



**HAL**  
open science

# Commande Prédicative pour le Véhicule Autonome

Iris Ballesteros Tolosana

► **To cite this version:**

Iris Ballesteros Tolosana. Commande Prédicative pour le Véhicule Autonome. Autre [cs.OH]. Université Paris Saclay (COMUE), 2018. Français. NNT : 2018SACL007 . tel-01863344

**HAL Id: tel-01863344**

**<https://theses.hal.science/tel-01863344>**

Submitted on 28 Aug 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Model Predictive Control for Autonomous Vehicles

Thèse de doctorat de l'Université Paris-Saclay  
préparée à CentraleSupélec

École doctorale n°580  
Sciences et Technologies de l'Information et de la Communication  
(STIC)  
Spécialité de doctorat: Automatique

Thèse présentée et soutenue à Gif-sur-Yvette, le 26 Janvier 2018, par

**Iris BALLESTEROS-TOLOSANA**

Composition du Jury :

M. Saïd MAMMAR Professeur, Univ. d'Evry Val-d'Essonne et ED STIC	Président
M. Arben CELA Professeur, Univ. Paris-Est, ESIEE Paris	Rapporteur
M. Antonios TZES Professeur, New York University Abu Dhabi	Rapporteur
M. François FAUVEL Ingénieur de Recherche, Renault	Examineur
M. Eduardo FERNANDEZ-CAMACHO Professeur, Universidad de Sevilla	Examineur
M. Sorin OLARU Professeur, CentraleSupélec-L2S	Directeur de thèse
M. Pedro RODRIGUEZ-AYERBE Professeur, CentraleSupélec-L2S	Co-Directeur de thèse
M. Renaud DEBORNE Ingénieur de Recherche, Renault	Invité
M. Guillermo PITA-GIL Ingénieur de Recherche, Renault	Invité



# Acknowledgments

Well, the moment has come...and here I am in front of my keyboard writing the Acknowledgments of my thesis. It always seemed so far, and still here we are. I cannot but think that this is the end of a great (not easy) period of my life. Of course, most of this greatness comes from the exceptional people I have met along the way : I will try to do my best in the following.

First and foremost, I would like to express my sincere gratitude to my supervisors Prof. Sorin Olaru and Prof. Pedro Rodriguez-Ayerbe for their continuous support and motivation. Your advice and guidance has helped me all along the time of this work and the writing of this thesis. Many thanks Sorin, for your funny emails, your amusing stories and the late-afternoon apples to keep me fed during the rush times. Thank you Pedro for your patience, your calm and the discussions in Spanish. It has been a great PLEASURE and I could not have imagined having better mentors for my PhD studies. I am also extremely grateful to Guillermo Pita-Gil, for conceiving this interesting and challenging PhD thesis topic and giving me opportunity to join his team in Renault to continue my professional career, working every day to make the Autonomous Car a reality. Renaud Deborne, for being my supervisor in Renault even if it was not the original plan, and always being there whenever I needed to help me solve so many surreal situations. For your great humor and your cool Terminator.

Besides my advisors, I would like to thank Prof. Arben Cela and Prof. Antonios Tzes for their time spent on the evaluation of this thesis, for their relevant questions and remarks that challenged me from various perspectives. Also, many thanks to the rest of my thesis committee : Prof. Saïd Mammari, M. François Fauvel and Prof. Eduardo Frendandez-Camacho for their insightful comments and encouragement.

Next, I would like to express my warmest gratitude to Prof. Jan Swevers for giving me the opportunity to visit their laboratory at the university of KU Leuven, for making me feel as one more proud member of the MECO research group and generously sharing their time and knowledge with me. Nothing to say about my colleagues from Office 01.17 : what happens in 01.17 stays in 01.17.

I am also extremely grateful to the TEMPO project, not only for providing the funding which allowed me to undertake this research, but also for giving me the opportunity to travel so much and attend many conferences, meeting so many interesting people and places on the way. The experience and exposure to such an environment provided was heavily enriching and rewarding. In addition, the members of the TEMPO project have greatly contributed to my personal and professional development during this time, we have founded a small TEMPO family with a friendship and collaboration network that will go beyond the end of the project. All the supervisors, Niels, Robin, Mohammad, Harsh, Sarmad, Goran, Bartolomeo, Bu-

lat, Gianni, Parisa, Deepak, Rajesh and Tahar : thank you for all the great moments around the world.

My PhD colleagues from Supelec, that amazing people with whom you share so much during this time and the ones that can understand and support you better. You are amazing people and I wish the best for you all. Specially the ones from *my* forever-alone office, for leaving the office all for myself but being there when needed. Finally, the Bling Ring team and my new indi-algerian b\*\*\*s family. The (many) sushi and the croissant au camembert. It would not have been the same without you people. Thank you very much for your support in the hardest moments and your humor, bringing such good and fun moments.

My PhD colleagues from Renault : Joan, Mihai, Ablamvi and Simon, for bringing such a nice atmosphere in the working place. The coffee pauses. Olivier Aubert for welcoming me in his team and his assistance and time spent with all the special procedures that a European Project brings into the picture. All the team members that always had nice words and a smile even in *livraison* weeks : Maud, Pascal, Yann, Pape, Thibault, Alain, Mathieu, Salam, Raphael, Sebastien, Virgile, Julien, Philippe, Emmanuel. The fusion team people, for being so welcoming when I joined the team towards the end of my thesis and creating a great atmosphere at work : Bruno, Bastien, Maxime, Omer, Nicole, Chrysanthi, Mathieu, Romain, Jorge and Salim.

Last not least, I would like to thank all my family and friends for their huge support. My little Enol, growing up so fast and becoming a marvelous kid. It is hard to be far away from you. But still, I feel close to all of you. You are all the most important people in my world and I would like to dedicate this thesis to you.

*Thank you very much to everyone.*

*Merci beaucoup à tous.*

*Muchas gracias a todos.*

# Résumé étendu de la thèse

## Introduction au sujet de thèse

L'industrie automobile évolue vers la conduite autonome, considérée comme l'avenir de cette industrie. Dans ce type de conduite, le conducteur devient un passager passif tandis que le véhicule prend en charge complètement de la tâche de conduite. Actuellement, plusieurs entreprises développent des prototypes qui sont capables de conduire de manière autonome, sans intervention humaine. Néanmoins, il y a encore un long chemin à parcourir entre la preuve de concept et la véritable extension de cette technologie pour le grand public.

Il y a plusieurs raisons qui justifient cela. La première est d'ordre économique car le prix actuel de la technologie la rend inabordable pour la plupart des consommateurs. La deuxième est l'acceptation de ces technologies par le conducteur ainsi que sa courbe d'apprentissage de tels systèmes, en vue d'une utilisation responsable. Enfin et non des moindres, les aspects législatifs et la nécessité de s'adapter à cette nouvelle problématique. Cela signifie qu'afin de permettre au grand public, d'accéder avec un coût abordable à ce type de technologie, la technique a besoin de gagner en maturité et d'être encadrée par un cadre juridique sûr.

Un processus d'évolution naturelle de l'industrie automobile est nécessaire, où les constructeurs envisagent que le développement des Systèmes Avancés d'Aide à la Conduite (ADAS) permettra à la technologie de devenir mature et de conduire progressivement au développement des véhicules autonomes. Lorsque nous évoquons les ADAS, nous considérons la prise en charge par le véhicule d'une certaine partie de la tâche de conduite, non pas à tous les niveaux de la décision mais sur un périmètre de scénarios définis et toujours sous supervision humaine. Ce terme couvre donc les systèmes qui, d'une certaine manière, visent à aider le conducteur dans sa tâche de conduite ou à en prendre le contrôle dans certaines situations, comme les manœuvres de parking, la conduite sur autoroutes ou sur routes « protégées ». Ces dispositifs améliorent la sécurité et le confort de l'expérience de conduite. Ce type de systèmes peut être considéré comme un premier pas vers la conduite semi-autonome et qui ouvrira alors la voie des véhicules entièrement automatisés.

## Architecture du système de contrôle - Schéma de hiérarchie

Il est fréquent de décrire la tâche de conduite pour un conducteur humain comme reposant sur trois niveaux de contrôle distincts, généralement connus sous le nom de stratégique (planification d'itinéraire), tactique (prise de décision, interaction avec le trafic environnant) et opérationnel (perception de l'environnement et contrôle).

La tâche principale des systèmes d'aide à la conduite (ADAS) est de remplacer, dans une certaine limite, le conducteur humain. Elle est alors divisée de manière analogue en plusieurs niveaux, pour reproduire le même type de logique. La Figure 1, illustre un schéma simplifié des niveaux tactique et opérationnel.

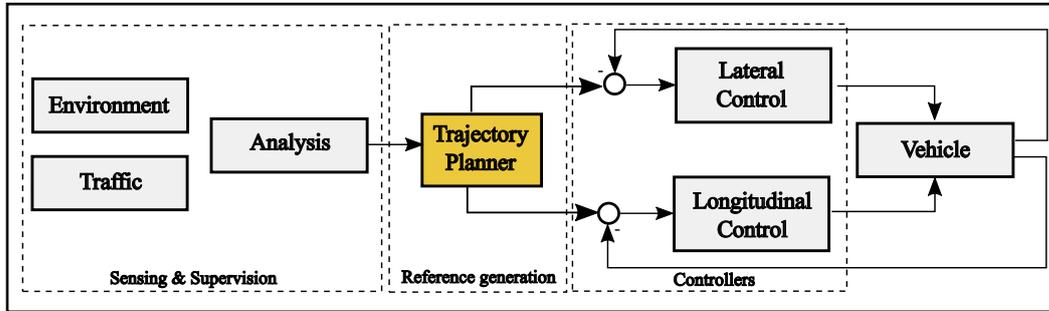


FIGURE 1 – Schéma de hiérarchie simplifié

Le niveau tactique est composé de l'analyse des éléments en interaction ainsi que de l'environnement du véhicule, ce qui inclut ses capteurs et ses algorithmes de fusion de données qui permettent le recueil de toutes ces données afin de mettre en place une analyse cohérente de l'environnement du véhicule. Dans un premier temps, notre travail dans cette s'est concentré sur le niveau opérationnel, où nous pouvons trouver les algorithmes de planification de trajectoire, qui fourniront les références appropriées au niveau inférieur, celui des contrôleurs. Dans notre cas, nous disposons d'une structure découplée pour le contrôle de la dynamique longitudinale et latérale du véhicule. Ce dernier thème représente une part importante de ce travail de thèse et décrit dans la partie I.

## Problèmes ouverts

La finalité des ADAS est d'être capable de conduire de façon autonome dans toutes les situations envisageables et ce avec un niveau de sécurité supérieur à celui d'un conducteur humain. Au-delà de des progrès technologiques en matière de développement, il est à noter que la conduite autonome dans l'exhaustivité des situations de trafic exige encore beaucoup d'avancées en matière de détection et de capacités de contrôle par rapport à ce qui est disponible dans l'état de l'art. Cela est en partie dû au fait que des conditions de conduite ne sont jamais identiques et donc c'est un point critique pour permettre d'assurer un comportement correct malgré les variations de paramètres du système ou la présence d'incertitudes. Lorsque l'on se focalise sur le développement des systèmes de contrôle pour les ADAS, les principaux axes de recherche visent à identifier des stratégies de contrôle robustes qui garantissent le niveau de performance attendu. De plus, des applications où la vie humaine est impliquée doit trouver les moyens d'assurer la sécurité du système pour les passagers du véhicule, ainsi que pour les éléments environnants. Dans ce contexte, les straté-

gies de contrôle qui assurent le traitement des différentes contraintes du système dès l'étape de conception un sujet clé.

En outre, une fois que les blocs de base fonctionnels de contrôle ont atteint maturité, fiabilité et performances satisfaisantes, des manœuvres de plus en plus complexes seront progressivement développées, comme par exemple le changement de voie et dépassement de véhicule. Ce type d'applications nécessite une maîtrise des tâches de niveau inférieur, comme le contrôle de la direction du véhicule. Après cela, les efforts de recherche seront orientés vers le développement d'autres types d'algorithmes nécessaires à la mise en œuvre de ces tâches complexes. C'est le cas des stratégies de planification de trajectoire, qui ont une longue histoire de recherche dans le domaine de la robotique. Pourtant, il reste un long chemin à parcourir pour les applications automobiles, où les environnements non contrôlés et très dynamiques en complexifient la conception.

## Structure de la thèse

Le manuscrit est divisé selon les deux principaux sujets d'application qui ont été étudiés lors de ce travail de thèse industrielle. La première partie met l'accent sur le contrôle contraintes de la dynamique latérale du véhicule. Elle est composée de trois chapitres, dont le contenu est brièvement présenté plus loin. Enfin, la seconde partie du document est quant à elle composée de deux chapitres, ayant pour thème la planification de trajectoire pour les manœuvres de dépassement dans les autoroutes.

### **PARTIE I : CONTRÔLE SOUS CONTRAINTES DE LA DYNAMIQUE LATÉRALE DU VÉHICULE**

#### **Chapitre 2 : Contexte théorique pour le contrôle sûr contraintes**

Ce chapitre vise à présenter un ensemble d'outils qui seront utilisés pour la conception du contrôle sous contraintes dans la première partie du manuscrit. L'idée est d'offrir au lecteur un rapide aperçu des concepts théoriques qui sont nécessaires pour comprendre les travaux présentés dans les deux chapitres suivants.

#### **Contrôle sous contraintes de la dynamique latérale du véhicule**

Ce chapitre présente deux approches génériques pour la conception d'un contrôleur pour le système Auto-Steering par suivi de cible, où l'incertitude liée à la variation de la vitesse est prise en considération de manière explicite. Cette variation de vitesse conduit à un modèle incertain, qui sera décrit par une dynamique couverte dans le cadre linéaire par une incertitude polytopique. Grâce à la mesure en ligne du paramètre, la dynamique du système est calculée au niveau de chaque itération de temps pour résoudre le problème d'optimisation du contrôleur par commande prédictive du véhicule. Des fonctions de Lyapunov dépendantes des paramètres et la théorie des ensembles invariants sont utilisés pour assurer la stabilité et la faisabilité d'une telle commande. Après cette étape de base de conception, le contrôle basé

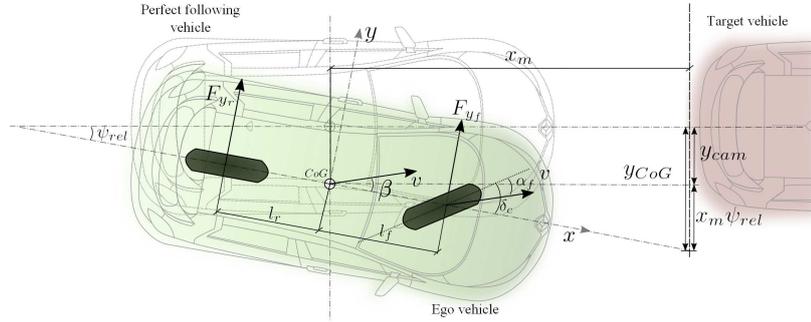


FIGURE 2 – Autosteer par poursuite des cible

sur l'interpolation, dont le principe est d'utiliser un contrôle de l'action construit comme une interpolation entre deux valeurs extrêmes précalculées, est étudié. A chaque instant de temps, deux problèmes de programmation linéaire sont résolus, ce qui conduit à un bon compromis entre performances et coût de calcul.

### Lane Centering Assistance System

L'attention de ce chapitre se concentre sur l'un des systèmes de contrôle de la dynamique latérale du véhicule connue sous le nom de système d'Assistance de Centrage de Voie (Lane Centering Assistance system ou encore LCA), qui est définie par un système Linéaire à Paramètres Variants (LPV) composé à partir des éléments de la dynamique du système le plus pertinents et de la courbure de la route, modélisée comme une perturbation additive à paramètre variant et borné.

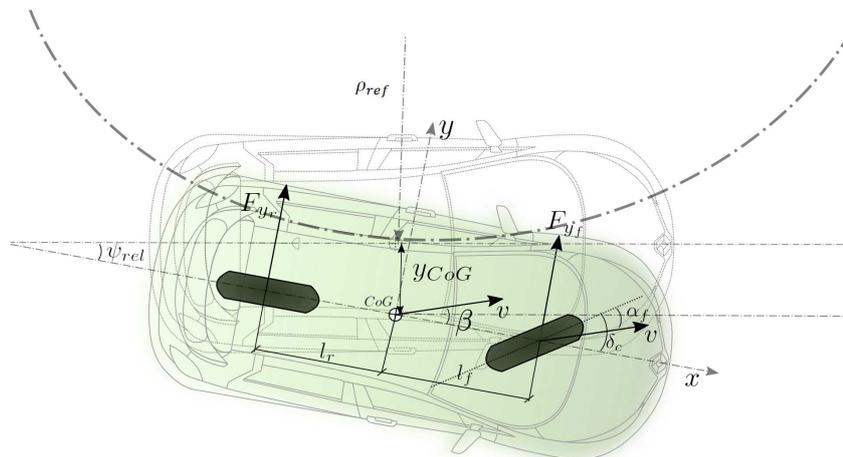


FIGURE 3 – Modèle bicyclette par rapport au centre de voie

D'un point de vue théorique, la théorie de l'invariance positive robuste est exploitée pour réaliser l'analyse des effets qu'une perturbation additive bornée et variable

a sur un contrôleur linéaire à paramètre variant, utilisés pour assurer la stabilité d'une stratégie de contrôle prédictif. Après cela, la vitesse du véhicule et la courbure de la route sont pris en compte dès l'étape de conception, afin de calculer un observateur-contrôleur par retour d'état qui garantit la performance du contrôle en présence de ces changements sur les conditions de conduite et les limites du système, qui sont traduits par des contraintes lors de la conception du contrôleur. En outre, la satisfaction des contraintes et la maximisation du domaine d'attraction sont considérés, afin de fournir une région de fonctionnement certifiée. Comme dernière partie de l'étude et afin de réduire le conservatisme introduit par de grandes variations des paramètres, une conception avec l'utilisation de multiples fonctions Lyapunov discontinues et la prise en compte de l'accélération maximale est proposée. La stabilité de la boucle fermée du système LPV par commutation est prouvée par l'exploitation des conditions de temps d'arrêt héritées de la mise en œuvre d'hystérésis.

## **PARTIE II : PLANIFICATION DES TRAJECTOIRE POUR CHANGEMENT DE VOIE ET DÉPASSEMENT SUR AUTOROUTES**

### **Chapitre 5 : Contexte théorique pour la planification de trajectoire**

En suivant l'organisation de la première partie, ce premier chapitre offre un aperçu des concepts théoriques utilisés à des fins de planification de trajectoire. Un état de l'état de l'art des méthodes numériques qui peuvent être utilisées pour résoudre des problèmes de contrôle optimal et des outils théoriques des arrangements des hyperplans sont introduites, sont présentés afin de permettre la compréhension du chapitre suivant.

### **Chapitre 6 : Planification de trajectoire sans collision sur autoroutes**

Le dernier chapitre du manuscrit aborde le problème du changement de voie du véhicule et de dépassement sur les routes dans le contexte d'une conduite assistée. Afin d'effectuer de telles manœuvres, il est fondamental de calculer des trajectoires

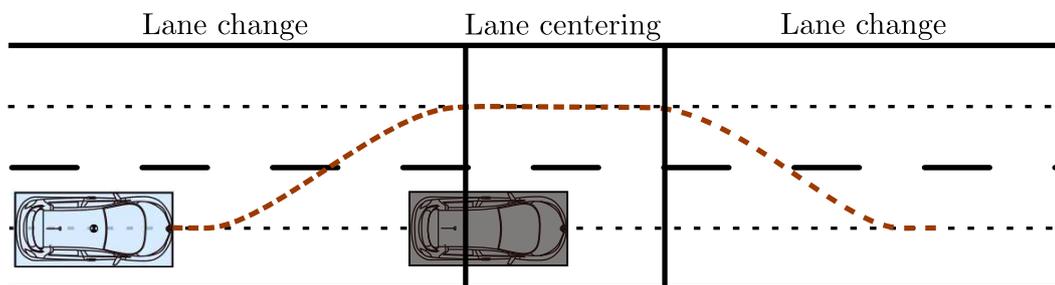


FIGURE 4 – Manoeuvre de changement de voie et dépassement

appropriées et confortables pour le conducteur, qui prennent en compte les limites du véhicule ainsi que les restrictions de sécurité. Au-delà des limitations internes et

dans un environnement aussi dynamique, l'élément essentiel à prendre en compte dans la conception est l'interaction avec les véhicules du trafic dans le voisinage immédiat. Ces véhicules, qui partagent l'environnement routier avec notre véhicule, vont définir une région faisable non-convexe qui est décrite dans le présent travail, en termes d'arrangements des hyperplans, prévoyant une formulation par entiers mixtes des contraintes d'anticollision. Les travaux antérieurs sur la réduction des variables

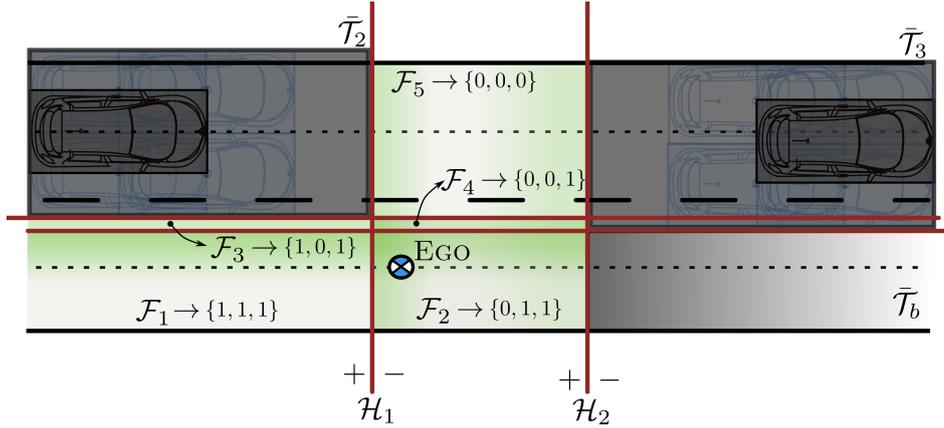


FIGURE 5 – Région faisable non-convexe - Hyperplan arrangement

binaires et les techniques de fusion de cellules sont considérés et appliqués à une énumération exhaustive des scénarios de dépassements possibles. Enfin, un problème de contrôle optimal est formulé et traduit en un problème de programmation non linéaire d'optimisation de dimension finie, résolue via une méthode d'approche de type *direct multiple-shooting*.

## Contributions de la thèse

Les Systèmes d'assistance à la conduite (ADAS) structurent l'axe principal de ce travail de thèse industrielle. Ce type de systèmes est considéré comme le premier pas vers le développement de la conduite autonome, visant à améliorer progressivement les connaissances technologiques et l'état de l'art des différentes tâches de conduite. Cette évolution permettra de proposer des stratégies de contrôle qui seront réalisables et applicables sur des véhicules de série et accessibles pour tout client. Par ailleurs, la synthèse d'algorithmes efficaces pour la mise en place des choix stratégiques de conduite représente un compromis entre les performances du système et la sécurité des passagers et va jouer un rôle fondamental pour l'avenir des solutions adoptées.

Dans la première partie de ce travail, nous avons mis l'accent dans l'une des principales tâches de conduite, qui est le contrôle de la direction du véhicule (par le volant) pour assurer le suivi d'un véhicule (Autosteer par poursuite de cible) ou le centre de la voie actuel (Lane Centering Assistance). Un accent particulier a été donné aux implications liées aux variations des paramètres de la dynamique du

système, issus de la variation de la vitesse. Nous avons adopté des principes de la commande LPV pour cette application automobile où la dynamique du système est affectée par de larges variations de la vitesse du véhicule.

En outre, la robustesse de la conception des contrôles à l'égard de l'impact de la courbure de la route, modélisée comme une perturbation additive variable et bornée a été étudiée à l'aide de la théorie des ensembles invariants positifs robustes, pour l'analyse et la conception du contrôleur LPV.

Il a été démontré que ces contrôleurs peuvent efficacement s'appuyer sur la synthèse de lois de commande LPV, fournissant des garanties de faisabilité et cela même en présence de grandes variations de vitesse au moyen d'une stratégie de contrôle commuté ou en prenant en compte les capacités d'accélération maximale du véhicule.

Cette question est motivée par la nécessité de disposer du plus grand domaine d'attraction en présence d'incertitudes, où la satisfaction des contraintes du système est certifiée dès l'étape de conception. De plus, les commandes prédictive (MPC) et par contrôle à base d'interpolation (IBC) sont étudiées afin d'assurer le respect des contraintes et l'élargissement du domaine d'attraction, fournis par la stratégie à l'horizon glissant. La principale contribution de la première partie est d'ordre méthodologique et prouve qu'une conception intégrée peut faire face à une large gamme de variations de paramètres et de perturbations additives et peut garantir la certification de la sécurité et du respect des contraintes sur le fonctionnement en boucle fermée. En particulier, le IBC est indiqué pour une formulation de la loi de commande très compacte basée sur l'optimisation.

En tant que perspectives de ce travail, la modélisation et l'intégration des études de conception du contrôle en présence de contraintes d'entrées variables devrait certainement être considérées. Plus particulièrement, la variation des limites de l'angle de direction à l'égard de la vitesse du véhicule représente une direction à privilégier. En outre, les dynamiques longitudinale et latérale couplées doivent être considérées, bien qu'elles fournissent des solutions plus complexes basées sur l'optimisation non linéaire. Ce type de modèle est capable de fournir des solutions pour le contrôle de manœuvres plus agressives, comme c'est le cas pour les manœuvres d'urgence.

Dans la deuxième partie de cette thèse, l'attention a été centrée sur les algorithmes de planification de trajectoire, où l'étude et l'analyse de l'environnement dynamique dans lequel la manœuvre est effectuée, ainsi que les limites physiques du véhicule, sont étudiées ensembles pour obtenir des trajectoires de changement de voie qui permettent d'effectuer ce type de manœuvre avec les garanties de sécurité, dont la certification de non collision. Comme étude spécifique, l'attention est focalisée sur la des méthodes de génération de trajectoire basée sur l'optimisation, où la réduction du niveau du jerk (variation d'accélération ou encore secousses) de la manœuvre est le principal objectif, et ce dans le but d'assurer le confort des passagers tout au long de l'exécution de la manœuvre. La contribution de cette deuxième partie est la proposition d'une méthode de planification de trajectoire basée sur l'optimisation qui fournit des trajectoires qui s'assurent le respect des contraintes de sécurité et de confort, tout en balayant les scénarii possibles en termes d'arrangements d'hyperplans autour des obstacles environnants. En outre, la réduction du

nombre de variables binaires nécessaires et les techniques de fusion de cellules sont considérées et appliquées à l'énumération exhaustive des scénarii de dépassement possibles menant à une *minimal* avec un impact positif sur la complexité du calcul qui est atténué de manière considérable.

## Conclusion

Pour résumer, nous assistons à l'évolution rapide dans le domaine des ADAS au cours des dernières années. En raison de plusieurs facteurs, il a fallu un certain temps pour que ces systèmes s'imposent depuis les premières briques théoriques jusqu'à la production industrielle de masse. Depuis quelques années, une demande croissante sur le marché se fait sentir également. De plus, il est envisageable qu'au cours des prochaines années, cette tendance soit encore plus marquée en raison de nouvelles réglementations (évolutions de l'Euro NCAP), où la note maximale des cinq étoiles sera uniquement accordée aux véhicules équipés d'ADAS. En outre, les conducteurs eux-mêmes commencent à apprécier les avantages de ce type de systèmes, et il devient de plus en plus disposé à payer le prix pour avoir accès à ces technologies.

La conduite autonome partielle sera une réalité dans les prochaines générations de véhicules particuliers. Cependant, nous devons garder à l'esprit qu'une réglementation appropriée doit être mise en place et de nombreuses questions difficiles d'homologation et de responsabilité doivent être traitées. Enfin, d'autres recherches sur l'évaluation des situations routières et des algorithmes pour la prise de décisions avec un niveau au moins équivalent à celui de la cognition humaine sont absolument nécessaires. La recherche fondamentale dans les années à venir va donc jouer un rôle essentiel pour que les concepts deviennent réalité.

# Contents

<b>List of figures</b>	<b>xvii</b>
<b>List of tables</b>	<b>xix</b>
<b>List of publications</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Advanced Driving Assistance Systems Overview . . . . .	2
1.2.1 System Architecture - Control hierarchy scheme . . . . .	3
1.2.2 Vehicle Instrumentation . . . . .	4
1.2.3 Vehicle dynamical modeling discussion . . . . .	5
1.3 Open problems . . . . .	9
1.4 Thesis structure . . . . .	10
1.5 Thesis contributions . . . . .	12
<b>PART I VEHICLE LATERAL DYNAMICS CONSTRAINED CONTROL</b>	<b>15</b>
<b>2 Theoretical background for constrained control</b>	<b>17</b>
2.1 Linear Parameter Varying systems . . . . .	18
2.1.1 General considerations on LPV models . . . . .	18
2.1.2 Polytopic representation of LPV systems . . . . .	19
2.2 Set invariance theory for input-free discrete-time LPV systems . . . . .	20
2.2.1 Polyhedral sets . . . . .	20
2.2.2 Ellipsoidal sets . . . . .	28
2.3 Control design for LPV systems . . . . .	32
2.3.1 LPV Control Design . . . . .	33
2.3.2 Model Predictive Control . . . . .	39
2.3.3 Interpolation based control . . . . .	44
2.4 Conclusion . . . . .	48
<b>3 Lateral Dynamics Control</b>	<b>49</b>
3.1 Problem formulation . . . . .	49
3.1.1 Autosteer constraints . . . . .	53
3.1.2 Note on $y_{cam}$ and $x_m$ . . . . .	53
3.2 Constrained Control for Lateral Dynamics . . . . .	54
3.2.1 Model Predictive Control for Lateral Dynamics . . . . .	54
3.2.2 Interpolation-Based Control for Lateral Dynamics . . . . .	56
3.3 Simulation Results: MPC vs IBC . . . . .	58
3.3.1 MPC feasibility analysis . . . . .	59

3.3.2	IBC feasibility analysis . . . . .	59
3.4	Conclusion . . . . .	63
<b>4</b>	<b>Lane Centering Assistance System</b>	<b>65</b>
4.1	Dynamical model for LCA . . . . .	66
4.1.1	Additive disturbances modeling . . . . .	67
4.1.2	Actuator dynamics . . . . .	70
4.1.3	Full dynamical model . . . . .	71
4.1.4	LCA system constraints . . . . .	73
4.2	Control design for LCA system . . . . .	73
4.2.1	LPV design and robustness analysis via mRPI tools . . . . .	74
4.2.2	LPV controller redesign for built-in robustness . . . . .	76
4.2.3	Simultaneous observer-controller design . . . . .	80
4.2.4	Advanced tools I: Switched LPV design . . . . .	84
4.2.5	Advanced tools II: Acceleration considerations . . . . .	87
4.2.6	MPC for LCA system control . . . . .	92
4.3	Conclusion . . . . .	95
 <b>PART II TRAJECTORY PLANNING FOR LANE CHANGE AND OVER-</b>		
<b>TAKING ON HIGHWAYS</b>		<b>97</b>
<b>5</b>	<b>Theoretical background for trajectory planning</b>	<b>99</b>
5.1	Optimal Control . . . . .	100
5.1.1	Basic Definitions . . . . .	100
5.1.2	Optimal Control Problem formulation . . . . .	101
5.1.3	Numerical methods for solving Optimal Control Problems . . . . .	101
5.1.4	Numerical Integration - Runge-Kutta scheme . . . . .	108
5.2	Flatness . . . . .	110
5.3	Hyperplane Arrangements . . . . .	111
5.3.1	Basic definitions . . . . .	111
5.3.2	Non-convex non-connected regions characterization . . . . .	113
5.3.3	Cell Merging . . . . .	115
5.3.4	Mixed Integer representation . . . . .	117
5.4	Conclusion . . . . .	122
<b>6</b>	<b>Collision-free trajectory planning on highways</b>	<b>125</b>
6.1	Problem formulation . . . . .	126
6.1.1	Trajectory planning strategies review . . . . .	127
6.1.2	Alternative obstacle avoidance formulations . . . . .	128
6.2	Exhaustive scenarii description . . . . .	129
6.3	Methodology . . . . .	133
6.3.1	Ego vehicle description . . . . .	133
6.3.2	Mathematical modeling of the highway scenarii: feasible non-convex region characterization . . . . .	134

---

6.3.3	Optimal Control Problem formulation for collision-free trajectory planning . . . . .	140
6.3.4	Trajectory reference transformation step . . . . .	146
6.4	Problem computation reduction . . . . .	150
6.5	Implementation remarks . . . . .	153
6.5.1	Receding horizon strategy . . . . .	153
6.5.2	Decision making algorithm . . . . .	153
6.6	Conclusion . . . . .	156
	<b>CONCLUSION AND FURTHER PERSPECTIVES</b>	<b>159</b>
	<b>Bibliography</b>	<b>163</b>



# List of Figures

1	Schéma de hiérarchie simplifié . . . . .	iv
2	Autosteer par poursuite des cible . . . . .	vi
3	Modèle bicyclette par rapport au centre de voie . . . . .	vi
4	Manoeuvre de changement de voie et dépassement . . . . .	vii
5	Région faisable non-convexe - Hyperplan arrangement . . . . .	viii
1.1	Simplified schema logic architecture . . . . .	3
1.2	Vehicle instrumentation. <i>Source: www.group.renault.com</i> . . . . .	4
1.3	2-DoF Bicycle model . . . . .	6
3.1	Auto-steering for target tracking . . . . .	50
3.2	Note on initial conditions . . . . .	55
3.3	$\Omega_P$ and $\Omega_{P(\nu)}$ comparison . . . . .	56
3.4	$\mathcal{C}_3$ and state admissible space $X$ . . . . .	57
3.5	$\Omega_{P(\nu)}$ , $\mathcal{C}_3$ and state admissible space $X$ . . . . .	57
3.6	System states trajectory . . . . .	61
3.7	System outputs trajectory . . . . .	61
3.8	System input trajectory . . . . .	62
3.9	IBC interpolation factor . . . . .	62
4.1	Bicycle model referenced to the center line . . . . .	66
4.2	Polytopic approximation of the curve $v_x \rho_m = v_x^2 \rho_m$ . . . . .	69
4.3	Polytopic representation from Table 4.1 data (comfort) . . . . .	70
4.4	Domain of attraction controller $\mathcal{E}(P_1)$ and computed mRPI set . . . . .	75
4.5	Admissible speed range in curve for the parameter-varying controller for fixed $\rho$ values . . . . .	75
4.6	Computed mRPI set $\mathcal{E}(P_i)$ for fixed $\rho_i$ values (Fig. 4.5) . . . . .	76
4.7	LPV Robust control design with RPI maximization domain of attraction . . . . .	77
4.8	mRPI ellipsoidal approximation for unmodeled additive disturbances . . . . .	78
4.9	Speed and Curvature Profile for LPV and Robust LPV designs com- parison . . . . .	78
4.10	System input trajectory for LPV and Robust LPV designs comparison . . . . .	78
4.11	System states trajectory for LPV and Robust LPV designs comparison . . . . .	79
4.12	System states trajectory and ellipsoidal invariant sets projections for LPV and Robust LPV designs comparison . . . . .	79
4.13	Robust domain of attraction for simultaneous robust observer-controller design, $\mathcal{E}(P_i)$ . . . . .	82
4.14	Speed and Curvature Profile . . . . .	82
4.15	System input trajectory Robust and Observer-based LPV designs comparison . . . . .	82

4.16	System states trajectory Robust and Observer-based LPV designs comparison . . . . .	83
4.17	System states trajectory and ellipsoidal invariant sets projections for Robust and Observer-based LPV designs comparison . . . . .	84
4.18	Hysteresis switching strategy . . . . .	86
4.19	Robust domain of attraction for parameter-dependent switched LPV design, $\mathcal{E}_{\sigma_i}(P_i)$ . . . . .	86
4.20	Speed and Curvature Profile for switched LPV design . . . . .	86
4.21	System input trajectory for switched LPV design . . . . .	87
4.22	System input trajectory for switched LPV design . . . . .	87
4.23	System states trajectory for switched LPV design . . . . .	88
4.24	System states trajectory and ellipsoidal invariant sets projections for switched LPV design . . . . .	88
4.25	Robust domain of attraction for parameter-dependent LPV design, $\mathcal{E}(P_i)$ with maximal acceleration considerations . . . . .	92
4.26	System input trajectory for LPV design with maximal acceleration considerations . . . . .	93
4.27	System states trajectory for LPV design with maximal acceleration considerations . . . . .	93
4.28	System states trajectory and ellipsoidal invariant sets projections for LPV design with maximal acceleration considerations . . . . .	94
5.1	Optimal control problem . . . . .	102
5.2	Optimal Control methods classification . . . . .	102
5.3	Direct single-shooting, NLP variables . . . . .	105
5.4	Multiple-shooting, NLP variables . . . . .	108
5.5	Hyperplane Arrangements . . . . .	113
5.6	Feasible and Unfeasible cells definition . . . . .	114
5.7	Karnaugh map representation for hyperplane arrangement in Fig. 5.6(b) . . . . .	116
5.8	Merged Cells . . . . .	117
5.9	Feasible cells binary enumeration . . . . .	121
5.10	Merged Cells . . . . .	122
6.1	Lane change and Overtaking maneuver . . . . .	126
6.2	Target definition schema . . . . .	129
6.3	Points detection and tracking at time $t_k$ . . . . .	132
6.4	$\bar{\mathcal{T}}_j$ definition . . . . .	135
6.5	Rotated target bounding box definition . . . . .	138
6.6	Hyperplane arrangement scenario 2.3 . . . . .	139
6.7	Non-convex feasible region definition: $\mathcal{C}_1 \cup \mathcal{C}_2$ . . . . .	140
6.8	Trajectory planning OCP constraint sources . . . . .	142
6.9	Cost Function value and Total time of the maneuver . . . . .	146
6.10	Computation time for maneuver generation . . . . .	146

---

6.11	Lane change generated trajectories in the absence of other vehicles (green) and collection of all feasible trajectories in the presence of a slower obstacle of type T1. . . . .	147
6.12	Scenario type 2.3 . . . . .	148
6.13	Lane change generated trajectory for Scenario 2.3 . . . . .	149
6.14	Flatness-based generated trajectory with 5-th order polynomial parameterization in between <i>waypoints</i> . . . . .	150
6.15	Flatness-based generated trajectory with linear parameterization in between <i>waypoints</i> , $\eta^{ref}(t)$ . . . . .	151
6.16	Flatness-based generated inputs with linear parameterization in between <i>waypoints</i> , $u^{ref}(t)$ . . . . .	152
6.17	$\lambda_k$ combinatorial search tree . . . . .	152
6.18	$\lambda_k$ simplified combinatorial search tree . . . . .	152
6.19	Hyperplane arrangement example for a curved road . . . . .	160



# List of Tables

1.1	Bicycle model states description . . . . .	9
1.2	Vehicle parameters definition . . . . .	9
3.1	Trajectory Constraints Definition for Autosteer System . . . . .	53
3.2	Interpolation based control tests . . . . .	60
4.1	Radius-speed values for design on highways [Vertet 2006] . . . . .	69
4.2	Trajectory Constraints Definition for LCA System . . . . .	73
6.1	Target definition . . . . .	129
6.2	Scenario definition . . . . .	131
6.3	Scenario feasible and unfeasible regions . . . . .	137
6.4	Numerical values for trajectory states and input constraints . . . . .	142



# List of publications

## 2 Patent applications

1. I. Ballesteros-Tolosana, Renaud Deborne, S. Oлару, J.Davins-Valldaura, M.Geoffriault. Feedforward enhancements for Lane Centering Assistance System.
2. I. Ballesteros-Tolosana, Renaud Deborne, S. Oлару, J.Davins-Valldaura. Model Predictive Control for Lane Centering Assistance System.

## Accepted Conference Papers

1. I. Ballesteros-Tolosana, S. Oлару, P. Rodriguez-Ayerbe, G. Pita-Gil, Renaud Deborne. Comparison of optimization-based strategies for constrained control of Auto-Steering Systems, European Control Conference, 2016.
2. I. Ballesteros-Tolosana, S. Oлару, P. Rodriguez-Ayerbe, G. Pita-Gil, Renaud Deborne. On the impact of additive disturbances on Auto Steering Systems. IFAC WC 2017.
3. I. Ballesteros-Tolosana, P. Rodriguez-Ayerbe, S. Oлару, Renaud Deborne, G. Pita-Gil. Lane Centering Assistance system design for large speed variation and curved roads. Conference in Control and Technology Applications, 2017.
4. I. Ballesteros-Tolosana, S. Oлару, P. Rodriguez-Ayerbe, G. Pita-Gil, Renaud Deborne. Optimization-based trajectory planning for overtaking in highways. Conference on Decisions and Control, 2017

## International conferences without proceedings

1. I. Ballesteros Tolosana, N. van Duijkeren, J. Swevers, S. Oлару. Optimization-based trajectory generation for overtaking on highways. Benelux Meeting on Systems and Control 2016.
2. I. Ballesteros-Tolosana, S. Oлару, P. Rodriguez-Ayerbe. Interpolation-based control for Auto-Steering systems, European Conference on Computational Optimization, 2016.



# Introduction

---

## Contents

<b>1.1</b>	<b>Motivation</b>	<b>1</b>
<b>1.2</b>	<b>Advanced Driving Assistance Systems Overview</b>	<b>2</b>
1.2.1	System Architecture - Control hierarchy scheme	3
1.2.2	Vehicle Instrumentation	4
1.2.3	Vehicle dynamical modeling discussion	5
1.2.3.1	Vehicle lateral dynamics modelization	6
<b>1.3</b>	<b>Open problems</b>	<b>9</b>
<b>1.4</b>	<b>Thesis structure</b>	<b>10</b>
<b>1.5</b>	<b>Thesis contributions</b>	<b>12</b>

---

## 1.1 Motivation

The automotive industry is evolving towards the *autonomous driving* concept, considered to be the future of this business. In this kind of driving, the human plays the role of a passive passenger while the vehicle is completely in charge of the driving task. It can be seen nowadays that several companies are developing prototypes [Behringer 2004] that are capable of autonomously driving the vehicle without human intervention. Nevertheless, there is a long way to go between the *proof-of-concept* and the actual spreading of the technology to the general public.

There are several reasons that can justify this time gap, to start with, the current price of the technology, which is not affordable for most of the vehicle consumers. Then, the drivers learning curve in view of a responsible use and acceptance of such technologies can be considered as an impediment. Not the least, the legal aspects need to adapt to the change of framework. This means that in order to fully offer this kind of technology at an affordable price to the general public, the technique needs to gain in maturity and to be comforted by a secure legal framework.

A natural evolution process is needed at the vehicle industry, where car manufacturers consider that the development of the Advanced Driving Assistance Systems (ADAS) will allow the technology to become mature and progressively lead to Autonomous Driving vehicles. When speaking of the ADAS systems, one can expect the vehicle to be in charge of certain driving tasks, not at all levels of decision but at certain delimited scenarios and always under human supervision. This term covers

the technological systems that, briefly speaking, aim to assist the driver or take over control of the driving task in certain situations, like parking lots, highways or protected roads, offering an improved safety and comfort experience. This kind of systems can be considered as a first generation of assisted or semi-autonomous driving, that will pave the way to fully automated vehicles [der Automobilindustrie eV 2015].

The ultimate ADAS of the future should be capable of automated driving in all conceivable situations at a safety level superior to that of a human driver. This is considered especially important, as the compensation for human error, accounting for 90 per cent of all accidents [Treat 1979],[Trucks 2013], is a prerequisite for accident-free traffic. In addition, the driving tasks can be broken up into basic functional components or systems that can be technically implemented and developed up to a certified level of maturity.

## 1.2 Advanced Driving Assistance Systems Overview

Despite the ADAS technology is a relatively fresh research field, it can be seen that there has been already an important amount of research dedicated to this kind of systems in the last three decades [Bengler 2014], [Winner 2015].

Initial realizations of driving assistance were mostly based on passive components whose main idea was to warn the driver or briefly correct vehicle's trajectory. Here we can name the Blind Spot Warning (BSW), Lane Departure Warning (LDW) which evolved lately to security mechanisms as Lane Keeping Assistance (LKA) [Enache 2008] systems. In the subsequent, more complex systems have been developed, where the main objective is focused on the complete automation of different driving tasks to improve the drivers comfort and safety. Some of these systems are the Parking Assistance, Automatic Emergency Braking (AEB), Longitudinal dynamics control, Lateral dynamics control or Lane Change Assistance systems.

First versions of Parking Assistance systems have been mostly focused on the steering wheel control, while the driver was still in charge of controlling the longitudinal movement by accelerating or braking as needed. More recent developments enable automotive components to be completely capable of executing the maneuver, without any human intervention.

In addition, a huge improvement on the vehicle's safety has been provided by the development of the Automatic Emergency Braking [Kusano 2012]. In a nutshell, this class of systems are in charge of braking the vehicle if the driver fails to react whenever facing an imminent longitudinal collision. More evolved systems will include active obstacle avoidance protection systems, like pedestrians, bicycles or other vehicles, that may arise provoking hazardous situations both for the vehicle or the environment [Coelingh 2010], [Dang 2012], indispensable to truly decrease traffic-related accidents. This line of developments came with a parallel developments in modeling in the human behavior and the interaction in an environment as the urban traffic.

Longitudinal dynamics control is a mature field [Vahidi 2003], where the main

objective is to regulate vehicle speed and longitudinal distance with the surrounding vehicles. This kind of systems, commonly known as Adaptive Cruise Control (ACC) systems [Winner 2014], are already offered by many car manufacturers, and seem to have been successfully accepted by the customers thus offering an important argument for the social acceptance of autonomous or semi-autonomous driving initiatives.

Then, lateral dynamics control has been also subject of several studies for a long time [Falcone 2007], [Rajamani 2006], [Palladino 2006]. Nevertheless, drivers acceptance of these kind of assistance systems seems to be more troublesome and needs a longer period of time to be completely settled on the market [Brookhuis 2001]. In principle, the main purpose of these systems is to control the steering wheel of the vehicle, considering different driving objectives. Here we recall the Auto-steer system, whose main purpose is to follow a vehicle that is in front of the controlled one at low speed. Then, the Lane Centering Assistance (LCA), the component in charge of following the center of the lane when driving in the highway, considering a broad range of vehicle speed.

Lastly, we have the Lane Change assistance, which has been already studied in several research work but in practice still remains a challenge for the existing real-life application, mostly due to the inherent complexity of the interaction with the surrounding dynamical environment [Schubert 2010].

### 1.2.1 System Architecture - Control hierarchy scheme

Generally, it is common to model the human driver's task as three levels of behavior and control, usually known as strategical (itinerary planning), tactical (decision making, traffic interaction) and operational (control and environment perception) levels respectively [Michon 1985].

The Advanced Driving Assistance Systems (ADAS) main task is to take the role of the human driver up to a certain point, so the general architecture structure is usually divided in an analogous way in several layers or levels, replicating the same kind of logic. In Figure 1.1 a simplified schema of the tactical and operational layers is shown.

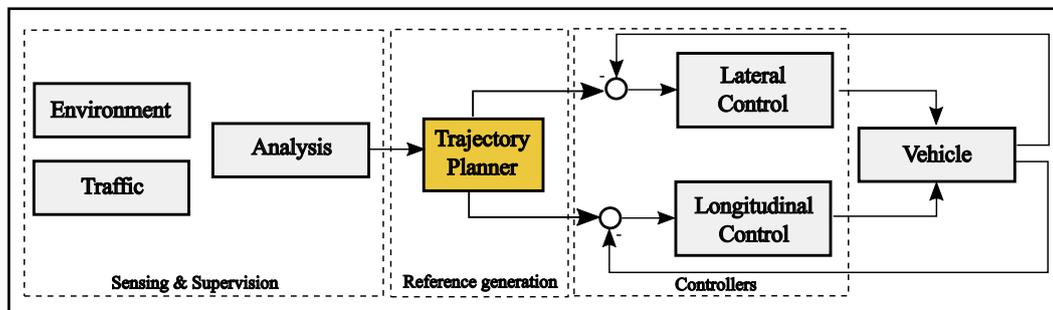


Figure 1.1 – Simplified schema logic architecture

The tactical level is comprised of the elements that interact and analyze the surroundings of the vehicle, which includes all the vehicle instrumentation (Section 1.2.2) and sensor data fusion algorithms that will collect all the data in order to set up a coherent vehicle environment analysis. As a first positioning, our focus in this thesis has been given to the operational layer, where we can find the trajectory planning algorithms (Part II), that will provide the appropriate references to the lower level controllers, in this case, having a decoupled structure for longitudinal and lateral dynamics control, this last topic representing also an important part of the present thesis as described in Part I.

### 1.2.2 Vehicle Instrumentation

Having an accurate knowledge of the framework in which the assisted vehicle is located is a fundamental problem of the ADAS systems: an automated vehicle cannot act properly without a deep understanding of the environment in which is driving. It is straightforward to notice that the surrounding conditions cannot be known in advance, as traffic interactions are highly dynamical and unpredictable, all emphasized by the fact that human-driven and semi-automated vehicles are *doomed* to co-exist during a certain, yet undefined, period of time.

All these factors generate the need to equip the vehicles with a series of on-board sensors (Figure 1.2) that allow to measure the relevant information of the environment and the vehicle. Generally speaking, we can distinguish between two kinds of sensors, denoted as exteroceptive or proprioceptive.

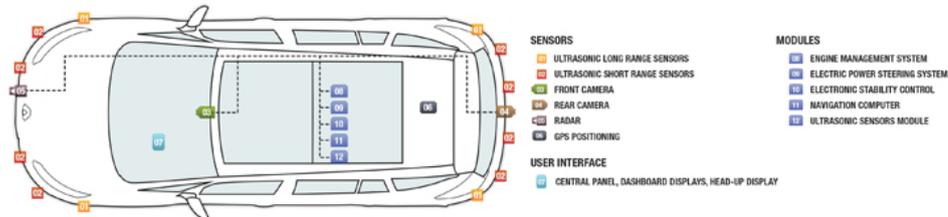


Figure 1.2 – Vehicle instrumentation. *Source: www.group.renault.com*

On one hand, the first type of sensors is in charge of providing environment measurements as well as the relative state of the vehicle. The main exteroceptive sensors for the auto-steering systems include a *camera* and one or several *radars*, which are in charge of providing the position of the vehicle with respect to the lines of the road as well as the movement and location of the surrounding elements. These technologies currently dominate the ADAS sector, having complementary capabilities, and the omission of one technology in favor of the other is not to be expected in the near future.

On the other hand, the second type of sensors has the objective of yielding own vehicle information. Here we include the measurement of the steering angle, by

means of a relative or absolute encoder and a gyro for the vehicle yaw rate. For the rest of LCA non-measured states, an observer that estimates their values is needed.

In addition to this, it has been shown in Figure 1.1 that the systems' correct operation is ensured by means of several algorithms that collect and analyze all information before providing it to the control logics. This data fusion algorithms are a entire topic by itself, and are out of the scope of this thesis work. Nevertheless, the schemes and methodologies presented here rely on the information provided by these blocks and here we can mention two different examples: firstly, there is the driver monitoring system through torque measurements, that is in charge of detecting if the steering wheel applied torque is above a minimum threshold, indicating that it is being hold by the human driver. This detection is important as the auto-steering assistance system needs to be disconnected for safety reasons. Then, we have the target selection and tracking algorithms, that are in charge of detecting the surrounding vehicles and provide the corresponding information to the lower level controllers.

### 1.2.3 Vehicle dynamical modeling discussion

Correct vehicle modelization plays a fundamental role for any kind of ADAS application, particularly in the kind of model-based control design and trajectory planning methods studied in this thesis. The dynamics of the vehicle have been studied and targeted in different contexts, [Sename 2013], [Fergani 2013], [Bokor 2005], [Jazar 2013] being of common knowledge in both mechanical and control fields.

The strategy to define the model used for the each one of the modules needs to be defined as a trade-off between complexity, performance and computational burden. In this context, it could be stated that is of common practice to consider a higher fidelity model for the lower level feedback controllers whereas the trajectory planner takes into account a simpler model [Gao 2010].

The main distinction can be made in between *dynamic* and *kinematic* models. The first kind of modelization is based on the study of the interacting forces between the vehicle and its environment, while the second group is mainly focused on the movement of the car, without the study of the forces that generate it. In [Carvalho 2015] it is provided an interesting performance comparison between the two groups of models.

Regarding dynamical models, we can find complete complex dynamical models, that consider vehicle movement in the three-dimensional space [Lee 2008]. This kind of models are mainly used for validation purposes and model-in-the-loop simulations, but remain too complex for control design or trajectory planning purposes. In order to reduce this complexity, it is of common practice to assume two-dimensional space movements, that is, planar displacements in the  $\{X, Y\}$  reference frame. In this context, 4-wheeled vehicle models have been used in few studies in the literature [Gao 2010] but still the community seems to retain the 2-wheeled vehicle models, this being the most widespread modelization for control design purposes, commonly denoted as *bicycle model*. In this framework, it is common to find a two or three

Degrees of Freedom (2-DoF or 3-DoF) vehicle models, where the main difference is the disregard of the longitudinal forces effects on the 2-DoF models, commonly used for auto-steering lateral dynamics control. On the contrary, both directional forces are commonly considered of interest in the case of more aggressive maneuvers, yielding a nonlinear 3-DoF bicycle model [Frasch 2013]. Nevertheless, in the wide literature related to assisted driving there is not a clear standard model accepted.

A survey of kinematic models for car-like vehicles was done by [Schubert 2008], where we can highlight the holonomic point-mass [Falcone 2008] vehicle model or de Dubins car [Muller 2007], commonly used for parking trajectory planning methods. In addition, for collision avoidance purposes a point-mass model is commonly accepted [Falcone 2007]. Needless to say, a higher fidelity representation on the higher level planner would facilitate the control task to the lower level controllers, but would complexify the required resolution algorithms, thus generating a needed of trade-off. However, even for trajectory planning, different complexity models are considered in the literature, according to the ultimate objective and aggressiveness of the generated maneuvers.

### 1.2.3.1 Vehicle lateral dynamics modelization

In [Carvalho 2015] a brief overview of the generally used models for vehicle steering control are outlined and [Rajamani 2006] provides a complete background on the topic and allow to understand deeply the interactions of this system with the environment. In the subsequent, a 2-DoF dynamical model based on the interacting lateral forces between the vehicle and the road surface through the tires is used. The adopted dynamical model is based on the widely known *bicycle model*, as it exhibits a good compromise between complexity and performance. The main idea of this model is to merge together each pair of wheels situated at the same axis, as shown in Figure 1.3.

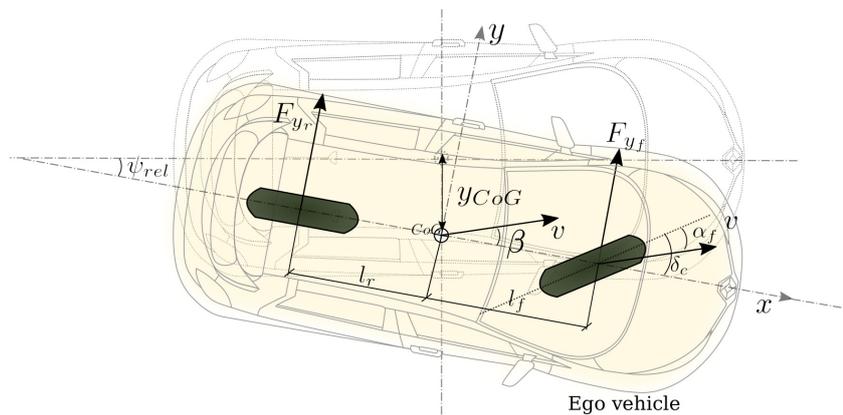


Figure 1.3 – 2-DoF Bicycle model

**Assumption 1.1** *For control design purposes, the lateral and longitudinal dynamics of the vehicle are decoupled, and only lateral forces effects are considered for the 2-DoF model.*  $\square$

This means that the interaction between lateral and longitudinal forces is neglected and only two degrees of freedom (DoF), namely the lateral position  $y$  and the heading angle  $\psi$ , are taken into consideration. Hence, only lateral interacting forces and vertical torque equilibrium define the dynamical equations,

$$\begin{aligned} m\ddot{y} &= \sum F_y, \\ I_z\ddot{\psi} &= \sum M_z. \end{aligned} \quad (1.1)$$

Considering the lateral and centripetal acceleration that will affect the vehicle,  $\ddot{y} = \ddot{y} + \dot{\psi}v_x$ , together with the lateral forces acting on the wheels we have

$$\begin{aligned} m(\ddot{y} + \dot{\psi}v_x) &= \sum F_y = F_{y_f} + F_{y_r}, \\ I_z\ddot{\psi} &= \sum M_z = l_f F_{y_f} - l_r F_{y_r}. \end{aligned} \quad (1.2)$$

The next part that needs to be analyzed is the study of lateral forces acting on the front and rear wheels,  $F_{y_f}$  and  $F_{y_r}$  respectively. Regarding auto-steering applications under regular driving conditions<sup>1</sup>, small slip angles<sup>2</sup> consideration is of common practice, thus the interacting forces between the tyres and the ground are restricted to their linear region [Pacejka 2005] (Assumption 1.2). However, the reader is referred to [Di Cairano 2013], [Falcone 2006], for auto-steering applications under more extreme driving conditions or coupled longitudinal and lateral vehicle dynamics modeling or tyres working on the nonlinear region [Liu 2010].

**Assumption 1.2** *Under normal driving conditions, small slip angle values restrict the range of operation of the lateral forces acting on the wheels to their linear region.*  $\square$

This linear dependence can be written as a proportional relationship of the lateral forces with respect to the cornering stiffness  $C$  of the tyres and the slip angle  $\alpha$ ,

$$\begin{aligned} F_{y_f} &= 2C_f\alpha_f = \bar{C}_f\alpha_f, \\ F_{y_r} &= 2C_r\alpha_r = \bar{C}_r\alpha_r. \end{aligned} \quad (1.3)$$

**Remark 1.1** *The cornering stiffness  $\bar{C}_r$  and  $\bar{C}_f$  in (1.3) contain a scaling factor of 2, which accounts for the effects of the two wheels present on the same axis that have been merged together when defining the bicycle modelization.*  $\square$

1. Understood as working in the linear region of the tyres, without performing aggressive steering maneuvers.

2. Orientative value:  $\alpha \in [-10^\circ, 10^\circ]$ . Nevertheless, it will depend on the vehicle tyres model, and the relationship would need to be studied for each case.

Then, the slip angle  $\alpha_*$  is defined as the angle between the orientation of the wheel and the vector speed of the wheel, depicted in Figure 1.3 under the following assumption.

**Assumption 1.3** *Only vehicles with front steering configurations are considered, i.e.  $\delta_r = 0$ .*  $\square$

Along these lines, the slip angle of the front wheel  $\alpha_f$  is given by

$$\alpha_f = \delta_f - \gamma_f, \quad (1.4)$$

while the rear slip angle is given by (Assumption 1.3)

$$\alpha_r = -\gamma_r, \quad (1.5)$$

where  $\gamma_f$  and  $\gamma_r$  stand for the angle between the longitudinal speed and the longitudinal axis of the vehicle at the front and rear wheels respectively. In the following, it is obtained from the ratio of the lateral to the longitudinal speed:

$$\tan(\gamma_f) = \frac{\dot{y} + l_f \dot{\psi}}{v_x}, \quad \tan(\gamma_r) = \frac{\dot{y} - l_r \dot{\psi}}{v_x}. \quad (1.6)$$

Using small angle approximations,  $\tan(\gamma_f) \approx \gamma_f$  and  $\tan(\gamma_r) \approx \gamma_r$ , we can directly substitute (1.6) on (1.4), (1.5) and obtain the lateral forces in (1.3) as

$$F_{y_f} = \bar{C}_f \left( \delta_f - \frac{\dot{y} + l_f \dot{\psi}}{v_x} \right), \quad F_{y_r} = \bar{C}_r \left( -\frac{\dot{y} - l_r \dot{\psi}}{v_x} \right). \quad (1.7)$$

When dealing with lane centering applications, it becomes of interest to do a change of reference, expressing some of the model states with respect to the road. In this way, the following change of coordinates has been considered

$$\begin{aligned} \dot{\psi}_{rel} &= \dot{\psi} - \dot{\psi}_{road} = \dot{\psi} - v_x \rho, \\ \dot{y}_{CoG} &= \dot{y} - v_x \psi_{rel}. \end{aligned} \quad (1.8)$$

The resulting dynamical model is shown in the following:

$$\begin{aligned} \dot{x}(t) &= A_c(v_x(t))x(t) + B_c u(t) \\ y(t) &= C_c x(t) \end{aligned} \quad (1.9)$$

with:

$$A_c(v_x(t)) = \begin{bmatrix} \frac{-(C_f l_f^2 + C_r l_r^2)}{I_z v_x(t)} & \frac{(C_f l_f - C_r l_r)}{I_z} & \frac{-(C_f l_f - C_r l_r)}{I_z v_x(t)} & 0 \\ 1 & 0 & 0 & 0 \\ \frac{(C_r l_r - C_f l_f)}{m v_x(t)} & \frac{(C_f + C_r)}{m} & \frac{-(C_f + C_r)}{m v_x(t)} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (1.10)$$

$$B_c = \begin{bmatrix} \frac{C_f l_f}{I_z} \\ 0 \\ \frac{C_f}{m} \\ 0 \end{bmatrix}, \quad C_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.11)$$

The state vector  $x \in \mathbb{R}^{n_x}$  is defined together with the control input  $u \in \mathbb{R}^{n_u}$  and the measured outputs  $y \in \mathbb{R}^{n_y}$  by the following sequence of system states, depicted in Table 1.1, with  $n_x = 4$ ,  $n_y = 3$  and  $n_u = 1$ .

$$\begin{aligned} x &= [\dot{\psi} \quad \psi_{rel} \quad \dot{y}_{CoG} \quad y_{CoG}]^T \\ y &= [\dot{\psi} \quad \psi_{rel} \quad y_{CoG}]^T \\ u &= \delta \end{aligned} \quad (1.12)$$

Table 1.1 – Bicycle model states description

State	Description
$\dot{\psi}$	Rate of change of vehicle orientation (yaw rate)
$\psi_{rel}$	Heading angle with respect to the road center line
$\dot{y}_{CoG}$	Lateral speed with respect to the road center line
$y_{CoG}$	Lateral offset with respect to the road center line
$\delta$	Front wheel steering angle

System matrices  $A_c(v_x(t)) \in \mathbb{R}^{n_x \times n_x}$ ,  $B_c \in \mathbb{R}^{n_x \times n_u}$  and  $C_c \in \mathbb{R}^{n_x \times n_y}$  depend on several vehicle constants and parameters, described in Table 1.2.

Table 1.2 – Vehicle parameters definition

Parameter	Description
$C_f$	Equivalent cornering stiffness of the front wheels [N/rad]
$C_r$	Equivalent cornering stiffness of the rear wheels [N/rad]
$l_f$	Longitudinal distance between CoG and front wheels axis [m]
$l_r$	Longitudinal distance between CoG and rear wheels axis [m]
$I_z$	Vehicle moment of inertia along the vertical axis [kgm <sup>2</sup> ]
$m$	Vehicle total mass [kg]
$v_x$	Longitudinal speed of the vehicle

### 1.3 Open problems

Beyond all the development concerning technological progress, it must be noted that automated driving in all conceivable traffic situations requires considerably more development on the sensing and control design capabilities than available at

the current state-of-the art. This is in part due to the fact that despite the enclosed context of the ADAS systems, driving conditions are never the same, so it is a critical feature to ensure a correct behavior under system's variation of parameters or in the presence of uncertainty. When focusing on the development of ADAS control system, research directions seek to come up with robust control strategies that will always keep a correct performance. At the same time, such control applications in which human life is involved must find the means to ensure system safety for the passengers of the vehicle, as well as for the surrounding elements. In this framework, control strategies that ensure hard constraints handling from the design stage is becoming a key subject.

In addition, once the basic functional control logic blocks are gaining in maturity, reliability and performances, more complex maneuvers will be progressively come into the picture, like the case of semi-automated lane change or overtaking. This group of applications will require to master the lower level tasks, like the lower level steering control. After that, the research efforts will be oriented towards the development of other kinds of algorithms needed to implement these complex tasks. This is the case of trajectory and planning strategies, which have a long research history in the field of robotic manipulators. Still, there is a long way to go when speaking of vehicle applications, where highly dynamic and uncontrolled environments make the task considerably more complex.

## 1.4 Thesis structure

The manuscript's structure has been divided according to the two main application topics that have been studied along this industrial thesis work. The first part, focused on the constrained control of vehicle lateral dynamics, is composed of three chapters, whose main content is briefly introduced in the subsequent. After that, the second part of the document is compounded by two chapters, where trajectory planning for overtaking in highways is the driving axis.

### **PART I: VEHICLE LATERAL DYNAMICS CONSTRAINED CONTROL**

#### **Chapter 2: Theoretical background for constrained control**

This chapter aims to introduce a set of tools that will be used in for the constrained control purposes in the first part of the manuscript. The idea is to offer to the reader a brief insight into the theoretical concepts that are needed to understand the work presented in the following two chapters.

#### **Chapter 3: Lateral Dynamics constrained control**

This chapter presents two different generic approaches for the control design of a vehicle Auto-Steering by Target tracking system, where the uncertainty produced by the variation on the speed is considered explicitly. This speed variation brings an uncertain model, that will be described by dynamics with polytopic uncertainty.

Thanks to the online measurement of the parameter, the system dynamics will be computed at each sample time to solve a Model Predictive Control optimization problem. Parameter-dependent Lyapunov functions and positive set invariance theory are used to ensure stability and feasibility. After that, Interpolation Based control, whose principle is to use a control action constructed as an interpolation between two computed extreme values, is studied. Each time, two linear programming problems are solved, which makes this approach to be a suitable trade-off between performance and computation cost.

#### **Chapter 4: Lane Centering Assistance System**

The attention of this chapter is focused on one of the lateral dynamics control system known as the Lane Centering Assistance system, which is defined by Linear Parameter-Varying (LPV) model comprising the most relevant system dynamics and the curvature of the road, modeled as a bounded parameter-varying additive disturbance. From a theoretical point of view, robust positive invariance theory is exploited in order to perform the analysis of the effects that a bounded parameter-varying additive disturbance has on a linear parameter-varying controller used to ensure stability of a model predictive control strategy. After that, vehicle speed and curvature of the road variations are considered at the design stage, in order to compute a suitable observer-based feedback controller that ensures performance under these changes on the driving conditions and system limitations, translated into control design constraints. In addition, constraint satisfaction and the maximization of the domain of attraction are considered, in order to provide a certified region of operation. As a last part of the study, in order to reduce the conservativeness introduced by large parameter variations, a discontinuous multiple parameter-dependent design and maximal acceleration considerations are proposed.

### **PART II: TRAJECTORY PLANNING FOR LANE CHANGE AND OVERTAKING ON HIGHWAYS**

#### **Chapter 5: Theoretical background for trajectory planning**

Accordingly to the structure of the first part, this first chapter includes an overview of the theoretical concepts used for trajectory planning purposes. In short, state-of-art of numerical methods that can be used to solve Optimal Control Problems and hyperplane arrangements theoretical tools are brought into the picture, ensuring an initial base for the understanding of the next chapter.

#### **Chapter 6: Collision-free trajectory planning on highways**

The final chapter of the manuscript addresses the problem of vehicle lane change and overtaking on highways in the assisted driving framework. In order to perform such maneuvers, it is fundamental to compute safe and comfortable trajectories that take into account the vehicle limitations as well as safety restrictions. On top of

the internal limitations, the critical feature to be considered in the design is the interaction with the surrounding vehicles when driving in such a dynamic environment. These vehicles, sharing the environment, define a non-convex feasible region which is described in the present work in terms of hyperplane arrangements leading to mixed-integer formulation of the anti-collision constraints. Previous work on the reduction of the necessary binary variables and cell merging techniques presented in Chapter 5 are considered and applied to an exhaustive enumeration of possible overtaking scenarios. Finally, a constrained optimal control problem is formulated and translated into a finite dimension non-linear programming problem via direct optimization multiple-shooting approach that is solved in a receding horizon fashion.

## 1.5 Thesis contributions

The overall topic of the thesis is Advanced Driving Assistance Systems (ADAS). In the first part of this thesis work, a particular focus is given to autosteering systems control, understood as lateral dynamics constrained control design and the implications of the parameter-varying nature of the system dynamics, provided by the variation of the speed of the vehicle. In addition, the robustness of the control design with respect to the impact of the curvature of the road, modeled as a bounded parameter-varying additive disturbance is studied using Robust Positive Invariance theory for the analysis and design of the Linear Parameter Varying (LPV) system. This subject is motivated by the need of a large robust domain of attraction, where system constraints satisfaction is certified from the design stage. On top of this, Model Predictive Control (MPC) and Interpolation Based Control (IBC) methods are studied in order to enforce constraint satisfaction and the enlargement of the domain of attraction, provided by the receding horizon strategy. The main contribution of the first part is methodological and proves that an integrated design can cope with a large range of parameter variation and the modeled additive disturbances and could provide certification for the safety and other constraints on the closed-loop functioning. In particular, IBC is shown to lead to a very compact optimization-based control law.

In the second part of this thesis, attention has been focused on the trajectory planning algorithms, where the study and analysis of the dynamical environment in which the maneuver will be performed, together with the physical limitations of the vehicle, are studied together to obtain suitable lane change trajectories that allow to perform this kind of maneuver with safety certifications, understood as anti-collision enhancements. Once this is set, attention is given to optimization-based trajectory generation, where the minimization of maneuver jerk is the main objective, ensuring passenger's comfort all along the trajectory execution. The contribution of this second part corresponds to an optimization based trajectory planning method that provides suitable trajectories that ensure safety and comfort constraints, all in addition to the exhaustive description of the possible scenarii in terms of hyperplane arrangements. Moreover, the reduction of the necessary binary

---

variables and cell merging techniques are considered and applied to the exhaustive enumeration of possible overtaking scenarii leading to a *minimal* representation in terms of binary variable.



Part I

VEHICLE LATERAL DYNAMICS  
CONSTRAINED CONTROL



# Theoretical background for constrained control

---

## Contents

---

<b>2.1</b>	<b>Linear Parameter Varying systems</b> . . . . .	<b>18</b>
2.1.1	General considerations on LPV models . . . . .	18
2.1.2	Polytopic representation of LPV systems . . . . .	19
<b>2.2</b>	<b>Set invariance theory for input-free discrete-time LPV systems</b> . . . . .	<b>20</b>
2.2.1	Polyhedral sets . . . . .	20
2.2.1.1	Basic definitions . . . . .	20
2.2.1.2	Polyhedral invariant sets . . . . .	22
2.2.2	Ellipsoidal sets . . . . .	28
2.2.2.1	Linear Matrix Inequalities framework . . . . .	28
2.2.2.2	Basic definitions for ellipsoids . . . . .	29
2.2.2.3	Other remarks involving ellipsoids . . . . .	30
2.2.2.4	Ellipsoidal invariant sets . . . . .	30
<b>2.3</b>	<b>Control design for LPV systems</b> . . . . .	<b>32</b>
2.3.1	LPV Control Design . . . . .	33
2.3.2	Model Predictive Control . . . . .	39
2.3.3	Interpolation based control . . . . .	44
<b>2.4</b>	<b>Conclusion</b> . . . . .	<b>48</b>

---

Constrained control theory can be considered by now a mature field, which is increasingly attracting the attention of the research community [Goodwin 2006], [Borrelli 2003], [Nguyen 2014], and industrial applications [Qin 2003]. Automotive applications are deeply engaged with safety certifications, thus this kind of control strategies are of special interest due to their inherent capability of enforcing the system physical limits as well as safety limitations from the control design stage.

In this chapter, we start by setting the theoretical base for the first part of the manuscript, bringing together the most significant ideas and concepts that will be further used for control design purposes. Our aim is not to offer a panorama of the constrained control design, as long as the topic and an exhaustive coverage of this methods is too complex to be covered in this framework and stays out of the scope

of the chapter. The point of view is conducted by the predictive strategies that have been effectively deployed in the present research work.

First of all, the class of systems that we have focused on are presented: discrete-time Linear Parameter Varying systems subject to parameter-varying additive disturbances. Secondly, we revisit the main concepts and procedures in the set invariance theory, as this theoretical tools allow to analyze essential regional features of the control strategies, such as stability and robustness.

Once the framework will be set, Section 2.3 presents three different control strategies for constrained systems that will be ultimately studied for the lateral dynamics control application in the subsequent chapters of this first part of the manuscript. The first control strategy (Section 2.3.1), denoted by LPV control design, is a controller synthesis founded on the computation of a stabilizing parameter-varying static feedback control law. Then, Implicit Model Predictive Control (Section 2.3.2), based on the resolution of a finite-horizon constrained optimal control problem will be described. At last, Interpolation Based Control (Section 2.3.3) is introduced, this last methodology bringing a series of novelties in the constrained control, presenting an appealing trade-off between performance and complexity.

## 2.1 Linear Parameter Varying systems

The class of Linear Parameter Varying (LPV) systems encompass the dynamical systems whose state-space representation depends linearly on an external time-varying parameter. In addition, we can mention also the quasi-LPV systems, where the varying parameter is a sub-state of the system dynamics itself.

### 2.1.1 General considerations on LPV models

**Definition 2.1** A linear parameter varying system is represented by a dynamical model that depends linearly on a time-varying parameter. The generic state-space representation for a dynamical system in discrete-time form depending linearly on a varying parameter  $\theta_k \in \mathbb{R}^{n_\theta}$  reads as follows,

$$\begin{aligned} x_{k+1} &= A(\theta_k)x_k + B(\theta_k)u_k + E(\theta_k)w_k \\ y_k &= C(\theta_k)x_k \end{aligned} \tag{2.1}$$

where  $x_k \in \mathbb{R}^{n_x}$ ,  $u_k \in \mathbb{R}^{n_u}$ ,  $y_k \in \mathbb{R}^{n_y}$  and  $w_k \in \mathbb{R}^{n_w}$  are the states, input, output and additive disturbances vectors respectively. The matrices  $A(\theta) : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x \times n_x}$ ,  $B(\theta) : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x \times n_u}$ ,  $C(\theta) : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_y \times n_x}$  and  $E(\theta) : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x \times n_w}$  define the system dynamical behavior.  $\square$

The characterization in (2.1) defines a family of admissible dynamical models. Accordingly, LPV analysis concerns assessing properties (such as control design, stability or disturbance rejection) that hold for the full family of LPV systems, rather than the sub-class of LTI system [Mohammadpour 2012].

**Remark 2.1** For each particular case, system dynamics may depend on several parameters  $\theta \in \Theta$ , with  $\Theta \in \mathbb{R}^{n_\theta}$  being the group of varying parameters appearing on the system dynamics. Moreover, not all the system matrices  $A$ ,  $B$ ,  $C$  and  $E$  will necessarily depend on the parameters  $\theta$  as it will be seen in the next chapters dedicate to specific automotive applications.  $\square$

Remark 2.1 opens the discussion to endless variety of possibilities. Nevertheless, as an special case and for the sake of clarity, let us continue the exposition under the following assumptions.

**Assumption 2.1** Input and output matrices  $B$  and  $C$  respectively are constant matrices and do not depend on any varying parameter  $\theta$ .  $\square$

Under Assumption 2.1, the LPV dynamical system reads as follows

$$\begin{aligned} x_{k+1} &= A(\theta_k)x_k + Bu_k + E(\theta_k)w_k \\ y_k &= Cx_k \end{aligned} \quad (2.2)$$

Controllability of LPV systems has been introduced in serveral references on the literature [Bokor 2005], [Mohammadpour 2012]. In the remaining of this part, the following assumption is considered:

**Assumption 2.2** LPV system (2.2) is controllable.  $\square$

### 2.1.2 Polytopic representation of LPV systems

There exist several ways of representing a dynamical system containing a parametric uncertainty, where structured feedback uncertainty, affine representation (nonlinear systems) and polytopic uncertainty description [Kothare 1996], [Bemporad 1999] are the most widespread methods.

**Assumption 2.3** The parameter  $\theta_k \in \Theta$ , where  $\Theta$  is a convex set in  $\mathbb{R}^{n_\theta}$ , also the vector  $\theta_k$  is measured at each sample time but unknown a priori on a future horizon.  $\square$

Whenever the LPV model is used for the prediction, the lack of future information lead to interpretation of the Assumption 2.3 in the sense that the varying parameter is lying in a bounded polytope  $\Theta \in \mathbb{R}^{n_\theta \times n_\theta}$ , defined by

$$\Theta = \text{ConvHull} \{ \theta_1, \dots, \theta_{n_v} \}, \quad (2.3)$$

with  $n_v$  being the number of parameter vertices and  $n_\theta$  the dimension of the parameter. Clearly, as the parameter  $\theta$  varies inside the convex polytope  $\Theta$ , the system matrices (2.2) vary inside a corresponding polytope, defined by the convex hull of the  $n_v$  local matrix vertices at the vertex of the parameters value.

$$\mathcal{P}_{AE} = \text{Conv} \{ [A_1(\theta_1) \quad E_1(\theta_1)], \dots, [A_{n_v}(\theta_{n_v}) \quad E_{n_v}(\theta_{n_v})] \}. \quad (2.4)$$

Hence, the system matrices in (2.2) (with a straightforward extension to the matrices in (2.1)) that depend on a varying parameter and under Assumption 2.3 can be obtained as a convex combination of the system's vertex realizations that define  $\mathcal{P}_{AE}$  (2.4),

$$A(\theta) = \sum_{i=1}^{n_v} \lambda_i A_i(\theta_i), \quad E(\theta) = \sum_{i=1}^{n_v} \lambda_i E_i(\theta_i), \quad (2.5)$$

with  $\lambda_i$  belonging to the unitary  $n_v$ -simplex,  $\lambda_i \in \Lambda$ ,

$$\Lambda = \left\{ \lambda_i \in \mathbb{R}^{n_v} : \sum_{i=1}^{n_v} \lambda_i = 1, \lambda_i \geq 0 \right\}. \quad (2.6)$$

## 2.2 Set invariance theory for input-free discrete-time LPV systems

Set invariance theory is a fundamental concept for the design of controllers for constrained systems by the fact that they characterize the regions guaranteeing a certain dynamical property which can be certified in conjunction with the constraints. The computation of this kind of sets represent an important step on the control synthesis and analysis, as these tools allow to assess critical features such as feasibility, stability or robustness of the strategy. A complete survey on the subject can be found in [Blanchini 1999] and the monography [Blanchini 2008]. Let us recall two formal definitions.

**Definition 2.2** [Blanchini 2007]. A set  $\mathcal{S} \subset \mathbb{R}^{n_x}$  is a positive invariant set for an autonomous discrete-time system  $x_{k+1} = f(x_k, \theta_k)$  if for any initial state  $x_0 \in \mathcal{S}$  the system evolution satisfies  $x_k \in \mathcal{S}$  for all future times  $k > 0$ .  $\square$

Moreover, it can be of interest as well to determine whether the trajectory of the system remains inside the given set in the presence of (bounded) additive disturbances in addition to the parametric uncertainties.

**Definition 2.3** [Blanchini 2007]. A set  $\mathcal{S} \subset \mathbb{R}^{n_x}$  is a robust positive invariant set for an autonomous discrete-time system  $x_{k+1} = f(x_k, \theta_k, w_k)$  if and only if for any initial state  $x_0 \in \mathcal{S}$ ,  $x_k \in \mathcal{S}$ , for  $k > 0$  independent of the bounded additive disturbance realization  $w_k \in W$ .  $\square$

Invariant sets are generally represented or approximated by means of polyhedral or ellipsoidal sets. The remaining of this section brings together useful concepts and definitions, setting up an appropriate set-theoretic framework for the subsequent.

### 2.2.1 Polyhedral sets

#### 2.2.1.1 Basic definitions

**Definition 2.4** An hyperplane  $\mathcal{H}(H, g)$  is a set of the form,

$$\mathcal{H}(H, g) = \{x \in \mathbb{R}^{n_x} : H^T x = g\}, \quad (2.7)$$

## 2.2. Set invariance theory for input-free discrete-time LPV systems 21

where  $H \in \mathbb{R}^{n_x \times n_x}$ ,  $g \in \mathbb{R}^{n_x \times 1}$ .  $\square$

**Definition 2.5** A closed half-space  $\mathcal{H}^+$ ,  $\mathcal{H}^-$  is a set of the form,

$$\begin{aligned}\mathcal{H}^+ &= \{x \in \mathbb{R}^{n_x} : H^T x \leq g\}, \\ \mathcal{H}^- &= \{x \in \mathbb{R}^{n_x} : H^T x > g\},\end{aligned}\tag{2.8}$$

where  $H \in \mathbb{R}^{n_x \times n_x}$ ,  $g \in \mathbb{R}^{n_x \times 1}$ .  $\square$

**Definition 2.6** A convex polyhedral set  $\mathcal{P}(H, g) \in \mathbb{R}^{n_x}$  is defined by the intersection of a finite set of halfspaces in  $\mathbb{R}^{n_x}$ ,

$$\mathcal{P}(H, g) = \{x \in \mathbb{R}^{n_x} : H^T x \leq g\},\tag{2.9}$$

with  $H \in \mathbb{R}^{n_x \times n_x}$  and  $g \in \mathbb{R}^{n_x \times 1}$ . This representation is known as the *half-space representation*. A polyhedral set contains the origin if and only if  $g \geq 0$ , and includes the origin in its interior if and only if  $g > 0$ .  $\square$

**Definition 2.7** A bounded polyhedral set is a *polytope*.  $\square$

A dual representation for a polytope is the *vertex representation* [Schneider 2014], where a polyhedron is defined by the convex hull of a series of points, denoted as vertex,  $v_i$ ,  $i = 1, \dots, p$ ,

$$\mathcal{P}(V) = \text{Conv}\{v_1, \dots, v_p\} = \left\{ x \in \mathbb{R}^{n_x} : x = \sum_{i=1}^p \alpha_i v_i \right\},\tag{2.10}$$

with  $\sum_{i=1}^p \alpha_i = 1$ ,  $\alpha_i \geq 0$  and  $v_i \in \mathbb{R}^{n_x}$  is the  $i$ -th column of the vertices matrix  $V \in \mathbb{R}^{n_x \times p}$ .

**Definition 2.8** The Minkowski sum of two polytopes,  $\mathcal{P}_1 \subset \mathbb{R}^{n_x}$ ,  $\mathcal{P}_2 \subset \mathbb{R}^{n_x}$  defined as

$$\mathcal{P}_1 \oplus \mathcal{P}_2 = \{x_1 + x_2 : x_1 \in \mathcal{P}_1, x_2 \in \mathcal{P}_2\}.\tag{2.11}$$

If the polytopes are given by their vertex representation (2.10), their Minkowski sum is computed as

$$\mathcal{P}_1 \oplus \mathcal{P}_2 = \text{Conv}\{v_1 i + v_2 j\}, \quad i = 1, \dots, p, \quad j = 1, \dots, q,\tag{2.12}$$

with  $p$  and  $q$  being the number of vertices of  $\mathcal{P}_1$  and  $\mathcal{P}_2$  respectively.  $\square$

**Definition 2.9** The Pontryagin difference of two polytopes,  $\mathcal{P}_1 \subset \mathbb{R}^{n_x}$ ,  $\mathcal{P}_2 \subset \mathbb{R}^{n_x}$  is the polytope,

$$\mathcal{P}_1 \ominus \mathcal{P}_2 = \{x_1 \in \mathcal{P}_1 : x_1 + x_2 \in \mathcal{P}_1, \forall x_2 \in \mathcal{P}_2\}.\tag{2.13}$$

$\square$

**Remark 2.2** Note that the Pontryagin difference and the Minkowski sum are complementary but not inverse operations. Given two polytopes,  $\mathcal{P}_1 \subset \mathbb{R}^{n_x}$ ,  $\mathcal{P}_2 \subset \mathbb{R}^{n_x}$ , it holds that  $\mathcal{P}_1 \oplus \mathcal{P}_2 \ominus \mathcal{P}_2 = \mathcal{P}_1$  but  $(\mathcal{P}_1 \ominus \mathcal{P}_2) \oplus \mathcal{P}_2 \subseteq \mathcal{P}_1$  and only under particular homothetic sets lead to equality.  $\square$

### 2.2.1.2 Polyhedral invariant sets

Let us consider a closed-loop stable LPV dynamical system in the form

$$x_{k+1} = A_{cl}(\theta_k)x_k + E(\theta_k)w_k, \quad (2.14)$$

where the closed-loop form corresponds to the stabilized system by means of a parameter-dependent gain, with  $u_k = K(\theta_k)x_k$ , thus  $A_{cl}(\theta_k) = A(\theta_k) + BK(\theta_k)$ . This uncertain closed-loop dynamics can be embedded in a polytopic representation as shown in Section 2.1.2,

$$A_{cl}(\theta) = \sum_{i=1}^{n_v} \lambda_i A_{cl}^i(\theta_i) = \sum_{i=1}^{n_v} \lambda_i (A_i(\theta_i) - BK_i(\theta_i)), \quad (2.15)$$

with  $\lambda_i$  lying in the unitary simplex (2.6) and  $A_{cl}^i(\theta_i)$  being the extreme realizations of the stabilized closed-loop dynamics.

The system dynamics are subject to polytopic constraints on the states and inputs,

$$x \in X, \quad X = \{x \in \mathbb{R}^{n_x} : H_X x \leq g_X\} \quad (2.16a)$$

$$u \in U, \quad U = \{u \in \mathbb{R}^{n_u} : H_U u \leq g_U\}, \quad (2.16b)$$

where  $X \subset \mathbb{R}^{n_x}$  and  $U \subset \mathbb{R}^{n_u}$  are respectively the bounded sets of admissible states and inputs, both containing the origin in their interior.

- MAXIMAL POSITIVELY INVARIANT AND CONSTRAINED ADMISSIBLE SET.

**Definition 2.10** [Gilbert 1991]. A set  $\Omega$  is positive invariant constrained admissible with respect to (2.14) with  $w_k = 0$  if  $x_0 \in \Omega$  implies that  $x_k \in \Omega$  for all future times,  $k > 0$ , without activating the constraints (2.16a), (2.16b). This is equivalent to

$$x_k \in \Omega \rightarrow x_{k+1} \in \Omega, \quad \forall k \text{ and } K(\theta_k)x_k \in U. \quad (2.17)$$

□

**Definition 2.11** [Gilbert 1991]. A set  $\Omega$  is a Maximal Admissible Set (MAS) or maximal positive invariant constrained admissible if it contains any other positive invariant constrained admissible set. □

In [Gilbert 1991] and [Borrelli 2015], the properties of this kind of sets are characterized and recursive algorithms are introduced in order to compute them. Based on these algorithms which involve closed operations over the class of polyhedral sets, we can obtain the positive invariant constrained admissible set  $\Omega$

$$\Omega = \{x \in \mathbb{R}^{n_x} : H_\Omega x \leq b_\Omega\} \quad (2.18)$$

## 2.2. Set invariance theory for input-free discrete-time LPV systems 23

for the constrained closed-loop stable parameter-varying system in (2.14) with  $w_k = 0$ , where  $H_\Omega$  and  $b_\Omega$  are computed recursively through Procedure 2.1.

**Procedure 2.1** Maximal Admissible Set (MAS) computation for LPV constrained stable closed-loop dynamics

1. Compute initial conditions at  $k = 0$  from the state-admissible space defined by the system constraints:

$$\Omega_0 = \{x \in \mathbb{R}^{n_x} : H_{\Omega_0} x \leq b_{\Omega_0}\}, \quad (2.19)$$

where

$$H_{\Omega_0} = \begin{bmatrix} H_X \\ H_U K_i \end{bmatrix}, \quad b_{\Omega_0} = \begin{bmatrix} g_X \\ g_U \end{bmatrix}, \text{ with } i = 1 \dots n_v. \quad (2.20)$$

Set  $\Omega_k = \Omega_0$ .

2. Compute image set  $\Omega_{k+1}^i = \{x \in \mathbb{R}^{n_x} : H_{\Omega_{k+1}}^i x \leq b_{\Omega_{k+1}}^i\}$ , with  $i = 1 \dots n_v$ , where

$$H_{\Omega_{k+1}}^i = \begin{bmatrix} H_{\Omega_k} \\ H_{\Omega_k} A_{cl}^i(\theta_i) \end{bmatrix}, \quad b_{\Omega_{k+1}}^i = \begin{bmatrix} b_{\Omega_k} \\ b_{\Omega_k} \end{bmatrix}. \quad (2.21)$$

3. Suppress redundant constraints.
4. Set  $\Omega_{k+1} = \bigcap_{i=1}^{n_v} \Omega_{k+1}^i$ .
5. If  $\Omega_{k+1} = \Omega_k$  stop and return  $\Omega_k$ . Else  $k = k + 1$  and go to Step 2.

Finite determinedness of the MAS set is guaranteed in the absence of additive disturbances by the the stability of the stable closed-loop dynamics [Blanchini 1996], [Gilbert 1991]. For the LPV systems represented in polytopic form, the finite determinedness is related to the robust asymptotic stability [Olaru 2008].

- MAXIMAL AND MINIMAL ROBUST POSITIVELY INVARIANT ADMISSIBLE SET.

**Definition 2.12** [Borrelli 2015]. A set  $\Omega$  is robust positive invariant and constrained admissible or, in short, robust positive invariant (RPI) with respect to (2.14) if  $x_0 \in \Omega$  implies that  $x_k \in \Omega$  for all future times,  $k > 0$ , despite the presence of bounded additive disturbances  $w_k \in \mathcal{W}$  and without activating the constraints. This is equivalent to

$$x_k \in \Omega \rightarrow x_{k+1} \in \Omega \text{ and } u_k \in U \quad \forall k \text{ and } \forall w \in \mathcal{W}. \quad (2.22)$$

□

In addition, in the presence of such additive disturbances, the analysis of the Maximal Robust Positively Invariant and Minimal Robust Positively Invariant

sets becomes of interest to study quantitatively the effect of the disturbances on the system dynamics.

**Definition 2.13** A Maximal Robust Positively Invariant (MRPI) set  $\Omega_M \subset X$  is a RPI set that contains any other RPI set contained in  $X$ .  $\square$

The computation of the MRPI,  $\Omega_M$ , can be done by means of a recursive algorithm, extracted from [Nguyen 2011], where more details on the topic and proof of convergence for (2.23) are detailed.

**Procedure 2.2** Maximal Robust Positive Invariant Set (MRPI) computation for LPV constrained stable closed-loop dynamics

$$\Omega_M(s) = \left\{ x : \begin{array}{l} \{x\} \subseteq X \\ \{A_i x\} \oplus E_i W \subseteq X \\ \dots \\ \{A_i^{s-1} x\} \oplus \bigoplus_{k=0}^{s-2} (A_i^k) E_i W \subseteq X \end{array} \right\}. \quad (2.23)$$

The presence of additive disturbances bounds the convergence of the system state to a region of the space around the origin. This region of the space can be delimited by the analysis of the minimal robust positive invariant set, which provides a quantitative measurement of the uncertainty introduced on the system dynamics due to the presence of additive disturbances: the larger the mRPI set is, the more the system is affected by the disturbances.

**Definition 2.14** A minimal Robust Positively Invariant set (mRPI)  $\Omega_m \subset X$  is the RPI set which is contained in any other RPI set contained in  $X$ .  $\square$

This set represents the set of states that can be reached from the origin under a bounded additive disturbance affecting the system dynamics. The exact computation of this kind of sets still remains a challenge for the general case of dynamics [Mayne 1997], and usually an  $\epsilon$ -outer approximation of this set needs to be computed [Rakovic 2005]. In the following, we present an algorithm following the lines of [Olaru 2010] where the approximation of the mRPI is computed starting from an initial RPI set, that is afterwards refined by means of a recursive strategy, adapted for the kind of LPV system dynamics we are studying in this work.

**Procedure 2.3** Minimal Robust Positive Invariant Set (mRPI) computation for LPV constrained stable closed-loop dynamics

1. Set initial condition,  $\Omega_{m_k} = \Omega_{M_k}$ , with  $\Omega_{m_k}$  being RPI (Procedure 2.2).
2. Compute image of the RPI set  $\Omega_{m_k}$  and add the bounded additive disturbances set  $W$ , for  $i = 1 \dots n_v$ ,

$$\Omega_{m_{k+1}} = \text{ConvHull}\{A_{cl}^i(\theta_i)\Omega_{m_k}\} \oplus \text{ConvHull}\{E_i(\theta_i)W\}. \quad (2.24)$$

3. If  $\Omega_{m_{k+1}} = \Omega_{m_k}$  stop and return  $\Omega_{m_k}$ . Else  $k = k + 1$  and go to Step 2.

- CONTROLLED POSITIVELY INVARIANT SET.

**Definition 2.15** Given a LPV dynamical system subject to polytopic constraints, a set  $\mathcal{C} \subseteq X$  is controlled positively invariant if  $x_0 \in \mathcal{C}$  implies that  $\exists u_k \in U$  such that  $x_k \in \mathcal{C}$  for all  $k > 0$  and  $\theta \in \Theta$ .

$$\mathcal{C}_\infty = \{x \in \mathbb{R}^{n_x} : \exists u \in \mathcal{U} \text{ s.t. } (A(\theta)x + Bu(\theta)) \in \mathcal{C}_\infty\}. \quad (2.25)$$

□

Ideally, the maximal controllable set  $\mathcal{C}_\infty$  would be of interest but this construction is finitely-determined only for particular cases and is particularly complex for certain parameter-varying systems.

In these cases, it is possible to compute the controlled positively invariant set of states that can be brought into the MAS set  $\Omega$  (2.18) in no more than  $N$  steps through a trajectory that fulfills system's constraints. The index  $N$  becomes a tuning variable in between the complexity of the construction and the "volume" of the invariant set.

**Definition 2.16** Given a positive invariant set  $\Omega \subset X$ , a set  $\mathcal{C}_N$  is N-step controlled invariant if any  $x_0 \in \mathcal{C}_N$  can be driven into a maximal admissible set  $\Omega$  in N-steps, while satisfying the constraints and applying admissible control inputs,  $u_k \in U$ .

Such a polytopic N-step controlled invariant set will be denoted as

$$\mathcal{C}_N = \{x \in \mathbb{R}^{n_x} : H_{\mathcal{C}_N}x \leq g_{\mathcal{C}_N}\}. \quad (2.26)$$

□

This set can be computed for the constrained LPV system by means of recursive pre-image set construction with respect to  $\Omega$  and for each extreme realization of the polytopic uncertainty description.

**Definition 2.17** [Kvasnica 2015]. A pre-image set  $\text{Pre}(\mathcal{S})$  is the set of states from which the evolution of the LPV system enters in the set  $\mathcal{S}$  in one time step.

$$\text{Pre}(\mathcal{S}) = \{x \in \mathbb{R}^{n_x} : \exists u \in U \text{ s.t. } (A(\theta)x + Bu(\theta)) \in \mathcal{S}\}. \quad (2.27)$$

□

**Procedure 2.4** Pre-image set computation for a LPV constrained controlled system.

1. Given a constrained controlled system,  $x_{k+1} = A(\theta)x_k + Bu_k(\theta)$ , with  $A(\theta)$  invertible and  $x \in \mathcal{S}$  and  $u \in U$ , the pre-image of a set  $\mathcal{S} \subset X$  is computed as follows,

$$\text{Pre}(\mathcal{S}) = \bigcap_{i=1}^{n_v} A_i^{-1}(\theta_i)(\mathcal{S} \oplus (-BU)). \quad (2.28)$$

In few words, the construction of  $\mathcal{C}_N$  (Procedure 2.5) is based on the iterative computation of the set of states from which we could reach the original set despite the worst parametric uncertainty and input constraints. Starting from a polyhedral set, the pre-image preserves the polyhedral structure and thus a polyhedral control invariant set  $\mathcal{C}_N$  is guaranteed.

**Procedure 2.5**  $N$ -step controlled invariant set  $\mathcal{C}_N$  computation for LPV constrained dynamics

1. Set initial conditions at  $k = 0$ ,  $\mathcal{C}_0 = \Omega$
2. Compute pre-image set,

$$\mathcal{C}_k^i = \{x \in X : \exists u \in U : H_C(A_i(\theta_i)x + Bu(\theta_i)) \leq g_C\}, \quad (2.29)$$

with  $i = 1 \dots n_v$ . Or equivalently (Procedure 2.4),

$$\text{Pre}(\mathcal{C}_k^i) = \bigcap_{i=1}^{n_v} A_i^{-1}(\theta_i)(\mathcal{C}_k^i \oplus (-BU)). \quad (2.30)$$

3. If  $\mathcal{C}_{k+1} = \mathcal{C}_k$  stop and return. Else  $k = k + 1$  and go to Step 2.

If the procedure is initiated with an invariant set  $\Omega$  then  $\mathcal{C}_N$  is ensured to be an invariant set too.

**Remark 2.3** With  $N$  going to infinity we obtain the maximal controllable set and thus  $\mathcal{C}_N$  is an approximation of this limit which preserves the robust controlled invariance properties. □

- ROBUST CONTROLLED POSITIVELY INVARIANT SET.

**Definition 2.18** Given a LPV dynamical system subject to polytopic constraints and bounded additive disturbances,  $w_k \in \mathcal{W}$ , a set  $\mathcal{C} \subseteq X$  is positively controlled invariant if  $x_0 \in \mathcal{C}$  implies that  $\exists u_k \in U$  such that  $x_k \in \mathcal{C}$  for all  $k > 0$  and  $\forall w_k \in \mathcal{W}$ .

$$\mathcal{C}_\infty = \{x \in \mathbb{R}^{n_x} : \exists u \in \mathcal{U} \text{ s.t. } (A(\theta)x + Bu(\theta) + E(\theta)w_k) \in \mathcal{C}_\infty\}. \quad (2.31)$$

□

Again, if the maximal robust controlled positive invariant set is not finitely determined, it is possible to compute the  $N$ -steps robust controlled positive invariant set,  $\mathcal{C}_N$ , for the uncertain system subject to bounded additive disturbances with a parameter varying disturbances matrix  $E(\theta)$ . Following the principles of Procedure 2.5, Procedure 2.7 shows how to compute  $\mathcal{C}_N$  in the presence of bounded additive disturbances, where the main difference is the consideration of the disturbances set when computing the robust pre-image set (Procedure 2.6), following a worst-case disturbances strategy.

**Procedure 2.6** Robust pre-image set computation for a constrained controlled system.

1. Given a constrained controlled LPV system,  $x_{k+1} = A(\theta)x_k + Bu_k(\theta) + E(\theta)w_k$ , with  $x \in \mathcal{S}$ ,  $w_k \in \mathcal{W}$  and  $u \in U$ .

$$Pre(\mathcal{S}) = \bigcap_{i=1}^{n_v} A_i^{-1}(\theta_i)((\mathcal{S} \ominus E_i(\theta_i)W) \oplus (-BU)). \quad (2.32)$$

**Procedure 2.7** Robust  $N$ -step controlled invariant set  $\mathcal{C}_N$  computation for LPV constrained dynamics subject to polytopic uncertain additive disturbances

1. Set initial conditions at  $k = 0$ ,  $\mathcal{C}_0 = \Omega$
2. Compute pre-image set,

$$\mathcal{C}_k^i = \{x \in \mathbb{R}^{n_x} : \exists u \in U : H_C(A_i(\theta_i)x + Bu + E(\theta_i)w_k) \leq g_C\}, \quad (2.33)$$

with  $i = 1 \dots n_v$ . Or equivalently (Procedure 2.6),

$$Pre(\mathcal{C}_k^i) = A_i^{-1}(\theta_i)((\mathcal{S} \ominus E_i(\theta_i)W) \oplus (-BU)). \quad (2.34)$$

3. Compute  $\mathcal{C}_{k+1}$  from the intersection of the intermediary sets,

$$\mathcal{C}_{k+1} = X \cap \left( \bigcap_{i=1}^{n_v} \mathcal{C}_k^i \right). \quad (2.35)$$

4. If  $\mathcal{C}_{k+1} = \mathcal{C}_k$  stop and return. Else  $k = k + 1$  and go to Step 2.

**Remark 2.4** *The Multi-parametric Toolbox [Herceg 2013] is a free-software tool that provides a series of performant and optimized methods to compute all this sets.*  $\square$

### 2.2.2 Ellipsoidal sets

Positive invariant sets are commonly approximated by means of ellipsoids. This kind of convex sets are widely used on the literature due to the compactness of their representation and the straightforward connection with the Linear Matrix Inequalities framework and quadratic Lyapunov functions.

#### 2.2.2.1 Linear Matrix Inequalities framework

First of all, let us briefly introduce Linear Matrix Inequalities (LMI) basic tools that will be used along this manuscript. For more details on the subject, the readers are referred to [Boyd 1994]. In practice, there exists nowadays a solid collection of open-software tools that allow to state and solve problems in the LMI framework in a simplified manner. We highlight the YALMIP toolbox, that offers both a modeling framework and a parser to different optimization problems [Lofberg 2005].

**Definition 2.19** [Boyd 1994]. A Linear Matrix Inequality (LMI) in a variable  $x \in \mathbb{R}^{n_x}$  has the form

$$F(x) \triangleq F_0 + \sum_{i=1}^{n_x} x_i F_i \succ 0, \quad (2.36)$$

where the symmetric matrices  $F_i = F_i^T \in \mathbb{R}^{m \times m}$ ,  $i = 0, \dots, n_x$  are given and  $F(x)$  is a positive definite matrix, that is,  $F(x) > 0$  is such that  $u^T F(x) u > 0$  for all nonzero  $u \in \mathbb{R}^n$ .  $\square$

#### Schur complement lemma

The Schur complement is a powerful tool that allows to translate convex quadratic inequalities in a LMI form. Let  $Q(x) = Q(x)^T$ ,  $R(x) = R(x)^T$ , and  $S(x)$  depend affinely on  $x$ . Then the LMI

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} \succ 0 \quad (2.37)$$

is equivalent to the following matrix inequalities

$$\begin{aligned} R(x) > 0, \quad Q(x) - S(x)R(x)^{-1}S(x)^T > 0 \\ Q(x) > 0, \quad R(x) - S(x)^T Q(x)^{-1}S(x) > 0. \end{aligned} \quad (2.38)$$

#### S-procedure for quadratic forms

The S-procedure allows to determine whether or not some quadratic function is (non)negative whenever some other quadratic functions are all (non)negative. The

## 2.2. Set invariance theory for input-free discrete-time LPV systems 29

problem that the S-procedure addresses can be transposed by a simple principle as follows: find when does the (non)negativity of a set of quadratic forms imply the (non)negativity of another.

**Definition 2.20** [Boyd 1994]. Let  $T_0 \dots T_p \in \mathbb{R}^{n_x \times n_x}$ , be symmetric matrices. We consider the following condition on  $T_0 \dots T_p$ ,

$$x^T T_0 x > 0 \text{ for all } x \neq 0 \text{ such that } x^T T_i x > 0, i = 1, \dots, p. \quad (2.39)$$

If there exists

$$\tau_1 \geq 0, \dots, \tau_p \geq 0 \text{ such that } T_0 - \sum_{i=1}^p \tau_i T_i > 0, \quad (2.40)$$

then (2.39) holds.  $\square$

### 2.2.2.2 Basic definitions for ellipsoids

**Definition 2.21** An ellipsoid  $\mathcal{E}(P, x_0)$  centered at  $x_0$  with shape matrix  $P$  is the convex set given by

$$\mathcal{E}(P, x_0) = \{x \in \mathbb{R}^{n_x} : (x - x_0)^T P^{-1} (x - x_0)\}, \quad (2.41)$$

with  $P \in \mathbb{R}^{n_x \times n_x}$  being a positive definite matrix. For the particular case of an ellipsoid centered at the origin,  $x_0 = 0$ , we have

$$\mathcal{E}(P) = \{x \in \mathbb{R}^{n_x} : x^T P^{-1} x \leq 1\}. \quad (2.42)$$

$\square$

There are alternative ways of representing ellipsoids. Let us consider the matrix  $Q$ , given by the Cholesky factor of the ellipsoid's shape matrix  $P = \sqrt{P}$  that satisfies,

$$Q^T Q = Q Q^T = P. \quad (2.43)$$

Considering  $Q$ , the above definition can be re-written as

$$\mathcal{E}(P) = \{x \in \mathbb{R}^{n_x} : (x - x_0)^T Q Q^T (x - x_0) \leq 1\} \quad (2.44)$$

$$\mathcal{E}(P) = \{x \in \mathbb{R}^{n_x} : \|Q^T (x - x_0)\| \leq 1\}, \quad (2.45)$$

which is equivalent to:

$$\mathcal{E}(P) = \{x \in \mathbb{R}^{n_x} : x = x_0 + Q^{-T} z, \|z\| \leq 1\}. \quad (2.46)$$

In the following, the size of an ellipsoid will become a property of interest. This attribute can be directly related with the length of the semi-axis of the ellipsoid, which is given by the square root of the eigenvalues  $\sqrt{\lambda_i}$  of the shape matrix  $P^{-1}$ , with  $i = 1 \dots n_x$  being the ellipsoidal dimension.

**Definition 2.22** [Hindi 2004]. The volume of an ellipsoid is proportional to the product of the eigenvalues of  $P^{-1}$ , or inversely proportional to the product of the eigenvalues of  $P$ . The direction of the semi-axis are given by the eigenvectors of the corresponding shape matrix.  $\square$

**Remark 2.5** *Ultimately, the product of the eigenvalues of the shape matrix can be related to the  $\det(P^{-1})$ , thus*

$$\text{Vol}(\mathcal{E}(P)) \propto \det(P^{-1}) \equiv \text{Vol}(\mathcal{E}(P)) \propto \frac{1}{\det(P)} \quad (2.47)$$

$\square$

**Remark 2.6** *Sometimes, it is desirable to use the  $\text{trace}(P^{-1})$  criterium, due to its linearity. The trace of a square matrix is defined as the sum of the elements of the main diagonal. Thus the maximization of the trace of a matrix is equivalent to the maximization of the sum of the eigenvalues of the matrix.*  $\square$

### 2.2.2.3 Other remarks involving ellipsoids

- INTERSECTION OF AN ELLIPSOID WITH A SUB-SUBSPACE. [Blanchini 2008]. The intersection of an ellipsoid with a lower dimension sub-subspace is also an ellipsoid  $\mathcal{E}_{\bar{x}}$ . The shape matrix of the intersection is obtained through a suitable matrix  $T$ ,

$$\mathcal{E}_{\bar{x}} = \{\bar{x}^T (TP^{-1}T^T)\bar{x} \leq 1, \bar{x} = Tx\}. \quad (2.48)$$

- PROJECTION OF AN ELLIPSOID. [Blanchini 2008]. The projection of an ellipsoid in a lower dimension sub-subspace is also an ellipsoid  $\mathcal{E}_{\bar{x}}$ . The shape matrix of the projected ellipsoid is obtained through a suitable matrix  $T$ ,

$$\mathcal{E}_{\bar{x}} = \{\bar{x}^T (TPT^T)^{-1}\bar{x} \leq 1, \bar{x} = Tx\}. \quad (2.49)$$

- ELLIPSOIDS CONTAINMENT TEST. Given two ellipsoids  $\mathcal{E}_1(P_1) = \{x \in \mathbb{R}^{n_x} : x^T P_1^{-1} x \leq 1\}$  and  $\mathcal{E}_2(P_2) = \{x \in \mathbb{R}^{n_x} : x^T P_2^{-1} x \leq 1\}$ , the objective is to check if  $\mathcal{E}_2(P_2) \subseteq \mathcal{E}_1(P_1)$ . We have

$$x^T P_2^{-1} x - x^T P_1^{-1} x \leq 0 \rightarrow x^T (P_2^{-1} - P_1^{-1}) x \leq 0. \quad (2.50)$$

We know that  $x^T x \geq 0$ , so in order to fulfill (2.50),  $(P_2^{-1} - P_1^{-1}) \prec 0$ .

### 2.2.2.4 Ellipsoidal invariant sets

Positive invariance concepts presented for polyhedral sets (Section 2.2.1.2) apply analogously to ellipsoidal positive invariant sets, so we will not duplicate here all the definitions already introduced. Instead, we focus exclusively on the basic procedures that can be used to compute this kind of ellipsoidal sets for a given LPV system in the form of (2.14)-(2.15) subject to symmetric polytopic constraints approximated by means of ellipsoids.

## 2.2. Set invariance theory for input-free discrete-time LPV systems 31

- **POSITIVE INVARIANT ELLIPSOIDAL SET.** The set  $\mathcal{E}(P) = \{x \in \mathbb{R}^{n_x} : x^T P^{-1} x \leq 1\}$  is positive invariant for the closed-loop stable LPV system  $x_{k+1} = A_{cl_i}(\theta_i)x_k$  if  $\forall x_k \in \mathcal{E}(P), x_{k+1} \in \mathcal{E}(P)$ . This is equivalent to

$$x^T P^{-1} x - x^T (A_{cl_i}^T(\theta_i) P^{-1} A_{cl_i}(\theta_i)) x \geq 0, \quad \text{with } i = 1, \dots, n_v, \quad (2.51)$$

or equivalently,

$$P^{-1} - A_{cl_i}^T(\theta_i) P^{-1} A_{cl_i}(\theta_i) \succeq 0. \quad (2.52)$$

This expression can be written in the LMI form by means of the Schur complement,

$$\begin{bmatrix} P^{-1} & A_{cl_i}^T(\theta_i) P^{-1} \\ P^{-1} A_{cl_i}(\theta_i) & P^{-1} \end{bmatrix} \succeq 0, \quad \text{with } i = 1, \dots, n_v. \quad (2.53)$$

After pre- and post-multiplication with the symmetric and full rank matrix  $\text{blkdiag}(P, P)$ , we obtain the following invariance condition.

$$\begin{bmatrix} P & P A_{cl_i}^T(\theta_i) \\ A_{cl_i}(\theta_i) P & P \end{bmatrix} \succeq 0, \quad \text{with } i = 1, \dots, n_v. \quad (2.54)$$

- **MAXIMAL AND MINIMAL ROBUST POSITIVE INVARIANT ELLIPSOIDAL SET.** The analysis of the robust positive invariant ellipsoidal sets condition can be studied on a similar basis to what has been presented for positive invariance, augmented with the explicit consideration of the presence of bounded additive disturbances affecting the system dynamics: an ellipsoid  $\mathcal{E}(P) = \{x \in \mathbb{R}^{n_x} : x^T P^{-1} x \leq 1\}$  is robust positive invariant ellipsoid if  $x_k \in \mathcal{E}(P) \implies x_{k+1} \in \mathcal{E}(P)$  in the presence of normalized bounded additive disturbances, that is  $w \in W$  and  $\|w\|_2 \leq 1$ ,

$$x_{k+1} P^{-1} x_{k+1} \leq 1, \quad (2.55)$$

which is equivalent to

$$(A_{cl_i} x_k + E_i w_k)^T P^{-1} (A_{cl_i} x_k + E_i w_k) \leq 1, \quad (2.56)$$

with  $i = 1, \dots, n_v$ , when

$$x_k^T P^{-1} x_k \leq 1 \quad (2.57a)$$

$$w^T w \leq 1. \quad (2.57b)$$

In a LMI form,

$$\begin{bmatrix} x_k & w_k \end{bmatrix} \begin{bmatrix} A_{cl_i} P^{-1} A_{cl_i} & A_{cl_i}^T P^{-1} E_i \\ E_i P^{-1} A_{cl_i} & E_i^T P^{-1} E_i \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} \leq 1, \quad (2.58)$$

with  $i = 1, \dots, n_v$ , when

$$\begin{bmatrix} x_k & w_k \end{bmatrix} \begin{bmatrix} P^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} \leq 1 \quad (2.59a)$$

$$\begin{bmatrix} x_k & w_k \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} \leq 1. \quad (2.59b)$$

By means of the S-procedure, the enforced condition for robust positive invariance (2.58) when (2.59a) and (2.59b) holds if  $\exists \tau_1, \tau_2 \in \mathbb{R}$  such that:

$$\begin{bmatrix} A_{cl_i} P^{-1} A_{cl_i} & A_{cl_i}^T P^{-1} E_i \\ E_i P^{-1} A_{cl_i} & E_i^T P^{-1} E_i \end{bmatrix} \preceq \tau_1 \begin{bmatrix} P^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \tau_2 \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (2.60)$$

for  $\tau_1 \geq 0$ ,  $\tau_2 \geq 0$  and  $\tau_1 + \tau_2 \leq 1$  with  $i = 1, \dots, n_v$ .

Reordering terms,

$$\begin{bmatrix} \tau_1 P^{-1} & \mathbf{0} \\ \mathbf{0} & \tau_2 \mathbf{I} \end{bmatrix} - \begin{bmatrix} A_{cl_i} \\ E_i \end{bmatrix} P^{-1} \begin{bmatrix} A_{cl_i} & E_i \end{bmatrix} \preceq 0, \quad (2.61)$$

and using the Schur complement, we finally obtain the equivalent BMI robust invariance criterium for the construction of the RPI ellipsoidal set:

$$\begin{bmatrix} \tau_1 P^{-1} & \mathbf{0} & A_{cl_i} \\ \mathbf{0} & \tau_2 \mathbf{I} & E_i \\ A_{cl_i} & E_i & P^{-1} \end{bmatrix} \succeq 0, \quad (2.62)$$

with  $i = 1, \dots, n_v$ .

By exploiting the worst case combination of scalar variables,  $\tau_1 + \tau_2 = 1 \rightarrow \tau = \tau_2 = 1 - \tau_1$ , the nonlinearity on the scalar values is eliminated, so one scalar variable can be dropped:

$$\begin{bmatrix} (1 - \tau)P & \mathbf{0} & P A_{cl_i}^T \\ \mathbf{0} & \tau \mathbf{I} & E_i^T \\ A_{cl_i} P & E_i & P \end{bmatrix} \succeq 0, \quad (2.63)$$

for  $0 < \tau < 1$  and  $i = 1, \dots, n_v$ .

Finally, this invariance condition can be exploited to compute the *maximal* or *minimal* robust invariant set by means of the maximization or minimization of the  $\text{trace}(P)$  subject to the system constraints in the LMI form [Nguyen 2014], [Luca 2011b].

## 2.3 Control design for LPV systems

Now that the most relevant theoretical tools are set, let's consider the problem of regulating to the origin a LPV system in the form (2.2) subject to polytopic constraints on the state, output and input (2.16a) - (2.16b).

We are going to consider three different control approaches: the first method is based on the definition of a linear static state-feedback or a parameter-dependent feedback that defines a common or parameter-dependent quadratic Lyapunov function for the LPV closed-loop system dynamics. In a second methodology, the domain of attraction of such LPV controller is enhanced by means of a receding horizon strategy, using a classical Model Predictive Control strategy whose stability and recursive feasibility is based on the previous LPV design. Finally, we look at the Interpolation Based Control, where a novel convex optimization-based control action is used to stabilize the system dynamics.

### 2.3.1 LPV Control Design

In the following, we recall the classical LPV control design in the LMI framework. This is a mature technique, based on the computation of a parameter-varying stabilizing state feedback gain. The exposition is organized in an incremental way, where we start from the basic constrained case, going through an enhanced constrained contractive control and ending up with the case of an observer-controller structure.

#### Classical LPV control design

Lyapunov stability theory has been largely used in classical control applications to proof stability of closed-loop system dynamics. This theory is based on the study of existence of Lyapunov functions describing or bounding the closed-loop dynamics of the system object of study.

In [de Oliveira 1999] a stability condition that allows to check the stability of discrete-time systems with polytopic uncertainties is presented. This approach is based on the computation of a common Lyapunov function, suitable for the full range of variation of the parameter. A direct application of these results is the computation of a parameter-dependent stabilizing gain for the constrained system: in the following we focus on the design for the unperturbed discrete parameter-dependent closed-loop system i.e.  $w_k = 0$ ,

$$x_{k+1} = A_k(\theta_k)x_k + Bu_k(\theta_k), \quad (2.64)$$

with  $A_k(\theta_k) = \sum_{i=1}^{n_v} \lambda_i A_i(\theta_i)$  and the parameter-dependent control law

$$u_k(\theta) = \sum_{i=1}^{n_v} \lambda_i K_i(\theta_i)x_k. \quad (2.65)$$

The controller design builds on the guaranteed decrease of the Lyapunov function, i.e.,

$$x_k^T P^{-1} x_k - x_{k+1}^T P^{-1} x_{k+1} \geq 0, \quad \text{with } i = 1, \dots, n_v. \quad (2.66)$$

Now, substituting the system dynamics and considering all the system realizations for  $i = 1 \dots n_v$ ,

$$P^{-1} - (A_i + BK_i)^T P^{-1} (A_i + BK_i) \succ 0. \quad (2.67)$$

By using the Schur complement, this condition can be rewritten as

$$\begin{bmatrix} P^{-1} & (A_i + BK_i)^T P^{-1} \\ P^{-1}(A_i + BK_i) & P^{-1} \end{bmatrix} \succeq 0. \quad (2.68)$$

Finally, by pre- and post-multiplication with the symmetric and full rank matrix,  $blkdiag(P)$  we obtain the condition for an ellipsoid  $\mathcal{E}(P)$  to be positive invariant,

$$\begin{bmatrix} P & P(A_i + BK_i)^T \\ (A_i + BK_i)P & P \end{bmatrix} \succeq 0. \quad (2.69)$$

Condition (2.69) contains bilinear terms issued from the products of  $K_i$  and  $P$ , and thus it is nonlinear. Nevertheless, it can be easily linearized by defining

$$Y_i = K_i P \quad (2.70)$$

and solving the LMIs,

$$\begin{bmatrix} P & PA_i^T + Y_i^T B^T \\ A_i P + BY & P \end{bmatrix} \succeq 0, \quad \text{with } i = 1, \dots, n_v. \quad (2.71)$$

The set of state constraints  $X = \{x \in \mathbb{R}^{n_x} : H_X x \leq g_X\}$  is symmetric and approximated by an ellipsoid for all states  $i = 1, \dots, n_x$ ,  $(H_{X_i}/g_{X_i})x \leq 1 \Leftrightarrow F_{i_x} x \leq 1$ . This kind of constraints can be captured by the following LMI for each state constraint  $F_{i_x}$ .

$$\begin{bmatrix} 1 & F_{i_x} P \\ P F_{i_x}^T & P \end{bmatrix} \succeq 0. \quad (2.72)$$

In the case of input constraints, we consider symmetric limits,  $-u_{min} = u_{max} = u_{max}$ . Then, we have one input constraint of the form  $-u_{i_{max}} \leq K_i \bar{x} \leq u_{i_{max}}$ . This is expressed in the LMI form as follows

$$\begin{bmatrix} u_{i_{max}}^2 & K_i P \\ P K_i^T & P \end{bmatrix} \succeq 0. \quad (2.73)$$

Performing again the linearizing change of variable (2.70) to formulate the constraints in a linear form,

$$\begin{bmatrix} u_{i_{max}}^2 & Y_{i_j}^T \\ Y_{i_j} & P \end{bmatrix} \leq 0, \quad \text{with } i = 1, \dots, n_v, \quad j = 1, \dots, n_x. \quad (2.74)$$

Solving the following optimization problem with the objective of enlarging the domain of attraction of the controller  $\max \text{trace}(P^{-1})$ , we obtain a control design that ensures asymptotic stability for all the states inside a quadratic domain of attraction for all the range of parameters and in the absence of additive disturbances with  $K_i(\theta_i) = Y_i P^{-1}$ . Furthermore, in the presence of (bounded) additive disturbances, it will ensure that the states remain in a neighborhood of the origin upon a input-to-state stability argument.

**Problem 2.1** *Classical LPV control design*

$$\min_{P, Y_i} \text{trace}(P) \quad (2.75)$$

subject to:

- Invariance condition (2.71). (2.76a)

- State constraints (2.72). (2.76b)

- Input constraints (2.74). (2.76c)

**Remark 2.7** *The LMIs are affine in the parameter through the affine dependence of  $A(\theta)$  on  $\theta$ . Since  $B$  is independent of  $\theta$ , the overall set of LMIs depend affinely on  $\theta$  in the same manner if we introduce an affine dependence of  $K$  and  $P$  in the parameter.*

Overall, we solve simultaneously the LMIs for each extreme value of the parameter range, allowing the controller to be different for each extreme realization of the system dynamics.  $\square$

After this development, in [Daafouz 2001] necessary and sufficient conditions for the computation of a parameter-varying Lyapunov functions which are quadratic on the system state and depend affinely on the uncertain parameter are presented.

**Theorem 2.1** [Daafouz 2001]. *The system  $x_{k+1} = A(\theta_k)x_k + Bu_k$  is poly-quadratically stable if and only if there exists symmetric positive definite matrices  $S_i$ ,  $S_j$  and  $G_i$  of appropriate dimensions such that*

$$\begin{bmatrix} G_i + G_i^T - S_i & G_i^T A_i^T \\ A_i G_i & S_j \end{bmatrix} > 0, \quad (2.77)$$

for all  $i = 1, \dots, n_v$  and  $j = 1, \dots, n_v$ . Where the parameter varying Lyapunov function is given by

$$\mathcal{P}(\theta_k) = \sum_{i=1}^{n_v} \lambda_i S_i^{-1}(\theta_i). \quad (2.78)$$

■

Again, this theoretical development is exploited to compute a parameter-varying stabilizing gain that ensures the closed-loop stability of the system dynamics, building up from the guaranteed decrease of the Lyapunov function, with an extra degree of freedom.

**Theorem 2.2** [Daafouz 2001]. *The system  $x_{k+1} = A(\theta_k)x_k + Bu_k(\theta_k)$  is poly-quadratically stable by a parameter-dependent control law  $u_k(\theta_k) = \sum_{i=1}^{n_v} \lambda_i K_i(\theta_i)x_k$  if and only if there exists symmetric positive definite matrices  $S_i$ ,  $S_j$  and  $G_i$ ,  $R_i$  of appropriate dimensions such that*

$$\begin{bmatrix} G_i + G_i^T - S_i & G_i^T A_i^T + R_i^T B^T \\ A_i G_i + B R_i & S_j \end{bmatrix} > 0, \quad (2.79)$$

for all  $i = 1, \dots, n_v$  and  $j = 1, \dots, n_v$ . Where the parameter dependent Lyapunov function is given by (2.78) and

$$K_i = R_i G_i^{-1}. \quad (2.80)$$

■

### Enhanced LPV control design

The solution of problem (2.1) leads to an ellipsoid  $\mathcal{E}(P)$  that is positive invariant but may not be contractive. This means that the system trajectories will stay inside the defined positive invariant ellipsoid but may not converge to the origin. In order to enhance this last property, we can opt for defining condition (2.71) as a strict inequality, enforcing implicitly a certain contractivity of the closed-loop dynamics. A second option is to define explicitly a *contraction factor*, denoted by  $\gamma > 0$  in the following, that ensures the controller's performance in terms of exponential decay of the Lyapunov function by the right hand side in (2.81),

$$x^T P^{-1} x - x^T (A_i + BK_i)^T P^{-1} (A_i + BK_i)^T x \geq \gamma (x^T P^{-1} x), \quad \text{with } i = 1, \dots, n_v, \quad (2.81)$$

which can be reformulated as

$$(1 - \gamma)P^{-1} - (A_i + BK_i)^T P^{-1} (A_i + BK_i) > 0, \quad (2.82)$$

with  $i = 1, \dots, n_v$ , thereby,

$$\begin{bmatrix} (1 - \gamma)P & P(A_i + BK_i)^T \\ (A_i + BK_i)P & P \end{bmatrix} \succeq 0, \quad \text{with } i = 1, \dots, n_v, \quad (2.83)$$

that can be transformed in the linear contractive invariance LMI condition (2.70) for a fixed  $\gamma$ ,

$$\begin{bmatrix} (1 - \gamma)P & PA_i^T + Y_i^T B^T \\ A_i P + BY & P \end{bmatrix} \succeq 0, \quad \text{with } i = 1, \dots, n_v. \quad (2.84)$$

On top of this formulation, it may be of interest to impose the inclusion of priority directions in the state space defined by  $z_i$  inside the computed ellipsoid. This allows to increase the domain of attraction of the designed controller in certain directions of special interest. This means that the largest invariant ellipsoid will include the point  $\sigma z_i$ , where  $\sigma$  is a scaling factor on the predefined direction in the state space. This kind of requirement is translated into an ellipsoidal containment constraint,  $\sigma z_i^T P^{-1} \sigma z_i < 1$ , that can be converted into a LMI form using one more time the Schur complement:

$$\begin{bmatrix} 1 & \sigma z_i^T \\ \sigma z_i & P \end{bmatrix} \succeq 0. \quad (2.85)$$

With all this in mind, we are in the position of formulating a constrained optimization problem that will enlarge the size of the obtained ellipsoid.

**Problem 2.2** *Enhanced LPV control design*

$$\min_{P, Y_i, \sigma} \{trace(P) + \sigma\} \quad (2.86)$$

subject to:

- Invariance condition (2.84). (2.87a)

- State constraints (2.72). (2.87b)

- Input constraints (2.74). (2.87c)

- Directional enhancements (2.85). (2.87d)

**Remark 2.8** *It has to be taken into account that this strategy enhances the controller performance in terms of convergence rate, but the size of the Positive Controlled Contractive sets will be relatively reduced, thus a trade-off between these two features needs to be done when defining the contraction factor value  $\gamma$ .*  $\square$

**Additive disturbances: robust LPV control design**

The following approach is based on the design of a robust LPV control law that maximizes the domain of attraction of the controller, understood as the domain of the state-space where system constraints are fulfilled even in the presence of additive disturbances  $w_k$ .

This maximization of the domain of attraction is directly related with the existence of a maximal robust positive invariant set (Section 2.2.2.4): we can formulate a constrained optimization problem [Nguyen 2015], where the robust positive invariance condition (2.63) for  $A_{cl_i} = A_i + BK_i$ , with  $K_i = Y_i P$  is enforced together with the system constraints in a LMI form.

**Problem 2.3** *Robust LPV control design*

$$\min_{P, Y_i, \sigma} \{\sigma + trace(P)\} \quad (2.88)$$

subject to:

- *Invariance condition*

$$\begin{bmatrix} (1-\tau)P & \mathbf{0} & PA_i^T + Y_i^T B^T \\ \mathbf{0} & \tau I & E_j^T \\ A_i P + BY_i & E_j & P \end{bmatrix} \succeq 0, \quad (2.89)$$

for all  $i = 1 \dots n_v$ ,  $j = 1 \dots n_v$ ,  $0 < \tau < 1$  and  $K_i = Y_i P^{-1}$ .

- *Constraints satisfaction (2.72), (2.74).*

- *Directional enhancements (2.85).*

### Simultaneous design of LPV observer-based static feedback controller

When dealing with real dynamical systems, it is very common to find that all not the model states are measurable, thus generating the need to include an estimation strategy to cope with the unmeasured but observable states. Although this thesis work is mainly focused on the control design for LPV discrete-time systems, some attention has been driven to the simultaneous design of a LPV observer and feedback controller, by extending the results from [Luca 2011b] to the LPV class of systems arising in the automotive application. Nevertheless, interested readers on the subject are addressed to this previous reference as well as to [Davins-Valldaura 2017], where a deeper study on the estimation for automotive steering applications is presented.

**Theorem 2.3** [Luca 2011b]. *Let us consider a discrete-time LPV system with parametric polytopic uncertainty. If there exist  $G_i = G_i^T \succ 0$ ,  $P_i = P_i^T \succ 0$ ,  $Q_{G_i}$ ,  $Q_{P_i}$ ,  $Y_i$ ,  $Z_i$  of appropriate dimensions, with  $i, l = [1 \dots n_v]$  and  $\tau > 0$ ,  $\beta \geq 0$ ,  $\lambda \geq 0$  such that:*

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i} - G_i & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau G_i & 0 \\ \star & \star & \star & G_j \end{bmatrix} \succ 0, \quad (2.90)$$

$$\begin{bmatrix} P_i & 0 & \tau P_i & \bar{A}_i^T Q_{P_i}^T - \bar{C}^T Z_i^T \\ \star & \lambda I & 0 & [Q_{P_i} \bar{E}_i - Z_i E_v]^T \\ \star & \star & \tau P_i & 0 \\ \star & \star & \star & Q_{P_i}^T + Q_{P_i} - P_j \end{bmatrix} \succ 0, \quad (2.91)$$

$$\tau - \beta \geq 0, \tau - \lambda \geq 0, \quad i, j = [1 \dots n_v], \quad (2.92)$$

then the system is input-to-state stable in the case of bounded norm disturbances  $\|w\|_2 \leq 1$ . The feedback and observer gains that stabilize the system are defined by:

$$K_i = Y_i Q_{G_i}^{-1}, \quad L_i = Q_{P_i}^{-1} Z_i, \quad i = [1 \dots n_v]. \quad (2.93)$$

We can calculate  $F(\theta)$  and  $L(\theta)$  at any working point as a convex combination of the vertex realizations:

$$K(\theta_k) = \sum_{i=1}^{n_v} \mu_{i_k} K_i(\theta_i) \quad \text{and} \quad L(\theta_k) = \sum_{i=1}^{n_v} \mu_{i_k} L_i(\theta_i). \quad (2.94)$$

with  $\sum_{i=1}^{n_v} \mu_{i_k} = 1$ ,  $\mu_{i_k} \geq 0$ . ■

Again, we can formulate a constrained optimization problem that takes into account system constraints.

**Problem 2.4** *Robust observer-based LPV control design*

$$\min_{P, Y_i, Z_i, Q_{P_i}, Q_{G_i}} \text{trace}(P_i) \quad (2.95)$$

subject to:

- ISS condition (2.90), (2.91), (2.92). (2.96a)

- State constraints:  $\begin{bmatrix} 1 & F_x^T Q_{G_i}^T \\ Q_{G_i} F_x & Q_{G_i}^T + Q_{G_i} - G_i \end{bmatrix} \succeq 0.$  (2.96b)

- Input constraints:  $\begin{bmatrix} Q_{G_i}^T + Q_{G_i} - G_i & Y_i^T \\ Y_i & u_{max}^2 I \end{bmatrix} \succeq 0.$  (2.96c)

- Output constraints:  $\begin{bmatrix} Q_{G_i}^T + Q_{G_i} - G_i & Q_{G_i}^T \bar{C}_i^T \\ \bar{C}_i Q_{G_i} & y_{max}^2 I \end{bmatrix} \succeq 0,$  (2.96d)

with  $i = 1, \dots, n_v.$

**2.3.2 Model Predictive Control**

In the last decades, the control research community has driven attention to Model Predictive Control (MPC) or Receding Horizon Control (RHC) [Camacho 2013], [Rawlings 2008], [J.M.Maciejowski 2002]. This strategy has progressively become a mature field, gaining great importance on the advanced multivariable constrained control industrial applications [Hrovat 2012]. MPC is an optimization based control strategy, where the plant model is used to predict the system's behavior along a certain time window, denoted as *prediction horizon*,  $N$ . Such prediction is initialized at the current state of the system, and each time instant  $k$ , the controller computes an optimal control sequence according to a defined criterium. After that, only the first computed optimal control input for the current time  $u_k$  is applied to the plant, while the tail of the optimal sequence is discarded. Then, the time window is shifted and the full procedure is repeated at time  $k + 1$  in a receding horizon fashion, taking into account the new available plant information.

The main feature of this kind of control that makes it desirable for industrial applications is its inherent capability of considering system constraints in a finite optimization framework. This allows to offer the best possible performance of the controlled plant, while staying inside the safety bounds or the actuation limits.

There exist two main trends of MPC: Implicit and Explicit MPC. Briefly speaking, Implicit MPC formulates a finite horizon Optimal Control Problem (OCP) which is solved online each time step [J.M.Maciejowski 2002].

Explicit MPC [Alessio 2009], [Bemporad 2002] computes offline the optimal control strategy for the full range of the admissible state-space, defining a control piecewise affine function when the system and constraints are linear. After that, the online computation is reduced to a point location problem. This kind of problem has an exponential complexity growth with respect to the number of states of the model

and the prediction horizon length, which prevents the application of this kind of MPC for large scale systems. Nevertheless, recent research work on the algorithms for the exploration of the regions [Ahmadi-Moshkenani 2016] and cell complexity reduction [Munir 2016] is working towards the attenuation of this major drawback of explicit MPC.

A second classification can be made depending on the model that is used for the plant behavior prediction: linear MPC deals with linear plant models (even though the dynamics of the closed-loop system is nonlinear due to the presence of constraints), which provide convex quadratic programs for the implicit formulations. Nonlinear MPC is focused on the study of systems with nonlinear dynamics, which generally yield non-convex and/or nonlinear optimization problems.

In this work, we have focused on the application of implicit linear MPC for the class of LPV systems, due to the LPV nature of the vehicle dynamics and the control architecture.

### Implicit MPC formulation for LPV systems

In the following, the formulation of a linear MPC for a given LPV system in the form of (2.2) is introduced. As it has been stated, this kind of controller is based on the resolution of a finite horizon optimal control problem, which is characterized by a cost function and a group of constraints.

Whenever the control objective is to stabilize the LPV system dynamics to the origin, the formulated cost function penalizes the deviation of the states together with the control signal along the prediction horizon  $N$ . Using quadratic norms, it takes the form

$$\min_U J(x_k, U) = \|x_N\|_P^2 + \sum_{k=1}^{N-1} \|x_k\|_Q^2 + \|u_k\|_R^2, \quad (2.97)$$

where  $Q \in \mathbb{R}^{n_x \times n_x}$ ,  $Q \succeq 0$  and  $R \in \mathbb{R}^{n_u \times n_u}$ ,  $R \succ 0$  represent respectively the positive definite state and input weighting matrices and  $\|x_N\|_P^2$  is the cost-to-go related to the controller stability guarantees that will be further detailed in the end of this section.

Next, in order to reformulate the control problem in a more convenient way, we express (2.97) in matrix form

$$J_k = X^T \tilde{Q} X + U^T \tilde{R} U, \quad (2.98)$$

where  $\tilde{Q} \in \mathbb{R}^{Nn_x \times Nn_x}$ ,  $\tilde{R} \in \mathbb{R}^{(N-1)n_u \times (N-1)n_u}$  represent all the weighting matrices along the prediction horizon and  $X \in \mathbb{R}^{Nn_x \times 1}$ ,  $U \in \mathbb{R}^{(N-1)n_u \times 1}$  are the vectors containing the sequence of predicted future states and inputs respectively,

$$X = [x_k^T, x_{k+1}^T, \dots, x_{k+N}^T]^T, \quad U = [u_k^T, u_{k+1}^T, \dots, u_{k+N-1}^T]^T. \quad (2.99)$$

The input vector  $U$  remains as the vector of optimization variables, while the state prediction vector  $X$  needs to be computed by the propagation of the system dynamics along the prediction horizon under the assumption of constant parameter

profile,

$$\begin{aligned}
x_{k+1} &= A(\theta_k)x_k + Bu_k + E(\theta_k)w_k \\
x_{k+2} &= A(\theta_k)x_{k+1} + Bu_{k+1} = A(\theta_k)^2x_k + A(\theta_k)Bu_k + Bu_{k+1} + \\
&\quad + A(\theta_k)E(\theta_k)w_k + E(\theta_k)w_{k+1} \\
&\quad \vdots \\
x_{k+N} &= A(\theta_k)x_{k+N-1} + Bu_{k+N-1}E(\theta_k)w_{k+N-1} = A(\theta_k)^Nx_k + A(\theta_k)^{N-1}Bu_k \\
&\quad + \dots + Bu_{k+N-1} + A(\theta_k)^{N-1}E(\theta_k)w_k + \dots + E(\theta_k)w_{k+N-1}.
\end{aligned} \tag{2.100}$$

In a compact matrix form:

$$X = \Psi(\theta_k) x_k + \Theta(\theta_k) U + \Gamma(\theta_k) W \tag{2.101}$$

with:

$$\Psi(\theta_k) = [ A(\theta_k) \quad A(\theta_k)^2 \quad \dots \quad A(\theta_k)^N ]^T, \tag{2.102}$$

$$\Theta(\theta_k) = \begin{bmatrix} B & 0 & \dots & 0 \\ A(\theta_k)B & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A(\theta_k)^{N-1}B & A(\theta_k)^{N-2}B & \dots & B \end{bmatrix}, \tag{2.103}$$

$$\Gamma(\theta_k) = \begin{bmatrix} E(\theta_k) & 0 & \dots & 0 \\ A(\theta_k)E(\theta_k) & E(\theta_k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A(\theta_k)^{N-1}E(\theta_k) & A(\theta_k)^{N-2}E(\theta_k) & \dots & E(\theta_k) \end{bmatrix}, \tag{2.104}$$

$$W = [ w_k \quad w_{k+1} \quad \dots \quad w_{k+N-1} ]^T. \tag{2.105}$$

**Remark 2.9** *The matrix  $W$  represents the prediction of the additive disturbances acting on the system in the future  $N$  steps. If the additive disturbances dynamical model is known, such prediction can be made by the propagation of the disturbances model along the horizon. If not, common standard assumptions are the constant additive disturbance  $w_{k+i} = w_k$ , for  $i = 1 \dots N-1$  or white noise modeling [Camacho 2013].*  $\square$

Substituting (2.101) in (2.98) and gathering together the terms depending on the optimization variable, a quadratic cost function is obtained:

$$\arg \min_U J_k = \arg \min_U \frac{1}{2} U^T H(\theta_k) U + F(\theta_k) U, \tag{2.106}$$

with  $H(\theta_k) = \Theta(\theta_k)^T \tilde{Q} \Theta(\theta_k) + \tilde{R}$ ,  $F(\theta_k) = 2 [x_k^T \Psi(\theta_k)^T + W^T \Gamma(\theta_k)] \tilde{Q} \Theta(\theta_k)$ .

Once this is set, input, state and output constraints can be included on the problem formulation, but they need to be reformulated in terms of the optimization

variable  $U$ . As an illustrative example, let us consider box constraints on the system state vector  $x \in X$  (2.16a). If we enforce such constraints on the state along the prediction horizon we obtain,

$$\begin{bmatrix} x_{min} \\ x_{min} \\ \vdots \\ x_{min} \end{bmatrix} \leq \begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+N} \end{bmatrix} \leq \begin{bmatrix} x_{max} \\ x_{max} \\ \vdots \\ x_{max} \end{bmatrix}. \quad (2.107)$$

Considering the prediction model (2.101) and reordering terms,

$$\underbrace{\begin{bmatrix} \Theta(\theta_k) \\ \Theta(\theta_k) \end{bmatrix}}_{G_x} U \leq \underbrace{\begin{bmatrix} x_{max} \\ \vdots \\ x_{max} \end{bmatrix}, - \begin{bmatrix} x_{min} \\ \vdots \\ x_{min} \end{bmatrix}}_{W_x}^T + \underbrace{\begin{bmatrix} -\Psi(\theta_k) \\ \Psi(\theta_k) \end{bmatrix}}_{T_x} x_k. \quad (2.108)$$

By concatenating the corresponding matrices  $G_\star$ ,  $T_\star$ ,  $W_\star$  for the input, output and states, we define a polyhedral feasible domain for the constrained problem,

$$GU \leq Tx_k + W. \quad (2.109)$$

Finally, the finite horizon optimal control problem consists of solving a quadratic programming (QP) in order to minimize the control input along the prediction horizon subject to linear inequality constraints.

**Problem 2.5** *MPC optimization problem*

$$U^* = \min_U \frac{1}{2} U^T H(\theta_k) U + F(\theta_k) U \quad (2.110)$$

subject to:

$$GU \leq Tx_k + W \quad (2.111a)$$

$$H(\theta_k) = \Theta(\theta_k)^T \tilde{Q} \Theta(\theta_k) + \tilde{R} \quad (2.111b)$$

$$F(\theta_k) = 2 [x_k^T \Psi(\theta_k)^T + W^T \Gamma(\theta_k)] \tilde{Q} \Theta(\theta_k) \quad (2.111c)$$

$$x_N \in \Omega \quad (2.111d)$$

Procedure 2.8 summarizes the Implicit MPC algorithm for LPV systems,

**Procedure 2.8** *Implicit Model Predictive Control*

1. Read measurement of current state  $x_k$  and parameter value  $\theta_k$  and update problem formulation.
2. Solve QP (2.110) subject to (2.111a) - (2.111d).
3. Obtain  $U^*$ .
4. Apply first optimal control  $U^*(1)$  to the plant.
5. Shift horizon one step forward.
6. Go to step 1.

**Remark 2.10** *Normally, the parameter trajectories are unknown a priori along the prediction horizon. A common way of proceeding is to fix its value according to the last available measurement for the propagation of the system dynamics model (2.101).*  $\square$

### Robust feasibility and stability

In a nutshell, the proof for stability of a finite-horizon MPC strategy is based on the use of a Lyapunov cost-to-go function [Mayne 2000], having a strictly decreasing cost function and ensures that the state and input converge to the origin. In addition, recursive feasibility is shown by means of proving that the feasibility of the optimization problem at one sample time implies feasibility at the next time step. This is handled by the definition of a robustly stabilizing controller for the unconstrained system, which defines the region of attraction of the origin, denoted *terminal set*, where the state  $x_N$  at the end of the prediction horizon is enforced to lay. Such sets can be defined within the class of polyhedral or ellipsoidal robustly terminal sets.

A first survey on the subject was introduced by [Bemporad 1999]. Then, an important study with a strong theoretical contribution was presented by [Kothare 1996], where a robust MPC controller for LPV systems with polytopic uncertainty is designed based on the online computation of a linear feedback controller that minimizes a worst-case infinite horizon objective function in the LMI framework. After this, [Cuzzola 2002] reduced the conservatism of the previous approach by using parameter-varying Lyapunov functions on the design, instead of a common one. In [Mao 2003] some initially incorrect proofs from [Cuzzola 2002] were rectified. Finally, [Wada 2004] introduced an additional degree of freedom in the LMI formulation, going one step forward in reducing such conservatism.

The major drawback of the previous approaches is induced by the computational online workload, coming from the calculation of the terminal invariant ellipsoid each time step. This tried to be attenuated in [Wan 2003], where the major part of the process effort is moved to the offline part: the basic idea is to compute a series of robust positive invariant sets for different initial states and the associated controllers beforehand, and store them in a lookup table. Then, the online part is in charge of determining which is the smallest of the computed sets that contains the current state and apply the associated optimal control. Although this approach reduces considerably the online computation effort, it still has a major disadvantage, which comes from the selection of the initial states for which the invariant sets are computed in advance.

These major problems on this kind of approaches drive the discussion to a recent work on the subject, presented by [Di Cairano 2015], where the terminal cost is based on the construction of a parameter-dependent Lyapunov function, following the lines of [Daafouz 2001]. This allows to achieve a finite-time receding-horizon formulation, by means of enforcing a cost decrease leading to a parameter-varying terminal cost and a stabilizing control law for the unconstrained system. The LMI

designed to obtain a parameter-dependent terminal cost for  $\theta \in \Theta$  is recalled in the following. We point the interested readers to [Di Cairano 2015] for further discussions on the subject and proof for Lemma 2.1.

**Lemma 2.1** *Let the symmetric positive definite matrices  $S_i$ ,  $S_l$  and  $G_i$ ,  $Y_i$ ,  $Y_l$  of appropriate dimensions, be such that*

$$\begin{bmatrix} G_i + G_i^T - S_i & (A_i G_i + B Y_i)^T & Y_i^T & G_i^T \\ (A_i G_i + B Y_i) & S_l & 0 & 0 \\ Y_i & 0 & R^{-1} & 0 \\ G_i & 0 & 0 & Q^{-1} \end{bmatrix} > 0 \quad (2.112)$$

subject to:

- State constraints:  $\begin{bmatrix} 1 & F_x^T Q_{G_i}^T \\ Q_{G_i} F_x & Q_{G_i}^T + Q_{G_i} - G_i \end{bmatrix} \succeq 0.$  (2.113a)

- Input constraints:  $\begin{bmatrix} Q_{G_i}^T + Q_{G_i} - G_i & Y_i^T \\ Y_i & u_{max}^2 I \end{bmatrix} \succeq 0.$  (2.113b)

for all  $i = 1, \dots, n_v$  and  $l = 1, \dots, n_v$ . Then  $G_i$  is full rank and the parameter-dependent Lyapunov function is given by  $P(\theta) = \sum_{i=1}^{n_v} \lambda_i P_i(\theta_i)$ , with  $P_i = S_i^{-1}$ , and the parameter-dependent stabilizing gain by  $K_i = Y_i G_i^{-1}$  which yields the stabilizing parameter dependent control law  $u(\theta) = \sum_{i=1}^{n_v} \lambda_i K_i(\theta_i) x_k$ .  $\square$

For ensuring stability, the obtained parameter-dependent Lyapunov function is used for the terminal cost formulation in the MPC design, adjusting each time step its value by an appropriate convex combination of the extreme values depending on the parameter value. In addition, a family of terminal sets is obtained for each one of the controllers following the lines presented in Section 2.2.

**Remark 2.11** *The number of degrees of freedom introduced in the problem through the matrices  $G_i$ ,  $S_i$ ,  $S_l$  can be again adapted to obtain a common Lyapunov function for all the parameter values, denoted  $P$ , by imposing  $S_i = S_l = S$  and  $G_i = G$  or a parameter-dependent one  $P(\theta)$ , that will change depending on the current measured value of the parameter.*  $\square$

### 2.3.3 Interpolation based control

This section introduces the main idea of the Interpolation Based Control (IBC), a relatively novel constrained control approach whose online computational load is considerably lower than the MPC method [Nguyen 2014]. The main idea on this strategy is to interpolate between two control action extreme values. Each time, two linear programming problems are solved, which makes this approach to be a suitable trade-off between performance and computational cost. Moreover, this control strategy is of interest for its ability to treat problem constraints and actuator limits in a systematic manner, as well as its capability to include changes of the behavior on a LPV system. This makes this controller a compelling candidate for embedded control applications.

### IBC formulation

The interpolation based control strategy is built on the fact that any state  $x_k$  which belongs to the controlled invariant set at a given time  $k$  can be expressed as a linear convex combination of two sub-states,  $x_{c_k}$  and  $x_{o_k}$ ,

$$x_k = \eta_k x_{c_k} + (1 - \eta_k) x_{o_k}, \quad (2.114)$$

where  $0 \leq \eta_k \leq 1$ ,  $x_{c_k} \in \mathcal{C}_N$ ,  $x_{o_k} \in \Omega$ , with  $\mathcal{C}_N$  being a Controlled Invariant Set (CIS) and  $\Omega$  being the Maximal Positive Invariant Constrained Admissible set (Section 2.2.1).

In an analogous way, the control action that would be applied to such state  $x_k$ , can be obtained from a convex combination of the control action that would be applied to each one of the previously defined sub-states,

$$u_k = \eta_k u_{c_k} + (1 - \eta_k) u_{o_k}, \quad (2.115)$$

where  $u_{o_k}$  is obtained using the parameter-varying stabilizing feedback gain  $K(\theta_k)$  computed at the interior of the maximal output admissible set  $\Omega$  and  $u_{c_k}$  is computed by solving an optimization-based control problem which exploits the control-invariance characteristics of the CIS,  $\mathcal{C}_N$ .

Once the basis are set, the first step is to compute the coefficient  $\eta_k$  that defines the affine convex decomposition of the state and the input (2.114), (2.115). An optimization problem whose objective will be to minimize this coefficient is build. This will provide a control action that is as close as possible to the unconstrained local controller performance, even when we are out of the MAS,  $\Omega$ .

$$\eta_k^* = \min_{x_{c_k}, x_{o_k}, \eta_k} \{\eta_k\} \quad (2.116)$$

subject to

$$H_{\mathcal{C}} x_{c_k} \leq g_{\mathcal{C}}, \quad x_{c_k} \in \mathcal{C}_N. \quad (2.117a)$$

$$H_{\Omega} x_{o_k} \leq g_{\Omega}, \quad x_{o_k} \in \Omega. \quad (2.117b)$$

$$x_k = \eta_k x_{c_k} + (1 - \eta_k) x_{o_k}, \quad \text{State convex decomposition (2.114)}. \quad (2.117c)$$

$$0 \leq \eta_k \leq 1, \quad \text{Affine parameter.} \quad (2.117d)$$

To transform the non-linear optimization problem (2.116), (2.117a)-(2.117d) in a linear programming problem, the following change of variables can be done [Nguyen 2013]:

$$r_{c_k} = \eta_k x_{c_k}, \quad (2.118)$$

$$r_{o_k} = (1 - \eta_k) x_{o_k}. \quad (2.119)$$

We know that  $x_{c_k} \in \mathcal{C}_N$  so it follows that  $r_{c_k} \in \eta_k \mathcal{C}_N$ . Similarly,  $x_{o_k} \in \Omega$  so  $r_{o_k} \in (1 - \eta_k) \Omega$ , yielding

$$H_{\mathcal{C}} r_{c_k} \leq \eta_k g_{\mathcal{C}}, \quad (2.120)$$

$$H_{\Omega} r_{o_k} \leq (1 - \eta_k) g_{\Omega}. \quad (2.121)$$

All considered, the resulting LP is formulated,

**Problem 2.6** *Interpolation factor optimization problem*

$$\eta_k^* = \min_{r_{c_k}, \eta_k} \{\eta_k\} \quad (2.122)$$

subject to:

$$H_C r_{c_k} \leq \eta_k g_C, \quad r_{c_k} \in \eta_k \mathcal{C}_N. \quad (2.123a)$$

$$H_\Omega(x_k - r_{c_k}) \leq (1 - \eta_k)g_\Omega, \quad r_{o_k} \in (1 - \eta_k)\Omega. \quad (2.123b)$$

and  $r_{o_k} = x_k - r_{c_k}$  from (2.119), (2.114).

$$0 \leq \eta_k \leq 1, \quad \text{Affine parameter.} \quad (2.123c)$$

In order to enforce a certain contractiveness on the controller, constraint (2.123c) can be substituted for the non-strict case with  $\eta < 1$ .

Once the optimal interpolation factor  $\eta_k^*$  is known, the states  $x_{c_k}$  and  $x_{o_k}$  can also be obtained as a by-product by means of (2.118) and (2.119).

After these steps, the only missing element is the admissible input signal corresponding to  $x_{c_k}$ , that is, an input signal  $u_{c_k}$  that will keep a state that belongs to the border of  $\mathcal{C}_N$  inside it. A simple one-step linear programming problem constructs this control action by maximizing the contraction factor  $\gamma_k$  that the computed input will produce on they state.

**Problem 2.7** *IBC contraction factor*

$$\gamma_k^* = \min_{u_{c_k}, \gamma_k} \{\gamma_k\} \quad (2.124)$$

subject to:

$$H_C(A_i x_{c_k} + B u_{c_k}) \leq \gamma_k g_C, \quad \text{Recursive feasibility condition, } x_{k+1} \in \mathcal{C}_N. \quad (2.125a)$$

$$H_U u_{c_k} \leq g_U, \quad \text{Input constraints.} \quad (2.125b)$$

$$0 \leq \gamma_k, \quad \text{Positive contractiveness.} \quad (2.125c)$$

with  $i = 1, 2, \dots, n_v$ .

As it can be seen (Procedure 2.9), the interpolation based control method consists of a pair of linear programming problems with  $n + 1$  arguments, which are considerably simpler than the  $N$ -step MPC optimization (QP).

**Procedure 2.9** Interpolation Based Control.

1. Read measurement of current state  $x_k$ .
2. Compute  $\eta^*$ ,  $r_{c_k}$  from LP (2.122), (2.123a)-(2.123c).
3. Obtain  $r_{o_k} = x_k - r_{c_k}$ ,  $x_{c_k}$  (2.118) and  $x_{o_k}$  (2.119).
4. Solve LP (2.124), (2.125a)-(2.125c) to obtain  $\gamma^*$  and  $u_{c_k}$ .
5. Compute  $u_{o_k} = K(\theta_k)x_{o_k}$ .
6. Obtain interpolated control action  $u_k$  (2.115).

**Stability and feasibility**

The interpolation based control scheme ensures asymptotic stability for all initial states inside the controlled invariant set,  $\mathcal{C}_N$ , with the interpolation factor  $\eta_k$  playing the role of a Lyapunov function on  $\mathcal{C}_N \setminus \Omega$  [Nguyen 2013]. Let us consider a non-negative candidate Lyapunov function,

$$V(x_k) = \eta_k^*, \quad \text{for all } x_k \in \mathcal{C}_N \setminus \Omega, \quad (2.126)$$

for any  $x_k \in \mathcal{C}_N \setminus \Omega$ , we can decompose the state (2.114)

$$x_k = \eta_k^* x_{c_k}^* + (1 - \eta_k^*) x_{o_k}^*. \quad (2.127)$$

Similarly (2.115),

$$u_k = \eta_k^* u_{c_k}^* + (1 - \eta_k^*) u_{o_k}^*. \quad (2.128)$$

Going one step forward,

$$x_{k+1} = A_i x_k + B u_k, \quad i = 1, \dots, n_v, \quad (2.129)$$

which corresponds to

$$x_{k+1} = \eta_k^* x_{c_{k+1}}^* + (1 - \eta_k^*) x_{o_{k+1}}^* \quad (2.130)$$

where

$$x_{c_{k+1}}^* = A_i x_{c_k}^* + B u_{c_k}^* \in \mathcal{C}_N, \quad (2.131a)$$

$$x_{o_{k+1}}^* = A_i x_{o_k}^* + B u_{o_k}^* \in \Omega. \quad (2.131b)$$

Which means that  $\eta_k^*$  provides a feasible decomposition for  $x_{k+1}$ . In addition, by solving (2.122) s.t. (2.123a)-(2.123c), we can obtain a different and optimal solution for the decomposition,

$$x_{k+1} = \eta_{k+1}^* x_{c_{k+1}}^* + (1 - \eta_{k+1}^*) x_{o_{k+1}}^*. \quad (2.132)$$

It follows that  $\eta_{k+1}^* \leq \eta_k^*$  and  $V(x_k)$  is a non-increasing function and a Lyapunov function (in the weak sense because the inequality is non-strict). In addition, by exploiting the invariance condition in  $\mathcal{C}_N$ , we can ensure that the state reaches the set  $\Omega$  in finite time, or equivalently, there will be a finite time  $k$  when  $\eta_k^* = 0$ . Moreover,

once the state enters in  $\Omega$ , the optimization problem has a trivial solution, with  $\eta_k^* = 0$ , so the interpolated controller will turn out to be the local contractive controller  $u_k = u_{o_k}$ , thus the interpolation based controller will ensure asymptotic stability for all  $x \in \mathcal{C}_N$ . In addition, by exploiting the controlled-invariance properties of  $\mathcal{C}_N$ , it can be proven that the problem (2.124) subject to (2.125a)-(2.125c) is recursively feasible [Nguyen 2013]. This can be checked if  $\forall x_k \in \mathcal{C}_N$  and constrained input,  $x_{k+1} \in \mathcal{C}_N$ :

$$H_U u_k \leq g_U, \quad (2.133a)$$

$$x_{k+1} = A_i x_k + B u_k \in \mathcal{C}_N. \quad (2.133b)$$

Input constraints inequality (2.133a) can be easily checked,

$$\begin{aligned} H_U u_k &= H_U (\eta_k u_{c_k} + (1 - \eta_k) u_{o_k}) \\ &= \eta_k H_U u_{c_k} + (1 - \eta_k) H_U u_{o_k} \\ &\leq \eta_k g_U + (1 - \eta_k) g_U = g_U, \end{aligned} \quad (2.134)$$

and (2.133b),

$$\begin{aligned} x_{k+1} &= A_i x_k + B u_k \\ &= A_i (\eta_k x_{c_k} + (1 - \eta_k) x_{o_k}) + B (\eta_k u_{c_k} + (1 - \eta_k) u_{o_k}) \\ &= \eta_k (A_i x_{c_k} + B u_{c_k}) + (1 - \eta_k) (A_i x_{o_k} + B u_{o_k}). \end{aligned} \quad (2.135)$$

Since  $A_i x_{c_k} + B u_{c_k} \in \mathcal{C}_N$  and  $(A_i x_{o_k} + B u_{o_k}) \in \Omega \subseteq \mathcal{C}_N$ , it follows that  $x_{k+1} \in \mathcal{C}_N$ .

**Remark 2.12** *The optimization with  $\eta_k < 1$  has a domain of guaranteed feasibility restricted when the parameter  $\theta$  is uncertain within  $\Theta$ . More than that, the optimization should be solved without the hard constraint  $\eta_k < 1$  thus allowing in the virtue of the feasibility and constraints satisfaction, an enlargement of the domain of attraction with a design formulation which is independent of the parameter realization.*

□

## 2.4 Conclusion

This chapter has set up the general framework for the constrained control design that will be developed in the remaining chapters of this first part of the thesis manuscript.

We will see in the next chapters that the vehicle lateral dynamics model can be included into the class of LPV systems, introduced at the first Section of this introductory chapter. After that, Set Invariance theory has been presented, in order to set the appropriate framework that allows to proof important properties such as stability and recursive feasibility of the Model Predictive Control and Interpolation Based Control, included in the last section of the chapter, together with classical control design strategies for LPV systems.

# Lateral Dynamics Control

---

## Contents

---

<b>3.1</b>	<b>Problem formulation</b>	<b>49</b>
3.1.1	Autosteer constraints	53
3.1.2	Note on $y_{cam}$ and $x_m$	53
<b>3.2</b>	<b>Constrained Control for Lateral Dynamics</b>	<b>54</b>
3.2.1	Model Predictive Control for Lateral Dynamics	54
3.2.2	Interpolation-Based Control for Lateral Dynamics	56
<b>3.3</b>	<b>Simulation Results: MPC vs IBC</b>	<b>58</b>
3.3.1	MPC feasibility analysis	59
3.3.2	IBC feasibility analysis	59
3.3.2.1	Numerical simulation	60
<b>3.4</b>	<b>Conclusion</b>	<b>63</b>

---

The main application area which is investigated in this first part of the thesis work is the control of the lateral dynamics of the vehicle. In this chapter, we focus on lateral dynamics control in the absence of exogenous additive disturbances and particularly on how this kind of constrained system can be modeled.

Model Predictive Control (MPC) and Interpolation Based Control (IBC) approaches are studied on a similar model-based design framework. Both are seen as powerful tools due to their inherent capability of ensuring constraint satisfaction in a systematic manner from the design stage, as well as their capability to include changes of the behavior of the system dynamics when the speed of the vehicle changes.

Finally, a comparison based on the performance and applicability of both control strategies is provided, based on numerical simulation of a selected scenario.

## 3.1 Problem formulation

Vehicle lateral dynamics control generally refers to the group of applications whose objective is to regulate the driving path of the vehicle, generally actuating on the steering wheel as a human driver would.

Several auto-steering control systems have already been developed in the literature and industry. First realizations of this kind of systems had the main objective of enhancing vehicle's *safety*, like the case of Lane Keeping Assistance [Enache 2008]



$$A_c(v_x) = \begin{pmatrix} \frac{-(C_f l_f^2 + C_r l_r^2)}{I_z v_x} & \frac{(C_f l_f - C_r l_r)}{I_z} & \frac{-(C_f l_f - C_r l_r)}{I_z v_x} & 0 \\ 1 & 0 & 0 & 0 \\ \frac{(C_r l_r - C_f l_f)}{m v_x} & \frac{(C_f + C_r)}{m} & \frac{-(C_f + C_r)}{m v_x} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (3.2)$$

$$B_c = \begin{pmatrix} \frac{C_f l_f}{I_z} \\ 0 \\ \frac{C_f}{m} \\ 0 \end{pmatrix}, \quad C_c = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -x_m & 0 & 1 \end{pmatrix}.$$

Where  $x(t)$  is the state vector  $[\dot{\psi}, \psi_{rel}, \dot{y}_{CoG}, y_{CoG}]^T \in \mathbb{R}^{n_x}$ .  $u$  is the control input  $\in \mathbb{R}^{n_u}$  which represents the steering angle  $\delta_c$  and  $y \in \mathbb{R}^{n_y}$  holds by the vector of system outputs, that will be  $[\dot{\psi}, y_{cam}]^T$  (Table 1.1).

System matrices  $A_c(v_x)$ ,  $B_c$ , and  $C_c$  depend on several vehicle parameters:  $C_f$  and  $C_r$  denote the cornering stiffness of the front and rear wheels respectively. The distances from the front and rear axis to the center of gravity are  $l_f$  and  $l_r$ ,  $m$  stands for the vehicle mass and  $I_z$  for the total yaw moment of inertia (Section 1.2.3.1). In addition to this, the observation matrix  $C_c$  depends on  $x_m$  which is the longitudinal distance with respect to the followed target.

Finally, one of the main parameters appearing in (3.2) is the longitudinal speed of the vehicle  $v_x$  which is time-varying. As it has been stated in Section 1.2.1, the lateral controller cooperates with the longitudinal vehicle dynamics control, that is an independent system which is in charge of the vehicle's longitudinal speed, acceleration and distance with the preceding vehicle, if any. These longitudinal control variables are measured or estimated at each sample time by the longitudinal logic, and provided to the lateral control system, that has no influence over them. Thus, the speed of the vehicle enters on the lateral controller as an external parameter, bringing an uncertain LPV model. In addition, with respect the framework of the present application, the speed is bounded, as the speed range in which the autosteer system is active is defined a priori by the system requirements,  $v_x \in [1, 40][\text{km/h}]$ .

Moreover, we can define a new parameter as the inverse of the longitudinal speed of the vehicle  $\nu = 1/v_x$ , in such a way that the state matrix  $A_c(\nu)$  presents a linear dependence on this parameter.

**Remark 3.1** *Such dynamical model depending on the parameter  $\nu = 1/v_x$  has a singularity when the vehicle is standing still,  $v_x = 0$ . This aspect can present serious drawbacks that prevent this model to be used for low speed applications, like parking maneuvers. Nevertheless, the current application will be only active in a speed range that does not include such singularity, thus remaining appropriate for the stated purposes.*  $\square$

**Assumption 3.1** *All the vehicle parameters, except the longitudinal speed, are considered known and fixed during the simulation, so they do not introduce any uncertainty to the model.*  $\square$

In the following, discrete-time formulation is used. For doing so, the forward Euler discretization method based on a truncated Taylor series expansion with  $T_s = 0.01[s]$  which has been applied to the system state-space continuous-time representation matrices displayed in (3.1), under the assumption of constant parameter evolution along the sampling window:

$$\begin{aligned} A(\nu_k) &= \mathbf{I} + A_c(\nu)T_s, \\ B &= B_cT_s. \end{aligned} \quad (3.3)$$

All by considering the time-varying (at integer multiples of  $T_s$ ), bounded and measured characteristics of the speed yields a discrete-time LPV model (Section 2.1). In this respect, we can perform a polytopic decomposition of the system matrix, such that

$$x_{k+1} = A(\nu_k)x_k + Bu_k. \quad (3.4)$$

Where the  $A(\nu)$ ,  $B$  matrices are described by after discretization (3.3), and  $A(\nu)$  satisfies:

$$A(\nu_k) = \sum_{i=1}^{n_v} \lambda_k A_i(\nu_i), \quad (3.5)$$

with  $\sum_{i=1}^{n_v} \lambda_k = 1$ ,  $\lambda_k \geq 1$ .

### Augmented state formulation

It has been set that the steering controller objective is to stabilize the vehicle lateral dynamics with respect to the target vehicle. For doing so, the system will need to act on the steering angle within the admissible bounds and by avoiding aggressive inputs that makes abrupt maneuvers. This means that when performing optimal control, the desired behavior is not one that minimizes the steering angle, but minimizes its rate of variation.

In this lines, we adapt the dynamical model by an *augmented state* formulation that will take the input as a state and will allow us to formulate the system dynamics related to the rate of change of the input,  $\Delta u_k = u_k - u_{k-1}$ , known as *velocity form*, which has additional interesting properties from the offset free tracking point of view [Pannocchia 2001].

The new state vector is

$$\bar{x}_k = [x_k^T \ u_{k-1}^T]^T. \quad (3.6)$$

The system dynamics will be expressed as follows

$$\bar{x}_{k+1} = \bar{A}(\nu_k)\bar{x}_k + \bar{B}\Delta u_k, \quad (3.7)$$

$$(3.8)$$

with:

$$\bar{A}(\nu_k) = \begin{bmatrix} A(\nu_k) & B \\ 0 & 1 \end{bmatrix} \quad \bar{B} = \begin{bmatrix} B \\ 1 \end{bmatrix}, \quad (3.9)$$

where (3.5) holds for  $\bar{A}(\nu_k)$  too.

### 3.1.1 Autosteer constraints

Real systems are commonly subject to constraints, coming from physical limitations on the system actuation: the steering angle cannot exceed a certain value, at the same time, there is a turning limit and rate of variation that avoids producing too aggressive maneuvers. State constraints often arise due to safety or environment restrictions: the lateral offset must be within the limits to keep the vehicle inside the corresponding lane and the lateral speed must be restricted to avoid uncomfortable maneuvers for the passengers.

The control design methods considered next are appealing for this constraint handling capability, which allows to guarantee a correct performance while having an optimal performance pushing the system close to its limits without running any risks.

Table 3.1 – Trajectory Constraints Definition for Autosteer System

Magnitude	Range	Definition
$\dot{\psi}$	$[\pm 0.05]$	Heading angle derivative [rad/s]
$\psi_{rel}$	$[\pm 0.5]$	Heading angle [rad]
$\dot{y}$	$[\pm 0.2]$	Lateral speed [m/s]
$y_{CoG}$	$[\pm 0.3]$	Lateral position [m]
$u$	$[\pm 0.17]$	Steering angle [rad]
$\Delta u$	$[\pm 0.52]$	Steering angle variation rate [rad/s]

System constraints on the states and input are defined by the numerical limits shown in Table 3.1, that define the polytopic sets of bounded admissible states and inputs  $X \subset \mathbb{R}^{n_x}$  and  $U \subset \mathbb{R}^{n_u}$ ,

$$x \in X, \quad X = \{x \in \mathbb{R}^{n_x} : H_x x \leq g_x\}. \quad (3.10a)$$

$$u \in U, \quad U = \{u \in \mathbb{R}^{n_u} : H_u u \leq g_u\}. \quad (3.10b)$$

### 3.1.2 Note on $y_{cam}$ and $x_m$

Magnitude  $x_m$  and output  $y_{cam}$  appearing in the state-space representation of the autosteer system are not considered for the control design, as complete measurements of the states are assumed to simplify the study in the subsequent. Nevertheless, when performing a complete analysis of this application, we notice that not all the system states are measured, and this will trigger the need of observation techniques. This note intends to briefly clear up this two elements appearing in the dynamical model.

First, we have  $x_m$ , which is defined by the longitudinal distance between the ego and the target vehicle,

$$x_m = v_x t_{follow} + d_{stop}, \quad (3.11)$$

where  $t_{\text{follow}}$  and  $d_{\text{stop}}$  represent tuned values that specify, respectively, the following time between the vehicles and the minimum safety distance that must be kept between the vehicles when a complete stop is reached. This longitudinal distance appears on the system observability matrix,  $C_c$  (3.2), so it can be proved that the autosteer system dynamics will be observable only for certain range of  $x_m$ . In [Davins-Valldaura 2017] a complete study on the subject is done and we point the reader to the respective details underlying the model we further use in the present study.

Vehicle instrumentation's (Section 1.2.2) camera provides the measurement of  $y_{cam}$ , which corresponds to the lateral offset between the ego vehicle and the target. Nevertheless, we need to transform this lateral offset between the real vehicles into an offset between the ego vehicle and a virtual perfect following vehicle,  $y_{CoG}$  (Figure 3.1)

$$y_{CoG} = x_m \psi_{rel} + y_{cam}. \quad (3.12)$$

This transformation takes into account the longitudinal distance that exists between the vehicles, that produces a certain *space delay* that needs to be considered by means of the term  $x_m \psi_{rel}$ .

## 3.2 Constrained Control for Lateral Dynamics

Once the base model is set, we continue the exposition with the design of a MPC (Section 2.3.2) and IBC (Section 2.3.3) for the autosteering system. For the control design purposes, complete measurements of the states are assumed in both cases.

### 3.2.1 Model Predictive Control for Lateral Dynamics

The application of implicit Model Predictive Control is based on the theoretical tools that have been introduced in Section 2.3.2. The main elements that need to be investigated and tuned whenever studying the implementation of this control strategy are listed and analyzed here.

**COST FUNCTION CRITERION.** Whenever developing an optimization-based control, we need to set the system objectives and tune the controller accordingly to obtain a desired optimal performance. In this control application, we seek for a system that converges to the origin, which represents the reference *Perfect following vehicle*. With this in mind, we define a cost function that minimizes the quadratic norm of the states and the system's input,

$$J(k) = \|\bar{x}_N\|_P^2 + \sum_{i=1}^{N-1} \|\bar{x}_i\|_Q^2 + \|\Delta u\|_R^2. \quad (3.13)$$

Once the cost function is defined, we need to tune the state and input weighting matrices  $Q \succeq 0$  and  $R \succ 0$ . Firstly, state weighting matrix  $Q$  can be used to give convergence priority to certain states of interest over the rest. From this point of view, the first relevant state is  $y_{CoG}$ , as that will generate a control tuning that

eliminates the vehicle lateral offset with respect to the target vehicle in priority. In addition, the importance of the heading angle correction cannot be neglected. In order to show this, Figure 3.2 pictures two different initial conditions, one of them denoted as *favorable* initial conditions, and the other *non-favorable* ones, depending on the sign relationship between  $y_{CoG}$  and  $\psi_{rel}$ . When the system is activated in a favorable position, the system lateral offset will instantaneously start converging to the origin. However, when the system is activated in a *non-favorable* initial condition, the lateral position will not exhibit a monotone decrease before the sign of the heading angle is corrected, thus, this state will have a considerable influence on the generalized convergence of the system states and must be tuned accordingly.

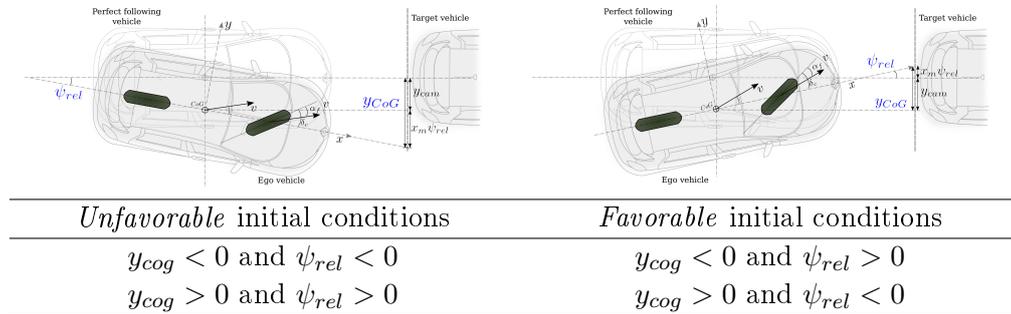


Figure 3.2 – Note on initial conditions

Then, we have input weighting matrix  $R$ , which is used to define the behavior of the controller and takes the form of scalar for the present model. If this value is set to a low constant, the control action is seen as *cheap*, so the optimization problem will tend to provide higher values in order to produce a fast and aggressive convergence on the states. On the contrary, if the control action has a higher cost, this takes the role of a penalty on the aggressiveness, and the convergence will be slower. As a consequence, a classical trade-off needs to be achieved along the tuning.

LPV PREDICTION MODEL. The autosteer system has been modeled by means of a LPV system, where the speed variation brings an uncertain model that will be described by a polytopic class of dynamics (3.7). Thanks to the online measurement of the parameter, the system dynamics will be updated at each sample time, using the available information (the current speed value). This means that the parameter value will be set as constant when propagating the system dynamics along the prediction horizon<sup>1</sup>.

SYSTEM CONSTRAINTS. State and input polytopic constraints described in Section 3.1.1 are included in the optimization problem. This allows to certify from the design stage that the optimal solution does not violate safety and physical limits of

1. If the future trajectory is known, and with it the future *desired* speed profile, one could make a more precise MPC formulation, computing the future system matrices using that information.

the autosteer system. The tuning of these values comes both from the actual physical limits and the expertise on the human perception when driving in an assisted vehicle.

**STABILITY AND FEASIBILITY.** Ensuring stability and recursive feasibility is based on the existence of a parameter-dependent stabilizing feedback gain in the neighborhood of the origin. Two different sets have been computed (Figure 3.3), corresponding to two different stabilizing control laws used to define the closed loop system (Section 2.3.2): on one hand, a common output admissible set, denoted by  $\Omega_P$ , can be obtained by considering the parameter-dependent stabilizing gain which is obtained with a common Lyapunov function  $P$  for the full range of the parameter variation. On the other hand, a parameter-dependent set,  $\Omega_{P(\nu)}$ , can be computed using a parameter-varying control based on the parameter-varying Lyapunov function  $P(\nu)$ . In the case of this second design, it is necessary to update each sample time the terminal set constraint  $x_N \in \Omega_{P(\nu)}$  to the last speed measurement.

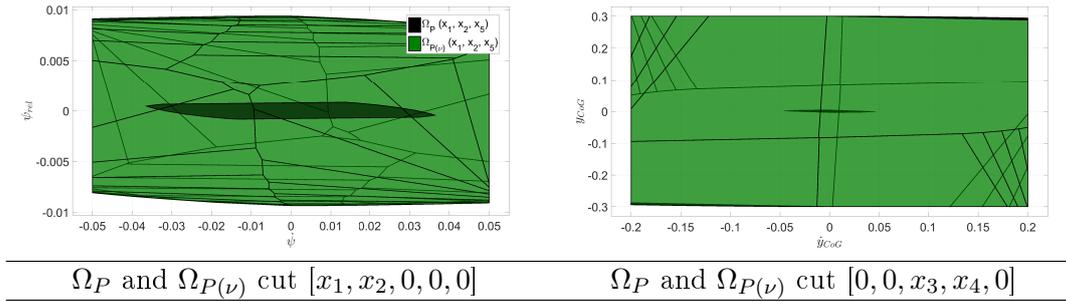


Figure 3.3 –  $\Omega_P$  and  $\Omega_{P(\nu)}$  comparison

Once all the elements are set, we are in the position of formulating an optimization problem in the lines of (2.110) subject to (2.111a)-(2.111d), which is solved online in a receding horizon fashion according to the classical implicit MPC formulation.

Before presenting the results, let us introduce an alternative design procedure which will allow a comparative study of the numerical simulations.

### 3.2.2 Interpolation-Based Control for Lateral Dynamics

Keeping the same control objective, the application of IBC introduced in Section 2.3.3 is studied. As it has been shown, the application of this kind of control is based on the decomposition of the current state into two sub-states, one belonging to the Maximal Controlled Invariant Set (CIS) and the other contained in the Maximal Admissible Set (MAS), providing a control action that is interpolated between the control values that would be applied to each one of these virtual states. As an initial step of the strategy, the design has to include the offline computation of the

corresponding invariant sets.

For the MAS set, we use the same ones we obtain when studying the system dynamics for the MPC design so it can be considered that both strategies share this element. In the case of the CIS (Section 2.2.1), its asymptotic iterative construction for the model of lateral dynamics leads to a relative complex set in  $\mathbb{R}^5$ , so the computational load becomes excessive. Due to this limitations, the 3-steps controlled invariant set,  $\mathcal{C}_3$  has been used in the present study as an approximation of the CIS (Figure 3.4).

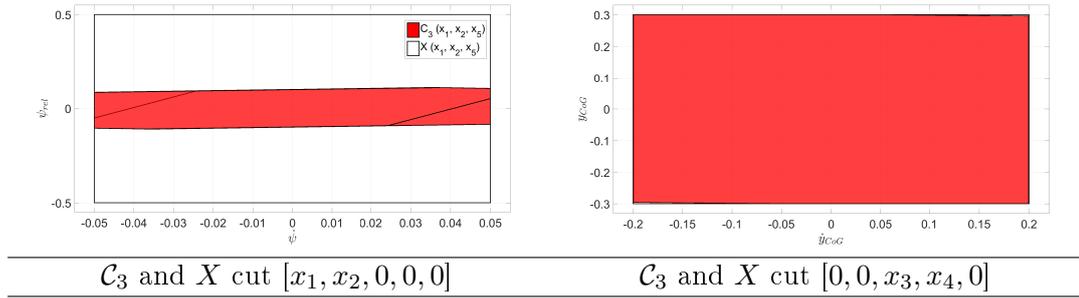


Figure 3.4 –  $\mathcal{C}_3$  and state admissible space  $X$

In order to have an illustration (and a relative measure), in Figure 3.5, the MAS  $\Omega_{P(\nu)}$  and the CIS  $\mathcal{C}_3$  for the LPV system are shown via cuttings together with the state admissible set  $X$  (3.10a). It can be seen that the limits in  $\dot{\psi}$ ,  $\dot{y}_{CoG}$  and  $y_{CoG}$  are close to the state constraints at the cutting values, while for  $\psi_{rel}$  the difference is relatively important, thus the initial conditions for this state will highly condition the feasibility of the control problem.

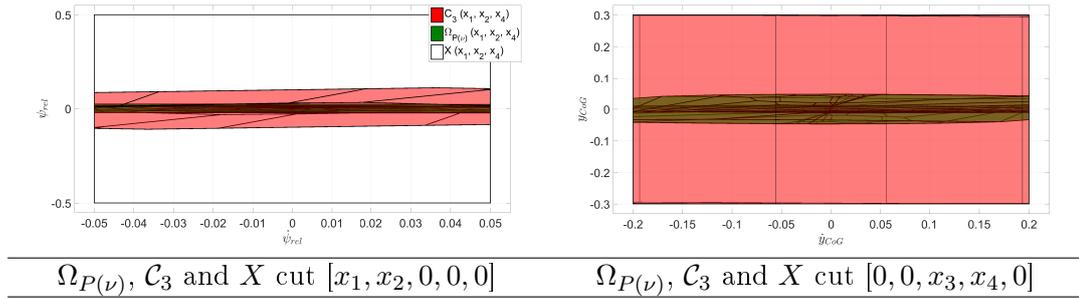


Figure 3.5 –  $\Omega_{P(\nu)}$ ,  $\mathcal{C}_3$  and state admissible space  $X$

### Enhanced feasibility scheme

From the construction of the invariant set in Figure 3.5, the strict inclusion of the CIS, or in the present case  $\mathcal{C}_N$ , in the satisfaction of the state constraints was obvious for the initial conditions inside these sets. However, the question arises on the treatment of the states that lay out of these boundaries, which would bring up

an unfeasible problem when solving (2.122), (2.123a)-(2.123c) in  $X \setminus \mathcal{C}_N$ . A simple way to avoid the infeasibility for the states out of  $\mathcal{C}_N$ , is to soften the constraints in order to contain the current state. The minimum scaled polyhedron  $\mathcal{C}'_N$  imposed on the interpolation factor that contains the current system state  $x_k$  is computed from the original  $\mathcal{C}_N$  solving the following LP:

$$\begin{aligned} \beta_k^* &= \min\{\beta\} \\ \text{s.t. } H_C x_k &\leq \beta b_C. \end{aligned} \quad (3.14)$$

Once  $\beta_k^*$  is known, the scaled set for the current iteration is defined  $\mathcal{C}'_N = \{x \in \mathbb{R}^n : H_{C'} x \leq b_{C'}\}$  with  $H_{C'} = H_C$  and  $b_{C'} = \beta_k^* b_C$ . This artifact ensures the feasibility in (2.124) s.t. (2.125a)-(2.125c). However the stability will be guaranteed only for  $\beta^* < 1$ , which is the guaranteed range of decrease independent of the parameter variation.

**Procedure 3.1** Interpolation Based Control with feasibility enhancements.

1. Read measurement of current state  $x_k$ .
2. If  $x_k \in \Omega$  compute  $u_{o_k} = K(\theta_{1_k})x_{o_k}$ . Set  $u_k = u_{o_k}$  and go to Step 8. else go to Step 3.
3. If  $x_k \in \mathcal{C}_N$  go to Step 4. else solve (3.14) s.t. (2.125a)-(2.125c) and compute  $\mathcal{C}'$ . Go to Step 4.
4. Compute  $\eta^*$ ,  $r_{c_k}$  from LP (2.122), (2.123a)-(2.123c). Obtain  $r_{o_k} = x_k - r_{c_k}$ ,  $x_{c_k}$  (2.118) and  $x_{o_k}$  (2.119)
5. Solve LP (2.124), (2.125a)-(2.125c) to obtain  $\gamma^*$  and  $u_{c_k}$ .
6. Compute  $u_{o_k} = K(\theta_{1_k})x_{o_k}$ .
7. Obtain interpolated control action  $u_k$  (2.115).
8. Return  $u_k$ .

**Remark 3.2** *There will be cases in which the current state is out of the computed  $\mathcal{C}_N$ , but only in a certain direction. The scaling technique is not always applicable, as (3.14) may produce a scaled polytope  $\mathcal{C}'$  that is out of the admissible state-space  $\mathcal{X}$ , and in these cases the intersection with the admissible set should be performed. These infeasibility issues are concerning the regions outside  $\mathcal{C}_N$  and it should be recalled that absolute guarantees of recursive feasibility can be obtained only within  $\mathcal{C}_\infty$ .*  $\square$

### 3.3 Simulation Results: MPC vs IBC

In this section the two controllers derived in Sections 3.2.1 and 3.2.2 are tested on a simulation basis. In order to present a relevant comparison for an usual configuration, a scenario with a trapezoidal profile of varying speed between the minimum and maximum values  $v_x \in [1, 40]$ [km/h] with the maximal longitudinal acceleration values<sup>2</sup> has been defined. The simulated vehicle will be running in a straight road

2.  $a_{max} = 4$ [m/s],  $a_{min} = -3$ [m/s]

and initialized at a perturbed initial state, simulating extreme conditions. All the results have been obtained using the open-source programming framework YALMIP [Lofberg 2005] and the optimization solver SEDUMI [Labit 2002].

### 3.3.1 MPC feasibility analysis

We have seen that the safe behavior of the controller is a critical feature when driving an application in which human life is involved. We want to ensure robust feasibility of the problem from the design stage. For doing so, we have to be able to steer the system state to the terminal set  $\Omega$  in  $N$  steps ahead in time, fulfilling all the system input and output constraints no matter the (bounded) speed of the vehicle.

This feature is mainly influenced by three arguments of the MPC design: first, the initial conditions  $x_0$ . Second, the length of the prediction horizon  $N$  and last, the size of the MAS,  $\Omega$ . If we start at initial conditions that are far away from the terminal set MAS, we will need to increase the prediction horizon, in order to get a feasible problem with  $x_N \in \Omega$ . At the same time, the size of the MAS will make this task easier to achieve at the price of an increase complexity of the respective polyhedral set within the optimization constraints.

In a first test the influence of the prediction horizon length and the terminal set constraint size has been analyzed. For doing that, we have run a simulation starting from the possible maximum values of the initial conditions  $x_0 = [0.05, 0.05, 0.2, 0.3, 0.15]^T$ , checking which would be the necessary prediction horizon length in order to derive a feasible optimization problem when the terminal set is either  $\Omega_P$  or  $\Omega_{P(\nu)}$ . With the first MPC choice, we have not been able to solve the problem, even considering a prediction horizon up to a prediction window of  $N = 500$  steps, which yields a problem without a practical meaning, due to the huge dimension of the optimization problem. On the other hand, the MPC controller with  $\Omega_{P(\nu)}$  provides a solution when  $N \geq 173$ . This is mainly due to the difference on the size of the terminal sets  $\Omega_P$  and  $\Omega_{P(\nu)}$ .

In addition to this, it has been showed in Figure 3.5 that the size of  $\Omega_{P(\nu)}$  in the  $\psi_{rel}$  dimension is very restrictive. That means that the domain of attraction for this particular state is relatively low, constraining the initial conditions for this angle at the 10% of its maximal value, 0.5[rad], being the most sensitive from the prediction horizon point of view.

**Remark 3.3** *It should be noted anyways that in practical implementations of the autosteer system,  $\psi_{rel}$  values remain small, reducing the possibility of worst case combinations of parameters and/or initial conditions, as those discussed from a theoretical perspective as limit cases herex.*  $\square$

### 3.3.2 IBC feasibility analysis

As a continuation of the study, IBC feasibility is analyzed. In this case the control technique is not affected by a prediction horizon but by the size and complexity of

the positive invariant sets used in the formulation.

This controller ensures feasibility to those points that belong to the original CIS,  $\mathcal{C}_{\mathcal{N}}$  (Section 2.3.3) or the scaled one  $\mathcal{C}'_{\mathcal{N}}$ . In order to check the closed loop behavior, we have performed several simulations for a grid of initial conditions, varying from  $[0, 0, 0, 0, 0]^T$  to the maximum values inside a  $\mathbb{R}^5$  box. Four different IBC controllers have been tested. In Table 3.2 it can be seen that the use of the bigger MAS,  $\Omega_{P(\nu)}$  together with the scaling technique for  $\mathcal{C}_{\mathcal{N}}$  (3.14) provides improved feasibility results.

Table 3.2 – Interpolation based control tests

MOAS	$\mathcal{C}_{\mathcal{N}}$	$N^\circ$ feasible $x_0$
$\Omega_{\bar{P}}$	Fixed	50.76%
	Scaled	57.36%
$\Omega_{P(\nu)}$	Fixed	80.24%
	Scaled	85.23%

### 3.3.2.1 Numerical simulation

In order to allow a detailed graphical analysis, a simulation case is presented together with the time-signals. The initial conditions have been fixed to  $x_0 = [0.025, 0.02, 0.15, 0.25, 0]^T$ , which represent a common situation in which the system would be activated. The speed profile with maximum acceleration and deceleration considerations used for the simulation, simulating aggressive variations on the parameter value. For the MPC, the prediction horizon has been fixed according to the minimal *Time-to-Collision* (TTC)<sup>3</sup>, that is generally fixed at 2[s], which ensures problem feasibility as well (Section 3.3.1). This corresponds to  $N = 200$ . Weighting matrices are set to

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and  $R = 1$ .

The first MPC design (using  $\bar{P}$  for the *terminal cost* formulation and  $\Omega_{\bar{P}}$  for the *terminal set* constraint) has a domain of attraction which makes the MPC policy unfeasible for this case study, being unable to solve the problem for such initial conditions. Even with  $N = 500$  the feasibility is not achieved, so it has been discarded from the test.

---

3. TTC is the time required for two vehicles to collide if they continue at their present speed and on the same path.

In Figures 3.7, 3.6 and 3.8, the time-trajectory for the MPC and IBC controller using  $P(\nu)$  for the *terminal cost* formulation and  $\Omega_{P(\nu)}$  for the *terminal sets* constraint are shown.

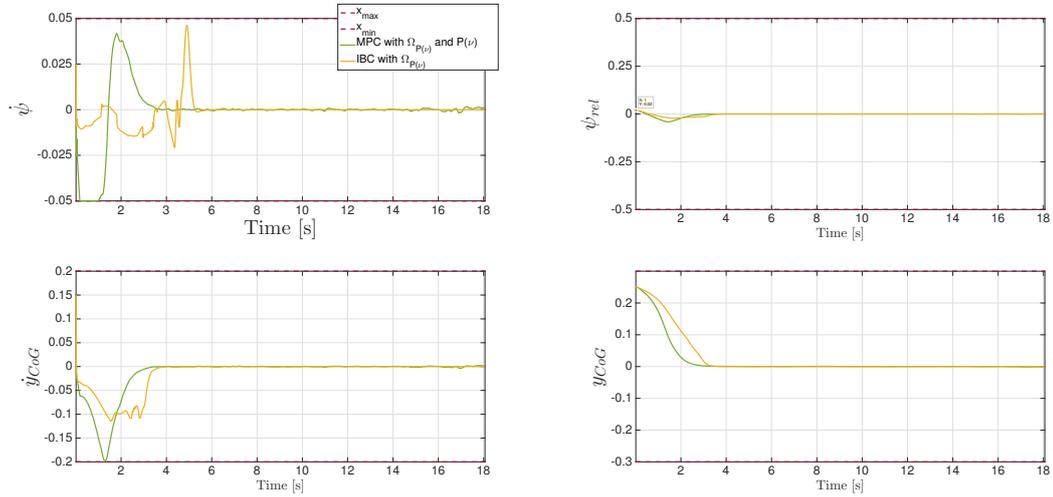


Figure 3.6 – System states trajectory

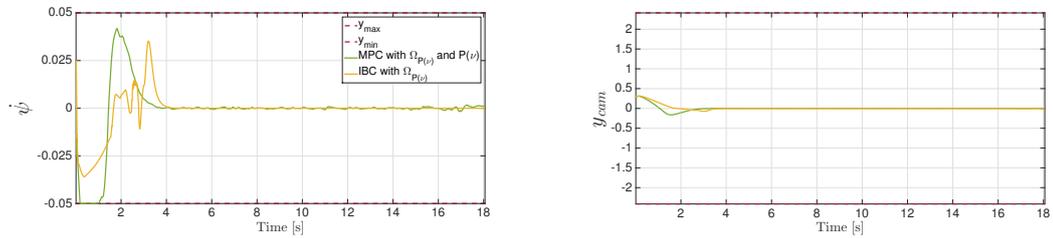


Figure 3.7 – System outputs trajectory

In Figure 3.9 it can be seen the interpolation parameter. Its positivity confirms the Lyapunov interpretation given in 2.3.3, the interpolation coefficient playing the role of a Lyapunov function. It is interesting to note as well that  $\eta_k^* = 0$  for all  $k \geq 245$ , implying that from that time instant, the state of the closed-loop system is in the invariant set  $\Omega$ . Note also the input constraint satisfaction, thus showing the ability of IBC to perform as a true active-constraint methodology. This is not the case for vertex control strategy which historically has the capability of handling a scaling of the maximal control invariant set as a basic element in the optimization-based design. Indeed, the vertex control has an active input control capability only on the boundary of the feasible set and doesn't use the full control capability in the interior of the feasible set.

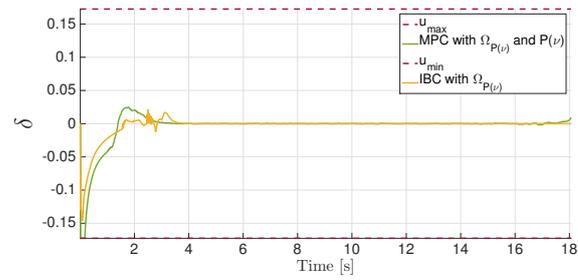


Figure 3.8 – System input trajectory

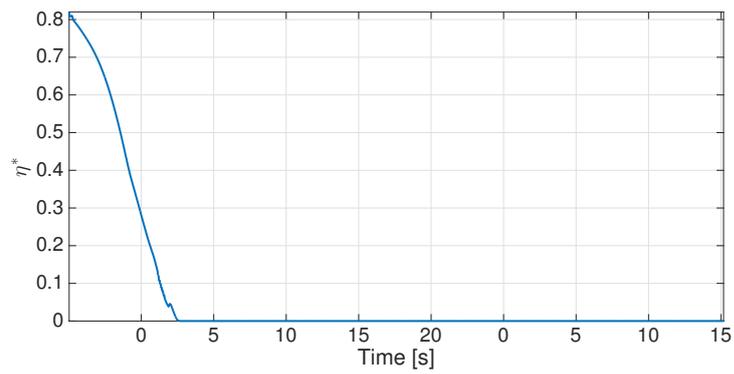


Figure 3.9 – IBC interpolation factor

## 3.4 Conclusion

Two different approaches for the design of autosteering target tracking control have been considered, when the system dynamics are described by a parameter-varying model where the parameter, speed of the vehicle, is bounded and measured.

On one hand, the MPC has been studied, starting from a recognized technique with a mature theoretical background. The online resolution of an optimization problem in the available embedded control units for the ADAS becomes nowadays feasible due to the computational power available for embedded control. However, the long prediction horizon remains the main bottle neck and the recent inverse optimality arguments [Nguyen 2014] offer the framework for compact representation of the same control law by an alternative low-complexity multi-parametric optimization.

As an alternative optimization-based control, the IBC proposes a computational complexity on a par with the one step horizon MPC controller, while keeping the constraints handling feature. This makes this controller an interesting candidate for applications where the computation power available is not high. The main problem of this kind of controller is the complexity of the positive controlled invariant set,  $\mathcal{C}_N$ . Although the computation of this set is made offline and does not affect the online load solver, this set increases tremendously its complexity each step we take backwards when computing it the backward reachability construction, thus, its size is relatively small comparing to the admissible state space set  $X$ . This drawback has been improved by a scaling technique that increases the domain of attraction of the interpolation based controller. However, this kind of controller would not be advisable for a system with complex positive invariant sets, as those issued from the iterative construction of [Dorea 1999] in large state space. The construction of controlled invariant sets of low complexity represents a fundamental off-line requirement for on-line computational efficiency. The research on these type of constructions are representing currently a very active trend and we choose not to report them here as they do not represent the main research topic for the present thesis. We point the reader to the recent papers [Hovd 2014], [Munir 2016], [Laraba 2016] for a complete account of the state of the art and mention that these can be further improved with polynomial type of controlled invariant sets.

From the application point of view, the system studied in the present work has been reduced in order to provide an scenario where the attention is focused in the variation of the speed. There are more features which can be considered: these include actuator dynamics and additive disturbances, like wind forces acting on the vehicle or the curvature of the road, which are covered in the next chapter.



# Lane Centering Assistance System

---

## Contents

---

<b>4.1</b>	<b>Dynamical model for LCA</b> . . . . .	<b>66</b>
4.1.1	Additive disturbances modeling . . . . .	67
4.1.2	Actuator dynamics . . . . .	70
4.1.3	Full dynamical model . . . . .	71
4.1.4	LCA system constraints . . . . .	73
<b>4.2</b>	<b>Control design for LCA system</b> . . . . .	<b>73</b>
4.2.1	LPV design and robustness analysis via mRPI tools . . . . .	74
4.2.2	LPV controller redesign for built-in robustness . . . . .	76
4.2.2.1	Comparative simulation LPV vs Robust LPV control design . . . . .	77
4.2.3	Simultaneous observer-controller design . . . . .	80
4.2.3.1	Comparative simulation LPV robust control and observer-based designs . . . . .	81
4.2.4	Advanced tools I: Switched LPV design . . . . .	84
4.2.5	Advanced tools II: Acceleration considerations . . . . .	87
4.2.6	MPC for LCA system control . . . . .	92
<b>4.3</b>	<b>Conclusion</b> . . . . .	<b>95</b>

---

This chapter devoted to the Lane Centering Assistance (LCA) System study closes the lateral dynamics constrained control design part. This application can be seen as the major goal of the auto-steering applications, as it is the base for both lateral control dynamics and for more complex maneuvers, like the lane change or overtaking, studied in the second part of this manuscript.

Up to this stage, we have studied LPV control considering a measured and bounded parameter, which is, the vehicle speed. In the following, we consider not only the internal variation of this parameter, but also the effect that other external elements have in the vehicle dynamics: the curvature of the road, modeled as a parameter-varying additive disturbance, is brought into the picture and included in the system model. In addition, actuator dynamics are included in the system modelization.

Once the complete model is set, this chapter is focused on the design of a pre-stabilizing LPV controller when the speed of the vehicle changes inside a large range of values. Thereafter, robust positive invariance theory is used to analyze the effect

of the curvature of the road on the obtained controller, and how the uncertainty decreases the domain of attraction of the LPV control. After this analysis, a redesign phase that explicitly takes the additive disturbances into account is performed to increase such domain of attraction, followed by the design of an observer-based robust LPV controller.

These robust designs have a correct performance, but still their domain of attraction does not seem satisfactory, so two different strategies are proposed afterwards to improve the control design. First, a switched LPV control strategy is proposed, which attenuates the effect of the large speed variation range. Then, the maximal acceleration capabilities of the vehicle are considered. This reduces the conservativeness of the previous approaches, induced by an infinite rate of variation of the parameter.

Finally, on top of the previous designs, a MPC scheme is used to enhance the constraint handling capacities and enlarge the domain of attraction thanks to the receding horizon strategy.

## 4.1 Dynamical model for LCA

We recall that the LCA system's main purpose is to stabilize the lateral dynamics of the vehicle to follow the center line of the lane by acting on the steering wheel of the vehicle (Figure 4.1). The dynamical model presented in the following comprises the most representative characteristics of the vehicle lateral dynamics behavior, as well as system actuator dynamics and the influence of the curvature of the road. The lateral dynamics are modeled once more by the *bicycle model*, which has already

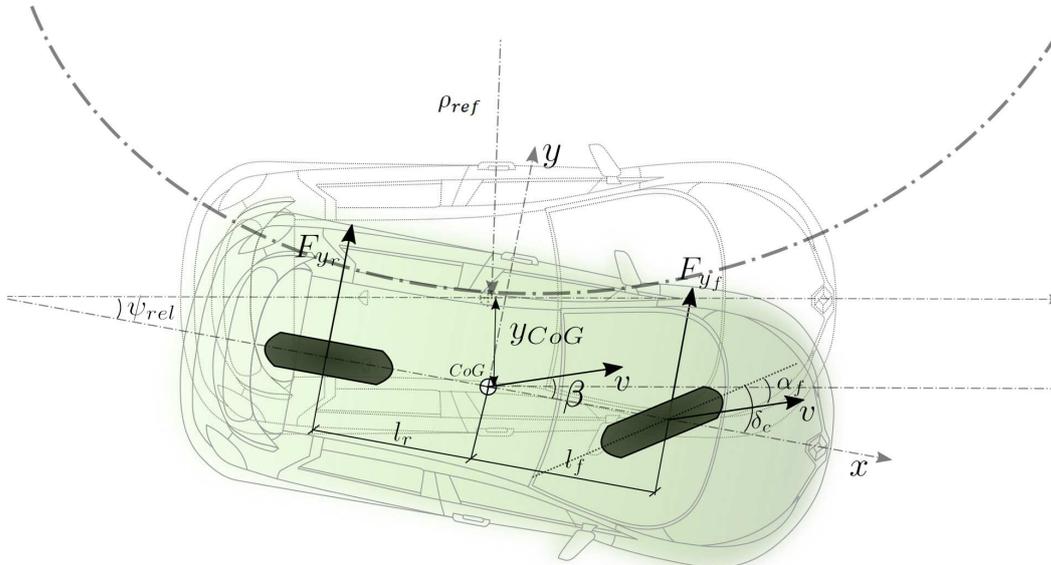


Figure 4.1 – Bicycle model referenced to the center line

been introduced in Section 1.2.3.1, so next we concentrate on the modelization of the road curvature effect and the actuator dynamics, before introducing the full dynamical model.

#### 4.1.1 Additive disturbances modeling

When driving in a dynamical environment, perturbations may arise due to different sources, such as a lateral slope on the road, difference of pressure in the tyres, actuation perturbations or crosswind [Hanke 2001]. This kind of perturbations can be modeled as an additive white noise  $[w_{n_f}, w_{n_r}]^T$  in the front and rear lateral forces  $F_{y_f}, F_{y_r}$ . (Section 1.2.3.1), yielding the disturbances matrix

$$H = \begin{bmatrix} \frac{l_f}{I_z} & 0 & \frac{-l_r}{I_z} & 0 & 0 & 0 \\ \frac{1}{m} & 0 & \frac{1}{m} & 0 & 0 & 0 \end{bmatrix}^T. \quad (4.1)$$

In addition to these, an important disturbance affecting the lateral dynamics is the curvature of the road  $w_k$ , due to its direct influence on the steering input signal. This perturbation affects the lateral dynamics behavior of the system, by adding an extra term to the lateral equations of motion, which is related to the centripetal acceleration that appears when driving in a curve

$$\ddot{y}_{CoG_p}(t) = -v_x^2(t)\rho(t).$$

In consequence, the required steering wheel angle is directly affected, as we will need to steer as much as it is needed to follow the curve and keep the vehicle centered. This effect is included in the system dynamics model by means of a bounded additive disturbance,  $\rho \in \mathcal{R}$ , defined by the disturbances matrix

$$E(v_x(t), v_x^2(t)) = [0, -v_x(t), -v_x^2(t), 0]^T, \quad (4.2)$$

with  $E(v_x(t), v_x^2(t)) \in \mathbb{R}^{n_x \times n_p}$ . It can be seen in (4.2) that this matrix depends on the vehicle speed, thus we have a parameter-varying disturbances matrix.

In the following, three different modeling abstractions are presented in decreasing conservativeness order. The first one is based on the independent consideration of the speed realizations appearing in  $E(v_x(t), v_x^2(t))$ . Then, a polytopic representation is proposed and thereafter refined to obtain a more precise representation of the disturbances.

#### Superposition principle

To begin with, the two realizations of the speed parameter,  $v$  and  $v^2$  are defined respectively as  $\nu_1$  and  $\nu_2$ , yielding  $E(\nu_{1_k}, \nu_{2_k})$ . Due to the linearity of the dynamics of the parameter-varying matrix  $E(\nu_1, \nu_2)$ , a superposition principle can be employed in order to separate the influence of the two uncertainty sources,  $E(\nu_1, \nu_2) = E_1(\nu_1) + E_2(\nu_2)$ , and analyze independently their impact on the closed loop dynamics. Again, this parameter uncertainty can be embedded in a polytopic

approach, obtaining each value inside the bounded working range of speed as a convex combination of the two extreme values:

$$E(\nu_{1_k}, \nu_{2_k}) = \sum_{i=1}^{n_v} \beta_i E_{1_i}(\nu_{1_k}) + \sum_{i=1}^{n_v} \eta_i E_{2_i}(\nu_{2_k}), \quad (4.3)$$

with  $\sum_{i=1}^{n_v} \beta_i = 1$ ,  $\beta_i \geq 0$ ,  $\sum_{i=1}^{n_v} \eta_i = 1$ ,  $\eta_i \geq 0$  and the matrix  $E$  defined by the addition of the matrices  $E_1$  and  $E_2$ , that are linear on the corresponding parameters,

$$E(\nu_{1_k}, \nu_{2_k}) = E_1(\nu_{1_k}) + E_2(\nu_{2_k}) = [0, -\nu_{1_k}, 0, 0, 0]^T + [0, 0, -\nu_{2_k}, 0, 0]^T.$$

Moreover, the dependency of the matrices  $E_1$  and  $E_2$  on the parameters  $\nu_{1_k}, \nu_{2_k}$  can be eliminated by scaling the disturbances boundaries with the maximum values of the parameter:

$$\begin{aligned} x_{k+1} &= A_{cl}(\nu_k)x_k + E(\nu_{1_k}, \nu_{2_k})\rho_k = \\ &= A_{cl}(\nu_k)x_k + E_1(\nu_{1_k})\rho_k + \bar{E}_2(\nu_{2_k})\rho_k = \\ &= A_{cl}(\nu_k)x_k + E_1 \underbrace{\nu_{1_k}\rho_k}_{\rho_{1_k}} + E_2 \underbrace{\nu_{2_k}\rho_k}_{\rho_{2_k}}, \end{aligned} \quad (4.4)$$

where  $E_1 = [0, -1, 0, 0, 0]^T$  and  $E_2 = [0, 0, -1, 0, 0]^T$  are now parameter independent constant matrices, and the pair of redefined disturbances are  $\rho_{1_k} \in \mathcal{R}_1$ ,  $\rho_{2_k} \in \mathcal{R}_2$ , with  $\mathcal{R}_1 = \nu_{1_{max}} \mathcal{R} = v_{x_{max}} \mathcal{R}$  and  $\mathcal{R}_2 = \nu_{2_{max}} \mathcal{R} = v_{x_{max}}^2 \mathcal{R}$ .

Note however, that the superposition of the effects of these two parameters will only offer a over approximation as long as the co-variance of the sources of disturbances is lost when considering this representation.

### Worst case polytopic representation

In order to preserve the existing relationship in between the parameters  $\nu_1$  and  $\nu_2$  it can be seen that the domain of variation of  $\nu_1, \nu_2$  is certainly defined by the curve  $f(v) = v^2$ , scaled by the constant value  $\rho_m$ , which is the maximal value of the road curvature.

The function can be represented by a polytopic embedding, approximated for example by a trapezoidal shape (Figure 4.2) for the sake of simplicity of the vertex representation. This is first limited by the segment  $[\rho_m \nu_{1_{min}}, \rho_m \nu_{2_{min}}], [\rho_m \nu_{1_{max}}, \rho_m \nu_{2_{max}}]$ . Then, we consider a line which is parallel to the first segment and tangent to the curve and the last two edges are defined by intersection of this line and the ones tangent at the extreme vertex. This provides a polyhedron with four vertices,  $n_v = 4$ , each one defining an extreme matrix realization  $E_i$  such that

$$E(\nu_{1_k}, \nu_{2_k}) = \sum_{i=1}^{n_v} \alpha_i E_i(\nu_{1_k} \rho_m, \nu_{2_k} \rho_m), \quad (4.5)$$

with  $\sum_{i=1}^{n_v} \alpha_i = 1$  and  $\alpha_i \geq 0$ . Alternatively, a containment optimization problem can be performed [Lombardi 2009] to obtain a tight polyhedral embedding of the curve.

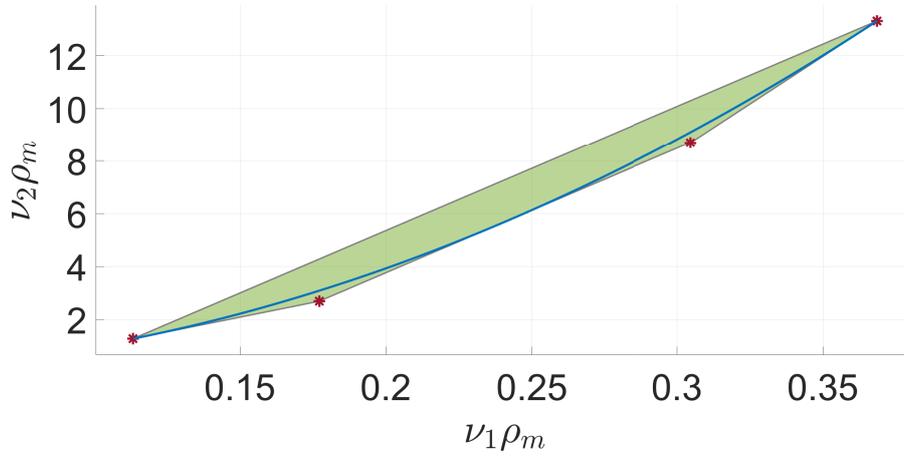


Figure 4.2 – Polytopic approximation of the curve  $v_x \rho_m = v_x^2 \rho_m$

Still, it is necessary to note that we are considering the same maximal value of the curvature of the road  $\rho_m$  for all the range of speed, which remains an important source of conservatism: road conception directives [Vertet 2006] dictate from the design stage of the infrastructure facilities that the maximal road curvature and the maximal driving speed regulation are directly related. These considerations are considered in the next formulation and integrated in the additive disturbance analysis.

### Refined polytopic representation

Following up with the polytopic representation, it is possible to go forward by means of considering that the maximal value of the curvature is speed-dependent too, that is, there exists a series of rules and conventions that are taken into account when the roads are designed, in order to ensure safety and comfort of the drivers. These conventions set a tight relation between the speed limit at a given road and the minimal radius for the road profile. Table 4.1 shows the convention for the design of highways in France.

Table 4.1 – Radius-speed values for design on highways [Vertet 2006]

Speed [km/h]	50	70	90	110	130
Comfort $R_{min}$ [m]	98	242	473	808	1267
Safety $R_{min}$ [m]	66	162	318	541	848

When we consider this formulation, we are highly reducing the conservatism of the previous two formulations, where a polytopic representation of the disturbances

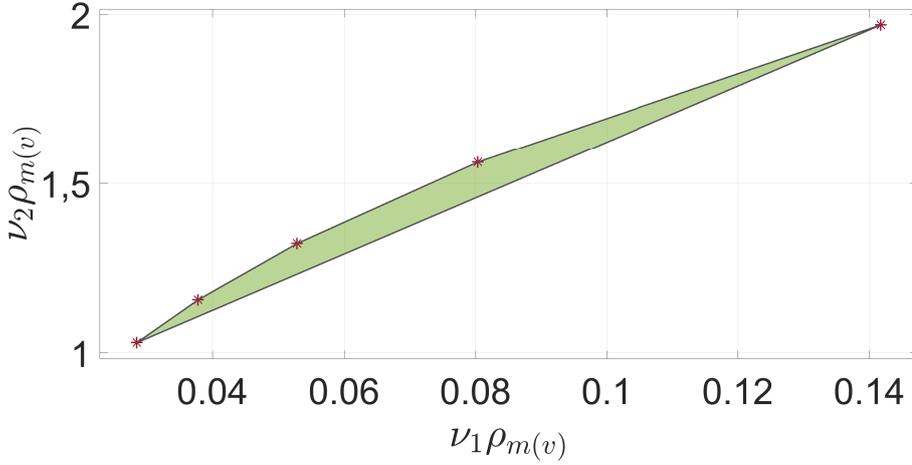


Figure 4.3 – Polyopic representation from Table 4.1 data (comfort)

matrix is proposed, whose defining vertex  $E_i$  (Figure 4.3) are defined by each one of the maximal curve-speed pairs shown in Table 4.1. In this way, the disturbances matrix can be parametrized as a function of a unique parameter, defined by the affine combination of the vertex defined polytope  $E_j(\gamma_j)$ :

$$E(\gamma) = \sum_{j=1}^{n_v} \alpha_j E_j(\gamma_j), \quad (4.6)$$

with  $\alpha_j$  laying in the unit simplex  $\Lambda_{n_v}$  given by  $\Lambda_{n_v} = \{\alpha \in \mathbb{R} : \sum_{j=1}^{n_v} \alpha_j = 1, \alpha_j \geq 0\}$  and  $n_v = 5$ .

Finally, the inclusion of the maximal curvature on the disturbances matrix allows to define a normalized disturbances vector,

$$w_j \in \mathcal{W}_j = \{w_j^T w_j \leq 1\}. \quad (4.7)$$

#### 4.1.2 Actuator dynamics

In the auto-steering context, it is necessary to actuate on the direction system [Reimpell 2001] as the driver would. With this in mind, a control action needs to be provided to the steering system via an actuator, in order to completely eliminate the human action. The wide majority of vehicles that are nowadays in the market are equipped with a Steering Assistance System (SAS), which supports the driver by reducing the needed effort when actuating on the steering column to turn the vehicle wheels. Some applications have been recently included to improve driver's ergonomics, like direction torque hardening at high speed.

In the present work, a steer-by-wire schema is considered. This kind of design is controlled electronically, eliminating any mechanical coupling between the steering

wheel and the wheels. Instead, two actuators are installed, generally two electric motors, which are in charge of steering the front wheels and the steering wheel independently. Both actuators are associated, providing the direction, speed and force of the steering wheel and sending it to the steering system at the wheels. In the same way, the driver receives an effort feedback through the steering wheel. This kind of system presents several advantages from the mechanical point of view, eliminating a great quantity of physical components, reducing the probability of mechanical failure and can be monitored for the possibility of electronic failures.

Both the assistance and the actuation actions for the auto-steering application are provided by the same (electric) motor, that will set the commanded position. There are other kinds of SAS, which include hydraulic or hybrid designs, interested readers are referred to [Enache 2008] for a deeper discussion on the topic. In addition to this, the installed instrumentation includes sensors that allow to detect the torque  $\tau$  and position of the steering column,  $\delta$ . Torque measurements are used to monitor the driver, and check if the measured torque is above a minimum threshold, indicating that the steering wheel is being hold. Otherwise, the auto-steering assistance system is disconnected for safety reasons.

In the following, the actuator dynamics are presented, considering an electric motor under position control, which is part of the steer-by-wire system and that is in charge of transforming the commanded steering angle  $\delta_c$  into front wheels steering angle  $\delta$ . This model follows the lines of the steering actuator presented in [Ackermann 2012], where a linearized steering actuator model transfer function is given by a second order system transfer function,

$$\delta = \frac{\omega^2}{s^2 + 2\xi\omega s + \omega^2} \delta_c. \quad (4.8)$$

Translated into a state-space representation, we obtain

$$\frac{d}{dt} \begin{bmatrix} \dot{\delta} \\ \delta \end{bmatrix} = \begin{bmatrix} -2\xi\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\delta} \\ \delta \end{bmatrix} + \begin{bmatrix} \omega^2 \\ 0 \end{bmatrix} \delta_c, \quad (4.9)$$

where the damping factor is represented by  $\xi$  and  $\omega$  stands for for the cut frequency of the modeled dynamics. Numerical values of the parameters are not included for confidentiality reasons.

### 4.1.3 Full dynamical model

Integrating all the elements together, we can formulate the complete dynamical model in view of LCA system control that includes the formulation of the lateral dynamics of the vehicle with respect to the road, the curvature influence and the actuator behavior.

$$\begin{aligned} \dot{x}(t) &= A_c(\nu(t))x(t) + B_c u(t) + E_c(\gamma(t))w(t) \\ y(t) &= C_c x(t) \end{aligned} \quad (4.10)$$

where  $A_c(\nu(t))$ ,  $B_c$ ,  $C_c$ ,  $E_C(\gamma)$  are defined by the concatenation of the corresponding models presented in the precedent sections,

$$A_c(\nu(t)) = \begin{pmatrix} \frac{-(C_f l_f^2 + C_r l_r^2)\nu(t)}{I_z} & \frac{(C_f l_f - C_r l_r)}{I_z} & \frac{-(C_f l_f - C_r l_r)\nu(t)}{I_z} & 0 & 0 & \frac{C_f l_f}{I_z} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{(C_r l_r - C_f l_f)\nu(t)}{m} & \frac{(C_f + C_r)}{m} & \frac{-(C_f + C_r)\nu(t)}{m} & 0 & 0 & \frac{C_f}{m} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2\xi\omega & -\omega^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix},$$

$$B_c = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \omega^2 \\ 0 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad E_c = \begin{pmatrix} 0 \\ -\frac{1}{\nu(t)}\rho(t) \\ -\frac{1}{\nu^2(t)}\rho(t) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

(4.11)

with the following state, output and input vectors (Table 1.1)

$$\begin{aligned} x &= [\dot{\psi} \quad \psi_{rel} \quad \dot{y}_{CoG} \quad y_{CoG} \quad \dot{\delta} \quad \delta \quad \int -y_{CoG} dt]^T \\ y &= [\dot{\psi} \quad \psi_{rel} \quad y_{CoG} \quad \delta \quad \int -y_{CoG} dt]^T \\ u &= \delta_c \end{aligned} \quad (4.12)$$

It has to be noted that this reformulation includes an integral action on the lateral position of the vehicle with respect to the road, in order to eliminate the steady-state error. In the following, system dynamics are discretized by a first order Euler method with  $T_s = 10\text{ms}$ .

Where matrices  $A(\nu)$ ,  $E(\gamma)$  satisfy:

$$\begin{aligned} A(\nu_k) &= \sum_{i=1}^{n_v} \lambda_k A_i(\nu_i), \quad \text{with } \sum_{i=1}^{n_v} \lambda_k = 1, \quad \lambda_k \geq 1, \\ E(\gamma_k) &= \sum_{j=1}^{n_v} \alpha_k E_j(\gamma_j), \quad \text{with } \sum_{j=1}^{n_v} \alpha_k = 1, \quad \alpha_k \geq 1. \end{aligned} \quad (4.13)$$

**Remark 4.1** *It is important to notice that the vehicle and actuator dynamics are modeled in terms of the front wheels steering angle  $\delta$ . Nevertheless, our actuator and the corresponding sensor is connected to the steering system through a rack. Thus, it is crucial to consider the reduction ratio  $i_r$  between the front wheels and the vehicle steering wheel when interacting with the system, sending the control signals  $\bar{\delta}_c$  or transforming the received the measurements  $\delta_{meas}$  in the proper reference.*

$$\begin{aligned} \bar{\delta}_c &= \delta_c i_r, \\ \delta &= \delta_{meas} / i_r. \end{aligned} \quad (4.14)$$

#### 4.1.4 LCA system constraints

Control design strategies considered in the following take into account LCA system constraints as well. States and input limits are defined by the numerical values shown in Table 4.2, that define the polytopic sets of bounded admissible states  $X \subset \mathbb{R}^{n_x}$  and inputs  $U \subset \mathbb{R}^{n_u}$ .

Table 4.2 – Trajectory Constraints Definition for LCA System

Magnitude	Range	Definition
$\dot{\psi}$	$[\pm 3.78]$	Heading angle derivative [rad/s]
$\psi_{rel}$	$[\pm 0.3]$	Heading angle [rad]
$\dot{y}$	$[\pm 1]$	Lateral speed [m/s]
$y_{CoG}$	$[\pm 2]$	Lateral position [m]
$\dot{\delta}$	$[\pm 0.43]$	Steering angle variation rate [rad/s]
$\delta$	$[\pm 0.52]$	Steering angle [rad]
$\delta_c$	$[\pm 0.52]$	Steering angle requested [rad]

## 4.2 Control design for LCA system

Let us recall that the LCA system objective is to control the vehicle to the center of the current lane. Due to the fact that the system dynamics have been expressed with respect to the road, such an objective is translated into a stabilization of system dynamics to the origin.

Robustness is a general concept that analyzes or takes into account the effect of mismatch between a nominal plant and the reality, as well as possible plant variations that may occur. In this section robustness of a LPV control design is analyzed via mRPI tools. After that, built-in robustness is considered from the design stage, and enhanced by means of a switching strategy and parameter variation considerations.

The main objective is to compute a certified range of operation for designed LPV controllers in the presence of parameter variations, additive disturbances and system constraints. In this way, it is important to delimit the region of the state-space in which the conceived tuning for the LPV system satisfies the system constraints regardless the driving conditions. Again, such region is approximated by an ellipsoid  $\mathcal{E}(P) = \{x^T P^{-1} x \leq 1\}$  centered at the origin, that is RPI with respect to the closed loop dynamics if  $\forall x \in \mathcal{E}(P)$  then  $A_{cl}(\nu)x_k + E(\gamma)\rho_k \in \mathcal{E}(P)$ ,  $\forall \rho_k \in \mathcal{R}$  and  $\forall \nu \in \mathcal{V}$ ,  $\forall \gamma \in \Gamma$ , with  $A_{cl}$  denoting the closed-loop system dynamics. Moreover, any initial state that belongs to  $\mathcal{E}(P)$  will satisfy system constraints and not leave the computed admissible set under any (bounded) parameter variation or disturbance. These ellipsoids  $\mathcal{E}(P)$  will be the limit set of all the trajectories of the system controlled with the LPV feedback gain (4.18). This means that all the trajectories that start on the computed ellipsoid will remain inside with certification of the constraints satisfaction and all the ones starting outside this region will converge

to it [Luca 2011b]. Nevertheless, in the later case no certification of constraint satisfaction is provided.

#### 4.2.1 LPV design and robustness analysis via mRPI tools

As a first approach, the application of a LPV classical control design presented in Section 2.3.1 by means of Problem 2.2 is used to stabilize the LCA model dynamics presented in Section 4.1.3. Moreover, enhancing constraints on the direction of the states  $\psi_{rel}$  and  $y_{CoG}$  have been included, due to their special importance on the application. The last parameter that needs to be tuned is the contraction factor of the system,  $(1 - \alpha)$ . This parameter increases the speed of convergence of the closed-loop dynamics, while decreasing the size of the obtained ellipsoid. A common approach in the literature is to consider a value which is slightly smaller than the largest eigenvalue of the system transformation matrix,  $A$ , in this case,  $(1 - \alpha) = (1 - 0.999)$ .

The obtained parameter-varying feedback stabilizing gain for the constrained system,

$$u_{1k}(\nu_k) = \sum_{i=1}^{n_v} \lambda_i K_{1i}(\nu_i) x_k, \quad (4.15)$$

with with  $\sum_{i=1}^{n_v} \lambda_i = 1$ ,  $\lambda_i \geq 0$ ,  $n_v = 2$  and

$$v_{x_1} = 50[\text{km/h}]$$

$$K_{11}(\nu_1) = [-0.0275, -0.8287, -0.5265, -0.1157, -0.3317, -17.7578, 0.0096]$$

$$v_{x_2} = 70[\text{km/h}]$$

$$K_{12}(\nu_2) = [-0.0220, -0.8462, -0.5489, -0.1173, 0.3361, -18.0146, 0.0097]$$

This design ensures asymptotic stability for all the states inside the domain of attraction of the controller (Figure 4.4) for all the range of speed and in the absence of additive disturbances. Furthermore, in the presence of (bounded) additive disturbances, it will ensure that the states remain in a neighborhood of the origin. Nevertheless, the presence of unmodeled additive disturbances that have not been considered on the design may drive the system state out of its domain of attraction, where the controller performance is lost or, at least, not guaranteed in the presence of both constraints and parameter variation.

The continuation of this section performs the analysis of the impact of the these unmodeled additive disturbances on the closed-loop dynamics, by means of the mRPI set (Procedure 2.3) tools, as the size of this set provides a measurement of the uncertainty that the bounded disturbances produce on the closed loop dynamics of a system: the larger the mRPI set is, the more the system is affected by the disturbances.

Along these lines, the corresponding mRPI set has been computed for the closed-loop LCA system, and proves to not be included in the domain of attraction of the nominal LPV control (Figure 4.4). This is not surprising since the tuning of the LPV controller is done based on aggressive contractiveness objectives in spite of

robustness. This means that the performance of this controller is not kept for any speed-curvature variation.

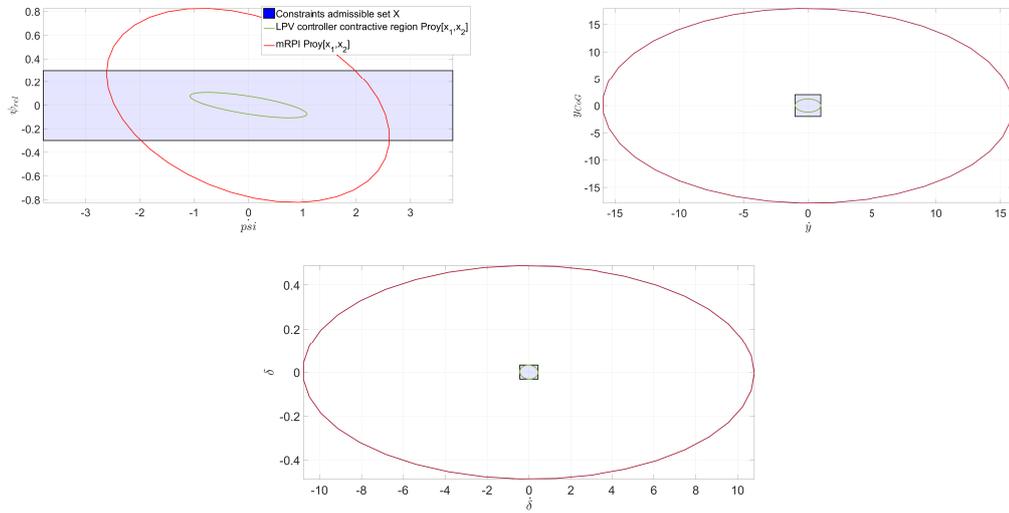


Figure 4.4 – Domain of attraction controller  $\mathcal{E}(P_1)$  and computed mRPI set

In practice, what can be done is the computation of the maximal range of speed variation that would be admissible for a given curvature value, naming *admissible* those speed limit values that provide a mRPI set which is smaller than the controller’s domain of attraction  $\mathcal{E}(P_1)$ . In Figure 4.5, the speed ranges for which the computed  $\mathcal{E}(P_{mRPI}) \subseteq \mathcal{E}(P_1)$  for fixed curvature values are shown<sup>1</sup>. It can be seen

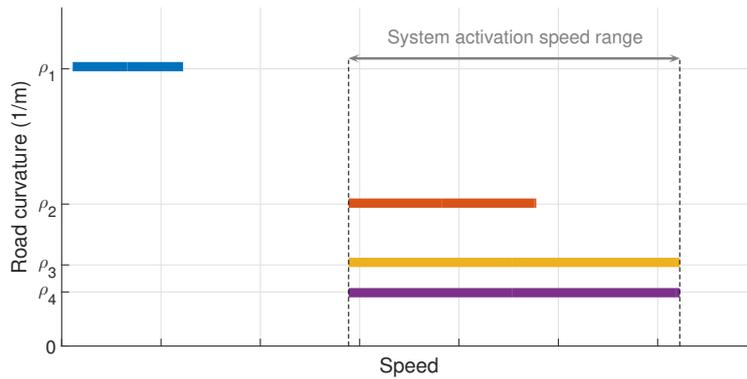


Figure 4.5 – Admissible speed range in curve for the parameter-varying controller for fixed  $\rho$  values

1. Due to confidentiality reasons the numerical details on the speed range and curvature values are omitted from the figure

that for lower curvature values the controller could be suitable in part of the speed range activation zone. However, if the curvature is higher, this kind of control would not be able to control the lateral dynamics of the system unless the speed was much lower than the range activation of the LCA system. Moreover, the size of the computed mRPI sets is relatively big compared to the size of the controller domain of attraction  $\mathcal{E}(P_1)$  (Figure 4.6), so the resulting invariant set  $X_N$  obtained from the refinement of  $\mathcal{E}(P'_N)$  turns out to be relatively small.

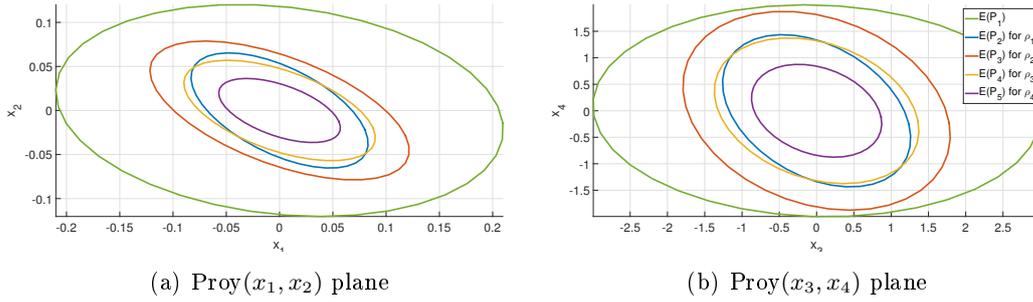


Figure 4.6 – Computed mRPI set  $\mathcal{E}(P_i)$  for fixed  $\rho_i$  values (Fig. 4.5)

#### 4.2.2 LPV controller redesign for built-in robustness

RPI tools used for the analysis of the previous design strategy can be used as well for synthesis purposes: in spite of pre-computing a stabilizing gain for the unperturbed system, the design is improved when the stabilizing gain computation is done at the same time as the RPI set computation, obtaining in this way a robust stabilizing gain that maximizes the RPI set, in which the input and output constraints are fulfilled in the presence of parameter-varying disturbances. This can be done by solving the LMI Problem 2.3 for a common RPI set.

The obtained parameter-varying linear state feedback gain provides a parameter-dependent control law leading to the limit state feedback gains with

$$u_{2k}(\nu_k) = \sum_{i=1}^{n_v} \lambda_i K_{2i}(\nu_i) x_k, \quad (4.16)$$

with  $\sum_{i=1}^{n_v} \lambda_i = 1$ ,  $\lambda_i \geq 0$ ,  $n_v = 2$  and

$$v_{x_1} = 50[\text{km/h}]$$

$$K_{21}(\nu_1) = [-0.1792, -1.9435, -0.0684, -0.1877, 0.2399, -1.2312, 0.0615]$$

$$v_{x_2} = 70[\text{km/h}]$$

$$K_{22}(\nu_2) = [-0.1892, -1.9962, -0.0713, -0.1929, -0.2486, -1.2918, 0.0632]$$

which maximizes the size of the RPI set in the presence of system constraints and disturbances (Figure 4.7).

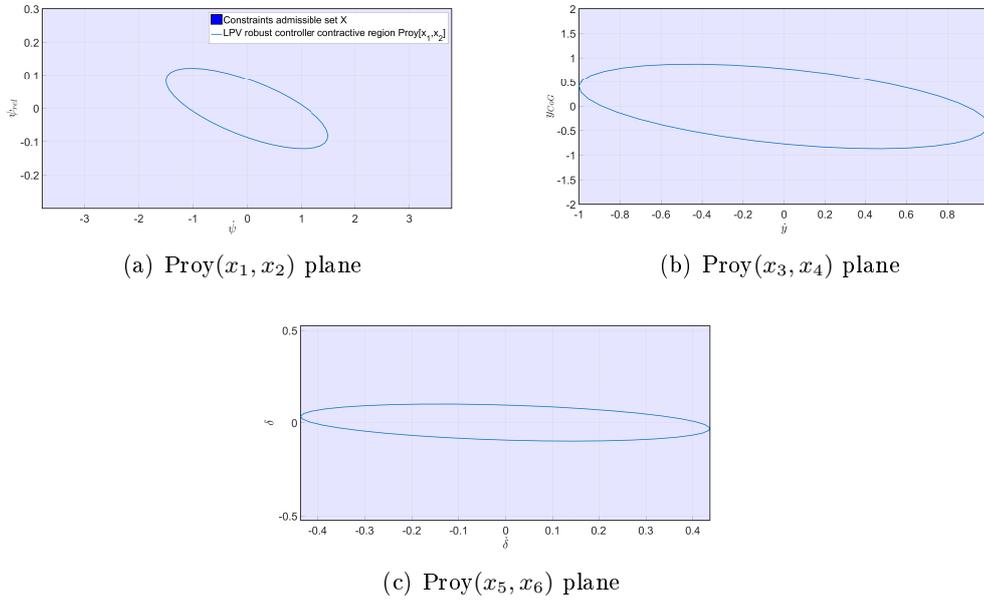


Figure 4.7 – LPV Robust control design with RPI maximization domain of attraction

**Remark 4.2** *As an alternative, we can also seek for a control law that minimizes the size of the mRPI set, that is, a control design whose main objective is to counteract the additive disturbances, by inverting the cost function objective in Problem 2.3:*

$$\min_{P_i, Y_i, \sigma} -\{\sigma + \text{trace}(P_i)\}.$$

□

In order to offer a complete robustness analysis of the redesigned LPV strategy with respect to the additive disturbances acting on the lateral dynamics of the system, we can compute the mRPI once more with  $E_j = H$  (4.1) in order to compute the mRPI set (ellipsoidal approximation) corresponding to the additive disturbances that can be modeled but cannot be predicted, which are represented by white noises that model other uncertainties acting on the lateral forces  $F_{y_f}, F_{y_r}$  (Section 4.1.1), shown in Figure 4.8.

#### 4.2.2.1 Comparative simulation LPV vs Robust LPV control design

In the following, we present the time evolution of the system states subject to a speed and curvature variation profile (Figures 4.9(a) and 4.9(b)), starting from the initial conditions defined by  $x_0 = [0.02, -0.05, 0, 1, 0, 0]$ , which represent a common situation in which the LCA system can be activated.

As expected, it can be seen in Figures 4.10 and 4.11 that the temporal responses of the robust LPV controller are slower than the ones of the LPV controller, nevertheless the domain of attraction is bigger as can be seen in Figure 4.12.

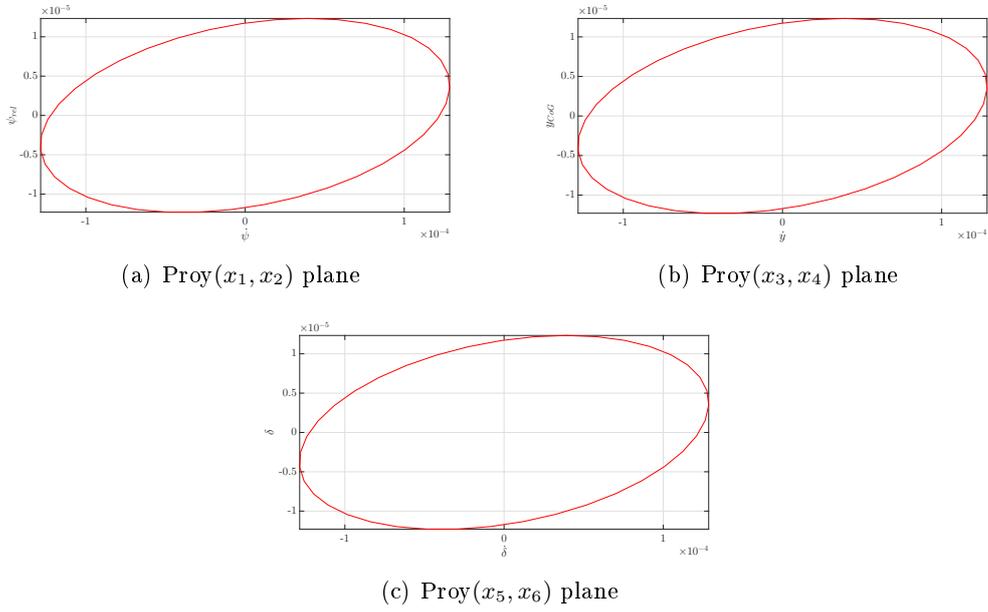


Figure 4.8 – mRPI ellipsoidal approximation for unmodeled additive disturbances

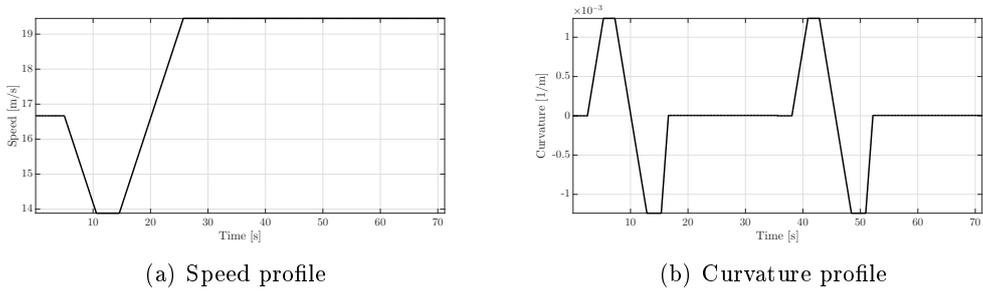


Figure 4.9 – Speed and Curvature Profile for LPV and Robust LPV designs comparison

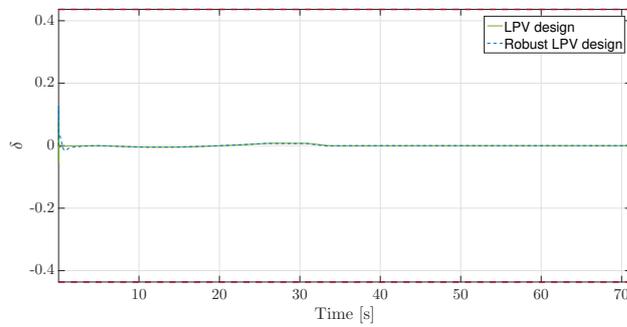


Figure 4.10 – System input trajectory for LPV and Robust LPV designs comparison

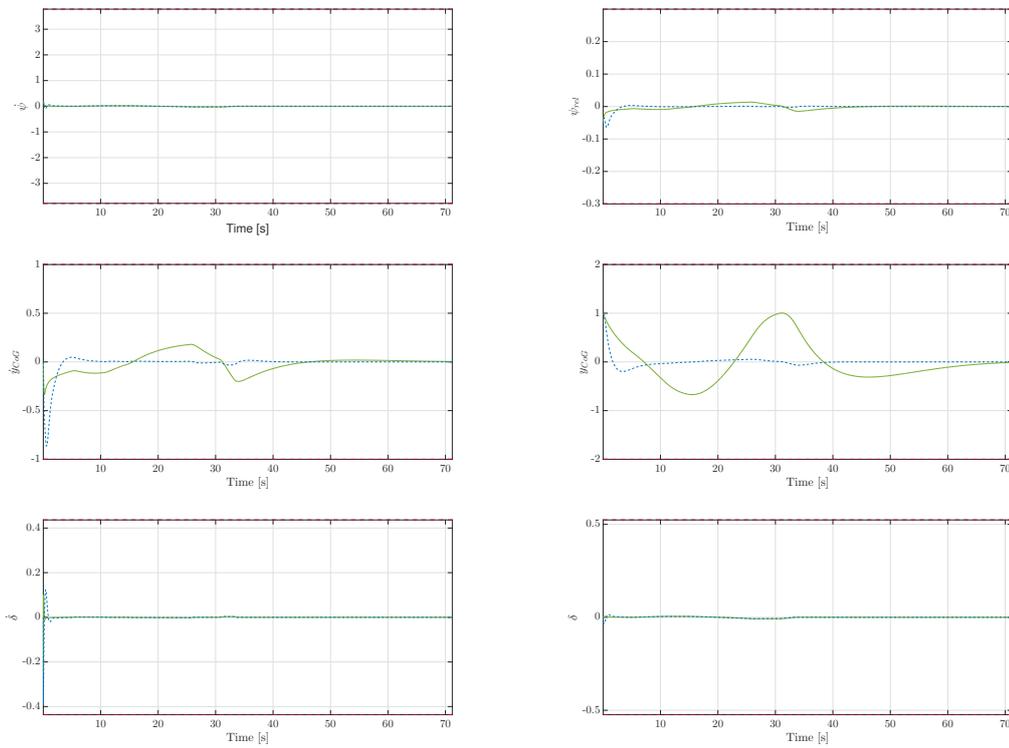


Figure 4.11 – System states trajectory for LPV and Robust LPV designs comparison

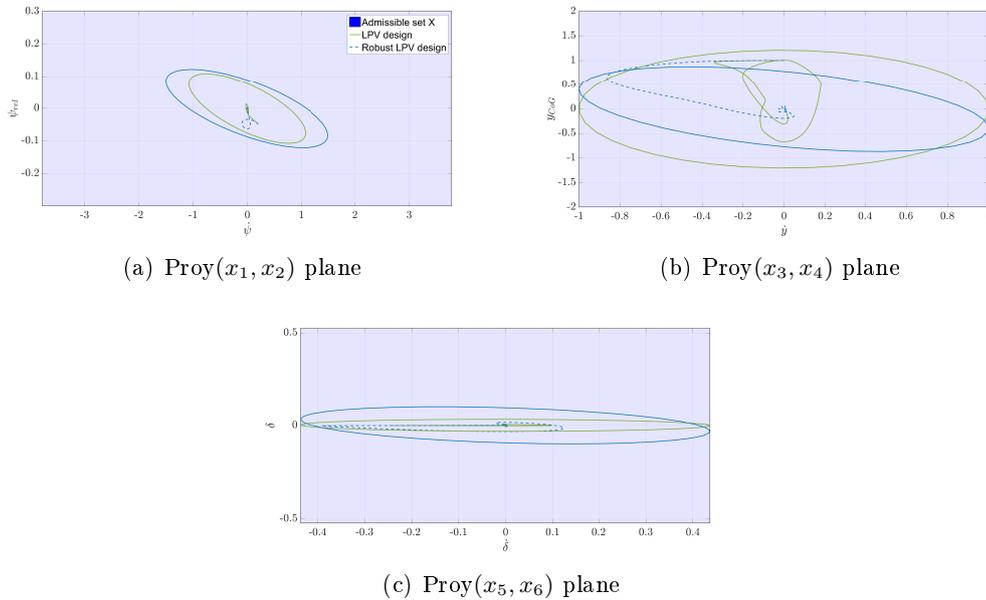


Figure 4.12 – System states trajectory and ellipsoidal invariant sets projections for LPV and Robust LPV designs comparison

### 4.2.3 Simultaneous observer-controller design

The next step that is performed is to consider that not all LCA system states are measured, thus an observer-based controller state feedback for the LPV system in the presence of parameter-varying additive disturbances and speed variations is obtained in this section. Moreover, parameter-dependent Lyapunov function are brought into the picture to improve the flexibility of the problem formulation.

First of all, we need to revisit observability of the LCA system dynamics.

#### Observability of the auto-steering system

Observability of a LPV system (4.10) can be dealt with based on the following theorem [Tóth 2007], [Sename 2013], which is recalled here for completeness.

**Theorem 4.1** *A LPV system is  $x_{k+1} = A(\theta_k)x_k + Bu_k$ ,  $y_k = Cx_k$  is completely observable if  $\text{rank } \mathcal{O}_n(\theta_{k:k+n-1}) = n$  for all  $k \in \mathbb{Z}$ , where  $\mathcal{O}_n(\theta_{k:k+n-1})$  is the parameter-varying state-observability matrix of the system defined as*

$$\mathcal{O}_n(\theta_{k:k+n-1}) = \begin{bmatrix} C \\ CA(\theta_k) \\ \vdots \\ C \prod_{l=1}^{n-2} A(\theta_{k+n-2-l}) \end{bmatrix}, \quad (4.17)$$

where  $\theta_{k:k+n-1} = [\theta_k, \dots, \theta_{k+n-1}]$ . ■

This means that the concept of observability is similar to the linear time-invariant (LTI) systems case but considering all the possible trajectories of the parameter  $\nu_k \in \mathcal{V}$ . Despite the fact that in [Sename 2013] it is shown that observability of the vertex realizations of the polytopic representation of the system dynamics for the extreme values of the parameter, is not a sufficient condition for the observability of the LPV system for the whole range of the parameter, we can exploit the structure of the LCA system dynamics in (4.10), (4.11) and show that it is an observable system no matter the speed value.

Let us consider the first two blocks of the observability matrix  $\mathcal{O}_n(\gamma)$  defined as in (4.17), obtaining  $[C, CA(\nu_k)]^T$ . From this two blocks, it is possible to obtain a full rank matrix that does not depend on the speed of the vehicle, for example  $[C, CA(\nu_k)_{(2:3 \times 7)}]^T$ , thus, the LCA system is observable no matter the parameter value.

#### Observer-based controller design

The problem design is formulated in terms of LMI, following the lines introduced in Section 2.3.1. The resolution of Problem 2.4 provides a suitable parameter-varying tuning that ensures input-to-state stability for a LPV system which is affected by additive disturbances. This approach allows to extend the speed variation range by 10[km/h] with respect to the previous approaches. The following results are computed for  $v \in [50 : 70][km/h]$ .

$$K(\nu_k) = \sum_{i=1}^{n_v=2} \mu_{i_k} K_i(\nu_i) \text{ and } L(\nu_k) = \sum_{i=1}^{n_v=2} \mu_{i_k} L_i(\nu_i). \quad (4.18)$$

with:

$$\begin{aligned}
 v_{x_1} = 50[\text{km/h}] \quad K_1(\nu_1) &= [0.1623, 1.6356, 0.0611, 0.0791, 0.2513, 0.9753, 0.0125] \\
 L_1(\nu_1) &= \begin{bmatrix} -0.2026 & 0.0399 & 2.2221 & -0.2363 & -0.0263 \\ -0.1095 & 0.9944 & 0.1537 & -0.0536 & -0.0022 \\ -5.1466 & 15.7732 & 43.6395 & -7.9471 & -0.2940 \\ -0.1654 & 0.1597 & 1.6289 & -0.1342 & -0.0053 \\ 0.3689 & -0.0002 & -0.8396 & -2.8496 & 0.0180 \\ -0.1077 & -0.0000 & 0.1355 & 0.9414 & -0.0021 \\ 0.0013 & -0.0008 & -0.0138 & 0.0011 & 1.0000 \end{bmatrix} \\
 v_{x_2} = 70[\text{km/h}] \quad K_2(\nu_2) &= [0.1667, 1.6606, 0.0598, 0.0784, 0.2515, 0.9782, 0.0124] \\
 L_2(\nu_2) &= \begin{bmatrix} -0.1862 & -0.0462 & 2.0594 & -0.1507 & -0.0244 \\ -0.1047 & 0.9943 & 0.0677 & -0.0148 & -0.0025 \\ -5.2740 & 16.1991 & 44.6131 & -8.0724 & -0.2878 \\ -0.1662 & 0.1617 & 1.6455 & -0.1457 & -0.0051 \\ 0.3557 & -0.0002 & -0.8315 & -2.7252 & 0.0177 \\ -0.1090 & -0.0000 & 0.1396 & 0.9508 & -0.0020 \\ 0.0013 & -0.0008 & -0.0140 & 0.0021 & 1.0000 \end{bmatrix}
 \end{aligned}$$

where the computed invariant ellipsoids  $P_i$  for the extreme parameter realization are shown in Figure 4.13.

It can be noticed that both ellipsoidal sets in Figure 4.13 are quite similar. This comes from the conservative consideration that the parameter variation can happen infinitely fast, so the invariant sets tend to overlap to ensure that no matter the change on the parameter, the state belongs to the affine combination of these two extreme realizations of the RPI set. This conservativeness source is dealt with in the subsequent of this section.

#### 4.2.3.1 Comparative simulation LPV robust control and observer-based designs

In the following, we picture the time evolution of the system states (Figure 4.16) subject to the same speed and curvature variation profile than in Section 4.2.2.1, (Figures 4.9(a) and 4.9(b)), starting from the same initial conditions defined by  $x_0 = [0.02, -0.05, 0, 1, 0, 0]$ . In this case, we will compare the performance of both robust controllers, where the observer-based design shows an improvement due to the additional degree of freedom introduced in the formulation by means of the parameter-varying Lyapunov functions  $P_i$  in exchange of a slower closed-loop convergence speed.

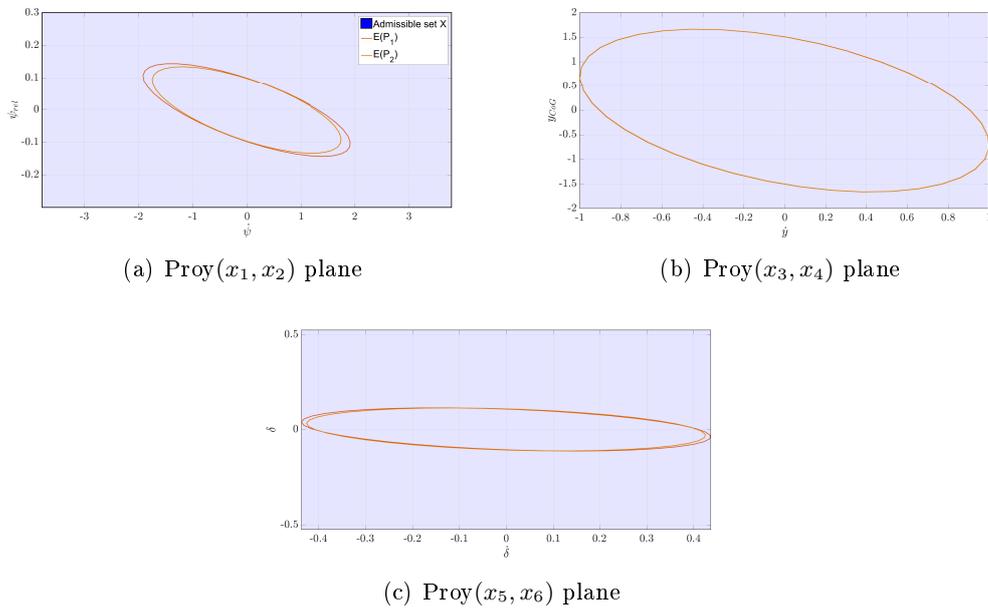


Figure 4.13 – Robust domain of attraction for simultaneous robust observer-controller design,  $\mathcal{E}(P_i)$

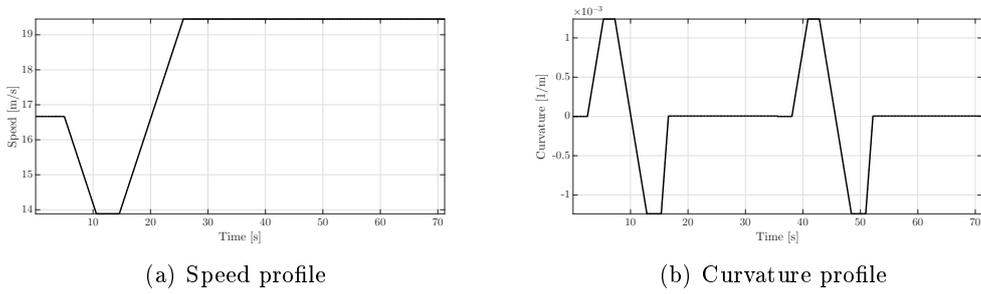


Figure 4.14 – Speed and Curvature Profile

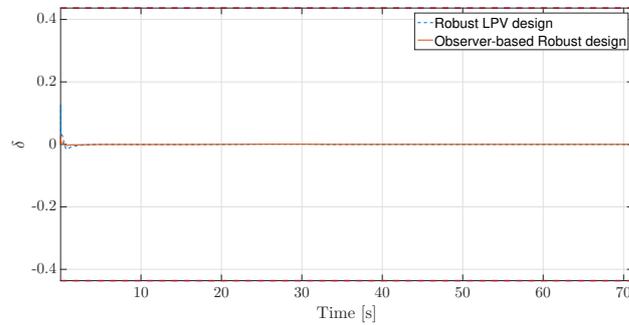


Figure 4.15 – System input trajectory Robust and Observer-based LPV designs comparison

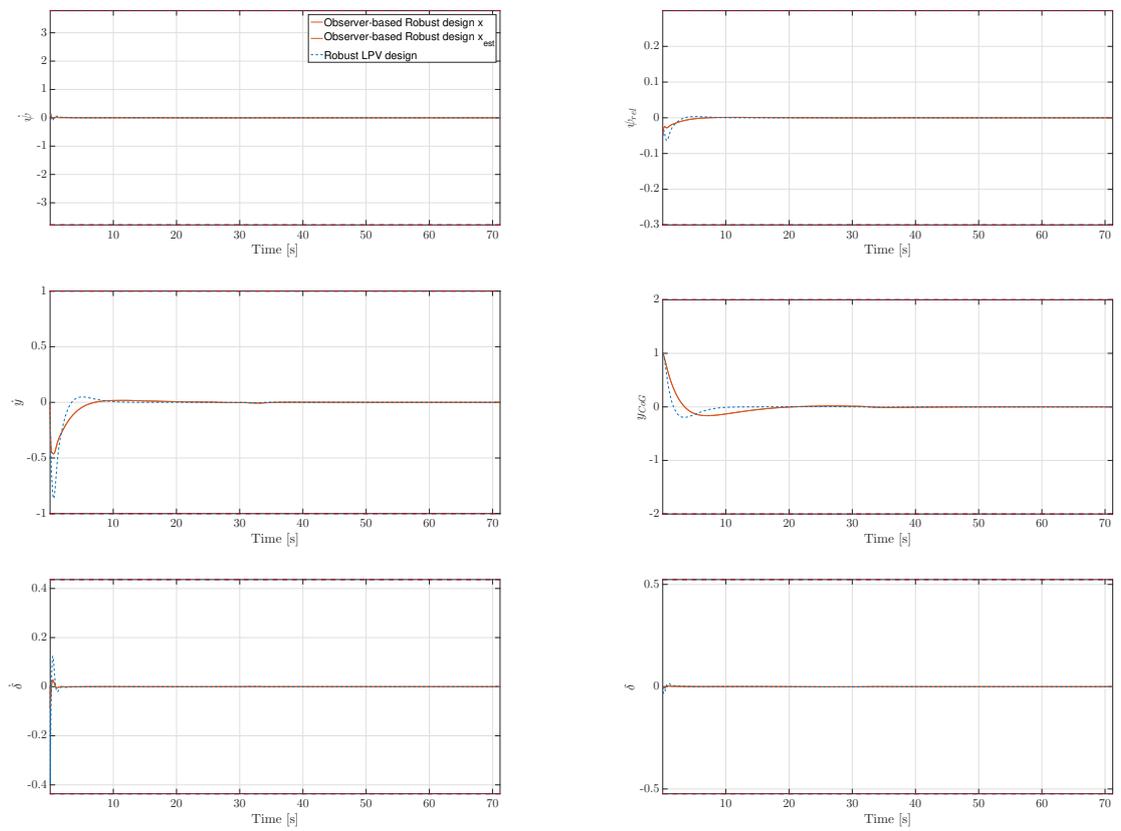


Figure 4.16 – System states trajectory Robust and Observer-based LPV designs comparison

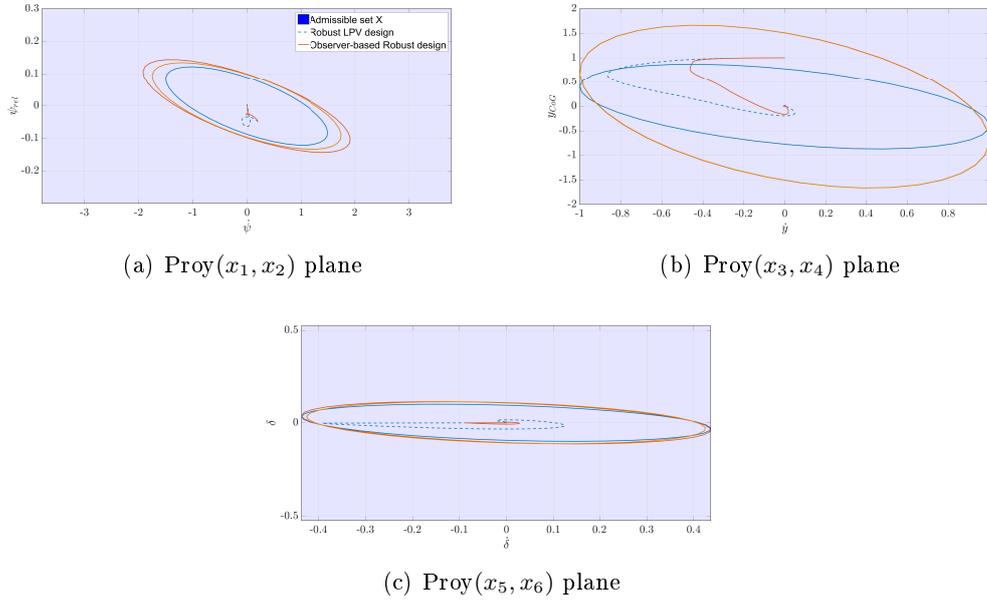


Figure 4.17 – System states trajectory and ellipsoidal invariant sets projections for Robust and Observer-based LPV designs comparison

#### 4.2.4 Advanced tools I: Switched LPV design

Where have progressively reached a LPV design with a proper performance and domain of attraction in the presence of speed variation and curved roads. Nevertheless, there are certain sources of conservativeness in the previous designs that have not been considered yet. The first one comes from the large range of variation on the vehicle speed: whenever we have a large parameter variation, it could remain conservative to use a single LPV controller over the whole admissible range of the parameter variation, up to the extent that only a solution existed for a 20[km/h] speed variation range in the previous designs. In order to reduce such conservativeness, it is possible to tune multiple LPV controllers, each one suitable for a defined parameter subspace. This provides more flexibility on the design stage, but global stability of the switched LPV system needs to be checked a posteriori.

In this thesis six different overlapped switching modes denoted by the subindex  $\sigma \in \{\sigma_1, \dots, \sigma_6\}$  defining an hysteresis switch [Lu 2004], [Lu 2006] have been considered. Each mode will be active at consecutive 20[km/h] intervals, covering the full range of speed  $v \in [30 : 150]$ km/h. In this way, a discontinuous parameter-dependent Lyapunov function (4.19) is defined over the range of operation by means of multiple parameter-dependent Lyapunov functions, each one ensuring the closed-loop stability on the corresponding parameter subspace.

$$V_\sigma(x, \nu) = x^T G_\sigma^{-1}(\nu)x. \quad (4.19)$$

For the LPV switched closed-loop system working under the hysteresis switching strategy to be stable, the value of the discontinuous Lyapunov function is not neces-

sary decreasing along the whole range of the parameter and may have discontinuities due to the switch. Nevertheless, it needs to be ensured that the switch is performed safely, this means that the value of the Lyapunov function has to be globally decreasing along the *dwell time* ( $k_{dw}$ ):

$$V_{\sigma_1}(x_k, \nu) > V_{\sigma_2}(x_{k+k_{dw}}, \nu). \quad (4.20)$$

In Figure 4.18, the evolution of a switching signal between  $\sigma_1$  and  $\sigma_2$  is shown. It can be seen that when the vehicle is running at low speed, the first mode  $\sigma_1$  with  $\bar{A}_{cl\sigma_1}(\nu)$  and the  $G_{\sigma_1}^{-1}$  matrix defining the Lyapunov function of the closed-loop system is active. Then, when the vehicle increases its speed, the overlapping region is reached. Nevertheless, the first mode is active until the switching surface  $\mathcal{S}_{12}$  is reached. If the controller switching fulfills (4.20), the LPV closed-loop switched system is ensured to be stable. This is translated into the following condition

$$x_k^T G_{\sigma_1}^{-1}(\nu) x_k - x_{k+k_{dw}}^T G_{\sigma_2}^{-1}(\nu) x_{k+k_{dw}} > 0, \quad (4.21)$$

or equivalently,

$$x_k^T (G_{\sigma_1}^{-1}(\nu) - A_{cl\sigma_1}^{k_{dw}}(\nu) G_{\sigma_2}^{-1}(\nu) A_{cl\sigma_1}^{k_{dw}}) x_k > 0, \quad (4.22)$$

where

$$A_{cl\sigma}^{k_{dw}}(\nu) = \prod_{d=k}^{d=k+k_{dw}-1} A_{cl\sigma}(\nu_d). \quad (4.23)$$

Via Schur complement [Boyd 1994], the following LMI is obtained

$$\begin{bmatrix} G_{\sigma_2}^{-1}(\nu) & A_{cl\sigma_1}^{k_{dw}} \\ \star & G_{\sigma_1}(\nu) \end{bmatrix} \succ 0. \quad (4.24)$$

In a similar way, when the vehicle is traveling at higher speed, the second mode  $\sigma_2$  with  $\bar{A}_{CL\sigma_2}(\nu)$  and  $G_{\sigma_2}$  is active. If then the vehicle decelerates, the switch will be performed if the switching surface  $\mathcal{S}_{21}$  is reached, and the following LMI must hold in order to ensure the global decrease of the discontinuous Lyapunov function:

$$\begin{bmatrix} G_{\sigma_1}^{-1}(\nu) & A_{cl\sigma_2}^{k_{dw}} \\ \star & G_{\sigma_2}(\nu) \end{bmatrix} \succ 0. \quad (4.25)$$

In practice, the switching strategy is based on the parameter  $\nu$ , defined by the vehicle speed. This means that the parameter variation is delimited by the maximal acceleration capabilities of the vehicle. In this way, once the subspace regions are defined and the respective closed-loop controllers are designed, the closed loop dynamics are known along the dwell time, and inequalities (4.24), (4.25) can be verified in a straightforward manner.

The obtained tuning provides the following ellipsoidal invariant sets for each one of the parameter sub-spaces, shown in Figure 4.19.

In Figures 4.20(a) and 4.20(b) a speed-curvature profile varying from 30 to 150 [km/h] with the different switching modes is introduced in order to show the performance of this switched LPV control design strategy. Figure 4.21 depicts the

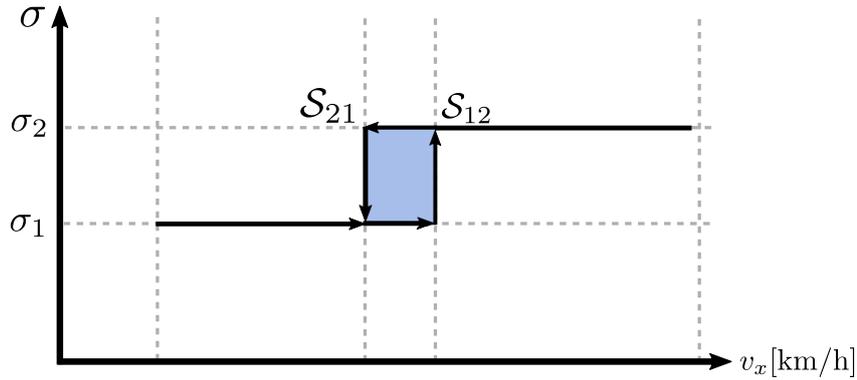
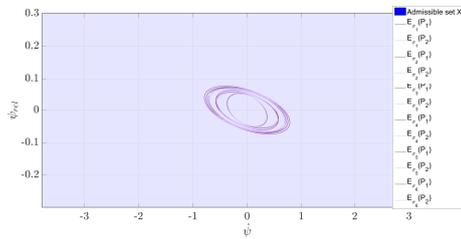
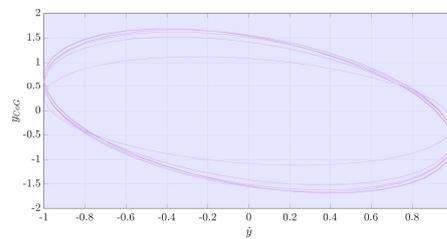


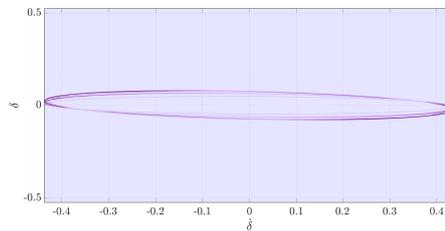
Figure 4.18 – Hysteresis switching strategy



(a) Proj( $x_1, x_2$ ) plane

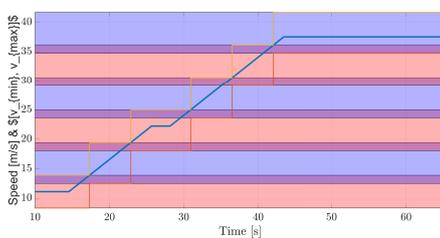


(b) Proj( $x_3, x_4$ ) plane

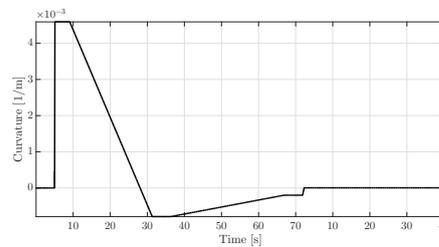


(c) Proj( $x_5, x_6$ ) plane

Figure 4.19 – Robust domain of attraction for parameter-dependent switched LPV design,  $\mathcal{E}_{\sigma_i}(P_i)$



(a) Speed profile



(b) Curvature profile

Figure 4.20 – Speed and Curvature Profile for switched LPV design

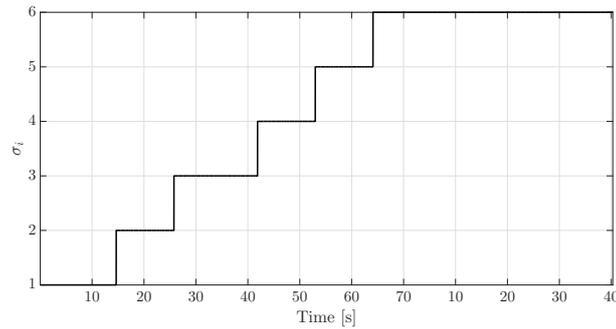


Figure 4.21 – System input trajectory for switched LPV design

evolution of the switching signal along the simulation, activating and deactivating the corresponding modes with the increment of the vehicle speed.

In Figures 4.22 and 4.23, we can observe that the system performance is kept even in the presence of large speed variations together with the consideration of curved roads, providing a control design suitable for the full range of the system operating conditions.

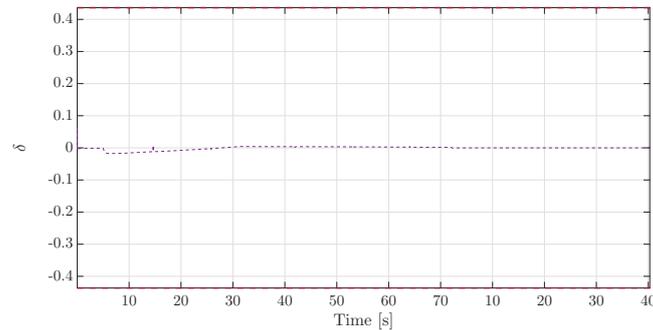


Figure 4.22 – System input trajectory for switched LPV design

#### 4.2.5 Advanced tools II: Acceleration considerations

A second source of conservatism is the consideration of a control design that needs to be performant in a large range and under any sudden change on the parameter value. Briefly speaking, such a design is theoretically seeking for a tuning which works under any infinitely huge variation on the parameter, which is translated in a reduced domain of attraction. In order to mitigate this effect, it can be taken into account the limited range of variation of the parameter.

The maximal parameter's rate of change  $\Delta\nu$  can be translated into a maximal variation of the Lyapunov function, including this information in the LMI condition of robust positive invariance (2.90), this information will be taken into account, and the infinite variation of the parameter is no longer assumed. The parameter value at

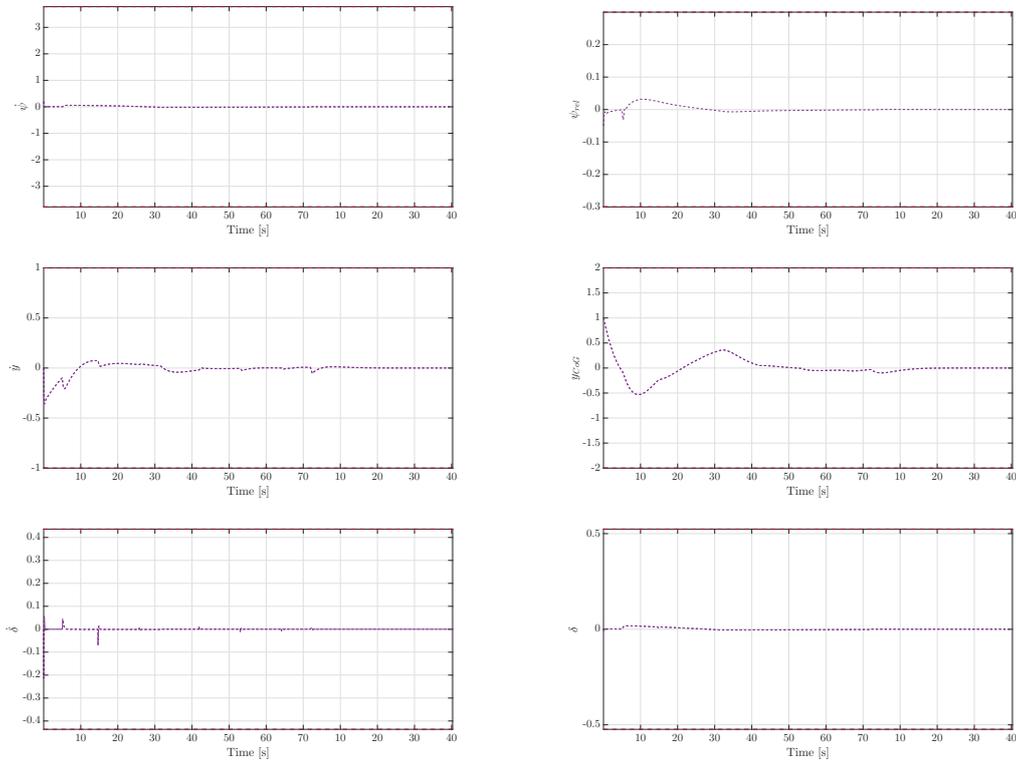


Figure 4.23 – System states trajectory for switched LPV design

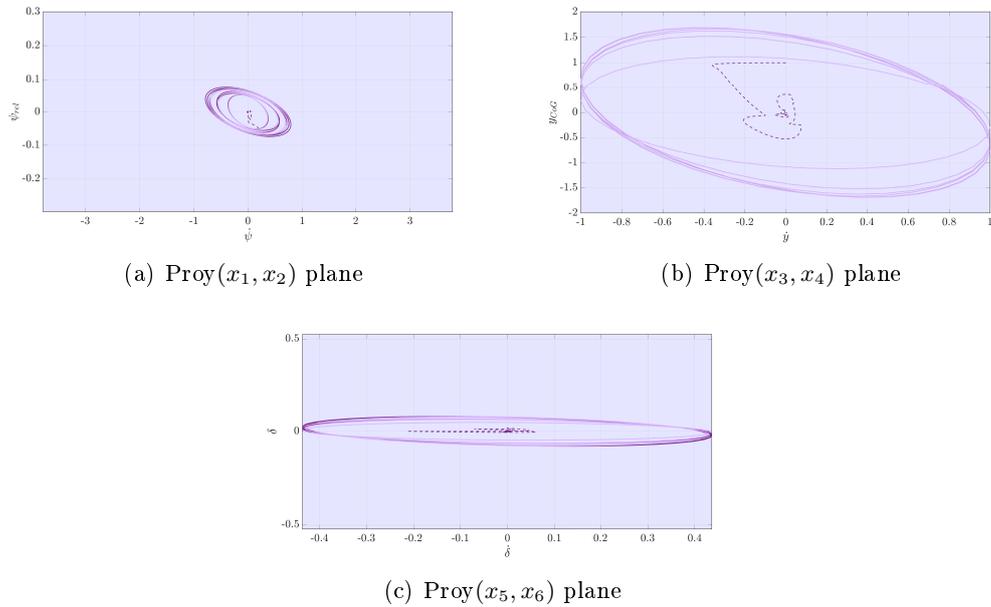


Figure 4.24 – System states trajectory and ellipsoidal invariant sets projections for switched LPV design

time  $k$  can be expressed as an affine combination of the parameter extreme values,

$$\nu_k = \sum_{i=1}^{n_v} \lambda_{i_k} \nu_i. \quad (4.26)$$

If we define the maximal variation on the parameter between time instant  $k$  and  $k + 1$  as  $\Delta\nu$ , we have

$$\|\nu_k\| - \|\Delta\nu\|_{max} \leq \|\nu_{k+1}\| \leq \|\nu_k\| + \|\Delta\nu\|_{max}, \quad (4.27)$$

For the robust control design purposes, we start from the definition of a decreasing parameter-varying Lyapunov function,

$$V_{k+1} - V_k = x_{k+1}^T \mathcal{P}(\nu_{k+1}) x_{k+1} - x_k^T \mathcal{P}(\nu_k) x_k < 0, \quad (4.28)$$

in LMI form,

$$\begin{bmatrix} x_k & \rho_k \end{bmatrix} \begin{bmatrix} A_{cl}^T(\nu) \mathcal{P}(\nu_{k+1}) A_{cl}(\nu) - \mathcal{P}(\nu_k) & A_{cl}^T(\nu) \mathcal{P}(\nu_{k+1}) E(\gamma_k) \\ \star & E^T(\gamma_k) \mathcal{P}(\nu_{k+1}) E^T(\gamma_k) \end{bmatrix} \begin{bmatrix} x_k \\ \rho_k \end{bmatrix}, \quad (4.29)$$

where  $A_{cl}(\nu) = (A(\nu_k) + BK(\nu))x_k$ , with

$$\mathcal{P}_k(\nu_k) = \sum_{i=1}^{n_v} \lambda_{i_k} P_i(\nu_i), \text{ with } i = 1 \dots n_v \quad (4.30)$$

being the Lyapunov function at time instant  $k$ . And

$$\mathcal{P}_k(\nu_{k+1}) = \sum_{i=1}^{n_v} \lambda_{i_{k+1}} P_i(\nu_i) \quad (4.31)$$

being the Lyapunov function at time instant  $k + 1$  with  $i = 1 \dots n_v$ .

In addition, the maximal variation on the parameter (4.27) can be translated into a maximal variation on the Lyapunov function, hence

$$\mathcal{P}_{k+1}(\nu_{k+1}) = \sum_{i=1}^{n_v} \lambda_{i_{k+1}} \mathcal{P}_{i_{k+1}}(\nu_{i_{k+1}}) = \quad (4.32)$$

$$= \sum_{i=1}^{n_v} \lambda_{i_k} \mathcal{P}_{i_k}(\nu_{i_k}) + \sum_{i=1}^{n_v} \Delta\lambda_{i_{k+1}} \mathcal{P}_{i_k}(\nu_{i_k}) = \quad (4.33)$$

$$= \mathcal{P}_k(\nu_k) + \sum_{i=1}^{n_v} \Delta\lambda_{i_{k+1}} \mathcal{P}_{i_k}(\nu_{i_k}). \quad (4.34)$$

**Assumption 4.1** *There is an order in the obtained Lyapunov functions for each one of the vertex realizations, we have*

$$\mathcal{P}_1 \prec \mathcal{P}_2 \prec \dots \mathcal{P}_{n_v}. \quad (4.35)$$

This assumption can turn to be conservative and can be proven that there exists solutions that do not verify this condition, for example by imposing that the obtained solutions  $\mathcal{P}_{k+1}(\nu_{k+1})$  are bounded by:

$$\mathcal{P}_k(\nu_k) + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (\mathcal{P}_i - \mathcal{P}_j) \prec \mathcal{P}_{k+1}(\nu_{k+1}) \prec \mathcal{P}_k(\nu_k) + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (\mathcal{P}_j - \mathcal{P}_i) \quad (4.36)$$

for  $\mathcal{P}_i \prec \mathcal{P}_j$ ,  $j = 1, \dots, n_v$  and  $j \neq i$ . That is, we assume that  $\mathcal{P}_i$  is the *smallest* of the obtained Lyapunov functions.

In the following, we seek for the LMI condition that ensures that (4.28) holds  $\forall x_k$  and  $\rho_k$  satisfying

$$x_{k+1}^T \mathcal{P}(\nu_{k+1}) x_{k+1} \leq 1, \quad (4.37a)$$

$$w^T w \leq 1. \quad (4.37b)$$

In LMI form,

$$\begin{bmatrix} x_k & w_k \end{bmatrix} \begin{bmatrix} P^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} \leq 1, \quad (4.38a)$$

$$\begin{bmatrix} x_k & w_k \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} \leq 1. \quad (4.38b)$$

By means of the S-procedure [Luca 2011a] we arrive to the RPI invariance condition with a parameter-varying Lyapunov function,

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i}^{-1} - P_k^{-1} & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau P_k^{-1} & 0 \\ \star & \star & \star & P_{k+1}^{-1} \end{bmatrix} \succ 0. \quad (4.39)$$

Considering the information on maximal variation on the Lyapunov function (4.36), equation (4.39) can be transformed in order to contain such information. In order to do this, we consider the worst case of variation for  $\mathcal{P}_k + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_i^{-1} - P_j^{-1}) \prec \mathcal{P}_{k+1}$ . We obtain,

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i}^{-1} - P_k^{-1} & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau P_k^{-1} & 0 \\ \star & \star & \star & P_k^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_i^{-1} - P_j^{-1}) \end{bmatrix} \succ 0. \quad (4.40)$$

Replacing  $P_k = \sum_{i=1}^{n_v} \lambda_i P_i$ , we have,

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i}^{-1} - P_i^{-1} & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau P_i^{-1} & 0 \\ \star & \star & \star & P_i^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_i^{-1} - P_j^{-1}) \end{bmatrix} \succ 0. \quad (4.41)$$

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i}^{-1} - P_h^{-1} & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau P_h^{-1} & 0 \\ \star & \star & \star & P_h^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_i^{-1} - P_j^{-1}) \end{bmatrix} \succ 0. \quad (4.42)$$

for  $h = 1, \dots, n_v$ , and  $h \neq i$ . Recalling Assumption 4.1, we are supposing that  $P_1$  is the smallest Lyapunov function. Nevertheless, we do not know a priori in which vertex of the polytopic system this holds. Thus, we need to check all the possibilities: the *smallest* value of  $P_i^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_i^{-1} - P_j^{-1})$  or  $P_j^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_j^{-1} - P_i^{-1})$  would be obtained for the *smallest*  $P_i$  for  $i = 1, \dots, n_v$ .

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i}^{-1} - P_i^{-1} & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau P_i^{-1} & 0 \\ \star & \star & \star & P_i^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_i^{-1} - P_j^{-1}) \end{bmatrix} \succ 0. \quad (4.43)$$

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i}^{-1} - P_h^{-1} & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau P_h^{-1} & 0 \\ \star & \star & \star & P_h^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_i^{-1} - P_j^{-1}) \end{bmatrix} \succ 0. \quad (4.44)$$

for  $h = 1, \dots, n_v$ , and  $h \neq i$ . Then can also be written as,

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i}^{-1} - P_i^{-1} & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau P_i^{-1} & 0 \\ \star & \star & \star & P_i^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_i^{-1} - P_j^{-1}) \end{bmatrix} \succ 0. \quad (4.45)$$

for  $i = 1, \dots, n_v$ , and

$$\begin{bmatrix} Q_{G_i}^T + Q_{G_i}^{-1} - P_h^{-1} & 0 & \tau Q_{G_i}^T & Q_{G_i}^T \bar{A}_i^T - Y_i^T \bar{B}^T \\ \star & \beta I & 0 & [\bar{E}_i \ 0]^T \\ \star & \star & \tau P_h^{-1} & 0 \\ \star & \star & \star & P_h^{-1} + \|\Delta\nu\|_{\max} \sum_{j=1, i \neq j}^{n_v} (P_j^{-1} - P_i^{-1}) \end{bmatrix} \succ 0. \quad (4.46)$$

for  $i = 1, \dots, n_v$ ,  $h = 1, \dots, n_v$  and  $h \neq i$ .

Considering this information at the design stage, we reduce the conservative consideration on the infinitely large variation of the parameter, thus we are able to compute a robust LPV control design for the full range of variation of the speed of the vehicle  $v \in [30 : 150]$ km/h and the curvature of the road additive disturbances.

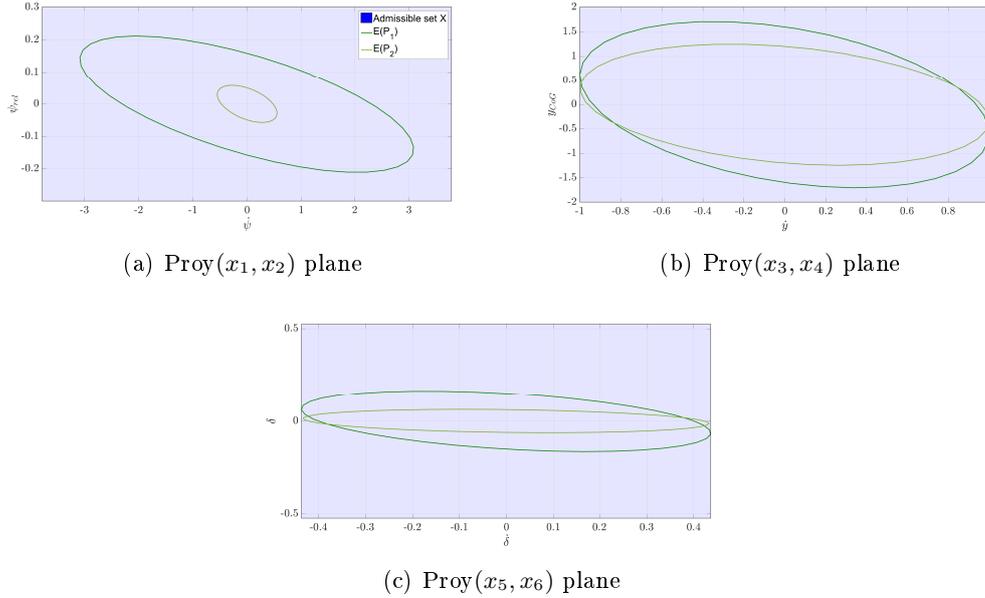


Figure 4.25 – Robust domain of attraction for parameter-dependent LPV design,  $\mathcal{E}(P_i)$  with maximal acceleration considerations

It is straightforward to notice in Figure 4.25 that the ellipsoidal sets for the extreme values of the parameter are not so similar to each other as in the previously developed designs, as the conservativeness of the previous designs produces this similarity between the extreme invariant sets, as we need to be able to change from one to the other infinitely fast.

Considering this controller design for the same speed-curvature values shown in Figure 4.20, we can see that a robust control design can be obtained for the full range of variation of the speed and in the presence of modeled parameter disturbances, as a convex combination of two unique extreme values,  $n_v = 2$ . Temporal results of the state and input trajectories are shown in Figures 4.26, 4.27 and 4.28.

#### 4.2.6 MPC for LCA system control

To end up the control design section, Model Predictive Control (MPC) (Section 2.3.2) strategy is proposed to enhance the constraints handling capabilities and the domain of attraction by using a finite horizon optimization, all by relaxing the linear structure of the LPV feedback computed in the previous designs.

Within this framework, it is necessary to foresee the geometry of the road tra-

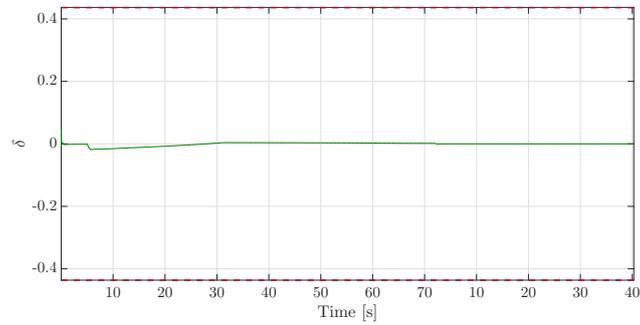


Figure 4.26 – System input trajectory for LPV design with maximal acceleration considerations

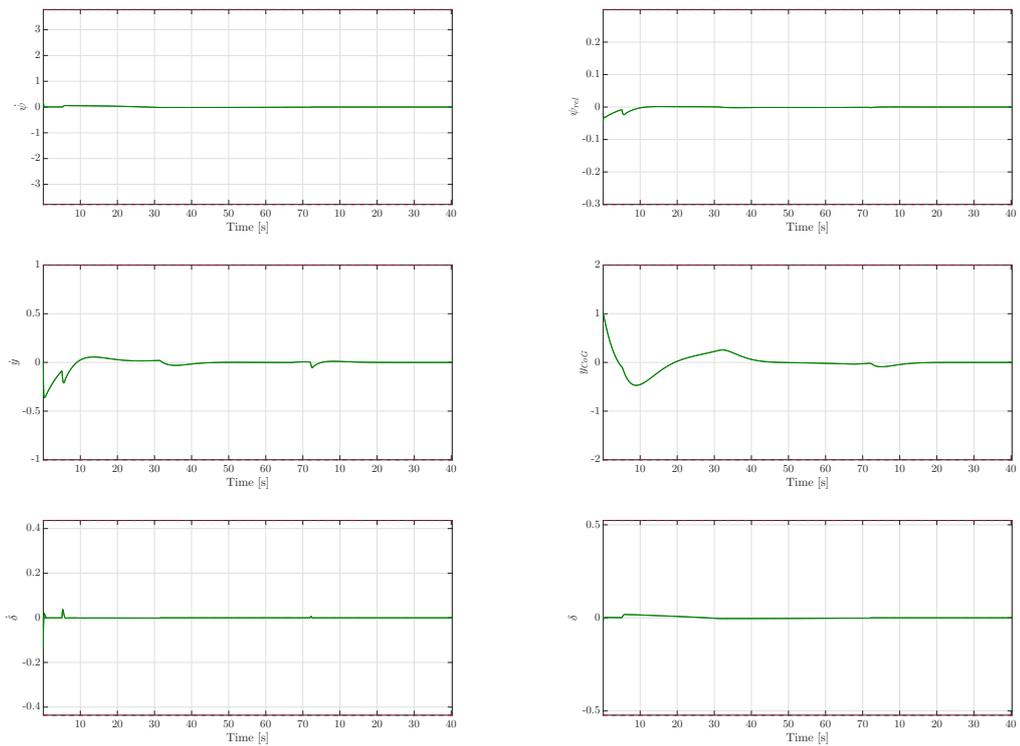


Figure 4.27 – System states trajectory for LPV design with maximal acceleration considerations

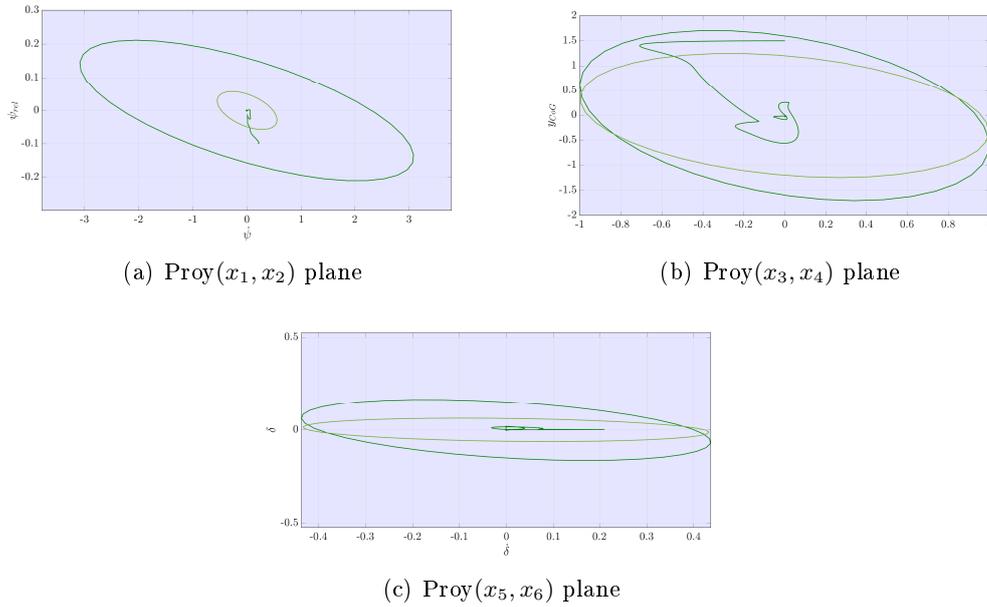


Figure 4.28 – System states trajectory and ellipsoidal invariant sets projections for LPV design with maximal acceleration considerations

jectory. Up to this point the curvature has been considered in terms of its bounded variation without any supplementary restrictions. However, local information on the longitudinal direction can be used for extrapolation. This allows transform the MPC prediction model to include the curvature of the road.

In the following, a third order polynomial is used  $c(\chi) = c_0 + c_1\chi + c_2\chi^2 + c_3\chi^3$ , where  $\chi$  represents the longitudinal distance with respect to the current position of the vehicle and  $c_i$  are the online measured coefficients of the reference trajectory that approximates the lane. From this model, it is possible to compute the curvature of the road at any distance  $\chi_k$  where we will be at any future time  $k$  by means of equation (4.47), where  $(.)'$  implies the derivative of the polynomial with respect to  $\chi$ .

$$\rho_k = \frac{c''(\chi_k)}{(1 + c'(\chi_k))^{(3/2)}} \quad (4.47)$$

When there is such a prediction model available (4.47) to calculate the incoming additive disturbance along the prediction horizon  $N$ , its effect can be incorporated to the predictive control strategy [J.M.Maciejowski 2002] by inclusion on the system dynamics model (4.10) and approximately cancel it by a suitable control action, commonly known as *feedforward* control. The final MPC problem formulation is

stated in the following

$$\begin{aligned}
\min_U J(x_k, w_k, U) &= \|x_N\|_P^2 + \sum_{k=1}^{N-1} \|x_k\|_Q^2 + \|u_k\|_R^2 \\
\text{s.t. } x_{k+1} &= A(\nu_k)x_k + Bu_k + E(\gamma_k)\rho_k \\
y_k &= Cx_k \\
x &\in X, u \in U \\
x_N &\in \tilde{X}_N
\end{aligned} \tag{4.48}$$

with  $Q$ ,  $R$ ,  $U$  and  $x_k$  defined as in Section 2.3.2 and  $\rho_k$  being the curvature of the road. Finally, the quadratic terminal cost  $P$  and terminal set  $\tilde{X}_N$  are defined by the RPI set.

**Remark 4.3** *System performance can be improved by means of using the curvature value measured at a certain longitudinal distance from the current vehicle position, as a human driver naturally does when looks at a point in the horizon.*

### 4.3 Conclusion

LCA system dynamics modeling in the presence of speed variation have been described by a parameter-varying model, where the parameter is bounded and measured. In the same way, different methods for modeling the effects of a curved road on the system have been introduced, by means of different parameter-varying additive disturbances models which differ on the level of conservativeness.

Once the system has been delimited, robust positive invariance theory has been exploited as a main tool to certify the behavior of a LPV controller that does not take the impact of the additive disturbances into account on the design stage. After that, the design of an ISS LPV observer-based controller design that maximizes the size of the RPI set, taking into account input, and internal states constraints has been proposed. In addition to this, two strategies that reduce the conservativeness have been studied: a LPV switched control strategy and acceleration considerations. The application of such techniques has allowed to largely expand the range of speed variation with respect to the initial robust LPV techniques.

Nevertheless, its working area is still limited, and a Model Predictive Control strategy that predicts the curvature of the road by means of a polynomial model and anticipates to its effect is considered. Moreover, a terminal cost and a parameter-varying stabilizing gain that maximizes the robust terminal set in the presence of system constraints and the modeled additive disturbances are designed to ensure the controller recursive stability for such scenarios.

Numerical simulations on an example scenario have been provided, showing the proposed solution performance in the presence of both speed and curvature variations.



Part II

TRAJECTORY PLANNING FOR LANE  
CHANGE AND OVERTAKING ON  
HIGHWAYS



# Theoretical background for trajectory planning

---

## Contents

---

<b>5.1</b>	<b>Optimal Control</b> . . . . .	<b>100</b>
5.1.1	Basic Definitions . . . . .	100
5.1.2	Optimal Control Problem formulation . . . . .	101
5.1.3	Numerical methods for solving Optimal Control Problems	101
5.1.3.1	Dynamic Programming . . . . .	102
5.1.3.2	Indirect methods . . . . .	103
5.1.3.3	Direct methods . . . . .	104
5.1.4	Numerical Integration - Runge-Kutta scheme . . . . .	108
<b>5.2</b>	<b>Flatness</b> . . . . .	<b>110</b>
<b>5.3</b>	<b>Hyperplane Arrangements</b> . . . . .	<b>111</b>
5.3.1	Basic definitions . . . . .	111
5.3.2	Non-convex non-connected regions characterization . . . . .	113
5.3.3	Cell Merging . . . . .	115
5.3.4	Mixed Integer representation . . . . .	117
5.3.4.1	MIP formulation complexity reduction . . . . .	118
<b>5.4</b>	<b>Conclusion</b> . . . . .	<b>122</b>

---

This chapter brings into focus the relevant theoretical framework for the second part of this manuscript, namely the trajectory generation in the context of overtaking or lane change maneuvers on highways and in the presence of other vehicles, described as possible obstacles.

The proposed framework is based on optimization, so the first section's attention is dedicated to the Optimal Control theory, providing a brief overview of most of the existing numerical methods. To complement this part, basic concepts of flatness theory are recalled, in view of addressing trajectory planning objectives. Finally, Section 5.3 introduces the main notions related to Hyperplane Arrangements, which constitute the theoretical foundation for the anti-collision constraints formulation in the proposed solution.

## 5.1 Optimal Control

Optimal Control theory is a broad topic itself, covering more than half-century of the control research. In the following, the major lines of some of the existing methods are drawn in order to provide an insight on this well established control problem. Furthermore, references are provided all along the section, and the reader is referred to those and the bibliographical references therein for and in depth coverage of the subject.

### 5.1.1 Basic Definitions

In order to establish the mathematical framework let us consider a dynamical model in the form of an Ordinary Differential Equation (ODE) whose behavior is characterized on a given time interval  $[t_{init}, t_{end}]$ . This represents the simplest case for describing the evolution of a controlled dynamic system, given by

$$\dot{\xi} = f(\xi(t), u(t), t) \quad (5.1)$$

where  $t \in \mathbb{R}$  is the time,  $u(t) \in \mathbb{R}^{n_u}$  are the controls, and  $\xi(t) \in \mathbb{R}^{n_x}$  is the state. The function  $f$  is a map from the states, controls and time to the range of change of the state, i.e.  $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times [t_{init}, t_{fin}] \rightarrow \mathbb{R}^{n_x}$ . Due to the explicit time dependence of the function  $f$ , this formulation covers the class of time-variant systems.

**Remark 5.1** *An ODE leads to the definition of a solution in terms of a function of one variable  $\xi(t)$ , which is the time. The case of differential equations with solutions characterized by functions of more than one variable are denoted as part of the class of Partial Differential Equations (PDE) and are not considered here, mainly due to the fact that the automotive applications as the one considered in the longitudinal and lateral control cases are seldom varying in space.*

**Definition 5.1** Initial Value Problem (IVP). Given a set of initial values for the dependent variables  $\xi(t)$ , called the *initial conditions*, one must determine their values at some other final point  $T_F$ . The problem would be stated as follows: *Compute the value of  $\xi(T_F)$  for some initial value at  $T_0 < T_F$  that satisfies (5.1) with the known initial value  $\xi(T_0) = \xi_{T_0}$ .* This kind of problem will thus be fully defined by the ODE and a given initial condition:

$$\dot{\xi} = f(\xi(t), u(t), t), \quad \xi(T_0) = \xi_{T_0} \quad (5.2)$$

□

**Definition 5.2** Boundary Value Problem (BVP). Given a set of initial and final values for the dependent variables, one must determine the dependent variables such that they have these specific values at two (or more) points, denoted as  $T_0$  and  $T_F$ . The conditions that define the dependent variables are called the *boundary conditions*,  $\xi(T_0) = \xi_{T_0}$ ,  $\xi(T_F) = \xi_{T_F}$ . □

**Definition 5.3** A Continuous Optimization problem is such that the decision variable is part of a smooth manifold, for example, real valued vectors  $x \in \mathbb{R}^{n_x}$ .  $\square$

**Definition 5.4** An Integer Optimization problem is such that the decision variable is an integer value  $z \in \mathbb{Z}^{n_z}$ , discrete or belongs to a set of binary choices  $z \in \{0, 1\}^{n_z}$ . This last case is also known as *combinatorial optimization*.  $\square$

**Remark 5.2** [Prodan 2015]. If an optimization problem has both, continuous and integer variables, it is called a *mixed-integer optimization problem*.  $\square$

### 5.1.2 Optimal Control Problem formulation

Getting closer to the models used in automotive dynamics representation, let us consider the following dynamical system:

$$\dot{\xi}(t) = f(\xi(t), u(t)) \quad (5.3)$$

where  $\xi(t)$  stands for the dynamical system states vector and  $u(t)$  denotes the system input. The problem of selection of a solution can be formulated in terms of a conventional Continuous Optimal Control problem (OCP) including Ordinary Differential Equations (ODE) constraints,

$$\min_{\xi(t), u(t), T_F} J = \min_{\xi(t), u(t), T_F} \int_{T_0}^{T_F} L(\xi(t), u(t)) dt + E(\xi(T_F)) \quad (5.4)$$

$$\begin{aligned} \text{s.t. } \quad & \xi(0) - \xi_{T_0} = 0, & \text{Initial conditions} \\ & \dot{\xi}(t) - f(\xi(t), u(t)) = 0, & \text{ODE model} \\ & q(\xi(t), u(t)) \geq 0, & \text{Path constraints} \\ & \xi(T_F) - \xi_{T_F} = 0, & \text{Final conditions} \end{aligned} \quad (5.5)$$

with  $t \in [T_0, T_F]$  and  $T_F$  being the time horizon length. This scalar value can be set beforehand or left as an optimization parameter.

The performance objective of the cost function, denoted  $L(\xi(t), u(t))$  is also known as the Lagrange term and the terminal cost  $E(\xi(T_F))$  is often referred to as Mayer term [Diehl 2014]. In Figure 5.1 a visualization of this generic OCP formulation together with the problem constraints is shown.

### 5.1.3 Numerical methods for solving Optimal Control Problems

There exists a wide range of numerical methods that can be used to solve constrained OCP (5.4, 5.5), the maturity of this field being sustained by the abundant literature concerned with this subject. In the subsequent, a brief overview based on [Betts 2010], [Bryson 1975], [Hull 2013], [Kirk 2012], [Diehl 2014] is presented.

The exposition is divided in three different approaches that can be considered when solving this kind of problems: Dynamic programming, Indirect and Direct methods (Figure 5.2).

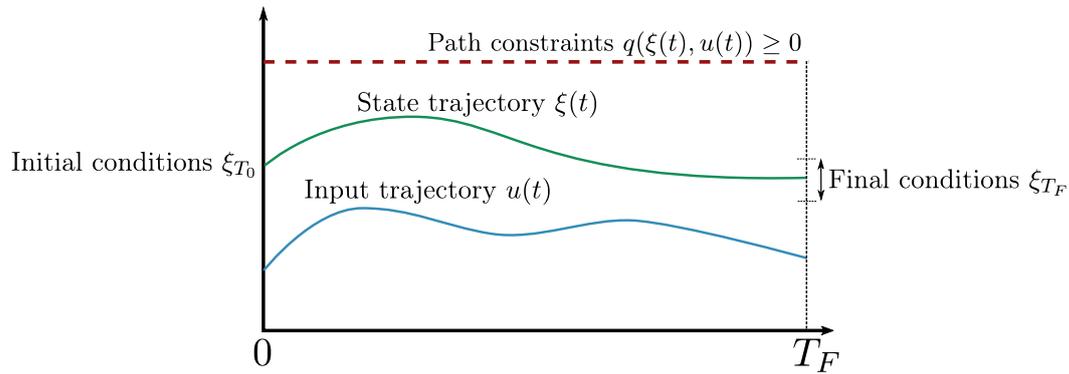


Figure 5.1 – Optimal control problem

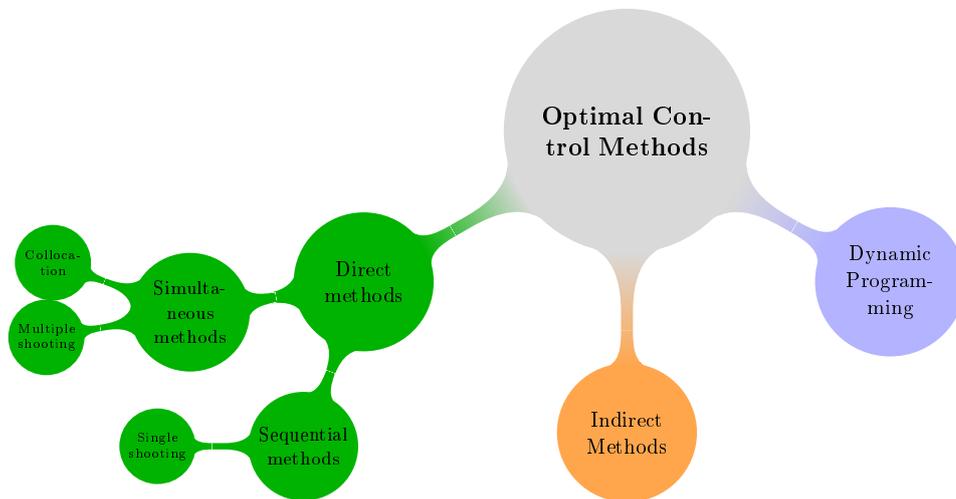


Figure 5.2 – Optimal Control methods classification

### 5.1.3.1 Dynamic Programming

This class of methods is based on the approach which considers that it is possible to enumerate in a suitable way all possible trajectories of the system dynamics, based on Bellman's principle of optimality.

**Definition 5.5** [Bellman 1957]. The *Principle of Optimality* states that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first optimal decision.  $\square$

In other words, this means that any subarc of an optimal trajectory is also optimal. This principle can either be applied on a discrete-time Dynamic Programming (DP) recursion [Bertsekas 1995] or in the continuous-time framework with infinitely small steps, leading to the Hamilton-Jacobi-Bellman (HJB) partial differential equations

(PDE) on the states [Kirk 2012]. This class of method exploits the principle of optimality for the computation of a sequence of trajectory (sub-arcs), which together define an optimal trajectory. Let us start from the optimal control problem cost function in (5.4), defining a BVP in the time interval  $[T_0, T_F]$ .

Starting from a given initial condition  $\bar{\xi}_{t_k}$ , with a time grid of  $K$  steps defined on the interval  $[T_0, T_F]$ , the cost function can be rewritten for a generic time sub-interval  $[t_k, t_{k+1}]$ ,

$$\min_{\xi(t), u(t), t_{k+1}} J_k(\xi(t_k), u(t), t_k) = \min_{\xi(t), u(t), t_{k+1}} \int_{t_k}^{t_{k+1}} L(\xi(t), u(t)) dt + E(\xi(t_{k+1})) \quad (5.6)$$

s.t.

$$\begin{aligned} \xi(t_k) - \bar{\xi}_{t_k} &= 0, & \text{Sub-arc initial conditions} \\ \dot{\xi}(t) - f(\xi(t), u(t)) &= 0, & \text{ODE model} \\ q(\xi(t), u(t)) &\geq 0, & \text{Path constraints} \\ \xi(t_{k+1}) - \xi_{t_{k+1}} &= 0, & \text{Sub-arc final conditions} \end{aligned} \quad (5.7)$$

where  $J_k(\xi(t_k), t_k)$  is denoted as the optimal cost-to-go at time  $t_k$  when starting at the defined state  $\bar{\xi}_{t_k}$ . Moreover, by means of the principle of optimality (Definition 5.5),  $E(\xi(t_{k+1}))$  corresponds with  $J_{k+1}(\xi(t_{k+1}), t_{k+1})$ , which will be defined by the cost-to-go of the following trajectory sub-arc, when starting at the given point  $\bar{\xi}_{t_{k+1}} = f(\xi(t_k), u)$ ,

$$J_k(\bar{\xi}(t_k)) = \arg \min_u L(\xi(t_k), u) + J_{k+1}(f(\xi(t_k), u)) \quad (5.8)$$

In this way, the main idea of this methodology is to propagate recursively the cost-to-go function (5.6, 5.7) backwards in time, starting from  $J_{T_F}(\xi(T_F), T_F) = E(\xi(T_F))$ , and computing the function value  $J_k(\xi(t_k), t_k)$  for the whole tabulated state space  $\bar{\xi}_{t_k} \in \mathbb{X}$  for the defined time grid with  $k \in \{K-1, \dots, 0\}$ . This essentially would be a matter of trying all the allowable control values at each of the allowable state points, identifying the optimal control  $u^*(\xi(t_k), t_k)$  with the one that provides the minimum value of the cost function,  $J_k^*(\bar{\xi}(t_k))$ .

$$u^*(\xi(t_k), t_k) = \arg \min_u L(\xi(t_k), u) + J_{k+1}^*(f(\xi(t_k), u)), \quad \text{for } k \in \{K-1, \dots, 0\} \quad (5.9)$$

### 5.1.3.2 Indirect methods

The basis of indirect methods is the Pontryagin's maximum principle [Pontryagin 1987], which is used to solve the first-order optimality conditions [Bryson 1975], [Hull 2013] of an OCP in order to obtain an optimal solution.

The solution of the OCP being considered is a control input function  $u(t)$  that minimizes the cost function (5.4) subject to the differential constraints and the given initial and final conditions (5.5). Let us recall that the Hamiltonian function is defined for the considered performance function (5.4) as

$$H(\xi(t), u(t), \lambda(t), t) = \mathcal{L}(\xi(t), u(t)) + \lambda^T f(\xi, u(t)) \quad (5.10)$$

where the Lagrangian will be defined as

$$\mathcal{L}(\xi(t), u(t)) = f(\xi, u(t)) + \lambda^T q(\xi, u(t)) \quad (5.11)$$

where  $\lambda$  represents the vector of lagrangian multipliers. considering the first order optimality conditions, we obtain the optimal control by minimizing the Hamiltonian function  $H$  with respect to  $u(t)$

$$\frac{\partial H}{\partial u} = 0 \rightarrow u^*(t) \quad (5.12)$$

Once the optimal signal  $u^*(t)$  is available, we compute the trajectories of all the states, that will provide the desired reference trajectories. These will be computed through the trajectories of the Lagrangian multipliers, using the adjoint differential equations,

$$\dot{\lambda}_i(t) = -\frac{\partial H}{\partial \xi_i} = 0 \quad (5.13)$$

Practically, integrating over time, we can obtain the Lagrange multipliers and states trajectories, whose parameters can be computed from the defined boundary conditions.

### 5.1.3.3 Direct methods

Direct methods seek for the optimal solution of the OCP by directly minimizing the problem's cost function. A main step on this methodologies is to perform the discretization of the state and/or control trajectories, approximating the original continuous time infinite dimension OCP by a finite dimensional Nonlinear Program (NLP), which can be afterwards solved by the corresponding strategies. This goal can be achieved by transcribing the infinite dimensional problem into a finite-dimensional approximation. This reformulation has three fundamental steps:

1. Convert the continuous control problem into a problem with a finite set of variables.
2. Solve the finite-dimensional problem using an optimization method (the NLP subproblem).
3. Assess the accuracy of the finite-dimensional approximation and if necessary repeat the transcription and optimization steps.

In the following, a review on the transcription methods that can be used to approximate an OCP into a NLP problem is provided, focusing on the first step of the process, that is, identifying the NLP variables, constraints and objective function. A further classification of the direct methods can be done regarding the discretization approach, denoted as sequential or simultaneous discretization direct approaches (Figure 5.2). In the first class, discretization is carried out only for the control vectors, while the simultaneous discretization will do for both states and control trajectories.

### Sequential approach

As it has been previously stated, only the control inputs are discretized in this strategy, leaving the states vector  $\xi(t)$  as a dependent variable of the control input vector  $u(t)$  and the initial state vector  $\xi_0$ . In other words, only the controls are the decision variables of the resulting NLP. The state trajectories are explicitly handled by an additional numerical integration method (Section 5.1.4) in a sequential manner, starting from  $\xi_0$  and using the computed controls to forward integrate the system dynamics.

- **SINGLE SHOOTING METHOD.** Control parameterization is performed by means of polynomials, piecewise constant functions or, more generally, piecewise polynomials. In the following, piecewise constant controls parameterization is considered, as it is the most widespread parameterization. A fixed grid with  $K$  steps is defined along the time horizon  $t_0 < t_1 < \dots < t_K = T_F$ , obtaining the vector of parameterized controls  $q_i \in \mathbb{R}^{n_u}$ , with  $i = 0 \dots K - 1$ . After that, the resulting control function  $u(t, q)$  can be obtained by

$$u(t, q) = q_i \text{ for } t \in [t_i, t_{i+1}] \quad (5.14)$$

Then, the state integration  $\xi(t, q)$  can be solved as a initial value problem over the entire prediction, using the controls  $u(t, q)$  (5.14). Finally, inequality

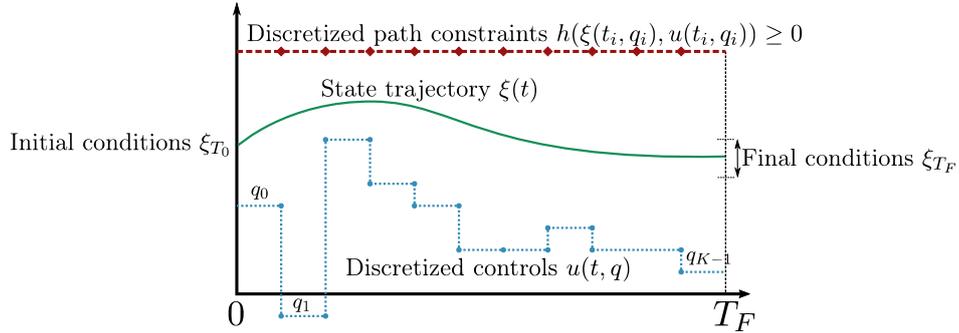


Figure 5.3 – Direct single-shooting, NLP variables

constraints are discretized and enforced at each one of the time grid nodes  $t_i$ , leading to the following NLP formulation:

$$\min_{q \in \mathbb{R}^{n_u}} \int_0^{T_F} L(\xi(t, q), u(t, q)) dt + E(\xi(T_F, q)) \quad (5.15)$$

s. t.

$$\begin{aligned} \xi(T_0) - \xi_{T_0} &= 0, & \text{Initial conditions} \\ h(\xi(t_i, q_i), u(t_i, q_i)) &\geq 0, & \text{Discretized path constraints} \\ \xi(T_F, q) - \xi_{T_F} &= 0, & \text{Final conditions} \end{aligned} \quad (5.16)$$

with  $i \in [0, K - 1]$ .

### Simultaneous approach

The second group of direct methods discretize the system states as well as the control inputs, hence the decision vector of the resulting NLP will consist of both the parametrized states and inputs, so it can be expected that the size of the resulting NLP problem is larger compared to the sequential approach. The direct simultaneous approach can in turn be divided into collocation and multiple shooting methods, which are outlined in the following.

- **COLLOCATION METHOD.** Direct collocation methods are based on the parameterization of the state and control trajectories using basis functions such as piecewise constant, polynomials or B-splines. They are both discretized over a relatively fine time grid, whose nodes  $t_i$  define the states  $s_i = \xi(t_i)$  and the control parameters,  $q_i$ , that yield the control function  $u_i(t, q_i)$ . For example, if a piecewise constant function is chosen as the control parameterization, we would obtain a constant control  $u(t) = q_i$  on each interval.

The core concept of the collocation method is to define a set of  $m$  collocation points on each subinterval of the grid  $t_i^{(1)}, \dots, t_i^{(m)}$ , also known as finite elements. There exists different strategies when defining the set of collocation points, where Legendre-Gauss (LG), Legendre-Gauss-Radau (LGR), and Legendre-Gauss-Lobatto(LGL) are the most popular ones [Pietz 2003].

Afterwards, the state trajectory can be approximated by a polynomial  $p_i(t, c_i) \in \mathbb{R}^{n_x}$ , with  $c_i \in \mathbb{R}^{n_x(m+1)}$  coefficients. Such polynomials integrate the trajectory over each interval  $[t_i, t_{i+1}]$ , by a suitable choice of the coefficients  $c_i$ , which is made by means of a set of algebraic equations that ensure that the computed polynomial is a realistic representation of the state trajectories, known as collocation conditions (5.17).

$$\begin{aligned}
 & s_i = p_i(t_i, c_i), && \text{Initial conditions} \\
 & \underbrace{f(p_i(t_i^{(1)}), c_i), u_i(t_i^{(1)}, q_i))}_{c_{i,1}} = \dot{p}_i(t_i^{(1)}, c_i), && \text{Model dynamics satisfaction} \\
 & \vdots \\
 & \underbrace{f(p_i(t_i^{(m)}), c_i), u_i(t_i^{(m)}, q_i))}_{c_{i,m}} = \dot{p}_i(t_i^{(m)}, c_i).
 \end{aligned} \tag{5.17}$$

These collocation conditions can be enforced as equality constraints of the generated NLP for each one of the collocation points, compacted as a vector equation  $g_i(s_i, c_i, q_i) = 0$ ,

$$g_i(s_i, c_i, q_i) = \begin{bmatrix} c_{i,0} - s_i \\ \dot{p}_i(t_i^{(1)}, c_i) - f(c_{i,1}, t_{i,1}) \\ \vdots \\ \dot{p}_i(t_i^{(m)}, c_i) - f(c_{i,m}, t_{i,m}) \end{bmatrix} = 0 \tag{5.18}$$

In addition, continuity between the different intervals of the time grid needs to be ensured by means of the following continuity conditions, enforced for each transition between the subintervals.

$$p(t_{i+1}, c_i) - s_{i+1} = 0 \text{ for } i = 0 \dots N \quad (5.19)$$

The ending step is to approximate the integrals  $\int_{i=0}^{i=K-1} L(\xi, u) dt$  by a quadrature formula, using the same collocation points, denoted as  $l_i(s_i, c_i, u_i)$  in the following.

Finally, the following NLP is formulated:

$$\min_{s, c, u} \sum_{i=0}^{K-1} l_i(s_i, c_i, u_i) + E(s_{T_F}) \quad (5.20)$$

s. t.

$$\begin{aligned} s_{T_0} - \xi_{T_0} &= 0, & \text{Initial conditions} \\ g_i(s_i, c_i, q_i) &= 0, & \text{Collocation conditions} \\ p(t_{i+1}, c_i) - s_{i+1} &= 0, & \text{Continuity conditions} \\ h(s_i, q_i) &\leq 0, & \text{Discretized path constraints} \\ s_{T_F} - \xi_{T_F} &\leq 0, & \text{Final conditions} \end{aligned} \quad (5.21)$$

with  $i \in [0, K - 1]$ .

- **MULTIPLE SHOOTING METHOD.** [Bock 1984], [Bock 2000]. Again, as part of the simultaneous approach group, both the state trajectory and the control input over the receding prediction horizon are discretized. The major difference with the collocation method and the sequential direct single-shooting method is that in the multiple shooting method, the system is separately integrated in each interval between the discretization (shooting) nodes, based on an initial guess of the initial state at each node  $s_i$ .

Accordingly, the prediction horizon  $T_F$  is divided in a time grid with  $K$  subintervals, and the input control signals are discretized on a piecewise constant grid,  $u(t) = q_i$ , for  $t \in [t_i, t_{i+1}]$  (5.14). After that, the ODE is solved independently at each one of the generated intervals, starting from a virtual initial value  $s_i$ , assigned to each grid node.

$$\begin{aligned} \dot{\xi} &= f(\xi(t), s_i), t \in [t_i, t_{i+1}] \\ \xi(t_i) &= s_i \end{aligned} \quad (5.22)$$

Once this is settled, a finite number of Initial Value Problems (IVP) are defined, so that the state trajectory pieces  $\xi_i(t_i, s_i, q_i)$  can be obtained by means of an integration approach (Section 5.1.4), providing the parametrized trajectory pieces  $l_i(s_i, q_i)$  over each subinterval.

$$l_i(s_i, q_i) = \int_{t_i}^{t_{i+1}} L(\xi(t_i, s_i, q_i)) dt \quad (5.23)$$

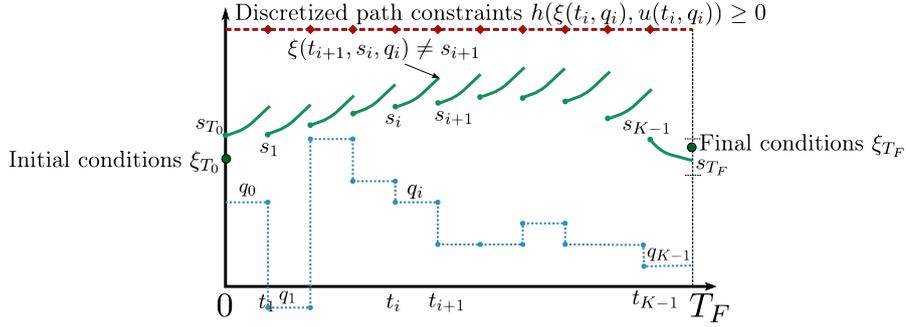


Figure 5.4 – Multiple-shooting, NLP variables

In order to avoid discontinuity on the trajectories between the final and initial state of the successive sub-intervals of the grid, continuity constraints between them need to be enforced explicitly. This is done by imposing  $s_{i+1} = \xi(t_{i+1}, s_i, q_i)$ .

All considered, the resulting NLP problem can be written as follows:

$$\min_{s, q} \sum_{i=0}^{K-1} l_i(s_i, q_i) + E(s_{T_F}) \quad (5.24)$$

s. t.

$$\begin{aligned} s_{T_0} - \xi_{T_0} &= 0, & \text{Initial conditions} \\ s_{i+1} - \xi(t_{i+1}, s_i, q_i) &= 0, & \text{Continuity conditions} \\ h(\xi(t_i, q_i), u(t_i, q_i)) &\geq 0, & \text{Discretized path constraints} \\ s(T_F) - \xi_{T_F} &= 0, & \text{Final conditions} \end{aligned} \quad (5.25)$$

with  $i \in [0, K]$ .

**Remark 5.3** *When dealing with real-time applications, a possible enhancement of the multiple shooting algorithm would be to exploit its nature and be implemented in a parallel computation fashion [Betts 1991]. In this kind of implementation, each segment could be implemented on an individual processor.*

□

#### 5.1.4 Numerical Integration - Runge-Kutta scheme

Previous section shows that shooting methods require the propagation of the system dynamics, so the numerical solution of the formulated IVP (Definition 5.1) is fundamental for the resolution of the OCPs via direct methods (Section 5.1.3.3). Numerical methods for solving the ODE IVP are a relative mature field in optimal control [Betts 2010], [Quirynen 2017].

The main objective of any numerical integration method is to take a step forward over a certain interval of time,  $t \in [t_i, t_{i+1}]$ , in order to find an approximate value for the point  $\tilde{\xi}_{i+1}$ , based on the previously computed solution values and the system ODE, written in the form (5.1). Formally, the integration of the system dynamics yields

$$\xi_{i+1} = \xi_i + \int_{t_i}^{t_{i+1}} \dot{\xi}(t) dt = \xi_i + \int_{t_i}^{t_{i+1}} f(\xi(t), u(t), t) dt \quad (5.26)$$

Traditionally, numerical integration methods are divided into one-step and multi-step approaches. Briefly speaking, the main difference between both strategies is that one-step methods use only the previous solution point, while multi-step approach consider a certain amount of previous points. In the following, attention is given to one-step Runge-Kutta (RK) methods, which are the most widely used. The basic idea of the RK methods is to perform several evaluations of the function at intermediate stage points, providing a higher accuracy on the resulting solution. In order to evaluate the integral (5.26), the time interval is divided into  $k$  subintervals,  $t_j = t_i + h_i \kappa_j$ , with  $0 \leq \kappa_1 \leq \kappa_2 \leq \dots \leq \kappa_k \leq 1$ , for  $1 \leq j \leq k$ . After defining the subdivided interval, a quadrature formula within a quadrature formula is applied:

$$\int_{t_i}^{t_{i+1}} f(\xi(t), u(t), t) dt \approx h_i \sum_{j=1}^k \beta_j \tilde{f}_j \quad (5.27)$$

where  $\tilde{f}_j \equiv f(t_j, \tilde{\xi}_j)$ , and  $\tilde{\xi}_j$  is the approximation of the system variables at the intermediate points of the subgrid, computed by means of the following expression:

$$\int_{t_i}^{t_j} f(\xi(t), u(t), t) dt \approx h_i \sum_{l=1}^k \alpha_{jl} f_l \quad (5.28)$$

with  $1 \leq j \leq k$ . Condensing the results from (5.26) - (5.28), the general formulation for the one-step  $k$ -stage Runge-Kutta scheme is obtained:

$$\xi_{i+1} = \xi_i + h_i \sum_{j=1}^k \beta_j f_{i_j} \quad (5.29)$$

with

$$f_{i_j} = f \left( \underbrace{\left[ h_i \sum_{l=1}^k \alpha_{jl} f_{i_l} \right]}_{\tilde{\xi}_j}, \underbrace{[t_i + h_i \kappa_j]}_{t_j} \right) \quad (5.30)$$

where  $\kappa_k, \beta_j, \alpha_{j_l}$  are known constants, often represented in a compact by means of the Butcher's array [Butcher 2016].

$$\begin{array}{c|ccc} \kappa_1 & \alpha_{11} & \dots & \alpha_{1k} \\ \vdots & \vdots & & \vdots \\ \kappa_k & \alpha_{k1} & \dots & \alpha_{kk} \\ \hline & \beta_1 & \dots & \beta_k \end{array} \quad (5.31)$$

A further classification of these methods can be made based on the  $\alpha_{ji}$  coefficients, denoted as explicit if  $\alpha_{ji} = 0$  for  $l \geq j$  and implicit otherwise.

To conclude this section, two common examples of the explicit k-stage Runge-Kutta schemes are introduced below.

### Explicit Euler integration method

The simplest explicit RK method is the forward Euler integration of order 1, with  $k = 1$ , whose Butcher array and common representation are depicted in (5.32), (5.33)

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \quad (5.32)$$

$$\tilde{\xi}_{j+1} = \xi_j + h_j f(x_j, t_j) \quad (5.33)$$

### Classical Runge-Kutta method

One of the most widespread integration methods is the Runge-Kutta Method of order four, usually denoted by RK4, whose Butcher array is defined by the coefficients.

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} \quad (5.34)$$

Considering the constant control input  $u_{const}$ , one step of the RK4 method proceeds as follows:

$$\begin{aligned} \kappa_1 &= f(\tilde{\xi}_i, u_{const}) \\ \kappa_2 &= f\left(\tilde{\xi}_i + \frac{h}{2}\kappa_1, u_{const}\right) \\ \kappa_3 &= f\left(\tilde{\xi}_i + \frac{h}{2}\kappa_2, u_{const}\right) \\ \kappa_4 &= f(\tilde{\xi}_i + h\kappa_3, u_{const}) \\ \tilde{\xi}_{i+1} &= \tilde{\xi}_i + \frac{h}{6}(\kappa_1 + 2\kappa_2 + 2\kappa_3 + \kappa_4) \end{aligned} \quad (5.35)$$

## 5.2 Flatness

Differential flatness (or flatness) was originally studied by [Fliess 1992]. Generally speaking, it is a property of some controlled dynamical systems which allows a complete parameterization of all system variables (states, inputs and outputs) in terms of a finite set of independent variables, called the *flat outputs*, and a finite number of its time derivatives. The flat outputs are internal variables to the system, so they are a function of the states and of a finite number of derivatives of the inputs. By specifying the desired flat outputs trajectories, the nominal state

and input trajectories are completely defined without integrating the system ODE. Nevertheless, establishing which are the flat outputs of a dynamical system is generally hard, as there is no systematic method to determine them, except on the case of linear systems and affine nonlinear single input cases [Sira-Ramirez 2004], [Fliess 1995].

**Definition 5.6** [Murray 2009]. A controlled nonlinear system expressed as in 5.1 is differentially flat if there exists a function  $f$  such that

$$z = f(\xi, u, \dot{u}, \dots, u^{(p)}) \quad (5.36)$$

and we can write the solutions of such nonlinear system as functions of  $z$ , denoted as the system's *flat output* and a finite number  $q$  of its derivatives

$$\begin{aligned} \xi &= f(z, \dot{z}, \dots, z^{(q)}), \\ u &= f(z, \dot{z}, \dots, z^{(q)}) \end{aligned} \quad (5.37)$$

□

**Remark 5.4** *The number of flat outputs  $n_z$  is equal to the number of system inputs.*

□

**Example 5.1** Let us consider a simple system, a nonholonomic integrator taken from [Brockett 1982]:

$$\dot{\xi}_1 = u_1, \quad \dot{\xi}_2 = u_3, \quad \dot{\xi}_3 = \xi_2 u_1 \quad (5.38)$$

This systems presents the differential flatness property, with  $z = (\xi_1, \xi_3)$ . This means that once the trajectories of the flat variables are known, the states trajectories can be obtained straightforward:

$$\xi_1 = z_1, \quad \xi_2 = \dot{\xi}_3 / \dot{\xi}_1 = \dot{z}_2 / \dot{z}_1, \quad \xi_3 = z_2 \quad (5.39)$$

◇

## 5.3 Hyperplane Arrangements

### 5.3.1 Basic definitions

In the following, a series of relevant definitions are recalled, following the nomenclature introduced in [Ziegler 2012], where the reader is addressed for further concepts study.

**Definition 5.7** An hyperplane arrangement  $\mathcal{A}(\mathbb{H})$  is a collection of regions generated by  $N$  hyperplanes  $\mathbb{H} = \{\mathcal{H}_i\}_{i=1:N}$  that partition the whole finite dimensional space in which they are defined.

□

**Definition 5.8** A cell or region  $\mathcal{A}(\boldsymbol{\sigma})$  represents a disjoint partition of the hyperplane arrangement  $\mathcal{A}(\mathbb{H})$  characterized by a sign tuple  $\boldsymbol{\sigma} \in \{+, -\}^N$ , defined as:

$$\mathcal{A}(\boldsymbol{\sigma}) = \bigcap_{i=1}^N \mathcal{H}_i^{\sigma^{(i)}} \quad (5.40)$$

□

**Remark 5.5** [Buck 1943]. Given an hyperplane arrangement  $\mathcal{A}(\mathbb{H})$ , the number of feasible regions  $\gamma(N)$  in relation with the space dimension  $n_x$ , and the number of hyperplanes  $N$ , is bounded by the Buck's formula:

$$\gamma(N) \leq \sum_{i=0}^{n_x} \binom{N}{i} \quad (5.41)$$

□

In this way, the whole state-space can be defined by an hyperplane arrangement  $\mathcal{A}(\mathbb{H})$  as a union of all the feasible disjoint cells  $\mathcal{A}(\boldsymbol{\sigma}_j)$ ,

$$\mathcal{A}(\mathbb{H}) = \bigcup_{j=1, \dots, \gamma(N)} \mathcal{A}(\boldsymbol{\sigma}_j) = \bigcup_{j=1, \dots, \gamma(N)} \left( \bigcap_{i=1}^N \mathcal{H}_i^{\sigma_j^{(i)}} \right) \quad (5.42)$$

where

$$\Sigma = \{\boldsymbol{\sigma}_j \in \{+, -\}^N : \mathcal{A}(\boldsymbol{\sigma}_j) \neq \emptyset\} \quad (5.43)$$

defines the collection of feasible tuples that provide combinations of the half-spaces (2.8) that yield non-empty intersections of regions and  $\gamma(N)$  is the number of cells, bounded by the Buck's formula (Remark 5.5). Efficient algorithms for mode enumeration can be found in [Avis 1996], [Geyer 2010].

**Example 5.2** Consider the hyperplane arrangement depicted in Figure 5.5(a), described by the hyperplanes  $\mathcal{H}_i$ , with  $i \in 1, \dots, 4$ , that is,  $N = 4$ . These divide the full space in 9 feasible cells, each one defined by a unique sign tuple  $\boldsymbol{\sigma}$ . The feasible combinations  $\Sigma$  (5.43) are  $(++--)$ ,  $(-+--)$ ,  $(---)$ ,  $(---+)$ ,  $(-+-+)$ ,  $(++-+)$ ,  $(++++)$ ,  $(-+++)$ ,  $(-+++)$ . In addition, the shadowed region corresponds to a cell  $\mathcal{A}(\boldsymbol{\sigma}) = \bigcap_{i=1}^{N=4} \mathcal{H}_i^{\sigma^{(i)}} = \mathcal{H}_1^- \cap \mathcal{H}_2^+ \cap \mathcal{H}_3^- \cap \mathcal{H}_4^+$ , given by the tuple  $\boldsymbol{\sigma} = (-+-+)$ . It can be seen that the remaining tuple combinations ( $2^4 - 9 = 7$ ) are unfeasible, for example  $\mathcal{A}(\boldsymbol{\sigma}) = \bigcap_{i=1}^{N=4} \mathcal{H}_i^{\sigma^{(i)}} = \mathcal{H}_1^- \cap \mathcal{H}_2^+ \cap \mathcal{H}_3^+ \cap \mathcal{H}_4^- = \emptyset$ .

Figure 5.5(b) shows a perturbed version of the original hyperplane arrangement. It can be seen that the arrangement in Figure 5.5(a) is not in general position, hence the bound given by Buck's formula is not reached (Remark 5.5). Moreover, the translation of the hyperplane generates new cells on the arrangement, that now has 10 regions.  $\diamond$

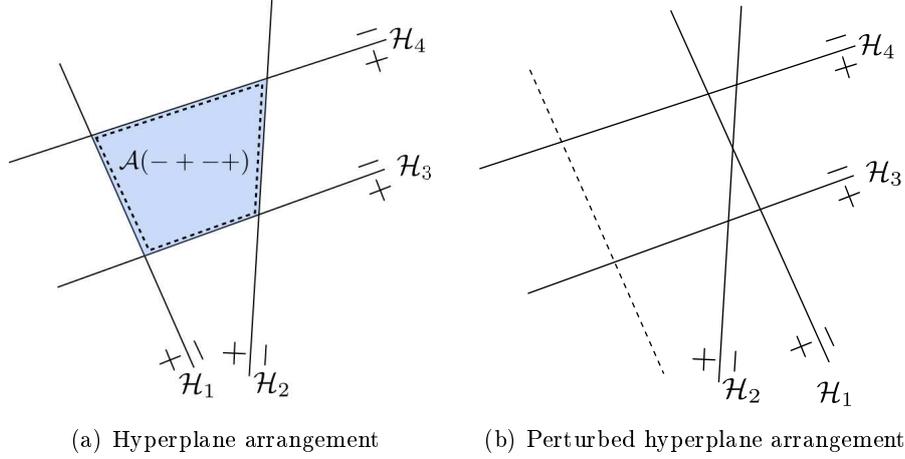


Figure 5.5 – Hyperplane Arrangements

### 5.3.2 Non-convex non-connected regions characterization

In practical applications, it may be of interest to divide the space into *feasible* ( $\mathbb{F}$ ) and *unfeasible* ( $\mathbb{T}$ ) regions. Typically, these regions are not necessarily convex or connected, still, they can be expressed as the union of convex cells ( $\mathcal{F}_l$  or  $\mathcal{T}_t$  respectively), covering the corresponding sub-space:

$$\mathbb{F} = \mathcal{C}(\mathbb{T}) = \bigcup_{l=1, \dots, N_f} \mathcal{F}_l \quad (5.44)$$

with  $\mathcal{F}_l = \{\mathcal{A}(\sigma_f) \subset \mathcal{A}(\mathbb{H}) : \sigma_f \in \Sigma_f\}$ , and  $\Sigma_f \subset \Sigma$ ,

$$\Sigma_f = \{\sigma \in \Sigma : \mathcal{A}(\sigma_f) \cap \mathbb{T} = \emptyset\} \quad (5.45)$$

representing the collection of feasible tuples. In a similar way, the forbidden region can be expressed as the union of the  $N_t$  unfeasible cells,

$$\mathbb{T} = \mathcal{C}(\mathbb{F}) = \bigcup_{t=1, \dots, N_t} \mathcal{T}_t \quad (5.46)$$

with  $\mathcal{T}_t \in \{\mathcal{A}(\sigma_t) \in \mathcal{A}(\mathbb{H}) : \sigma_t \in \Sigma_t\}$ , and  $\Sigma_t \subset \Sigma$ ,

$$\Sigma_t = \{\sigma \in \Sigma : \mathcal{A}(\sigma_t) \cap \mathbb{F} = \emptyset\} \quad (5.47)$$

representing the collection of unfeasible tuples.

**Remark 5.6** Note that  $\Sigma_f \cap \Sigma_t = \emptyset$  and that  $\Sigma_f \cup \Sigma_t = \Sigma$ , according to the definition in (5.43).  $\square$

In these lines, there are two ways of proceeding. The first option would be to pre-define an hyperplane arrangement and afterwards define the existing forbidden cells, while in a second approach the forbidden regions can be represented by polyhedra whose defining hyperplanes generate an hyperplane arrangement.



$--++)(---++)(---+++)(---++-)(-+++-)(-++++-)(+--+ + + -)(+ + + + -)(+ + + - + -)(+ + + - + +)(- - + + + +)(+ - + - + +)\}.$

**Remark 5.8** *As a practical note, it can be noted that neighboring cells only have one different sign on their defining tuple.*

It can be seen that both approaches have their pros and cons, in the first one, the representation is simpler, but the obstacles are highly over-approximated. On the other hand, the second approach is more precise, at the expense of a higher complexity.  $\diamond$

### 5.3.3 Cell Merging

The number of convex cells describing a non-convex non-connected region can be quite important, and the analysis within cell merging techniques [Prodan 2012], [Prodan 2015], [Geyer 2004] that allow to obtain a simplified representation is of interest in most of the applications.

**Definition 5.9** The union of cells  $\mathcal{A}(\sigma)$  that preserve the same sign value over a subset of indices  $i \subset \mathcal{I}$  of their sign tuple  $\sigma$  and present all the possible combinations for the rest of indices  $k \subset \mathcal{I}$  is a merged cell  $\mathcal{A}(\sigma^*)$ , defined as follows,

$$\mathcal{A}(\sigma^*) = \bigcup_{\sigma^*} \mathcal{A}(\sigma) = \bigcap_{\sigma^*(i) \neq *', i \in \mathcal{I}} \mathcal{H}_i^{\sigma^*(i)} \quad (5.48)$$

with  $\sigma^* \in \{+, *, -\}^N$  being the sign tuple of the merged cell, where  $\sigma^*$  is defined as  $\sigma^*(i) = \sigma(i)$  and  $\sigma^*(k) = *'$ , with  $i$  and  $k$  described above.  $\square$

There exist several algorithms that allow to compute merged cells. A first trend is based in a "branch and bound" strategy proposed in [Geyer 2004]. A second approach is based on the use of Boolean algebra, where the merging problem is translated into the optimization of a Boolean function given in the Sum of Products (SOP) form (for example, via Karnaugh maps. In this approach, merged cells will be described by the Boolean miniterms of the optimized function. Such a Boolean function can be defined from the collection of the hyperplane arrangement tuples, by means of the following Theorem.

**Theorem 5.1** [Prodan 2015]. *Consider the forbidden and admissible regions (5.44), (5.46) characterized respectively by the sign tuples (5.45), (5.47). Then, the feasible region is compactly described as a union of merged cells (5.48):*

$$\mathbb{F} = \bigcup_{\sigma^*} \mathcal{A}(\sigma^*), \quad (5.49)$$

where  $\sigma^* \in \{-, *, +\}^N$  are given by the sum of products representation of the Boolean function  $f : \{-, +\}^N \rightarrow \{0, *, 1\}$  verifying:

$$\begin{aligned} f(\sigma) &= 0, \quad \forall \sigma \in \Sigma_f \\ f(\sigma) &= 1, \quad \forall \sigma \in \Sigma_t \\ f(\sigma) &= *, \quad \forall \sigma \in \{-, +\}^N \setminus \Sigma \end{aligned} \quad (5.50)$$

□

*Proof:* Using this formulation, we obtain a truth table for the function in (5.50), where the value '0' is assigned to a forbidden cell, and '1' represents a feasible one. Moreover the symbol '\*' is assigned to the rest of cells, that is, the ones that are related to an empty tuple combination. If we simplify the obtained Boolean function and express it on its canonical SOP, each one of the terms will describe a region of the form (5.48), which is equivalent to (5.49). ■

In [Prodan 2015] the techniques to build up these Boolean functions maps are revisited in detail.

**Example 5.4** We continue with the hyperplane arrangement depicted in Figure 5.6(b). For the small number of hyperplanes appearing on this example, graphical merging cell techniques based on Boolean algebra and Karnaugh maps are an attractive approach. First of all, the Boolean function has been obtained by means of Theorem 5.1. Then, Figure 5.7 shows the corresponding Karnaugh map and the constructed groups, obtaining the reduced canonical SOP form of the original function (5.51).

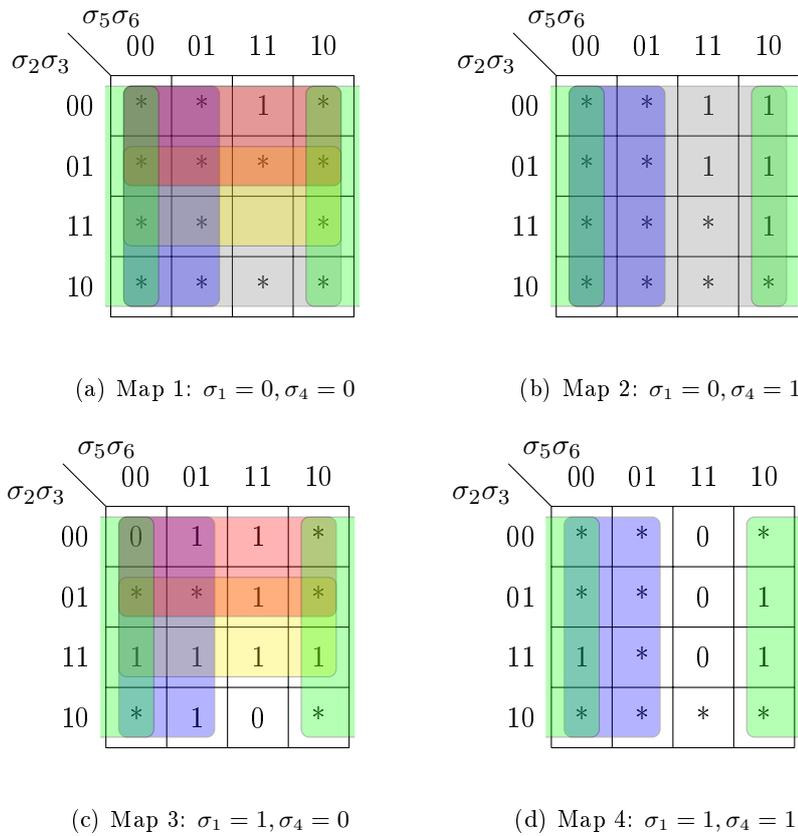


Figure 5.7 – Karnaugh map representation for hyperplane arrangement in Fig. 5.6(b)

$$f(\boldsymbol{\sigma}) = \sigma(\bar{1}) + \sigma(\bar{5}) + \sigma(\bar{6}) + \sigma(\bar{2})\sigma(\bar{4}) + \sigma(3)\sigma(\bar{4}), \quad (5.51)$$

so the merged cells  $\mathcal{A}(\boldsymbol{\sigma}^*)$  (5.48), shown in (Figure 5.8), are defined by the following collection of tuples  $\boldsymbol{\sigma}^* \in \Sigma^* = \{(-****), (****-), (*****), (*--**), (**+-**)\}$ .

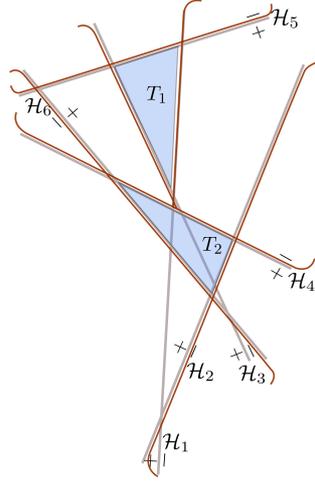


Figure 5.8 – Merged Cells

◇

### 5.3.4 Mixed Integer representation

In the constrained optimal control framework, it may be of interest to enforce the inclusion of a certain state  $\xi$  on a given non-convex and/or non-connected feasible region. If an hyperplane arrangement  $\mathcal{A}(\mathbb{H})$  (5.42) is defined on the state-space, such non-convex region can be expressed as the union of  $N_f$  convex disjoint cells which are defined by a collection of  $N$  hyperplane inequalities (2.8), that is,  $\xi \in \mathbb{F}$  (5.44).

Mixed Integer Programming (MIP) framework provides a set of tools that allow to express such containment condition in an extended space  $\mathbb{X} \times \{0, 1\}^N$  defined by the state space and a set of auxiliary binary variables,  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N) \in \{0, 1\}^N$ :

$$\begin{aligned} -h_i^T \xi &\leq -g_i + M\alpha(1) \\ &\vdots \\ -h_i^T \xi &\leq -g_i + M\alpha(N) \\ \sum_{i=1}^N \alpha(i) &\leq N - 1 \end{aligned} \quad (5.52)$$

with  $i = 1 \dots N$  and  $M$  being a scalar relatively bigger than the variables at the right hand side of the inequalities. The use of this scalar value on the formulation, together

with the binary variables, allows to manage the collection of constraints, relaxing the inequality when the corresponding hyperplane is not active for a given cell. This approach is denoted in the MIP framework as the *big M* strategy [Vielma 2008].

**Remark 5.9** [Prodan 2015]. *For a finite selection of a sufficiently large  $M$ , the following Linear Programming (LP) problem can be solved, which allows to obtain a bounded  $M$  that makes the inequalities redundant with respect to the state space  $\mathbb{X}$ .*

$$M = \max_{i, M_i \geq 0} \left( \max_{\xi \in \mathbb{X}} \{M_i = g_i - h_i \xi\} \right) \quad (5.53)$$

□

**Example 5.5** Let us continue with the analysis of the hyperplane arrangement from Figure 5.6(b), where any cell of the the feasible space can be expressed in the augmented space as:

$$\begin{aligned} -h_1 \xi &\leq -g_1 + M\alpha(1) \\ -h_2 \xi &\leq -g_2 + M\alpha(2) \\ -h_3 \xi &\leq -g_3 + M\alpha(3) \\ -h_4 \xi &\leq -g_4 + M\alpha(4) \\ -h_5 \xi &\leq -g_5 + M\alpha(5) \\ -h_6 \xi &\leq -g_6 + M\alpha(6) \\ \sum_{i=1}^6 \alpha_i(i) &\leq 5 \end{aligned} \quad (5.54)$$

For example, for  $(\alpha_1, \dots, \alpha_N) = (1, 0, 0, 0, 1, 1)$ , we would obtain the region  $\mathcal{F}_6$ , by activating the corresponding  $\mathcal{H}_{\{2,3,4\}}^-$  inequalities, while the rest of them would be made redundant by means of the scalar value  $M$ . ◇

**Remark 5.10** *It must be noted that the last condition that is enforced on the formulation (5.52) prevents the solution to degenerate (with  $\alpha = \mathbf{0}$ ), and ensures that at least one constraint is active.* □

### 5.3.4.1 MIP formulation complexity reduction

We have shown that the representation of the containment into a non-convex non-connected region can be divided into a collection of convex sub-problems, by means of a augmented state-binary space and mixed-integer programming formulation. Nevertheless, it is straightforward to notice that the number of hyperplanes defining the arrangement,  $N$ , increases the number of binary variables that is needed in the formulation and as a by-product, directly affects the computational burden of the generated MIP problem.

In the subsequent, the techniques that allow to reduce the number of binary variables are revisited. We start by introducing the *logarithmic representation* of

the binary part, where one binary tuple  $\beta^l$  assigned to each feasible convex cell  $\mathcal{F}_l$  is enough to distinguish between the cells of the arrangement. Following up with this enumeration logic, the least number of regions, the least number of binary variables, so *cell merging* techniques (Section 5.3.3) are also a powerful tools to lighten the computational burden.

Based on a binary representation, each one of the cells of the hyperplane arrangement can be coded in terms of  $N_0$  bits, assigning a unique number to each one of them, starting from zero and taking the numbers successively:

$$N_0 = \lceil \log_2(N_f) \rceil \quad (5.55)$$

Let us recall Proposition 5.1 from [Stoican 2011b].

**Proposition 5.1** [Stoican 2011b]. *A mapping  $\alpha_l(\beta) : \{0, 1\}^{N_0} \rightarrow \mathbb{R}$  which verifies that  $\alpha_l(\beta^l) = 0$  and  $\alpha_l(\beta^j) \geq 0$  for any  $j \neq l$  is given by:*

$$\alpha_l(\beta) = \sum_{k=1}^{N_0} p_k^l, \quad \text{where } p_k^l = \begin{cases} \beta_k, & \text{if } \beta_k^l = 0 \\ 1 - \beta_k, & \text{if } \beta_k^l = 1 \end{cases} \quad (5.56)$$

where  $\beta_k$  denotes the  $k^{\text{th}}$  variable and  $\beta_k^l$  its value for the tuple associated to the region  $\mathcal{F}_l$  (5.44), with  $l = 1 \dots N_f$ .  $\square$

This kind of mapping keeps the linear nature of the expression appearing on the inequalities (5.52), while reducing the number of necessary binary variables to enumerate each cell. Nevertheless, it has to be taken into account that there may be some tuples that provide an empty region, that is, their representation in the extended space is degenerated. To avoid this situation, the following logic is included in the formulation, in order to enforce the unfeasibility of such tuple realizations:

**Corollary 5.1** [Stoican 2011b]. *Let there be a tuple  $\beta^l \in \{0, 1\}^{N_0}$ . The point it describes is made unfeasible with respect to the constraint:*

$$-\sum_{k=1}^{N_0} p_k^l \leq -\epsilon \quad (5.57)$$

with  $p_k^l$  defined as in Proposition 5.1 and  $\epsilon \in \{0, 1\}$  being a scalar value that makes the corresponding tuple inequalities to vanish.  $\square$

In this way, any variable  $\alpha$  is written as a linear combination in the space of variables  $\beta = (\beta_1, \dots, \beta_{N_0})$ , following Proposition 5.1 for the feasible tuples (5.43) and Corollary 5.1 for the non-allocated ones, which will be individually assigned to each cell.

The full non-convex non-connected feasible region  $\mathbb{F}$  (5.44) is then reformulated and defined by the following set of inequalities in the extended space:

$$\mathcal{F}_l \begin{cases} \sigma_l(1)h_1\xi \leq \sigma_l(1)g_1 \\ \vdots \\ \sigma_l(N)h_N\xi \leq \sigma_l(N)g_N \end{cases} + M\beta^l \quad (5.58)$$

$$\vdots$$

for  $l = 1 \dots N_l$  covering the number of cells on the feasible space with their corresponding sign tuples  $\sigma_l \in \Sigma_f$  (5.45) and the tuple of assigned binary variables  $\beta^l$ .

**Example 5.6** Let us continue with the analysis of the hyperplane arrangement from Figure 5.6(b) with  $N_l = 17$ . By considering the binary representation, it is enough to use  $N_0 = \lceil \log_2(17) \rceil = 5$  binary variables to distinguish between the cells of the arrangement. Using the list of feasible cells, together with the variable mapping stated in Proposition 5.1, we can assign a unique number for each cell, coded in the corresponding tuples  $\beta^l$  (Figure 5.9), obtaining the following mixed-integer representation of the feasible region is as follows:

$$\begin{aligned}
 \mathcal{F}_1 & \left\{ \begin{array}{l} h_1\xi \leq g_1 \\ h_2\xi \leq g_2 \\ h_3\xi \leq g_3 \\ h_4\xi \leq g_4 \\ -h_5\xi \leq -g_5 \\ -h_6\xi \leq -g_6 \\ \vdots \end{array} \right. + M(\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5) \\
 \mathcal{F}_2 & \left\{ \begin{array}{l} h_1\xi \leq g_1 \\ h_2\xi \leq g_2 \\ h_3\xi \leq g_3 \\ -h_4\xi \leq -g_4 \\ -h_5\xi \leq -g_5 \\ -h_6\xi \leq -g_6 \\ \vdots \end{array} \right. + M(1 + \beta_1 + \beta_2 + \beta_3 + \beta_4 - \beta_5) \\
 \mathcal{F}_3 & \left\{ \begin{array}{l} h_1\xi \leq g_1 \\ h_2\xi \leq g_2 \\ h_3\xi \leq g_3 \\ -h_4\xi \leq -g_4 \\ -h_5\xi \leq -g_5 \\ h_6\xi \leq g_6 \\ \vdots \end{array} \right. + M(1 + \beta_1 + \beta_2 + \beta_3 - \beta_4 + \beta_5) \\
 & \vdots
 \end{aligned} \tag{5.59}$$

Since only 17 tuples of the 32 in total are related to a feasible cell, we need to add the remaining inequalities to ensure that the following unallocated tuples remain unfeasible:

$$\begin{aligned}
 -(2 - \beta_1 + \beta_2 + \beta_3 + \beta_4 - \beta_5) & \leq 0.5 \\
 -(2 - \beta_1 + \beta_2 + \beta_3 - \beta_4 + \beta_5) & \leq 0.5 \\
 -(3 - \beta_1 + \beta_2 + \beta_3 - \beta_4 - \beta_5) & \leq 0.5 \\
 \dots \text{remaining unallocated tuples} &
 \end{aligned} \tag{5.60}$$

◇

A second simplification that can be considered comes in a straightforward manner from the tools introduced in Section 5.3.3, where a reduced number of regions

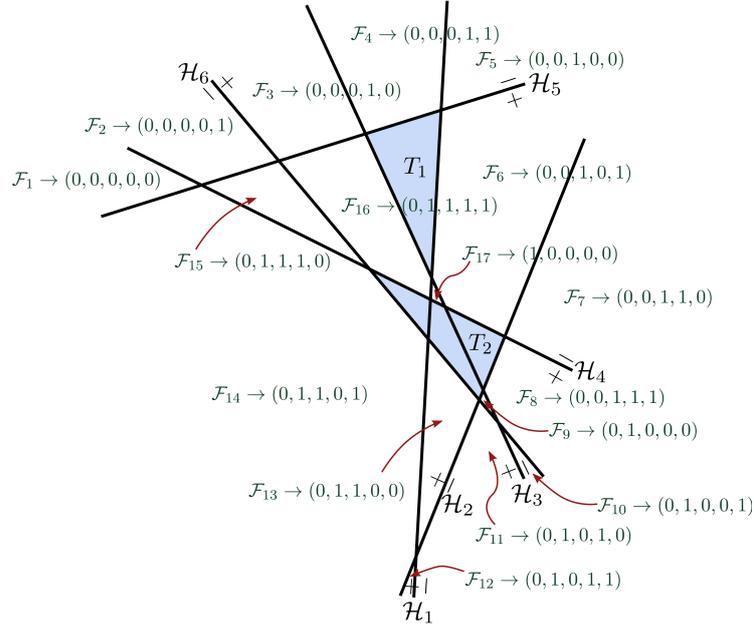


Figure 5.9 – Feasible cells binary enumeration

implies the need of less binary variables  $\beta$ . Moreover, the representation is made more compact by means of the following Remark 5.11.

**Remark 5.11** *There is no need to describe each one of the individual cells in (5.58) by the full collection of  $N$  hyperplanes defining the arrangement. Instead, the number of inequalities can be cut down to just the  $N_{I_a}$  active ones of each feasible region  $F_i$ , eliminating the redundant hyperplane inequalities from the description.*  $\square$

**Example 5.7** We continue with the hyperplane arrangement from Figure 5.6(b). Once a cell merging technique has been applied (Example 5.4, Section 5.3.3),  $N_0 = \lceil \log_2(5) \rceil = 3$  binary variables suffice to define the convex feasible region (Figure 5.10).

$$\begin{aligned}
 \mathcal{M}_1 & \left\{ \begin{array}{l} -h_1\xi \leq -g_1 \\ +M(\beta_1 + \beta_2 + \beta_3) \end{array} \right. \\
 \mathcal{M}_2 & \left\{ \begin{array}{l} -h_5\xi \leq -g_5 \\ +M(1 + \beta_1 + \beta_2 - \beta_3) \end{array} \right. \\
 \mathcal{M}_3 & \left\{ \begin{array}{l} -h_6\xi \leq -g_6 \\ +M(1 + \beta_1 - \beta_2 + \beta_3) \end{array} \right. \\
 \mathcal{M}_4 & \left\{ \begin{array}{l} -h_2\xi \leq -g_2 \\ -h_4\xi \leq -g_4 \\ +M(2 + \beta_1 - \beta_2 - \beta_3) \end{array} \right. \\
 \mathcal{M}_5 & \left\{ \begin{array}{l} h_3\xi \leq g_3 \\ -h_4\xi \leq -g_4 \\ +M(1 - \beta_1 + \beta_2 + \beta_3) \end{array} \right.
 \end{aligned} \tag{5.61}$$

Similarly to Example 5.6, the remaining unallocated tuples are made unfeasible by

means of the following inequalities:

$$\begin{aligned}
 -(2 - \beta_1 + \beta_2 + \beta_3) &\leq 0.5 \\
 -(2 - \beta_1 - \beta_2 + \beta_3) &\leq 0.5 \\
 -(3 - \beta_1 - \beta_2 - \beta_3) &\leq 0.5
 \end{aligned}
 \tag{5.62}$$

◇

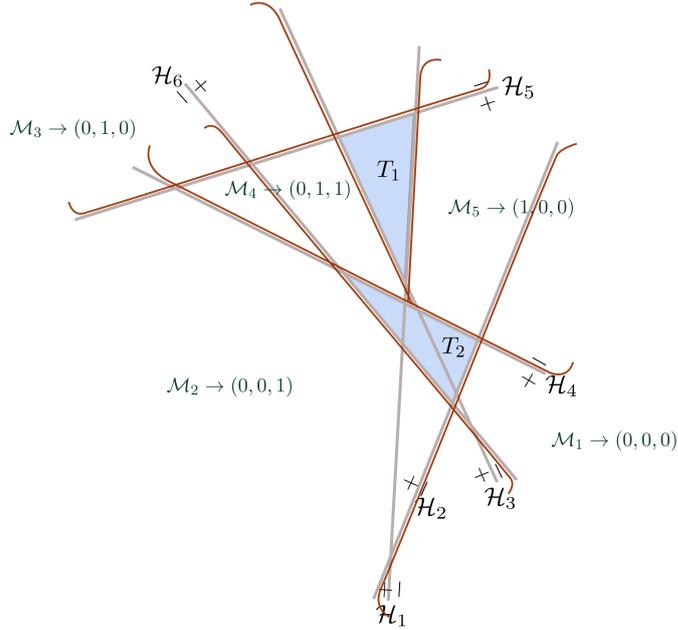


Figure 5.10 – Merged Cells

## 5.4 Conclusion

This chapter has brought into the picture the main tools that will be used in the following chapter for an optimization-based trajectory planning with anti-collision enhancements algorithm. The present thesis is the first work which aims to bring together the optimal control approach with a hyperplane arrangement modeling from the collision avoidance. This methodology is one of the contribution of the thesis and will be illustrated in the automotive application next.

Several state-of-art optimization techniques for the resolution of Optimal Control Problems have been reviewed. Firstly, Dynamic Programming has been introduced. In this methodology, most of the computational burden is made offline, generating a series of optimal control inputs depending on the current state, providing the solution that represents the global minimum. Nevertheless, such kind of technique requires afterwards to store this information in a tabular form, that shall be extracted afterwards on the online execution. This represents the major drawback of this method denoted by Bellman as *the curse of dimensionality*, which confines its

---

practical application to systems without a high number of states. In addition, as the optimal input computation is made offline, system constraints must be fixed a priori, limiting this technique to static environments. This makes this method inapplicable for the trajectory planning case, where we are dealing with a dynamic environment, and anti-collision constraints may never be the same: we cannot predict beforehand how the other vehicles may behave. This flexibility limitation appears as well when considering Indirect Methods, that search for an analytical solution of the optimal control by means of the resolution of the optimality conditions. In addition, the need of knowing when the constraints are activated adds an extra degree of rigidity. In other words, this technique allows to obtain the knowledge of the *form* of the optimal control and a statement of the two-point boundary-value problem, which, when solved, yields an explicit relationship for the optimal control, but is complex to apply in the presence of system constraints. Due to all these limitations of the previous methods, a direct optimization method will be considered in the following chapter, due to the capability of these kind of methods to deal with inequality constraints arising from path limitations in a systematic way without suffering of dimensionality problems.

As a complement, differential flatness concept has been introduced and it will be exploited to reduce the complexity of the model used at the generation stage. To finish with, Hyperplane Arrangements theory has been described together with a series of illustrative examples that allow to understand this not so well known technique, used in the next chapter to define the feasible free space for the trajectory generation algorithm.



# Collision-free trajectory planning on highways

---

## Contents

---

<b>6.1</b>	<b>Problem formulation</b>	<b>126</b>
6.1.1	Trajectory planning strategies review	127
6.1.2	Alternative obstacle avoidance formulations	128
<b>6.2</b>	<b>Exhaustive scenarii description</b>	<b>129</b>
<b>6.3</b>	<b>Methodology</b>	<b>133</b>
6.3.1	Ego vehicle description	133
6.3.2	Mathematical modeling of the highway scenarii: feasible non-convex region characterization	134
6.3.3	Optimal Control Problem formulation for collision-free trajectory planning	140
6.3.3.1	Objective function	141
6.3.3.2	Problem constraints	141
6.3.3.3	Complete OCP formulation	142
6.3.3.4	Illustrative simulation	145
6.3.4	Trajectory reference transformation step	146
6.3.4.1	Numerical simulation	148
<b>6.4</b>	<b>Problem computation reduction</b>	<b>150</b>
<b>6.5</b>	<b>Implementation remarks</b>	<b>153</b>
6.5.1	Receding horizon strategy	153
6.5.2	Decision making algorithm	153
<b>6.6</b>	<b>Conclusion</b>	<b>156</b>

---

This chapter converges towards the algorithm that has been proposed to solve the problem of overtaking and lane change in a generic formulation based on optimization. We start from a formal description of the overtaking and lane change maneuvers, defining this ADAS function's objective. Then, different methods that can be used to treat this kind of maneuvers are revisited in a brief review of the state-of-the-art main trends on the trajectory planning methods and obstacle avoidance alternative formulations.

Afterwards, the chosen methodology is detailed, starting from the delimitation of the collection of scenarii that may arise when performing this kind of maneuvers

and the formulation of the anti-collision constraints. Then, a mixed-integer OCP is developed together with a series of tools that allow to lighten the a priori combinatorial computational burden of the formulation and bring it to a polynomial formulation in terms of binary variables. Ultimately, the goal is to give the reader an insight on the implementation details and understand the specific framework for the overtaking and lane change algorithm. Simulation results close this second part of the manuscript.

## 6.1 Problem formulation

Before proposing a formal description, let us begin with the discussion on what exactly an overtaking and lane change means in the ADAS framework. This can be stated in words: *An optimal lane change maneuver is such that drives the assisted vehicle to the adjacent lane by means of a comfortable and safe movement, after a driver's request is received through an indicator activation. In addition, an overtaking maneuver is related to a similar action performed in order to overpass a vehicle, where the main difference comes from the fact that the initial and final lane of the movement are the same one and it is performed in the presence of other vehicles.*

In practice, the overtaking maneuver can be seen as an operation composed of three stages, identified with a lane change phase, a lane centering phase and a second lane change movement that brings back the vehicle to the original lane (Figure 6.1). This means that both maneuvers can be executed by an assistance system that is equipped with the lane change, the lane centering (Chapter 4) and the cruise adaptive control features (Chapter 1).

**Remark 6.1** *Overtaking maneuver passing over two lanes can be considered as a consecutive lane change maneuver.*

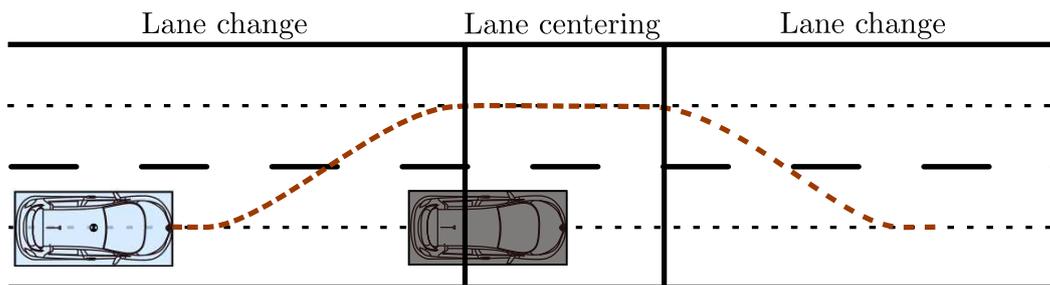


Figure 6.1 – Lane change and Overtaking maneuver

Technically speaking, an *optimal*, *comfortable* and *safe* maneuver is translated in terms of aggressive movements which need to be avoided, limiting the variation on the vehicle acceleration, that is, jerk. At the same time, the lateral acceleration and all system states must be maintained in safe bounds, all units stay within the lane

boundaries and the actuator constraints must be complied with. Likewise, on top of the internal limitations, safety is directly related with the exogenous factors, like the presence of surrounding vehicles, where the certification of no collision represents a critical feature to be considered in the design stage.

In Figure 1.1 a schematic representation of the overall ADAS control hierarchy has been depicted. In the following, attention is driven to the trajectory planner layer, while underlying control loops are assumed to be functioning in closed loop with appropriate feedback controllers (as discussed in the first part of the manuscript). The trajectory generator receives external information about the road (such as lane width, number of lanes or computed curvature of the road). Also, with respect to the obstacles, the relative position and movement is considered known with the onboard instrumentation. As a result this, the vehicle state vector is also available.

### 6.1.1 Trajectory planning strategies review

There exists a broad variety of trajectory planning methods. Traditionally, non-optimization based strategies [Chee 1994] have been used to generate trajectories for point-to-point motion. These are mostly based on predefined continuous-curvature paths, where trapezoidal, B-splines, clothoids or cosine profiles can be pointed out. The choice of these classes of curves aim to avoid non-smooth trajectories, that result in undesirable wear of tires [Vorobieva 2013].

A second group of methods are the exploration-based techniques, like the Rapidly Random Exploring Trees (RTT) algorithms [Pepy 2006], [Hwan 2011], where model dynamics are integrated to explore the surrounding environment and decide on the best feasible path. However, there are still critical issues, related to the computational burden, that prevent this technique to be used in real-time applications. A certain attention has been recently driven to this kind of techniques in [Weiss 2017], where the combination of the exploration technique with constraint admissible positively invariant sets provides a promising new insight with appropriate theoretical guarantees for the method.

There is also an important amount of work where optimization-based methods are used. Within this framework, 5th order polynomial parameterization is widely used, which results from the resolution of an Optimal Control Problem (OCP) via indirect optimization techniques [Rathgeber 2015]. As it has been stated in the previous chapter, the major drawback of this approach is the inherent difficulty to consider trajectory constraints, mainly related to the non-convexity of the feasible domain.

However, recently, direct optimization techniques are generating an increasing interest due to its ability to manage constrained systems in a systematic manner and important advances on the real-time capabilities on the resolution of the optimization problems [Diehl 2005]. In the same time, in [Mercy 2014] time-optimal collision-free trajectories are computed for a holonomic vehicle in terms of spline parameterizations of the trajectory, exploiting its geometric properties to reduce the

number of constraints. In [Gao 2010] a two-level Model Predictive Control (MPC) scheme is proposed, where the main objective is to enhance collision avoidance when following the center of the lane in terms of a distance-based term in the cost function. With this in mind, [Gao 2012] and [Frasch 2013] introduce a spatial reformulation of the vehicle model dynamics to eliminate model's speed dependency and formulate the anticollision constraints in terms of state constraints.

Finally, flatness-based trajectory planning methods [Murray 2009], [Milam 2003] can be seen as a complementary methodology to Optimal Control which is based on a property of some systems, that allows to compute the trajectories without having to integrate their dynamics: given a differentially flat system (Section 5.2), all its feasible trajectories can be written as a function of the flat output and its derivatives (Definition 5.6). This property is specially useful for trajectory generation applications, where the trajectories can be planned in the flat output space and afterwards being mapped to the appropriate inputs space. Nevertheless, the complexity of this methods arise from the nonlinearity of the system state limits when expressing them as flat output constraints, rendering nonlinear optimization problems.

### 6.1.2 Alternative obstacle avoidance formulations

The inclusion of obstacle avoidance capabilities in the trajectory planning algorithms remains a complex problem within dynamic environments, thus several techniques have been developed in the literature to formulate this particular kind of problem.

Potential fields methods (PFM) rapidly gained importance in the obstacle avoidance for robot manipulators framework, due to its simplicity in dealing with obstacle integration in the classical trajectory design. The main idea builds up from the definition of a potential field function around each one of the obstacles that need to be avoided, whose value increases inversely to the distance to the obstacle. With this in mind, the minimization of a cost function including the potential field functions will generate a path that does not collide with the obstacles. As a word of caution, in [Koren 1991] a series of disadvantages and identified problems of this kind of methods were analyzed, among which we can mention local minima problems due to the presence of several obstacles or non-passing trajectories when two obstacles are close to each other. All in all, active research interest has been manifested in this direction in connection with the automotive trajectory planning framework.

Focusing on the optimization-based trajectory planning methods, obstacle avoidance can be formulated by means of system state constraints, enforcing the containment of the solution in a certain region of the state-space, corresponding to the region free of collision. In these lines, separating hyperplane theory [Ziegler 2012] has been proposed in [Mercy 2014], where the enforcement of existence of a separating hyperplane between the obstacle and the ego vehicle ensures that both bodies remain separated along the generated trajectory. An alternative formulation is suggested in [Stoican 2011a], where hyperplane arrangement theory is exploited to define the free space for the trajectory generation purposes. The major advantage of this method

is the pre-definition of the separating hyperplanes, avoiding its online computation each iteration, thus reducing the computation burden. As a downside, this kind of method induces a mixed-integer formulation, which has been enhanced with certain simplifications as proposed in [Janeček 2017] and the enclosed overtaking and lane change framework allows to reduce the computational burden of this approach, as it will be presented later on this chapter.

## 6.2 Exhaustive scenarii description

This section characterizes the collection of possible situations that may arise in terms of collision avoidance formulation in the lane change and overtaking maneuver, allowing to delimit the context of the algorithm and argument for the choice of suitable modeling formalism. In this work, the addressed general scenario considers a two or three lanes one-way road (i.e. highways), where the final objective is to perform a lane change or overtaking maneuver (Section 6.1) in the presence of other vehicles, denoted as *objects* in the subsequent.

These vehicles, sharing the environment, can be located by mutual exclusion in three different relative positions with respect to the controlled vehicle, denoted as *ego*. This means that all the possible objects on this kind of roads are easily identified as one of the following predefined types (Table 6.1), and then characterized in real-time by its dimensions and movement information (and prediction), provided by the data fusion algorithms (Section 6.2).

Table 6.1 – Target definition

Target	Description	Schema (Figure 6.2)
Type 1	Object at original lane in front of Ego	T1
Type 2	Object at objective lane behind Ego	T2
Type 3	Object at objective lane in front of Ego	T3

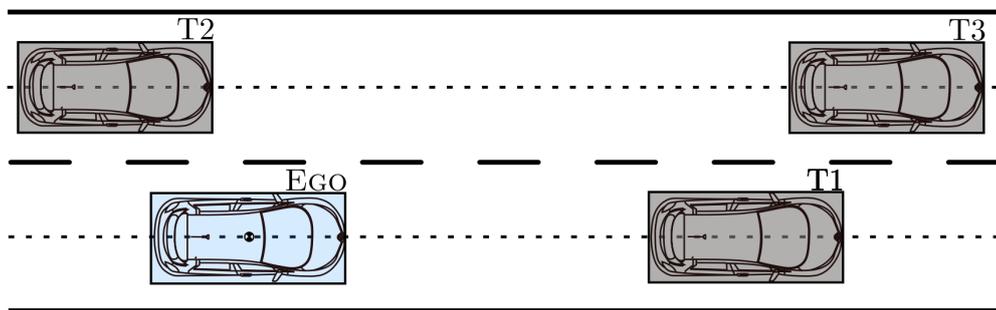


Figure 6.2 – Target definition schema

**Assumption 6.1** *The right lines are functioning with a lower speed that render*

*infeasible double lane undertaking maneuvers as long as the ego vehicle is in the range of action on a lane positioned at the left.*

**Assumption 6.2** *The cars manoeuvring on the original lane behind the ego car are governed by a certified algorithm that avoids collision from behind with the ego vehicle and interdict the undertaking maneuvers*

Once these object types have been defined, they can be combined in order to predefine the context in which the maneuver will take place. A complete description of the considered scenarii when changing to the left lane is shown in Table 6.2. Likewise, the same object types and symmetric scenarii can be considered when performing a change to the right lane.

### Target detection

Surrounding vehicles are detected by the vehicle's on-board instrumentation (1.2.2), whose captured information is provided to the sensor fusion algorithm. In the following, a brief description of the logic and the available information about the obstacles is given.

For the object detection and tracking feature, input information mainly comes from the radars located at the front bumper and back corners of the vehicle. This sensor will provide a collection of points, that are correlated to a object with a shape (i.e. a rectangle). Measured information comprises relative position with respect to the center of gravity of the ego vehicle and its derivatives up to the acceleration, together with the relative heading angle of the detected shape (6.1). A time-stamp  $t_k$  defining the detection time is assigned to each group of measurements.

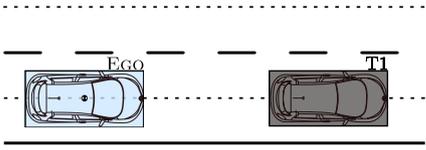
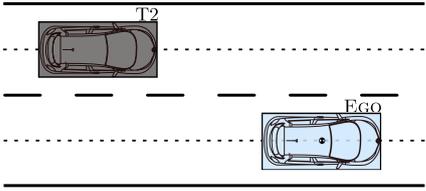
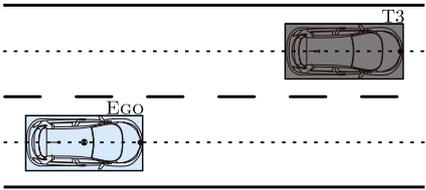
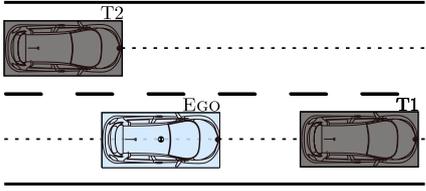
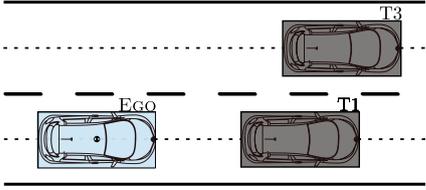
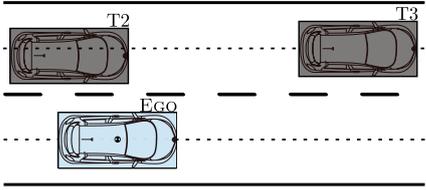
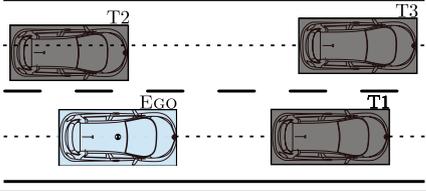
$$\xi_{object_k} = [p_k, v_k, a_k, \psi_k, \dot{\psi}_k]^T \quad (6.1)$$

where  $p_k = [p_{x_k}, p_{y_k}]^T$ ,  $v_k = [v_{x_k}, v_{y_k}]^T$  and  $a_k = [a_{x_k}, a_{y_k}]^T$  are provided in relative coordinates with respect to the ego vehicle.

Moreover, the object type (i.e. motorbike, truck, car) can be detected by evaluating the Radar Cross Section (RCS) of the object. This factor is influenced by several characteristics of the detected vehicle, where we can identify its building material or size. The higher the RCS is, the easier the object is detected. This feature allows to define a catalog of predefined vehicles, where the default size is registered and can be used in the case of a detection failure.

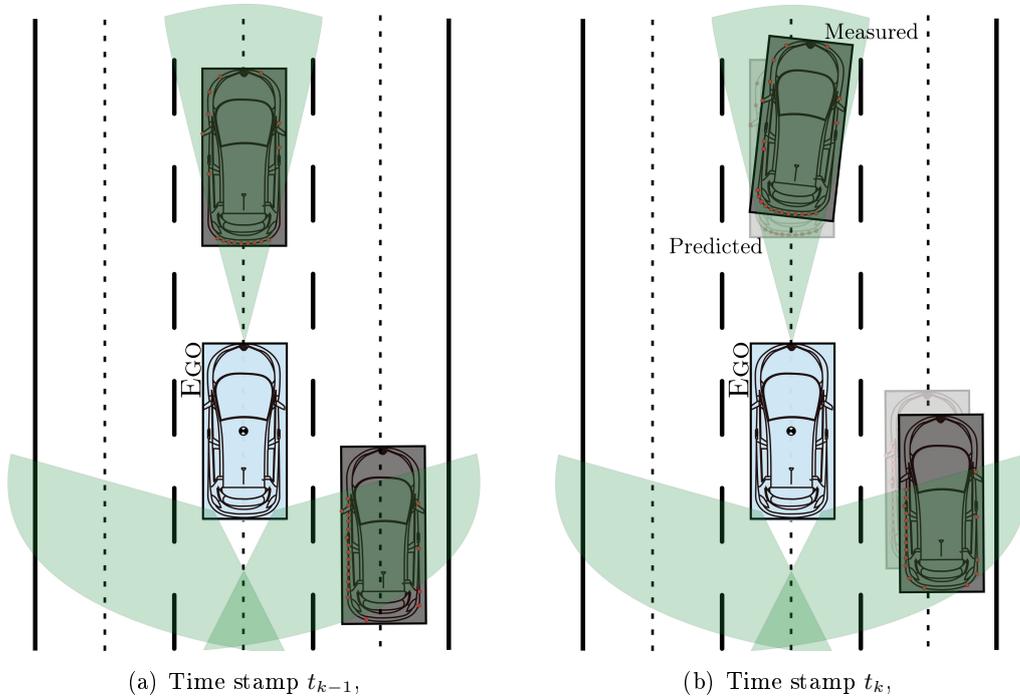
Whenever a new group of data at a generic time  $t_k$  is available, the information is checked against the processed in the previous time stamp  $t_{k-1}$ , and tracking and detection algorithm is executed (Algorithm 1).

Table 6.2 – Scenario definition

SCENARIO	DESCRIPTION	SCHEMA
<b>Scenario 1 One object detected</b>		
1.1	T1	
1.2	T2	
1.3	T3	
<b>Scenario 2 Two objects detected</b>		
2.1	T1&T2	
2.2	T1&T3	
2.3	T2&T3	
<b>Scenario 3 Three objects detected</b>		
3	T1&T2&T3	

**Algorithm 1** Target detection and tracking (pseudocode)

- 1: **Input:**
- 2: Read information  $\hat{\xi}_{object_{i_{k-1}}}$  at previous time stamp  $t_{k-1}$ , (Figure 6.3(a)).
- 3: Measure new states  $\xi_{object_{i_k}}$  and assign  $t_k$  as current time stamp (Figure 6.3(b)).
- 4: **Tracking:**
- 5: **for** Each detected object  $i$  at previous time stamp  $t_{k-1}$  **do**
- 6:   Update initial conditions:  $\xi_{object_{i_0}} = [p_{i_{k-1}}, v_{i_{k-1}}, a_{i_{k-1}}, \psi_{i_{k-1}}, \dot{\psi}_{i_{k-1}}]^T$ .
- 7:   Compute predicted states for time  $t_k$ :  $\tilde{\xi}_{object_{i_k}}$  by a constant acceleration or constant turn ratio and speed model from  $T_0 = t_{k-1}$  to  $T_F = t_k$  (Figure 6.3(b)).
- 8: **end for**
- 9: Compare distribution of predicted  $\tilde{\xi}_{object_{i_k}}$  and measured  $\xi_{object_{i_k}}$  states at time stamp  $t_k$ , compute Mahalanobis distance [Mahalanobis 1936], that provides a measurement on the similarity between two multidimensional variables.
- 10: Identify *tracked* objects by maximum superposing area  $\chi_{ij}^2$ , and/or define *new* objects, if any.
- 11: **Estimation:**
- 12: **for** Each new or tracked object,  $i$  **do**
- 13:   Compute  $\hat{\xi}_{object_{i_k}}$  by Kalman estimation.
- 14:   Save object states  $\hat{\xi}_{object_{i_k}}$  and time stamp  $t_k$  values for next iteration.
- 15: **end for**
- 16: **return** Estimated state  $\hat{\xi}_{object_{i_k}} = [\hat{p}_k, \hat{v}_k, \hat{a}_k, \hat{\psi}_k, \hat{\dot{\psi}}_k]^T$  and object bounding box length  $L_i$ , and width  $W_i$ .

Figure 6.3 – Points detection and tracking at time  $t_k$

## 6.3 Methodology

Once the obstacles and scenarii are characterized, we are ready to describe in a mathematical form the trajectory planning problem by means of the theoretical background that has been set up in Chapter 5 and formally describe the available data for the decision making.

First of all, the selected vehicle model is described. After that, we focus on the description of the feasible space in which the ego vehicle can move over, defined as a non-convex feasible region which is described in terms of hyperplane arrangements, leading to a mixed-integer formulation of the anticollision constraints. Moreover, the reduction of the necessary binary variables and cell merging techniques are considered and applied to the exhaustive enumeration of possible overtaking scenarii leading to a *minimal* representation in terms of binary variable (and representing one of the contributions of the present approach). Concerning the on-line control, attention is given to the formulation of a constrained optimal control problem which is translated into a finite dimension non-linear programming problem via direct optimization multiple-shooting approach that is solved in a receding horizon fashion, leading to the desired reference trajectory after a suitable change of coordinates.

### 6.3.1 Ego vehicle description

The first assumption coming from lower level controllers structure, shown in Figure 1.1, is the decoupling between the longitudinal and the lateral dynamics. This given schema provides a sub-optimal solution for the problem of trajectory generation. Nonetheless, such a control design structure has been set to allow a parallel development process for industrialization purposes. As stated in Section 6.1, the aim of the present section is to design a technique which provides the reference trajectory for the lower level controllers, which are already available and will function based on higher fidelity models. Hence, it will be acceptable to approximate the longitudinal and the lateral motion of the vehicle by a first-order differential equation in the jerk.

$$\begin{aligned} \dot{\xi}(t) &= A\xi(t) + Bu(t) \\ \xi(t) &= [x(t) \quad \dot{x}(t) \quad \ddot{x}(t) \quad y(t) \quad \dot{y}(t) \quad \ddot{y}(t)]^T \\ u(t) &= [\ddot{x}(t) \quad \ddot{y}(t)]^T \end{aligned} \quad (6.2)$$

where  $x(t)$ ,  $y(t)$  represent the longitudinal and lateral position respectively and  $\{\dot{\cdot}\}$  denotes time derivatives. State matrix  $A$  is a two block diagonal matrix, where each block is defined by a Jordan block,  $J_{0,3}$  and  $B = [0, 0, 1, 0, 0, 1]^T$ . The longitudinal position reference point will be set at the current position of the vehicle, while the lateral reference is set at the center of the lane from which the vehicle is starting the maneuver. The whole planning procedure is stated in these coordinates and represent the first stage of the trajectory generation.

In a second stage, the planned trajectory is used to generate the reference states provided to the underlying controllers. This transformation from planning to

reference trajectory is performed by the use of the *differentially flat* vehicle kinematic model (Section 5.2), which reads as follows,

$$\begin{aligned}\dot{x}(t) &= V(t)\cos(\psi(t)) \\ \dot{y}(t) &= V(t)\sin(\psi(t)) \\ \dot{\psi}(t) &= \frac{V(t)}{L}\tan(\delta(t))\end{aligned}\tag{6.3}$$

with  $\psi(t)$  being the heading angle of the vehicle and  $L$  its length. Input signals of the model are the front wheels steering angle  $\delta$  and vehicle speed  $V(t) = \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}$ . The state and input vectors are represented by

$$\begin{aligned}\eta(t) &= [x(t), y(t), \psi(t)]^T \\ u(t) &= [\delta(t), V(t)]^T\end{aligned}\tag{6.4}$$

The objective here is to provide the reference  $[\eta^{ref}(t), u^{ref}(t)]$  that guides the vehicle from one lane to the adjacent one. Since the adopted kinematic model is differentially flat, [Murray 2009], [Sira-Ramirez 2004], the system states and inputs can be computed as a function of the *flat outputs*  $z(t)$  and a finite number of its time derivatives. In this particular case,

$$z(t) = [z_1(t), z_2(t)] = [x(t), y(t)]^T\tag{6.5}$$

Hence, once the trajectory for the flat outputs and its derivatives has been computed by means of the model (6.2), we can obtain the required references by using a predefined basis function for the flat outputs as follows:

$$\begin{aligned}\eta^{ref}(t) &= \left[ z_1(t), z_2(t), \operatorname{atan}\left(\frac{\dot{z}_2(t)}{\dot{z}_1(t)}\right) \right] \\ u^{ref}(t) &= \left[ \operatorname{atan}\left( L \frac{\ddot{z}_2(t)\dot{z}_1(t) - \dot{z}_1(t)\ddot{z}_2(t)}{(\dot{z}_1^2(t) + \dot{z}_2^2(t))^{3/2}} \right), \sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)} \right]\end{aligned}\tag{6.6}$$

This strategy represents an open-loop control policy, so the mismatch between the model and the real system needs to be regulated based on a feedback loop, that is executed at the operational level (Section 1.2.1). On top of this, the saturations can play an important role in the tracking performances according to their activation in the process of transition from the model (6.2) to (6.3) and in general in the trajectory planning step.

### 6.3.2 Mathematical modeling of the highway scenarii: feasible non-convex region characterization

We start by the account of the feasible region for the overtaking and lane change maneuver: *Whenever performing an overtaking or lane change maneuver on a highway, the feasible space will be defined by the road physical limits, together with the*

presence of other vehicles in the surroundings. These will be defined by their shape, size, relative position and movement. In addition, safety traffic standards must be considered, such as the minimum safety distance that must be kept between all the vehicles and the interdiction of undertaking. In practice, the neighboring vehicles are seen as obstacles that can be identified with a polyhedral shape (the convex hull of its extreme points), which is ultimately correlated to a collection of hyperplanes that partition the maneuver space into feasible and unfeasible regions (Section 5.3.2). Thus, in order to define the corresponding hyperplane arrangement for each scenario, we need to compute the location of the limiting hyperplanes, which is directly obtained from the relative position and size of the targets, that are obtained from the bounding box information provided by the ego vehicle on-board instrumentation and the target detection and tracking algorithm (Section 6.2).

In addition, it needs to be noted that a point-mass model (6.2) has been chosen to represent the ego vehicle. This means that only a point located at its center of gravity is considered and its neglected dimensions could produce a collision by entering in the forbidden region of the space even if the center of gravity remains inside the feasible non-convex region. This can be overcome in terms of polyhedral Minkowski addition (2.8) of the ego vehicle shape  $E$  to each one of the targets,  $\bar{T}_j = T_j \oplus -E$ . In this way, the ego vehicle can be appropriately condensed to a point with respect to an obstacle that integrates the information of the collision geometry. In addition to this, as it has been established at the beginning of this section, road safety legislation needs to be considered, and the ego vehicle must keep a certain longitudinal distance  $d_{sfty}$  with the targets, so each one of the obstacles' polyhedral approximation has been enlarged horizontally to enforce such distance (Figure 6.4).

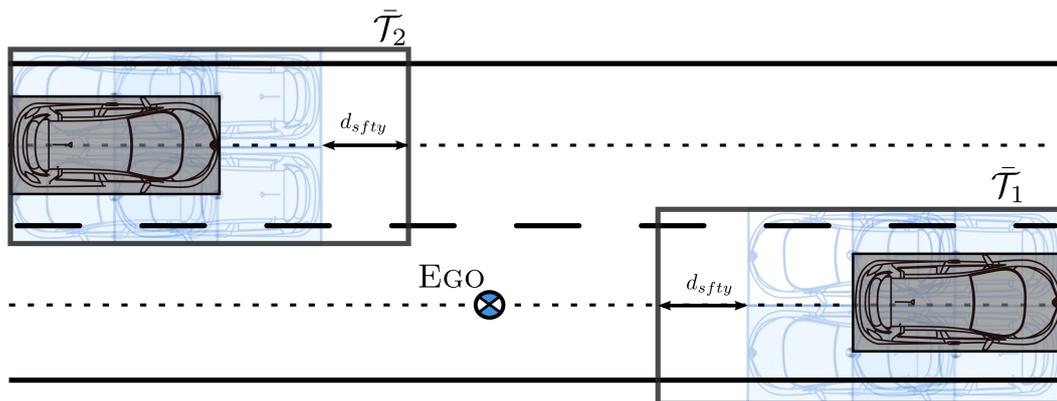


Figure 6.4 –  $\bar{T}_j$  definition

Once this is settled, hyperplane arrangements theory (Section 5.3) can be used to divide the space into feasible and unfeasible cells, where the latest will be defined by the presence of the obstacles ( $\bar{T}_j$ , with  $j = 1 \dots n_{obstacles}$ ) that need to be avoided and road traffic rules.

**Remark 6.2** *At this point, we may ask what is set first, the hyperplane arrangement or the obstacles? The answer is that it depends on the problem at hand. Should the design start with predefined obstacles (polyhedral sets) and find the hyperplane arrangement associated to them? Or, should the procedure start with a predefined hyperplane arrangement (e.g., by grid-ing the space and assigning to the resulting cells admissible/forbidden values)? Each approach has its merits. On one hand, gridding may produce over-approximations of the obstacles, in exchange of a reduced/fixed complexity and, on the other hand, pre-defining the hyperplane arrangement can speed up the anti-collision constraints computation.*  $\square$

In the kind of scenario we are working with, pre-definition of the hyperplane arrangement is an appealing option, as the enclosed collection of scenarii that may arise in terms of the lane change or overtaking maneuvers allows to pre-define the situation and speeding up the anticollision constraints. Typically, for the defined scenarii, this feasible region will be non-convex, still, it can be expressed as the union of  $l$  scenario-dependent convex overlapping cells  $\mathcal{F}_l$  covering the whole feasible space (5.46). Table 6.3<sup>1</sup> shows the feasible (green) and unfeasible (grey) cells for each one of the scenarii introduced in Section (6.2) under Assumption 6.3.

**Assumption 6.3** *The ego vehicle is the only one changing the lane. This rule can be enforced in the case of communicating vehicles or simply handled by a supervisory level which switch the autonomous driving to a different (safety) maneuver whenever the hypothesis is not fulfilled.*

However, it may be conservative to consider that only the ego vehicle is performing a lane change when we are driving in a dynamic environment such as the highway. In addition, available current camera image processing algorithms are able to detect target's lateral indicators, and thus target maneuver, known by *cut-in cut-out* maneuver, can be detected on the data fusion algorithms level. Hence, if a target indicator is detected, an update on the convex feasible region is needed, where the overtaking of the target which is changing the lane is forbidden. This provides the scenarii description depicted in Table 6.3.

Furthermore, it must be considered that the collection of scenarii and the corresponding hyperplane arrangement have been defined for a set of predefined targets (Table 6.2) that are in a straight position, that is, their heading angle is not considered.

**Assumption 6.4** *Rotated targets, defined by its length ( $L_{target}$ ), width ( $W_{target}$ ) and rotation angle ( $\psi$ ), are overapproximated by a bounding box whose heading angle is zero (Figure 6.5). The size of the approximating box is defined by*

$$\begin{aligned} L_{box} &= W_{target} \sin(\psi) + L_{target} \cos(\psi) \\ W_{box} &= W_{target} \cos(\psi) + L_{target} \sin(\psi) \end{aligned} \quad (6.7)$$

$\square$

---

1. The squemas are not drawn in a real scale.



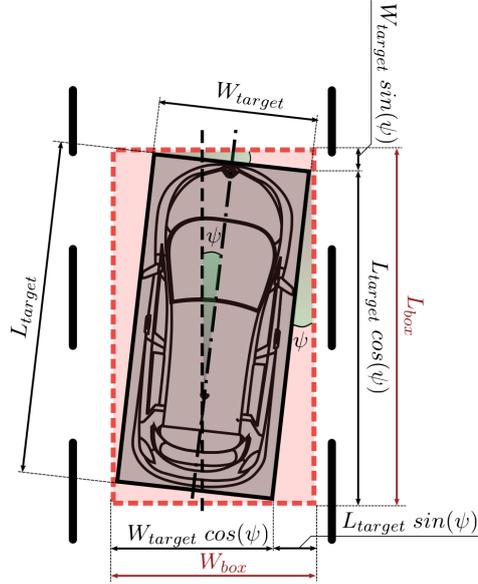


Figure 6.5 – Rotated target bounding box definition

Assumption 6.4 allows to generalize the application framework of the pre-defined scenarii (Tables 6.2 and 6.3) and the corresponding hyperplane arrangements to a wider collection of situations that may arise, at the cost of overapproximating the target size. Once the feasible and unfeasible regions are defined, the notions of merging techniques (Section 5.3.3) are applied to minimize the number of cells that can be described by a reduced amount of binary variables (Section 5.3.4.1). In this way, the containment of the flat output  $z(t)$  (6.5) inside the corresponding convex merged feasible regions  $\mathcal{F}_l$  can be enforced in terms of mixed-integer formulation (Section 5.3.4).

**Procedure 6.1** Feasible non-convex region definition for each scenario.

1. Obstacle definition (Section 6.2)
2. Hyperplane arrangement definition (5.7).
3. Identify feasible (5.44), (5.45) and forbidden cells (5.46), (5.47).
4. Study cell merging possibilities (5.48).
5. Define binary variables (5.55).
6. Formulate non-convex region inequalities (5.58).

In order to provide a better exposition of Procedure 6.1, an illustrative example is depicted in the following.

**Example 6.1** Figure 6.6 shows a representation of an Scenario type 2.3, where a target of type T2 and type T3 are present (Table 6.2). The final objective here is to compute the region of space where the ego vehicle must stay during the maneuver, that is,  $\bar{\xi} = [\xi(1), \xi(4)]^T \in \mathcal{C}(\bar{\mathbb{T}})$ .

This feasible space is defined by the complement of the union of the two regions delimited by the targets and road standards  $\bar{\mathbb{T}} = \bar{\mathcal{T}}_2 \cup \bar{\mathcal{T}}_3 \cup \mathcal{T}_b$ . We consider the

collection of  $\mathbb{H} = \{\mathbb{H}\}_{i=1:4}$  hyperplanes, from which we obtain a total of 9 cells (Section 5.3.1). On one hand, the unfeasible ones are defined by the presence of the two targets  $\bar{\mathcal{T}}_2 = \mathcal{H}_1^+ \cap \mathcal{H}_4^-$ ,  $\bar{\mathcal{T}}_3 = \mathcal{H}_2^- \cap \mathcal{H}_3^-$  and  $\bar{\mathcal{T}}_b = \mathcal{H}_2^- \cap \mathcal{H}_3^+$  (5.46), defined unfeasible to avoid undertaking maneuvers. On the other hand, the non-convex

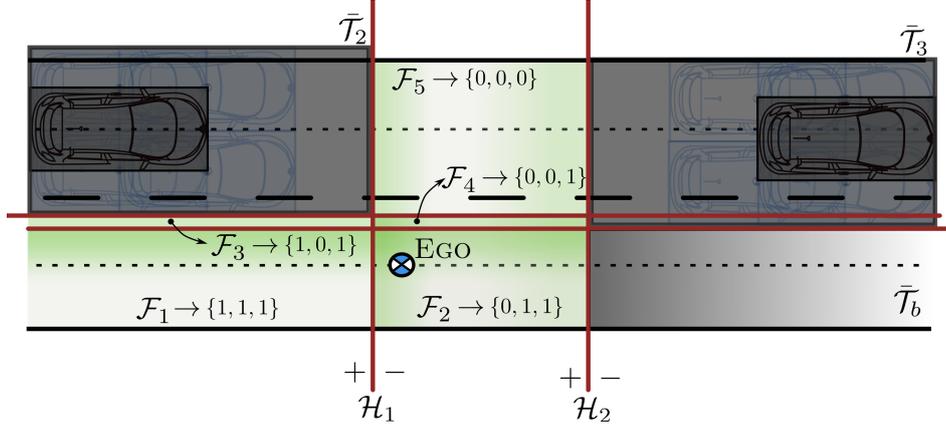


Figure 6.6 – Hyperplane arrangement scenario 2.3

feasible region is defined by 5 cells (5.44),  $\mathcal{F}_j$ ,  $j = 1 \dots 5$ . Following the lines introduced in Section 5.3.4.1, we can describe the feasible region  $\mathcal{C}(\bar{\mathbb{T}})$  with the following set of inequalities (6.8), expressed in the form (5.58). Moreover, it can be seen that each one is described by unique tuple  $\beta \in \{0, 1\}^{N_0}$  with  $N_0 = \log_2(5) = 3$  (5.55).

$$\begin{aligned}
 \mathcal{F}_1 & \begin{cases} h_1 \bar{\xi}(t) \leq g_1 \\ h_3 \bar{\xi}(t) \leq g_3 \end{cases} & +M(3 - \beta_1 - \beta_2 - \beta_3) \\
 \mathcal{F}_2 & \begin{cases} -h_1 \bar{\xi}(t) \leq -g_1 \\ h_3 \bar{\xi}(t) \leq g_3 \\ h_2 \bar{\xi}(t) \leq g_2 \end{cases} & +M(2 + \beta_1 - \beta_2 - \beta_3) \\
 \mathcal{F}_3 & \begin{cases} h_1 \bar{\xi}(t) \leq g_1 \\ -h_3 \bar{\xi}(t) \leq -g_3 \\ h_4 \bar{\xi}(t) \leq g_4 \end{cases} & +M(2 - \beta_1 + \beta_2 - \beta_3) \\
 \mathcal{F}_4 & \begin{cases} -h_1 \bar{\xi}(t) \leq -g_1 \\ -h_3 \bar{\xi}(t) \leq -g_3 \\ h_2 \bar{\xi}(t) \leq g_2 \\ h_4 \bar{\xi}(t) \leq g_4 \end{cases} & +M(1 + \beta_1 + \beta_2 - \beta_3) \\
 \mathcal{F}_5 & \begin{cases} -h_1 \bar{\xi}(t) \leq -g_1 \\ -h_4 \bar{\xi}(t) \leq -g_4 \\ h_2 \bar{\xi}(t) \leq g_2 \end{cases} & +M(\beta_1 + \beta_2 + \beta_3)
 \end{aligned} \tag{6.8}$$

The next natural step is to review the possibility of cell merging (Section 5.3.3), to analyze if the feasible region can be further simplified. For the lane change maneuver, the scenarios are simple enough to study cell merging by means of a Karnaugh map (Section 5.3.3), obtaining a simplified formulation with  $C_f$  regions, where  $f \in \{1, 2\}$ .

In this manner,  $C_1 = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3 \cup \mathcal{F}_4$  defined by the tuple  $\sigma_1^* = (*, *, -)$  and  $C_2 = \mathcal{F}_4 \cup \mathcal{F}_5$  with  $\sigma_2^* = (-, -, *)$ . Accordingly, only one binary variable  $\lambda$  is now needed to define the anticollision constraints,  $N_o = \log_2(2) = 1$ .

$$\begin{aligned}
 C_1 & \begin{cases} h_2 \bar{\xi}(t) \leq g_2 \\ h_4 \bar{\xi}(t) \leq g_4 \end{cases} & +M(\lambda) \\
 C_2 & \begin{cases} -h_1 \bar{\xi}(t) \leq -g_1 \\ h_3 \bar{\xi}(t) \leq g_3 \\ h_2 \bar{\xi}(t) \leq g_2 \end{cases} & +M(1 - \lambda)
 \end{aligned} \tag{6.9}$$

◇

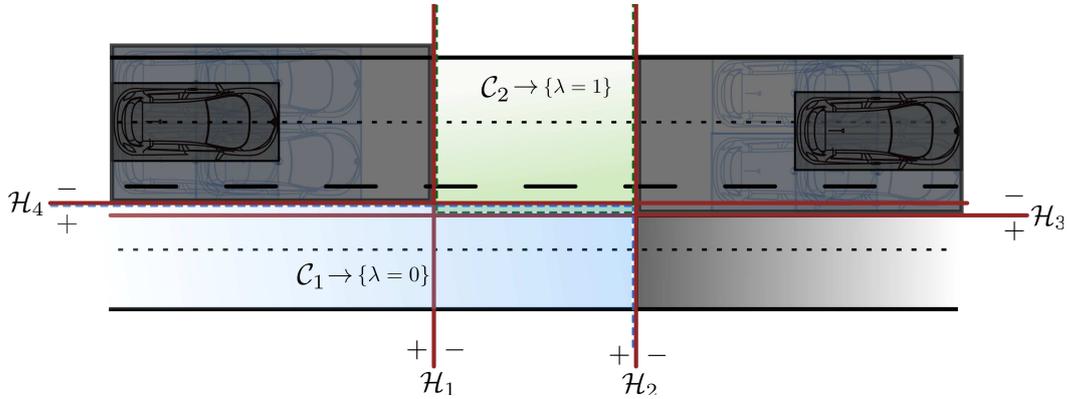


Figure 6.7 – Non-convex feasible region definition:  $C_1 \cup C_2$

**Remark 6.3** For the other scenarii depicted in Table 6.2, two convex feasible regions can be defined in a similar manner, thus only one binary variable is needed in each case, with the exception of Scenarios 1.2 and 1.3, where there is a unique convex feasible region.

### 6.3.3 Optimal Control Problem formulation for collision-free trajectory planning

In the subsequent, the main objective of trajectory planning problem is established. After that, the collection of constraint inequalities that ensure the fulfillment system and environment limitations are defined to finally get to the final OCP problem formulation for collision-free trajectory planning on highways.

In practice, a direct multiple shooting strategy (Section 5.1.3.3) with  $K$  sub-intervals has been used to transform the constrained OCP into such NLP, which is afterwards solved by one of the efficient state-of-the-art available NLP solving methods, like the sequential quadratic programming (SQP) or the real-time iteration scheme [Diehl 2006], [Diehl 2005].

### 6.3.3.1 Objective function

Maximizing the driver comfort is one of the main priorities of ADAS, alongside ensuring safety through collision avoidance and stabilizing the vehicle dynamics. Driver comfort is typically measured via the jerk levels perceived by the passengers, very sudden movements that would arise in aggressive or evasive maneuvers, which would be only used in emergency situations, are to be avoided. A successful trajectory generator ought to effectively address these priorities.

The solution that maximizes passenger comfort but does not unnecessarily extend the maneuver<sup>2</sup> is defined. Hence, an objective function which minimizes the control input, regarding the final time of the maneuver is settled, providing a suitable trajectory which renders a trade-off between a minimum-jerk trajectory and maneuver total time.

$$\min_{\xi(t), u(t), T_F} \int_{T_0}^{T_F} u^T(t) \bar{Q} u(t) dt + \bar{R} T_F^2 \quad (6.10)$$

### 6.3.3.2 Problem constraints

Constraints fulfillment guarantees is one of the most attractive features of the on-line constrained optimization-based approaches. Generally, constraint sources include vehicle limitations, which must be considered when designing control systems as well as for performing trajectory planning strategies, together with maneuver characterization and environment limitations, that need to be included in the formulation (Figure 6.8).

- **SYSTEM LIMITS.** This kind of constraints are defined by the dynamical behaviour of the system, which is encoded by the model ODE (6.2). In addition, its physical restrictions and actuation limits, are formulated by state and input constraints in a polytopic form (2.16a, 2.16b). Here we can include maximal acceleration capability of the car or the maximal desired jerk of the maneuver.
- **MANEUVER CHARACTERIZATION.** Constraints on the initial and final states define the boundary conditions of the desired trajectory, providing the starting and final point to the algorithm. Moreover, the maneuver can be characterized by additional features, where we can mention the maximum duration of the total maneuver time, which is enforced by traffic regulations or the directionality of the lane change (right-to-left or left-to-right).
- **ENVIRONMENT LIMITATIONS.** This type of exogeneous restrictions comprise the collection of constraints that come from external elements to the ego vehicle. Here we can mention the road speed limits or width of the lanes  $W_l$ .

---

2. In the absence of obstacles, if time is not included in the problem cost function  $J$ , it is trivial to notice that the optimal solution renders a enlarged maneuver (or  $T_F \approx T_{max}$  if constrained) over space and time in such a way the jerk is kept close to zero.

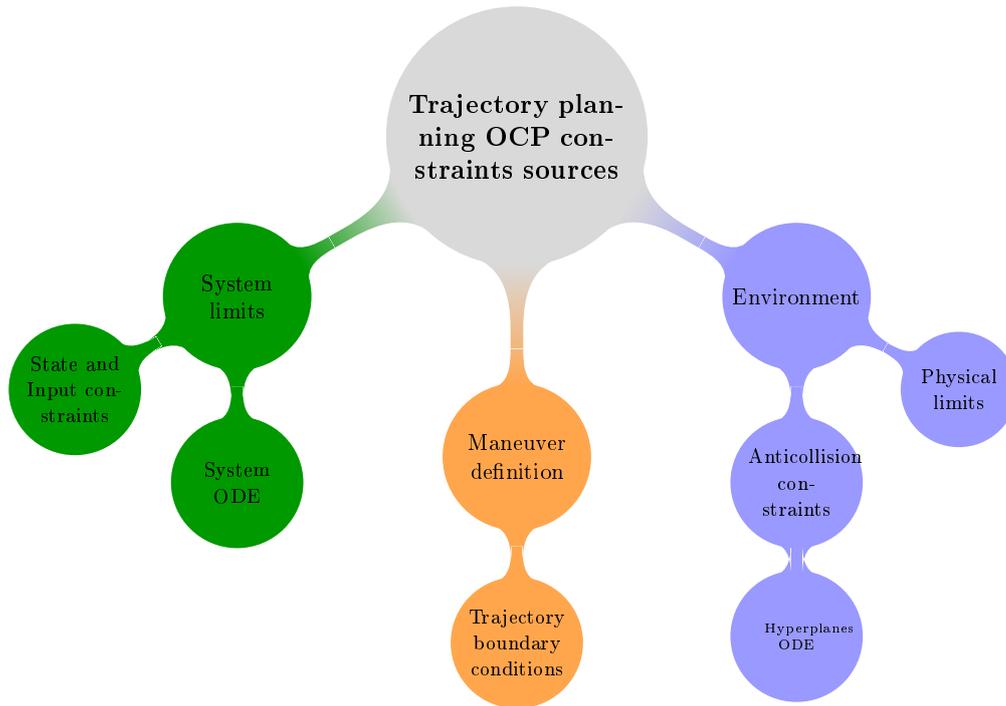


Figure 6.8 – Trajectory planning OCP constraint sources

On top of this, anticollision constraints are enforced by means of the feasible non-convex domain of the space where the vehicle must stay during the maneuver.

In this context, the defined numerical values for the state and input limits are depicted in Table 6.4.

Table 6.4 – Numerical values for trajectory states and input constraints

State	Range	Definition
$\dot{x}(t)$	[60, 130]	Longitudinal speed by road limits [km/h]
$\ddot{x}(t)$	[ $\pm 3$ ]	Longitudinal accel. by comfort limits [m/s <sup>2</sup> ]
$\ddot{\dot{x}}(t)$	[ $\pm 1.3$ ]	Longitudinal jerk by comfort limits [m/s <sup>3</sup> ]
$y(t)$	[ $-0.5W_l, 1.5W_l$ ]	Lateral displacement by road width [m]
$\dot{y}(t)$	[ $\pm 2.5$ ]	Lateral speed by comfort limits [m/s]
$\ddot{y}(t)$	[ $\pm 0.5$ ]	Lateral accel. by comfort limits [m/s <sup>2</sup> ]
$\ddot{\dot{y}}(t)$	[ $\pm 0.7$ ]	Lateral jerk by comfort limits [m/s <sup>3</sup> ]

### 6.3.3.3 Complete OCP formulation

Previous sections have described all the ingredients that are needed to formulate the general planning optimization problem to be solved to obtain a suitable collision-

free trajectory, that will be translated (via flatness) into a suitable reference provided to the lower level controllers. We see in this section that the OCP is formulated as an optimization that involves a quadratic cost function and the description of the non-convex anticollision constraints formulated in terms of hyperplane arrangements, that introduce an integer binary variable in the formulation.

$$\min_{\xi(t), u(t), T_F} \int_{T_0}^{T_F} u^T(t) \bar{Q} u(t) dt + \bar{R} T_F^2 \quad (6.11)$$

$$\text{subject to} \quad T_F \leq T_{max}, \quad \text{Maximum maneuver time} \quad (6.12a)$$

$$\xi(0) - \xi_{T_0} = 0, \quad \text{Ego initial conditions} \quad (6.12b)$$

$$\xi(T_F) - \xi_{T_F} = 0 \quad \text{Ego final conditions} \quad (6.12c)$$

$$q(\xi(t), u(t)) \geq 0, \quad \text{Path constraints (Table 6.4)} \quad (6.12d)$$

$$\dot{\xi}(t) - f(\xi(t), u(t)) = 0 \quad \text{Ego ODE model (6.2)} \quad (6.12e)$$

$$\bar{\xi} \in \mathcal{C}(\bar{\mathbb{T}}) \quad \text{Anticollision constraints (5.58)} \quad (6.12f)$$

$$g_i(0) - g_{i0} = 0, \quad \text{Hyperplanes } \mathcal{H}_i \text{ initial conditions} \quad (6.12g)$$

$$\dot{g}_i(t) - f(g_i(t), u_i) = 0, \quad \text{Hyperplanes } \mathcal{H}_i \text{ ODE (6.13)} \quad (6.12h)$$

with  $t \in [0, T_F]$  and  $\bar{Q} \succeq 0, \bar{R} \succ 0$  denoting the weighting matrices. The total time of the maneuver  $T_F$  has been left as an optimization parameter within a certain value that limits the maximal duration of the generated trajectory,  $T_{max}$  (6.12a). In addition to this, (6.12b) defines the initial conditions  $\xi_0$ , which are set according to latest measurements. (6.12d) includes the path constraints that take into account actuator, road and comfort limits on the system states (Section 6.3.3.2). Final conditions (6.12c) are designated at the objective lane, with zero acceleration and jerk values. Final speed is adapted accordingly and left as a free optimization parameter if a target type T2 or T3 (Table 6.1) is involved. If not, desired set speed is used.

In addition, if other vehicles are detected on the scene, anticollision constraints (6.12f - 6.12h) are considered, enforcing the inclusion of the trajectory waypoints into the scenario-dependent non-convex feasible region  $\bar{\xi} \in \mathcal{C}(\bar{\mathbb{T}})$  (Table 6.3), where  $\bar{\xi} = [x, y]^T$ . In this way, it is ensured that the position way-points of the generated trajectory corresponds to a collision-free maneuver and remains inside the feasible space. Nevertheless, to be able to ensure this containment along the full generated trajectory, it is necessary to take into account the fact that the target vehicles are moving (Remark 6.4).

**Remark 6.4** *The feasible region is defined by the limiting hyperplanes of the target vehicles  $\bar{T}_j$ , which are moving and changing their position along the period of time in which the maneuver is calculated and executed.*  $\square$

The main consequence coming from this fact is that the feasible region is evolving, thus the anticollision constraints are time-dependent, making it necessary to predict

their position along the time window in which the maneuver is computed. For this purpose, it has been considered that the hyperplanes  $\mathcal{H}_i$  (2.4) defining the scenario-dependent hyperplane arrangement have constant orientation, so only the right hand side part of the inequalities depends on time  $g_i(t)$ .

Accordingly, the dynamics of each hyperplane  $i \in [1, N]$  have been modelled as a constant speed integrator (6.13), where the input  $u_i$  is defined by the target's longitudinal or lateral speed for vertical or horizontal hyperplanes respectively. If target acceleration was to be considered, the order of the integrator could be accordingly augmented to consider such information.

$$\dot{g}_i(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} g_i(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i \quad (6.13)$$

Once the full infinite-dimensional OCP is defined, the corresponding finite-dimensional NLP is formulated and solved to generate the collision-free trajectory (Section 5.1.3). In our implementation, a direct method (5.1.3.3) is used to solve the OCP. More precisely, a direct multiple shooting approach is used.

The major reason that leads to the general preference for direct methods when solving OCPs in most engineering applications is that indirect methods are very accurate, but they are based on the resolution of the necessary conditions for an optimal trajectory (Section 5.1.3.2). This means that the knowledge on active and inactive inequality constraints in advance is required. Moreover, any switching of inactive and active constraints need to be parameterized up front. State-space methods will suffer from the same limitation as well, together with their inherent *curse of dimensionality* (Section 5.1.3.1).

It is straightforward to realize that this is a sufficient condition to rule out this two methods, as the studied application needs a strategy that is able to deal with highly dynamic scenarios, where the location of the targets can be parametrized in terms of the exhaustive scenario description, but their behaviour and movement cannot be predicted beforehand.

Including all this preliminary considerations, the resulting NLP problem is formulated:

$$\min_{s, q, \lambda_k, T_F} \sum_{k=0}^{K-1} l_k(s_k, q_k, T_F) + E(s_{T_F}) \quad (6.14)$$

subject to

$$T_F \leq T_{max}, \text{ Maximum maneuver time} \quad (6.15a)$$

$$s(0) - s_{T_0} = 0, \text{ Ego initial conditions} \quad (6.15b)$$

$$s(T_F) - s_{T_F} = 0, \text{ Ego final conditions} \quad (6.15c)$$

$$s_{k+1} - \xi(t_{k+1}, s_k, q_k) = 0, \text{ Continuity conditions} \quad (6.15d)$$

$$h(\xi(t_k, q_k), u(t_k, q_k)) \geq 0, \text{ Discretized path constraints} \quad (6.15e)$$

$$g_i(0) - g_{i0} = 0, \text{ Hyperplanes initial conditions} \quad (6.15f)$$

$$\mathcal{F}_l \begin{cases} \sigma_l(1)h_1\bar{\xi} \leq \sigma_l(1)g_{i_k} \\ \vdots \\ \sigma_l(N_a)h_{N_a}\bar{\xi} \leq \sigma_l(N_a)g_{N_{a,k}} \end{cases} + M\lambda_k, \text{ Discretized anticollision constraints} \quad (6.15g)$$

$$\lambda_k \in \{0, 1\}^{N_0}, \text{ Binary variable} \quad (6.15h)$$

with  $k \in [0, K]$ ,  $i \in [0, N_a]$  and  $l \in [1, N_l]$ . (6.15a) keeps the same form as in (6.12a). Then, (6.15b- 6.15c) correspond to the initial and final states of  $s_i$ , which is the initial guess at each node of the time grid, which arises from the application of the multiple-shooting strategy to transcribe the infinite dimensional OCP into the NLP, together with the continuity constraints (6.15d) between consecutive subintervals.

In addition, (6.15f - 6.15h) translate the anticollision constraints (6.12f - 6.12h), where the RK4 numerical integration method is used to compute the predicted  $i$ -th hyperplane right hand side  $g_{i_k}$  at the corresponding  $k$ -th time node, considering a constant input and starting from the initial conditions (6.15f), updated according to the last available target measurements. (6.15g) lists the group of defining inequalities for each feasible cell, which will be activated by means of the binary variables vector  $\lambda_k$  (6.15h).

#### 6.3.3.4 Illustrative simulation

In the following, we compare two generated trajectories: the first one is a single lane change maneuver in the absence of other vehicles, so no anti-collision constraints are needed on the formulation, the obtained solution is a pure optimal jerk-time trajectory (Figure 6.11). Then, we introduce another vehicle of type T1 in the scene, which is driving at a relative speed of  $-10$ [m/s]. This slower vehicle produces the generation of several feasible trajectories, where the difference between them is the moment when the switch in the binary variable indicating the state containment in the convex feasible region  $\mathcal{C}_1$  or  $\mathcal{C}_2$  is performed. Two main kinds of trajectories can be distinguished, where some of them are based on aggressive maneuvers with

early stage switch or maneuvers where the ego vehicle brakes before changing the lane. We can see on Figure 6.9 the values of the cost function and total time of the maneuver with respect to the multiple shooting node  $k$  when the switching takes place. It can be seen that the ones with a later switch are closer to the optimal solution in the absence of obstacles.

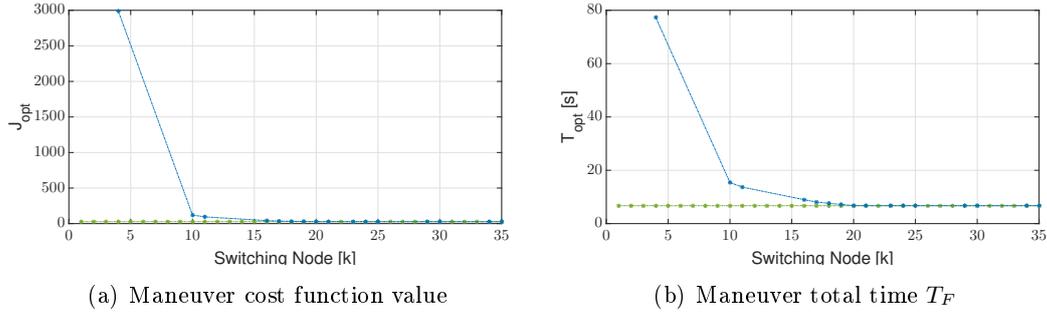


Figure 6.9 – Cost Function value and Total time of the maneuver

Figure 6.10 shows the computation time of each one of the generated maneuvers in a i7-6700 CPU at 3.4GHz. As expected, the inclusion of anticollision enhancements ensures the generation of a safe trajectory from the design stage, but at the cost of more computational resources. The numerical solution is computed using the open-source framework YALMIP [Lofberg 2005].

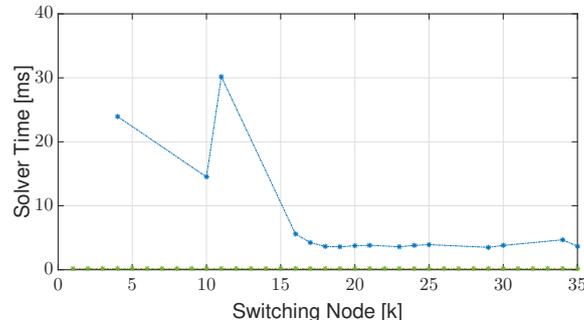


Figure 6.10 – Computation time for maneuver generation

### 6.3.4 Trajectory reference transformation step

Once the planning step is executed, and the formulated NLP (6.14), subject to (6.15a)-(6.15h) is solved, the planned trajectories for system (6.2) are available. As it has been previously introduced in Section 6.3.1, underlying controllers references will be provided in a different set of coordinates, that is,  $\eta^{ref}$ ,  $u^{ref}$ , that can be obtained from the flat non-holonomic kinematic model (6.3).

Two different strategies could be considered at this stage. First, the computed trajectories for  $\xi$  can be seen as collision-free intermediate waypoints, used as bound-

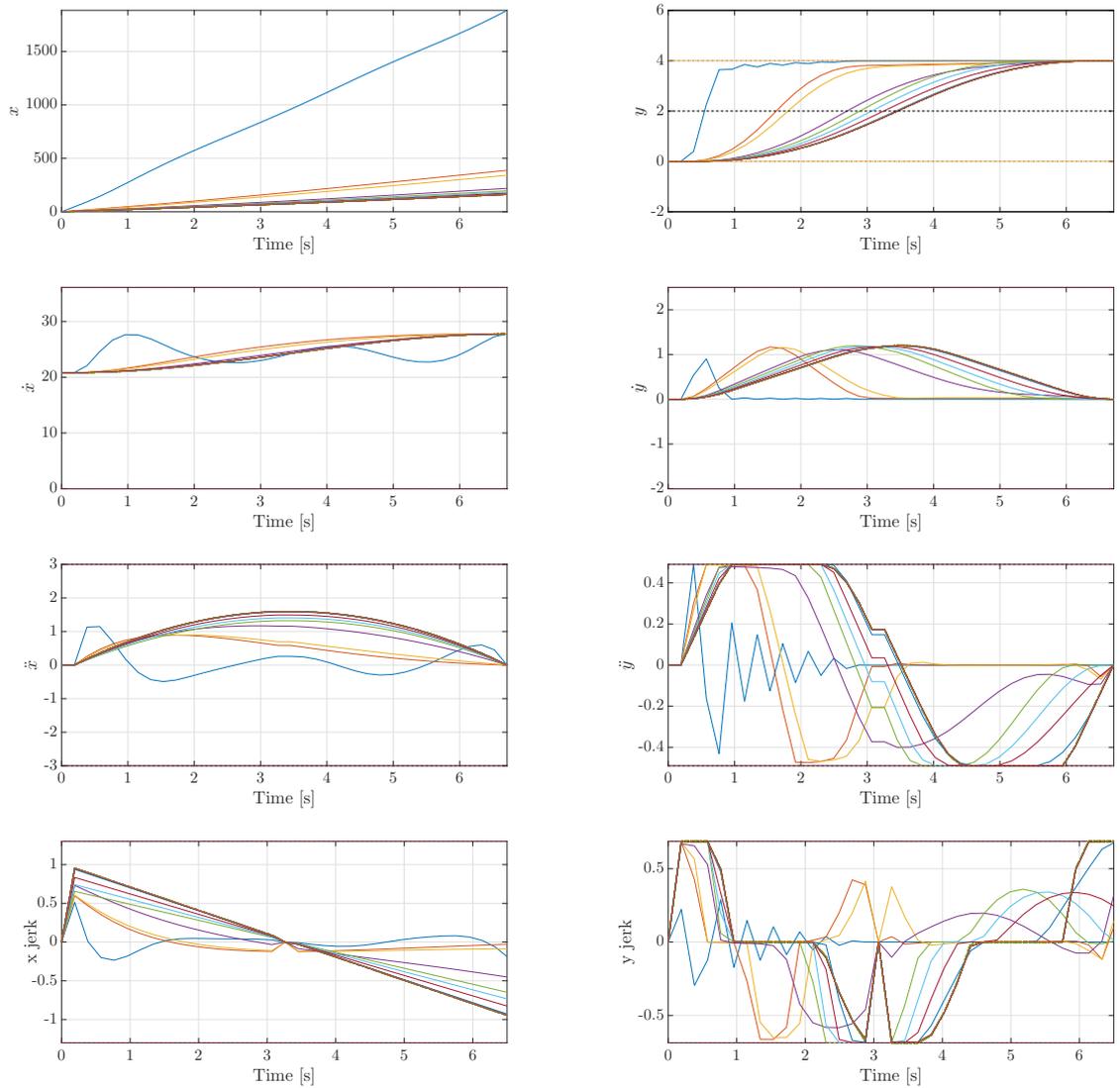


Figure 6.11 – Lane change generated trajectories in the absence of other vehicles (green) and collection of all feasible trajectories in the presence of an slower obstacle of type T1.

ary conditions to perform a replanning of the final trajectory with a parameterization in the flatness-based framework (Section 6.3.1). Nevertheless, this strategy has a major drawback: computational burden is clearly increased, as we are computing the trajectory twice. Moreover, if constraints on the reference states  $\eta^{ref}$  are to be added, they would result in non-linear constraints on the flat output, so computation load is increased anyways. In addition, an oscillatory behaviour in between the way-points is induced in the trajectory, as we are fitting 5th-order polynomials in between the points. This is shown in Figure 6.14.

Alternatively, the strategy that has been considered is to directly define the flat output by means of the optimal trajectories of  $\xi$ , that is,  $z(t) = \bar{\xi}(t)$  and the corresponding derivatives. Then, linear evolution in between the waypoints is considered, avoiding the oscillatory behaviour on the generated trajectory (Figure 6.15) and reducing the computational burden. In this lines, we do not strictly parametrize the flat output, but we use an OCP to compute its trajectories. Again, the formulated OCP (6.11), (6.12a)-(6.12h)) does not consider constraints on the  $\eta^{ref}$  states, so saturations need to be checked a posteriori.

#### 6.3.4.1 Numerical simulation

This section presents the results from the generation of a reference trajectory for a lane change in the presence of surrounding vehicles in a Scenario type 2.3 (Table 6.2). The objective is to generate a trajectory that changes the lane in the presence of these two vehicles and ends up the maneuver in between both of them, staying within the safety distance and internal kinematic constraints. Again, the numerical solution is computed using the open-source framework YALMIP [Lofberg 2005] and the formulated NLP by means of the multiple-shooting method for its discretization has been solved by means of the open-source NLP solver IPOPT [Wächter 2006].

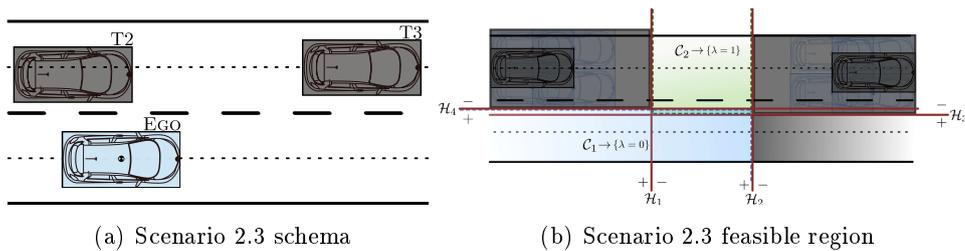


Figure 6.12 – Scenario type 2.3

Target T3 is initially at 120[m] ahead the ego vehicle, driving at a positive relative speed  $-2.5$ [m/s]. The lateral relative distance between both is 3.2[m]. Target T2 is approaching the ego vehicle from 70[m] behind with a relative speed of 2.5[m/s]. This kind of scenario will produce an adaptation on the longitudinal speed in order to get incorporated into the objective lane without interceding on the target T2 trajectory and not get too close to target T3. The time evolution of the generated profiles are shown in Figures 6.13. It can be seen that the generated trajectory

produces a longitudinal and lateral accelerated movement, allowing the ego vehicle to get incorporated in the existing gap between the obstacles without collision. Once this is achieved, the trajectory smoothly converges to the final conditions.

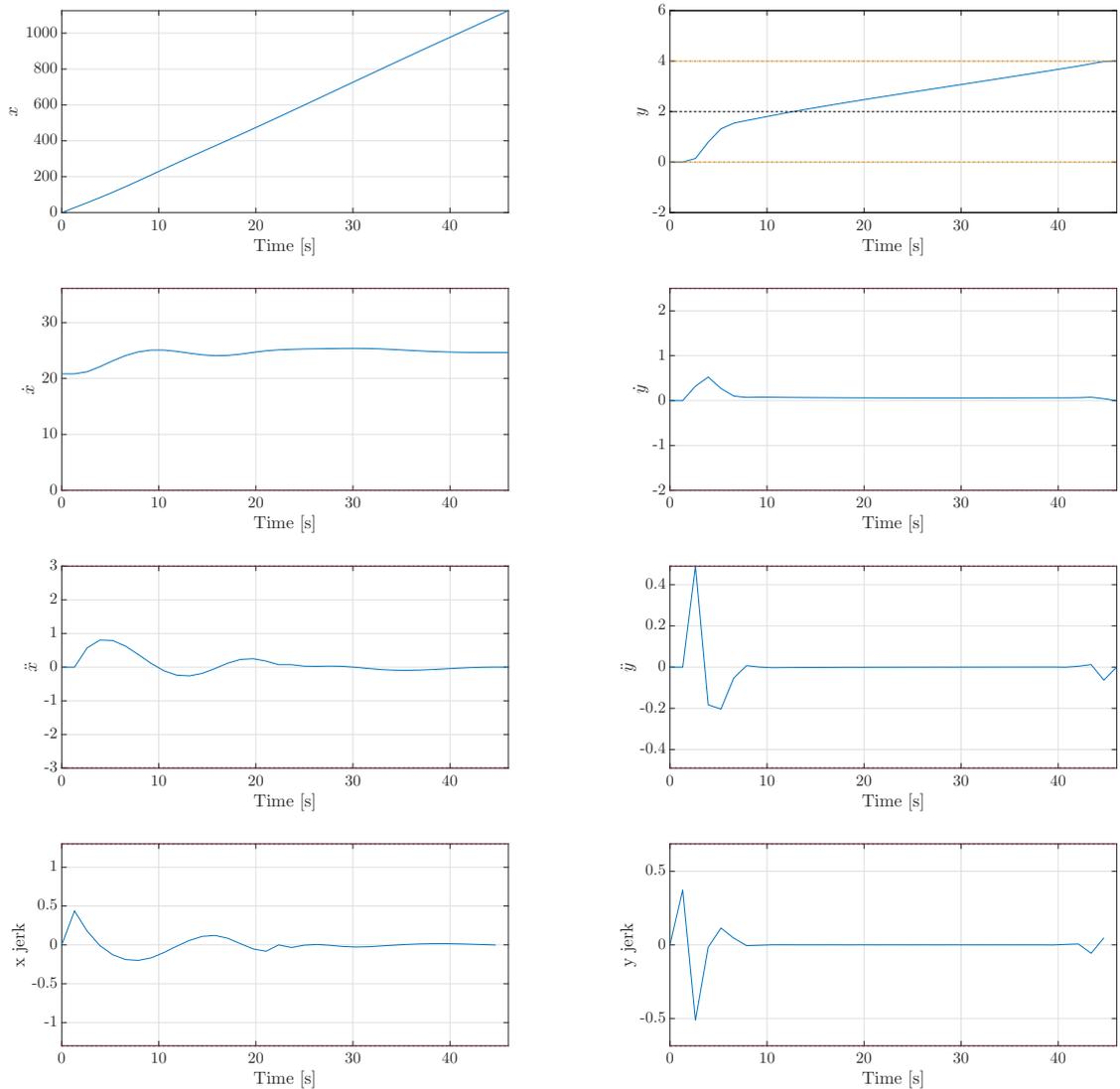


Figure 6.13 – Lane change generated trajectory for Scenario 2.3

Now, we can use the generated optimal trajectory as a base that ensures that we have a collision free trajectory. Now, in a second stage, we generate trajectories based on the flatness properties of model (6.3), using the first reference path as *waypoints* that will define the initial and final conditions of the subintervals.

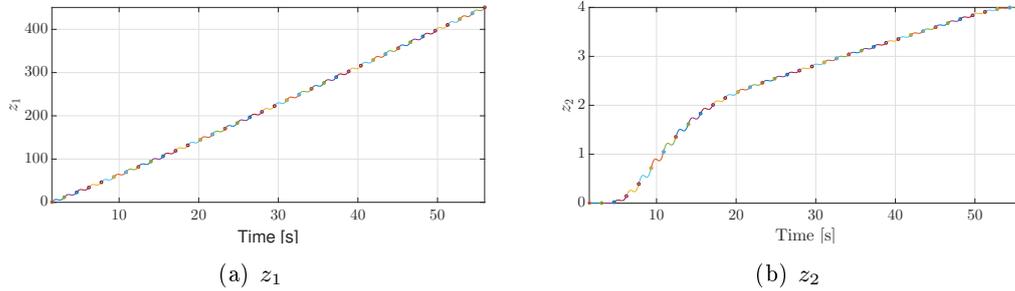


Figure 6.14 – Flatness-based generated trajectory with 5-th order polynomial parameterization in between *waypoints*

## 6.4 Problem computation reduction

The inclusion of the anticollision constraints by means of inequalities (6.15g) translates the original OCP into a mixed-integer problem formulation, due to the presence of the binary decision variable,  $\lambda_k \in \{0, 1\}^{N_0}$ , that defines which feasible convex region ( $\mathcal{F}_l$ ) is active along the discretized trajectory waypoints. It must be noted that this kind of problem is NP-hard, and has important computation drawbacks due to its combinatorial nature (Figure 6.17).

**Theorem 6.1** *Given a combinatorial search tree (Figure 6.17), any combination that belongs to a branch derived from an unfeasible combination node will also be unfeasible in all the sub-branches and can be directly suppressed from the exploration algorithm.*

*Proof:* Any combination derived from an unfeasible node will preserve the unfeasible sequence, and remains unfeasible independent of the choice of the successive values of the array. This is practically related to a collision in the early part of the trajectory and can be eliminated from the enumeration. ■

**Theorem 6.2** *Any branch with more than one switch on the binary variable is sub-optimal and can be pruned for the search tree.*

*Proof:* The switch on the binary variable value occurs when the trajectory changes from one convex feasible region  $\mathcal{C}_1$  with  $\lambda_k = 0$  to the another one  $\mathcal{C}_2$ , with  $\lambda_k = 1$ . Any trajectory that contains a second switch in practice involves going back to the first convex region  $\mathcal{C}_1$ , which mathematically is translated on an increase on  $\xi(4) = y$ .

Let us consider a constrained time-optimal motion planning, where the objective is to minimize the total time in which the maneuver is executed. The optimal solution of this kind of problem provides for a linear system is a bang-bang control input, with  $n - 1$  switches [Patil 2013], [Olech 1966], with  $n$  being the number of states. If we consider the lateral dynamics sub-model, with  $\tilde{\xi}_1(\tau) = [y(\tau), \dot{y}(\tau), \ddot{y}(\tau)]^T$  and

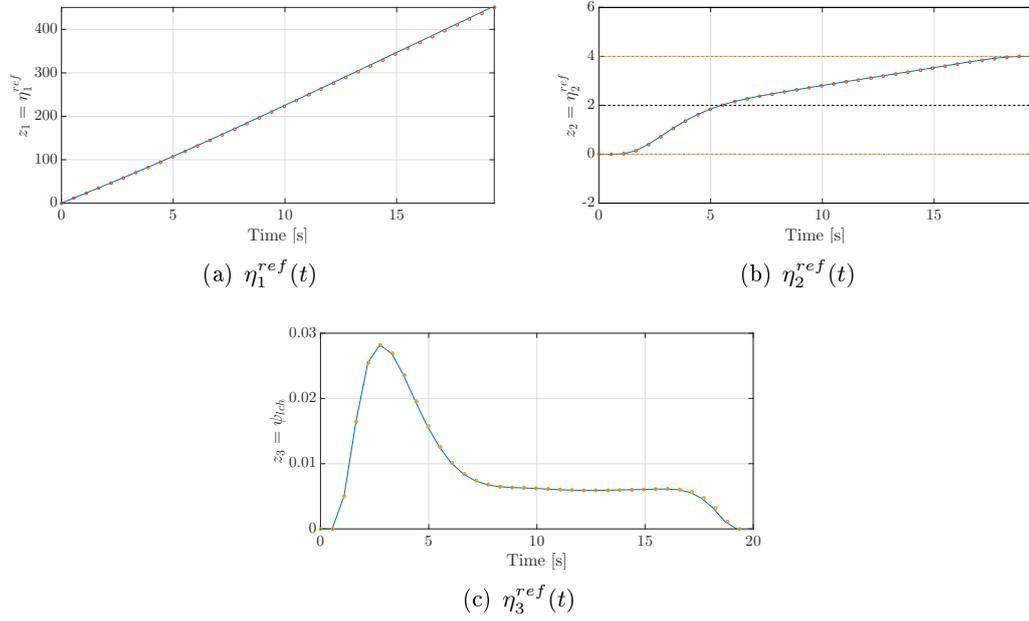


Figure 6.15 – Flatness-based generated trajectory with linear parameterization in between *waypoints*,  $\eta^{ref}(t)$

$\tilde{u}_1(\tau) = \ddot{y}(\tau)$  being the input, the time-optimal solution will have  $n - 1 = 2$  switches on the trajectory jerk, that is, the optimal control input is piecewise constant with at most two discontinuities. In a similar way, if now we consider a double integrator subsystem where acceleration is the control input,  $u_2(\tau) = \ddot{y}(\tau)$  and  $\tilde{\xi}_2(\tau) = [y(\tau), \dot{y}(\tau)]^T$ , we would obtain 1 switch on the acceleration trajectory. Finally, we consider  $u_3(\tau) = \dot{y}(\tau)$  and  $\tilde{\xi}_3(\tau) = y(\tau)$ , the optimal solution will have no switches at the control input and the state  $\tilde{\xi}_3(\tau) = y(\tau)$  trajectory is monotonic.

The inclusion of the control input on the performance index produces a trade-off between arrival time and comfort of the maneuver. This means that the optimal control input will not necessarily hit the constraints. Nevertheless, the monotonicity on the  $\xi(4) = y$  is still kept. ■

In order to reduce the computational burden of this kind of problems, an iterative approach can be introduced, where the sequence of values for the parameter  $\lambda_k$  at each node of the time gridding is defined before solving the constrained NLP (6.14, 6.15a-6.15h) and furthermore reduce the computational complexity. In this way, the integer variable is eliminated from the decision variables, it would still be necessary to solve  $2^K$  NLPs, one for each possible combinations of the  $\lambda_k$  sequence. Nevertheless, by means of Theorem 6.1, a simplified combinatorial enumeration can be obtained (Figure 6.18), where the branch elimination has been made in terms of the one-switching condition introduced by Theorem 6.2. In addition to this, an early-stopping criteria can be used from the unfeasibility guarantee condition, presented in Lemma 6.1, which allows to stop the computation once an unfeasible

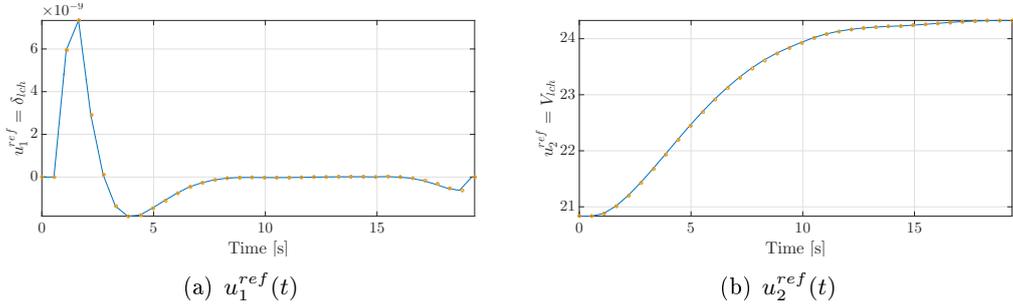


Figure 6.16 – Flatness-based generated inputs with linear parameterization in between waypoints,  $u^{ref}(t)$

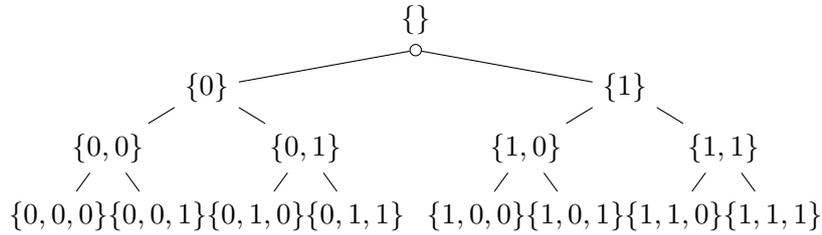


Figure 6.17 –  $\lambda_k$  combinatorial search tree

constrained NLP is obtained for a given  $\lambda_k$  sequence (Algorithm 2).

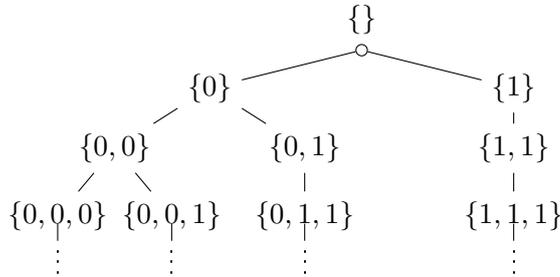


Figure 6.18 –  $\lambda_k$  simplified combinatorial search tree

Considering these simplifications, the maximal number of NLP that need to be solved is drastically reduced from  $2^K$  to  $K$  at the worst case scenario (changing from the exponential complexity to a polynomial and even linear complexity with respect to the number of binary variables), where all the  $\lambda$  sequences provide a feasible solution. Finally, robustness is guaranteed from the superposition of both convex feasible regions, ensuring that the state is contained in at least one of the feasible regions at all times  $t_k$ .

**Lemma 6.1** *Given an unfeasible trajectory provided by a branch  $\underbrace{\{0 \dots 0\}}_p, \underbrace{\{1 \dots 1\}}_m$ , any other trajectory obtained for a sequence with  $m - k$  ones is unfeasible too. In a similar way, given an unfeasible trajectory obtained from a sequence with  $p$  zeros, any other which contains  $p - k$  zeros is unfeasible.*

*Proof:* Let us consider a trajectory that starts at the first convex region,  $\mathcal{C}_1$  with  $\lambda_k = 0$ . A trajectory where the switch occurs at time  $k = p + 1$  being unfeasible means that the system cannot reach  $\mathcal{C}_2$ , in  $p + 1$  steps, due to system's constraints. If such trajectory is not possible, it is straightforward to notice that any other trajectory with an earlier switch at time  $k = p - i$  will violate the constraints in a similar way and be unfeasible too. Equivalently, an unfeasible trajectory that stays  $m$  steps in the second convex region is in practice a trajectory that does not get into the second convex region in time, and the final conditions cannot be reached without violating the constraints. This means that any trajectory that stays  $m - i$  steps will suffer from the same kind of unfeasibility. ■

## 6.5 Implementation remarks

### 6.5.1 Receding horizon strategy

Once a suitable trajectory that performs the lane change maneuver is generated at a given time  $t$ , it will be provided to the lower level controllers as the reference to track. Nevertheless, it has to be taken into account that the vehicle is driving a dynamic environment, so the conditions that are used to generate the trajectory at the first place can change along the effective time of the maneuver realization. Moreover, there will be unmodeled dynamics and disturbances that need to be considered. In the same way, the lane change maneuver could be aborted or interrupted due to numerous situations that are treated at the strategical level, producing the need of a new trajectory that would bring back the vehicle to the initial lane, for example. As a result of these factors in the decision making, a receding horizon strategy is used, recomputing the reference trajectory in an iterative way by means of Algorithm 2, providing replanned references every  $T_{straj}$  under normal driving circumstances or triggered by the strategical level if an exception arises.

### 6.5.2 Decision making algorithm

The decision making algorithm is a key component for the automated lane change and overtaking system. Initiating and aborting this kind of maneuver are inherently decisions of this logic. In this thesis work, no attempt is made to introduce an intelligent algorithm regarding this matter. Instead, some key points that this kind of algorithm needs to address are listed, but a much deeper study on the matter would be needed for a series implementation (and remain outside of the scope of the present work).

---

**Algorithm 2** Trajectory Generation Block (pseudocode)
 

---

```

1: Initialize:
2: Select scenario: define convex feasible regions  $\mathcal{C}_j$ 
3: Define final maneuver conditions  $\xi_{T_F}$ 
4: Update initial conditions  $\xi_0, k_{f_0}, \lambda = \{1\}^{1 \times N}$ .
5: Update  $T_{max} = T_{max_{ich}} - T_{elapsed}$ 
6: for  $t_k = 1 : K$  do
7:   for Each hyperplane  $i$  do
8:     Integrate hyperplane dynamics (6.13)
9:   end for
10:  if  $\bar{\xi}_0 \ni C_2$  then
11:     $\lambda = \{0, \lambda(1 : (N - 1))\}$ 
12:  end if
13:  Solve constrained NLP (6.14) subject to (6.15a - 6.15h)
14:  if Feasible trajectory found then
15:    Compute trajectory total cost  $J_j$  function (6.10)
16:    if  $J_j < J_{j-1}$  then
17:      Save new feasible trajectory
18:    else Keep previous feasible trajectory
19:    end if
20:  else Uneasible trajectory
21:    break (Lemma 6.1)
22:  end if
23: end for
24: Compute lower level controllers reference signals (6.6)
25: return  $\eta^{ref}(t), u^{ref}(t)$ 

```

---

- LANE CHANGE REQUEST. Flag activation through lateral indicator, activated by the driver. This will define the laterality of the lane change maneuver: *right-to-left* or *left-to-right*.
- DEFINE MANEUVER CHARACTERISTICS. Detect and process targets information, if any. Check and select scenario, kinematical constraints, convex feasible regions for anticollision constraints and final point of the maneuver. Decide if the maneuver is can take place.
- MANEUVER RE-DEFINITION. If environment changes are detected, change the previously stated maneuver characteristics.
- MANEUVER COMPUTATION. Following the receding horizon strategy, a new trajectory computation will be requested every  $T_{straj}$  seconds, or if an exception arises.
- HANDLING UNFEASIBILITY OR ABORT MANEUVER It can occur that the optimization problem is infeasible. Infeasibility can occur because of several reasons, here we can mention,
  - Initial or final state are out of trajectory planner constraints.
  - Final state is not reachable from the current position for the constrained problem case.
  - Anticollision constraints are unfeasible.
  - Target vehicles change their lane and scenario becomes unfeasible.

If such a situation arises, decision making algorithm is the one in charge to decide the strategy to follow. If the scenario has become unfeasible, a new target position may be defined, and a new trajectory that bring the vehicle back to the initial lane may be generated. Moreover, if unfeasibility arises from the optimization problem, an alternative plan can be applied. The solution algorithm generates a trajectory of length  $T_F$ . Hence, the last feasible solution of the NLP has a predicted optimal trajectory for several control intervals in the future. If the previously computed trajectory is stored and the optimization problem provides an error exit flag, we could take the most recent solution or use the previous optimal trajectory. Note that no guarantee exists that the system will recover from the infeasibility using this strategy, but it provides a tool that is considered a better alternative compared to the usage of the latest unfeasible solution.

- LANE CHANGE COMPLETION. Once the trajectory has been completed, activate LCA system to follow the center of the new lane until new lane change is requested.

## 6.6 Conclusion

A crucial component that is needed for automated maneuvers such as lane changing or overtaking, is the trajectory generation algorithm. For safe operation, the algorithm needs to be sufficiently sophisticated to capture the real-time complexity of the problem along the maneuver. However, available computational resources and measurements are constrained by the typically low cost hardware utilized in passenger cars.

An optimization-based trajectory generation strategy to perform lane change or overtaking maneuvers in multiple-lanes one-way roads in the presence of surrounding vehicles has been proposed. Overall, the whole control problem is decomposed in a hierarchical structure, where a high level planner provides a collision-free reference to the lower level tracking controllers. This chapter has been focused on the first layer, where the reference trajectory is computed by means of a NLP that is obtained from the formulated optimal control problem via a widely known direct optimization method, the multiple shooting approach. This optimization problem includes a simplified model of the vehicle dynamics and ensures that the computed collision-free trajectory maximizes passenger comfort and fulfills vehicle kinematic constraints. Once this trajectory is obtained, the differentially flat vehicle kinematic model is used to adapt the reference and provide suitable signals to the controllers.

Optimal control and hyperplane arrangements theoretical tools have been used to define the anticollision constraints that are to be used at the generation phase. To the best of our knowledge the present work is the first to provide an exhaustive description in terms of hyperplane arrangements for the non-convex feasible region in which the ego vehicle has to stay when performing the maneuver by means of the union of convex cells, defined by a set of polyhedra and a unique binary variable. Moreover, a thorough scenario description has been presented for the overtaking and lane changing in connection with optimal control problems. This enclosed framework has lead to an important simplification from the computational point of view, and the inherent complexity of the mixed-integer nature of the formulation has been attenuated.

# CONCLUSION AND FURTHER PERSPECTIVES



# Conclusion and further perspectives

---

Advanced Driving Assistance Systems (ADAS) has been the conducting axis of this industrial thesis work. This kind of systems are considered to be the initial steps towards the research and development of future autonomous driving vehicles, helping to progressively improve the technological knowledge and state-of-art of different driving tasks. This evolution will help to come up with control strategies that are feasible and applicable for real world products accessible for all customers. At the same time, efficiently make driving choices as a trade off between system performance and passenger safety will play a fundamental role for the future adopted solutions.

In the first part of this work, we have focused in one of the main driving tasks, which is the control of the steering wheel in order to follow another vehicle (Autosteer by target tracking) or the center of the current lane (Lane Centering Assistance). We have adopted state-of-art LPV control theory for this automotive application when the system dynamics are affected by broad vehicle speed variations and curved roads. It has been shown that this lower level controllers can efficiently rely on LPV control designs, providing feasibility guarantees even for large speed variations by means of an enhanced switching strategy or by taking into account the maximal acceleration capabilities of the vehicle.

On top of this formulation, Model Predictive Control is increasingly gaining importance in the industrial framework, due to its optimal performance and system together with system's and environmental constraints satisfaction guarantees. At the same time, computationally cheaper strategies like Interpolation Based Control are proven to be a promising alternative on the constrained optimization control theory. As further perspectives of this work, the modelization and control design studies in the presence of parameter-varying input constraints should definitely be considered. More particularly, the variation of the steering angle's limits with respect to the vehicle speed is a fundamental problem that should be addressed in the future for the case of steering control applications. In addition, coupled longitudinal and lateral dynamics can be considered, although this would provide more complex solutions based on nonlinear optimization that may pose complexity problems when considering the full ADAS structure.

The second part of this work has been focused on a maneuver-oriented development, where the need of generating a driving reference path comes into the picture. An extensive description of the possible scenarios that may arise when driving in one-way roads (i.e. highways) and its characterization for anti-collision constraints



are beginning to appreciate the advantages of these kind of systems, increasingly becoming more willing to pay their price.

Partly autonomous driving is becoming a reality within the next few generations of passenger cars, and even highly automated driving no longer appears completely out of reach. However, we must keep in mind that a suitable regulation framework must be developed and many arduous issues of homologation and liability have to be addressed. Finally, further research on situation assessment and improvement on the decision making algorithms that are capable of being in a par with human cognition capabilities is highly needed. Fundamental research in the upcoming years will play a vital role to bring these concepts to the real world.



# Bibliography

- [Ackermann 2012] Jürgen Ackermann. *Robust control: Systems with uncertain physical parameters*. Springer Science & Business Media, 2012. (Cited on page 71.)
- [Ahmadi-Moshkenani 2016] Parisa Ahmadi-Moshkenani, Sorin Olaru and Tor Arne Johansen. *Further results on the exploration of combinatorial tree in multi-parametric quadratic programming*. In Control Conference (ECC), 2016 European, pages 116–122. IEEE, 2016. (Cited on page 40.)
- [Alessio 2009] Alessandro Alessio and Alberto Bemporad. *A survey on explicit model predictive control*. In Nonlinear model predictive control, pages 345–369. Springer, 2009. (Cited on page 39.)
- [Avis 1996] David Avis and Komei Fukuda. *Reverse search for enumeration*. Discrete Applied Mathematics, vol. 65, no. 1-3, pages 21–46, 1996. (Cited on page 112.)
- [Behringer 2004] Reinhold Behringer, Sundar Sundareswaran, Brian Gregory, Richard Elsley, Bob Addison, Wayne Guthmiller, Robert Daily and David Bevy. *The DARPA grand challenge-development of an autonomous vehicle*. In Intelligent Vehicles Symposium, 2004 IEEE, pages 226–231. IEEE, 2004. (Cited on page 1.)
- [Bellman 1957] RICHARD Bellman. *Dynamic programming*. Princeton, USA: Princeton University Press, vol. 1, no. 2, page 3, 1957. (Cited on page 102.)
- [Bemporad 1999] Alberto Bemporad and Manfred Morari. *Robust model predictive control: A survey*. In Robustness in identification and control, pages 207–226. Springer, 1999. (Cited on pages 19 and 43.)
- [Bemporad 2002] Alberto Bemporad, Manfred Morari, Vivek Dua and Efstratios N Pistikopoulos. *The explicit linear quadratic regulator for constrained systems*. Automatica, vol. 38, no. 1, pages 3–20, 2002. (Cited on page 39.)
- [Bengler 2014] Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Christoph Stiller and Hermann Winner. *Three decades of driver assistance systems: Review and future perspectives*. IEEE Intelligent Transportation Systems Magazine, vol. 6, no. 4, pages 6–22, 2014. (Cited on page 2.)
- [Bertsekas 1995] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995. (Cited on page 102.)
- [Betts 1991] John T Betts and William P Huffman. *Trajectory optimization on a parallel processor*. Journal of Guidance, Control, and Dynamics, vol. 14, no. 2, pages 431–439, 1991. (Cited on page 108.)

- [Betts 2010] John T Betts. Practical methods for optimal control and estimation using nonlinear programming, volume 19. Siam, 2010. (Cited on pages 101 and 108.)
- [Blanchini 1996] Franco Blanchini and Stefano Miani. *On the transient estimate for linear systems with time-varying uncertain parameters*. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 43, no. 7, pages 592–596, 1996. (Cited on page 23.)
- [Blanchini 1999] Franco Blanchini. *Survey paper: Set invariance in control*. Automatica (Journal of IFAC), vol. 35, no. 11, pages 1747–1767, 1999. (Cited on page 20.)
- [Blanchini 2007] Franco Blanchini and Stefano Miani. Set-theoretic methods in control. Springer Science & Business Media, 2007. (Cited on page 20.)
- [Blanchini 2008] Franco Blanchini and Stefano Miani. Set-theoretic methods in control. Springer, 2008. (Cited on pages 20 and 30.)
- [Bock 1984] Hans Georg Bock and Karl-Josef Plitt. *A multiple shooting algorithm for direct solution of optimal control problems*. PROCEEDINGS OF THE IFAC WORLD CONGRESS, 1984. (Cited on page 107.)
- [Bock 2000] HG Bock, MM Diehl, DB Leineweber and JP Schlöder. *A direct multiple shooting method for real-time optimization of nonlinear DAE processes*. In Nonlinear Model Predictive Control, pages 245–267. Springer, 2000. (Cited on page 107.)
- [Bokor 2005] József Bokor and Gary Balas. *Linear parameter varying systems: A geometric theory and applications*. In 16th IFAC World Congress, Prague, volume 1, pages 1–11, 2005. (Cited on pages 5 and 19.)
- [Borrelli 2003] Francesco Borrelli. Constrained optimal control of linear and hybrid systems, volume 290. Springer, 2003. (Cited on page 17.)
- [Borrelli 2015] F. Borrelli, A. Bemporad and M. Morari. Predictive control for linear and hybrid systems. 2015. (Cited on pages 22 and 23.)
- [Boyd 1994] Stephen Boyd, Laurent El Ghaoui, Eric Feron and Venkataramanan Balakrishnan. Linear matrix inequalities in system and control theory. SIAM, 1994. (Cited on pages 28, 29 and 85.)
- [Brockett 1982] Roger W Brockett. *Control theory and singular Riemannian geometry*. In New directions in applied mathematics, pages 11–27. Springer, 1982. (Cited on page 111.)
- [Brookhuis 2001] Karel A Brookhuis, Dick De Waard and Wiel H Janssen. *Behavioural impacts of advanced driver assistance systems—an overview*. EJTIR, vol. 1, no. 3, pages 245–253, 2001. (Cited on page 3.)

- [Bryson 1975] Arthur Earl Bryson. Applied optimal control: optimization, estimation and control. CRC Press, 1975. (Cited on pages 101 and 103.)
- [Buck 1943] Robert Creighton Buck. *Partition of space*. The American Mathematical Monthly, vol. 50, no. 9, pages 541–544, 1943. (Cited on page 112.)
- [Butcher 2016] John Charles Butcher. Numerical methods for ordinary differential equations. John Wiley & Sons, 2016. (Cited on page 109.)
- [Camacho 2013] Eduardo F Camacho and Carlos Bordons Alba. Model predictive control. Springer Science & Business Media, 2013. (Cited on pages 39 and 41.)
- [Carvalho 2015] Ashwin Carvalho, Stéphanie Lefèvre, Georg Schildbach, Jason Kong and Francesco Borrelli. *Automated Driving: The Role of Forecasts and Uncertainty-A Control Perspective*. European Journal of Control, 2015. (Cited on pages 5 and 6.)
- [Chee 1994] Wonshik Chee and Masayoshi Tomizuka. *Vehicle lane change maneuver in automated highway systems*. California Partners for Advanced Transit and Highways (PATH), 1994. (Cited on page 127.)
- [Coelingh 2010] Erik Coelingh, Andreas Eidehall and Mattias Bengtsson. *Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking*. In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pages 155–160. IEEE, 2010. (Cited on page 2.)
- [Cuzzola 2002] Francesco A Cuzzola, Jose C Geromel and Manfred Morari. *An improved approach for constrained robust model predictive control*. Automatica, vol. 38, no. 7, pages 1183–1189, 2002. (Cited on page 43.)
- [Daafouz 2001] Jamal Daafouz and Jacques Bernussou. *Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties*. Systems & control letters, vol. 43, no. 5, pages 355–359, 2001. (Cited on pages 35 and 43.)
- [Dang 2012] Thao Dang, Jens Desens, Uwe Franke, Dariu Gavrilă, Lorenz Schäfers and Walter Ziegler. *Steering and evasion assist*. In Handbook of intelligent vehicles, pages 759–782. Springer, 2012. (Cited on page 2.)
- [Davins-Valldaura 2017] Joan Davins-Valldaura, Franck Plestan, Saïd Moussaoui and Guillermo Pita-Gil. *Design and Optimization of Nonlinear Observers for Road Curvature and State Estimation in Automated Vehicles*. IEEE Transactions on Intelligent Transportation Systems, 2017. (Cited on pages 38 and 54.)

- [de Oliveira 1999] Maurício C de Oliveira, Jacques Bernussou and José C Geromel. *A new discrete-time robust stability condition*. Systems & control letters, vol. 37, no. 4, pages 261–265, 1999. (Cited on page 33.)
- [der Automobilindustrie eV 2015] Verband der Automobilindustrie eV. *Automation: From Driver Assistance Systems to Automated Driving*. VDA Magazine-Automation, 2015. (Cited on page 2.)
- [Di Cairano 2013] Stefano Di Cairano, H Eric Tseng, Daniele Bernardini and Alberto Bemporad. *Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain*. Control Systems Technology, IEEE Transactions on, vol. 21, no. 4, pages 1236–1248, 2013. (Cited on pages 7 and 50.)
- [Di Cairano 2015] Stefano Di Cairano. *Model adjustable predictive control with stability guarantees*. In American Control Conference (ACC), 2015, pages 226–231. IEEE, 2015. (Cited on pages 43 and 44.)
- [Diehl 2005] Moritz Diehl, Hans Georg Bock and Johannes P Schlöder. *A real-time iteration scheme for nonlinear optimization in optimal feedback control*. SIAM Journal on control and optimization, vol. 43, no. 5, pages 1714–1736, 2005. (Cited on pages 127 and 140.)
- [Diehl 2006] Moritz Diehl, Hans Georg Bock, Holger Diedam and P-B Wieber. *Fast direct multiple shooting algorithms for optimal robot control*. In Fast motions in biomechanics and robotics, pages 65–93. Springer, 2006. (Cited on page 140.)
- [Diehl 2014] Moritz Diehl. *Lecture notes on Optimal Control and Estimation*, 2014. (Cited on page 101.)
- [Dorea 1999] Carlos Eduardo Trabuco Dorea and JC Hennes. *(A, B)-invariant polyhedral sets of linear discrete-time systems*. Journal of optimization theory and applications, vol. 103, no. 3, pages 521–542, 1999. (Cited on page 63.)
- [Doumiati 2013] Moustapha Doumiati, Olivier Sename, Luc Dugard, John-Jairo Martinez-Molina, Peter Gaspar and Zoltan Szabo. *Integrated vehicle dynamics control via coordination of active front steering and rear braking*. European Journal of Control, vol. 19, no. 2, pages 121–143, 2013. (Cited on page 50.)
- [Enache 2008] Nicoleta Minoiu Enache. *Assistance préventive à la sortie de voie*. PhD thesis, Université d’Evry-Val d’Essonne, 2008. (Cited on pages 2, 49 and 71.)
- [Falcone 2006] Paolo Falcone, Francesco Borrelli, Jahan Asgari, H Eric Tseng and Davor Hrovat. *A real-time model predictive control approach for autonomous*

- active steering*. Nonlinear Model Predictive Control for Fast Systems, Grenoble, France, 2006. (Cited on page 7.)
- [Falcone 2007] Paolo Falcone, Francesco Borrelli, Jahan Asgari, Hongtei Eric Tseng and Davor Hrovat. *Predictive active steering control for autonomous vehicle systems*. Control Systems Technology, IEEE Transactions on, vol. 15, no. 3, pages 566–580, 2007. (Cited on pages 3 and 6.)
- [Falcone 2008] Paolo Falcone, Francesco Borrelli, H Eric Tseng, Jahan Asgari and Davor Hrovat. *A hierarchical model predictive control framework for autonomous ground vehicles*. In American Control Conference, 2008, pages 3719–3724. IEEE, 2008. (Cited on page 6.)
- [Fergani 2013] Soheib Fergani, Lghani Menhour, Olivier Sename, Luc Dugard and BD’Andrea Novel. *Study and comparison of non linear and LPV control approaches for vehicle stability control*. In Control & Automation (MED), 2013 21st Mediterranean Conference on, pages 303–310. IEEE, 2013. (Cited on page 5.)
- [Fliess 1992] Michel Fliess, Jean Lévine, Philippe Martin and Pierre Rouchon. *On differentially flat nonlinear systems*. In IFAC SYMPOSIA SERIES, pages 159–163, 1992. (Cited on page 110.)
- [Fliess 1995] Michel Fliess, Jean Lévine, Philippe Martin and Pierre Rouchon. *Flatness and defect of non-linear systems: introductory theory and examples*. International journal of control, vol. 61, no. 6, pages 1327–1361, 1995. (Cited on page 111.)
- [Frasch 2013] Janick V Frasch, Andrew Gray, Mario Zanon, Hans Joachim Ferreau, Sebastian Sager, Francesco Borrelli and Moritz Diehl. *An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles*. In Control Conference (ECC), 2013 European, pages 4136–4141. IEEE, 2013. (Cited on pages 6 and 128.)
- [Gao 2010] Yiqi Gao, Theresa Lin, Francesco Borrelli, Eric Tseng and Davor Hrovat. *Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads*. In ASME 2010 dynamic systems and control conference, pages 265–272. American Society of Mechanical Engineers, 2010. (Cited on pages 5 and 128.)
- [Gao 2012] Yiqi Gao, Andrew Gray, Janick V Frasch, Theresa Lin, Eric Tseng, J Karl Hedrick and Francesco Borrelli. *Spatial predictive control for agile semi-autonomous ground vehicles*. In Proceedings of the 11th international symposium on advanced vehicle control, 2012. (Cited on page 128.)
- [Geyer 2004] Tobias Geyer, Fabio D Torrioni and Manfred Morari. *Optimal complexity reduction of piecewise affine models based on hyperplane arrangements*.

- In American Control Conference, 2004. Proceedings of the 2004, volume 2, pages 1190–1195. IEEE, 2004. (Cited on page 115.)
- [Geyer 2010] Tobias Geyer, Fabio D Torrisi and Manfred Morari. *Efficient mode enumeration of compositional hybrid systems*. International Journal of Control, vol. 83, no. 2, pages 313–329, 2010. (Cited on page 112.)
- [Gilbert 1991] Elmer G Gilbert and Kok Tin Tan. *Linear systems with state and control constraints: The theory and application of maximal output admissible sets*. Automatic Control, IEEE Transactions on, vol. 36, no. 9, pages 1008–1020, 1991. (Cited on pages 22 and 23.)
- [Goodwin 2006] Graham Goodwin, María M Seron and José A De Doná. *Constrained control and estimation: an optimisation approach*. Springer Science & Business Media, 2006. (Cited on page 17.)
- [Hanke 2001] Oliver Hanke, Torsten Bertram and Manfred Hiller. *Analysis and control of vehicle dynamics under crosswind conditions*. In Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on, volume 1, pages 331–336. IEEE, 2001. (Cited on page 67.)
- [Herceg 2013] Martin Herceg, Michal Kvasnica, Colin N Jones and Manfred Morari. *Multi-parametric toolbox 3.0*. In Control Conference (ECC), 2013 European, pages 502–510. IEEE, 2013. (Cited on page 28.)
- [Hindi 2004] Haitham Hindi. *A tutorial on convex optimization*. In In Proceedings of the 23th American Control Conference, pages 3252–3265. IEEE, 2004. (Cited on page 30.)
- [Hovd 2014] Morten Hovd, Sorin Olaru and George Bitsoris. *Low Complexity constraint control using contractive sets*. IFAC Proceedings Volumes, vol. 47, no. 3, pages 2933–2938, 2014. (Cited on page 63.)
- [Hrovat 2012] Davor Hrovat, Stefano Di Cairano, H Eric Tseng and Ilya V Kolmanovskiy. *The development of model predictive control in automotive industry: A survey*. In Control Applications (CCA), 2012 IEEE International Conference on, pages 295–302. IEEE, 2012. (Cited on page 39.)
- [Hull 2013] David G. Hull. *Optimal control theory for applications*. Springer Science & Business Media, 2013. (Cited on pages 101 and 103.)
- [Hwan 2011] Jeon Jeong Hwan, Sertac Karaman and Emilio Frazzoli. *Anytime computation of time-optimal off-road vehicle maneuvers using the RRT*. In Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pages 3276–3282. IEEE, 2011. (Cited on page 127.)
- [Janeček 2017] Filip Janeček, Martin Klaučo, Martin Kalúz and Michal Kvasnica. *OPTIPLAN: A Matlab Toolbox for Model Predictive Control with Obstacle*

- Avoidance*. IFAC-PapersOnLine, vol. 50, no. 1, pages 531–536, 2017. (Cited on page 129.)
- [Jazar 2013] Reza N Jazar. *Vehicle dynamics: theory and application*. Springer Science & Business Media, 2013. (Cited on page 5.)
- [J.M.Maciejowski 2002] J.M.Maciejowski. *Predictive control with constraints*. Prentice Hall, 2002. (Cited on pages 39 and 94.)
- [Kirk 2012] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012. (Cited on pages 101 and 103.)
- [Koren 1991] Yoram Koren and Johann Borenstein. *Potential field methods and their inherent limitations for mobile robot navigation*. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1398–1404. IEEE, 1991. (Cited on page 128.)
- [Kothare 1996] Mayuresh V Kothare, Venkataramanan Balakrishnan and Manfred Morari. *Robust constrained model predictive control using linear matrix inequalities*. *Automatica*, vol. 32, no. 10, pages 1361–1379, 1996. (Cited on pages 19 and 43.)
- [Kusano 2012] Kristofer D Kusano and Hampton C Gabler. *Safety benefits of forward collision warning, brake assist, and autonomous braking systems in rear-end collisions*. *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pages 1546–1555, 2012. (Cited on page 2.)
- [Kvasnica 2015] Michal Kvasnica, Bálint Takács, Juraĵ Holaza and Deepak Ingole. *Reachability Analysis and Control Synthesis for Uncertain Linear Systems in MPT*. IFAC-PapersOnLine, vol. 48, no. 14, pages 302–307, 2015. (Cited on page 26.)
- [Labit 2002] Y Labit, Dimitri Peaucelle and D Henrion. *SeDuMi interface 1.02: a tool for solving LMI problems with SeDuMi*. In *Computer Aided Control System Design, 2002. Proceedings. 2002 IEEE International Symposium on*, pages 272–277. IEEE, 2002. (Cited on page 59.)
- [Laraba 2016] Mohammed-Tahar Laraba, Morten Hovd, Sorin Olaru and Silviu-Iulian Niculescu. *A bilevel optimization approach for D-invariant set design*. IFAC-PapersOnLine, vol. 49, no. 10, pages 235–240, 2016. (Cited on page 63.)
- [Lee 2008] Jaehoon Lee, Jonghyun Lee and Seung-Jin Heo. *Full vehicle dynamic modeling for chassis controls*. In *Proceedings of the 9th International Federation of Automotive Engineering Societies (FISITA) Student Congress*. Citeseer, 2008. (Cited on page 5.)

- [Liu 2010] Li Liu, Shuming Shi, Shuiwen Shen and Jiangwei Chu. *Vehicle planar motion stability study for tyres working in extremely nonlinear region*. Chinese journal of mechanical engineering, no. 2, page 185, 2010. (Cited on page 7.)
- [Lofberg 2005] Johan Lofberg. *YALMIP: A toolbox for modeling and optimization in MATLAB*. In Computer Aided Control Systems Design, 2004 IEEE International Symposium on, pages 284–289. IEEE, 2005. (Cited on pages 28, 59, 146 and 148.)
- [Lombardi 2009] Warody Lombardi, Sorin Olaru and Silviu Iulian Niculescu. *Invariant sets for a class of linear systems with variable time-delay*. In Control Conference (ECC), 2009 European, pages 4757–4762. IEEE, 2009. (Cited on page 68.)
- [Lu 2004] Bei Lu and Fen Wu. *Switching LPV control designs using multiple parameter-dependent Lyapunov functions*. Automatica, vol. 40, no. 11, pages 1973–1980, 2004. (Cited on page 84.)
- [Lu 2006] Bei Lu, Fen Wu and SungWan Kim. *Switching LPV control of an F-16 aircraft via controller state reset*. IEEE transactions on control systems technology, vol. 14, no. 2, pages 267–277, 2006. (Cited on page 84.)
- [Luca 2011a] Anamaria Luca. *Outils ensemblistes d’analyse et de synthèse des lois de commande robustes pour des systèmes incertains*. PhD thesis, Supélec, 2011. (Cited on page 90.)
- [Luca 2011b] Anamaria Luca, Pedro Rodriguez-Ayerbe and Didier Dumur. *Control of disturbed LPV systems in a LMI setting*. IFAC Proceedings Volumes, vol. 44, no. 1, pages 4149–4154, 2011. (Cited on pages 32, 38 and 74.)
- [Mahalanobis 1936] Prasanta Chandra Mahalanobis. *On the generalized distance in statistics*. Proceedings of the National Institute of Sciences (Calcutta), vol. 2, pages 49–55, 1936. (Cited on page 132.)
- [Mao 2003] Wei-Jie Mao. *Robust stabilization of uncertain time-varying discrete systems and comments on an improved approach for constrained robust model predictive control*. Automatica, vol. 39, no. 6, pages 1109–1112, 2003. (Cited on page 43.)
- [Mayne 1997] David Q Mayne and WR Schroeder. *Robust time-optimal control of constrained linear systems*. Automatica, vol. 33, no. 12, pages 2103–2118, 1997. (Cited on page 24.)
- [Mayne 2000] David Q Mayne, James B Rawlings, Christopher V Rao and Pierre OM Scokaert. *Constrained model predictive control: Stability and optimality*. Automatica, vol. 36, no. 6, pages 789–814, 2000. (Cited on page 43.)

- [Mercy 2014] Tim Mercy, Wannes Van Loock, Goele Pipeleers and Jan Swevers. *Time-optimal motion planning in the presence of moving obstacles*. 2014. (Cited on pages 127 and 128.)
- [Michon 1985] John A Michon. *A critical view of driver behavior models: what do we know, what should we do?* In Human behavior and traffic safety, pages 485–524. Springer, 1985. (Cited on page 3.)
- [Milam 2003] Mark Bradley Milam. Real-time optimal trajectory generation for constrained dynamical systems. California Institute of Technology, 2003. (Cited on page 128.)
- [Mohammadpour 2012] Javad Mohammadpour and Carsten W Scherer. Control of linear parameter varying systems with applications. Springer Science & Business Media, 2012. (Cited on pages 18 and 19.)
- [Muller 2007] Bernhard Muller, Joachim Deutscher and Stefan Grodde. *Continuous curvature trajectory design and feedforward control for parking a car*. IEEE Transactions on Control Systems Technology, vol. 15, no. 3, pages 541–553, 2007. (Cited on page 6.)
- [Munir 2016] Sarmad Munir, Morten Hovd, Guillaume Sandou and Sorin Olaru. *Controlled contractive sets for low-complexity constrained control*. In Computer Aided Control System Design (CACSD), 2016 IEEE Conference on, pages 856–861. IEEE, 2016. (Cited on pages 40 and 63.)
- [Murray 2009] Richard M Murray. *Optimization-based control*. California Institute of Technology, CA, 2009. (Cited on pages 111, 128 and 134.)
- [Nguyen 2011] Nam Nguyen, Sorin Olaru and Florin Stoican. *On maximal robustly positively invariant sets*. In 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO), pages 6–pages, 2011. (Cited on page 24.)
- [Nguyen 2013] H-N Nguyen, Sorin Olaru and Per-Olof Gutman. *Control synthesis for switched systems with control and state constraints*. In Control & Automation (MED), 2013 21st Mediterranean Conference on, pages 225–230. IEEE, 2013. (Cited on pages 45, 47 and 48.)
- [Nguyen 2014] Hoai-Nam Nguyen. *Constrained control of uncertain, time-varying, discrete-time systems*. Lecture Notes in Control and Information Sciences, vol. 451, 2014. (Cited on pages 17, 32, 44 and 63.)
- [Nguyen 2015] Hoai-Nam Nguyen, Sorin Olaru, Per Olof Gutman and Morten Hovd. *Constrained control of uncertain, time-varying linear discrete-time systems subject to bounded disturbances*. IEEE Transactions on Automatic Control, vol. 60, no. 3, pages 831–836, 2015. (Cited on page 37.)

- [Olaru 2008] Sorin Olaru and S-I Niculescu. *Predictive control for linear systems with delayed input subject to constraints*. IFAC Proceedings Volumes, vol. 41, no. 2, pages 11208–11213, 2008. (Cited on page 23.)
- [Olaru 2010] Sorin Olaru, JA De Doná, MM Seron and Florin Stoican. *Positive invariant sets for fault tolerant multisensor control schemes*. International Journal of Control, vol. 83, no. 12, pages 2622–2640, 2010. (Cited on page 24.)
- [Olech 1966] Czesław Olech. *Extremal solutions of a control system*. Journal of Differential Equations, vol. 2, no. 1, pages 74–101, 1966. (Cited on page 150.)
- [Pacejka 2005] Hans Pacejka. Tire and vehicle dynamics. Elsevier, 2005. (Cited on page 7.)
- [Palladino 2006] Luca Palladino. *Analyse comparative de différentes lois de commande en vue du contrôle global du châssis*. PhD thesis, Université Paris Sud-Paris XI, 2006. (Cited on page 3.)
- [Pannocchia 2001] Gabriele Pannocchia and James B Rawlings. *The velocity algorithm LQR: a survey*. 2001. (Cited on page 52.)
- [Patil 2013] Deepak U Patil, Ameer K Mulla, Debraj Chakraborty and Harish Pillai. *Gröbner basis computation of feedback control for time optimal state transfer*. In Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, pages 3243–3248. IEEE, 2013. (Cited on page 150.)
- [Pepy 2006] Romain Pepy, Alain Lambert and Hugues Mounier. *Path planning using a dynamic vehicle model*. In Information and Communication Technologies, 2006. ICTTA'06. 2nd, volume 1, pages 781–786. IEEE, 2006. (Cited on page 127.)
- [Pietz 2003] Jesse Allen Pietz. *Pseudospectral collocation methods for the direct transcription of optimal control problems*. PhD thesis, Rice University, 2003. (Cited on page 106.)
- [Pontryagin 1987] Lev Semenovich Pontryagin. Mathematical Theory of Optimal Processes. CRC Press, 1987. (Cited on page 103.)
- [Prodan 2012] Ionela Prodan, Florin Stoican, Sorin Olaru and Silviu-Iulian Niculescu. *Enhancements on the hyperplanes arrangements in mixed-integer programming techniques*. Journal of Optimization Theory and Applications, vol. 154, no. 2, pages 549–572, 2012. (Cited on page 115.)
- [Prodan 2015] Ionela Prodan, Florin Stoican, Sorin Olaru and Silviu-Iulian Niculescu. Mixed-integer representations in control design: Mathematical foundations and applications. Springer, 2015. (Cited on pages 101, 115, 116 and 118.)

- [Qin 2003] S Joe Qin and Thomas A Badgwell. *A survey of industrial model predictive control technology*. Control engineering practice, vol. 11, no. 7, pages 733–764, 2003. (Cited on page 17.)
- [Quirynen 2017] Rien Quirynen. *Numerical Simulation Methods for Embedded Optimization*. PhD thesis, KU Leuven, 2017. (Cited on page 108.)
- [Rajamani 2006] Rajesh Rajamani. Vehicle dynamics and control, chapter 2 Lateral Vehicle Dynamics. Springer, 2006. (Cited on pages 3 and 6.)
- [Rakovic 2005] Sasa V Rakovic, Eric C Kerrigan, Konstantinos I Kouramas and David Q Mayne. *Invariant approximations of the minimal robust positively invariant set*. IEEE Transactions on Automatic Control, vol. 50, no. 3, pages 406–410, 2005. (Cited on page 24.)
- [Rathgeber 2015] Christian Rathgeber, Franz Winkler, Xiaoyu Kang and Steffen Müller. *Optimal Trajectories for Highly Automated Driving*. World Academy of Science, Engineering and Technology, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, vol. 9, no. 6, pages 969–975, 2015. (Cited on page 127.)
- [Rawlings 2008] James B Rawlings and DQ Mayne. *Model predictive control*. In The 2008 Spring National Meeting, 2008. (Cited on page 39.)
- [Reimpell 2001] Jornsen Reimpell, Helmut Stoll and Jurgen Betzler. The automotive chassis: engineering principles. Butterworth-Heinemann, 2001. (Cited on page 70.)
- [Schneider 2014] Rolf Schneider. Convex bodies: the brunn–minkowski theory. Number 151. Cambridge university press, 2014. (Cited on page 21.)
- [Schubert 2008] Robin Schubert, Eric Richter and Gerd Wanielik. *Comparison and evaluation of advanced motion models for vehicle tracking*. In Information Fusion, 2008 11th International Conference on, pages 1–6. IEEE, 2008. (Cited on page 6.)
- [Schubert 2010] Robin Schubert, Karsten Schulze and Gerd Wanielik. *Situation assessment for automatic lane-change maneuvers*. IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 3, pages 607–616, 2010. (Cited on page 3.)
- [Sename 2013] Olivier Sename, Peter Gaspar and József Bokor. Robust control and linear parameter varying approaches: application to vehicle dynamics, volume 437. Springer, 2013. (Cited on pages 5 and 80.)
- [Sira-Ramirez 2004] Hebertt Sira-Ramirez and Sunil K Agrawal. Differentially flat systems. CRC Press, 2004. (Cited on pages 111 and 134.)

- [Stoican 2011a] Florin Stoican, Sorin Olaru, José A De Doná and Maria M Seron. *Zonotopic ultimate bounds for linear systems with bounded disturbances*. In Proceedings of the 18th IFAC World Congress, volume 28, pages 9224–9229, 2011. (Cited on page 128.)
- [Stoican 2011b] Florin Stoican, Ionela Prodan and Sorin Olaru. *Enhancements on the hyperplane arrangements in mixed integer techniques*. In Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pages 3986–3991. IEEE, 2011. (Cited on page 119.)
- [Tóth 2007] Roland Tóth, Federico Felici, Peter SC Heuberger and Paul MJ Van den Hof. *Discrete time LPV I/O and state space representations, differences of behavior and pitfalls of interpolation*. In Control Conference (ECC), 2007 European, pages 5418–5425. IEEE, 2007. (Cited on page 80.)
- [Treat 1979] John R Treat, NS Tumbas, ST McDonald, D Shinar, Rex D Hume, RE Mayer, RL Stansifer and NJ Castellan. *Tri-level study of the causes of traffic accidents: final report. Executive summary*. 1979. (Cited on page 2.)
- [Trucks 2013] Volvo Trucks. *EUROPEAN ACCIDENT RESEARCH AND SAFETY REPORT 2MI3*. 2013. (Cited on page 2.)
- [Vahidi 2003] Ardalan Vahidi and Azim Eskandarian. *Research advances in intelligent collision avoidance and adaptive cruise control*. IEEE transactions on intelligent transportation systems, vol. 4, no. 3, pages 143–153, 2003. (Cited on page 2.)
- [Vertet 2006] Martine Vertet and Sylvain Giauxserand. *Comprendre les principaux paramètres de conception géométrique des routes*. Sétra, 2006. (Cited on pages xix and 69.)
- [Vielma 2008] Juan Pablo Vielma and George L Nemhauser. *Modeling disjunctive constraints with a logarithmic number of binary variables and constraints*. In International Conference on Integer Programming and Combinatorial Optimization, pages 199–213. Springer, 2008. (Cited on page 118.)
- [Vorobieva 2013] Hélène Vorobieva, Nicoleta Minoiu-Enache, Sébastien Glaser and Saïd Mammar. *Geometric continuous-curvature path planning for automatic parallel parking*. In Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on, pages 418–423. IEEE, 2013. (Cited on page 127.)
- [Wächter 2006] Andreas Wächter and Lorenz T Biegler. *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*. Mathematical programming, vol. 106, no. 1, pages 25–57, 2006. (Cited on page 148.)

- [Wada 2004] Nobutaka Wada, Koji Saito and Masami Saeki. *Model predictive control for linear parameter varying systems using parameter dependent Lyapunov function*. In Circuits and Systems, 2004. MWSCAS'04. The 2004 47th Midwest Symposium on, volume 3, pages iii–133. IEEE, 2004. (Cited on page 43.)
- [Wan 2003] Zhaoyang Wan and Mayuresh V Kothare. *An efficient off-line formulation of robust model predictive control using linear matrix inequalities*. Automatica, vol. 39, no. 5, pages 837–846, 2003. (Cited on page 43.)
- [Weiss 2017] Avishai Weiss, Claus Danielson, Karl Berntorp, Ilya Kolmanovsky and Stefano Di Cairano. *Motion planning with invariant set trees*. In Control Technology and Applications (CCTA), 2017 IEEE Conference on, pages 1625–1630. IEEE, 2017. (Cited on page 127.)
- [Winner 2014] Hermann Winner and Michael Schopper. *Adaptive Cruise Control*. Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort, pages 1–44, 2014. (Cited on page 3.)
- [Winner 2015] Hermann Winner, Stephan Hakuli, Felix Lotz and Christina Singer. Handbook of driver assistance systems: Basic information, components and systems for active safety and comfort. Springer Publishing Company, Incorporated, 2015. (Cited on page 2.)
- [Ziegler 2012] Günter M Ziegler. Lectures on polytopes, volume 152. Springer Science & Business Media, 2012. (Cited on pages 111 and 128.)

**Title:** Model Predictive Control for the Autonomous Vehicle

**Keywords:** Model Predictive Control, Lateral Dynamics Control, Trajectory Planning, ADAS.

**Abstract:** The thesis work contained in this manuscript is dedicated to the Advanced Driving Assistance Systems, which has become nowadays a strategic research line in many car companies. This kind of systems can be seen as a first generation of assisted or semi-autonomous driving, that will set the way to fully automated vehicles.

The first part focuses on the analysis and control of lateral dynamics control applications - Autosteer by target tracking and the Lane Centering Assistance System (LCA). In this framework, safety plays a key role, bringing into focus the application of different constrained control techniques for linear parameter-varying (LPV) models. Model Predictive Control (MPC) and Interpolation Based Control (IBC) have been the selected ones in the present work.

In addition, it is a critical feature to design robust control systems that ensure a correct behavior under system's variation of parameters or in the presence of uncertainty. Robust Positive Invariance (RPI) theory tools are considered to design robust LPV control strategies with respect to large vehicle speed variations and curvature of the road changes.

The second axis of this thesis is the optimization-based trajectory planning for overtaking and lane change in highways with anti-collision enhancements. To achieve this goal, an exhaustive description of the possible scenarios that may arise is presented, allowing to formulate an optimization problem which maximizes passenger comfort and ensures system constraints' satisfaction.

**Acknowledgement:** The work leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 607957 (TEMPO).

**Titre:** Commande Prédicative pour le Véhicule Autonome

**Mots Clés :** Commande Prédicative, Contrôle de la Dynamique Latérale, Planification de Trajectoire, ADAS.

**Résumé:** Le travail de thèse décrit dans ce manuscrit concerne les Systèmes Avancés d'Aide à la Conduite (ADAS) qui sont devenus de nos jours un axe de recherche stratégique chez de nombreux constructeurs automobiles. Ce type de systèmes peuvent être considérés comme la première génération de dispositifs de conduite assistée ou semi-autonome et qui ouvrira la voie aux véhicules pleinement autonomes. La première partie de ce manuscrit concerne l'analyse et la commande pour les applications de contrôle de la dynamique latérale du véhicule – autoguidage par suivi de cible et aide au maintien au centre de la voie (LCA). Dans ce cadre, la sécurité joue un rôle clé, mettant en lumière la mise en œuvre différentes techniques de commande contrainte pour des modèles linéaires à paramètres variants (LPV). La commande prédictive (MPC) et la commande par interpolation (IBC) ont été sélectionnés dans ce travail.

De plus, la conception d'un système de commande robuste qui assure un comportement correct malgré la variation des paramètres du système ou la présence d'incertitudes est une caractéristique critique. Les outils de la théorie de l'invariance positive robuste (RPI) sont pris en considération pour la conception de stratégies de commande robustes LPV par rapport aux larges variations de la vitesse véhicule et aux changements de courbure de la route. Le second axe de cette thèse est la planification optimale de trajectoire pour les manœuvres de dépassement et de changement de voie sur autoroute, avec réduction des risques de collision. Pour atteindre cet objectif, la description exhaustive des scénarios possible est présentée, permettant de formuler un problème d'optimisation qui maximise le confort du conducteur et assure la satisfaction des contraintes du système.

**Remerciements :** Le travail qui a abouti à ces résultats a reçu le financement du People Programme (Marie Curie Actions), Septième programme-cadre de la Communauté européenne (FP7/2007-2013) sous la convention de subvention REA no 607957 (TEMPO).



