Planification et exécution de mouvements pour un robot bi-guidable: une approche basée sur la platitude différentielle

Jorje Hermosillo

▶ **To cite this version:**

Jorje Hermosillo. Planification et exécution de mouvements pour un robot bi-guidable: une approche basée sur la platitude différentielle. Autre [cs.OH]. Institut National Polytechnique de Grenoble - INPG, 2003. Français. NNT : . tel-00147377

HAL Id: tel-00147377

https://theses.hal.science/tel-00147377

Submitted on 16 May 2007

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.

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

No. attribué par la bibliothèque

THESE

Pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité: Imagerie, Vision, Robotique

préparée au Laboratoire GRAphique, VIsion, Robotique à l'INRIA Rhône-Alpes dans le cadre de l'Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information

---

présentée et soutenue par

Jorge HERMOSILLO VALADEZ

le 23 juin 2003

Titre:

Motion Planning & Feedback Control of Bi-steerable Robots: an Approach Based on Differential Flatness

Planification de Mouvements et Commande de Robots Bi-guidables : Une Approche Basée sur la Platitude Différentielle

Directeur de thèse : Christian Laugier

Jury

Président	Augustin Lux
Rapporteurs	Jean-Paul Laumond Dominique Meizel
Examinateurs	Sepanta Sekhavat (Co-encadrant) Christian Laugier Bernard Espiau Vincent Dupourqué

A ma chère compagne Alicia et
nos plus belles créations : Amaury, Danilo et Araí...

Remerciements

Cette thèse a été financée par le Conseil National de Science et Technologie du Mexique (CONACYT) et la Société Française d'Exportation de Ressources (SFERE). Toute ma reconnaissance à leur support et confiance.

Durant ma formation au sein de l'équipe SHARP de l'INRIA Rhône-Alpes, nombreux sont ceux et celles qui m'ont soutenu et encouragé, à qui je souhaite exprimer ma reconnaissance. Je tiens tout d'abord à remercier Christian Laugier, directeur de recherche à l'INRIA, pour m'avoir accepté dans son équipe et soutenu tout au long de ce parcours.

Je souhaite adresser mes plus sincères remerciements aux membres de mon jury. Ainsi, je remercie Monsieur Augustin Lux, Professeur INPG, pour avoir accepté de présider le jury. J'ai apprécié en particulier la participation de Monsieur Vincent Dupourqué, PDG de Robosoft, qui s'est déplacé de loin pour être présent en tant qu'examinateur. Je tiens à remercier Monsieur Bernard Espiau, directeur de l'INRIA Rhône-Alpes qui m'a fait l'honneur et le plaisir de participer. Ma plus sincère reconnaissance à mes rapporteurs, Monsieur Dominique Meizel, Professeur à l'Université de Compiègne, pour les discussions que nous avons eu et Monsieur Jean-Paul Laumond, directeur de recherche CNRS, pour son encouragement et l'intérêt qu'il a porté à ce mémoire.

Les meilleurs sentiments et ma plus vive gratitude à mon encadrant de thèse, Sépanta Sekhavat, chargé de recherche INRIA, dont l'esprit critique et formel ont influencé de manière incontestable ce document. Il est indéniable que sa rigueur d'analyse et sa grande perspicacité scientifique ont marqué mon esprit. Pourtant, et il le sait, cela n'a pas été facile de travailler ensemble. Nous avons eu beaucoup de différents au cours de nos collaborations. Mais qu'il sache que je garderai les fruits positifs de ces jours d'accords et désaccords. En particulier, je n'oublierai jamais (et je pense que lui non plus d'ailleurs) cette phrase philosophique qui nous apportait toujours de l'espoir dans les moments fatidiques:

≪ *aghh... too much complicated... Do some simulation!* ≫ Jiangzhou Lu

J'ai eu la grande chance d'avoir travaillé en équipe avec les nombreux collaborateurs du projet SHARP. Ainsi, je remercie tous ceux qui à un moment ou à un autre m'ont donné un coup de main, que se soit technique ou encore dans le va-et-viens de la vie quotidienne. En particulier je garde un souvenir agréable de César Mendoza, François Boux de Casson, Diego d'Aulignac, David Raulo, Anton Deguet, Rémis Balaniuk, Ivan Costa, Kenneth Sundaraj, Christophe Coué, Kamel Mekhnacha, Carla Koike, Jaun-Manuel Ahuactzin, Cédric Pradalier, Olivier Malrait et Olivier Lebeltel pour nommer quelques uns... Je pense aussi à mon ami Frédéric Large, compagnon infatigable de bureau, qui a toujours su mettre de la bonne ambiance dans l'équipe. Je garde aussi de bons souvenirs des permanents: Thierry Fraichard, Emmanuel Mazer et Pierre Bessière ; leur recul scientifique m'a toujours aidé à cadrer mon travail.

Les mots me manquent pour exprimer mon immense reconnaissance à ma famille :

Alicia, Amaury, Araí et Danilo, qui m'ont supporté et soutenu inconditionnellement durant ces longues années de croissance. Je leur dédie tous les fruits de mon travail. Qu'ils sachent qu'ils étaient toujours dans mon esprit, même si mon comportement et mes absences semblaient dire le contraire...

Motion Planning & Feedback Control of Bi-steerable Robots: an Approach Based on Differential Flatness

Jorge Hermosillo Valadez

INRIA Rhône-Alpes
ZIRST, 655 AV. DE L' EUROPE
38334 ST. ISMIER CEDEX

June 2003

Abstract

This dissertation addresses the problem of planning and executing motions for a special class of four-wheel robot with double steering axles: we call *bi-steerable car* a vehicle capable of steering its rear wheels in function of the front steering angle. The differential equations describing the control system set new problems of motion planning and control in mobile robotics. The methodology employed to solve these problems is framed within the theory of Nonlinear Control and in particular *differential flatness*. The objective is to find state and control transformations, putting the system in normal forms with structural properties convenient for control design purposes. This dissertation shows first that the bi-steerable car is *flat*. For these systems, effective solutions can be found in the literature and resort to the structural properties of the *flat* or *linearizing output*. Notwithstanding, the major difficulty, and open problem in the general case, is to find transformations yielding a flat output of the system. Hence we first establish theoretical results framed in the theory of Pfaffian systems and in particular those concerned by the Engel's theorem. For such systems, we propose a systematic approach leading to a necessary condition on the coordinates transformations yielding a linearizing output. We apply this approach to the bi-steerable car. We consider the symmetries of our problem in order to effectively compute a flat output. We make this result effective by proposing the first complete motion planner for this kind of mechanical structure. We next establish results in the framework of flatness and *feedback linearization* regarding the bi-steerable car. This allowed us to fully linearize the system and to synthesize a simple control law to solve the trajectory tracking problem. Experimental results validate these theoretical issues.

Keywords:
Nonholonomic Motion Planning, Mobile Robotics, Double-steering axles vehicle
Differential flatness, Endogeneous feedback

Contents

List of Figures

Planification de Mouvements et Commande de Robots Bi-guidables : Une Approche Basée sur la Platitude Différentielle

Introduction

1 Résumé

Cette thèse porte sur les problèmes de planification et d'exécution de mouvements pour une classe de robots de type voiture sans essieu fixe. Nous définissons une voiture *bi-guidable* comme un véhicule capable d'orienter ses roues arrières en fonction de la commande de braquage de l'essieu avant. Un tel système présente des avantages de manœuvrabilité accrue dans des espaces encombrés tels ceux rencontrés par les petits véhicules urbains.

Plusieurs structures mécaniques, allant de la voiture conventionnelle aux systèmes tracteur-remorque, ont fait l'objet d'études approfondies sur la planification non-holonome et la commande non-linéaire depuis le milieu des années 1980 ([Lau86], [Lat91a], [LC92], [Lau98], [RFLM93a], [RFLM93b], [TMS95], [BTS95], [LSL99]). Des recherches menées dans ces domaines ont permis de dégager des modèles formels pour ces robots conduisant à des solutions efficaces d'un point de vue algorithmique. Nous trouvons ainsi dans la littérature des solutions à des problèmes de commande en boucle ouverte et de stabilisation en boucle fermée pour certaines classes de systèmes : *nilpotents* [LS93], *chaînés*[MS93] et *plats* [FLMR95b]. Jusqu'à maintenant, la voiture bi-guidable n'a pas été étudiée avec ces approches. En effet, la cinématique bi-guidable relève d'une complexité nouvelle pour laquelle aucune solution existante ne peut-être appliquée directement.

Dans cette thèse, nous proposons une approche basée sur la *platitude différentielle* [FLMR95b]. Un système est dit *plat* si son comportement peut être complètement décrit par un nouvel ensemble de fonctions différentiellement indépendantes, appelé *sortie plate* ou encore *sortie linéarisante*, elles-mêmes fonction des variables constitutives du système et de leurs dérivées. Toute trajectoire du système peut alors s'obtenir à partir de cet ensemble de fonctions sans intégrer d'équations différentielles. Formellement, la platitude se définit par une relation d'équivalence [Mar92]. Deux systèmes F et G sont équivalents s'il existe une correspondance biunivoque entre les trajectoires de F et celles de G. Par définition, un système est plat s'il est équivalent à un ensemble de fonctions arbitraires.

Cette dissertation établi d'abord l'existence d'une sortie linéarisante pour la voiture bi-guidable. La difficulté majeure est de transformer ce résultat d'existence en un résultat effectif permettant de construire la sortie plate. En effet, nous montrons que le problème de trouver une sortie plate pour la voiture bi-guidable n'a pas une solution immédiate et demande l'utilisation d'outils mathématiques très sophistiqués ainsi qu'une approche méthodique de calcul.

La question : "*comment trouver la sortie plate?*" est ouverte dans le cas général.

Quelques approches trouvées dans la littérature, concernant certains robots non-holonomes, reposent sur l'analyse des *équations extérieures* décrivant les contraintes du système. Elles correspondent à des formes différentielles de degré 1. Les outils d'analyse sont issus de la théorie des systèmes de Pfaff. En effet, les théorèmes de Pfaff, d'Engel et de Goursat donnent des conditions suffisantes pour établir l'existence de nouvelles coordonnées, dans lesquelles il est possible de paramétrer de façon arbitraire des solutions aux équations extérieures. Il s'avère que la sortie plate d'un système de commande fait partie de ce nouvel ensemble de coordonnées. Or, malgré ces résultats, il n'existe pas dans la littérature une démarche formelle de calcul explicite de cet ensemble ou de la sortie plate dans le cas général.

Nous apportons une réponse concernant les systèmes à deux entrées et quatre variables d'état appartenant à la classe des systèmes d'Engel. Pour cette classe, nous proposons une démarche systématique qui repose sur la démonstration du théorème, aidant au calcul d'une sortie plate.

Nous appliquons cette démarche pour calculer de manière explicite la sortie plate de la voiture bi-guidable. Nous rendons ce résultat opérationnel en proposant le premier planificateur complet pour la voiture bi-guidable.

Nos travaux portent ensuite sur la stabilisation du système autour de la trajectoire de référence. Dans le domaine de la commande non-linéaire, la platitude différentielle a une interprétation par bouclage dynamique. Le lien entre équivalence et bouclage est le suivant [Mar92]: si F et G sont deux systèmes équivalents, il est possible de trouver un bouclage dynamique particulier (dit *bouclage endogène*[1]) et un changement de coordonnées qui transforment F en G étendu par des intégrateurs purs. Dans ce sens, un système plat est un système linéarisable par bouclage endogène [FLMR99].

Dans ce contexte, nous menons une étude approfondie de la dynamique de la sortie plate et trouvons de nouvelles relations complémentaires aux précédantes. Ces nouvelles relations complètent l'étude sur la platitude des robots bi-guidables et nous permettent de linéariser le système.

Nous profitons de ces résultats pour stabiliser les trajectoires du robot en employant des méthodes issues de la théorie de la commande des systèmes linéaires. Nous synthétisons une loi de commande par retour d'état afin de résoudre le problème du suivi de la trajectoire de référence. Nous obtenons des résultats en simulation qui valident notre approche.

Enfin, Nous poursuivons la validation de ces aspects théoriques dans une phase d'expérimentation.

2 Contributions

Nos principales contributions se résument ainsi:

[1] Ce type de bouclage est appelé ainsi car il est "engendré" par les variables du système et leurs dérivées (aucune variable exogène au système est employée).

- Nous montrons que la voiture bi-guidable est un système plat. Nous donnons une condition nécessaire et suffisante sur la relation angle de braquage arrière–angle de braquage avant pour garantir la platitude.

- Nous proposons une démarche systématique pour calculer une sortie plate concernant un système général d'Engel. La démarche aboutit à une condition nécessaire se traduisant par deux équations différentielles partielles où les inconnues sont les transformations recherchées.

- Nous résolvons le problème complet de planification non-holonome pour la voiture bi-guidable. A cette fin, nous utilisons la démarche proposée. Le calcul explicite de la sortie plate profite des symétries du problème en considérant les invariances par translation rotation de la trajectoire dans le plan.

- Nous établissons des résultats d'équivalence entre la voiture bi-guidable et un système linéaire commandable et nous donnons la forme chaînée du système bi-guidable. Ces résultats permettent de résoudre le problème d'exécution de trajectoires pour la voiture bi-guidable et ouvrent la possibilité de résoudre d'autres problèmes de stabilisation via la forme chaînée.

3 Plan du manuscrit

Chapitre I

Dans un premier temps, nous proposons un état de l'art en matière de planification et commande de systèmes non-holonomes. Ceci nous amène à présenter les points essentiels de chaque domaine. En particulier, nous discutons quelques notions de *commandabilité*, les méthodes de guidage (ou commande en boucle ouverte) trouvées dans la littérature, ainsi que les problèmes et solutions liés à la stabilisation en boucle fermée. Il s'avère que des solutions efficaces d'un point de vue algorithmique peuvent être trouvées, lorsque les systèmes étudiés appartiennent à certaines classes de systèmes : *nilpotents*, *chaînés* et *plats*. Nous justifions le choix retenu (la platitude différentielle) pour résoudre les problèmes qui nous concernent. En effet, la platitude s'avère à la fois une méthode formelle de classement de systèmes linéarisables par bouclage endogène, ainsi qu'une démarche élégante et puissante pour résoudre de manière intégrale et simple des problèmes de commande en boucle ouverte et fermée.

Chapitre II

Dans ce deuxième chapitre, nous introduisons la cinématique du robot bi-guidable. Nous montrons d'abord que le système est plat. A cette fin, nous utilisons des résultats concernant le rang de l'algèbre de Lie pour les systèmes à deux entrées. Il s'agit ensuite de convaincre le lecteur de l'intérêt de la platitude pour résoudre des problèmes de planification. Nous proposons une étude comparative entre le robot bi-guidable et son "cousin" le robot de type voiture conventionnelle. Nous rappelons les équations différentielles décrivant leur cinématique, qui reposent sur des hypothèses de roulement sans glissement ([Lat91b],[LSL98]). Malgré les proches similitudes entre l'une et l'autre, les résultats

établis chez la voiture (reposant sur la platitude de celle-ci) sont loin de s'appliquer directement sur un robot bi-guidable pour résoudre la problématique qui nous concerne. Cette discussion est présentée en fin de chapitre où nous montrons que le problème de trouver une sortie plate pour le nouveau système est beaucoup plus complexe et demande l'utilisation d'outils mathématiques très sophistiqués ainsi qu'une approche méthodique de calcul.

CHAPITRE III

Nous abordons la partie centrale du sujet. Nous discutons d'abord un état de l'art sur les méthodes employées dans la littérature pour résoudre des problèmes de commande en boucle ouverte pour certains robots non-holonomes. Ces méthodes reposent sur une ré-paramétrisation de la variété différentielle. Il s'agit de trouver des transformations de l'état et de la commande, afin de mettre le système sous une forme dite *normale*. L'intérêt est qu'il est beaucoup plus simple, d'une part, d'étudier les solutions aux équations différentielles décrivant le système, et d'autre part de synthétiser des contrôleurs pour la commande. Le noyau dur de cette approche est constitué par le calcul explicite des transformations requises.

La question : "*comment trouver la sortie plate?*" est ouverte dans le cas général. Nous proposons une démarche systématique pour le cas des systèmes à deux entrées et quatre variables d'état, qui repose sur la démonstration du théorème d'Engel. Nous aboutissons à une condition nécessaire donnant un ensemble d'équations différentielles faisant intervenir les transformations du calcul de la sortie plate pour de tels systèmes.

Nous appliquons cette démarche à la voiture bi-guidable. Nous trouvons une sortie plate et les transformations inverses grâce à l'exploitation des symétries du problème et à l'identification d'un repère tournant par rapport au robot. Ces symétries s'expriment par les invariances de la planification et du suivi de trajectoires par translation et rotation dans le plan. En prenant en compte ces invariances, il est possible de simplifier les calculs en considérant un robot immobile. En effet, la seule variation importante concerne la courbure de la trajectoire, qui ne dépend que de l'angle de braquage des roues.

Nous faisons ensuite une adaptation d'un planificateur local (ou méthode de guidage) conçu à l'origine pour des systèmes chariot à remorque [SLL+97]. Ce planificateur lie deux points de l'espace de configurations par une courbe suffisament lisse calculée dans l'espace plat. Afin de garantir la complétude du planificateur global, la méthode de guidage permet l'introduction d'un point de rebroussement vérifiant la propriété topologique (**TP**). Schématiquement, un planificateur vérifie **TP** si, lorsque deux points de l'espace de configurations (q_1, q_2) sont suffisament proches, c'est à dire qu'ils appartiennent à une boule B_{q_1,q_2}, le chemin menant de q_1 à q_2 reste à l'intérieur d'un voisinage contenant B_{q_1,q_2}. L'adaptation consiste à introduire un point de rebroussement entraînant un chemin symétrique entre deux points de l'espace de configurations, tout en respectant **TP**. Nous montrons par des simulations que cette approche résout le problème de planification pour la voiture bi-guidable d'une manière plus efficace en termes de manœuvres réalisées.

CHAPITRE IV

Le problème de planification de chemins une fois résolu, nos travaux portent ensuite sur la stabilisation du système autour d'une trajectoire de référence. Puisque le voiture bi-guidable est un système plat, une solution naturelle consiste à exploiter les propriétés de

la sortie plate.

Nous établissons des résultats d'équivalence entre la voiture bi-guidable et un système linéaire commandable écrit sous sa forme canonique de Brunovsky ([Kai80]). En effet, nous trouvons des transformations, complémentaires aux précédentes, par une étude approfondie de la dynamique de la sortie plate. Ces relations établissent notamment le lien entre la commande du robot et la commande du système équivalent exprimé en coordonnées plates. Ces nouvelles relations complètent l'étude sur la platitude des robots bi-guidables. Grâce à elles, nous calculons de façon explicite la *forme chaînée* du système. Ceci ouvre la possibilité d'explorer de nombreuses méthodes de stabilisation du robot en un point.

Nous profitons de ces résultats pour calculer un bouclage linéarisant. Ainsi, nous cherchons à stabiliser les trajectoires du robot en employant des méthodes issues de la théorie de la commande des systèmes linéaires. Nous synthétisons une loi de commande par retour d'état afin de résoudre le problème du suivi de la trajectoire de référence. Nous obtenons des résultats en simulation qui valident notre approche.

CHAPITRE V

Dans ce chapitre nous commençons par discuter l'incertitude présente, issue de l'incomplétude de nos modèles vis-à-vis d'un robot réel ([BDL+99a], [BDL+99b], [LDBM00]). Nous présentons ensuite notre plateforme expérimentale, le robot Cycab. Nous poursuivons par l'introduction du système de localisation, indispensable pour réduire l'incertitude relative à l'état courant de notre robot. Enfin nous discutons plusieurs résultats obtenus au cours des expérimentations.

CHAPITRE VI

Nous concluons ce rapport en résumant nos principales contributions et discutons des possibles axes de recherche à poursuivre. En particulier nous soulignons la nécessite de prendre davantage en compte l'incertitude liée à nos modèles ainsi que le besoin d'incorporer dans l'architecture de contrôle des comportements réactifs et des méthodes de planification en ligne.

Motion Planning & Feedback Control of Bi-steerable Robots: An Approach Based on Differential Flatness

Introduction

1 Context of our work

This dissertation is concerned about motion planning and execution issues for a four-wheel vehicle with a double steering capability. We call *bi-steerable car* a vehicle capable of deflecting its rear wheels in function of the front steering angle. To our knowledge, our work is the first to tackle the motion planning and feedback linearization problems introduced by this mechanical structure.

The motion planning issue in Robotics began to be formalized in the early 1980's [Lat91a]. The formalization of the problem relies on the notion of *configuration space*, inspired from Mechanics. The configuration of a robot is the specification of the position and orientation of every part of the robot with respect to a Cartesian reference frame embedded in its workspace. The configuration space of a robot, denoted $\mathcal{CS}$, is the space of all its possible configurations. Each configuration of the robot is represented by a point in $\mathcal{CS}$. The geometric formulation of the problem considers the motion of rigid bodies amidst obstacles in the three-dimensional Euclidean space. Some geometric relations between the bodies may appear for a given robotic system; a typical example is a robot manipulator composed of rigid links connected by joints. These relations are translated into equations between the configuration parameters. They restrict the degrees of freedom of the robot and are called *holonomic* kinematic constraints. Together with no-collision constraints, imposed by the obstacles in the workspace, they yield forbidden (or *non-free*) configurations. In this framework, the basic motion planning problem reduces to exploring the *free* configuration space in order to find a *connected path* joining a start and a goal configuration. Solutions to this problem resort to algorithmic geometry techniques. By the mid 1980's, the difficulty of the basic issue increases with the introduction of additional constraints on the velocities of the configuration parameters ([Lau86]). Such constraints are called *nonholonomic*. They define the allowable velocities of the system at each point of $\mathcal{CS}$. If the system is prevented from moving in some directions of $\mathcal{CS}$, a natural question is to know whether all possible configurations in $\mathcal{CS}$ are *reachable*? This turns out to be a *controllability* question. Now the problem is not only geometric but requires additional analysis tools coming from Nonlinear Control theory.

Motion planning is essentially an *open-loop* control task: a nominal trajectory is computed off-line with a priori knowledge about the environment. The classical approach in Robotics consists in *following* (or tracking) this trajectory as accurately as possible at the execution phase. If we tried to perform this trajectory without taking care of the

actual evolution of the robot, it is very likely that the resulting motion would not be as expected: unconsidered factors at running time would make the robot deviate from the nominal path. *Feedback controllers*, driven by the current task error, are intended to overcome these problems so as to achieve some degree of robustness. This is why a *closed-loop* (or feedback) control strategy is necessary at the execution phase in order to achieve the task.

During the last two decades, there has been intensive research activity in Nonholonomic Motion Planning (NMP) and control of wheeled mobile robots (e.g. see [LC92], [Lau98]). Nonholonomic constraints arise typically in these robots owing to the rolling contact between the wheels and the ground. Wheeled robots for which NMP and control problems have been addressed are essentially of three kinds (see for instance [Lat91a], [LC92], [Lau98]): the cart or *unicycle* robot, the conventional car or *bicycle* model and *tractor-trailer* systems of diverse structures ([RFLM93a], [RFLM93b], [TMS95], [BTS95], [LSL99]). More recently car-like robots with a double steering capability have gained applicability[2]. This is probably due to the fact that the double-steering ability entails advantages over conventional cars, both from dynamics and kinematics point of views ([Whi90]): improved dynamical stability, enhanced maneuverability and reduced swept volume; even car manufacturers have been interested in rear wheel steering since the 1980's ([SFS86]). However the bi-steerable system, as defined herein, has not been studied in the general case by the Robotics community, who begins to pay attention to the challenges one needs to face ([Hem94], [WQ01]). In particular the motion planning and control problems for this kind of vehicle have never been addressed in the general case. Our work aims at bringing a contribution in this direction.

2 Problem and contribution

Basic to any control task is a formal representation of the mechanical device. Modeling a nonholonomic robot involves differential equations describing the kinematics of the system. These differential equations are parameterized by the control inputs of the device and constitute the *control system* model. Solving motion planning problems, for instance, means finding the sequence of inputs in time that *steer* the robot from the start to the goal configuration. This might be a very difficult task in the general case due to the nonlinear nature of the control system; it represents a typical problem of optimal control.

In this context, there exist some classes of systems whose properties allow to simplify NMP and control problems to a great extent. Over the last decade, theoretical surveys in Robotics and Nonlinear Control have yield criteria allowing for the classification of some nonholonomic control systems. Two of these classes are *differentially flat* [FLMR95b, FLMR99] systems and those that can be put in the *chained* form [MS93]. They possess structural properties leading to simple solutions of NMP and feedback control problems. Indeed, there exist powerful methods ([LS93],[MS93], [TMS95]) to control the chained form in order to drive the system exactly to the goal configuration. On the other hand,

[2] e.g. the "πCar" prototype of IEF ("Institut d'Electronique Fondamentale" of Paris-Sud University.) and the Cycab robot at INRIA in France and NTU in Singapore.

flat systems admit a new set of variables, called the *flat* or *linearizing* output. It contains all the dynamical information of the original control system. This implies that we can compute the trajectory of the original system from that of the flat output (and vice-versa) without integrating any differential equation. It follows that flat outputs can be used immediately for trajectory generation and tracking ([RFLM93a], [FLM$^+$97], [LL97]). Moreover, the properties of the linearizing output allow us to deal with a linear control system. Since Linear Control theory offers a solid and complete framework for solving closed-loop tasks, there is a great interest in exploring these properties.

In order to be able to exploit these properties, the original control system must verify necessary and sufficient conditions allowing for its classification. In a second step, coordinate transformations must be undertaken allowing either to obtain the chained form or to find a flat output of the system. Finding these transformations is in fact the major obstacle in exploiting the chained form or flatness and the literature includes very few works on their effective computation ([MR94], [TMS95], [Pom97]).

The study of control models associated to car-like and tractor-trailer robots have led to their complete characterization in terms of flatness and the chained form. The kinematics of the bi-steerable car introduce a different complexity than the one found in these robots. Our contribution concerns the following questions: what are the structural properties of this new kind of kinematics? How to find the convenient transformations yielding the flat coordinates or the chained form? And how can we exploit their properties in order to solve motion planning and control issues regarding the bi-steerable robot?

After showing that the bi-steerable car is *flat*, we turned to the central question: "*how to compute a flat output for this system?*" In answering this question, we were interested in a general formulation of the problem. We are therefore concerned by control systems whose configuration space is of dimension 4 and which are subject to two nonholonomic constraints. In this respect, we outline a systematic approach putting forward the key steps to compute a particular flat output: one allowing to put the control system in its chained form. The outcome is a *necessary condition* on the coordinate transformations required to obtain this flat output. Hopefully, the methodology will be helpful in analyzing similar problems. This result is to our knowledge new. It allowed us to compute a flat output for the general bi-steerable car, by considering the symmetries of the problem, and thus to solve the motion planning issue for this kind of robot. The properties of the flat output pushed us to investigate further into the relation between the nonlinear bi-steerable kinematics and its underlying controllable linear system. The outcome is the full linearization of the bi-steerable car, leading to a simple solution for the trajectory tracking problem.

In view of these issues, the main contributions of our work are as follows:

- We show that the bi-steerable car is differentially flat. We give a necessary and sufficient condition on the front-to-rear steering function f in order to guarantee flatness at all points of $\mathcal{CS} \subset \mathbb{R}^4$. This entails typically that for a linear front-to-rear coupling function of the form $f(\varphi) = k\varphi$ we have that the system is flat for $k \neq 1$.
- We outline key steps to compute a flat output for a controllable control system of dimension 4 subject to two nonholonomic constraints. The outcome is a necessary

condition on the coordinate transformations required to obtain this flat output. The condition translates into a pair of PDEs where the unknowns are the transformations sought.

- We solve the complete motion planning problem for the bi-steerable car. To this end, we apply the aforementioned methodology to find a set of flat coordinates for the system. The effective computation of the flat output includes symmetry considerations: we take into account the invariance of the problem with respect to Euclidean transformations in the plane.

- We establish results between the bi-steerable car and its equivalent controllable linear system, and give the explicit chained form of the bi-steerable robot. These results allowed us to solve the trajectory tracking issue for the general bi-steerable car and open the possibility to tackle other control tasks via the chained form.

To our knowledge, our work is the first to tackle motion planning and feedback linearization issues for this kind of nonholonomic robot. The outline of the document follows.

3 Document layout

CHAPTER I. In this chapter we introduce the formal framework of our work. We discuss some relevant aspects about the state-of-the-art in motion planning and feedback control of wheeled mobile robots. This chapter frames the theoretical aspects of our work which come in Chapters II through IV.

CHAPTER II. In this chapter we introduce the bi-steerable robot. We firstly turn to its kinematics model. We then establish some connections between typical approaches for solving the path-planning problem in the context of flatness. This shows how the solutions to the problem resort to the properties of the flat output. We then show that the bi-steerable car is flat and discuss the core of the problem we need to face in finding a flat output for the system. It soon appears, that the complexity of this new problem resorts to advanced mathematical tools in order to address it effectively.

CHAPTER III. In this chapter we propose a systematic approach to find the transformations that send the original coordinates of a class of two-input control systems to its flat coordinates. The explicit computation of this flat output for the general bi-steerable system is then outlined. This allowed us to solve the path-planning problem for this kind of robot, as illustrated with simulation results.

CHAPTER IV. In this chapter we present equivalence issues concerning the bi-steerable car. As a matter of fact flatness is defined through an equivalence relation ([Mar92]). Two systems are said to be equivalent if it is possible to compute the trajectories of the one from those of the other (and vice-versa) without solving any differential equation. In this respect we find further flat transformations in order to establish the desired results. This allowed us to address the exact linearization of the system and to explicitly give its chained form. By linearizing the bi-steerable car, we tackle the closed-loop control problem using Linear Control techniques as illustrated with simulation results.

CHAPTER V. In this chapter we address the experimental validation of our work. We have conducted experimentations using a real bi-steerable platform, the Cycab robot of the INRIA Rhône-Alpes. To this end, the robot requires essential localization abilities so as to reduce the uncertainty associated to its current state. Accordingly, the robot must rely on its proprioceptive and exteroceptive sensors. We shall introduce the localization system used, based on an absolute localization scheme and an odometry module. The experimental assessment of our work is given subsequently.

CHAPTER VI. We conclude by discussing the results and guidelines for future work.

Chapter I

Motion Planning and Control of Nonholonomic Wheeled Robots

In this chapter we are interested in understanding the main issues in motion planning and feedback control of nonholonomic wheeled robots[1]. The aim of this chapter is therefore to give a literature review on these topics. This will allow us to justify the approach chosen: the property of *differential flatness* to address both problems regarding the bi-steerable car.

1 Introduction

Robotics as a research field constitutes a challenging framework for integrating and experimenting constantly evolving concepts in engineering and science. One of the last frontiers in robotics is the *motion autonomy* paradigm. In this framework, the capability to plan actions has been considered as essential in achieving autonomous robots [Lat91b].

Basic to a motion planning task is a formal representation of the real mechanical device. In modeling a robotic system, one must take into account the physical constraints restricting its overall motion and/or the relative motion between its parts. Indeed, most mechanical systems are subject to kinematic constraints, which may be classified in various ways. The simplest ones are *holonomic* kinematic constraints. They are expressed by equations connecting the configuration variables. They describe geometric relations between the bodies or parts of a given robotic system. Examples of systems subject to holonomic constraints are the bead of an abacus sliding on the supporting wire, or a robot manipulator composed of rigid links connected by joints. The motion planning problem under these constraints is of a purely geometric nature: the aim is to join any two end-points in $\mathcal{CS}$ through a connected clear (obstacle-free) path. This problem is known as the *piano mover*[2]. The main difficulty concerns the analysis of the connectivity of

[1] There are indeed wheeled robots which are holonomic; e.g. omnidirectional robots like the $Nomadic^{TM}$ robot.

[2] The situation of a piano mover is here illustrative of the geometric formulation of the problem. Indeed,

the free configuration space $\mathcal{CS}$; solutions to this problem resort to algorithmic geometry techniques [Lat91b].

A different kind of kinematic constraints involves nonintegrable equations between the derivatives of the configuration variables. These are known as *nonholonomic* constraints and define the allowable velocities at every point in $\mathcal{CS}$. A typical example is a rolling disc. The assumption that the disc rolls without slipping implies that there are "forbidden" velocities during its motion: the disc is not allowed to slide orthogonally to the tangential direction of its trajectory. If at every point in $\mathcal{CS}$ the system is prevented from moving in some directions, a natural question is to know whether all possible configurations in $\mathcal{CS}$ are *reachable*? This turns out to be a *controllability* question, the answer of which resorts to analysis tools from the theory of Control.

The links between Nonholonomic Motion Planning (NMP) and Control have been formalized since the first studies about the subject ([Lau86], [Mur90]). The control model of a nonholonomic robot involves differential equations describing the kinematics of the system. These differential equations are parameterized by the control inputs of the device and constitute the *control system*. Solving the motion planning problem means finding the sequence of controls in time, *steering* the robot from the start to the goal configuration. This might be a very difficult task in the general case due to the nonlinear nature of the control system; it represents a typical problem of Optimal Control.

NMP is not only a geometric issue but turns out to be also an *open-loop* control problem. This implies that the sole finding of a connected path between obstacle-free endpoints in $\mathcal{CS}$ does not a priori mean the mechanical system will reach its goal. The path must be compatible with the kinematic constraints. The core of the problem is to steer the system along a *feasible* path while keeping it away from the obstacles. Steering nonholonomic systems in general is a difficult problem owing to their nonlinear character; its analysis requires tools from Differential Geometry and Nonlinear Control. This accounts for the research carried out during the last decade in this domain ([LC92], [Lau98]).

On the other hand, once a collision-free trajectory has been computed, there is a second problem which consists in executing this one. The classical approach in Robotics consists in *following* (or tracking) this trajectory as accurately as possible at the execution phase. In the planning phase, the resulting commands are computed off-line, according to a priori knowledge of the task and the state of the environment. If we tried to perform this trajectory without taking care of the actual evolution of the robot, it is very likely that the resulting motion would not be as expected: unmodeled events at running time (e.g. occasional slipping of the wheels) would prevent the successful completion of the task. In Automatic Control, feedback improves system performance even in the presence of external disturbances and/or initial errors. To this end, real-time sensor measurements are used to reconstruct the robot state. This information is *fed-back* to the controlling input so as to undertake a suitable action. In this respect, the major concern is the stability of the system; the problem is how to make the error, in the state of the robot, converge to zero in a stable fashion. The *closed-loop* control of a nonholonomic robot concerns the stabilization of a nonlinear system. This nonlinear character is, again, at

the task consists in finding the collision-free connected sequence of placements in the 3-dimensional space of an object having 6-degrees of freedom (3 translations and 3 rotations).

the basis of the difficulty of the problem.

In this context, there exist some classes of systems whose properties allow to simplify NMP and feedback control problems to a great extent. These systems possess structural properties that have been thoroughly studied and characterized over the last decade ([LS93],[MS93], [FLMR95b]). They allow to find effective solutions to open-loop and closed-loop control problems.

What follows is our understanding of some relevant issues concerning NMP and feedback control of wheeled robots. We are interested in addressing these problems for a mechanical system showing a new kinematics, namely the bi-steerable car. Hence, as far as NMP is concerned, we are particularly interested in discussing the control aspects (the steering problem) rather than the algorithmic ones, including complexity issues (see for instance [LSL98]).

We start by introducing elementary concepts in robotics such as configuration space and kinematic constraints. We then discuss how nonholonomic constraints affect the motion planning problem; this will introduce fundamental notions concerning the controllability of nonlinear systems.

We move on to the core of NMP regarded as an open-loop control issue: how to effectively steer the robot. We discuss how this problem may be solved using Optimal Control techniques and by some special classes of control systems, including the *chained form* and *flatness*. This will allow us to put forward how motion planning problems may be effectively solved by resorting to the properties of these classes.

On the other hand, steering nonholonomic robots in the presence of obstacles rises topological issues that need to be considered. Hence we briefly discuss a topological property for some motion planners.

We finally discuss the state-of-the-art in closed-loop control of nonholonomic wheeled robots. This will help in introducing further properties of flatness for trajectory tracking purposes.

We conclude the chapter with a discussion on the integral character of differential flatness for solving both open and closed-loop control problems. We will also show the interest of flatness in connection with the chained form of control systems.

2 Configuration space and configuration variables

A configuration of a robot is the specification of the position an orientation of every part of the robot with respect to a Cartesian reference frame embedded in its workspace. The configuration space of a robot, denoted $\mathcal{CS}$, is the space of all its possible configurations. Hence each configuration of the robot is represented by a point in $\mathcal{CS}$.

A typical representation of a point in $\mathcal{CS}$ consists in the vector of generalized coordinates $\mathbf{q} = (q_1, q_2, \ldots, q_n)$, where n is the dimension of $\mathcal{CS}$. For example, the configuration of a car-like robot is represented by a point $\mathbf{q} = (x_r, y_r, \theta)$. Here (x_r, y_r) are the coordinates of a point of the robot (e.g the midpoint of the rear axle) in some Cartesian frame

of the workspace. The third parameter θ is the orientation of the main axis of the car relatively to the x axis of the Cartesian frame.

The configuration space of a mechanical system made of rigid bodies is a smooth manifold [Lat91b]. For instance, the configuration space of a two-dimensional rigid body translating and rotating in $\mathbb{R}^2$ is $\mathcal{CS} = \mathbb{R}^2 \times S^1$, where S^1 denotes the unit circle.

In general, the motions of a mechanical system are constrained in a particular way. Typically, a system consisting in a sequence of rigid links connected by joints (e.g. robot manipulators) induces geometric constraints on the configuration parameters. On the other hand, a rolling contact for instance (e.g. a wheel on a surface) induces constraints on the velocities of the configuration parameters. We introduce now these two kinds of kinematic constraints which are at the origin of motion planning problems.

3 Kinematic constraints

3.1 Holonomic constraints

A holonomic constraint reduces the degrees of freedom of the mechanical system. It is expressed by an equation relating the configuration variables and possibly time:

$$F(\mathbf{q}, t) = 0 \quad \mathbf{q} \in \mathcal{CS}, \tag{I.1}$$

Assuming that F is smooth, it is possible to solve for one of the generalized coordinates in function of the others and time. For many robotic practical applications the constraint (I.1) is time-independent; we shall focus our attention on these kind of constraints throughout this dissertation.

Equation (I.1) defines a $(n-1)$-dimensional submanifold of $\mathcal{CS}$. It is in fact the configuration space of the robot. The $n-1$ remaining generalized coordinates represent the independent degrees of freedom of the system. Hence a holonomic constraint reduces the configuration space of the robot. In general, k independent holonomic constraints reduce the configuration space of a mechanical system to a sub-manifold of dimension $n-k$.

Example: the pendulum

A simple example is a spherical pendulum in 3-dimensional (3D) space constrained to remain at a fixed distance from the origin. The mathematical representation of such a constraint is therefore:

$$x^2 + y^2 + z^2 - R^2 = 0.$$

Physically, the manifold on which the motion can be described is the sphere of radius R, so the configuration space has been reduced to two dimensions: the surface of the sphere. In this case the generalized coordinates may be two independent angles expressing the position of a point on the sphere; i.e. $\mathbf{q} = (\alpha, \beta)$ where α, β are say the latitude and the longitude. □

3.2 Nonholonomic constraints

A (time-independent) nonholonomic constraint is expressed as a nonintegrable equation involving derivatives of the configuration variables:

$$G(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0} \quad \dot{\mathbf{q}} \in \mathbf{T_q M}, \tag{I.2}$$

where $T_{\mathbf{q}}M$ is the tangent space of $\mathcal{CS}$ at $\mathbf{q}$. If the constraint is linear in the velocities one can write:

$$G(\mathbf{q}, \dot{\mathbf{q}}) = \omega(\mathbf{q}) \cdot \dot{\mathbf{q}} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \omega_{\mathbf{i}}(\mathbf{q}) \dot{\mathbf{q}}_{\mathbf{i}} = \mathbf{0} \quad \dot{\mathbf{q}} \in \mathbf{T_q M}.$$

In the sequel, we shall consider only nonholonomic constraints of this kind. As we will see, such a constraint reduces the space of velocities in $\mathcal{CS}$ by one dimension. It defines a $(n-1)$-dimensional subspace of the tangent space, specifying the velocities achievable by the system at every point of $\mathcal{CS}$. More generally, k nonholonomic constraints define a $(n-k)$-dimensional subspace of allowable velocities at every point of the configuration space.

Typically, car-like robots are subject to this kind of constraints. Indeed, in the utmost simplification the assumption is that there is a pure rolling contact (with no slippage) between the surface of the road and the wheels. This contact precludes a lateral displacement of the wheels. Thus the admissible velocities are restricted to those collinear to the free rolling direction. At each point of the configuration space the constraints are nonintegrable—i.e. they cannot be expressed in the form (I.1)—and for this reason are called nonholonomic.

Example: the unicycle

A typical nonholonomic mobile robot is the two-wheel differential drive robot (usually stabilized with castors: additional self-oriented wheels). Its 2-dimensional (2D) representation is shown in Figure I.1.(a).

The assumption that the wheels roll without slipping leads to a further simplification: the so called *unicycle* model. In this model an imaginary wheel is placed at the middle point of the wheels axle, noted R (see Figure I.1). This amounts to consider a disc of centre R rolling on the horizontal xy plane. The disc is constrained to move so that its plane is always vertical.

The configuration space is $\mathcal{CS} = \mathbb{R}^2 \times S^1$ (the position of the robot in the plane and its orientation). One is free to chose any point belonging to the robot in order to represent it in $\mathcal{CS}$. For convenience we chose the point R and denote by (x_r, y_r) its coordinates in the plane, with respect to a global reference frame. By denoting θ the orientation of the robot, a configuration in $\mathcal{CS}$ is given by the point $\mathbf{q} = (x_r, y_r, \theta)$. Furthermore, we denote by (v, ω) the linear and angular speeds of the robot. In this case, v corresponds to the linear speed of R.

This time, coordinates (x_r, y_r) and θ are not anymore independent. They are connected by the constraint of "rolling": a change in the position of the point of contact,

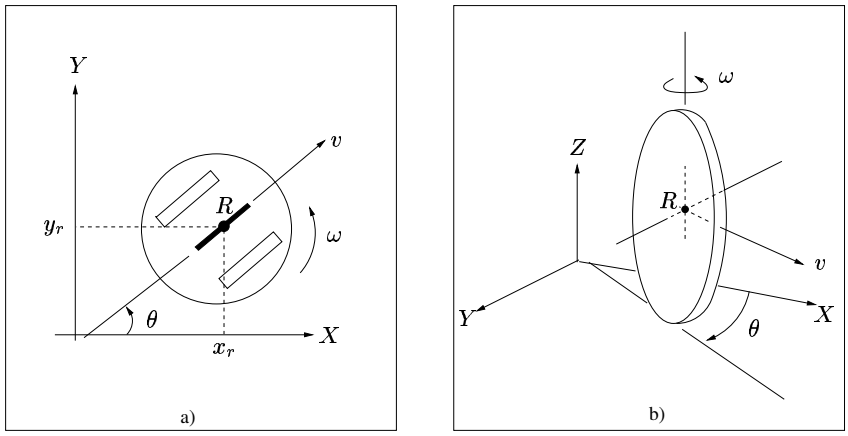


Figure I.1: (a) 2-D model of a simple nonholonomic wheeled robot and (b) its corresponding 3D *unicycle* model.

between the disc and the ground, will induce a change in the orientation of the former. However the constraint does not involve an equation between the coordinates as in (I.1). Rather, it says that the point of contact is stationary—i.e. momentarily at rest—implying that the constraint applies to the velocities of the coordinates. This means that, at any point in $\mathcal{CS}$, not all differential motions are admissible; in other words they are subject to the following constraint:

$$\sin(\theta)\mathrm{d}x_r - \cos(\theta)\mathrm{d}y_r + 0\mathrm{d}\theta = 0, \tag{I.3}$$

which is nonintegrable. Hence for any trajectory of this system, the velocity of R points in the direction θ. The admissible velocities are those that verify the constraint (I.3). □

4 Nonholonomic motion planning as a control issue

We are now interested in understanding the motion planning problem under nonholonomy.

Roughly speaking, there are two constraints that must be addressed (either separately or simultaneously). The first one consists in avoiding the obstacles; this is a geometric issue. The second one is the consideration of the nonholonomic constraints imposed by the particular mechanical system.

More generally, NMP addresses two problems [LSL98]:

1. The ***decision problem***: it consists in determining whether there **exists** a collision-free admissible path.

2. The ***complete problem***: if a this path actually exists, then it has to be **computed**. Solutions to the complete problem are called *exact* methods. There are indeed *approximated* methods, which give a solution up to a discretization of the configuration

space, and *probabilistic* methods which give a solution with probability **1** provided the computing time tends to infinity.

In order to address effectively NMP issues, the comprehension of how a nonholonomic constraint of the form (I.3) affects the robot's motion is of primary importance. Given such an equality constraint, a first natural question is to know whether it is ***integrable***. For, if it is, we know that such a constraint reduces the configuration space. If it is nonintegrable a second justified question is to know whether all possible configurations in $\mathcal{CS}$ are ***accessible***.

Both questions concern the decision problem and their answers are given by Differential Geometry and Nonlinear Control. More precisely, a nonholonomic constraint entails a certain robot kinematics. This kinematics is usually modeled by a set of differential equations, whose free parameters are the control inputs of the system. Such model defines a *control system*. Its controllability properties are better understood in the differential geometric setting of nonlinear control theory.

We recall now some fundamental notions that will be of utility in the sequel. We start with control systems without drift and then discuss basic controllability notions regarding the decision problem. We end the section by discussing main obstacles towards the complete issue. We assume the reader is acquainted with notions about vector fields, Lie brackets, distributions, etc.. The unfamiliar reader may refer to Appendix A on page 140 for a brief recall on these and related concepts.

4.1 Driftless control systems

A nonlinear control system is usually a set of differential equations of the form

$$\dot{x} = f(x,u) \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m,$$

where the manifold structure of $\mathcal{CS}$ is considered locally as $\mathbb{R}^n$. When the system is affine in the controls one can write it as a weighted combination of vector fields:

$$\dot{x} = f_0(x) + u_1 f_1(x) + \cdots + u_m f_m(x),$$

where each f_i is a vector field defined on a open subset of $\mathbb{R}^n$ and f_0 is called the drift. In this case the controls u_i correspond to accelerations or torques of the physical robot.

Systems without drift may appear when the controls are speeds instead of accelerations. For motion planning purposes this is a fair assumption in most cases. Indeed, it is not uncommon to find robots whose control architectures encompass low-level speed-servoing loops. In this case we deal with kinematic models expressed as a driftless system of the form:

$$\dot{x} = \sum_{i=1}^{m} u_i(t)\, f_i(x) \quad x \in \mathbb{R}^n. \tag{I.4}$$

At every point of the configuration space, the admissible vector fields define a vector space $\Delta(x) \subset T_xM$ of dimension m. The distribution

$$\Delta = span\{f_1, \ldots, f_m\}$$

defines the system constraints in terms of admissible differential motions associated to the vector fields $\{f_1, \ldots, f_m\}$.

Example: kinematics model of the unicycle

Suppose that the controls of the system are the linear and angular speeds (v, ω). In this case, a point in $\mathcal{CS}$ is called the *state* of the robot. Let us note the state vector $x = (x_r, y_r, \theta)$. The kinematics model of the unicycle may be given by any weighted combination of admissible vector fields—i.e. those associating to each configuration variable a velocity verifying the constraint (I.3). Thus a typical example of kinematics model for the unicycle is given by the following set of differential equations:

$$\begin{pmatrix} \dot{x_r} \\ \dot{y_r} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega. \tag{I.5}$$

They verify the nonholonomic constraint (I.3).

Model (I.5) shows that the dimension of the space of achievable velocities at all x is smaller than the dimension of the configuration space. Indeed, at every $x \in \mathcal{CS}$ the controls (v, ω) stress respectively the two vector fields:

$$f_1(x) = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{pmatrix}, \quad f_2(x) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

generating thus a 2-dimensional space. The distribution associated to system (I.5) is $\Delta = span\{f_1, f_2\}$. □

4.2 Nonholonomic constraints and the decision problem

4.2.1 Integrability of a nonholonomic constraint

Consider a nonholonomic equality constraint

$$F(x, \dot{x}) = 0 \quad \dot{x} \in T_x M.$$

For every $x \in \mathcal{CS}$, equation (4.2.1) determines a $(n-1)$-distribution $\Delta(x)$, which is an hyperplane included in the tangent space $T_x M$.

The Frobenius theorem (see e.g. [BCG$^+$91]) gives a necessary and sufficient condition to determine if (4.2.1) is integrable. This one asserts that such an equality constraint is integrable (or holonomic) if and only if the distribution (Δ), describing the system, is closed under the Lie bracket operation. That is, any additional vector field generated by bracketing between the original vector fields belongs to Δ (i.e. no further dimension is added). A distribution verifying the Frobenius theorem is said to be involutive.

On the other hand, the question of whether a nonholonomic constraint reduces the set of *accessible* configurations translates in a *controllability* issue. The control Lie algebra

associated with a given distribution Δ, denoted by $LA(\Delta)$, is the smallest distribution which contains Δ and is closed under the Lie bracket operation. A key issue in nonholonomic motion planning concerns the study of $LA(\Delta)$, giving information about the controllability of the system. We now review some important results.

4.2.2 Small-time controllability of driftless systems

Controllability of nonlinear systems has been studied for long (at least 6 decades). The controllability problem arises when we ask whether two points of a given manifold M are accessible from each other by trajectories tangent to a given distribution (defining the system constraints)—e.g. see [BJR98].

This question has been answered in a constructive manner, with several shaded meanings, by different authors. We can for instance mention the pioneering work of Chow [Cho39] who settled the so called Lie Algebra Rank Condition (LARC), defining accessibility in the wider sense. The condition of Chow (taken from [BJR98]) reads:

Definition 4.1 (Chow's condition—LARC) : *For a given system (I.4), the vector fields $f_1, \ldots, f_m$ and their iterated brackets $[f_i, f_j], [[f_i, f_j], f_k]$, etc. span the tangent space T_xM at every point of M.*

A system satisfying this condition is said to be *maximally nonholonomic*. For control systems without drift there is a stronger accessibility concept (see e.g. [LSL98]):

Definition 4.2 (small-time controllability) *A driftless system is locally controllable from x if the set of points reachable from x by an admissible trajectory contains a neighbourhood of x. It is small-time controllable from x if the set of points reachable from x before a given time T contains a neighbourhood of x for any T.*

Small-time controllability of driftless control systems states that it is possible to find a trajectory from some point x^0 to any point contained in a neighborhood, **little though it would be**, of x^0.

A necessary and sufficient condition for small-time controllability is due to Sussmann and Jurdjevic [SJ72]. It says that symmetric[3] driftless systems are small-time controllable if and only if the LARC condition is verified:

Theorem 4.1 ([SJ72]) *A symmetric system without drift is small-time controllable from x iff the rank of the vector space spanned by the family of vector fields f_i together with all their Lie brackets is n at x.*

Example: Lie algebra of the unicycle

[3] A symmetric driftless system is one for which the control domain is symmetric with respect to the origin. This is the case of most mobile robots of the type *car-like*. One exception is the Dubins' car[Dub57] which does not admit backward motions.

Consider the unicycle kinematics model:

$$\begin{aligned} \dot{x}_r &= v\cos(\theta) \\ \dot{y}_r &= v\sin(\theta) \\ \dot{\theta} &= \omega. \end{aligned}$$

Let us analyse the Lie algebra associated to this system. The state vector is $x = (x_r, y_r, \theta) \in \mathbb{R}^3$ and the associated distribution is

$$\Delta = span\{f_1, f_2\}, \quad f_1(x) = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{pmatrix}, \quad f_2(x) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Computing the first brackets yields

$$f_3(x) = [f_1, f_2](x) = \frac{\partial f_2}{\partial x} f_1(x) - \frac{\partial f_1}{\partial x} f_2(x) = \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \\ 0 \end{pmatrix},$$

Now if we set $E_2 = span\{f_1, f_2, f_3\}$, it is readily verified that $dim\, E_2 = n = 3$. Therefore, the Lie algebra of the unicycle verifies $LARC$: the system is small-time controllable for every $x \in \mathbb{R}^3$.

We remark that g_3 is the vector field corresponding to the (forbidden) direction of motion, perpendicular to the main axis of the robot. □

The relevance of this theorem, to nonholonomic motion planning, becomes evident in view of the following result [LSL98]: the decision problem for a nonholonomic system is the same as the decision problem for the associated holonomic one (i.e it is decidable), provided that the system is symmetric small-time controllable.

As a consequence, if a control system is symmetric small-time controllable then the complete problem may be solved in a two-step approach. The first step is to find any collision-free path between an initial and a final configuration for the unconstrained (holonomic) system. If this collision-free path does not exist then we know that the complete problem cannot be solved. Otherwise, the second step consists in approximating this path, by steering the mechanical system between collision-free configurations. Thanks to small-time controllability, it is possible to approximate the collision-free holonomic path, no matter how close this path is to the obstacles.

4.3 From controllability to path planning: the complete problem

These results give answers to the decision problem for particular systems under nonholonomic constraints. Unfortunately they do not solve the complete problem of effectively producing a trajectory.

Furthermore, it is observed that nonholonomy arises typically when there are less controls than configuration variables– i.e. $m < n$. For the car-like robot for instance $n = 3$ (x, y, θ) and $m = 2$ (e.g. the linear and the angular speeds). As a consequence any

path in the 3D configuration space is not necessarily an *admissible* path for the system. An admissible trajectory is a continuous function, parameterized in time, defined on some interval $[0, T]$ to $\mathcal{CS}$. An admissible path is the image of an admissible trajectory in $\mathcal{CS}$. The goal of nonholonomic motion planning is therefore to provide *collision-free admissible trajectories* in the configuration space of the robot.

Computing an admissible trajectory in a single step may be a very difficult problem. An alternative solution consists in translating the nonholonomic constraints into geometric constraints (e.g. curvature) in order to find feasible paths. Then the problem consists in transforming such a path into a trajectory. Transforming an admissible path into an admissible trajectory is a classical problem in Optimal Control. This problem has been investigated by the robotics community in the field of articulated manipulators [SHH92] and in the case of tractor-trailer systems [Lam97].

A planner computing admissible paths irrespective of the obstacles is called a *local path planner* or *steering method*. Its task is therefore, given any two points q_i and q_f in $\mathcal{CS}_{free}$, to compute a feasible path between them. Today there is no algorithm that guarantees **any** nonholonomic system to reach an accessible goal exactly. One might as well say that steering the robot stills the major difficulty in NMP problems. The difficulty stems from the intrinsic nonlinear nature of nonholonomic systems.

The following sections collect some results concerning steering methods for nonholonomic wheeled robots. In a first time we will discuss the optimal control approach, including some results regarding car-like robots. Subsequently, we shall introduce particular nonholonomic systems possessing specific structural properties. These particular systems are of great interest because their properties have been well understood and characterized. The interest is that many nonholonomic robots fall into one of these categories for which there exist a few efficient motion planning and closed-loop control strategies.

5 Steering nonholonomic robots

5.1 Steering with optimal control

5.1.1 Introduction

Perhaps the most well-formulated method for finding trajectories of a general control system is given by optimal control theory. By giving a cost to each trajectory, one can limit the search to those trajectories which minimize the cost.

One of the most significant issues in optimal control is the Pontryagin's Maximum Principle (PMP)— see e.g [BJR98]. PMP gives a set of differential equations which are satisfied by all *extremal* curves (see below) of a cost functional[4].

For a mechanical system, this cost functional involves the line integral of the La-

[4] A functional is a real valued function on a vector space, usually of functions.

grangian[5]. Indeed, it is important to realize that solutions to the Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = 0. \tag{I.6}$$

solve an *extreme* (or optimal) path problem between two points in the configuration space. The problem can be stated as that of finding the path $q(t) \in \mathbb{R}^n$, $t_o \leq t \leq t_1$, such that the integral

$$S = \int_{t_0}^{t_1} L(q(t), \dot{q}(t), t)dt \tag{I.7}$$

is minimal. Extremal curves are those which render this integral stationary. The minima of the cost functional belongs to this broader class of curves which play a central role in optimal control.

The classical variational calculus studies the variation of this integral under perturbations of the path $q(t)$. One substitutes the initial path $q(t)$ with the new path

$$q_\epsilon(t) = q(t) + \epsilon\delta q(t),$$

where $\delta q(t)$ is an arbitrary vector-valued function on the segment $[t_0, t_1]$.[6] The "fundamental lemma" of the calculus of variations says that if the variation of the integral (I.7) vanishes, then the Lagrange equation (I.6) is verified in the interval (t_0, t_1) (e.g. see [Gol80]).

PMP is the generalization of the Lagrange's problem of the calculus of variation. Given a control system, PMP provides necessary conditions on the inputs (controls) that minimize the cost functional.

5.1.2 Minimum-length paths for car-like robots

Finding optimal paths for nonholonomic robots is a difficult problem in the general case (e.g. see [SB98] for a survey). Regarding car-like robots, the pioneering and seminal efforts of Dubins [Dub57] and of Reeds & Shepp [RS90] were resumed later by some authors (e.g. [BCL92]), using Optimal Control theory. It was found that the original sufficient family of 48 optimal paths, proposed by Reeds & Shepp, could be reduced to 46 (the paths are combinations, including cusp points, of arc of circles of diverse length with straight lines).

These results were later completed by [SL96] to construct a synthesis (or network) of optimal paths for the Reeds & Shepp car. In fact, PMP gives a local characterization of optimal paths. Hence the authors resort to symmetry considerations and global geometric constructions in order to build up solutions steering the robot from any point in $\mathcal{CS}$ to

[5] In Lagrangian mechanics one starts by writing down the Lagrangian of the system under study: $L = T - U$, where T is the kinetic energy and U is the potential energy. Both are expressed in terms of generalized coordinates $(q, \dot{q}) \in \mathbb{R}^n \times \mathbb{R}^n$.

[6] In classical mechanics the notation δq refers to a virtual (infinitesimal) displacement of a system. This one corresponds to a change in the configuration of the system as the result of any arbitrary infinitesimal change of the coordinates δq, consistent with the forces and constraints imposed on the system at the given instant t [Gol80].

the origin. To this end, they determine domains from where the origin is attainable by using members of the family of optimal paths. The boundaries of the domains can be computed analytically. The observation that these domains may overlap led the authors to make new partitions of the free space. These partitions define cells from where the origin is attainable by a single optimal path. There are still however some regions for which several equivalent solutions may be obtained. By choosing a particular solution in those regions one can achieve a synthesis of optimal paths. This construction constituted the first example of a regular synthesis for a nonholonomic system in a 3D space [SB98].

A similar procedure to construct such a synthesis was also applied to the Dubins' car [SL96]. Regarding this system, [BCL94] were interested in the problem of optimal paths when they control the angular acceleration of the car instead of its angular velocity. They considered a bounded curvature derivative constraint and found that the shortest paths are a concatenation of straight lines and of arcs of clothoids.

5.1.3 Discussion

Optimal steering methods like the aforementioned have been successfully used in some nonholonomic motion planners (e.g. see [LJTM94] for a pioneering work). Notwithstanding, finding optimal paths for a given kinematics is a difficult problem. In some cases though, it is possible to sacrifice optimality in the interests of finding reasonable paths in an efficient way.

Over the last decade, theoretical surveys in Robotics and Nonlinear Control have yield criteria allowing for the classification of some nonholonomic control systems. Systems pertaining to these classes possess structural properties allowing for effective solutions of open-loop and closed-loop control issues. The solution process starts by identifying the class to which the control system belongs to. In a second step, coordinate transformations must be undertaken allowing to obtain a "normal form" of the particular class. The interest is that many properties of the original system can be elucidated by using the normal form.

Amongst these classes, we find *nilpotent*, *chained-form* and *flat* systems. We continue this section by discussing how the steering issues can be solved for those control systems belonging to any of these classes.

5.2 Steering nilpotent control systems

In linear algebra a nilpotent transformation is one with a power (degree of composition) that is the zero map. In Nonlinear Control theory, a control system is nilpotent as soon as the Lie products of the input vector fields vanish from some given length. Therefore, a basis of vector fields $\{f_1, \ldots, f_m\}$ is nilpotent if there exists an integer k such that all Lie products of length greater than k are zero; in that case k is called the order of nilpotency.

A distribution (or control system) is said to be *nilpotentizable* if one is able to find a family of vector fields that span the distribution and generate a nilpotent Lie algebra of finite dimension. It is *feedback nilpotentizable* if it can be made nilpotent by a feedback

transformation.

One of the first exact steering methods for nilpotent systems is due to Lafferriere and Sussmann [LS93]. We shall point out the major steps of the method in order to illustrate the interest of transforming a control system into a nilpotent one.

Suppose a small-time controllable system Σ defined by vector fields $f_i(x)$ ($i = 1, \ldots, m$; $x \in \mathbb{R}^n$) is given. The first step consists in defining a basis of the Lie algebra of the system. An example of such a family of linearly independent vector fields is the *P. Hall basis* (see also [MS93] for a few examples).

The approach of [LS93] starts by assuming a holonomic path is given. This path is the *desired motion* for the *extended* system $\{f_1, \ldots, f_m, f_{m+1}, \ldots, f_r\}$. The $\{f_{m+1}, \ldots, f_r\}$ are higher order Lie brackets of the f_i, chosen so that $\{f_1, \ldots, f_r\}$ span $\mathbb{R}^n$ at all x and such that they form a P. Hall basis.

The authors use then the Campbell-Hausdorff formula in combination with P. Hall bases so as to decompose the desired motion in a suitable way. Loosely speaking, the Campbell-Hausdorff formula describes the composition of flows as a single flow (e.g. see [Lau93]):[7]

$$e^{\tau f_1} \circ e^{\tau f_2} = e^{\tau f_1 + \tau f_2 - \frac{1}{2}[f_1, f_2] + \cdots}.$$

It tells us how a nonholonomic system can reach any point in a neighborhood of a starting point. It yields a method for explicitly computing paths in a neighborhood of a point.

Assuming that $B_1, B_2, \ldots, B_s$ is a P.Hall basis, the idea is to convert the desired motion into a flow of the form

$$S = e^{\alpha_1 B_1 + \alpha_2 B_2 + \alpha_3 B_3 + \cdots},$$

where the coefficients α_i are the P. Hall coordinates of S. If the system is **nilpotent**, the Campbell-Hausdorff formula yields an **exact** formulation of S.

The aim is to achieve this flow for the original (non-extended) system. To this end, the authors propose an algorithm where the main idea is to solve separately for each exponential factor, and then concatenate the results. The desired motion is achieved by choosing piecewise constant inputs to generate the flow. Alternatively piecewise-polynomial inputs, yielding a continuous control, allow also to achieve this flow.

The steering method of [LS93] is a general strategy for solving motion planning problems for driftless systems. In the particular case of nilpotent systems the method is exact. For systems that are not nilpotent, the authors consider an iterated utilization of the algorithm. They consider the system as if it were nilpotent of order k and a neighborhood condition to stop the process as close to the goal as wanted.

True, when the system is not nilpotent, but is nilpotentizable, a challenging problem is to find the appropriate transformations. In this respect, a particular class of nilpotent systems corresponds to the *chained form* of nonlinear control systems. This "canonical" form has aroused the interest of the Robotics and Automatic Control communities since its introduction by Murray and Sastry [MS93]. In order to better understand the interest of the chained form we shall discuss the work of [MS93].

[7] see Appendix A for a recall on flows.

5.3 Chained form of control systems

In [MS93], the authors were interested first in steering a particular canonical form of first and second order systems. A first order system is one for which the first level of brackets, together with the input vector fields, span the tangent space at each configuration; in classical mechanics such systems are called *contact structures* (e.g. the unicycle). Systems of higher order are therefore those for which the number of brackets needed to span $\mathbb{R}^n$ is greater than 1 (e.g. the car-like robot).

Previous work by Brockett ([Bro81]) showed that the optimal controls to steer a contact structure, between an arbitrary initial and final configuration, are sinusoids at integrally related frequencies (i.e. $2\pi, 2\cdot 2\pi, \cdots, (m/2)\cdot 2\pi$). Based on this result, in [MS93] the authors propose an algorithm to steer their canonical form using sinusoidal inputs. The idea is to control in a first time the directions directly driven by the inputs, so that the corresponding configuration variables are taken to their final value. Then by iteratively applying sinusoidal controls at integrally related frequencies, it is possible to steer the subsequent variables; and this, making the first ones to execute periodic cycles (see example below).

In order to extend this algorithm to higher order canonical systems with maximal growth, they next turned to a nilpotent form by using P. Hall bases. They soon realized that it would be very difficult to use sinusoids to steer such systems in the general case. Hence they focused on a special subclass of nilpotent systems, which they refer to as the canonical chained form.

The chained form of a two-input nonholonomic control system is given by the following set of differential equations:

$$\begin{aligned}
\dot{z}_1 &= u_1 \\
\dot{z}_2 &= u_2 \\
\dot{z}_3 &= z_2\, u_1 \\
&\vdots \\
\dot{z}_n &= z_{n-1}\, u_1,
\end{aligned}$$

or in a more compact form:

$$\dot{z} = g_1(z)u_1 + g_2(z)u_2,$$

where

$$\begin{aligned}
g_2(z) &= \frac{\partial}{\partial z_2} \\
g_1(z) &= \frac{\partial}{\partial z_1} + z_2\frac{\partial}{\partial z_3} + \cdots + z_{n-1}\frac{\partial}{\partial z_n}
\end{aligned}$$

[MS93] showed that these systems could be steered using sinusoids at integrally related frequencies and give sufficient conditions for finding a coordinate change and a

static feedback[8] putting a nonholonomic system into its chained form.

Example: chained form of the unicycle

The unicycle may be put into the chained form. Indeed, let the state vector be (x, y, θ) and the control input be (v_1, v_2). The system equations read

$$\begin{aligned} \dot{x} &= v_1 \cos(\theta) \\ \dot{y} &= v_1 \sin(\theta) \\ \dot{\theta} &= v_2. \end{aligned}$$

Then the following coordinate change

$$\begin{aligned} z_1 &= x \\ z_2 &= \tan(\theta) \\ z_3 &= y \end{aligned}$$

and the static feedback

$$\begin{aligned} v_1 &= \frac{u_1}{\cos(\theta)} \\ v_2 &= u_2 \cos^2(\theta), \end{aligned}$$

put the system into the chained form:

$$\begin{aligned} \dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1. \end{aligned}$$

Straight computations allow to verify the claim.

Clearly, z_1 and z_2 may be steered independently to their final values at the cost of letting z_3 drift away from its initial value. Then if we apply sinusoidal inputs of the form [MS93]:

$$\begin{aligned} u_1 &= \alpha \sin(\omega \cdot t) \\ u_2 &= \beta \cos(k \cdot \omega \cdot t), \end{aligned}$$

with $k = 1$ and during a period $T = 2\pi/\omega$, a direct integration yields

$$\begin{aligned} z_1(T) &= z_1(0) \\ z_2(T) &= z_2(0) \\ z_3(T) &= z_3(0) + \frac{\alpha\beta}{2\omega}. \end{aligned}$$

[8] A feedback is achieved when some information from the evolution of the state—or from the output if only part of the state is measured—must return to the input. A static feedback is one for which this information is memory-less; i.e. it does not depend on the dynamics of any other variable.

By appropriately choosing α, β it is possible to steer z_3 to its desired value while making the other variables to perform a complete cycle.

In the general case one has [MS93]:

$$\begin{aligned} z_1(T) &= z_1(0) \\ &\vdots \\ z_{k+1}(T) &= z_{k+1}(0) \\ z_{k+2}(T) &= z_{k+2}(0) + \frac{\alpha^k \beta}{k!(2\omega)^k}. \end{aligned}$$

□

The study of chained systems has given rise to significant and ongoing research particularly in the domain of feedback control (see Section 6.1.2). In the domain of motion planning, we can mention the work of Tilbury [TMS95], who addressed the open-loop control of the N-trailer (or general tractor-n-trailer) system. In this case, [TMS95] proposes a single step steering approach, using sinusoidal inputs of the form

$$\begin{aligned} u_1 &= a_0 + a_1 \sin(\omega t) \\ u_2 &= b_0 + b_1 \cos \omega t + b_2 \cos 2\omega t + \cdots + b_{m-2} \cos(m-2)\omega t, \end{aligned}$$

as well as piece-wise constant and polynomial inputs.

We shall also mention the work of Sekhavat [SL98] who gave conditions for steering chained systems, using sinusoids in constrained spaces accounting hence for small-time controllability (see Section 5.5).

The chained form of control systems is closely related to *flatness* (the precise meaning of this assertion will be given in section 7.3). For certain flat systems some practical and efficient steering methods have been proposed in the literature. We discuss now this particular class of nonlinear control systems.

5.4 Steering flat systems

Differential flatness (or flatness for short) finds its application niche in the theory of nonlinear control systems [Mar92]. In 1995, Fliess, Lévine, Martin and Rouchon introduced the concept of flatness in a differential algebraic setting [FLMR95b]. In a later publication [FLMR99], the authors framed flatness within the setting of differential geometry. In these papers, they discuss the fundamentals of flatness.

Loosely speaking, a system

$$\dot{x} = f(x, u) \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m \tag{I.8}$$

is said to be differentially flat (or flat) if there exists a set of differentially independent functions $y = \{y_1, ..., y_m\}$ called the *flat* or *linearizing* output, such that:

- the linearizing output can be expressed as a function of the system state x and the derivatives of the controlling input u:

$$y = h(x, u, \dot{u}, ..., u^{(k)}).$$

- any system variable (state and control) can be expressed only from the elements of the linearizing output and their successive derivatives:

$$x = A(y, \dot{y}, ..., y^{(l)})$$

$$u = B(y, \dot{y}, ..., y^{(l)}).$$

Thus it is possible to compute trajectories for system (I.8) from y, without integrating any differential equation; as the elements of y are free, any trajectory $y = y(t)$ determines a trajectory $(x(t), u(t))$ and vice-versa.

The striking advantage of flatness comes therefore from this fact: the trajectories of the original control system can be computed from arbitrary time-laws of the free (unconstrained) components of the flat output; and this without solving any differential equation. This property is what makes flatness attractive for motion planning purposes—i.e. for the open-loop control of the system—especially when we are faced with complex systems.

One example of this concerns the work of Rouchon *et al* [RFLM93a]. In this note, flatness was exploited as a useful property to plan paths for a mobile robot with n trailers. For this system, the coordinates of the middle point of the n^{th} trailer axle are the linearizing outputs[9]. This means that any feasible path in the system configuration space $\mathbb{R}^2 \times S^1$ can be deduced from any sufficiently smooth path followed by the trailer reference point in the workspace (i.e. $\mathbb{R}^2$). Indeed, as soon as a smooth path for the trailer is obtained, its absolute orientation is the direction of the tangent vector to the path and its relative angle (w.r.t. the next trailer or the robot direction) can be deduced from the curvature of the path [RFLM93a].

In chapter II, we shall discuss in more detail how motion planning problems have been addressed using flatness. Suffice it to say for now, that flat outputs allow to use parameterized curves (e.g. polynomials) so as to solve path planning problems. The only constraints that apply are the initial and goal configurations and their derivatives.

The steering method of [RFLM93a] did not take into account the possibility to provide cusp points—i.e. the possibility to execute maneuvers. Yet this aspect is of particular importance in the presence of obstacles. Indeed, if the space to maneuver is very cluttered, there is an evident interest in reaching locally (i.e. without too much deviating from the current configuration) nearby configurations of the free space. Small-time controllable robots in general allow for this local maneuvering capability. Exploring it is essential in order to achieve a *complete* motion planner. In this case, obstacle avoidance is a problem of a topological nature as discussed below.

[9]The utilization of the term *output* in the plural here means only that the set defining the linearizing output contains more than one element.

5.5 Steering with obstacles: topological considerations

5.5.1 Small-time controllability and topological issues

Recall a nonholonomic robot can directly move only in some directions. Hence taking small-time controllability into account requires deep analysis. Indeed, two configurations close to each other in the Euclidean space may be at an infinite distance in the configuration space; i.e. inaccessible the one from the other. This is due to the fact that nonholonomic constraints partition the configuration space into disconnected submanifolds. As a result, the topology in the configuration space may have little resemblance to the one induced by the Riemannian metric in the Euclidean space. Holonomic systems do not have this problem. Indeed, since any smooth path in the configuration space is an admissible path in the Euclidean space, the topologies of both spaces remain the same: the Riemannian distance is a natural one for both [Lau93]. For nonholonomic systems the notion of *distance* in the configuration space becomes crucial. The analysis of the motions of nonholonomic systems can lead to metrics inducing finer topologies. Once the metric has been defined, the key issue is to analyse the nature of the associated topology.

The first work rising this problem comes from Laumond *et al* [LJTM94]. They used the arc length of optimal paths for car-like robots in order to define a metric. A ball of radius r corresponding to this metric is the set of all the points in the configuration space reachable by a path of length (or cost) lesser than r. This metric is sub-Riemannian. Laumond *et al* show that the associated topology is the same as the Euclidean one. This fact entailed a topological property allowing to determine reachable sets in the presence of obstacles by using optimal paths.

Later, Sekhavat and Laumond [SL98] proposed a more general topological property. We recall now the main results.

5.5.2 Topological property

Let $q = (x_1, x_2, \ldots, x_n)$ be a point in the configuration space $\mathcal{CS}$. Let $\mathrm{d}_{\mathcal{CS}}$ be the following distance over $\mathcal{CS}$:

$$\mathrm{d}_{\mathcal{CS}}(q_1, q_2) = \sum_{i=1}^{n} |x_i^1 - x_i^2|$$

where x_i^j stands for the i-th coordinate of q_j.

The set of configurations q_2 such that $\mathrm{d}_{\mathcal{CS}}(q_1, q_2) < \epsilon$ is denoted by $B(q_1, \epsilon)$; this is the ball centered at q_1 of radius ϵ.

Let $\mathcal{C}$ be the set of feasible paths defined over an interval of the type $[0, T]$. A steering function $Steer$ is a mapping from $\mathcal{CS} \times \mathcal{CS}$ into $\mathcal{C}$:

$$(q_1, q_2) \rightarrow Steer(q_1, q_2)$$

where $Steer$ is defined over the interval $[0, T]$, such that $Steer(q_1, q_2)(0) = q_1$, $Steer(q_1, q_2)(T) = q_2$.

Definition 5.1 (Topological Property [SL98]) *Steer verifies the topological property iff:*

$$\forall \epsilon > 0,\ \exists \eta > 0,\ \forall (q_1, q_2) \in (\mathcal{CS})^2,$$
$$\mathrm{d}_{\mathcal{CS}}(q_1, q_2) < \eta \Rightarrow \forall t \in [0, T],\ \mathrm{d}_{\mathcal{CS}}(Steer(q_1, q_2)(t), q_1) < \epsilon$$

This is a global property that not only takes into account small-time controllability but also holds uniformly everywhere [LSL98]. It says that for each neighborhood $B(q_1, \epsilon)$ of a configuration q_1, there exists a neighborhood $B(q_1, \eta)$ such that for every configuration $q_2 \in B(q_1, \eta)$, the path corresponding to $Steer(q_1, q_2)(t)$ remains within $B(q_1, \epsilon)$.

Now let us equip $\mathcal{C}$ with a metric $\mathrm{d}_{\mathcal{C}}$ between paths defined as follows. Γ_1 and Γ_2 being two paths on $[0, 1]$, $\mathrm{d}_{\mathcal{C}}(\Gamma_1, \Gamma_2) = \max_{t \in [0,1]} \mathrm{d}_{\mathcal{CS}}(\Gamma_1(t), \Gamma_2(t))$. A sufficient condition for a steering method to verify the topological property **TP** is given in [SL98]:

1. *Steer* must be continuous w.r.t. the topology associated with $\mathrm{d}_{\mathcal{C}}$ and
2. the path $Steer(q, q)$ must be reduced to the point q.

One can find in the literature a few works addressing the motion planning issue in the presence of obstacles (e.g. see [LSL98] for a comprehensive survey). In this context, a general motion planning scheme (see Chapter III), taking into account the topological property above, was applied to the tractor-trailer system: using a local planner based on sinusoids [SL98], and another based on the flatness of the tractor-trailer [SLL+97], [LSL99].

6 Feedback control of nonholonomic systems

Up to this point, we have reviewed some specific ways of steering nonholonomic systems. Motion (trajectory) planning and open-loop control are essentially synonyms as opposed to feedback control. However, the planning and control phases are closely related during the execution of a given task.

In general, a closed-loop controller results from the superposition of a feedback action to a coherent open-loop (feedforward) term[10]. In the planning phase, the resulting feedforward commands are computed off-line, according to a priori knowledge of the motion task and the state of the environment. If we tried to perform this trajectory without taking care of the actual evolution of the robot, unmodeled events at running time (e.g. occasional slipping of the wheels or erroneous initial localization) would prevent the successful completion of the task. Feedback controllers, driven by the current task error, are therefore intended to overcome these problems so as to achieve some degree of robustness. To this end, real-time sensor measurements are used to reconstruct the robot state. This information is sent back to the controlling input so as to undertake a suitable action. Thus *feedback laws* or *compensators* aim at computing the action best adapted for a particular control task.

Control tasks usually encountered in Robotics fall into one of the following categories: stabilization about a point, path following and trajectory tracking. The nonlinear nature of nonholonomic robots is at the origin of the main difficulty for synthesizing suitable feedback laws. The simplest problem in which this difficulty may be understood is the stabilization to an equilibrium point. Paradoxically, in theoretical control this problem is the hardest to solve regarding nonholonomic robots. It is also called the *regulation* or *stability* issue. It consists in making an equilibrium point of the system (conventionally the origin) locally (or globally) asymptotically stable[11]. There are indeed other control problems which are simpler than this one. In particular *output stability* (where there is a given function $y = h(x)$ and one wishes to make $y(t) \to 0$ using suitable control actions) and trajectory tracking (i.e. the problem of making $y(t)$ follow a reference trajectory $y^*(t)$), which can be seen as an output stability issue for an "error" signal $e(t) = y(t) - y^*(t)$.

The next sections make a literature review about feedback control strategies in order to cope with the aforementioned control tasks. We first discuss main obstructions to stabilize nonholonomic systems. Subsequently, we review solutions to the path following and trajectory tracking problems regarding nonholonomic wheeled robots.

We should emphasize that the most important task to us, from an application point of view, is trajectory tracking; more especially since we assume that a nominal precomputed trajectory from the planning phase will be given. Hence we should concentrate much more our discussion on solutions to this rather than the other two problems.

[10] However a feedback-like action may be achieved with only open-loop commands. Indeed, we may use repeated open-loop phases re-planned at higher rates using new sensor data to gather information on the actual state. In the limit, continuous sensing and re-planning leads to a feedback solution [DLOS98].

[11] see Appendix C page 154 for a brief recall on this terminology.

6.1 Stabilization of nonholonomic control systems

A natural question in theoretical control concerns the possibility to extend results from linear control theory to nonlinear systems. It is a well known fact that any linear control system which is controllable can be asymptotically stabilized by means of continuous state feedback laws [Kai80, Chapter 3]; that is, the open-loop property (controllability) implies that the closed-loop one (stability) can be achieved. Research work in Nonlinear Control has revealed that this is not any more the case for nonholonomic robots. This was shown by Brockett [Bro83] who proved that nonholonomic systems fail to be (locally) stabilized by means of smooth (or even continuous) static (time-invariant) state feedbacks.

Let us recall the condition given by Brockett.

6.1.1 Obstructions for stability

The problem of stabilization about the origin by means of a regular static state feedback is the following. Given a control system

$$\dot{x} = f(x,u), \quad f(0,0) = 0 \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m,$$

the problem is to find a feedback law

$$u = k(x), \quad k(0) = 0,$$

making the closed-loop system

$$\dot{x} = f(x, k(x))$$

asymptotically stable about $x = 0$. One says that the feedback law k is regular if it is smooth (or of class $\mathcal{C}^\infty$) on $\mathbb{R}^n \backslash \{0\}$[Son99].

The famous condition of Brockett for the existence of such a feedback law may be restated as follows [Son99]:

Theorem 6.1 (Brockett [Bro83]) *If there is a stabilizing feedback which is regular and continuous at zero, then the map $(x,u) \mapsto f(x,u)$ is open at zero.*

In other words, the image $f(x,u)$ contains a neighborhood of 0.

Nonholonomic systems fail to verify this condition as the cart example (unicycle robot) shows. To illustrate this, consider a cart robot having a reference frame placed at the middle point of the wheels axle. Its position in the plane and its orientation are represented by the state vector (x_1, x_2, θ). The controls (u_1, u_2) are respectively the linear and angular speeds. We have seen that the system dynamics may be modeled by:

$$\begin{aligned} \dot{x}_1 &= u_1 \cos(\theta) \\ \dot{x}_2 &= u_1 \sin(\theta) \\ \dot{\theta} &= u_2. \end{aligned}$$

Then the test fails, since there are no points of the form $(0, \varepsilon_1, \varepsilon_2)$ that belong to the image of the following map

$$\mathbb{R}^5 \to \mathbb{R}^3 : (x_1, x_2, \theta, u_1, u_2) \mapsto f(x,u) = (u_1 \cos(\theta), u_1 \sin(\theta), u_2)$$

for $\theta \in (-\pi/2, \pi/2)$, unless $\varepsilon_1 = 0$.

More generally, it is impossible to continuously stabilize any system without drift

$$\dot{x} = f_1(x)u_1 + \cdots + f_m(x)u_m$$

if $m < n$ and $rank\,[f_1(0), \ldots, f_m(0)] = m$ (this includes all totally nonholonomic mechanical systems) [Son99].

The stabilization of nonlinear systems in general is an active research area (see e.g. [LS98] for a new approach on stability and [Cor98] for open problems on feedback stabilization). The negative result given by Brockett is at the origin of work leading to effective solutions to the problem. Let us discuss some of the current approaches.

6.1.2 Some approaches to the regulation problem

In view of the obstacle pointed out by Brockett, two solutions have been proposed to solve the regulation problem under nonholonomic constraints.

On the one hand, early stabilizing controllers using discontinuous feedbacks were discussed in Bloch *et al*[BMR90]. Canudas de Wit and Sørdalen [CdWS92] proposed piecewise continuous laws to stabilize unicycle robots.

On the other hand, an alternative solution to the stability problem consists in using periodic time-varying feedbacks. Pioneering work in this field is due to Samson [Sam91, SAA91b]. Furthermore, results obtained from Coron [Cor92, Cor94] showed that driftless systems can be stabilized globally and asymptotically by means of time-varying control laws.

A considerable amount of research in this direction has been published, particularly regarding the chained form of control systems. Let us mention for instance the work of Sørdalen and Egeland [SO95]. They combined time-varying and non-smooth state feedbacks to achieve a control law exponentially stabilizing a nonholonomic two-input chained system of arbitrary dimension.

Samson proposed time-varying laws for linearized two-input chained systems [Sam95]. They achieve a polynomial asymptotic rate of convergence. To improve this, Morin and Samson [MS97] use backstepping techniques (Lyapunov-based design of robust feedback laws), achieving exponential stabilization with a lower-bound estimate of the asymptotic rate of convergence.

We now discuss a probably more important problem from an application point of view: the stabilization of mobile robots around a reference path.

6.2 Path following and trajectory tracking

6.2.1 Path following

In the path following task the controller is given a geometric description of the desired motion in the Cartesian space. This information is usually available through a path parameter σ, which can be in particular the arc length along the path. For this task, one is only interested in the geometric gap between the robot and the path. Hence time dependence is not relevant: the time evolution of the path parameter is usually free. Accordingly, the command inputs can be arbitrarily scaled with respect to time without changing the robot path. In this context, it is customary to set the input corresponding to the robot forward velocity to an arbitrary constant or time-varying value. Hence the other input (e.g. the angular velocity) is available for control. Thus the path following problem consists in stabilizing to zero the geometric gap between the robot and the path.

Typical approaches consist in building controllers issued from a geometric analysis of the problem (see e.g. [DLOS98]).

6.2.2 Trajectory tracking

On the other hand, a trajectory is a geometric path with an associated timing law. In the trajectory tracking task the robot must follow the desired Cartesian path with a specific velocity profile; in other words the problem is equivalent to tracking a robot moving along the nominal path. In this case the objective is to stabilize an error in position and orientation using the two control inputs.

Several approaches to this task can be found in the literature. Let us discuss some of them.

Geometric approach The first closed-loop controllers, from [KKMN91] and [SAA91a], rely on a geometric reasoning about the problem (in a very similar way to what is done in the path following context).

In [KKMN91], they introduce a Lyapunov function to determine the structure of the controller. It stabilizes the error in position and orientation, with respect to a "reference" robot moving with nominal linear and angular speeds. The controller can be used for many nonholonomic mobile robots in the sense that it gives linear and angular control speeds. Indeed, the authors assume intermediate blocks between the controller and the robotic system. This allows for converting general speed commands into robot-specific controls.

In [SAA91a] they propose a particular control law to stabilize a cart robot (unicycle) around a reference trajectory. However, the feedback law may fit as well to other systems, as it has been demonstrated in particular for the tractor-trailer robot HILARE [LSL99].

Tracking via linearization Another technique consists in linearizing the system either approximatively or exactly.

An example of a general control law issued from the former approach is [WTS+92]. They first compute the linear approximation of the system about a point pertaining to the trajectory. Then they propose a controller that locally exponentially stabilizes the resulting linear time-varying system. The control law uses a linear state feedback, whose construction involves an integral function, similar to the controllability matrix of the linear system.

On the other hand, exact linearization is especially interesting in that the nonlinear system may be actually considered as a linear one, with respect to different coordinates. The idea is to search for state transformations and some kind of feedback in order to obtain a linear system. This approach is known as *feedback linearization*. We shall discuss this technique in more detail since our contribution concerns the *exact* feedback linearization of the bi-steerable car (see Chapter IV).

Feedback linearization approach Pioneering work on feedback linearization is due to Sampei *et al* [STIN91]. They used coordinate transformation and time-scaling ([SF86]) techniques to linearize the tractor-trailer in order to make it track a straight line. The approach consisted in studying the dynamics of the middle point of the rearmost (trailer) axle. The idea was to parameterize the system dynamics with respect to the x coordinate of that point instead of the time. The objective is to make the robot track a "running" point traveling along the x axis. A linearizing coordinate transformation and a static feedback entails a linear dynamics with respect to the coordinate x. Thus the authors proposed some control laws ensuring the stability of the system for this particular case.

In the same order of ideas, *dynamic* feedbacks[12] began to be explored also in the early 1990's with the work of d'Andréa-Novel, Bastin and Campion [dNBC92]. Their solution was based on results from [JR80] (see Chapter IV). [dNBC92] proposed "linearizing outputs" and dynamic compensators in order to achieve full linearization of wheeled mobile robots, including the car-like (tricycle model). This technique was also used in [dNMS92], where dynamic compensation achieved full linearization of a rigid manipulator having less controls than articulated joints. We shall discuss this technique in more detail in Chapter IV, where the interest should become more clear.

In the general case, the problem of finding a dynamic feedback entailing full linearization is a difficult one (see e.g. [CLM91], [MR94] for few significant results). However, flatness characterizes a particular class of nonlinear systems that can be linearized by a dynamic feedback called *endogenous* (meaning that the original—endogenous—variables of the system are transformed without creation of new *exogenous* variables).

An interesting observation, pointed out in [FLMR99], is that even if there are tight links relating flatness and dynamic feedback linearization, the two concepts are distinct. Indeed, flatness is a property of the trajectories of a system. It does not imply that the system ought to be turned into a linear one. However, when a system is flat it implies that its nonlinearity is well characterized and that this structure may be exploited in the control design.

[12] Unlike static feedbacks, a dynamic feedback does depend on the evolution in time of some other variable.

In view of these results and those discussed regarding the motion planning problem, it follows that the most useful properties of flatness and flat outputs are their immediate utilization for trajectory generation and tracking. We shall discuss further this integral character of flatness in Section 7.

Feedback linearization of nonholonomic robots The linearizability structure properties of flat systems has led to feedback strategies solving tracking problems for car-like and tractor-trailer systems.

Exact feedback linearization of flat systems, including the tractor with n-trailers and car-like robots, was first addressed in references [RFLM93a] and [FLMR95a]. Using the same approach, the tracking problem was also solved for the general (off-hooked) tractor-trailer and presented in [FLM$^+$97].

In all these cases, the tracking problem is solved via endogenous feedback linearization.

7 Conclusions and discussion

At this stage, we are ready to derive some conclusions and establish the grounds justifying the approach we have chosen to solve our problem. We recall the problem we are interested in is the following.

- How to plan and execute collision-free motions for a bi-steerable robot.

As far as this problem is concerned we now make some concluding remarks, issued from our analysis of the state of the art.

7.1 Mobile robots found in the literature

To our knowledge, the nonholomic wheeled robots formally classified in Robotics (either as nilpotent[13], chained or flat), until present, include: the unicycle robot, the car-like (which could be seen as an extension of the latter), the single steering-tractor pulling multiple trailers— STMT (an extension of the car-like robot), the multiple steering-tractors and multiple trailers— MTMT (the most simple case of which is the fire truck [BTS95]), and the single tractor and single off-hooked trailer— STSOT.

For those systems that do not classify in any of the structures previously discussed, some research work has been done so as to find a nilpotent approximation for them. This is for instance the case in [VLO97], where a constructive method for finding a nilpotent approximation of the tractor with two off-hooked trailers is given.

Regarding double-steering vehicles, most of the research has focused on control schemes, considering the dynamics of the wheels-ground contact [SFS86], [Whi90]. A

[13] nilpotentizable

path-following controller based on the kinematics of the center of mass was proposed in [Hem94]. In [WQ01], a particular strategy is envisaged to compute the velocity profile along a (a priori given) collision-free path for the center of mass.

7.2 Flatness: an integral approach to motion planning and control

Even if a general approach such as optimal control allows to solve many motion planning problems, a great majority of these has not a practical solution. However, many (computationally) effective solutions exist for nilpotent, chained and flat systems.

In this context, flatness offers an integral approach to motion planning and feedback control, in the sense that we can solve both steering and feedback control issues. Not only that, but also we can view flat systems as a generalization of chained-form systems. The following results reported in the literature account for this.

7.3 Flatness as a formal framework for system classification

The concept of flatness makes part of the research work aiming at characterizing nonlinear systems which are *equivalent* to linear ones by means of dynamic feedback (see e.g. [FLMR99]). The question about equivalence of systems of differential equations has gained in recent years considerable attention (see [W.F90], [NRM95] for two examples) since the early results from Elie Cartan about *absolute equivalence* [Car14]. Fliess and co-workers introduced flatness [FLMR95b] through the notion of *endogenous equivalence* [Mar92]. This general notion has provided a formalization framework of system classification and linearization by a restricted kind of dynamic feedbacks called endogenous (see Chapter IV). In this setting, the simplest class is made up of flat systems [FLMR99]. In particular, a flat system is linearizable by means of dynamic feedback and coordinate change. In view of this, there is an elementary reason why we should address our problem regarding of the bi-steerable car in the framework of flatness: a complete and formal system classification can be achieved.

Fundamental work on flatness has turned around its characterization. Necessary and sufficient conditions for flatness have been given by Rouchon [Rou94] and Martin and Rouchon [MR95], [MR94] for a wide range on nonlinear systems. In this latter reference, an important result is a necessary and sufficient condition for two-input driftless systems to be dynamic feedback linearizable and especially flat. A direct consequence of their result is that such a system can be converted, around every point of a dense open subset, into a chained-form system using only static feedback [MR94]. Extra regularity conditions are given by Murray [Mur94], who gives a necessary and sufficient condition for putting a two-input driftless system into chained form around a *given* point by static feedback.

Notwithstanding, the hard problem is to obtain the coordinate transformations required either to find the flat output or to put the system in its chained form. For two input driftless systems this problem is equivalent for either transformations. Indeed, the chained form of a two-input driftless control system contains a single chain. Observing this chain, one can notice that the trajectories of the first and last coordinates ($z_1(t)$ and

$z_n(t)$) completely define all the state variables of the chained-form system: we only need to differentiate these functions so as to find the subsequent "chained" coordinates. It follows that z_1 and z_n are the flat outputs of the system. As a consequence a coordinate change transformation is completely defined by the first and last coordinates of the chain as functions of the original coordinates.

7.4 Methodological tools

On the other hand, there is no unicity in the choice of the flat output and therefore there are many possible transformations into chained form. Finding a flat output demands a methodological approach. However, the literature includes very few works on the effective computation of such transformations and this remains a difficult problem in general. The rare contributions (e.g. [MR94], [TMS95], [Pom97]) resort to methodological tools from the theory of *Pfaffian* systems [BCG+91].[14] Notwithstanding, in the majority of situations a certain amount of physical insight about the system and guess work are required.

7.5 Concluding remarks and contributions

In this context, a classification effort of the bi-steerable system would prove to be useful. Indeed, if we characterize the kinematics of the bi-steerable car, not only our understanding of the system will increase, but also we open the possibility to solve our problem. The questions we aim at answering are the following. What are the structural properties of the bi-steerable kinematics? How to find the convenient transformations yielding the flat coordinates or the chained form? And how to exploit them so as to solve the motion-planning and feedback control problems for such a system?

The purpose of the forthcoming chapters is therefore to introduce the kinematics of the bi-steerable robot and to show that it is a differentially flat system. In this respect, we shall point out the main difficulty in finding a flat output. It is our purpose to give some methodological guidelines in order to compute a particular flat output for controllable systems of dimension 4 and subject to 2 nonholomic constraints. In doing this, we will show that taking into account symmetry considerations will be instrumental in finding a flat output for the bi-steerable car. Moreover, further analysis will lead to equivalence relations that will complete our survey on the flatness property of the system. All these issues allowed us to solve our problem as we will show with simulations and experimental results.

Remarks

Nonholonomic assumptions. One could expect that the modeling assumptions made here— in particular the rolling without slippage— are valid to a certain extent. This at least insofar as one is able to remain within the conditions for which these assumptions are

[14] A Pfaffian system is a set, assumed to be of constant dimension, of linearly independent differential equations. In mechanics, Pfaffian systems arise naturally from the kinematic constraints.

plausible (e.g. low speed). Throughout this document therefore, we pretend to establish results on models that, although remain far from the reality, allow for reasoning on properties that will have to be validated in an experimental phase. Thence the uncertainty related to these models will have to be taken into account in a specific way (see Chapter V).

Bounded curvature constraints for mobile robots. An additional kinematic constraint in mobile robots is when the turning radius is bounded. This induces constraints on the curvature (and may be its derivatives) of the path followed by the robot. Typically car-like robots are subject to this kind of constraint, for the steering wheel is usually subject to mechanical stops. This constraint may also apply for two-wheel drive robots if it is not allowed to turn around the Z-axis.

Adding constraints of this kind increases the complexity of the path planning task and in particular the formulation of the optimal paths under such constraints [SB98]. In this respect, suboptimal solutions based on clothoids ([Sch98]) and in general using *cubic spirals* ([KH89, KH97]) have been explored for the car-like robot. These efforts aimed at increasing the tracking performance (avoiding the robot to stop) as well as reducing slippage due to unwelcome curvature discontinuities.

We shall indeed consider bounded steering angles regarding our kinematic analysis about the bi-steerable car. However we had not addressed the motion planning issue from the point of view of the aforementioned references.

Chapter II

Introduction to the Bi-steerable Kinematics

The purpose of this chapter is twofold. Firstly, the aim is to introduce the motivations that aroused our interest for studying the bi-steerable kinematics, beyond the fact that it is an innovative mechanical structure. To this end, we shall discuss the kinematical properties of the bi-steerable system.

On the other hand, the material presented here should account for the solution we propose for the complete motion planning and the feedback control problems, presented in subsequent chapters. To this end, we first establish some connections between typical approaches for solving the path-planning problem in the context of flatness. This shows how the solutions to the problem resort to the properties of the flat output. A natural question is thence to ask whether the bi-steerable system is flat and what is the flat output. In this respect, we shall introduce the flatness property of the control system associated to the kinematics model of the bi-steerable car. We then discuss first attempts to find a flat output for this control system. This shall reveal that the problem is far from being obvious, justifying a deeper analysis.

1 Introduction

During the last two decades, vehicles with double steering ability have aroused the attention of several research groups. Amongst these, there are some researchers—including car-constructor's laboratories—particularly interested in vehicle response quality from a dynamics dynamics point of view [SFS86, Whi90, Hem94, WQ01]. On the other hand, prototype platforms showing double steering capability have been developed aiming at exploring new ways of intelligent transportation systems[1].

The fact that car constructors have been interested in the double steering feature

[1] Examples of these platforms are the *Cycab* robot, designed at INRIA in France, and the $\pi - Car$ prototype of IEF (Institut d'Electronique Fondamentale, Université Paris-Sud).

indicates that the enhancements in the vehicle performance are substantial. But what makes a bi-steerable car so attractive? How does it compare to a conventional car-like robot? We will give some answers to these questions by turning to the geometric models for these two vehicles. These models will lead our discussions throughout this dissertation. Notwithstanding these models are great simplifications of the real systems, they allow for a detailed analysis of their structural properties.

In this respect, we know the car-like robot is flat and that solutions to the motion planning problem in the context of flatness resort to the properties of the flat output [FLMR95b]. Motivated by the fact that the car-like and the bi-steerable kinematics are seemingly close, a natural question is to ask whether the bi-steerable system is flat and whether a flat output could be intuitively found. In this respect, we show the control system associated to the bi-steerable kinematics is flat. This opens effectively the possibility to exploit existing solutions to the motion planning and feedback control problems. However in order to exploit these solutions, the flat output must be found. Hence we will discuss first attempts in finding a flat output for the bi-steerable car. It soon appears that the richer kinematics of the bi-steerable system entails further complexity in the analysis of the problem and in the synthesis of its solution.

In the sequel we will also refer to the term *bi-steerable* by using the acronym BiS.

2 Kinematics of the conventional car-like robot

The car-like vehicle has been the canonical case-study in nonholonomic motion planning and control research for the last two decades [Lat91b],[LSL98].

Typically, the system moves with respect to a global Cartesian frame $\{X, Y, Z\}$. The configuration space $\mathcal{CS} = \mathbb{R}^2 \times S^1$ has local coordinates (x, y, θ), where (x, y) are usually the coordinates of one of the middle points of an axle and θ is the heading of the robot (i.e. the orientation of the body with respect to the X-axis). A typical geometric representation for this kind of robot is given in Figure II.1.

This geometric model relies on results from classical mechanics. The latter tells us that the perpendicular lines to the velocity vectors of all points of the solid should coincide at a single point, namely the instantaneous gyration centre G. Hence, assuming the system is moving in the plane $X-Y$ with angular velocity $\vec{\Omega} = \dot{\theta} \cdot \vec{k}_Z$ (where $\|\vec{k}_Z\| = 1$), the following relation holds for any point A pertaining to the solid:

$$\vec{V}_A = \vec{\Omega} \times \vec{GA} \Rightarrow \|\vec{V}_A\| = |v_A| = |\dot{\theta}| \cdot |\overline{GA}|. \tag{II.1}$$

At this stage, the reader may have noticed that, in order to have a unique intersection point G, the front wheels must have different attitudes—i.e. different deviation angles. This translates into a geometric relationship called the Alexander-Maddocks condition [AM89]. However, for modeling purposes, the steering angle φ is assumed to be applied to an imaginary wheel placed at point F. Hence simplifying assumptions lead to consider this steering angle φ equal to the median angle between the left and right wheels. Furthermore, the rolling without slipping assumption makes the velocity vectors $\vec{V}_R$ and $\vec{V}_F$ to "point" in the direction collinear to each imaginary wheel respectively. As a consequence, a

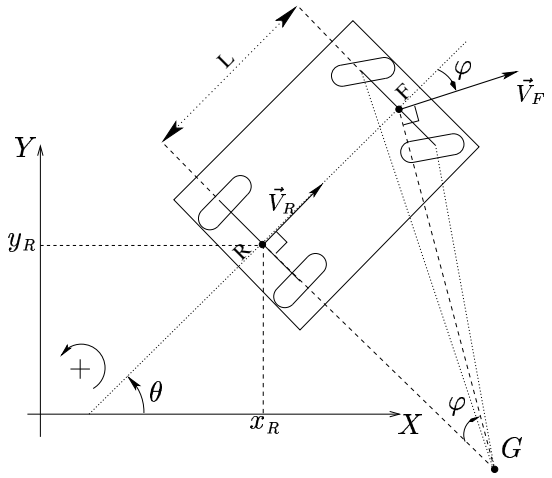


Figure II.1: The geometric model of a car-like robot

typical variation of the geometric construction II.1 is the so called "bicycle" (as opposed to unicycle) model, where the analysis is focused on these two imaginary wheels.

Figure II.1 agrees with the modeling of the nonholonomic constraints imposed by the rear and front wheels. Thus from this geometric construction and from classical mechanics results one can derive the kinematics model of the system. This is a customary practice in many references, where the motion kinematics of the car-like robot is given by well-known systems of differential equations (e.g see [Lat91b]). We recall now the car-like kinematics models.

Typical geometric constructions allow for placing a robot reference frame either at point R or F. Hence the controls of the robot are usually the steering angle φ and the linear speed of either points R (v_R) or F (v_F). A further assumption is that the steering angle φ is bounded. In this respect, typical range values for φ are $]-\pi/2, \pi/2[$.

In this setting, the kinematics model of this vehicle is given by system (II.2), when the robot reference frame is placed at point R:

$$\begin{cases} \dot{x_R} &= v_R \cdot \cos(\theta) \\ \dot{y_R} &= v_R \cdot \sin(\theta) \\ \dot{\theta} &= v_R \cdot \frac{\tan(\varphi)}{L}. \end{cases} \tag{II.2}$$

If now we place the robot reference frame at point F the system becomes (II.3):

$$\begin{cases} \dot{x_F} &= v_F \cdot \cos(\theta + \varphi) \\ \dot{y_F} &= v_F \cdot \sin(\theta + \varphi) \\ \dot{\theta} &= v_F \cdot \frac{\tan(\varphi)}{L}. \end{cases} \tag{II.3}$$

3 Kinematics of the Bi-Steerable car

Let us now turn to the kinematics of a BiS-car. We proceed in a similar way than for the car-like case.

Figure II.2 represents the geometric model of a BiS-car. This geometric construction is inspired from the one presented in [BGMPG99] for a particular bi-steerable car, where the function f is a constant of proportionality—i.e. $f(\varphi) = k \cdot \varphi$. Geometric model II.2 will be our support throughout the discussion below.

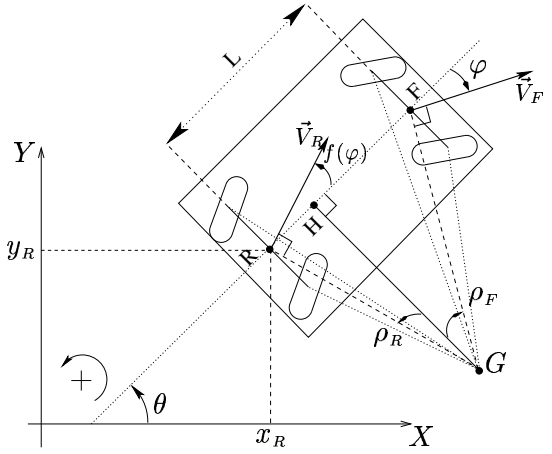


Figure II.2: Kinematics of a BiS-car

Making the same assumptions of no slippage of the wheels, the orientations of the velocities of points R and F are respectively given by the rear and front steering angles. Hence, we deduce the position of the instantaneous turning center of the solid, represented by the point G. We call H the orthogonal projection of G on the main axis of the vehicle. The reader should keep in mind this point as we will come back to it in Section 5.

From the geometric representation of Figure II.2 we have the following algebraic expressions:

$$\begin{cases} \overline{RH} = -\overline{HF} \cdot \frac{\tan(f(\varphi))}{\tan(\varphi)} \\ \overline{RH} + \overline{HF} = L \end{cases} \tag{II.4}$$

which give:

$$\begin{cases} \overline{RH} = -L \cdot \frac{\cos(\varphi)\cdot\sin(f(\varphi))}{\sin(\varphi - f(\varphi))} \\ \overline{HF} = L \cdot \frac{\cos(f(\varphi))\cdot\sin(\varphi)}{\sin(\varphi - f(\varphi))} \end{cases} \tag{II.5}$$

where L is the distance between the back and front axles, also called the longitudinal wheels base.

These expressions allow us to find relations for ρ_R and ρ_F, distances of points R and F respectively to point G:

$$\begin{cases} \rho_R = L \cdot \left| \frac{\cos(\varphi)}{\sin(\varphi - f(\varphi))} \right| \\ \rho_F = L \cdot \left| \frac{\cos(f(\varphi))}{\sin(\varphi - f(\varphi))} \right| \end{cases} \tag{II.6}$$

Again, equation (II.1) from classical mechanics leads to:

$$|\dot{\theta}| = \frac{|v_R|}{\rho_R} = \frac{|v_F|}{\rho_F} \Rightarrow |v_R| = |v_F| \cdot \frac{\rho_R}{\rho_F} = \left| v_F \cdot \frac{\cos(\varphi)}{\cos(f(\varphi))} \right| \tag{II.7}$$

Assuming furthermore that the steering angle is bounded within $]-\pi/2, \pi/2[$, these relations enable us to establish the kinematic model of the bi-steerable robot. With a reference frame located at the mid point of the rear axle (point R) we have:

$$\begin{cases} \dot{x_R} &= v_R \cdot \cos(\theta + f(\varphi)) \\ \dot{y_R} &= v_R \cdot \sin(\theta + f(\varphi)) \\ \dot{\theta} &= v_R \cdot \frac{\sin(\varphi - f(\varphi))}{L \cdot \cos(\varphi)}. \end{cases} \tag{II.8}$$

Alternatively, the kinematics model of the BiS-car with a reference frame located at point F is:

$$\begin{cases} \dot{x_F} &= v_F \cdot \cos(\theta + \varphi) \\ \dot{y_F} &= v_F \cdot \sin(\theta + \varphi) \\ \dot{\theta} &= v_F \cdot \frac{\sin(\varphi - f(\varphi))}{L \cdot \cos(f(\varphi))}. \end{cases} \tag{II.9}$$

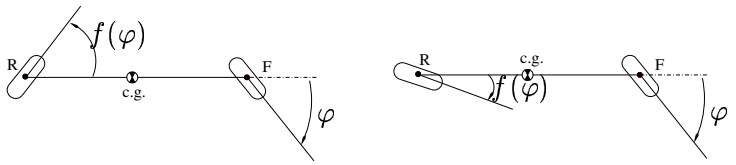
Notice the close similarity between models (II.8)–(II.9) and (II.2)–(II.3).

We will come back to this issue in Section 5. For now, let us concentrate on the improvements of the bi-steerable kinematics compared to those of the conventional car. This will account for the growing interest in favour of the former against the latter.

4 Maneuverability enhancements in BiS-cars

The possibility to steer the rear wheels as a function of the front wheels entails two kinds of rear-to-front wheel attitudes: deflection in an opposite or similar direction as shown in Figure II.3. The convenience of having either attitude at the rear wheels stems from an increased maneuverability. A brief explanation of the reasons behind this assertion may be given here.

The maneuverability enhancements may be studied from a dynamics or a kinematics standpoint. Even though the kinematics point of view is more in connection with motion planning concerns, the dynamics point of view shows how rear wheel steering changes the



(a) Opposite rear-to-front attitude (b) Similar rear-to-front attitude

Figure II.3: Two possible rear wheels steering directions with respect to front steering.

vehicle response at low and high speeds (e.g. see [Whi90]). In this respect, some surveys (e.g. [SFS86], [Whi90]) are illustrative of both the benefits and technical difficulties of rear wheel steering, either alone or in combination with front steering.

We shall start by discussing the advantages of double steering and outline some of the underlying problems from a dynamics point of view.

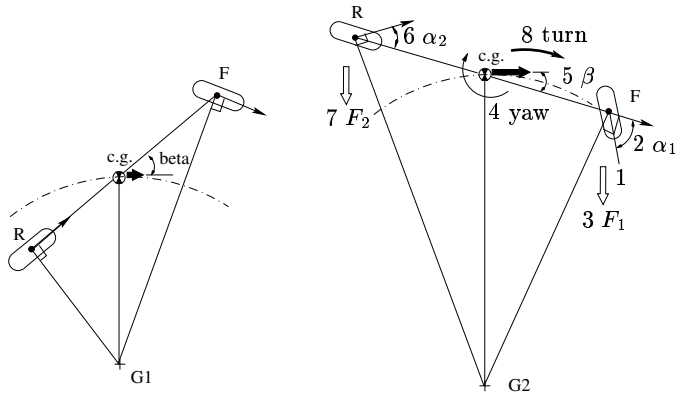
4.1 Double steering from a dynamics point of view

Vehicles in motion generally have a small lateral velocity component instead of moving exactly forward. The *side-slip angle* is the resulting small angle between a vehicle's longitudinal axis and its velocity vector in the plane of the road. Likewise, each tire has a slip angle, between its free rolling direction and its actual velocity vector, associated with a lateral force from the road surface (e.g. see [SCM02]). For low levels of lateral acceleration, lateral motion of a tire occurs by deformation as it rolls. The resulting lateral tire forces produce a total lateral force and a yawing on the vehicle.

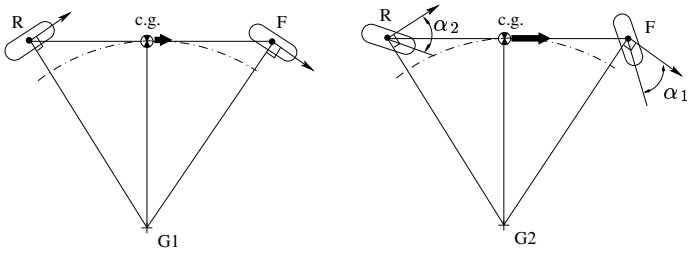
The interest shown in the double steering by some research groups has yield comprehensive insight about the dynamics response of vehicles having this mechanical feature. For instance, when an "opposite-deflection" rear wheel steering (RWS) mechanism is considered, the response of the vehicle differs significantly from the front wheel steering (FWS) response at low speeds [Whi90]. This is because the RWS vehicle must first move opposite to the desired steering direction in order to perform a turn; this reverse action is the cause of the unique transient characteristics of RWS vehicles. However, as the speed increases, the reverse action part of the RWS response becomes smaller in magnitude and shorter in time relative to the overall vehicle response [Whi90].

On the other hand, some of the advantages of coordinated double steering (i.e. bi-steerable as we have defined it) against FWS are discussed by Sano *et al* [SFS86]. One significant advantage is illustrated in Figure II.4 showing steady-states of a turning maneuver for a conventional FWS car and for a bi-steerable car.

Parts (a) and (b) of the figure show the dynamics of the motion of a conventional front-steering car. At very low speeds the assumptions of rolling without slipping lead to the motion model (a). The small angle β (exaggerated in the figure) giving the alignment of the vehicle to its path of motion is the side-slip angle. At high speeds (see part (b) of



(a) Car-like robot at low speed. (b) Car-like robot at high speed.



(c) Bi-steerable robot at low speed. (d) Bi-steerable robot at high speed.

Figure II.4: One advantage of using rear steering angles as a function of the front steering controls according to a dynamics analysis made by [SFS86]. (a) and (b): The car-like robot at low and high speeds respectively. The body is said to slip at angle β. (c) and (d): The bi-steerable kinematics could lead, if controlled appropriately, to the elimination of the side-slip angle of the body yielding a faster lateral response. The explanation is in the text.

the figure) the forces involved are larger. Dynamics of the tire-ground contact impose to each wheel a side-slip angle α. The vehicle centre of gravity is said to slip with angle β.

To appreciate the interest of the double steering capability, the analysis should turn to the transient behaviour of the vehicle— i.e. what happens between the moment the steering angle is applied and the steady-states. Refer to Figure II.4.(b), where the series of events before the car starts turning have been labeled from 1 to 8.

In fact, the vehicle motions can be divided into two parts according to [SFS86]. The first part corresponds to the rotation around the vehicle centre of gravity *c.g.* (event series from 1 to 4). This rotation corresponds to a yaw acceleration response in the presence of a force due to the change in the direction of the steering wheels. Indeed, to make the tire generate a lateral force, a side-slip angle is needed. The slip angle α_1 results from forces brought about by the vehicle inertia and friction. Once this slip angle has settled the lateral force F_1 creates a moment about the centre of gravity. Hence the rotation (yaw) occurs in the early stages of the whole series due to the moment created by F_1 (see Figure II.4 part (b)).

The second part of the whole motion is the revolution around the instantaneous turning centre $G2$, which occurs in the latter stages of the whole series (event series from 5 to 8). Indeed, because of the yaw acceleration response, the rear wheels are eventually subject to a lateral force inducing a slip angle $\alpha 2$. Depending on the road conditions (the friction coefficient) and the tire dynamics (e.g. speed), the rear slip angle α_2 generates a lateral force F_2. Both forces F_1 and F_2 bring about a centripetal force on the body resulting in vehicle turn.

In summary, before settling down to a steady-state revolution, the following steps occur (see Figure II.4 part (b)):

1. The vehicle steers its front wheels.
2. This generates a slip angle $\alpha 1$.
3. This in turn produces a lateral force $F1$,
4. causing the vehicle to rotate around its center of gravity.
5. The body slips with a side-slip angle β.
6. In response to the yaw moment, the rear wheels slip at angle $\alpha 2$.
7. This causes a lateral force $F2$,
8. joining force $F1$ in the generation of a centripetal force making the vehicle turn around point $G2$.

Thus the vehicle rotation around the center of gravity corresponds to yaw responses, while the revolution around the turning center ($G2$) corresponds to lateral acceleration responses. Since the angular acceleration of the vehicle around its center of gravity is related to its body yaw inertia, the larger the body side-slip angle is needed, the longer the delay before the vehicle is settled in the steady-state turning (see [SFS86] for details).

In their survey [SFS86], Sano *et al.* show, that if the rear wheels are controlled in an appropriate manner, the body side-slip angle (β) can be kept at zero in a steady-state. In this case the vehicle rotation around the center of gravity is no longer needed. Figures II.4 (c) and (d) illustrate this. The vehicle can start turning the moment the steering input is operated, reducing hence the delay in lateral acceleration response to steering.

A key parameter in the analysis of [SFS86] is the steer angle ratio of rear to front wheel $k = \frac{\varphi_{rear}}{\varphi_{front}}$. The authors suggest a few mechanical arrangements and a control function to determine the rear steering angle in function of the front one. The idea is to be able to switch from "similar" to "opposite" deflections in function of the amplitude of the steering angle of the front wheel as depicted in Figure II.5. This method makes also natural assumptions about the speed of the vehicle: i.e. the fact that at high speeds the amplitude of the steering angle is rather small, contrary to low speeds where the angles are likely to be large.

Sano *et al.* present interesting simulations and experiments with real vehicles equipped with a mechanical coupling between the front and the rear wheels. This coupling

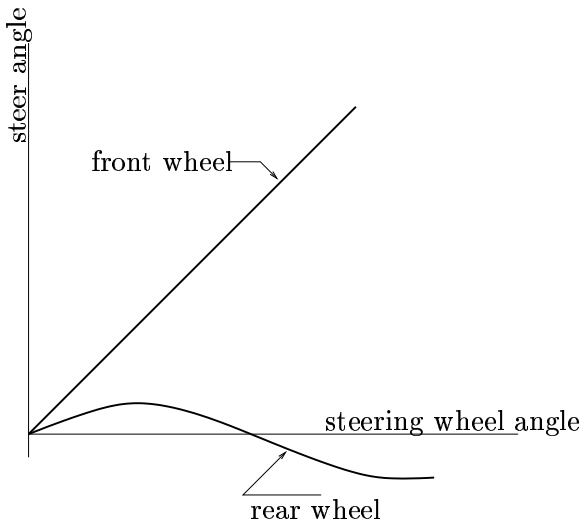


Figure II.5: The controlling function, according to Sano *et al* [SFS86], in order to determine $\varphi_{rear} = k \cdot \varphi_{front}$.

follows a relation $\varphi_{rear} = k\varphi_{front}$, with $-1 < k < 0.6$. Their results show for instance that a lane change is better negotiated (more accurately[2] and with greater stability) when $k \geq 0.3$.

These surveys are illustrative of the maneuverability enhancements from a dynamics standpoint. As far as motion planning is concerned, most of the analysis is done from the kinematics point of view. In this case there are also non-trivial advantages that we shall discuss next.

4.2 Double steering from a kinematics point of view

The possibility of deflecting the rear wheels in an opposite direction regarding the front steering angle leads to an increased maneuverability in cumbersome spaces. To illustrate this, consider the application of the same steering controls to the conventional car and to the BiS-car, to perform a simple parking maneuver as the one shown in Figure II.6. It is clear that the BiS-robot shows a higher penetration ability.

In this case of opposite deflection between rear and front wheels, the higher maneuverability is owing to the reduced turning radius of the BiS-vehicle compared to that of the car-like robot. Indeed, comparing their geometric representations (see Figures II.1 and II.2), we see that for a given steering angle φ, the instantaneous turning radius of the car (i.e. the distance $\overline{GR}$) is greater than its counterpart for the BiS-car (i.e. the distance $\overline{GH}$):

$$\overline{GR}_{car} = \frac{L\cos(\varphi)}{\sin(\varphi)} \geq \overline{GH}_{BiS} = \frac{L\cos(\varphi)\cos(f(\varphi))}{\sin(\varphi - f(\varphi))}$$

Figure II.7 shows the corresponding geometric construction.

[2]The subjects of the experiments where asked to maintain the car in the middle of the lane at 80km/h.

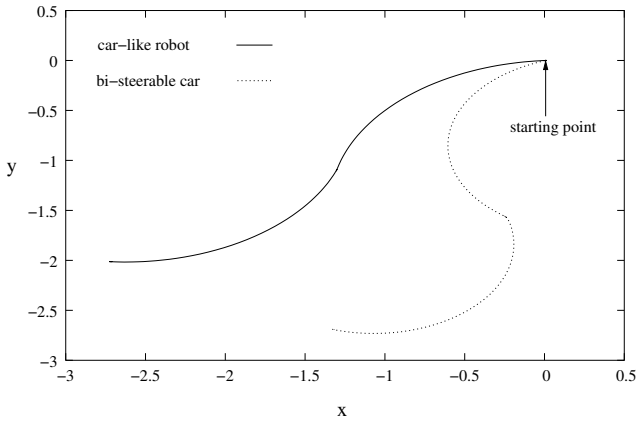


Figure II.6: Bi-steerable car's maneuverability characteristics: Simulated traces of the point (x_R, y_R) using models (II.2) and (II.8). These traces are obtained when applying the same typical commands, for a parallel parking maneuver, to a conventional car-like robot (top track) and to a BiS-car for which the rear-front coupling relation is a constant of proportionality held to 0.7(bottom track).

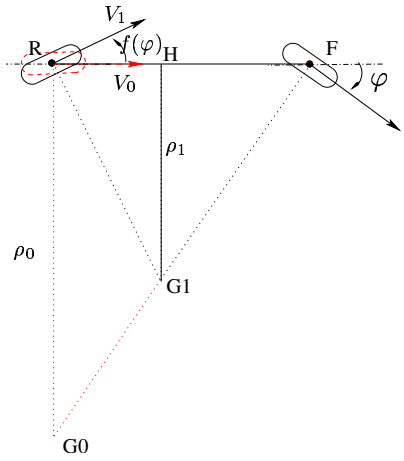
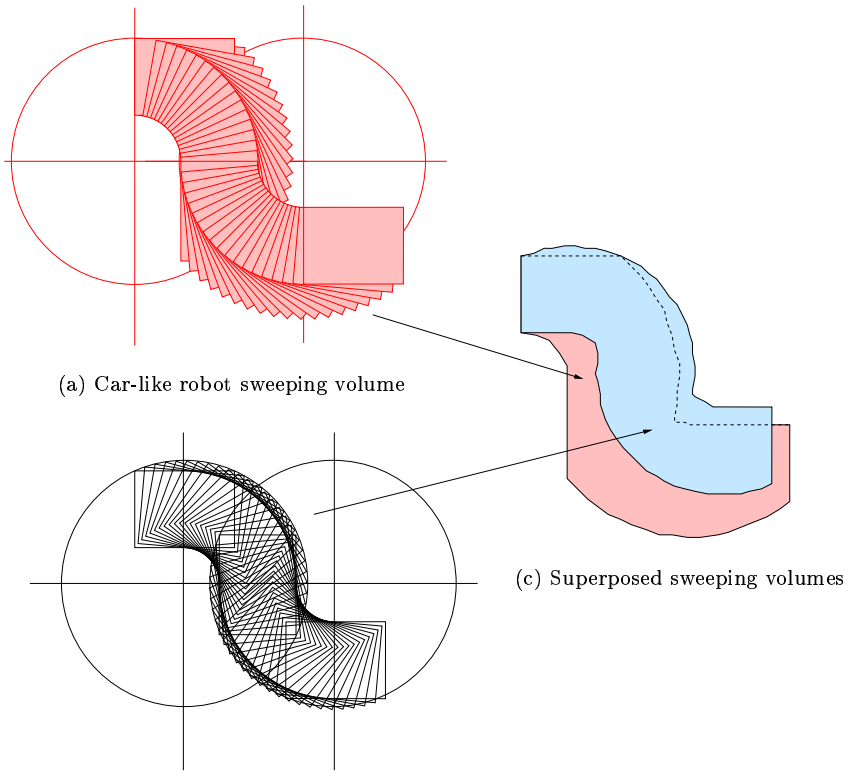


Figure II.7: Bi-steerable car's maneuverability characteristics: The turning radius is reduced owing to the deflection of the rear wheels while maintaining the front wheel steered.

As a consequence we have that the turning speed of the BiS-vehicle is greater than that of the car for a given value of φ.

Another major consequence is that the sweeping volume of the solid in motion is also reduced. This not also enhances the maneuverability in cluttered environments but also reduces the risk of collision. To appreciate this better consider Figure II.8 showing the sweeping volumes of a car-like robot (filled polygon) and that of a BiS-car (unfilled polygon) when performing a parking maneuver.



(a) Car-like robot sweeping volume

(b) Bi-steerable robot sweeping volume

(c) Superposed sweeping volumes

Figure II.8: (a) and (b): Sweeping volumes for car-like and bi-steerable robots respectively, when applying hypothetical maximum steering angles. The assumption is the same curvature radius can be achieved for both systems. The car-like instantaneous gyration centre is aligned with the rear ends of the vehicle, while that of the BiS-car is aligned with middle points of the body. Both robots have the same dimension. (c): When superposing both volumes, it can be readily seen the weeping volume of the conventional car is larger than that of the bi-steerable car.

5 Introduction to the path planning issue and flatness

5.1 Introduction

In view of all these advantages it is not surprising to find experimental platforms in some research laboratories around the world[3] and real vehicles (e.g. see [WQ01]) showing double steering kinematics. It would be worth to know whether a general solution could be found aiming at the motion autonomy paradigm, at least for the bi-steerable system as we have defined it. In this respect, the path planning issue is the first obstacle to address.

Solutions to the motion planning problem in the context of flatness resort to the properties of the flat output [FLMR95b]. The most simple example illustrating these properties is the car-like robot. In order to introduce the reader to the advantages of flatness for planning paths, we will present in this section the case study of the conven-

[3] *ibid.*(1)

tional car-like robot. The aim is to give some "landmarks" hopefully useful throughout Chapters III and IV.

5.2 Flatness in car-like robots

System dynamics Let us recall the kinematics model of the car-like robot:

$$\begin{cases} \dot{x_R} &= v_R \cdot \cos(\theta) \\ \dot{y_R} &= v_R \cdot \sin(\theta) \\ \dot{\theta} &= v_R \cdot \frac{\tan(\varphi)}{L}. \end{cases} \tag{II.10}$$

The rolling contact between the surface of the road and the wheels of the vehicle leads to nonholonomic constraints. For the car-like robot the velocity of point R must be tangent to the main axis of the car. This translates into the following nonholonomic constraint, which may be obtained by eliminating the control speed in equation (II.10):

$$-\dot{x}_R \sin(\theta) + \dot{y}_R \cos(\theta) = 0 \tag{II.11}$$

The left hand side of equation (II.11) is called a ***differential 1-form***[4], defined at a point $q := (x_R, y_R, \theta) \in \mathcal{CS}$:

$$\omega(q)(f) = -\mathrm{d}x_R \sin(\theta) + \mathrm{d}y_R \cos(\theta) + \mathbf{0}\mathrm{d}\theta \in \mathbb{R}. \tag{II.12}$$

The notation used is on purpose. What this equality says is that at a given point q, the 1-form (II.12) is a linear function mapping a tangent vector $f \in T_xM$ (the tangent space at q) to $\mathbb{R}$.

Thus equation (II.11) has the following geometric interpretation. At every point q, equation (II.11) defines a hyperplane of T_xM. This hyperplane contains all tangent vectors verifying (II.11) and hence defines a distribution $\Delta(q)$ of dimension 2. This distribution is precisely the one associated to the control system of (II.10). Therefore, finding solutions of (II.11) is equivalent to solve (II.10).

In fact, looking for solutions to (II.11) instead of (II.10) is actually more convenient. This comes from the fact that powerful analysis tools are at our disposal in the theory of *Pfaffian systems*. We shall come back to this idea in the following chapter. For now, the reader should keep in mind that the path planning problem can be solved using either approach.

Flat output of the system Back to the car-like robot, equation (II.11) suggests that if we knew the motion of point R in time, we would be able to directly obtain from the first derivatives of (x_R, y_R) the orientation of the vehicle: $\tan(\theta) = \dot{y}_R/\dot{x}_R$. Furthermore, if during the motion of point R we also knew the second derivatives of (x_R, y_R), we would obtain the front-steering angle. Indeed, from equations (II.2) we have that $\tan(\varphi) = L\dot{\theta}/v_R$.

[4]See Appendix B on page 148 for a recall on 1-forms.

From these observations we conclude that all the variables of system (II.2) can be obtained directly from x_R, y_R and their successive derivatives. Therefore, to any 2D curve $(x_R(t), y_R(t))$ corresponds (locally) a unique 3D trajectory $(x_R(t), y_R(y), \theta(t))$ with a set of controls (v_R, φ).

Flatness and motion planning The property shown by point R in the car-like robot example can be found in other physical systems; e.g. the inverted pendulum in a plane, the ideal vertical take-off aircraft, the induction motor and the mobile robot with trailers [FLMR99, RFLM93b, MR96]. The formalization of this property leads to the concept of *differential flatness*, firstly introduced by Fliess and co-workers in [FLMR95b].

The striking advantage of flatness for planning a path from an initial configuration q_i to a final configuration q_f should be now clear. In the car-like example, a flat output is the point R so that $y = (y_1, y_2) = (x_R, y_R)$. Since y_1 and y_2 are differentially independent (i.e. not related by any differential equation), it is possible to employ arbitrary curves for $y_1(t)$ and $y_2(t)$ in the flat-output space (e.g. parameterized polynomials) in order to connect y_i and y_f having only initial and final constraints on their time derivatives—i.e. those imposed at the initial and final conditions $(q_i, q_f, \dot{q}_i, \dot{q}_f, \cdots)$. Hence, the problem of finding a path in the 3-dimensional configuration space can be equivalently solved in the 2-dimensional *flat space*.

5.3 Is the bi-steerable car flat?

The compelling question is: can we apply these results to the BiS-car?

The close similitude between kinematic models of the car—equation (II.10)—and the BiS-car:

$$\begin{cases} \dot{x_R} &= v_R \cdot \cos(\theta + f(\varphi)) \\ \dot{y_R} &= v_R \cdot \sin(\theta + f(\varphi)) \\ \dot{\theta} &= v_R \cdot \frac{\sin(\varphi - f(\varphi))}{L \cdot \cos(\varphi)}, \end{cases} \tag{II.13}$$

reasonably leads to the conjecture that both systems share common properties. To verify this, we shall answer a first question: is the BiS-car flat?

Hence in the following section we investigate the differential flatness of the bi-steerable car.

6 Conditions of flatness for the BiS-car

We now proceed to show the bi-steerable car is flat and will give necessary conditions on the function $f(\varphi)$ for this. We assume the reader is already acquainted with notions about differential geometry and nonlinear control systems (i.e. vector fields, Lie brackets, distributions, etc.). Again, the unfamiliar reader may refer to Appendix A on page 140 for a brief recall on these and related concepts.

Notice that, like almost any car-like platform, our system has a bounded steering angle, so that $-\frac{\pi}{2} < \varphi < \frac{\pi}{2}\,[\pi]$. For the same reason, the function $f(\varphi)$ must be chosen so that it does not reach this bound at the rear wheels and in particular, $f(\varphi) \neq \frac{\pi}{2}\,[\pi]$.

Therefore, in the sequel we assume that

$$\cos(\varphi) \neq 0$$

and

$$\cos(f(\varphi)) \neq 0.$$

6.1 Conditions for two-input driftless systems

In order to study the flatness of the bi-steerable car, we will focus on its control system. Hence we add φ to the configuration vector, considering its derivative as the second control of the robot.

To alleviate notation in this section, we will denote by $X = (x, y, \theta, \varphi)$ the state vector, with θ the orientation of the robot, φ the front steering angle and x, y the Cartesian coordinates of R (see Figure II.2). Therefore the configuration space (or the state space in control theory) of the bi-steerable car is $\mathcal{CS} = R^2 \times (S^1)^2$.

In this setting the bi-steerable system dynamics is given by the following driftless control system with 4 states and 2 inputs:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} \cos(\theta + f(\varphi)) \\ \sin(\theta + f(\varphi)) \\ \frac{\sin(\varphi - f(\varphi))}{L\cos(\varphi)} \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \omega. \qquad (\Sigma)$$

Furthermore, to avoid confusion between the function $f(\varphi)$ and the usual vector field notation, we denote the two driving vector fields of (Σ) by:

$$V_1(X) = \begin{pmatrix} \cos(\theta + f(\varphi)) \\ \sin(\theta + f(\varphi)) \\ \frac{\sin(\varphi - f(\varphi))}{L\cos(\varphi)} \\ 0 \end{pmatrix}, \; V_2(X) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \qquad (II.14)$$

For such a system, there exists a necessary and sufficient condition of flatness given by Rouchon [MR94]:

Theorem 6.1 ([MR94]) *A system without drift in n states and 2 controls ($n \geq 2$) is flat* iff *for $i = 0, \ldots, n-2$, the vector spaces $E_i(X)$ spanned by the system vector fields and their Lie brackets of length less than or equal to i, satisfy at all points: dim $E_i(X) = i + 2$.*

We give now some precisions about this theorem. Using the Lie bracket operator recursively on the input vector fields $\{V_1, \cdots, V_m\}$, one can associate a sequence E_i of

distributions to a system defined by some distribution $\Delta = span\{V_1, \cdots, V_m\}$. To this end, one sets:

$$\Delta = E_0 = span\{V_1, \cdots, V_m\},$$

and the next distributions are defined as:

$$E_{i+1} = E_i + [E_i, E_i], \quad i = 1, 2, \ldots,$$

where $[E_i, E_i]$ is the short notation for all possible bracketings between vector fields of E_i.

The sequence of E_i is called a *filtration* of Δ.

Hence Theorem 6.1 states that a driftless system with two inputs is flat iff:

- E_i are regular distributions: which means that the vector space $E_i(X)$ has the same dimension for all X,
- The dimension of E_i increases by one and exactly one at each bracketing until reaching the dimension of the system.

In the case of the bi-steerable control system, the growth of the filtration of E_0 is clearly dependent on the choice of the function $f(\varphi)$. In the next section, we use the Theorem 6.1 to get a necessary and sufficient condition on f for the flatness of the bi-steerable car.

Remark 6.1 *Flatness implies strong accessibility and hence controllability [FLMR95b]. Typically in our case, a system verifying Theorem 6.1 verifies the Lie Algebra Rank Condition (LARC) implying controllability. The converse is in general not true.*

6.2 Necessary and sufficient condition on $f(\varphi)$ for flatness

Let us introduce some notation that will be used thoroughly in the sequel. For any scalar function of the unique variable φ, $\lambda(\varphi)$, we note $\lambda'(\varphi)$ its derivative with respect to φ. In particular, $f'(\varphi)$ denotes the derivative of f with respect to φ.

The system vector fields are V_1 and V_2— see (II.14). Computing the successive Lie brackets of the vector fields that will be of our use and letting

$$\xi(\varphi) = \frac{(1 - f'(\varphi))\cos(\varphi - f(\varphi))\cos\varphi + \sin(\varphi - f(\varphi))\sin\varphi}{L\cos^2\varphi},$$

we get:

$$V_3(X) = [V_2, V_1](X) = \begin{pmatrix} -f'(\phi)\sin(\theta + f(\phi)) \\ f'(\phi)\cos(\theta + f(\phi)) \\ \xi(\phi) \\ 0 \end{pmatrix}$$

$$V_4(X) = [V_1, V_3](X) = \xi(\phi) \begin{pmatrix} \sin(\theta + f(\phi)) \\ -\cos(\theta + f(\phi)) \\ 0 \\ 0 \end{pmatrix} - \frac{f'(\phi)\sin(\phi - f(\phi))}{L\cos(\phi)} \begin{pmatrix} \cos(\theta + f(\phi)) \\ \sin(\theta + f(\phi)) \\ 0 \\ 0 \end{pmatrix}$$

$$V_5(X) = [V_2, V_3](X) = -f"(\phi) \begin{pmatrix} \sin(\theta + f(\phi)) \\ -\cos(\theta + f(\phi)) \\ 0 \\ 0 \end{pmatrix} - f'(\phi)^2 \begin{pmatrix} \cos(\theta + f(\phi)) \\ \sin(\theta + f(\phi)) \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \xi'(\phi) \\ 0 \end{pmatrix}$$

Or in vector field notation:

$$\begin{aligned} V_3(X) = [V_2, V_1](X) &= -f'(\varphi)\sin(\theta + f(\varphi))\tfrac{\partial}{\partial x} + f'(\varphi)\cos(\theta + f(\varphi))\tfrac{\partial}{\partial y} + \xi(\varphi)\tfrac{\partial}{\partial \theta} \\ V_4(X) = [V_1, V_3](X) &= \left(\xi(\varphi)\sin(\theta + f(\varphi)) - \tfrac{f'(\varphi)\sin(\varphi - f(\varphi))}{L\cos(\varphi)}\cos(\theta + f(\varphi))\right)\tfrac{\partial}{\partial x} \\ &\quad - \left(\xi(\varphi)\cos(\theta + f(\varphi)) + \tfrac{f'(\varphi)\sin(\varphi - f(\varphi))}{L\cos(\varphi)}\sin(\theta + f(\varphi))\right)\tfrac{\partial}{\partial y} \\ V_5(X) = [V_2, V_3](X) &= \left(-f"(\varphi)\sin(\theta + f(\varphi)) - f'(\varphi)^2\cos(\theta + f(\varphi))\right)\tfrac{\partial}{\partial x} \\ &\quad + \left(f"(\varphi)\cos(\theta + f(\varphi)) - f'(\varphi)^2\sin(\theta + f(\varphi))\right)\tfrac{\partial}{\partial y} \\ &\quad + \xi'(\varphi)\tfrac{\partial}{\partial \theta} \end{aligned}$$

According to Theorem 6.1, the system is flat iff:

- dim $E_0(X) = span\{V_1(X), V_2(X)\} = 2$
- dim $E_1(X) = span\{V_1(X), V_2(X), V_3(X)\} = 3$
- dim $E_2(X) = span\{V_1(X), \cdots, V_5(X)\} = 4$

for all X.

Let us verify this by first considering the case of the configurations where

$$f'(\varphi) = 0.$$

• **Case f'**$(\varphi) = 0$ In this case one can check easily that

$$[V_1\ V_4\ V_3\ V_2]_{f'(\varphi)=0} = \begin{bmatrix} \cos(\theta + f(\varphi)) & \frac{\cos f(\varphi)}{L\cos^2\varphi}\sin(\theta + f(\varphi)) & 0 & 0 \\ \sin(\theta + f(\varphi)) & -\frac{\cos f(\varphi)}{L\cos^2\varphi}\cos(\theta + f(\varphi)) & 0 & 0 \\ \frac{\sin(\varphi - f(\varphi))}{L\cos(\varphi)} & 0 & \frac{\cos f(\varphi)}{L\cos^2\varphi} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and that the determinant of this matrix is non zero (indeed, by assumption $\cos(\varphi) \neq 0$ and $\cos f(\varphi) \neq 0$). Therefore the system clearly respects the condition of Theorem 6.1 at points where $f'(\varphi) = 0$.

Thus from now on we assume

$$f'(\varphi) \neq 0.$$

We have:

• **dim $\mathbf{E}_0 = 2$** This is obvious.

• **dim $\mathbf{E}_1 = 3$** One can check that

$$\lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3 = 0 \Longleftrightarrow$$

$$\lambda_1 = \lambda_2 = 0 \ \ and \ \ \lambda_3 f'(\varphi) = 0 \ \ and \ \ \lambda_3 \xi(\varphi) = 0.$$

Since we are in a case where $f'(\varphi) \neq 0$ we also have $\lambda_3 = 0$ and therefore $(V_1, V_2, V_3)(X)$ is a free family for any configuration X.

• **dim $\mathbf{E}_2 = 4$** The distribution E_2 has two vector fields (V_4 and V_5) more than E_1 of dimension 3. The dimension of E_2 included in the tangent bundle of the $\mathcal{CS}$ cannot exceed $dim\ \mathcal{CS} = 4$. Therefore we just have to check that at any point X, at least one of the two vectors $V_4(X), V_5(X)$ does not belong to $E_1 = span\{V_1, V_2, V_3\}$. Let us consider successively the case of V_4 and V_5.

One can show that:

$$\lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3 + \lambda_4 V_4 = 0 \Longleftrightarrow$$

$$\begin{cases} \lambda_1 - \frac{\sin(\varphi - f(\varphi))}{L\cos(\varphi)} f'(\varphi)\lambda_4 &= 0 \\ f'(\varphi)\lambda_3 - \xi(\varphi)\lambda_4 &= 0 \\ \frac{\sin(\varphi - f(\varphi))}{L\cos(\varphi)}\lambda_1 + \xi(\varphi)\lambda_3 &= 0 \\ \lambda_2 &= 0 \end{cases}$$

whose discriminant function is:

$$\Lambda_1(\varphi) = f'(\varphi)^2 \cos^2(\varphi) + \cos^2(f(\varphi)) - 2f'(\varphi)\cos(\varphi)\cos(f(\varphi))\cos(\varphi - f(\varphi)). \quad \text{(II.15)}$$

Then we show that for any configuration where $f'(\varphi) \neq 0$ the following statement

$$\lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3 + \lambda_4 V_4 = 0 \Longleftrightarrow \lambda_i = 0 \ \ \forall i$$

is equivalent to:

$$\Lambda_1(\varphi) \neq 0.$$

In the same way,

$$\lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3 + \lambda_5 V_5 = 0 \Longleftrightarrow$$

$$\begin{cases} \lambda_1 - f'(\varphi)^2\lambda_5 & = & 0 \\ f'(\varphi)\lambda_3 + f''(\varphi)\lambda_5 & = & 0 \\ \frac{\sin(\varphi - f(\varphi))}{L\cos(\varphi)}\lambda_1 + \xi(\varphi)\lambda_3 + \xi'(\varphi)\lambda_5 & = & 0 \\ \lambda_2 & = & 0 \end{cases}$$

and we show that there is a function

$$\Lambda_2(\varphi) = f''(\varphi) + 2f'(\varphi)^2\tan(f(\varphi)) - 2f'(\varphi)\tan(\varphi) \tag{II.16}$$

such that for any configuration X where $f'(\varphi) \neq 0$,

$$\dim E_2(X) = 4 \Longleftrightarrow \Lambda_2(\varphi) \neq 0.$$

Thus we can state the following Proposition.

Proposition 6.1 *Given a Bi-steerable car (Σ) with a characteristic function f between the rear and the front steering angle.*

The system is flat iff f is such that, for any front steering angle φ,

$$\Lambda_1(\varphi) \neq 0 \quad or \quad \Lambda_2(\varphi) \neq 0,$$

with Λ_1 and Λ_2 including f and defined respectively in equations (II.15) and (II.16).

Example

Consider linear functions f of the form

$$f(\varphi) = k\varphi, \quad k \in \mathbb{R}.$$

Then we have:

$$\Lambda_1(\varphi) = k^2\cos^2(\varphi) + \cos^2(k\varphi) - 2k\cos\varphi\cos k\varphi\cos(\varphi - k\varphi)$$

and

$$\Lambda_2(\varphi) = 2k^2\tan(k\varphi) - 2k\tan(\varphi).$$

One can check that

$$\begin{array}{rcl} \Lambda_2(\varphi) = 0 & \Longleftrightarrow & k^2\tan(k\varphi) = k\tan(\varphi) \\ & \Longleftrightarrow & k \in \{0, 1, -1\} \text{ or } \varphi = 0, \end{array}$$

since the tan function is a strictly increasing bijection.

Let us investigate each case. If $k = 0$ then $\Lambda_1(\varphi) = 1$. Therefore, Proposition 6.1 implies that the system is flat as expected since this is the case of the conventional car-like robot. If $k = -1$ then $\Lambda_1(\varphi) = 0 \Longleftrightarrow \cos 2\varphi = -1 \Longleftrightarrow \varphi \equiv \frac{\pi}{2}\,[2\pi]$, which is against our assumption. Then again, Proposition 6.1 implies that the system is flat. For the case $k = 1$, $\Lambda_1(\varphi) = \Lambda_2(\varphi) = 0$ and from Proposition 6.1, the system is not flat as expected since this system is not even controllable. Indeed, it corresponds to a bi-steerable car for which the front and rear steering angles are always equal: the system can only perform

translations without ever being able to change its orientation. Finally for $\varphi = 0$, one can check easily that $\Lambda_1(0) = 0 \Longleftrightarrow k = 1$. To summarize, the only case where there is a φ such that $\Lambda_1(\varphi) = \Lambda_2(\varphi) = 0$ is when $k = 1$. □

Therefore, Proposition 6.1 implies the following result:

Corollary 6.1 *Given a Bi-steerable car (Σ) with a linear relation (of factor k) between the rear and the front steering angle, the system is flat for all $k \neq 1$.*

7 Finding a flat output: what is the problem?

Up to this point we have set necessary and sufficient conditions for a bi-steerable system to be flat. We now turn to the next compelling question: What is the flat output?

The intention of this section is to show that the problem is not so straight forward as seemingly appears from the close similitude between the car-like and the bi-steerable systems.

7.1 Getting insight into the problem

Let us recall the car-like robot kinematics—equations (II.17) and Figure II.9 below—and the bi-steerable kinematics—equations (II.18) and Figure II.10:

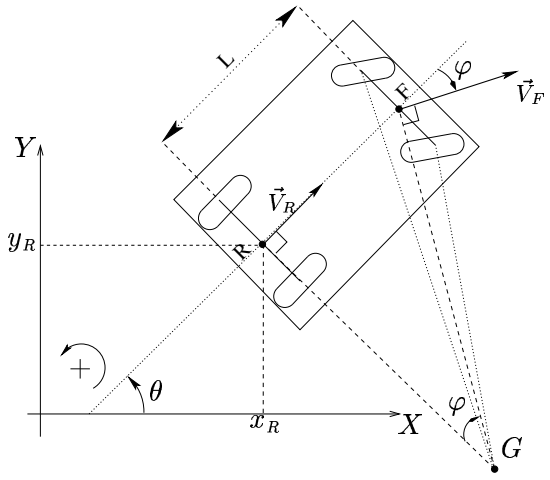


Figure II.9: The geometric model of a car-like robot

$$\begin{cases} \dot{x_R} &= v_R \cdot \cos(\theta) \\ \dot{y_R} &= v_R \cdot \sin(\theta) \\ \dot{\theta} &= v_R \cdot \frac{\tan(\varphi)}{L}, \end{cases} \tag{II.17}$$

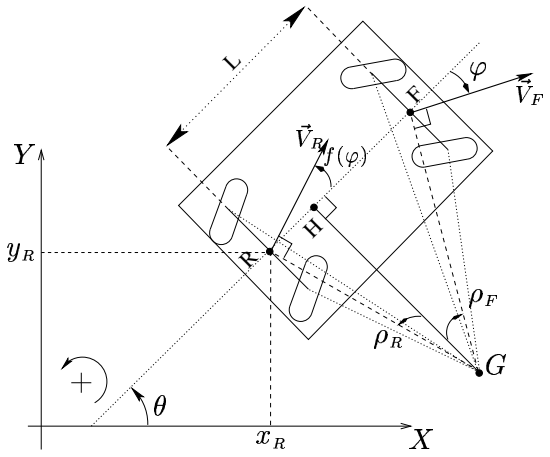


Figure II.10: Kinematics of a BiS-car

$$\begin{cases} \dot{x_R} &= v_R \cdot \cos(\theta + f(\varphi)) \\ \dot{y_R} &= v_R \cdot \sin(\theta + f(\varphi)) \\ \dot{\theta} &= v_R \cdot \frac{\sin(\varphi - f(\varphi))}{L \cdot \cos(\varphi)}. \end{cases} \tag{II.18}$$

Comparing equations (II.18) with equations (II.17), a different complexity arises from the bi-steerable car's model: the BiS-robot has an intrinsic tight coupling between the configuration variable θ and the control variable φ.

When trying to use the point R coordinates as the linearizing output, this tight coupling makes it impossible to deduce θ and φ from $x_R(t), y_R(t)$, which therefore do not seem to be the flat outputs for the BiS-car.

Looking at other nonholonomic systems, the following constant may be observed: If we consider the case of a classical tractor with n-trailers, the linearizing outputs are the coordinates of R the middle point of the axle of the last trailer. Just as for the car-like robot, the velocity vector of R directly yields the orientation of the last trailer. If we consider the case of more complex systems, such as the tractor-trailer with off-axle hitch (see [RFLM93b]), the linearizing outputs correspond to the Cartesian coordinates of a virtual point which position changes with respect to the robot body but which velocity is again parallel to some characteristic orientation of the system[5].

Therefore, a natural first guess for the BiS-car's linearizing output is a point whose velocity is parallel to the BiS-car main axis. We show that the projection H of the instantaneous center of rotation G on the main axis of the BiS-car has such a characteristic (see Figure II.10), even if the relative position of H to the robot changes along the trajectory as equations (II.5), rewritten here, show (e.g. see Figure II.11):

[5] Actually this orientation is that of the straight line passing by the middle points of tractor and trailer wheel axles.

$$\begin{cases} \overline{RH} = -L \cdot \frac{\cos(\varphi)\cdot\sin(f(\varphi))}{\sin(\varphi - f(\varphi))} \\ \overline{HF} = L \cdot \frac{\cos(f(\varphi))\cdot\sin(\varphi)}{\sin(\varphi - f(\varphi))} \end{cases} \tag{II.19}$$

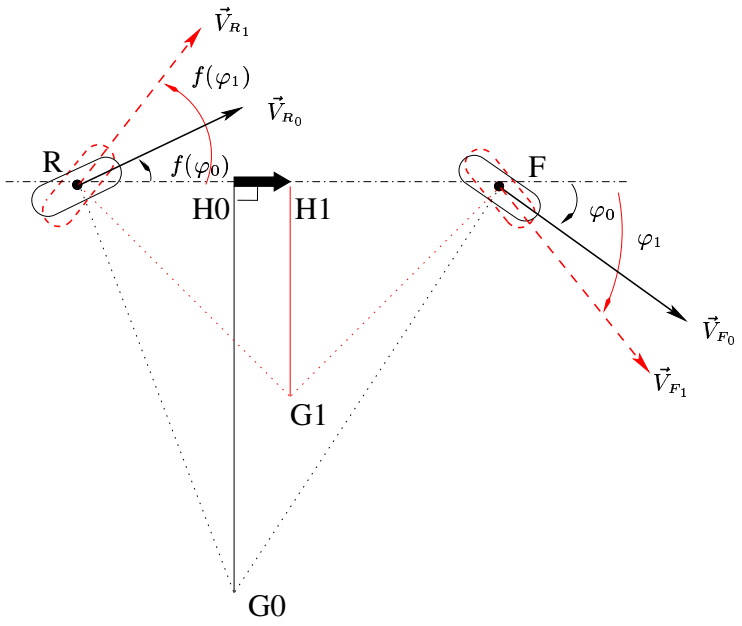


Figure II.11: Relative motion of the orthogonal projection on the main axis of the robot of the instantaneous point of rotation G.

Indeed, consider the absolute reference frame $[\mathcal{R}_a]$ and the reference frame $[\mathcal{R}_c]$ attached to the robot. Then for the point H, classical mechanics gives:

$$\vec{V}_{[\mathcal{R}_a]}(H) = \vec{V}_{[\mathcal{R}_c]}(H) + \vec{V}_{[\mathcal{R}_c/\mathcal{R}_a]}(H),$$

where $\vec{V}_{[\mathcal{R}_a]}(H)$ (resp. $\vec{V}_{[\mathcal{R}_c]}(H)$) is H's velocity in the absolute (resp. BiS-car's) frame and $\vec{V}_{[\mathcal{R}_c/\mathcal{R}_a]}(H)$ is the *dragging* velocity of the frame $[\mathcal{R}_c]$ in $[\mathcal{R}_a]$ at the point H. Actually, $\vec{V}_{[\mathcal{R}_c/\mathcal{R}_a]}(H)$ corresponds to the absolute velocity of H if it were fixed in $[\mathcal{R}_c]$ (which means fixed with respect to the BiS-car).

Thus classical mechanics gives:

$$GH \perp \vec{u}_\theta \Longrightarrow \vec{V}_{[\mathcal{R}_c/\mathcal{R}_a]}(H) = (\dot{\theta}\vec{u}_z \wedge \vec{GH}) \;//\; \vec{u}_\theta$$

In other respects, the relative motion of H with respect to the Cycab is always along its main axis. Therefore :

$$\vec{V}_{[\mathcal{R}_c]}(H) \;//\; \vec{V}_{[\mathcal{R}_c/\mathcal{R}_a]}(H) \;//\; \vec{V}_{[\mathcal{R}_a]}(H) \;//\; \vec{u}_\theta$$

Which implies that for any trajectory of the robot and at any point, the bi-steerable car is parallel to the curve followed by H in the plane.

Therefore, the curve followed by H gives some incomplete informations on the robot position and an exact information (through its tangent) on the robot orientation but it is not enough to fully determine the position of the robot, φ and the inputs. Notice that in the case of a car-like robot(i.e. $f(\varphi) = -k\varphi$, $k = 0$), H position is fixed with respect to the robot for any φ and (x_H, y_H) do represent the flat outputs of the car. However, for the general bi-steerable car, the position of H moves on the main axis of the robot when φ varies (see Equation (II.19)). This is the main reason why we cannot prove that x_H, y_H are the linearizing outputs.

7.2 Simplistic path planner for a BiS-car

Notice however, that for a given $\varphi = \varphi_0$, the position of H on the main axis is fixed, so the point H follows an arc of circle whose curvature is:

$$\kappa_0 = \frac{\sin(\varphi_0 - f(\varphi_0))}{L \cos\varphi_0 \cos(f(\varphi_0))}$$

Hence, the motion of the BiS-car for a constant φ is equivalent to the one of a unicycle situated on the main axis of the robot at a distance $\overline{RH}$ from the rear axle, following a curve with a constant curvature κ_0. This observation could be exploited, even if it means to over-constrain the system, by building a restricted class of path planners for the BiS-car by applying (e.g. see [SH00]) Dubins' work [Dub57] or Reeds and Shepp's work [RS90] to the virtual unicycle described above.

8 Summary and conclusions

In this chapter we have introduced the kinematics of the general bi-steerable car. We have established connections between this system and the car-like robot, pointing out the main advantages of the former over the latter. These advantages account for the interest of studying the bi-steerable kinematics from a robotics point of view.

Regarding the car-like robot and tractor with n-trailer systems, elegant solutions to the motion planning and control problems have been found by exploiting their flatness property. Motivated by the close similarities between the car-like and the bi-steerable systems, we wanted to know whether the latter is flat and whether we could find an intuitive flat output.

In this respect, we have given a condition of differential flatness ([MR94]) for the bi-steerable car. This condition led eventually to a necessary condition on the front-to-rear steering function f in order to guarantee flatness at all points of $\mathcal{CS} \subset \mathbb{R}^4$. Typically, for a linear front-to-rear coupling function of the form $f(\varphi) = k\varphi$ we have that the system is flat for $k \neq 1$.

The further complexity of the bi-steerable system pushed us to search for hypothetical solutions to the problem of finding a flat output. These solutions were basically explored by using a *linear* geometric approach. However this shed no light on plausible flat outputs.

As a matter of fact the core of the problem we have to address is finding relations between variables of the robot and some "output variables". The latter should be (a priori) the coordinates of some point of the 2D-space. This set of coordinates, together with its time-derivatives, should give us sufficient information in order to recover the state and controls of the system.

The problem stated in this way turned ourselves to sophisticated mathematical tools to search for a solution using a *differential* geometric approach. Hence the following chapter discusses the problem and its solution, leading to the explicit computation of feasible paths for the general bi-steerable system.

Chapter III

Engel normal form and path planning for the BiS-car

In this chapter we solve the path planning problem for the bi-steerable car. To this end we needed to *find* a flat output for the system. It is our aim to show how to *compute* a set of transformations, sending the original system coordinates to a particular set of flat coordinates, namely those that put the system in its chained form.

In answering the question "*how to compute a flat output for the bi-steerable car?*", we were interested in a general formulation of the problem. We were interested in finding a flat output for a driftless control distribution of dimension 2 in $\mathbb{R}^4$. Hence our contribution is resumed as follows:

- We consider small-time controllable driftless systems, whose states live in $\mathbb{R}^4$ and subject to 2 nonholonomic constraints. For such systems we outline a systematic approach and give a necessary condition to compute the flat output allowing to put the system in its chained form.
- We solve the complete path planning problem for the bi-steerable car. Indeed, we apply the methodology to find a set of flat coordinates for the bi-steerable car. This flat output allowed us to solve for the first time the complete path planning problem for this kind of mechanical structure, as shown by simulation results.

1 Introduction

The problem we want to solve in this chapter may be stated as follows. Suppose that the system is defined by a set differential equations of the form:

$$\dot{x} = g(x, u) \quad (x, u) \in \mathbb{R}^n \times \mathbb{R}^m.$$

The general open loop control problem is to find a trajectory $(x(t), u(t))$ steering the system from an intial state x_i at $t = 0$ to a final state x_f at $t = T$. When the system

is controllable there is an infinite number of possible trajectories and hence of controls allowing for the achievement of the transition. Choosing amongst these trajectories is a basic problem in (optimal) control theory, where the aim is to look for those controls that minimize a certain criteria.

Alternatively, the problem may be addressed from a geometric point of view. Instead of focusing on the directions in which the system is allowed to move—i.e. the driving vector fields corresponding to the degrees of freedom of the system—we turn to the analysis of the system defined by its nonholonomic constraints. For, suppose that the system constraints are a set of differential equations of the form

$$\boldsymbol{\alpha}(x, dx) = 0. \tag{III.1}$$

Instead of looking for optimal solutions in the control space of the system, we look for a way to *parameterize integral curves* solving the system of differential equations (III.1), defining the kinematic constraints of the system. This is the subject of the present chapter.

The results obtained in this chapter concern the theory of *exterior differential systems* [BCG$^+$91], particularly Pfaffian (which will be defined in section 2).

Roughly speaking, a *Pfaffian system* is a finite collection of (*exterior*) differential equations

$$\alpha_1 = 0, \ldots, \alpha_s = 0, \tag{III.2}$$

where each α_i is a linear 1-form[1]. In 1814-15 Pfaff pioneered in this field by studying solutions to a single equation $\alpha = 0$ ([BCG$^+$91]).

Typically, such exterior equations arise in mechanical systems subject to kinematic constraints. For instance, s nonholonomic constraints in mobile robots give rise to an exterior differential system of the form (III.2).

Today, the theory of Pfaffian systems gathers an important number of powerful theorems characterizing systems which can be put in a *normal form* by a change of coordinates. A normal form corresponds to a different representation of the original exterior equations, expressed with respect to a new coordinate chart that needs to be found. The interest of the normal form is that several important properties of the original system can be elucidated (e.g. linearizability properties). Moreover, solutions to the normal form can be obtained from arbitrary functions. Indeed, amidst the new coordinates, there is a subset to which arbitrary functions can be assigned. In fact, this subset represents a *flat output* of the original system. Complete solutions are then found by successive differentiation of the functions. By computing the inverse transformations, it is possible to derive solutions for the original system of equations.

Notwithstanding the powerful results from the theory, the major difficulty is still to find the appropriate transformations for the coordinate change. In this respect, finding the flat output of any control system in the general case is a (very) difficult problem. A systematic approach to solve it would consist in establishing the required computations in order to find the coordinate change in the general case. This is a very ambitious objective

[1] see Appendix B for a brief recall.

and to our knowledge there are barely a few methodologies aiming at formalizing the steps to undertake for particular classes of systems.

The rare contributions in this direction concern mainly a few approaches developing the powerful machinery of Pfaffian systems. In the robotics field one example is [TMS95]. In that paper, the authors are interested in studying the path planning problem for the general tractor with n-trailers system. They present an algorithm for *finding* a family of transformations putting the nonholonomic constraints of the system into the so called Goursat normal form. The Goursat's normal form is in fact the dual of the chained form of $(s+2)$-dimensional control systems, where s is the number of nonholonomic constraints. Hence the path planning problem of the general tractor with n-trailers is solved for the chained system as it was discussed in Chapter I (see [TMS95] for details).

The algorithm presented in [TMS95] relies on the theorems of Pfaff, Engel and Goursat giving sufficient conditions for finding the required transformations. However, the explicit computation of these transformations remained a guess work [TMS95].

Another example, this time in theoretical Control, concerns a specific class of control systems [Pom97]. In that paper, the author gives necessary and sufficient conditions for the existence of linearizing outputs for such general systems with drift. They lead to Partial Differential Equations (PDE) that the pair of linearizing outputs must satisfy. However, the PDEs for a general system of this class leads to overwhelming computations of compatibility conditions. Hence the author proposes to use the *infinitesimal Brunovsky form* [Pom97] that allows to write different, more tractable, PDEs coming from an integrability condition for one forms: the unknowns are then some coefficients of transformations that act on pairs of differential forms instead of the linearizing outputs themselves. The interest is that solutions to this second set of PDEs yield a linearizing basis of one-forms, and, it is proved that, the first differentials of the flat outputs are *always* linear combinations of the elements of the basis.

Another work is [FLM$^+$97]. In that paper the authors mention how they relied upon the work of [MR94] (see section 2) to find the flat output for linearising the general tractor pulling an off-hooked trailer. The method used to compute the flat output included considerations on the invariance of the problem leading to an elliptic integral. Since in this case the codistribution is 2-dimensional, the flat-output construction was inspired from the study of Engel form systems (which are defined in section 2).

Our contribution follows this direction. The aim of our work is to outline a systematic approach for the explicit computation of a particular flat output for a "general" codistribution of dimension 2. By "general" we mean that the codistribution should verify Engel's conditions as will be explained in section 2. Indeed, the flat output we shall look for yields a system of nonholonomic constraints written in Engel normal form, corresponding to the chained form of 2-dimensional distributions. As a result we give a necessary condition on the transformations required in order to obtain this flat output. This translates into a set of PDEs which, to our knowledge, constitute a new result.

The approach yields a methodology susceptible of being applied to any system verifying Engel's conditions. This is in particular the case of the bi-steerable car, for which we compute a flat output. To reach our goal, symmetry considerations are instrumental in

the construction of the transformations sought. Indeed, we will see in the next chapter how the concept of flatness derives actually from the general notion of *endogenous equivalence*, where symmetry plays as well an important role.

Moreover, we show how this flat output allowed us to adapt the local planner [SLL+97], originally designed for tractor-trailer systems, in order to compute feasible paths verifying the topological property introduced in Chapter I (section 5.5). We then integrate the steering method into a global path-planning scheme ([LSL99])—i.e. taking into account the obstacles—yielding the first complete path planner for a bi-steerable system.

In the following sections, we shall start by pointing out few concepts on Pfaffian systems. Together with Appendix B, these concepts should suffice to understand the systematic approach for computing the flat output in the general (Engel) case. Then we apply the approach to the specific case of the bi-steerable car, for which we give a flat output. We finally show how this flat output allowed us to solve the path planning problem for the bi-steerable car.

2 Pfaffian systems

Recall (see Appendix B) that an *exterior differential system* can be viewed (pointwise) as a system of *exterior* equations on T_xM:

$$\alpha_1 = 0, \ldots, \alpha_s = 0$$

where each $\alpha_i \in \Omega^k(M)$ is a smooth k-form. A solution to an exterior differential system is any submanifold N of M which satisfies

$$\alpha_i(x)|_{T_xN} \equiv 0$$

for all $x \in N$ and all $i \in \{1, \ldots, s\}$.

Definition 2.1 *An exterior differential system of the form*

$$\alpha_1 = \alpha_2 = \ldots = \alpha_s = 0$$

where each α_i are independent 1-forms on a n-dimensional manifold is called a Pfaffian system of dimension s.

Definition 2.2 *A set of linearly independent 1-forms $\alpha_1, \ldots, \alpha_s$ in the neighbourhood of a point is said to satisfy the* Frobenius condition *if one of the following equivalent conditions hold:*

1. *$d\alpha_i$ is a linear combination of $\alpha_1, \ldots, \alpha_s$.*
2. *$d\alpha_i \wedge \alpha_1 \wedge \cdots \wedge \alpha_s = 0$ for $1 \leq i \leq s$.*
3. *$d\alpha_i = \sum_{j=1}^{s} \theta_j \wedge \alpha_j$ for some forms $\theta_j \in \Omega(M)$.*

When $d\alpha_i$ is a linear combination of $\alpha_1, \ldots, \alpha_s$ the following expression is frequently used

$$d\alpha_i \equiv 0 \; mod \, \alpha_1, \ldots, \alpha_s \; 1 \leq i \leq s$$

where the *mod* operation is implicitly performed over the algebraic ideal generated by the α_i; in other words if

$$I := \{\omega \in \Omega(M) \,|\, \omega = \sum_{i=1}^{s} \theta_i \wedge \alpha_i \;\; for \; some \; \theta_i \in \Omega(M)\}.$$

then we write

$$d\alpha_i \equiv 0 \; mod \; I.$$

Theorem 2.1 (Frobenius theorem for codistributions) *Let I be an algebraic ideal generated by the independent 1-forms $\alpha_1, \ldots, \alpha_{s=n-m}$ such that the Frobenius condition is satisfied. Then in a neighbourhood of x there is a coordinate system $y_1, \ldots, y_n$ such that I is generated by $dy_{m+1}, \ldots, dy_n$*

A proof can be found in [BCG+91].

The Frobenius theorem shows that a completely integrable system takes a very simple form upon a proper choice of local coordinates. It actually gives a "normal form" of a completely integrable system; i.e. the system can be written locally as

$$dy_{m+1} = \cdots = dy_n = 0$$

in a suitable coordinate system. The maximal integral manifolds are

$$y_{m+1} = const, \cdots, y_n = const,$$

and are therefore of dimension m. We say that the system defines a *foliation*, of dimension m and codimension $n - m$, of which these submanifolds are *leaves*.

When the system is not completely integrable (as nonholonomic ones), one is interested in studying the integral manifolds of the codistribution defined by the system constraints. In this respect, *normal forms* allow us to elucidate several important properties of the system, and in particular of its solutions (or integral manifolds).

We shall concentrate ourselves in the Engel form, which concerns codistributions of codimension 2 (more precisely $s = 2$ and $n = 4$). But before, we need to introduce additional notions as follows.

Let M be a manifold, with local coordinates x, on which a driftless system

$$\Sigma : \; \dot{x} = \sum_{i=1}^{m} f_i(x) u_i$$

is defined.

To the independent set of vector fields $\{f_1, \ldots, f_m\}$ is naturally associated a distribution and also a codistribution $\{f_1, \ldots, f_m\}^{\perp}$ on M.

Given 1-forms $\omega_i \in \Omega^1(M)$ such that $\{f_1, \ldots, f_m\}^\perp = \{\omega_1, \ldots, \omega_{s=n-m}\}$, Σ can be equivalently defined as the solution of the *Pfaffian system*:

$$\omega_1 = 0, \ldots, \omega_s = 0$$

To each driftless system is associated a *derived flag* which is the set of ideals

$$I^{(0)} \supset \ldots \supset I^{(k)}$$

such that

$$I^{(0)} = \{\omega_1, \ldots, \omega_s\}$$

is the ideal generated by ω_i's and

$$I^{(j+1)} = \{\eta \in I^{(j)} : d\eta \equiv 0 \mod I^{(j)}\}.$$

Definition 2.3 *The* rank *of a form ω is an integer r defined by $(d\omega)^r \wedge \omega \neq 0$ and $(d\omega)^{r+1} \wedge \omega = 0$.*

We are now ready to state the theorem of Engel and conclude this section.

Theorem 2.1 (Engel) *Let I be a two dimensional codistribution*

$$I = \{\omega_1, \omega_2\}$$

of four variables. If the derived flag satisfies

$$\begin{array}{l} dim I^{(1)} = 1 \\ dim I^{(2)} = 0 \end{array} \tag{III.3}$$

then there exists coordinates y_1, y_2, y_3, y_4 such that I may be written in normal form:

$$I = \{dy_4 - y_3 dy_1, dy_3 - y_2 dy_1\} \tag{III.4}$$

As a matter of fact the Engel normal form may be written as (e.g. see [BCG+91]):

$$I = \{dy - y'dx, dy' - y''dx\},$$

with local coordinates x, y, y', y''. The interest of the Engel normal form should be now clear. If a system is put into Engel normal form, then the "general solution" is visibly given by

$$y = f(x), \;\; y' = f'(x), \;\; y'' = f''(x),$$

where $f(x)$ is an arbitrary function of x. In this case, "general solution" means a solution for which $dx \neq 0$.[2] Clearly, a flat output corresponds to the pair $\{x, y\}$.

With these elements, we are ready to discuss a systematic approach leading to the computation of this flat output.

[2] One says that dx is an *independence condition*.

The solution proposed is inspired from the work of Martin and Rouchon [MR94]. In that note, the authors show that a 2-input driftless system is flat if, and only if, the associated Pfaffian system is a contact system. Moreover, they indicate how to compute the flat output via the Pfaff normal form of the differential form generating the last non zero system of the derived flag.

Since the Pfaffian system associated to the bi-steerable car contains two equations and four variables, finding the flat-output consists in finding the change of coordinates associated to its Engel normal form.

The work presented in the following section aims at bringing a contribution in this direction, namely by lemma 3.1 [SRH01]. The lemma states a *necessary* condition on the transformations required to send the original coordinates of a 2-dimensional codistribution to its flat coordinates yielding the Engel normal form.

3 Flat output of an Engel system

3.1 Preliminary discussion

Consider a manifold M of dimension 4 and a Pfaffian system on M defined by:

$$\omega_1 = 0 \quad \omega_2 = 0 \tag{III.5}$$

where $\omega_1, \omega_2 \in \Omega^1(M)$ are independent 1-forms. It has been shown [MR94] that such a system is flat if and only if its derived flag satisfies (III.3). Indeed, the derived flag (III.3) is precisely the dual of the filtration

$$E_{i+1} = E_i + [E_i, E_i], \quad i = 1, 2, 3$$

constructed in Chapter II (section 6).

Notice that other possibilities for the derived flag ($dimI^{(1)} = 2$ or $dimI^{(1)} = 1$, $dimI^{(2)} = 1$) correspond to a non controllable system (e.g. from the Lie Algebra Rank Condition) and each instance of the problem is actually equivalent to a system of lower dimension.

If we consider the Engel form (III.4) of the system (III.5), (y_1, y_4) are clearly the flat outputs of the system. Indeed in $\{y_1, y_2, y_3, y_4\}$ coordinates, the system (III.5) can be equivalently written in the chained form:

$$\begin{cases} \dot{y_1} &= u_1 \\ \dot{y_2} &= u_2 \\ \dot{y_3} &= y_2 u_1 \\ \dot{y_4} &= y_3 u_1 \end{cases}$$

where obviously all variables of the system can be obtained from (y_1, y_4) and their derivatives.

Now generally, ω_1 and ω_2 are not expressed in the $\{y_i\}$ coordinates. Therefore, in order to take a practical benefit of flatness we have to compute the coordinates change which put the system into Engel form and explicitly obtain the flat output expressions.

The main point to compute the coordinates change $\{x_i\} \longrightarrow \{y_i\}$ is to notice that the Engel form of the system is adapted to the derived flag (see the proof of the Engel theorem [BCG$^+$91]). One says that a basis of one-forms $\{\alpha_i\}$ is adapted to the derived flag if

$$I^{(i)} = \{\alpha_1, \ldots, \alpha_{(s_i)}\}$$

where s_i is a strictly decreasing sequence of integers. In other words, an adapted basis is one in which the derived systems are calculated by dropping elements from the end of the basis.

Therefore, for Engel systems:

$$I^{(1)} = \{dy_4 - y_3 dy_1\}.$$

Notice that any one-form η belonging to $I^{(1)}$ can be expressed as a linear combination of elements $\{dy_1, dy_4\}$ of the new basis. Moreover, any one-form generating $I^{(1)}$ in the original basis $\{dx_1, dx_2, dx_3, dx_4\}$ is collinear to the one-form: $dy_4 - y_3 dy_1$. These two facts are instrumental in establishing a condition for the coordinate change. We discuss now a systematic approach leading to the condition.

3.2 Computing the flat output of an Engel differential system

Let us compute a 1-form of $I^{(1)}$ in the original system coordinates (x_1, x_2, x_3, x_4) of a point $x \in M$, in which $\{\omega_1, \omega_2\}$ are expressed. Given $\{dx_1, dx_2, dx_3, dx_4\}$ the associated basis of $\Omega^1(M)$, there are scalar functions θ^i_j on M such that:

$$\omega_i = \sum_{j=1}^{4} \theta^i_j dx_j \qquad i = 1, 2 \tag{III.6}$$

For $\eta \in \Omega^1$:

$$\eta \in I^{(1)} \Longrightarrow \eta \in I \ \ and \ \ d\eta \equiv 0 \ mod \ I$$

Therefore there are scalar functions α_i on M such that:

$$\eta = \alpha_1 \omega_1 + \alpha_2 \omega_2 \tag{III.7}$$

which implies

$$\begin{array}{rcl} d\eta & = & d\alpha_1 \wedge \omega_1 + \alpha_1 d\omega_1 + d\alpha_2 \wedge \omega_2 + \alpha_2 d\omega_2 \\ & \equiv & \alpha_1 d\omega_1 + \alpha_2 d\omega_2 \ \ mod \ I \end{array} \tag{III.8}$$

Now from (III.6):

$$d\omega_i = \sum_{j=1}^{4} d\theta^i_j \wedge dx_j \qquad i = 1, 2$$

and expressing $d\theta^i_j$ in $\{dx_i\}$ basis using their partial derivatives one gets:

$$d\omega_i = \sum_{1 \le j < k \le 4} \left(\frac{\partial \theta^i_j}{\partial x_k} - \frac{\partial \theta^i_k}{\partial x_j}\right) dx_j \wedge dx_k \qquad i = 1, 2 \tag{III.9}$$

Since ω_1 and ω_2 are independent, from (III.6) one can express two of the $\Omega^1(M)$ basis vectors in function of ω_1, ω_2 and the other vectors. Without loss of generality, assume those basis vectors are dx_1, dx_2, (III.6) leads to:

$$dx_1 \equiv \beta_3^1 dx_3 + \beta_4^1 dx_4 \mod \{\omega_1, \omega_2\}$$

$$dx_2 \equiv \beta_3^2 dx_3 + \beta_4^2 dx_4 \mod \{\omega_1, \omega_2\}$$

where β_i^j's are scalar functions: $M \longrightarrow \mathbb{R}$ involving only θ_j^k's. Injecting these expressions in (III.9), the exterior product properties give:

$$d\omega_1 \equiv \lambda_1 dx_3 \wedge dx_4 \mod \{\omega_1, \omega_2\}$$

$$d\omega_2 \equiv \lambda_2 dx_3 \wedge dx_4 \mod \{\omega_1, \omega_2\}$$

where λ_1, λ_2 are computable functions of θ_i^j's and their partial derivatives. Therefore (III.8) becomes:

$$d\eta \equiv (\alpha_1\lambda_1 + \alpha_2\lambda_2) dx_3 \wedge dx_4 \mod I$$

Therefore a sufficient condition for η to belong to $I^{(1)}$ is:

$$\alpha_1\lambda_1 + \alpha_2\lambda_2 = 0$$

Hence, from equation (III.7):

$$\eta = \lambda_2\omega_1 - \lambda_1\omega_2 = \sum_{j=1}^{4} (\lambda_2\theta_j^1 - \lambda_1\theta_j^2) dx_j \tag{III.10}$$

is a 1-form of $I^{(1)}$ and therefore collinear to $dy_4 - y_3 dy_1$. Via the above computations, we have the explicit expression of η in x-coordinates:

$$\eta = f_1 dx_1 + f_2 dx_2 + f_3 dx_3 + f_4 dx_4$$

where the f_i's are known x functions. Since $\eta \wedge d\eta \neq 0$, there exist $i \neq j$ such that the differential form $f_i dx_i + f_j dx_j$ has a rank > 1. Assume that f_1 and f_2 are such functions. Now set

$$y_1 = P^1(x) \quad y_4 = P^4(x)$$

Then:

$$dy_i = \sum_{j=1}^{4} \frac{\partial P^i}{\partial x_j} dx_j \qquad i = 1, 4$$

Again since dy_1 and dy_4 are independent and η is collinear to $dy_4 - y_3 dy_1$, for x_3, x_4 fixed, the mapping $(x_1, x_2) \mapsto (y_1, y_4)$ is bijective. Set (Q^1, Q^2) its inverse:

$$x_1 = Q^1(y_1, y_4, x_3, x_4) \quad x_2 = Q^2(y_1, y_4, x_3, x_4)$$

Thus dx_1 and dx_2 are linear combinations of dy_1, dy_4, dx_3 and dx_4 and η reads:

$$\eta = g_1 dy_1 + g_4 dy_4 + \bar{f}_3 dx_3 + \bar{f}_4 dx_4$$

where $\bar{f}_3, \bar{f}_4$ are known functions of (Q^1, Q^2, x_3, x_4) and of the partial derivatives of the Q^i with respect to x_3 and x_4:

$$\bar{f}_3 = f_3 + f_1\frac{\partial Q^1}{\partial x_3} + f_2\frac{\partial Q^2}{\partial x_3}$$

and

$$\bar{f}_4 = f_4 + f_1\frac{\partial Q^1}{\partial x_4} + f_2\frac{\partial Q^2}{\partial x_4}$$

Therefore we have the following lemma:

Lemma 3.1 *Assume that η, the differential form generating $I^{(1)}$, reads*

$$\eta = f_1 dx_1 + f_2 dx_2 + f_3 dx_3 + f_4 dx_4$$

where the f_i are known x-functions with f_1 and f_2 non zero independent x-functions. The coordinates change giving the flat outputs (y_1, y_4) takes the form

$$x_1 = Q^1(y_1, y_4, x_3, x_4) \quad x_2 = Q^2(y_1, y_4, x_3, x_4)$$

where the unknown functions Q^1 and Q^2 are solution of the quasi-linear first order partial differential systems:

$$f_3 + f_1\frac{\partial Q^1}{\partial x_3} + f_2\frac{\partial Q^2}{\partial x_3} = 0$$

$$f_4 + f_1\frac{\partial Q^1}{\partial x_4} + f_2\frac{\partial Q^2}{\partial x_4} = 0$$

In the next section we study the specific case of the bi-steerable car and show the explicit computation of the flat output. This calculating process uses the necessary condition of Lemma 3.1. Further computations leading to the flat output include symmetry considerations in a similar way to [RFLM93b], where the computation of the flat output of the general—off-hooked— one-trailer system led to an elliptic integral.

As a matter of fact, the analysis of the invariance of the system under Euclidean transformations in the plane (e.g. translations and rotations) is instrumental in the construction of the transformations sought.

4 Flat output for the BiS-car

We recall that the notation $\lambda'(\varphi)$ implies that λ is a scalar function of the unique variable φ and $\lambda'(\varphi)$ is its derivative with respect to φ. Moreover, we denote by $\lambda^{(k)}$ the total derivative of λ of order k. For an arbitrary angle α let us denote by $\vec{u}_\alpha$ the unitary vector of direction α and by $\vec{u}_\alpha^{\perp}$ the unitary vector of direction $\alpha + \frac{\pi}{2}$.

We recall that we represent a configuration of the system by a point (x, y, θ, φ) of the manifold $M = R^2 \times (S^1)^2$ (of dimension 4)[3]. In this case, x, y are the Cartesian coordinates of the middle point of the front axle (point F in Figure III.1), θ is the orientation of the car in the absolute reference frame and φ is the angle of the front wheels with respect to the car (see Figure III.1).

[3]Notice that one could eventually take $\mathcal{CS} \subset R^4$ instead of a subset of M.

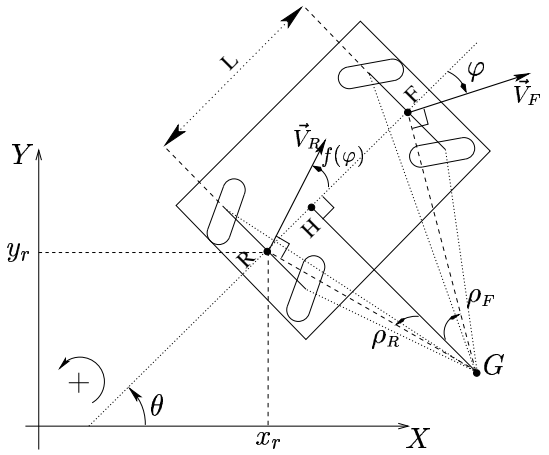


Figure III.1: Kinematics of a BiS-car

The kinematic constraints imposed on the system are due to the rolling without slipping of the wheels which means that the instantaneous velocity of each wheel is parallel to its orientation:

$$\begin{cases} \dot{y}\cos(\theta+\varphi) - \dot{x}\sin(\theta+\varphi) &= 0 \\ \dot{y}_r\cos(\theta+f(\varphi)) - \dot{x}_r\sin(\theta+f(\varphi)) &= 0 \end{cases}$$

with (x_r, y_r) being the coordinates of point R, the middle point of the rear axle.

4.1 Systematic approach (Lemma 3.1)

The equivalent Pfaffian system on M is $I^{(0)} = \{\omega_1, \omega_2\}$ with:

$$\begin{aligned} \omega_1 &= \cos(\theta+\varphi)dy - \sin(\theta+\varphi)dx \\ \omega_2 &= \cos(\theta+f(\varphi))dy - \sin(\theta+f(\varphi))dx - L\cos(f(\varphi))d\theta \end{aligned} \qquad \text{(III.11)}$$

where L is the longitudinal wheels base. From Chapter II-section 6, we know the derived flag of the bi-steerable car satisfies the Engel's conditions. Hence it can be put into the Engel normal form.

Therefore, we know that the derived flag of system (III.11) leads to a one-dimensional ideal:

$$I^{(1)} = \{\eta \in I^{(0)} : d\eta \equiv 0 \, mod \, I^{(0)}\} \subset I^{(0)}.$$

Then if we set η, a generator of $I^{(1)}$:

$$\eta = \mu_1\omega_1 + \mu_2\omega_2$$

we have

$$d\eta = \underbrace{(d\mu_1 \wedge \omega_1 + d\mu_2 \wedge \omega_2)}_{\equiv 0 \, mod \, I^{(0)}} + \mu_1 d\omega_1 + \mu_2 d\omega_2.$$

Hence the condition for $\eta \in I^{(1)}$ reads

$$\mu_1 d\omega_1 + \mu_2 d\omega_2 = 0. \tag{III.12}$$

Back to our system we have:

$$\begin{aligned} d\omega_1 &= -\sin(\theta+\varphi)(d\theta+d\varphi)\wedge dy \\ &\quad -\cos(\theta+\varphi)(d\theta+d\varphi)\wedge dx \\ d\omega_2 &= -\sin(\theta+f(\varphi))(d\theta+f'(\varphi)d\varphi)\wedge dy \\ &\quad -\cos(\theta+f(\varphi))(d\theta+f'(\varphi)d\varphi)\wedge dx \\ &\quad +Lsin(f(\varphi))f'(\varphi)d\varphi\wedge d\theta, \end{aligned}$$

and later on, using (III.11) to express dx and dy in function of $d\theta$, ω_1, ω_2, we get

$$d\omega_1 \equiv \frac{-L\lambda_1(\varphi)}{\sin(\varphi - f(\varphi))} d\theta\wedge d\varphi \quad mod\ I^{(0)}$$

$$d\omega_2 \equiv \frac{-L\lambda_2(\varphi)}{\sin(\varphi - f(\varphi))} d\theta\wedge d\varphi \quad mod\ I^{(0)},$$

with

$$\begin{aligned} \lambda_1(\varphi) &= \cos(f(\varphi)) \\ \lambda_2(\varphi) &= \cos(\varphi)f'(\varphi). \end{aligned}$$

Then, to verify (III.12) one can set $\mu_1 = -\lambda_2$ and $\mu_2 = \lambda_1$ so we obtain

$$\eta = \xi_1(\theta,\varphi)\mathrm{d}x - \xi_2(\theta,\varphi)\mathrm{d}y - L\lambda_1(\varphi)\cos(f(\varphi))\mathrm{d}\theta \tag{III.13}$$

where

$$\begin{aligned} \xi_1(\theta,\varphi) &= \lambda_2(\varphi)\sin(\theta+\varphi) - \lambda_1(\varphi)\sin(\theta+f(\varphi)) \\ \xi_2(\theta,\varphi) &= \lambda_2(\varphi)\cos(\theta+\varphi) - \lambda_1(\varphi)\cos(\theta+f(\varphi)). \end{aligned}$$

Then we know that if we find variables y_1 and y_2 such that at each point $p = \{x, y, \theta, \varphi\}$:

$$\eta = k_1(p)dy_1 + k_2(p)dy_2$$

for some scalar functions k_1, k_2 then y_1, y_2 are the flat outputs.

On the other hand, one can prove that for our system the flat outputs are only function of the state variables [MR95]. Considering our specific system and the symmetry of the problem (invariance with respect to the translations and rotations of the car) it is sound to consider (y_1, y_2) as the Cartesian coordinates of a point whose relative position with respect to the robot does not depend on the position and the orientation of the vehicle. Let us call such a general point H, whose Cartesian coordinates (x_H, y_H) are the flat output. Therefore the coordinates of H in the vehicle frame can be expressed as follows (see Figure III.2):

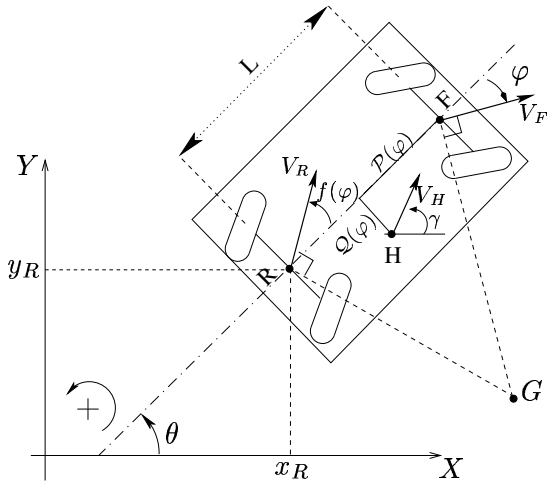


Figure III.2: Kinematics model of a bi-steerable car showing the coordinates of the flat output (point H) with respect to the reference frame of the robot placed at point F.

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_H \\ y_H \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \mathcal{P}(\varphi) \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} + \mathcal{Q}(\varphi) \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix} \tag{III.14}$$

with $\mathcal{P}$ and $\mathcal{Q}$ the unknown functions that we have to determine.

By computing dx, dy in function of $dy_1, dy_2, d\theta, d\varphi$ from (III.14) and substituting them in the expression of η above we get the set of PDE of the Lemma 3.1.

$$\begin{aligned} &\xi_2(\theta,\varphi)(\,\mathcal{P}(\varphi)\sin(\theta) + \mathcal{Q}(\varphi)\cos(\theta)\,) \\ &+\xi_1(\theta,\varphi)(\,\mathcal{P}(\varphi)\cos(\theta) - \mathcal{Q}(\varphi)\sin(\theta)\,) \\ &-\lambda_1 L\cos(f(\varphi)) = 0 \end{aligned} \tag{III.15}$$

$$\begin{aligned} &-\mathcal{P}'(\varphi)(\,\xi_1(\theta,\varphi)\cos(\theta) - \xi_2(\theta,\varphi)\sin(\theta)\,) \\ &+\mathcal{Q}'(\varphi)(\,\xi_1(\theta,\varphi)\sin(\theta) + \xi_2(\theta,\varphi)\cos(\theta)\,) = 0 \end{aligned} \tag{III.16}$$

These results can also be obtained by directly applying the lemma 3.1 to equations (III.13) and (III.14).

After some simplifications all coordinates but φ disappear in the PDE and we get:

$$\begin{aligned} &\mathcal{P}(\varphi)[\cos^2(\varphi)f'(\varphi) - \cos^2(f(\varphi))] \\ &+\mathcal{Q}(\varphi)[\cos(\varphi)\sin(\varphi)f'(\varphi) - \cos(f(\varphi))\sin(f(\varphi))] \\ &-L\cos^2(f(\varphi)) = 0 \end{aligned} \tag{III.17}$$

$$\begin{aligned} &\mathcal{P}'(\varphi)[\cos(\varphi)\sin(\varphi)f'(\varphi) - \cos(f(\varphi))\sin(f(\varphi))] \\ &+\mathcal{Q}'(\varphi)[\cos^2(\varphi)f'(\varphi) - \cos^2(f(\varphi))] = 0 \end{aligned} \tag{III.18}$$

It is now possible to look for solutions to this system of PDE. Indeed, by differentiating equation (III.17) we can express $\mathcal{P}'$ in function of $\mathcal{Q}$ and $\mathcal{Q}'$. Then substituting in

(III.18) yields a first order ordinary differential equation, whose solutions may be found by using standard techniques (e.g. using the method of the variation of the constant to obtain Q). Thence we would get $\mathcal{P}$ and y_1, y_2.

However, from a practical point of view, such a solution is not yet quite satisfactory. Indeed, $\mathcal{Q}$ will be computed through a double (enclosed) numerical integration which gives no hint on how to compute the inverse transformation—i.e. the expressions of the original coordinates in function of the flat output and its derivatives.

Looking closer at (III.17) and (III.18), an interesting observation can be made. This is the subject of the following section.

4.2 Further computations: the turning frame

Let us define the vector:

$$\vec{t} = \cos(\varphi) f'(\varphi) \vec{u}_{\theta+\varphi} - \cos(f(\varphi)) \vec{u}_{\theta+f(\varphi)}.$$

It is a combination of a vector parallel to the front wheel and a second one parallel to the rear wheel. Refer to Figure III.3. Expressing $\vec{t}$ and $\vec{FH}$ in the world's frame we get:

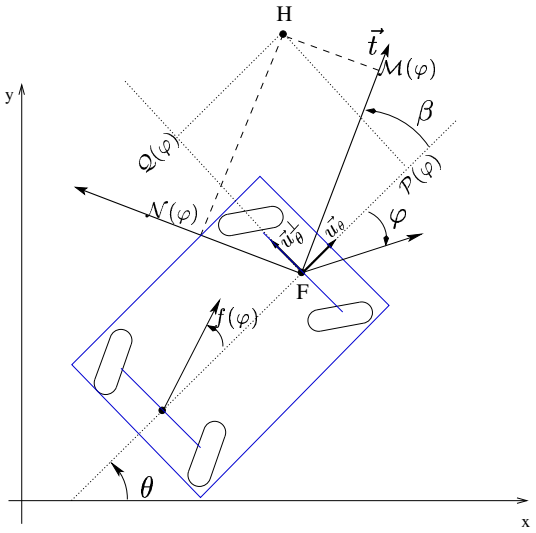


Figure III.3: Frames and coordinates.

$$\begin{aligned} \vec{FH} &= \mathcal{P}(\varphi)\vec{u}_\theta + \mathcal{Q}(\varphi)\vec{u}_\theta^\perp \\ \vec{t} &= \mathcal{A}(\varphi)\vec{u}_\theta + \mathcal{B}(\varphi)\vec{u}_\theta^\perp \end{aligned}$$

with

$$\begin{aligned} \mathcal{A}(\varphi) &= \cos^2(\varphi) f'(\varphi) - \cos^2(f(\varphi)) \\ \mathcal{B}(\varphi) &= \cos(\varphi)\sin(\varphi) f'(\varphi) - \cos(f(\varphi))\sin(f(\varphi)) \end{aligned}$$

Notice that with these definitions, equations (III.17) and (III.18) read:

$$\begin{cases} \mathcal{P}(\varphi)\mathcal{A}(\varphi) + \mathcal{Q}(\varphi)\mathcal{B}(\varphi) - L\cos^2(f(\varphi)) = 0 \\ \mathcal{P}'(\varphi)\mathcal{B}(\varphi) + \mathcal{Q}'(\varphi)\mathcal{A}(\varphi) = 0 \end{cases} \tag{III.19}$$

Thus equations (III.19) have the following geometric interpretation:

$$\vec{FH} \cdot \vec{t} = L\cos^2(f(\varphi)) \tag{III.20}$$

$$\left[\frac{d\vec{FH}}{d\varphi}\right]_{\boldsymbol{R}_\theta} // \vec{t} \tag{III.21}$$

In other words (III.19) tell us that there is a moving vector in the robot's frame that conveys information both on the position of H relative to the robot and on its motion as the wheels turn. Indeed $\vec{t}$ varies in function of φ and θ, however the projection of $H(y_1, y_2)$ on $\vec{t}$ is known— equation (III.20). Moreover, as we have seen, the position of H with respect to the car is only function of φ and its infinitesimal variation in the car frame $[\boldsymbol{R}_\theta]$ is parallel to $\vec{t}$ for any φ— equation (III.21). This shows the interest of the vector $\vec{t}$.

It seems therefore more interesting to express H in the frame attached to $\vec{t}$.

With the following notations:

$$\beta(\varphi) = \widehat{(\vec{u}_\theta, \vec{t})} = \tan^{-1}\frac{\mathcal{B}(\varphi)}{\mathcal{A}(\varphi)} \tag{III.22}$$

$$\vec{FH} = \mathcal{M}\vec{u}_\beta + \mathcal{N}\vec{u}_\beta^{\perp},$$

one can prove using (III.20) and (III.21) that

$$\mathcal{M}(\varphi) = \frac{L\cos^2(f(\varphi))}{\sqrt{\mathcal{A}^2(\varphi) + \mathcal{B}^2(\varphi)}} \tag{III.23}$$

and

$$\mathcal{N}'(\varphi) + \mathcal{M}(\varphi)\beta'(\varphi) = 0 \tag{III.24}$$

which yields:

$$\mathcal{N}(\varphi) = -\int_0^\varphi \frac{L\cos^2(f(u))(\mathcal{B}'(u)\mathcal{A}(u) - \mathcal{A}'(u)\mathcal{B}(u))}{(\mathcal{A}^2(u) + \mathcal{B}^2(u))^{\frac{3}{2}}}\,\mathrm{d}u \tag{III.25}$$

Thus the projection of the problem in the turning frame attached to $\vec{t}$ allows us to have tractable expressions of the H coordinates. Typically $\mathcal{M}(\varphi)$ is expressed analytically and $\mathcal{N}(\varphi)$ is the primitive of a simple expression. Therefore we have:

$$\begin{array}{rcl} \mathcal{P}(\varphi) & = & \mathcal{M}(\varphi)\cos(\beta(\varphi)) - \mathcal{N}(\varphi)\sin(\beta(\varphi)) \\ \mathcal{Q}(\varphi) & = & \mathcal{M}(\varphi)\sin(\beta(\varphi)) + \mathcal{N}(\varphi)\cos(\beta(\varphi)) \end{array} \tag{III.26}$$

Hence injecting (III.26) into (III.14) yields the expressions of the flat output (y_1, y_2) in function of the system variables (x, y, θ, φ).

4.3 Inverse expressions

The formulation of the problem in the new frame associated to the vector $\vec{t}$ also allows for the computation of the original coordinates (x, y, θ, φ) in function of the flat outputs and their derivatives.

Considering the invariance of the problem, one can prove that the curvature κ of the curve followed by H during the motion is only function of φ. Then we can compute the relation $\kappa(\varphi)$ by considering the case where the car does not move and only turns its wheels at a speed $\dot{\varphi} = 1$ (i.e $\varphi = t$) inducing a motion of H.

Therefore, in this specific case, the absolute velocity of H is equal to its relative velocity with respect to the car. This velocity is $(\mathcal{P}'(\varphi), \mathcal{Q}'(\varphi))$ and has an angle $\beta(\varphi)$ relatively to the car— see (III.21).

To carry on the computations of $\mathcal{P}', \mathcal{Q}'$, we differentiate hence equations (III.26) with respect to φ and take into account equation (III.24). This gives

$$\begin{aligned} \mathcal{P}'(\varphi) &= (\mathcal{M}'(\varphi) - \beta'(\varphi)\mathcal{N}(\varphi))\cos(\beta(\varphi)) \\ \mathcal{Q}'(\varphi) &= (\mathcal{M}'(\varphi) - \beta'(\varphi)\mathcal{N}(\varphi))\sin(\beta(\varphi)) \end{aligned}$$

On the other hand, the curvature is defined as the variation of the angle of the tangent vector to the curve in function of the arc length along the curve. Hence we have

$$\mathcal{K}(\varphi) = \frac{\beta'(\varphi)}{\sqrt{\left(\frac{dx_H}{ds}\right)^2 + \left(\frac{dy_H}{ds}\right)^2}},$$

which in our case gives:

$$\kappa = \mathcal{K}(\varphi) = \frac{\beta'(\varphi)}{\mathcal{M}'(\varphi) - \beta'(\varphi)\mathcal{N}(\varphi)} \tag{III.27}$$

where,

$$\kappa = \frac{\det\left(y^{(1)}, y^{(2)}\right)}{\left(\left[y_1^{(1)}\right]^2 + \left[y_2^{(1)}\right]^2\right)^{3/2}} \tag{III.28}$$

is the curvature of the path described by the flat output having a regular parameterization; i.e. such that the denominator is different from zero.

On the other hand, by Engel's theorem we know that η is collinear to a one-form:

$$dy_2 - g_3 dy_1.$$

Let us express η in function of dy_1 and dy_2. To this end, we differentiate expression (III.14) and inject it in (III.13) so as to eliminate dx, dy. After simplification—using (III.15) and (III.16)—we get

$$\eta = \xi_1(\theta, \varphi)\mathrm{d}y_1 \quad - \quad \xi_2(\theta, \varphi)\mathrm{d}y_2$$

Since $\eta \in I$, on the codistribution defining our system, we have

$$\eta = 0 \Rightarrow \frac{dy_2}{dy_1} = \frac{\xi_1(\theta,\varphi)}{\xi_2(\theta,\varphi)}$$

And now, functions ξ_1 and ξ_2 may be rewritten as:

$$\begin{aligned} \xi_1(\theta,\varphi) &= -L\sqrt{\mathcal{A}^2(\varphi)+\mathcal{B}^2(\varphi)}\sin(\theta+\beta(\varphi)) \\ \xi_2(\theta,\varphi) &= -L\sqrt{\mathcal{A}^2(\varphi)+\mathcal{B}^2(\varphi)}\cos(\theta+\beta(\varphi)). \end{aligned}$$

Therefore, if we let γ to be the orientation of the velocity vector of H (see Figure III.2) we have the following useful result:

$$\gamma = (\theta+\beta(\varphi))\,[\pm\pi] = \tan^{-1}\left(\frac{dy_2}{dy_1}\right). \tag{III.29}$$

The usefulness of (III.29) will be better appreciated in Chapter IV. Notice however that this equation gives us valuable information relating θ, φ and (dy_1, dy_2).

Hence by knowing the curve $y_1(t), y_2(t)$ of the flat outputs during the motion we can compute $\varphi(t)$ through $\kappa(t)$ (by inverting the expression $\mathcal{K}(\varphi)$). Then from $\beta(\varphi)$, and the orientation γ of the velocity of H, we get the orientation of the robot θ. Finally, we compute x, y, using (III.14).

5 Qualitative behaviour of the flat output

The graphs included at the end of the chapter show the behaviour of the flat output for a rear-to-front coupling function $f(\varphi) = k\varphi$ and for some $-1 \leq k < 1$.

Figure III.11 shows the behaviour of coordinate functions $\mathcal{P}(\varphi)$ and $\mathcal{Q}(\varphi)$ for some $k \leq 0$. Notice that $k = 0$ is the car-like robot (i.e. $\mathcal{P}(\varphi) = -L$, $\forall\varphi$) and as k moves towards -1 the coordinate $\mathcal{P}$ moves from point R to point F and settles at a constant value $-\frac{1}{2}L$ w.r.t. this last point.

We wanted to observe as well the behaviour of the flat coordinates when $k > 0$, which means that the front and rear wheels turn to the same side. Figure III.13 shows the equivalent graph of Figure III.12 for some positive k. One can observe that the relative motion of $\mathcal{P}$ with respect to the point F is considerably faster than in the previous case and in the opposite direction with respect to point R— the car-like case (see Figure III.14).

On the other hand one can also appreciate that, irrespective of the sign of k, the coordinate $\mathcal{Q}(\varphi)$ moves very little with respect to the longitudinal axis of the vehicle. However one should keep in mind that $\mathcal{Q}$ also intervenes by its derivative, and it is evident that its motion velocity with respect to φ may be important at some points. Interesting to observe as well is the fact that the curve of $\mathcal{Q}(\varphi)$ flattens out less for $k > 0$ than for $k < 0$ in the neighbourhood of $\varphi = 0$ as shown in Figure III.14.

We then turned to the characteristic angle $\beta(\varphi)$ which behaviour is shown in Figure III.15. Notice that the dynamics of β are more important when $k > 0$.

Finally the curvature κ as a function of φ— relation (III.27) —is depicted in Figure III.16. An interesting observation is that the function $\tan(\varphi)$, that is the curvature function for the car-like robot, appears as a "symmetry axis" of curvature functions (III.27) with opposite sign in k.

Remark 5.1 *Its is possible to show that the expression (III.27) is undetermined at* $\varphi = 0$. *However the computation of the limits (i.e. at* $0-$ *and at* $0+$*) gives*

$$\lim_{\varphi\to 0-} \mathbf{K}(\varphi) = \lim_{\varphi\to 0+} \mathbf{K}(\varphi) = 0.$$

Remark 5.2 *For functions* $f(\varphi) = k\,\varphi$ *where* $-1 \leq k < 1$, *Maple simulations indicate that the function (III.27) has asymptotic behaviour around values of* $\varphi = \pm(\frac{\pi}{2} + \epsilon)$ *for* $-0.1 < \epsilon < 0$. *We found no explanation justifying this phenomenon.*

These two remarks should indicate that it is possible to find a "better" flat output, since there is no unicity in this choice. We shall remark that it is yet difficult to find one flat output. However, this obviously opens the possibility for further research in this direction.

Notwithstanding these remarks, the results obtained during simulations show the validity of the approach as discussed in the next section.

6 Path planning for the BiS-car

We show now that the flat output introduced in this chapter may help in the computation of feasible paths for the general bi-steerable car. To this end we introduce first the motion planner we have used and subsequently we present simulation results showing feasible paths for the BiS-car.

6.1 Introduction to the motion planner

It is known (see e.g. [Lau98]) that it is possible to plan collision-free paths for small-time controllable nonholonomic robots in two steps: firstly, one can compute a collision-free geometric path irrespective of the nonholonomic constraints; secondly one can approximate this path in different ways, taking into account the specific kinematic constraints of the robot, by using different local planners.

A first version of this motion planning scheme was presented in [LJTM94]. The whole planner relied upon a local planner computing optimal paths for car-like robots and was successfully applied to the robot HILARE at LAAS.

This scheme was generalised in [SL98] by substituting the optimality of the steering method with a weaker constraint. The completeness of the planner was ensured by a local planner taking into account the small-time controllability of the system. This means that the local planner verified the topological property (**TP**) introduced in Chapter I:

Definition 6.1 (Topological Property [SL98])

Let $q = (x_1, x_2, \ldots, x_n)$ be a point in the configuration space $\mathcal{CS}$. Let $\mathrm{d}_{\mathcal{CS}}$ be the following distance over $\mathcal{CS}$:

$$\mathrm{d}_{\mathcal{CS}}(q_1, q_2) = \sum_{i=1}^{n} |x_i^1 - x_i^2|$$

The set of configurations q_2 such that $\mathrm{d}_{\mathcal{CS}}(q_1, q_2) < \epsilon$ is denoted by $B(q_1, \epsilon)$; this is the ball centered at q_1 of radius ϵ.

Let $\mathcal{C}$ be the set of feasible paths defined over an interval of the type $[0, T]$. A steering function is a mapping from $\mathcal{CS} \times \mathcal{CS}$ into $\mathcal{C}$:

$$(q_1, q_2) \rightarrow Steer(q_1, q_2)$$

where Steer is defined over the interval $[0, T]$, such that $Steer(q_1, q_2)(0) = q_1$, $Steer(q_1, q_2)(T) = q_2$.

Then, Steer verifies the topological property iff:

$$\forall \epsilon > 0,\ \exists \eta > 0,\ \forall (q_1, q_2) \in (\mathcal{CS})^2,$$
$$\mathrm{d}_{\mathcal{CS}}(q_1, q_2) < \eta \Rightarrow \forall t \in [0, T],\ \mathrm{d}_{\mathcal{CS}}(Steer(q_1, q_2)(t), q_1) < \epsilon$$

□

This general scheme was applied to the tractor-trailer system using a local planner based on sinusoidal inputs [SL98]. It has also been implemented with another local planner ([SLL+97]) based on the flatness of the tractor-trailer [LSL99].

Even if the steering method [SLL+97] aimed at tractor-trailer systems, it is possible to adapt the approach to the bi-steerable car. This stems from the fact that the representation of a point in the flat space is equivalent for both systems. Indeed, recall that (roughly speaking) for the bi-steerable car the following information is required: the plane-coordinates of the flat output (x_H, y_H), the angle γ giving the orientation of the velocity vector of the flat output, and the curvature κ giving the steering angle φ. In the case of the tractor-trailer system the flat output are the coordinates of the middle point of the rarest trailer [RFLM93a]. The corresponding variables for γ and κ in that case yield respectively information about the orientation of the last trailer and about the relative orientation between the latter and subsequent trailers.

Hence we have adapted the motion planning scheme [LSL99] so as to be able to compute feasible paths for the bi-steerable car. In this respect, we shall briefly discuss the steering method [SLL+97] and outline a further modification we made.

6.2 Steering method for flat systems verifying TP

The local planner proposed in [SLL+97] was conceived to steer *tractor-trailer* systems by exploiting their flatness. We now discuss the principle of this local planner (see Figure III.4).

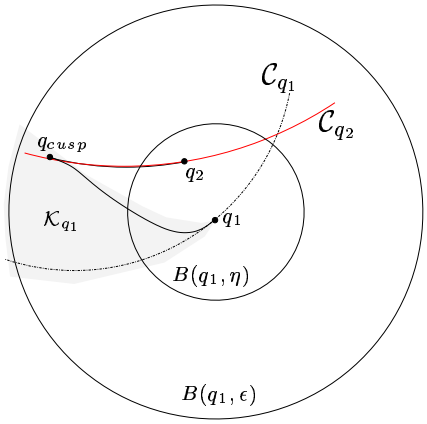


Figure III.4: Local planner for a flat system, as proposed by [SLL$^+$97], respecting the topological property.

Given two configurations in $\mathcal{CS}$, q_1 and q_2, the approach consists first in finding the corresponding flat points and then a curve between these two points. The constraints imposed to the curve at these end-points include the coordinates of these points (i.e. their position $M := (x, y)$ in the plane) as well as the curvature κ and the orientation of the velocity vector of the flat output (that we will call for convenience γ). Hence q_1(resp. q_2) intrinsically defines a curvature and a tangential orientation at M_1(resp. M_2). The aim is to compute a parameterized curve $\mathcal{C}_{q_1,q_2}(t)$ such that $\mathcal{C}_{q_1,q_2}(t=0) = (x_1, y_1, \kappa_1, \gamma_1)$ and $\mathcal{C}_{q_1,q_2}(t=1) = (x_2, y_2, \kappa_2, \gamma_2)$.

The parameters $(x_i, y_i, \kappa_i, \gamma_i)$, where $i = \{1, 2\}$, define at each ending point M_i a canonical curve $\mathcal{C}_{qi}(t)$ with constant $\kappa = \kappa_i$ and whose tangent vector at M_i has orientation γ_i. Hence this canonical curve may be either a circle or a straight line. The idea is then to interpolate the canonical curves as follows:

$$\mathcal{C}_{q_1,q_2}(t) = \alpha(t)\mathcal{C}_{q_2}(t) + (1 - \alpha(t))\mathcal{C}_{q_1}(t) \tag{III.30}$$

with $\alpha(t)$ being a polynome whose degree and coefficients are so that the conditions on $\mathcal{C}_{q_1,q_2}(t)$ at end points (i.e. at $t = 0$ and $t = 1$) are met (e.g. tangency conditions and continuity of κ; see [SLL$^+$97] for details).

Since the end objective is to join points q_1 and q_2 by paths verifying **TP**, a cusp point (forcing a change in the direction of motion) may prove to be necessary to reach lateral neighbors without leaving a vicinity of q_1. Thus the cusp point must be chosen according to the intuitive idea that: *the closer the points to each other, the shorter the path joining them*. This notion entails the following consequence: if both corresponding points M_1 and M_2 lie on the same canonical curve then the interpolation process (III.30) should yield exactly this canonical curve.

According to **TP**, if q_1 and q_2 are "sufficiently" close to each other—i.e. they lie within a ball $B(q_1, \eta)$—then $\mathcal{C}_{q_1,q_2}(t)$ remains within a ball $B(q_1, \epsilon)$. To construct the planner, [SLL$^+$97] aimed first at finding the set of points reachable from q_1 without a

cusp and such that for all $q \in B(q_1, \eta)$, $\mathcal{C}_q$ intersects this set. This set turns out to be a sort of cone $\mathcal{K}_{q_1}$ (see Figure III.4) starting from q_1 and developing around $\mathcal{C}_{q_1}$ [SLL+97].

The idea is the following: provided q_1 and q_2 are sufficiently close to each other, if one can follow a path starting from q_1 without leaving the ball $B(q_1, \epsilon)$, then one can reach the intersection q_{csup} between this path and the canonical curve associated to q_2 without leaving $B(q_1, \epsilon)$. Henceforth one can reach q_2 from q_{csup} by simply following the canonical curve of the former, since this curve introduces small local changes in the configuration variables. Accordingly, **TP** is verified since $\mathcal{C}_{q_1,q_2}(t)$ will remain within $B(q_1, \epsilon)$.

Therefore, by construction of $\mathcal{K}_{q_1}$ we have that $\exists \eta$ such that $\forall q \in B(q_1, \eta)$, $\mathcal{C}_q$ intersects $\mathcal{K}_{q_1}$ without leaving the ball $B(q_1, \epsilon)$. Hence if $q_2 \in B(q_1, \eta)$, there is a cusp point q_{cusp} within the intersection of $\mathcal{K}_{q_1}$ and $\mathcal{C}_{q_2}$. Henceforth, it is possible to reach q_{cusp} from q_1 and then follow $\mathcal{C}_{q_2}$ from q_{cusp} down to q_2 without leaving $B(q_1, \epsilon)$. This is depicted in Figure III.4 (see [SLL+97] for details).

6.3 Our contribution: symmetrical local planner

Seeking to improve performance we have modified the local planner above by choosing another cusp point as follows.

As a matter of fact the cusp point shown in Figure III.4 corresponds to a path going from q_1 to q_2. Equivalently, another cusp point on $\mathcal{C}_{q_1}$ may be found if one decided to travel from q_2 to q_1. Assume that we call $q_{2_{cusp}}$ the first cusp point on $\mathcal{C}_{q_2}$ and $q_{1_{cusp}}$ the equivalently obtained counterpart on $\mathcal{C}_{q_1}$. Hence we have chosen to interpolate between these points in order to define a new cusp point between q_1 and q_2. This new cusp point q_{cusp} is shown in Figure III.5.

The new planner so built respects **TP**. The proof follows.

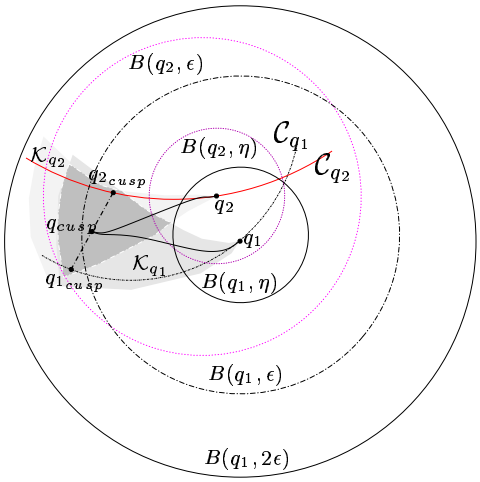


Figure III.5: Modification to the local planner allowing for a symmetrical cusp point and respecting the topological property.

From the design of the local planner ([SLL+97]) we know that the following assertion

holds:

$$\forall \epsilon,\ \exists \eta \text{ such that } \forall q_0 \in \mathcal{CS},\ \forall q \in B(q_0,\eta),\ \mathcal{C}_q \cap \mathcal{K}_{q0} \neq \emptyset$$

and we have a way for choosing q_{cusp} in this intersection.

Let $d(\cdot,\cdot)$ be the Euclidean distance. If $d(q_1,q_2) < \eta$ then

$$\begin{aligned} \mathcal{C}_{q_1} \cap \mathcal{K}_{q_2} \neq \emptyset \Rightarrow q_{1_{cusp}} \in \mathcal{K}_{q_2} \\ \mathcal{C}_{q_2} \cap \mathcal{K}_{q_1} \neq \emptyset \Rightarrow q_{2_{cusp}} \in \mathcal{K}_{q_1} \end{aligned}$$

It follows that

$$\begin{aligned} q_{1_{cusp}} \in \mathcal{C}_{q_1} \subset \mathcal{K}_{q_1} \quad &\Rightarrow \quad q_{1_{cusp}} \in \mathcal{K}_{q_1} \cap \mathcal{K}_{q_2} \\ q_{2_{cusp}} \in \mathcal{C}_{q_2} \subset \mathcal{K}_{q_2} \quad &\Rightarrow \quad q_{2_{cusp}} \in \mathcal{K}_{q_1} \cap \mathcal{K}_{q_2}. \end{aligned}$$

On the other hand, let us remind that the structure of a cone $\mathcal{K}_{q_i}$ is such that any segment linking any point of the cone to any point of its main axis $\mathcal{C}_{q_i}$ remains inside the cone. This entails that

$$[q_{1_{cusp}}, q_{2_{cusp}}] \in \mathcal{K}_{q_1} \cap \mathcal{K}_{q_2}$$

Now chose any point $q_{cusp} \in [q_{1_{cusp}}, q_{2_{cusp}}]$. Let us define $\mathcal{C}_{q_{cusp},q_2}$ as the curve obtained by interpolation (III.30) between the canonical curves respectively associated to q_{cusp} and q_2. Then by construction ([SLL+97]) we have:

$$\mathcal{C}_{q_1,q_{cusp}} \subset B(q_1,\epsilon) \; and \; \mathcal{C}_{q_{cusp},q_2} \subset B(q_2,\epsilon),$$

and since

$$\mathcal{C}_{q_1,q_2} = \mathcal{C}_{q_1,q_{cusp}} \circ \mathcal{C}_{q_{cusp},q_2},$$

we conclude that

$$\forall \epsilon > 0,\ \exists \eta > 0 \text{such that} \forall q_1,\ \forall q_2 \in B(q_1,\eta),\ \mathcal{C}_{q_1,q_2} \subset B(q_1,2\epsilon).$$

This is illustrated in Figure III.5. □

What we have shown is that the new way of choosing q_{cusp} leads to a planner that still respects **TP**.

In this way we have achieved the following improvements. Firstly, we have turned the local planner into a symmetrical one—i.e. in either direction: $q_1 \to q_2$ or $q_2 \to q_1$ the planner computes the same path. Secondly, we avoid to execute the second half of the path by following an arc of circle (which was not necessarily optimal). Finally, the interest is that we use one cusp point for two interpolations instead of one. Hence if the topology of $\mathcal{CS}$ allows for it, this enhances the capability to reach a final point by using fewer cusps.

As mentioned we adapted the motion planning scheme [LSL99] by using this new local planner, based on the flat output we have computed for the bi-steerable car. In this respect, it constitutes the first complete path planning scheme for a bi-steerable robot. In the following section we present simulation results validating the claim.

6.4 Simulation results

The flat output introduced in this chapter allowed us for computing feasible paths for the bi-steerable car as demonstrated by the simulations performed using the motion planning scheme discussed above. Essentially, the motion planner first builds a collision-free holonomic path using the *Random Path Planning RPP* [BL91] approach. Then, this path is approximated by a sequence of collision-free feasible paths computed by the new steering method as proposed in section 6.3. Roughly speaking, the algorithm consists in trying first to connect the initial and final configurations through a feasible (collision-free) path. If this attempt fails, the program picks up the final configuration and an intermediate configuration, selected randomly amongst those created during the holonomic search. Hence the local planner computes the corresponding (collision-free) path in the flat space. Finally, the resulting path is smoothed. From this path it is possible to compute a feasible trajectory for the robot (this will be discussed in the forthcoming chapter).

We have conducted simulations with a rear steering function $f(\varphi) = k \cdot \varphi$. Figure III.6 shows an elementary path computed by the steering method as proposed by [SLL+97] (left image) and the resulting path using the cusp-point modification as proposed in section 6.3.

Figure III.6: Left: Elementary path with a cusp point as proposed in Figure III.4; Right: The corresponding path when the cusp point is computed as shown in Figure III.5

The reduction of the number of maneuvers using the new approach to compute the cusp point is validated as shown in Figure III.7.

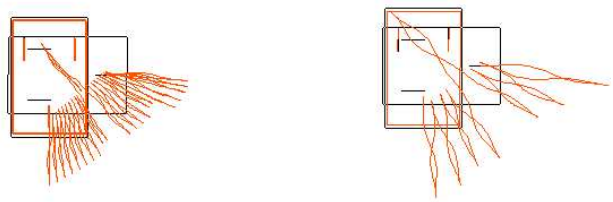


Figure III.7: The difference in maneuvering using "old" (left side) cusp points (46 paths) against "new" (right side) cusp points (16 paths) without smoothing.

The new local planner was integrated in the global planning scheme. Figure III.8

shows the maneuvering enhancements induced by the bi-steerable capability (right image) against the conventional car-like robot (left image). The difference between negative against positive rear deflection of the wheels with respect to the front steering angle is depicted in Figures III.9 and III.10. Notice that the maneuvering capacity is reduced with positive rear deflection but larger (wide-open) turns are achieved— i.e. paths with a minor curvature derivative are obtained.

Figure III.8: Smoothed feasible paths for a car-like robot (left image) $f(\varphi) = k \cdot \varphi$, $k = 0$ and a BiS-car

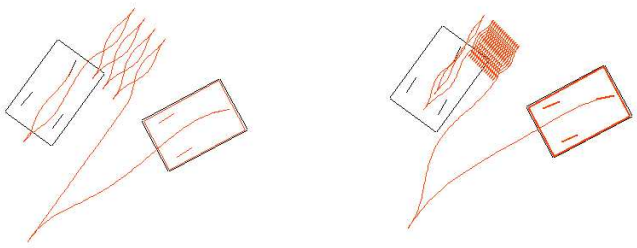


Figure III.9: Maneuvering ability with negative (left) and positive (right) steering with $|k| = 0.3$.

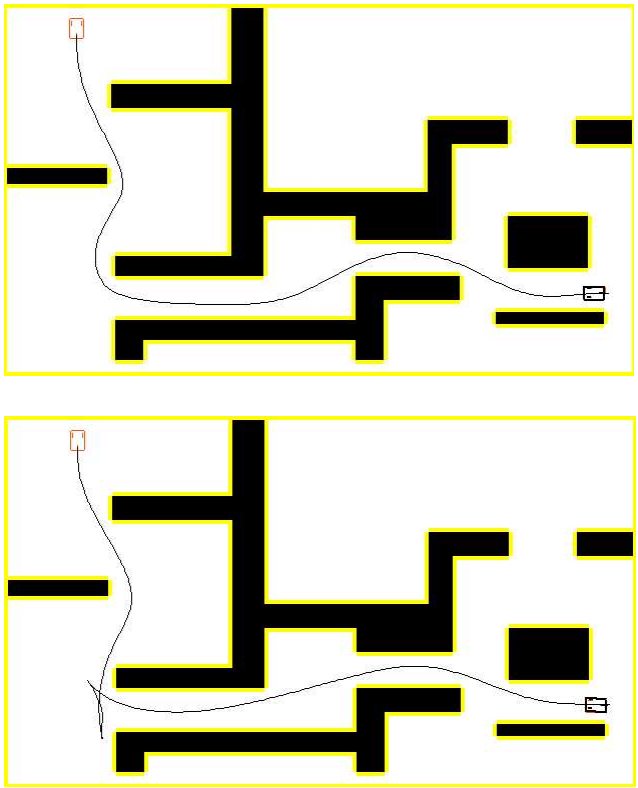


Figure III.10: Feasible paths with negative (left) and positive (right) rear steering angle ($|k| = 0.3$).

7 Conclusions

In this chapter the problem of finding the flat output of the bi-steerable car was addressed from a differential geometric point of view. The idea was to look for a way to *parameterize integral curves* solving the system of differential equations, defining the kinematic constraints.

For mobile robots, exterior differential equations arise directly from the nonholo-

nomic constraints. Hence the problem was formalized in the setting of Pfaffian systems. The theory of Pfaffian systems allows to characterize those systems which can be put in a *normal form* by a change of coordinates. The normal form corresponds to a different representation of the original system, expressed with respect to a new set of coordinates. The flat output of the system belongs to this new set. Solutions to the normal form can be obtained from arbitrary functions assigned to the flat output. The original system integral sub-manifolds are then obtained from inverse coordinate transformations.

Since the Pfaffian system associated to the bi-steerable car contains two equations and four variables, finding the flat-output consisted in finding the change of coordinates associated to its Engel normal form. However, the literature includes very few works on the effective computation of such normal form and the flat output.

The work presented in this chapter aimed at bringing a contribution in this direction. We give a *necessary* condition based on a set of PDE, allowing to find the flat coordinate transformations in order to write a 2-dimensional codistribution in its Engel normal form.

We applied the methodology for explicitly computing the flat output of the bi-steerable car. By inspection of the PDE resulting from the systematic approach and some symmetry considerations, the clue was to spot a turning frame with respect to the reference frame of the robot, so that computable expressions of the flat output followed. Symmetry considerations again led subsequently to the inverse expressions allowing us to compute the state variables of the robot from the flat output and a number of its derivatives.

This flat output allowed us to compute feasible paths between points of the free configuration space. In this sense it constitutes half the way to the solution of the motion planning and control of the bi-steerable car. However, the path planner introduced in this chapter constitutes the first planner for a bi-steerable system with general rear steering function $f(\varphi)$.

In the following chapter we will address the second half of the problem and in particular we will address the equivalence issue. Indeed, a nonlinear flat system is equivalent to a linear controllable one by a special class of dynamic feedback [FLMR95b]. We will have therefore to switch to the theory of control of nonlinear systems where flatness plays a fundamental role.

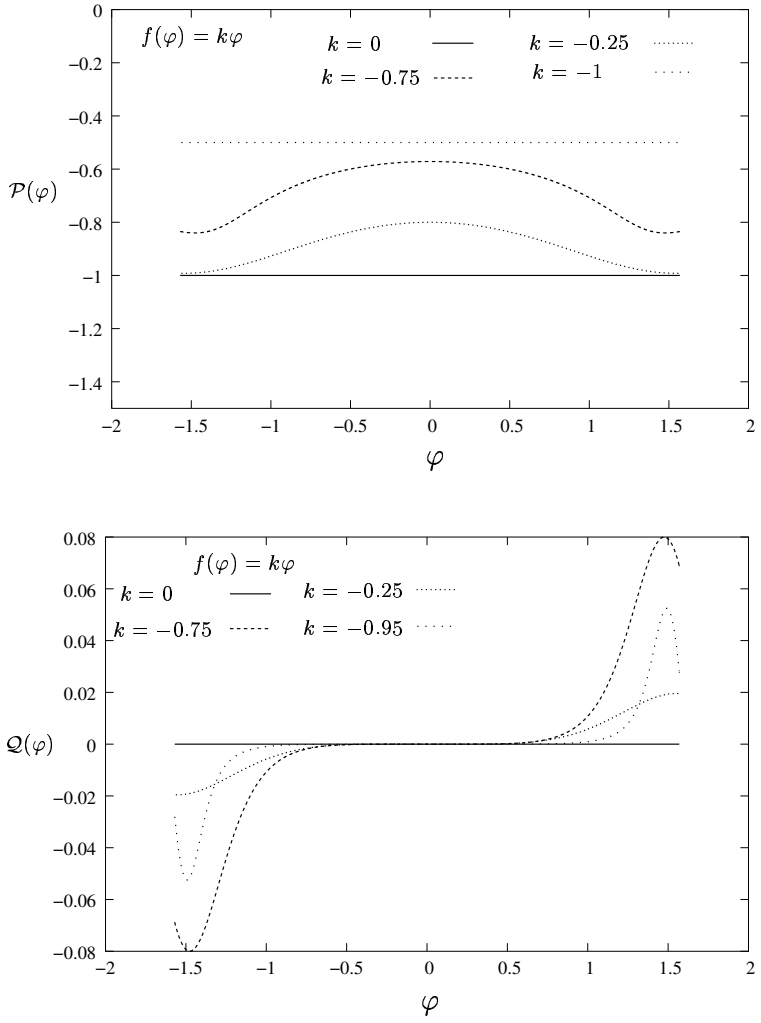


Figure III.11: Flat coordinates $\mathcal{P}(\varphi)$and $\mathcal{Q}(\varphi)$ when $f(\varphi) = k\varphi$, $k \leq 0$ and $L = 1$. We remark that when $k = 0$ and $k = -1$ the coordinate $\mathcal{Q}(\varphi)$ in both cases coincides.

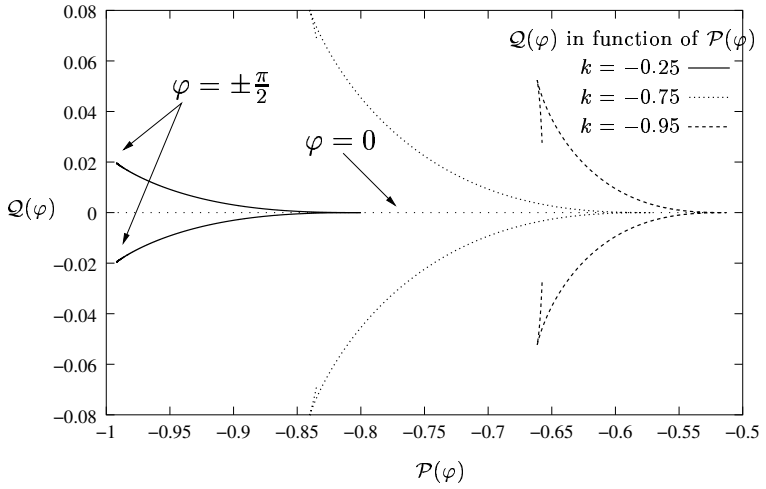


Figure III.12: $\mathcal{Q}(\varphi)$ in function of $\mathcal{P}(\varphi)$ for $-\frac{\pi}{2} < \varphi < \frac{\pi}{2}$ when $k < 0$.

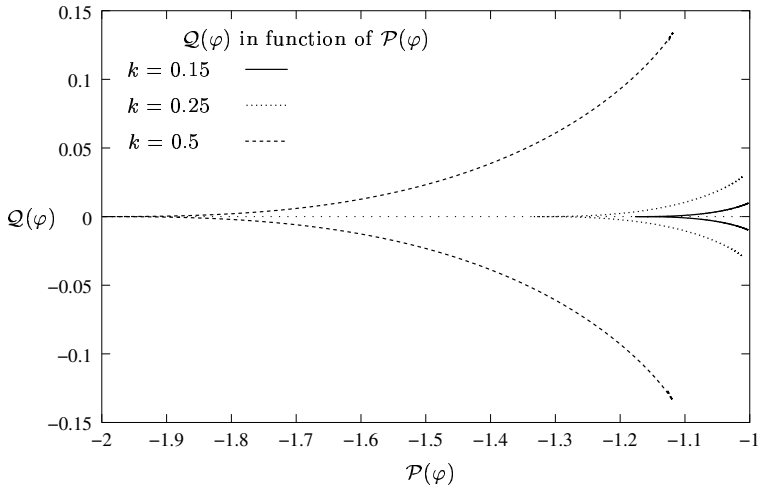


Figure III.13: $\mathcal{Q}(\varphi)$ in function of $\mathcal{P}(\varphi)$ for $-\frac{\pi}{2} < \varphi < \frac{\pi}{2}$ when $k > 0$.

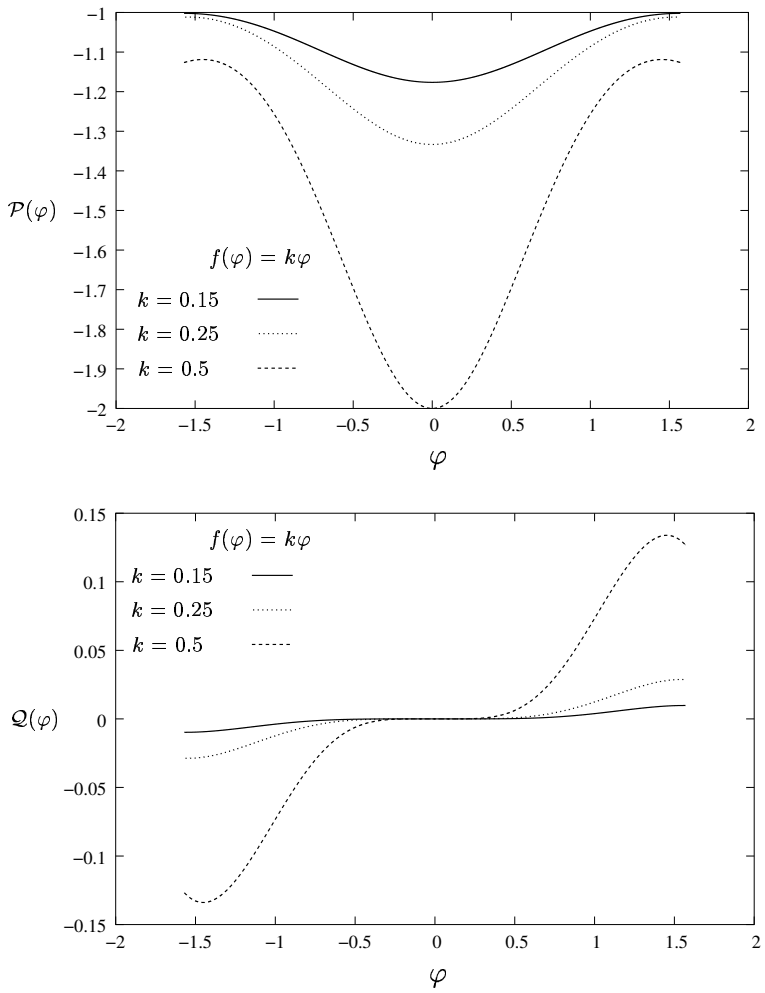


Figure III.14: Flat coordinates $\mathcal{P}(\varphi)$and $\mathcal{Q}(\varphi)$ when $f(\varphi) = k\varphi$, $k \geq 0$, $L = 1$.

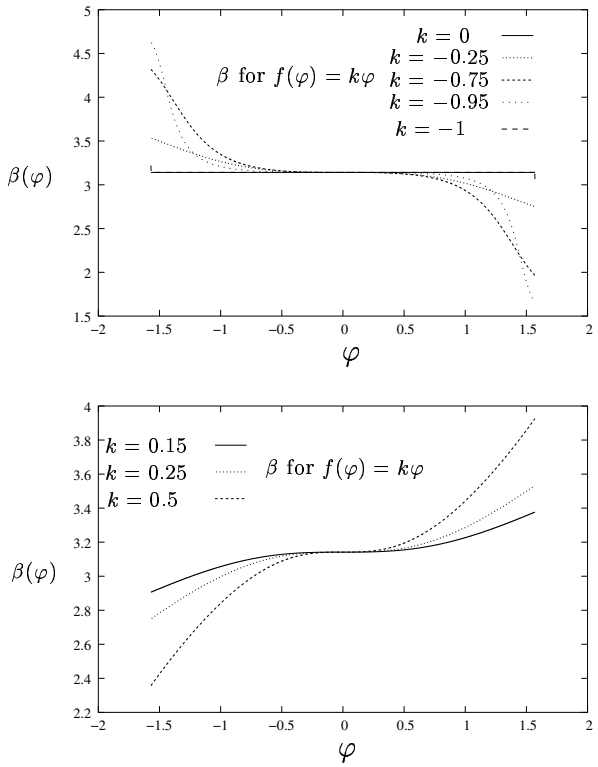


Figure III.15: Characteristic angle $\beta(\varphi)$ when $f(\varphi) = k\varphi$ $-1 \leq k < 1$. We remark that β is around π indicating that the flat output is "behind" the point F; in other words, the same values of coordinates $\mathcal{P}(\varphi)$ and $\mathcal{Q}(\varphi)$ with opposite sign would have been obtained if the reference frame of the robot had been placed at point R, in which case β varies in a neighbourhood of 0.

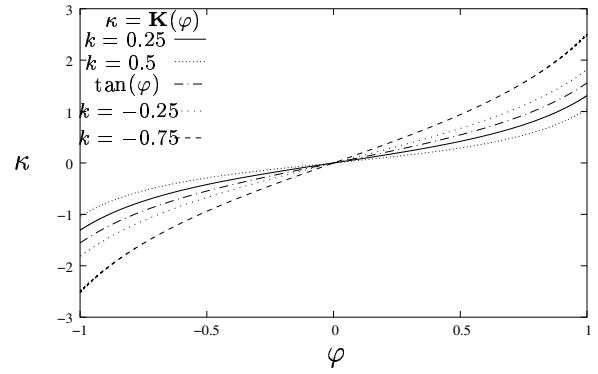


Figure III.16: $\kappa = \mathbf{K}(\varphi)$. Notice that the function $\tan(\varphi)$ is practically a "symmetry axis" between curvatures for $k > 0$ and $k < 0$.

Chapter IV

Endogenous Equivalence and Linearization of the BiS-car

Up to this point we have been able to compute feasible geometric paths for the general bi-steerable system. The aim of this chapter is to provide means for planning trajectories and synthesizing feedback control laws. In this respect our contribution is as follows:

- We establish results concerning the equivalence between the bi-steerable car and its underlying controllable linear system. In particular the linearizing feedback and the explicit formulation of the chained form of the system are deduced.
- We solve the trajectory tracking issue for the general bi-steerable car, by stabilizing its equivalent linear system, and have opened the possibility to solve the regulation problem by formulating explicitly its chained form [HS03]. To our knowledge, our work is the first to tackle these problems regarding bi-steerable systems.

1 Introduction

The subject of this chapter is the analysis of the bi-steerable system structure as a nonlinear control system. The end objective is to synthesize a *feedback control law* (or compensator) allowing us to perform trajectory tracking. In general, the problem may stated as follows: given the nonlinear control system

$$\dot{x} = f(x, u),$$

whose output function

$$y = \lambda(x)$$

follows a particular trajectory $y(t)$, compute the trajectory $(x(t), u(t))$ for $t \geq 0$ (knowing that the output's behaviour is influenced by the input through x). In other words we are facing an inversion problem whose solution involves the design of a feedback control law.

In general, every nonlinear system is a particular case when looking for solutions to any control task. In Nonlinear Control it is therefore desirable to provide means for classifying control systems, allowing to give them a *normal form* of special interest. This normal form allows us to elucidate several important properties of the system, and in particular to synthesize feedback laws adapted to the control task [Isi95].

A particular normal form of special interest is the so called *Brunovsky canonical form* of linear systems (see [Kai80] for a comprehensive description of canonical forms for linear systems). A linear system written in this form consists in a chain of pure integrators as shown in Figure IV.2. A nonlinear control system potentially transformable into a linear one by means of feedback is said to be *feedback linearizable*.

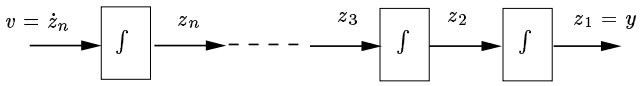


Figure IV.1: Brunovsky canonical form of a linear control system.

Feedback linearization is a powerful tool in nonlinear control and has received considerable attention during the last 30 years (see for instance [JR80], [Isi95], [MR94], [Pom97]). The interest is twofold. Firstly, from a mathematical viewpoint (system theory) there is an interest for the classification of nonlinear systems under feedback transformations so that the problem is to characterize those nonlinear systems which are feedback equivalent to linear ones. On the other hand, if one is able to compensate nonlinearities by feedback then the modified system possesses all control properties of its linear equivalent and linear control theory can be used in order to study it and(or) to achieve the desired control properties.

Flatness is the outcome of research aiming at characterizing nonlinear systems which are equivalent to linear ones, by means of a restricted class of dynamic feedbacks called endogenous ([FLMR99]). In this respect [Mar92] and [FLMR95b] introduce the notion of *endogenous equivalence*, providing a framework to the study of system classification and linearization via this kind of feedbacks (see section 3).

In this chapter we are interested in going more deeply into the *differential* flatness of the bi-steerable car. On one hand, the aim is to establish the equivalence between the system and its underlying linear structure. On the other hand, we aim at finding means to synthesize a control law allowing us to stabilize the system trajectories around a nominal reference.

The fact is that both problems have been partially solved with the results obtained in the previous chapter. Indeed, we have already found coordinates allowing us to parameterize solutions of the differential equations, defining the system constraints, for given initial conditions. In other words we have established up to this point results concerning state-space transformations only. However, in the setting of flatness, equivalence between any two systems F and G entails a one-to-one correspondence between the trajectories of F and those of G (see section 3). Hence we shall further investigate the dynamics of both the flat output and the bi-steerable systems, in order to set trajectories equivalence results. In support to our analysis, we shall rely on results established by [Mar92] and

[FLMR99].

On the other hand, stabilizing trajectories is a control design problem involving some kind of feedback which solution depends on the type of system and the trade-offs between *stability* and *robustness*. Since the bi-steerable car is flat, a natural solution to the problem is to exploit the properties of the linearizing output. In this respect, we shall find the equivalent linear system of the bi-steerable car in order to synthesize a feedback control law. The chapter is structured as follows.

We start by introducing the formulation of the feedback linearization problem in general and giving a brief recall on exact state feedback linearization in section 2. This will lay down the grounds required to understand the principle of dynamic feedback linearization discussed in section 3. Then in section 4 we introduce the methodology followed in order to achieve our goals. Hence equivalence results concerning the BiS-car are presented in section 5. These results allowed us also to compute the explicit formulation of the chained form of the system. The interest in the latter stems from the possibility to use results found in the growing literature about the control of nonholonomic systems put in this form (see e.g. [MS97]). We move on to section 6 where we discuss two possible ways for computing the linearizing feedback. As a consequence, we deduce a feedback control law allowing for trajectory tracking. Finally in section 7 we show by simulation results that the linearization of the BiS-car allows us to stabilize the 4-dimensional system trajectories.

Throughout all the chapter we will use notations and definitions found in the Appendix A.

2 Introduction to feedback linearization

2.1 General notions

We start this section by formulating the problem of feedback linearization in the general case (see e.g. [CLM89], [MR94]).

The problem of feedback linearization of a (smooth) control system

$$\dot{x} = f(x, u)$$

defined on an open subset $X \times U$ of $\mathbb{R}^n \times \mathbb{R}^m$ consists in finding a (smooth) dynamic feedback (or dynamic compensator)

$$\begin{aligned} u &= a(x, z, v) \\ \dot{z} &= b(x, z, v) \end{aligned}$$

defined on an open subset $\tilde{X} \times Z \times V$ of $X \times \mathbb{R}^p \times \mathbb{R}^q$ such that the closed-loop system

$$\begin{aligned} \dot{x} &= f(x, a(x, z, v)) \\ \dot{z} &= b(x, z, v) \end{aligned}$$

is diffeomorphic on $\tilde{X} \times Z$ to a controllable linear system.

To be precise, one says that the system is *linearizable at a point* (x^0, u^0) if $\tilde{X}$ is a neighbourhood of x^0 and $a(\tilde{X}, Z, V)$ is a neighbourhood of u^0. And say that it is *linearizable* if it is linearizable at every point of a dense subset of $X \times U$. Clearly, when a system is linearizable at a point, it is linearizable on a neighbourhood of that point. The following example (taken from [MR94]) should be illustrative.

Example

Consider the driftless nonlinear control system

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2\, u_1 \end{aligned}$$

It is linearizable using the feedback

$$\begin{aligned} \dot{z}_1 &= v_1 \\ u_1 &= z_1 \\ u_2 &= \frac{v_2 - x_2\, v_1}{z_1} \end{aligned}$$

and the diffeomorphism

$$(x_1, x_2, x_3, z_1) \mapsto \xi := (\xi_1, \xi_2, \xi_3, \xi_4) = (x_1, z_1, x_3, x_2\, z_1)$$

Indeed, one can check that, in the new coordinates, the system becomes:

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= v_1 \\ \dot{\xi}_3 &= \xi_4 \\ \dot{\xi}_4 &= v_2, \end{aligned}$$

which is obviously linear. Hence the system is linearizable only at points (x^0, u^0) such that $u_2^0 \neq 0$. In particular the system is not linearizable at equilibrium points. □

Remark that the assumption is that the state x of the system is available for measurements, and we let the input of the system to depend on this state and, possibly, on external reference signals.

Back to our problem definition, if the value of the control at time t depends only on the values, at the same instant of time, of the state x and of the external reference input, then the control is said to be a *Static State Feedback Control.* This translates into $dim\{z\} = 0$; i.e. the feedback is memoryless and hence can be viewed as a particular case. Otherwise, as is the case of the example above, we say that a *Dynamic State Feedback Control* is implemented.

In terms of characterization of the problem, the simplest case concerns the static one. From a theoretical viewpoint, the problem of static feedback linearization was solved by Jakubczyk and Respondek [JR80]. This led to a complete characterization of those

nonlinear systems which can be transformed into a linear system by means of state space diffeomorphisms

$$\xi = \Phi(x)$$

and static feedback.

Perhaps a brief review concerning the underlying concepts leading to the solution of the static feedback linearization problem can be of utility in understanding the steps discussed further ahead in this chapter. In what follows, we shall therefore discuss fundamental notions and few nonlinear control techniques in order to achieve full linearization of a general nonlinear system, via state static feedback.

2.2 Static state feedback linearization

In this situation $dim\{z\} = 0$. The point of departure of the whole analysis is the notion of relative degree of the system [Isi95].

2.2.1 Relative degree of a nonlinear control system

The nonlinear control system

$$\dot{x} = f(x) + g(x)u \tag{IV.1}$$

with output

$$y = \lambda(x)$$

has relative degree r at point x^0 if:

(i) $L_g L_f^k \lambda(x) = 0$ for all x in a neighborhood of x^0 and all $k < r - 1$

(ii) $L_g L_f^{r-1} h(x^0) \neq 0$.

Interesting remarks from [Isi95] elucidate this notion. In particular, let us suppose that at some time t^0 the system is at state $x(t^0) = x^0$. Assume we want to calculate the value of $y(t)$ and of its derivatives with respect to time $y^{(k)}(t)$ for $k = 1, 2, \ldots$ at $t = t^0$. It is worth noticing that the relative degree r is exactly equal to the number of times one has to differentiate the output $y(t) = \lambda(x(t))$ at time $t = t^0$ in order to have the value $u(t^0)$ of the input explicitly appearing.

This implies that if no relative degree is found for any $k \geq 0$, the Taylor series expansion of $y(t)$ at the point $t = t^0$ takes the form

$$y(t) = \sum_{k=0}^{\infty} L_f^k \lambda(x^0) \frac{(t - t^0)^k}{k!},$$

which means that $y(t)$ is a function depending only on the initial state and not on the input.

As a matter of fact, it can be shown (see e.g. [Isi95]), that the functions

$$\lambda(x^0), L_f\lambda(x^0), \ldots, L_f^{r-1}\lambda(x^0)$$

are linearly independent and hence can be used in order to define, at least partially, a local coordinates transformation around x^0.

In this setting, linearization results follow easily as discussed next.

2.2.2 Coordinate transformation and static feedback

In the ideal case the relative degree $r = n$ and hence one can obtain a fully linearized system. We will show next how to perform the linearization. In fact, the methodology is constructive and outlines the linearization process in the "general case"; that is, when the suitable "output function" has been obtained.

In view of the discussion of the previous section, it is possible to define, locally around x^0, a coordinate transformation taking the form

$$\xi = \Phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_r(x)) = (\lambda(x), L_f\lambda(x), \ldots, L_f^{r-1}\lambda(x)).$$

The linearization is henceforth realized as follows. First notice that

$$\frac{d\xi_1}{dt} = \frac{\partial\lambda}{\partial x}\frac{dx}{dt} = L_f\lambda(x(t)) = \xi_2(t).$$

Furthermore, as a consequence of $r = n$:

$$\frac{d\xi_n}{dt} = L_f^n\lambda(x(t)) + L_gL_f^{n-1}\lambda(x(t))u(t), \tag{IV.2}$$

so that in ξ coordinates one has $\dot{\xi}_n = b(\xi) + a(\xi)u$, with $a(\xi)$ nonsingular by definition (recall that the analysis is done around x^0 where $L_gL_f^{n-1}\lambda(x) \neq 0$).

Hence the system (IV.1) will be described, after coordinate change, by equations of the form

$$\begin{array}{lcl} \dot{\xi}_1 & = & \xi_2 \\ \dot{\xi}_2 & = & \xi_3 \\ & \cdots & \\ \dot{\xi}_{n-1} & = & \xi_n \\ \dot{\xi}_n & = & b(\xi) + a(\xi)u. \end{array}$$

Suppose now that one chooses the following state feedback control law

$$u = \frac{1}{a(\xi)}(-b(\xi) + v) \tag{IV.3}$$

which is obtained by inversion of (IV.3) after setting $\dot{\xi}_n = v$. We have the right to do this since, locally around $\xi^0 = \Phi(x^0)$, $a(\xi) = a(\Phi(x)) \neq 0$. Notice that in original coordinates, the feedback reads:

$$u = \frac{1}{L_gL_f^{n-1}\lambda(x)}(-L_f^n\lambda(x) + v),$$

Thus we have constructed a feedback of the form

$$u = \frac{1}{a(\Phi(x))}(-b(\Phi(x)) + v) = \alpha(x) + \beta(x)v. \tag{IV.4}$$

with β nonsingular.

Therefore system (IV.1), after coordinate change and static feedback, becomes the following linear and controllable system in Brunovsky canonical form—i.e. a chain of n integrators:

$$\begin{array}{rcl} \dot{\xi}_1 & = & \xi_2 \\ \dot{\xi}_2 & = & \xi_3 \\ & \cdots & \\ \dot{\xi}_{n-1} & = & \xi_n \\ \dot{\xi}_n & = & v. \end{array} \tag{IV.5}$$

Notice that the linearization is full; that is, the dimension of the state-space in ξ coordinates is the same as that of the original control system, and therefore we say that *exact linearization* is achieved.

A compelling question is to ask under which conditions the above transformations can be undertaken. The following section states necessary and sufficient conditions for exact static state feedback linearization for a single-input single-output nonlinear control system[1].

2.3 Conditions for exact static state feedback linearization

Theorem 2.1 ([JR80]) *Suppose a system*

$$\dot{x} = f(x) + g(x)u \tag{IV.6}$$

is given. The State Space Exact Linearization Problem (SELP) is solvable near a point x^0 (i.e. there exists an "output" function $\lambda(x)$ for which the system has relative degree n at x^0) if and only if the following conditions are satisfied

(i) the matrix $\left[g(x^0)\ ad_f g(x^0)\ \ldots\ ad_f^{n-2} g(x^0)\ ad_f^{n-1} g(x^0)\right]$ has rank n,

(ii) the distribution $D = span\{g, ad_f g, \ldots, ad_f^{n-2} g\}$ is involutive near x^0.

It is worth remarking that the condition (i) has the following interesting interpretation. Suppose the vector field $f(x)$ has an equilibrium at $x^0 = 0$. The *linear approximation* of the system (IV.6) at $x = 0$ reads:

$$\dot{x} = Ax + Bu,$$

where

$$A = \left[\frac{\partial f}{\partial x}\right]_{x=0} \qquad B = g(0).$$

[1] We will just recall established results for a single-input single output system, since the solution for the multiple-input multiple-output problem, under certain conditions, follows easily (see e.g. [Isi95]).

It can be shown (e.g. see [Isi95]) that condition (i) of Theorem 2.1 (written at $x^0 = 0$) is equivalent to the condition

$$rank\big(B, AB, \ldots, A^{n-1}B\big) = n.$$

That is, (i) implies that a necessary condition for the solution of (SELP) is that the linear approximation of the nonlinear control system must be controllable at $x = 0$.

At this stage, a last remark about the implication of theorem 2.1 is imperative. If *(ii)* holds the distribution D is involutive. By the Frobenius theorem (e.g. see Appendix A) the system is integrable. In the general (multi-dimensional) case, the equivalent condition applies to the filtration of the system's distribution (see [Isi95, Theorem 5.2.3, p.233]). Accordingly, we can conclude that nonholonomic systems (and generally those systems which do not verify the conditions above[2]), are not (fully) static state feedback linearizable.

What this implies is that, for such systems, it is however possible to linearize a subsystem at the expense of leaving the remaining part of the system nonlinear. In this respect, there exist many important results in the literature of theoretical control (see e.g. [Isi95], [CLM91]) addressing this and related problems, such as input-output decoupling (i.e. in a multiple-input multiple-output system each output channel is only affected by a single input), and full linearization via dynamic feedback.

In what follows, we shall discuss few results on dynamic feedback linearization in order to introduce the benefits of flatness in this context.

3 Feedback linearization, flatness and equivalence

3.1 Dynamic feedback linearization

3.1.1 Full linearization via dynamic compensation

As anticipated, full linearization via state static feedback cannot be achieved for nonholonomic systems in general. This is because the integrability condition is not satisfied. In other words, we can not conclude on the existence of an "output function" yielding a relative degree $r = n$. The rather fundamental subtlety is that the relative degree is invariant under static feedback and state space transformations [Isi95]. The idea is therefore to resort to dynamic feedbacks in order to achieve the required relative degree. The following discussion about the work of [dNBC92] illustrates the point.

One of the first results concerning this technique, applied in the robotics field, is due to d'Andréa-Novel, Bastin and Campion [dNBC92]. In order to achieve full linearization of wheeled mobile robots, including the car-like (tricycle model), [dNBC92] proposed, based on results from [JR80], "linearizing outputs" and dynamic compensators of the

[2]Or simply systems verifying the conditions but for which the relative degree is $r < n$

form

$$\begin{aligned} u &= \alpha(x,z) + \beta(x,z)v \\ \dot{z} &= a(x,z) + b(x,z)v. \end{aligned}$$

β being a nonsingular matrix and v an auxiliary input.

The authors consider 3 kinds of robots whose dynamical models were systems with drift, described by smooth vector fields of the form:

$$\Sigma : \qquad \dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i \quad x \in \mathbb{R}^n \; u \in \mathbb{R}^m.$$

The principle of the approach was very similar to what we have discussed above in the sense that suitable output functions yield the relative degree required. Hence the first step consisted in choosing output functions

$$y_i = \lambda_i(x), \;\; i = 1, \cdots, m$$

and then apply the *dynamic extension algorithm* [DM85] (e.g. see [Isi95]) on Σ and y_i. A brief digression is necessary at this stage in order to explain this algorithm.

Dynamic Extension Algorithm This algorithm finds its origins in the *structure algorithm* [Sil69, Hir79]. The idea is to delay the appearance of some inputs (or combination of inputs) to higher derivatives of the output, hoping with this that new inputs will show up in a decoupled manner—i.e. such that the output has full rank with respect to the inputs. The delay consists in adding integrators to those inputs (or combination of inputs) of which the output is fully dependent. Adding integrators in this way amounts to consider the dynamics of a new system driven by an auxiliary input and having its own state. To illustrate this, consider an example where a 2-input (u_1, u_2) system is assumed to have state x, and a 2-dimensional output function $y = (y_1, y_2)$ whose first derivative depends only on u_1. An integrator, driven by the auxiliary input v_1, is added to u_1 so that the new extended state becomes (x, ζ) and now the output has full rank with respect to (u_1, u_2).

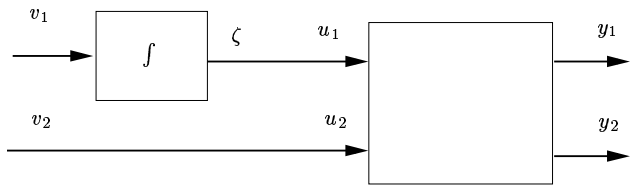


Figure IV.2: Dynamical Extension Algorithm in a simple case (taken from [Isi95]).

Back to the work of [dNBC92], the method consisted in applying the algorithm to y_i in order to have a decoupled system of the form:

$$y_k^{(\rho_k)} = w_k, \;\; k = 1, \cdots, m,$$

where $y_k^{(i)}$ denotes the (i)-th derivative of y_k with respect to time, ρ_k is the relative degree of Σ and w_k is the new auxiliary input. The decoupled system is such that each output is only affected by a single control input.

In order to get full linearization, a n_e-dimensional extended system is required. Here, n_e is the dimension of the original system plus the number of added integrators and is such that:

$$\sum_{i=1}^{m} \rho_i = n_e.$$

This example shows the interest of using dynamic feedbacks looking for full linearization. As for static feedbacks, in theoretical control the objective is to give conditions under which general systems may be linearized. Let us discuss some significant results about this subject.

3.1.2 Conditions for dynamic feedback linearization

In contrast to the static case, characterization of dynamic feedback linearization is still an open problem in the general case [MR94]. Notwithstanding, we have at our disposal a certain number of significant results.

Let us cite for instance [CLM89]. In that note, the authors showed that for single-input (general) control systems static and dynamic feedback linearizability are equivalent. Moreover, they showed that a driftless nonlinear control system is never linearizable at an equilibrium point, since its linear approximation is not controllable. Later, the same authors gave in [CLM91] sufficient conditions for dynamic feedback linearization of control systems with drift, by using and dynamic compensators and extended state space diffeomorphisms (e.g. see the example above):

$$\xi = \Phi(x, z)$$

By the mid 1990's [MR94] recast the formulation of the problem using maps with some adequate properties rather than feedbacks. The outcome was a sufficient condition on the derived flag for feedback linearization of driftless control systems. Basically, if the driftless system states live in $\mathbb{R}^n$ and the inputs are defined on $\mathbb{R}^m$, the condition says that if the derived flag satisfies $dim\, I^k(x) = n - m - k$ for $k = 0, \ldots, n - m$ then the system is feedback linearizable. The condition proved also to be necessary in the 2 input case.

As a direct result in this case, systems verifying the flag condition are flat and can be put into the chained form by static (and invertible) feedback [MR94]. The chained form is interesting because it may be used as a local model for a linearizable driftless system around a singular point, such as an equilibrium point, where the linear approximation is not controllable.

All these results account for the importance conferred to the problem of feedback linearization. Notwithstanding, this problem may be seen as part of a more fundamental

issue which concerns system equivalence and system classification. In this respect, flatness has played a fundamental role as discussed in the following section.

3.2 Feedback linearization, system equivalence and classification

As a matter of fact, static-state feedbacks and diffeomorphisms provide us with a natural transformation group preserving the state manifold [FLMR99], [MR94]. Two systems are said to be equivalent with respect to this group if they can be transformed into each other by some element of the group. The concept of a group, arises from the algebraic abstraction of the notion of symmetry (e.g. see [Lan86]). Symmetry is a notion common to algebra and geometry, where an important example is $SO(2)$: the group of rotations in the plane. In the theory of differential equations, the symmetry group of a system is the largest *local* (i.e. expressed in local coordinates) *group of transformations* acting on the independent and dependent variables of the system, with the property that it transforms solutions of the system to other solutions. Moreover, the system of differential equations is invariant under some local group transformations (in particular the *prolonged group*)[3] [Olv93].

As far as dynamic feedbacks are concerned, there are few results (e.g. see [W.F90], [NRM95]) linking dynamic feedback linearization with the notion of *absolute equivalence* from Elie Cartan [Car14]. However, in the general case there are simple examples that show the corresponding transformation may not be one-to-one and hence dynamic feedbacks do not form a group [FLMR99]; therefore equivalence results cannot be established.

In this context, the general notion of *endogenous equivalence* ([Mar92],[FLMR95b]) has provided a formalization framework of system classification and linearization by a restricted class of dynamic feedbacks called endogenous: two systems F and G are equivalent if there is a bijective correspondence between the trajectories of F and those of G. Moreover, if F and G are two equivalent systems, one can find an *endogenous* dynamic feedback and a diffeomorphism which transforms F into G *extended* by pure integrators (see section 4).

In particular, a flat system is linearizable by means of dynamic feedback and coordinate change. By definition, a control system is flat if it is equivalent, in the previous sense, to a system without dynamics, described by a collection of independent variables, the flat output, having the same number of components as the number of control variables [FLMR99]. Besides, linear controllable systems with the same number of inputs are proved (see the latter reference) to be equivalent, in this sense, to such system without dynamics. Hence dynamic feedback linearization results are deduced.

Moreover, the linearization is exact and since the feedback is endogenous it is invertible ([MR94]), except probably at singular points. The main property of an endogenous feedback is to be reversible up to pure integrators. This means that equivalence by endogenous feedback respects many important properties of the system such as controllability and linearizability [Mar92].

In view of this discussion, there is a fundamental reason why we should address

[3] See section 4.1 for the notion of prolongation.

the feedback linearization problem of the bi-steerable car in the framework of flatness: a complete and formal system classification may be done. Before we address this problem, namely the equivalence issue concerning the bi-steerable car, we shall lay down first the methodological grounds that guided our work. This is the subject of the following section.

4 Endogenous equivalence

We need to give first formal definitions of additional concepts and introduce some notation required later . Most of the following definitions can be found in [Olv93].

4.1 Prolongation and jets

Given a smooth real-valued function $\lambda(x) = \lambda(x_1, \ldots, x_p)$ of p independent variables, there are

$$p_k \equiv \left(\begin{array}{c} p+k-1 \\ k \end{array} \right) = \frac{(p+k-1)!}{k!(p-1)!}$$

different k-th order partial derivatives of λ. The multi-index notation

$$\partial_J \lambda(x) = \frac{\partial^k \lambda(x)}{\partial x_{j_1} \partial x_{j_2} \cdots \partial x_{j_k}} \lambda$$

stands for the partial derivative of λ of order k, with respect to the j-th variable; where $J = (j_1, \ldots, j_k)$ is an unordered k-tuple of integers, with entries $1 \leq j_k \leq p$ indicating which derivatives are being taken. More generally, if $\lambda : X \to U$ is a smooth function from $X \subset \mathbb{R}^p$ to $U \subset \mathbb{R}^q$, so $u = \lambda(x) = (\lambda_1(x), \ldots, \lambda_q(x))$, there are $q \cdot p_k$ numbers $u^\alpha_J = \partial_J \lambda_\alpha(x)$ needed to represent all the different k-th order derivatives of the components of λ at a point x. We let $U_k \equiv \mathbb{R}^{q \cdot p_k}$ be the Euclidean space of this dimension, endowed with coordinates u^α_J corresponding to $\alpha = 1, \ldots, q$, and all multi-indices $J = (j_1, \ldots, j_k)$ of order k. Furthermore set $U_{(n)} = U \times U_1 \times \cdots \times U_n$ to be the Cartesian product space, whose coordinates represent all the derivatives of functions $u = \lambda(x)$ of all orders from 0 to n.

Given a smooth function $u = \lambda(x)$, so $\lambda : X \to U$, there is an induced function $u_{(n)} = \mathbf{pr}^{(n)} \lambda(x) \in U_{(n)}$, called the n-th *prolongation* of λ, which is defined by the equations

$$u^\alpha_J = \partial_J \lambda_\alpha(x).$$

Thus $\mathbf{pr}^{(n)} \lambda$ is a function from X to the space $U_{(n)}$, and for each x in X, $\mathbf{pr}^{(n)} \lambda(x)$ is a vector whose entries represent the values of λ and all its derivatives up to order n at the point x. For instance, consider the case $p = 2$,$q = 1$ and $n = 2$, with X having coordinates in $\mathbb{R}^2$: $(x_1, x_2) = (x, y)$ and $U \subset \mathbb{R}$ a single coordinate u. Then we have

$$\mathbf{pr}^{(2)} \lambda(x, y) = (u; u_x, u_y; u_{xx}, u_{xy}, u_{yy}) = (\lambda; \frac{\partial \lambda}{\partial x}, \frac{\partial \lambda}{\partial y}; \frac{\partial^2 \lambda}{\partial x^2}, \frac{\partial^2 \lambda}{\partial x \partial y} \frac{\partial^2 \lambda}{\partial y^2}).$$

The total space $X \times U_{(n)}$, whose coordinates represent the independent variables, the dependent variables *and* the derivatives of the dependent variables up to order n is called

the n-th order *jet space* of the underlying space $X \times U$. The n-th prolongation $\mathtt{pr}^{(n)}\lambda(x)$ is also known as the n-jet of λ.

With these preliminary notions, we are ready to introduce the differential geometric framework necessary to the generalization of the notion of endogenous equivalence.

4.2 Vector fields in infinite-dimensional differential geometry

In order to be able to say that transformations by dynamic feedback (a priori of unknown dimension) are "diffeomorphisms", there is a necessity to develop a convenient framework for manipulating functions and other objects, which depend on a *finite* but not a priori fixed number of variables. Typically, we need to consider a new set of coordinates possibly made up with an unbound number of derivatives of u (i.e. the control). This implies to work in the infinite jet space of U, with an arbitrarily big, but a priori unknown, number of jets of u. This enables the formalization of the bijective relations between the trajectories of two equivalent systems (see section 4.4).

We proceed now to lay down the notion of a system in an infinite coordinate set according to [FLMR99].

Consider the dynamics

$$\dot{x} = g(x, u) \tag{IV.7}$$

where g is smooth on an open subset $X \times U \subset \mathbb{R}^n \times \mathbb{R}^m$, x is the state and u is the control input.

The integral curves of (IV.7) are described by smooth functions $t \mapsto (x(t), u(t))$, parameterized by initial conditions only; that is, initial conditions of the form: $\zeta_0 = (x_0, u_0, \dot{u}_0, \cdots, u_0^{(\mu)}, \cdots)$, where the derivatives of u of any order at time $t = 0$ are noted $u_0^{(\mu)}$, with $\mu \geq 0$.

The original coordinates (x, u) are thus completed by the infinite sequence of coordinates $\xi = (x, u, u^{(1)}, \ldots, u^{(\mu)}, \ldots) \in X \times U \times \mathbb{R}_m^\infty$, where $\mathbb{R}_m^\infty = \mathbb{R}^m \times \mathbb{R}^m \times \cdots$ denotes the product of a countably infinite number of copies of $\mathbb{R}^m$.

In this context, a smooth function is a function smoothly depending on a finite (but arbitrary) number of coordinates. Then if we prolong the original vector field g as

$$\mathbf{G}(\zeta) = (g(x, u), \dot{u}, \ddot{u}, \cdots)$$

(IV.7) reads

$$\dot{\zeta} = \mathbf{G}(\zeta).$$

Hence the following definition.

Definition 4.1 *A* classic system *is a pair* $(X \times U \times \mathbb{R}_m^\infty, \mathbf{G})$ *where* $\mathbf{G}$ *is a smooth vector field on* $X \times U \times \mathbb{R}_m^\infty$.

4.3 Notations

We introduce now notations thoroughly used in the sequel.

We note $\bar{u} := \left(u, u^{(1)}, \ldots, u^{(\nu)}\right)$ to design u and its derivatives up to order $\nu > 0$, a "sufficiently big" integer (i.e. arbitrarily big but a priori unknown). $\bar{U} := U \times U_1 \times \cdots \times U_\nu \subset \mathbb{R}^m \times \cdots \times \mathbb{R}^m_\nu$ designs $\nu+1$ copies of $\mathbb{R}^m$ with ν arbitrarily big. Moreover, we introduce a notation that will be useful in writing vector field prolongations: $\tilde{x} = (x, \bar{u}) = (x, u, u^{(1)}, \ldots, u^{(\nu)})$ for ν arbitrarily big. Suppose a given system $\dot{x} = g(x, \bar{v})$, we note $\tilde{g}$ the prolonged system $g_{(\nu)}$:

$$\begin{aligned} \dot{x} &= g(x, \bar{u}) \\ \dot{u}^{(\nu)} &= \varpi \end{aligned}$$

where the state is the vector $(x, u, \ldots, u^{(\nu)})$ and the input is $\varpi = u^{(\nu+1)}$. Hence we also write

$$\dot{\tilde{x}} = \tilde{g}(\tilde{x}, \varpi).$$

Typically, for a mapping λ defined on $X \times \bar{U} \subset \mathbb{R}^n \times (\mathbb{R}^m)^{\nu+1}$ we denote $\frac{\partial \lambda}{\partial x} f = \sum_{i=1}^n \frac{\partial \lambda}{\partial x_i} f_i$ and $\frac{\partial \lambda}{\partial u^{(k)}} u^{(k+1)} = \sum_{i=1}^m \frac{\partial \lambda}{\partial u_i^{(k)}} u_i^{(k+1)}$. We design by $\dot{\lambda}$ its prolongation (or total derivative) with respect to $\tilde{g}$ so that $\dot{\lambda} := D\lambda\tilde{g} = L_{\tilde{g}}\lambda$, where D stands for the total derivative:

$$D\lambda = \frac{\partial \lambda}{\partial x} + \sum_{k=0}^{\nu} \frac{\partial \lambda}{\partial u^{(k)}}.$$

Finally let us recall the notations introduced in the previous Chapter: $\vec{u}_{(.)}$ (resp. $\vec{u}_{(.)^\perp}$) is the unitary vector in the direction (.) (resp. the direction (.) + $\frac{\pi}{2}$); for any real-valued function, depending on the unique variable φ: $\mathcal{F}(\varphi)$, we note $\partial\mathcal{F}(\varphi)/\partial\varphi$ as $\mathcal{F}'(\varphi)$.

We can finally proceed to lay down the methodological grounds of our work.

4.4 Endogenous feedback equivalence

Suppose a system $\dot{x} = g(x, \bar{u})$, defined on $X \times \bar{U}$, with $X \times \bar{U} \subset \mathbb{R}^n \times (\mathbb{R}^m)^{k+1}$. Denote by N a point $(x, \bar{u})$ of $X \times \bar{U}$. Suppose another system $\dot{y} = h(y, \bar{v})$ defined on $Y \times \bar{V} \subset \mathbb{R}^p \times (\mathbb{R}^m)^{k+1}$ and denote by P a point $(y, \bar{v})$ of $Y \times \bar{V}$.

The following definition concerns the core of the notion of endogenous equivalence and of flatness.

Definition 4.2 (Endogenous feedback equivalence [Mar92]) *Two systems g and h are equivalent at (N_0, P_0) if there exists four analytical mappings $\psi : Y \times \bar{V} \to \mathbb{R}^n$, $\phi : X \times \bar{U} \to \mathbb{R}^p$, $\delta : Y \times \bar{V} \to \mathbb{R}^m$, $\alpha : X \times \bar{U} \to \mathbb{R}^m$ such that $N_0 = (\psi(P_0), \bar{\delta}(P_0))$,*

$P_0 = (\phi(N_0), \bar{\alpha}(N_0))$ and for all $(x, y, \bar{u}, \bar{v}) \in X \times Y \times \bar{U} \times \bar{V}$,

$$\begin{aligned} x &= \psi(\phi(x, \bar{u}), \bar{\alpha}(x, \bar{u})) && \text{(IV.8)} \\ u &= \delta(\phi(x, \bar{u}), \bar{\alpha}(x, \bar{u})) && \text{(IV.9)} \end{aligned}$$

$$\begin{aligned} y &= \phi(\psi(y, \bar{v}), \bar{\delta}(y, \bar{v})) && \text{(IV.10)} \\ v &= \alpha(\psi(y, \bar{v}), \bar{\delta}(y, \bar{v})) && \text{(IV.11)} \end{aligned}$$

$$\begin{aligned} h(\phi(x, \bar{u}), \bar{\alpha}(x, \bar{u})) &= D\phi \cdot \tilde{g}(x, \bar{u}), && \text{(IV.12)} \\ g(\psi(y, \bar{v}), \bar{\delta}(y, \bar{v})) &= D\psi \cdot \tilde{h}(y, \bar{v}). && \text{(IV.13)} \end{aligned}$$

Two systems g and h are equivalent if there exist points (N_0, P_0) such that g and h are equivalent in (N_0, P_0).

In other words, if $(x(t), u(t))$ is a trajectory of g then $(\phi(x(t), \bar{u}(t)), \bar{\alpha}(x(t), \bar{u}(t)))$ is a trajectory of h; moreover, there is a unique trajectory $(y(t), v(t))$ of h such that $(x(t), u(t)) = (\psi(y(t), \bar{v}(t)), \bar{\delta}(y(t), \bar{v}(t)))$ and vice-versa [Mar92].

On the other hand, the following theorem will help us to establish the actions to be undertaken in order to achieve our goal (linearization of the BiS-car).

Theorem 4.1 ([FLMR99]) *Consider two systems $(X \times U \times \mathbb{R}^\infty_m, \mathbf{G})$ and $(Y \times V \times \mathbb{R}^\infty_s, \mathbf{H})$, respectively describing the dynamics*

$$\begin{aligned} \dot{x} &= g(x, u), \quad (x, u) \in X \times U \subset \mathbb{R}^n \times \mathbb{R}^m && \text{(IV.14)} \\ \dot{y} &= h(y, v), \quad (y, v) \in Y \times V \subset \mathbb{R}^r \times \mathbb{R}^s && \text{(IV.15)} \end{aligned}$$

Assume that systems $(X \times U \times \mathbb{R}^\infty_m, \mathbf{G})$ and $(Y \times V \times \mathbb{R}^\infty_s, \mathbf{H})$ are differentially equivalent. Then, $s = m$ and there exists an endogenous dynamic feedback

$$\begin{aligned} u &= \sigma(x, z, w) \\ \dot{z} &= a(x, z, w) \quad z \in Z \subset \mathbb{R}^q && \text{(IV.16)} \end{aligned}$$

such that the closed-loop system (IV.14)–(IV.16) is diffeomorphic to (IV.15) prolonged by sufficiently many integrators.

Where "(IV.15) prolonged by sufficiently many integrators" means:

$$\begin{aligned} \dot{y} &= g(y, v) \\ \dot{v} &= v^{(1)} \\ \dot{v}^{(1)} &= v^{(2)} \\ &\vdots \\ \dot{v}^{(\nu)} &= w. \end{aligned}$$

or, using notations defined above

$$\dot{\tilde{y}} = \tilde{g}(\tilde{y}, w) \tag{IV.17}$$

According to theorem 4.1, in order to achieve our goal, we ought to find a dynamics of the form (IV.15), which should be proven to be equivalent to (IV.18). Subsequently, we shall compute the (endogenous) feedback (IV.16) allowing for the dynamics (IV.15) to be put into form (IV.17). This is what we will discuss in detail in what follows.

5 Equivalence issues for the BiS-car

5.1 Preliminaries

Recall that assuming that $\varphi \in]-\frac{\pi}{2}, \frac{\pi}{2}[$, then for a robot reference frame placed at point F (see Figure IV.3), the 2-input driftless control system of a BiS-car is:

$$\left\{ \begin{array}{lcl} \dot{x}_F &=& v_F \cos(\theta + \varphi) \\ \dot{y}_F &=& v_F \sin(\theta + \varphi) \\ \dot{\theta} &=& v_F \mathcal{F}(\varphi) \\ \dot{\varphi} &=& \omega_\varphi \end{array} \right. \tag{IV.18}$$

where $\mathcal{F}(\varphi) = \frac{\sin(\varphi - f(\varphi))}{L\cos(f(\varphi))}$. We write system dynamics (IV.18) in the more convenient form:

$$\dot{x} = g(x, u) \quad (x, u) \in X \times U \subset \mathbb{R}^4 \times \mathbb{R}^2, \tag{Σ_x}$$

where[4] $x = (x_F, y_F, \theta, \varphi)$ is the state of the robot (i.e. position, orientation and front steering angle) and $u = (v_F, \omega_\varphi)$ is the control input (i.e. the linear and the front-steering speeds).

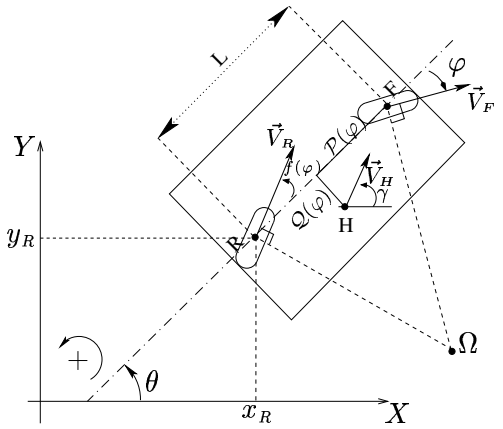


Figure IV.3: Bi-steerable car model showing the flat output (point H) with respect to the reference frame of the robot placed at point F.

[4]Lower case letters without index stand for vector variables.

The results established so far tell us that a flat output $y = (y_1, y_2)$ of the system are the coordinates of a point H in the world's Cartesian frame $\vec{P}_H := (x_H, y_H) = (y_1, y_2)$, computed as a function of the state as follows (see Figure IV.3):

$$\vec{P_H} = \vec{P_F} + \mathcal{P}(\varphi)\vec{u}_\theta + \mathcal{Q}(\varphi)\vec{u}_{\theta\perp} \tag{IV.19}$$

where $\mathcal{P}(\varphi)$ and $\mathcal{Q}(\varphi)$ are coordinate functions relative to the robot's reference frame:

$$\begin{array}{rcl} \mathcal{P}(\varphi) & = & \mathcal{M}(\varphi)\cos(\beta(\varphi)) - \mathcal{N}(\varphi)\sin(\beta(\varphi)) \\ \mathcal{Q}(\varphi) & = & \mathcal{M}(\varphi)\sin(\beta(\varphi)) + \mathcal{N}(\varphi)\cos(\beta(\varphi)) \end{array} \tag{IV.20}$$

with

$$\mathcal{M}(\varphi) = \frac{L\cos^2(f(\varphi))}{(\mathcal{A}^2(\varphi) + \mathcal{B}^2(\varphi))^{\frac{1}{2}}}$$

$$\mathcal{N}(\varphi) = -\int_0^\varphi \frac{L\cos^2(f(u))(\mathcal{B}'(u)\mathcal{A}(u) - \mathcal{A}'(u)\mathcal{B}(u))}{(\mathcal{A}^2(u) + \mathcal{B}^2(u))^{\frac{3}{2}}}\,\mathrm{d}u$$

and

$$\beta(\varphi) = \tan^{-1}\frac{\mathcal{B}(\varphi)}{\mathcal{A}(\varphi)}.$$

Moreover, $\mathcal{P}'$ and $\mathcal{Q}'$ are given by:

$$\begin{array}{rcl} \mathcal{P}'(\varphi) & = & (\mathcal{M}'(\varphi) - \beta'(\varphi)\mathcal{N}(\varphi))\cos(\beta(\varphi)) \\ \mathcal{Q}'(\varphi) & = & (\mathcal{M}'(\varphi) - \beta'(\varphi)\mathcal{N}(\varphi))\sin(\beta(\varphi)) \end{array} \tag{IV.21}$$

Conversely, x can be found from y and a finite number of its derivatives. More precisely, if one knows the flat output's velocity along the path it describes

$$\dot{\vec{P}}_H = v_H\vec{u}_\gamma = \sqrt{\dot{x}_H^2 + \dot{y}_H^2}\cdot\vec{u}_\gamma \tag{IV.22}$$

$$\gamma = \arctan\left(\dot{y}_2/\dot{y}_1\right) \tag{IV.23}$$

and the curvature of the path

$$\kappa = \frac{\det\left(y^{(1)}, y^{(2)}\right)}{\left(\left[y_1^{(1)}\right]^2 + \left[y_2^{(1)}\right]^2\right)^{3/2}}, \tag{IV.24}$$

then one can obtain the state of the robot as follows:

$$\left\{\begin{array}{l} \varphi = \mathcal{K}^{-1}(\kappa) \\ \theta = \gamma - \beta(\varphi) \\ \vec{P_F} = \vec{P_H} - \mathcal{P}(\varphi)\vec{u}_\theta - \mathcal{Q}(\varphi)\vec{u}_{\theta\perp} \end{array}\right. \tag{IV.25}$$

where

$$\kappa = \mathcal{K}(\varphi) = \frac{\beta'(\varphi)}{\mathcal{M}'(\varphi) - \beta'(\varphi)\mathcal{N}(\varphi)}. \tag{IV.26}$$

These expressions are formal relations between x and y together with its derivatives up to order 2.

On the other hand, since y is the flat output, its control system dynamics is simply given by

$$\dot{y} = h(y, v) = v \quad (y, v) \in Y \times U \subset \mathbb{R}^2 \times \mathbb{R}^2, \tag{Σ_h}$$

with $v = (v_1, v_2)$ being arbitrary functions of time.

In what follows, we shall find new expressions allowing to establish the equivalence relations from Definition 4.2. The point is that our current results (IV.19) and (IV.25) entail two mappings, fore-shadowing the transformations (IV.10) and (IV.8) respectively:

$$\begin{array}{rcl} (IV.19) & \Rightarrow & \phi(x) = y \\ (IV.25) & \Rightarrow & \psi(y, \bar{v}) = x \end{array} \tag{IV.27}$$

with $\nu = 1$; i.e. $\bar{v} = (v, v^{(1)})$.

Intuitively, in order to prove that (Σ_x) is equivalent to (Σ_h), we shall look into the flat output dynamics in order to formally establish those and the other transformations; i.e. (IV.9) and (IV.11). Moreover we shall prove also that the differential equivalence relations (IV.12) and (IV.13) hold. Furthermore we will find new relations that will be instrumental in the proof of equivalence:

$$\begin{array}{rcl} \sin(\varphi - \beta) & = & -\mathcal{M}\mathcal{F} \\ \cos(\varphi - \beta) & = & \frac{\mathcal{M}'\mathcal{F}}{\beta'}. \end{array} \tag{IV.28}$$

This is the purpose of the following section. We start by establishing results concerning **differential equivalence**—i.e. relations (IV.12) and (IV.13)—and then the **endogenous transformations**—i.e. mappings (IV.8)–(IV.11).

5.2 Differential equivalence

To alleviate notation, in the sequel we will drop the explicit dependence on φ in functions that depend only on this variable (i.e. all functions introduced in section 5.1).

We will first assume that x follows the flow of g and investigate the behaviour of $\dot{y}$. We will refer to the bi-steerable car when saying *system dynamics* and to the flat output when saying *flat dynamics*.

5.2.1 Flat output controls such that $D\phi \cdot g(x, u) = h(\phi(x), \alpha(x, u))$

The assumption here is that x follows the flow of g.

Differentiating equation (IV.19) with respect to time yields

$$\dot{\vec{P}}_H = D\phi(x) \cdot g(x, u) = v_F \vec{u}_{(\theta+\varphi)} + (\mathcal{P}\vec{u}_{\theta^\perp} - \mathcal{Q}\vec{u}_\theta)\dot{\theta} + (\mathcal{P}'\vec{u}_\theta + \mathcal{Q}'\vec{u}_{\theta^\perp})\omega_\varphi$$

where we have insisted in the fact that x follows the flow of (Σ_x). In this equation we can distinguish two terms. The first term is the *transport* velocity of point H:

$$(\dot{\vec{P}}_H)_{transp} = v_F \vec{u}_{(\theta+\varphi)} + (\mathcal{P}\vec{u}_{\theta^\perp} - \mathcal{Q}\vec{u}_\theta)\dot{\theta}.$$

The second term is the relative speed of point H with respect to the robot:

$$(\dot{\vec{P}}_H)_{rel} = (\mathcal{P}'\vec{u}_\theta + \mathcal{Q}'\vec{u}_{\theta^\perp})\omega_\varphi$$

Notice that this relative term is completely controlled by the steering command.

Since x follows the flow of g we can express the angular speed of the robot $\dot{\theta}$ with respect to the speed control v_F and φ (see system (IV.18)). Then projecting into directions $(\vec{u}_\theta, \vec{u}_{\theta^\perp})$ we have that

$$\dot{\vec{P}}_H = (v_F\varepsilon_1(\varphi) + \omega_\varphi\mathcal{P}')\vec{u}_\theta + (v_F\varepsilon_2(\varphi) + \omega_\varphi\mathcal{Q}')\vec{u}_{\theta^\perp},$$

with:

$$\begin{aligned}\varepsilon_1(\varphi) &= \cos(\varphi) - \mathcal{Q}\mathcal{F}(\varphi)\\ \varepsilon_2(\varphi) &= \sin(\varphi) + \mathcal{P}\mathcal{F}(\varphi)\\ \mathcal{F}(\varphi) &= \tfrac{\sin(\varphi - f(\varphi))}{L\cos(f(\varphi))}\end{aligned}$$

From (IV.23) and (IV.25), the velocity of the flat output is collinear to a vector of direction

$$\gamma = \theta + \beta(\varphi). \tag{IV.29}$$

Thus let us define a direct reference frame, attached to point F: $[\mathcal{R}]_\gamma \equiv (\vec{u}_\gamma, \vec{u}_{\gamma^\perp})$, so that expressing the flat output dynamics with respect to this frame yields

$$\begin{aligned}\dot{\vec{P}}_H &= [v_F(\cos(\varphi - \beta) - \mathcal{N}\mathcal{F}) + \omega_\varphi(\mathcal{M}' - \beta'\mathcal{N})]\vec{u}_\gamma\\ &\quad + v_F[\sin(\varphi - \beta) + \mathcal{M}\mathcal{F}]\vec{u}_{\gamma^\perp}\end{aligned}$$

where $(\varphi - \beta)$ is the angle subtended between the velocity vector of F ($\vec{V}_F$) and the velocity vector of H ($\vec{V}_H$) (see Figure IV.3), and where $\mathcal{M}, \mathcal{N}$ are defined in equation (IV.20).

Therefore, from equation (IV.22) and the expression above, we obtain

$$\begin{aligned}\|\dot{\vec{P}}_H\| &= v_H\\ &= v_F[\cos(\varphi - \beta) - \mathcal{N}\mathcal{F}] + \omega_\varphi[\mathcal{M}' - \beta'\mathcal{N}]\end{aligned} \tag{IV.30}$$

This gives the first of the new relations (IV.28):

$$\sin(\varphi - \beta) = -\mathcal{M}\mathcal{F}$$

Alternatively, we can explore equation (IV.29) in the following way.

$$\dot{\gamma} = \dot{\theta} + \beta'\dot{\varphi}.$$

But $\dot{\gamma} = \kappa\sqrt{\dot{y}_1^2 + \dot{y}_2^2} = \mathcal{K}v_H$. Since again, we are differentiating γ along g we have that $\dot{\varphi} = \omega_\varphi$ and $\dot{\theta} = v_F\mathcal{F}$. Therefore we obtain a second expression for v_H:

$$\|\dot{\vec{P}}_H\| = v_H = \frac{v_F\mathcal{F} + \omega_\varphi\beta'}{\mathcal{K}}. \tag{IV.31}$$

Equating (IV.30) and (IV.31) yields the second of the new relations (IV.28):

$$\cos(\varphi - \beta) = \frac{\mathcal{M}'\mathcal{F}}{\beta'}.$$

We have now at our disposal two possible expressions relating the open-loop controls of the robot to the control input of the flat output: (IV.31) and (IV.30). Since $v = (v_1, v_2)$ is an arbitrary function of time, it suffices to set:

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \alpha(x,u) = \frac{v_F\mathcal{F} + \omega_\varphi\beta'}{\mathcal{K}} \begin{pmatrix} \cos(\theta+\beta) \\ \sin(\theta+\beta) \end{pmatrix} \tag{IV.32}$$

so that when x follows the flow of (Σ_x), y follows the flow of (Σ_h). In other words, by setting $\alpha(x,u)$ as we did we have that

$$D\phi \cdot g(x,u) = h(\phi(x), \alpha(x,u))$$

This proves that (IV.12) holds. We are now interested in the inverse analysis as explained next.

5.2.2 Bi-steerable car controls such that $D\psi \cdot h(y,\bar{v}) = g(\psi(y,\bar{v}), \delta(y,\bar{v}))$

The assumption here is that y follows the flow of h.

We shall now differentiate the mapping (IV.25): $x = \psi(y,\bar{v})$ with respect to the flow of (Σ_h). The reader should keep in mind that $\varphi = \mathcal{K}^{-1}(\kappa)$, where κ is defined by (IV.24). Thus

$$\dot{\varphi} = \frac{d\kappa}{dt}\frac{1}{\mathcal{K}'(\mathcal{K}^{-1}(\kappa))} = \frac{d\kappa}{dt}\frac{1}{\mathcal{K}'(\varphi)} \simeq \lambda(v, v^{(1)}, v^{(2)}).$$

Then

$$\begin{aligned} \dot{\theta} &= \tfrac{d}{dt}\gamma - \dot{\varphi}\beta' \\ &= \kappa v_H - \dot{\varphi}\beta' = \mathcal{K}v_H - \dot{\varphi}\beta'. \end{aligned}$$

where $\gamma = \tan^{-1}\left(\frac{\dot{y}_2}{\dot{y}_1}\right)$. Finally

$$\dot{\vec{P}}_F = \dot{\vec{P}}_H - (\mathcal{P}\vec{u}_{\theta^\perp} - \mathcal{Q}\vec{u}_\theta)\dot{\theta} - (\mathcal{P}'\vec{u}_\theta + \mathcal{Q}'\vec{u}_{\theta^\perp})\omega_\varphi$$

Therefore, if we want φ to follow the bi-steerable system dynamics then we just need to set

$$\omega_\varphi = \frac{d\kappa}{dt}\frac{1}{\mathcal{K}'(\varphi)}.$$

The same argument applies for θ which gives

$$\dot{\theta} = v_F\mathcal{F} = \mathcal{K}v_H - \omega_\varphi\beta'.$$

Henceforth we just need to set

$$v_F = \frac{\mathcal{K}v_H - \omega_\varphi\beta'}{\mathcal{F}}$$

with v_H given by (IV.22).

Therefore, the open-loop controls of the robot in function of the trajectory of point H are given by the mapping

$$\begin{pmatrix} \omega_\varphi \\ v_F \end{pmatrix} = \delta(y, \bar{v}) = \begin{pmatrix} \frac{d\kappa}{dt}\frac{1}{\mathcal{K}'(\varphi)} \\ \frac{\mathcal{K}v_H - \omega_\varphi \beta'}{\mathcal{F}} \end{pmatrix} \tag{IV.33}$$

It remains therefore to show that (x_F, y_F) are also consistent with the flow of g.

Let us differentiate the mapping ψ (IV.25) with respect to time

$$\dot{\vec{P}}_F = \dot{\vec{P}}_H - (\mathcal{P}\vec{u}_{\theta^\perp} - \mathcal{Q}\vec{u}_\theta)(\mathcal{K}v_H - \dot{\varphi}\beta') - (\mathcal{P}'\vec{u}_\theta + \mathcal{Q}'\vec{u}_{\theta^\perp})\dot{\varphi}.$$

Projecting in $[\mathcal{R}]_\gamma \equiv (\vec{u}_\gamma, \vec{u}_{\gamma^\perp})$ gives

$$\dot{\vec{P}}_F = v_H\vec{u}_\gamma - (\mathcal{K}v_H\mathcal{Q} - \dot{\varphi}(\mathcal{P}' + \mathcal{Q}\beta'))[c_\beta\vec{u}_\gamma - s_\beta\vec{u}_{\gamma^\perp}] - (\mathcal{K}v_H\mathcal{P} + \dot{\varphi}(\mathcal{Q}' + \mathcal{P}\beta'))[c_\beta\vec{u}_{\gamma^\perp} + s_\beta\vec{u}_\gamma]$$

where the notation $c_\beta \equiv \cos(\beta)$, $s_\beta \equiv \sin(\beta)$ was employed.

After a few manipulations using expressions above, we find

$$\dot{\vec{P}}_F = v_H\vec{u}_\gamma - (\mathcal{K}v_H\mathcal{N} - \dot{\varphi}\mathcal{M}')\vec{u}_\gamma - (\mathcal{K}v_H\mathcal{M} - \dot{\varphi}\mathcal{M}\beta')\vec{u}_{\gamma^\perp}$$

Again, if we want x to follow the flow of (Σ_x) we just need to set

$$\begin{cases} \dot{\varphi} &= \frac{d\kappa}{dt}\frac{1}{\mathcal{K}'(\mathcal{K}^{-1}(\kappa))} = \omega_\varphi \\ \mathcal{K}v_H &= \dot{\gamma} = \dot{\theta} - \dot{\varphi}\beta' = v_F\mathcal{F} - \omega_\varphi\beta' \end{cases}$$

in order to obtain

$$\dot{\vec{P}}_F = v_H\vec{u}_\gamma - [v_F\mathcal{F}\mathcal{N} - \omega_\varphi(\mathcal{M}' - \beta'\mathcal{N})]\vec{u}_\gamma - v_F\mathcal{M}\mathcal{F}\vec{u}_{\gamma^\perp}.$$

If v_H is actually controlled as given by (IV.32) we obtain

$$\begin{aligned} \dot{x}_F\cos(\gamma) + \dot{y}_F\sin(\gamma) &= v_F\mathcal{F}\frac{\mathcal{M}'}{\beta'} \\ -\dot{x}_F\sin(\gamma) + \dot{y}_F\cos(\gamma) &= -v_F\mathcal{M}\mathcal{F} \end{aligned}$$

and finally, after substituting the new relations (IV.28), we get

$$\begin{aligned} \dot{x}_F &= v_F\cos(\theta + \varphi) \\ \dot{y}_F &= v_F\sin(\theta + \varphi) \end{aligned}$$

Therefore we can conclude that the mappings ψ (IV.25) and δ (IV.33) entail the differential equivalence:

$$D\psi \cdot h(y, \bar{v}) = g(\psi(y, \bar{v}), \delta(y, \bar{v}))$$

and thus the claim that (IV.13) holds is proved.

5.3 Endogenous transformations

This case is a bit more delicate. Indeed, the equivalence implies that we must be able to go from one point, say N_0, of the trajectory of **G** to a point, say P_0, of the trajectory of **H** and, from P_0, we shall be able to come back to exactly the same original point N_0.

It is clear that in order to recover the state and controls of the robot, we need the derivatives of the flat output up to order 3 (mapping δ requires $d\kappa/dt$). To prove the equivalence we should proceed by steps.

5.3.1 Endogenous transformation $x = \psi(\phi(x), \bar{\alpha}(x, \bar{u}))$

First we shall argue that, as x follows the flow of g, the prolongations of mapping (IV.32)—giving the further derivatives of y—*must* induce an overall consistency in the flat output state, thus inducing the desired results in the robot state. Indeed, the first prolongation of α reads

$$\dot{\alpha}(x, u) = \alpha^{(1)}(x, u, \dot{u}) = v^{(1)}.$$

Therefore, if x follows g then, in order to have $D\phi \cdot g(x, u) = h(\phi(x), \alpha(x, u))$ we *must* set $v^{(1)}$ such that

$$\kappa = \frac{\det\left(v, v^{(1)}\right)}{\|v\|^3} = \mathcal{K}(\varphi),$$

which entails

$$\psi|_{\varphi}(\phi(x), \alpha(x, u), \alpha^{(1)}(x, \dot{u})) = \mathcal{K}^{-1}(\kappa) = \mathcal{K}^{-1}(\mathcal{K}(\varphi)) = \varphi.$$

where $\psi|_{\varphi}$ stands for the restriction of ψ to φ.

On the other hand v is ruled by (IV.32). Hence

$$\tan^{-1}\left(\frac{v_2}{v_1}\right) = \theta + \beta(\varphi),$$

and therefore

$$\psi|_{\theta}(\phi(x), \bar{\alpha}(x, \bar{u})) = \tan^{-1}\left(\frac{v_2}{v_1}\right) - \beta(\mathcal{K}^{-1}(\kappa)) = \theta,$$

with $\bar{\alpha} = (\alpha, \dot{\alpha})$ and $\bar{u} = (u, \dot{u})$. It remains to recover (x_F, y_F). From (IV.19) and (IV.25), it should be now evident that

$$x = \psi(\phi(x), \bar{\alpha}(x, \bar{u}))$$

5.3.2 Endogenous transformation $y = \phi(\psi(y, \bar{v}), \delta(y, \bar{v}))$

This one is trivial from the composition of mappings $(IV.19) \circ (IV.25)$:

$$y = \phi(\psi(y, v, v^{(1)})).$$

5.3.3 Endogenous transformation $u = \delta(\bar{\alpha}(x, \bar{u}))$

If $\dot{x} = g(x, u)$, following the same arguments than in section 5.3.1, we must have

$$\kappa = \mathcal{K}(\varphi)$$

and the second prolongation of α

$$\ddot{\alpha}(x, u) = \alpha^{(2)}(x, u, \dot{u}, \ddot{u}) = v^{(2)},$$

must be such that

$$d\kappa/dt \simeq k\left(v, v^{(1)}, v^{(2)}\right) = \omega_\varphi \mathcal{K}'(\varphi).$$

And therefore

$$\delta|_{\omega_\varphi}(\kappa, d\kappa/dt) = \omega_\varphi,$$

so that the result for v_F follows from (IV.32) and (IV.33).

5.3.4 Endogenous transformation $v = \alpha(\psi(y, \bar{v}), \delta(y, \bar{v}))$

Here $\dot{y} = h(y, v)$ so that from (IV.33) we have the results for u. Moreover, from (IV.25) we get $\varphi = \mathcal{K}^{-1}(\kappa)$. Now,

$$\alpha(\psi(y, \bar{v}), \delta(y, \bar{v})) = \sqrt{v_1^2 + v_2^2} \begin{pmatrix} \cos(\tan^{-1}(v_2/v_1)) \\ \sin(\tan^{-1}(v_2/v_1)) \end{pmatrix}$$

and hence

$$\alpha(\psi(y, \bar{v}), \delta(y, \bar{v})) = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = v$$

These results allow us to claim that:

Claim 5.1 *According to the Definition 4.2, systems (Σ_x) and (Σ_h) are endogenous equivalent.*

This conclusion is more vivid when depicted as in Figure IV.4 showing that, when $\bar{v}$ is excited, if $(y(t), v(t))$ is a trajectory of g, there is a unique corresponding trajectory $(x(t), u(t))$ of f. The converse is also true.

We finally state a last, but not least, result regarding equivalence. In the following section we give the explicit chained form of the bi-steerable car.

5.4 Chained form of the bi-steerable car

As pointed out in [MR94], a 2-input driftless system that is linearizable by dynamic feedback can be converted, around every point of a dense open subset, into a chained system ([MS93]) using only static feedback.

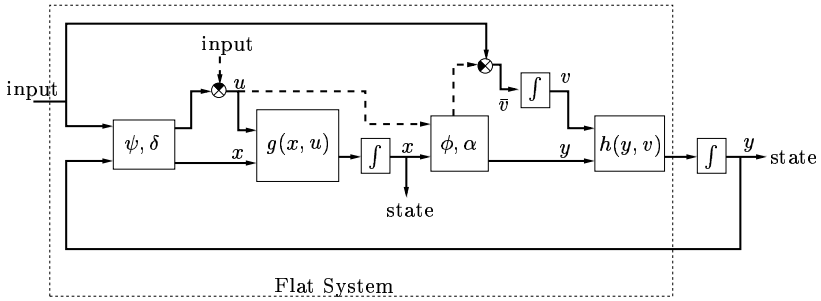


Figure IV.4: Endogenous equivalence between systems $g(x,u)$ (Σ_x) and $h(y,v)$ (Σ_h). The continuous lines show the "active" flow when the control input $\bar{v}$ is excited; i.e. the effect of u is discarded as shown by the discontinuous lines.

In view of this result, and as a corollary of the mappings just defined, we propose the formulation of the chained form of the bi-steerable car. Indeed, the endogenous transformations previously discussed allow us to derive the static feedback transformations giving the chained form of a bi-steerable car, by using the reference frame $[\mathcal{R}]_\gamma$.

The interest stems from the possibility to use results found in the growing literature about point-stabilization of nonholonomic systems put in this form (see e.g. [MS97].

The equivalent chained form of the bi-steerable system (IV.18)

$$\begin{cases} \dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1 \\ \dot{z}_4 &= z_3 u_2 \end{cases}$$

can be obtained from the following coordinates change:

$$\begin{aligned} z_1 &= x_H \\ z_2 &= \mathcal{K}/\cos(\gamma)^3 \\ z_3 &= \tan(\gamma) \\ z_4 &= y_H \end{aligned}$$

together with the static feedback:

$$\begin{aligned} u_1 &= v_H \cos(\gamma) \\ u_2 &= \mathcal{K}'/\cos(\gamma)^3\, \omega_\varphi + 3\mathcal{K}^2 \sin(\gamma)/\cos(\gamma)^5\, v_H \end{aligned}$$

where $\mathcal{K}$ is the curvature defined by equation (IV.26), $\gamma = \theta + \beta(\varphi)$ and v_H is given by either (IV.30) or (IV.31).

A proof follows.

First notice that, in flat coordinates, the bi-steerable control system reads with respect to $[\mathcal{R}]_\gamma$:

$$\begin{cases} \dot{x}_H &= v_H \cos(\gamma) \\ \dot{y}_H &= v_H \sin(\gamma) \\ \dot{\gamma} &= v_H \kappa \end{cases} \tag{IV.34}$$

where v_H is function of the control inputs of the robot (v_F, ω_φ) as stated by either (IV.30) or (IV.31). Hence the first equation of the chained form is simply obtained by setting

$$u_1 = v_H \cos(\gamma)$$

so that $\dot{z}_1 = \dot{x}_H = u_1$.

Then (IV.34) becomes

$$\begin{cases} \dot{x}_H &= u_1 \\ \dot{y}_H &= u_1 \tan(\gamma) \\ \dot{\gamma} &= u_1 \frac{\kappa}{\cos(\gamma)}. \end{cases}$$

By next choosing $z_4 = y_H$ we obtain

$$\dot{z}_4 = \dot{y}_H = u_1 \tan(\gamma).$$

Hence we get $\dot{z}_4 = z_3 u_1$ by choosing $z_3 = \tan(\gamma)$; this gives

$$\dot{z}_3 = \frac{1}{\cos(\gamma)^2}\dot{\gamma} = \frac{\kappa}{\cos(\gamma)^3}u_1.$$

Thus we get $\dot{z}_3 = z_2 u_1$ by choosing $z_2 = \frac{\mathcal{K}}{\cos(\gamma)^3}$; then we have

$$\dot{z}_2 = \mathcal{K}'/\cos(\gamma)^3\,\omega_\varphi + 3\mathcal{K}^2 \sin(\gamma)/\cos(\gamma)^5\, v_H,$$

and hence $\dot{z}_2 = u_2$.

□

We are now interested in stabilizing system (IV.18) around a reference trajectory. In doing so we explore the fact that, thanks to the flatness property, this system is diffeomorphic to a linear controllable one through endogenous dynamic feedback (Theorem 4.1 above). The following section presents the computation of such a feedback for the general bi-steerable car allowing us to deal with a linear system.

6 Feedback linearization of the BiS-car

A direct result of Theorem 4.1 above and the mappings we have defined is that, since (Σ_x) is flat, there exists an *endogenous*[5] dynamic feedback

$$\begin{aligned} u &= a(x, z, w) \\ \dot{z} &= b(x, z, w), \end{aligned} \tag{IV.35}$$

leading to the following closed-loop system:

$$y_1^{(3)} = w_1, \quad y_2^{(3)} = w_2$$

[5]meaning that the original—endogenous—variables of the system are transformed without creation of new *exogenous* variables.

with $(w_1, w_2) = w$ arbitrary functions of time.

This corresponds to (Σ_h) prolonged by 2 pure integrators. We shall alternatively write the linear system as:

$$\dot{\tilde{y}} = \tilde{h}(\tilde{y}, w)$$

where $\tilde{y} = (y, v^{(1)}, v^{(2)})$ is the 6-dimensional state vector.

At this stage, two possible ways for computing the feedback may be envisaged. The first would consist in following [FLMR99](Theorem 8). A second alternative is to use nonlinear control techniques to compute this feedback. We shall discuss both ways.

6.1 Feedback linearization by computing an endogenous feedback

According to [FLMR99](Theorem 8), the feedback is constructed as follows.

Notice first that the mapping ψ (defined in section 5.3.1), under new notation, reads:

$$x = \psi(\tilde{y}, w)$$

and that from section 5.2.2 we have

$$g(\psi(\tilde{y}, w), \delta(\tilde{y}, w)) = \tilde{h}(\tilde{y}, w)$$

There is therefore a splitting of $\tilde{y}$ such that the mapping

$$\tilde{y} \mapsto K(\tilde{y}) = (\psi(\tilde{y}), \tilde{y}_c) = (x, z)$$

is invertible; i.e. given x and some elements (z) of $\tilde{y}$, it is possible to recover $\tilde{y}$ completely. Hence the feedback is obtained as follows [FLMR99]:

$$\begin{aligned} u &= \alpha(K^{-1}(x, z), w) \\ \dot{z} &= \tilde{h}_c(K^{-1}(x, z), w) \end{aligned}$$

where $\tilde{h}_c$ is the projection of $\tilde{h}$ corresponding to $\tilde{y}_c$.

Clearly we have now all the transformations required to compute the feedback. Indeed, one can verify that if one knows the state x and also the current values of $z := \{y_1^{(1)}, y_1^{(2)}\} \subset \tilde{y}$,[6] then the mapping $\tilde{y} = K^{-1}(x, z)$ can be computed from expressions (IV.19), (IV.23) and (IV.24). With this and inputs (w_1, w_2) one can compute v_H and $\frac{d\kappa}{dt}$ to obtain the robot controls from (IV.33).

A second alternative is to use nonlinear control techniques to compute this feedback. We discuss now this approach.

[6] Notice that from x and using (IV.30) we can compute $y_1^{(1)}$. If we keep record of its evolution in time, then we can compute $y_1^{(2)}$ by differences.

6.2 Feedback linearization using nonlinear control techniques

As a matter of fact, the curvature (and hence φ) of the curve described by the trajectory of H is invariant with respect to Euclidean transformations in the plane. Due to this symmetry property, it seems more convenient to chose the intrinsic parameters of the curve, namely its arc length s and curvature $\kappa(s)$, to describe the motion of H.

Hence let us denote by $\mathcal{C}_H$ the actual curve of H admitting a natural arc length parameterization s. The curve $\mathcal{C}_H$ is defined by points P_H: $s \mapsto P_H(s)$ such that, by definition of a natural parameterization, $\forall P_H(s) \in \mathcal{C}_H$, $\|\mathrm{d}P_H/\mathrm{d}s\| = 1$ (see for instance [FLMR95a]).

The utilization of the arc length, to parameterize the curve described by H, leads naturally to the use of a Frenet reference frame $(\vec{\tau}, \vec{\nu})$ associated to the flat output and admitting direct orientation such that $\vec{\tau} \equiv \vec{u}_\gamma$ as shown in Figure IV.5 below.

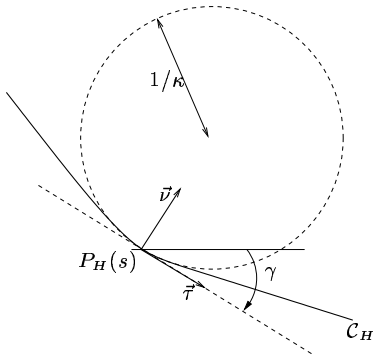


Figure IV.5: Frenet reference frame associated to the flat output when an arc-length parameterization is used on the trajectory of H.

In this setting we have

$$\mathrm{d}\vec{\tau}/\mathrm{d}s = \kappa\,\vec{\nu}, \quad \mathrm{d}\vec{\nu}/\mathrm{d}s = -\kappa\,\vec{\tau}.$$

Now consider a smooth mapping $t \mapsto s_r = \sigma_r(t)$, yielding a new time scale so that

$$v_H = \frac{\mathrm{d}s}{\mathrm{d}t} = \xi_1\,\dot{\sigma}_r,$$

where $\xi_1 = \frac{\mathrm{d}s}{\mathrm{d}\sigma_r}$ is near to 1 when the actual trajectory of H is close to the reference trajectory. Hence in the sequel derivations with respect to s_r, denoted: $(') \equiv \mathrm{d}/\mathrm{d}s_r$—not to be confused with $(')\equiv \partial/\partial\varphi$—yield the following rule: $\frac{\mathrm{d}}{\mathrm{d}s_r} = \frac{1}{\dot{\sigma}_r}\frac{\mathrm{d}}{\mathrm{d}t}$.

In this setting, the open-loop controls of the robot in function of the trajectory of point H read

$$\begin{aligned} \omega_\varphi &= \xi_1\,\dot{\sigma}_r \left(\frac{\mathrm{d}\kappa}{\mathrm{d}s}\right) \Big/ \mathcal{K}' \\ v_F &= \frac{\xi_1\,\dot{\sigma}_r - \omega_\varphi(\mathcal{M}' - \beta'\mathcal{N})}{\cos(\varphi-\beta) - \mathcal{N}\mathcal{F}} \end{aligned}$$

and dynamics (IV.22) now reads

$$\vec{P_H}' = \xi_1 \vec{\tau}.$$

If we differentiate twice and let $\xi_2 = \xi_1'$ we obtain

$$\vec{P_H}''' = (\xi_2' - \kappa^2 \xi_1^3)\vec{\tau} + (3\xi_2\xi_1\kappa + \xi_1^3 \frac{\mathrm{d}\kappa}{\mathrm{d}s})\vec{\nu} \tag{IV.36}$$

Notice that this system is invertible with respect to $(\xi_2', \mathrm{d}\kappa/\mathrm{d}s)$ as soon as $\xi_1 \neq 0$. Hence the dynamic compensator in time

$$\begin{aligned} \dot{\xi}_2 &= (\varpi_1 + \kappa^2 \xi_1^3)\, \dot{\sigma}_r \\ \dot{\xi}_1 &= \xi_2\, \dot{\sigma}_r \\ v_H &= \xi_1\, \dot{\sigma}_r \end{aligned}$$

and the static feedback

$$\frac{\mathrm{d}\kappa}{\mathrm{d}s} = \frac{\varpi_2 - 3\kappa\xi_2\xi_1}{\xi_1^3}$$

where ϖ_1, ϖ_2 are the new inputs in the σ_r scale, linearize the system as (IV.36) becomes:

$$\vec{P_H}''' = \varpi_1\vec{\tau} + \varpi_2\vec{\nu}.$$

6.3 Discussion

Let us look closer at this latter result. As a matter of fact, in the previous development we are applying the extended state space diffeomorphism

$$(x, \xi_1, \xi_2) \mapsto \tilde{y} = (\phi(x), \alpha(x,u), \dot{\alpha}(x,u,\dot{u}))$$

defining the 6-dimensional vector $\tilde{y} = (\tilde{y}_1, \tilde{y}_2)$ as the new state, with $\tilde{y}_i = (y_i^{(0)}, y_i^{(1)}, y_i^{(2)})$ for $i = 1, 2$.

From the inversion of (IV.36) we obtain the linearizing feedback which can be written in the following form (compare with (IV.4)):

$$\begin{pmatrix} \xi_2' \\ \frac{\mathrm{d}\kappa}{\mathrm{d}s} \end{pmatrix} = \boldsymbol{\alpha}(\tilde{y}) + \boldsymbol{\beta}(\tilde{y}) \begin{pmatrix} \varpi_1 \\ \varpi_2 \end{pmatrix},$$

Notice that the new controls in time are obtained easily from the following scalar products:

$$\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} y_1^{(3)} \\ y_2^{(3)} \end{pmatrix} = \begin{pmatrix} \vec{P_H}^{(3)} \cdot \vec{\tau} \\ \vec{P_H}^{(3)} \cdot \vec{\nu} \end{pmatrix}. \tag{IV.37}$$

Hence each output channel becomes a chain of two integrators and we have a controllable linear system of the form:

$$\dot{\tilde{y}} = \underbrace{\begin{bmatrix} \mathbf{A}_{c1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{c2} \end{bmatrix}}_{\mathbf{A}} \tilde{y} + \underbrace{\begin{bmatrix} \mathbf{B}_{c1} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{c2} \end{bmatrix}}_{\mathbf{B}} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \tag{IV.38}$$

where the canonical matrices $\mathbf{A}_{ci}$ and $\mathbf{B}_{ci}$, $i = 1, 2$ read as follows

$$\mathbf{A}_{ci} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \mathbf{B}_{ci} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

Figure IV.6 below illustrates the linearizing feedback and the equivalent linear controllable system.

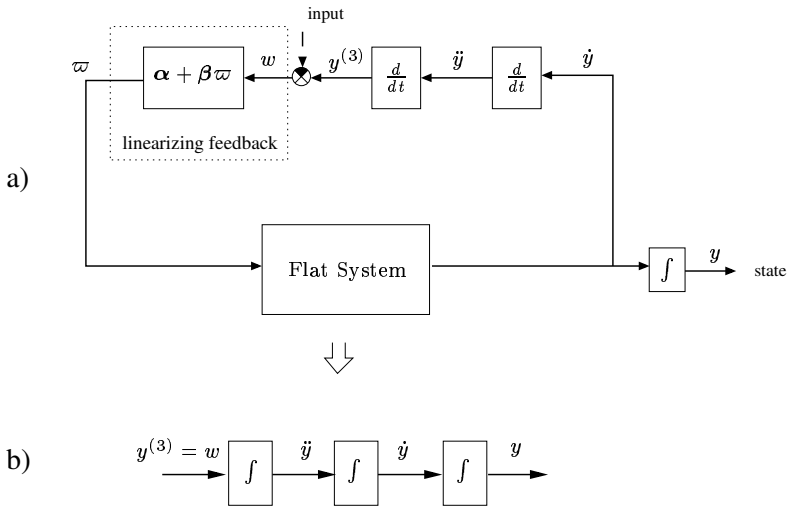


Figure IV.6: a) Linearizing feedback; b) Equivalent linear controllable system (IV.38) in Brunovsky form.

7 Trajectory tracking using a linear control law

We now explore the linear structure of system (IV.38) in order to stabilize trajectories of (IV.18). To achieve this goal we will use standard linear control techniques as discussed below.

7.1 The control law

From (IV.37) we see that the auxiliary input w is obtained from $y^{(3)}$ which is actually the controlling input.

For a real system, one can be sure that in the presence of a reference trajectory y^* there will be an error between this and the actual trajectory y of the system. What we are looking for therefore is to stabilize the error $e = (y - y^*)$. From system (IV.38), the error dynamics reads

$$\dot{\tilde{e}} = \mathbf{A}\tilde{e} + \mathbf{B}y^{(3)}. \tag{IV.39}$$

Therefore, from Linear Control theory we know that we can close the loop, at a second step, to stabilize the reference trajectory y^* with the control law:

$$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} y_1^{*(3)} \\ y_2^{*(3)} \end{pmatrix} - \begin{bmatrix} \mathbf{K}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_2 \end{bmatrix} \begin{pmatrix} \tilde{e}_1 \\ \tilde{e}_2 \end{pmatrix}$$

where

$$\mathbf{K}_i = (k_{i,0},\ k_{i,1},\ k_{i,2}) \quad i = 1, 2.$$

Thus the control law reads:

$$w_i = (y_i^*)^{(3)} - \sum_{j=0}^{2} k_{i,j} \left(y_i^{(j)} - (y_i^*)^{(j)} \right) \qquad i = 1, 2 \tag{IV.40}$$

and the coefficients $k_{i,j}$ must be chosen in order that the linear time invariant error dynamics

$$e_i^{(3)} = \sum_{j=0}^{2} k_{i,j} e_i^{(j)} \qquad i = 1, 2$$

with $e_i^{(j)} = \left(y_i^{(j)} - (y_i^*)^{(j)} \right)$, is stable.

In other words, we are searching to stabilize the trajectory of y, including its derivatives up to order 3, by means of a feed-forward reference term $(y_i^*)^{(3)}$ as (exciting) input modified by weighted correcting error terms $e_i^{(3)}$— see Figure IV.7.

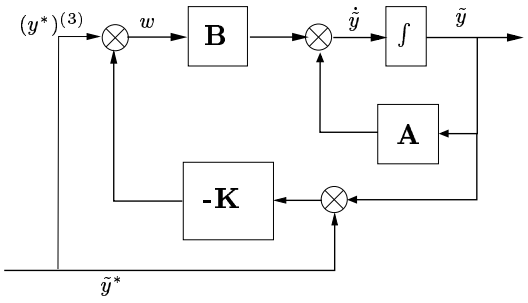


Figure IV.7: Feedback control law for the equivalent linearized system.

It is now clear what kind of reference trajectory is required as discussed next.

7.2 From feasible paths to feasible trajectories

We recall that the outcome of the local planner introduced in Chapter III is a feasible path. A feasible path corresponds to a geometric-parameterized curve (e.g. a curve parameterized by arc-length) whose course-speed may have been chosen arbitrarily. The problem that concerns us here is the transformation of such a path into a trajectory: i.e. finding a timing-law for the path.

The question on how to compute a parameter morphism in order to achieve a time-optimal speed profile along a given path has been already addressed in the literature. The problem is of algorithmic nature. Indeed, it is not possible to compute an optimal

profile without being able to foresee the curvature constraints imposed by the path and, at the same time, take into account dynamic constraints: linear and yaw acceleration limits for instance. This re-parameterization of the curve is therefore a key stage in obtaining a feasible trajectory (accounting for the dynamic constraints) from a feasible path (accounting for the kinematic constraints).

We have seen in sections 6.2 and 7.1 that the choice of an arc length parameterization conveniently accounts for geometric invariance and for natural parameterizations of the curve. In particular we saw that if $\mathcal{C}_H$ is the actual curve of H admitting a natural arc length parameterization s, then $\forall P_H(s) \in \mathcal{C}_H,\ \|\mathrm{d}P_H/\mathrm{d}s\| = 1$. In practice however this might not be the case so one could ask simply for a regular parameterization $t \to s = \sigma(t)$ such that $\|\mathrm{d}P_H/\mathrm{d}s\| \neq 0$ for all s.

In these respects, we have relied upon the work of [LSL99] in order to compute the aforementioned re-parameterization, taking into account speed and acceleration bounds. The input of the algorithm are the velocity bounds together with the linear and yaw acceleration limits of the robot and the feasible path in the flat space. The outcome is hence a time-sampled curve in the configuration space together with a set of nominal or reference controls.

The work of [LSL99] was conceived for tractor-n-trailer systems for which the flat output coincides with a physical point of the robot. In our case, instead of obtaining the reference trajectory and controls for the robot we had to yield the reference trajectory of the flat output— that is, the nominal values of (y_1, y_2) and their successive derivatives up to order 3. Indeed, we aim at stabilizing system (IV.38), for which a reference path is given, by using the control law (IV.40).

We now discuss the simulation results obtained.

7.3 Simulation results

We take up notations defined in Section 5.

In a first time a re-parameterization of the reference curve $\mathcal{C}_{Hr}$ of H is computed from [LSL99]. In this case a regular parameterization

$$t \mapsto \sigma_r(t)$$

is used.

In order to cope with this and to be able to apply all the machinery developed in Section 7.1 we will define an auxiliary control variable

$$\xi_{1_r} = \frac{\mathrm{d}P_{Hr}}{\mathrm{d}\sigma_r}$$

as the reference speed of H along $\mathcal{P}_{Hr}$ in the σ_r scale.

In this case

$$v_H = \frac{\mathrm{d}P_H}{\mathrm{d}t} = \xi_1 \dot{\sigma}_r,$$

where $\xi_1 = \frac{\mathrm{d}P_H}{\mathrm{d}\sigma_r}$ is near to ξ_{1_r} when the actual trajectory of H is close to the reference trajectory. An example of this is shown in Figure (IV.8) below.

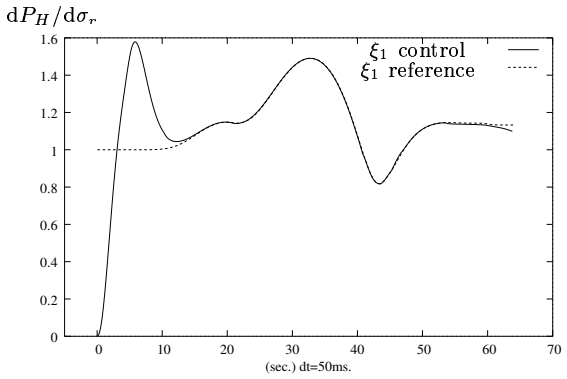


Figure IV.8: Simulation of typical control and reference curves for the flat output (point H) during a simple forward motion.

Notice that the control law (IV.40) will be applied for each channel of (IV.39):

$$\dot{\tilde{e}}_i = \mathbf{A}_{ci}\,\tilde{e}_i + \mathbf{B}_{ci}\,w_i \quad i = 1, 2$$

where we remind that

$$\mathbf{A}_{ci} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \mathbf{B}_{ci} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

For each output channel, the feedback control law (IV.40) is a matrix of the form

$$F_B = \mathbf{A}_{ci} - \mathbf{B}_{ci}\mathbf{K}_i = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -k_1 & -k_2 & -k_3 \end{pmatrix}$$

In order for this feedback to be stable, the matrix F_B must be *Hurwitz*. This implies that the eigen values of F_B must have negative real parts. The eigen values of F_B are obtained from its characteristic polynomial:

$$det(\lambda I_3 - F_B) = \lambda^3 + k_3\lambda^2 + k_2\lambda + k_1 = (\lambda + \frac{1}{d_1})(\lambda + \frac{1}{d_2})(\lambda + \frac{1}{d_3}).$$

The gains of the closed-loop system, $k_{i,j}$ $(i = 1, 2\ j = 1, 2)$, are therefore set by identification with the coefficients of this characteristic polynomial with eigen-values $\{\frac{1}{d_1}, \frac{1}{d_2}, \frac{1}{d_3}\}$. Thus we have:

$$\begin{aligned} k_1 &= \frac{1}{d_1 d_2 d_3} \\ k_2 &= \frac{1}{d_2 d_3} + \frac{1}{d_1 d_3} + \frac{1}{d_1 d_2} \\ k_3 &= \frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3} \end{aligned}$$

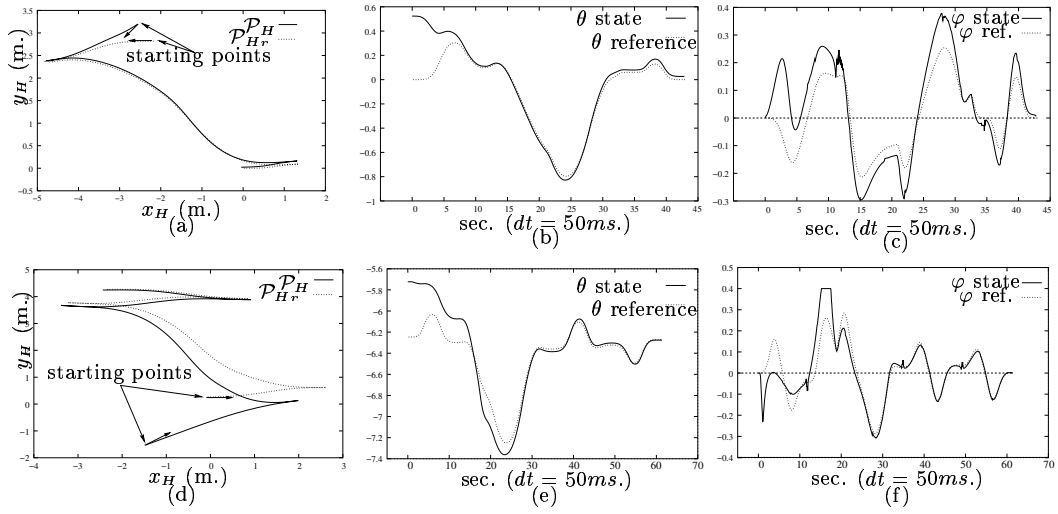


Figure IV.9: Simulation results for parking maneuvers using a BiS-car with $f(\varphi) = -0.6\varphi$ (series a–c) and $f(\varphi) = -\varphi$ (series d–f), showing respectively: the actual and reference curves for H; the state θ during the execution and the state φ (bounded to the physical limitations of the real robot). In both simulation series we set an error of $\pi/6$ in θ and of 5% in the parameter L. In series (a)–(c) the initial error in x_F and y_F is of $0.5m$. In series (d)–(f) this error is of $1.5m$.

For simulations we set $d_1 = 0.610$, $d_2 = d_1/1.5$ and $d_3 = d_2/1.5$ for both forward ($\dot{\sigma}_r > 0$) and backward ($\dot{\sigma}_r < 0$) motions. For all poles, the frequency is 20Hz.

Figures IV.9.a–IV.9.f show simulation results when introducing an initial error in the configuration space and a model error through the parameter L (the distance between the front and rear wheels). The arrows at starting points indicate the direction of the motion. Notice that even if the initial error in the first series is smaller than in the second one, the control is more sensitive in the case where $f(\varphi) = -0.6\varphi$. This is due to the fact that the relative velocity of the flat output (function of $\mathcal{P}$', $\mathcal{Q}$') is higher in the first than in the second case. Indeed, when $f(\varphi) = -\varphi$ the flat output is the same as that of a car with a non-steerable rear wheel placed at $\frac{1}{2}L$ of $[FR]$. Hence the stabilization is (relatively) easier.

8 Conclusions

Flatness characterizes nonlinear control systems which are equivalent to linear ones, by means of a restricted class of dynamic feedbacks called endogenous ([FLMR99]). In this respect the notion of *endogenous equivalence* provides a framework to the study of system classification and linearization via this class of feedbacks. In this setting, two systems F and G are equivalent if there is a bijective correspondence between the trajectories of F and those of G. Moreover, if F and G are two equivalent systems, one can find an endogenous dynamic feedback and a diffeomorphism which transforms F into G extended by pure integrators.

In this chapter we were interested in two issues: first, we established endogenous equivalence results regarding the bi-steerable system; second, we found means to synthesize a control law allowing us to stabilize the system trajectories around a nominal reference. To this end, we found the relations between the flat output dynamics and the robot controls. This led subsequently to complementary relations allowing to prove the differential equivalence between the bi-steerable car and the flat output dynamics extended by two integrators. As a consequence, we have tackled the trajectory tracking problem for the general bi-steerable car by finding its equivalent linear controllable system.

As a corollary of these results, it was possible to find the explicit chained form of the bi-steerable system, opening the possibility to use state-of-the art stabilization techniques for nonholonomic systems put into this form. In particular, time-varying point-stabilization techniques can be used.

The theoretical aspects were validated in simulation.

The next chapter addresses the integration of the results obtained, namely a complete motion planning and execution control scheme.

Chapter V

Experimental Issues

In this chapter we aim at an experimental validation of the integral approach of flatness to motion planning and feedback control of a real bi-steerable car. To this end, the building blocks introduced up this point, namely the motion planner and control law, need to be completed with a third element. This one must take into account the real character of the physical world and the robot: proprioceptive and exteroceptive perception of the system's real state are needed.

1 Introduction

At this stage, we are at the crossroads between the modeling of and the real system itself. What this implies is that the formal representations of a real bi-steerable system used up this point are ineluctably incomplete: our models do not correspond to reality. Indeed, the models we have introduced in previous chapters stem from simplifying assumptions of the reality, and hence *hide* many variables of the physical system.

The incompleteness issue sets us the following question: to what extent our theoretical results can be applied to the real system? Accordingly, we are interested in validating experimentally the results obtained in simulation. In the setting of probabilistic (Bayesian) inference and learning, recent work [BDL+99a], [BDL+99b], [LDBM00] has shown that incompleteness (the mismatch between models and reality) can be translated into uncertainty. Therefore the question about how to cope with the uncertainty when dealing with a real robot is inescapable.

In Robotics, there are many ways in which uncertainty may be taken into account: from low-level reactiveness, going through environment-perception modeling to task/mission reconfiguration. For instance, at the motion planning level, a classic way to deal with the uncertainty related to the environment is to grow the obstacles and the robot; this aims at reducing the risk of collision at the execution time. Notwithstanding, the incompleteness of our models makes that our (model-based) computations (e.g. robot commands) will still introduce errors during the execution. These errors translate into

deviations from the expected (nominal) motion, thus preventing the robot from fulfilling the task as planned.

In this respect, we have seen that the control law, introduced in the previous chapter, takes into account certain deviations from the nominal parameters of the robot and trajectory. However the control law needs to know the error between the nominal trajectory and the actual state of the robot. Hence the robot requires essential localization abilities. Accordingly, the robot must rely on its proprioceptive and exteroceptive sensors in order to reduce the uncertainty about its current state in the environment. In this chapter we shall discuss this problem and our experimental results.

In a first time, we introduce in section 2 our experimental bi-steerable platform. This will allow us to better understand the motivations of the global (exteroceptive) localization system. This one is the work of [PS02], who designed and implemented a robust absolute localization system introduced in section 3.1. Aiming at further robustifying the localization system, we designed an odometry (proprioceptive) localization module for the bi-steerable car. However, in the interests of consistency, we decided to take the discussion of the odometry module, together with its uncertainty propagation model, to Appendix D. The experimental assessment of our work is thence given in Section 4.

2 Experimental bi-steerable platform

The Cycab robot (see Figure V.1) is for the moment proposed as an academic platform in research laboratories such as INRIA in France and NTU in Singapore. It is however intended to be part of a new public transportation system based on small electric vehicles specifically designed for zones of limited access to regular automobiles.

Figure V.1: The INRIA Rhône-Alpes' Cycab robot.

The Cycab has diverse original characteristics: low profile platform allowing to adapt multiple sorts of cabins for different applications; distributed control architecture via a CAN bus, interfacing sensors and actuators in a modular fashion; ergonomic manual

control through a joystick; wire-less Ethernet link for communications with an eventual remote host; and on-board human-machine interface through a touch-screen and keyboard.

The particular characteristic which concerns us here is its bi-steerable capability. Making a little history, there exist two generations of Cycab robots:

- The first generation concerned a bi-steerable platform with a real mechanical link between the rear wheels and the front wheels. This mechanical link is such that $f(\varphi) = -0.7\varphi$. As a matter of fact the choice of the constant was not a design parameter—i.e. the objective was to have a bi-steerable platform no matter of the actual rear-to-front relation. Thus the rear-to-front factor was identified later by calibration procedures [BGMPG99].

- A second generation is a platform with a rear-to-front link realised by software—i.e. $\varphi_{rear} = f(\varphi_{front})$ with f arbitrarily set by software.

The Cycab is endowed with modest computational capacity. Notwithstanding we could validate experimentally the first steps towards motion autonomy; namely the motion planning and trajectory tracking abilities that have been discussed in previous chapters. To this end however, and in view of the incompleteness of the models we used, the uncertainty dimension must be taken into account. Therefore we shall now discuss a third essential building block in any architecture of an autonomous robot: the localization system.

3 Localization system

The localization system we aimed at for our experiments is based on a global (absolute) SLAM (Simultaneous Localization and Map-building) method [PS02] and a self-localization odometry module.

Owing to the dimensions of our robot and of the environment, the SLAM approach allowed us to have a reliable and robust localization system. Notwithstanding its remarkable performance, we wanted to rely as well on a self-localization module based on odometry. The interest for this is justified in cluttered environments. Indeed, the occlusion of landmarks in such environments prevents from global localization, in which case, it would be of great interest to trust as long as possible on the odometry itself.

Therefore we first introduce the absolute localization system. This will allow us to subsequently discuss the interest of the odometry module.

3.1 Localization system (I): exteroceptive localization

In order to carry on safe experiments we needed a robust exteroceptive localization system. We have used the work of [PS02] for this purpose.

The objective was to generate a map which could be used for localization during motion planning and execution. The specification of the system included reliability, accuracy and flexibility. What follows gives an overview on how these goals were reached.

Localization by detection of natural features in the environment is often subject to failure and not very accurate. In order to ensure reliability, it was decided to install artificial landmarks in the environment leading to an accurate detection. Hence the artificial landmarks are cylinders covered with reflector sheets, specially designed for our Sick laser range-finder. They are easy to detect, and can be localized with great accuracy (see below). Figure V.2 shows the Cycab, its laser sensor and the landmarks.

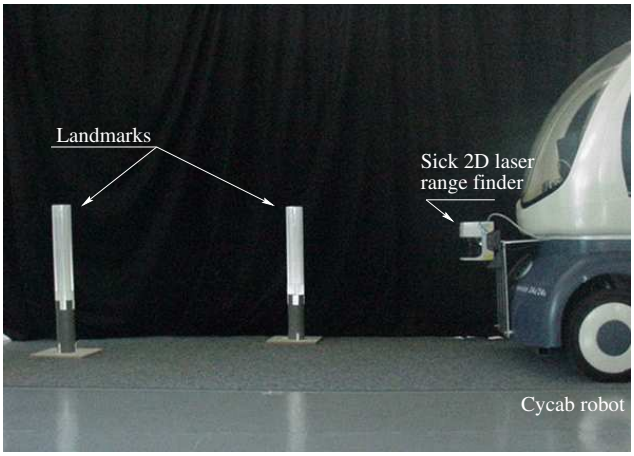


Figure V.2: Cycab robot and landmarks

The Cycab robot is the size of a golf cab, capable of attaining up to 30Km/h. Its "natural" environment is the car-park area of the INRIA Rhône-Alpes (about $10000m^2$). Hence it was decided to equip the environment with non permanent beacons in order to keep flexibility. Consequently, the localization system ought to be able to learn the current state of the car-park area using SLAM— Simultaneous Localization And Mapping —methods. However, the beacons—even though they are easy to detect—are not signed. Therefore, the problem of robust matching (between new observations and the current map) had to be handled simultaneously with the map building and localization problems. [PS02] proposes a robust algorithm called *SMLAM* (Simultaneous Matching Localization And Mapping), based on invariant features (e.g. distance between landmarks and angles between three of them) and graph theory tools.

Owing to the accuracy of the laser range-finder, to the good choice of the landmarks, and to the strength of the SLAM methods, the accuracy of the localization system was evaluated to the following value: about 10 centimeters in position and 2 degrees in orientation (see [PS02] for more details about the way these values were evaluated).

3.2 Localization system (II): Odometry for a BiS-car

Notwithstanding the remarkable performance of this absolute localization system, we sought to robustify it even more by means of an odometry module. In this respect,

odometry is necessary for two main reasons:

- Odometry is at the heart of uncertainty estimations when data fusion is performed with exteroceptive sensors;

- A reasonably accurate odometer may alleviate momentarily from computations (possibly heavy) for absolute localization; this is particularly true when for some reason absolute position readings are unavailable (e.g. occlusion of landmarks), in which case it would be of great interest to trust as long as possible on the odometry itself.

On the other hand, modeling odometry for a BiS-robot sets challenges that are not encountered in typical (non-steerable) mobile robots for which many work has been done. Let us mention for instance the work of [Wan88]. He proposed an odometry model for the rear axle of a car-like robot, together with its associated general uncertainty model under Gaussian hypothesis. Another pioneering work is due to [BF95]. They introduced a methodology yielding a benchmark test in order to calibrate systematic errors for differential drive robots. Systematic errors are observable on rectangular closed trajectories by solving geometric relationships. Finally, [CK97] resumed the odometry of [Wan88] by proposing a similar model and a closed-form for the propagation of uncertainty.

Odometry (dead-reckoning) is a basic localization system which is at the lowest levels of a control architecture. One can say that it is a low level sensor whose position predictions are model based.

Odometry aims at *predicting* the robot position and orientation (or heading) from proprioceptive measurements. Typically, encoders on two separated wheels are used to indicate both incremental change in heading (through their difference) as well as incremental distance traveled (through their average). This is known as differential heading odometry. Other variants combine information on the differential distance traveled in conjunction with a measure of the heading of the robot by means of a compass or through a gyroscope (measuring the speed of the change in heading).

Whatever the technique used, we are necessarily driven to make modeling choices involving ourselves in a design process. The outcome is an odometry model which is inevitably (though fortunately) a great simplification of the real system and hence is incomplete. This incomplete character may translate into some a priori knowledge about the "quality" of the model, but can be quantified by the notion of uncertainty [LDBM00]. Furthermore, the model contains parameters, related to the physical robot, that participate in the overall uncertainty about odometry predictions. Hence we are also driven to identify the nature of the uncertainty and the way we will treat it.

In order to keep the overall consistency of our dissertation, we discuss a possible odometry model for the bi-steerable car, together with its corresponding uncertainty model, in Appendix D.

We now address the last part of this chapter. What follows is the experimental phase concerning our work. In particular we shall discuss the integration of the computational

blocks required to validate the theoretical developments, regarding the flatness of the bi-steerable car.

4 Motion planning and trajectory execution settings

4.1 Obstacle map and user-planner interface

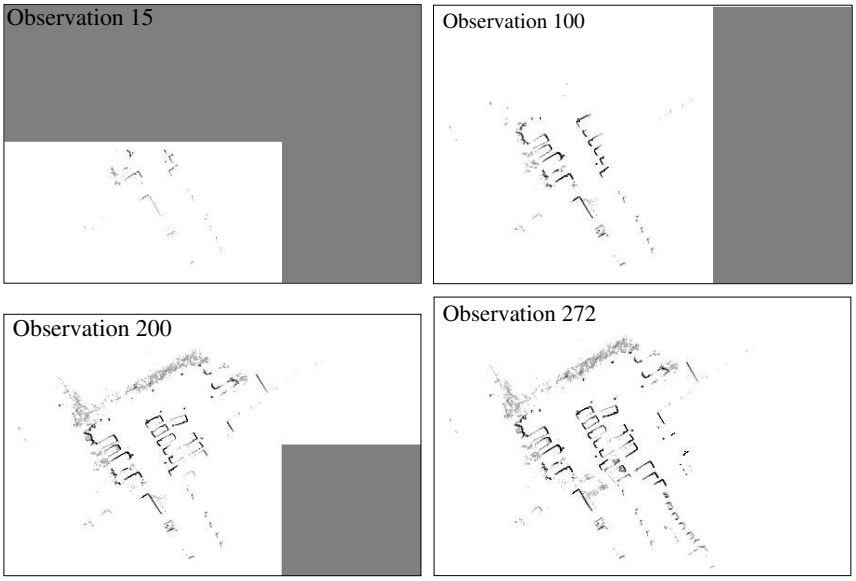


Figure V.3: Obstacle map evolution from [PS02]: Experimental images during the obstacle/landmarks map-building phase. The vehicle is driven within the car-park area as long as needed. Simultaneously, the laser range sensor is used to detect the landmarks to build-up the localization map.

The global localisation system introduced in Section 3.1 provides also a map of observed obstacles in order to plan safe paths. To achieve this goal, an occupancy grid was built on the environment. This structure gives informations correlated with the probability that a given place is occupied by an obstacle.

Both maps (landmarks and obstacle) are built online, in real-time, by the robot during the environment-exploration phase. Figure V.3 shows how the obstacle map evolves while we are exploring the environment. This map is made of small patches which are added according to the need of the application. In this way, the map can be extended in any direction, as long as memory is available. Once the map-building phase has finished, the obstacle map is converted into a pixmap and passed to the Motion Planning stage.

On the other hand, the User-Planner interface in the Cycab is achieved through a *touch-screen* superposed to a 640×480 pixels LCD display. The display may be used in text mode or graphics mode by direct access to video memory (by using SVGA-lib). Additionally, the keyboard can be used for the entrance of data.

The interface is used to display the current position of the robot within its environment and to capture the goal position entered by the user. These positions together with

the obstacle map is passed to the motion planner. The output path is then displayed allowing the user to validate the path or start a new search.

Figure V.4 shows the outcome of the motion planner introduced in Chapter 6 using an obstacle map generated as described above.

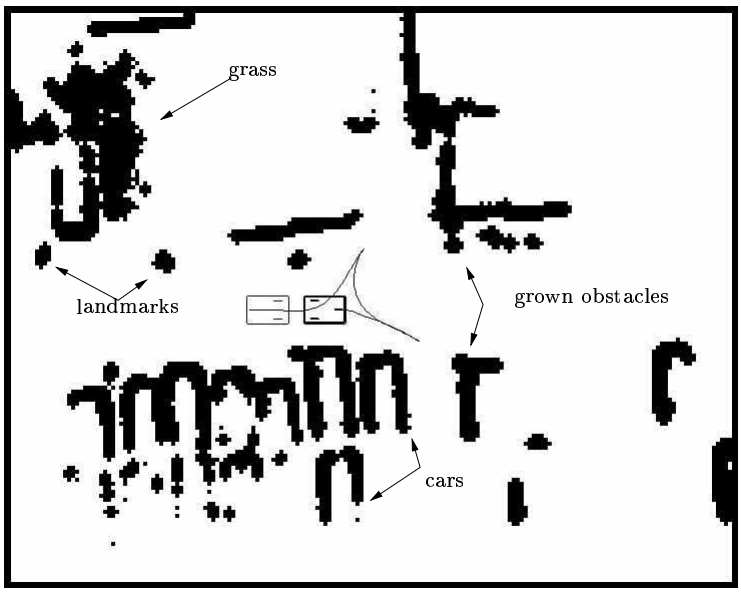


Figure V.4: Simulated path computed by the motion planner in a work-station using a real obstacle map generated by the previously described map-building stage. The obstacles are grown as well as the robot before computing the path.

Finally, the reference trajectory is generated using the regular parameterization of the path discussed in Chapter IV (Section 7.2) and the user is requested to accept to start the execution of the trajectory.

The control law, introduced in Chapter IV (section 7), is then used to track the nominal trajectory as discussed next.

4.2 Experimental results

We tested the integration of these modules in the Cycab robot. The aim was to validate the flatness approach to motion planning and control of the BiS-car and to get insight into the limitations of the whole motion scheme.

During the experiments the front-to-back steering function was set to $f(\varphi) = -\varphi$ and the speed of the robot was limited to $1.5ms^{-1}$. The gains of the closed-loop system, $k_{i,j}$, were set by identification with the coefficients of a characteristic polynomial with eigen-values $\{\frac{1}{d_1}, \frac{1}{d_2}, \frac{1}{d_3}\}$. We have set $d_1 = 1.40$ for forward motions and $d_1 = 1.41$ for backward motions, $d_2 = d_1/1.5, d_3 = d_2/1.5$. For all poles, the frequency is 20Hz.

The computation power on-board the Cycab is a *Pentium II*^{TM} 233MHz running a RedHat^{TM} Linux system. All programs were written in C/C++ language. The control

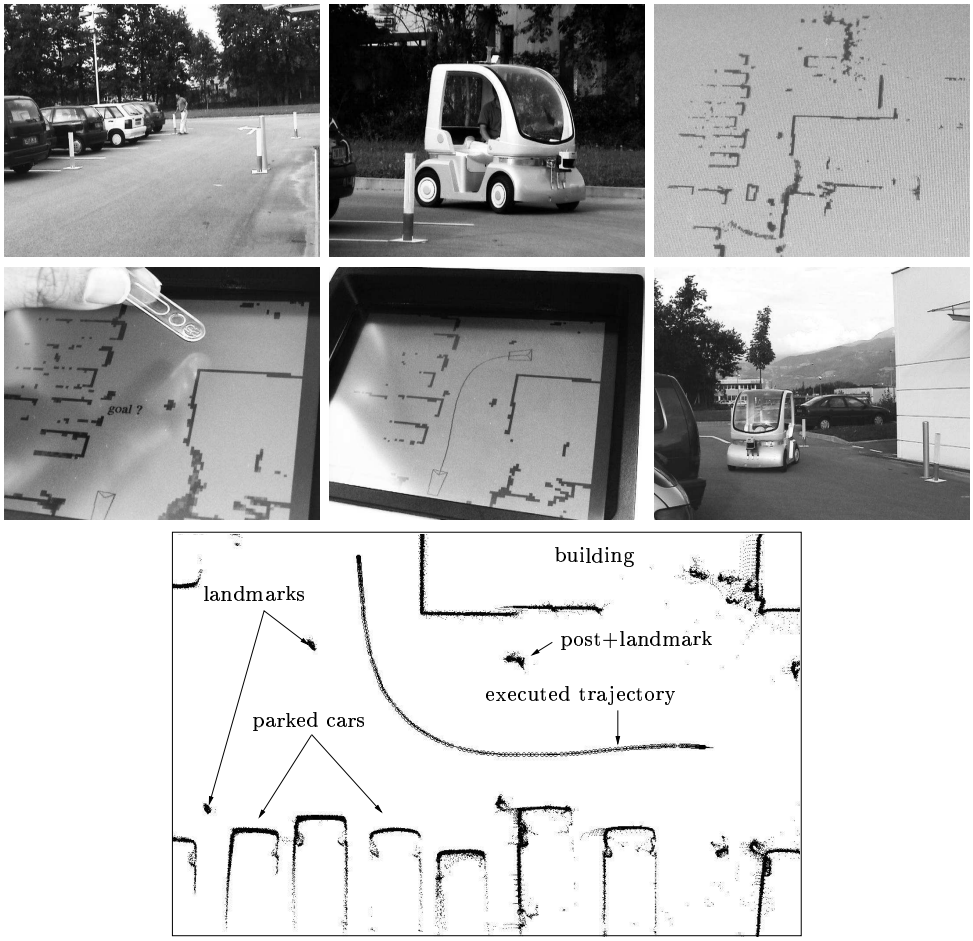


Figure V.5: An experimental setting showing from left top to bottom right: The arbitrary placing of the landmarks; the manual driving phase for landmark and obstacle map-building; the obstacle map generated together with the current position of the robot as seen on the LCD display; the capture of the goal position given by the user by means of the touch-screen; the reference path found by the motion planner; the execution of the maneuver using the linear control law (IV.40); and the executed trajectory recovered after the experiment.

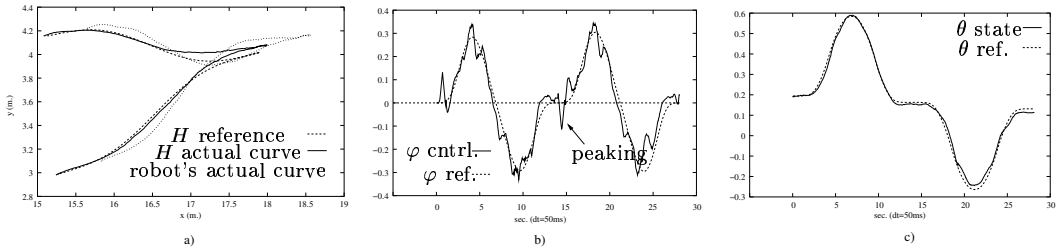


Figure V.6: Parking maneuver experimental setting: a) Reference curve for point H and actual curves for this point and for the robot (i.e. point F); b) Reference and actual curves of the steering angle control φ (the control system is sensible at cusp points inducing "peaks"); c) Reference and actual orientation of the robot θ.

rate of the robot was fixed at $50ms$. The throughput rate of the laser range-finder is limited to $140ms$[1]; therefore the control system relies momentarily in odometry readings.

Figure V.5 is a set of pictures showing a complete application integrating the map-building & localization system together with the motion planning and control schemes based on flatness of the BiS-car.

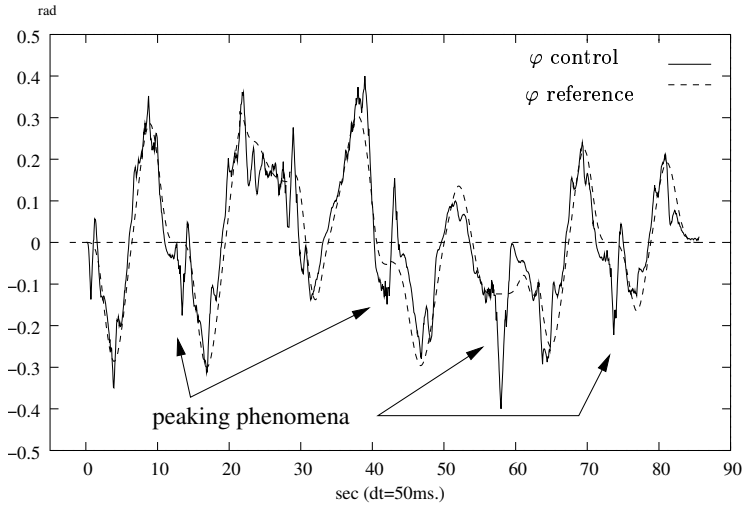


Figure V.7: Singularities at cusp points due to the fact that $v_H \approx 0$. The phenomenon is produced in the linearizing feedback by the undetermination of the derivative of the curvature ($d\kappa/ds$) at these points.

Figure V.6 shows the evolution of the flat output and the robot during a short parking maneuver. A couple of remarks may be done regarding these curves. The controller is able to track the reference curves for the 4-dimensional state (x, y, φ, θ). The performance of the controller is of good quality if we consider that the speed of the robot is low. The peaking phenomenon shown for the steering angle is intrinsic to the feedback linearization at cusp points. This is better appreciated in Figure V.7.

Finally, Figure V.8 shows another parking maneuver. Incidentally, during the experimentation there was an error in the localization system described in Section 3.1. The control law was able to stabilize the robot though, showing to be robust against sensor noise.

5 Conclusions

The computation capacity on-board our experimental platform allowed us to validate the theoretical approach regarding the flatness property of the bi-steerable car. The technical developments discussed allowed us to successfully integrate a basic application, representing the first step towards motion autonomy of this new kind of transportation system [HPS+03]. Taking further steps requires work regarding the safety and robustness of the

[1]This rate is fair enough for our needs, even though we could use a real-time driver.

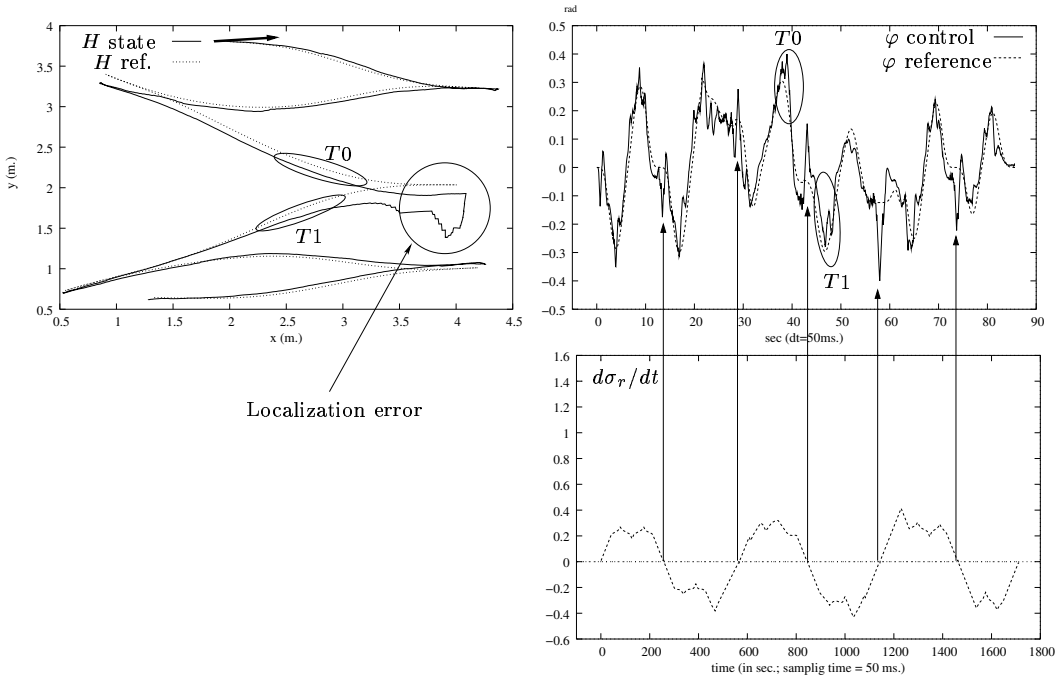


Figure V.8: Experimental curves showing the flat output during the execution of a parking maneuver (the fat arrow shows the starting point). Incidentally there was an error in the localization system "introducing a jump" in the robot's position. The system was able to remain stable though.

execution accounting further for uncertainty issues and for a dynamical environment. In particular, obstacle avoidance techniques and dynamic planning ([LSSL02]) ought to be integrated in the future as well as formal verification issues in the software architecture (e.g. using Orccad[SEKR91] development tool).

Chapter VI

Conclusions

1 Concluding Remarks

The work reported in this dissertation is at the crossroads of nonholonomic motion planning and feedback control. We have been interested in studying a recent double-steering nonholonomic robot allowing for exceptional maneuverability capacities. We call *bi-steerable car* a vehicle capable of steering its rear wheels in function of the front steering angle.

The study presented in this dissertation is original in the sense that characterizes the kinematics of the bi-steerable car which had not been studied in the literature. Indeed, the robotics community begins to pay attention to the motion planning problem for this kind of robot. To our knowledge, we have presented the first path planner for a general bi-steerable car and the first full state control based on feedback linearization. To this end we explored the properties of differentially flat systems and in particular of the flat (or linearizing) output.

In a first time, we were interested in solving the complete motion planning problem for the bi-steerable car using flatness. This faced us with the difficulty of finding the flat output. The problem was formalized in the setting of Pfaffian systems. The theory of Pfaffian systems gathers an important number of powerful theorems characterizing systems which can be put in a *normal form* with respect to a new set of coordinates. The flat output of the system belongs to this new set. Since the Pfaffian system associated to the bi-steerable car contains two equations and four variables, finding the flat-output consists in computing the change of coordinates associated to its *Engel normal form*. The Engel normal form is the dual of the chained form of a four-state, two-input control distribution. Notwithstanding the powerful results from the theory of Pfaffian systems, the major obstacle is still to find the appropriate transformations for the coordinate change.

On the other hand, we were interested in exploring the properties of the linearizing output for feedback control purposes. This led us to the theory of feedback linearization in Nonlinear Control. In this context, flatness characterizes nonlinear control systems

which are equivalent to linear controllable ones, by means of a restricted class of dynamic feedbacks called endogenous. In this respect the notion of *endogenous equivalence* provides a framework to the study of system classification and linearization via this class of feedbacks. Since Linear Control theory offers a solid and complete framework for solving closed-loop tasks, we were interested in exploring these properties to solve the trajectory tracking problem.

The details of our contribution are as follows.

- We first introduced the kinematics of the general bi-steerable robot and showed that it belongs to the class of nonlinear systems known as differentially flat. In particular we have given a necessary and sufficient condition on the front-to-rear steering function f in order to guarantee flatness at all points of $\mathcal{CS} \subset \mathbb{R}^4$. This entails typically that for a linear front-to-rear coupling function of the form $f(\varphi) = k\varphi$ we have that the system is flat for $k \neq 1$.
- We outline key steps for the explicit computation of the flat output for a codistribution of dimension 2, verifying the Engel's conditions. The outcome is a necessary condition on the coordinate transformations required to obtain this flat output. The condition translates into a pair of PDEs where the unknowns are the transformations sought.
- We applied this condition to the bi-steerable system. By exploring the PDEs and the invariance of the problem, with respect to Euclidean transformations in the plane, we found the flat output of the bi-steerable car.
- This allowed us to adapt a local planner originally conceived for tractor-trailer systems [SLL$^+$97]. The adaptation led to a symmetrical planner, verifying the *Topological Property* [SL98] and therefore taking into account the small-time controllability of the system. The steering method thus obtained was subsequently introduced in the frame of a general path-planning scheme [LJTM94, SLL$^+$97, LSL99].
- We completed the set of mappings allowing to prove the endogenous equivalence between the bi-steerable car and the flat output dynamics extended by two integrators. As a corollary of these results, we give the chained form of the bi-steerable car, opening the possibility to explore point-stabilization techniques. Finally, we have tackled the trajectory tracking problem for the general bi-steerable car by stabilizing its equivalent linear controllable system.

To our knowledge our work is the first to address motion planning and feedback linearization issues regarding bi-steerable robots. Simulation and experimental results have demonstrated the effectiveness and limitations of the approach. There are still open questions and thus we propose some guidelines for future work as follows.

2 Open questions and guidelines for future work

The advantages of the bi-steerable over the single front-steering system have been made patent in theoretical surveys in vehicle design and dynamic control. One may wonder

about the technical implementation problems or the economical factors that have prevented the car manufacturers from commercially launching standard models of the sort. However it is not unlikely that the double steering capability will gain even more applicability. In this respect, more research work needs to be done in order to get further insight into the technical issues.

In this respect, we can mention a few guidelines.

1. More experimentations need to be done in order to further evaluate the approach reported in this dissertation. In this respect, other front-to-rear functions should be explored (e.g. a rear angle varying sinusoidally with respect to the front steering).

2. What are the optimal paths for the bi-steerable car? In view of the advantages and applications of this kind of mechanical structure, this question is interesting though remains difficult to address.

3. Independent control of the rear steering angle yields a 5-state 3-input system. The system is seemingly flat ([MR95]).

   (a) What is the flat output? In this respect, more general and tractable methodological tools are required. The possibility to characterize symmetries in a more general framework should be a promising research direction.

   (b) How to use the additional degree of freedom? In this sense, exploring several combinations of rear-to-front control schemes may yield interesting results (from an application point of view: e.g. car parking). In this framework, a probabilistic approach to motion planning including a random choice on $f(\varphi)$ could be considered.

4. The solutions proposed in this dissertation for the planning and feedback control problems of the bi-steerable car show the utility of flatness. They shed light on some of the properties of the models used. There is however a big gap between these models and the real system. In this respect more robust control strategies remain to be explored as well as new AI techniques aiming at autonomous vehicles of this class. In this respect, a deeper analysis of the system dynamics stills the major obstacle in developing robust control methods.

Appendix A

Nonlinear control systems

Introduction

No one would deny that mathematics is the fundamental tool of any scientist or engineer looking for insight into a new problem at hand—this is an evidence for biologists of today looking for modeling phenomena of microscopic life. As far as the study of the bi-steerable system is concerned, nonlinear control systems and differential geometry are the formal frames within which we should lead the discussions developed throughout this dissertation.

A-1 Preliminary notions

Before giving the definitions of some concepts used throughout this document, let us give some preliminary notions that might be of utility, at least for the reader unfamiliar with the material presented here. Let us start this introductory Section with a statement that we found in [Lan86] about what one understands as *formal*: A list of rules or of axioms or of methods of proof which can be applied without attention to the "meaning" but which give results which do have the correct interpretation. Thus this Section is basically "informal" (most of the material presented here can be found in [Lan86, Chapters VI and VII]).

A-1.1 Tangent spaces, cotangent spaces and vector fields

Consider two functions $x = g(t)$ and $y = h(t)$ describing the motion of a point in the plane. If these functions have continuous derivatives, giving values x,y in the set $U \in R^2$ where the function $z = f(x, y)$ also has two continuous first partial derivatives, then $z = f(g(t), h(t))$ is a function of (suitable values of) t with the continuous derivative

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \frac{\partial z}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial z}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t}. \tag{A-1}$$

This chain rule has several different aspects.
First, think of $dx = (dx/dt)dt$ as an infinitesimal change in x, caused by the (equally) infinitesimal change dt in t. Then, multiplying (A-1) by dt and canceling gives

$$\mathrm{d}z = \frac{\partial z}{\partial x}\mathrm{d}x + \frac{\partial z}{\partial y}\mathrm{d}y. \tag{A-2}$$

This expression is called the total differential of z due to the infinitesimal changes dx and dy.
Starting from a point x_0, y_0 with finite changes $x - x_0$ and $y - y_0$, the formula (A-2) suggests a linear approximation $z - z_0$ to the change in z:

$$z - z_0 = \left(\frac{\partial z}{\partial x}\right)_0 (x - x_0) + \left(\frac{\partial z}{\partial y}\right)_0 (y - y_0). \tag{A-3}$$

In the chain rule (A-1), $\mathrm{d}z/\mathrm{d}t$ can be regarded as an "inner product" of two "vectors", as follows

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) \cdot \left(\frac{\mathrm{d}x}{\mathrm{d}t}, \frac{\mathrm{d}y}{\mathrm{d}t}\right). \tag{A-4}$$

The first factor on the right is called the *gradient* of $z = f(x, y)$; it is defined at each point of the plane, and is written

$$(\nabla f)_0 = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right)_{x=x_0, y=y_0}. \tag{A-5}$$

This vector "points" in the direction of the maximum rate of increase of the function f, and has that rate as length. The function f determines one such vector at each point of the plane. At each point, all such vectors for all f form a two-dimensional vector space, called the *cotangent* space attached to the plane at that point[1].

The second vector in the product (A-4) depends on the functions $x = g(t)$ and $y = h(t)$. They describe a continuous *path* passing through point x_0, y_0; such a path is called a *parametrized curve.* Such a curve is the trajectory of the moving point. At time t_0, where $x = x_0$, and $y = y_0$, the velocity of this moving point is the second factor of (A-4)

$$\left(\frac{\mathrm{d}x}{\mathrm{d}t}, \frac{\mathrm{d}y}{\mathrm{d}t}\right)_{t=t_0} = (g'(t_0), h'(t_0)).$$

It is called the *tangent vector* of the path at the point. All the tangent vectors to the trajectories through the point (x_0, y_0) form a two-dimensional space, called the *tangent space* T_0 to the plane at this point[2].

[1]Notice that a basis for such vector space, in this particular case, is $\{dx, dy\}$.
[2]Notice, that a basis for this vector space is $\{\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\}$.

The "product" (A-4) is a real-valued function of two vectors, one from each space. This function is linear in each vector when the other is held constant, so is said to be *bilinear*. As a matter of fact, the cotangent space is *dual* to the tangent space.

On the other hand, every smooth function f has a gradient ∇f as in (A-5). In particular the coordinates x and y are smooth functions, with gradients $\nabla x = (1, 0)$ an $\nabla y = (0, 1)$. Hence every gradient can be expressed at each point as a linear combination of these two gradients, in the form

$$\nabla f = \left(\frac{\partial f}{\partial x}\right) \nabla x + \left(\frac{\partial f}{\partial y}\right) \nabla y.$$

Except for notation, this is just the definition

$$df = \left(\frac{\partial f}{\partial x}\right) dx + \left(\frac{\partial f}{\partial y}\right) dy.$$

of the *total differential*. Thus the differential, born as an infinitesimal, may be defined to be the gradient ∇f — a vector in the cotangent space.
The tangent vector at t_0 to the path $g(t), h(t)$ also determines the usual tangent line to the path, with the parametric equations

$$x - x_0 = g'(t_0)(t - t_0), \quad y - y_0 = h'(t_0)(t - t_0), \tag{A-6}$$

where $x_0 = g(t_0)$ and $y_0 = h(t_0)$.

The chain rule (A-1) also has a 3-dimensional interpretation. The function $z = f(x, y)$ represents a height z above (or below) the point (x, y) in the plane, and so may be pictured by a smooth surface S at these heights above some portion of the plane. The tangent plane π to this surface at a point $p = (x_0, y_0, z_0 = f(x_0, y_0))$ is by definition the plane (if there is one) containing all the tangent lines at p to all the smooth curves on S passing through p. Such a smooth trajectory is given by $x - g(t)$, $y = h(t)$ and $z = f(g(t), h(t))$; its tangent line at p is given by the parametric equation (A-6) plus the corresponding equation for z:

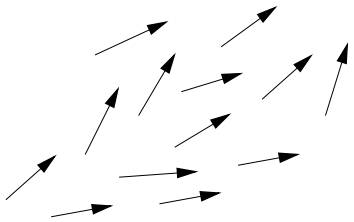
$$z - z_0 = \left(\frac{dz}{dt}\right)_0 (t - t_0). \tag{A-7}$$

But equations (A-6) and (A-7) together satisfy the linear equation (A-3) for the approximate change $z - z_0$ in z. This linear equation (A-3) represents a plane in 3-space; since it is satisfied by (A-6) and (A-7) for any curve $g(t)$, $h(t)$, $f(g(t), h(t))$ of S, it must be the tangent plane to the surface S.

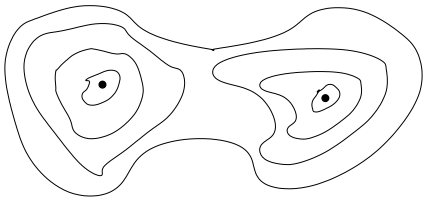
In this way, the chain rule combines ideas from geometry (tangent planes), from mechanics (velocity vectors), from calculus (linear approximation), and from algebra (dual spaces). It gives meaning to the "total differential".

Some of these ideas are more vivid in pictures. Thus gradients of a function $f(x, y)$ defined in the whole (x, y)-plane give a vector at each point in the plane —hence a *vector field* in the plane (Figure A-1.a). Alternatively the loci where $f(x, y) =$ constant give the

family of curves in the plane — the *contour lines* for f (Figure A-1.b). When f is smooth the gradient vectors, if non-zero, are orthogonal to the contour lines; for topography, they represent the direction of fastest ascent.



a.) A vector field



b.) Contour lines

Figure A-1: A vector field and contour lines such that $f(x, y)$ is constant.

A-1.2 Space, Motion and Differential geometry

Space and motion are closely tied, from physics to physical exercise. Perceptions of space and of motions in space have led mathematicians to describe a wide variety of formal geometrical structures. In Euclidean geometry, the phenomena of space is analyzed in terms of lines, triangles, angles, and congruence; in brief such a geometry is primarily linear. Other geometrical phenomena involve curved lines in the plane or twisted curves and curved surfaces in three-dimensional space. By approximating these curves by straight lines, the methods of calculus come into play, leading to the subject of differential geometry. One important concept in differential geometry is curvature. The resulting elementary methods of analyzing curvature lead inevitably to a study of the intrinsic geometry of surfaces.

A smooth *path* (that is, a *parameterized curve*) in the x, y plane is given by a pair of smooth functions

$$x = g(t), \quad y = h(t) \tag{A-8}$$

defined for some interval of the real parameter t. The tangent vector at any point $p = (g(t_0), h(t_0))$ of the path is the vector

$$(g'(t_0), h'(t_0)) \tag{A-9}$$

in the tangent space T_p of the plane at that point. If the tangent vector is never zero, the path is said to be *regular*. The parameter t can be changed, say to $u = k(t)$, provided the smooth function k has $k'(t) \neq 0$ throughout the interval in t. Such a change alters the length of the tangent vector (A-9) but not its direction—and gives the same set of points (x, y). This collection of points is the curve traced out by the path (A-8).

One first wants the length of such curve. To measure the length from t_1 to t_2 we have

$$\int_{t_1}^{t_2} [g'(t)^2 + h'(t)^2]^{1/2} dt = \int (dx^2 + dy^2)^{1/2}. \tag{A-10}$$

The length s from t_1 up to t is then the integral

$$s = \int_{t_1}^{t} (dx^2 + dy^2)^{1/2}; \tag{A-11}$$

then s determines t, so one may use this *arc length* s as a parameter in place of t. The differential of this arc length s is then given by the quadratic expression

$$ds^2 = dx^2 + dy^2; \tag{A-12}$$

it is the Pythagorean theorem in infinitesimal form.

For curvature, the circle provides a typical example. The smaller the radius, the greater the curvature. Here one defines the curvature of a circle to be $\kappa = 1/r$. For a general curve the natural approach is then to approximate the curve at each point p by a circle through that point. It is the limiting curvature of a circle, passing at points p and two neighbor points p' and p'', called the *osculating* circle, that defines the curvature of the given curve at point p. From this description as a limit, one can obtain an analytic formula giving the curvature κ at each point as a function of the parameter value t at that point:

$$\kappa = \frac{g'(t)h''(t) - h'(t)g''(t)}{(g'(t)^2 + h'(t)^2)^{3/2}}. \tag{A-13}$$

Here we assumed a regular parameterization, as defined above, so that the denominator is never zero.

The arc length s, measured from some starting point, provides an intrinsic parameter, and the curvature κ at each point is more naturally expressed as a function $\kappa = \kappa(s)$ of the arc length s up to that point.

Now in view of the "intrinsic" geometric definition of arc length and of curvature, these quantities are said to be Euclidean *invariants* of the curve, in the sense that they are unaltered by rotation and translation transformations.

In brief, the lines, circles and planes of elementary Euclidean geometry can be used to approximate the curves (and later the surfaces) in space, thereby making the ideas of calculus apply in a geometric context.

A-2 Differential geometry concepts

A-2.1 Manifolds

The following definitions are taken from several references including [Lat91b], [Olv93] and [Isi95].

Continuous map Let E and F be two topological spaces. A map $f : E \to F$ is continuous iff the inverse image of any open set (in the topology of F) is an open set (in the topology of E).

Homeomorphism Let $f : X \subset \mathbb{R}^n \to Y \subset \mathbb{R}^m$ be a map between two subsets of two topological spaces. It is a homeomorphism if f is bijective and continuous (or $\mathcal{C}^0$), and the inverse map $f^{-1} : Y \to X$ is also continuous (in the subspace topologies of X and Y). X and Y are said to be homeomorphic if such a map exists.

Diffeomorphism Let $f : X \subset \mathbb{R}^n \to Y \subset \mathbb{R}^m$ be a map between two subsets of two Euclidean spaces. It is a diffeomorphism (resp. a $\mathcal{C}^n$ diffeomorphism) if f is bijective and, f and the inverse map $f^{-1} : Y \to X$ are differentiable (resp. $\mathcal{C}^n$) (in the subspace topologies of X and Y). X and Y are said to be diffeomorphic if such a map exists.

Manifold A topological space M is a manifold if every point $x \in M$ has an open neighborhood homeomorphic to an open ball of $\mathbb{R}^n$, for some n independent of x. The number n is the dimension of the manifold.

Roughly speaking, and as far as the material presented here is concerned, a manifold may be seen (at least locally) as an open subset of the Euclidean space [Olv93].

A-2.2 Tangent space and tangent bundle

Tangent space Suppose C is a smooth curve on a manifold M, parameterized by $\sigma : I \to M$, where I is a subinterval of $\mathbb{R}$. In local coordinates $x = (x_1, \ldots, x_m)$, C is given by m smooth functions $\sigma(\varepsilon) = (\sigma_1(\varepsilon), \ldots, \sigma_m(\varepsilon))$ of the real variable ε. At each point $x = \sigma(\varepsilon)$ of C the curve has a *tangent vector*, namely the derivative

$$\dot{\sigma}(\varepsilon) = \frac{d(\sigma)}{d\varepsilon} = (\dot{\sigma}_1(\varepsilon), \ldots, \dot{\sigma}_m(\varepsilon)).$$

The collection of all tangent vectors to all possible curves passing through a given point x in M is called the *tangent space* to M at x, and is denoted by T_xM.

Tangent bundle The collection of all tangent spaces corresponding to all points x in M is called the *tangent bundle* of M, denoted by

$$TM = \bigcup_{x \in M} T_x M.$$

If M is of dimension m, then TM is of dimension $2m$.

A-2.3 Vector fields, distributions and integral curves

Vector field A *vector field* f is a function on M that assigns a tangent vector $f(x) \in T_x M$ to each point $x \in M$, with $f(x)$ varying smoothly from point to point. In local coordinates, a vector field is a column vector

$$f(x) = \begin{pmatrix} f^1(x) \\ \vdots \\ f^m(x) \end{pmatrix},$$

where each $f^i(x)$ is a smooth function of x.

If M is an m-dimensional manifold, then $T_x M$ is an m-dimensional vector space, with

$$\partial/\partial x_1, \ldots, \partial/\partial x_m$$

providing a basis for $T_x M$ in the given local coordinates $(x_1, \ldots, x_m)$. Hence an alternative notation for a vector field f is as follows:

$$f(x) = f^1(x)\frac{\partial}{\partial x_1} + f^2(x)\frac{\partial}{\partial x_2} + \cdots + f^m(x)\frac{\partial}{\partial x_m}.$$

Distribution Suppose now an independent set of vector fields $\{f_1, \ldots, f_m\}$. To this set is naturally associated a *distribution* on M. A distribution $\Delta(x)$, $x \in M$ is the subspace of $T_x M$ which is spanned by the tangent vectors assigned at x by the vector fields $\{f_1, \ldots, f_m\}$. Thus the dimension of the distribution at x is defined by the dimension of the vector space

$$\Delta(x) = span\{f_1(x), \ldots, f_m(x)\} \subset T_x M.$$

A distribution is therefore identified by a set of vector fields, say $\{f_1, \ldots, f_m\}$; when the explicit dependence on the point is dropped, the alternative notation is used

$$\Delta = span\{f_1, \ldots, f_m\}$$

to denote the assignment as a whole.

Integral curve An *integral curve* of a vector field f is a smooth parameterized curve $x = \sigma(\varepsilon)$ whose tangent vector at any point coincides with the value of f at the same point:

$$\dot{\sigma}(\varepsilon) = f(\sigma(\varepsilon))$$

for all ε. In local coordinates, $x = \sigma(\varepsilon) = (\sigma_1(\varepsilon), \ldots, \sigma_m(\varepsilon))$ must be a solution to the autonomous system of ordinary differential equations

$$\frac{dx_i}{d\varepsilon} = f^i(x), \quad i = 1, \ldots, m, \tag{A-14}$$

where $f^i(x)$ are the coefficients of f at x. For $f^i(x)$ smooth, the standard existence and uniqueness theorems for systems of ordinary differential equations guarantee that there is a unique solution to (A-14) for each set of initial data

$$\sigma(0) = x^0. \tag{A-15}$$

This in turn implies the existence of a unique *maximal* integral curve $\sigma : I \to M$ passing through a given point $x^0 = \sigma(0) \in M$, where "maximal" means that it is not contained in any longer integral curve.

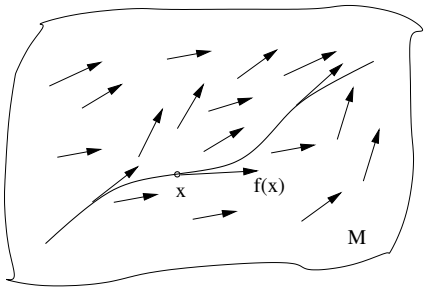


Figure A-2: Vector field and integral curve on a manifold.

A-2.4 Flows

If f is a vector field, we denote by $\Psi(\varepsilon, x)$ the parameterized maximal integral curve passing through x in M at $\varepsilon = 0$ and call Ψ the *flow* generated by f (see Figure A-2). Thus for each $x \in M$, and ε in some interval I_x containing 0, $\Psi(\varepsilon, x)$ will be a point on the integral curve passing through x in M. The flow of a vector field has the basic properties:

$$\Psi(\delta, \Psi(\varepsilon, x)) = \Psi(\delta + \varepsilon, x), \quad x \in M,$$

for all $\delta, \varepsilon \in \mathbb{R}$ such that both sides of the equation are defined,

$$\Psi(0, x) = x,$$

and

$$\frac{d}{d\varepsilon}\Psi(\varepsilon, x) = f(\Psi(\varepsilon, x))$$

Let us introduce the following notation:

$$e^{(\varepsilon f)}x \equiv \Psi(\varepsilon, x)$$

Hence the above properties can be restated as

$$e^{[(\delta+\varepsilon)f]}x = e^{(\delta f)}e^{(\varepsilon f)}x$$

whenever defined,

$$e^{(0f)}x = x,$$

and

$$\frac{d}{d\varepsilon}[e^{(\varepsilon f)}x] = f(e^{(\varepsilon f)}x)$$

for all $x \in M$.

The most important operation on vector fields is their Lie bracket or commutator. The next definition is a coordinate-free one.

A-2.5 Lie brackets

For vector fields f defined on M and for smooth functions $\lambda : M \to \mathbb{R}$

$$f(\lambda) = \frac{d}{d\varepsilon}\lambda(e^{(\varepsilon f)}x)$$

for all $x \in M$.

Definition If f and g are vector fields on M, then their *Lie bracket* $[f, g]$ is the unique vector field satisfying

$$[f, g](\lambda) = f(g(\lambda)) - g(f(\lambda))$$

for all smooth functions $\lambda : M \to \mathbb{R}$.

The following theorem gives a geometric interpretation of the Lie bracket of two vector fields (see Figure A-3).

Theorem A-2.1 ([Olv93]) *Let f and g be smooth vector fields on a manifold M. for each $x \in M$, the commutator*

$$\psi(\varepsilon, x) = e^{(-\sqrt{\varepsilon}g)}e^{(-\sqrt{\varepsilon}f)}e^{(\sqrt{\varepsilon}g)}e^{(\sqrt{\varepsilon}f)}x$$

defines a smooth curve for sufficiently small $\varepsilon \geq 0$. The Lie bracket $[f, g](x)$ is the tangent vector to this curve at the end-point $\psi(0, x) = x$:

$$[f, g](x) = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0+}\psi(\varepsilon, x).$$

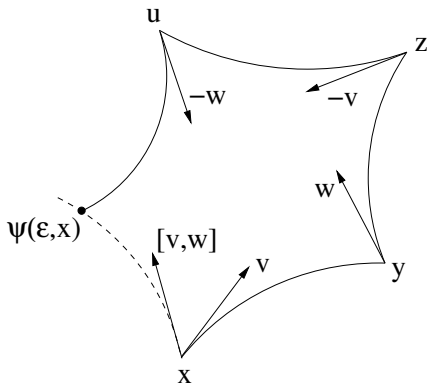


Figure A-3: Geometric interpretation of the Lie bracket of two vector fields (taken from [Olv93]).

Lie derivative in local coordinates Let λ be a real valued function and f be a vector field, both defined on a subset U of $\mathbb{R}^n$. The Lie derivative of λ along f is noted $L_f\lambda$ and has the usual definition

$$L_f\lambda(x) = \frac{\partial\lambda}{\partial x}f(x) = \sum_{i=1}^{n}\frac{\partial\lambda}{\partial x_i}f^i(x)$$

at each x of U.

When the operation is repeated, e.g. along another vector field g, we have

$$L_gL_f\lambda(x) = \frac{\partial\big(L_f\lambda\big)}{\partial x}g(x).$$

If the operation is repeated k times along the same vector field, we write $L_f^k\lambda$ and the following recursion formula is used

$$L_f^k\lambda(x) = \frac{\partial\big(L_f^{k-1}\lambda\big)}{\partial x}f(x)$$

with $L_f^0\lambda(x) = \lambda(x)$.

Lie bracket in local coordinates Let f and g be two vector fields, both defined on a subset U of $\mathbb{R}^n$. The Lie bracket is the new vector field noted $[f,g]$ and defined as

$$[f,g](x) = \frac{\partial g}{\partial x}f(x) - \frac{\partial f}{\partial x}g(x)$$

at each x of U, and where $\frac{\partial g}{\partial x}$ and $\frac{\partial f}{\partial x}$ denote the Jacobian matrices of the mappings g and f respectively.

Repeated bracketing is possible and is noted $[f,[f,\ldots,[f,g]]]$ which has the following alternative notation

$$ad_f^kg(x) = [f, ad_f^{k-1}g](x)$$

for any $k \geq 1$, setting $ad_f^0g(x) = g(x)$.

A-3 Elements of control systems classification

We now give some elements of classification of nonlinear control systems. These elements help in the formalisation of some system properties—e.g. controllability.

Lie product length Given vector fields $\{f_1, \ldots, f_m\}$, the length of a Lie product is recursively defined as:

$$l(f_i) = 1 \qquad i = 1, \ldots, m$$
$$l([A, B]) = l(A) + l(B),$$

where A and B are themselves Lie products.

Regular systems Let $\Delta = span\{f_1, \ldots, f_m\}$ be a distribution associated to a nonlinear control system of the form:

$$\Sigma : \quad \dot{x} = \sum_{i=1}^{m} u_i(t)\, f_i(x) \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m.$$

Define $E_1 = \Delta$ and

$$E_i = E_{i-1} + [E_1, E_{i-1}] \ \ (i = 2, 3, \ldots),$$

where

$$[E_1, E_{i-1}] = span\{[f, g] : \ f \in E_1,\ g \in E_{i-1}\}.$$

The set of all E_i's is called a *filtration* associated with Δ. At every point $x \in \mathbb{R}^n$, each $E_i(x)$ defines a linear subspace of the tangent space spanned by the driving (input) vector fields $\{f_1, \ldots, f_m\}$, plus the vector fields generated by taking up to $i - 1$ Lie brackets. A filtration is said to be regular at a point x^0 if

$$\mathrm{r}ank\ E_i(x) = \mathrm{r}ank\ E_i(x^0) \quad \forall x \in X,$$

where $X \subset \mathbb{R}^n$ is a neighborhood of x^0. A singular point is one for which the above equality does not hold, in which case the filtration is singular at that point. A system is said to be regular if the corresponding filtration is regular.

Degree of nonholonomy During the construction of a regular filtration, E_i either gains or maintains dimension, in which case the construction stops. If $\mathrm{r}ank\ E_{i+1} \neq \mathrm{r}ank\ E_i$, then $\mathrm{r}ank\ E_{i+1} > \mathrm{r}ank\ E_i$. Clearly, $\mathrm{r}ank\ E_i \leq n$. If a filtration is regular, then there exists an integer $p < n$ such that $E_{p-1} \neq E_p = E_{p+1} = \cdots$. The integer p is referred to as the *degree of nonholonomy* of the system. It corresponds to the minimal length of the Lie bracket required to span the tangent space[3].

[3] Formally it should be distinguished between the degree of nonholonomy at regular and singular points. The degree of nonholonomy of the system would corresponds therefore to the upper bound of all degrees of nonholonomy defined locally[LSL98].

Growth and relative growth vectors It is sometimes useful to record the dimension of each E_i. Thus, suppose that p is the degree of nonholonomy of the system at a point x and assume we have k nonholonomic equality constraints. The *growth vector* $r \in \mathbb{Z}^p$ at x is defined as the sequence $r := (r_1, \ldots, r_p)$, where $r_1 = (n-k) \leq r_2 \leq r_p = n$ and

$$r_i = \mathrm{r}ank\ E_i(x).$$

The *relative* growth vector $\sigma \in \mathbb{Z}^p$ is defined as: $\sigma_i = r_i - r_{i-1}$ and $r_0 := 0$. For a distribution with finite rank, it is possible to compute for every step an upper bound of this parameter. If at each step σ is equal to this upper bound then the system is said to have *maximum growth.*

Frobenius Theorem Recall that at a point x of the manifold M, the distribution $\Delta(x)$ is the subspace of T_xM which is spanned by the tangent vectors assigned at x by vector fields $f_1, \ldots, f_d$. The dimension of the distribution at a point is defined as the dimension of the subspace $\Delta(x)$. A vector field f belongs to a distribution Δ if $f(x) \in \Delta(x)$ for all $x \in M$.

A distribution Δ is called *involutive* if given any two vector fields f_1 and f_2 belonging to the distribution, their Lie bracket also belongs to the distribution, i.e.

$$f_1, f_2 \in \Delta \Rightarrow [f_1, f_2] \in \Delta$$

A distribution is called *integrable* if there exists a submanifold N of M such that the tangent space of N at x equals $\Delta(x)$. The submanifold N is called the *integral manifold* of the distribution Δ.

Theorem A-3.1 (Frobenius theorem for distributions) *A distribution $\Delta(x)$ is integrable if and only if it is involutive.*

Appendix B

Exterior differential systems

We will mention few concepts about exterior differential systems. For a complete presentation of the required material we refer the reader to [BCG$^+$91].

Preliminary notions

Tensors

Definition B-0.1 *Let $(V, \mathbb{R})$ denote a finite dimensional vector space over $\mathbb{R}$. The dual space associated with $(V, \mathbb{R})$ is defined as the space of all linear mappings $f : V \to \mathbb{R}$. The dual space of V is denoted as V^* and the elements of V^* are called* covectors. *V^* is a vector space over $\mathbb{R}$ with $dim\,(V^*) = dim\,(V)$ for the operations of addition and scalar multiplication defined by:*

$$\begin{aligned} (\alpha + \beta)(v) &= \alpha(v) + \beta(v) \\ (c\alpha)(v) &= c \cdot \alpha(v) \end{aligned}$$

Furthermore, if $\{v_1, \ldots, v_n\}$ is a set of basis vectors for V, then the set of linear functions $\phi^i : V \to \mathbb{R}$, $1 \leq i \leq n$, defined by:

$$\phi^i(v_j) = \begin{cases} 0 = if\, i \neq j \\ 1 = if\, i = j \end{cases}$$

form a basis of V^* called the dual basis.

A multi-linear function $T : V^k \to \mathbb{R}$ is called a covariant tensor of order k or simply a k-tensor. The set of all $k-$tensors on V is denoted $\mathcal{L}^k(V)$. Note that $\mathcal{L}^1(V) = V^*$, the dual space of V. Therefore, we can think of covariant tensors as generalized covectors.

Definition B-0.2 *Let f be an arbitrary k-tensor on V. If σ is a permutation of $\{1, \ldots, k\}$, we define f^σ by the equation*

$$f^\sigma(v_1, \ldots, v_k) = f(v_{\sigma(1)}, \ldots, v_{\sigma(k)})$$

Since f is linear in each of its variables, so is f^σ. The tensor f is said to be symmetric *if $f = f^e$ for each elementary permutation e, and it is said to be* alternating *if $f = -f^e$ for every elementary permutation e.*

We denote the set of all alternating k-tensors on V by $\Lambda^k(V^*)$. We have $\mathcal{L}^1(V) = \Lambda^1(V^*) = V^*$ and we define $\Lambda^0(V^*) = \mathbb{R}$.

Ideals

Definition B-0.3 *Given an algebra $(V, \odot)$, a subspace $W \subset V$ is called an* algebraic ideal *if $x \in W$, $y \in V$ implies that $x \odot y$ and $y \odot x \in W$.*

Definition B-0.4 *Let $(V, \odot)$ be an algebra, and $I \subset V$ an ideal. Two vectors $x, y \in V$ are said to be* equivalent mod I *if and only if $x - y \in I$. This equivalence is denoted*

$$x \equiv y \ mod \ I$$

k-forms The dual space of T_xM at each $x \in M$ is called the cotangent space to a manifold M at x and is denoted by T_x^*M. The collection of all cotangent spaces,

$$TM^* := \bigcup_{x \in M} T_x^*M$$

is called the cotangent bundle. Similarly, we can form the bundles

$$\mathcal{L}^k(M) := \bigcup_{x \in M} \mathcal{L}^k(T_xM)$$

$$\Lambda^k(M) := \bigcup_{x \in M} \Lambda^k(T_x^*M)$$

Tensor fields are constructed on a manifold M by assigning to each point x of the manifold a tensor. A $k-$tensor field on M is a section of $\mathcal{L}^k(V)$, i.e. a function ω assigning to every $x \in M$ a $k-$tensor $\omega(x) \in \mathcal{L}^k(T_xM)$. At some point $x \in M$, $\omega(x)$ is a function mapping k-tuples of tangent vectors of T_xM to $\mathbb{R}$, that is $\omega(x)(f^1, f^2, \ldots, f^k) \in \mathbb{R}$ is a multi-linear function of tangent vectors $f^1, f^2, \ldots, f^k \in M$. In particular, if ω is a section of $\Lambda^k(M)$ then ω is called a differential form of order k or a $k-$form on M. In this case, $\omega(x)$ is an alternating $k-$tensor $\in \Lambda^k(T_x^*M)$ at each point $x \in M$. The space of all $k-$forms will be denoted by $\Omega^k(M)$ and the space of all forms on M is simply

$$\Omega(M) := \Omega^0(M) \oplus \cdots \oplus \Omega^n(M)$$

Exterior derivative and wedge product

Exterior derivative Recall that a 0-form on a manifold M is a function $\lambda : M \to \mathbb{R}$. The differential $d\lambda$ of a 0-form λ is defined pointwise as the 1-form

$$d\lambda(x)(f_x) = f_x(\lambda).$$

It acts on a vector field f_x to give the directional derivative of λ in the direction of f_x at x.

Let $(x_1, \ldots, x_n)$ be local coordinate functions around a point $x \in M$. Then

$$\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}$$

is a basis for $T_x M$.

Consider the differentials of the coordinate functions

$$dx_i(x)(f_x) = f_x(x_i)$$

By evaluating these differentials at the basis tangent vectors of $T_x M$ we obtain

$$dx_i(x)(\frac{\partial \lambda}{\partial x_i}) = \delta_{ij} = \begin{cases} 0 = \text{if } i \neq j \\ 1 = \text{if } i = j \end{cases}$$

(i.e. for any point x and any tangent vector $v \in T_x M$, if $\{v^1, \ldots, v^n\}$ are coordinates for v at x, then $dx_k(x)(v) = v^k$). Therefore the $dx_i(x)$ are the dual basis of $T_x M$. Using this basis, we have now that for a 0-form

$$d\lambda = \sum_{i=1}^{n} \frac{\partial \lambda}{\partial x_i} dx_i$$

More generally, the *differentiation* is then defined as an operator transforming a k-form into a $(k+1)$-form.

$$d : \Omega^k(M) \to \Omega^{k+1}(M)$$

Wedge product On $\Omega(M)$, the space of the differential forms on M, one defines a *wedge product* $\wedge$ such that for a k-form ω and a p-form α, $\omega \wedge \alpha$ is a $(k+p)$-form. The wedge product is associative, distributive and skew-commutative: $\alpha \wedge \omega = (-1)^{kp} \omega \wedge \alpha$.

It allows us to simply define a basis for all $\Omega^k(M)$ using a coordinate system of M. Indeed, given such a chart $\{x_1, \ldots, x_n\}$ on M, any k-form ω can be expressed as :

$$\omega = \sum_{1 \leq i_1 < \ldots < i_k \leq n} \theta_{i_1, \ldots, i_k} dx_{i_1} \wedge \ldots \wedge dx_{i_k}$$

where θ_i's are 0-forms (i.e scalar functions on M) and dx_i's the 1-forms associated to the coordinate system.

Moreover, the differentiation of ω is defined as:

$$d\omega = \sum_{1 \leq i_1 < \ldots < i_k \leq n} d\theta_{i_1, \ldots, i_k} \wedge dx_{i_1} \wedge \ldots \wedge dx_{i_k}$$

where

$$d\theta_{i_1, \ldots, i_k} = \sum_{j=1}^{n} \frac{\partial \theta_{i_1, \ldots, i_k}}{\partial x_j}, \quad \text{for } 1 \leq i_1 < \ldots < i_k \leq n$$

We say that a k-form is of class C^∞ (or smooth) if and only if the functions $\theta_{i_1,\ldots,i_k}$ are of class C^∞ (or smooth).

The following theorem resumes the above notions and some important properties of the differential operator.

Theorem B-0.2 *Let M be a manifold and let $x \in M$. Then the exterior derivative is the unique linear operator*

$$d : \Omega^k(M) \to \Omega^{k+1}(M)$$

for $k \geq 0$, that satisfies,

1. *If α is a 0-form, then $d\alpha$ is the 1-form*

$$d\alpha(x)(f_x) = f_x(\alpha)$$

2. *If $\omega_1 \in \Omega^k(M)$, $\omega_2 \in \Omega^\ell(M)$ then*

$$d(\omega_1 \wedge \omega_2) = d\omega_1 \wedge \omega_2 + (-1)^k \omega_1 \wedge d\omega_2$$

3. *For every form ω, $d(d\omega) = 0$.*

Distributions and codistributions

Distributions Recall that a distribution $\Delta(x)$, $x \in M$ is the subspace of T_xM which is spanned by the tangent vectors assigned at x by the vector fields $\{f_1, \ldots, f_d\}$ and is denoted by $\Delta(x) = span\{f_1(x), \ldots, f_d(x)\}$. Alternatively, when the explicit dependence on the point is dropped, we write

$$\Delta = span\{f_1, \ldots, f_d\}.$$

A distribution is called integrable if there exists a submanifold N of M such that the tangent space of N at x equals $\Delta(x)$. The submanifold N is called the integral manifold of the distribution.

Codistributions Similarly, one can assign to each point $x \in M$ a set of 1-forms. The span of these 1-forms at each point will be a subspace of the cotangent space T_x^*M. This assignment is called a codistribution and is denoted by $\Theta(x) = span\{\omega_1(x), \ldots, \omega_d(x)\}$ or alternatively:

$$\Theta = span\{\omega_1, \ldots, \omega_d\}.$$

There is a notion of duality between codistributions and distributions. Given a distribution Δ, for each x in a neighborhood X, consider all the 1-forms which pointwise annihilate all vectors in $\Delta(x)$:

$$\Delta^\perp(x) = span\{\omega(x) \in T_x^*M : \omega(x)(f) = 0, \ \forall f \in \Delta(x)\}.$$

$\Delta^\perp(x)$ is a subspace of T_x^*M and is therefore a codistribution. We call $\Delta^\perp$ the annihilator or dual of Δ.

Conversely, given a codistribution Θ, we construct the dual distribution pointwise as

$$\Theta^\perp(x) = span\{v \in T_xM : \omega(x)(v) = 0,\ \forall \omega(x) \in \Omega(x)\}.$$

If N is an integral manifold of a distribution Δ and v is a vector in the distribution Δ at a point x (and consequently in T_xM), then for any $\alpha \in \Delta^\perp$, $\alpha(x)(v) = 0$. This must also be true for any integral curve of the distribution. Therefore given a codistribution $\Theta = span\{\omega_1, \ldots, \omega_s\}$, an integral curve of the codistribution is a curve $c(t)$ whose tangent $c'(t)$ at each point satisfies, for $i = 1, \ldots, s$:

$$\omega_i(c(t))(c'(t)) = 0.$$

Exterior differential systems

Exterior algebra and ideals The space of all forms on a manifold M,

$$\Omega(M) := \Omega^0(M) \oplus \cdots \oplus \Omega^n(M)$$

together with the wedge product is called the exterior algebra on M.

Algebraic ideal An algebraic ideal I is a subspace such that if

$$\alpha \in I,\ \ then\ \alpha \wedge \beta \in I\ for\ any\ \beta \in \Omega(M).$$

An ideal $I \subset \Omega(M)$ is said to be *closed with respect to exterior differentiation iff*

$$\alpha \in I \Rightarrow d\alpha \in I$$

An algebraic ideal which is closed with respect to exterior differentiation is called a *differential ideal.*

A finite collection of forms, $\Sigma := \{\alpha_1, \ldots, \alpha_K\}$ generates an algebraic ideal

$$I_\Sigma := \{\omega \in \Omega(M) \,|\, \omega = \sum_{i=1}^{K} \theta_i \wedge \alpha_i \ \ for\ some\ \theta_i \in \Omega(M)\}.$$

We can also talk about the differential ideal $\mathcal{I}_\Sigma$ generated by Σ.

Exterior differential system

Definition B-0.5 *An* exterior differential system *is a finite collection of equations*

$$\alpha_1 = 0, \ldots, \alpha_r = 0$$

where each $\alpha_i \in \Omega^k(M)$ *is a smooth k-form. A solution to an exterior differential system is any submanifold* N *of* M *which satisfies*

$$\alpha_i(x)|_{T_x N} \equiv 0$$

for all $x \in N$ *and all* $i \in \{1, \ldots, r\}$.

The following theorem allows us to either work with the generators of an ideal or with the ideal itself.

Theorem B-0.3 *Given an exterior differential system*

$$\alpha_1 = 0, \ldots, \alpha_K = 0$$

and the corresponding differential ideal $\mathcal{I}_\Sigma$ *generated by the collection of forms*

$$\Sigma := \{\alpha_1, \ldots, \alpha_K\}.$$

An integral submanifold N *of* M *solves the system of exterior equations iff it also solves the equation* $\pi = 0$ *for every* $\pi \in \mathcal{I}_\Sigma$.

Appendix C

Basics on Stability Analysis

Modern control theory relies strongly on the state-space analysis approach. We recall here some of the principles and tools related to this approach.

State variables and state space

The state of a dynamic system is a set of variables (called state variables) such that the knowledge of these variables at $t = t_0$, together with the knowledge of the input for $t \geq t_0$, completely determines the behavior of the system for any time $t \geq t_0$[Oga90]. The state variables is hence the n-tuple $(x_1, x_2, \ldots, x_n)$ of variables needed to fully describe the system. Note that state variables are not necessarily measurable physical quantities. Note also that if you have a dynamical system whose output depends on time and input, it must involve elements that memorize the values of the input as a function of time. Since integrators in continuous-time control systems serve as memory devices, the outputs of such integrators can be considered as the variables that define the internal state of the dynamical system. In this case, the outputs of the integrators serve as state variables.

All the possible values of the state variables $(x_1, x_2, \ldots, x_n)$ define an n-dimensional space whose coordinate axes consist of the x_i axes $i \in (1, 2, \ldots, n)$. At any time, the set $(x_1, x_2, \ldots, x_n)$ defines a single point in the State space and hence $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is a vector in this space, called the State vector.

State-space equations

A dynamical system is frequently modeled by a set of linear (or nonlinear) differential equations, relating the behavior of the output to the input as a function of time. Since state variables "memorize" the internal state of the system, it is possible to express the dynamics of the system as a set of differential equations relating the dynamics of the state

and the output to the actual state, the input and possibly the time[1]. Mathematically, if the input of the system is represented by a vector (multiple input) $\mathbf{u}$, the output is the vector (multiple output) $\mathbf{y}$ and the the state vector is $\mathbf{x}$, then the following expressions define control systems in state-space regarding linear autonomous systems (we will not focus on time-varying systems) :

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$
$$\dot{\mathbf{y}}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

where $\mathbf{A}$ is the state matrix, $\mathbf{B}$ is the input matrix, $\mathbf{C}$ is the output matrix and $\mathbf{D}$ is the direct transmission matrix.

Nonlinear autonomous control systems are modeled:

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

Stability

The definitions, theorems and ideas discussed in this section are borrowed from [Kha96].
Equilibrium point:

Definition C-0.1 *The equilibrium point $x = 0$ of $\dot{x} = f(x)$ is :*

- *stable if, for each $\epsilon > 0$, there is $\delta = \delta(\epsilon) > 0$ such that :*

$$\| x(0) \| < \delta \Rightarrow \| x(t) \| < \epsilon, \forall t \geq 0$$

- *unstable if not stable.*
- *asymptotically stable if it is stable and δ can be chosen such that :*

$$\| x(0) \| < \delta \Rightarrow \lim_{t \to \infty} x(t) = 0$$

This definition states clearly the concept of stability around the origin: if the trajectories of the system start at a neighborhood δ (possibly function of ϵ) of the equilibrium point, then they remain within a certain ball of radius ϵ containing this equilibrium point, and our system is said to be stable. Asymptotic stability is a stronger condition since we force the system not only to remain within the ball defined by ϵ but also to converge to the equilibrium point in the long run.

Lyapunov Stability : Consider the following autonomous system :

$$\dot{x} = f(x) \tag{C-1}$$

[1]This is the difference between time-invariant and time-varying systems, for which the dependence on time of the internal state is explicit.

where $x \in \mathbb{R}^n$ and f is smooth. Suppose (C-1) admits a stability (or equilibrium) point. We are interested in determining the stability of (C-1) without computing its actual solutions (i.e. without solving the system). In 1892, Lyapunov showed that it is possible to find functions whose behavior along trajectories of (C-1) enable us to derive conclusions about its stability.

Theorem C-0.4 (Lyapunov) *Let $x = 0$ be an equilibrium point for (C-1) and $U \subset \mathbb{R}^n$ be a domain containing $x = 0$. Let $V : U \to \mathbb{R}$ be a continuously differentiable function, such that :*

$$V(0) = 0 \text{ and } V(x) > 0 \text{ in } U - \{0\} \tag{C-2}$$

$$\dot{V}(x) \leq 0 \text{ in } U \tag{C-3}$$

Then, $x = 0$ is stable. Moreover if

$$\dot{V}(x) < 0 \text{ in } U \tag{C-4}$$

then $x = 0$ is asymptotically stable.

where

$$\dot{V}(x) = \frac{d}{dt}V(x) = \frac{\partial V}{\partial x}f(x)$$

A continuously differentiable function satisfying C-2 and C-3 is called a *Lyapunov function*[2]. The set of points x such that $\dot{V}(x) = 0$ define a surface $V(x) = c$ called a *Lyapunov surface* or *level surface*. The condition $\dot{V} \leq 0$ implies that when a trajectory crosses a Lyapunov surface it moves inside the set $\Omega_c = \{x \in \mathbb{R}^n | V(x) \leq c\}$ and can never come out again. Moreover, if $\dot{V} < 0$, the trajectory moves from one Lyapunov surface to an inner Lyapunov surface with a smaller c, showing that $V(x) = c$ shrinks to the origin as time progresses (see Figure C-1).

A function V is said to be **positive definite**(resp. **negative definite**) if $\forall x \in \mathbb{R}^n$ $\mathbf{V(x) > 0}$(resp. $\mathbf{V(x) < 0}$). It is said to be **positive semidefinite**(resp. **negative semidefinite**) if $\forall x \in \mathbb{R}^n$ $\mathbf{V(x) \geq 0}$(resp. $\mathbf{V(x) \leq 0}$).

Suitable Lyapunov function candidates are quadratic functions of the form :

$$V(x) = x^T P x = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}x_i x_j$$

where P is a real symmetric positive definite matrix.

[2]A more rigurous definition may involve "comparison functions" of the class $\mathcal{K}_\infty$, $\alpha : \mathbb{R}^+ \to \mathbb{R}^+$ which are continuous, strictly increasing, unbounded and satisfy $\alpha(0) = 0$. Thence the definition may be completed by the following sentence:

$$\exists \alpha, \beta \in \mathcal{K}_\infty,\ \alpha(|x|) \leq V(x) \leq \beta(|x|)\ \forall x \in \mathbb{R}^n.$$

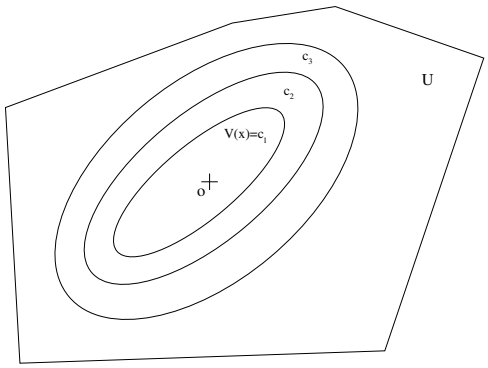


Figure C-1: Level surfaces of a Lyapunov function (i.e. set of points x such that $\dot{V}(x) = 0$) where $c_1 < c_2 < c_3$

There is no methodology in finding a Lyapunov function for any particular problem. It is a trial and error approach, apart from the energy equations that can be formulated for electrical or mechanical systems. This is also reinforced by the fact that the conditions in the theorem are only sufficient, since failure of a Lyapunov function candidate to satisfy the conditions for stability or asymptotic stability does not mean that the equilibrium point is not stable or asymptotically stable.

Asymptotic stability is an important problem in control theory. Its proof may be particularly difficult. When the origin is asymptotically stable, we are interested in knowing how far can we go from it in order to keep the stability properties of the system. We define thus a *region of attraction* also referred to as the *attraction domain*. Suppose that $\phi(t, x)$ is the solution of (C-1) passing by x at $t = 0$. The region of attraction is therefore defined as the set of all points x such that $\lim_{t\to\infty} \phi(t, x) = 0$.

Trying to find such a domain analytically might be very difficult. It is however possible to estimate it using Lyapunov functions satisfying conditions of asymptotic stability of theorem C-0.4 and such that the sets $\Omega_c = \{x \in \mathbb{R}^n | V(x) \leq c\}$ are bounded and contained in U. For, if these conditions are met, we know that every trajectory starting inside Ω_c remains in Ω_c and approaches the origin as $t \to \infty$. Thus Ω_c is an estimate of the region of attraction[3]. Even more importantly is that if this region is proved to be $\mathbb{R}^n$, we say that the system is *globally asymptotically stable.*

The next theorem formalizes the conditions for global asymptotic stability:

Theorem C-0.5 (Barbashin-Krasovskii) *Let $x = 0$ be an equilibrium point for (C-1). Let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuous differentiable function such that*

$$V(0) = 0 \text{ and } V(x) > 0, \quad \forall x \neq 0 \tag{C-5}$$

$$\| x \| \to \infty \Rightarrow V(x) \to \infty \tag{C-6}$$

$$\dot{V}(x) < 0, \quad \forall x \neq 0 \tag{C-7}$$

then $x = 0$ is globally asymptotically stable.

[3]Better estimations of this region may be found in advanced stability analysis methods (see [Kha96]).

Lyapunov Stability for Time-invariant Linear Systems : The next theorem applies to linear time-invariant systems:

Theorem C-0.6 (taken from [Kha96]) *The equilibrium point $x = 0$ of the linear time-invariant system : $\dot{x} = Ax$ is stable if and only if all eigenvalues λ_i of A satisfy $Re(\lambda_i) \leq 0$ and every eigenvalue with $Re(\lambda_i) = 0$ has an associated Jordan block of order one. The equilibrium point $x = 0$ is (globally) asymptotically stable if and only if all eigenvalues of A satisfy $Re(\lambda_i) < 0$.*

When all eigenvalues of A satisfy $Re(\lambda_i) < 0$, A is called a *Hurwitz matrix.*

Asymptotic stability can alternatively be investigated using Lyapunov's method. Recall that a candidate Lyapunov function is:

$$V(x) = x^T P x$$

where P is a real symmetric positive definite matrix. The derivative of V along the trajectories of the linear system $\dot{x} = Ax$ is given by:

$$\dot{V}(x) = -x^T Q x$$

where Q is a symmetric matrix defined by:

$$PA + A^T P = -Q \tag{C-8}$$

In analyzing the stability of $\dot{x} = Ax$ we would need to chose a positive definite matrix P and then check for the negative definiteness of $\dot{V}(x)$. It is however possible to undertake the reverse procedure, that is, to chose Q as a real symmetric positive definite matrix and solve (C-8) for P. If (C-8) has a positive definite solution, we can then conclude that the origin is asymptotically stable :

Theorem C-0.7 *A matrix A is a stability matrix; that is, $Re\lambda_i < 0$ for all eigenvalues of A, if and only if for any given positive definite matrix Q, there is a positive definite matrix P that satisfies the Lyapunov equation (C-8). Moreover, if A is a stability matrix, then P is the unique solution of (C-8).*

Lyapunov Stability of time-invariant Non-linear Autonomous Systems : The next theorem applies to non-linear systems of the form $\dot{x} = f(x)$:

Theorem C-0.8 (Lyapunov's indirect method) *Let $x = 0$ be an equilibrium point for the non-linear system :*

$$\dot{x} = f(x)$$

where $f : D \to \mathbb{R}^n$ is continuously differentiable and D is a neighborhood of the origin. Let

$$A = \frac{\partial f}{\partial x}(x)\Big|_{x=0}$$

Then,

1. *The origin is asymptotically stable if* $Re(\lambda_i) < 0$ *for all eigenvalues of* A.
2. *The origin is unstable if* $Re(\lambda_i) > 0$ *for one or more of the eigenvalues of* A.

Appendix D

Preliminary Study on the Bi-steerable Odometry

In this appendix, we propose a model for the odometry of the bi-steerable robot and a "general" model for dealing with the uncertainty. By "general" we mean that the model can be particularised under specific assumptions, regarding the nature of the uncertainty in the odometry model. To this end, we use a Bayesian formalism. Hence we give a general *description*[1] of the uncertainty of our odometry model. For our purposes, we use the methodology of Lebeltel [LDBM00] as a design tool.

Let us start by discussing elements to take into consideration when designing the odometry of a bi-steerable car ([HPS02]). The assumption is that the robot is equipped with either absolute or relative encoders in the four wheels.

D-1 Modeling odometry for a BiS-car: a few considerations

Standard *Differential Heading* odometry results ([Wan88], [CK97]) cannot be applied directly to the bi-steerable robot, for there is no fixed axle in this case. Hence we discuss some modeling choices for an axle with steering wheels, for odometry purposes.

D-1.1 Modeling an axle with steerable wheels for odometry

According to the *rolling compatibility conditions* given by Alexander-Maddocks [AM89], when the wheels are steered each one turns a different amount. Hence the mechanical design of a steering axle is such that each wheel has different attitude as soon as the steering wheel turns. Consequently the axle has a single instantaneous centre of gyration and describes an exact circle when turning (see Figure D-1.a).

At this stage geometric modeling choices can be made. If we tried to compute

[1] Here *description* has a precise meaning as pointed out further in the text.

the arc length ΔS (described by the middle of the axle) we would probably improve the accuracy of the odometry at the cost of increased complexity. However, it is not sure that the required information to compute ΔS would be available. Indeed, the control input usually considered is the rotation angle of the steering wheel. If we wanted to make exact computations, we would need the exact values of angles φ_1 and φ_2, which may not be available. For instance, as far as our robot is concerned we do not have this information. In fact, the actual robot is controlled through a single steering command (like for a mechanical steering wheel). Thus the assumption that the actual system corresponds to the one depicted in Figure D-1.b seems a reasonable one for modeling purposes. For this reason, we decided to look for a trade-off between accuracy and low complexity in the following way.

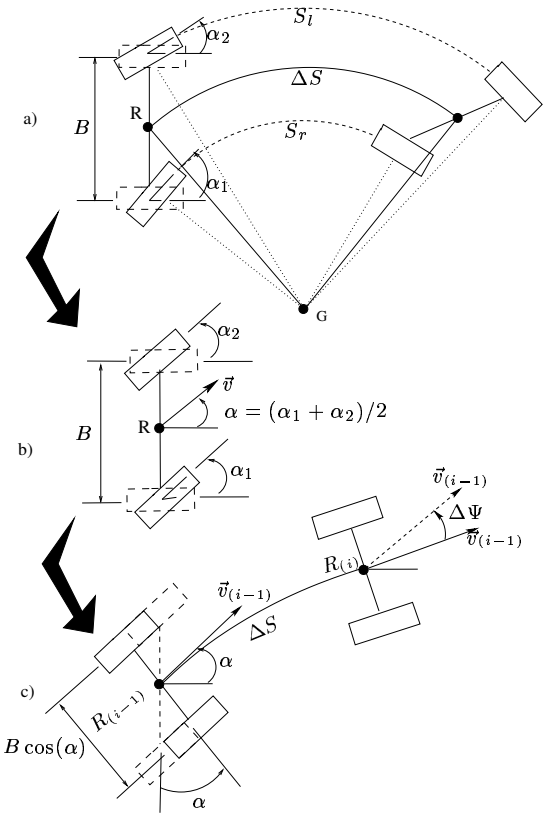


Figure D-1: a) Mechanical steering capability of a BiS-car; b) Geometric model for kinematics study: a single steering angle $\alpha = (\alpha_1 + \alpha_2)/2$ is assumed; c) Alternative modeling for odometry purposes: we make the axle to pivot round the middle point as soon as the steering angle is applied. Thus the wheels base becomes $B\cos(\alpha) \leq B$.

We have seen that for the study of the kinematics of the vehicle the modeling choice was as shown in Figure D-1.b. This could be a first choice if the relations linking the wheels' speeds to the kinematics of point R are known.

For our odometry purposes, we decided to keep this choice but with a different geometrical interpretation: we pivot the steering axle round the middle point. This leads naturally to a *differential heading* odometry model, entailing a "shortening" of the axle

(see [HPS02]):

$$\begin{cases} x_i &= x_{i-1} + \Delta S \cos(\alpha_{i-1} + \theta_{i-1} + \frac{1}{2}\Delta\theta) \\ y_i &= y_{i-1} + \Delta S \sin(\alpha_{i-1} + \theta_{i-1} + \frac{1}{2}\Delta\theta) \\ \theta_i &= \theta_{i-1} + \Delta\theta \end{cases} \tag{D-1}$$

Indeed, the original wheels base B becomes $B\cos(\alpha)$ as represented in Figure D-1.c. This is the model used during the experiments reported in Chapter V.

This model can be justified in the following way. In fact, one can imagine that the virtual wheels (Figure D-1.c) will describe exactly the same trace than the "real" wheels (Figure D-1.b) but shifted back—i.e. they start and stop sooner. Moreover, if we assume that our actual mechanical system corresponds to (Figure D-1.b), then the traces $S_{r/l}$ of these wheels are exactly equal to those of the virtual wheels. Thus it makes sense to perform computations with (Figure D-1.c) using actual data coming from the robot.

We now discuss an uncertainty propagation model for odometry. In order to keep our study independent of the odometry modeling choices, the following section aims at establishing an error propagation model for a general odometry function.

D-2 Uncertainty model as a joint probability distribution

D-2.1 Introduction

When designing a dead-reckoning sensor, it is important to account for its *accuracy* (i.e. how much the sensor reading—mean value—is close to the *true* value; usually issued from a more accurate instrument) and *precision* (i.e. the statistical deviation (or uncertainty) from the mean sensor reading). In this respect, the Robotics community has dedicated considerable effort in addressing questions of how errors should be represented and how they can be eliminated through calibration [Wan88], [BF95], [CK97], [BTNO99]. However, more fundamental questions that begin to be addressed concern the nature of the errors and of the uncertainty.

Regarding accuracy in odometry, analytical solutions for navigational error propagation are proposed in [Kel01]. The analysis focus on the general solution to the linearized dynamics of the odometry model. This method can be used to determine the accuracy of the target system, when comparing the approximate linearized results to exact numerical solutions.

On the other hand, recent work in AI applied to robot perception and control, have addressed the uncertainty issue in the setting of Bayesian inference and learning [BDL+99a]. In this framework, the notion of uncertainty stems from a fundamental problem: any model representing the real (physical) system is incomplete by nature. A simple illustration of this statement is the following. When we look at the modeling process depicted in Figures D-1.(a) to (c) above, elementary questions arise: how much do we know about any of the models with respect to the real system? Can we render this knowl-

edge computable? Faced with these questions, Bessiere *et al* account for the utilization of probabilities to formally translate incompleteness into uncertainty [BDL+99b]. The theoretical foundations rely in the maximum entropy principle. The core of the approach is based on the *description* paradigm. Formally, a description is the joint probability distribution, determined upon *a priori* knowledge and experimental data, of a set of variables. In this respect, Lebeltel proposes a methodology for Bayesian robot programming, that has been shown to be also effective for designing model based sensors [LDBM00].

Besides the modeling of uncertainty, we are interested in propagating it along time and eventually reduce it. In order to cope with this task, a common agreement is to embrace estimation theory and in particular filtering techniques. Filtering concerns the propagation in time of the uncertainty about the state in function of the uncertainty of the model and its parameters. The process consists in confronting *predicted* information about the state of the robot, at the time new information is obtained via *observations*.

In the presence of statistical knowledge about some of the variables, the filtering process resorts to Bayesian calculus. The aim is to compute the *a posteriori* probability distribution (or density function) of the state in function of the *a priori* estimated distribution (i.e. from the previous iteration) and current observations. The result is a density function accounting for the global uncertainty of the state at a precise instant in time. The general approach usually involves the computation of an integral encompassing the joint probability distribution of all the variables. In this case, the estimation of the state may lead to very difficult computational problems (e.g. see [MMB00]). However, if the system is linear and assumed to be Gaussian, it is possible to use the Kalman Filter (KF) [Kal60]; the advantage being the possibility to compute analytically the sought distribution and to obtain an optimal estimate of the state. If the system is nonlinear, as in the majority of real situations, the linearization of the model leads to a sub-optimal estimation process known as the Extended Kalman Filter (EKF).

As for any measurement instrument, the known sources of uncertainty are assimilated to what we call *systematic* errors— i.e. quantifiable and hence susceptible of being calibrated. Other sources (spurious and hence very difficult to assess) account for what are called *non-systematic* errors. Typically, sensor models are calibrated by identifying the statistical information of the parameters encompassing uncertainty (systematic errors). Eventually, some exteroceptive sensor is used in order to reduce the cumulation of uncertainty due to systematic and non-systematic errors.

In this section we propose to use the methodology of Lebeltel [LDBM00] (see below) as tool for obtaining a Bayesian uncertainty model for a general odometry model. We shall justify this choice once we would have introduced the methodology.

D-2.2 Design methodology

The methodology employed is the one presented in [LDBM00], but somehow adapted for the design of the uncertainty model for odometry. We summarize here the main steps:

1. ***Specification of the variables***: In a first-degree analysis the variables come directly from the odometry model. However, the set of variables could increase ar-

bitrarily with new a priori knowledge. The complete specification of the variables requires following the next steps:

(a) The definition of the variables and their respective validity domains.

(b) The *explicit* statement of interdependence relationships between the variables by means of conditional probability distributions. This step allows to simplify the joint probability distribution according to our assumptions.

(c) The formulation of a probabilistic *question* that will be eventually used (see point 3 below). The "question" is the target probability distribution. It involves some unknown variables and others whose values are observed at the time the question is asked. The reason to define the question at this stage is that the uncertainty of some variables can be neglected. Indeed, as the question may involve the observation of some variables, at the time the question is asked, the uncertainty of those variables is meaningless and therefore useless.

(d) The specification of the parametric forms of the uncertainty related to each variable. This reflects the specific knowledge (subjective or statistical) we have about the system under assessment.

2. ***Identification of the parameters*** The parameters of the forms above must be defined according to either the *a priori* knowledge of the designer or from a set of experimental data enabling its identification.

3. ***Utilization*** Once the set of variables has been defined and the parametric forms have been identified we have a completely defined description. Hence it is possible now to interrogate this description by means of the probabilistic question. The answer is therefore a new probability distribution on our target variables.

A few reasons can be given here to justify the approach for our purposes. First, the Bayesian formalism enables natural expansions for data fusion purposes, aiming at global localization systems (e.g. see [TFBD01]). Second, the methodology of Lebeltel allows for a clear statement of independence assumptions *explicitly* reflected in the uncertainty model. Finally, we think that the methodology and the Bayesian formalism allow for a common discussion framework and for a certain kind of *traceability*. By this we mean that we are able to come back to the assumptions that originated the model; in other words we can confront the assumptions we made to a new situation or knowledge about our system.

We are now ready to discuss the application of this design process to the case of a general odometry model for the bi-steerable car.

D-3 Uncertainty model for the odometry

We are looking for a tractable expression defining the uncertainty model of a general odometry model

$$X_i = f(X_{i-1}, P) \tag{D-2}$$

where P are the model parameters. Let us apply the above methodology.

1. ***Variables specification:***

(a) Definition:

We shall assume that the robot is equipped with digital encoders allowing for reading the relative position of the wheels (i.e. its rolling displacement) and their absolute orientation. This seems a sound assumption since it is not uncommon to find low-cost relative encoders mounted on each wheel, and a single, more expensive, absolute encoder at the steering shaft. We adopt the usual convention that counter clockwise deflections induce positive readings. Moreover, we assume that we are dealing with a 4-wheel drive robot. Hence a typical configuration is that each wheel encoder is placed at the motor shaft followed by speed-reduction gears coupling the wheel to the motor (see Figure D-2).

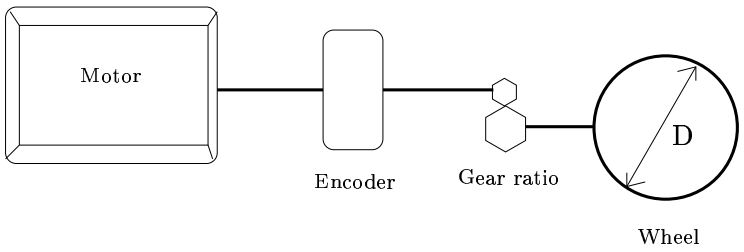


Figure D-2: Typical configuration assumed for our system

Let us call k_r(resp. k_l) the resolution of the right(resp. left) wheel encoder (in pulses/revolution); $E = (E_r, E_l, \alpha)$ the vector of encoder readings containing respectively the right and left encoder values (in pulses) and the steering encoder (in radians); let D_r(resp. D_l) be the diameter of the right (resp. left) wheel (in meters) and N_r (resp. N_l) be the right (resp. left) wheel-to-motor gear ratio.

We thus have that each gain of wheel encoder, in meters per pulse (m./pulse), is given by

$$\zeta_r = \frac{\pi D_r}{N_r \cdot k_r}, \quad \zeta_l = \frac{\pi D_l}{N_l \cdot k_l} \tag{D-3}$$

Then the length of the path ΔS and the angular displacement $\Delta\theta$, between two samples, in equations (D-2) are given by

$$\begin{aligned} \Delta S &= \frac{\zeta_r E_r + \zeta_l E_l}{2} \\ \Delta\theta &= \frac{\zeta_r E_r - \zeta_l E_l}{B\cos(\alpha)}, \end{aligned} \tag{D-4}$$

We thus have the following set of variables:

State	the state vector	$X = (x, y, \theta)$
Sensor	the encoders vector	$E = (E_r, E_l, \alpha)$
System	the robot's parameters vector	$A = (\zeta_r, \zeta_l, B)$

Equations (D-2) express the dynamics of the state as a function of these vectors; that is:

$$X_i = f(X_{i-1}, E_i, A)$$

We therefore know that the description is given, using Bayes rules, by the joint probability density

$$p\,(X_i X_{i-1} E_i A) = p\,(X_{i-1})\,p\,(E_i|X_{i-1})\,p\,(A|X_{i-1}E_i)\,p\,(X_i|X_{i-1}E_i A) \qquad \text{(D-5)}$$

Description (D-5) can be simplified under certain assumptions. At this stage, it is possible to formulate in an explicit way conditional independence hypothesis so as to simplify the model.

(b)Independence hypothesis:

We will assume that the state and the encoder readings are statistically independent; that is: current values from the encoders are uncorrelated to the previous position of the robot. Furthermore, the variables A concern only the characteristics of the robot components. Hence they are uncorrelated to the actual reading of the encoders and do not depend on the position of the robot. Under these hypothesis, the joint probability distribution (D-5) can be re-written as

$$p\,(X_i X_{i-1} E_i A) = p\,(X_{i-1})\,p\,(E_i)\,p\,(A)\,p\,(X_i|X_{i-1}E_i A) \qquad \text{(D-6)}$$

At this stage, it is possible to elucidate the physical meaning of (D-6). In fact, equation (D-6) involves systematic errors coming from $p(E)$, representing the error induced by the encoders[2] and from $p(A)$, which is the uncertainty associated to the parameters of the robot (encoder gains).

(c)Probabilistic question: the general uncertainty model

Now we shall set the question we are interested in. In fact this question will constitute the uncertainty propagation model of the odometry (D-2).

The probability distribution we are interested in, during the utilisation of the odometry model, is the following.

$$\begin{aligned} p\,(X_i|E_i) &= \frac{p\,(E_i)\int\int p\,(X_{i-1})\,p\,(A)\,p\,(X_i|X_{i-1}E_i A)\,dX_{i-1}\,dA}{p\,(E_i)} \\ p\,(X_i|E_i) &= \int\int p\,(X_{i-1})\,p\,(A)\,p\,(X_i|X_{i-1}E_i A)\,dX_{i-1}\,dA \end{aligned} \qquad \text{(D-7)}$$

A few comments about this probability distribution follow. Notice that the question reads: *what is the density function of the current position of the robot, knowing the*

[2]Typically, the constructor gives the resolution of the encoder from which it is possible to compute a quantization error.

value of the encoder readings at iteration i?. The answer reads: *the probability distribution sought is given by all possible values taken by the cumulated uncertainty* $p(X_{i-1})$*, by all possible values of the robot parameters given by* $p(A)$ *and, for each of these values, by the probability distribution* $p(X_i|X_{i-1}EA)$*, associated to odometric predictions given by* $f(X_{i-1}, E, A)$*—i.e. model (D-2).* We shall precise that the distribution $p(X_{i-1})$ contains the cumulated uncertainty since the beginning of the process and until the previous iteration. Hence we shorten notation and write: $p(X_{i-1})$, instead of $p(X_{i-1}|E_{i-1}, \ldots, E_0)$. This implies in turn that the *question-answer* process during the robot's motion, can be viewed as a Markov chain. What we mean is that, at each iteration i, the result of the question (i.e. $p(X_i|E_i)$) is inserted iteratively into the next one so as to compute a new answer.

Equation (D-7) is therefore the general uncertainty propagation model of the odometry model (D-2) under the aforementioned hypothesis.

(d)Parametric forms:

Typically at this stage, Gaussian parametric forms may be assumed. Indeed, the maximum entropy principle leads naturally to this kind of probability distributions as the number of data increases. As a consequence, the propagated uncertainty is usually seen as an ellipsoidal growing around the mean state of the robot. Moreover, Gaussian distributions together with a linear system (or its linearization) allow to use the computational efficient Kalman Filter so as to propagate the uncertainty. Notwithstanding, Gaussian and linear hypothesis may be over constraining assumptions. Yet another possibility is to use particle filters. They allow to avoid the Gaussian and linear assumptions, but resort to numerical integrations yielding approximative results [AMGC02].

In order to illustrate this step, we will assume Gaussian hypothesis. True, Gaussian hypothesis leads to a *Normal* distribution. It will remain always Normal if the following conditions are met

- It is initialized to a Normal distribution.
- The model of the system is linear (or linearized).
- All the other distributions are Normal.

Initialization implies that the distribution starts with some valid (arbitrarily set) parameters.

In what follows the notation herein will be used:

$$p(\chi) \equiv \mathcal{N}ormal(\chi, \hat{\chi}, \Sigma_\chi),$$

which stands for a (multivariate) *Normal* distribution on the variable χ, centered around a mean value (or first moment) $\hat{\chi}$ and with a covariance matrix (or second moment) Σ_χ.

Mathematically, if the dimension of χ is n_χ we have

$$\mathcal{N}ormal(\chi, \hat{\chi}, \Sigma_\chi) = \frac{1}{(2\pi)^{\frac{n_\chi}{2}}\sqrt{\det(\Sigma_\chi)}} \exp^{-\frac{1}{2}(\chi-\hat{\chi})\,\Sigma_\chi^{-1}\,(\chi-\hat{\chi})^T}.$$

We are now ready to give the parametric forms of the necessary distributions.

$p(X_{i-1})$

If we assume that X_{i-1} is Normally distributed, the parametric form is:

$$p(X_{i-1}) \equiv \mathcal{N}ormal(X_{i-1}, \hat{X}_{i-1}, \Sigma_{X_{i-1}}). \tag{D-8}$$

$\hat{X}_{i-1}$ may be for instance the most likely value for X_{i-1} and $\Sigma_{X_{i-1}}$ is the *a posteriori* uncertainty of the whole process, which becomes the *a priori* uncertainty in the next step.

Notice that the choice of a Normal distribution assumes an a priori knowledge about the initial position of the robot and its associated uncertainty.[3]

$p(A)$

This distribution stems from the fact that the actual diameter of the wheels as well as the actual wheels base cannot be known exactly. We shall assume hence that the encoder gains have a Gaussian distribution whose parameters can be eventually characterized. Thus we have:

$$p(A) \equiv \mathcal{N}ormal(A, \hat{A}, \Sigma_A), \tag{D-9}$$

where $\hat{A} := (\hat{\zeta}_r, \hat{\zeta}_r, \hat{B})$ and

$$\Sigma_A := \begin{bmatrix} \sigma^2_{\zeta_r} & 0 & 0 \\ 0 & \sigma^2_{\zeta_l} & 0 \\ 0 & 0 & \sigma^2_B \end{bmatrix}$$

is the covariance matrix of A.

$p(X_i|X_{i-1}E_iA)$

This is the uncertainty associated to the next odometric prediction. Again we consider Gaussian hypothesis to have:

$$p(X_i|X_{i-1}E_iA) \equiv \mathcal{N}ormal(X_i, f(X_{i-1}, E_i, A), \Sigma_f), \tag{D-10}$$

where Σ_f is the following covariance matrix:

$$\Sigma_f = \Sigma_{X_{i-1}} + J_A \cdot \Sigma_A \cdot J_A^T, \tag{D-11}$$

with $J_A \equiv \partial f / \partial A$.

2. ***Parameter identification***

   It should be clear now that we only need to characterize the parameters of the normal distribution $p(A)$ above—i.e. we need to find $\hat{A}$ and Σ_A through a calibration

[3] Another choice could have been a *Uniform* distribution. This case corresponds to the problem of the "kidnaped robot", where there is no a priori knowledge about the starting position.

process. Indeed, for $\Sigma_{X_{i-1}}$ we just need to give an initial value (typically a Dirac pulse is assumed if the initial position is given) and let the *filtering* process (in this case the EKF) compute its subsequent values.

3. ***Utilization of the question***

   Once all the parametric forms have been identified we have a completely defined description. Hence it is possible now to ask our probabilistic question; that is, we may now compute the distribution (D-7). As the robot moves, a new mean value $\hat{X}_i$ is computed from model (D-2), and a new uncertainty matrix Σ_{X_i} is obtained from (D-7).

Bibliography

[AM89] J. C. Alexander and J. H. Maddocks. On the kinematics of wheeled mobile robots. *Int. Journal of Robotics Research*, 8(5):15–27, October 1989.

[AMGC02] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, 2002.

[BCG+91] R.L. Bryant, S.S. Chern, R.B Gardner, H.L Goldschmidt, and P.A. Griffiths. *Exterior Differential Systems*. Springer-Verlag, 1991.

[BCL92] J.-D. Boissonnat, A. Cérézo, and J. Leblond. Shortest paths of bounded curvature in the plane. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Nice (FR), May 1992.

[BCL94] J.-D. Boissonnat, A. Cérézo, and J. Leblond. A note on shortest paths in the plane subject to a constraint on the derivative of the curvature. Research Report 2160, Inst. Nat. de Recherche en Informatique et en Automatique, Rocquencourt (FR), January 1994.

[BDL+99a] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer, and K. Mekhnacha. Interprétation ou description (I): Proposition pour une théorie probabiliste des systèmes cognitifs sensori-moteurs. *Intellectica*, 26–27(1–2):257–311, 1999.

[BDL+99b] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer, and K. Mekhnacha. Interprétation ou description (II): Fondements mathématiques de l'approche F+D. *Intellectica*, 26–27(1–2):313–336, 1999.

[BF95] Johann Borenstein and Liqiang Feng. Correction of systematic odometry errors in mobile robots. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Pittsburgh, PA (US), October 1995.

[BGMPG99] G. Baille, Ph. Garnier, H. Mathieu, and R. Pissard-Gibollet. Le cycab de l'inria rhône-alpes. Technical Report 229, Inst. Nat. de Recherche en Informatique et en Automatique, Montbonnot (FR), April 1999.

[BJR98] A. Bellaïche, F. Jean, and J.-J. Risler. *Robot Motion Planning and Control*, chapter Geometry of Nonholonomic Systems. Springer-Verlag, 1998. Lecture notes in control and Information Sciences.

[BL91] J. Barraquand and J.C. Latombe. Robot motion planning: a distributed representation approach. *Int. Journal of Robotics Research*, vol.10(6):628–649, 1991.

[BMR90] A.M. Bloch, N.H. McClamroch, and M. Reyhanoglu. Controllability and stabilizability properties of a nonholonomic control system. In *Proc. of the Conf. on Decision and Control*, pages 1312–1314, Honolulu, HI, December 1990.

[Bro81] R.W. Brockett. Control theory and singular remannian geometry. In *New Directions in Applied Mathematics*, pages 11–27. New York: Springer-Verlag, 1981.

[Bro83] R.W. Brockett. Asymptotic stability and feedback stabilization. In R.W. Brockett, R.S. Millman, and H.J. Sussmann, editors, *Differential Geometric Control Theory*, pages 181–191, Boston, MA: Birkhäuser, 1983.

[BTNO99] M Bak, Larsen T.D., Andersen N.A., and Ravn O. Auto-calibration of systematic odometry errors in mobile robots. In *SPIE, Mobile Robots XIV*, Boston, Massachusets, 1999.

[BTS95] L.G. Bushnell, D.M. Tilbury, and S.S. Sastry. Steering three-input nonholonomic systems: The fire truck example. *IEEE Trans. Autom. Contr.*, vol.14(4):366–381, August 1995.

[Car14] Elie Cartan. Sur l'équivalence absolue de certains systèmes d'équations différentielles et certaines familles de courbes. *Bull. Soc. Math. France*, 42:12–48(*Oeuvres Complètes* pp. 1133–1168), 1914.

[CdWS92] C. Canudas de Wit and O.J. Sørdalen. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Trans. Autom. Contr.*, vol.37(11):1791–1797, November 1992.

[Cho39] W.L. Chow. Systeme von linearen partiellen differential gleichungen erster ordnung. *Math. Ann.*, (117):98–105, 1939.

[CK97] Kok Seng Chong and Lindsay Kleeman. Accurate odometry and error modelling for a mobile robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM (US), April 1997.

[CLM89] B. Charlet, J. Lévine, and R. Marino. On dynamic feedback linearization. *Systems and Control Letters*, **13**:143–151, 1989.

[CLM91] B. Charlet, J. Lévine, and R. Marino. Sufficient conditions for dynamic state feedback linearization. *SIAM J.Control and Optimization.*, vol.29(1):38–57, January 1991.

[Cor92] Jean-Michel Coron. Global asymtotic stabilization for controllable systems without drift. *Math. Control Signals Systems*, **5**:295–312, 1992.

[Cor94] Jean-Michel Coron. On the stabilization of controllable and observable systems by an output feedback law. *Math. Control Signals Systems*, **7**:187–216, 1994.

[Cor98] Jean-Michel Coron. On the stabilization of some nonlinear control systems: results,tools, and applications. *Nonlinear Analysis, Differential Equations, and Control,* F.H. Clarke and R.Stern Eds., NATO Advanced Study Institute, Montreal, pages 307–367, Kluwer, 1998.

[DLOS98] A. De Luca, G. Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. In J.-P. Laumond, editor, *Robot motion planning and control*, volume 229 of *Lecture Notes in Control and Information Science*, pages 171–253. Springer, 1998.

[DM85] J. Descusse and C.H. Moog. Decoupling with dynamic compensation for strong invertible affine nonlinear systems. *Int. J. Control*, vol. 42(6):1387–1398, 1985.

[dNBC92] B. d'Andrea Novel, G. Bastin, and G. Campion. Dynamic feedback linearization of nonholonomic wheeled mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2527–2532, Nice (FR), May 1992.

[dNMS92] B. d'Andréa Novel, Ph. Martin, and R. Sépulchre. Full feedback linearization of a class of mechanical systems. In S.Kimura and H.Kodama, editors, *Recent Advances in Mathematical Theory of Systems, Control, Network and Signal Processing II (MTNS 91, Kobe, Japan)*, pages 327–333, Mita press. 1992.

[Dub57] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–517, 1957.

[FLM+97] M. Fliess, J. Lévine, P. Martin, F. Ollivier, and P. Rouchon. Controlling nonlinear systems by flatness. In C.I. Byrnes, B.N. Datta, D.S. Gilliam, and C.F. Martin, editors, *sacitfc*, pages 137–154, Boston, MA: Birkhäuser, 1997.

[FLMR95a] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Design of trajectory stabilizing feedback for driftless flat systems. In *Proc. of the European Control Conference*, pages 1882–1887, Rome, Italy, september 1995.

[FLMR95b] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defects of nonlinear systems: introductory theory and examples. *Int. Journal of Control*, 61(6):1327–1361, 1995.

[FLMR99] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. A lie-bäcklund approach to equivalence and flatness of nonlinear systems. *IEEE Trans. Autom. Contr.*, vol.44(5):922–937, May 1999.

[Gol80] Herbert Goldstein. *Classical Mechanics.* Addison-Wesley, 1980. Second edition.

[Hem94] A. Hemami. A control scheme for low speed automated vehicles with double steering. In *Proc. of the Conf. on Decision and Control*, pages 2452–2454, Lake Buena Vista, FL (US), December 1994.

[Hir79] R.M. Hirschorn. Invertibility of multivariable nonlinear control systems. *IEEE Trans. Autom. Contr.*, vol. AC-24:855–865, December 1979.

[HPS02] J. Hermosillo, C. Pradalier, and S. Sekhavat. Modelling odometry and uncertainty propagation for a bi-steerable car. In *Proc. of the IEEE Intelligent Vehicle Symp.*, Versailles (FR), June 2002. Poster session.

[HPS+03] J. Hermosillo, C. Pradalier, S. Sekhavat, Ch. Laugier, and G. Baille. Towards motion autonomy of a bi-steerable car: Experimental issues from map-building to trajectory execution. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Taipei (TW), May 2003.

[HS03] J. Hermosillo and S. Sekhavat. Feedback control of a bi-steerable car using flatness; application to trajectory tracking. In *Proc. of the American Control Conference*, Denver, CO (US), June 2003.

[Isi95] Alberto Isidori. *Nonlinear Control Systems.* Springer-Verlag, 1995. 3rd. edition.

[JR80] B. Jakubczyk and W. Respondek. On linearization of control systems. *Bulletin de l'Académie Polonaise des Sciences, Série Sci. Math.*, vol.28(9—10):517–522, 1980.

[Kai80] Thomas Kailath. *Linear Systems.* Prentice-Hall Information and System Sciences Series, 1980.

[Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering; Transactions of the ASME*, pages 35–45, March 1960.

[Kel01] Alonzo Kelly. General solution for linearized systematic error propagation in vehicle odometry. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Hawaii, HI (US), October-November 2001.

[KH89] Y. Kanayama and B. I. Hartman. Smooth local path planning for autonomous vehicles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 1265–1270, Scottsdale, AZ (US), May 1989.

[KH97] Y. Kanayama and B. I. Hartman. Smooth local path planning for autonomous vehicles. *Int. Journal of Robotics Research*, 16(3):263–283, 1997.

[Kha96] Hassan K. Khalil. *Nonlinear Systems.* Prentice-Hall, 1996.

[KKMN91] Y. Kanayama, Y. Kimura, F. Myazaki, and T. Noguchi. A stable tracking control method for a non-holonomic mobile robot. In *Proc. of the IEEE-RSJ Int. Workshop on Intelligent Robots and Systems*, volume 2, pages 1236–1241, Osaka (JP), 1991.

[Lam97] F. Lamiraux. *Robots Mobiles à Remorque : de la Planification de chemins à l'Exécution de Mouvements.* PhD thesis, Institut National Polytechnique de Toulouse, 1997. LAAS-CNRS Report 97327.

[Lan86] Saunders Mac Lane. *Mathematics Form and Function.* Springer-Verlag, 1986. New-York Berlin Heidelberg Tokyo.

[Lat91a] J.-C. Latombe. *Robot motion planning.* Kluwer Academic Press, 1991.

[Lat91b] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, USA, 1991.

[Lau86] J.-P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. In *Proc. of the Int. Conf. on Intelligent Autonomous Systems*, pages 346–354, Amsterdam (NL), December 1986.

[Lau93] J.-P. Laumond. Singularities and topological aspects in nonholonomic motion planning. In Zexiang Li and J.F. Canny, editors, *Nonholonomic Motion Planning*, pages 149–199. Kluwer Academic Publishers, 1993.

[Lau98] J.P. Laumond, editor. *Robot Motion Planning and Control.* Springer-Verlag, 1998.

[LC92] Z. Li and J. F. Canny, editors. *Nonholonomic Motion Planning*, volume 192 of *The Kluwer Int. Series in Engineering and Computer Science.* Kluwer Academic Press, 1992.

[LDBM00] O. Lebeltel, J. Diard, P. Bessière, and E. Mazer. A Bayesian framework for robotic programming. In *Proc. of the Int. Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, Paris (FR), July 2000.

[LJTM94] J.-P. Laumond, P. E. Jacobs, M. Taïx, and R. M. Murray. A motion planner for non-holonomic mobile robots. *IEEE Trans. Robotics and Automation*, 10(5):577–593, October 1994.

[LL97] F. Lamiraux and J.P. Laumond. Flatness and small-time controllability of multi-body mobile robots: applications to motion planning. In *European Conf. on Control*, 1997.

[LS93] G. Lafferriere and H.J. Sussmann. A differential geometric approach to motion planning. In Zexiang Li and J.F. Canny, editors, *Nonholonomic motion planning*, pages 235–270. Kluwer Academic, 1993.

[LS98] Winfried Lohmiller and Jean-Jacques E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, Elsevier Science Ltd., 1998.

[LSL98] J.P. Laumond, S. Sekhavat, and F. Lamiraux. *Robot Motion Planning and Control*, chapter Guidelines in Nonhonomic Motion Planning. Springer-Verlag, 1998. Lecture notes in control and Information Sciences. To appear.

[LSL99] F. Lamiraux, S. Sekhavat, and J.-P. Laumond. Motion planning and control for Hilare pulling a trailer. *IEEE Trans. Robotics and Automation*, 15(4):640–652, August 1999.

[LSSL02] F. Large, S. Sekhavat, Z. Shiller, and C. Laugier. Using non-linear velocity obstacles to plan motions in a dynamic environment. In *Proc. of the Int. Conf. on Control, Automation, Robotics and Vision*, Singapore (SG), December 2002.

[Mar92] Ph. Martin. *Contribution à l'étude des systèmes différentiellement plats.* Thèse de doctorat, Ecole des Mines de Paris, Paris (FR), December 1992.

[MMB00] K. Mekhnacha, E. Mazer, and P. Bessière. A robotic CAD system using a Bayesian framework. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Takamatsu (JP), November 2000.

[MR94] P. Martin and P. Rouchon. Feedback linearization and driftless systems. *Math. Contr. Signal Syst.*, vol.7:pp.235–254, 1994.

[MR95] Phillipe Martin and Pierre Rouchon. Any (controllable) driftless system with m inputs and $m + 2$ states is flat. In *Proc. of the Conf. on Decision and Control*, 1995.

[MR96] Phillipe Martin and Pierre Rouchon. Flatness and sampling control of induction motors. In *IFAC Motion Control*, pages 389–394, San Francisco, US, 1996.

[MS93] R.M. Murray and S.S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Trans. Autom. Contr.*, vol.38(5):700–716, May 1993.

[MS97] P. Morin and C. Samson. Application of backstepping techniques to the time-varying exponential stabilisation of chained form systems. *European Journal of Control*, (3):15–36, 1997.

[Mur90] R.M. Murray. *Robotic Control and Nonholonomic Motion Planning.* Ph.D. Thesis, Univ. of California at Berkeley, December 1990.

[Mur94] Richard M. Murray. Nilpotent bases for a class of non-integrable distributions with applications to trajectory generation for nonholonomic systems. *Math. Contr. Signal Syst.*, 7:58–75, 1994.

[NRM95] M. Van Nieuwstadt, M. Rathinam, and R.M. Murray. Differential flatness and absolute equivalence of nonlinear control systems. Technical report 94006, California Institute of Technology, Pasadena, CA, August 1995.

[Oga90] Katsuhiko Ogata. *Modern Control Engineering.* Prentice-Hall, 1990. International Editions.

[Olv93] Peter J. Olver. *Applications of Lie Groups to Differential Equations.* Springer-Verlag, 1993. 2nd. edition.

[Pom97] Jean-Baptiste Pomet. On dynamic feedback linearization of four-dimensional affine control systems with two inputs. *ESAIM:COCV.*, vol.2:151–230, June 1997.

[PS02] C. Pradalier and S. Sekhavat. Concurrent localization, matching and map building using invariant features. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, 2002.

[RFLM93a] P. Rouchon, M. Fliess, J. Lévine, and P. Martin. Flatness and motion planning: the car with n trailers. In *Proc. of the European Control Conference*, pages 1518–1522, Groningen, 1993.

[RFLM93b] P. Rouchon, M. Fliess, J. Lévine, and P. Martin. Flatness, motion planning and trailer systems. In *IEEE Int. Conf. Dec. Contr.*, pages 2700–2705, San Antonio, Texas, December 1993.

[Rou94] P. Rouchon. Necessary condition and genericity of dynamic feedback linearization. *J. Math. Systems Estimation Control*, 4(2), 1994.

[RS90] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[SAA91a] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1136–1141, Sacramento, CA (US), April 1991.

[SAA91b] C. Samson and K. Ait-Abderrahim. Feedback stabilization of a nonholonomic wheeled mobile robot. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1242–1246, Osaka (JP), November 1991.

[Sam91] C. Samson. Velocity and torque feedback control of a nonholonomic cart. In C. Canudas de Wit, editor, *Advanced Robot Control*, volume 162 of *Lecture Notes in Control and Inform. Sci.*, pages 125–151. Springer-Verlag, 1991.

[Sam95] C. Samson. Control of chained systems, application to path following and time-varying point-stabilization of mobile robots. *IEEE Trans. Autom. Contr.*, vol.40(1):64–77, January 1995.

[SB98] P. Souères and J.D. Boissonnat. Optimal trajectories for nonholonomic mobile robots. In J.-P. Laumond, editor, *Robot motion planning and control*, volume 229 of *Lecture Notes in Control and Information Science*, pages 93–170. Springer, 1998.

[Sch98] A. Scheuer. *Planification de chemins à courbure continue pour robot mobile non-holonome*. Thèse de doctorat, Inst. Nat. Polytechnique de Grenoble, Grenoble (FR), January 1998.

[SCM02] J. Stephant, A. Charara, and D. Meizel. Force model comparison on the wheel-ground contact for vehicle dynamics. In *IEEE Intelligent Vehicle Symposium*, Versailles, France, June 2002.

[SEKR91] D. Simon, B. Espiau, K. Kapellos, and Pissard-Gibollet R. The orccad architecture. *Int. Journal of Robotics Research, Special issues on Integrated Architectures for Robot Control and Programming*, vol.17(4):338–359, April 1991.

[SF86] Mitsuji Sampei and Katsuhisa Furuta. On time scaling for nonlinear systems: application to linearization. *IEEE Trans. Autom. Contr.*, vol.AC.31(5):pp.459–462, may 1986.

[SFS86] S. Sano, Y. Furukuwa, and S. Shiraishi. Four wheel steering system with rear wheel steer angle controlled as a function of steering wheel angle. In *Int. congress and Exposition*, Detroit, Michigan, February 24-28 1986. SAE paper 860625.

[SH00] S. Sekhavat and J. Hermosillo. The cycab robot: a differentially flat system. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Takamatsu (JP), November 2000.

[SHH92] Zvi Shiller and Lu Hsueh-Hen. Computation of path constrained time optimal motions with dynamic singularities. *Journal of Dynamic Systems, Measurement, and Control*, 114:34–40, March 1992.

[Sil69] L.M. Silverman. Inversion of multivariable linear systems. *IEEE Trans. Autom. Contr.*, vol. AC-14(6):270–276, June 1969.

[SJ72] H.J. Sussmann and V.J. Jurdjevic. Controllability of nonlinear systems. *J.Differential Equations*, vol. 12:95–116, 1972.

[SL96] P. Souères and J.-P. Laumond. Shortest paths synthesis for a car-like robot. *IEEE Trans. Autom. Contr.*, 41(5), May 1996.

[SL98] S. Sekhavat and J.-P. Laumond. Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems. *IEEE Transaction Robotics Automat.*, 14(5):671–680, October 1998.

[SLL+97] S. Sekhavat, F. Lamiraux, J-P. Laumond, G. Bauzil, and A. Ferrand. Motion planning and control for hilare pulling a trailer: experimental issues. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3306–3311, Albuquerque, NM (US), April 1997.

[SO95] O.J. Sørdalen and Egeland O. Exponential stabilization of nonholonomic chained systems. *IEEE Trans. Autom. Contr.*, vol.40(1):35–49, January 1995.

[Son99] Eduardo D. Sontag. Stability and stabilization: discontinuities and the effect of disturbances. *Nonlinear Analysis, Differential Equations, and Control,* F.H. Clarke and R.Stern Eds., NATO Advanced Study Institute, Montreal, pages 551–598, Kluwer,1999.

[SRH01] S. Sekhavat, P. Rouchon, and J. Hermosillo. Computing the flat outputs of engel differential systems the case study of the bi-steerable car. In *Proc. of the American Control Conf.*, Arlington, VA (US), June 2001.

[STIN91] M. Sampei, T. Tamura, T. Itoh, and M. Nakamichi. Path tracking control of trailer-like mobile robot. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 193–198, Osaka (JP), November 1991.

[TFBD01] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1):99–142, January 2001.

[TMS95] D. Tilbury, R. Murray, and S. Sastry. Trajectory generation for the n-trailer problem using goursat normal form. *IEEE Transaction Automat. Contr.*, 40(5):802–819, 1995.

[VLO97] M. Venditelli, J.P. Laumond, and G. Oriolo. Nilpotent approximation of nonholonomic systems with singularities: A case study. Technical report 97441, LAAS, Toulouse, France, July 1997.

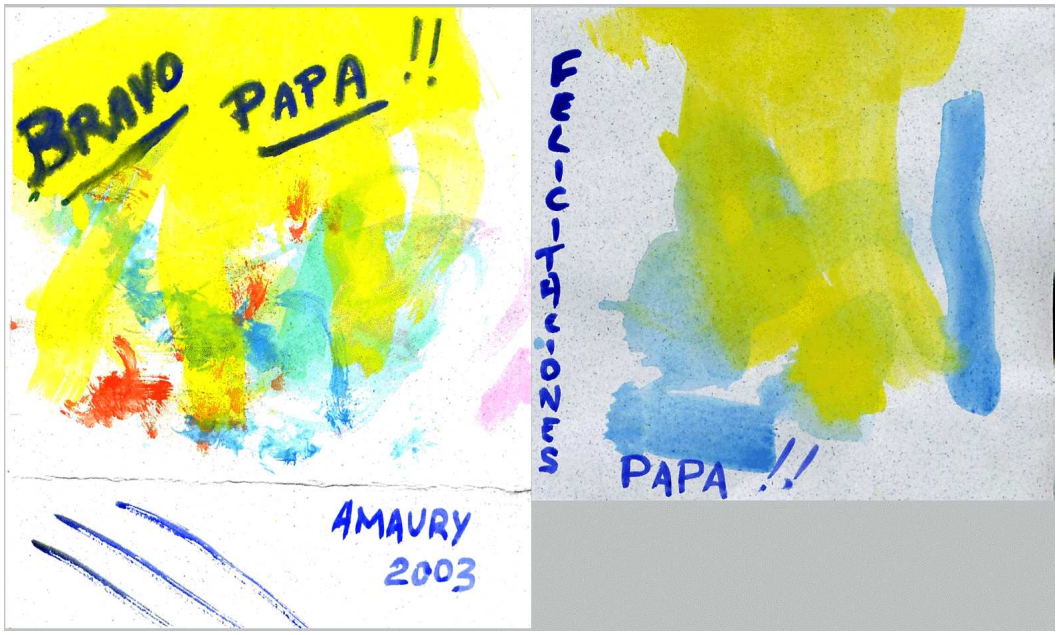
[Wan88] C. Ming Wang. Location estimation and uncertainty analysis for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA (US), April 1988.

[W.F90] Shadwick W.F. Absolute equivalence and dynamic feedback linearization. *Systems and Control Letters*, 15:35–39, 1990.

[Whi90] J.C. Whitehead. Rear wheel steering dynamics compared to front steering. *Journal of Dynamic Systems, Measurement and Control*, vol.112:88–93, March 1990.

[WQ01] Danwei Wang and Feng Qi. Trajectory planning for a four-wheel-steering vehicle. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3320–3325, Seoul, Korea, 2001.

[WTS+92] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J.P. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1999–2004, Nice (FR), May 1992.


BRAVO PAPA !!
AMAURY
2003
FELICITACIONES
PAPA !!

Résumé

Cette thèse porte sur les problèmes de planification et d'exécution de trajectoires pour une classe de robots de type voiture sans essieu fixe. Nous définissons une voiture *bi-guidable* comme un véhicule capable d'orienter ses roues arrières en fonction de l'angle de braquage avant. Les équations différentielles décrivant un tel système posent de nouveaux problèmes de commande en robotique mobile. Nous abordons ces problèmes dans le cadre de la *platitude différentielle*. Un système est dit *plat* si son comportement peut être complètement décrit par un ensemble de variables différentiellement indépendantes, appelé *sortie plate* ou *sortie linéarisante*, elles-mêmes fonction des variables du système et de leurs dérivées : toute trajectoire du système peut s'obtenir à partir de cet ensemble de fonctions sans intégrer d'équations différentielles. Pour ces systèmes, il existe dans la littérature des solutions efficaces profitant des propriétés de la sortie plate. Afin d'exploiter ces solutions, cette dissertation établi d'abord l'existence d'une sortie linéarisante pour la voiture bi-guidable. La difficulté majeure, question ouverte dans le cas général, est de transformer ce résultat d'existence en un résultat effectif permettant de construire la sortie plate. Nous proposons une démarche systématique dans le cadre de la théorie des systèmes de Pfaff, aidant au calcul d'une sortie linéarisante pour un système vérifiant les conditions d'Engel. Nous appliquons cette démarche à la voiture bi-guidable afin de résoudre le problème de planification non-holonome. Nous nous intéressons ensuite au suivi de trajectoires ou la commande en boucle fermée. Dans ce contexte, la platitude implique la linéarisation par bouclage dynamique. Nous établissons des résultats permettant de calculer ce bouclage pour la voiture bi-guidable. Nous synthétisons une loi de commande pour systèmes linéaires afin de résoudre le problème d'exécution des trajectoires. Enfin, nous montrons la validité de ces résultats théoriques par des expérimentations menées sur un véhicule électrique réel à double essieu orientable. Cette thèse contient un résumé étendu en français, le contenu principal étant en anglais.

Mots clés :
Planification non-holonome, robotique mobile, véhicule à deux essieux commandables, platitude différentielle, bouclage endogène.