MOTION PLANNING FOR WHEELED NONHOLONOMIC SYSTEMS

by

Linda Grace Bushnell

Memorandum No. UCB/ERL M94/67

12 September 1994

MOTION PLANNING FOR WHEELED NONHOLONOMIC SYSTEMS

by

Linda Grace Bushnell

Memorandum No. UCB/ERL M94/67

12 September 1994

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

MOTION PLANNING FOR WHEELED NONHOLONOMIC SYSTEMS

by

Linda Grace Bushnell

Memorandum No. UCB/ERL M94/67

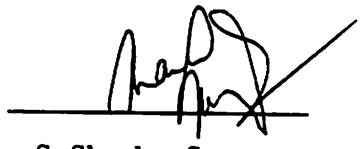
12 September 1994

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Motion Planning for Wheeled Nonholonomic Systems
by
Linda Grace Bushnell
Doctor of Philosophy in Engineering-Electrical Engineering and Computer Sciences
University of California at Berkeley
Professor S. Shankar Sastry, Chair



S. Shankar Sastry
Chair

This dissertation solves important cases of the motion planning problem for wheeled nonholonomic systems. Given initial and final positions and orientations of a mobile robot in its environment workspace, the problem is to generate a path specifying a continuous sequence of positions and orientations that do not collide with the workspace obstacles and to generate the control inputs needed to steer the robot along this path.

The two dual methods of geometric nonlinear control theory and exterior differential systems for transforming kinematic models of wheeled nonholonomic systems with two or more inputs into *chained form* or *Goursat normal form* are presented. Conversion to chained form using vector field methods only gives sufficient conditions, but is easy to apply. Conversion to Goursat normal form gives necessary and sufficient conditions, but requires using subtleties of exterior differential systems. Once the system is in chained form or Goursat normal form, various open-loop, point-to-point steering methods can easily be constructed to steer the mobile robot between any two given configurations. Algorithms are given for steering with sinusoidal, polynomial and piecewise constant control inputs. The examples used to illustrate the theory include a fire truck, or tiller truck, and a multiple-steering, multiple-trailer mobile robot. These systems are drift-free and the nonholonomic behavior comes from non-slipping constraints on the wheels.

For a mobile robot configured as a car pulling trailers connected by off-axle hitches,

an upper bound is computed on the maximal distance that the trailers and kingpin hitches swing off the lead car's path when the car changes from a straight line to an arc of a circle, or vice versa. The trailers are shown to exponentially converge to their steady-state circular paths when the lead car is moving on a circular path. If the turning radius of the lead car is upper bounded, then a reduced visibility graph method is proposed to find a collision-free path. Otherwise, path planners from the literature for a car-like mobile robot are modified. The methodology presented in this dissertation guarantees that the trailers do not collide with the obstacles for forward motions of the lead car.

This dissertation is dedicated with loving memories to my sister

Victoria Anne Bushnell

and with loving thoughts to my husband

Gandis Gediminas Mažeika

Contents

List of Figures

Acknowledgements

First and foremost, I would like to thank my research advisor and mentor, Shankar Sastry for his guidance and motivation during my study of nonholonomic systems. It was due to Shankar that I was able to attend many conferences and meet many interesting people. His enthusiasm for research has inspired me to continue working in the control systems area.

I am grateful to my two other dissertation committee members: John Canny, who gave precise comments on the path planning chapter in this dissertation and Jerry Marsden, who not only thoroughly read this dissertation, but also was interested in my future research plans. I would also like to thank Felix Wu for being on my qualifying exam committee.

Thanks to all of my co-workers for making the Robotics Lab a fun place to do research, especially my nonholonomic research partners, Dawn Tilbury and Greg Walsh, who were and are outstanding roll models for me; Charles Coleman, Neil Getz, Datta Godbole, Tony Lindsey, Brian Mirtich and Jeff Wendlandt, who, with their enthusiasm for science, were always willing to discuss interesting problems; and Shahram Shahruz, who encouraged me to pursue my first "invited session" at a major control systems conference. The efforts of Anant Sahai and Matthew Secor, who worked on the interactive simulation project used in Chapter 6 of this dissertation, were greatly appreciated.

Thanks to Richard Murray for many inspiring discussions on this research and to Jean-Paul Laumond for suggesting some nonholonomic path planners to look at. I would like to acknowledge Captain David Orth, tillerman George Fisher and driver Bob Humphrey at Berkeley Fire Department Station No. 2 for their demonstration of parallel parking and backing up their fire truck.

I greatly benefited from my summer internships at at IBM Yorktown Heights, NY and at PG&E Research in San Ramon, CA. I would like to thank Ralph Hollis and Paul Mauldin for these experiences.

I would like to thank Sanjoy Mitter and Kathleen O'Sullivan for my productive stay at MIT as a visiting scholar during the Fall semester of 1992. It was during this visit that I was able to spend quality time with my sister during her last months.

Heartfelt thanks to my parents, David and Nijolė Bushnell, who, through their constant love and guidance during my thirteen-year undergraduate and graduate careers, helped me achieve my goal.

Most of all, I would like to thank my husband, Gandis Mažeika, who, after four years of driving back and forth between UC Davis and Berkeley, said he "drove around the world to woo me." *Aš tave myliu, mano saldusis.*

Without a doubt, this research could not have been accomplished without the generous financial support from NSF under grant IRI 90-14490 and ARO under grant DAAL 03-91-G0191.

Chapter 1

Introduction

This dissertation solves some problems of motion planning for wheeled nonholonomic systems. Specifically, several control inputs are designed to steer systems such as a car, a car pulling trailers, a fire truck, and a multiple-steering, multiple-trailer vehicle between two given configurations in a given amount of time. For some systems, obstacles may be taken into account in planning the path. Even though only the kinematic models of the systems, i.e., without mass and forces are considered, the problem is interesting since the number of control inputs is less than the dimension of the state space of the system. In addition to wheeled nonholonomic systems, a satellite [82], a falling cat [21], a hopping robot in flight (hopping gymnast) [10, 33], and multiple-fingered robotic hands [55] are all examples of nonholonomic systems. A system is called *nonholonomic* if either rolling constraints are imposed on the system from bodies in contact with each other that roll without slipping, or conservation laws (dynamic constraints) are observed on the system when angular momentum is conserved. Mobile robots and multiple-fingered robotic hands have linear velocity constraints that require the wheels or the fingers to roll without slipping. The satellite, falling cat, and the hopping gymnast conserve angular momentum when they move in space.

This research extends two dual methods of analysis for nonholonomic systems with two inputs to systems with greater than two inputs. The approaches are geometric nonlinear control theory and the theory of exterior differential systems, and each has its own advantages. Geometric nonlinear control was used in the development of a class of systems, called *chained form*, which can be easily steered using sinusoidal inputs. Murray and Sastry [57] presented sufficient conditions for transforming kinematic equations of

nonholonomic systems with two inputs (as for example, the car, which has the two inputs of driving and steering) to a chained form with one chain. Extending the same two-input problem, Sørdalen [68] and Tilbury *et al.* [76] were able to convert a car pulling N trailers to chained form.

In geometric nonlinear control theory, nonholonomic systems are considered not from the point of view of their constraints, but rather from the point of view of a control system with the allowable motions in the span of the input vector fields. The dual to the chained form is the Goursat normal form. Although the mathematics literature abounds with the theory of exterior differential systems (see [15, 25, 87] as a survey), only recently have there been attempts to apply this machinery to solve general control problems for steering nonholonomic systems. On the one hand, conversion to chained form using vector field methods only gives sufficient conditions, but is easy to apply; it is analogous to the method for exact linearization of a nonlinear system [28]. Conversion to Goursat normal form, on the other hand, gives necessary and sufficient conditions, but requires using subtleties of exterior differential systems. We found it beneficial to look at a system from both the vector field and exterior differential systems points of view.

The unique contributions of this dissertation are the presentation of necessary and sufficient conditions for converting a nonholonomic system with two or more inputs into Goursat normal form, sufficient conditions for transforming these systems to chained form, and steering methods to control chained form systems. A three-input system (the fire truck, or tiller truck) and a multiple-input system (multiple-steering, multiple-trailer vehicle) are introduced as examples of nonholonomic systems that can be converted and controlled in chained form. The steering algorithms presented construct sinusoidal, polynomial, and piecewise constant control inputs. In addition, numerical simulation results are given that show how extra steering wheels result in greater maneuverability.

Minimizing the distance between any two configurations (as in [62] for a car) is difficult to define because there may be more than one trajectory or vehicle to consider. In the multiple-steering, multiple-trailer vehicles, for example, we may want to optimize the trajectories of all the trailers, not just the lead car. The input effort (maximum input value), defined as some weighted combination of the driving velocity and the steering velocities of the system, could also be minimized. The methods presented in this dissertation, namely the sinusoidal, polynomial, and piecewise constant methods, however, are optimal in the sense that they minimize the number of reversals in the trajectories. We have found that for

a parallel parking type of trajectory, sinusoidal inputs work well since the vehicle makes one reversal during the entire trajectory. Using polynomial inputs for this trajectory, however, requires dividing the path into two parts and treating the reverse part of the maneuver as a second trajectory. Generally, the polynomial and piecewise constant methods work well when there are no reversals in the trajectory.

Another unique contribution of this dissertation is the presentation of a path planning algorithm for a car pulling trailers connected by off-axle hitches. If the lead car's turning radius is upper bounded by the radius of an "enlarged" circular robot superimposed on the car, then a visibility graph algorithm is given to plan a collision-free path. If the turning radius is not constrained, then an alternate algorithm that modifies existing obstacle avoidance planners that use a configuration space approach to plan a path for a single car to be planners for the multiple-trailer vehicle is proposed. Both algorithms use the additional restriction that the lengths of the links in the kingpin hitching are all equal. The key difference between designing path planners for single cars and multiple-trailer vehicles relies on defining an off-tracking bound, which is the maximal distance that the trailers and kingpin hitches deviate from the lead car's track when the car changes from one path to another. In addition, the trailers are shown to exponentially converge to their steady-state circular paths when the lead car is moving on a circular path. The methodology presented in this dissertation guarantees that the trailers do not collide with the obstacles for forward motions of the lead car.

There are many applications of this research for path planning of wheeled nonholonomic systems. One such application is for training drivers of trucks with multiple trailers that have off-axle hitching. Using this research, we suggest attaching "whiskers" to the truck to notify the driver how close she can drive near the obstacles. The length of the whiskers would be directly proportional to how many trailers the vehicle has. Then, if the driver avoids hitting obstacles with the whiskers, the trailers will also avoid those obstacles. Another application of this research is to help in steering multiple-trailer vehicles in manufacturing plants, nuclear power plants or any area unsafe for human operators or made up of narrow passageways. The goal is to have fully automated vehicles in these areas or to assist an operator by steering other axles in the vehicle. The extra steering wheels give greater maneuverability in the narrow, winding passageways.

The outline of this dissertation is as follows. Chapter 2 gives the mathematical preliminaries for both geometric nonlinear control theory and exterior differential systems.

Chapter 3 presents the methods for transforming nonholonomic systems with two or more inputs into a multiple-input chained form and extended Goursat normal form. The controllability of chained form systems is also discussed. Chapter 4 presents different steering methods for the system in chained form and stabilization issues. Chapter 5 gives examples to illustrate the theory. The fire truck and a multiple-steering, multiple-trailer system are the main examples presented. The systems are transformed into chained form or Goursat normal form and numerical simulations are performed. The simulation results are presented for different trajectories such as parallel parking, turning left and right corners, and changing lanes. The fire truck's performance is compared to a similar vehicle without the tiller steering. Chapter 6 presents an obstacle avoidance algorithm for a car pulling many trailers. We find that a collision-free path need only be planned for an "enlarged" front car and the trailers will avoid the same obstacles. Chapter 7 presents open problems in the area of nonholonomic path planning. Chapter 8 summarizes the results presented in this dissertation.

Chapter 2

Mathematical Preliminaries

In this chapter, tutorials on geometric nonlinear control theory and exterior differential systems will be presented. These will be referred to in the next chapter when the two dual methods for converting a multiple-input nonholonomic system to either chained form or Goursat normal form are discussed.

2.1 A Tutorial for Geometric Nonlinear Control Theory

In this section, a variety of results from geometric nonlinear control theory, differential geometry and introductory nonholonomic systems will be presented, covering the Frobenius Theorem, the concept of controllability, and exact linearization, which will all prove useful when analyzing nonholonomic system. To minimize the mathematical prerequisites, all calculations are performed in $\mathbb{R}^n$. All of the essential ideas, however, are covariant and thus carry over to the context of manifolds. We suggest Isidori [28] and Nijmeijer and van der Schaft [59] for an introduction to nonlinear control theory, Spivak [70] for an introduction to differential geometry, and Murray, *et al.* [54] for an introduction to nonholonomic motion planning. The presentation of [54] is followed here.

2.1.1 Vector Fields and Flows

The tangent space to $\mathbb{R}^n$ at a point $x \in \mathbb{R}^n$ is denoted as $T_x\mathbb{R}^n$. A *vector field* $f : \mathbb{R}^n \to T_x\mathbb{R}^n$ defined on an open set $U \subset \mathbb{R}^n$ is a smooth map, assigning to each point $x = (x_1, \ldots, x_n) \in U$ the n-dimensional tangent vector $f(x) \in T_x\mathbb{R}^n$. The map f is said to be *smooth*, or $C^\infty(\mathbb{R}^n)$, if its partial derivatives of any order with respect to $(x_1, \ldots, x_n)$

exist and are continuous, *analytic* if it is smooth and for each point $x_0 \in \mathbb{R}^n$ there exists a neighborhood U of x_0 such that the Taylor series expansion of f at x_0 converges to $f(x)$ for all $x \in U$, and *meromorphic* if it is a ratio of analytic functions. The map f is represented in coordinates as a column vector

$$f = \begin{pmatrix} f^1(x) \\ \vdots \\ f^n(x) \end{pmatrix}$$

or if $x_1, \ldots, x_n$ are local coordinates for $\mathbb{R}^n$, f is written as

$$f(x) = f^1(x)\frac{\partial}{\partial x_1} + \cdots + f^n(x)\frac{\partial}{\partial x_n} ,$$

where the symbol $\frac{\partial}{\partial x_i}$ is the i^{th} basis element for $T_x\mathbb{R}^n$ with respect to a given set of coordinates. A vector field is smooth if each $f^i(x)$ is smooth. Alternatively, a vector field can be thought of as the right hand side of a differential equation

$$\dot{x} = f(x) . \tag{2.1}$$

The symbol $\frac{\partial}{\partial x_i}$ reminds us that vector fields act by differentiation in the sense that the derivative of a smooth function $h : \mathbb{R}^n \to \mathbb{R}$ along a vector field g is denoted as the mapping $L_g : \mathbb{R} \to \mathbb{R}$,

$$L_g h(x) = \frac{\partial h}{\partial x} g(x) .$$

This is also called the *Lie derivative* of h along g.

The *flow* of a vector field is defined to be the solution of the differential equation (2.1). That is, $\phi_t^f : \mathbb{R}^n \to \mathbb{R}^n$ is the state of the differential equation at time t starting from $x \in \mathbb{R}^n$ at time $t = 0$. It therefore satisfies the differential equation

$$\frac{d}{dt}\phi_t^f(x) = f(\phi_t^f(x)) .$$

A vector field is said to be *complete* if its flow is defined for all t. From the existence and uniqueness theorem of ordinary differential equations, ϕ_t^f is a local diffeomorphism (or, global on its domain of definition) of $\mathbb{R}^n$ onto itself for each fixed t. It also satisfies the following group property

$$\phi_t^f \circ \phi_s^f = \phi_{t+s}^f \tag{2.2}$$

for all t, s, where $\circ$ stands for the composition of the two flows.

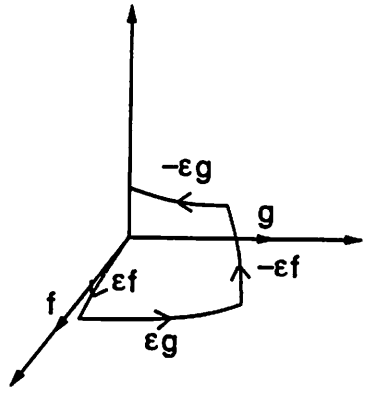


Figure 2.1: A Lie bracket motion resulting from flowing around a square defined by two vector fields.

2.1.2 Lie Brackets and the Frobenius Theorem

Given two vector fields f and g, the composition of the flow of f for t seconds with the flow of g for s seconds is given by the map $\phi_s^g \circ \phi_t^f$. In general, this map is different from the composition in the reverse order $\phi_t^f \circ \phi_s^g$. Figure 2.1 illustrates a Lie bracket motion.

Proposition 1 *The net motion from traveling along f for ϵ seconds, g for ϵ seconds, $-f$ for ϵ seconds, and $-g$ for ϵ seconds starting at the point $x(0) = x_0$ satisfies*

$$\phi_\epsilon^{-g} \circ \phi_\epsilon^{-f} \circ \phi_\epsilon^{g} \circ \phi_\epsilon^{f}(x_0) = x_0 + \epsilon^2[f,g](x_0) + O(\epsilon^3)\ . \tag{2.3}$$

The notation $O(\epsilon^3)$ represents terms of order ϵ^3 and higher. That is, if we take the left hand side of equation (2.3) and divide by ϵ^3, the limit of this expression as ϵ approaches zero is not necessarily zero.

Proof. The Taylor series in ϵ is evaluated for the differential equation

$$\dot{x} = f(x)u_1 + g(x)u_2$$

with the inputs $u := (u_1, u_2)$ being

$$u(t) = \begin{cases} (1,0) & t \in [0,\epsilon) \\ (0,1) & t \in [\epsilon, 2\epsilon) \\ (-1,0) & t \in [2\epsilon, 3\epsilon) \\ (0,-1) & t \in [3\epsilon, 4\epsilon) \end{cases}$$

for $\epsilon > 0$. After the first step,

$$\begin{aligned} x(\epsilon) &= x(0) + \epsilon\dot{x}(0) + \frac{1}{2}\epsilon^2\ddot{x}(0) + O(\epsilon^3) \\ &= x_0 + \epsilon f(x_0) + \epsilon^2 \frac{1}{2}\frac{\partial f}{\partial x}(x_0) f(x_0) + O(\epsilon^3) , \end{aligned}$$

where $O(\epsilon^i)$ represents terms of order ϵ^i and higher. The second step gives

$$\begin{aligned} x(2\epsilon) &= x(\epsilon) + \epsilon g(x(\epsilon)) + \epsilon^2 \frac{1}{2}\frac{\partial g}{\partial x}(x(\epsilon)) g(x(\epsilon)) + O(\epsilon^3) \\ &= x_0 + \epsilon(f(x_0) + g(x_0)) \\ &\quad + \epsilon^2 \left(\frac{1}{2}\frac{\partial f}{\partial x}(x_0) f(x_0) + \frac{\partial g}{\partial x}(x_0) f(x_0) + \frac{1}{2}\frac{\partial g}{\partial x}(x_0) g(x_0) \right) + O(\epsilon^3) , \end{aligned}$$

where we have used the fact that $g(x_0 + \epsilon x_1) = g(x_0) + \epsilon \frac{\partial g}{\partial x}(x_0) x_1 + O(\epsilon^2)$. The third step gives

$$\begin{aligned} x(3\epsilon) &= x(2\epsilon) - \epsilon f(x(2\epsilon)) + \epsilon^2 \frac{1}{2}\frac{\partial f}{\partial x}(x(2\epsilon)) f(x(2\epsilon)) + O(\epsilon^3) \\ &= x_0 + \epsilon g + \epsilon^2 \left(\frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g + \frac{1}{2}\frac{\partial g}{\partial x} g \right) + O(\epsilon^3) , \end{aligned}$$

where all functions are evaluated at x_0. At the last step,

$$\begin{aligned} x(4\epsilon) &= x(3\epsilon) - \epsilon g(x(3\epsilon)) + \epsilon^2 \frac{1}{2}\frac{\partial g}{\partial x}(x(3\epsilon)) g(x(3\epsilon)) + O(\epsilon^3) \\ &= x_0 + \epsilon^2 \left(\frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g \right) + O(\epsilon^3) . \end{aligned}$$

The net motion $x(4\epsilon) - x_0$ is defined to be the *Lie bracket* between the two vector fields f and g up to order ϵ^3; that is,

$$\begin{aligned} [f, g](x) &:= \frac{\partial g}{\partial x} f(x) - \frac{\partial f}{\partial x} g(x) \\ \mathrm{ad}_f^k g(x) &:= [f(x), \mathrm{ad}_f^{k-1} g(x)] , \end{aligned}$$

where $\mathrm{ad}_f^0 g(x) := g(x)$.

□

If the two vector fields commute, i.e., $[f, g] = 0$, then there is no net motion. In fact, in this case $\phi_t^f \circ \phi_s^g = \phi_s^g \circ \phi_t^f$. For $s = t$, the calculation follows easily from the first two steps of the proof of Proposition 1 and the assumption that $\frac{\partial g}{\partial x} f(x) = \frac{\partial f}{\partial x} g(x)$. In general, we have the following proposition (see [1] page 282 for a different proof that treats vector fields as differential operators).

Proposition 2 *Let $x \in \mathbb{R}^n$ and vector fields f and g be defined on $\mathbb{R}^n$ with flows ϕ and ψ, respectively. Then $[f,g] = 0$ if and only if $\phi_t \circ \psi_s = \psi_s \circ \phi_t$.*

Proof.[84] We first define the mapping $h : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \mapsto \mathbb{R}^n$ by $h(s,t,x) := \phi_t^{-1} \circ \psi_s \circ \phi_t \circ x$, and show that the following are equivalent:

(i) $\phi_t \circ \psi_s = \psi_s \circ \phi_t$

(ii) $h(s,t,x) = \psi_s$

(iii) $\frac{\partial}{\partial t} h = 0$ for all s and x

(iv) $\frac{\partial}{\partial s}\frac{\partial}{\partial t} h = 0$

(v) $[f,g] = 0$.

The first equivalence between (i) and (ii) is clear by the definition of h.

If (iii) is true, then h is not a function of t. Thus, $h(s,t,x) = h(s,0,x) = \phi_0^{-1} \circ \psi_s \circ \phi_0 \circ x = \psi_s \circ x$ since ϕ_t at $t = 0$ is the identity map. Conversely, if (ii) is true, then $\frac{\partial}{\partial t} h = 0$ since ψ_s is not a function of t.

By composition of flows,

$$\begin{aligned} h(s,t,x) &= \phi_t^{-1}(t, \psi_s(s, \phi_t(t,x))) \\ \left.\frac{\partial h(s,t,x)}{\partial t}\right|_{t=0} &= D_1\phi_t^{-1}|_{t=0} + D_2\phi_t^{-1} D_2\psi_s D_1\phi_t|_{t=0} \ , \end{aligned} \tag{2.4}$$

where D_i represents the derivative with respect to the i^{th} argument of the function. If (iii) is true, then clearly (iv) is true. If (iv) is true, then $\frac{\partial}{\partial t} h$ would not be a function of s. Equation (2.4), however, shows that it is a function of s. Therefore, the only way for (iv) to be true is if $\frac{\partial}{\partial t} h = 0$ for all s.

Taking the derivative of this expression with respect to s gives

$$\left.\frac{\partial}{\partial s}\frac{\partial h(s,t,x)}{\partial t}\right|_{t=0} = \left.\frac{\partial}{\partial s}\left(D_1\phi_t^{-1} + D_2\phi_t^{-1} D_2\psi_s D_1\phi_t\right)\right|_{t=0} . \tag{2.5}$$

$D_2\phi_t^{-1}$ at $t = 0$ is the identity matrix since $D_2\phi_t$ at $t = 0$ is the identity matrix. Also, $D_1\phi_t = \frac{d}{dt}\varphi_t = f$ by the definition of the flow of f. Furthermore, when $s = 0$, $D_1\phi_t^{-1}(t, \psi_s(s,x))|_{t=0} = -f(0, \psi_s(s,x))$; therefore $\frac{\partial}{\partial s} D_1\phi_t^{-1}(t, \psi_s(s,x))|_{t=0} = -D_2 f D_1\psi_s = -Df\ g$. Similarly, we find $D_2\psi_s = Dg$. Combining this information with equation (2.5) gives

$$\left.\frac{\partial}{\partial s}\frac{\partial h(s,t,x)}{\partial t}\right|_{t=0} = -Df\ g + Dg\ f = [f,g] \ ,$$

showing that (iv) and (v) are equivalent. □

The Lie bracket has three basic properties (see [28] for the proof).

Proposition 3 (Properties of the Lie Bracket) *Consider the vector fields f, g, h defined on $\mathbb{R}^n$ and real numbers r_1, r_2. The Lie bracket satisfies the following properties:*

(i) $[f, g] = -[g, f]$ *(skew commutativity)*

(ii) $[f, [g, h]] + [g, [h, f]] + [h, [f, g]] = 0$ *(Jacobi identity)*

(iii) $[r_1 f + r_2 h, g] = r_1[f, g] + r_2[h, g]$ *and* $[f, r_1 g + r_2 h] = r_1[f, g] + r_2[f, h]$ *(bilinearity over* $\mathbb{R}$*).*

Note that a *Lie product* is defined as a nested set of Lie brackets. In addition, the following properties of the Lie bracket will be useful. Given vector fields f, g and a real-valued smooth functions α, β, λ all defined on an open set $U \subset \mathbb{R}^n$,

$$[\alpha f, \beta g](x) = \alpha(x)\beta(x)[f, g] + \alpha(x)(L_f\beta(x))g(x) - \beta(x)(L_g\alpha(x))f(x) \quad (2.6)$$

$$L_{[f,g]}\lambda(x) = L_f L_g \lambda(x) - L_g L_f \lambda(x) \quad (2.7)$$

for $x \in \mathbb{R}^n$.

A *distribution* Δ smoothly assigns a subspace of $T\mathbb{R}^n$ to each point in $\mathbb{R}^n$. Using a basis of smooth vector fields $f_1, \ldots, f_d$, a distribution is defined as

$$\Delta = \text{span}\{f_1, \ldots, f_d\} .$$

Δ assigns to each point $x \in U \subset \mathbb{R}^n$ a subspace spanned by the values at x of the smooth vector fields f_i defined on U. At any point $x \in U$, Δ is a linear subspace of the tangent space

$$\Delta(x) = \text{span}\{f_1(x), \ldots, f_d(x)\} \subset T_x\mathbb{R}^n .$$

The vector fields $f_1, \ldots, f_d$ are called the *local generators* of Δ, since any vector field $f \in \Delta$ can be written as

$$f(q) = \sum_{i=1}^{d} a_i(q) f_i(q)$$

for q in a neighborhood of x, where a_i are smooth functions. Thus, the distribution is a fundamental object. If the basis vector fields $f_1, \ldots, f_d$ are thought of as the right hand side of the differential equation

$$\dot{x}(t) = f_1(x)u_1(t)a + \cdots + f_d(x)u_d(t) ,$$

then Δ is the subspace of allowable velocities of this system.

A distribution, Δ, is said to be *involutive* if it is closed under the Lie bracket operator:

$$f, g \in \Delta \Longrightarrow [f, g] \in \Delta \ .$$

The *involutive closure* of Δ, denoted as $\bar{\Delta}$, is the smallest distribution containing Δ such that

$$f, g \in \bar{\Delta} \Longrightarrow [f, g] \in \bar{\Delta} \ .$$

A distribution is said to be *regular* or *nonsingular* if the dimension of the subspace $\Delta(x)$ does not vary with x. A regular distribution $\Delta(x) = \text{span}\{f_1(x), \ldots, f_d(x)\}$, defined on an open set U of $\mathbb{R}^n$, is said to be *completely integrable* if for each point x_0 of U there exist a neighborhood U^o of x_0 and $n - d$ real-valued smooth functions $h_1, \ldots, h_{n-d}$ such that

$$dh_i \cdot f_j(x) = 0 \ , \quad f_j \in \Delta \quad 1 \leq i \leq n - d$$

for all $x \in U^o$.

Theorem 4 (Frobenius) *A nonsingular distribution is completely integrable if and only if it is involutive.*

See [28] for a proof.

In analogy to defining vector fields on $T_x\mathbb{R}^n$, their dual can be defined on $T_x^*\mathbb{R}^n$ (the set of linear functions taking values in $T_x\mathbb{R}^n$): for each $x \in \mathbb{R}^n$, define $\omega(x) \in T_x^*\mathbb{R}^n$ to be a *one-form*. In local coordinates $(x_1, \ldots, x_n)$ for $\mathbb{R}^n$, a smooth one-form is written as

$$\omega(x) = \omega_1(x)dx_1 + \cdots + \omega_n(x)dx_n \ ,$$

where each ω_i is smooth. The symbols dx_i represent the dual basis and are defined as

$$dx_i \left(\frac{\partial}{\partial x_j} \right) = \delta_{ij} \ ,$$

where δ_{ij} is the Kronecker delta function, which takes values 1 for $i = j$ and 0 for $i \neq j$. A one-form acts on a vector field to give a real-valued function on $\mathbb{R}^n$:

$$\omega \cdot f = \left(\sum_{i=1}^{n} \omega_i dx_i \right) \cdot \left(\sum_{j=1}^{n} f^j \frac{\partial}{\partial x_j} \right) = \sum_{i=1}^{n} \omega_i f^i \ .$$

A distribution spanned by linearly independent one-forms

$$\Omega(x) := \text{span}\{\omega^1(x), \ldots, \omega^k(x)\} \tag{2.8}$$

smoothly assigns a subspace of $T_x^*\mathbb{R}$ to each $x \in \mathbb{R}^n$. In analogy to vector fields, any one-form $\omega \in \Omega$ can be written as

$$\omega(q) = \sum_{i=1}^{k} a^i(q)\omega^i(q)$$

for q in a neighborhood of x, where a^i are smooth functions of q. Thus, Ω is a unique subspace of $T_x^*\mathbb{R}^n$, while the one-forms ω^i are a non-unique basis of Ω.

2.1.3 An Introduction to Nonholonomic Systems

A *nonholonomic* constraint on a system is a non-integrable kinematic constraint. For wheeled systems, these constraints express that the relative velocity between the contact point on the rolling wheel and the ground is zero. A system is said to be nonholonomic if its motions are constrained by nonholonomic constraints.

Let a configuration of the robot system be represented by n parameters, or coordinates, $(x_1, \ldots, x_n)$ where n is the dimension of the configuration space. For example, the configuration of a two-axle kinematic car could be represented by (x, y, θ, ϕ), where x and y are the Cartesian coordinates of the center of the rear axle, θ is the body orientation, and ϕ is the angle of the front (steerable) wheels. The configuration space for this car is $\mathbb{R}^2 \times S^1 \times S^1$, where S^1 denotes the unit circle. In the following, we assume, without loss of generality, that the configuration space is $\mathbb{R}^n$ and follow the presentation in [8]. Suppose the motion of the general system is constrained by a scalar constraint of the form

$$F(x, t) = 0\ , \tag{2.9}$$

where F is a smooth function with a non-zero derivative. This equation can be used to solve for one of the x_i in terms of the other x_j, $j \neq i$, thereby reducing the dimension of the configuration space to $n - 1$. This constraint (2.9) is called a ***holonomic equality constraint*** of the system. In general, if there are k independent constraints of the form (2.9), the configuration space can be reduced to be $(n - k)$-dimensional. There can also be holonomic inequality constraints of the form $F(x, t) < 0$ or ≤ 0.

Now suppose the motion of the system is constrained by the scalar constraint

$$G(x, \dot{x}, t) = 0 \ , \tag{2.10}$$

where $\dot{x} \in T_x\mathbb{R}^n$, the space of velocities of the system. The constraint (2.10) is holonomic if it is integrable, that is, if $\dot{x}$ appears linearly in equation (2.10) so that the constraint has the form $\frac{d}{dt}F(x,t) = 0$, which can then be integrated to give equation (2.9). If (2.10) is not integrable, it is said to be a *nonholonomic equality constraint*, restricting the space of velocities achievable by the system at any configuration to an $(n-1)$-dimensional linear subspace of $T_x\mathbb{R}^n$ without affecting the dimension of the configuration space $\mathbb{R}^n$. In general, if there are k independent constraints of the form (2.10), the space of achievable velocities is reduced to an $(n-k)$-dimensional subspace of $T_x\mathbb{R}^n$. As above, there can also be nonholonomic inequality constraints of the form $G(x, \dot{x}, t) < 0$ or ≤ 0.

There are two types of motions caused by two rigid bodies in contact: rolling and sliding. For rolling with sliding motion, the nonholonomic constraint (2.10) is a nonlinear expression in $\dot{x}$. For rolling without sliding, the nonholonomic constraint is linear in $\dot{x}$. The latter is assumed throughout this dissertation.

One of the first tasks in analyzing nonholonomic systems is to convert them into control systems. The kinematic constraints of the nonholonomic system with state space $x \in \mathbb{R}^n$ are written as

$$\omega^i \cdot \dot{x} = 0 \qquad i = 1, \ldots, k \qquad k \leq n$$

where the one-forms ω^i are written as

$$\omega^i(x) = \omega^i_1(x)dx_1 + \cdots + \omega^i_n(x)dx_n \ .$$

The one-form dx_i represents the basis dual to the basis $\frac{\partial}{\partial x_i}$ on $T_x\mathbb{R}^n$.

For a given distribution $\Delta = \text{span}\{g_0, \ldots, g_{n-k-1}\}$, its corresponding *codistribution* $\Omega = \text{span}\{\omega^1, \ldots, \omega^k\}$ is defined to be the subspace of $T^*_x\mathbb{R}^n$ that is the annihilator of Δ, i.e., Ω is the set of all one-forms that vanish on Δ:

$$\omega^i \cdot g_j = 0 \quad i = 1, \ldots, k, \quad j = 0, \ldots, n-k-1 \ . \tag{2.11}$$

Alternatively, given a set of nonholonomic constraints $\Omega = \text{span}\{\omega^1, \ldots, \omega^k\}$ for a system, there exist vector fields g_j, $j = 0, \ldots, n-k-1$, that annihilate the one-forms ω^i as in

equation (2.11) such that the g_j are smooth and linearly independent over the ring of smooth functions. We say that $\Delta = \Omega^{\perp}$.

The control system arising from the given kinematic constraints is thus written as

$$\dot{x} = g_0(x)u_0 + \cdots + g_{n-k-1}(x)u_{n-k-1} \ ,$$

where the u_j are the freely specified *control inputs* to the system, and the g_j are called the *input vector fields.*

2.1.4 Nonlinear Controllability

From the above discussion, we can now treat the nonholonomic motion planning problem as a control problem and concentrate on the controllability issues. We further restrict our consideration to control systems without drift (when all the control inputs are set to zero, the system does not drift):

$$\dot{x} = g_0(x)u_0 + \cdots + g_m(x)u_m \tag{2.12}$$

with x in an open set $U \subset \mathbb{R}^n$, $u \in \mathbb{R}^{m+1}$, $m+1 \leq n$ and g_j smooth, linearly independent vector fields defined on U. It is also assumed that the g_j are complete (flows defined for all time). This system is said to be *controllable* if for any $x_0, x_f \in U$, there exists a $T > 0$ and an input $u : [0,T] \to \mathbb{R}^{m+1}$ such that $x(0) = x_0$ and $x(T) = x_f$. In the following, the formulation of Hermann and Krener [26] is applied to drift-free systems (2.12).

The system is said to be *small-time locally controllable at* x_0 if nearby points can be reached in arbitrarily small amounts of time and the system stays near x_0 at all times. Given x_0 and x_f in an open set $V \subseteq \mathbb{R}^n$, define

$$\begin{aligned} \mathcal{R}^V(x_0, T) \ := \ & \{x \in \mathbb{R}^n | \text{ there exists } u : [0,T] \to \mathbb{R}^{m+1} \text{ that steers (2.12) from} \\ & x(0) = x_0 \text{ to } x(T) = x_f \text{ and satisfies } x(t) \in V \text{ for } 0 \leq t \leq T\} \ , \end{aligned}$$

which is the set of states that are reachable from x_0 in time T that remain in V, and

$$\mathcal{R}^V(x_0, \leq T) := \bigcup_{0<\tau\leq T} \mathcal{R}^V(x_0, \tau) \ .$$

A control system is said to be small-time locally controllable, or just *locally controllable*, if $\mathcal{R}^V(x_0, \leq T)$ contains a neighborhood of x_0 for all neighborhoods V of x_0 and $T > 0$.

Let $\Delta := \text{span}\{g_0, \ldots, g_m\}$ be the input distribution associated with the control system (2.12) and recall that $\bar{\Delta}$ is the involutive closure of Δ. $\bar{\Delta}$ is referred to as the *controllability distribution.* Controllability of the system (2.12) can be characterized in terms of the Lie algebra generated by the smooth, linearly independent input vector fields g_j. The conditions for local controllability are given by the following theorem [18].

Theorem 5 (Chow) *The distribution $\bar{\Delta}_x = T_x\mathbb{R}^n$ for all $x \in U \Rightarrow$ the interior of the set $\mathcal{R}^V(x_0, \leq T)$ is not empty $\Longleftrightarrow$ the system (2.12) is locally controllable on U.*

See [54] for a proof. Chow's Theorem states that if the system (2.12) can be steered in every direction using Lie bracket motions of any order, then it is controllable. In practice, local controllability can be checked by checking the rank of the controllability distribution $\bar{\Delta}$. This is referred to as the *controllability rank condition.* Chow's Theorem is a non-constructive procedure for generating trajectories for the control system (2.12). A constructive method is the subject of Chapters 3 and 4 of this dissertation.

2.1.5 Exact Linearization

One of the main topics in this dissertation is the conversion of the control system (2.12) to a canonical chained form, which is easy to steer. One method used for this constructive transformation resembles that for linearizing a nonlinear system. Consider the following nonlinear system with drift

$$\begin{aligned} \dot{x} &= f(x) + g_1(x)u_1 + \cdots + g_m(x)u_m \\ y_i &= h_i(x) \qquad i = 1, \ldots, m\,, \end{aligned} \tag{2.13}$$

with x in an open set $U \subset \mathbb{R}^n$, $u \in \mathbb{R}^m$, g_j are smooth, linearly independent vector fields defined on U, and h_i are smooth functions defined on U, and define the distributions

$$\begin{aligned} G_0 &:= \text{span}\{g_1, \ldots, g_m\} \\ G_i &:= \text{span}\{\text{ad}_f^k g_j : 1 \leq k \leq i, 1 \leq j \leq m\} \quad \text{for} \quad 1 \leq i \leq n-1\,. \end{aligned} \tag{2.14}$$

A multiple-input, multiple-output (MIMO) system (2.13) is said to have *vector relative degree* $\gamma_1, \ldots, \gamma_m$ at x_0 if

(i) $L_{g_j} L_f^k h_i(x) \equiv 0$ for all $1 \leq j \leq m$, $0 \leq k < \gamma_i - 1$, $1 \leq i \leq m$ and for all x in a neighborhood of x_0

(ii) the $m \times m$ matrix $A(x)$ with elements $[L_{g_j} L_f^{\gamma_i - 1} h_i]_{i,j}$ is nonsingular at $x = x_0$.

Assuming that the system (2.13) has vector relative degree $\gamma_1 + \cdots + \gamma_m = n$ at x_0 with respect to the system outputs $h_1(x), \ldots, h_m(x)$, the following theorem can be stated that gives the necessary and sufficient conditions for transforming a nonlinear system with drift to a linear form via state feedback and a coordinate transformation. The theorem holds within a neighborhood of x_0 (this is in contrast to Jacobian linearization, which linearizes around a point) and the size of the neighborhood is unspecified.

Theorem 6 (Exact Linearization via Feedback) *Given an initial state x_0 for the above system (2.13) with drift $f(x)$ and $g(x) := [g_1(x) \cdots g_m(x)]$ having full rank at $x = x_0$, then there exists a neighborhood U of x_0, a feedback $u = \alpha(x) + \beta(x)v$, and a coordinate transformation $z = \Phi(x)$ defined on U such that*

$$\begin{aligned} \dot{z} &= Az + Bv \\ y &= Cz \end{aligned} \tag{2.15}$$

if and only if the following conditions are satisfied:

(i) the distribution G_i has constant dimension near x_0 for $0 \leq i \leq n-1$

(ii) the distribution G_{n-1} has dimension n

(iii) the distribution G_i is involutive for $0 \leq i \leq n-2$.

The exact state space linearization problem for single-input systems was posed and solved by Brockett [12]. For multiple-input systems, the problem was solved by Jakubczyk and Respondek [31]. Su [71] and Hunt, Su, and Meyer [27] independently presented a different formulation in addition to a procedure for constructing the linearizing transformation.

For reference, the *Brunovsky normal form* of the system (2.15) is

$$\begin{aligned} \dot{z}_0^i &= v_i \\ \dot{z}_1^i &= z_0^i \\ &\vdots \\ \dot{z}_{n_i}^i &= z_{n_i - 1}^i \end{aligned} \tag{2.16}$$

for $1 \leq i \leq m$ with $n_1 + \cdots + n_m + m = n$.

2.2 A Tutorial for Exterior Differential Systems

In this section, a tutorial for exterior differential systems is presented. This is the basis for understanding how to convert a multiple-input nonholonomic systems to extended Goursat normal form. A good collection of references for exterior differential systems is [25, 15, 11, 87, 70, 17, 22, 76, 44].

2.2.1 Exterior Algebra

An exterior algebra is first constructed on p-vectors for $p = 0, \ldots, n$, then basic properties of these algebras are given.

Define $\Lambda^0 L := \mathbb{R}$ to be the space of real numbers, or 0-vectors, with elements $a, b, c, \ldots$.

Define $\Lambda^1 L := L = \mathbb{R}^n$ as an n-dimensional vector space over $\mathbb{R}$, or the space of 1-vectors, with elements $\alpha, \beta, \gamma, \ldots$. Let $\{\sigma^1, \ldots, \sigma^n\}$ be a basis for L. The dimension of $\Lambda^1 L$ is $\begin{pmatrix} n \\ 1 \end{pmatrix} = n$, the number of combinations of n things taken one at a time.

Define $\Lambda^2 L$ to be the space of 2-vectors by defining the wedge product as a skew-symmetric bilinear map $\wedge : \Lambda^1 L \times \Lambda^1 L \to \Lambda^2 L$ that satisfies

(i) $(a_1\alpha^1 + a_2\alpha^2) \wedge \beta = a_1(\alpha^1 \wedge \beta) + a_2(\alpha^2 \wedge \beta)$ where $a_1, a_2 \in \mathbb{R}$ and $\alpha^1, \alpha^2, \beta$ are 1-vectors

(ii) $\alpha \wedge (b_1\beta^1 + b_2\beta^2) = b_1(\alpha \wedge \beta^1) + b_2(\alpha \wedge \beta^2)$ where $b_1, b_2 \in \mathbb{R}$ and β^1, β^2, α are 1-vectors

(iii) $\alpha \wedge \alpha = 0$

(iv) $\alpha \wedge \beta = -\beta \wedge \alpha$.

The notation $\alpha \wedge \beta$ is called the *exterior product* of the vectors α and β. That is, $\wedge$ is a bilinear, associative, distributive, non-commutative product mapping $\Lambda^1 L \times \Lambda^1 L \to \Lambda^2 L$. A basis for $\Lambda^2 L$ is given by $\{\sigma^i \wedge \sigma^j \ , \ 1 \leq i < j \leq n\}$, since

$$\alpha = \sum_i a_i \sigma^i \quad \text{and} \quad \beta = \sum_j b_j \sigma^j$$

combine as

$$\alpha \wedge \beta = \sum_{i<j} (a_i b_j - a_j b_i)\ \sigma^i \wedge \sigma^j \ .$$

It follows that the dimension of $\Lambda^2 L$ is $\begin{pmatrix} n \\ 2 \end{pmatrix}$, the number of combinations of n things taken two at a time.

$\Lambda^p L$ for $2 \leq p \leq n$ is the space of p-vectors consisting of all formal sums

$$\sum a(\alpha^1 \wedge \cdots \wedge \alpha^p) \quad a \in \mathbb{R}$$

with the properties

(i) $(a\alpha + b\beta) \wedge \alpha^2 \wedge \cdots \wedge \alpha^p = a(\alpha \wedge \alpha^2 \wedge \cdots \wedge \alpha^p) + b(\beta \wedge \alpha^2 \wedge \cdots \wedge \alpha^p)$

(ii) $\alpha^1 \wedge \cdots \wedge \alpha^p = 0$ if for some $i \neq j$ we have $\alpha^i = \alpha^j$

(iii) $\alpha^1 \wedge \cdots \wedge \alpha^p$ changes sign if any two α^i are interchanged.

If the indices $H := \{h_1, h_2, \ldots, h_p \ : \ 1 \leq h_1 < h_2 < \cdots < h_p \leq n\}$ are defined, the basis for $\Lambda^p L$ is given by the totality of $\sigma^H = \sigma^{h_1} \wedge \cdots \wedge \sigma^{h_p}$. Then any λ in $\Lambda^p L$ can be written as

$$\lambda = \sum_H a_H \sigma^H \ ,$$

where the sum is over all the ordered sets H. The dimension of $\Lambda^p L$ is $\begin{pmatrix} n \\ p \end{pmatrix}$, the number of combinations of n things taken p at a time. Thus, the dimension of $\Lambda^n L$ is 1. Λ^p is not defined for $p > n$, but sometimes it is convenient to set $\Lambda^p = 0$ for $p > n$.

The *exterior multiplication* operator, $\wedge$, is therefore defined in general as

$$\wedge : \ (\Lambda^p L) \times (\Lambda^q L) \rightarrow \Lambda^{p+q} L$$

and has the following properties for $\lambda \in \Lambda^p L$ and $\mu \in \Lambda^q L$:

(i) $\lambda \wedge \mu$ is associative: $\lambda \wedge (\mu \wedge \gamma) = (\lambda \wedge \mu) \wedge \gamma$

(ii) $\lambda \wedge \mu$ is bilinear in λ, μ: for $a_1, a_2, b_1, b_2 \in \mathbb{R}$,

$$\begin{aligned} (a_1\lambda^1 + a_2\lambda^2) \wedge \mu &= a_1(\lambda^1 \wedge \mu) + a_2(\lambda^2 \wedge \mu) \\ \lambda \wedge (b_1\mu^1 + b_2\mu^2) &= b_1(\lambda \wedge \mu^1) + b_2(\lambda \wedge \mu^2) \end{aligned}$$

(iii) $\lambda \wedge \mu$ is anticommutative: $\mu \wedge \lambda = (-1)^{pq} \lambda \wedge \mu$, where μ is a p-form and λ is a q-form.

Thus, the exterior product over the vector space L is used to form the vector space $\Lambda^p L$.

Proposition 7 *The vectors* $\alpha^1, \ldots, \alpha^s$ *are* linearly dependent *if and only if*

$$\alpha^1 \wedge \cdots \wedge \alpha^s = 0 \ . \tag{2.17}$$

2.2.2 Differential Forms

The exterior product over a vector space L^* (the dual of L) can now be used to form a vector space $\Lambda^p(L^*) := \Omega^p(L)$. As before, everything will be defined on the space $\mathbb{R}^n$ (instead of on a manifold). The tangent space to $\mathbb{R}^n$ at a point x, $T_x\mathbb{R}^n$, is a vector space of dimension n. The vector space $\Lambda^p(T_x\mathbb{R}^n)$ consists of all p-vectors. Attaching $\Lambda^p(T_x\mathbb{R}^n)$ to each point $x \in \mathbb{R}^n$ gives a bundle structure on $\mathbb{R}^n$, denoted by $\Lambda^p(\mathbb{R}^n)$. Similarly, the bundle $\Omega^p(\mathbb{R}^n)$ is defined using the dual space $T_x^*\mathbb{R}^n$. An element of $\Omega^p(\mathbb{R}^n)$ is called an *exterior differential p-form* on $\mathbb{R}^n$, $\Omega^p(\mathbb{R}^n)$ is called the totality of p-forms on $\mathbb{R}^n$, and $\Omega(\mathbb{R}^n)$ is called the *exterior differential algebra* of p-forms on $\mathbb{R}^n$.

Given the local coordinates $x_1, x_2, \ldots, x_n$ on $\mathbb{R}^n$, the dual bases are defined as

$$T_x\mathbb{R}^n = \text{span}\left\{\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\right\}$$

$$T_x^*\mathbb{R}^n = \text{span}\{dx_1, \ldots, dx_n\}$$

such that

$$dx_i\left(\frac{\partial}{\partial x_i}\right) = \delta_{ij} \ ,$$

where δ_{ij} is the Kronecker delta function, which takes values 1 for $i = j$ and 0 for $i \neq j$. In these coordinates, a vector field $f : \mathbb{R}^n \to T_x\mathbb{R}^n$ is written as

$$f(x) = \sum_{i=1}^{n} f^i(x)\frac{\partial}{\partial x_i} \ ,$$

where $f^i(x)$ are smooth functions.

The $\Omega^p(\mathbb{R}^n)$ spaces (spaces of p-forms) for $p = 0, 1, \ldots, n$ are constructed as follows, noting that:

(i) $\Omega^0(\mathbb{R}^n) := T_x^*\mathbb{R}^n$ is defined to be a space of smooth functions, called 0-forms, on $\mathbb{R}^n$

(ii) $\Omega^1(\mathbb{R}^n)$ is defined to be the space of covector fields, or one-forms.

A one-form, $\omega : T_x\mathbb{R}^n \to \mathbb{R}^n = \Omega^1(\mathbb{R}^n)$, is written as

$$\omega(x) = \sum_{j=1}^{n} \omega_j(x)dx_j \ ,$$

where $\omega_j(x)$ are smooth functions. A *p-form* is written in this basis as

$$\omega = \sum_H a_H(x)\, dx^H$$

for a multi-index $H := \{h_1, \ldots, h_p : 1 \leq h_1 < \cdots < h_p \leq n\}$, where $dx^H := dx^{h_1} \wedge \cdots \wedge dx^{h_p}$ and a_H are smooth functions on an open set U in $\mathbb{R}^n$. If ω is the p-form above and η is the q-form

$$\eta = \sum_K b_K(x)\, dx^K ,$$

then

$$\omega \wedge \eta = \sum_{H,K} a_H b_K\, dx^H \wedge dx^K .$$

Exterior Derivatives

Define the *exterior derivative* on $\Omega^k(\mathbb{R}^n)$ as the unique map

$$d : \Omega^k(\mathbb{R}^n) \to \Omega^{k+1}(\mathbb{R}^n)$$

with the properties

(i) $d\alpha$ is linear in α: $d(a_1\alpha^1 + a_2\alpha^2) = a_1 d\alpha^1 + a_2 d\alpha^2$, where $a_1, a_2 \in \mathbb{R}$ and α^1, α^2 are k-forms

(ii) $d\lambda$ satisfies the product rule: $d(\lambda \wedge \mu) = d\lambda \wedge \mu + (-1)^k \lambda \wedge d\mu$, where λ is a k-form and μ is a one-form

(iii) $d(d\alpha) = 0$, i.e., $d^2\alpha = 0$ for any k-form α

(iv) for a 0-form α, i.e., $\alpha = f \in C^\infty(\mathbb{R}^n)$ relative to a local coordinate chart, df is the one-form that is the differential of f: $df = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i$.

A k-form α is said to be *closed* if $d\alpha = 0$. It is called *exact* if there exists a $(k-1)$-form β such that $\alpha = d\beta$. The following lemma associates these two concepts.

Lemma 8 (Poincaré Lemma) *A closed form is locally exact, i.e., if $d\alpha = 0$, then there is a neighborhood about each point on which $\alpha = d\beta$.*

To illustrate the use of the exterior derivative, consider the following example. For $\mathbb{R}^3$ with $x := (x_1, x_2, x_3)$ the 0-form f is written as

$$df = \frac{\partial f}{\partial x_1}\, dx_1 + \frac{\partial f}{\partial x_2}\, dx_2 + \frac{\partial f}{\partial x_3}\, dx_3 \ ,$$

and is called the *differential* of f. For the one-form

$$\omega = P\, dx_1 + Q\, dx_2 + R\, dx_3 \ ,$$

$$d\omega = \left(\frac{\partial R}{\partial x_2} - \frac{\partial Q}{\partial x_3}\right) dx_2 \wedge dx_3 + \left(\frac{\partial P}{\partial x_3} - \frac{\partial R}{\partial x_1}\right) dx_3 \wedge dx_1 + \left(\frac{\partial Q}{\partial x_1} - \frac{\partial P}{\partial x_2}\right) dx_1 \wedge dx_2$$

is called the *curl* of ω. For the two-form

$$\alpha = A\, dx_2 \wedge dx_3 + B\, dx_3 \wedge dx_1 + C\, dx_1 \wedge dx_2,$$

$$d\alpha = \left(\frac{\partial A}{\partial x_1} + \frac{\partial B}{\partial x_2} + \frac{\partial C}{\partial x_3}\right) dx_1 \wedge dx_2 \wedge dx_3$$

is called the *divergence* of α.

2.2.3 Exterior Differential Systems

Some of the mathematical tools from exterior differential systems that are used in this dissertation will now be presented. A more thorough description can be found in [15].

An *algebraic ideal* $\mathcal{I} \subset \Omega(\mathbb{R}^n)$ is a collection of smooth differential forms on $\mathbb{R}^n$ that satisfies

(i) if $\alpha^1, \alpha^2 \in \mathcal{I}$ and $f_1, f_2 \in C^\infty(\mathbb{R}^n)$, then $f_1\alpha^1 + f_2\alpha^2 \in \mathcal{I}$

(ii) if $\alpha \in \mathcal{I}$ and $\beta \in \Omega(\mathbb{R}^n)$, then $\alpha \wedge \beta \in \mathcal{I}$.

Given a collection $\mathcal{F}$ of smooth differential forms, the smallest algebraic ideal of smooth differential forms that includes $\mathcal{F}$ is called the *algebraic ideal generated by* $\mathcal{F}$.

An *exterior differential system* $\mathcal{I}$ on $\mathbb{R}^n$ is an algebraic ideal that, in addition, is closed under exterior differentiation, i.e.,

$$\text{for any } \alpha \in \mathcal{I},\ d\alpha \in \mathcal{I} \ .$$

In short notation, this is written as $d\mathcal{I} \subset \mathcal{I}$. The set $d\mathcal{I}$ consists of exterior derivatives of elements of $\mathcal{I}$.

A *Pfaffian system*, denoted by I, on $\mathbb{R}^n$ is a submodule of the module of differential one-forms $\Omega^1(\mathbb{R}^n)$ over the commutative ring of smooth functions $C^\infty(\mathbb{R}^n)$. A *codimension two* Pfaffian system is generated by a set of $n-2$ linearly independent one-forms

$$\alpha^0 = \alpha^1 = \cdots = \alpha^{n-3} = 0 ,$$

as

$$I = \left\{ \sum_{k=0}^{n-3} f_k \, \alpha^k , \quad \text{for} \quad f_k \in C^\infty(\mathbb{R}^n) \right\} . \tag{2.18}$$

In the context of wheeled nonholonomic systems, the one-forms are the kinematic constraints of the system, i.e., the directions in which the system cannot move instantaneously. Since the kinematic constraints are imposed on the system in a neighborhood of some initial point, the Pfaffian system is also local.

Let I be defined as the codistribution spanned by a set of linearly independent one-forms

$$I = \{\alpha^0, \ldots, \alpha^{n-3}\} .$$

The codistribution is the annihilator of the distribution of allowable velocities of the nonholonomic system, as described in Section 2.1. The ideal generated by I is

$$\mathcal{I} = \{I\} = \{\sigma \in \Omega \, : \, \sigma \wedge \alpha^0 \wedge \cdots \wedge \alpha^{n-3} = 0\} .$$

$\mathcal{I}$ is *integrable* if there exists functions $h_0, \ldots, h_{n-3}$ such that $\mathcal{I} = \{dh_0, \ldots, dh_{n-3}\}$.

The Frobenius Theorem 4 of nonlinear control theory has the following dual in exterior differential systems.

Theorem 9 (Frobenius Integration Theorem) *Consider a Pfaffian system generated by linearly independent one-forms $\alpha^0, \ldots, \alpha^{n-3}$. If these one-forms satisfy the Frobenius condition*

$$d\alpha^k = \sum_{i=0}^{n-3} \theta_k^i \wedge \alpha^i \quad 0 \le k \le n-3 \tag{2.19}$$

for some one-forms θ_k^i, then the Pfaffian system is completely integrable, i.e., there exist functions f_k^i, and h_i satisfying

$$\alpha^k = \sum_{i=0}^{n-3} f_k^i dh_i \quad \textit{for} \quad 0 \le k \le n-3 .$$

See [15] or [22] for a proof. The Frobenius condition (2.19) is equivalent to

$$d\alpha^k \wedge \alpha^0 \wedge \cdots \wedge \alpha^{n-3} = 0 \quad 0 \leq k \leq n-3 \,. \tag{2.20}$$

A useful tool in analyzing exterior differential systems is the derived flag. Define the codistributions

$$\begin{aligned} I^{(0)} &:= I = \{\alpha^0, \alpha^1, \ldots, \alpha^{n-3}\} \\ I^{(i)} &:= \{\lambda \in I^{(i-1)} \,:\, d\lambda \equiv 0 \mod I^{(i-1)}\} \end{aligned}$$

for $i = 1, \ldots, N$, where N is the step in which this procedure terminates, i.e., $I^{(N+1)} = I^{(N)}$. The notation *mod* I means modulo the algebraic ideal generated by I; that is, given two one-forms α and β in Ω, $\alpha \equiv \beta \mod I$ if there exists a one-form $\eta \in I$ such that $\alpha = \beta + \eta$. The set of codistributions

$$I = I^{(0)} \supset I^{(1)} \supset I^{(2)} \supset \cdots \supset I^{(N)}$$

is called the *derived flag* of the Pfaffian system I and N is called the *derived length.* If I is completely integrable, then $N = 0$ and $I^{(1)} = I^{(0)}$ by the Frobenius Theorem 9. In fact, $I^{(N)}$ is always integrable since $d\alpha \equiv 0 \mod I^{(N)}$ for all $\alpha \in I^{(N)}$. $I^{(N)}$ is the *largest* integrable subsystem contained in I. Therefore, if $I^{(N)}$ is non-empty, e.g., contains one-forms $\alpha^0, \ldots, \alpha^{k-1}$, then there exists functions $h_0, \ldots, h_{k-1}$ such that $I^{(N)} = \{dh_0, \ldots, dh_{k-1}\}$. This means the system is not controllable since there exist algebraic functions that give a foliation of the state space and the solution trajectories of I are constrained to lie on level surfaces of $\{h_0, \ldots, h_{k-1}\}$.

A Pfaffian system is called *nonholonomic* if $I^{(1)}$ is a proper subset of I. The following discussion is restricted to the class of systems that are *maximally nonholonomic*, or completely controllable, i.e., $I^{(N)} = \{0\}$. Consider the derived flag with basis $\{\alpha^i\}$ *adapted to the derived flag*, that is, a basis such that the basis of $I^{(j+1)}$ is a subset of the basis of $I^{(j)}$ for $j = 0, \ldots, N-1$:

$$\begin{aligned} I^{(0)} := I &:= \{\alpha^0, \alpha^1, \ldots, \alpha^{n-3}\} \\ I^{(1)} &:= \{\alpha^0, \alpha^1, \ldots, \alpha^{n-4}\} \\ &\vdots \\ I^{(N-1)} &:= \{\alpha^0\} \\ I^{(N)} &:= \{0\} \,. \end{aligned} \tag{2.21}$$

Theorem 10 (Chow) *Given a Pfaffian system generated by smooth, linearly independent one-forms $\alpha^0, \ldots, \alpha^{n-3}$ and a derived flag for this system, there exists a path $x(t)$ between any two points satisfying*

$$\alpha^i \cdot \dot{x} = 0 \quad 0 \leq i \leq n-3$$

if and only if there exists an integer K such that $I^{(K)} = \{0\}$.

The proof consists of converting the problem to the vector field version and following the proof in [54].

A basic problem in exterior differential systems is to study the *integral manifolds*, or solutions, of a system $\alpha = 0$. In local coordinates, this is a system of partial differential equations. For example,

$$c_1(x_1, \ldots, x_n)\, dx_1 + \cdots + c_n(x_1, \ldots, x_n)\, dx_n = 0$$

is called a *Pfaffian equation.* Pfaff's problem is to find the integral manifolds of maximal dimension. Consider Pfaff's problem and solution for a exterior differential system consisting of one equation

$$\alpha = 0\,, \tag{2.22}$$

where α is a one-form. The *rank* of α is the smallest integer r such that

$$\begin{aligned} (d\alpha)^r \wedge \alpha &\neq 0 \\ (d\alpha)^{r+1} \wedge \alpha &= 0\,. \end{aligned}$$

Theorem 11 (Pfaff's Problem) *Assume α in equation (2.22) has constant rank r in a neighborhood of $x_0 \in \mathbb{R}^n$. Then there exists a coordinate system $x_1, \ldots, x_n$ (possibly in a smaller neighborhood of x_0) such that α can be written in the* normal form

$$\alpha := dx_1 + x_2 dx_3 + \cdots + x_{2r} dx_{2r+1}\,.$$

The proof can be found in [15]. For the rank 1 case, the proof reduces to finding two functions f_1 and f_2 satisfying

$$\begin{aligned} d\alpha \wedge \alpha \wedge df_1 = 0 \quad &\text{and} \quad \alpha \wedge df_1 \neq 0 \\ \alpha \wedge df_1 \wedge df_2 = 0 \quad &\text{and} \quad df_1 \wedge df_2 \neq 0 \end{aligned}$$

from which α can be scaled such that

$$\alpha := df_2 + g_1 df_1 = dx_1 + x_2 dx_3\,.$$

Theorem 12 (Engel's Normal Form) *Given a codimension two Pfaffian system in* $\mathbb{R}^4$ *with the derived flag satisfying*

$$\dim I^{(1)} = 1 \quad \textit{and} \quad I^{(2)} = \{0\} \ ,$$

there exist local coordinates z_0^0, z_0, z_1, z_2 *such that*

$$I = \{dz_2 - z_1 dz_0^0,\ dz_1 - z_0 dz_0^0\} \ , \tag{2.23}$$

which is called Engel's normal form.

Proof [15, 76]. The proof is repeated here since the concepts will be used later when a Pfaffian system is transformed into Goursat normal form.

Choose a basis for the codimension two Pfaffian system $I = \{\alpha^0, \alpha^1\}$ that is adapted to the derived flag. By a dimension count, $d\alpha^0 \wedge \alpha^0 \neq 0$ and $(d\alpha^0)^2 \wedge \alpha^0 = 0$, giving α^0 rank 1. Hence, Pfaff's Theorem 11 can be used to find coordinates z_0^0, z_1, z_2 such that $\alpha^0 = dz_2 - z_1 dz_0^0$.

By construction $\alpha^0 \in I^{(0)}$, i.e., $d\alpha^0 \wedge \alpha^0 \wedge \alpha^1 = 0$. Since $d\alpha^0 = -dz_1 \wedge dz_0^0$, α^1 must have the structure

$$\alpha^1 = a\ dz_1 + b\ dz_0^0 \mod \alpha^0 \ .$$

Since $\alpha^1 \neq 0$, a and b cannot both be zero. There are two cases to consider to find the coordinate z_0.

Case 1: ($a \neq 0$). Since α^1 is only determined mod α^0, α^1 can be scaled by a non-zero function as

$$\frac{1}{a}\alpha^1 = dz_1 + \frac{b}{a} dz_0^0 \mod \alpha^0 \ . \tag{2.24}$$

Choosing $z_0 = -b/a$ then completes the set of coordinates for the basis for the codistribution I, which is in Engel's normal form. Notice that this basis is a transformed version of the original basis

$$\begin{aligned} \bar{\alpha}^0 &= \alpha^0 &&= dz_2 - z_1 dz_0^0 \\ \bar{\alpha}^1 &= \tfrac{1}{a}\alpha^1 + \lambda\alpha^0 &&= dz_1 - z_0 dz_0^0 \end{aligned}$$

where λ is chosen such that equation (2.24) is an equality, i.e., no mod α^0.

Case 2: ($b \neq 0$). In this case, α^1 is scaled as

$$\frac{1}{b}\alpha^1 = \frac{a}{b}dz_1 + dz_0^0 \mod \alpha^0 ,$$

and z_0 is chosen as $-a/b$ to complete the set of coordinates for the basis for the codistribution I, which is in the normal form

$$\begin{aligned} \bar{\alpha}^0 &= dz_2 - z_1 dz_0^0 \\ \bar{\alpha}^1 &= dz_0^0 - z_0 dz_1 . \end{aligned} \tag{2.25}$$

This normal form is diffeomorphic to Engel's normal form via the following change of coordinates:

$$\begin{aligned} &\eta_0^0 = z_1 \\ &\eta_0 = z_0 \qquad\qquad \Longrightarrow \quad \bar{\alpha}^0 = d\eta_2 - \eta_1 d\eta_0^0 \\ &\eta_1 = z_0^0 \qquad\qquad\quad\ \ \bar{\alpha}^1 = d\eta_1 - \eta_0 d\eta_0^0 \\ &\eta_2 = z_2 - z_1 z_0^0 . \end{aligned}$$

□

A generalization of Engel's normal form is stated as follows.

Theorem 13 (Goursat Normal Form) *Given a codimension two Pfaffian system*

$$I = \{\alpha^0, \ldots, \alpha^{n-3}\} \tag{2.26}$$

in $\mathbb{R}^n$, *if there exists an integrable one-form* $\pi \neq 0 \mod I$ *satisfying the Goursat congruences*

$$\begin{aligned} d\alpha^i &\equiv \pi \wedge \alpha^{i+1} \mod \alpha^0, \ldots, \alpha^i , \ 0 \leq i \leq n-4 \\ d\alpha^{n-3} &\neq 0 \mod I , \end{aligned} \tag{2.27}$$

then there exist local coordinates $z_0^0, z_0, z_1, \ldots, z_{n-2}$ *such that*

$$I = \{dz_{n-2} - z_{n-3}dz_0^0, \ \ldots, \ dz_1 - z_0 dz_0^0\} . \tag{2.28}$$

Equation (2.28) is called the *Goursat normal form* of the Pfaffian system (2.26).

This theorem requires the existence of a basis $\{\alpha^i\}$ and a one-form π of certain structure. The basis is, however, adapted to the derived flag of the system, as can be verified by a simple calculation. Therefore, if we start with the basis adapted to the derived flag, we only need to find a one-form π that satisfies the Goursat congruences (2.27). Finding

this π can be difficult and involves further scaling of the basis while preserving the adapted structure.

Reference [15] gives a complete proof of Theorem 13. The following algorithm for converting a codimension two Pfaffian system into Goursat normal form summarizes the proof. It is based on the Gardner and Shadwick algorithm [24], which is a feedback linearization version of this algorithm.

Algorithm 1 ***(Conversion to Goursat Normal Form)***
Consider a codimension two Pfaffian system $I = \{\alpha^0, \ldots, \alpha^{n-3}\}$ in $\mathbb{R}^n$. The following steps will transform the Pfaffian system into the Goursat normal form (2.28).

(i) Construct a basis $I = \{\omega^0, \ldots, \omega^{n-3}\}$ that is adapted to the derived flag of the Pfaffian system.

(ii) Find an integrable one-form π for which the Goursat congruences (2.27) are satisfied with this basis. The coordinate z_0^0 is chosen such that $dz_0^0 = \pi$.

(iii) From the Goursat congruences, ω^0 and ω^1 satisfy $d\omega^0 \wedge \omega^0 \wedge \omega^1 = 0$. Therefore, use the proof of Engel's Theorem 12 to find coordinates z_{n-2}, z_{n-3}, z_{n-4} such that

$$\begin{aligned} \omega^0 &= dz_{n-2} - z_{n-3}dz_0^0 \\ \omega^1 &= dz_{n-3} - z_{n-4}dz_0^0 \, . \end{aligned}$$

This may involve scaling the basis of ω^i while preserving the adapted structure and a change of coordinates to convert between the two normal forms in the proof of Engel's Theorem 12.

(iv) Find the remaining coordinates by algebraically solving the equations

$$\omega^k = dz_{n-k-2} - z_{n-k-3}dz_0^0 \mod \omega^0, \ldots, \omega^k \quad k = 2, \ldots, n-3 \, . \tag{2.29}$$

The proof of the Goursat normal form Theorem 13 is essentially to show that equation (2.29) always has a solution.

Remark. The Goursat normal form is dual to what is called a *two-input, single-generator chained form*, which was introduced in Murray and Sastry [57]. Considering the codimension

two case in $\mathbb{R}^n$, the system in Goursat normal form (2.28) can be written as a control system in chained form by choosing the vector fields

$$\begin{aligned} g_0 &= \frac{\partial}{\partial z_0^0} + z_0^0 \frac{\partial}{\partial z_1} + \cdots + z_{n-3} \frac{\partial}{\partial z_{n-2}} \\ g_1 &= \frac{\partial}{\partial z_0^0}\,, \end{aligned} \tag{2.30}$$

which form a basis for the distribution annihilated by $I = \{\alpha^0, \ldots, \alpha^{n-3}\}$. Thus, the problem of finding a basis for the constraints α^i in Goursat normal form is equivalent to finding a feedback transformation to the chained form (this transformation will be discussed in detail in Section 3.1).

Finding the integrable one-form π for the Goursat normal form can be difficult. In [52], Murray completely characterizes the set of codimension two systems that are equivalent to a system in chained or Goursat normal form. Let $I = \{\alpha^0, \ldots, \alpha^{n-3}\}$ be a codistribution on $\mathbb{R}^n$ and write $\Delta = I^\perp$ for the distribution that annihilates I. For example, $\Delta = \text{span}\{g_0, g_1\}$ with g_0 and g_1 as in equation (2.30). Two filtrations are defined as

$$\begin{array}{lll} E_0 := \Delta & & G_0 := \Delta \\ E_1 := E_0 + [E_0, E_0] & & G_1 := G_0 + [G_0, G_0] \\ E_2 := E_1 + [E_1, E_1] & & G_2 := G_1 + [G_1, G_0] \\ \vdots & & \vdots \\ E_{i+1} := E_i + [E_i, E_i] & & G_{i+1} := G_i + [G_i, G_0]\,. \end{array} \tag{2.31}$$

The filtration $\{G_i\}$ is the one that usually appears in the context of nonlinear feedback linearization (see equation (2.14)) and contains all Lie brackets up to order i. The filtration $\{E_i\}$ also contains Lie brackets of order i, but may also contain higher-order Lie products (up to order 2^{i-1}). This is due to the recursive construction of E_i as opposed to the iterative construction of G_i. The filtration $\{E_i\}$ is the sequence of distributions that is perpendicular to the derived flag of $I = \Delta^\perp$. Using these filtrations, [52] gives the following two results.

Theorem 14 (Murray) *There exists a basis $\{\omega^0, \ldots, \omega^{n-3}\}$ for a codimension two Pfaffian system $I = \Delta^\perp$ that is in Goursat normal form (2.28) if and only if*

$$\dim E_i = \dim G_i = i + 2 \quad \textit{for} \quad 0 \le i \le n-2\,.$$

See [52] for a proof. The following corollary uses the fact that the Goursat normal form and the chained form are dual forms.

Corollary 15 (Murray) *There exists a feedback transformation that converts a two-input nonholonomic system into chained form if and only if*

$$\dim E_i = \dim G_i = i+2 \quad \textit{for} \quad 0 \le i \le n-2 \; .$$

Consider now what happens when there are more than two inputs, which corresponds to a Pfaffian system with codimension greater than two. In general, a Pfaffian system $I = \{\alpha^0, \ldots, \alpha^{s-1}\}$ is said to have *codimension* m if $s+m=n$, the dimension of the state space $\mathbb{R}^n$. A *tower* of the derived flag must first be defined. In the codimension two case, there is only one tower in the derived flag, which consists of the basis adapted to the derived flag in equation (2.21).

Definition 1 (Towers of a derived flag) *Let I be a collection of $n_1 + \cdots + n_m$ smooth linearly independent one-forms defined on an open set $U \subset \mathbb{R}^n$ with $n = m+1+\sum_{j=1}^{m} n_j$,*

$$I \;=\; \{\omega_1^j, \omega_2^j, \ldots, \omega_{n_j}^j : j = 1, \ldots, m\} \; . \tag{2.32}$$

Thus, I is a codimension $m+1$ Pfaffian system. Let the one-form $\pi \neq 0 \mod I$ be such that for $j = 1, \ldots, m$,

$$\begin{aligned} d\omega_k^j &\equiv \pi \wedge \omega_{k+1}^j \mod I^{(n_j-k)} \; , \; k = 1, \ldots, n_j - 1 \\ d\omega_{n_j}^j &\neq 0 \mod I \; . \end{aligned} \tag{2.33}$$

These congruences imply that the derived flag associated with the system I has the form

$$I^{(i)} = \{\omega_1^j, \omega_2^j, \ldots, \omega_{n_j-i}^j : j = 1, \ldots, m\} \quad i = 0, \ldots N \; ,$$

where N is the step where $I^{(N)} = I^{(N+1)}$. If $i \ge n_j$, then none of the constraints $\omega^j := (\omega_1^j, \omega_2^j, \ldots, \omega_{n_j}^j)$ will appear in the i^{th} derived system. Under this construction, we say the derived flag of I has m towers. *The set of relations (2.33) will be referred to as the* extended Goursat congruences.

In other words, when considering Pfaffian systems with codimension $m+1$, we use a derived flag with m towers. This is the dual to what is called a single-generator, $(m+1)$-input, chained form with m chains.

If the filtration $\{E_i\}$ as in equation (2.31) is constructed for $\Delta = \text{span}\{g_0, \ldots, g_m\}$, this filtration is still the sequence of distributions that is perpendicular to the derived flag of

$I = \Delta^{\perp}$ with m towers. Theorem 14 cannot, however, be extend to this higher codimension case, since the distribution F_i may contain more vector fields than E_i due to the many possible combinations of Lie brackets between the $g_j \in \Delta$. Thus for $m > 1$ (more than two inputs), the filtration $\{F_i\}$ cannot be guaranteed to grow as $\{E_i\}$ grows.

Gardner and Shadwick [24] extend Theorem 13 to a system with codimension greater than two with the following theorem.

Theorem 16 (Extended Goursat Normal Form) *Let U be an open subset of $\mathbb{R}^n$ and*

$$I = \{\omega_1^j, \omega_2^j, \ldots, \omega_{n_j}^j : j = 1, \ldots, m\}$$

be a collection of $n_1 + \cdots + n_m$ smooth linearly independent one-forms defined on U, where $n = m + 1 + \sum_{j=1}^{m} n_j$. If there exists an integrable one-form $\pi \neq 0 \mod I$ such that the extended Goursat congruences (2.33) are satisfied, then there exists a set of n coordinates on U such that I can be written as

$$I = \{dz_{n_j}^j - z_{n_j-1}^j dz_0^0, \ldots, \; dz_1^j - z_0^j dz_0^0 \; : \; j = 1, \ldots, m\} \,. \tag{2.34}$$

See [24, 52] for proof. Equation (2.34) represents an extended Goursat normal form system with m towers.

Remark. It is necessary to have an integrable π in order to construct the coordinates for the extended Goursat normal form. This was not stated in [24] since the π for their control system was always equal to dt. If the one-form π in the above theorem is not integrable, then the Frobenius Theorem 9 cannot be used to find the z_i^j coordinates for I [77, 53]. In the case where only one tower is the longest, Lemma 17 below shows that if there exists any π that satisfies the extended Goursat congruences, then there exists an *integrable* π' that also satisfies these congruences (under a rescaling of the basis one-forms). In the case where $n_1 = n_2$, or at least two towers have the longest length, however, this is no longer true. Therefore, assuming π is integrable in Theorem 16 is necessary.

Lemma 17 *Consider the case where one tower has the longest length. If there exists a single one-form $\pi \neq 0 \mod I$ that satisfies the extended Goursat congruences (2.33), then there exists an integrable one-form π' that also satisfies these congruences.*

Proof. Without loss of generality, consider the case when the first tower is the longest. Given that there exists a $\pi \neq 0$ that satisfies the extended Goursat congruences (2.33), the

derived flag has the structure

$$\begin{aligned} I^{(n_1-1)} &= \{\omega_1^1\} \\ I^{(n_1)} &= \{0\} . \end{aligned}$$

The last Goursat congruence

$$d\omega_1^1 = \pi \wedge \omega_2^1 \mod \omega_1^1 \tag{2.35}$$

implies that ω_1^1 has rank 1. Thus, from the solution to Pfaff's problem, there exists a function f_1 satisfying the equation $d\omega_1^1 \wedge \omega_1^1 \wedge df_1 = 0$. Substituting $d\omega_1^1$ from (2.35) gives $\pi \wedge \omega_2^1 \wedge \omega_1^1 \wedge df_1 = 0$, which shows that df_1 is linearly dependent on π, ω_2^1, and ω_1^1, i.e.,

$$df_1 = k_0(x)\ \pi + k_1(x)\ \omega_1^1 + k_2(x)\ \omega_2^1 ,$$

where $x = (x_1, \ldots, x_n)$ are the local coordinates on $\mathbb{R}^n$. Define $\pi' = df_1$ and note that any such π' with $k_0 \neq 0$ also satisfies the extended Goursat congruences (2.33). □

Chapter 3

Conversion to Chained and Extended Goursat Normal Forms

In this chapter, methods are developed for converting the kinematic model of nonholonomic wheeled vehicles into chained form and extended Goursat normal form. The two methods presented are duals of each other: geometric nonlinear control theory and the theory of exterior differential systems. Sufficient conditions for converting to chained form are derived using vector field methods. Using techniques from exterior differential systems, both necessary and sufficient conditions can be derived for transforming into extended Goursat normal form.

3.1 Conversion to Chained Form

In this section, a method to convert systems into a special canonical form of systems called *chained form systems* is presented. Once in chained form, the system can be easily steered using various methods discussed in Chapter 4.

We are interested in steering mechanical systems with nonholonomic, or non-integrable, linear velocity constraints

$$\omega^i(x) \cdot \dot{x} = 0, \quad i = 1, 2, \ldots, k \ ,$$

where $x \in \mathbb{R}^n$ is the state of the system and the $\omega^i(x) \in \mathbb{R}^n$ are row vectors, or one-forms. For mobile robots, these constraints arise when a wheel rolls without slipping on a surface, expressing that the relative velocity of the two points in contact is zero.

The ω^i, are assumed to be linearly independent and smooth. The corresponding codistribution $\Omega(x) = \text{span}\{\omega^1(x), \ldots, \omega^k(x)\}$ has dimension k. Therefore, an $(n-k)$-dimensional distribution $\Delta(x) = \text{span}\{g_0(x), \ldots, g_{n-k-1}(x)\}$ can be found with all $g_j(x) \in \mathbb{R}^n$, such that $\Delta = \Omega^\perp$, i.e., $\omega^i(x) \cdot g_j(x) = 0$ for all $\omega^i \in \Omega$, $g_j \in \Delta$. Then a kinematic system with the above nonholonomic constraints can be represented as a control system with inputs u_j as follows:

$$\dot{x} = g_0(x)u_0(t) + \cdots + g_{n-k-1}(x)u_{n-k-1}(t) \; .$$

The motion planning problem therefore consists of controlling the drift-free system

$$\dot{x}(t) = g_0(x)u_0(t) + \cdots + g_m(x)u_m(t)$$

where x is in the open set $U \subset \mathbb{R}^n$, $u_j(t) \in \mathbb{R}$, $m+1 \leq n$, and the g_j are smooth, linearly independent vector fields. All subsequent conditions are assumed to hold on the open set U. Given x^0 and x^f, the goal is to find a control law $u = (u_0(t), \ldots, u_m(t))$ to steer $x(0) = x^0$ to $x(T) = x^f$ on the time interval $[0, T]$.

Chained form systems were first introduced by Murray and Sastry [57] as a class of systems inspired by Brockett [13] to which one could convert a number of interesting examples, including a car and a car with one trailer, and for which it was easy to derive sinusoidal steering control laws. These examples have two inputs and their chained forms have one chain. If the system meets certain sufficient conditions allowing it to be transformed into what is called a *single-chain, single-generator* chained form, defined by

$$\begin{aligned} \dot{z}_0 = v_0 \qquad \dot{z}_1 &= v_1 \\ \dot{z}_2 &= z_1 v_0 \\ &\vdots \\ \dot{z}_{n-1} &= z_{n-2} v_0 \end{aligned} \tag{3.1}$$

after a nonlinear change of coordinates and state feedback, then the system may be steered by setting the inputs v_0 and v_1 to be sinusoids at integrally related frequencies. This is called a *chained form system* because the derivative of each state depends on the state directly above it in a chained fashion. This particular chained form is reminiscent of a Brunovsky normal form. Indeed, with the input v_0 set to 1, the coordinates $z_1, \ldots, z_{n-1}$ are in Brunovsky canonical form (2.16). Chained form systems, however, are nonlinear,

drift-free, and bilinear in the input and state variables. The input v_0 that appears in the chain is called the *generating* input, thus the name "single-generator chained form." A more general chained form is discussed in Chapter 7 (see also [57]) that can have more than one generating input and multiple chains leading down from each input. In this section, we are only interested in chained form systems that have a single generator.

In the following, systems with three or more inputs are converted into a chained form with more than one chain. In this special form, the system can be controlled using sinusoidal, polynomial, or piecewise constant input functions as described in Chapter 4.

The outline of this section is as follows. In Section 3.1.1, sufficient conditions are presented for transforming a three-input nonholonomic system into a two-chain, single-generator chained form. In Section 3.1.2, the controllability of chained form systems is discussed. In Section 3.1.3, these ideas are generalized to $(m+1)$-inputs.

3.1.1 Converting to Two-chain, Single-generator Chained Form

As will be seen in the example section, finding the control inputs $\{u_j\}$ that will steer the state $x \in \mathbb{R}^n$ from an initial configuration to a final configuration may be difficult if the kinematic equations are complicated. Converting to chained form greatly simplifies this task: the structure of the chained form system allows one to easily construct sinusoidal, polynomial, or piecewise constant control inputs.

Deriving conditions to transform a nonholonomic system with two or more inputs into chained form is straightforward when we recall the method for exact linearization of a nonlinear system with drift via state feedback and a coordinate transformation as presented in Section 2.1.5. In analogy to this method, the following theorem that gives sufficient conditions for transforming a three-input, drift-free, nonholonomic system into chained form can be stated.

Theorem 18 ***(Converting to Two-chain, Single-generator Chained Form)*** *Consider a three-input, drift-free, nonholonomic system*

$$\dot{x} = g_0(x)u_0 + g_1(x)u_1 + g_2(x)u_2 \tag{3.2}$$

with smooth, linearly independent input vector fields g_0, g_1, g_2. There exists a feedback

transformation on some open set $U \subset \mathbb{R}^n$

$$\begin{aligned}(\xi, \zeta, \eta) &= \Phi(x) \\ u &= \beta(x)v\end{aligned}$$

that transforms the system (3.2) *into two-chain, single-generator chained form*

$$\begin{array}{lllllll} \dot{\xi}_0 &=& v_0 & \dot{\zeta}_0 &=& v_1 & \dot{\eta}_0 &=& v_2 \\ & & & \dot{\zeta}_1 &=& \zeta_0 v_0 & \dot{\eta}_1 &=& \eta_0 v_0 \\ & & & \vdots & & & \vdots & & \\ & & & \vdots & & & \dot{\eta}_{n_2} &=& \eta_{n_2-1} v_0 \\ & & & \dot{\zeta}_{n_1} &=& \zeta_{n_1-1} v_0 & & & \end{array} \tag{3.3}$$

if there exists a basis f_0, f_1, f_2 *for* $\Delta_0 := span\{g_0, g_1, g_2\}$ *that has the form*

$$\begin{aligned} f_0 &= \frac{\partial}{\partial x_1} + \sum_{i=2}^{n} f_0^i(x) \frac{\partial}{\partial x_i} \\ f_1 &= \sum_{i=2}^{n} f_1^i(x) \frac{\partial}{\partial x_i} \\ f_2 &= \sum_{i=2}^{n} f_2^i(x) \frac{\partial}{\partial x_i} \end{aligned} \tag{3.4}$$

such that the distributions

$$\begin{aligned} G_0 &:= span\{f_1, f_2\} \\ G_1 &:= span\{f_1, f_2, ad_{f_0} f_1, ad_{f_0} f_2\} \\ &\vdots \\ G_{n-1} &:= span\{ad_{f_0}^i f_1, ad_{f_0}^i f_2 : 0 \le i \le n-1\} \end{aligned} \tag{3.5}$$

have constant dimension on U, *are all involutive, and* G_{n-1} *has dimension* $n-1$ *on* U.

Proof. First denote the dimension of each distribution as $d_i := \dim G_i$. By construction, $d_0 = 2$ and by assumption, $d_{n-1} = n-1$. Since $G_0 \subseteq G_1 \subseteq \cdots \subseteq G_{n-1}$,

$$2 = d_0 \le d_1 \le \cdots \le d_{n-1} = n-1 \, .$$

Let n_1 be the smallest integer less than n such that

$$\begin{aligned} \dim G_{n_1} &= n-1 \\ \dim G_{n_1-1} &= n-2 \end{aligned}$$

and let n_2 be the integer when $\dim G_{n_2-1}$ first drops by two. Thus, $\mathrm{ad}^i_{f_0} f_1$ and $\mathrm{ad}^i_{f_0} f_2$ each give new directions up to some level, n_2, when one chain saturates and the other chain continues to give new directions until the state space is spanned:

$$d_i = \begin{cases} 2(i+1) & i = 0, \ldots, n_2 \\ (i+1)+(n_2+1) & i = n_2, \ldots, n_1 \\ n-1 & i = n_1, \ldots, n-1 \end{cases}$$

with $n_2 + n_1 + 2 = n - 1$. Without loss of generality, this proof will use the case $n_1 \geq n_2$. A basis for G_{n-1} can then be chosen that is given by the first n_1 brackets of f_0 with f_1 and the first n_2 brackets of f_0 with f_2:

$$G_{n-1} = \cdots = G_{n_1} = \mathrm{span}\{f_1, \mathrm{ad}_{f_0} f_1, \ldots, \mathrm{ad}^{n_1}_{f_0} f_1,\ f_2, \mathrm{ad}_{f_0} f_2, \ldots, \mathrm{ad}^{n_2}_{f_0} f_2\}\ ,$$

where f_1 and f_2 have been renumbered if necessary.

Because of the special form (3.4) of the vector fields, none of the vector fields in G_{n-1} has an entry $\frac{\partial}{\partial x_1}$, thus

$$\mathrm{span}\{f_0,\ f_1, \mathrm{ad}_{f_0} f_1, \ldots, \mathrm{ad}^{n_1}_{f_0} f_1,\ f_2, \mathrm{ad}_{f_0} f_2, \ldots, \mathrm{ad}^{n_2}_{f_0} f_2\}$$

has dimension n on U. Then since the distribution G_{n-1} is involutive and of dimension $d_{n-1} = n - 1$ on U, Frobenius Theorem 4 shows that there exists $n - d_{n-1} = 1$ smooth function $h_0 : U \to \mathbb{R}$ such that $dh_0 \cdot X = 0$ for all $X \in G_{n-1}$. Furthermore, $dh_0 \cdot f_0(x) = a_0(x) \neq 0$. With f_0 in the special form of equation (3.4), h_0 can be chosen to be x_1, which gives $dh_0 \cdot f_0(x) = 1$. It can also be verified that none of the vector fields in G_{n-1} has an entry in the first coordinate, giving $dh_0 \cdot X = 0$ for all $X \in G_{n-1}$. By the dimension argument, $G_i = G_{n-1}$ for $i = n_1, \ldots, n-2$, thus $dh_0 \perp G_i$ for $i = n_1, \ldots, n-1$.

The distribution G_{n_1-1} drops dimension by one by removing the vector field $\mathrm{ad}^{n_1}_{f_0} f_1$ from $G_{n-1} = G_{n_1}$. Since this distribution is involutive and $d_{n_1-1} = n - 2$, there exist two smooth functions whose derivatives span $G^{\perp}_{n_1-1}$. One of these functions is h_0 since dh_0 annihilates $G_{n_1} \supseteq G_{n_1-1}$. Let h_1 be the second function independent of h_0 and note that $dh_1 \cdot \mathrm{ad}^{n_1}_{f_0} f_1(x) := a_1(x) \neq 0$.

At the next step, the vector field $\mathrm{ad}^{n_1-1}_{f_0} f_1$ is removed from G_{n_1-1} to get the involutive distribution G_{n_1-2} of dimension $d_{n_1-2} = n - 3$. The one-forms dh_0 and dh_1 annihilate G_{n_1-2}. Using the property of Lie derivatives given in equation (2.7) of Chapter 2

and the fact that dh_1 annihilates G_{n_1-1}, we find $dL_{f_0}h_1$ annihilates G_{n_1-2} since

$$\begin{aligned} 0 &= dh_1 \cdot \mathrm{ad}_{f_0}^j f_k = L_{\mathrm{ad}_{f_0}^j f_k} h_1 = L_{[f_0, \mathrm{ad}_{f_0}^{j-1} f_k]} h_1 = L_{f_0} L_{\mathrm{ad}_{f_0}^{j-1} f_k} h_1 - L_{\mathrm{ad}_{f_0}^{j-1} f_k} L_{f_0} h_1 \\ &= -dL_{f_0}h_1 \cdot \mathrm{ad}_{f_0}^{j-1} f_k \quad 1 \le j \le n_1 - 1 \quad k = 1,2\,. \end{aligned} \tag{3.6}$$

Furthermore,

$$a_1(x) = dh_1 \cdot \mathrm{ad}_{f_0}^{n_1} f_1 = -dL_{f_0}h_1 \cdot \mathrm{ad}_{f_0}^{n_1-1} f_1\,.$$

This procedure continues to the distribution G_{n_2}, which is annihilated by dh_0, dh_1, $dL_{f_0}h_1$, $\ldots$, $dL_{f_0}^{n_1-n_2-1}h_1$.

Starting with the involutive distribution G_{n_2-1}, the distributions drop dimension by two. For G_{n_2-1}, the vector fields $\mathrm{ad}_{f_0}^{n_2} f_1$ and $\mathrm{ad}_{f_0}^{n_2} f_2$ are removed from G_{n_2}. The one-forms dh_0, dh_1, $dL_{f_0}h_1$, $\ldots$, $dL_{f_0}^{n_1-n_2-1}h_1$ annihilate $G_{n_2-1} \subseteq G_{n_2}$. In addition, equation (2.7) and the fact that $dL_{f_0}^{n_1-n_2-1}h_1$ annihilates G_{n_2} give

$$dL_{f_0}^{n_1-n_2}h_1 \cdot \mathrm{ad}_{f_0}^{j-n_1+n_2} f_k = 0 \quad 1 \le j \le n_1 - 1 \quad k = 1,2\,,$$

which shows $dL_{f_0}^{n_1-n_2}h_1$ annihilates G_{n_2-1}. There is one more function whose differential also annihilates G_{n_2-1}; we call it h_2 and note that $dh_2 \cdot \mathrm{ad}_{f_0}^{n_2} f_2(x) := a_2(x) \neq 0$.

At the next step, the vector fields $\mathrm{ad}_{f_0}^{n_2-1} f_1$ and $\mathrm{ad}_{f_0}^{n_2-1} f_2$ are removed from G_{n_2-1} to get the involutive distribution G_{n_2-2} of dimension $d_{n_2-2} = 2(n_2 - 1)$. The one-forms $dh_0, dh_1, dL_{f_0}h_1, \ldots, dL_{f_0}^{n_1-n_2}h_1, dh_2$ annihilate $G_{n_2-2} \subseteq G_{n_2-1}$. To see that the one-forms $dL_{f_0}^{n_1-n_2+1}h_1$ and $dL_{f_0}h_2$ also annihilate G_{n_2-2}, we use equation (2.7) and the fact that $dL_{f_0}^{n_1-n_2}h_1$ and dh_2 annihilate G_{n_2-1} to get

$$dL_{f_0}^{n_1-n_2+1}h_1 \cdot \mathrm{ad}_{f_0}^{j-n_1+n_2-1} f_k = 0 \quad 1 \le j \le n_1 - 1 \quad k = 1,2$$

and

$$0 = dh_2 \cdot \mathrm{ad}_{f_0}^j f_k = -dL_{f_0}h_2 \cdot \mathrm{ad}_{f_0}^{j-1} f_k \quad 1 \le j \le n_2 - 1 \quad k = 1,2\,.$$

Furthermore,

$$a_2(x) = dh_2 \cdot \mathrm{ad}_{f_0}^{n_2} f_2 = -dL_{f_0}h_2 \cdot \mathrm{ad}_{f_0}^{n_2-1} f_2\,.$$

This procedure continues to the distribution G_0, which is annihilated by dh_0, dh_1, $\ldots$, $dL_{f_0}^{n_1-1}h_1$, dh_2, $\ldots$, $dL_{f_0}^{n_2-1}h_2$.

In summary, the three functions h_0, h_1, and h_2 have been found such that

$$\begin{aligned} &dh_0 \perp G_j && 0 \le j \le n-1 \\ &dL_{f_0}^k h_1 \perp G_j && 0 \le j \le n_1 - 1, \quad 0 \le k \le n_1 - 1 - j \\ &dL_{f_0}^k h_2 \perp G_j && 0 \le j \le n_2 - 1, \quad 0 \le k \le n_2 - 1 - j \; . \end{aligned} \tag{3.7}$$

These three functions are used to define the chained form coordinates as follows

$$\begin{aligned} \xi_0 = h_0 \qquad \zeta_0 &= L_{f_0}^{n_1} h_1 & \eta_0 &= L_{f_0}^{n_2} h_2 \\ \zeta_1 &= L_{f_0}^{n_1-1} h_1 & \eta_1 &= L_{f_0}^{n_2-1} h_2 \\ &\vdots & &\vdots \\ &\vdots & \eta_{n_2-1} &= L_{f_0} h_2 \\ \zeta_{n_1-1} &= L_{f_0} h_1 & \eta_{n_2} &= h_2 \\ \zeta_{n_1} &= h_1 \; . \end{aligned} \tag{3.8}$$

To verify that the above coordinate transformation is valid, we show it is a local diffeomorphism. First the derivatives of the coordinate transformation, Φ, are calculated with respect to x. This is then multiplied on the right by a nonsingular matrix M whose columns are the n independent vector fields $f_0, f_1, \mathrm{ad}_{f_0} f_1, \ldots, \mathrm{ad}_{f_0}^{n_1} f_1, f_2, \mathrm{ad}_{f_0} f_2, \ldots, \mathrm{ad}_{f_0}^{n_2} f_2$:

$$\frac{\partial \Phi}{\partial x} \cdot M = \begin{bmatrix} dh_0 \\ dL_{f_0}^{n_1} h_1 \\ \vdots \\ dL_{f_0} h_1 \\ dh_1 \\ dL_{f_0}^{n_2} h_2 \\ \vdots \\ dL_{f_0} h_2 \\ dh_2 \end{bmatrix} \begin{bmatrix} f_0 & f_1 & \mathrm{ad}_{f_0} f_1 & \cdots & \mathrm{ad}_{f_0}^{n_1} f_1 & f_2 & \mathrm{ad}_{f_0} f_2 & \cdots & \mathrm{ad}_{f_0}^{n_2} f_2 \end{bmatrix} \tag{3.9}$$

$$= \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 \\ * & \pm a_1(x) & * & \cdots & \cdots & * & * & \cdots & \cdots & * \\ * & 0 & \pm a_1(x) & \ddots & & \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & * & \vdots & & \ddots & * \\ * & 0 & 0 & \cdots & 0 & a_1(x) & 0 & \cdots & \cdots & 0 \\ * & 0 & * & \cdots & \cdots & * & \pm a_2(x) & * & \cdots & * \\ \vdots & 0 & 0 & \ddots & & \vdots & 0 & \pm a_2(x) & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & * \\ * & 0 & 0 & \cdots & 0 & * & 0 & \cdots & 0 & a_2(x) \end{bmatrix}.$$

The functions $a_1(x)$ and $a_2(x)$ are nonzero by definition. Equation (3.7) is used to get the zeros in the matrix. It can be shown that the above matrix, under row operations, is similar to a nonsingular diagonal matrix with

$$1, \pm a_1(x), \ldots, a_1(x), \pm a_2(x), \ldots, a_2(x)$$

on the diagonal, and thus has full rank. Therefore the Jacobian matrix $\frac{\partial \Phi}{\partial x}$ must also be nonsingular locally, implying $(\xi, \zeta, \eta) = \Phi(x)$ is a local diffeomorphism and a valid coordinate transformation on the open set U by the inverse function theorem.

To compute the input transformation, we take derivatives of the transformed coordinates, cancelling terms by using the zero entries of the above matrix $\frac{\partial \Phi}{\partial x} \cdot M$:

$$\begin{aligned} \dot{\xi}_0 &= u_0 \\ \dot{\zeta}_0 &= L_{f_0}^{n_1+1} h_1 u_0 + L_{f_1} L_{f_0}^{n_1} h_1 u_1 + L_{f_2} L_{f_0}^{n_1} h_1 u_2 \\ \dot{\zeta}_1 &= L_{f_0}^{n_1} h_1 u_0 = \zeta_0 u_0 \\ &\vdots \\ \dot{\zeta}_{n_1} &= L_{f_0} h_1 u_0 = \zeta_{n_1-1} u_0 \\ \dot{\eta}_0 &= L_{f_0}^{n_2+1} h_2 u_0 + L_{f_2} L_{f_0}^{n_2} h_2 u_2 \\ \dot{\eta}_1 &= L_{f_0}^{n_2} h_2 u_0 = \eta_0 u_0 \\ &\vdots \\ \dot{\eta}_{n_2} &= L_{f_0} h_2 u_0 = \eta_{n_2-1} u_0 \, . \end{aligned}$$

Therefore the input transformation

$$\begin{aligned} v_0 &= u_0 \\ v_1 &= L_{f_0}^{n_1+1} h_1 u_0 + L_{f_1} L_{f_0}^{n_1} h_1 u_1 + L_{f_2} L_{f_0}^{n_1} h_1 u_2 \\ v_2 &= L_{f_0}^{n_2+1} h_2 u_0 + L_{f_2} L_{f_0}^{n_2} h_2 u_2 \end{aligned}$$

will result in the two-chain, single-generator chained form (3.3). □

Remark. The condition that $G_0, \ldots, G_{n-1}$ all be involutive is somewhat redundant, as in the exact linearization conditions in [28] (Section 5.2.6, page 256), since the involutivity of some distributions in the sequence may imply the involutivity of others.

3.1.2 Controllability of Chained Form Systems

In this section a system in two-chain, single-generator chained form is shown to be completely controllable. Since controllability is unaffected by state feedback and coordinate transformation, it will follow that the original system is also completely controllable.

Theorem 19 ***(Controllability of Two-chain, Single-generator Chained Form Systems)*** *The three-input, two-chain, single-generator chained form system in equation (3.3), where* $(\xi, \zeta, \eta) \in U \subset \mathbb{R}^n$ *and* $n = n_1 + n_2 + 3$, *is completely controllable.*

Proof. The chained form system equations are written as

$$\begin{pmatrix} \dot{\xi} \\ \dot{\zeta} \\ \dot{\eta} \end{pmatrix} = X_0(\xi, \zeta, \eta) v_0 + X_1(\xi, \zeta, \eta) v_1 + X_2(\xi, \zeta, \eta) v_2 \,,$$

where $(\xi, \zeta, \eta) = (\xi_0, \zeta_0, \ldots, \zeta_{n_1}, \eta_0, \ldots, \eta_{n_2})$ and the corresponding input vector fields are

$$\begin{aligned} X_0 &= \frac{\partial}{\partial \xi_0} + \sum_{i=1}^{n_1} \zeta_{i-1} \frac{\partial}{\partial \zeta_i} + \sum_{i=1}^{n_2} \eta_{i-1} \frac{\partial}{\partial \eta_i} \\ X_1 &= \frac{\partial}{\partial \zeta_0} \\ X_2 &= \frac{\partial}{\partial \eta_0} \,. \end{aligned}$$

Recall from Section 2.1 that a system of the form $\dot{z} = \sum_{i=0}^{m} X_i(z) v_i$ is completely controllable if the involutive closure of the distribution $\Delta_0 = \text{span}\{X_0, \ldots, X_m\}$ at each

configuration is equal to the entire state space $\mathbb{R}^n$. Define $\Delta_0 = \text{span}\{X_0, X_1, X_2\}$. The existence of n independent vector fields in the involutive closure, $\bar{\Delta}_0$, will imply complete controllability. Consider the n-dimensional distribution, a subset of $\bar{\Delta}_0$, resulting from taking successive Lie brackets with X_0:

$$\Delta = \text{span}\{X_0, X_1, \text{ad}_{X_0}X_1, \ldots, \text{ad}_{X_0}^{n_1}X_1, X_2, \text{ad}_{X_0}X_2, \ldots, \text{ad}_{X_0}^{n_2}X_2\}$$

$$= \text{span}\left\{ \begin{array}{ccccccccc} 1 & 0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \zeta_0 & 0 & 1 & \ddots & & & & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & & & & \vdots \\ \zeta_{n_1-1} & \vdots & & \ddots & 1 & \ddots & & & \vdots \\ 0 & \vdots & & & 0 & 1 & \ddots & & \vdots \\ \eta_0 & \vdots & & & \vdots & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\ \eta_{n_2-1} & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 & 1 \end{array} \right\}.$$

The columns are linearly independent vector fields for each $z \in U$. Therefore the system is completely controllable. □

Remark. It could have proven directly that the original system is completely controllable by defining Δ to be the span of the columns of the nonsingular matrix M in equation (3.9).

3.1.3 Multiple-input, Single-generator Chained Form Systems

The above results can be generalized to $(m+1)$-input nonholonomic systems.

Proposition 20 ***(Converting to m-chain, Single-generator Chained Form)*** *Consider the drift-free nonholonomic system*

$$\dot{x} = g_0(x)u_0 + \cdots + g_m(x)u_m \tag{3.10}$$

with smooth, linearly independent input vector fields g_j. There exists a feedback transformation on some open set $U \subset \mathbb{R}^n$

$$z = \Phi(x)$$
$$u = \beta(x)v$$

that transforms the system (3.10) *into m-chain, single-generator chained form*

$$
\begin{array}{lllllllll}
\dot{z}_0^0 = v_0 & \dot{z}_0^1 &=& v_1 & \dot{z}_0^2 &=& v_2 & \cdots & \dot{z}_0^m &=& v_m \\
& \dot{z}_1^1 &=& z_0^1 v_0 & \dot{z}_1^2 &=& z_0^2 v_0 & \cdots & \dot{z}_1^m &=& z_0^m v_0 \\
& \vdots & & & \vdots & & & & \vdots & & \\
& \dot{z}_{n_1}^1 &=& z_{n_1-1}^1 v_0 & \vdots & & & & \vdots & & \\
& & & & \dot{z}_{n_2}^2 &=& z_{n_2-1}^2 v_0 & & \vdots & & \\
& & & & & & & & \dot{z}_{n_m}^m &=& z_{n_m-1}^m v_0
\end{array}
\tag{3.11}
$$

if there exists a basis $f_0, \ldots, f_m$ *for* $\Delta_0 := span\{g_0, \ldots, g_m\}$ *that has the form*

$$
\begin{aligned}
f_0 &= \frac{\partial}{\partial x_1} + \sum_{i=2}^{n} f_0^i(x) \frac{\partial}{\partial x_i} \\
f_j &= \sum_{i=2}^{n} f_j^i(x) \frac{\partial}{\partial x_i} \quad 1 \le j \le m
\end{aligned}
$$

such that the distributions

$$
G_j = span\{ad_{f_0}^i f_1, \ldots, ad_{f_0}^i f_m : 0 \le i \le j\} \quad 0 \le j \le n-1
$$

have constant dimension on U, *are all involutive, and* G_{n-1} *has dimension* $n-1$ *on* U.

The proof follows the same method as in the proof of Theorem 18.

Finding the coordinate transformation to the $(m+1)$-input chained form uses the same method as for the three-input case. We first find $m+1$ smooth functions $h_0, \ldots, h_m$ and use the following formula:

$$
\begin{array}{lllllllll}
z_0^0 = h_0 & z_0^1 &=& L_{f_0}^{n_1} h_1 & z_0^2 &=& L_{f_0}^{n_2} h_2 & \cdots & z_0^m &=& L_{f_0}^{n_m} h_m \\
& z_1^1 &=& L_{f_0}^{n_1-1} h_1 & z_1^2 &=& L_{f_0}^{n_2-1} h_2 & \cdots & z_1^m &=& L_{f_0}^{n_m-1} h_m \\
& \vdots & & & \vdots & & & & \vdots & & \\
& z_{n_1-1}^1 &=& L_{f_0} h_1 & \vdots & & & & \vdots & & \\
& z_{n_1}^1 &=& h_1 & z_{n_2-1}^2 &=& L_{f_0} h_2 & & \vdots & & \\
& & & & z_{n_2}^2 &=& h_2 & & z_{n_m-1}^m &=& L_{f_0} h_m \\
& & & & & & & & z_{n_m}^m &=& h_m \, .
\end{array}
\tag{3.12}
$$

Then the $m+1$ chained form inputs are set to be

$$
v_i := \dot{z}_0^i \quad 0 \le i \le m \, .
$$

Proposition 21 ***(Controllability of m-chain, Single-generator Chained Form Systems)*** *The $(m+1)$-input, m-chain, single-generator chained form system in equation (3.11), where $z \in U \subset \mathbb{R}^n$ and $n = m + 1 + \sum_{j=1}^{m} n_j$, is completely controllable.*

The proof follows the same method as in the proof of Theorem 19.

3.2 Conversion to Extended Goursat Normal Form

In this section, a method is presented that only uses the constraints on the system to transform the kinematics of a mobile robot into extended Goursat normal form. The constraints can be written as $\alpha^i = 0$, where the α^i are one-forms on the robot's configuration space. In the previous section, nonholonomic systems were considered not from the point of view of their constraints, but rather from the point of view of a control system with the allowable motions in the span of the input vector fields. Sufficient conditions in terms of the vector fields were given for converting multiple-input control systems to chained form. The conditions presented in this section, however, are necessary and sufficient.

The dual to the chained form, in the sense of one-forms, is the Goursat normal form. Although the mathematics literature abounds with the theory of exterior differential systems (see [15, 25, 87] as a survey), only recently have there been attempts to apply this machinery to solve general control problems in steering nonholonomic systems.

Murray [52] first described the connection between the chained form and the Goursat normal form. Tilbury *et al.* [76] applied these results to show how to convert the system of a car with N trailers, a two-input control system or codimension two Pfaffian system, into Goursat normal form. The calculations in this context were simplified by the use of a coordinatization of the state space introduced by Sørdalen in [68].

In fact, the techniques of Sørdalen are a way of systematically converting mobile robot systems of N trailers into two-chained form by noticing that the trajectory of the (x, y) position of the last trailer determines the evolution of all the state variables of the system. Tilbury *et al.* [78] also use this technique for transforming the kinematic model of a multiple-steering, multiple-trailer mobile robot system to multiple-input chained form. The physical intuition about the system is used to identify the states that determine all of the trajectories of the system. These states become the bottoms of the chains of integrators in

the chained form (3.11)

$$z_0^0(t), z_{n_1}^1(t), \ldots, z_{n_m}^m(t) \ ,$$

and the rest of the coordinate transformation is found through differentiation

$$z_i^j = \frac{\dot{z}_{i+1}^j}{\dot{z}_0^0} \quad 1 \leq j \leq m, \quad 0 \leq i \leq n_j - 1 \ . \tag{3.13}$$

In related work by Fliess *et al.* [23, 64], the idea that certain variables determine the entire state of the system has been formalized in a more general setting, and these system variables are referred to as *flat outputs.* The formal definition of flatness is given in the language of differential algebra and will not be discussed here. Informally [64], a set of outputs $y = h(x, u)$ with $x \in \mathbb{R}^n$ and $u, y \in \mathbb{R}^m$ is said to be (differentially) *flat* for the meromorphic[1] system $\dot{x} = f(x, u)$ if all of the system variables (states and inputs) are differentiable functions of the outputs y, i.e., x and u are meromorphic functions of the outputs y and finitely many of their derivatives. Intuitively, the flat outputs are outputs with respect to which the system has no zero dynamics in the sense of nonlinear systems [28]. A system is called *differentially flat* if a set of flat outputs can be found. Moreover, there may be many choices for the flat, or linearizing, outputs. The multiple-input chained form of (3.11) is differentially flat with flat outputs $z_0^0, z_{n_1}^1, \ldots, z_{n_m}^m$, although chained form systems with more than one generator are not, in general, flat.

In this section tools from exterior differential systems (refer to Section 2.2) are used to convert Pfaffian systems of codimension *greater than two* into *extended* Goursat normal form. Recall that a Pfaffian system of k one-forms (constraints) on a state space $\mathbb{R}^n$ is said to have codimension $n - k$. We refer to Theorem 16 from Section 2.2, which states the necessary and sufficient conditions for the existence of a coordinate transformation to extended Goursat normal form. If one cannot constructively find these coordinates, however, the theorem is not useful for nonholonomic motion planning since the steering algorithms use the transformed coordinates. A constructive algorithm for finding the extended Goursat normal form coordinates is proposed in this section and modified in Section 5.4, in the context of an example, to include the method of partial prolongations.

If the constraints of the Pfaffian system

$$I \ = \ \{\omega_1^j, \omega_2^j, \ldots, \omega_{n_j}^j : j = 1, \ldots, m\} \tag{3.14}$$

[1]The proof of flatness in [64] requires that the space for x be a field, not a ring, which can only be guaranteed if $f(x, u)$ is a meromorphic function.

satisfy the extended Goursat congruences

$$\begin{aligned} d\omega_k^j &\equiv \pi \wedge \omega_{k+1}^j \mod I^{(n_j-k)}\ ,\ k=1,\ldots,n_j-1 \\ d\omega_{n_j}^j &\not\equiv 0 \mod I \end{aligned} \tag{3.15}$$

for $j = 1,\ldots,m$, then, by Theorem 16, there exist coordinates for the extended Goursat normal form

$$I = \{dz_{n_j}^j - z_{n_j-1}^j dz_0^0,\ \ldots,\ dz_1^j - z_0^j dz_0^0\ :\ j=1,\ldots,m\}\ . \tag{3.16}$$

In this section, a purely algebraic algorithm is presented that finds the coordinates. The algorithm is similar to the one used in [24] for linearization to Brunovsky normal form.

For codimension two systems, or systems with only one tower, $I = \{\omega_1^1, \omega_2^1, \ldots, \omega_{n_1}^1\}$ with $n_1+2 = n$, the transformation is straightforward: the generator coordinate, z_0^0 and the coordinate for the bottom of the tower, z_{n_1} are found from the solution to Pfaff's problem (Theorem 11, Section 2.2), then the rest of the coordinates are found through differentiation by equation (3.13). This is summarized in Algorithm 1 from Section 2.2 and presented in detail in [76].

The difficulty with having more than one tower is that the constraints must be modified to decouple the towers so that the solution to Pfaff's problem can be used to find the coordinates for each tower. The following discussion is restricted to the case of the first tower being the longest tower so that if a π is found that satisfies the extended Goursat congruences, an integrable π' can be found that satisfies these congruences by using Lemma 17 from Section 2.2. For the case $n_1 = n_2$, or when at least two towers have the longest length, if a π is found that satisfies the extended Goursat congruences, we may not be able to construct an integrable π'. There is no restriction on the lengths $n_2, n_3, \ldots, n_m$.

Algorithm 2 ***(Conversion to Extended Goursat Normal Form)***

Step 1 *For the Pfaffian system*

$$I = \{\alpha^0, \alpha^1, \ldots, \alpha^{n-m-2}\} \tag{3.17}$$

on $\mathbb{R}^n$, *compute the derived flag*

$$\begin{aligned} I^{(0)} = I &= \{\alpha^0, \alpha^1, \ldots, \alpha^{n-m-2}\} \\ I^{(i)} &= \{\lambda \in I^{(i-1)} : d\lambda \equiv 0 \mod I^{(i-1)}\} \end{aligned}$$

for $i = 1, \ldots, N$, where N is the step in which this procedure terminates, i.e., $I^{(N+1)} = I^{(N)}$. The one-forms α^i come from the kinematic constraints of the system.

Step 2 *Construct m towers (3.14) using an integrable π such that the Goursat congruences (3.15) are satisfied.*

Step 3 *From the Goursat congruence*

$$d\omega_1^1 = \pi \wedge \omega_2^1 \mod I^{(n_1-1)} ,$$

the rank of ω_1^1 is one since $d\omega_1^1 \wedge \omega_1^1 \neq 0$ and $(d\omega_1^1)^2 \wedge \omega_1^1 = 0$. Therefore, use the solution to Pfaff's problem (Theorem 11) to compute the coordinates for the first tower of the normal form as follows:

(i) Define $df_1 := \pi$. This satisfies the first Pfaff equation

$$d\omega_1^1 \wedge \omega_1^1 \wedge df_1 = 0 \quad \textit{and} \quad \omega_1^1 \wedge df_1 \neq 0 .$$

Find a function f_2 satisfying the second Pfaff equation

$$\omega_1^1 \wedge df_1 \wedge df_2 = 0 \quad \textit{and} \quad df_1 \wedge df_2 \neq 0 . \tag{3.18}$$

(ii) Define $z_0^0 := f_1$ as the generator coordinate and $z_{n_1}^1 := f_2$ as the coordinate corresponding to the bottom of the first chain.

(iii) Since ω_1^1 satisfies the Pfaff equation (3.18), it can be modified as

$$\omega_1^1 = b(q)\ df_2 - a(q)\ dz_0^0$$

for some smooth functions $a(q)$ and $b(q)$, where $q \in \mathbb{R}^n$ is the total state of the system. $b(q) \neq 0$ for if it were, $d\omega_1^1 = dz_0^0 \wedge da(q) = 0 \mod \omega_1^1$, which contradicts the Goursat congruences (3.15). Therefore, this constraint can be modified as

$$\bar{\omega}_1^1 = df_2 - \frac{a(q)}{b(q)}\ dz_0^0 .$$

Define the coordinate $z_{n_1-1}^1$ to be $a(q)/b(q)$.

(iv) Modify ω_i^1 for $i = 2, \ldots, n_1$ by using the form constraints in $I^{(n_1-i)}$ from the Goursat congruence equation (3.15) to satisfy

$$\bar{\omega}_i^1 = dz_{n_1-i+1}^1 - c_{n_1-i}^1(q)\ dz_0^0$$

and define $z_{n_1-i}^1$ to be $c_{n_1-i}^1$. This is a purely algebraic step that gives all the coordinates for the first tower.

Step 4 *Compute the coordinates for the second tower as follows:*

(i) Modify ω_1^2 *by using the form constraints in* $I^{(n_2-1)}$ *from the Goursat congruence equation (3.15) to satisfy the equation*

$$\bar{\omega}_1^2 = dh(q) - c(q)\, dz_0^0 \,, \tag{3.19}$$

where $h(q)$ *and* $c(q)$ *are smooth functions. The rank of* $\bar{\omega}_1^2$ *is one since* $d\bar{\omega}_1^2 \wedge \bar{\omega}_1^2 \neq 0$ *and* $(d\bar{\omega}_1^2)^2 \wedge \bar{\omega}_1^2 = 0$.

(ii) This satisfies the first Pfaff equation with $g_1 = f_1$,

$$d\bar{\omega}_1^2 \wedge \bar{\omega}_1^2 \wedge dg_1 = 0 \quad \textit{and} \quad \bar{\omega}_1^2 \wedge dg_1 \neq 0 \,.$$

The second Pfaff equation is satisfied with $g_2 = h$ *from equation (3.19),*

$$\bar{\omega}_1^2 \wedge dg_1 \wedge dg_2 = 0 \quad \textit{and} \quad dg_1 \wedge dg_2 \neq 0 \,.$$

(iii) Define $z_{n_2}^2 := g_2 = h$ *as the coordinate corresponding to the bottom of the second chain.*

(iv) From equation (3.19), define the coordinate $c(q)$ *to be* $z_{n_2-1}^2$.

(v) Modify ω_i^2 *for* $i = 2, \ldots, n_2$ *by using the form constraints in* $I^{(n_2-i)}$ *from the Goursat congruence equation (3.15) to satisfy*

$$\bar{\omega}_i^2 = dz_{n_2-i+1}^2 - c_{n_2-i}^2(q)\, dz_0^0$$

and define $c_{n_2-i}^2$ *to be* $z_{n_2-i}^2$. *This is a purely algebraic step that gives the coordinates of the second tower.*

Step 5 *Compute the coordinates for the other towers using the same method as for the second tower.*

The above algorithm constructs coordinates $\{z_i^j\}$ such that

$$\begin{aligned} I &= \{\omega_1^j, \ldots, \omega_{n_j}^j \,:\, j = 1, \ldots, m\} \\ &= \{dz_{n_1}^1 - z_{n_1-1}^1\, dz_0^0, \ldots,\ dz_1^1 - z_0^1\, dz_0^0, \ldots,\ dz_{n_m}^m - z_{n_m-1}^m\, dz_0^0, \ldots,\ dz_1^m - z_0^m\, dz_0^0\}, \end{aligned}$$

which is the dual of the multiple-input, single-generator chained form in equation (3.11).

In the preceding algorithm, there is flexibility in the choice of $\bar{\omega}_1^j$. More precisely, an "integrating factor", $\gamma(q)$, needs to be chosen such that

$$\gamma(q)\ \bar{\omega}_1^j = dz_{n_j}^j - z_{n_j-1}^j\ dz_0^0\ . \tag{3.20}$$

In practice, it is difficult to choose $\gamma(q)$ appropriately to solve for the $z_{n_j-1}^j$ coordinate. Therefore, a method using partial prolongations of the exterior differential system that more readily yields extended Goursat normal form coordinates is developed Section 5.4 in the context of an example. The basic idea is to use partial prolongations to add more one-forms to some of the towers in the system so that it is easier to find an integrating factor for equation (3.20) and to solve for the coordinates. These one-forms are the constraints of "virtual axles" strategically added onto the multiple-steering, multiple-trailer system. The example considered only has two towers in its derived flag, but the procedure developed in Section 5.4 easily extends to general m tower systems.

In summary, this chapter has presented two methods for transforming a multiple-input drift-free nonholonomic system into chained form and extended Goursat normal form. The first method gave sufficient conditions for transforming the kinematic system to a multiple-chain, single-generator chained form using a coordinate transformation and state feedback. In this special form, the system was shown to be completely controllable. The second method presented an algorithm with necessary and sufficient conditions to find the extended Goursat normal form coordinates. The algorithm used the fact that the one-form π that satisfies the extended Goursat congruences is integrable. Once the system is in extended Goursat normal form or multiple-input chained form, there are many methods available for steering and stabilization of such systems, which are the topics presented in the next chapter.

Chapter 4

Steering and Stabilization

In this chapter, different open-loop methods for steering wheeled nonholonomic systems in chained form between two given configurations are presented. The first algorithm uses sinusoidal control inputs; steering one level in the chains at a time in a step-by-step fashion, or steering all levels in the chains at once. The steering method using polynomial control inputs was presented in detail in [57, 78, 76] and will be briefly mentioned here. The method of steering nonholonomic systems using piecewise constant inputs was first introduced in [49] as multirate digital control. The basic idea behind each of the steering methods is to parameterize the input space with at least as many parameters as there are states, integrate the chained form equations symbolically, and then solve for the input parameters in terms of the desired initial and final states.

As mentioned in the introductory chapter, we make no attempt to find the "optimal" control inputs since the criteria for optimality may change with the different mobile robot systems considered. We have found, however, that although all of the steering algorithms will find a path between any two given configurations, the resulting trajectories look "nicer" for some methods than for others. Using the step-by-step sinusoidal method is not recommended in practice, but only included here to show the ease of steering at each level in the chained form. The all-at-once sinusoidal method is best used for trajectories that have a reversal, such as parallel parking. Polynomial and piecewise constant control inputs work better for trajectories without reversals.

The control problem considered in this chapter is stated as follows: given a system

of equations in extended Goursat normal form

$$\begin{aligned} I \;=\; \{&dz^1_{n_1} - z^1_{n_1-1}\,dz^0_0,\; dz^1_{n_1-1} - z^1_{n_1-2}\,dz^0_0, \dots,\; dz^1_1 - z^1_0\,dz^0_0, \\ &\dots,\; dz^m_{n_m} - z^m_{n_m-1}\,dz^0_0,\; dz^m_{n_m-1} - z^m_{n_m-2}\,dz^0_0, \dots,\; dz^m_1 - z^m_0\,dz^0_0\}, \end{aligned} \tag{4.1}$$

or chained form

$$\begin{array}{lllllllll} \dot{z}^0_0 = v_0 & \dot{z}^1_0 &=& v_1 & \dot{z}^2_0 &=& v_2 & \cdots & \dot{z}^m_0 &=& v_m \\ & \dot{z}^1_1 &=& z^1_0 v_0 & \dot{z}^2_1 &=& z^2_0 v_0 & & \dot{z}^m_1 &=& z^m_0 v_0 \\ & \vdots & & & \vdots & & & & \vdots & & \\ & \dot{z}^1_{n_1} &=& z^1_{n_1-1} v_0 & \vdots & & & & \vdots & & \\ & & & & \dot{z}^2_{n_2} &=& z^2_{n_2-1} v_0 & \ddots & \vdots & & \\ & & & & & & & & \dot{z}^m_{n_m} &=& z^m_{n_m-1} v_0\,, \end{array} \tag{4.2}$$

a desired initial state $\xi^0 := (z^0, z^1, \dots, z^m)^0$ and final state $\xi^f := (z^0, z^1, \dots, z^m)^f$, and a time $T > 0$, find inputs $\{v_i(t) :\; t \in [0,T],\; 0 \le i \le m\}$ that will steer the system from the initial state to the final state in $[0,T]$.

Various approaches for feedback stabilization of chained form systems are also briefly mentioned. Although most of the work in this area has concentrated on two-input systems, the decoupled form of the multiple-input chained form system allows the techniques to be generalized.

4.1 Steering with Sinusoidal Inputs

In this section, two algorithms are presented for steering a system in chained form from a given initial configuration to a desired final configuration. Both algorithms use sinusoidal inputs, but the first algorithm steers the system by steering one level in the chained form at a time, and the second algorithm steers all of the states in the chained form at the same time. Steering chained form systems with sinusoids was introduced by Murray and Sastry [55]. We have found steering with the step-by-step sinusoidal method not to be practical in the sense that it produces trajectories with more reversals than are minimally needed, and recommend replacing it by the all-at-once sinusoidal method or one of the other two methods presented in this chapter.

As was stated in the introductory chapter, using sinusoids to steer chained form systems is optimal in the sense that the "input effort" (maximum input value) is minimized.

Consider the optimal control problem for the chained form system

$$\dot{\xi} = B(\xi)\, v$$

where $\xi : \mathbb{R}^n \mapsto \mathbb{R}^n$, $v : \mathbb{R} \mapsto \mathbb{R}^{m+1}$, and $B(\xi) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times(m+1)}$ that is written as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\int_0^1 |v|^2 dt \\ \text{subject to} & \xi(0) = \xi^0 \quad \text{and} \quad \xi(1) = \xi^f \ . \end{array}$$

The solution to this problem is periodic: sinusoidal for $n = 3$ and elliptic for $n > 3$. For example, if we consider the system in $\mathbb{R}^3$

$$\begin{aligned} \dot{\xi}_1 = v_0 \qquad \dot{\xi}_2 &= v_1 \\ \dot{\xi}_3 &= \xi_2 v_0 \end{aligned}$$

with $\xi(0) = \xi^0$ and $\xi(1) = \xi^f$, the optimal control problem becomes

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\int_0^1 (\dot{\xi}_1^2 + \dot{\xi}_2^2) dt \\ \text{subject to} & \xi(0) = \xi^0 \ , \ \xi(1) = \xi^f \ , \ \text{and } \dot{\xi}_3 - \xi_2 v_0 = 0 \ . \end{array}$$

By the Calculus of Variations, the Lagrangian is

$$L(\xi, \dot{\xi}) = \frac{1}{2}(\dot{\xi}_1^2 + \dot{\xi}_2^2) + \lambda(\dot{\xi}_3 - \xi_2 \dot{\xi}_1) \ .$$

Solving the Euler-Lagrange equation

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\xi}} - \frac{\partial L}{\partial \xi} = 0$$

gives

$$\begin{pmatrix} \dot{v}_0 \\ \dot{v}_1 \end{pmatrix} = \begin{pmatrix} 0 & \lambda \\ -\lambda & 0 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} = \Lambda \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} ,$$

where Λ is a constant skew-symmetric matrix. Solving for $v = (v_0, v_1)$ gives

$$v(t) = e^{\Lambda t} v(0) \ .$$

By Rodrigues' formula, with $\Lambda = \lambda S$ where S is unit skew matrix,

$$e^{\lambda S} = I + S \sin\lambda + S^2 \ (1 - \cos\lambda) \ .$$

This gives

$$e^{\Lambda t} = \cos\lambda t I + \frac{\sin\lambda t}{\lambda t}\Lambda \ ,$$

which shows that the inputs are sinusoids. If the additional assumption is made that $t = 1$, $\xi^0 = (0,0,0)$, and $\xi^f = (0,0,a)$ for some constant a, then $\lambda = 2\pi n$ and $e^{\Lambda} = I$, showing that the inputs are sinusoids at integrally related frequencies.

The above shows the existence of sinusoids as inputs to steer systems in chained form for the case of $x \in \mathbb{R}^3$ when the cost is the input effort. This does not mean that sinusoids are the best practical method for steering these systems, as will be seen in the examples in Chapter 5.

For the first algorithm, consider the system in the chained form (4.2) with the lengths of the chains such that $n_i \leq n_{i+1}$. This step-by-step sinusoidal algorithm exploits the decoupling of the chains, allowing for simultaneous steering of each level. The main idea, considering for a moment only the first chain with z^1, is that if $v_0 = \alpha \sin \omega t$ and $v_1 = \beta \cos \ell\omega t$, then $\dot{z}_0^1$ will have a frequency component at $\ell\omega$, $\dot{z}_1^1$ will have a frequency component at $(\ell - 1)\omega$, ..., and $\dot{z}_\ell^1$ will have a frequency component at zero. By simple integration over one period, this yields net movement in z_ℓ^1 while $z_0^1, \ldots, z_{\ell-1}^1$ return to their previous values. Thus, at the ℓ^{th} step in the algorithm, the states at the ℓ^{th} level in the chain are driven to their final positions.

The following algorithm is an extension of the algorithm for two-input systems in Murray and Sastry [56].

Algorithm 3 ***(Step-by-step Steering with Sinusoids for Multiple-input Systems)***

Step 0 *Steer the top-level coordinates, $\{z_0^j,\ j = 0, \ldots, m\}$ by choosing constant values for $v_0, v_1, \ldots, v_m$ on the time interval $[0, T)$:*

$$\begin{aligned} v_0 &= \frac{1}{T}((z_0^0)^f - (z_0^0)^0) \\ v_1 &= \frac{1}{T}((z_0^1)^f - (z_0^1)^0) \\ &\vdots \\ v_m &= \frac{1}{T}((z_0^m)^f - (z_0^m)^0)\ . \end{aligned}$$

Step 1 *Steer the coordinates at the first level down by choosing a sinusoid on v_0 and out-*

of-phase sinusoids on the other inputs with $\omega = \frac{2\pi}{T}$ *over the time interval* $[T, 2T)$*:*

$$\begin{aligned} v_0 &= \alpha \sin \omega t \\ v_1 &= \beta \cos \omega t \\ v_2 &= \gamma \cos \omega t \\ &\vdots \\ v_m &= \nu \cos \omega t \ . \end{aligned}$$

Choose $\alpha, \beta, \ldots, \nu$ *such that*

$$\begin{aligned} (z_1^1)^f - z_1^1(T) &= \frac{\alpha\beta}{2\omega} T \\ &\vdots \\ (z_1^m)^f - z_1^m(T) &= \frac{\alpha\nu}{2\omega} T \ , \end{aligned}$$

which causes the states $\{z_1^j, \ j = 1, \ldots, m\}$ *to reach their final values at time* $2T$*.*

Step k *(for* $k = 2, \ldots, n_m$*) Steer the coordinates at level* k *from the top. If* $n_i < k \leq n_{i+1}$*, then only chains* $i+1, \ldots, m$ *will be affected. A single frequency sinusoid is used for the first input, while multiple frequency sinusoids are used for the other inputs with* $\omega = \frac{2\pi}{T}$ *over the time interval* $[kT, (k+1)T)$*:*

$$\begin{aligned} v_0 &= \alpha \sin \omega t \\ v_1 &= 0 \\ &\vdots \\ v_i &= 0 \\ v_{i+1} &= \zeta \cos k\omega t \\ &\vdots \\ v_m &= \nu \cos k\omega t \ . \end{aligned}$$

Choose $\zeta, \ldots, \nu$ *such that*

$$\begin{aligned} (z_k^{i+1})^f - z_k^{i+1}(kT) &= \frac{\alpha^k \zeta}{(2\omega)^k k!} T \\ &\vdots \\ (z_k^m)^f - z_k^m(kT) &= \frac{\alpha^k \nu}{(2\omega)^k k!} T \ , \end{aligned}$$

which causes the states $\{z_k^j,\ j = i+1, \ldots, m\}$ *to reach their desired final values at time* $(k+1)T$.

After each step k, the states closer to the top of the chain than level k will have returned to the same values they reached after the previous step $(k-1)$. The states lower in the chain than level k will move as a result of the inputs at step k by some amount. This movement is ignored since those states are steered to their desired final values in subsequent iterations.

Although this method works well, it can be tedious in practice because of the many steps that are needed. In addition, the trajectories that are generated consist of many segments and do not always follow a direct path between the start and goal configurations.

In [76] an "all-at-once" sinusoidal method was proposed for the two-input case; we extended it here to multiple-input systems. In this method, only one step is used with all of the necessary frequencies set in the inputs:

$$\begin{aligned} v_0 &= a_0 + a_1 \sin \omega t \\ v_1 &= b_0 + b_1 \cos \omega t + \cdots + b_{n_1} \cos n_1 \omega t \\ &\vdots \\ v_m &= \nu_0 + \nu_1 \cos \omega t + \cdots + \nu_{n_m} \cos n_m \omega t \ . \end{aligned} \tag{4.3}$$

The existence of the parameters $a_0, a_1, b_0, \ldots, b_{n_1}, \ldots, \nu_0, \ldots, \nu_{n_m}$ is stated in the following proposition, which was proven for single-chain systems in [76]. The main idea of the proof of the proposition is to symbolically integrate the chained form equations (4.2) with the all-at-once sinusoidal inputs (4.3) to get $\xi(t) := (z^0(t), z^1(t), \ldots, z^m(t))$, which are functions of the initial state and input parameters. If $\xi(T)$ is evaluated with $T = 2\pi/\omega$, all the sinusoidal functions integrate to 1 or 0. Setting $\xi(T)$ to be the given final state ξ^f, gives a set of n polynomials in the $n+1$ input parameters $a_0, a_1, b_0, \ldots, b_{n_1}, \ldots, \nu_0, \ldots, \nu_{n_m}$. This proposition guarantees local existence of solutions to these equations.

Proposition 22 ***(Steering with All-at-once Sinusoids)*** *Consider the multiple-input chained system in equation (4.2) with initial and final states such that* $|\xi^0 - \xi^f| < \delta$ *for some* $\delta > 0$ *sufficiently small. Then there exist input parameters* $a_0, a_1, b_0, \ldots, b_{n_1}, \ldots, \nu_0, \ldots, \nu_{n_m}$ *such that the inputs in equation (4.3) steer the system from the initial to final state in time* $T = 2\pi/\omega$.

Proof. Without loss of generality, consider the case where the number of chains is $m = 2$. Let $\xi(t) := (z_0^0(t), z_0^1(t), \ldots, z_{n_1}^1(t), z_0^2(t), \ldots, z_{n_2}^2(t))$ and define the following map for $n = m + 1 + \sum_{j=1}^{m} n_j$:

$$\begin{aligned} \phi : \mathbb{R}^n &\mapsto \mathbb{R}^n \\ (a_0, b_0, \ldots, b_{n_1}, c_0, \ldots, c_{n_2}) &\mapsto \xi^f \ . \end{aligned}$$

Define $\phi(a_0, b_0, \ldots, b_{n_1}, c_0, \ldots, c_{n_2})$ to be the value of $\xi(T)$ when the chained form system (4.2) with $m = 2$ is integrated over $[0, T]$, starting at the given initial state ξ^0 with the inputs (4.3). We choose $a_1 \neq 0$ and show that ϕ is a local diffeomorphism.

Let $\{e_i\}_{i=1}^n$ be the standard basis for $\mathbb{R}^n$ and let ϵ be small. With the input parameterized by ϵe_1,

$$v_0 = \epsilon + a_1 \sin \omega t \qquad v_1 = 0 \qquad v_2 = 0 \ ,$$

the chained form equations are integrated and evaluated at T to give

$$\phi(\epsilon e_1) = \xi^0 + (\epsilon T; 0, o(\epsilon), \ldots, o(\epsilon); 0, o(\epsilon), \ldots, o(\epsilon)) \ ,$$

where $o(\epsilon)$ represents terms of linear and higher order in ϵ. For $k = 2, \ldots, n_1 + 2$ with an input parameterized by ϵe_k,

$$v_0 = a_1 \sin \omega t \qquad v_1 = \epsilon \cos((k-2)\omega t) \qquad v_2 = 0 \ ,$$

the chained form equations are integrated and evaluated at T to give

$$\phi(\epsilon e_k) = \xi^0 + (0; 0, \ldots, 0, p_k(\epsilon), o(\epsilon), \ldots, o(\epsilon); 0, \ldots, 0)$$

with

$$p_k(\epsilon) = \frac{a_1^{k-2}\, T}{(k-2)!(2\omega)^{k-2}} \ . \tag{4.4}$$

Similarly, for $k = n_1 + 3, \ldots, n_1 + n_2 + 3$ with an input parameterized by ϵe_k,

$$v_0 = a_1 \sin \omega t \qquad v_1 = 0 \qquad v_2 = \epsilon \cos((j-2)\omega t) \quad j = k - n_1 - 1 \ ,$$

the chained form equations are integrated and evaluated at T to give

$$\phi(\epsilon e_k) = \xi^0 + (0; 0, \ldots, 0; 0, \ldots, 0, p_j(\epsilon), o(\epsilon), \ldots, o(\epsilon))$$

with $p_j(\epsilon)$ defined as in (4.4). These n_1+n_2+3 directional derivatives are linearly independent, implying that the Jacobian of ϕ is nonsingular. Therefore, ϕ is a local diffeomorphism and the parameters for the inputs (4.3) can be found by selecting a value for a_1 and taking the inverse transformation $\phi^{-1}(\xi^f)$. In practice, the final state is not within a δ ball around the initial state. In this case, the two given states are connected by a finite number of δ balls, and the above method is applied within each ball. It is not clear how to apply this method when the transformation to chained form has singularities since in the original coordinates, the parameter δ may be a function of x. Extending this proof to the case of m chains is tedious, but straightforward since the chains are decoupled. □

Both of the sinusoidal methods require one more parameter than state and this parameter, α in Algorithm 3 and a_1 in Proposition 22, is the magnitude of the first input v_0. The main drawback to this approach is that there will be some interference between the levels, although not between chains, which requires solving nonlinear algebraic equations for the input parameters.

4.2 Steering with Polynomial Inputs

In this section, the method of steering multiple-input chained form systems in equation (4.2) with polynomial inputs is presented. This method was introduced in [76] for two-input systems and extended to multiple-input systems in [78].

In this method, the first input, v_0, is constant over the entire trajectory and the other inputs are Taylor polynomials

$$\begin{aligned} v_0 &= 1 \\ v_1 &= b_0 + b_1 t + \cdots + b_{n_1} t^{n_1} \\ v_2 &= c_0 + c_1 t + \cdots + c_{n_2} t^{n_2} \\ &\vdots \\ v_m &= \nu_0 + \nu_1 t + \cdots + \nu_{n_m} t^{n_m} \end{aligned} \tag{4.5}$$

with the number of parameters on each input chosen to be equal to the number of states in its chain. The time needed to steer the system is determined from the change in the z_0^0 coordinate,

$$T = (z_0^0)^f - (z_0^0)^0. \tag{4.6}$$

Let $\xi(t) := (z_0^0(t), z^1(t), \ldots, z^m(t))$. Integrating the chained form equations (4.2) with inputs as in (4.5), initial condition $\xi(0) = \xi^0$, and evaluating at time T, the parameters $b_0, \ldots, b_{n_1}, c_0, \ldots, c_{n_2}, \ldots, \nu_0, \ldots, \nu_{n_m}$ can be found in terms of the initial and final states from setting $\xi(T) = \xi^f$. Since the chains are decoupled, each chain's parameters can be independently found from the equations

$$\begin{aligned} M_b(T)\begin{bmatrix} b_0 \\ \vdots \\ b_{n_1} \end{bmatrix} + f(z^1(0), T) &= \begin{bmatrix} (z_0^1)^f \\ \vdots \\ (z_{n_1}^1)^f \end{bmatrix} \\ &\vdots \\ M_\nu(T)\begin{bmatrix} \nu_0 \\ \vdots \\ \nu_{n_m} \end{bmatrix} + f(z^m(0), T) &= \begin{bmatrix} (z_0^m)^f \\ \vdots \\ (z_{n_m}^m)^f \end{bmatrix}. \end{aligned} \tag{4.7}$$

The entries of the matrices have the form

$$M_{ij} = \frac{(j-1)!\, T^{i+j-1}}{(i+j-1)!},$$

showing that the matrices are nonsingular for $T \neq 0$. Equations (4.7) are easily solved for the parameters $b_i, c_i, \ldots, \nu_i$ by inverting the matrices, which are linear in the parameters.

A word of caution is needed if by chance equation (4.6) yields a time $T = 0$. In this case, this method will not work. This corresponds in a mobile robot system to the "parallel-parking" maneuver. An easy way to remedy this situation is to pick an intermediate point, with z_0^0 not equal to the given $(z_0^0)^0$, $(z_0^0)^f$, and then plan the path in two pieces. For the parallel-parking trajectory, the intermediate point is chosen to be that point where the vehicle would change directions to start backing up.

4.3 Steering with Piecewise Constant Inputs

In this section, a piecewise constant steering method is presented that steers the chained form system in equation (4.2). The method of steering nonholonomic systems using multirate digital control was first introduced by Monaco and Normand-Cyrot [49]. If the system can be discretized exactly, this method will generate exact point-to-point trajectories. For our purposes, these controls are interpreted as piecewise constant inputs. An algorithm using piecewise constant inputs for steering mobile robots was given in [75] in

the context of the fire truck example, and in [78] for steering a multiple-steering, multiple-trailer vehicle.

As in the method with polynomial inputs, the first input v_0 in equation (4.2) is chosen to be constant over the entire trajectory and the other m inputs are parameterized. Having v_0 constant over the whole trajectory guarantees the linearity of the equations that need to be solved for the other input parameters. The other inputs are chosen to be piecewise constant. For the first input v_1, for example, the time interval is divided into $n_1 + 1$ intervals

$$0 \leq t_1 \leq \cdots \leq t_{n_1+1}$$

and v_1 is set to a constant value over each $[t_k, t_{k+1})$. To make sure that the resulting equations have a solution, each input should switch constant values at least as many times as there are states in its chain. For this discussion, n_i is assumed to be $\leq n_{i+1}$ in equation (4.2) so that the m^{th} input will have the largest number of switches.

Let the time for the trajectory be denoted as T. The first input is chosen to be constant over the entire trajectory:

$$v_0(t) = v_0^D \quad \text{for } t \in [0, T] \,,$$

where v_0^D is chosen such that the first chained form state z_0^0 will go from its initial to its final position over the time period, i.e.,

$$v_0^D = \frac{1}{T}\left((z_0^0)^f - (z_0^0)^i\right) \,. \tag{4.8}$$

The other inputs are chosen to be piecewise constant as follows. Let the switching times be chosen as

$$0 = t_0^j < t_1^j < \cdots < t_{n_j+1}^j = T \quad 1 \leq j \leq m \,.$$

There are $n_j + 1$ switching times for each input $v_1, \ldots, v_m$ since there are $n_j + 1$ states in the j^{th} chain. Many different methods are available for choosing these times; we choose them so that for the m^{th} input (with the most switching times) the holding times between switches will be equal. The switching times for the other inputs are then chosen to be some subset of the switching times for the m^{th} input. Therefore, the inputs are of the form

$$v_j(t) = v_{j,k}^D \quad \text{for } t \in [t_k, t_{k+1}) \quad 1 \leq j \leq m.$$

These inputs are substituted into equation (4.2) and the equations are integrated using a symbolic manipulation software package. The final state can be expressed in terms of the inputs and the initial state as

$$\begin{bmatrix} z_0^j \\ z_1^j \\ \vdots \\ z_{n_j}^j \end{bmatrix} (T) = M_j(v_0^D, z^j(0)) \begin{bmatrix} v_{j,0}^D \\ v_{j,1}^D \\ \vdots \\ v_{j,n_j}^D \end{bmatrix}$$

where the matrices M_j are assured to be nonsingular whenever the first input v_0^D is nonzero [49]. As in steering with polynomial inputs, if the first input is zero from equation (4.8), then a slight modification of this method is necessary. One can either add a piecewise constant input to v_0 using at least two time periods, or an intermediate point can be chosen and the path can be planned as two separate trajectories.

4.4 Stabilization of Multiple-input Chained Form Systems

In this section, some methods from the literature for stabilization of chained form systems are briefly discussed. These systems are open-loop controllable, as shown in the previous sections by the various point-to-point steering algorithms, but are not stabilizable to a point by pure smooth static-state feedback (see Brockett [14]). The reason for this is that chained form systems $\dot{x} = \sum_{j=0}^{m} g_j(x)u_j$ fail the necessary condition that the mapping

$$\begin{aligned} \gamma : \mathbb{R}^n \times \mathbb{R}^m &\rightarrow \mathbb{R}^n \\ \gamma : (x, u) &\mapsto \sum_{j=0}^{m} g_j(x)u_j \end{aligned}$$

be onto an open set containing the origin. This mapping for chained form systems fails to map small regions in $\mathbb{R}^n \times \mathbb{R}^m$ into small regions about the origin in $\mathbb{R}^n$. Bearing this result in mind, various researchers have tried to stabilize such systems by time-varying or non-smooth state feedback.

Many of the algorithms for point stabilization require the system to be in chained form. For two-input systems, a class of smooth, time-varying control laws for local and global asymptotic stabilization to a point was presented by Teel *et al.* [74]. This method consists of taking the chained form system and converting it into power form, which has

the structure

$$\begin{aligned}
\dot{y}_0 &= v_0 \\
\dot{y}_1 &= v_1 \\
\dot{z}_1 &= y_0 v_1 \\
\dot{z}_2 &= \frac{1}{2}(y_0)^2 v_1 \\
&\vdots \\
\dot{z}_{n-2} &= \frac{1}{(n-2)!}(y_0)^{n-2} v_1 \ .
\end{aligned}$$

It was shown that the following control laws locally asymptotically stabilize the origin of the power form system:

$$\begin{aligned}
v_0 &= -y_0 + \left(\sum_{k=1}^{n-2}(z_k)^2\right)(\cos t - \sin t) \\
v_1 &= -y_1 + \sum_{k=1}^{n-2} c_k z_k \cos(kt)
\end{aligned}$$

for constants $c_k < 0$.

This procedure was extended by Walsh and Bushnell [85] to locally asymptotically stabilize the origin of $(m+1)$-input, m-chain, single-generator chained form systems. As above, the chained form system is first converted to power form

$$\begin{aligned}
\dot{y}_j &= v_j \quad \text{for} \quad 0 \le j \le m \\
\dot{z}_k^j &= \frac{1}{k!}(y_0)^k v_j \quad \text{for} \quad 1 \le k \le n_j \ , \quad 0 \le j \le m \ ,
\end{aligned}$$

where z_k^j is the k^{th} level state in chain j. The origin $(y, z) = (0, 0)$ is locally asymptotically stable under the action of the controls

$$\begin{aligned}
v_0 &= -y_0 + \left(\sum_{j=1}^{m}\sum_{k=1}^{n_j}(z_k^j)^2\right)(\cos t - \sin t) \\
v_j &= -y_j + \sum_{k=1}^{n_j} c_k^j z_k^j \cos(kt) \quad \text{for} \quad 1 \le j \le m \ ,
\end{aligned}$$

where each $c_k^j < 0$. This reference also uses the fire truck as an example to illustrate the proposed stabilizing control law.

In Walsh *et al.* [83], a technique for stabilizing nonholonomic systems to trajectories is presented. An explicit control law is constructed to locally exponentially stabilize the

system to a desired trajectory, which has been generated by an open-loop path planner. The method is to linearize the system about the given trajectory, check to see if the resulting time-varying linear system is uniformly completely controllable, and if so, construct a linear time-varying feedback control law to locally stabilize the system about the trajectory.

In Murray *et al.* [58], a non-smooth, time-varying feedback control law achieving local exponential convergence to a neighborhood of the origin for two-input chained form systems was presented.

Sørdalen and Egeland [69] present a method of globally stabilizing about the origin with exponential convergence for a two-input, chained form system. Here, the feedback control laws were developed for the system in chained form instead of power form.

Pomet [61] presents a constructive approach for deriving a time-varying smooth feedback control law that can be applied to globally uniformly asymptotically stabilizing chained form systems to the origin.

In summary, different methods for steering wheeled nonholonomic systems in chained form and various approaches for feedback stabilization were described. The steering method using sinusoids in a step-by-step manner was used to show the construction of chained form systems, but is not advised in practice. The "all-at-once" sinusoidal method is easier to use, and works well for paths that require at least one reversal. Polynomial and piecewise constant control inputs are recommended for paths without reversals. If the path does have a reversal, as in the parallel parking maneuver, one can use polynomial or piecewise constant inputs if the path is planned in two parts: first for the forward direction, then for the reverse direction.

Chapter 5

Examples of Wheeled Nonholonomic Systems

In this chapter, the theory discussed in earlier chapters is illustrated with examples of wheeled nonholonomic systems. Two mobile robot examples are considered in detail: the fire truck, which has three axles and three inputs (steering on the front and back axles plus driving), and an "extended fire truck" that has five axles and three inputs. The introduction of the fire truck system allowed us to generalize many of the ideas developed for two-input nonholonomic systems as presented by Murray and Sastry [57], where a two-axle car was converted to chained form, and by Sørdalen [68], where a car pulling N trailers was converted to chained form. Considering the more general multiple-input wheeled nonholonomic systems also prompted us to look into the theory of exterior differential systems, extending the ideas of Tilbury *et al.* [76], where a car pulling N trailers was converted to Goursat normal form, and dualizing the ideas presented in Tilbury *et al.* [78], where a general multiple-steering, multiple-trailer system was converted to chained form by the method of dynamic state feedback.

Our investigation of the fire truck system has been fundamental to understanding multiple-steering, multiple-trailer nonholonomic systems, which may be used in practice in manufacturing plants, nuclear power plants, or any area unsafe for human operators.

The outline of this chapter is as follows. In Section 5.1, the kinematic equations of the fire truck system are converted to chained form. In Section 5.2, the kinematics of the fire truck are converted to extended Goursat normal form, showing the differences and

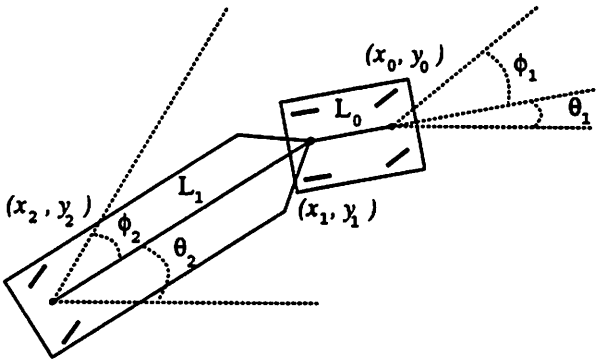


Figure 5.1: The configuration of the fire truck.

similarities between the two dual conversion methods. In Section 5.3, numerical simulation results of the fire truck system are presented, demonstrating the advantage of having the extra steering wheel. In Section 5.4, a five-axle, two-steering mobile robot is converted to extended Goursat normal form. In Section 5.5, numerical simulation results of this "extended fire truck" are presented. In Section 5.6, converting mobile robots configured with off-axle, or kingpin, hitching to chained form or Goursat normal form is studied.

5.1 Converting the Fire Truck to Chained Form

In this section, the kinematic model of a fire truck mobile robot is converted to chained form using the results stated in Section 3.1. Fire trucks are used to carry aerial ladders, tools, and equipment and have the main purpose of rescue and ventilation. They are mainly used by fire departments in large cities in the United States and have great maneuverability through narrow city streets due to the extra steering on the third axle, or tiller.

5.1.1 The Fire Truck System

The fire truck is an example of a three-input nonholonomic system. It is mathematically modeled as two planar rigid bodies supported by three axles. Assuming axle-to-axle hitching, the support of the trailer is over the center of the rear axle of the truck. The first and third axles are allowed to pivot, while the middle axle is rigidly fixed to the truck's body. The wheels are assumed to roll but not slip, thus giving linear velocity constraints.

The derivation of the kinematic equations for the fire truck refers to Figure 5.1, where the two rigid bodies are emphasized. The states of the kinematic model, all functions of time, are chosen as follows: (x_1, y_1) is the Cartesian location of the center of the rear axle of the truck, ϕ_1 is the steering angle of the front wheels with respect to the truck's body, and θ_1 is the orientation of the truck's body with respect to the horizontal axis of the inertial frame. The states $(x_2, y_2, \phi_2, \theta_2)$ are described similarly for the trailer, except that ϕ_2 is the angle of the rear wheels with respect to the trailer's body.

Let the distance between the front and rear axles of the truck be L_0, and the distance between the centers of the rear axles of the truck and trailer be L_1. This gives the holonomic constraints

$$\begin{aligned} x_0 &= x_1 + L_0 \cos\theta_1 \qquad y_0 = y_1 + L_0 \sin\theta_1 \\ x_2 &= x_1 - L_1 \cos\theta_2 \qquad y_2 = y_1 - L_1 \sin\theta_2 \ . \end{aligned}$$

Thus, the six coordinates $x := (x_1, y_1, \phi_1, \theta_1, \phi_2, \theta_2)$ are sufficient to represent the positions and orientations of the truck, trailer, and wheels.

For a mechanical system with wheels rolling and turning on a surface, the non-slipping constraint states that the velocity of a body in the direction perpendicular to each wheel must be zero. In terms of coordinates, for a wheel centered at location (x, y) and at an angle φ with respect to the horizontal axis of the fixed frame, the constraint is written as

$$0 = v_x \sin\varphi - v_y \cos\varphi \ .$$

In order to simplify the kinematic model of the fire truck, each pair of wheels is modeled as a single wheel centered at the midpoint of the axle[1]. Requiring that the wheels do not slip gives the three linear velocity constraints

$$\begin{aligned} 0 &= \dot{x}_1 \sin(\theta_1 + \phi_1) - \dot{y}_1 \cos(\theta_1 + \phi_1) - \dot{\theta}_1 L_0 \cos\phi_1 \\ 0 &= \dot{x}_1 \sin\theta_1 - \dot{y}_1 \cos\theta_1 \\ 0 &= \dot{x}_1 \sin(\theta_2 + \phi_2) - \dot{y}_1 \cos(\theta_2 + \phi_2) + \dot{\theta}_2 L_1 \cos\phi_2 \ . \end{aligned}$$

[1]It may be shown (see the appendix) that in fact the two wheels have different angles and their normals all intersect at a single point. If, for example, ϕ_1^1 and ϕ_1^2 are the angles of the front wheels of the truck, a holonomic constraint can be derived that eliminates one of these two variables. Thus, only one of the wheel angles needs to be in the state of the system.

These constraints can be expressed more compactly as $\omega^i(x) \cdot \dot{x} = 0$, where the covectors $\omega^i(x)$ are written in coordinates in $\mathbb{R}^6$ as

$$\begin{aligned}
\omega^1(x) &= [\ \sin(\theta_1+\phi_1) \quad -\cos(\theta_1+\phi_1) \quad 0 \quad -L_0\cos\phi_1 \quad 0 \quad 0 \quad] \\
\omega^2(x) &= [\ \sin\theta_1 \quad -\cos\theta_1 \quad 0 \quad 0 \quad 0 \quad 0 \quad] \\
\omega^3(x) &= [\ \sin(\theta_2+\phi_2) \quad -\cos(\theta_2+\phi_2) \quad 0 \quad 0 \quad 0 \quad L_1\cos\phi_2 \quad]\ .
\end{aligned} \tag{5.1}$$

The corresponding codistribution is $\Omega(x) = \text{span}\{\omega^1(x), \omega^2(x), \omega^3(x)\}$. Since Ω has dimension three and the state space is of dimension six, a three-dimensional distribution $\Delta(x) := \text{span}\{g_0(x), g_1(x), g_2(x)\}$ can be found such that $\omega^i(x) \cdot g_j(x) = 0$, for all $\omega^i \in \Omega$, $g_j \in \Delta$. A simple calculation will show that the following vector fields form a basis for Δ:

$$g_0 = \begin{pmatrix} \cos\theta_1 \\ \sin\theta_1 \\ 0 \\ \frac{1}{L_0}\tan\phi_1 \\ 0 \\ -\frac{1}{L_1}\sec\phi_2\sin(\phi_2-\theta_1+\theta_2) \end{pmatrix} \quad g_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad g_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}. \tag{5.2}$$

The nonholonomic constraints $\omega^i(x) \cdot \dot{x} = 0$ for all $\omega^i \in \Omega$ are equivalent to having $\dot{x} \in \Delta$, i.e., $\dot{x}$ is a linear combination of vector fields in Δ with functions of x as coefficients. Therefore the kinematic model of the fire truck as a control system with three inputs can be written as

$$\dot{x} = g_0(x)u_0 + g_1(x)u_1 + g_2(x)u_2\ . \tag{5.3}$$

The basis $\{g_0, g_1, g_2\}$ for Δ is chosen so that the input u_0 corresponds to the forward driving velocity of the truck, u_1 corresponds to the steering velocity of the front wheels of the truck and u_2 corresponds to the steering velocity of the rear wheels of the trailer.

5.1.2 Converting to Chained Form

Theorem 18 in Section 3.1.1 is now used to find the chained form equations for the fire truck system. The vector fields f_0, f_1, f_2, which will be shown to satisfy the conditions

of the proposition, are

$$f_0 = \begin{pmatrix} 1 \\ \tan\theta_1 \\ 0 \\ \frac{1}{L_0}\sec\theta_1\tan\phi_1 \\ 0 \\ -\frac{1}{L_1}\sec\theta_1\sec\phi_2\sin(\phi_2-\theta_1+\theta_2) \end{pmatrix}, \; f_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \; f_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \qquad (5.4)$$

where $f_0 = \sec\theta_1 g_0$ with g_0 as in equation (5.2). These can be considered as the original input vector fields after the input transformation $\tilde{u}_0 = u_0\cos\theta_1$. In this representation, three of the states are controlled directly: x_1, ϕ_1, ϕ_2, so their velocities are the inputs.

The distributions G_i in equation (3.5) are now constructed, and their involutivity will be checked. By way of notation, define $f_3 := \mathrm{ad}_{f_0} f_1$, $f_4 := \mathrm{ad}_{f_0} f_2$ and $f_5 := \mathrm{ad}^2_{f_0} f_1$. Recall that $x = (x_1, y_1, \phi_1, \theta_1, \phi_2, \theta_2)$.

$$\begin{aligned}
G_0 &= \mathrm{span}\{f_1, f_2\} \\
G_1 &= \mathrm{span}\{f_1, \mathrm{ad}_{f_0} f_1, f_2, \mathrm{ad}_{f_0} f_2\} \\
G_2 &= \mathrm{span}\{f_1, \mathrm{ad}_{f_0} f_1, \mathrm{ad}^2_{f_0} f_1, f_2, \mathrm{ad}_{f_0} f_2, \mathrm{ad}^2_{f_0} f_2\} \\
&= \mathrm{span}\{f_1, \mathrm{ad}_{f_0} f_1, \mathrm{ad}^2_{f_0} f_1, f_2, \mathrm{ad}_{f_0} f_2\} \quad = \mathrm{span}\{f_1, f_3, f_5, f_2, f_4\} \\
&= \mathrm{span}\left\{ \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{-1}{L_0}\sec^2\phi_1\sec\theta_1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{L_0}\sec^2\phi_1\sec^3\theta_1 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L_0L_1}\cos(\phi_2+\theta_2)\sec^2\phi_1\sec\phi_2\sec^3\theta_1 \end{pmatrix}, \right. \\
&\qquad \left. \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L_1}\cos(\theta_2-\theta_1)\sec^2\phi_2\sec\theta_1 \end{pmatrix} \right\}
\end{aligned}$$

The distribution G_2 has dimension $n-1 = 5$ on $U = \{x : \theta_1 - \theta_2, \phi_1, \phi_2, \theta_1 \neq \pm\frac{\pi}{2}\} \subset \mathbb{R}^6$ since the five vector fields that define G_2 along with f_0 are linearly independent,

i.e.,

$$\det[f_0\ f_1\ f_2\ f_3\ f_4\ f_5] = \frac{-1}{L_0^2 L_1}\cos(\theta_1-\theta_2)\sec^4\phi_1\sec^2\phi_2\sec^5\theta_1\ .$$

It may also be verified that G_0, G_1 and G_2 are involutive on U and that the functions $h_0 = x_1$, $h_1 = y_1$ and $h_2 = \theta_2$ satisfy

$$\begin{aligned} &dh_0 \perp G_j \qquad j = 0,1,2 \\ &dh_1 \perp G_j \qquad j = 0,1 \\ &dL_{f_0}h_1 \perp G_0 \\ &dh_2 \perp G_0\ , \end{aligned}$$

which is equation (3.7) from Section 3.1.1 with $n = 6$, $n_1 = 2$, $n_2 = 1$, and $G_2 = G_3 = G_4 = G_5 = G_{n-1}$. Note that there is a lack of uniqueness in the h functions.

Remark. In the open set $U \subset \mathbb{R}^6$, the fire truck is not in the *jack-knife* configuration, that is, the wheels are less than $\pi/2$ relative to their respective bodies, and the truck's body angle is not $\pm\pi/2$. Sometimes using higher-order Lie brackets allows the spanning of the whole space $\mathbb{R}^6$, including the singularities. Out of all the distributions with at most second order Lie brackets, the distribution

$$\text{span}\{f_1, \text{ad}_{f_0}f_1, \text{ad}^2_{f_0}f_1, f_2, \text{ad}^2_{f_0}f_2\}$$

was found to span $U' = \{x : \phi_1, \phi_2, \theta_1 \neq \pm\frac{\pi}{2}\} \subset \mathbb{R}^6$, which eliminates the singularity at the jack-knife configuration but still requires $\theta_1 \neq \pm\pi/2$. Allowing third order Lie brackets, the distributions

$$\text{span}\{f_1, \text{ad}_{f_0}f_1, \text{ad}^2_{f_0}f_1, \text{ad}^3_{f_0}f_1, f_2\}$$
$$\text{span}\{f_1, \text{ad}_{f_0}f_1, \text{ad}^3_{f_0}f_1, f_2, \text{ad}^2_{f_0}f_2\}$$

both span U', and

$$\text{span}\{f_1, \text{ad}_{f_0}f_1, \text{ad}^3_{f_0}f_1, f_2, \text{ad}_{f_0}f_2\}$$

spans U.

The coordinate transformation $(\xi, \zeta, \eta) = \Phi(x)$ from equation (3.8) is computed

as follows:

$$\begin{aligned}
\xi_0 = \quad h_0 \quad &= x_1 \\
\zeta_0 = \; L_{f_0}^2 h_1 \; &= \frac{1}{L_0} \tan\phi_1 \sec^3\theta_1 \\
\zeta_1 = \; L_{f_0} h_1 \; &= \tan\theta_1 \\
\zeta_2 = \quad h_1 \quad &= y_1 \\
\eta_0 = \; L_{f_0} h_2 \; &= -\frac{1}{L_1} \sin(\phi_2 - \theta_1 + \theta_2) \sec\phi_2 \sec\theta_1 \\
\eta_1 = \quad h_2 \quad &= \theta_2 \; .
\end{aligned} \tag{5.5}$$

This is a valid coordinate transformation since the matrix $\frac{\partial \Phi}{\partial x}$ is nonsingular.

The derivatives of the chained form coordinates are taken with respect to time and substituted into $\dot{x}$ from $\dot{x} = f_0 u_0 + f_1 u_1 + f_2 u_2$ with f_i as in equation (5.4). This gives the following equations showing the required state feedback to transform the system equations into the two-chain, single-generator chained form:

$$\begin{aligned}
\dot{\xi}_0 &= \tilde{u}_0 = v_0 \\
\dot{\zeta}_0 &= \frac{3}{L_0^2} \tan^2\phi_1 \tan\theta_1 \sec^4\theta_1 \, \tilde{u}_0 + \frac{1}{L_0} \sec^2\phi_1 \sec^3\theta_1 \, u_1 = v_1 \\
\dot{\zeta}_1 &= \frac{1}{L_0} \tan\phi_1 \sec^3\theta_1 \, \tilde{u}_0 = \zeta_0 v_0 \\
\dot{\zeta}_2 &= \tan\theta_1 \, \tilde{u}_0 = \zeta_1 v_0 \\
\dot{\eta}_0 &= \frac{1}{L_0 L_1} \cos(\phi_2 + \theta_2) \tan\phi_1 \sec\phi_2 \sec^3\theta_1 \, \tilde{u}_0 \\
&\quad + \frac{1}{L_1^2} \cos(\phi_2 - \theta_1 + \theta_2) \sin(\phi_2 - \theta_1 + \theta_2) \sec^2\phi_2 \sec^2\theta_1 \, \tilde{u}_0 \\
&\quad - \frac{1}{L_1} \cos(\theta_2 - \theta_1) \sec^2\phi_2 \sec\theta_1 \, u_2 = v_2 \\
\dot{\eta}_1 &= -\frac{1}{L_1} \sin(\phi_2 - \theta_1 + \theta_2) \sec\phi_2 \sec\theta_1 \, \tilde{u}_0 = \eta_0 v_0 \; .
\end{aligned} \tag{5.6}$$

Remark. Using the fire truck system as a reference, we can infer characteristics of chained form systems. The generating input, v_0 is related to the driving velocity of the system by $u_0 = \sec\theta_1 v_0$. For the fire truck, the first chain corresponds to the first two axles (one steerable, one passive) and the second chain corresponds to the third steerable axle. In general multiple-steering, multiple-trailer systems, the single-generator chained form can be applied with the number of chains corresponding to the number of steerable axles in the system. The paper by Tilbury *et al.* [78] shows that such a composite system can be

converted into multiple-input chained form; but that, in general, dynamic state feedback is needed to achieve the transformation. This will be discussed in the context of an example in Section 5.4.

5.2 Converting the Fire Truck to Extended Goursat Normal Form

In this section, Algorithm 2 described in Section 3.2 is used to transform the kinematic equation of a three-input wheeled nonholonomic system into extended Goursat normal form. The example of the fire truck, as shown in Figure 5.1, is used again since it provides a simple illustration.

With the state space parameterized by $x = (x_1, y_1, \phi_1, \theta_1, \phi_2, \theta_2)$ as in the previous section, the three nonholonomic linear velocity constraints for the fire truck can be written as one-forms as follows (refer to equation (5.1)):

$$\begin{aligned}
\alpha^0 &= \sin(\theta_1 + \phi_1)dx_1 - \cos(\theta_1 + \phi_1)dy_1 - L_0 \cos\phi_1 d\theta_1 \\
\alpha^1 &= \sin\theta_1 dx_1 - \cos\theta_1 dy_1 \\
\alpha^2 &= \sin(\theta_2 + \phi_2)dx_1 - \cos(\theta_2 + \phi_2)dy_1 + L_1 \cos\phi_2 d\theta_2 \ .
\end{aligned}$$

The Pfaffian system associated with the fire truck can therefore be written as

$$I = \{\alpha^0, \alpha^1, \alpha^2\}.$$

The first step in Algorithm 2 is to compute the derived flag for this system. First, the basis of the constraints is completed with

$$\begin{aligned}
\alpha^3 &:= \cos\theta_1 dx_1 + \sin\theta_1 dy_1 \\
\alpha^4 &:= d\phi_1 \\
\alpha^5 &:= d\phi_2 \ .
\end{aligned}$$

The exterior derivatives of α^0, α^1, and α^2 are then computed, writing them in terms of the

basis of two-forms given by $\{\alpha^i \wedge \alpha^j : 0 \le i < j \le 5\}$:

$$\begin{aligned} d\alpha^0 &= \frac{\sin(\phi_1+\theta_1)}{\cos\phi_1\cos\theta_1\sin\phi_1+(1-\sin^2\phi_1)\sin\theta_1}\,\alpha^3\wedge\alpha^5 \mod I \qquad (5.7)\\ &\neq 0 \mod I \\ d\alpha^1 &= -\frac{\sec\phi_1}{L_1}\,\alpha^0\wedge\alpha^5 \mod \alpha^1 \qquad (5.8)\\ &\equiv 0 \mod I \\ d\alpha^2 &= \sec\phi_2\cos(\theta_1-\theta_2)\,\alpha^4\wedge\alpha^5 \mod I \qquad (5.9)\\ &\neq 0 \mod I\,. \end{aligned}$$

From this representation, α^1 is the only constraint that drops to the $I^{(1)}$ level in the derived flag:

$$\begin{aligned} I = I^{(0)} &= \{\alpha^0,\alpha^1,\alpha^2\} \\ I^{(1)} &= \{\alpha^1\} \\ I^{(2)} &= \{0\}\,. \end{aligned}$$

The second step of Algorithm 2 constructs two towers by finding an integrable π to satisfy the Goursat congruences (3.15). Choose

$$\omega_1^1 = \alpha^1\,.$$

The Goursat congruence for this one-form is $d\omega_1^1 = \pi\wedge\omega_2^1 \mod \omega_1^1$. From equation (5.8), if

$$\begin{aligned} \tilde\pi &= \alpha^5 \neq 0 \mod I \\ \omega_2^1 &= \frac{1}{L_1}\sec\phi_1\alpha^0\,, \end{aligned}$$

then $d\omega_1^1 = \tilde\pi\wedge\omega_2^1 \mod \omega_1^1$. On the other hand, if π is chosen as

$$\pi := \cos\theta_1\tilde\pi + \sin\theta_1\omega_1^1 = dx_1\,,$$

it is integrable and $d\omega_1^1 = \pi\wedge\omega_2^1 \mod \omega_1^1$. Since there are only three constraints in I, there are two towers: one with $n_1 = 2$ and the other with $n_2 = 1$,

$$\begin{aligned} I = I^{(0)} &= \{\omega_1^1,\omega_2^1,\omega_1^2\} \\ I^{(1)} &= \{\omega_1^1\} \\ I^{(2)} &= \{0\} \end{aligned}$$

where $\omega_1^2 := \alpha^2$.

The third step of Algorithm 2 solves the coordinate transformation for the first tower by using the solution to Pfaff's problem (see Theorem 11 in Section 2.2). The rank of the one-form ω_1^1 is one, since

$$\begin{aligned} d\omega_1^1 &= \cos\theta_1 d\theta_1 \wedge dx_1 + \sin\theta_1 d\theta_1 \wedge dy_1 \\ d\omega_1^1 \wedge \omega_1^1 &= d\theta_1 \wedge dy_1 \wedge dx_1 \neq 0 \end{aligned}$$

and

$$d\omega_1^1 \wedge d\omega_1^1 \wedge \omega_1^1 = (\cos\theta_1 d\theta_1 \wedge dx_1 + \sin\theta_1 d\theta_1 \wedge dy_1) \wedge d\theta_1 \wedge dy_1 \wedge dx_1 = 0 .$$

With a rank of one, there exists a function f_1 satisfying

$$d\omega_1^1 \wedge \omega_1^1 \wedge df_1 = 0 \quad \text{and} \quad \omega_1^1 \wedge df_1 \neq 0 .$$

Clearly, $df_1 = dx_1 = \pi$ satisfies this equation.

Let the coordinate z_0^0 be defined from π as $z_0^0 = x_1$. A function f_2 satisfying Pfaff's equation is

$$\omega_1^1 \wedge df_1 \wedge df_2 = 0 \quad \text{and} \quad df_1 \wedge df_2 \neq 0 , \tag{5.10}$$

noting that

$$\omega_1^1 \wedge \pi = \cos(\phi_1 + \theta_1) dx_1 \wedge dy_1 .$$

Therefore, let $f_2 = y_1$ and set the last coordinate in the first chain to be $z_2^1 = f_2 = y_1$.

The solution to Pfaff's problem gives $\omega_1^1 \wedge \pi \wedge df_2 = 0$, which means ω_1^1 is a linear combination of π and df_2,

$$\omega_1^1 = -\cos\theta_1 df_2 + \sin\theta_1 \pi .$$

We would like to rescale ω_1^1 to be of the form

$$\bar{\omega}_1^1 = dz_2^1 - z_1^1 dz_0^0 .$$

Therefore, let

$$\bar{\omega}_1^1 = -\sec\theta_1 \omega_1^1 = dz_2^1 - \tan\theta_1 dz_0^0 ,$$

from which the coefficient of dz_0^0 is set to $z_1^1 = \tan\theta_1$.

The one-form ω_2^1 is next modified to be of the form

$$\bar{\omega}_2^1 \;=\; dz_1^1 - z_0^1 dz_0^0 \; .$$

To do this, let $\bar{\omega}_2^1$ be a linear combination of ω_2^1 and $\bar{\omega}_1^1$

$$\begin{aligned}
\bar{\omega}_2^1 &= -\sec^2\theta_1\omega_2^1 - \frac{1}{L_1}\sec^2\theta_1\sec\phi_1\cos(\phi_1+\theta_1)\bar{\omega}_1^1 \\
&= -\sec^2\theta_1\frac{1}{L_1}\sec\phi_1(\sin(\phi_1+\theta_1)dx_1 - \cos(\phi_1+\theta_1)dy_1 \\
&\quad -L_1\cos\phi_1 d\theta_1) - \frac{1}{L_1}\sec^2\theta_1\sec\phi_1\cos(\phi_1+\theta_1)(dy_1 - \tan\theta_1 dx_1) \\
&= \sec^2\theta_1 d\theta_1 - \frac{1}{L_1}\sec^3\theta_1\tan\phi_1 dx_1
\end{aligned}$$

and set the coefficient of $dx_1 = dz_0^0$ to be $z_0^1 = -\frac{1}{L_1}\sec^3\theta_1\tan\phi_1$. This completes the coordinate transformation for the first tower.

The fourth step in Algorithm 2 finds a coordinate transformation for the second tower. This does not involve a Goursat congruence equation since there is only a single one-form in the second tower. First, the rank of $\omega_1^2 := \alpha^2$ is computed to be one as follows:

$$\begin{aligned}
d\omega_1^2 &= \cos(\phi_2+\theta_2)(d\theta_2\wedge dx_1 + d\phi_2\wedge dx_1) + \sin(\phi_2+\theta_2)(d\theta_2\wedge dy_1 + d\phi_2\wedge dy_1) \\
&\quad +L_1\sin\phi_2 d\phi_2\wedge d\theta_2
\end{aligned}$$

$$\begin{aligned}
d\omega_1^2\wedge\omega_1^2 &= -\cos^2(\phi_2+\theta_2)(d\theta_2\wedge dx_1\wedge dy_1 + d\phi_2\wedge dx_1\wedge dy_1) \\
&\quad -L_1\cos\phi_2\cos(\phi_2+\theta_2)d\phi_2\wedge dx_1\wedge d\theta_2 - L_1\sin\phi_2\cos(\phi_2+\theta_2)d\phi_2\wedge d\theta_2\wedge dy_1 \\
&\quad -L_1\cos\phi_2\sin(\phi_2+\theta_2)d\phi_2\wedge dy_1\wedge d\theta_2 + L_1\sin\phi_2\sin(\phi_2+\theta_2)d\phi_2\wedge d\theta_2\wedge dx_1 \\
&\quad +\sin^2(\phi_2+\theta_2)(d\theta_2\wedge dy_1\wedge dx_1 + d\phi_2\wedge dy_1\wedge dx_1) \quad \neq 0
\end{aligned}$$

and

$$d\omega_1^2\wedge d\omega_1^2\wedge\omega_1^2 \;= 0 \; .$$

With a rank of one, there exists a function f_1 satisfying

$$d\omega_1^2\wedge\omega_1^2\wedge dg_1 = 0 \quad \text{and} \quad \omega_1^2\wedge dg_1 \neq 0 \; .$$

If g_1 is chosen to be $\cos(\phi_2+\theta_2)dx_1 + \sin(\phi_2+\theta_2)dy_1$, then the equation is satisfied. For the second tower, however, it must be that $dg_1 = df_1 = \pi$ in order for the coordinate

transformation to work. In this case, ω_1^2 must be modified by adding terms in the one-form basis I,

$$\begin{aligned}\bar{\omega}_1^2 &= \omega_1^2 + \cos(\phi_2+\theta_2)\bar{\omega}_1^1 \\ &= (\sin(\phi_2+\theta_2) - \cos(\phi_2+\theta_2)\tan\theta_1)dx_1 + L_2\cos\phi_2 d\theta_2 \\ &= \sec\theta_1 \sin(\phi_2+\theta_2-\theta_1)dx_1 + L_2\cos\phi_2 d\theta_2 \\ &= d\theta_2 + \frac{\sec\theta_1\sin(\phi_2+\theta_2-\theta_1)}{L_2\cos\phi_2}dx_1 \ .\end{aligned}$$

If $dg_1 = df_1 = \pi = dx_1$, then $d\omega_1^2 \wedge \omega_1^2 \wedge dg_1 = 0$ and $\omega_1^2 \wedge dg_1 \neq 0$. From the equality

$$\bar{\omega}_1^2 \wedge \pi = d\theta_2 \wedge dx_1 \ ,$$

a function g_2 satisfying Pfaff's equation (5.10) is found to be $g_2 = \theta_2$. The last coordinate in the second chain is set as $z_1^2 = g_2 = \theta_2$.

The one-form $\bar{\omega}_1^2$

$$\bar{\omega}_1^2 = dz_1^2 - z_0^2 dz_0^0 = d\theta_2 - z_0^2 dx_1$$

is already in the correct form for reading off the next coordinate. Setting z_0^2 to be the coefficient of dx_1, gives

$$z_0^2 = \frac{\sec\theta_1\sin(\phi_2+\theta_2-\theta_1)}{L_2\cos\phi_2} \ .$$

Therefore, we have found coordinates z_i^j such that

$$I = \{\bar{\omega}_1^1, \omega_2^1, \bar{\omega}_1^2\} = \{dz_2^1 - z_1^1 dz_0^0,\ dz_1^1 - z_0^1 dz_0^0,\ dz_1^2 - z_0^2 dz_0^0\} \ ,$$

which is the dual of the chained form

$$\begin{aligned} \dot{z}_0^0 &= v_0 \quad \dot{z}_0^1 = v_1 \quad \dot{z}_0^2 = v_2 \\ &\qquad \dot{z}_1^1 = z_0^1 v_0 \quad \dot{z}_1^2 = z_0^2 v_0 \\ &\qquad \dot{z}_2^1 = z_1^1 v_0 \ . \end{aligned}$$

5.3 Simulation of the Fire Truck

In this section, numerical simulation results for the fire truck system are presented. The symbolic manipulation software *Mathematica*[2] and the numerics software package *MATLAB*[3] are used to perform the simulations and plot the results. The simulation is

[2]Copyright Wolfram Research, Inc.
[3]Copyright The Math Works, Inc.

performed on the system in chained form

$$\begin{aligned} \dot{\xi}_0 &= v_0 \qquad \dot{\zeta}_0 = v_1 \qquad \dot{\eta}_0 = v_2 \\ &\qquad \dot{\zeta}_1 = \zeta_0 v_0 \qquad \dot{\eta}_1 = \eta_0 v_0 \\ &\qquad \dot{\zeta}_2 = \zeta_1 v_0 \,. \end{aligned} \tag{5.11}$$

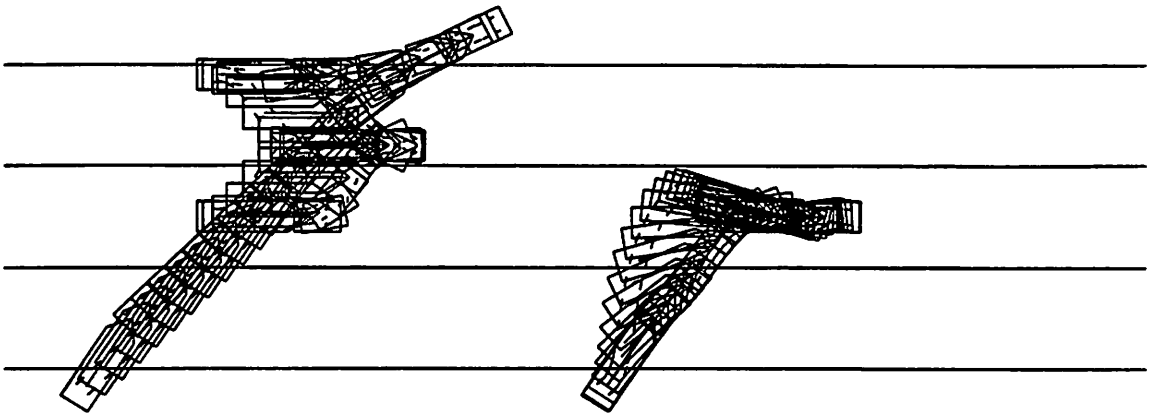
The transformed states are steered from an initial configuration to a final configuration by using the transformed inputs as constructed in the step-by-step sinusoidal steering algorithm, the all-at-once sinusoidal steering algorithm, or the polynomial steering algorithm. Then the inverse coordinate transformation

$$\begin{aligned} x_1 &= \xi_0 \\ y_1 &= \zeta_2 \\ \phi_1 &= \tan^{-1}(L_0\, \zeta_0 \cos^3(\tan^{-1}\zeta_1)) \\ \theta_1 &= \tan^{-1}\zeta_1 \\ \phi_2 &= \eta_1 - \tan^{-1}\zeta_1 + \tan^{-1}(L_1\, \eta_0 \cos(\tan^{-1}\zeta_1)\sec(\eta_1 - \tan^{-1}\zeta_1)) \\ \theta_2 &= \eta_1 \end{aligned} \tag{5.12}$$

with $L_0 = 1$ and $L_1 = 4$ is calculated to extract the trajectory of the fire truck in the original coordinates. The total vehicle length for all of the simulations is 6.5 units. The results are presented for the parallel-parking maneuver using sinusoidal inputs. An arbitrary trajectory with both step-by-step sinusoidal inputs and all-at-once sinusoidal inputs is used to show the difference between these two steering methods. To see the advantage of having the extra steering wheel, both left and right corner trajectories and a change-lane trajectory using polynomial inputs for the fire truck and the same system without tiller steering are simulated.

Remark. Since the coordinate transformation to chained form is a diffeomorphism on the open set $U = \{x : \theta_1 - \theta_2, \phi_1, \phi_2, \theta_1 \neq \pm\frac{\pi}{2}\} \subset \mathbb{R}^6$, these singular points must be avoided when the initial and final configurations are chosen for the simulation. One practical solution is to plan a path that does not start or end the fire truck in a singular configuration, then rotate the resulting trajectory about the origin to yield the desired trajectory. This will be explained later.

Remark. In order to limit the size of the configuration states or the inputs, one must take into account the coordinate transformation to chained form since the simulation is



Step-by-step sinusoidal inputs. All-at-once sinusoidal inputs.

Figure 5.2: Comparing the step-by-step and all-at once sinusoidal methods for initial configuration $(x_1, y_1) = (4, 0)$ with the body angles $\theta_1 = 0.78$, $\theta_2 = 1$ and steering angles $\phi_1 = 0$, $\phi_2 = 0.21$ and final configuration $(x_1, y_1) = (12, 5)$ with all body and steering angles at zero.

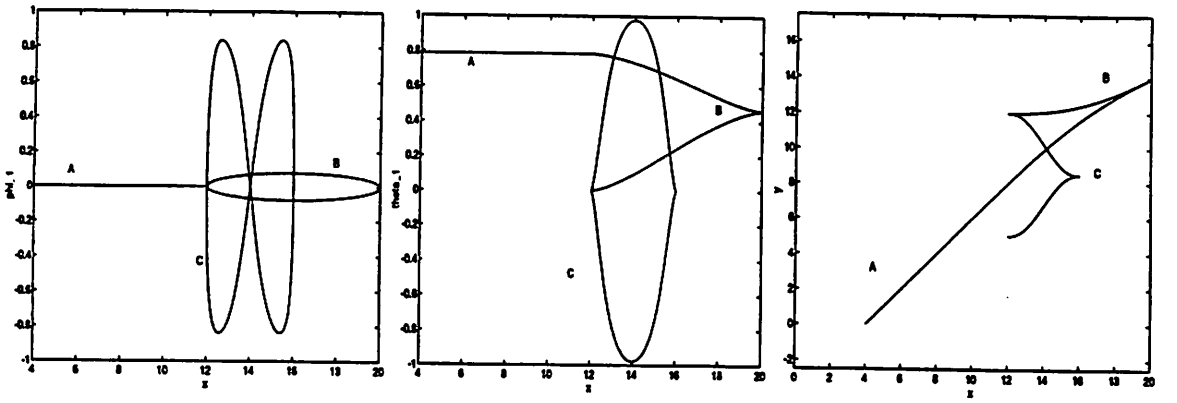
performed on the system in chained form. From the equations of the fire truck system, the steering wheel angles can take values between $-90°$ and $90°$. In reality[4], the front steering wheel angle is limited to $-45° \leq \phi_1 \leq 45°$ and the tiller steering wheel angle is limited to $-15° \leq \phi_2 \leq 15°$.

Figure 5.2 compares the step-by-step and the all-at-once sinusoidal steering methods for arbitrary initial and final configurations. We found that the all-at-once sinusoidal method was easier to execute and produces a more direct trajectory. The all-at-once method uses the inputs

$$\begin{aligned} v_0 &= a_0 + a_1 \sin \omega t \\ v_1 &= b_0 + b_1 \cos \omega t + b_2 \cos(2\omega t) \\ v_2 &= c_0 + c_1 \cos \omega t \end{aligned} \tag{5.13}$$

in one step. The step-by-step method, however, is performed in three separate steps. Figure 5.3 shows the phase plots that correspond to the trajectory resulting from using step-by-step sinusoidal inputs shown in Figure 5.2. The first part of the path, labeled A, corresponds to Step 0 and uses constant input to steer the transformed coordinates ξ_0, ζ_0, η_0 to their final values. In the original coordinates with small angles, this means x_1 and

[4]Data from Berkeley Fire Department Station No. 2.



ϕ_1 vs. x_1 θ_1 vs. x_1 y_1 vs. x_1

Figure 5.3: Sample phase plots from using step-by-step sinusoidal inputs showing the Lissajous figures for the trajectory in Figure 5.2.

ϕ_1 are steered to their final positions while the other four states drift. The second part, labeled B, corresponds to Step 1 and uses sinusoidal input to steer ζ_1 and η_1 to their final values. Referring to equation (5.12) with small angles, this means the body orientations θ_1 and θ_2 are steered to their final positions. The wheel orientation ϕ_2 is now also at its final position. The last part, labeled C, drives y_1 to its final position and returns the other states to their final positions.

The state x_1 is controlled directly by the input u_0 and therefore is moved in the direction of vector field f_0. Similarly, ϕ_1 is moved in the f_1 direction. Taking first-order Lie brackets shows that θ_1 is moved in the direction $\mathrm{ad}_{f_0} f_1$. The state y_1 is driven in the direction of the second-order Lie bracket $\mathrm{ad}^2_{f_0} f_1$. The number of loops is determined by the order of the Lie bracket needed to get net motion in a desired direction. Referring to the portion of the trajectory labeled C in Figure 5.3, in order to get net motion in the y_1 direction, the ϕ_1-x_1 phase plot has a Lissajous figure with two loops and the θ_1-x_1 phase plot has one loop.

Figure 5.4 shows the parallel parking maneuver that results from using the step-by-step sinusoidal steering method or the all-at-once sinusoidal steering method. In the original coordinates, this corresponds to steering the fire truck from $y_1 = 5$ to zero with all other coordinates starting and ending at zero. The inputs for this trajectory are found from the last step of the step-by-step sinusoidal algorithm, where the only state that must be changed is $y_1 = \zeta_2$, the last coordinate of the ζ chain, or from the all-at-once sinusoidal method with inputs as in equation (5.13) with $a_0 = b_0 = b_1 = c_0 = c_1 = 0$, $a_1 = 2$,

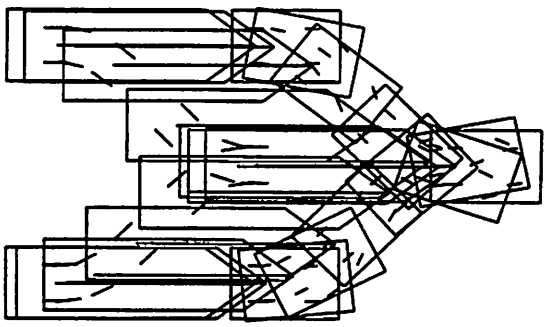


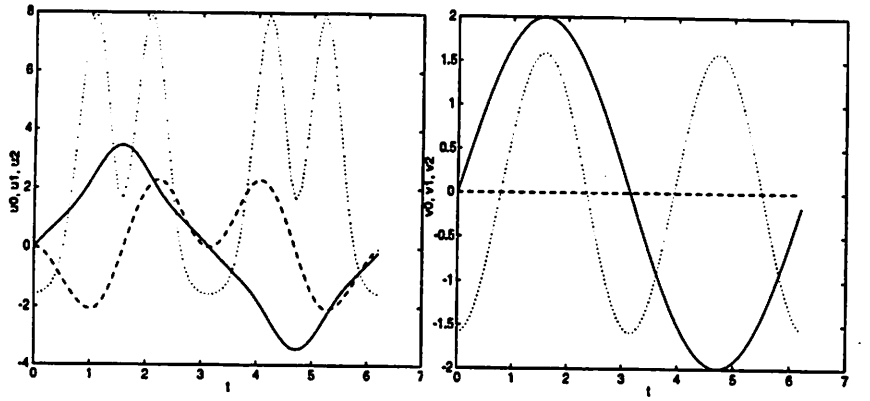
Figure 5.4: The parallel parking trajectory for the fire truck using sinusoidal inputs. The fire truck starts at $y_1 = 5$ and ends at zero, with all other coordinates starting and ending at zero.

$b_2 = -1.5915$, and $\omega = 1$ over the interval $[0, 2\pi]$.

In Figure 5.5, the chained form and original input functions needed to parallel park the fire truck as in Figure 5.4 are shown. The chained form inputs are the open-loop control laws for the system in two-chain, single-generator chained form. The physical inputs, however, depend on the states of the system, as can be seen in the following equations:

$$\begin{aligned} u_0 &= \sec\theta_1 v_0 \\ u_1 &= L_0 \cos^2\phi_1 \cos^3\theta_1 \left(v_1 - v_0 \frac{3}{L_0^2} \tan^2\phi_1 \tan\theta_1 \sec^4\theta_1 \right) \\ u_2 &= \frac{-L_1 \cos^2\phi_2 \cos\theta_1}{\cos(\theta_2 - \theta_1)} \left(v_2 - v_0 \left(\frac{\cos(\phi_2 + \theta_2)\tan\phi_1}{L_0 L_1 \cos\phi_2 \cos^3\theta_1} \right.\right. \\ &\quad \left.\left. + \frac{\cos(\phi_2 - \theta_1 + \theta_2)\sin(\phi_2 - \theta_1 + \theta_2)}{L_1^2 \cos^2\phi_2 \cos^2\theta_1} \right) \right) . \end{aligned}$$

Polynomial inputs are now used to show the control design and performance differences when the fire truck does not have tiller steering. For the system without a tiller, the following coordinate transformation to chained form (from using $h_1 = x_2$ and $h_2 = y_2$ in equation (3.8) from Section 3.1 without the η chain and following the procedure as for



Physical inputs. Chained form inputs.

Figure 5.5: Physical and chained form inputs for the parallel parking trajectory shown in Figure 5.4. For the inputs, u_0, v_0 are the solid lines, u_1, v_1 are the dotted line, and u_2, v_2 are the dashed line.

the fire truck, or following the procedure in [76]) is used

$$\begin{aligned}
\xi_0 &= x_2 \\
\zeta_0 &= \frac{\sec^4\theta_2 \sec^2(\theta_1-\theta_2)}{L_0 L_1^2}\left(3L_0\sin^2(\theta_1-\theta_2)\tan\theta_2 - L_0\tan(\theta_1-\theta_2) + \frac{L_1\tan\phi_1}{\cos(\theta_1-\theta_2)}\right) \\
\zeta_1 &= \frac{1}{L_1}\sec^3\theta_2\tan(\theta_1-\theta_2) \\
\zeta_2 &= \tan\theta_2 \\
\zeta_3 &= y_2 \,,
\end{aligned} \tag{5.14}$$

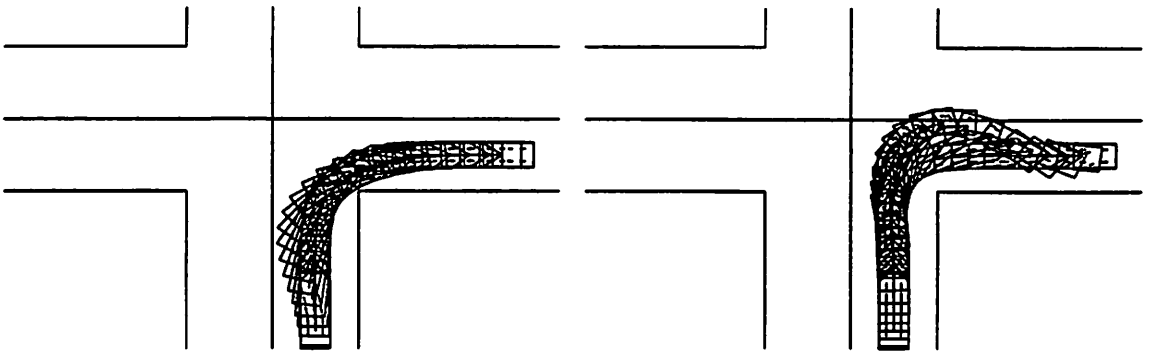
which kinematically is a two-axle car pulling one trailer.

For the fire truck in the chained form of equation (5.11), there are two chains, one of length three and one of length two, so the following polynomial control inputs are used

$$\begin{aligned}
v_0 &= 1 \\
v_1 &= b_0 + b_1 t + b_2 t^2 \\
v_2 &= c_0 + c_1 t \,.
\end{aligned}$$

For the fire truck without tiller steering of equation (5.14), there is only one chain of length four, so the two inputs are

$$\begin{aligned}
v_0 &:= \dot{\xi}_0 = 1 \\
v_1 &:= \dot{\zeta}_0 = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \,.
\end{aligned}$$



With tiller steering. Without tiller steering.

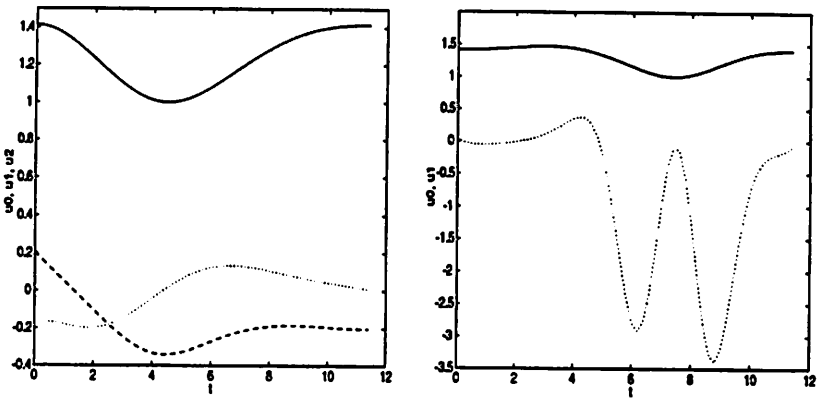
Figure 5.6: The fire truck with and without tiller steering for a 90° right hand turn in an intersection. The fire truck is steered from an initial state $(x_1, y_1) = (0, 4)$ with the body angles $\theta_1 = \theta_2 = \pi/2$ and steering angles $\phi_1 = \phi_2 = 0$ to a final state $(x_1, y_1) = (9.6, 10.6)$ with body and steering angles all zero.

In all of the following simulations, the initial and final states of the two systems are the same.

Figures 5.6 and 5.8 show the advantage of having tiller steering when making 90° right and left hand turns. For both of these trajectories, the fire truck goes through the singular point $\theta_1 = \pi/2$. To avoid this singularity for the right hand turn, the trajectory was simulated with the initial and final configurations at $\pi/4$ and $-\pi/4$, respectively. Then the entire resulting path was rotated by $\pi/4$. The left hand turn was simulated similarly, noting that the path is wider than the right hand turn in the same intersection due to the convention of driving on the right hand side of the road.

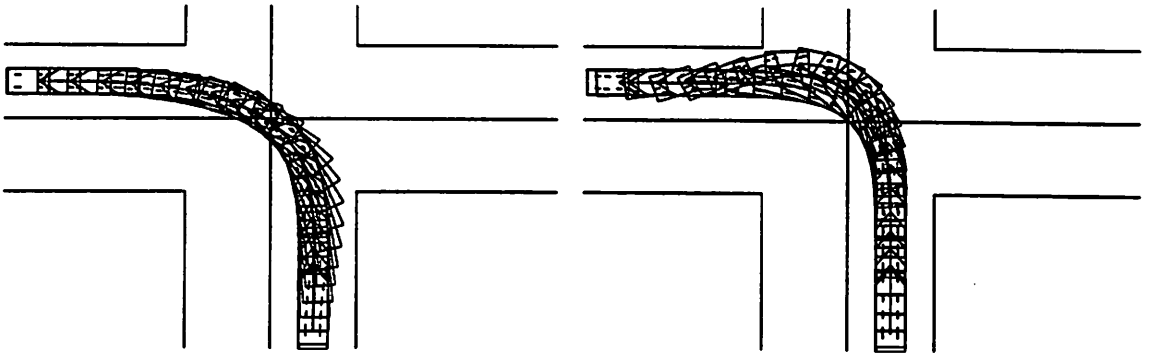
The inputs with and without tiller steering for the right hand turn trajectory are shown in Figure 5.7, which shows that the steering velocity for the system without tiller steering, which is just the steering velocity of the front wheels, switches back and forth more. Of greater interest, however, is that the magnitude of the input u_1 without tiller steering is larger than the inputs of the fire truck with tiller steering. In some sense, the control for the system without tiller steering has to work *harder*.

Figure 5.9 shows how the tiller steering assists the vehicle when it changes lanes on a freeway. The vehicle moves a total of 13 units in the x direction and 5 units in the y



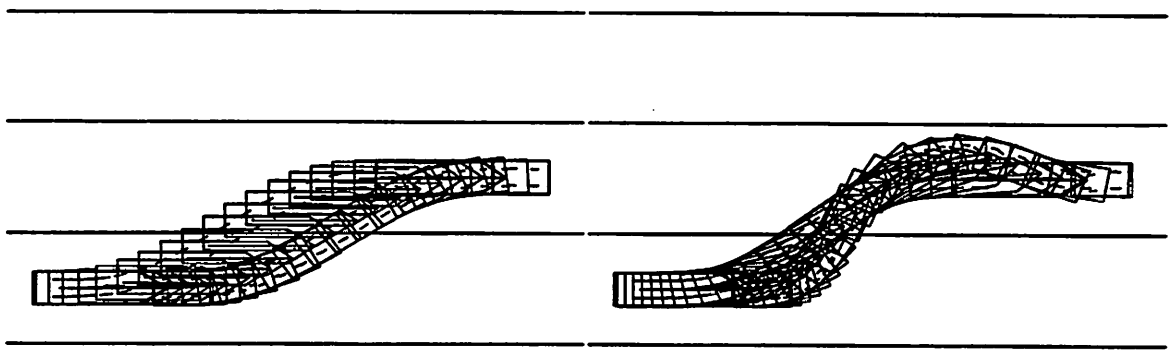
With tiller steering. Without tiller steering.

Figure 5.7: The inputs for the fire truck with and without tiller steering for the 90° right hand turn trajectory shown in Figure 5.6. The driving velocity u_0 is the solid line, the steering velocity of the front wheels u_1 is the dotted line, and the steering velocity of the tiller u_2 is the dashed line.



With tiller steering. Without tiller steering.

Figure 5.8: The fire truck with and without tiller steering for a 90° left hand turn in the same intersection. The fire truck is steered from an initial state $(x_1, y_1) = (0, 4)$ with the body angles $\theta_1 = \theta_2 = \pi/2$ and steering angles $\phi_1 = \phi_2 = 0$ to a final state $(x_1, y_1) = (-14, 15)$ with body angles π and steering angles at zero.



With tiller steering. Without tiller steering.

Figure 5.9: The fire truck with and without tiller steering for changing lanes on a freeway. The fire truck is steered from an initial position $(x_1, y_1) = (4, 0)$ with the body and steering angles all zero to a final position $(x_1, y_1) = (17, 5)$ with body and steering angles all zero.

direction.

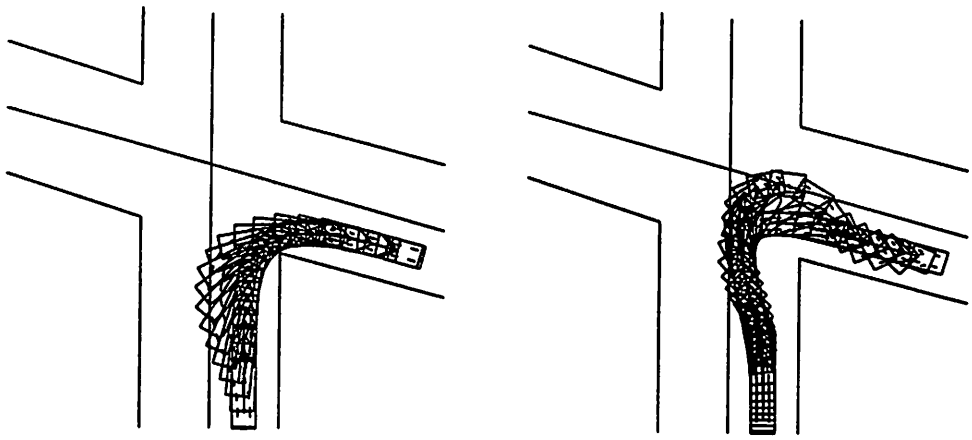
Figures 5.10 and 5.11 show the simulation results of steering the fire truck with and without tiller steering through a 75° right hand turn. These results confirm our findings that the system without tiller steering works harder than with tiller steering. In addition, for the right hand corner trajectories, the system without a tiller crosses over into the lane of on-coming traffic. All of the above maneuvers are smoother for the fire truck, which justifies our initial hypothesis that the tiller adds maneuverability.

5.4 Converting the Extended Fire Truck to Extended Goursat Normal Form

In this section, Algorithm 2 of Section 3.2 is used to transform a mobile robot with five axles and two steering wheels into an extended Goursat normal form

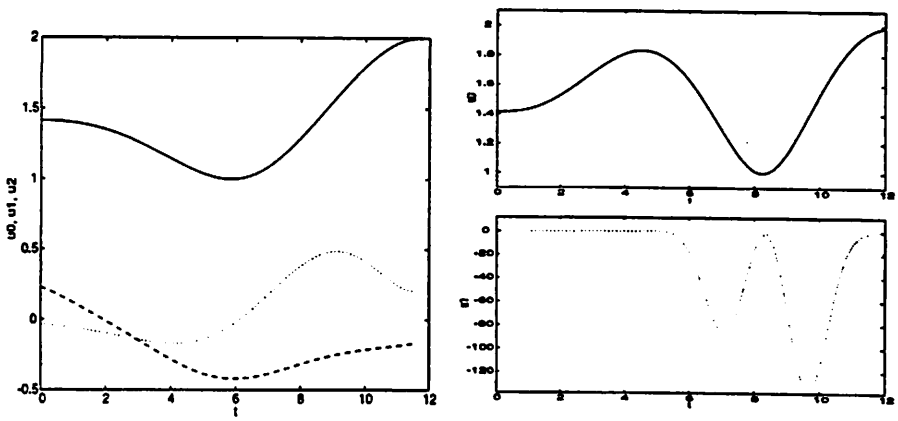
$$I = \{dz^1_{n_1} - z^1_{n_1-1}\, dz^0_0, \ldots,\ dz^1_1 - z^1_0\, dz^0_0,\ dz^2_{n_2} - z^2_{n_2-1}\, dz^0_0, \ldots,\ dz^2_1 - z^2_0\, dz^0_0\}$$

with two towers. Our main goal is to easily find the extended Goursat normal form coordinates, z^j_i. Following the steps of Algorithm 2 exactly for this mobile robot system, this goal cannot be achieved since Step 4 of the algorithm does not give the coordinates for the second



With tiller steering. Without tiller steering.

Figure 5.10: The fire truck with and without tiller steering for a 75° right hand turn in an intersection. The fire truck is steered from an initial position $(x_1, y_1) = (0, 4)$ with the body angles $\theta_1 = \theta_2 = \pi/2$ and steering angles $\phi_1 = \phi_2 = 0$ to a final state $(x_1, y_1) = (9.6, 10.9)$ with body angles $\theta_1 = \theta_2 = -\pi/12$ and steering angles at zero.



With tiller steering. Without tiller steering.

Figure 5.11: The inputs for the fire truck with and without tiller steering for the 75° right hand turn trajectory shown in Figure 5.10. The driving velocity u_0 is the solid line, the steering velocity of the front wheels u_1 is the dotted line, and the steering velocity of the tiller u_2 is the dashed line.

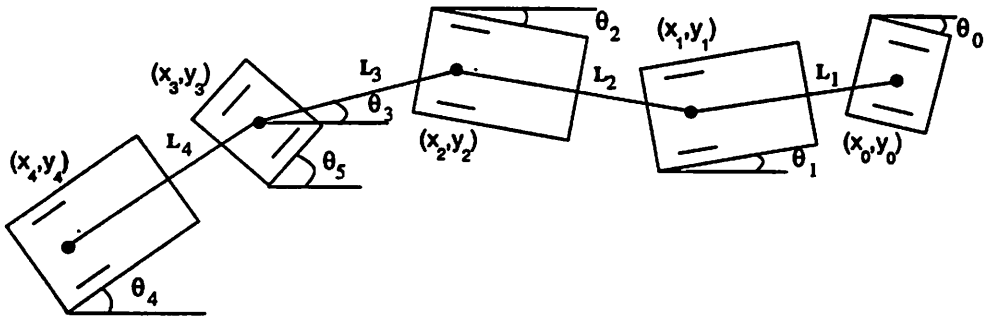


Figure 5.12: Configuration of a five-axle, two-steering mobile robot.

tower. Finding one of these coordinates involves solving a partial differential equation that we do not know how to solve. To resolve this, a modified procedure is suggested involving the use of partial prolongations. This will be discussed in more detail in the following.

Consider now the five-axle, two-steering mobile robot as shown in Figure 5.12. The front *steering train* consists of three axles: the front steering wheel and the next two passive axles. The second steering train consists of the second steering wheel and the passive axle behind it. The one-form constraints for this system, labeling the axles with the constraints α^0 to α^4 from right to left, are

$$\begin{aligned}
\alpha^0 &= \sin\theta_0\, dx_0 - \cos\theta_0\, dy_0 = 0 \\
\alpha^1 &= \sin\theta_1\, dx_1 - \cos\theta_1\, dy_1 = 0 \\
\alpha^2 &= \sin\theta_2\, dx_2 - \cos\theta_2\, dy_2 = 0 \\
\alpha^3 &= \sin\theta_5\, dx_3 - \cos\theta_5\, dy_3 = 0 \\
\alpha^4 &= \sin\theta_4\, dx_4 - \cos\theta_4\, dy_4 = 0\,.
\end{aligned}$$

The Pfaffian system associated with this mobile robot is written as $I = \{\alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4\}$ and the derived flag has the form

$$\begin{aligned}
I = I^{(0)} &= \{\ \alpha^0,\ \alpha^1,\ \alpha^2,\ \alpha^3,\ \alpha^4\} \\
I^{(1)} &= \{\ \ \ \ \alpha^1,\ \alpha^2,\ \ \ \ \alpha^4\} \\
I^{(2)} &= \{\ \ \ \ \ \ \ \alpha^2\} \\
I^{(3)} &= \{\ 0\}\,,
\end{aligned}$$

which shows the two groupings of constraints corresponding to the two steering trains.

Let the state space be parameterized by $q := (x, y, \theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. The Cartesian coordinates of the axles are related by the holonomic constraints

$$x_{i-1} = x_i + L_i\cos\theta_i \qquad y_{i-1} = y_i + L_i\sin\theta_i \quad i = 1,2,3,4\,.$$

Proposition 23 *The five-axle mobile robot with steering on the first and fourth axles, with $L_3 \neq L_4$ as shown in Figure 5.12, whose Pfaffian system is extended by partial prolongation can be transformed into extended Goursat normal form.*

Proof. First the constraints are written with $(x, y) = (x_2, y_2)$ since the α^2 steering train is the longer. The state space $\mathbb{R}^8$ is thus parameterized by $q := (x_2, y_2, \theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. This front steering train's coordinates are constructed as stated in Algorithm 2. In deriving the second steering train's coordinates using $z_0^0 = x_2$, however, we find that the partial differential equation that yields the coordinate z_2^2 cannot be integrated. For the constraint

$$\alpha^4 = \sin\theta_4\, dx - \cos\theta_4\, dy + L_3 \cos(\theta_4 - \theta_3)\, d\theta_3 + L_4\, d\theta_4\ ,$$

the Goursat congruences allow only α^1 and α^2 to be used to scale α^4. We would like to modify α^4 to be of the form

$$\alpha^4 = dz_2^2 - z_1^2\, dz_0^0\ ,$$

which would give the coordinate z_2^2. Using only α^1 and α^2 to scale α^4, the partial differential equation

$$dz_2^2 = \gamma(q) L_3 \cos(\theta_4 - \theta_3)\, d\theta_3 + \gamma(q) L_4\, d\theta_4 \tag{5.15}$$

must be integrated for some integrating factor $\gamma(q)$. Finding the factor $\gamma(q)$ proved to be so difficult in the case $L_3 \neq L_4$ that we opted for a new procedure to find the coordinates[5].

Motivated by our recent work [78] and the literature on partial prolongations of exterior differential systems [66, 67], Algorithm 2 can be improved to handle this case. The basic idea used was introduced in [78], which uses dynamic state feedback to augment the state space of a multiple-steering, multiple-trailer system so that the kinematic equations are easily convertible to multiple-input chained form. The new augmented states are interpreted as "virtual" axles extending in front of the steerable axles (except the lead car). The number of virtual axles added to a steering train equals the total number of passive axles in front of it.

In exterior differential systems, augmenting the system corresponds to adding the new constraints of the virtual axles to the Pfaffian system. Moreover, the derived flag

[5]If $L_3 = L_4$, an integrating factor can be easily found. For let $\gamma\omega := dz_2^2$ in equation (5.15) with $L_3 = L_4 = 1$. Then $d\omega \wedge \omega = 0$, which means there exists an f and g such that $\omega = f dg$.

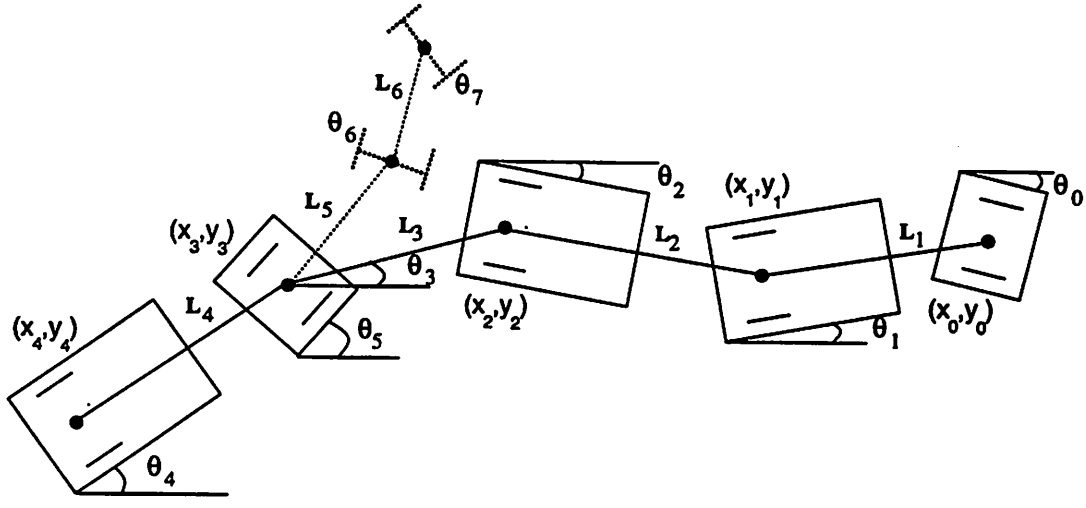


Figure 5.13: Configuration of a five-axle, two-steering mobile robot with virtual axles added in front of the second steerable axle.

changes to our advantage, allowing the use of more forms to modify the constraints to have the structure

$$\gamma(q)\ \bar{\alpha} = dz_i - c(q)\ dz_0^0\ ,$$

from which the coordinate z_{i-1} is assigned to be $c(q)$.

Using this new method, the constraints are now written using the Cartesian coordinates $(x, y) = (x_4, y_4)$ and partial prolongation is used to add two new constraints to the system. These constraints, ω^1 and ω^2, correspond to the two virtual axles with angles θ_6 and θ_7, respectively, added onto the second steering train as shown in Figure 5.13.

The state space $\mathbb{R}^8$ is now parameterized by $q := (x_4, y_4, \theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ and the derived flag is computed to be

$$\begin{aligned}
I = I^{(0)} &= \{\ \alpha^0,\ \alpha^1,\ \alpha^2,\ \omega^2,\ \omega^1,\ \alpha^3,\ \alpha^4\} \\
I^{(1)} &= \{\ \ \ \ \alpha^1,\ \alpha^2,\ \ \ \ \omega^1,\ \alpha^3,\ \alpha^4\} \\
I^{(2)} &= \{\ \ \ \ \ \ \ \alpha^2,\ \ \ \ \ \ \alpha^3,\ \alpha^4\} \\
I^{(3)} &= \{\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \alpha^4\} \\
I^{(4)} &= \{\ 0\}
\end{aligned}$$

showing the two steering trains, where the constraints are written as

$$\begin{aligned}
\alpha^0 &= \sin\theta_0\, dx_4 - \cos\theta_0\, dy_4 - L_4\cos(\theta_0-\theta_4)\, d\theta_4 - L_3\cos(\theta_0-\theta_3)\, d\theta_3 \\
&\quad -L_2\cos(\theta_0-\theta_2)\, d\theta_2 - L_1\cos(\theta_0-\theta_1)\, d\theta_1 \\
\alpha^1 &= \sin\theta_1\, dx_4 - \cos\theta_1\, dy_4 - L_4\cos(\theta_1-\theta_4)\, d\theta_4 - L_3\cos(\theta_1-\theta_3)\, d\theta_3 \\
&\quad -L_2\cos(\theta_1-\theta_2)\, d\theta_2 \\
\alpha^2 &= \sin\theta_2\, dx_4 - \cos\theta_2\, dy_4 - L_4\cos(\theta_2-\theta_4)\, d\theta_4 - L_3\cos(\theta_2-\theta_3)\, d\theta_3 \\
\alpha^3 &= \sin\theta_5\, dx_4 - \cos\theta_5\, dy_4 - L_4\cos(\theta_5-\theta_4)\, d\theta_4 \\
\alpha^4 &= \sin\theta_4\, dx_4 - \cos\theta_4\, dy_4 \\
\omega^1 &= \sin\theta_6\, dx_4 - \cos\theta_6\, dy_4 - L_5\cos(\theta_6-\theta_5)\, d\theta_5 - L_4\cos(\theta_6-\theta_4)\, d\theta_4 \\
\omega^2 &= \sin\theta_7\, dx_4 - \cos\theta_7\, dy_4 - L_6\cos(\theta_7-\theta_6)\, d\theta_6 - L_5\cos(\theta_7-\theta_5)\, d\theta_5 \\
&\quad -L_4\cos(\theta_7-\theta_4)\, d\theta_4\ . \qquad (5.16)
\end{aligned}$$

First choose $\pi = dx_4$, which is integrable, and modify the constraints in the first tower, which is the α^4 tower. The Goursat congruences for this system are

$$\begin{aligned}
d\alpha^4 &= dx_4 \wedge \alpha^3 \mod \alpha^4 \\
d\alpha^3 &= dx_4 \wedge \omega^1 \mod \alpha^2, \alpha^3, \alpha^4 \\
d\omega^1 &= dx_4 \wedge \omega^2 \mod \alpha^1, \alpha^2, \alpha^3, \alpha^4, \omega^1 \\
d\omega^2 &\neq 0 \mod I \\
d\alpha^2 &= dx_4 \wedge \alpha^1 \mod \alpha^2, \alpha^3, \alpha^4 \\
d\alpha^1 &= dx_4 \wedge \alpha^0 \mod \alpha^1, \alpha^2, \alpha^3, \alpha^4, \omega^1 \\
d\alpha^0 &\neq 0 \mod I\ . \qquad (5.17)
\end{aligned}$$

Following the notation of Algorithm 2, set $\omega_1^1 := \alpha^4$ and scale this constraint as

$$\bar{\omega}_1^1 = -\sec\theta_4\, \omega_1^1 = dy_4 - \tan\theta_4\, dx_4\ . \qquad (5.18)$$

This one-form has rank 1 since

$$\begin{aligned}
d\bar{\omega}_1^1 \wedge \bar{\omega}_1^1 = -\sec^2\theta_4\, d\theta_4 \wedge dx_4 \wedge dy_4 &\neq 0 \\
d\bar{\omega}_1^1 \wedge d\bar{\omega}_1^1 \wedge \bar{\omega}_1^1 &= 0\ .
\end{aligned}$$

Therefore from the solution to Pfaff's problem, $f_1 = x_4$ satisfies

$$d\bar{\omega}_1^1 \wedge \bar{\omega}_1^1 \wedge df_1 = 0 \quad \text{and} \quad \bar{\omega}_1^1 \wedge df_1 \neq 0$$

and $f_2 = y_4$ satisfies

$$\bar{\omega}_1^1 \wedge df_1 \wedge df_2 = 0 \quad \text{and} \quad df_1 \wedge df_2 \neq 0 .$$

Setting the *generating* coordinate $z_0^0 = f_1 = x_4$ and the coordinate for the bottom of the first chain $z_4^1 = f_2 = y_4$, the next coordinate is read as the coefficient of dx_4 in equation (5.18) as $z_3^1 = \tan\theta_4$.

The second constraint $\omega_2^1 := \alpha^3$ in the first tower is used to find the next coordinate, z_2^1. From the Goursat congruences (5.17), α^2 and α^4 can be used to transform ω_2^1 into the form

$$\omega_2^1 = dz_3^1 - z_2^1 \, dz_0^0 .$$

The one-form ω_2^1 is scaled, $\bar{\omega}_1^1$ is used to eliminate the dy_4 term

$$\begin{aligned}
\bar{\omega}_2^1 &= -\frac{1}{L_4}\sec^2\theta_4 \sec(\theta_5 - \theta_4)\,(\omega_2^1 + \cos\theta_5\ \bar{\omega}_1^1) \\
&= \sec^2\theta_4 d\theta_4 - \frac{1}{L_4}\sec^3\theta_4 \tan(\theta_5 - \theta_4)\ dx_4 \\
&= dz_3^1 - \frac{1}{L_4}\sec^3\theta_4 \tan(\theta_5 - \theta_4)\ dz_0^0 ,
\end{aligned}$$

and the coefficient of dz_0^0 is set to be z_2^1.

The third constraint $\omega_3^1 := \omega^1$ in the first tower is modified as follows, noting that

$$\begin{aligned}
dz_2^1 &= \frac{1}{L_4}\sec^3\theta_4 \sec^2(\theta_5 - \theta_4)\ d\theta_5 + \frac{1}{L_4}\,(3\sec^3\theta_4 \tan\theta_4 \tan(\theta_5 - \theta_4) \\
&\quad - \sec^3\theta_4 \sec^2(\theta_5 - \theta_4))\ d\theta_4 \\
&:= \beta_5(q)\ d\theta_5 + \beta_4(q)\ d\theta_4 .
\end{aligned}$$

Define the coefficients

$$\begin{aligned}
k_0(q) &= \frac{1}{L_5}\sec(\theta_6 - \theta_5)\beta_5(q) \\
k_1(q) &= \frac{1}{L_4}\sec(\theta_5 - \theta_4)\,(k_0(q)L_4\cos(\theta_6 - \theta_4) + \beta_4(q)) \\
k_2(q) &= (k_0(q)\cos\theta_6 + k_1(q)\cos\theta_5)\sec\theta_4 .
\end{aligned}$$

The one-form ω_3^1 is scaled to match the dz_2^1 coefficient of $d\theta_5$, ω_2^1 is used to modify the coefficient of $d\theta_4$ to match that of dz_2^1, and ω_1^1 is used to eliminate the dy_4 term

$$\begin{aligned}
\bar{\omega}_3^1 &= k_0(q)\ \omega_3^1 + k_1(q)\ \omega_2^1 + k_2(q)\ \omega_1^1 \\
&= dz_2^1 - (k_0\sin\theta_6 + k_1\sin\theta_5 + k_2\sin\theta_4)\ dz_0^0 ,
\end{aligned}$$

from which the coordinate z_1^1 is read as the coefficient of the dz_0^0 term.

The fourth constraint $\omega_4^1 := \omega^2$ in the first tower is modified in a similar manner, using all of the other constraints to get the form

$$\bar{\omega}_4^1 = dz_1^1 - z_0^1\, dz_0^0 \ ,$$

from which the coefficient of dz_0^0 is assigned to be the last coordinate in the first chain, z_0^1. The explicit calculations are omitted here. Thus, all of the Goursat congruences (5.17) are satisfied for the first tower.

In the second tower, the first constraint $\omega_1^2 := \alpha^2$ has rank 1 since

$$d\omega_1^2 \wedge \omega_1^2 \neq \ 0 \quad \text{and} \quad d\omega_1^2 \wedge d\omega_1^2 \wedge \omega_1^2 = \ 0 \ .$$

Therefore from Pfaff's problem it must be that $g_1 = f_1 = x_4$, as in the first tower, since the same generator is used in the second tower. The function g_1 satisfies

$$d\omega_1^2 \wedge \omega_1^2 \wedge dg_1 = \ 0 \quad \text{and} \quad \omega_1^2 \wedge dg_1 \neq \ 0 \ .$$

Define the coefficients

$$\begin{aligned} k_0(q) &:= \frac{1}{L_3}\sec(\theta_2 - \theta_3) \\ k_1(q) &:= \frac{1}{L_4}\sec(\theta_5 - \theta_4)\,(k_0(q) L_4 \cos(\theta_2 - \theta_4)) \\ k_2(q) &:= (k_0(q)\cos\theta_2 + k_1(q)\cos\theta_5)\sec\theta_4 \ . \end{aligned}$$

First, ω_1^2 is scaled to have a term $d\theta_3$. Then, ω_2^1 is used to eliminate the $d\theta_4$ term and ω_1^1 is used to eliminate the dy_4 term:

$$\begin{aligned} \bar{\omega}_1^2 &= k_0(q)\ \omega_1^2 + k_1(q)\ \omega_2^1 + k_2(q)\ \omega_1^1 \\ &= dz_3^2 - (k_0 \sin\theta_2 + k_1 \sin\theta_5 + k_2 \sin\theta_4)\ dz_0^0 \ , \end{aligned}$$

giving the coordinate $z_3^2 = \theta_3$. The coordinate z_2^2 is the coefficient of the dz_0^0 term. The solution to Pfaff's problem gives $g_2 = \theta_3 = z_3^2$, which satisfies

$$\bar{\omega}_1^2 \wedge dg_1 \wedge dg_2 = \ 0 \quad \text{and} \quad dg_1 \wedge dg_2 \neq \ 0 \ .$$

The next two constraints, $\omega_2^2 := \alpha^1$ and $\omega_3^2 := \alpha^0$ are modified in a similar manner as above. The coordinates z_1^2 and z_0^2 are found from these calculations, respectively.

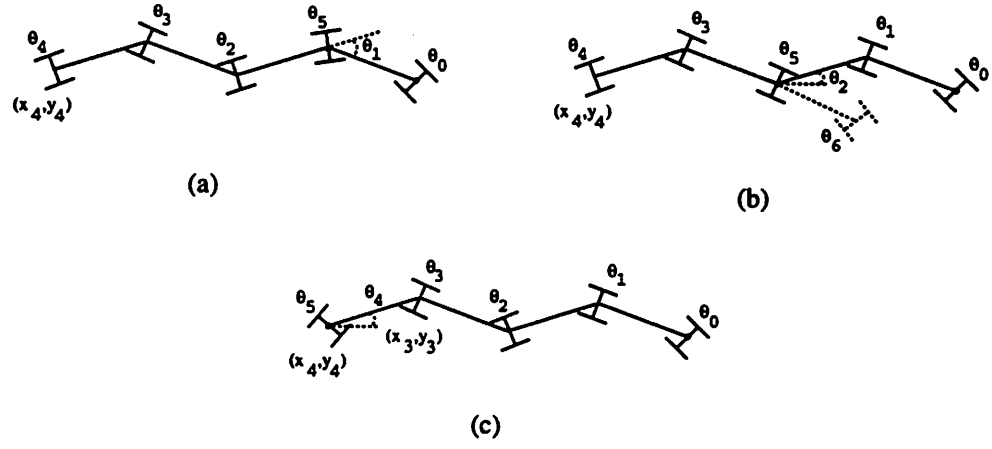


Figure 5.14: The three other cases to consider for a five-axle, two-steering mobile robot. The placement of the second steering axle changes the analysis.

The resulting coordinates transform the system into the extended Goursat normal form:

$$\begin{aligned} I &= \{\bar{\omega}_1^1,\ \bar{\omega}_2^1,\ \bar{\omega}_3^1,\ \bar{\omega}_4^1,\ \bar{\omega}_1^2,\ \bar{\omega}_2^2,\ \bar{\omega}_3^2\} \\ &= \{dz_4^1 - z_3^1\, dz_0^0,\ dz_3^1 - z_2^1\, dz_0^0,\ dz_2^1 - z_1^1\, dz_0^0,\ dz_1^1 - z_0^1\, dz_0^0, \\ &\quad dz_3^2 - z_2^2\, dz_0^0,\ dz_2^2 - z_1^2\, dz_0^0,\ dz_1^2 - z_0^2\, dz_0^0\}\ . \end{aligned}$$

□

The three other steering configurations for the example of a five-axle, two-steering mobile robot, as shown in Figure 5.14, are now considered.

Proposition 24 *The five-axle mobile robot with steering on the first two axles, as shown in Figure 5.14 (a), can be transformed into extended Goursat normal form without partial prolongation.*

Proof. The derived flag for this mobile robot, labeling the axles with the constraints α^0 to α^4 from right to left, is

$$\begin{aligned} I = I^{(0)} &= \{\ \alpha^0,\ \alpha^1,\ \alpha^2,\ \alpha^3,\ \alpha^4\} \\ I^{(1)} &= \{\ \alpha^2,\ \alpha^3,\ \alpha^4\} \\ I^{(2)} &= \{\ \alpha^3,\ \alpha^4\} \\ I^{(3)} &= \{\ \alpha^4\} \\ I^{(4)} &= \{\ 0\}\ . \end{aligned}$$

in Figure 5.14(b). This corresponds to the new constraint ω^1. The new derived flag is

$$\begin{aligned} I = I^{(0)} &= \{ \ \alpha^0, \ \alpha^1, \ \omega^1, \ \alpha^2, \ \alpha^3, \ \alpha^4\} \\ I^{(1)} &= \{ \qquad \alpha^1, \qquad \alpha^2, \ \alpha^3, \ \alpha^4\} \\ I^{(2)} &= \{ \qquad\qquad\qquad \alpha^3, \ \alpha^4\} \\ I^{(3)} &= \{ \qquad\qquad\qquad\qquad \alpha^4\} \\ I^{(4)} &= \{ \ 0\} , \end{aligned} \tag{5.21}$$

which "pushes down" the second steering train so that α^1 can now use α^2 in addition to α^3 and α^4 for modification into the form

$$\alpha^1 = dz_2^2 - z_1^2 dz_0^0 \ .$$

The first tower's (second steering train's) coordinates are found as in Algorithm 2, with one extra coordinate due to the new ω^1 constraint. The second tower's (front steering train's) coordinates are found as in Algorithm 2, using the new Goursat congruences with the extra form constraint. The Goursat congruences are satisfied with $\pi = dx_4$ and Pfaff's problem yields $z_0^0 = x_4$, $z_4^1 = y_4$ and $z_2^2 = \theta_2$.

□

Proposition 26 *The five-axle mobile robot with steering on the first and last axles, as shown in Figure 5.14 (c), can be transformed into extended Goursat normal form without partial prolongation.*

Proof. The derived flag in this case is

$$\begin{aligned} I = I^{(0)} &= \{ \ \alpha^0, \ \alpha^1, \ \alpha^2, \ \alpha^3, \ \alpha^4\} \\ I^{(1)} &= \{ \qquad \alpha^1, \ \alpha^2, \ \alpha^3\} \\ I^{(2)} &= \{ \qquad\qquad \alpha^2, \ \alpha^3\} \\ I^{(3)} &= \{ \qquad\qquad\qquad \alpha^3\} \\ I^{(4)} &= \{ \ 0\} , \end{aligned}$$

which shows four axles in the front steering train and one axle, α^4, in the second steering train. Write the constraints with $(x, y) = (x_3, y_3)$ since the α^3 tower is the longer. The state space $\mathbb{R}^8$ is thus parameterized by $q := (x_3, y_3, \theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. The first tower is then the second steering train and Algorithm 2 can be used to find the coordinates for both towers. The extended Goursat congruences are satisfied with $\pi = dx_3$ and Pfaff's problem yields $z_0^0 = x_3$, $z_4^1 = y_3$ and $z_1^2 = \theta_4$.

□

Remark. In essence, partial prolongation is used to "push down" those towers in the derived flag that correspond to the steering trains that have extra virtual axles added on. For example in case (b) above, the second tower in the derived flag (5.19) consists of the third, fourth and fifth axles. The second tower is structured such that the last passive axle in the steering train is the one whose constraint drops through to the system $I^{(2)}$, the second to last axle's constraint drops through to $I^{(1)}$, while the constraint of the steerable axle only appears in the top system $I^{(0)}$. Thus, when one virtual axle is added onto the second steering train, this becomes the new virtual steering wheel, causing the real steering wheel to be treated as a passive axle. This results in the new derived flag shown in equation (5.21). The net gain of using partial prolongation is that more constraints can be used to modify the one-forms in the second tower of the derived flag of the Pfaffian system so that the extended Goursat congruences can be satisfied.

In summary, a methodology for finding a coordinate transformation to extended Goursat normal form for multiple-steering, multiple-trailer systems is as follows.

Methodology using Algorithm 2

Step 1 Compute the derived flag using the given constraints.

Step 2 Separate the derived flag into towers corresponding to the steering trains, i.e., all of the axles in a steering train are in the same tower.

Step 3 If partial prolongation is needed, add new constraints to those towers corresponding to those steering trains that have virtual axles added to them.

Step 4 Using the longest tower as the first tower, the (x, y) coordinate is the coordinate of the axle whose constraint drops to the bottom of this tower. Pfaff's problem gives $f_1 = x$ and $f_2 = y$, which correspond to the z_0^0 coordinate and the last coordinate, $z_{n_1}^1$, in the first chain of the extended Goursat normal form, respectively. Follow Algorithm 2 using $\pi = dx$.

Step 5 For the j^{th} tower, Pfaff's problem requires that $g_1 = f_1 = x$ still, and g_2 is chosen to be the hitch angle between the j^{th} and $(j-1)^{st}$ steering trains. This hitch angle then becomes the last coordinate, $z_{n_j}^j$, in the j^{th} chain of the extended Goursat normal form. Follow Algorithm 2 using $\pi = dx$.

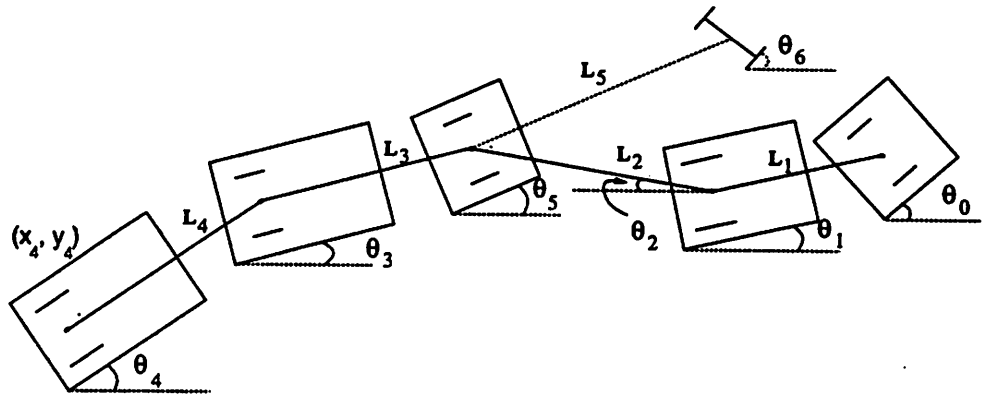


Figure 5.15: The five-axle, two-steering mobile robot used in the simulations.

5.5 Simulation of the Extended Fire Truck

In this section, numerical simulation results for the five-axle, two-steering mobile robot system described in Section 5.4 (as case (b)) and shown in Figure 5.15 are presented. This simulation is not as extensive as for the fire truck, since only two passive trailers are being added to that system. The coordinate transformation to chained form is similar to that of the fire truck, choosing $z_0^0 = x_4$, $z_4^1 = y_4$, and $z_2^2 = \theta_2$ as the bottoms of the chains and using equation (3.12) from Section 3.1 to find the rest of the coordinates.

Once the kinematic equations of the extended fire truck are in multiple-input chained form, which includes the extra state θ_6 from the virtual axle, the system can be steered by one of the algorithms discussed in Chapter 4. For the following simulations, the lengths of the hitches are $L_1 = L_3 = L_4 = 5$, $L_2 = 3$ and $L_5 = 1$.

The first trajectory is the parallel parking maneuver with initial point $(x, y) = (0, 20)$ and final point $(x_4, y_4) = (0, 0)$ (the coordinates of the midpoint of the last axle) and all of the body angles (including the virtual axle at angle θ_6) aligned with the horizontal axis in both the initial and final configurations. The simulation is performed on the system equations in chained form using polynomial control inputs. As noted in Section 4.2, polynomial inputs are not immediately suited to this type of trajectory since the time needed to steer the system, computed from equation (4.6), would be zero and the algorithm would fail. Therefore the trajectory is planned in two steps, choosing an intermediate point $(x, y) = (30, 10)$.

Figure 5.16 shows the resulting parallel parking trajectory. The path taken by the

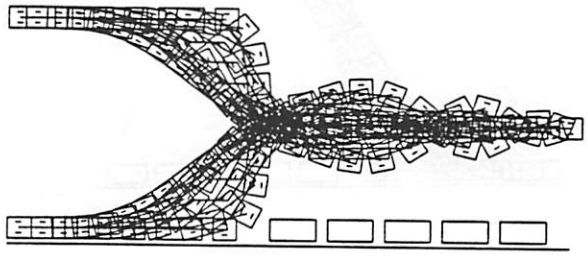


Figure 5.16: The parallel parking trajectory for the extended fire truck using polynomial inputs. The vehicle starts at $y_4 = 20$ and ends at zero, with all other coordinates starting and ending at zero.

virtual axle is not shown. The second path simulated is shown in Figure 5.17. This shows the system backing up from a far distance of $(x_4, y_4) = (35, 35)$ into the same parking space.

5.6 Mobile Robots with Kingpin Hitching

In this section, the class of mobile robots with off-axle hitches is studied. The goal is to determine if these vehicles are convertible to chained form and if they are controllable. In the following, a two-axle car pulling one trailer attached by a kingpin hitch is transformed into Goursat normal form, hence chained form. When there are two or more trailers, however, the system cannot be transformed into chained form. In addition, the procedure for transforming the fire truck with axle-to-axle hitching into chained form, as presented in Section 5.1, applies with no modification when the hitch point is off the axle.

Recall from Section 5.3, that a two-axle car pulling one trailer attached by an axle-to-axle hitch (the fire truck without tiller steering) is transformable to chained form. A two-axle car pulling N trailers with axle-to-axle hitching was shown to be convertible to chained form in [68, 76].

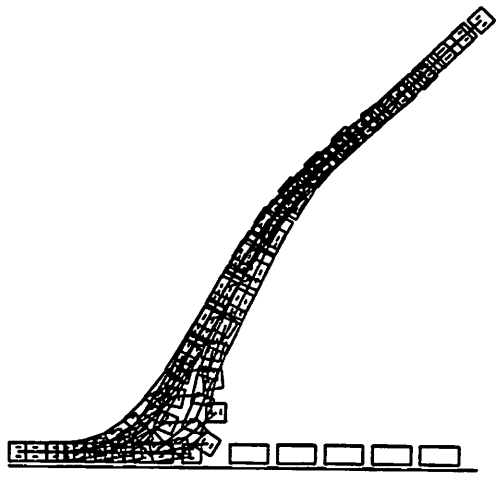


Figure 5.17: The back-up trajectory for the extended fire truck using polynomial inputs. The vehicle starts at $(x_4, y_4) = (35, 35)$ and ends at zero, with all other coordinates starting and ending at zero.

5.6.1 A Car Pulling One Trailer with Kingpin Hitching

First consider the system of a two-axle car pulling one trailer that is attached by a kingpin hitch as shown in Figure 5.18. In [65] (see also [45]), the linearizing outputs were given that show the system is *flat*, essentially meaning that there exists local coordinates to transform the system into Goursat normal form. The explicit calculations proving the system can be transformed into Goursat normal form were omitted from both [65] and [45] and will be presented here.

Using the notation shown in Figure 5.18 with absolute angles, the kinematic model of this system can be derived. The non-slipping constraints for the three axles from front to back are written as one-forms

$$\begin{aligned} \alpha^0 &= \dot{x}_0 \sin\theta_0 - \dot{y}_0 \cos\theta_0 \\ \alpha^1 &= \dot{x}_1 \sin\theta_1 - \dot{y}_1 \cos\theta_1 \\ \alpha^2 &= \dot{x}_2 \sin\theta_2 - \dot{y}_2 \cos\theta_2 \ . \end{aligned}$$

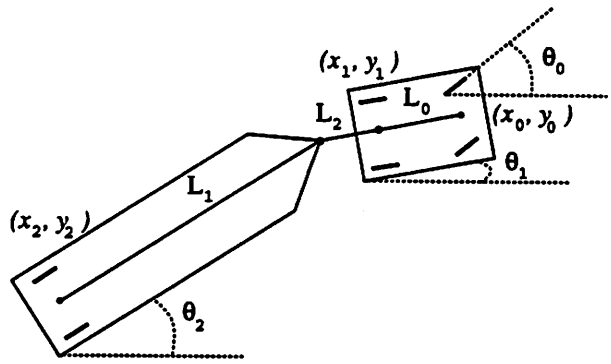


Figure 5.18: Configuration of a two-axle car pulling one trailer attached by a kingpin hitch. This system can be converted into chained form.

Using the equalities

$$x_0 = x_1 + L_0\cos\theta_1 \qquad y_0 = y_1 + L_0\sin\theta_1$$
$$x_2 = x_1 - L_1\cos\theta_2 - L_2\cos\theta_1 \qquad y_2 = y_1 - L_1\sin\theta_2 - L_2\sin\theta_1$$

the three constraints can be expressed in coordinates in $\mathbb{R}^5$ as as

$$\begin{aligned}
\alpha^0(x) &= [\ \sin\theta_0 \ \ -\cos\theta_0 \ \ 0 \ \ -L_0\cos(\theta_0-\theta_1) \ \ 0\ ] \\
\alpha^1(x) &= [\ \sin\theta_1 \ \ -\cos\theta_1 \ \ 0 \ \ 0 \ \ 0\ ] \\
\alpha^2(x) &= [\ \sin\theta_2 \ \ -\cos\theta_2 \ \ 0 \ \ L_2\cos(\theta_2-\theta_1) \ \ L_1\ ]
\end{aligned}$$

where the state space is parameterized by $x := (x_1, y_1, \theta_0, \theta_1, \theta_2) \in \mathbb{R}^5$.

Since the state space is of dimension five and there are three constraints on the system, two vector fields g_0 and g_1 can be found such that $\alpha^i(x) \cdot g_j(x) = 0$, for $i = 1, 2, 3$ and $j = 1, 2$. A simple calculation shows that the following vector fields form a basis for $\Delta(x) := \text{span}\{g_0(x), g_1(x)\}$:

$$g_0 = \begin{pmatrix} \cos\theta_1 \\ \sin\theta_1 \\ 0 \\ \frac{1}{L_0}\tan(\theta_0-\theta_1) \\ -\frac{1}{L_1}\left(\sin(\theta_2-\theta_1) + \frac{L_2}{L_0}\cos(\theta_2-\theta_1)\tan(\theta_0-\theta_1)\right) \end{pmatrix}, \quad g_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

Thus, the kinematic model of the two-axle car pulling one trailer with kingpin hitching can be written as

$$\dot{x} = g_0(x)u_0 + g_1(x)u_1$$

where the input u_0 corresponds to the forward driving velocity of the car and u_1 corresponds to the steering velocity of the front wheels of the car.

For the following $f_0 := \sec\theta_1 g_0$ and $f_1 := g_1$ are used as a basis for Δ. The filtrations E_i and G_i are formed as follows

$$\begin{aligned} E_0 &= \Delta & G_0 &= \Delta \\ E_1 &= E_0 + [E_0, E_0] & G_1 &= G_0 + [G_0, G_0] \\ E_2 &= E_1 + [E_1, E_1] & G_2 &= G_1 + [G_1, G_0] \\ E_3 &= E_2 + [E_2, E_2] & G_3 &= G_2 + [G_2, G_0] \end{aligned}$$

and Theorem 14 and Corollary 15 from Section 2.2 are used to show that the system can be transformed into chained form.

Using the symbolic manipulation software package *Mathematica*,

$$\begin{aligned} \dim E_0 = \dim G_0 &= \dim \operatorname{span}\{f_0, f_1\} = 2 \\ \dim E_1 = \dim G_1 &= \dim \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1\} = 3 \\ \dim E_2 = \dim G_2 &= \dim \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1, [f_1, \operatorname{ad}_{f_0} f_1]\} = 4 \end{aligned}$$

since $[f_1, \operatorname{ad}_{f_0} f_1] = 2\tan(\theta_0 - \theta_1)\operatorname{ad}_{f_0} f_1$. Using $E_2 = \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1\}$, E_3 is formed as follows

$$E_3 = \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1, \operatorname{ad}^3_{f_0} f_1, [f_1, \operatorname{ad}^2_{f_0} f_1], [\operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1]\} .$$

It has dimension five since $[f_1, \operatorname{ad}^2_{f_0} f_1]$ is in the span of $\{f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1\}$ and $[\operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1]$ is in the span of $\{f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1, \operatorname{ad}^3_{f_0} f_1\}$. Additionally,

$$G_3 = \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1, \operatorname{ad}^3_{f_0} f_1, [f_1, \operatorname{ad}^2_{f_0} f_1]\}$$

has dimension five from above. Thus, since

$$\dim E_i = \dim G_i = i + 2 \quad i = 0, 1, 2, 3$$

Corollary 15 from Section 2.2 gives that there exists a feedback transformation that transforms the system into chained form.

Rouchon *et al.* [65] geometrically computed the flat, or linearizing, outputs $h = (h_1, h_2)$ for this system to be

$$\begin{aligned} h_1 &= x_1 + L_1\cos\theta_2 + \rho(\theta_1 - \theta_2)\frac{L_1\sin\theta_2 - L_2\sin\theta_1}{\sqrt{L_2^2 + L_1^2 - 2L_1L_2\cos(\theta_1 - \theta_2)}} \\ h_2 &= y_1 + L_1\sin\theta_2 + \rho(\theta_1 - \theta_2)\frac{L_1\cos\theta_1 - L_2\cos\theta_2}{\sqrt{L_2^2 + L_1^2 - 2L_1L_2\cos(\theta_1 - \theta_2)}} \end{aligned} \tag{5.22}$$

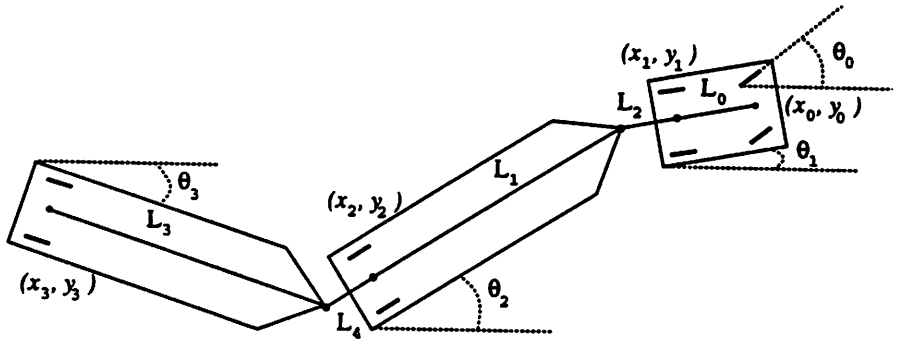


Figure 5.19: Configuration of a two-axle car pulling two trailers attached by kingpin hitches. This system cannot be converted into chained form.

with

$$\rho(\theta_1 - \theta_2) := L_1 L_2 \int_{\pi}^{2\pi+\theta_1-\theta_2} \frac{\cos\sigma}{\sqrt{L_2^2 + L_1^2 - 2L_1 L_2 \cos\sigma}} d\sigma \ .$$

Rouchon found, through a numerical calculation, that the local coordinates $z_0^0 = h_1$ and $z_3^1 = h_2$ yield the chained form system

$$\begin{aligned} \dot{z}_0^0 = v_0 \qquad \dot{z}_0^1 &= v_1 \\ \dot{z}_1^1 &= z_0^1 v_0 \\ \dot{z}_2^1 &= z_1^1 v_0 \\ \dot{z}_3^1 &= z_2^1 v_0 \ . \end{aligned}$$

The key numerical calculation was that

$$z_2^1 = \dot{z}_3^1 / z_0^0 = (a\sin\alpha - b\sin\beta)/(a\cos\alpha - b\cos\beta) \ ,$$

from which the coordinates z_1^1 and z_0^1 could be found by differentiation.

5.6.2 A Car Pulling Two Trailers with Kingpin Hitching

Consider the system of a two-axle car pulling two trailers attached by kingpin hitches as shown in Figure 5.19. In [65] it was mentioned that for two or more trailers, this system cannot be transformed into Goursat normal form; the explicit calculations are derived here.

Using the notation shown in Figure 5.19, the four non-slipping constraints are written from front to back

$$\begin{aligned}
\alpha^0 &= \dot{x}_0 \sin\theta_0 - \dot{y}_0 \cos\theta_0 \\
\alpha^1 &= \dot{x}_1 \sin\theta_1 - \dot{y}_1 \cos\theta_1 \\
\alpha^2 &= \dot{x}_2 \sin\theta_2 - \dot{y}_2 \cos\theta_2 \\
\alpha^3 &= \dot{x}_3 \sin\theta_3 - \dot{y}_3 \cos\theta_3
\end{aligned}$$

in coordinates in $\mathbb{R}^6$ as

$$\begin{aligned}
\alpha^0(x) &= [\ \sin\theta_0 \quad -\cos\theta_0 \quad 0 \quad -L_0\cos(\theta_0-\theta_1) \quad 0 \quad 0\ ] \\
\alpha^1(x) &= [\ \sin\theta_1 \quad -\cos\theta_1 \quad 0 \quad 0 \quad 0 \quad 0\ ] \\
\alpha^2(x) &= [\ \sin\theta_2 \quad -\cos\theta_2 \quad 0 \quad L_2\cos(\theta_2-\theta_1) \quad L_1 \quad 0\ ] \\
\alpha^3(x) &= [\ \sin\theta_3 \quad -\cos\theta_3 \quad 0 \quad L_2\cos(\theta_3-\theta_1) \quad (L_1+L_4)\cos(\theta_3-\theta_2) \quad L_3\ ]
\end{aligned}$$

by using the constraints

$$\begin{aligned}
x_0 &= x_1 + L_0\cos\theta_1 & y_0 &= y_1 + L_0\sin\theta_1 \\
x_2 &= x_1 - L_1\cos\theta_2 - L_2\cos\theta_1 & y_2 &= y_1 - L_1\sin\theta_2 - L_2\sin\theta_1 \\
x_3 &= x_2 - L_4\cos\theta_2 - L_3\cos\theta_3 & y_3 &= y_1 - L_4\sin\theta_2 - L_3\sin\theta_3\ .
\end{aligned}$$

The state space is parameterized by $x := (x_1, y_1, \theta_0, \theta_1, \theta_2, \theta_3) \in \mathbb{R}^6$.

A basis for the distribution annihilated by the α^i contains $g_1 = (0,0,1,0,0,0)'$ and

$$g_0 = \begin{pmatrix}
\cos\theta_1 \\
\sin\theta_1 \\
0 \\
\frac{1}{L_0}\tan(\theta_0-\theta_1) \\
-\frac{1}{L_1}\left(\sin(\theta_2-\theta_1) + \frac{L_2}{L_0}\cos(\theta_2-\theta_1)\tan(\theta_0-\theta_1)\right) \\
\frac{-1}{L_3}\left(\sin(\theta_3-\theta_1) + \frac{L_2}{L_0}\cos(\theta_3-\theta_1)\tan(\theta_0-\theta_1) + \beta(L_1+L_4)\cos(\theta_3-\theta_2)\right)
\end{pmatrix} \tag{5.23}$$

where $\beta = -\frac{1}{L_1}\left(\sin(\theta_2-\theta_1) + \frac{L_2}{L_0}\cos(\theta_2-\theta_1)\tan(\theta_0-\theta_1)\right)$. Thus, the kinematic model of the two-axle car pulling two trailers with kingpin hitching can be written as

$$\dot{x} = g_0(x)u_0 + g_1(x)u_1$$

where the input u_0 corresponds to the forward driving velocity of the car and u_1 corresponds to the steering velocity of the front wheels of the car.

Again $f_0 := \sec\theta_1 g_0$ and $f_1 := g_1$ are chosen as a basis for Δ and the recursive and iterative filtrations E_i and G_i are formed as follows

$$\begin{aligned} E_0 &= \Delta & G_0 &= \Delta \\ E_i &= E_0 + [E_{i-1}, E_{i-1}] & G_i &= G_0 + [G_0, G_{i-1}] \qquad i = 1, \ldots, n-2 \,. \end{aligned}$$

Using *Mathematica*,

$$\begin{aligned} \dim E_0 = \dim G_0 &= \dim \operatorname{span}\{f_0, f_1\} = 2 \\ \dim E_1 = \dim G_1 &= \dim \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1\} = 3 \\ \dim E_2 = \dim G_2 &= \dim \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1, [f_1, \operatorname{ad}_{f_0} f_1]\} = 4 \end{aligned}$$

since $[f_1, \operatorname{ad}_{f_0} f_1] = 2\tan(\theta_0 - \theta_1)\operatorname{ad}_{f_0} f_1$. Using $E_2 = \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1\}$,

$$E_3 = \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1, \operatorname{ad}^3_{f_0} f_1, [f_1, \operatorname{ad}^2_{f_0} f_1], [\operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1]\} \tag{5.24}$$

has dimension six. But,

$$G_3 = \operatorname{span}\{f_0, f_1, \operatorname{ad}_{f_0} f_1, \operatorname{ad}^2_{f_0} f_1, \operatorname{ad}^3_{f_0} f_1, [f_1, \operatorname{ad}^2_{f_0} f_1]\} \tag{5.25}$$

only has dimension five. Thus, since

$$\begin{aligned} \dim E_i &= \dim G_i = i + 2 \quad i = 0, 1, 2 \\ \dim E_3 &= 6 \neq \dim G_3 = 5 \,, \end{aligned}$$

Corollary 15 shows that there does not exists a feedback transformation that converts the system to chained form. Extending the same argument, the mobile robot system with more than two trailers with kingpin hitching cannot be transformed into chained form.

Remark. It is conjectured that in the limit as the hitch length, L_4, between the two trailers goes to zero, the dimension of the distribution $E_3(L_4)$ in equation (5.24) approaches the dimension of the distribution G_3 in equation (5.25), allowing for this system to be transformed into chained form.

Remark. A robot is said to be *completely controllable* if it can reach every point in its free configuration space within the same connected component of its current configuration. Controllability of an N-body mobile robot system with the standard axle-to-axle hitching is proven by Laumond [36]. Controllability of a two-body mobile robot system with the

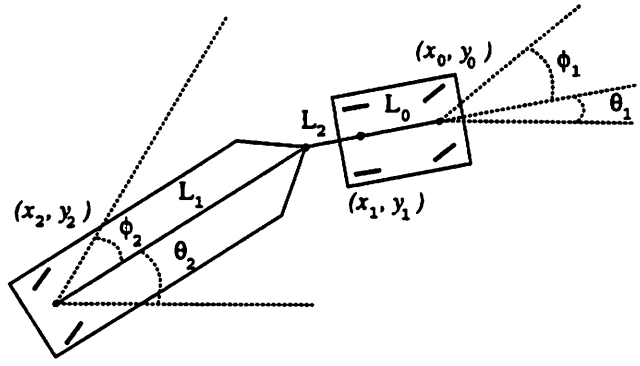


Figure 5.20: Configuration of the fire truck with off-axle hitching.

non-standard kingpin hitching is also shown by Laumond [37]. All of the systems mentioned in this section are controllable, even though they cannot be transformed into chained form, since we can find n linearly independent vector fields that span $\mathbb{R}^n$, namely

$$f_0, f_1, \mathrm{ad}_{f_0} f_1, \mathrm{ad}^2_{f_0} f_1, \ldots, \mathrm{ad}^{n-2}_{f_0} f_1 \ .$$

5.6.3 The Fire Truck with Kingpin Hitching

The procedure for transforming the fire truck with axle-to-axle hitching to chained form, as presented in Section 5.1, applies with no modification when the attachment point between the truck and trailer is not located at the center of the rear axle of the truck, but at some distance off of the axle. The kinematic equations and the h functions needed for transforming this system to chained form are found as follows.

Refer to Figure 5.20 for the system states and parameters and define $x := (x_1, y_1, \phi_1, \theta_1, \phi_2, \theta_2)$. The three covectors representing the non-slipping constraints have the form

$$\begin{aligned}
\omega^1(x) &= [\ \sin(\theta_1+\phi_1) \quad -\cos(\theta_1+\phi_1) \quad 0 \quad -L_0\cos\phi_1 \quad 0 \quad 0 \quad] \\
\omega^2(x) &= [\ \sin\theta_1 \quad -\cos\theta_1 \quad 0 \quad 0 \quad 0 \quad 0 \quad] \\
\omega^3(x) &= [\ \sin(\theta_2+\phi_2) \quad -\cos(\theta_2+\phi_2) \quad 0 \quad L_2\cos(\phi_2+\theta_2-\theta_1) \quad 0 \quad L_1\cos\phi_2 \quad] \ .
\end{aligned}$$

Following the method used for the fire truck in Section 5.1, a control system can be constructed with vector fields g_0, g_1, g_2 that are orthogonal to $\Omega(x) = \mathrm{span}\{\omega^1(x), \omega^2(x), \omega^3(x)\}$. The vector fields are chosen such that the inputs have the interpretation that u_0 is the driving velocity of the rear wheels of the truck, u_1 is the steering velocity of the front axle, and

u_2 is the steering velocity of the tiller. The control system is then written as

$$\dot{x} = g_0(x)u_0 + g_1(x)u_1 + g_2(x)u_2 \ , \tag{5.26}$$

with $g_1 = (0,0,1,0,0,0)'$, $g_2 = (0,0,0,0,1,0)'$ and

$$g_0 = \begin{pmatrix} \cos\theta_1 \\ \sin\theta_1 \\ 0 \\ \frac{1}{L_0}\tan\phi_1 \\ 0 \\ \frac{-1}{L_0 L_1}\sec\phi_2(L_0\sin(\phi_2 - \theta_1 + \theta_2) + L_2\cos(\phi_2 - \theta_1 + \theta_2)\tan\phi_1) \end{pmatrix} .$$

It is straightforward to check that equation (5.26) reduces to the original fire truck system in equation (5.3) when $L_2 = 0$.

The chained form coordinates for this system are found in an analogous manner to those for the original fire truck system (see Section 5.1) using

$$\begin{array}{lll} f_0 = \sec\theta_1 g_0 & f_1 = g_1 & f_2 = g_2 \\ h_0 = x_1 & h_1 = y_1 & h_2 = \theta_2 \ . \end{array}$$

Note that these are the same h functions used for the fire truck system with axle-to-axle hitching. The coordinate transformation to chained form is given by

$$\begin{aligned} \xi_0 = \quad h_0 \quad &= x_1 \\ \zeta_0 = \quad L_{g_0}^2 h_1 \quad &= \frac{\sec^3\theta_1\tan\phi_1}{L_0} \\ \zeta_1 = \quad L_{g_0} h_1 \quad &= \tan\theta_1 \\ \zeta_2 = \quad h_1 \quad &= y_1 \\ \eta_0 = \quad L_{g_0} h_2 \quad &= \frac{1}{L_0 L_1}\sec\theta_1\sec\phi_2(L_0\sin(\theta_1 - \phi_2 - \theta_2) - L_2\cos(\theta_1 - \phi_2 - \theta_2)\tan\phi_1) \\ \eta_1 = \quad h_2 \quad &= \theta_2 \ . \end{aligned} \tag{5.27}$$

This is a valid coordinate transformation since the matrix $\frac{\partial\Phi}{\partial x}$ is nonsingular. The chained form inputs v_0, v_1, v_2 are the derivatives of the coordinates ξ_0, ζ_0, η_0, respectively.

In summary, the main examples used to illustrate the conversion to chained form and extended Goursat normal form were a fire truck and a multiple-steering, multiple-trailer system. We found that for the fire truck system, there was no difference in the two

conversion methods; both approaches were straightforward and easy to use. In chained form, the fire truck was steered using sinusoids at integrally related frequencies in both a step-by-step method and an all-at-once method. Additionally, the fire truck with and without tiller steering was steered using polynomial inputs to show the advantage of having tiller steering. In extended Goursat normal form, the extended fire truck was steered using polynomial inputs. Simulation results showed the effectiveness of the open-loop, point-to-point control algorithms.

For the five-axle system with two steering wheels, however, the method of converting to extended Goursat normal form was easier to use. Even though there is a theorem giving necessary and sufficient conditions for the existence of a coordinate transformation to extended Goursat normal form, it was shown that for some arrangements of the steerable and passive axles, finding this transformation may be difficult. This was resolved using partial prolongations. It is of interest to know necessary and sufficient conditions for using partial prolongations to transform a general multiple-steering, multiple-trailer system into extended Goursat normal form. For larger systems, converting to extended Goursat normal form should be easier than transforming to chained form since the input vector fields would not have to be computed; the kinematic constraints of the system can be used directly.

The mobile robots configured with off-axle hitching that could be transformed into chained form, specifically the fire truck and a car pulling a single trailer, are steerable with the control inputs presented in Chapter 4. The car pulling more than one trailer with off-axle hitching, however, could not be transformed into chained form, but is still controllable.

Chapter 6

Obstacle Avoidance

In this chapter, the problem of avoiding obstacles for a mobile robot configured as a car pulling trailers connected by off-axle, or kingpin, hitches is solved. Given initial and final positions and orientations of a mobile robot in its environment workspace, we would like to generate a path specifying a continuous sequence of positions and orientations that do not collide with the workspace obstacles. Navigating a mobile robot in an environment full of obstacles can be considered more formally as a problem of finding a collision-free path for a point in the robot's configuration space [42]. The *configuration space*, or *C-space*, is the space where the robot is represented as a point and into which the obstacles are mapped. The obstacles are "grown" in the configuration space so that planning a motion of the robot relative to the obstacles in the workspace is equivalent to planning the motion of a point relative to the enlarged obstacles in the configuration space.

The motivating application of this research is automatically controlling a car with many trailers through areas with corridors or lanes such as manufacturing plants or nuclear power plants. The goal is to have fully automated vehicles in areas where it is unsafe for human operators or to assist an operator by steering other axles in the vehicle. The extra steering wheels give greater maneuverability in the narrow, winding passageways.

In the following, a path planning algorithm is developed that will plan a collision-free path for a car pulling trailers. Instead of using axle-to-axle hitching, however, the more general *kingpin* hitching is considered, where the axles are connected by a kingpin (or kingbolt, the point of articulation) between the bodies. Furthermore, the problem is restricted to analyzing the special case of equal length kingpin hitches. The key difference between path planners for a single car and planners for a multiple-trailer vehicle relies on

defining an off-tracking bound for the trailers and kingpin hitches, which is defined as the maximal distance deviated from the lead car's track when the car moves from a straight line to an arc of a circle, or vice versa, in the forward direction. In addition, the trailers are shown to exponentially converge, with respect to the distance traveled by the lead car, to their steady-state circular path when the lead car is moving on a circular path.

If the turning radius of the lead car is upper bounded by the radius of an "enlarged" circular robot (the radius depends on the number of trailers and the off-tracking bounds mentioned above) that is superimposed on the car, then a visibility graph method is proposed to find a collision-free path for the entire vehicle. If the turning radius is not constrained, an alternate algorithm is presented: existing path planners for a single car that use a configuration space approach, in which the environment obstacles are mapped into the configuration space and a path is planned for a point robot in the configuration space, are modified by using the same enlarged circular robot to grow the obstacles. The potential field method and methods unique to car-like robots will be modified.

The outline of the chapter is as follows: in Section 6.1, relevant path planners from the literature are surveyed along with a discussion of the different types of path planning methods. In Section 6.2, operating trucks pulling trailers in reverse is discussed, using studies from the literature. In Section 6.3, upper bounds are calculated on the off-tracking of the trailers and kingpin hitches and the trailers are shown to exponentially converge to their steady state path with respect to the distance traveled by the lead car. In Section 6.4, simulation results are presented for a car pulling one trailer and a multiple-trailer mobile robot to illustrate the off-tracking. In Section 6.5, our obstacle avoidance algorithm is described in detail.

6.1 Literature Survey of Path Planning Methods

For general path planning of robots, the three basic approaches are the roadmap, cell decomposition, and potential field (see Latombe [34] for a more thorough description of all methods). The roadmap approach characterizes the connectivity of the robot's free configuration space as a network, or *roadmap*, of one-dimensional curves lying in the free space or its closure. The initial and goal configurations are connected to points in the resulting roadmap, which is then searched for a path between these points. Various algorithms produce roadmaps called the visibility graph, Voronoi diagram, and silhouette.

The cell decomposition approach is guaranteed to find a path if one exists. This method decomposes the robot's free configuration space into cells in which it is easy to construct a path between any two configurations. The potential field method discretizes the robot's configuration space into a grid of configurations, which is then searched for a free path using a potential field heuristic. In the following, a path planner is called *correct* if it always produces an admissible (collision-free) path and *complete* if it always finds a path if one exists. A distinction is also made between whether the path planning algorithm is an exact or approximate method and, if possible, the time complexity of the algorithm is given.

Classical geometric path planning algorithms that generate trajectories for holonomic robots in constrained spaces, however, may not be feasible for mobile robots with their nonholonomic kinematic constraints. These general algorithms are solutions to the classical piano mover problem, which consists of moving a piano through obstacles without any constraints on the allowable movements. The problem of moving the piano in the 3-dimensional Euclidean space is transformed into the problem of moving a point in the configuration space. The existence of a collision-free trajectory for the piano is characterized by the existence of a connected component in the admissible configuration space. The kinematic constraints of the mobile robot, however, cause the number of degrees of freedom to be less than the dimension of the configuration space. Therefore, an arbitrary path in the admissible configuration space does not necessarily correspond to a feasible trajectory for the mobile robot.

In this section, a survey of the literature on path planners for mobile robots is presented, concentrating on planners that use the visibility graph method, the potential field method, or a method unique to robots with nonholonomic constraints.

In general geometric planning algorithms described by Latombe [34] that use a configuration space formulation, the procedure for planning a path for a car with multiple trailers would be to enclose the whole mobile robot in one circle and use the resulting circular robot to "grow" the obstacles in the robot's configuration space. Moravec [51, 50] was the first to grow obstacles using circles. This procedure essentially represents the environment obstacles in the robot's configuration space. The collision-free path for the resulting "point robot" is then planned in the configuration space. This method produces very conservative paths when the robot is large.

Reeds and Shepp [62] completely characterize the shortest paths between any two configurations for a two-axle vehicle that is allowed to move in the reverse direction in an

environment without obstacles. A set of paths that is *sufficient* (always contains a shortest path) and contains at most 68 paths is given by an explicit formula. They show that any path with greater than two reversals can be reduced to a path with at most two reversals that may be as long or shorter than the original one. Dubins [20], on the other hand, derives shortest paths for a system similar to a car-like robot when no reversals are allowed. The velocity of the system is held constant and the path is made up of straight line segments and arcs of circles. A sufficient set of paths is given in which there are only at most six candidates for each pair of initial and final configurations. The shortest path out of the six is then selected.

In [35], Laumond presents an exact, but uncomplete approach for finding collision-free smooth trajectories for a circular (radius r) mobile robot whose turning radius is lower bounded by r_0. Reversals are not allowed in this planner. The path planning problem is transformed into a finding polygonal lines in a dual space of the configuration space, called the *space of centers of curvature.* The admissible configuration space is obtained by an isotropic growth of the obstacles by the radius r. The existence of smooth trajectories is characterized by the existence of paths made up of line segments, circle arcs of radius r_0, or contact arcs of curvature less than $1/r_0$ at any point. The algorithm consists of searching for the centers of curvature. The search yields the shortest path, although the time complexity of the algorithm cannot be precisely evaluated. The algorithm is more efficient when the space is more constrained since the space of solutions is reduced.

A good recent review of nonholonomic motion planning is given by Li and Canny, eds. [41]. In this collection, Jacobs and Canny [29] present a complete path planner that calculates a smooth, approximate path for a mobile robot with a minimum turning radius and no reversals. This paper extends Dubin's [20] results to collision-free trajectories. First, a finite set of canonical trajectories that satisfy the nonholonomic constraints is defined. Then, orientations and positions of the endpoints of the trajectories are quantized, giving a finite number of possible trajectory segments. A graph search algorithm is then applied that finds a path (if one exists) in $O(\frac{n^3}{\delta}\log n + A\log(\frac{n}{\delta}))$ time, where n is the number of free trajectories and δ characterizes the robustness of the path and the accuracy of the approximation. The initial collision-free subpaths go between points on the obstacle boundaries and are smoothly concatenated into the final path.

Laumond, Jacobs, Taix, and Murray [30, 39] present a complete and exact motion planner for a mobile robot whose turning radius is lower bounded. Reeds and Shepp curves

are used as the basis for their algorithm. A geometric path planning algorithm is used to compute a collision-free, minimum length path, ignoring the kinematic and curvature constraints of the robot and allowing for reversals. The path is then transformed into one that obeys the nonholonomic and curvature constraints by using subgoals along the initial path. The basic idea is to try to reach the goal configuration from the current configuration by using the shortest path. If this shortest path intersects an obstacle, subgoals are generated from the initial holonomic collision-free path. A finite number of recursive subdivisions may be used before finding a collision-free trajectory. The subpaths are concatenated to make the final path, which is then optimized for near-minimal length.

Mirtich and Canny [48] present a path planner for a two-axle car that is also based on following a nonfeasible (does not consider the nonholonomic constraints) path as in [30]. The planner uses a novel approach that builds a one-dimensional maximal clearance skeleton, or roadmap, through the robot's configuration space using a shortest feasible path metric (rather than the usual Euclidean metric) that captures the nonholonomy of the mobile robot. Once the skeleton between the start and goal configurations has been constructed, it is covered with shortest feasible path balls that lie completely in the free configuration space. Then a series of "jumps" can be made between a finite number of points on the skeleton that connect the start to the goal. Canny's general roadmap algorithm [16] is then applied to find a feasible path with near minimum number of reversals. This approach is approximate and the complexity of the path is an increasing function of the arc length of the path and the number of reversals in the path.

Laumond [38] found that planning a path for a mobile robot through highly constrained spaces may lead to a trajectory with many back-up maneuvers. Wilfong [86] and Tournassoud and Jehl [79] both present heuristic path planning algorithms for a mobile robot with limited steering range when the environment consists of lanes or corridors, such as in a flexible automated factory. Wilfong presents a correct algorithm that computes the motion of an autonomous vehicle between two given configurations with a minimum number of turns in $O(m^2)$ time, where m is the number of lanes in the environment. The algorithm requires $O(m^2(n^2+\log m))$ pre-processing time, where n is the number of corners on the polygonal obstacles. The minimum free radius for each turn in the designated lanes is computed, then the motion of the vehicle is obtained by constructing and searching a directed graph for a path. This algorithm does not allow reversals and does not necessarily find the shortest path. Heuristics are added to guarantee that the mobile robot does not

"hug" the obstacles. Tournassoud and Jehl, on the other hand, present a heuristic path planning algorithm that divides the robot's free configuration space into pairs of adjacent *cones*, where each pair defines a possible turn for the robot. A Voronoi diagram is constructed, which yields the free paths that maximize the clearance between the mobile robot and the obstacles. For polygonal obstacles, the Voronoi diagram consists of straight line segments and parabolic segments. The planning is performed using a cost function that combines the length of a path and a penalty when reversals are necessary.

In the potential field method, the robot's configuration space is discretized into a rectangular grid of configurations. The point robot in the configuration space acts under a certain potential field that is constructed using the obstacles and the goal configuration. The negated gradient of the total potential field is treated as an artificial force applied to the robot, causing the robot to be drawn to the desired goal position and repelled from the obstacles. The major disadvantage to this method is that since it is a fastest descent optimization method, it may get caught in local minima of the potential field. If the total potential field has local minima, the minima are connected by a graph that is searched until the goal is attained. These local minima may arise, for example, when the robot gets trapped in simple concavities formed by the obstacles, or when conflict occurs when various control points are concurrently attracted to their respective goal position.

Barraquand and Latombe [9, 6, 7] present a potential field method for path planning for mobile robots with reversals and for manipulator arms. Both the workspace and the configuration space of the robot are discretized using a hierarchical bitmap representation. The obstacles are modeled as distributed bitmap descriptions, rather than the usual semi-algebraic descriptions. Their approach uses potential fields attached to specific points on the boundary of the robot. A potential field is computed in $O(a + b \log b)$ time, where a is the number of points in the bitmap array and b is the number of points in an "augmented skeleton," which is a generalized Voronoi diagram of the robot's free workspace that includes a link to the goal configuration. All of the potential fields are summed into one potential field defined in the configuration space, which is then searched heuristically for a collision-free path. For searching the graph of local minima of the potential field, they present a brute force method, which uses a "best-first" algorithm to search for the local minima in the discretized configuration space and works well for robots with degrees of freedom less than or equal to four, and a Monte-Carlo procedure, which is better for systems with large degrees of freedom. The brute-force method is complete at the maximal

resolution of the bitmap, while the Monte-Carlo procedure is not complete. For mobile robots, this algorithm is designed to minimize the number of back-up maneuvers.

Laumond, Sekhavat, and Vaisset [40] apply the potential field method of Barraquand and Latombe above to mobile robots with trailers with reversals. A collision-free path based on the discretization of the configuration space (using potential fields as in [9]) is first planned that does not consider the nonholonomic constraints of the vehicle. In the absence of obstacles, this path is then approximated by a sequence of feasible, near-optimal paths (using a scheme based on the method in [39]) that are computed numerically. The last step of the algorithm smooths out the first path to yield a solution. When the degree of nonholonomy of the system is d, the complexity of the approximation part of the algorithm is $O(\epsilon^d)$, where ϵ is the smallest distance from the initial geometric path to an obstacle.

The visibility graph is a non-directed graph whose nodes are the initial and final configuration of the point robot and all of the vertices of the obstacles in the workspace. The nodes that "see each other" are connected by straight line segments that do not intersect the interiors of the obstacles. The resulting graph is searched by an optimization method for the shortest (with respect to the Euclidean metric) semi-free (may touch boundary of the obstacles) path between the start and goal configurations. If there exists a path between the start and goal configurations, then there exists a shortest path and this path is on the visibility graph.

Lozano-Pérez and Wesley [43] present a collision avoidance algorithm that computes the shortest path for a polyhedral (polygons on the plane) robot moving among known convex polyhedral obstacles (any obstacle can be modeled in this way to any desired accuracy). The visibility graph approach for a point robot is generalized to be used for a non-point robot by growing the obstacles by the robot. This is done, for example, for a circular robot by displacing the obstacle vertices away from the obstacles at least by the radius of the robot. The circular robot then moves so that the center point goes through the displaced vertices. For general robots, a method is presented that takes into account the orientation when growing the obstacles.

Rohnert [63] constructs a *reduced* visibility graph, which contains only that part of the visibility graph that is relevant for finding the shortest path between two points. The problem considered has n total vertices on f disjoint convex polygons in the Euclidean plane. It is shown that the shortest path uses only the edges of the polygons and the *supporting segments* of pairs of polygons that do not intersect other polygons. The supporting segments

for two polygons are the common tangent links between two polygons. The reduced visibility graph is computed in $O(n+f^2 \log n)$ pre-processing time and the shortest path between two points is found in $O(f^2+n \log n)$ time and $O(n+f^2)$ space using Dijkstra's algorithm [19], which is a standard single source shortest path algorithm.

Vegter [80] presents a dynamic algorithm that maintains a *visibility diagram* (a generalization of the visibility graph) when the set of N line segments (the obstacles) in the plane is allowed to change when a line segment is deleted or added to the set. The main result is that the visibility diagram can be maintained in $O(\log^2 N + K \log N)$ time, where K is the total number of arcs of the visibility graph that are created or distroyed when a line segment is added or deleted. The algorithm can be used to plan a feasible path of a rod moving through the N line segment obstacles in $O(N^2)$ time.

Jiang *et al.* [32] find the minimum time smooth path, as opposed to the usual minimum distance paths, for a mobile robot without generating the robot's configuration space. A reduced visibility graph is constructed, which is then mapped to a feasible reduced visibility graph using heuristics to include the robot's kinematics. Finally, a polynomial-time algorithm is used to search the graph for a safe, minimum time smooth path.

6.2 Operating Tractor-trailer Systems in Reverse

In a California Department of Public Works triple-trailer study [60], an actual truck tractor pulling three trailers was used to test the backing maneuver. The vehicle was configured as having two axles on the truck, one axle on the first trailer, and two axles on both the second and third trailers. The first trailer was connected to the truck tractor with a fifth wheel type hitch, while the second and third trailers were connected by a pintle hook on the trailer in front over which is placed the eye of a tow bar of the trailer in back. This can be treated in our terms as axle-to-axle hitching between the truck and the first trailer and off-axle (kingpin) hitching between the other trailers. The total vehicle length was 94 feet 4 inches with each of the trailers being 26 feet 6 inches. For the first test, the triple trailer started off in a straight line, then backed up until it had folded to the point of "impending damage" to the vehicle. It was found that the vehicle could back up 47 to 65 feet before jack-knifing. For the second test, the triple trailer was run through 180 degrees of a 60-foot radius curve, then backed up. It was found that it could only back up 30 feet before approaching damage. The report concluded that one should never back up one of

these vehicles unless to clear a stalled vehicle in a lane of traffic.

An instructor's manual for training tractor-trailer drivers [2] states that the process of backing up tractor-trailers is a combination of "jacking" and "chasing." The tractor is first turned until a jack-knife angle occurs between the tractor and trailer. Once the trailer is heading in the correct direction, the tractor is then turned so as to follow the trailer. The idea is to use the rear tractor axle as the trailer's steering axle, so that one steers in the opposite direction of the desired trailer path. This procedure is used to train drivers to back up tractor-trailers along a straight line, into an alley dock, and during parallel parking.

For the system with axle-to-axle hitching and any number of trailers, there is no difficulty operating in reverse, since the system can be transformed into chained form as shown in [68, 76]. Once in chained form, any one of the steering methods presented in Chapter 4 can be used to steer the vehicle in the reverse direction. For kingpin hitching, as was shown in Section 5.6, the system with one trailer is transformable to chained form, and hence steerable in the reverse direction. For two or more trailers, however, the system is not transformable into chained form. The system is still controllable in reverse, but the steering methods for chained form systems cannot help us steer these systems.

As an experiment, the system of a single-axle car pulling two trailers connected with equal length kingpin hitches was steered in the reverse direction using the interactive steering program that will be described in Section 6.4. Using jacking and chasing maneuvers, it was possible to keep the first trailer moving in front of the car, but the second trailer reached the jack-knife configuration rather quickly.

6.3 Off-tracking Bounds

In this section, fundamental calculations needed for the proposed obstacle avoidance algorithm for a car pulling trailers connected with off-axle hitches are computed. The calculations include computing an upper bound on the off-tracking of the trailers and kingpin hitches and proving that the trailers exponentially converge to their steady state path with respect to the distance traveled by the lead car.

In [46, 60], *off-tracking* is defined as the deviation to the right or left of the trailers' axles from the path of the front steering axle during a turn at slow speed. In [47], a truck pulling one or more trailers is said to be *stable* if the path of each trailer does not swing or deviate more than three inches to either side of the truck when the vehicle moves in a

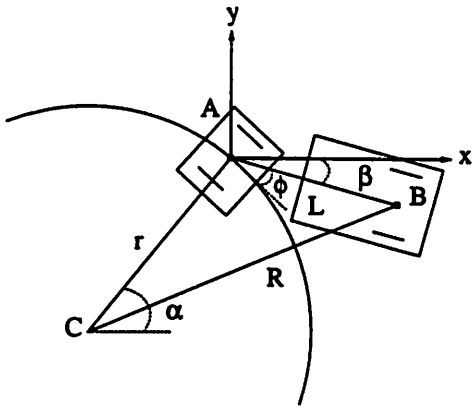


Figure 6.1: The car is traveling along a circle of radius r. For axle-to-axle hitching, the trailer is shown to exponentially converge, with respect to the arc length traveled by the lead car, to a circle with the same center.

straight line on a level smooth paved surface.

Below the idea of off-tracking is quantified for the case of a car pulling one trailer (a two-axle system). The results can be easily extended for a car with more than one trailer by making all of the subsequent calculations for the last trailer in the convoy. In this section, the trailer is shown to exponentially converge to a circular path when the car is traveling on a circular path (for straight line, the radius is infinity). In addition, under the condition of equal length kingpin hitches, an upper bound is computed[1] on the amount the trailer and kingpin hitch deviate from the car's path for the two special cases: (1) the transition from a straight line to an arc of a circle of radius r, and (2) the transition from an arc of a circle of radius r to a straight line. The goal is to find a single upper bound that bounds both the off-tracking of the trailer and the off-tracking of the kingpin hitch for the entire trajectory when the car is moving in the forward direction. The reverse direction is left as an open problem (see Chapter 7). An upper bound for the first case is computed using the off-tracking of the kingpin hitch, while an upper bound for the second case is computed using the off-tracking of the trailer. The bounds are computed with respect to the distance traveled by the lead car. In the following, "path of the car" and "path of the trailer" refer to the trajectory of the center of the respective axle.

Consider Figure 6.1, where the lead car in a two-axle system is traveling counter-clockwise around a circular path of radius r. We will show that the trailer exponentially

[1]Thanks to Brian Mirtich for helping to formulate the proofs presented in this section.

converges to a steady-state circular path with the same center (when the vehicle travels clockwise, the derivation is similar). Before attempting this, coordinates must be assigned to the system. The center of the car's axle, A, is assigned to be the origin of an x-y moving frame. A fixed reference frame is attached to the center, C, of the circle. The car has traveled an angle α with respect to the fixed x-axis. The trailer, which is connected to the car with an axle-to-axle hitch of length L, is at an angle ϕ from the tangent line to the circle at point A and at an angle β from the moving x-axis. Let (x_1, y_1) be the coordinates of the car (point A) and (x_2, y_2) be the coordinates of the trailer (point B).

Lemma 27 *The velocity of the lead car, v_1, is related to the velocity of the trailer, v_2, as*

$$v_2 = v_1 \cos\phi \ . \tag{6.1}$$

Proof. Using the fixed frame, differentiating $(x_2 - x_1)^2 + (y_2 - y_1)^2 = L^2$ with respect to time gives

$$2(x_2 - x_1)(\dot{x}_2 - \dot{x}_1) + 2(y_2 - y_1)(\dot{y}_2 - \dot{y}_1) = 0 \ . \tag{6.2}$$

Defining the angle

$$\beta := \pi + \alpha + \pi/2 + \phi = \alpha + \phi - \pi/2 \tag{6.3}$$

gives $x_2 - x_1 = L\cos\beta$ and $y_2 - y_1 = L\sin\beta$. By calculation,

$$\begin{aligned} \dot{x}_2 - \dot{x}_1 &= v_2\cos(\beta+\pi) - v_1\cos(\alpha+\pi/2) = -v_2\cos\beta + v_1\sin\alpha \\ \dot{y}_2 - \dot{y}_1 &= v_2\sin(\beta+\pi) - v_1\sin(\alpha+\pi/2) = -v_2\sin\beta - v_1\cos\alpha \ . \end{aligned}$$

Substituting this into equation (6.2) gives

$$\begin{aligned} 0 &= L\cos\beta(-v_2\cos\beta + v_1\sin\alpha) + L\sin\beta(-v_2\sin\beta - v_1\cos\alpha) \\ &= -v_2 + v_1\sin(\alpha-\beta) = -v_2 + v_1\sin(\pi/2-\phi) = -v_2 + v_1\cos\phi \ . \end{aligned}$$

□

Lemma 28 *The angle β changes with respect to time as*

$$\frac{d\beta}{dt} = -\frac{v_1}{L}\sin\phi \ . \tag{6.4}$$

Proof. The angle the trailer makes with the moving frame x-axis can be defined as

$$\beta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) .$$

Using the $\frac{d}{dt}\arctan u = \dot{u}/(1+u^2)$, the derivative of β with respect to time is

$$\begin{aligned}
\frac{d\beta}{dt} &= \frac{1}{1+\left(\frac{y_2-y_1}{x_2-x_1}\right)^2}\left(\frac{(\dot{y}_2-\dot{y}_1)(x_2-x_1)-(\dot{x}_2-\dot{x}_1)(y_2-y_1)}{(x_2-x_1)^2}\right) \\
&= \frac{1}{L^2}((\dot{y}_2-\dot{y}_1)(x_2-x_1)-(\dot{x}_2-\dot{x}_1)(y_2-y_1)) \\
&= \frac{1}{L^2}((-v_2\sin\beta - v_1\cos\alpha)L\cos\beta - (-v_2\cos\beta + v_1\sin\alpha)L\sin\beta) \\
&= -\frac{v_1}{L}\cos(\alpha-\beta) = -\frac{v_1}{L}\cos(\pi/2-\phi) = -\frac{v_1}{L}\sin\phi .
\end{aligned}$$

□

Lemma 29 *The angle ϕ changes with respect to α as*

$$\frac{d\phi}{d\alpha} = -\left(1+\frac{r}{L}\sin\phi\right) . \tag{6.5}$$

Proof. From equation (6.3) and Lemma 28,

$$\frac{d\beta}{dt} = \frac{d\alpha}{dt} + \frac{d\phi}{dt} = -\frac{v_1}{L}\sin\phi . \tag{6.6}$$

The angle traveled by the car around the circle is defined as

$$\alpha(t) = \alpha_0 + \frac{1}{r}\int_0^t v_1(\tau)d\tau , \tag{6.7}$$

where the initial angle, α_0, is assumed to be zero without loss of generality. Substituting the derivative of α into equation (6.6),

$$\frac{d\phi}{dt} = -\frac{v_1}{r} - \frac{v_1}{L}\sin\phi = \frac{d\phi}{d\alpha}\frac{d\alpha}{dt} = \frac{d\phi}{d\alpha}\frac{v_1}{r}$$

gives equation (6.5). □

From this lemma, when $\phi = 0$, $\frac{d\phi}{d\alpha} = -1$, i.e., when the trailer is aligned with the car, it is still moving in towards the center of the circle.

Theorem 30 *The angle ϕ locally exponentially converges to the steady state value*

$$\phi_s = \lim_{\alpha\to\infty}\phi(\alpha) = \arcsin\left(\frac{-L}{r}\right) \pm 2\pi n \quad n \geq 0 . \tag{6.8}$$

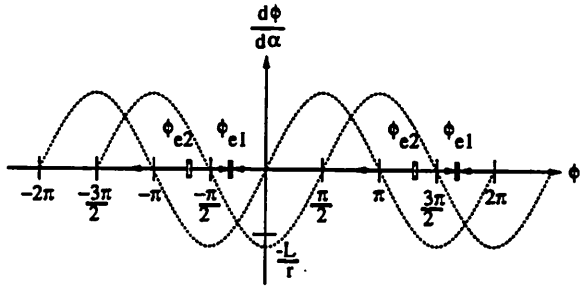


Figure 6.2: The phase portrait of ϕ. The two equilibrium points are ϕ_{e1} (stable) and ϕ_{e2} (unstable). The arrows indicate the direction ϕ moves when near an equilibrium point.

Proof. The equilibrium points of the periodic, nonlinear equation

$$\frac{d\phi}{d\alpha} = f(\phi) = -\left(1 + \frac{r}{L}\sin\phi\right) \tag{6.9}$$

are

$$\begin{aligned}
\phi_{e1} &= \arcsin\left(\frac{-L}{r}\right) \pm 2\pi n \quad n \geq 0 \\
\phi_{e2} &= \pi - \arcsin\left(\frac{-L}{r}\right) \pm 2\pi n \quad n \geq 0 \quad = -\arcsin\left(\frac{-L}{r}\right) \pm \pi m \quad m \geq 1 \text{ odd}\,.
\end{aligned}$$

To determine the stability of the system (6.9), the Liapunov linearization method [81] is used. Consider the linearization of equation (6.9) about the equilibrium point $\phi_e \in \{\phi_{e1}, \phi_{e2}\}$:

$$\frac{d\phi}{d\alpha} = \left.\frac{\partial f(\phi)}{\partial \phi}\right|_{\phi=\phi_e} \phi\,.$$

This gives the autonomous linear equation

$$\frac{dz}{d\alpha} = -\frac{r}{L}\cos\phi_e\, z\,, \tag{6.10}$$

where $z := \phi - \phi_e$ is the linearized variable. The linearized system has the solution

$$z(\alpha) = z(0)\exp\left(-\frac{r}{L}\cos\phi_e\,\alpha\right)\,.$$

When $\phi_e = \phi_{e1}$, $\cos\phi_{e1} \geq 0$ and $\frac{dz}{d\alpha} \leq 0$. Thus, z converges exponentially to zero, or ϕ converges to ϕ_{e1} exponentially. Therefore, the equilibrium point ϕ_{e1} is locally exponentially stable. When $\phi_e = \phi_{e2}$, $\cos\phi_{e2} \leq 0$ and $\frac{dz}{d\alpha} \geq 0$. Thus, z diverges exponentially away from zero, or ϕ diverges away from ϕ_{e2} exponentially. Therefore, the equilibrium point ϕ_{e2} is locally unstable. This is illustrated in the phase portrait in Figure 6.2.

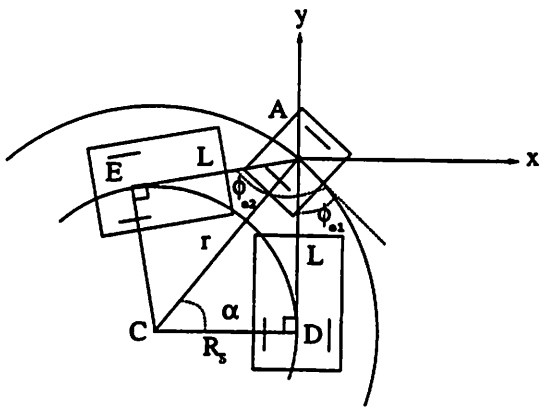


Figure 6.3: The trailer exponentially converges to a circle of radius R_s passing through point D. The unstable equilibrium occurs when the trailer is at point E.

Since the linearized system (6.10) is locally exponentially stable in a neighborhood of the stable equilibrium point $\phi_s = \phi_{e1}$, the nonlinear system (6.9) is also locally exponentially stable in a neighborhood of ϕ_{e1}. □

Figure 6.3 shows the stable and unstable equilibrium positions for this vehicle. The stable position has the trailer at point D with ADC a right triangle. If ϕ is restricted as $|\phi| \leq \pi/2$, i.e., where the trailer avoids the jack-knife positions, then ϕ never reaches the unstable equilibrium. Under this assumption, $r > L$. The case with $r = L$ is unrealistic since this corresponds to $R = 0$ and $\phi = -\pi/2$, i.e., the trailer is sitting at the center of the circle (see Section 6.4, case (c)).

Lemma 31 *The distance of the trailer (point B) from the center of the circle (point C) changes with respect to α as*

$$\frac{dR}{d\alpha} = -\frac{rL}{R}\cos\phi\left(1 + \frac{r}{L}\sin\phi\right) . \tag{6.11}$$

Proof. Referring to Figure 6.1, the law of cosines on the triangle ABC gives

$$\begin{aligned} R^2 &= r^2 + L^2 - 2rL\cos(\pi/2 + \phi) \\ R &= (r^2 + L^2 + 2rL\sin\phi)^{1/2} . \end{aligned} \tag{6.12}$$

Taking the derivative with respect to α gives

$$\frac{dR}{d\alpha} = \frac{dR}{d\phi}\frac{d\phi}{d\alpha} = \frac{1}{2R}(2rL\cos\phi)\frac{d\phi}{d\alpha} .$$

The result follows from substituting in equation (6.5) for $\frac{d\phi}{d\alpha}$. □

Theorem 32 *For $r > L$, R locally exponentially converges to the steady state value*

$$R_s = \lim_{\alpha \to \infty} R(\alpha) = \sqrt{r^2 - L^2} \ . \tag{6.13}$$

Proof. From equation (6.12),

$$\phi = \arcsin\left(\frac{R^2 - r^2 - L^2}{2rL}\right) \pm 2\pi n \quad n \geq 0 \ ,$$

which can be used to write the left hand side of equation (6.11) as a function of R:

$$\begin{aligned} \frac{dR}{d\alpha} &= f(R) = -\frac{rL}{R}\sqrt{1 - \left(\frac{R^2 - r^2 - L^2}{2rL}\right)^2}\left(1 + \frac{R^2 - r^2 - L^2}{2L^2}\right) \\ &= -\frac{R^2 - r^2 + L^2}{4RL^2}\sqrt{(2rL)^2 - (R^2 - r^2 - L^2)^2} \ . \end{aligned} \tag{6.14}$$

The equilibrium points of the nonlinear equation (6.14) are

$$R_{e1} = \sqrt{r^2 - L^2} \qquad R_{e2} = r \pm L \ .$$

R_{e1} corresponds to the angle $\phi = \phi_s$ of equation (6.8) and R_{e2} corresponds to the angle $\phi = \pm m\pi/2$ for m odd, i.e, the trailer is in the jack-knife position.

To determine the stability of the system (6.14) around these equilibrium points, the Liapunov linearization method [81] is again used. Consider the linearization of equation (6.14) about the equilibrium point $R_e \in \{R_{e1}, R_{e2}\}$,

$$\frac{dR}{d\alpha} = \left.\frac{\partial f(R)}{\partial R}\right|_{R=R_e} R \ ,$$

which gives the autonomous linear equation

$$\begin{aligned} \frac{dz}{d\alpha} &= -\frac{1}{2L^2}\left(\sqrt{(2rL)^2 - (R_e^2 - r^2 - L^2)^2} - \frac{(R_e^2 - r^2 + L^2)(R_e^2 - r^2 - L^2)}{\sqrt{(2rL)^2 - (R_e^2 - r^2 - L^2)^2}}\right. \\ &\quad - \left.\frac{(R_e^2 - r^2 + L^2)}{2R_e^2}\sqrt{(2rL)^2 - (R_e^2 - r^2 - L^2)^2}\right) z \qquad = -A\, z \end{aligned}$$

where $z := R - R_e$ is the linearized variable. When $R_e = R_{e1}$,

$$\frac{dz}{d\alpha} = -\frac{1}{L}\sqrt{r^2 - L^2}\ z \ ,$$

which has the solution

$$z(\alpha) = z(0)\exp\left(-\frac{\sqrt{r^2 - L^2}}{L}\ \alpha\right) \quad r > L \ . \tag{6.15}$$

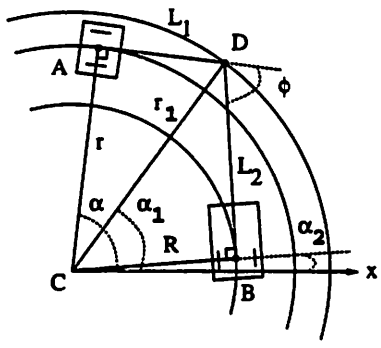


Figure 6.4: For kingpin hitching, the lead car travels along a circle of radius r and the kingpin travels along a circle of radius r_1. The trailer is shown to converge to a circle of radius R with $R^2 = r^2 + L_1^2 - L_2^2$.

Therefore, z converges exponentially to zero, or R converges to R_{e1} exponentially. Therefore, the equilibrium point $R_s = R_{e1}$ is locally exponentially stable.

When $R_e = R_{e2}$, the function A is not well defined, since the square root term in $\frac{dz}{d\alpha}$ is zero at R_{e2}. For our purposes, this case can be ignored. Refer back to Theorem 30, which shows the stable and unstable modes of the physical system. □

By the above results, the following lemmas that consider the case of a two-axle system with kingpin hitching can be stated.

Lemma 33 *If the lead car of a two-axle system with kingpin hitching of lengths L_1 and L_2 travels along a path of radius r, then the trailer will converge to a circle of radius*

$$R = \sqrt{r^2 + L_1^2 - L_2^2}\ , \tag{6.16}$$

provided $r^2 + L_1^2 \geq L_2^2$.

Proof. The proof refers to Figure 6.4. The system of the kingpin hitch and the trailer is equivalent to the car and trailer system treated above. Therefore, setting $r = r_1$ and $L = L_2$ in equation (6.13) gives $R = \sqrt{r_1^2 - L_2^2}$. Since triangle CAD is a right triangle, $r_1 = \sqrt{r^2 + L_1^2}$. □

Lemma 34 *If the lead car of a two-axle system with a kingpin hitch travels a distance s on a circle of radius r, the kingpin travels a distance $s_1 = s$ and the trailer travels a distance*

$$s_2 \ = \ \int_0^t v_1(\tau)\cos\phi(\tau)d\tau\ , \tag{6.17}$$

where ϕ is the relative angle between the trailer and the car.

Proof. The proof refers to Figure 6.4. Let C be the origin of the x-y fixed frame. The lead car (point A) travels an arc length s counterclockwise around a circle of radius r through an angle $\alpha = s/r$ at a velocity v_1. The kingpin hitch (point D) travels a distance s_1 on a circle of radius r_1 through an angle $\alpha_1 = s_1/r_1$ with velocity v_D. The trailer (point B) travels a distance s_2 on a circle of radius R through an angle $\alpha_2 = s_2/R$ at a velocity v_2.

Without loss of generality, assume α is initially zero. Then, by equation (6.7),

$$s = \alpha r = \int_0^t v_1(\tau)d\tau \ .$$

By Lemma 27, $v_D = v_1$. Therefore, the kingpin hitch travels a distance

$$s_1 \ = \ \alpha_1 r_1 = \int_0^t v_D(\tau)d\tau = \int_0^t v_1(\tau)d\tau = s$$

from its initial point. By Lemma 27, $v_2 = v_1 \cos\phi$. Therefore, the trailer travels a distance

$$s_2 \ = \ \alpha_2 R = \int_0^t v_2(\tau)d\tau = \int_0^t v_1(\tau)\cos\phi(\tau)d\tau$$

from its initial point. □

Lemma 35 *If the lead car of a two-axle system with equal length kingpin hitches travels along a path of radius r, then the trailer will exponentially converge to the same circle.*

Proof. The result follows directly from setting $L_1 = L_2$ in equation (6.16). □

The next two theorems assume equal length kingpin hitches, i.e., $L := L_1 = L_2$. The goal is to find a single upper bound that bounds both the off-tracking of the trailer and the off-tracking of the kingpin hitch for the entire trajectory. The first theorem computes an upper bound using the off-tracking of the kingpin hitch when the car changes from a straight line to an arc of a circle of radius r. The second theorem computes an upper bound using the off-tracking of the trailer when the car changes from an arc of a circle of radius r to a straight line. The bounds are computed with respect to the distance traveled by the lead car.

Theorem 36 *If the lead car of a two-axle system with equal length kingpin hitching changes from a straight line to an arc of a circle of radius r, then an upper bound on the off-tracking of the trailer, $\tilde{z}$, and the off-tracking of the kingpin hitch, z, is*

$$\tilde{z}(\alpha) \leq z(\alpha) \leq z_1 := r\left(\frac{\sqrt{\lambda^2+1}}{\lambda} - 1\right) \tag{6.18}$$

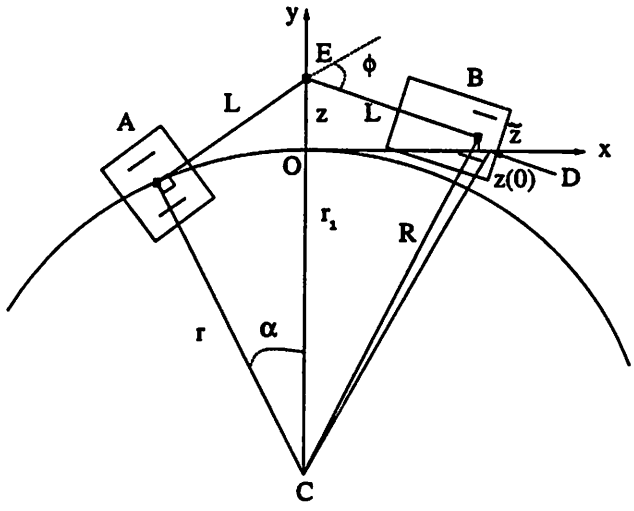


Figure 6.5: An upper bound on the off-tracking of the trailer and kingpin hitch is computed when the lead car changes its path from a straight line DO to an arc of a circle of radius r.

where α is the arc length traveled by the lead car from the instant the car switches to the circle and $\lambda := r/L$.

Proof. The proof refers to Figure 6.5. The car travels from the right to the origin O along a straight line of length $2L$, then at O switches to the arc of a circle of radius r. The kingpin hitch is at a distance L from the origin (point D) initially, and at point E when the car switches to the circle. The trailer is at a distance $2L$ from the origin initially, and at point B at the switching time.

From Lemma 33, $R \leq r_1$ and $r_1 \geq r$, therefore $\tilde{z} \leq z$, i.e., an upper bound needs to be computed on the off-tracking of the kingpin hitch.

Let γ be that angle where z is maximum. This occurs when the kingpin hitch is at the y-axis (as drawn in the figure). Thus, when $\alpha = \gamma$, the length of CE is $\sqrt{r^2 + L^2} = \frac{r}{\lambda}\sqrt{\lambda^2 + 1}$ and $\gamma = \arctan(1/\lambda)$. For $0 \leq \alpha \leq \gamma$, z increases from zero to its maximum value, giving a bound z_1 as in equation (6.18).

For $\alpha \geq \gamma$, equation (6.15) gives the bound

$$z(\alpha) \;=\; z(0)\exp\left(-\alpha\sqrt{\lambda^2-1}\right) \leq z_2 := z(0)\exp\left(-\sqrt{\lambda^2-1}\tan^{-1}\frac{1}{\lambda}\right) \qquad (6.19)$$

with the initial off-tracking of the kingpin hitch

$$z(0) = \sqrt{r^2 + L^2} - r = r\left(\frac{\sqrt{\lambda^2+1}}{\lambda} - 1\right).$$

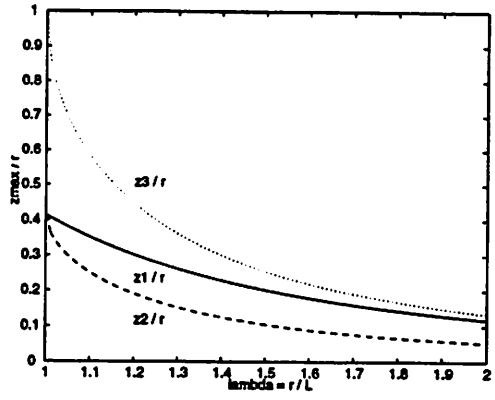


Figure 6.6: Comparing the three off-tracking bounds as a function of $\lambda = r/L$ when the car switches from a straight line to an arc of a circle. The solid line is z_1/r, the dashed line is z_2/r, and the dotted line is z_3/r.

Figure 6.6 shows that the bound z_1 from equation (6.18) is always greater than the bound z_2 from (6.19). Therefore, the first bound is used as the maximum distance the kingpin hitch and the trailer will swing off the car's path for this trajectory. □

Theorem 37 *If the lead car of a two-axle system with equal length kingpin hitching changes from an arc of a circle of radius r to a straight line, then an upper bound on the off-tracking of the trailer, z, is*

$$z(\alpha) \leq z_3 := r\left(1 - \frac{\sqrt{\lambda^2 - 1}}{\lambda}\right) \tag{6.20}$$

where α is the arc length traveled by the lead car from the instant the car switches to the circle and $\lambda := r/L$.

Proof. The proof refers to Figure 6.7. The car travels counterclockwise around the circle, then switches to the straight line at the origin O. The trailer is at point D at the switching time, and at point B when the car has moved a distance L. The kingpin hitch is at point E initially and follows the straight line path of the car.

From Lemma 33, $R \leq r_1$. Therefore, the trailer will swing into the circle during this maneuver. In contrast to the previous theorem, the maximum bound is calculated on the off-tracking of the trailer (the kingpin off-tracking is actually zero for this case).

Let d be that distance traveled by the car from the origin where z is maximum. This occurs when the kingpin hitch is at the origin (as drawn in the figure). Thus, when

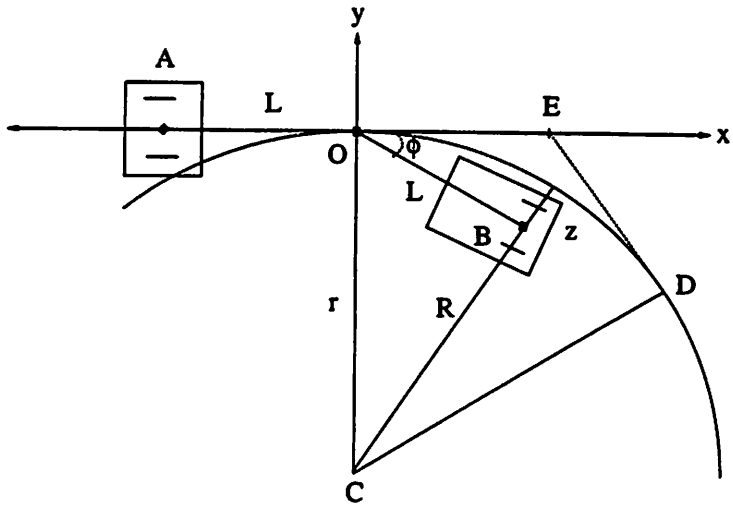


Figure 6.7: An upper bound on the off-tracking of the trailer is computed when the lead car changes its path from an arc of a circle of radius r to a straight line.

$d = L$, $R = \sqrt{r^2 - L^2}$ and an upper bound on z is given by

$$r - R = r - \sqrt{r^2 - L^2} = r - r\,\frac{\sqrt{\lambda^2 - 1}}{\lambda},$$

which is the bound in equation (6.20).

□

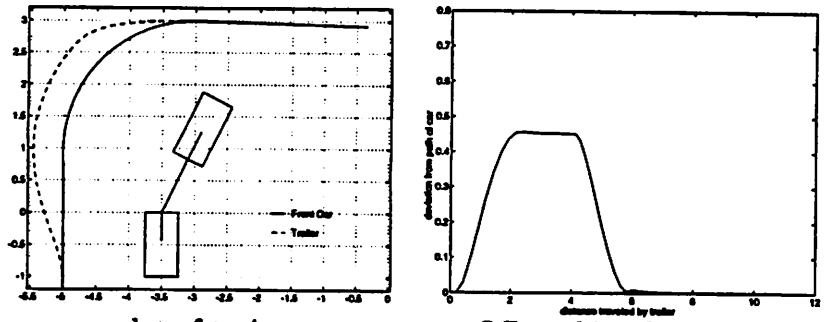
Figure 6.6 shows that bound z_3 from equation (6.20) is greater than bound z_1 from equation (6.18). Therefore, in a corner trajectory, the trailer and kingpin hitch initially swing out of the circle, the trailer converges to the circle, then the trailer will swing into the circle. The results show that the swing-out is less than the swing-in.

6.4 Simulation of Off-tracking

Using the interactive steering software package that we[2] developed, we were able to acquire experimental data supporting the theorems for the off-tracking bounds calculated in the previous section. In this section, the off-tracking bounds and the convergence rates are simulated for a car pulling one trailer and a car pulling three trailers.

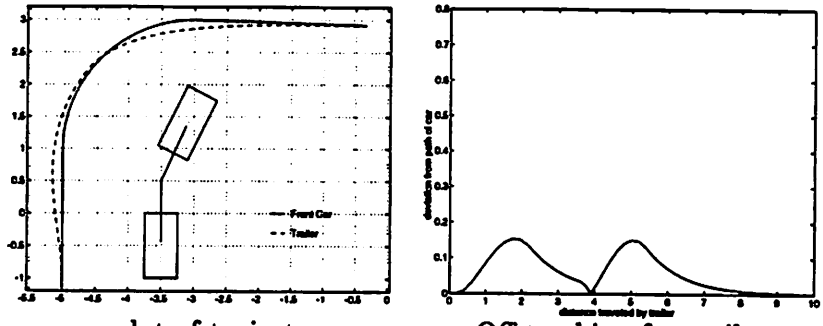
Consider first the case of a car with one trailer that is attached by means of a kingpin hitch with lengths L_1 and L_2 with total length of 3 units. To investigate what

[2]Thanks to Anant Sahai and Matthew Secor for writing the software code on a *SiliconGraphics* workstation and for performing the simulations presented in this section.



x-y plot of trajectory. Off-tracking for trailer.

Figure 6.8: Case (a): $L_1 = 1.5$, $L_2 = 0.5$. The car (solid line) pulling one trailer (dashed line) with unequal hitch lengths making a right hand turn.



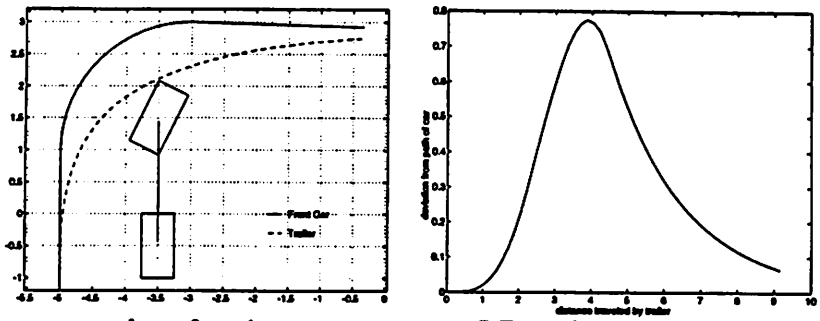
x-y plot of trajectory. Off-tracking for trailer.

Figure 6.9: Case (b): $L_1 = L_2 = 1.0$. The car (solid line) pulling one trailer (dashed line) with equal hitch lengths making a right hand turn.

happens under different hitching configurations, these lengths are varied for the three test cases: (a) $L_1 > L_2$, (b) $L_1 = L_2$, and (c) $L_1 = 0$. The vehicle is driven through a sharp right turn of radius $r = 2.0$ and the resulting trajectories of the centers of the axles of the car and trailer are plotted. In addition, the trailer's off-tracking is plotted as a function of the distance traveled by the trailer for each case.

For case (a) with $L_1 = 1.5 > L_2 = 0.5$ as shown in Figure 6.8, the trailer's path swings out and rapidly settles to a constant distance away from the circular path of the car. From equation (6.16), this off-tracking distance is $R_s - r = \sqrt{6} - 2.0 \approx 0.449$, which is the maximum off-tracking for the entire path.

For case (b) with $L_1 = L_2$ as shown in Figure 6.9, the trailer closely follows the path of the front car. The initial off-tracking to the left of the car's track is called the *swing-out* [72]. For the first half of the path, the maximum off-tracking for the kingpin



x-y plot of trajectory. Off-tracking for trailer.

Figure 6.10: Case (c): $L_1 = 0$, $L_2 = 2.0$. The car (solid line) pulling one trailer (dashed line) with axle-to-axle hitching making a right hand turn.

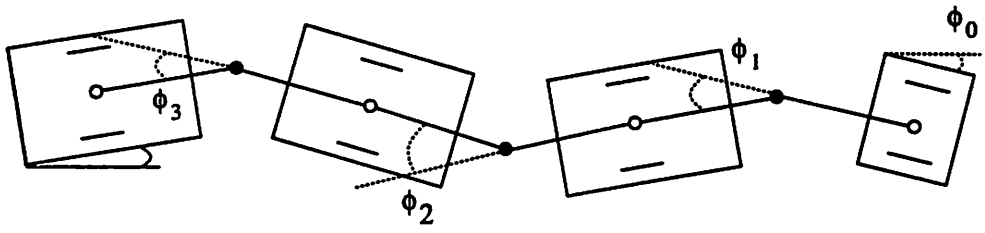


Figure 6.11: Configuration of a car pulling three trailers with kingpin hitching.

hitch and the trailer is computed from equation (6.18) with $\lambda = 2$ to be 0.236. For the second half of the path, the maximum off-tracking is computed from equation (6.20) with $\lambda = 2$ to be 0.267.

The third case (c) with $L_1 = 0$ and $L_2 = 2 = r$ as shown in Figure 6.10 is the axle-to-axle hitching configuration that has been used widely in the literature. The figure shows that the trailer has a large off-tracking to the right of the car's track. In an intersection, this would cause an intrusion beyond the pavement's edge.

In comparing the three cases, the case with equal kingpin hitching yields the best results; the trailer follows more closely to the lead car's path than with the other two hitching configurations.

The interactive software package was used to drive a car pulling three trailers as shown in Figure 6.11. Figure 6.12 shows the trajectories of the centers of the four axles as the vehicle is driven through an obstacle field. The lengths of all of the hitches were equal and set to 1.0, giving a total vehicle length of 7 units. As the figure indicates, the path swept out by the car and its trailers is not much larger than that of a single car. This allows the system

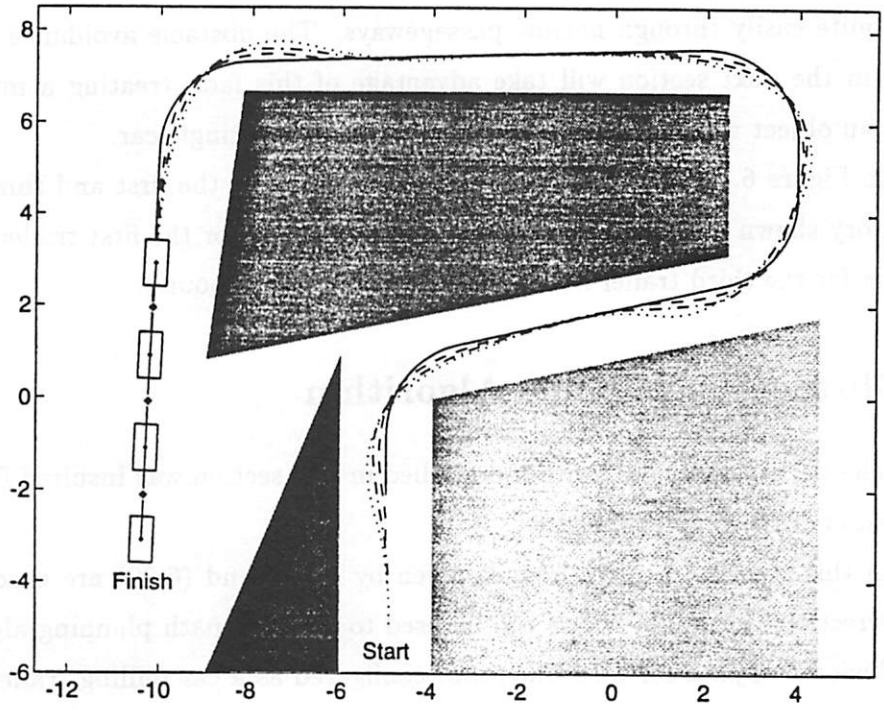


Figure 6.12: Experimental data showing path of a car pulling three trailers through an obstacle field. The lengths of the hitches are all set to 1.0, giving a total vehicle length of 7 units. The front car's trajectory is the solid line.

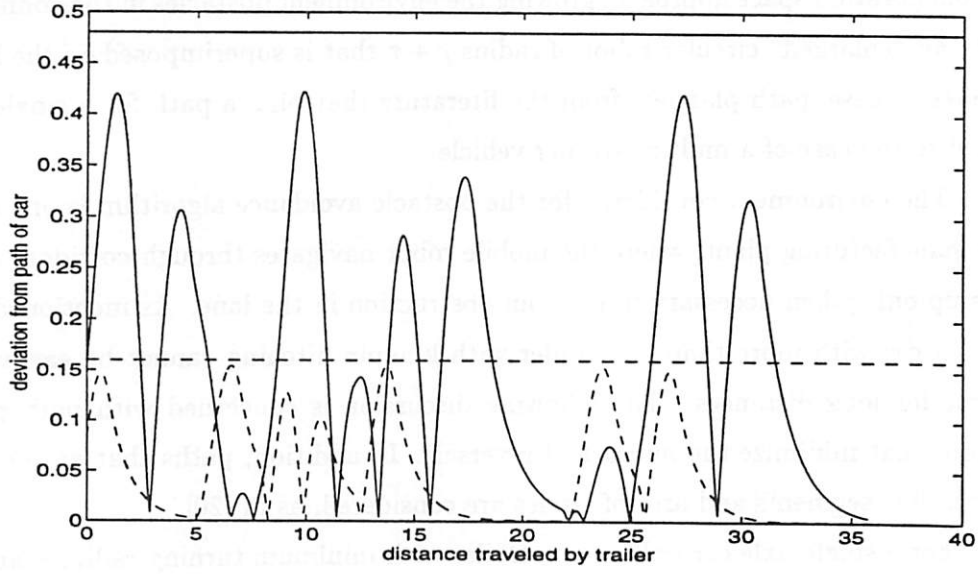


Figure 6.13: The actual off-tracking in the obstacle field for the third (solid line) trailer is less than three times the off-tracking of the first (dotted line) trailer.

to travel quite easily through narrow passageways. The obstacle avoidance methodology presented in the next section will take advantage of this fact, treating a multiple-trailer vehicle as an object whose extent is very close to that of a single car.

In Figure 6.13, the actual off-tracking is shown for the first and third trailers for the trajectory shown in Figure 6.12. A conservative bound for the first trailer is 0.16. The off-tracking for the third trailer is less than three times this bound.

6.5 Obstacle Avoidance Algorithm

The path planning algorithm described in this section was inspired from the simulation results of the previous section.

In this section, the two bounds given by (6.18) and (6.20) are used to define a "trailer correction factor," τ, which will be used to design a path planning algorithm that finds a collision-free path for a mobile robot configured as a car pulling trailers connected by kingpin hitches. The path planning algorithm is then stated for two cases. If the turning radius of the lead car is upper bounded by τ plus the radius, ρ, of a circular robot superimposed on its body, then a visibility graph method can be used to plan a path for the entire vehicle. If the turning radius is not constrained, an alternate algorithm is given that uses a configuration space approach, growing the environment obstacles in the configuration space by an "enlarged" circular robot of radius $\rho + \tau$ that is superimposed on the lead car. In the second case, path planners from the literature that plan a path for a single car are enhanced to the case of a multiple-trailer vehicle.

The environment considered for the obstacle avoidance algorithm is one of a factory or manufacturing plant, where the mobile robot navigates through corridors or lanes, backing up only when necessary to clear an obstruction in the lane. As mentioned in Section 6.2, a car with more than one trailer with kingpin hitching cannot be easily steered in reverse for long distances. The following discussion is concerned with path planning algorithms that minimize the number of reversals. In addition, paths that are only made of straight line segments and arcs of circles are considered, as in [20].

For a single-axle car pulling one trailer with minimum turning radius r and equal length off-axle hitching, define $Z := \max(z_1, z_3)$, where z_1 is the upper bound calculated in equation (6.18) and z_3 is the upper bound calculated in equation (6.20). Thus, Z represents the worst possible off-tracking for a trailer and kingpin hitch over all permissible paths of

the lead car.

In the previous section, Figure 6.13 shows that the deviation of the third trailer from the lead cars path is less than three times the deviation of the first trailer. Using this, a conservative trailer correction factor, τ, can be defined for a car pulling N trailers. We propose to add the upper bounds for each trailer to define $\tau := N \cdot Z$. The linear scaling is justified since the convergence is exponential in the distance traveled and the bounds can be propagated backwards in the trailer system.

Remark. It is of interest to explore the concept of making the trailer correction factor time varying, such as having it also depend on the radius of curvature of the circle the lead car is currently traveling on, which would enable the path planner to produce more agile trajectories. In addition, for the visibility graph method, the trailer correction factor must take into account the possibility that the trailers may have a turning radius different from the lead car. This is left as an open problem (see Chapter 7).

The proposed path planning algorithm can now be stated. In both versions, the key idea is to use a circular robot of radius $\rho+\tau$, where ρ is the radius of a circle around the lead car and τ is the trailer correction factor, to compute the distances to the environment obstacles.

If the turning radius, r, of the lead car is upper bounded by $\rho + \tau$, then the following reduced visibility graph method can be used to plan a path for the entire vehicle. The car and trailers are hitched together by equal length off-axle hitches.

Algorithm A

Step 1 Check that the turning radius, r, of the lead car is upper bounded by $\rho + \tau$, the radius of the circular robot constructed from the dimension of the lead car increased by the trailer correction factor.

Step 2 Approximate the generalized polygonal environment obstacle by a set of n line segments to a desired accuracy. To do this, choose ϵ to be the error parameter for the distance between a point in the given obstacle and the approximated obstacle. The number of segments used to approximate an obstacle is proportional to $1/\sqrt{\epsilon}$.

Step 3 Grow the approximated obstacles in the configuration space using the enlarged circular robot of radius $\rho + \tau$.

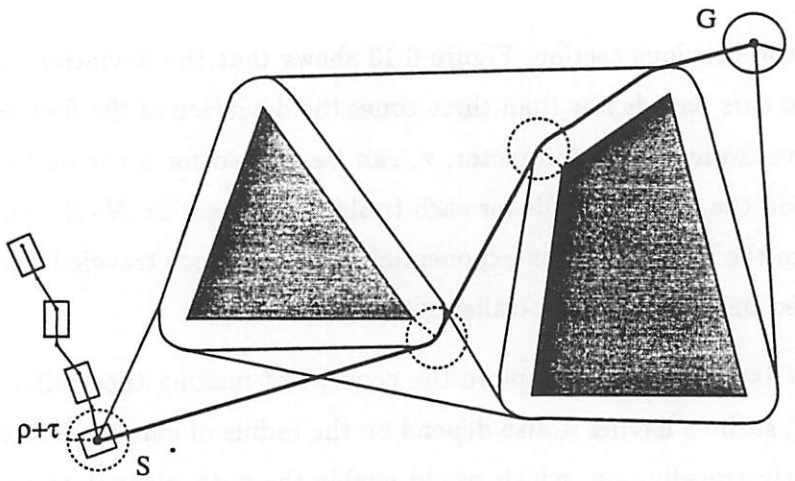


Figure 6.14: The obstacles are grown by a circular robot of radius $\rho + \tau$, which is superimposed on the lead car of a three-trailer system, and the reduced visibility graph is constructed (links in figure). The shortest path between the start, S, and goal, G, configurations is shown in bold.

Step 4 Construct the visibility graph by connecting the nodes (start and goal configurations and all the vertices of the approximated polygonal obstacles) that "see each other" with a link. This link will not intersect the obstacles.

Step 5 As in Rohnert [63], consider only the useful supporting segments, or common tangents, to a pair of obstacles.

Step 6 Search the resulting reduced visibility graph for the shortest path using Dijkstra's algorithm [19].

Figure 6.14 illustrates how the environment obstacles (shown as convex polygons) are grown for a car pulling three trailers connected by off-axle hitches. A circular robot of radius $\rho+\tau$ is used to grow the obstacles, which consist of straight line segments and arcs of a circle of radius equal to the turning radius of the car, r. In the figure, the grown obstacles are drawn with r approximately equal to $\rho + \tau$.

The construction of the visibility graph is also shown in Figure 6.14. The visibility graph consists of straight line segments and arcs of a circle of radius equal to the turning radius of the car, r. If the turning radius is upper bounded by $\rho + \tau$, this algorithm is complete (it will always find a path if one exists) and correct (if a path between the start

and goal configurations exists, then there exists an admissible path). In addition, with $r \leq \rho + \tau$, the visibility graph method will produce a path without reversals that is, in fact, the shortest path. The overall complexity of Algorithm A is $O(n^2/\epsilon)$, where n is the total number of segments used to approximate the polygonal obstacles and ϵ is the error term for the approximation.

Algorithm A is similar to the path planners of Lozano-Pérez and Wesley [43] and Rohnert [63] as presented in Section 6.1, which plan a collision-free shortest path for a circular robot moving among convex polygonal obstacles in the plane. In our case of a multiple-trailer mobile robot moving among generalized polygons, the key difference is approximating the generalized polygons and using an enlarged circular robot to grow the obstacles in the configuration space.

If there is no constraint on the turning radius of the car, r, then the above algorithm cannot be used. In this case, the following algorithm, which modifies existing path planners from the literature that plan a path for a single car to be a path planner for a multiple-trailer vehicle can be used. The class of path planners considered use a configuration space approach, where the environment obstacles are mapped into the robot's configuration space and a path is planned for a point robot in the configuration space. The car and trailers are hitched together by equal length off-axle hitches.

Algorithm B

Step 1 Construct the smallest circle of radius ρ that encloses the lead car only (the car is assumed to have the widest body; if it does not, use the widest trailer's body), making it a circular robot.

Step 2 Increase the radius of the circular robot by the trailer correction factor, τ, to account for the multiple trailers.

Step 3 Grow the environment obstacles in the configuration space using the enlarged circular robot of radius $\rho + \tau$.

Step 4 Plan the trajectory for the point robot in the configuration space within the resulting highway through the obstacles using a potential field method or a method specific to car-like robots.

As an example of how to apply this method to plan a path for a multiple-trailer

vehicle moving in an obstacle field, consider the path planner of Laumond [35] presented in Section 6.1. In this planner, a collision-free smooth trajectory is found for a circular robot of radius ρ whose turning radius is lower bounded. To modify this planner for a multiple-trailer vehicle, a circular robot is used that is constructed from a circle enclosing the lead car of radius ρ increased by the trailer correction factor, τ. The first step grows the obstacles by the radius of the circular robot, $\rho + \tau$. Then the space of centers of curvature (dual space to configuration space) is searched for the shortest path, which is the solution.

A similar planner by Laumond *et al.* [30, 39] could be modified in the same way. The first step consists of using a general geometric path planner, which is modified to use a circular robot of radius $\rho+\tau$, to compute a collision-free path that ignores the nonholonomic constraints of the vehicle. The second step transforms this geometric path to one that obeys the nonholonomic and curvature constraints by exactly characterizing, without considering the obstacles, the minimal length constrained path connecting any two configurations of the robot. To acknowledge the obstacles, a set of subgoals is chosen along the initial geometric path by requiring that the minimum length constrained path connecting successive pairs of configurations is still collision-free under the kinematic constraints of the robot. The subpaths are then concatenated to make the final trajectory, which is optimized for near-minimal length.

As an example of how to apply Algorithm B to a potential field method for planning a path for a multiple-trailer vehicle moving in an obstacle field, consider the path planner of Laumond *et al.* [40] presented in Section 6.1. A collision-free path is first planned without taking into account the nonholonomic constraints of the mobile robot. To do this, the numerical potential field method of Barraquand and Latombe [9, 6, 7] is used. In this method, the environment workspace of the robot is modeled as a multi-scale pyramid of two-dimensional bitmap arrays. Each element of the array may be thought of as a cell with value 1 if the cell intersects an obstacle and value 0 otherwise. For every cell of value 0, the L^1 distances to the obstacles are computed, normalizing the distance between two neighboring cells to be one. This construction yields a Voronoi diagram of the workspace, showing the connectivity of the obstacle-free environment. The robot's configuration space is also modeled as a multi-resolution grid pyramid whose resolution is consistent with the resolution of the workspace grid and the dimension of the robot. It is at this step that the planner is modified to use a circular robot of radius $\rho + \tau$ to compute the distance to the obstacles. Potential fields (navigation functions) are attached to the boundary of this circular robot

at certain "control points," which are then summed into one potential field defined in the configuration space of the mobile robot. The negative gradient of this field is used to pull the robot to the desired position. To search the graph of the local minima of the potential field, a probabilistic method is used that generates random motions when the gradient method ends up in a local minimum. The resulting path is then smoothed out to remove unnecessary motions. Next, in the absence of obstacles, the resulting path is numerically approximated by a sequence of feasible, near-optimal paths using the procedure described in Laumond *et al.* [39]. A cost function is defined, and a numerical solution to the optimal control problem gives the near-optimal path. Finally, this near-optimal geometric path is approximated by a sequence of optimal paths that are collision-free. If the geometric path is obstacle-free, the algorithm terminates; else, the path is subdivided. These subdivisions are then checked for collisions. This recursive subdividing continues until a collision-free path is found. The resulting path is smoothed out by iteratively repeating the subdividing procedure on randomly chosen configuration pairs to yield a solution.

In summary, a methodology for path planning in the presence of obstacles for a car pulling trailers with kingpin hitching has been presented. The main idea was to consider the lead car alone as a circular robot whose radius is increased by the "trailer correction factor" defined in this chapter. This factor depended on the number of trailers and the calculation of upper bounds on the off-tracking of the trailers and kingpin hitches. The obstacles were then mapped into the configuration space using this enlarged circular robot. The methodology guarantees that the trailers will avoid the same obstacles that the lead car avoids.

Chapter 7

Open Problems

In this chapter, open problems in the area of nonholonomic systems are presented. The first open problem considers the issue of regions of validity. The second problem considers using vector field methods to find necessary and sufficient conditions for transforming to single-generator chained form. The third generalizes the notion of chained form systems with one generator, as considered in this dissertation, to chained form systems with more than one generator. The fourth open problem considers issues of obstacle avoidance for mobile robot systems. The last problem considers controlling the system of a car pulling trailers connected by kingpin hitches in the reverse direction.

7.1 Regions of Validity

In Section 2.1.5 a theorem was presented that gives necessary and sufficient conditions for transforming a nonlinear system with drift into a linear form. The transformation only holds within a neighborhood of the initial state of the system. In addition, in Chapter 3 two methods for transforming a nonholonomic system into multiple-input chained form or extended Goursat normal form were presented, where the transformations are also only valid within a neighborhood of the initial state. The size of neighborhood, however, is unknown. In all three of these methods, the question that arises is: ***how useful are these theorems for systems that have large motions?*** As with the use of normal forms in bifurcation theory, it is an important open problem to study the regions of validity of these methods for the nonholonomic systems considered in this dissertation.

As a way to approach this problem, I suggest applying the conversion methods to

many examples of nonholonomic systems and observing any patterns in the local and global results. The goal is to understand why certain systems produce local or global results.

7.2 Single-generator Systems

In this dissertation, necessary and sufficient conditions were given for transforming a multiple-input nonholonomic system into extended Goursat normal form using techniques from exterior differential systems. Using vector field methods, however, only sufficient conditions were given to convert to multiple-chain, single-generator chained form. I encountered difficulty in trying to derive necessary conditions for this transformation using vector field methods. It is an open problem *to use vector field methods to derive necessary and sufficient conditions for transforming a wheeled nonholonomic system into chained form.*

In Chapter 2, a theorem was presented that gives necessary and sufficient conditions for transformation to chained form or Goursat normal form in terms of the dimensions of certain filtrations associated with the system. This theorem was only stated for the case of two inputs. It is an open problem *to generalize Murray's dimension count theorem (see Theorem 14) to the case with more than two inputs.*

7.3 Multiple-generator Systems

The most general form of a chained form system in $\mathbb{R}^n$ is the $(m+1)$-input, $m(m+1)$-chain, $(m+1)$-generator chained form, which is written as

$$\begin{aligned}
&\dot{z}_j^0 = v_j && j = 0, \ldots, m \\
&\dot{z}_{ij}^1 = z_i^0 v_j && i > j \quad \text{and} \quad z_{ij}^1 := z_j^0 z_i^0 - z_{ji}^1 \quad i < j \\
&\dot{z}_{ij}^k = z_{ij}^{k-1} v_j && i \neq j; \quad k = 2, 3, \ldots n_{ij} \ ,
\end{aligned}$$

where the $v_0, \ldots, v_m$ are referred to as the *generators* and $\sum_{i,j;i\neq j} n_{ij} + 3 = n$.

For example, $m = 2$ gives a three-generator chained form system with six chains:

$$\begin{array}{llll}
\dot{z}_0^0 = & v_0 & \dot{z}_1^0 = & v_1 \qquad\qquad \dot{z}_2^0 = \quad v_2 \\
\dot{z}_{10}^1 = & z_1^0 v_0 & (z_{01}^1 := & z_1^0 z_0^0 - z_{10}^1) \quad (z_{02}^0 := \quad z_2^0 z_0^0 - z_{20}^0) \\
\dot{z}_{20}^1 = & z_2^0 v_0 & \dot{z}_{21}^1 = & z_2^0 v_1 \qquad\qquad (z_{12}^0 := \quad z_2^0 z_1^0 - z_{21}^1) \\
\dot{z}_{10}^k = & z_{10}^{k-1} v_0 & \dot{z}_{01}^k = & z_{01}^{k-1} v_1 \qquad\qquad \dot{z}_{02}^k = \quad z_{02}^{k-1} v_2 \\
\dot{z}_{20}^k = & z_{20}^{k-1} v_0 & \dot{z}_{21}^k = & z_{21}^{k-1} v_1 \qquad\qquad \dot{z}_{12}^k = \quad z_{12}^{k-1} v_2
\end{array} \tag{7.1}$$

with $k = 2, \ldots, n_{ij}$.

It is an open problem *to classify the set of nonholonomic systems, perhaps under a prolongation, that can be converted to this form*, and *if vector field (differential systems) methods can be used for the transformation.* This dissertation uses vector field methods to derive sufficient conditions for converting the kinematic model of a wheeled nonholonomic system into chained form with a single generator. For our most general mobile robot example, the multiple-steering, multiple-trailer system, the number of chains corresponds to the number of steering trains in the system and the lengths of the chains correspond to the lengths of the steering trains (including virtual axles added via dynamic state feedback or partial prolongation). The single generator, v_0 in our case, is the velocity of the front car. It is unknown what happens when there is more than one generator of if this corresponds to having more than one driving input.

It is also an open problem *whether nonholonomic systems can be converted to the dual of equation (7.1) using an exterior differential systems method.* For example, for $m = 2$ as above, I have the following conjecture that generalizes the extended Goursat normal form theorem (see Theorem 16 in Section 2.2).

Conjecture 38 *For nonholonomic systems with three inputs, converting to the dual of the chained form system (7.1),*

$$\begin{aligned} I \;=\; \{&dz_k^{10} - z_{k-1}^{10} dz_0^0,\ k = 1, \ldots, n_{10};\ dz_k^{20} - z_{k-1}^{20} dz_0^0,\ k = 1, \ldots, n_{20}; \\ &dz_k^{01} - z_{k-1}^{01} dz_1^0,\ k = 1, \ldots, n_{01};\ dz_k^{21} - z_{k-1}^{21} dz_1^0,\ k = 1, \ldots, n_{21}; \\ &dz_k^{02} - z_{k-1}^{02} dz_2^0,\ k = 1, \ldots, n_{02};\ dz_k^{12} - z_{k-1}^{12} dz_2^0,\ k = 1, \ldots, n_{12}\} \ , \end{aligned}$$

requires finding three integrable one-forms π_0, π_1, π_2 such that the following extended Goursat congruences are satisfied:

$$\begin{aligned}
d\omega_k^{10} &\equiv \pi_0 \wedge \omega_{k+1}^{10} \mod I^{(n_{10}-k)} \quad k = 1, \ldots, n_{10} - 1 \ , \quad d\omega_{n_{10}}^{10} \not\equiv 0 \mod I \\
d\omega_k^{20} &\equiv \pi_0 \wedge \omega_{k+1}^{20} \mod I^{(n_{20}-k)} \quad k = 1, \ldots, n_{20} - 1 \ , \quad d\omega_{n_{20}}^{20} \not\equiv 0 \mod I \\
d\omega_k^{01} &\equiv \pi_1 \wedge \omega_{k+1}^{01} \mod I^{(n_{01}-k)} \quad k = 1, \ldots, n_{01} - 1 \ , \quad d\omega_{n_{01}}^{01} \not\equiv 0 \mod I \\
d\omega_k^{21} &\equiv \pi_1 \wedge \omega_{k+1}^{21} \mod I^{(n_{21}-k)} \quad k = 1, \ldots, n_{21} - 1 \ , \quad d\omega_{n_{21}}^{21} \not\equiv 0 \mod I \\
d\omega_k^{02} &\equiv \pi_2 \wedge \omega_{k+1}^{02} \mod I^{(n_{02}-k)} \quad k = 1, \ldots, n_{02} - 1 \ , \quad d\omega_{n_{02}}^{02} \not\equiv 0 \mod I \\
d\omega_k^{12} &\equiv \pi_2 \wedge \omega_{k+1}^{12} \mod I^{(n_{12}-k)} \quad k = 1, \ldots, n_{12} - 1 \ , \quad d\omega_{n_{12}}^{12} \not\equiv 0 \mod I \ .
\end{aligned}$$

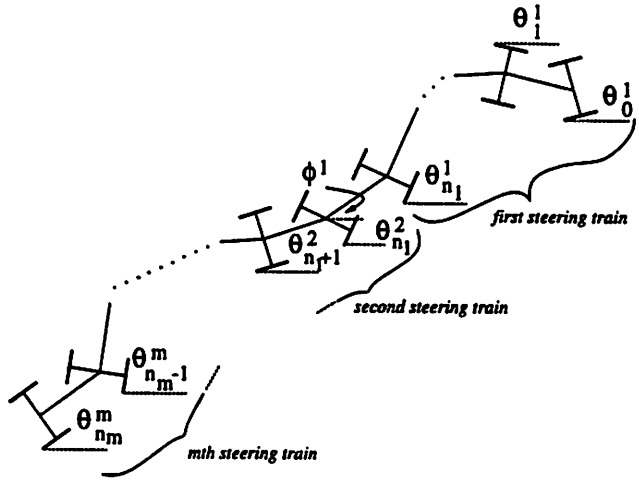


Figure 7.1: The configuration of a multiple-steering, multiple-trailer mobile robot.

7.4 Obstacle Avoidance for Mobile Robots

A path planning algorithm was presented in Chapter 6 for the two-input nonholonomic system of a car pulling trailers that are connected by kingpin hitches. This planner relied on defining off-tracking bounds for the trailers and kingpin hitches. These bounds were then used to define a "trailer correction factor," which is used to enlarge a circular robot superimposed on the lead car in order to grow the environment obstacles in the configuration space. The trailer correction factor used in Chapter 6 was defined to be an upper bound on the off-tracking times the number of trailers. Linearly adding in a fixed bound for each trailer was shown to be conservative in the simulation results. It is an open problem *to define a less conservative trailer correction factor, possibly time varying, to be used in the obstacle avoidance planner presented in this dissertation.* In addition, for the visibility graph method, if the trailers have a turning radius different from that of the lead car, then it is questionable whether the off-tracking bounds can be added for the trailer correction factor. It is an open problem *to define a trailer correction factor that takes into account the possibility of the trailers having a turning radius different from the lead car.*

It is known, both from practical experience and from the theory in this dissertation, that for the fire truck (a three-axle nonholonomic system), having the extra steering wheel for the back axle greatly improves maneuverability along narrow city streets. Numerical simulation results were presented that support this claim in Section 5.3. The general mobile

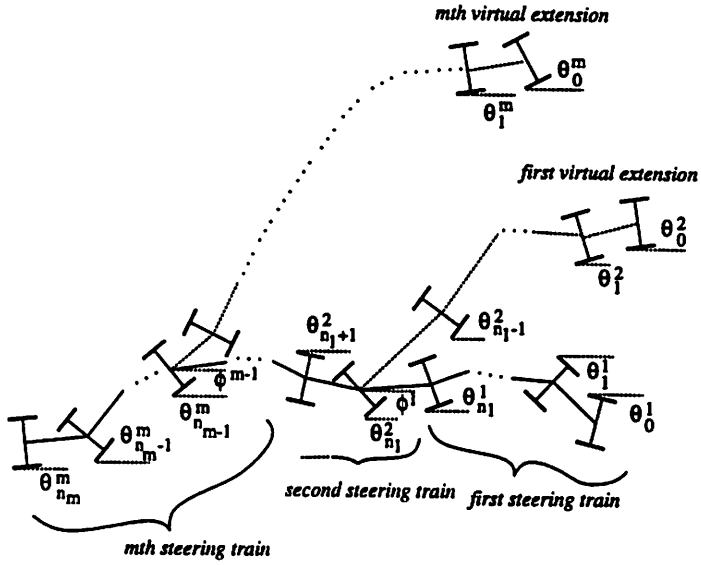


Figure 7.2: The multiple-steering, multiple-trailer mobile robot with added virtual axles, which are used to convert the system to chained form or Goursat normal form.

robot with multiple trailers and multiple steering wheels is configured as in Figure 7.1. This system was shown to be convertible to chained form in [78] by using dynamic state feedback. This conversion method required adding virtual axles in front of the steerable axles, as was explained in Section 3.2 (see Figures 7.2 and 5.13 from Section 5.3). It is an open problem *to develop an obstacle avoidance path planner for the fire truck and for more general systems of cars pulling trailers with steering on some of the trailers.* It is hoped that the path planner presented in this dissertation will be extendible to this case.

7.5 Steering Mobile Robots with Trailers in Reverse

In calculating the off-tracking bounds for the trailers and kingpin hitches for a car pulling trailers in Section 6.3 and simulating the mobile robot system in Section 6.4, only the forward direction was considered. In principle, it should be possible to drive this system in the reverse direction, too, by tracking the coordinates of the last trailer and working back up to determine the coordinates of the lead car. It is an open problem *to investigate whether or not driving the vehicle in reverse is the same as driving the back wheels for the case with equal length kingpin hitches* and *to calculate the off-tracking bounds for the trailers and kingpin hitches when the vehicle is moving in reverse.*

Chapter 8

Summary

This dissertation developed methods for converting the controlled kinematics of wheeled nonholonomic vehicles to chained form using vector field methods and to extended Goursat normal form using exterior differential systems methods. The two methods are duals of each other. Conversion to chained form only gives sufficient conditions, but is easy to apply. Conversion to Goursat normal form, however, gives necessary and sufficient conditions, but requires using subtleties of exterior differential systems.

Different steering algorithms were presented that steer the mobile robots open-loop between any two given configurations. Steering with sinusoids is the natural method for controlling systems in chained form, due to its structure, and was developed originally for optimal control problems. In practice, we have found that steering with sinusoids not to be optimal in the sense that it produces trajectories with more reversals than are minimally needed. The step-by-step sinusoidal method presented in this dissertation emphasizes the chained form structure, but is not recommended in practice. The all-at-once sinusoidal method works well for trajectories that have a reversal, such as parallel parking. Other methods that were presented use polynomial control inputs and piecewise constant control inputs; both work well for trajectories without reversals.

How to convert to chained and extended Goursat normal forms and how to use the proposed steering algorithms were illustrated with the examples of a fire truck and an extended fire truck. Generalization to multiple-steering, multiple-trailer systems is straightforward, but computationally tedious. Additionally, the fire truck with and without tiller steering was steered using polynomial inputs to show the advantage of having tiller steering: the fire truck had greater maneuverability and could execute the same maneuvers in

a narrower space and with less steering effort, which agrees with the experience of real fire trucks. The procedure for transforming to chained form also applied with no modification when the attachment point between the cab and the trailer is not located at the center of the rear axle of the cab, but at some distance off of the axle, as is commonly the case for cars pulling trailers.

Another goal of this research was to develop a new path planner for a car pulling trailers connected by off-axle hitches in the presence of obstacles. This dissertation presented such an algorithm using a visibility graph method if the lead car's turning radius is upper bounded by the radius of an "enlarged" circular robot superimposed on the car. If the turning radius of the lead car is not constrained, an alternate algorithm was proposed that modifies existing path planners that use a configuration space approach for planning paths for a single car to be path planners for the multiple-trailer vehicle. The key difference between designing path planners for single cars and multiple-trailer vehicles relied on defining an off-tracking bound, which is the maximal distance that the trailers and kingpin hitches deviate from lead car's track when the car moves from one path to another in the forward direction. In addition, it was shown that the trailers exponentially converge, with respect to the distance traveled by the lead car, to their steady-state circular path when the lead car is traveling on a circular path.

The applications of this research are various. In [47], a common error of the drivers of trucks with multiple trailers that have off-axle hitching was that of excessive steering inputs. The report suggested training the drivers to minimize the steering inputs to avoid weaving and swaying instabilities. One method suggested by this dissertation is to attach flexible "whiskers" to the lead car to notify the driver how close she can come near the obstacles. The length of the whiskers would be directly proportional to how many trailers the vehicle has and the off-tracking bounds. Then, if the driver avoids hitting obstacles with the whiskers, the trailers will also avoid these obstacles. Another application is to use the motion planning methods developed here to automatically control vehicles with multiple trailers in manufacturing plants, nuclear power plants, or any areas that have narrow passageways. For vehicles with more than one steerable axle, the methods presented here may be used to automatically control the whole vehicle, or just the steerable axles behind the human driver.

Bibliography

[1] R. Abraham, J. E. Marsden, and T. S. Ratiu. *Manifolds, Tensor Analysis and Applications.* Applied Mathematical Sciences 75. Springer-Verlag, 2nd edition, 1988.

[2] Federal Highway Administration. Model curriculum for training tractor-trailer drivers. Technical report, U. S. Department of Transportation, 1986. Instructor's Manual, Part I.

[3] J. C. Alexander. On the motion of a trailer-truck. *SIAM Review*, 26(4):579, 1984. Problem 84-19.

[4] J. C. Alexander. On the motion of a trailer-truck. *SIAM Review*, 27(4):579–579, 1985. Solutions to Problem 84-19.

[5] J. C. Alexander and J. H. Maddocks. On the maneuvering of vehicles. *SIAM Journal of Applied Mathematics*, 48(1):38–51, 1988.

[6] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. Technical Report STAN-CS 89-1285, Stanford University, 1989.

[7] J. Barraquand and J.-C. Latombe. On non-holonomic mobile robots and optimal maneuvering. In J.-D. Boissonnat and J.-P. Laumond, editors, *Revue D'intelligence Artificielle*, pages 77–104. Hermes, Paris, 1989.

[8] J. Barraquand and J.-C. Latombe. On nonholonomic mobile robots and optimal maneuvering. In *4th International Symposium on Intelligent Control*, Albany, NY, 1989.

[9] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *International Journal of Robotics and Control*, 10(6):628–649, December 1991.

[10] M. Berkemeier and R. Fearing. Control of a two-link robot to achieve sliding and hopping gaits. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 286–291, Nice, France, 1992.

[11] W. M. Boothby. *An introduction to differentiable manifolds and Riemannian geometry.* Academic Press, 1986.

[12] R. W. Brockett. Feedback invariants for non-linear systems. *IFAC Congress*, 6:1115–1120, 1978.

[13] R. W. Brockett. Control theory and singular Riemannian geometry. In P. J. Hilton and G. S. Young, editors, *New Directions in Applied Mathematics*, pages 11–27. Springer-Verlag, New York, 1981.

[14] 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. Birkhäuser, 1983.

[15] R. L. Bryant, S. S. Chern, R. B. Gardner, H. L. Goldschmidt, and P. A. Griffiths. *Exterior Differential Systems.* Springer-Verlag, 1991.

[16] J. F. Canny. *The complexity of robot motion planning.* M.I.T. Press, Cambridge, 1988.

[17] S. S. Chern. *Lectures on Exterior Differential Systems.* University of California, Berkeley, 1979.

[18] W.-L. Chow. Über systeme von linearen partiellen differentialgleichungen erster ordnung. *Math. Annalen*, 117:98–105, 1940–41.

[19] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[20] 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*, 79:497–516, 1957.

[21] C. Fernandes, L. Gurvitz, and Z. X. Li. Optimal nonholonomic motion planning for a falling cat. In Z. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, chapter 10. Kluwer Academic Publishers, 1993.

[22] H. Flanders. *Differential Forms with Applications to the Physical Sciences.* Academic Press, 1963. Also available from Dover Publications, Inc., New York, 1989.

[23] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. On differentially flat nonlinear systems. In *Proceedings of the IFAC Nonlinear Control Systems Design Symposium (NOLCOS)*, pages 408–412, Bordeaux, France, 1992.

[24] R. B. Gardner and W. F. Shadwick. The GS algorithm for exact linearization to Brunovsky normal form. *IEEE Transactions on Automatic Control*, 37(2):224–230, 1992.

[25] P. A. Griffiths. *Exterior Differential Systems and the Calculus of Variations.* Birkhäuser, 1983.

[26] R. Hermann and A. J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22:728–740, 1977.

[27] L. R. Hunt, R. Su, and G. Meyer. Design for multi-input nonlinear systems. In R. W. Brockett, R. S. Millman, and H. J. Sussmann, editors, *Differential Geometric Control Theory*, pages 268–298. Birkhäuser, 1983.

[28] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, 2nd edition, 1989.

[29] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In Z. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, chapter 8. Kluwer Academic Publishers, 1993. Also in *Proceedings of IEEE International Conference on Robotics and Automation* 1989.

[30] P. Jacobs, J.-P. Laumond, and M. Taix. A complete iterative motion planner for a car-like mobile robot. *Journées de Géométrie Algorithmique*, June 1990. INRIA (sophia-Antipolis).

[31] B. Jakubczyk and W. Respondek. On linearization of control systems. *Bull. Acad. Polonaise Sci. Ser. Sci. Math.*, 28:517–522, 1980.

[32] K. Jiang, L. D. Seneviratne, W. W. E. Earles, and W. S. Ko. Minimum-time smooth path planning for a mobile robot with kinematic constraints. In *Proceedings of the IEEE International workshop on emerging technologies and factory automation - Technology for the intelligent factory*, pages 531–536, World Congress Centre, Melbourne, Australia, 1992. CRL Publishing ltd, London.

[33] D. Koditschek and M. Bühler. Analysis of a simplified hopping robot. *International Journal of Robotics and Control*, 10(6):587–605, 1991.

[34] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, 1991.

[35] J.-P. Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1120–1123, 1987.

[36] J.-P. Laumond. Nonholonomic motion planning versus controllability via the multibody car system example. Technical Report STAN-CD 90-1345, Stanford University, 1990.

[37] J.-P. Laumond. Singularities and topological aspects in nonholonomic motion planning. In Z. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, chapter 5. Kluwer Academic Publishers, 1993.

[38] J.-P. Laumond. Feasible trajectories for a mobile robot with kinematic and environment constraints. In Proceedings of Intelligent Autonomous Systems, December 1986. Amsterdam.

[39] J.-P. Laumond, P. Jacobs, M. Taix, and R. M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 1994. In press. Also available as LAAS technical report 92413, 1992.

[40] J.-P. Laumond, S. Sekhavat, and M. Vaisset. Collision-free motion planning for a nonholonomic mobile robot with trailers. In *Proceedings of the IFAC Symposium on Robot Control*, Capri, Italy, September 1994.

[41] Z. Li and J. F. Canny, editors. *Nonholonomic Motion Planning.* Kluwer Academic Publishers, 1993.

[42] T. Lozano-Pérez. Spatial planning: a configuration space approach. *IEEE Transactions on Computers*, 32(2), 1983.

[43] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the Association for Computing Machinery (ACM)*, 22(10):560–570, 1979.

[44] J. E. Marsden and T. S. Ratiu. *An Introduction to Mechanics and Symmetry.* Math Dept., University of California at Berkeley, In preparation, 1994.

[45] P. Martin and P. Rouchon. Systems without drift and flatness. In *Mathematical Theory of Networks and Systems (MTNS)*, 1993. To be published in book form by Springer-Verlag.

[46] W. R. J. Mercer, J. R. Billing, and M. E. Wolkowicz. Test and demonstration of double and triple trailer combinations. Technical Report TVS-CV-82-109, The Transportation Technology and Energy Division, Ontario Ministry of Transportation and Communications, Downsview, Ontario, August 1982.

[47] E. C. Mikulcik. Hitch and stability problems in vehicle trains. Technical Report CR 7502, Road and Motor Vehicle Traffic Safety Branch, Ministry of Transport, Ottawa, Ontario, Canada, November 1973.

[48] B. Mirtich and J. Canny. Using skeletons for nonholonomic path planning among obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2533–2540, 1992.

[49] S. Monaco and D. Normand-Cyrot. An introduction to motion planning under multirate digital control. In *Proceedings of the IEEE Control and Decision Conference*, pages 1780–1785, Tucson, Arizona, 1992.

[50] H. P. Moravec. *Robot rover visual navigation.* UMI Research Press, Ann Arbor, Michigan, 1981.

[51] H. P. Moravec. Visual mapping by a robot rover. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, Tokyo, Japan, August, 1979.

[52] R. M. Murray. Nilpotent bases for a class of non-integrable distributions with applications to trajectory generation for nonholonomic systems. Technical Report CIT/CDS 92-002, California Institute of Technology, Pasadena, 1992.

[53] R. M. Murray. Applications and extensions of Goursat normal form to control of nonlinear systems. In *Proceedings of the IEEE Control and Decision Conference*, pages 3425–3430, 1993.

[54] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robot Manipulation*. CRC Publishers, 1994.

[55] R. M. Murray and S. S. Sastry. Grasping and manipulation using multifingered robot hands. In R. W. Brockett, editor, *Robotics: Proceedings of Symposia in Applied Mathematics*, volume 41, pages 91–128. American Mathematical Society, 1990.

[56] R. M. Murray and S. S. Sastry. Steering nonholonomic systems in chained form. In *Proceedings of the IEEE Control and Decision Conference*, pages 1121–1126, 1991.

[57] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.

[58] R. M. Murray, G. Walsh, and S. S. Sastry. Stabilization and tracking for nonholonomic control systems using time-varying state feedback. In *Proceedings of the IFAC Nonlinear Control Systems Design Symposium (NOLCOS)*, pages 182–187, Bordeaux, France, 1992.

[59] H. Nijmeijer and A. J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, 1990.

[60] California Department of Public Works. Triple trailer study in California. Technical report, State of California, Business and Transportation Action Agency, Department of Public Works, Sacramento, California, March 1972.

[61] J. Pomet. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *Systems and Control Letters*, 18:139–145, 1992.

[62] 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.

[63] H. Rohnert. Shortest paths in the plane with convex polygonal obstacles. *Information Processing Letters*, 23(2):71–76, 1986.

[64] P. Rouchon, M. Fliess, J. Lévine, and P. Martin. Flatness and motion planning: the car with n trailers. In *Proceedings of the European Control Conference*, pages 1518–1522, Groningen, The Netherlands, 1993.

[65] P. Rouchon, M. Fliess, J. Lévine, and P. Martin. Flatness, motion planning and trailer systems. In *Proceedings of the IEEE Control and Decision Conference*, pages 2700–2705, San Antonio, Texas, 1993.

[66] W. F. Shadwick. Absolute equivalence and dynamic feedback linearization. *Systems and Control Letters*, 15:35–39, 1990.

[67] W. M. Sluis. *Absolute Equivalence and its Applications to Control Theory.* PhD thesis, Department of Pure Mathematics, University of Waterloo, Waterloo, Ontario, Canada, 1992.

[68] O. J. Sørdalen. Conversion of the kinematics of a car with N trailers into a chained form. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 382–387, Atlanta, Georgia, 1993.

[69] O. J. Sørdalen and O. Egeland. Exponential stabilization of chained nonholonomic systems. In *Proceedings of the European Control Conference*, pages 1438–1443, Groningen, The Netherlands, June 1993.

[70] M. Spivak. *A Comprehensive Introduction to Differential Geometry.* Publish or Perish Inc., Berkeley, CA, 1979.

[71] R. Su. On the linear equivalents of nonlinear systems. *Systems and Control Letters*, 2:48–52, 1982.

[72] R. Sweatman, R. George, Y. Tso, and E. Ramsay. A study of heavy vehicle swept path performance. Technical Report ARRB SR 48, Australian Road Research Board, 1991.

[73] J. L. Synge. A steering problem. *Quart. Appl. Math.*, 31:295–302, 1973.

[74] A. R. Teel, R. M. Murray, and G. C. Walsh. Nonholonomic control systems: From steering to stabilization with sinusoids. In *Proceedings of the IEEE Control and Decision Conference*, pages 1603–1609, Tucson, Arizona, December 1992.

[75] D. Tilbury and A. Chelouah. Steering a three-input nonholonomic system using multi-rate controls. In *Proceedings of the European Control Conference*, pages 1428–1431, Groningen, The Netherlands, June 1993.

[76] D. Tilbury, R. Murray, and S. Sastry. Trajectory generation for the N-trailer problem using Goursat normal form. In *Proceedings of the IEEE Control and Decision Conference*, pages 971–977, San Antonio, Texas, December 1993. To appear in *IEEE Transactions on Automatic Control.*

[77] D. Tilbury and S. Sastry. On Goursat normal forms, prolongations, and control systems. Technical Report UCB/ERL M94/16, University of California at Berkeley, March 1994.

[78] D. Tilbury, O. Sørdalen, L. Bushnell, and S. Sastry. A multi-steering trailer system: Conversion into chained form using dynamic feedback. In *Proceedings of the IFAC Symposium on Robot Control*, Capri, Italy, September 1994. To appear in *IEEE Transactions on Robotics and Automation.*

[79] P. Tournassoud and O. Jehl. Motion planning for a mobile robot with a kinematic constraint. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1785–1790, 1988.

[80] G. Vegter. Dynamically maintaining the visibility graph. In F. Dehne, J.-R. Sack, and N. Santoro, editors, *Proceedings of the 2nd workshop algorithms data structures (WADS)*, volume 519 of *Lecture Notes in Computer Science*, pages 425–436. Springer-Verlag, 1991.

[81] M. Vidyasagar. *Nonlinear Systems Analysis.* Prentice Hall, second edition, 1993.

[82] G. Walsh. On reorienting rigid linked bodies using internal motions. In *Proceedings of the IEEE Control and Decision Conference*, December 1991.

[83] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J.-P. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 39(1):216–222, 1994.

[84] G. C. Walsh, 30 August 1994. Personal communication.

[85] G. C. Walsh and L. G. Bushnell. Stabilization of multiple input chained form control systems. In *Proceedings of the IEEE Control and Decision Conference*, pages 959–964, San Antonio, Texas, December 1993. To appear in *Systems and Control Letters.*

[86] G. T. Wilfong. Motion planning for an autonomous vehicle. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 529–533, 1988.

[87] K. Yang. *Exterior Differential Systems and Equivalence Problems.* Kluwer Academic Publishers, 1992.

Appendix A

Appendix

Derivation of the Wheel-angle Condition

In this appendix, a novel derivation of the *wheel-angle* condition that was studied by [3, 4, 5, 73] for a two-axle car with front wheel steering is presented. This condition is used to simplify the derivation of the kinematic model of the autonomous wheeled nonholonomic systems presented in this dissertation. The net result is that one only needs to keep track of one of the wheel angles on each axle, and the other wheel angle can be derived from the wheel-angle condition. For the examples in this dissertation, each axle is modeled instead as one wheel at the center of the axle.

When driving a two-axle car on a curve in practice, the two front wheels are at different angles (see Figure A.1): the inner wheel to the curvature is at a sharper angle than the outer wheel. The wheel-angle condition states that the normals to the two steerable front wheels and the normal to the rear wheels all intersect at point P, which is the center of rotation of the circular path that the car is traveling on. This is derived geometrically as

$$\frac{L}{\tan\phi_2} = \frac{L}{\tan\phi_1} - 2a$$

or

$$L\cos\phi_2\sin\phi_1 = L\cos\phi_1\sin\phi_2 + 2a\sin\phi_1\sin\phi_2\ . \tag{A.1}$$

To derive the wheel-angle condition (A.1) using the vector field method presented in this dissertation, first let the states of the model be $(x, y, \theta, x_1, y_1, \phi_1, x_2, y_2, \phi_2)$, where (x, y) is the Cartesian location of the center of the rear axle of the car, θ is the orientation

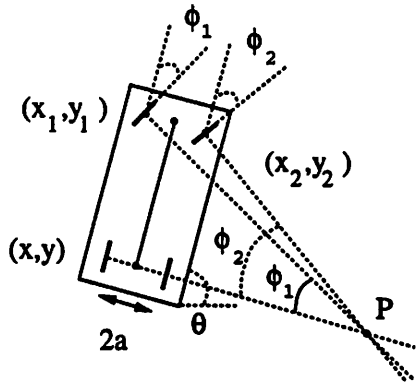


Figure A.1: Configuration of a two-axle car showing the intersecting normals. The three normals to the wheels must all intersect at point P, which is the center of rotation of the car.

of the car's body with respect to the horizontal axis of the inertial frame, (x_1, y_1) and (x_2, y_2) are the Cartesian coordinates of the center pivot points of the left and right front wheels, respectively, and ϕ_1 and ϕ_2 are the steering angles of the left and right front wheels, respectively, with respect to the car's body. Let the distance between the front and rear axles of the car be L and the width of the axles each be $2a$. The two pivot points (x_1, y_1) and (x_2, y_2) can be written in terms of the center of the rear axle of the car as

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} L\cos\theta - a\sin\theta \\ L\sin\theta + a\cos\theta \end{pmatrix}$$
$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} L\cos\theta + a\sin\theta \\ L\sin\theta - a\cos\theta \end{pmatrix} . \qquad \text{(A.2)}$$

These constraints are holonomic in the sense that they reduce by four the number of variables needed to specify the state of the system. The five coordinates $q = (x, y, \theta, \phi_1, \phi_2)$ are sufficient to represent the positions and orientations of the car and wheels.

For mechanical systems with wheels rolling and turning on a surface, the non-slipping constraint states that the velocity of the body in the direction perpendicular to each wheel must be zero. This can be stated in terms of coordinates as follows: for a wheel centered at location (x, y) and at an angle φ with respect to the horizontal axis of the fixed frame, $0 = v_x \sin\varphi - v_y \cos\varphi$.

For a two-axle car, the three non-slipping constraints

$$\begin{aligned} 0 &= \dot{x}\sin\theta - \dot{y}\cos\theta \\ 0 &= \dot{x}_1\sin(\theta+\phi_1) - \dot{y}_1\cos(\theta+\phi_1) \\ 0 &= \dot{x}_2\sin(\theta+\phi_2) - \dot{y}_2\cos(\theta+\phi_2) \end{aligned}$$

can be expressed in the equivalent form by using (A.2) as

$$\begin{aligned} 0 &= \dot{x}\sin\theta - \dot{y}\cos\theta \\ 0 &= \dot{x}\sin(\theta+\phi_1) - \dot{y}\cos(\theta+\phi_1) - \dot{\theta}L\cos\phi_1 - \dot{\theta}a\sin\phi_1 \\ 0 &= \dot{x}\sin(\theta+\phi_2) - \dot{y}\cos(\theta+\phi_2) - \dot{\theta}L\cos\phi_2 + \dot{\theta}a\sin\phi_2 \,. \end{aligned}$$

These constraints are non-integrable, or nonholonomic, and will not further reduce the reachable configuration space. They can be expressed more compactly as $\omega^i(q)\cdot\dot{q} = 0$, where the entire state is represented as $q = (x, y, \theta, \phi_1, \phi_2)$ and the covectors $\omega^i(q)$ are expressed in coordinates in $\mathbb{R}^5$ as

$$\begin{aligned} \omega^1(q) &= [\ \sin\theta & -\cos\theta & 0 & 0 & 0\ ] \\ \omega^2(q) &= [\ \sin(\theta+\phi_1) & -\cos(\theta+\phi_1) & -L\cos\phi_1 - a\sin\phi_1 & 0 & 0\ ] \\ \omega^3(q) &= [\ \sin(\theta+\phi_2) & -\cos(\theta+\phi_2) & -L\cos\phi_2 + a\sin\phi_2 & 0 & 0\ ]\,. \end{aligned} \tag{A.3}$$

Since $\omega^i(q)\cdot\dot{q} = 0$ for $i = 1, 2, 3$,

$$0 = \sin\phi_2\,\omega^2\cdot\dot{q} - \sin\phi_1\,\omega^3\cdot\dot{q}\,.$$

Substituting in the ω's, gives

$$\begin{aligned} 0 &= \sin\phi_2\left(\sin(\theta+\phi_1)\dot{x} - \cos(\theta+\phi_1)\dot{y} - (L\cos\phi_1 + a\sin\phi_1)\dot{\theta}\right) \\ &\quad - \sin\phi_1\left(\sin(\theta+\phi_2)\dot{x} - \cos(\theta+\phi_2)\dot{y} - (L\cos\phi_2 - a\sin\phi_2)\dot{\theta}\right) \\ &= \dot{x}\left(\sin\phi_2\sin(\theta+\phi_1) - \sin\phi_1\sin(\theta+\phi_2)\right) \\ &\quad -\dot{y}\left(\sin\phi_2\cos(\theta+\phi_1) - \sin\phi_1\cos(\theta+\phi_2)\right) \\ &\quad -\dot{\theta}\left(\sin\phi_2(L\cos\phi_1 + a\sin\phi_1) - \sin\phi_1(L\cos\phi_2 - a\sin\phi_2)\right)\,. \end{aligned}$$

Expanding and rearranging terms

$$\begin{aligned}
0 &= \dot{x}\,(\sin\phi_2(\sin\theta\cos\phi_1+\cos\theta\sin\phi_1)-\sin\phi_1(\sin\theta\cos\phi_2+\cos\theta\sin\phi_2)) \\
&\quad -\dot{y}\,(\sin\phi_2(\cos\theta\cos\phi_1-\sin\theta\sin\phi_1)-\sin\phi_1(\cos\theta\cos\phi_2-\sin\theta\sin\phi_2)) \\
&\quad -\dot{\theta}\,(L(\sin\phi_2\cos\phi_1-\sin\phi_1\cos\phi_2)+a(\sin\phi_2\sin\phi_1+\sin\phi_1\sin\phi_2)) \\
&= \dot{x}\,(\sin\theta(\sin\phi_2\cos\phi_1-\sin\phi_1\cos\phi_2)+\cos\theta(\sin\phi_2\sin\phi_1-\sin\phi_1\sin\phi_2)) \\
&\quad -\dot{y}\,(\cos\theta(\sin\phi_2\cos\phi_1-\sin\phi_1\cos\phi_2)-\sin\theta(\sin\phi_2\sin\phi_1-\sin\phi_1\sin\phi_2)) \\
&\quad -\dot{\theta}\,(L\sin(\phi_2-\phi_1)+2a\sin\phi_1\sin\phi_2)
\end{aligned}$$

gives the equation

$$0=\dot{x}\sin\theta\sin(\phi_2-\phi_1)-\dot{y}\cos\theta\sin(\phi_2-\phi_1)-\dot{\theta}\,(L\sin(\phi_2-\phi_1)+2a\sin\phi_1\sin\phi_2)\ .$$

But $\omega^1\cdot\dot{q}=0$, where $q=(x,y,\theta,\phi_1,\phi_2)$, i.e, $\dot{x}\sin\theta-\dot{y}\cos\theta=0$. Therefore

$$0 = \dot{x}\sin\theta\sin(\phi_2-\phi_1)-\dot{y}\cos\theta\sin(\phi_2-\phi_1)\ ,$$

which means

$$0 = \dot{\theta}\,(L\sin(\phi_2-\phi_1)+2a\sin\phi_1\sin\phi_2)\ . \tag{A.4}$$

For a car to roll without slipping, two cases arise in equation (A.4). The first case is when $\dot{\theta}=0$, which corresponds to pure translation of the car. The two front wheels are parallel to each other, causing the center of rotation to be at infinity. The second case is when $\dot{\theta}\neq 0$. This yields a pure rotation about some point:

$$0 = L\sin(\phi_2-\phi_1)+2a\sin\phi_1\sin\phi_2\ ,$$

which is the wheel-angle condition. In this case, the two front wheels cannot be parallel to each other; it must be that $\phi_1\neq\phi_2$. In a pure rotation, the inner wheel to the curvature is at a sharper angle than the outer wheel.

In this dissertation, the wheel-angle condition is used to keep track of only one of the front wheel angles of the mobile robot, say ϕ_1. Then the wheel angle ϕ_2 can always be found from equation (A.1).