A MULTI-STEERING TRAILER SYSTEM: CONVERSION INTO CHAINED FORM USING DYNAMIC FEEDBACK

by

D. Tilbury, O. Sørdalen, L. Bushnell, and S. Sastry

Memorandum No. UCB/ERL M93/55

19 July 1993

A MULTI-STEERING TRAILER SYSTEM: CONVERSION INTO CHAINED FORM USING DYNAMIC FEEDBACK

by

D. Tilbury, O. Sørdalen, L. Bushnell, and S. Sastry

Memorandum No. UCB/ERL M93/55

19 July 1993

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

A Multi-Steering Trailer System: Conversion into Chained Form using Dynamic Feedback*

D. Tilbury, O. Sørdalen, L. Bushnell, S. Sastry

Electronics Research Laboratory
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720

email: dawnt@eecs.berkeley.edu

July 19, 1993

Abstract. In this paper, we examine in detail the kinematic model of an autonomous mobile robot system consisting of a chain of steerable cars and passive trailers, connected together with rigid bars. We define the state space and kinematic equations of the system, modeling the pair of wheels on each axle as able to roll but not slip. We then investigate how this system of kinematic equations may be converted into a multi-input chained form. The advantages of the chained form are that many methods are available for the open-loop steering of such systems as well as for point-stabilization.

In order to convert the system to this multi-input chained form, we use dynamic state feedback. We draw some motivation from the very simple example of a kinematic unicycle and the relationships of the angular velocities therein, and we show how the dynamic state feedback that we use corresponds to adding, in front of the steerable cars, a chain of *virtual axles* which diverges from the original chain of trailers.

We briefly discuss how some of the methods which have been proposed for steering and stabilizing two-input chained form systems can be generalized to multi-chained systems. For concreteness, we also present two different example systems: a fire truck (three axles) and a five-axle, two-steering system. Simulation results for a parallel-parking maneuver for the five-axle system are included in the form of margin movies.

*This research was supported in part by NSF under grant IRI-9014490. D. Tilbury would like to acknowledge an AT&T Ph.D. Fellowship for partial support of this work. The research of O. Sørdalen was supported in part by the Center of Maritime Control Systems at the Norwegian Institute of Technology and SINTEF.

A MULTI-STEERING TRAILER SYSTEM: CONVERSION INTO CHAINED FORM USING DYNAMIC FEEDBACK

by

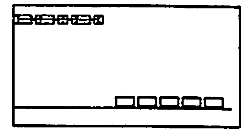
D. Tilbury, O. Sørdalen, L. Bushnell, and S. Sastry

Memorandum No. UCB/ERL M93/55

19 July 1993

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720



1. Introduction

In this paper we consider and solve the motion planning problem for a car-like mobile robot pulling a combination of n passive trailers and $m-1$ car-like robots. The controls available to the system are the velocity (throttle) of the lead car and the steering velocities of all m car-like robots. We refer to the system as a multi-steering n-trailer system. It can be thought of as a generalization of an n-trailer system, in which the only two controls available were the driving and steering velocities of the lead car. The motion planning problem for the single-steering n-trailer system was considered and solved by us in [20, 24]. Mobile robot systems of this kind are of interest in practical applications; part of the motivation for this work came from our work on the fire truck [5, 23] (the fire truck is an example of a system with three axles and two steering wheels, corresponding to the driver in the front and tiller in the rear). Also, we have been told anecdotally about the construction of such n-trailer systems with multi-steering for use in nuclear environments [6] and also for baggage handling in new airports [10]. In all of these applications it is felt that the multi-steering systems will be more maneuverable than a single-steering system.

Such systems can be modeled as having one constraint on each axle; namely, that the wheels are allowed to roll but not to slip; this constraint is nonholonomic or nonintegrable and will not reduce the reachable configuration space of the mobile robot. As in our earlier work we have not been explicit in taking into account the existence of differentials in systems of this kind which results in the two wheels of a single axle moving through different amounts in the course of a turn [1].

The present system appears at first glance to be a straightforward extension of the systems we considered in [5, 20, 23, 24], but our main motivation in writing this paper is to show how much richer and more complex the current system is in its structure. The strategy we will use for motion planning is to first convert the kinematic equations into a multi-chained form. The transformation of the system into chained form, however, requires dynamic state feedback. Motivated by the physical structure of the constraints involved in this particular problem, the dynamic state feedback that we use consists of adding *virtual axles* to each of the steerable cars in the system. The resulting extended system can be put into chained form. This chained form, which was introduced in [16] and used in our earlier work on the single-steering n-trailer system in [20, 24], enables us to use a variety of steering and stabilization techniques that we have developed in previous work. A particularly intriguing aspect of this work is its connection with an emerging body of literature in differentially flat systems by Fliess and his co-workers [8, 19]. In their work they have shown that chained form systems are a special case of what are known as differentially flat systems; the *bottoms* of the chains in the chained form play the role of flat outputs. For the two input case, it was pointed out by Martin [13] and Murray [15] that, modulo somewhat different regularity conditions, chained forms are equivalent to flat systems for the type of drift-free systems that arise in nonholonomic motion planning. The results of the current paper appear to indicate that this is not true for more than two inputs without allowing for the possibility of dynamic state feedback. As such this provides a valuable counterpoint to the results of Gardner and Shadwick [9] and our own results in [4].

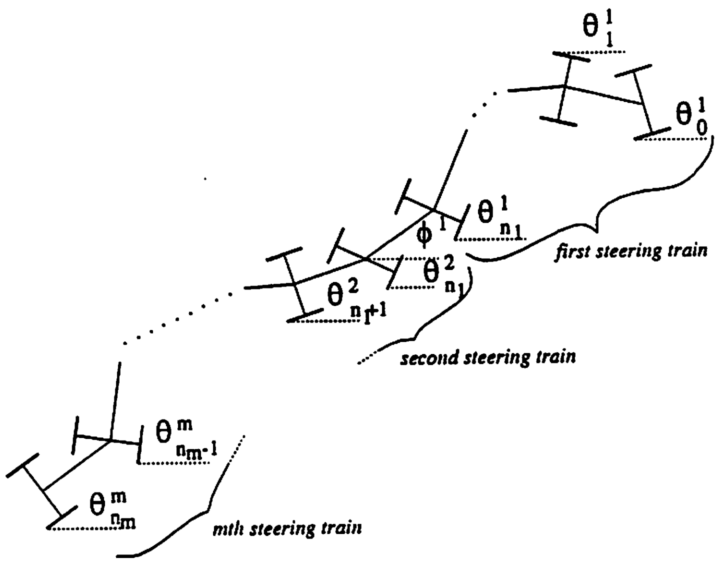
The outline of our paper is as follows: in Section 2, we develop a kinematic model of the multi-steering n-trailer system with the rolling without slipping constraints on each of the axles. In Section 3, we discuss the choice of coordinates for conversion to chained form and include the motivation for the dynamic state feedback corresponding to the addition of virtual axles. We explicitly show the change of variables required to put the extended system consisting of both real and virtual axles in chained form. In Section 4, we collect methods for steering and stabilization of the chained form systems associated with the extended multi-steering system. Section 5 contains two examples, one of which is the fire truck, worked out to demonstrate the methods of Sections 2–4. The transformation to chained form for the fire truck presented in Section 5 is different from that proposed in [5, 23] since it involves the use of a virtual axle for steering the rear axle. The question of when dynamic state feedback is absolutely necessary for this class of systems is thus an open one. Section 6 contains some concluding remarks.

2. The System Model

We consider a multi-steering trailer system, *i.e.* a system of n (passive) trailers and m (steerable) cars linked together by rigid bars. A sketch of such a system is given in Figure 1. We assume each body (trailer or car) has only one axle, since, as we have shown in [24], a two-axle car is equivalent (under coordinate transformation and state feedback) to a one-axle car towing one trailer.

2.1. Configuration Space. The active or steering axles are numbered from front to back, starting with 1 and going up to m, and the passive axles are numbered similarly from 1 to n. There are a total of $n + m$ axles in the system. The angle of each passive axle with respect to the horizontal will be represented by θ_i^j where $i \in \{1, \dots, n\}$ is the axle number and $j \in \{1, \dots, m\}$ is the number of the steering wheel most directly in front of that axle. We will call each steerable axle together with the passive axles directly behind it a *steering train.*

The steerable axles may be interspersed among the passive axles in any fashion. We will denote the indices of the passive axles which are directly in front of the steerable axles by $n_1, \dots, n_{m-1}$. By convention, we will assume that the first axle is steerable, and we define $n_0 = 0$. The superscripts that we use associate the set of passive trailers behind each active car with that car. The angle of the first axle with respect to the horizontal is denoted by θ_0^1. If there are n_1 passive trailers in the first steering train, their angles will be denoted $\theta_1^1, \dots, \theta_{n_1}^1$. The axle directly behind the first steering train is steerable, and its angle with respect to the horizontal will be $\theta_{n_1}^2$, the superscript representing that it is in the second steering train. The (passive) axles behind the second steering wheel will be denoted $\theta_{n_1+1}^2, \dots, \theta_{n_2}^2$; the angle of the third steering wheel will be $\theta_{n_2}^3$, and so forth. For convenience



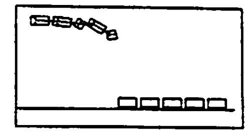


FIGURE 1. A multi-trailer system with n (passive) trailers and m (active) steering wheels.

of notation we will define $n_m \equiv n$, although in general the last axle will not be steerable. If the last axle is steerable, then we will have $n_{m-1} = n_m$.

This system as defined is a very general system, and includes the following as special cases:

(1) the standard n trailer system of [12, 16, 20, 24] corresponds to $m = 1$.
(2) the fire truck of [5, 23] corresponds to $m = 2$, $n_2 = n_1 = 1$.

The angles of the axles alone will not suffice to express the state of the system; we also need to know the angles of the rigid bars that connect each of the steerable axles to the axle in front of it. We denote by ϕ^j the absolute angle (with respect to the horizontal) of the bar connecting the $(j+1)^{st}$ steered axle to the last axle of the j^{th} steering train (which may be either steered or passive). This can be considered to be the angle of the bar connecting the $(j+1)^{st}$ steering train to the j^{th} steering train.

Additionally, the Cartesian position of the system is needed in the definition of the state. The (x, y) position of *any one* of the axles, along with all of the angles described above, will determine the state of the system. For reasons which will be explained in the sequel, we will choose the x and y positions of the *last* axle as state variables.

Therefore, the configuration of a trailer system consisting of n trailers and m cars with steering is completely given by

$$\xi = [\theta_0^1, \ldots, \theta_{n_1}^1, \phi^1, \theta_{n_1}^2, \ldots, \theta_{n_2}^2, \phi^2, \ldots, \theta_{n_{m-1}}^m, \ldots, \theta_{n_m}^m, y, x]^T \in (S^1)^{n+2m-1} \times \mathbb{R}^2 .$$

2.2. Kinematic Equations. There are many different ways to construct a kinematic model for this sort of system. The most direct way is as follows: the differential form constraints representing the non-slipping of the wheels could be written in terms of the state variables, and then the input vector fields would be constructed as the right null space of these constraints. Dualizing the $n+m$ constraints (one for each axle) in a configuration space of $n+2m+1$ state variables, we get a control system with $m+1$ inputs. The obvious choice for m of these inputs is the angular velocities of the m steering wheels, and the remaining input can be chosen as the linear velocity of the first car. The input vector fields for the steering inputs are constant vector fields of the form $[0\cdots0\;1\;0\cdots0]^T$ with the 1 in the $(n_0+1)^{st}, (n_1+2)^{nd}, \ldots,$ $(n_{m-1}+m)^{th}$ locations respectively. The driving vector field takes on a much more complex form which would take a good deal of organization and bookkeeping to derive in its general form just from looking at the form constraints.

For this reason, we have chosen to construct the kinematic model somewhat differently. Although we will eventually consider the linear velocity of the front car as one of our inputs, we will find it convenient to define the kinematic equations in terms of the linear velocity of the last body. The projections of this velocity onto the horizontal and vertical directions will be the derivatives of the state variables x and y respectively. We can then proceed towards the front of the train, recursively defining the velocities of each body in terms of the linear and angular velocities of the bodies behind it in the train. This procedure defines all the derivatives of the angles of the passive axles as well as the derivatives of the hitch angles. All that remains are the derivatives of the steering angles, and these we define to be the inputs.

As noted above, we will start at the rear of the train and let the linear velocity of the last body be denoted $v^m_{n_m}$ (recall that the angle of the last body is $\theta^m_{n_m}$). Then the derivatives of x and y are the projections of this velocity onto the horizontal and vertical directions,

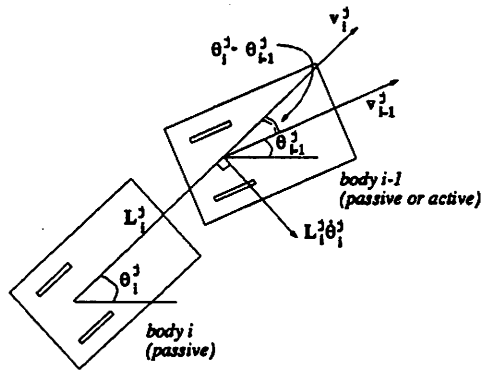
$$\begin{aligned}\dot{y} &= \sin\theta^m_{n_m} v^m_{n_m}\\ \dot{x} &= \cos\theta^m_{n_m} v^m_{n_m}\ .\end{aligned}$$

Define v^j_i to be the the linear velocity of the axle with angle θ^j_i. We can now write down the relationships between the linear and angular velocities of adjacent bodies. There are two cases to consider, corresponding to the rear body being a passive trailer or an active car.

We consider the case of a passive trailer first; refer to Figure 2 for clarification. Although this figure has been drawn for two passive trailers, we will note here that these calculations are still valid when the front body $(i-1)$ has an active steering wheel.

The linear velocity of body $i-1$ can be broken into its two perpendicular components: one is in the direction of the linear velocity of body i, and the other is along the direction of the angular velocity of body i. The two linear velocities are related by the cosine of the angle between them,

$$v^j_i = \cos(\theta^j_{i-1} - \theta^j_i)v^j_{i-1},$$



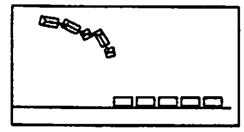


FIGURE 2. Showing the velocity relationships between adjacent bodies when the rear body is a passive trailer.

and the projection of the angular velocity is by the sine of the difference angle,

$$L_i^j \dot{\theta}_i^j = \sin(\theta_{i-1}^j - \theta_i^j) v_{i-1}^j . \qquad (1)$$

The relationship (1) gives the kinematic equations for the angles of the passive trailers, θ_i^j.

Things are only slightly more complicated when the rear body is an active car instead of a passive trailer. Refer to Figure 3 for a diagram of the case where the rear body has an active steering wheel.

Projecting onto the line connecting the two bodies, we can find the relationship between the two linear velocities,

$$v_{n_j}^{j+1} \cos(\theta_{n_j}^{j+1} - \phi^j) = v_{n_j}^j \cos(\theta_{n_j}^j - \phi^j) ,$$

that is, both velocities are multiplied by the cosine of the angle between the velocity vector and the connecting bar.

Now, because the velocity of the rear body, $v_{n_j}^{j+1}$, is no longer perpendicular to the angular velocity vector $L_{n_j}^j \dot{\phi}^j$, this linear velocity will also contribute to the angular velocity $\dot{\phi}^j$. Adding up the contributions of the two linear velocities, we obtain the relationship

$$L_{n_j+1}^j \dot{\phi}^j = \sin(\theta_{n_j}^{j+1} - \phi^j) v_{n_j}^{j+1} - \sin(\theta_{n_j}^j - \phi^j) v_{n_j}^j . \qquad (2)$$

The relationship (2) for $j \in \{1, \ldots, m-1\}$ defines the kinematics of the hitch angles.

All that remains to be determined are the derivatives of the steering wheel angles. Since these variables are free, that is they are not constrained by the kinematics, we can consider them to be controlled by the inputs,

$$\dot{\theta}_{n_{j-1}}^j = \omega^j .$$

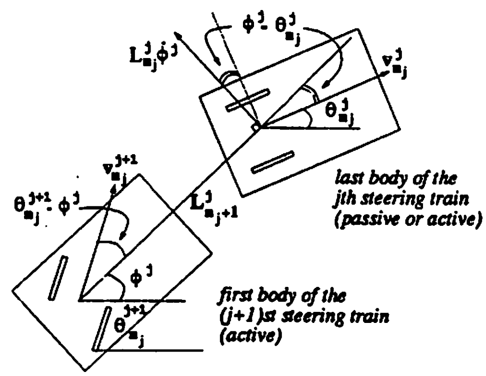


FIGURE 3. Showing the velocity relationships between two bodies when the rear body is an active car.

Combining all of these equations, we find the complete kinematic model of a trailer system consisting of n trailers and m cars with steering to be

$$\begin{aligned}
\dot\theta^j_{n_{j-1}} &= \omega^j, \quad j \in \{1,\dots,m\} \\
\dot\theta^j_i &= \frac{1}{L^j_i}\sin(\theta^j_{i-1}-\theta^j_i)v^j_{i-1}, \quad j \in \{1,\dots,m\}, i \in \{n_{j-1}+1,\dots,n_j\} \\
\dot\phi^j &= \frac{1}{L^j_{n_j+1}}[\sin(\theta^{j+1}_{n_j}-\phi^j)v^{j+1}_{n_j} - \sin(\theta^j_{n_j}-\phi^j)v^j_{n_j}], \; j \in \{1,\dots,m-1\} \\
\dot y &= \sin\theta^m_{n_m} v^m_{n_m} \\
\dot x &= \cos\theta^m_{n_m} v^m_{n_m} ,
\end{aligned} \tag{3}$$

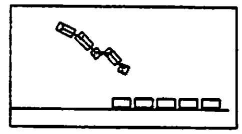
where we repeat the two velocity relationships here for reference,

$$\begin{aligned}
v^j_i &= \cos(\theta^j_{i-1}-\theta^j_i)v^j_{i-1}, \quad j \in \{1,\dots,m\}, i \in \{n_{j-1}+1,\dots,n_j\} \\
v^{j+1}_{n_j} &= \sec(\theta^{j+1}_{n_j}-\phi^j)\cos(\theta^j_{n_j}-\phi^j)v^j_{n_j} \quad j \in \{1,\dots,m-1\} .
\end{aligned} \tag{4}$$

The inputs are the angular velocities of the steering axles, $\{\omega^1,\dots,\omega^m\}$, and the linear velocity of the first car, v^1_0.

3. CONVERSION TO MULTI-INPUT CHAINED FORM

Now that we have the kinematic behavior of the multi-steering system, we will show how these equations can be converted to multi-input chained form. The rationale for this conversion is that once the equations are in this chained form, there are many methods available for either steering or stabilizing the system (some of these methods are discussed in Section 4).



3.1. Multi-input Chained Form. For definiteness in what follows, we indicate the form of a multi-input chained form system:

$$
\begin{array}{llllllll}
\dot{z}_0^0 = u^0 & \dot{z}_0^1 = u^1 & \dot{z}_0^2 = u^2 & \cdots & \dot{z}_0^m = u^m \\
 & \dot{z}_1^1 = z_0^1 u^0 & \dot{z}_1^2 = z_0^2 u^0 & & \dot{z}_1^m = z_0^m u^0 \\
 & \vdots & \vdots & & \vdots \\
 & \dot{z}_{n_1+1}^1 = z_{n_1}^1 u^0 & \vdots & & \vdots \\
 & & \dot{z}_{n_2+1}^2 = z_{n_2}^2 u^0 & \ddots & \vdots \\
 & & & & \dot{z}_{n_m+1}^m = z_{n_m}^m u^0 \, .
\end{array}
\qquad (5)
$$

We call this 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 Brunovsky normal form, especially when the input u^0 is set to 1. However, we emphasize that chained forms are bilinear in the input and state variables and not linear. Furthermore they are drift-free.

The state equations in each chain in (5) are multiplied by u^0; we call this the *generating* input. A more general chained form [16] can have more than one generating input, and thus multiple chains leading down from each input. In this paper, we will only be interested in this particular chained form (5) which has a single generator.

Chained form systems were first introduced in [16] as a class of systems inspired from [2] 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 steering control laws. Some sufficient conditions for converting two-input drift-free systems into chained form were presented in [16]. In later work [5] in the context of steering the fire truck (a three-input system with no drift), we gave sufficient conditions for converting a multi-input system into a multi-input chained form. Necessary and sufficient conditions for converting two-input systems into chained form were given in [15], where a connection was made between the chained form and its dual in the terminology of one-forms, called the Goursat normal form. In [24] we applied these results to show transformations to convert the system of a car with n trailers into the two-input chained form. The calculations in this context were simplified by the use of a coordinatization of the state space of the car with n trailers introduced by [20].

In fact, the techniques of Sørdalen were a way of systematically converting systems of n trailers into 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. From the form of the multi-input chained form equation (5), it is clear that the trajectories of the states at the *bottoms* of each chain, that is $z_0^0(t), z_{n_1+1}^1(t), \ldots, z_{n_m+1}^m(t)$ will determine the trajectories of all the states through the relationships

$$z_i^j = \dot{z}_{i+1}^j / \dot{z}_0^0 \quad j \in \{1, \ldots, m\},\ i \in \{0, \ldots, n_j\}. \qquad (6)$$

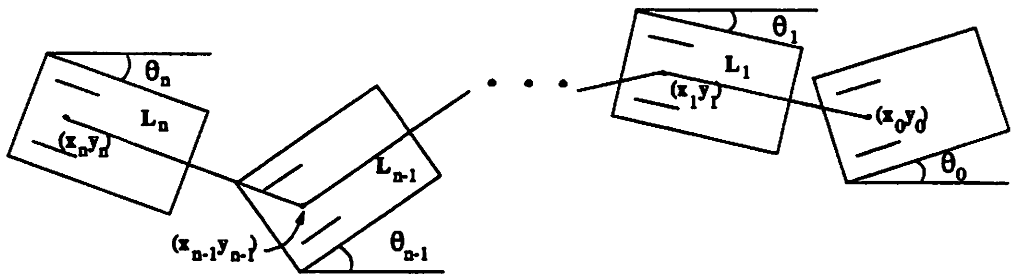


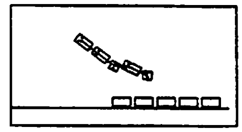
FIGURE 4. The N-trailer system, with a steering wheel at the front of the chain.

The technique that we use in this paper to convert the multi-steering trailer system into chained form is now straightforward to explain. We use our physical intuition about the system to identify those states which determine the trajectories of the system. These states become the bottoms of the chains of integrators in the chained form, as mentioned above, and the rest of the coordinate transformation is found through differentiation (equation 6). We then verify that the transformation found in this manner is a local diffeomorphism and therefore a valid change of coordinates. We know of no generalization to the Goursat normal form for multi-input systems which would give necessary and sufficient conditions for converting into multi-input chained form using dynamic state feedback.

In related work (see [8, 19]) the idea that certain variables determine the entire state of the system has been formalized in a more general setting, and these system variables have been referred to as *flat outputs.* The formal definition of flatness is given in the language of differential algebra and will not be discussed here. Instead, we will mention the definition [19] that a set of outputs $y = h(x,u)$ is called *flat* for the system $\dot{x} = f(x,u)$ if all of the system variables (states and inputs) are differential functions of the outputs y; that is x and u are meromorphic functions of the outputs y and finitely many of their derivatives. Equivalently, the flat outputs are outputs with respect to which the system has no zero dynamics [11]. A system is called *differentially flat* if a set of flat outputs can be found. Moreover, there may be many choices for the flat outputs. The multi-input chained form of (5) is differentially flat with flat outputs $z_0^0, z_{n_1+1}^1, \ldots, z_{n_m+1}^m$, although chained form systems with more than one generator are not in general flat.

3.2. The Single-steering and Multi-steering N-trailer Examples. By way of example, consider the system of n trailers with only one steering wheel (assumed to be at the front of the chain). It can be easily seen that the trajectory of the x and y positions of the midpoint of the last axle will determine the entire system trajectory. Indeed from looking at Figure 4, it can be noted that the ratio of the derivatives of these two quantities will give the tangent of the body angle of the last trailer:

$$\tan\theta_n = \dot{y}_n/\dot{x}_n.$$



The midpoint of the second-to-last trailer axle can now be expressed as a function of the midpoint of the last axle and the angle of the last trailer using the hitch relationship:

$$\begin{aligned} x_{n-1} &= x_n + L_n \cos\theta_n \\ y_{n-1} &= y_n + L_n \sin\theta_n \ , \end{aligned}$$

and knowing these two quantities as functions of time will allow the angle of the second-to-last trailer to be found

$$\tan\theta_{n-1} = \dot{y}_{n-1}/\dot{x}_{n-1}.$$

This iterative procedure will determine all of the state variables of the system up to and including the angle of the steering wheel. The steering input can then be found as the derivative of the steering angle $\dot{\theta}_0$, and the driving input as the linear velocity of the first trailer, $v = \dot{x}_0\cos\theta_0 + \dot{y}_0\sin\theta_0$.

We use a similar argument to show that for the multi-steering system of Figure 1, the (x, y) position of the last trailer along with all the hitch angles $\{\phi^1, \ldots, \phi^{m-1}\}$ between steering trains will determine the entire state of the system. Consider the last steering train: using the technique described above for the n-trailer single-steering case, we can find all the angles of the trailers up to and including that of the last (or m^{th}) steering wheel. However, the hitch angle ϕ^{m-1} ahead of this steering wheel is a free variable in the sense that it will not be determined by anything behind it. This is the reason that its evolution as a function of time is needed to specify the entire state of the system. The knowledge of the hitch angle ϕ^{m-1} will allow us to find the Cartesian coordinates of the midpoint of the last axle of the second-to-last steering train, and we use this information to find the angles of all the axles in this steering train, and so forth until we get to the front of the entire chain.

We will show that if we choose these states $x, \phi^1, \ldots, \phi^{m-1}, y$ as the bottoms of the chains in a multi-input chained form, and define the rest of the states in the chained form through equation (6), we will need to augment the state by dynamic feedback so that the derivatives of the steering inputs do not appear in the coordinate transformation. The states that we add have the physical interpretation of virtual axles in front of the steering wheels, diverging from the original chain of trailers. The inputs to this extended system are the steering velocities of the front car in each virtual chain. The number of virtual axles that need to be added in front of each steering wheel is equal to the number of passive axles that are located in front of it in the actual system.

We note here that another possible choice for the states at the bottoms of the chains are the y^i values of the midpoints of the axles in front of each of the steering wheels. The resulting chained form is the same, with the same number of states added through dynamic extension. The coordinate transformation required to put the system into the chained form, however, will be different.

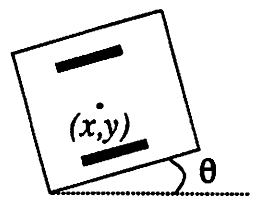


FIGURE 5. The unicycle model. The robot is allowed to drive forwards or backwards and to spin about its center axis.

3.3. Extending the System with "Virtual" Trailers. The kinematic equations for the multi-trailer system, as derived in Section 2, are drift-free and affine in the input, and can be written in the form

$$\dot{x} = G(x)\, u\ .$$

Extending the system with dynamic state feedback corresponds to adding states z, a new input ν, and defining a feedback for u

$$\begin{aligned} \dot{z} &= \alpha(x, z, \nu) \\ u &= \beta(x, z) + \gamma(x, z)\, \nu\ . \end{aligned}$$

However, to keep the resulting system drift-free and affine in the inputs, we will require that

$$\begin{aligned} \alpha(x, z, \nu) &= \alpha(x, z)\, \nu \\ \beta(x, z) &= 0\ . \end{aligned}$$

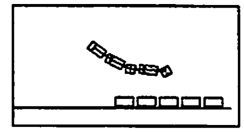
As mentioned above, the states z that we add through dynamic feedback will be interpreted physically as the angles of "virtual" axles in front of the steering wheels, diverging from the chain of trailers. The new input v will correspond to the driving velocity together with the steering velocities of the *virtual* cars, and the feedback on u will be defined such that the actual steerable cars are controlled through the chain of virtual trailers.

For some insight into this formulation of virtual axles, consider the well-known example of a unicycle, sketched in Figure 5. This is also the model of the Hilare family of mobile robots at LAAS, Toulouse. The body is allowed to drive either forwards or backwards and to spin about its axis.

The kinematic model takes as inputs the linear velocity v and the angular velocity ω of the body,

$$\begin{aligned} \dot{x} &= \cos\theta\ v \\ \dot{y} &= \sin\theta\ v \\ \dot{\theta} &= \omega\ . \end{aligned} \tag{7}$$

Since the system is drift-free, the relative degree of any choice of outputs will be equal to one (the input appears directly in every state equation). In the sequel, when considering the multi-steering trailer system, we will be interested in finding the relative degree of the states with respect to the *steering* inputs. Here we only



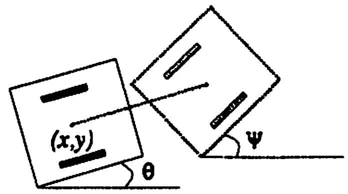


FIGURE 6. A unicycle with a "virtual" extension, interpreted as another axle added in front of the original robot.

note that the relative degree of the body angle θ with respect to the steering input is equal to one.

Consider the dynamic feedback

$$\begin{aligned} \dot{\psi} &= \alpha \\ \omega &= \tan(\psi - \theta)\ v\ . \end{aligned}$$

The extended system (x, y, θ, ψ) now satisfies the equations

$$\begin{aligned} \dot{x} &= \cos\theta\ v \\ \dot{y} &= \sin\theta\ v \\ \dot{\theta} &= \tan(\psi - \theta)\ v \\ \dot{\psi} &= \alpha\ . \end{aligned} \tag{8}$$

The added state ψ has an attractive physical interpretation of being the angle of another axle added in front of the original steering wheel, and the new input α is the steering velocity of this "virtual" wheel. This is represented in Figure 6. In addition, we see that the relative degree of the body angle θ with respect to the (virtual) steering input is now equal to two.

The linear velocity at the front axle can be denoted by v_f, and it is related to the linear velocity of the rear wheel by the cosine of the angle between them,

$$v = \cos(\psi - \theta)\ v_f\ .$$

Thus, an equivalent way to write the equations (8) would be

$$\begin{aligned} \dot{x} &= \cos\theta\ v \\ \dot{y} &= \sin\theta\ v \\ \dot{\theta} &= \sin(\psi - \theta)\ v_f \\ \dot{\psi} &= \alpha\ . \end{aligned}$$

Remark. We note here that any valid trajectory $\gamma = (x, y, \theta, \phi)$ of the system (8) can be projected down, via the standard projection $\pi : \mathbb{R}^2 \times (S^1)^2 \to \mathbb{R}^2 \times S^1$, to give a valid trajectory $\pi(\gamma) = \zeta = (x, y, \theta)$ of (7). Also, for any trajectory ζ of (7) for which $\theta(t)$ is C^1 *and* for which $\dot{\theta} = 0$ whenever $\dot{x} = \dot{y} = 0$, there exists a trajectory γ such that $\pi(\gamma) = \zeta$. Trajectories where the unicycle spins about its axis without moving either forwards or backwards cannot be achieved with the extended model.

This is the motivation for our dynamic state feedback that adds virtual axles to the system. Each virtual axle that we add in front of a steering wheel increases by one the relative degree of its hitch angle with respect to the steering input at the hitch.

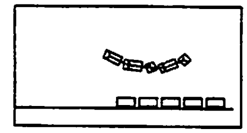
3.4. Virtual Extension for the Multi-Steering System. We now describe in detail how the kinematic model (3) is locally converted to a multi-input chained form using dynamic state feedback and a coordinate transformation.

As we described in Section 3.1, we will choose the states at the bottoms of the chains to be $x, \phi^1, \ldots, \phi^{m-1}, y$. Consider the front-most hitch angle ϕ^1; this will become the state at the bottom of the first chain, or $z^1_{n_1+1}$. Its relative degree with respect to the first steering input ω^1 is equal to $n_1 + 2$, or one more than the number of axles in the first steering train. We will need to differentiate ϕ^1 a total of $n_1 + 2$ times in order to define all the states z^1 in the first chain by equation (6). However, since $\dot{\phi}^1$ depends on all the angles *behind* it in the trailer system according to equation (3), the relative degree of ϕ^1 with respect to any of the other steering inputs $\omega^2, \ldots, \omega^m$ will be equal to two.

We do not want the derivatives of these inputs to appear in our coordinate transformation, so we will increase the relative degree of ϕ^1 with respect to the other steering inputs by adding n_1 virtual axles in front of each steering axle $\theta^j_{n_{j-1}}$ for $j \in \{2, \ldots, m\}$. The virtual inputs we temporarily denote by $\tilde{\omega}^j$, the angular velocities of the axles at the front of each virtual chain.

If we now move to consider the second hitch angle ϕ^2 which will be the state at the bottom of the second chain, we can find its relative degree with respect to the new virtual input $\tilde{\omega}^2$, which is equal to $n_2 + 2$, or one more than the number of passive axles in the second (extended) steering train. By our construction, the derivative of ϕ^2 will not depend on the first steering input ω^1 (recall that we have defined everything in terms of the velocity of the last trailer). The relative degree of ϕ^2 with respect to the other virtual steering inputs $\tilde{\omega}^3, \ldots, \tilde{\omega}^m$ is equal to $n_1 + 2$. Thus we must add $n_2 - n_1$ more virtual axles in front of each steering wheel $\theta^3_{n_2}, \ldots, \theta^m_{n_{m-1}}$ so that the derivatives of the virtual steering inputs will not appear in the coordinate transformation. Again, we temporarily denote the derivatives of the axles at the front of each new virtual chain as the virtual inputs $\hat{\omega}^3, \ldots, \hat{\omega}^m$.

We continue similarly for $\phi^3, \ldots, \phi^{m-1}, y$ and when we have finished, we will have added n_j virtual axles in front of the j^{th} steering wheel, as we have sketched in Figure 7. We note that now there are the same number of passive axles between an axle θ^j_i on the chain and any (virtual) steering wheel, and that this is the same as the number of passive axles between the axle θ^j_i and the front steering wheel θ^1_0. After we have added these virtual axles, each steering train (except the first) is considered to be augmented by the virtual axles that appear in front of it. The j^{th} steering train now contains n_j axles, of which only $n_j - n_{j-1}$ are real (physical). The only axles which we can consider as steerable in this formulation are the first



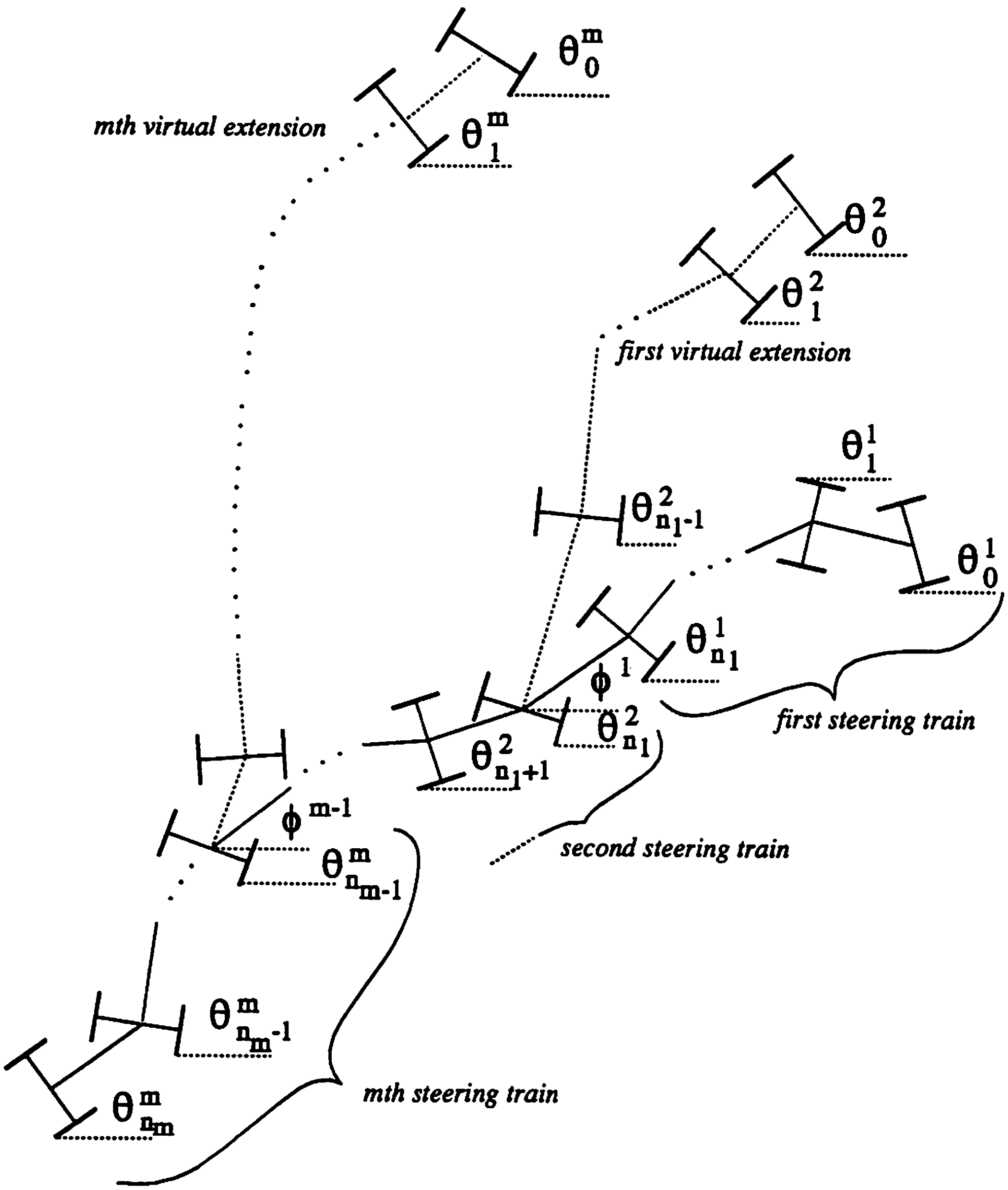


FIGURE 7. The multi-steering system, showing the virtual axles that must be added to convert the system into multi-input chained form.

axles of each virtual extension, or θ_0^j for $j \in \{1, \ldots, m\}$.

The state variables that we have introduced, which correspond to the angles of these virtual trailers, we denote by θ_i^j for $j \in \{2, \ldots, m\}$, $i \in \{0, \ldots, n_{j-1} - 1\}$. We define their derivatives as if they were actual axles,

$$\begin{aligned} \dot{\theta}_0^j &= \rho^j, \quad j \in \{2, \ldots, m\} \\ \dot{\theta}_i^j &= \frac{1}{L_i^j} \sin(\theta_{i-1}^j - \theta_i^j) v_{i-1}^j, \quad j \in \{2, \ldots, m\},\ i \in \{1, \ldots, n_{j-1} - 1\}\ , \end{aligned} \tag{9}$$

where L_i^j is an arbitrarily chosen positive parameter (usually chosen to be equal to one for simplicity). The velocities of the virtual axles are defined in the same manner as the velocities of the real axles (4),

$$v_i^j = \cos(\theta_{i-1}^j - \theta_i^j) v_{i-1}^j, \quad j \in \{1, \ldots, m\},\ i \in \{1, \ldots, n_{j-1}\}\ , \tag{10}$$

where the velocities $v_{n_{j-1}}^j$ of the actual steering wheels were also given in (4).

We will denote the new (fictitious) inputs as ρ^j, $j \in \{1, \ldots, m\}$, where as a notational convenience we will denote $\rho^1 := \omega^1$, since no virtual axles need be added in front of the first car. These inputs ρ^j represent the angular velocity of the front car in each virtual extension. In effect, we no longer directly control the angular velocities of the steering wheels which are in the middle of the chain, they are controlled indirectly through this virtual steering train. Therefore the states in our extended system will be functions of the (true) inputs ω^j and their derivatives. The true kinematic inputs ω^j are of course the derivatives of the actual steering angles,

$$\omega^j = \dot{\theta}_{n_j - 1}^j = \sin(\theta_{n_{j-1}-1}^j - \theta_{n_{j-1}}^j) v_{n_{j-1}-1}^j \quad j \in \{2, \ldots, m\}. \tag{11}$$

The complete configuration space of the extended system can now be defined as

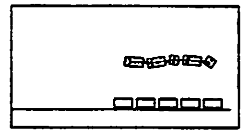
$$q = [x, \theta_0^1, \ldots, \theta_{n_1}^1, \phi^1, \theta_0^2, \ldots\ldots, \phi^{m-1}, \theta_0^m, \ldots, \theta_{n_m}^2, y]^T \in (S^1)^{N+2m-1} \times \mathbb{R}^2$$

where $N = \sum_{j=1}^m n_j$ is the number of passive axles, both real and virtual, in the extended system.

We now assume that the configuration of the system with the dynamic feedback is in a subset D of the extended configuration space, defined as the set where all of the relative angles between adjacent axles and hitches are less than $\pi/2$,

$$\begin{aligned} D \triangleq \{q \in (S^1)^{N+2m-1} \times \mathbb{R}^2\ : &|\theta_i^j - \theta_{i+1}^j| < \pi/2, \\ &|\theta_{n_j}^{j+1} - \phi^j| < \pi/2, \\ &|\phi^j - \theta_{n_j}^j| < \pi/2, \\ &|\theta_{n_m}^m| < \pi/2 \quad \}\ . \end{aligned}$$

The kinematic equations are well-defined on this set.



3.5. The Kinematics of the Extended System. In an effort to write the kinematic model in a compact form and also to show the triangular structure of the coordinate transformation, the following vectors are introduced:

$$\begin{aligned} \underline{\theta}_i^j &\triangleq [\theta_i^j, \dots, \theta_{n_j}^j, \phi^j] \\ \underline{\theta}_i^m &\triangleq [\theta_i^m, \dots, \theta_{n_m}^m, y] \\ \underline{\Theta}_i^j &\triangleq [\underline{\theta}_i^j, \underline{\theta}_i^{j+1}, \dots, \underline{\theta}_i^m] \,. \end{aligned} \tag{12}$$

To help clarify this notation, we will describe these vectors in detail. In general terms, the superscripts (j) of the vectors refer to the j^{th} steering train for $j \in \{1, \dots, m\}$. The subscripts (i) refer to the tails of the steering train starting from the i^{th} trailer (which is real if $i \geq n_{j-1}$ and virtual if $0 \leq i < n_{j-1}$). Thus, we have:

- The vector $\underline{\theta}_i^j$, for $j \in \{1, \dots, m-1\}$, $i \in \{0, \dots, n_j\}$, refers to the angles of the axles in the j^{th} steering train behind (and including) the level of the i^{th} axle. We have found it convenient to also include the hitch angle ϕ^j behind the j^{th} steering train. Thus, in particular, the vector $\underline{\theta}_0^j$ contains the angles of *all* the axles, both virtual and real, in the j^{th} steering train. Recall that θ_0^j is the angle of the virtual steering axle for the j^{th} train (except when $j = 1$, in which case it is the real steering axle). The vector $\underline{\theta}_{n_{j-1}}^j$ contains all the angles of *only the real axles* in the j^{th} steering train, including the angle of the (true) j^{th} steering axle, $\theta_{n_{j-1}}^j$.
- The vector $\underline{\theta}_i^m$, for $i \in \{0, \dots, n_m\}$, refers to the angles of the axles in the m^{th} steering train behind (and including) the level of the i^{th} axle. Since there is no hitch angle behind the last steering train, we have included the position y of the midpoint of the last axle for symmetry with the other vectors $\underline{\theta}_i^j$. Recall from Section 3.1 that the y position and the hitch angles ϕ^j will be the bottoms of the chains in the multi-input chained form equations.
- The vector $\underline{\Theta}_i^j$ groups all of the angles of the steering trains from j to m for the axles from i to the end of the j^{th} train n_j. For example, $\underline{\Theta}_i^m$ contains only angles of axles in the last steering train, from axle i to the rearmost axle of the train. The vector $\underline{\Theta}_0^1$ contains all of the angles of all the axles, both real and virtual, in the entire extended system.

We now will derive an extended kinematic model in the subset D. To this end we introduce the new fictitious input v as the linear velocity in the x direction of the last body in the last steering train. This will become the generating input u^0 in the multi-input chained form of equation (5),

$$v \triangleq \cos\theta_{n_m}^m v_{n_m}^m \,. \tag{13}$$

The linear velocities of the other bodies v_i^j can then be expressed as multiples of this generating velocity v, where we combine here the definitions (4), (10), (13):

$$
\begin{aligned}
v_{n_m}^m &= \sec\theta_{n_m}^m v \\
v_{i-1}^j &= \sec(\theta_{i-1}^j - \theta_i^j)v_i^j, \quad j \in \{1,\dots,m\},\ i \in \{1,\dots,n_j\} \\
v_{n_j}^j &= \sec(\theta_{n_j}^j - \phi^j)\cos(\theta_{n_j}^{j+1} - \phi^j)v_{n_j}^{j+1} \quad j \in \{1,\dots,m-1\}\ .
\end{aligned} \tag{14}
$$

We wish to eliminate the use of this recursive notation and write all the linear velocities as some function of the states and the generating input v:

$$v_i^j = s_i^j(\underline{\Theta}_i^j)v\ .$$

The linear velocity at a wheel θ_i^j will depend on all the difference angles behind it, $\underline{\theta}_i^j, \underline{\theta}_{n_j}^{j+1}, \dots, \underline{\theta}_{n_{m-1}}^m$. Although the vector $\underline{\Theta}_i^j$ contains more angles than this, we write the velocity v_i^j as a function of $\underline{\Theta}_i^j$. From looking at the equations (14), it can be shown that the function $s_i^j(\underline{\Theta}_i^j)$ will have the form:

$$
\begin{aligned}
s_i^j(\underline{\Theta}_i^j) = \sec\theta_{n_m}^m &\left[\prod_{k=i}^{n_j-1}\sec(\theta_k^j - \theta_{k+1}^j)\right] \\
&\prod_{l=j}^{m-1}\left\{\left[\prod_{r=n_l}^{n_{l+1}-1}\sec(\theta_r^{l+1} - \theta_{r+1}^{l+1})\right]\sec(\theta_{n_l}^l - \phi^l)\cos(\theta_{n_l}^{l+1} - \phi^l)\right\},
\end{aligned} \tag{15}
$$

and we note that this function is smooth in D.

We would now like to write the derivative of the states in a compact form,

$$\dot{\underline{\Theta}}_i^j = \underline{F}_j^i(\underline{\Theta}_{i-1}^j)$$

for some function $\underline{F}_j^i$ which we will define. We start by looking back at the derivatives of the angles θ_i^j, which have the same form for both actual (3) and virtual (9) axles,

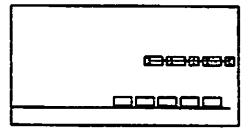
$$\dot{\theta}_i^j = \frac{1}{L_i^j}\sin(\theta_{i-1}^j - \theta_i^j)v_{i-1}^j\ .$$

We define the function f_i^j to be the derivative of θ_i^j divided by the velocity v,

$$
\begin{aligned}
f_i^j(\underline{\Theta}_{i-1}^j) &\triangleq \frac{1}{L_i^j}\sin(\theta_{i-1}^j - \theta_i^j)s_{i-1}^j(\underline{\Theta}_{i-1}^j) \\
&= \frac{1}{L_i^j}\tan(\theta_{i-1}^j - \theta_i^j)s_i^j(\underline{\Theta}_i^j)
\end{aligned} \tag{16}
$$

for $j \in \{1,\dots,m\}$, $i \in \{1,\dots,n_j\}$. We now define the functions $f_{n_j+1}^j$ to be the derivatives of the hitch angles, ϕ^j, divided by the velocity v. Recall that the equation for the kinematics of ϕ was given in (3)

$$\dot{\phi}^j = \frac{1}{L_{n_j+1}^j}[\sin(\theta_{n_j}^{j+1} - \phi^j)v_{n_j}^{j+1} - \sin(\theta_{n_j}^j - \phi^j)v_{n_j}^j],$$



so that the functions $f^j_{n_j+1}$ are

$$f^j_{n_j+1}(\underline{\Theta}^j_{n_j}) \triangleq \frac{1}{L^j_{n_j+1}}[\sin(\theta^{j+1}_{n_j} - \phi^j)s^{j+1}_{n_j} - \sin(\theta^j_{n_j} - \phi^j)s^j_{n_j}]$$
$$= \frac{1}{L^j_{n_j+1}}[\tan(\theta^{j+1}_{n_j} - \phi^j) - \tan(\theta^j_{n_j} - \phi^j)]\cos(\theta^{j+1}_{n_j} - \phi^j)s^{j+1}_{n_j}(\underline{\Theta}^{j+1}_{n_j}) \quad (17)$$

for $j \in \{1, \ldots, m-1\}$. The final function that we define here is the derivative of the y coordinate of the last trailer. From (3) we can see that equating $\dot{y} = f^m_{n_m+1}$ will give:

$$f^m_{n_m+1}(\underline{\Theta}^m_{n_m}) \triangleq \tan\theta^m_{n_m} \ . \quad (18)$$

The kinematic model (3) with the dynamic feedback (9) can then be rewritten locally in the following manner:

$$\begin{aligned}
\dot{\theta}^j_0 &= \rho^j, \quad j \in \{1, \ldots, m\} \\
\dot{\theta}^j_i &= f^j_i(\underline{\Theta}^j_{i-1})\ v, \quad j \in \{1, \ldots, m\}, i \in \{1, \ldots, n_j\} \\
\dot{\phi}^j &= f^j_{n_j+1}(\underline{\Theta}^j_{n_j})\ v, \quad j \in \{1, \ldots, m-1\} \\
\dot{y} &= f^m_{n_m+1}(\underline{\Theta}^m_{n_m})\ v \\
\dot{x} &= v \ .
\end{aligned} \quad (19)$$

By way of notation, we define

$$\begin{aligned}
\underline{f}^j_i &\triangleq [f^j_i, \ldots, f^j_{n_j+1}] \\
\underline{F}^j_i &\triangleq [\underline{f}^j_i, \underline{f}^{j+1}_i, \ldots, \underline{f}^m_i]^T
\end{aligned}$$

so that the local kinematic model with dynamic feedback (19) can now be written compactly as

$$\begin{aligned}
\dot{\theta}^j_0 &= \rho^j, \quad j \in \{1, \ldots, m\} \\
\dot{\underline{\Theta}}^1_1 &= \underline{F}^1_1(\underline{\Theta}^1_0)\ v \\
\dot{x} &= v \ .
\end{aligned} \quad (20)$$

3.6. Conversion to Multi-input Chained Form. We are now interested in finding a coordinate transformation to a multi-input chained form. As we stated before, the bottoms of the chains for the multi-input chained form will be the states $x, \phi^1, \ldots, \phi^{m-1}, y$, and the other coordinates will be found through differentiation.

The first chain has only one coordinate,

$$z^0_0 \triangleq x \ , \quad (21)$$

whose derivative will act as our generator (remember we used as our generating input $v = \cos\theta^m_{n_m} v^m_{n_m} = \dot{x}$). The m^{th} chain will be the longest, and its last coordinate is the other Cartesian coordinate,

$$z^m_{n_m+1} \triangleq y \ .$$

The other chains have the hitch angles at the bottom,

$$z^j_{n_j+1} \triangleq \phi^j, \quad j \in \{1, \dots, m-1\}, \tag{22}$$

and the remaining coordinates are found through the relationship (6).

This can be written more specifically as follows. Recalling that the derivatives of $\phi^1, \dots, \phi^{m-1}, y$, were defined as $f^j_{n_j+1}$ for $j = 1, \dots, m$, we can see that

$$\dot{z}^j_{n_j+1} = f^j_{n_j+1}(\underline{\Theta}^j_{n_j})\ v \ ,$$

and so the second-to-last coordinate in each chain will be

$$z^j_{n_j} \triangleq f^j_{n_j+1}(\underline{\Theta}^j_{n_j}) \ . \tag{23}$$

The general form of the new coordinates is

$$z^j_{i-1} \triangleq L_{\underline{F}^j_i} L_{\underline{F}^j_{i+1}} \cdots L_{\underline{F}^j_{n_j}} f^j_{n_j+1}, \quad j \in \{1, \dots, m\},\ i \in \{1, \dots, n_j\} \tag{24}$$

where $L_{\underline{F}} h$ denotes the Lie derivative of the function h along the vector $\underline{F}$. We note here that each coordinate z^j_i is a function of the state variables $\underline{\Theta}^j_i$.

The input transformation is defined by taking the derivatives of the first states in the chains for $j \in \{1, \dots, m\}$

$$u^j \triangleq \frac{\partial z^j_0}{\partial \underline{\Theta}^j_0} \dot{\underline{\Theta}}^j_0 = \frac{\partial z^j_0}{\partial \theta^j_0} \rho^j + \frac{\partial z^j_0}{\partial \underline{\tilde{\Theta}}^j_0} \dot{\underline{\tilde{\Theta}}}^j_0 \tag{25}$$

$$u^0 \triangleq v \tag{26}$$

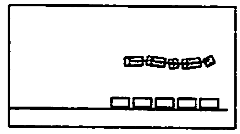
where $\underline{\tilde{\Theta}}^j_0$ is defined such that

$$\underline{\Theta}^j_0 = [\theta^j_0, (\underline{\tilde{\Theta}}^j_0)^T]^T$$

and $\dot{\underline{\tilde{\Theta}}}^j_0$ is found from (12) and (20).

Theorem 1. *Let the coordinates z^j_i, $j \in \{0, \dots, m\}$, $i \in \{0, \dots, n_j + 1\}$, be given by (21)–(24) and the inputs u^j, $j \in \{0, \dots, m\}$, be given by (25)–(26). Then the following equations are satisfied:*

$$\begin{array}{llll}
\dot{z}^0_0 = u^0 & \dot{z}^1_0 = u^1 & \dot{z}^2_0 = u^2 \quad \cdots & \dot{z}^m_0 = u^m \\
& \dot{z}^1_1 = z^1_0 u^0 & \dot{z}^2_1 = z^2_0 u^0 & \dot{z}^m_1 = z^m_0 u^0 \\
& \quad \vdots & \quad \vdots & \quad \vdots \\
& \dot{z}^1_{n_1+1} = z^1_{n_1} u^0 & \quad \vdots & \quad \vdots \\
& & \dot{z}^2_{n_2+1} = z^2_{n_2} u^0 \quad \ddots & \quad \vdots \\
& & & \dot{z}^m_{n_m+1} = z^m_{n_m} u^0 \ .
\end{array} \tag{5}$$



Proof. The chained form follows directly from the definitions of the coordinates and input transformation along with the local kinematic model (20).

To show that this coordinate transformation is a local diffeomorphism we will exhibit its triangular structure and the nonsingularity of the Jacobian at the origin.

Let the state of the entire system (including the extension) be

$$\begin{aligned} q &= [x, \underline{\theta}_0^1, \underline{\theta}_0^2, \cdots, \underline{\theta}_0^m] \\ &= [q^0, q^1, q^2, \cdots, q^m] \end{aligned}$$

where we have partitioned the states as follows:

$$\begin{aligned} q^0 &= x \\ q^1 &= \underline{\theta}_0^1 = [\theta_0^1, \theta_1^1, \cdots, \theta_{n_1}^1, \phi^1] \\ &\vdots \\ q^{m-1} &= \underline{\theta}_0^{m-1} = [\theta_0^{m-1}, \theta_1^{m-1}, \cdots, \theta_{n_{m-1}}^{m-1}, \phi^{m-1}] \\ q^m &= \underline{\theta}_0^m = [\theta_0^m, \theta_1^m, \cdots, \theta_n^m, y] \ . \end{aligned}$$

The dimension of the state space is easily seen to be $1 + (n_1 + 2) + \cdots + (n_m + 2) = N + 2m + 1$, as before.

The new states z we will partition in an analogous way, $z = [z^0, z^1, \cdots, z^m]$ where

$$z^j = [z_0^j, \cdots, z_{n_j+1}^j] \ .$$

We note here that not only do q and z have the same dimension, but q^j and z^j also have equivalent dimensions (equal to $n_j + 2$ for $j \neq 0$ and 1 for $j = 0$).

The Jacobian of the coordinate transformation,

$$J = \frac{\partial z}{\partial q}$$

is block-upper triangular and the elements on the diagonal are nonzero on the open set D of interest. This will be shown most easily by considering J block-by-block,

$$[J]_{j,k} = \frac{\partial z^j}{\partial q^k} \in \mathbb{R}^{n_j+2} \times \mathbb{R}^{n_k+2} \ .$$

Since z_i^j is a function of $\underline{\Theta}_i^j$, it is a function only of $q^j, q^{j+1}, \ldots, q^m$. This implies that the blocks $J_{j,k} = 0$ whenever $j > k$; *i.e.* all blocks below the diagonal of J are zero.

We will now show that each diagonal block $J_{j,j}$ is upper-triangular with nonzero diagonal entries. Since the first block is only one-by-one, we will consider it first. By definition $z_0^0 = x = q^0$, thus $J_{0,0} = 1$ which is nonzero.

We will now consider the other diagonal blocks, $J_{j,j}$, and we note first of all that the lower-right entry in each block is equal to 1, since by definition

$$\begin{aligned} z_{n_j+1}^j &= q_{n_j+1}^j \\ &= \begin{cases} \phi^j & j \in \{1, \ldots, m-1\} \\ y & j = m \ . \end{cases} \end{aligned}$$

To calculate the next diagonal entry, we note that $z^j_{n_j}$, which was defined in equation (23), is a function of $\underline{\Theta}^j_{n_j}$, and the dependence on $q^j_{n_j} = \theta^j_{n_j}$ is through a tangent function. Therefore,

$$\frac{\partial z^j_{n_j}}{\partial q^j_{n_j}} = \begin{cases} \frac{1}{L^j_{n_j+1}} \sec^2(\theta^j_{n_j} - \phi^j)\cos(\theta^{j+1}_{n_j} - \phi^j) s^{j+1}_{n_j}(\underline{\Theta}^{j+1}_{n_j}) & j \in \{1, \ldots, m-1\} \\ \sec^2 \theta^m_{n_m} \quad j = m \end{cases}$$

where the function s^j_i is the velocity function defined in (15) and is nonzero in D.

The other diagonal entries are found similarly. Since each z^j_i is a function of only $\underline{\Theta}^j_i$, and depends on $\theta^j_i = q^j_i$ through a tangent function, each diagonal element of the Jacobian matrix will be a product of secants and cosines of difference angles and will be nonzero on D (indeed it will be equal to one at the origin).

The input transformation can also be seen to be nonsingular. If we define $\underline{\rho}$ to be the vector of the virtual inputs,

$$\underline{\rho} = [\rho^1, \cdots, \rho^m],$$

and $\underline{u}$ to be the vector of the transformed inputs,

$$\underline{u} = [u^1, \cdots, u^m],$$

then we can look at the Jacobian matrix of this transformation,

$$J' = \frac{\partial \underline{u}}{\partial \underline{\rho}}.$$

We claim that this matrix is upper-triangular as well, and that its diagonal elements are nonzero.

Each of the inputs u^j for $j \in \{1, \ldots, m\}$ depends only on the derivatives of the states $\underline{\Theta}^j_0$, and so $J'_{j,k} = 0$ whenever $j > k$. The diagonal entries are equal to

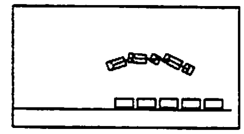
$$\frac{\partial u^j}{\partial \rho^j} = \frac{\partial z^j_0}{\partial \theta^j_0} = \frac{\partial z^j_0}{\partial q^j_0}$$

by (25), and this is the same as one of the diagonal entries of J above, and is nonzero by the same argument. □

4. Steering Chained Form Systems

Once a system is in multi-input chained form, many different algorithms can be used to steer it. We briefly describe three methods in this section; all three of them were presented in [16, 23, 24] for steering two-input systems in chained form. The basic idea behind each of the three methods is to parameterize the input space with at least as many parameters as there are states, integrate the chained form equations symbolically, and finally, solve for the input parameters in terms of the desired initial and final states.

We also mention various approaches for feedback stabilization of chained form systems. Although most of the work in these areas have concentrated on two-input



systems, the decoupled form of the multi-input chained form system will allow the techniques to be generalized in a straightforward manner.

In this section we will not deal with any particular system of trailers or nonlinear equations, but only with the multi-chained form equation of (5):

$$\begin{array}{llll} \dot z_0^0 = u^0 & \begin{aligned} \dot z_0^1 &= u^1 \\ \dot z_1^1 &= z_0^1 u^0 \\ &\vdots \\ \dot z_{n_1+1}^1 &= z_{n_1}^1 u^0 \end{aligned} & \begin{aligned} \dot z_0^2 &= u^2 \\ \dot z_1^2 &= z_0^2 u^0 \\ &\vdots \\ &\vdots \\ \dot z_{n_2+1}^2 &= z_{n_2}^2 u^0 \end{aligned} \quad \cdots \ddots & \begin{aligned} \dot z_0^m &= u^m \\ \dot z_1^m &= z_0^m u^0 \\ &\vdots \\ &\vdots \\ &\vdots \\ \dot z_{n_m+1}^m &= z_{n_m}^m u^0 . \end{aligned} \end{array} \tag{5}$$

The problem that we consider in this section is given a system of equations in the above form, and a desired initial and final state, find inputs $\{u^i(t) \, : \, t \in [0,T), \, i = 0,\ldots,m\}$ which will steer the system from the initial state to the final state.

4.1. Steering with Polynomial Inputs. One approach to the point-to-point steering problem is to hold the first input u^0 constant and identically equal to one over the entire trajectory. The time needed to steer is then determined from the change in the z_0^0 coordinate,

$$T = (z_0^0)^f - (z_0^0)^i. \tag{27}$$

We choose the parameters for the remaining inputs as coefficients of a Taylor polynomial,

$$\begin{aligned} u^1 &= a_0 + a_1 t + \ldots + a_{n_1+1} t^{n_1+1} \\ u^2 &= b_0 + b_1 t + \ldots + b_{n_2+1} t^{n_2+1} \\ &\vdots \\ u^m &= \nu_0 + \nu_1 t + \ldots + \nu_{n_m+1} t^{n_m+1} \end{aligned} \tag{28}$$

with the number of parameters on each input chosen to be equal to the number of states in its chain. The chained form equations can be integrated symbolically and the input parameters $a_j, b_j, \ldots, \nu_j$ can be found in terms of the initial and final states. This is a fairly simple procedure since all of the equations that need to be solved are linear. A symbolic manipulation program can be used quite readily to do this.

Of course, if the time needed for steering is zero from equation (27), then this method will not work. This case corresponds in the physical system to the "parallel-parking" direction, or no change in the x coordinate. The easiest way to remedy this situation is to first choose an intermediate point and then plan the path in two pieces.

4.2. Steering with Piecewise Constant Inputs. This steering method was originally inspired by multirate digital control [14], but is most easily understood in terms of motion planning simply as piecewise constant inputs. The first input u^0 is chosen to be constant over the entire trajectory. This choice will ensure the linearity of the equations that need to be solved for the other input parameters, as well as generate "nice" trajectories (since this input is related to the driving input of the multi-trailer system, a constant u^0 will usually transform to a uni-directional velocity, or equivalently no backups).

The other inputs are chosen to be piecewise constant, and to ensure that the resulting equations have a solution, each input should have at least as many switches as there are states in its chain. There will need to be the largest number of switches on the m^{th} input since it will always have the longest chain.

The time for the trajectory can be chosen arbitrarily as T. As stated before, the first input is chosen to be constant over the entire trajectory,

$$u^0(t) = u^0_{\mathrm{D}} \quad \text{for } t \in [0, T)$$

where the magnitude of the first input is chosen such that the first chained form state will go from its initial to its final position over the time period,

$$u^0_{\mathrm{D}} = \left[(z^0_0)^f - (z^0_0)^i\right]/T\,. \tag{29}$$

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

$$0 = t^j_0 < t^j_1 < \ldots < t^j_{n_j+2} = T\,,$$

where we need $n_j + 2$ switching times for each input since there are $n_j + 2$ states in the j^{th} chain. There are many different methods available for choosing these times. We will most commonly choose them so that for the m^{th} input, which has the most switching times, the holding times will be equal. We then choose the switching times for the other inputs to be some subset of the switching times for the m^{th} input. The j^{th} input will be of the form:

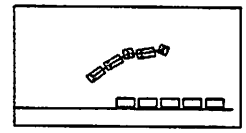
$$u^j(t) = u^{j,k}_{\mathrm{D}} \quad \text{for } t \in [t_k, t_{k+1})\,.$$

When the chained form equations are integrated using these input values, the final state can be expressed in terms of the inputs and the initial state as

$$\begin{bmatrix} z^j_0 \\ z^j_1 \\ \vdots \\ z^j_{n_j+1} \end{bmatrix}(T) = \Delta^j(u^0_{\mathrm{D}}, z^j(0)) \begin{bmatrix} u^{j,0}_{\mathrm{D}} \\ u^{j,1}_{\mathrm{D}} \\ \vdots \\ u^{j,n_j+1}_{\mathrm{D}} \end{bmatrix}$$

where the matrices Δ^j are assured to be nonsingular whenever the first input u^0_{D} is nonzero [14].

Similarly to the previous section, if the first input does come out to be zero from equation (29), then a slight modification of this method is necessary. A multirate input can also be added on u^0, using at least two time periods, or an intermediate



point can be chosen and the path can be planned in two steps. This case corresponds in the physical system to the parallel-parking direction.

We note here that the inputs we are choosing to be piecewise constant are not the velocities of the steering wheels, but the chained form inputs, which are nonlinear functions of the states and the *virtual* inputs (except in the case of the front-most steering input).

4.3. Steering with Sinusoidal Inputs. A method for steering multi-chained systems with sinusoids was proposed in [5]. This method is step-by-step and uses one step to steer each level of the chain (although the states of all chains at the same level can be steered simultaneously). Since the longest chain in our definition is length $n_m + 2$, this is the number of steps that will be needed.

The algorithm is sketched as follows:

Step 0: Steer the top-level coordinates, $\{z_0^j, \;\; j = 0, \ldots, m\}$ by choosing constant values for $u^0, u^1, \ldots, u^m$ on the time interval $[0, T)$.

Step 1: Steer the coordinates at the first level down by choosing a sinusoid on u^0 and out-of-phase sinusoids on u^j,

$$\begin{aligned} u^0 &= \alpha \sin \omega t \\ u^1 &= \beta \cos \omega t \\ u^2 &= \gamma \cos \omega t \\ &\vdots \\ u^m &= \nu \cos \omega t \end{aligned}$$

over a time period $[T, 2T)$, with appropriate choice of $\alpha, \beta, \ldots, \nu$ so that at time $2T$, the states $\{z_1^j, \;\; j = 1, \ldots, m\}$, have achieved their desired final values.

Step k: ($k = 2, \ldots, n_m + 1$). Steer the coordinates at the k^{th} level from the top. If $n_i < k \leq n_{i+1}$, then only chains $i+1, \ldots, m$ will be affected. Again, we will choose a single frequency sinusoid on the first input, but now we choose multiple frequency sinusoids on the other inputs:

$$\begin{aligned} u^0 &= \alpha \sin \omega t \\ u^1 &= 0 \\ &\vdots \\ u^i &= 0 \\ u^{i+1} &= \zeta \cos k\omega t \\ &\vdots \\ u^m &= \nu \cos k\omega t \end{aligned}$$

over a time period $[kT, (k+1)T)$, with appropriate choice of $\zeta, \ldots, \nu$ so that at time kT, the states $\{z_k^j, \;\; j = i+1, \ldots, m\}$, are at their desired final values.

We note that after each step k, the states closer to the top of the chain than level k will have returned to their values 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; we disregard these because we steer those states to their desired final values in subsequent iterations.

Although this method works perfectly well, and the magnitudes of the sinusoids are simple to solve for, the algorithm can be tedious in practice because of the many steps that are needed. The trajectories that are generated consist of many segments and do not always follow a very direct path between the start and goal.

Therefore, we also propose an "all-at-once" sinusoids method which is an extension of that detailed in [24] for the two input single chain case. We only use one step, and we put all of the necessary frequencies into the inputs.

$$\begin{aligned} u^0 &= \alpha_0 + \alpha \sin \omega t \\ u^1 &= \beta_0 + \beta_1 \cos \omega t + \cdots + \beta_{n_1+1} \cos(n_1+1)\omega t \\ &\vdots \\ u^m &= \nu_0 + \nu_1 \cos \omega t + \cdots + \nu_{n_m+1} \cos(n_m+1)\omega t \ . \end{aligned}$$

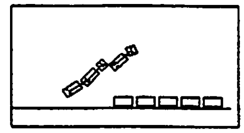
The input parameters are found in the same manner as in the other methods. The chained form equations are integrated symbolically, evaluated at time T, and the parameters are solved for as a function of the initial and final states.

The main drawback to this approach is that there will be some interference between the levels (although not between chains) and solving for the input parameters will require solving nonlinear algebraic equations. In the simple cases that we have explored, this has not been a problem for a symbolic manipulator.

When this method is implemented on a multi-steering trailer system, the first input, which always goes through one period, will transform back to the driving input, which will usually change direction (at least one backup). This seems to work well when parallel-parking type maneuvers are desired. The free parameter α can be adjusted to change the distance that the trailer system drives forward before it backs up.

4.4. Stabilization for Multi-input Chained Form Systems. We now briefly discuss some methods from the literature for stabilization of chained form systems. These systems are open-loop controllable, as shown above by the various point-to-point steering algorithms, but are not stabilizable by pure smooth static-state feedback [3]. 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 [22]. This procedure was extended in [25] to locally asymptotically stabilizing the origin of $(m+1)$-input, m-chain, single-generator chained form systems. This method consists of taking the chained



form system and converting it into power form, which has the structure (for the three-input case):

$$\begin{array}{rcl}
\dot{x}_1 &=& v_1
\end{array}
\qquad
\begin{array}{rcl}
\dot{y}_1 &=& v_2\\
\dot{y}_2 &=& x_1 v_2\\
\dot{y}_3 &=& \frac{1}{2}x_1^2 v_2\\
&\vdots&\\
\dot{y}_j &=& \frac{1}{(j-2)!}x_1^{j-2} v_2
\end{array}
\qquad
\begin{array}{rcl}
\dot{z}_1 &=& v_3\\
\dot{z}_2 &=& x_1 v_3\\
\dot{z}_3 &=& \frac{1}{2}x_1^2 v_3\\
&\vdots&\\
&\vdots&\\
\dot{z}_k &=& \frac{1}{(k-2)!}x_1^{k-2} v_3
\end{array}$$

The control laws, which are time-varying functions of the state, will stabilize the system in power form.

Many of the other results which have been presented in the literature for two-input chained form systems could also be extended to multi-input single-generator chained form systems in a straightforward manner. Thus we will briefly describe some of these results to give the reader an idea of the possibilities available for stabilizing a system once it has been converted into chained form.

In [17], 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. A method for globally stabilizing about the origin with exponential convergence rates was proposed in [21] for a two-input, single-chain chained form system. The feedback control laws were developed for the system in chained form instead of power form. And finally, [18] presents a constructive approach for deriving a time-varying smooth feedback control law which can be applied to globally uniformly asymptotically stabilize chained form systems to the origin.

5. Examples of Multi-Steering Trailer Systems

Some examples of systems that fit into the class of multi-steering n-trailer but have been examined in previous papers include the n-trailer system with one steering wheel (see, for example, [20]) and the fire truck system [5] which has three axles and two steering wheels. To illustrate the procedure that we presented in this paper for converting multi-steering trailer systems into chained form, we will follow through the algorithm given in Section 3 for two example systems.

5.1. Fire Truck Example. Although the fire truck example has been examined extensively in previous work, we will also consider it in terms of the algorithm described in this paper, since the formulation is somewhat different than in [5]. In that paper, the bottoms of the chains in the multi-input chained form were chosen to be the (x, y) position of the *passive* axle along with the angle of the trailer (see Figure 8). Because of the relative simplicity of the three-axle system, that choice allowed us to put the kinematic equations into multi-input chained form without using dynamic state feedback. The fire truck fits into the class of multi-steering trailer systems, thus we can also convert the kinematic equations into multi-input chained form using a virtual extension (and a different choice of

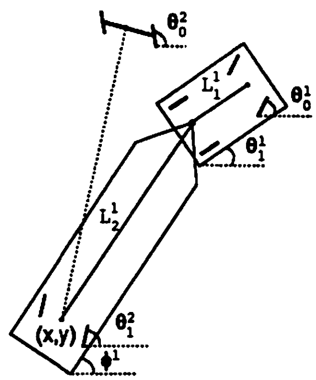


FIGURE 8. A sketch of the fire truck system showing the virtual extension that is added in front of the rear steering wheel. These vehicles are equipped with a long ladder on the trailer and are used by fire departments in large cities in the United States. The extra steering wheel at the rear of the trailer is used for improved maneuverability on narrow city streets.

states at the bottoms of the chains). Although this extension is not necessary for this particular system, we know of no systematic procedure for transforming a general multi-steering trailer system into multi-input chained form without using the sort of virtual extension that we propose in this paper.

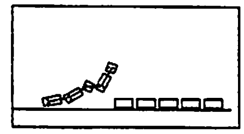
The kinematic equations for the fire truck can be obtained from equation (3):

$$\begin{aligned}
\dot\theta_0^1 &= \omega^1 \\
\dot\theta_1^2 &= \omega^2 \\
\dot\theta_1^1 &= \frac{1}{L_1^1}\sin(\theta_0^1-\theta_1^1)\ v_0^1 \\
\dot\phi^1 &= \frac{1}{L_2^1}[\sin(\theta_1^2-\phi^1)\ v_1^2 - \sin(\theta_1^1-\phi^1)\ v_1^1] \\
\dot y &= \sin\theta_1^2\ v_1^2 \\
\dot x &= \cos\theta_1^2\ v_1^2
\end{aligned} \tag{30}$$

where from equation (4) the velocities are related by

$$\begin{aligned}
v_1^1 &= \sec(\theta_1^1-\phi^1)\cos(\theta_1^2-\phi^1)\ v_1^2 \\
v_0^1 &= \sec(\theta_0^1-\theta_1^1)\ v_1^1\ .
\end{aligned}$$

The system has two steering trains, the first has length two and the second has length one. Since there is one passive axle in front of the second steering train, we will augment this train by the addition of one virtual axle as described in Section 3.4. The angle of this virtual axle is denoted θ_0^2. A sketch of the extended system is shown in Figure 8.



The kinematics of the second steering axle θ_1^2 are thus altered. We no longer steer it directly, but through the virtual steering wheel. Instead of $\dot{\theta}_1^2 = \omega^2$ we will have:

$$\dot{\theta}_1^2 = \frac{1}{L_1^2}\tan(\theta_0^2 - \theta_1^2)v_1^2$$

where the velocity v_1^2 is the linear velocity of the steering axle. The virtual input will be the angular velocity of the virtual steering wheel,

$$\dot{\theta}_0^2 = \rho^2 \ .$$

The generating input for the multi-input chained form is the linear velocity of the last body in the horizontal direction,

$$v = \cos\theta_1^2 \ v_1^2$$

and we can then find the velocities of all the other bodies in terms of this quantity from equation (14).

The bottoms of the chains in the multi-input chained form are the (x, y) coordinates of the rear axle and the hitch angle ϕ^1, and the rest of the coordinates are found through differentiation according to equation (6),

$$\begin{array}{llll} z_0^0 \triangleq x & z_2^1 \triangleq \phi^1 & z_2^2 \triangleq y \\ & z_1^1 \triangleq f_2^1(\Theta_1^1) = \dot{\phi}^1/v & z_1^2 \triangleq f_2^2(\Theta_1^2) = \dot{y}/v \\ & z_0^1 \triangleq L_{E_1^1} f_2^1 = \dot{z}_1^1/v & z_0^2 \triangleq L_{E_1^2} f_2^2 = \dot{z}_1^2/v \end{array}$$

(the chains are written upside down here to show the order in which the coordinates are calculated: starting at the bottom). The resulting equations are in multi-chained form:

$$\begin{array}{lll} \dot{z}_0^0 = u^0 & \dot{z}_0^1 = u^1 & \dot{z}_0^2 = u^2 \\ & \dot{z}_1^1 = z_0^1 u^0 & \dot{z}_1^2 = z_0^2 u^0 \\ & \dot{z}_2^1 = z_1^1 u^0 & \dot{z}_2^2 = z_1^2 u^0 \end{array}$$

where the input $u^0 := v$ is defined as the velocity of the last trailer in the horizontal direction, and the other two inputs u^1 and u^2 are defined as the derivatives of z_0^1 and z_0^2, respectively.

Once we have the system in chained form, we can steer it from point to point or stabilize it to a point using one of the methods presented in Section 4. The controls u^0, u^1, and u^2 that result from whichever method was chosen can be transformed back into the original coordinates to give the inputs v, ω^1 and the virtual input ρ^2. The actual input ω^2 can be calculated from the virtual input ρ^2 and the state trajectory.

5.2. Another Example. We will also go through an example here of a five-axle system with two steering wheels. The system is depicted in Figure 9. In effect, this system is a fire truck with two passive trailers. We note that with these extra trailers, the (x, y) position of the first passive axle, along with the trailer angle ϕ^1, will no longer define the entire state of the system. Although it is possible that this system could be converted to multi-input chained form without using dynamic

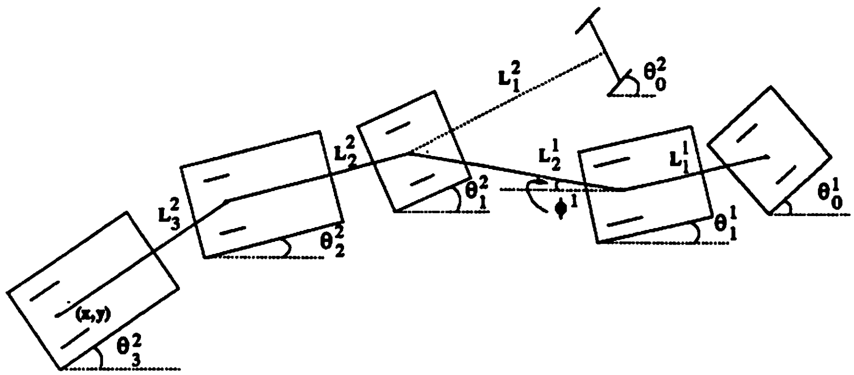
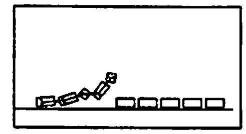


FIGURE 9. The five-axle, two-steering system showing the virtual extension which is added in front of the second steering wheel. Such a system could be envisioned as being used in a nuclear power plant or in a mining area where maneuverability around narrow passageways in danger zones is of utmost importance.

feedback, such a transformation has not been found. The virtual extension allows us to define a relatively simple transformation into multi-input chained form for any multi-steering trailer system, and once the system is in this chained form, it can be steered or stabilized using one of the methods outlined in Section 4. A parallel-parking maneuver for the system shown in Figure 9 is shown in the margin movies of this paper. More details on the generation of that trajectory will be given in Section 5.3; first, we will present the conversion to chained form.

The kinematic equations for this system can be written down from equation (3)

$$
\begin{aligned}
\dot{\theta}_0^1 &= \omega^1 \\
\dot{\theta}_1^2 &= \omega^2 \\
\dot{\theta}_1^1 &= \frac{1}{L_1^1} \sin(\theta_0^1 - \theta_1^1)\, v_0^1 \\
\dot{\theta}_2^2 &= \frac{1}{L_2^2} \sin(\theta_1^2 - \theta_2^2)\, v_1^2 \\
\dot{\theta}_3^2 &= \frac{1}{L_3^2} \sin(\theta_2^2 - \theta_3^2)\, v_2^2 \\
\dot{\phi}^1 &= \frac{1}{L_2^1} [\sin(\theta_1^2 - \phi^1)\, v_1^2 - \sin(\theta_1^1 - \phi^1)\, v_1^1] \\
\dot{y} &= \sin \theta_3^2\, v_3^2 \\
\dot{x} &= \cos \theta_3^2\, v_3^2
\end{aligned}
\tag{31}
$$



where the velocities are related by equation (4)

$$\begin{aligned} v_2^2 &= \sec(\theta_2^2 - \theta_3^2)\; v_3^2 \\ v_1^2 &= \sec(\theta_1^2 - \theta_2^2)\; v_2^2 \\ v_1^1 &= \sec(\theta_1^1 - \phi^1)\cos(\theta_1^2 - \phi^1)\; v_1^2 \\ v_0^1 &= \sec(\theta_0^1 - \theta_1^1)\; v_1^1 \end{aligned}$$

The system consists of two steering trains, the first has length two and the second has length three. We will need to add one virtual axle to the second steering train since there is one passive axle in front of its steering car; see Figure 9.

This new state corresponding to the angle of the virtual axle is denoted as θ_0^2, and the kinematics of θ_1^2 must be changed to represent that the angular velocity of the second steering wheel is no longer controlled directly by the input ω^2 but indirectly through the virtual steering wheel θ_0^2,

$$\dot{\theta}_1^2 = \tan(\theta_0^2 - \theta_1^2)\; v_1^2$$

and the input ρ^2 will now control the virtual steering velocity $\dot{\theta}_0^2$,

$$\dot{\theta}_0^2 = \rho^2 \; .$$

When needed, the real input ω^2 can be calculated as the derivative of the angle θ_1^2,

$$\omega^2 = \tan(\theta_0^2 - \theta_1^2)\; v_1^2$$

where v_1^2 is the velocity of the actual steering wheel θ_1^2 as was detailed in Section 3.

The generating input in the multi-input chained form is

$$v = \cos\theta_3^2\; v_3^2$$

which is the linear velocity of the last trailer in the horizontal direction. We can now calculate the velocities of all the other bodies in terms of this quantity from equation (14). For the coordinate transformation to chained form, as described in Section 3, the bottoms of the chains will be the (x, y) position of the end of the last trailer and the angle ϕ^1 of the hitch connected to the second steering wheel. To find the other coordinates in the chained form we differentiate, for example:

$$z_3^2 = \dot{z}_4^2 / \dot{z}_0^0 \; .$$

The complete coordinate transformation is too complicated to include here but may be obtained from the first author via electronic mail. We will show the structure of the coordinates:

$$z_0^0 \triangleq x$$

$$\begin{aligned} z_2^1 &\triangleq \phi^1 \\ z_1^1 &\triangleq f_2^1(\underline{\Theta}_1^1) = \dot{\phi}^1 / v \\ z_0^1 &\triangleq L_{\underline{F}_1^1} f_2^1 = \dot{z}_1^1 / v \end{aligned}$$

$$\begin{aligned} z_4^2 &\triangleq y \\ z_3^2 &\triangleq f_4^2(\underline{\Theta}_3^2) = \dot{y}/v \\ z_2^2 &\triangleq L_{\underline{F}_3^2} f_4^2 = \dot{z}_3^2 / v \\ z_1^2 &\triangleq L_{\underline{F}_2^2} L_{\underline{F}_3^2} f_4^2 = \dot{z}_2^2 / v \\ z_0^2 &\triangleq L_{\underline{F}_1^2} L_{\underline{F}_2^2} L_{\underline{F}_3^2} f_4^2 = \dot{z}_1^2 / v \; , \end{aligned}$$

and for this particular system, we have the resulting multi-input chained form:

$$
\begin{array}{lccc}
\dot z_0^0 = u^0 \qquad & \dot z_0^1 = u^1 \qquad & \dot z_0^2 = u^2 \\
 & \dot z_1^1 = z_0^1 u^0 & \dot z_1^2 = z_0^2 u^0 \\
 & \dot z_2^1 = z_1^1 u^0 & \dot z_2^2 = z_1^2 u^0 \\
 & & \dot z_3^2 = z_2^2 u^0 \\
 & & \dot z_4^2 = z_3^2 u^0 .
\end{array}
$$

5.3. A Parallel-Parking Trajectory. Once the kinematic equations are in multi-input chained form, we can steer the system by one of the algorithms discussed in Section 4. As an illustration, we will discuss how we achieved the parallel parking maneuver shown in the margins of this paper for the five-axle, two-steering system described in Section 5.2. The system parameters we have chosen to be $n = 3$ (three passive axles), $m = 2$ (two steering wheels), and the lengths of the hitches as $L_1^1 = L_2^2 = L_3^2 = 5$, and $L_2^1 = 3$.

We want to steer the system from an initial point of $(x, y) = (0, 20)$ to a final point of $(x, y) = (0, 0)$, where (x, y) are the coordinates of the midpoint of the last axle, and all of the body angles aligned with the horizontal axis in both the initial and final configurations. We will use the polynomial inputs in the chained form equations to plan the trajectory.

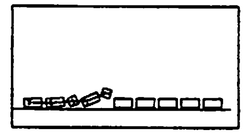
As noted in Section 4.1, polynomial inputs are not immediately suited to this type of trajectory since the time needed to steer the system, computed from equation (27), would come out to be zero and the algorithm would fail. Therefore we have planned the trajectory in two steps, choosing an intermediate point $(x, y) = (30, 10)$. The virtual angles we chose equal to zero in both the initial and final states, and the virtual hitch length we chose as $L_1^2 = 1$. The procedure is first to transform the initial and final states into the chained form coordinates. Using the polynomial inputs methods discussed in Section 4.1, the chained form inputs needed to steer the system are found. These inputs can then be transformed back to the original coordinates to find the virtual inputs, and the real inputs can finally be calculated using the relationship (11).

The simulation was performed on the system in the chained form coordinates, then the inverse coordinate transformation was used on the simulation data to obtain the trajectory in the original coordinates. A movie animation was made of this trajectory; scenes from this movie are shown in the margins of the paper. The path taken by the virtual axle is not shown.

6. Summary

We have presented in this paper a systematic method for convert the kinematic model of a multi-trailer system with n passive trailers and m steerable cars into a multi-input chained form. The advantages of having the system in a multi-input chained form are that many algorithms exist for both steering and stabilizing systems in chained form.

We defined a multi-steering trailer system, and computed its kinematic equations using the constraint that the wheels are allowed to roll but not slip. We then



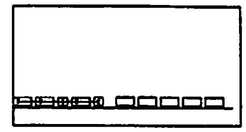
detailed a procedure to transform these equations into a multi-input chain form. The method that we proposed involved adding virtual axles to the system in a form of dynamic state feedback. We motivated the "virtual" extension that we proposed by examining the connections between the kinematic models of a unicycle and a unicycle with one trailer.

Three steering methods for multi-input chained form systems were discussed briefly to show the many different algorithms available. Additionally, various stabilization methods for multi-input chained form systems were mentioned. Finally, two example systems were presented to show the details of the procedure for converting to multi-input chained form. A simulation of a parallel parking maneuver for a five-axle two-steering vehicle was described and clips from the movie animation were shown in the margins of this paper.

Acknowledgements. For the margin movies we thank Anita Flynn, Joe Jones, and Patrick O'Donnell of the MIT AI Laboratory. We would also like to thank Greg Walsh for his help in animating the simulations for the videotape which accompanies this paper.

References

1. J. C. Alexander and J. H. Maddocks. On the maneuvering of vehicles. *SIAM Journal of Applied Mathematics*, 48(1):38–51, 1988.
2. R. W. Brockett. Control theory and singular Riemannian geometry. In *New Directions in Applied Mathematics*, pages 11–27. Springer-Verlag, New York, 1981.
3. R. W. Brockett. Asymptotic stability and feedback stabilization. In R. W. Brockett, R. S. Millman, and H. J. Sussman, editors, *Differential Geometric Control Theory*, pages 181–191. Birkhauser, 1983.
4. L. Bushnell, D. Tilbury, and S. S. Sastry. Extended Goursat normal forms with applications to nonholonomic motion planning. In *Proceedings of the IEEE Conference on Decision and Control*, 1993. To Appear.
5. L. Bushnell, D. Tilbury, and S. S. Sastry. Steering chained form nonholonomic systems using sinusoids: The firetruck example. In *Proceedings of the European Control Conference*, pages 1432–1437, Groningen, The Netherlands, 1993.
6. C. Canudas de Wit. Personal communication.
7. J. Descusse and C. H. Moog. Decoupling with dynamic compensation for strong invertible affine nonlinear systems. *International Journal of Control*, 42:1387–1398, 1985.
8. 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.
9. 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.
10. G. Giralt. Personal communication.
11. A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, 2nd edition, 1989.
12. J-P. Laumond. Nonholonomic motion planning versus controllability via the multibody car system example. Technical Report STAN-CS-90-1345, Stanford University, Department of Computer Science, December 1990.
13. P. Martin and P. Rouchon. Systems without drift and flatness. In *Mathematical Theory of Networks and Systems*, 1993. To appear.
14. S. Monaco and D. Normand-Cyrot. An introduction to motion planning under multirate digital control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1780–1785, Tucson, Arizona, 1992.
15. 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, 1992.
16. R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, May 1993.
17. 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.
18. J. Pomet. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. In *Systems and Control Letters*, volume 18, pages 139–145, 1992.
19. P. Rouchon, M. Fliess, J. Levine, 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.

20. 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.
21. 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, 1993.
22. A. Teel, R. M. Murray, and G. Walsh. Nonholonomic control systems: From steering to stabilization with sinusoids. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1603–1609, Tucson, Arizona, 1992.
23. D. Tilbury and A. Chelouah. Steering a three-input nonholonomic system using multirate controls. In *Proceedings of the European Control Conference*, pages 1428–1431, Groningen, The Netherlands, 1993.
24. D. Tilbury, R. Murray, and S. Sastry. Trajectory generation for the N-trailer problem using Goursat normal form. Technical Report UCB/ERL M93/12, Electronics Research Laboratory, University of California at Berkeley, 1993. To appear in *Proceedings of the IEEE Conference on Decision and Control*, San Antonio, December 1993.
25. G. C. Walsh and L. G. Bushnell. Stabilization of multiple input chained form control systems. Technical Report UCB/ERL M93/43, Electronics Research Laboratory, University of California at Berkeley, 1993. To appear in *Proceedings of the IEEE Conference on Decision and Control*, San Antonio, December 1993.