

## EXPERIMENTAL COMPARISON OF TRAJECTORY TRACKERS FOR A CAR WITH TRAILERS

Francesco Bullo    Richard M. Murray

*Control and Dynamical Systems, California Institute of Technology,*  
bullo@indra.caltech.edu,    <http://avalon.caltech.edu/~nonholomobile>

**Abstract.** In this paper various design techniques are applied to the trajectory tracking problem for a mobile robot with trailers. Using simulations and experiments, we evaluate linear and nonlinear designs on the basis of implementation issues, stability and performance. After a careful design of their gains, the various feedback controllers have very close performance measures. In both the simulations and the experiments, all the controllers show a strong dependence on the knowledge of the reference trajectory. The flatness of the system is exploited in precomputing this quantity.

**Keywords.** tracking, nonlinear control, path planning, feedforward control

### 1. INTRODUCTION

In this paper we evaluate some “modern” solutions to the trajectory tracking problem for a car with trailers. We implement some nonlinear techniques recently proposed in the literature as well as a standard gain scheduling approach based on linear design. Performance, robustness and implementation issues are our criteria in drawing conclusions upon the aspects of each controller.

On the theoretical side, nonlinear techniques have nominal stability, as proven in recent papers by (Samson 1995), (Fliess *et al.* 1995) and (Sampei *et al.* 1995). This is proven with respect to the nominal model and with the assumption of exact measurements and exact input actuation. A more classic and practical approach involves Jacobian linearization and scheduling of linear point stabilizers, designed for example using LQR techniques. These are the techniques that we want to compare. As the numerical and experimental results show, a particularly important aspect of the controller design consists of the implementation of a trajectory generation

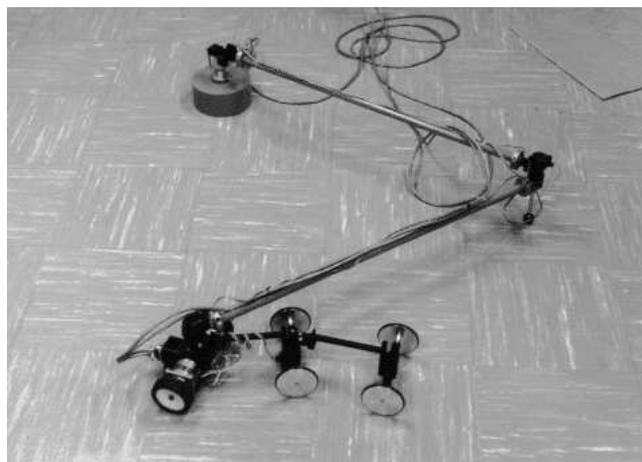


Fig. 1. The nonholomorphic mobile robot.

module. From a theoretical viewpoint, the kinematic model of a car with trailers is a differentially flat system, implying that trajectory generation is a straightforward problem (Fliess *et al.* 1993).

We validate our numerical conclusions on the nonholomorphic mobile robot, depicted in Figure 1. This experimental device has been previously used to evaluate non-

---

\* Research supported in part by NSF Grant CMS-9502224 and AFOSR Grant F49620-95-1-0419.

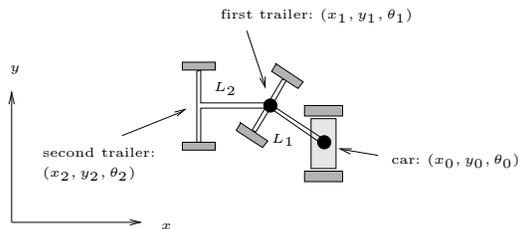


Fig. 2. Nonholonomic mobile robot: notations.

holonomic point stabilizers, see (M’Closkey and Murray 1994) and (M’Closkey and Murray 1995). The main experimental limitations are due to unstable dynamics, limited actuator bandwidth and input saturation.

The article is organized as follows. In Section 2, we describe the experimental device and introduce the model. In Section 3 we design the various controllers, both feedback and feedforward parts. We report on the numerical and experimental comparison in Section 4 and our conclusions are found in Section 5.

## 2. EXPERIMENTAL SETUP AND MODEL

The experimental device is a two-wheeled car with two attached trailers. Two independent stepper motors drive the wheels of the front car: they saturate at a speed of about 0.25 m/s and provide a maximum acceleration of 0.5 m/s<sup>2</sup>. Optical encoders, mounted on a passive two-link manipulator, measure position and orientation of the system. The kinematic map from angle encoders to the position of the rear trailer is straightforward and is not presented here. The experiment is interfaced to an 80386 computer running a real-time control kernel. The servo loop runs at 50 Hertz, all the signals pass through a 5th order Butterworth filter with cut-off frequency of 25 Hertz. The feedback control action is computed at a sample rate of 12.5 Hertz. We refer to (M’Closkey and Murray 1994) for more details.

We denote the position of the  $i$ th trailer as the element  $g_i = (x_i, y_i, \theta_i) \in SE(2)$ , as shown in Figure 2. The shape of the device is given by  $(r_1, r_2) := (\theta_0 - \theta_1, \theta_1 - \theta_2) \in S^1 \times S^1$ . To exploit the flatness of the system, we choose the coordinates of the last trailer  $(x_2, y_2, \theta_2)$  to parametrize the absolute position of the mechanical system in  $SE(2)$ . Let  $v_0$  and  $\omega_0$  denote the linear and angular velocity of the first trailer  $g_0$ . Assuming that the wheels of the robot roll without slipping, a kinematic model is written as

$$\begin{aligned}
 \dot{x}_2 &= \cos(\theta_2)v_2 \\
 \dot{y}_2 &= \sin(\theta_2)v_2 \\
 \dot{\theta}_2 &= (1/L_2) \tan(r_2)v_2 \\
 \dot{r}_2 &= \left( \frac{1}{L_1} \frac{\tan r_1}{\cos r_2} - \frac{1}{L_2} \tan r_2 \right) v_2 \\
 \dot{r}_1 &= \omega_2,
 \end{aligned} \tag{1}$$

where we have performed the feedback transformation  $v_2 = \cos(r_2) \cos(r_1)v_0$  and  $\omega_2 = \omega_0 - \frac{\tan r_1}{L_1 \cos r_2}v_0$ .

An implicit assumption in the kinematic model is that we directly control the velocity of the wheels. Indeed, the stepper motors work in an open loop configuration and there is no guarantee that the commanded speed will actually be achieved. The ultimate justification for this (first principle) model comes from the successful experimental results.

## 3. CONTROL DESIGN

We decompose the controller design into two separate steps. First a reference trajectory is computed and then it is fed forward into a closed loop system, where a feedback action is implemented. We call this design paradigm a two degree of freedom (2 DOF) system design and we compare it with the simpler 1 DOF design, where no feedforward of the full reference trajectory is implemented. The theoretical and practical motivations of this approach are well-explained in (Van Nieuwstadt and Murray 1995).

### 3.1 Feedforward design: trajectory generation

We generate reference trajectories for the car with trailers exploiting the flatness properties of the model (1), see (Fliess *et al.* 1993). In (Fliess *et al.* 1995) flatness is coupled with a “time scaling” procedure to avoid the singularity that exists at zero velocity ( $v_2 = 0$ ). Indeed an arc length parametrization allows for the design of realistic trajectories in which both backward and forward movements are allowed. To test controllers we designed four trajectories: back up straight, back up along a sinusoidal path, parallel park, and load into a dock. These last two are depicted in Figure 3.

### 3.2 Feedback designs

Here we give a brief account of the linear and nonlinear design methodologies implemented. Given the state  $(g, s) = (x_2, y_2, \theta_2, r_2, r_1)$  and the reference  $(g_{\text{ref}}, s_{\text{ref}})$ , we compute the error signal as  $(g_{\text{ref}}^{-1}g, s - s_{\text{ref}})$ ; this definition is useful in some of the designs.

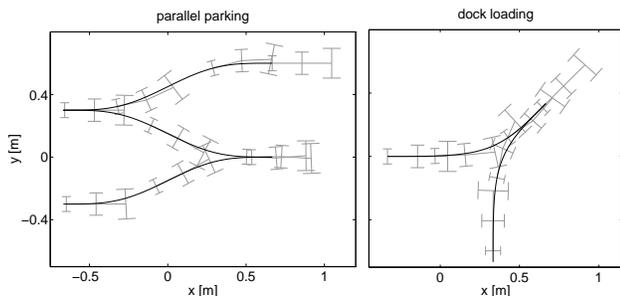


Fig. 3. Reference trajectories: the solid line depicts the  $(x, y)$  path of the rear trailer coordinates. In gray we draw a few sketches of the full state trajectory. The two pictures represent parallel parking and dock loading manouvers, respectively.

**Gain scheduling** We patch together a (global, non-linear) feedback controller by scheduling some linear point stabilizers for the Jacobian linearization of (1). By using the notion of error defined above, the linearization does not depend on  $g_2 = (x_2, y_2, \theta_2)$ , but only on  $(r_2, r_1, v_2)$ . Therefore we schedule only with respect to these latter variables. The total number operating points is 100. We label this controller GS.

**Time scaled feedback linearization** Dynamic feedback linearization is performed on model (1) by adding three integrators before the input  $v_2$ , but a singularity appears for  $v_2 = 0$ . This problem is solved by using the time scaling procedure given in (Fliess *et al.* 1995), to which we refer for further details. In the transformed variables, tracking is easily achieved through a linear feedback; we label this controller TSFL.

**Chained form** Converting the system to chained form is a common feature of many approaches. The chained form of model (1) is  $\dot{z}_1 = u_1$ ,  $\dot{z}_i = u_1 z_{i+1}$  for  $i = 2, 3, 4$  and  $\dot{z}_5 = u_2$ . Assume now that  $\{z_{\text{ref},i}\}, \{u_{\text{ref},j}\}$  is the reference trajectory. If it is a straight line ( $z_{\text{ref},i} = 0$  for  $i = 2, \dots, 5$ ), then convergence is guaranteed by

$$\begin{aligned} u_1 &= -k_1(z_1 - z_{\text{ref},1}) \\ u_2 &= -u_1 \sum_{i=2}^5 k_i z_i. \end{aligned} \quad (2)$$

where the gains  $k_i$  satisfy the Hurwitz stability criterion and where for simplicity of exposition we have assumed  $u_{\text{ref},1} > 0$ . This is the approach in (Sampei *et al.* 1995) and (Samson 1995), to which we refer for more details. To deal with generic trajectories, we implement a modification of the controller in (2). In agreement with the overall philosophy of 2 DOF system design, we employ

$$u_2 = -u_1 \sum_{i=2}^5 k_i (z_i - z_{\text{ref},i}). \quad (3)$$

CRITERIA	CONTROLLERS		
	GS	TSFL	CF
$\ e\ _2$ worst	140	133	141
$\ e\ _2$ average	64.4	58.7	60
$\ e_{\text{group}}\ _2$	57.6	51.9	53.4
$\ e_{\text{shape}}\ _2$	6.75	6.77	6.64
$\ y\ _\infty$	0.146	0.142	0.146
$\ y\ _2$	0.227	0.215	0.221
$ y_{\text{final}} $	0.0187	0.0247	0.027
$\ u\ _2$	0.518	0.512	0.52
$\ u\ _\infty$	0.131	0.165	0.165
$\ u - u_{\text{ref}}\ _2$	0.0667	0.037	0.0328
$\ u - u_{\text{ref}}\ _\infty$	0.115	0.115	0.115

Table 1. Automatic gain design: the controllers' gains are maximized under the constraint of  $\|u - u_{\text{ref}}\|_\infty \leq 0.115$  m/s.

While we provide no convergence proof for this feedback, simulations and experiments show a large improvement over the controller in (2), see Subsection 4.3. We label this controller CF.

### 3.3 Implementation issues

A first judgment on the various design choices is based on the complexity and reliability of the implementations. Gain scheduling controllers are simple to implement and have very little computational need. Instead, due to their extreme ‘‘algebraic’’ complexity, nonlinear controllers require the help of some symbolic package, like Mathematica, and have high computational needs.

A very important implementation aspect for each controller is the design of the gains. Indeed this is a crucial task if any fair comparison is to be attempted. Therefore we relied upon an *automatic gain design* procedure, which despite being somewhat arbitrary, has the advantage of being independent of the controller. The main idea motivating this procedure is the following: assume we assign a set of weights to the state and input variables, also assume that we are given a ball of initial errors and an upper bound to the input magnitude. Then we design LQR controllers using the weights and we use the remaining degree of freedom (the relative weight between state and inputs) to meet the input upper bound while allowing the initial conditions to vary in the error ball. These are the detailed steps:

1. A set of weights is assigned to each state variable through a kinematic analysis of the system. A ratio of 10 to 1 was employed for errors in the  $y$  direction over errors in the  $x$  direction. Call  $Q$  this weighting matrix. Also, a natural weight  $R_0$  is associated to the (original) inputs, by recalling the linear rela-

tionship between the velocities of the front wheels and the linear and angular velocity  $(v_0, \omega_0)$ .

2. We consider a ball of initial errors, described by 0.1 m in the position variables  $(x_2, y_2)$  and  $\pi/20$  in the angular variables  $(\theta_2, r_2, r_1)$ . The input ball is given by  $\|u - u_{\text{ref}}\|_{\infty} \leq 0.115$  m/s, where  $u = (v_2, \omega_2)$  and  $u_{\text{ref}}$  is the nominal input to the reference trajectory described in step 4 below. Note that we are here using the norm of the inputs after the feedback transformation.
3. Assume our system is tracking a straight trajectory, so that  $\theta_2 = r_2 = r_1 = 0$ . Then we linearize the feedback transformation from  $(v_0, \omega_0)$  to  $(v_2, \omega_2)$  to obtain a weight  $R_2$  on the new inputs. Now we compute the GS controller's gains with an LQR design with weights  $(Q_2, \lambda R_2)$ , where  $\lambda$  is for now a free positive parameter. Regarding the nonlinear controllers we follow the same procedure: using the linearization of the state and feedback transformation, we obtain weight matrices on transformed states and inputs. Then we perform an LQR design with  $\lambda$  as free parameter. For the TSFL case, a low weight is assigned to the controller's state.
4. Given the state and input balls computed in step 2, we compute the minimum allowable  $\lambda$ , by running the following set of simulations: track a straight backward trajectory starting from initial points in the ball of initial errors and compute  $\|u - u_{\text{ref}}\|_{\infty}$  as the maximum over all these tests. With an iterative procedure, we set  $\lambda$  (hence the weight on the inputs) to be as low as possible, while keeping  $\|u - u_{\text{ref}}\|_{\infty} \leq 0.115$  m/s. The total number of simulation runs for each controller is 52; each initial condition is simulated for 20 seconds.

We report in Table 1 the results of the final simulation. In the following section we describe all the quantities included in the table: here only  $\|u - u_{\text{ref}}\|_{\infty}$  is of interest.

## 4. COMPARISON

Here we present some quantitative results obtained simulating the controllers in the closed loop system. We formulate some claims on the behavior of the feedback designs and we validate them on the experimental device.

### 4.1 Performance in simulations

We evaluate the feedback controllers on the four trajectories by starting them from various initial conditions: in particular, we allow the initial error to vary in the set described by  $\|(x_2, y_2)\|_2 \leq 0.2$  m, together with  $|\theta_2|, |r_2|$  and  $|r_1| \leq 0.1\pi$ . There are a total of 64 different initial

CRITERIA	CONTROLLERS		
	GS	TSFL	CF
$\ e\ _2$ worst	134	119	145
$\ e\ _2$ average	44.6	37.7	45.2
$\ e_{\text{group}}\ _2$	39.7	33.1	40.6
$\ e_{\text{shape}}\ _2$	4.94	4.55	4.62
$\ y\ _{\infty}$	0.269	0.28	0.283
$\ y\ _2$	0.297	0.267	0.309
$ y_{\text{final}} $	0.0353	0.0395	0.0622
$\ u\ _2$	0.548	0.598	0.545
$\ u\ _{\infty}$	0.176	0.246	0.263
$\ u - u_{\text{ref}}\ _2$	0.0842	0.0761	0.0485
$\ u - u_{\text{ref}}\ _{\infty}$	0.214	0.22	0.224

Table 2. Comparison of feedback controllers. Simulations are performed on 4 trajectories and with 64 different initial conditions.

CRITERIA	CONTROLLERS		
	GS	TSFL	CF*
$\ e\ _2$ worst	453	318	201
$\ e\ _2$ average	143	128	98.4
$\ e_{\text{group}}\ _2$	127	111	85.3
$\ e_{\text{shape}}\ _2$	15.2	16.9	13.1
$\ y\ _{\infty}$	0.334	0.292	0.289
$\ y\ _2$	0.516	0.451	0.407
$ y_{\text{final}} $	0.179	0.182	0.0964
$\ u\ _2$	0.613	0.566	0.521
$\ u\ _{\infty}$	0.176	0.269	0.24
$\ u - u_{\text{ref}}\ _2$	0.133	0.0751	0.0443
$\ u - u_{\text{ref}}\ _{\infty}$	0.202	0.225	0.202

Table 3. Same simulation set as in the previous table, but setting the reference values of  $(r_2, r_1)$  to zero. \*: the results for the CF controller are restricted to only three out of four trajectories, since the closed loop system was unstable on the sinusoidal manouver.

conditions for each trajectory. The model used in the simulations includes a saturation function on the output of the controllers: this block reproduces the saturation of the stepper motors in velocity at 0.25 m/s and in acceleration at 0.5 m/s<sup>2</sup>. Note that the size of the error ball is now larger than the one in the automatic gain design. Indeed all controllers hit the saturation limits in one run or another.

Out of this large amount of data we summarize some quantitative results in Table 2. We include the following error measures: for each trajectory we compute the 2-norm of the error state (as described in Subsection 3.2) and we normalize it with respect to the size of the initial error. In the table, we report the worst case, the average and how the average decomposes into the group and

shape variables. The next three values report on the behavior of the error in the  $y_2$  direction. The last four values describe the size of the input  $u = (v_2, \omega_2)$ .

In Table 3, we report on a second set of simulations, in which the controllers have only a partial knowledge of the reference trajectory. In particular, the reference value for the shape variables  $(r_2, r_1)$  is set to zero. From these numerical results, we conclude the following:

1. The various designs perform very similarly in the size of both error and input signals; this is due to the careful choice of gains. The time scaled feedback linearization controller TSFL is consistently better than the others, even though no great advantage is achieved over the linear design.
2. All the controller designs depend very strongly on the full knowledge of the reference trajectory. This is proven by noting the performance decay from Table 2 to Table 3.

#### 4.2 Stability regions in simulation

An estimate of the controllers' stability region is computed by running the following simulation set: track a straight backward trajectory starting with initial errors in the set described by  $x_2 = y_2 = \theta_2 = 0$ ,  $|r_2|$  and  $|r_1| \leq 0.4\pi$ . The results are depicted in Figure 4: in a  $(r_1, r_2)$  plane each point corresponds to a simulation; given the symmetry of the results, only a half plane is depicted. We draw an "x" when the car with trailer jack-knives and a "+" when the error in  $y_2$  reaches the value of 0.1 m (but the state still converges in the end). A light gray "o" corresponds to a successful run. The simulations were performed with the saturation block described before, where the acceleration limit was set to  $0.25 \text{ m/s}^2$  (half the experimental value), to emphasize the differences between controllers.

Note the different region of stability for the various controllers. The linear design seems to have a wider stability region than the nonlinear ones. In particular, for both nonlinear controllers, there exists a region of initial errors, characterized by  $r_1$  and  $r_2$  having opposite sign, for which convergence is achieved only after a fairly large transient response.

#### 4.3 Validation

To validate the qualitative results obtained so far, we run some appropriate experiments on the nonholomorphic mobile robot.

**Hard initial conditions for nonlinear designs** The stability analysis of the previous section is confirmed by the fact that from certain initial conditions the

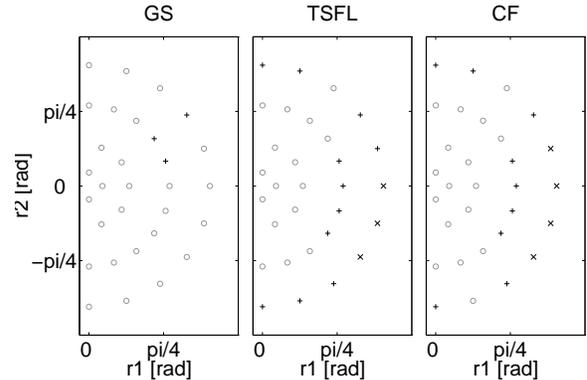


Fig. 4. Region of stability of the controllers tracking a backward trajectory. Only half of the  $(r_1, r_2)$  plane is depicted because of the symmetry of the results. R "x" means that along the simulation either  $r_1$  or  $r_2$  became larger than  $0.45\pi$ , "+" means that  $y_2$  became larger than 0.1 m and a gray "o" means the run was successful.

scheduled design behaves better than both nonlinear controllers. When the trailers' position is very skewed, the "best" thing to do is go forward to straighten the system's shape, before backing up to track the trajectory. This is what the GS controller does in the first picture of Figure 5. The initial skewed configuration of the system is drawn in black. In gray one can notice the different behavior of the nonlinear designs, which only allow for negative velocities.

#### Two degree of freedom controller design

To illustrate the vast performance improvement due to the feedforward module, we have tested the various controllers with and without feedforward. In Figure 6 the GS and CF controllers are tested on the sinusoidal trajectory and on the parallel parking manouever. In the simulations without feedforward, only the reference values of  $x_2$  and  $y_2$  are known. As foreseen, the availability of a full state trajectory makes the difference: in the left column the controllers converge, while in the right one they do not.

## 5. DISCUSSION

Here we formalize the results obtained in the previous sections and draw some conclusions.

1. The first principles model, described in Section 2, is precise enough for design purposes. This holds true in the closed loop, as long as due care is given to the actuators' bandwidth, input saturation and other experimental limitations.
2. The controllers' behavior depends very strongly on their gains. Hence, a fair comparison of the various design methods is impossible unless we agree upon

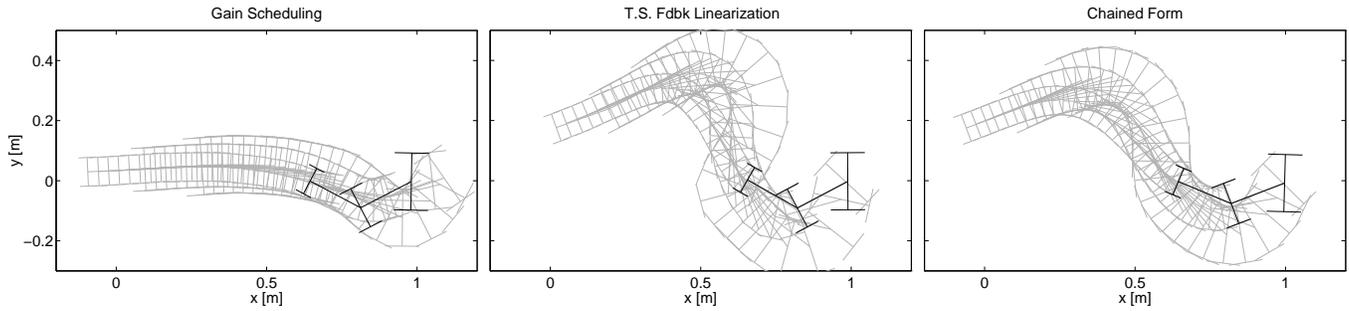


Fig. 5. Experimental comparison of controllers starting from an initial position with very skewed trailers (and tracking a straight backward path). From left to right: Gain Scheduling, Time Scaling Feedback Linearization, Chained Form: three light gray trailer systems are drawn each two seconds of experiment.

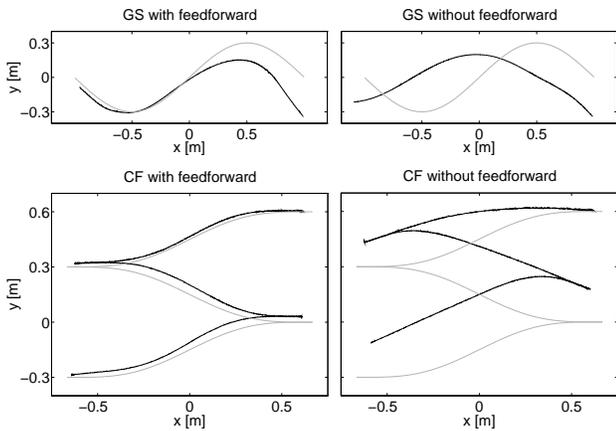


Fig. 6. Experimental comparison of controllers with and without feedforward. We depict the  $(x_2, y_2)$  path of the experimental device in black and the reference trajectory in gray. In the left column, GS and CF with feedforward, in the right column without. In the first row, the simulations are started with an initial error  $y_2 = -0.3$  m. In the second row, no initial error is introduced.

a way of designing weights for each technique. This is the purpose of the “automatic gain design” algorithm described in Subsection 3.3. Much more theoretical work is needed in this area.

3. The feedback designs show a similar average and worst case performance over all trajectories. In the simulations, the time scaled feedback linearization controller has the best performance and the gain scheduling design has a greater stability region.
4. Two degree of freedom controller design guarantees superior performances over one degree of freedom. In other words, the efficiency of each controller showed a strong dependence on the availability of a precomputed full state reference trajectory.

This study has outlined the need for more sophisticated nonlinear techniques and for a deeper understanding

of the ones available. Indeed the trajectory generation module is critical for linear and nonlinear controllers and it is here that the differential flatness of the model plays a decisive role.

*Acknowledgments:* The authors would like to thank Robert M’Closkey, Sudipto Sur and Michiel van Nieuwstadt for their help and support.

## 6. REFERENCES

- Fliess, M., J. Lévine, P. Martin and P. Rouchon (1993). Flatness, motion planning and trailer systems. In: *CDC*. San Antonio, Texas. pp. 2700–2705.
- Fliess, M., J. Lévine, P. Martin and P. Rouchon (1995). Design of trajectory stabilizing feedback for driftless flat systems. In: *ECC*. Rome. pp. 1882–1887.
- M’Closkey, R. T. and R. M. Murray (1994). Experiments in exponential stabilization of a mobile robot towing a trailer. In: *ACC*. Baltimore, Maryland. pp. 988–993.
- M’Closkey, R. T. and R. M. Murray (1995). Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. California Institute of Technology. Technical Report CIT/CDS 95–012. Submitted, *T. on Automatic Control*. Available electronically via <http://avalon.caltech.edu/cds>.
- Van Nieuwstadt, M. and R.M. Murray (1995). Approximate trajectory generation for differentially flat systems with zero dynamics. In: *CDC*. New Orleans, Louisiana. pp. 4224–4230.
- Sampei, M., T. Tamura, T. Kobayashi and N. Shibui (1995). Arbitrary path tracking control of articulated vehicles using nonlinear control theory. *T. Control Systems Technology* **3**(1), 125–131.
- Samson, C. (1995). Control of chained systems. Application to path following and time-varying point-stabilization of mobile robots. *T. Automatic Control* **40**(1), 64–77.