# Stabilization of Trajectories for Systems with Nonholonomic Constraints

G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J. P. Laumond

*Abstract*— **A new technique for stabilizing nonholonomic systems to trajectories is presented. It is well known (see [2]) that such systems cannot be stabilized to a point using smooth static-state feedback. In this note, we suggest the use of control laws for stabilizing a system about a trajectory, instead of a point. Given a nonlinear system and a desired (nominal) feasible trajectory, the note gives an explicit control law which will locally exponentially stabilize the system to the desired trajectory. The theory is applied to several examples, including a car-like robot.**

## I. Introduction

There has been a great deal of recent research on the problem of stabilizing a system with nonholonomic (nonintegrable) constraints on its velocities [1], [5], [12]. Of course, from Brockett's necessary conditions for stability [3], one may demonstrate that systems with nonintegrable velocity constraints cannot be stabilized to a point with smooth static-state feedback. Given this result, researchers have offered both nonsmooth feedback laws [6] and time-varying feedback laws [12] for stabilizing simple mobile robots to a point. However, it is fair to say that these approaches are not yet fully general.

We would like to take advantage of the large body of work on generating open-loop paths for nonholonomic systems (see for example [10], [11]) and make our controller robust to modeling errors and initial condition position errors by stabilizing about *trajectories* instead of points. Given a feasible trajectory for the system generated by an open-loop path planner, we can compute the linearization of the system about this nominal trajectory. If the linear time-varying system thus obtained is uniformly completely controllable in a certain sense (to be made explicit in Section II), we define a linear time-varying feedback law which will locally stabilize the system about this nominal trajectory.

Thus, the problem this note solves is: given a nonholonomic system, a feasible desired trajectory to follow, a known clearance between obstacles, and a measure of accuracy of the sensors, find a control law which will stabilize the system to this path, avoiding the obstacles robustly in the face of disturbances.

In Section II, we present our control law and show it to be exponentially convergent. In the following sections, we apply this control law to various nonholonomic systems, including the system generated by the so-called Heisenberg control algebra, a wheeled mobile robot called Hilare, and a front wheel drive car.

In the examples, we focus on mobile robots with an objective of creating a composite controller that will: first, have off-line computation of a trajectory which avoids the obstacles [9]; second, apply the control law given here to stabilize the system to the open-loop collision-free trajectory; third, while executing, use sensors to detect possible collisions due to poor *a priori* information. In this case, new information can be used to update the model of the environment and restart the process. Such a controller would be able to reject many types of disturbances including noise in the sensors, initial condition errors, and errors introduced along the trajectory.

Manuscript received March 14, 1992; revised November 17, 1992. This work was supported in part by the NSF under grants ECS-87-1998 and IRI 90-14490, and by the CNRS. G. Walsh was supported in part by an NSF fellowship.

G. Walsh, D. Tilbury, and S. Sastry are with the Electronics Research Laboratory, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720.

R. Murray is with the Department of Mechanical Engineering, California Institute of Technology, Pasadena, CA 91125.

J. P. Laumond is with the Laboratoire d'Automatique et d'Analyse des Systèmes, Toulouse, France.

IEEE Log Number 9213574.

## II. An Exponentially Stabilizing Control Law

We consider a system

$$\dot{x} = f(x, u) \tag{1}$$

with $x \in \mathbb{R}^n$ as the state of the system and $u \in \mathbb{R}^p$ as the input. The function $f(x, u)$ will be $C^2$ with regards to $x$ and $u$. In addition to the system, we will be given the nominal trajectory $x^0(t)$ and the corresponding nominal input $u^0(t)$.

*Remarks:* We shall focus on systems where $f(x, 0)$ is identically zero. Systems like this are called "drift-free" and encompass most of the models used in the literature. However, the method we present here is general enough to include systems like (1) which have nonzero drift terms. Thus the proofs will include the drift terms, although the worked examples are all drift-free.

Inspired by the result on linear time-varying control systems found in [4], we have chosen the following control law.

*Proposition 1 (A Stabilizing Control Law):* Given a system of the form (1), a desired trajectory $x^0(\cdot)$, with corresponding nominal input $u^0(\cdot)$, both bounded, define

$$A(t) := \frac{\partial f}{\partial x}(x^0(t), u^0(t))$$
$$B(t) := \frac{\partial f}{\partial u}(x^0(t), u^0(t)).$$

Let $\Phi(t, t_0) \in \mathbb{R}^{n \times n} \times \mathbb{R}$ be the state transition matrix of $A(t)$, that is $\Phi(t, t_0)$ satisfies $\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0)$ with $\Phi(t_0, t_0) = I$. Further, for a given $\alpha > 0$, define

$$H_c(t_0, t) = \int_{t_0}^{t} \exp(6\alpha(t_0 - \tau))\Phi(t_0, \tau)B(\tau)B(\tau)^T\Phi(t_0, \tau)^T\, d\tau.$$

If there exists a $\delta$ such that $H_c(t, t + \delta)$ is bounded away from singularity[1] uniformly in $t$, then define $P_c(t)$ as follows:

$$P_c(t) := H_c^{-1}(t, t + \delta).$$

Now, if there exist two numbers $p_c^m$, $p_c^M$ such that

$$0 < p_c^m I < P_c(t) < p_m^M I \qquad \forall t \in \mathbb{R}_+.$$

Then, for any function $\gamma(t): \mathbb{R}_+ \to [\frac{1}{2}, \infty)$, continuous and bounded, the linear time-varying feedback law:

$$u = u^0 - \gamma(t)B^T(t)P_c(t)(x - x^0) \tag{2}$$

locally, uniformly, exponentially stabilizes the system (1) to the desired trajectory $x^0(t)$ at a rate of at least $2\alpha p_c^m (p_c^M)^{-1} > 0$, that is for all $t > t_0$, as long as $\|x(t_0) - x^0(t_0)\| < \epsilon$ for some $\epsilon > 0$

$$\|x(t) - x^0(t)\| \le \|x(t_0) - x^0(t_0)\| \cdot \sqrt{\frac{p_c^M}{p_c^m}} \exp(-2\alpha p_c^m (p_c^M)^{-1}(t - t_0)).$$

[1] If the linear time-varying system is uniformly completely controllable over intervals of length $\delta > 0$ then $H_c(t, t + \delta)$ is uniformly invertible.

*Proof:* First, define the error signal $e$ and error input $v$ as

$$e = x - x^0$$
$$v = u - u^0(t).$$

We solve for the dynamics of these error signals using the Taylor series expansions

$$\dot{e} = f(x^0 + e, u^0 + v) - f(x^0, u^0)$$
$$= A(t)e + B(t)v + o(e, v, t).$$

Here all terms with dependencies on $x^0$, $u^0$ are rewritten as functions of time. We have defined $o(e, v, t)$ to be the higher order terms

$$o(e, v, t) = f(x^0 + e, u^0 + v) - f(x^0, u^0) - A(t)e - B(t)v.$$

Note that since our control law defines the error input $v = -\gamma(t)B^T(t)P_c(t)e$, we may rewrite $o(e, v, t)$ so that it depends only on $e$ and $t$; call this $\hat{o}(e, t)$. Note that $x^0(t)$, $u^0(t)$ bounded implies $B(t)$ is bounded. Further since $\gamma(t)$ and $P_c(t)$ are bounded we have

$$\|v\| = \left\|\gamma(t)B^T(t)P_c(t)e\right\| \le K\|e\|$$

for some $K < \infty$. From this it follows that:

$$\lim_{\|e\|\to 0} \sup_{t\ge 0} \frac{\|\hat{o}(e, t)\|}{\|e\|} = 0. \tag{3}$$

That is, $\hat{o}(e, t)$ is uniformly higher order in $e$.

Thus

$$\dot{e} = A(t)e + B(t)v + o(e, v, t)$$
$$\dot{e} = \tilde{A}(t)e + \hat{o}(e, t) \tag{4}$$

with

$$\tilde{A}(t) = A(t) - \gamma(t)B(t)B^T(t)P_c(t).$$

As in [4], we choose a Lyapunov function

$$V(e, t) = e^T P_c(t)e \tag{5}$$

and calculate its time derivative along trajectories of the system (4). It may be verified that:

$$\dot{P}_c(t) = -6\alpha P_c(t) - P_c(t)A(t) - A^T(t)P_c(t)$$
$$+ P_c(t)B(t)B^T(t)P_c(t) - \exp(-6\alpha\delta)P_c(t)$$
$$\cdot [\Phi(t, t+\delta)B(t+\delta)B^T(t+\delta)$$
$$\cdot \Phi^T(t, t+\delta)]P_c(t).$$

Thus the time derivative of the Lyapunov function is:

$$\dot{V}(e, t) = -e^T[6\alpha P_c(t) + (2\gamma(t) - 1)P_c(t)B(t)B^T(t)P_c(t)]e$$
$$- e^T \exp(-6\alpha\delta)P_c(t)\Phi(t, t+\delta)$$
$$\cdot B(t+\delta)B^T(t+\delta)\Phi^T(t, t+\delta)P_c(t)e$$
$$+ 2e^T P_c(t)\hat{o}(e, t) \tag{6}$$

Note that if $\gamma(t) \ge \frac{1}{2}$, $\forall t$, then the first two terms in (6) are less than or equal to $-6\alpha p_c^m\|e\|^2$. Now by (3), there exists $\epsilon > 0$ such that

$$\|\hat{o}(e, t)\| \le \alpha p_c^m (p_c^M)^{-1}\|e\|. \qquad \forall e \text{ such that } \|e\| \le \epsilon.$$

This implies that

$$\left|2e^T P_c(t)\hat{o}(e, t)\right| \le 2\alpha p_c^m\|e\|^2. \qquad \forall \|e\| \le \epsilon.$$

Proceeding as in [4], we see

$$\dot{V}(e, t) \le -4\alpha p_c^m\|e\|^2 \tag{7}$$

Further we may state that:

$$\dot{V}(e, t) \le -4\alpha p_c^m (p_c^M)^{-1} V(e, t) \tag{8}$$

so that

$$V(e, t) \le V(e_0, t_0)\exp(-4\alpha p_c^m (p_c^M)^{-1}(t - t_0))$$
$$\|e(t)\| \le \|e(t_0)\|\sqrt{\frac{p_c^M}{p_c^m}}\exp(-2\alpha p_c^m (p_c^M)^{-1}(t - t_0)) \tag{9}$$

which gives us the specified convergence rate for the error. □

*Convergence Rate:* This convergence rate is independent of $p_c^m$ and $p_c^M$. To demonstrate this, use an exponential weighting function:

$$z = e y(t)$$

with

$$y(t) = \exp(\alpha t)y(0)$$

$z$ satisfies

$$\dot{z} = (\tilde{A}(t) + \alpha I)z + \hat{o}(e, t)y.$$

We will pick the same Lyapunov equation and calculate its derivative, as before

$$V(z, t) = z^T P_c(t)z$$
$$\dot{V}(z, t) \le -4\alpha p_c^m\|z\|^2 + 2z^T P_c(t)\hat{o}(e, t)y.$$

Given the exponential convergence of $e$ when it starts sufficiently close to the origin, we may say that after some time $\tau$ that

$$\|\hat{o}(e, t)\| \le \frac{p_c^m}{p_c^M}\alpha\|e\|.$$

Thus the last factor may be bounded as follows:

$$\left|2z^T P_c(t)\hat{o}(e, t)y\right| \le 2\alpha p_c^m\|z\|^2.$$

Thus we may write, as before, that

$$\dot{V}(z, t) \le -2\alpha p_c^m\|z\|^2.$$

Following (8) and (9), we will obtain the same convergence rate, $\alpha p_c^m (p_c^M)^{-1}$. However, we may note that

$$\|z\| = \exp(\alpha(t - t_0))|y_0|\,\|e\|.$$

Thus, if $z$ is exponentially convergent at a rate $\alpha p_c^m (p_c^M)^{-1}$ after some time $\tau$, then $e$ is exponentially convergent at a rate $\alpha p_c^m (p_c^M)^{-1} + \alpha > \alpha$ after some time $\tau$, thus for a sufficiently large $k$ we may state that

$$\|z\| \le k\exp(-\alpha(t - t_0))\|z_0\|.$$

*Information Considerations:* For some regulator applications, it is desirable not to need information on the future trajectory of the system. To deal with this concern, define $P_r(t)$, similar to $P_c(t)$, again, assuming the inverse in the formula exists

$$P_r(t) \equiv (H_c(t, t - \delta))^{-1}.$$

Notice that this matrix is dependent on past values of the trajectory and not on future values. As before, if there exists two numbers $p_r^m$ and $p_r^M$ such that

$$0 < p_r^m I < P_r(t) < p_r^M I \qquad \forall t \in \mathbb{R}_+$$

then for any $\gamma(t)$: $\mathbb{R}_+ \to [\frac{1}{2}, \infty)$, continuous and bounded, the linear time-varying feedback law:

$$u = u^0 - \gamma(t)B(t)^T P_r(t)(x - x^0)$$

locally uniformly exponentially stabilizes the system (1) at a rate greater than $\alpha$.

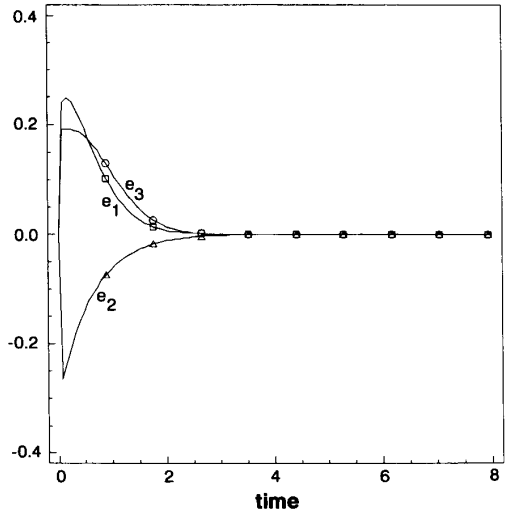


Fig. 1. Plot of errors $e$ versus time. The errors all quickly converge to zero for this path.

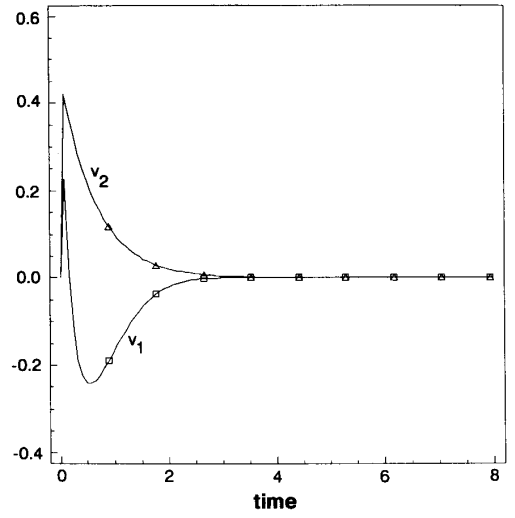


Fig. 2. Graph of the error inputs $v$ versus time. Note how all inputs are bounded and smooth.

The proof is similar to that of the last control law. It is useful to note that:

$$\begin{aligned}\dot{P}_r(t) = &-6\alpha P_r(t) - P_r(t)A(t) - A^T(t)P_r(t)\\ &- P_r(t)B(t)B^T(t)P_r(t)\\ &- \exp(6\alpha\delta)P_r(t)\Phi(t, t-\delta)B(t-\delta)B^T(t-\delta)\\ &\cdot \Phi^T(t, t-\delta)P_r(t).\end{aligned}$$

We have applied this control law to three example systems. The first is chosen because its simple structure allows for the explicit computation of the control laws. The second is the Hilare-like mobile robot, without drift, and the third example is a front wheel drive car.

## III. Example: The Heisenberg Control Algebra

Here we will consider one of the simplest nonholonomic systems: the system whose control Lie algebra is the so-called Heisenberg algebra with two generators [2]. The differential equations are as follows:

$$\begin{aligned}\dot{x}_1 &= u_1\\ \dot{x}_2 &= u_2\\ \dot{x}_3 &= x_2 u_1.\end{aligned} \tag{10}$$

This system's straightforward structure allows us to compute the control laws in closed form. We will investigate two trajectories for this sytem, a "trivial" trajectory which is just a point, and a straight line. Because of the simple structure of this system, much of the control law can be found without reference to the specific form of the desired trajectory.

For the "trivial" trajectory, the system stays fixed at a given point for all time. Note that stabilizing to this trajectory is equivalent to finding a point stabilization feedback law. The trajectory is degenerate in the sense that both nominal inputs are zero. We choose our desired point $x^0(t)$ to be the origin, $(0, 0, 0)$.

It may be shown that in this case, $H_c(t_0, t)$ has the following form (for $\alpha = 1$):

$$H_c(t_0, t) = \begin{bmatrix} 1 - e^{(t_0 - t)} & 0 & 0 \\ 0 & 1 - e^{t_0 - t} & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Since the matrix $H_c(t, t+\delta)$ is not invertible for any choice of $\delta$, we cannot find the matrix $P_c(t) = H_c^{-1}(t, t+\delta)$ which is used in the definition of the control law, and therefore the method presented in this note cannot be used to stabilize the system (10) to a point.

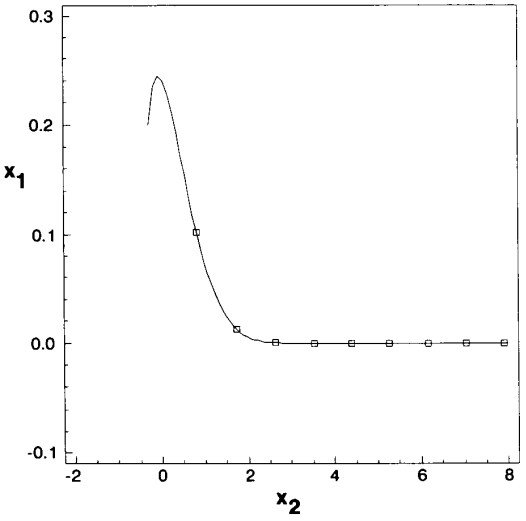


Fig. 3. This $x_1$, $x_2$ phase plot shows the actual trajectory, projected onto the $(x_1, x_1)$ plane. The desired trajectory is a straight line along the $x_2$ axis.

The second sample trajectory corresponds to a straight line in state-space described by $x^0(t) = (0, t, 0)$, with nominal input $u^0(t) = (0, 1)$. Since the matrix $H_c$ is invertible, our strategy will work. In fact, the determinant of $H_c$ is independent of time, as can be seen by the following formula (where $\alpha = 1$)

$$\det(H_c(t, t+\delta)) = 1 - e^{3\delta} + (3 + \delta^2)(e^{-2\delta} - e^{-\delta}).$$

We can choose any value $\delta$ for which the previous expression is nonzero. In our simulation, $\delta = 1$ was chosen. (Note: nearly identical simulation results are obtained for the trajectory is given by $u^0(t) = (1, 0)$).

The initial error for this simulation was $(0.2, -0.3, 0.2)$. The simulation was run for 8 sec, and the results are given in Figs. 1–3, showing the error coordinates $e(t)$ and the error inputs $v(t)$ versus time.

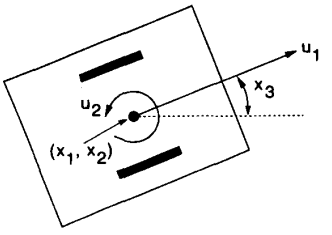


Fig. 4. Model of the mobile robot Hilare.

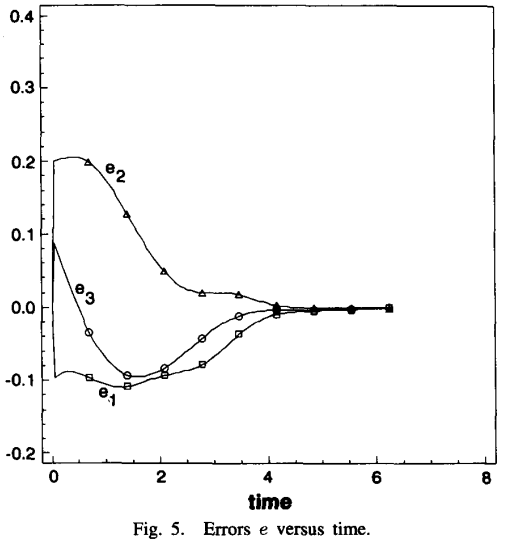


Fig. 5. Errors $e$ versus time.

## IV. Trajectory Stabilization for a Simple Nonholonomic Mobile Robot: Hilare

Hilare is a wheeled mobile robot (see Fig. 4) built at LAAS, Laboratoire d'Automatique et d'Analyse des Systèmes, located in Toulouse, France [8]. This robot has two parallel wheels which can be controlled independently. By commanding the same velocity to both wheels, the robot moves in a straight line. By commanding velocities with the same magnitude but opposite directions, the robot pivots about the vertical axis. Although the actual input is the acceleration, we perform only a kinematic analysis and assuming that we can control the velocity of each wheel. If these wheel velocities are the inputs, one may model Hilare as follows:

$$\begin{aligned} \dot{x}_1 &= \cos(x_3)u_1 \\ \dot{x}_2 &= \sin(x_3)u_1 \\ \dot{x}_3 &= u_2. \end{aligned} \tag{11}$$

Here the coordinates $(x_1, x_2)$ represent the position of the robot in the plane, and $x_3$ is its orientation (see Fig. 4).

Again, one would hope the system's straightforward structure allows the control laws to be computed in closed form. Indeed, a straightforward transformation of both the coordinates and the inputs would convert this system to a form like the last example which is nilpotent. It is not clear whether we can depend on finding in general such transformations, so the control law will be applied only to the untransformed system.

In this form and for the nominal trajectories $x^0$ that we have chosen to simulate, the integrals do not have a closed form. Thus, we cannot directly compute the control law so we must compute the matrix $P_c$

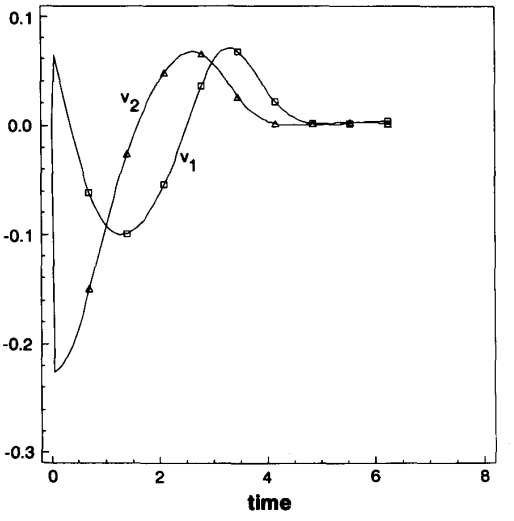


Fig. 6. Error inputs $v$ versus time.

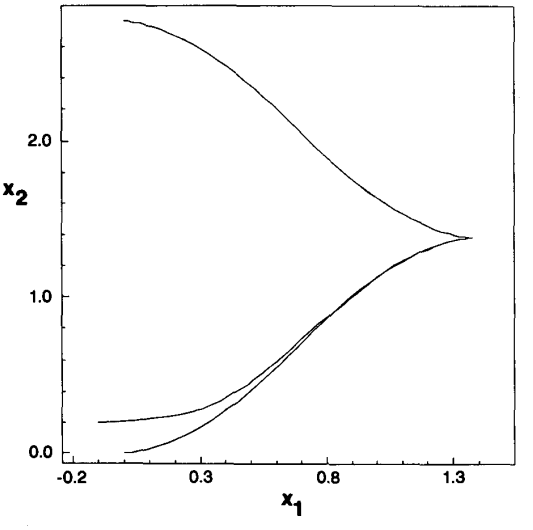


Fig. 7. Phase plot showing the nominal and actual trajectories projected onto the $(x_1, x_2)$ plane. The desired trajectory starts at $(0, 0)$ whereas the actual trajectory has an initial offset of $(-0.1, 0.2)$.

and the control law numerically. In doing so the following identity is useful:

$$\dot{\Phi}(t, t+\delta) = A(t)\Phi(t, t+\delta) - \Phi(t, t+\delta)A^T(t+\delta).$$

The first nominal trajectory for this system (Hilare) is generated by the inputs $u_1 = \sin(t)$, $u_2 = \cos(t)$. We set $\alpha = 0.1$, $\delta = 1.0$. After one cycle this input steers the system in the direction given by the Lie bracket of the two input vector fields.[2] The initial condition was chosen to be $(-0.1, 0.2, 0.1)$, and the simulation was run for $2\pi$ secs. See Figs. 5–7 for the simulation results in this case.

The second nominal trajectory to which we have applied our stabilization procedure is a circular path. This choice was inspired by the work of Reeds and Shepp [11], who showed that time-optimal

[2]One may write this model as $\dot{x} = g_1(x)u_1 + g_2(x)u_2$, so the Lie bracket of the two input vector fields would be $[g_1, g_2]$.

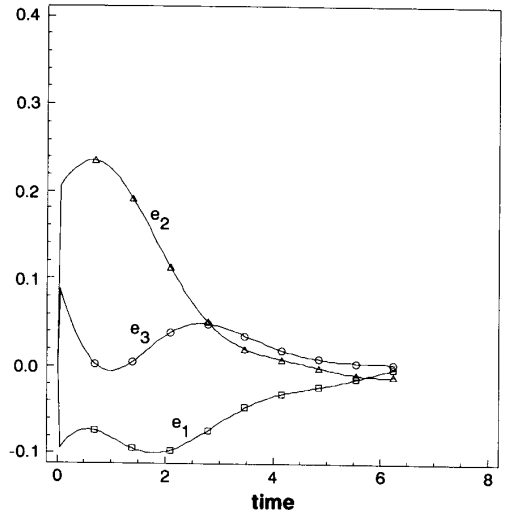


Fig. 8. The errors $e$ versus time.

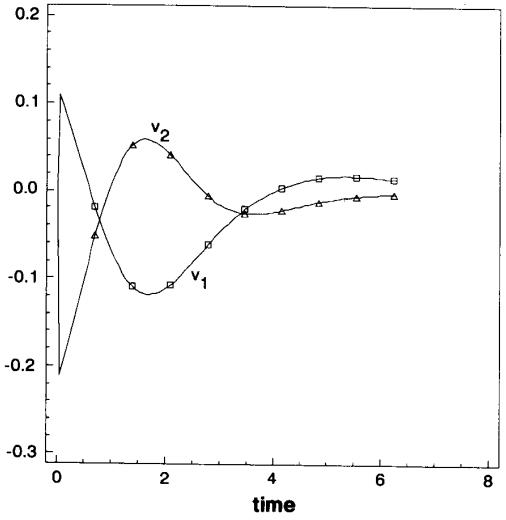


Fig. 9. Error inputs $v$ versus time.

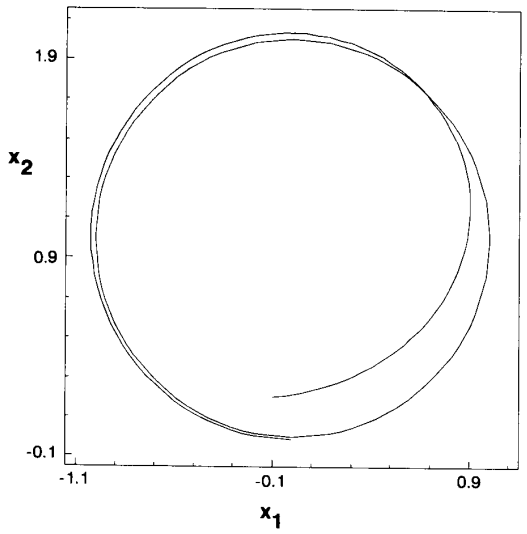


Fig. 10. Phase plot showing the nominal and actual trajectories, projected onto the $(x_1, x_2)$ plane. The desired trajectory is a circle.

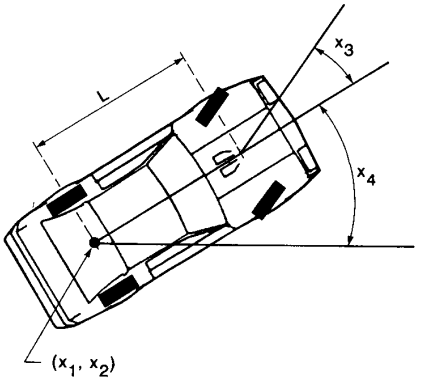


Fig. 11. The front wheel drive car.

paths for Hilare-like robots with actuator limits consist of straight-line segments and arcs of circles.

The nominal input for this trajectory is $u^0 = (1, 1)$. We set $\alpha = 0.1$, $\delta = 1.0$ as before. We again choose an initial condition error of $(-0.1, 0.2, 0.1)$, and run the simulation for $2\pi$ sec. See Figs. 8–10 for the simulation results in this case.

Although we have used the same values of $\alpha$, $\delta$ and initial error as in the previous example, the convergence seems less rapid, indicating that the convergence rate depends on the chosen trajectory. However, the convergence rate is also a function of $\alpha$, which we are free to choose. If we needed faster convergence, we could simply choose a larger $\alpha$.

## V. Trajectory Stabilization for a Front Wheel Drive Car

Consider a front-wheel drive car as shown in Fig. 11. Assuming velocities as inputs, the kinematic equations of the car (derived in [10]) are:

$$\begin{aligned} \dot{x}_1 &= \cos(x_3)\cos(x_4)u_1 \\ \dot{x}_2 &= \cos(x_3)\sin(x_4)u_1 \\ \dot{x}_3 &= u_2 \\ \dot{x}_4 &= \frac{1}{L}\sin(x_3)u_1 \end{aligned} \tag{12}$$

where $(x_1, x_2)$ is the position of the car in the plane, $x_3$ is the angle of the front wheels with respect to the car (or the steering wheel angle), $x_4$ is the orientation of the car with respect to some reference frame, and the constant $L$ is the length of the wheel base. This system is controllable [10], although Lie brackets of order 2 must be taken to demonstrate this.

For the example trajectory we chose the nominal input $u^0 = (\sin(t), \cos(2t))$, roughly corresponding to a parallel-parking maneuver (see Fig. 14). This choice was inspired by [10] though in form it is a simpler input than their algorithm would have generated. Again, we chose $\alpha = 0.1$, $\delta = 1.0$. After one period ($T = 2\pi$), this input steers the system in the direction given by the Lie bracket of order two of the two input vector fields (i.e., one may write the

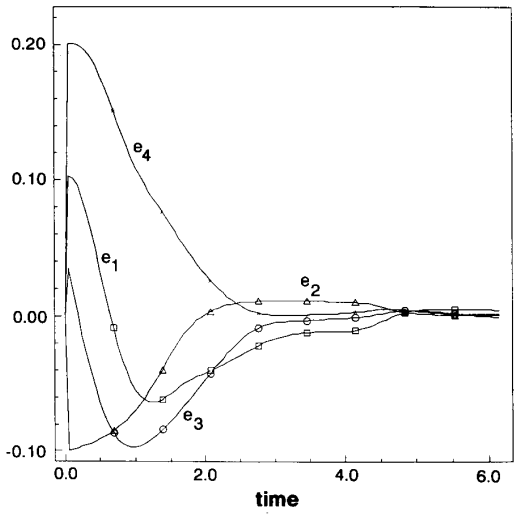


Fig. 12. Plot of errors $e$ versus time.

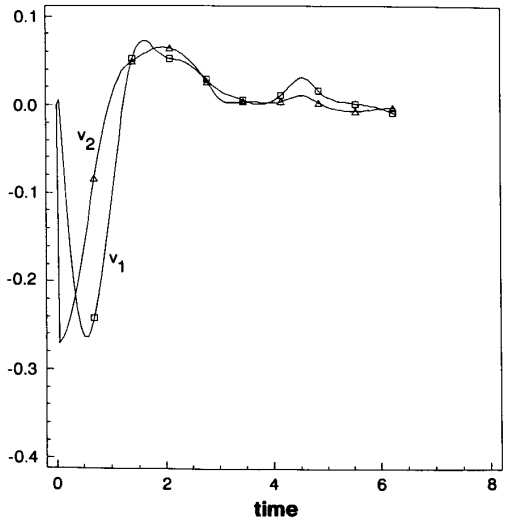


Fig. 13. The control inputs $v$ versus time. Note that they are bounded, smooth, and go to zero.

system as $\dot{x} = g_1(x)u_1 + g_2(x)u_2$, thus the Lie bracket of order 2 would be $[g_1, [g_1, g_2]]$). Because the equations for this example are not simple, we have not tried to find $H_c$ in closed form; all of the computations were done by the simulation program. The initial condition was chosen to be (0.1, −0.1, 0.05, 0.2), and the simulation was run for $2\pi$ seconds. Figs. 12–14 show the results. Note the rapid convergence to zero in the error terms.

## VI. Conclusions

The control law and simulation results presented in this paper suggest that for nonholonomic systems, stabilizing to a trajectory is a more appropriate problem to consider than stabilizing to a point. It should be noted that for drift-free systems, *all* points are equilibrium points (in the sense that with zero input, the system will remain at rest).

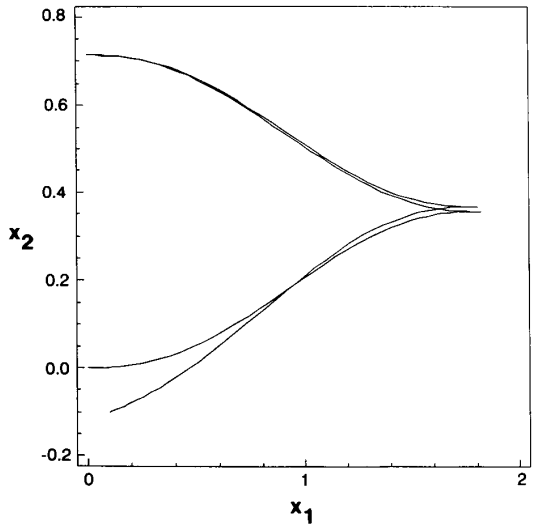


Fig. 14. This phase plot shows the desired and the actual trajectories projected onto the $(x_1, x_2)$ plane (the orientation of the car and the steering wheel angle are not shown). The desired trajectory is the one which starts at (0, 0). Note how quickly and smoothly the control law stabilizes the system to this trajectory.

However, if one adopts our point of view, one must also face the problem of finding feasible trajectories. Excellent work has been done [7], [9]–[11], [13] in this area, and methods for finding trajectories exist for a wide range of nonholonomic systems (including all of the examples found in this note).

In the context of nonkinematic models such as the real Hilare robot, the inputs are not the motor velocities but the torques, and thus the problems involving drift should also be examined. The control law presented here can be applied to stabilize systems with drift. However, it is not yet as clear how to generate the nominal trajectories for nonholonomic systems with drift.

The control law presented in this paper is robust to three types of error: initial condition errors, perturbations introduced along the trajectory, and noise in the sensor data. We have only shown the convergence results when there is an error in the initial condition, but it can be seen that the effects of the other two types of errors also are reduced using this law.

In summary, the path to a composite controller for mobile robots is: 1) utilize the path planners to generate a nominal open-loop trajectory; 2) apply the control law developed in this paper to stabilize the system to this nominal trajectory; 3) during operation of the robot, low-level sensor data can be used to avoid collisions caused by *a priori* errors in the knowledge of the environment. This new knowledge can be used to plan a new feasible nominal trajectory and find its associated stabilizing control law.

## Acknowledgment

Part of this research was done at the Laboratoire d'Automatique et d'Analyse des Systèmes in Toulouse, France during the summer of 1991. G. Walsh, D. Tilbury, and S. Sastry are grateful for the warm welcome they received there. The authors would like to thank Dr. Georges Giralt for many inspiring discussions and Andrew Teel for suggesting the two remarks after proposition one.

## REFERENCES

[1] G. Bastin and G. Campion. "Feedback control of nonholonomic mechanical systems." *Adv. Robot Contr.*, 1991.

[2] R. W. Brockett. *Control Theory and Singular Riemannian Geometry.* New York: Springer-Verlag, 1981.

[3] ——, "Asymptotic stability and feedback stabilization," in R. W. Brockett, R. S. Millman, and H. J. Sussman, eds, *Differential Geometric Control Theory.* Boston: Birkhauser, 1983, pp. 181–191.

[4] V. H. L. Cheng, "A direct way to stabilize continuous-time and discrete-time linear time-varying systems," *IEEE Trans. Automat. Contr.*, vol. AC-24, no. 4, pp. 641–543, 1979.

[5] J. M. Coron, "Global asymptotic stabilization for systems without drift," Université de Paris-Sud, Orsay, 1991.

[6] C. Canudis de Wit and O. J. Sordalen, "Exponential stabilization of mobile wheeled robots with nonholonomic constraints," In: *IEEE Conf. Decision Contr.*, pp. 692–697, 1991.

[7] G. Lafferriere and H. J. Sussmann, Motion planning for controllable systems without drift: a preliminary report. Tech. Rep., New York Univ. and Rutgers Univ., June 1990.

[8] J-P. Laumond, "Controllability of a multibody mobile robot," In: *IEEE Internat. Conf. Contr. Appl.*, pp. 1033–1038, Pisa, Italy, 1991.

[9] J-P. Laumond and T. Siméon, "Motion planning for a two degrees of freedom mobile robot with towing," in, *IEEE Int. Conf. Contr. Appl.*, 1991.

[10] R. Murray and S. Sastry. "Steering nonholonomic systems using sinusoids," in *IEEE Conf. Decision Contr.*, pp. 2097–2101, 1990.

[11] J. A. Reeds and L. A. Shepp. "Optimal paths for a car that goes both forwards and backwards." *Pacific J. Math.*, vol. 145 no. 2, 1990.

[12] Claude Samson. "Feedback stabilization of a nonholonomic car-like mobile robot," In: *IEEE Conf. Decision Contr.*, 1991.

[13] D. Tilbury, J-P. Laumond, R. Murray, S. Sastry, and G. Walsh. "Steering car-like robots with trailers using sinusoids," in *IEEE Int. Conf. Robotics Automation,* 1992.


# Control of Vibrational Systems

William C. Karl, George C. Verghese, and Jeffrey H. Lang

***Abstract*—Vibrational systems, characterized by their second-order or "spring-mass" nature, occur throughout engineering. Examples range from lightly damped structures, such as the proposed space station, to regional power system models. While the study of such vibrational systems has a long and rich history, concern has traditionally focused on the properties of *passive*, uncontrolled systems. The growing neeed to actively control large, complex systems of this type leads us to the study of controlled vibrational systems. We provide stability assuring constraints for compensator design. These constraints directly use the initial system model, thus avoiding the need for order reduction techniques with the associated problems of "spill-over" from the full to the reduced system model.**

## I. INTRODUCTION

Vibrational systems, characterized by their second-order or "spring-mass" nature, occur throughout engineering. Examples range from lightly damped structures, such as oil derricks and the proposed space station, to regional power system models. While the study of such vibrational systems has a long and rich history, traditionally, concern has focused on properties of *passive* systems, where active control of the complete system dynamics has *not* been a primary objective [1], [2]. As a consequence, the development of special results, techniques, and tools for properly and *directly* exploiting the special features of vibrational systems has, for the most part, been neglected. Our note presents constraints on vibrational system compensators that guarantee the stability of the compensated system, and that are relatively easy to use in a design setting. More detail may be found in [3].

In Section II, we start by reviewing vibrational systems and their stability from the standpoint of this work. Section III brings in the control issue by considering the addition of a compensator to the vibrational system model given in Section II. The compensator is incorporated into the model in a way that preserves the physical intuition inherent in the second-order form. Simple sufficiency conditions for the stability of the closed-loop system are then presented in the form of compensator design constraints in the spirit of [4]. A guide to notation is given as Appendix A. The proof of our main result is outlined in Appendix B.

Manuscript received May 18, 1992; revised April 30, 1993. This work partially supported by Center for Intelligent Control Systems, U.S. Army Research Office under grant DAAL03-92-G-0115, by the U.S. Air Force Office of Sponsored Research under contract AFOSR-92-J-0002, by the C. S. Draper Laboratory under contract number DL-H-441678, and by the Lockheed Missiles and Space Company under contract number LS90B4860F.

The authors are with the Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 9213575.

## II. VIBRATIONAL SYSTEMS

### A. System Definition

For this work, a vibrational system is taken to be any system that can be modeled by a set of coupled, second-order differential equations, as follows:

$$I\ddot{x} + D\dot{x} + Kx = Pu$$

$$y = Q\dot{x} + Rx \tag{1}$$

where the superscript dots denote differentiation with respect to time, and $x$, $u$, and $y$ are $n$-, $m$-, and $p$-dimensional vectors of (generalized) positions, actuator inputs, and sensor outputs, respectively. The coefficient matrices $D$ and $K$ are square and real, $I$ is the identity matrix, and $P$, $Q$, and $R$ are real matrices representing the external coupling of the system.

The model in (1) follows *naturally* from the widely used structural modeling schemes, such as the finite element method, and strong physical associations may be made with each term (eg. $D$ being a mass-normalized damping coefficient matrix and $K$ a normalized spring or stiffness coefficient matrix [5]). We preserve the flavor of these interpretations by maintaining the natural second-order form in the work to follow.

### B. Latent Values, Latent Vectors, and Stability

The stability of the uncontrolled system (1) is determined by the (Smith) zeros or latent values of the corresponding matrix polynomial [6], [7]

$$V(s) = Is^2 + Ds + K \tag{2}$$

The zeros or latent values and associated latent vectors of (2) are, respectively, the set of solutions, $s_i$ and $v_i$, to the equation

$$V(s_i)v_i = 0, \qquad v_i \neq 0.$$

The system (1) is asymptotically stable if and only if the latent values of (2) are in the open left half-plane. We will use the term