Pergamon

PII: S0005-1098(97)00052-6

Automatica, Vol. 33, No. 8, pp. 1571-1577, 1997

0005-1098/97 $17.00 + 0.00

Brief Paper

Output Stabilization of Square Nonlinear Systems

ROBERT MAHONY,† IVEN MAREELS‡ GEORGES BASTIN§ and GUY CAMPION§

Key Words—Output regulation; internal stability; robot navigation; stability domains; stabilizing feedback; singular control.

Abstract—This paper considers the problem of designing static-state feedback laws for output regulation of square affine nonlinear systems. The approach taken is to use input–output decoupling techniques to simplify the output regulation task into separate single-input single-output regulation tasks. In the case where the input-output decoupling matrix is full-rank, this approach yields the well-known input–output linearizing feedback law. In the case where the input–output decoupling matrix is rank-degenerate, it is shown that a static-state control law for output regulation can be constructed as long as the system can be input–output decoupled via dynamic feedback. The internal stability of the closed-loop system obtained using this approach is analysed. ©1997 Elsevier Science Ltd.

1. *Introduction and problem statement*

In recent years the question of state stabilization of nonlinear systems has been an area of significant development. The foundation of this interest can be traced back to the fundamental result of Brockett (1983), which showed that for affine systems where the number of inputs is strictly less than the number of states, no smooth static-state feedback law exists that asymptotically stabilizes the system state. To overcome this difficulty, authors have concentrated on two approaches: the use of time-varying control laws (see e.g. Pomet, 1992; Coron, 1992), and the use of discontinuous and non-smooth static-state control laws (see e.g. the early work by Sussmann (1979) and the more recent work by Canudas de Wit and Sordalen (1991), Kolmanovsky *et al.* (1994) and Khennouf and Canudas de Wit (1995)). A connection between these methods is presented in Coron and Rosier (1994). In comparison, the task of output regulation of a dynamic system has not been strongly pursued. This question has a strong practical motivation, since many physical systems have as many control inputs as control objectives (or outputs), though the dynamics of the system may contain additional 'internal' states. In such situations, the techniques developed for full-state stabilization need not be employed to achieve the desired control objectives. An application area in which these issues arise is in the control of kinematic models of mobile robots. These models have the advantage that they are simple nonlinear dynamic systems (d'Andrea-Novel *et al.*, 1996) that nevertheless display many of the characteristic difficulties associated with controlling general nonlinear systems. An example of the output regulation problem is the task of 'parking' the robot at the origin without proscribing the orientation of the steering wheels.

In this paper we consider general nonlinear affine control systems with the same number of inputs as outputs. Such systems are known as *square* systems. The output regulation task is approached by transforming the system into a form where each output can be individually regulated by a single input. The main tool employed to achieve this end is the dynamic extension algorithm (Nijmeijer and Van der Schaft, 1990, Section 8.2, especially pp. 263–264). In the case where the input–output decoupling matrix is full-rank, this approach yields the well-known input–output linearizing feedback law. The closed-loop system generated in this manner is studied for internal-stability properties and a theorem is given that characterises a subset of initial conditions for which the closed-loop system is internally stable. The situation is similar to that considered by authors studying peaking phenomena in cascaded systems (Saberi *et al.*, 1990; Sussmann and Kokotovic, 1991), where certain initial conditions generate transients in the system that become unbounded in finite time. In the case where the input–output decoupling matrix is rank-degenerate, however, the controller obtained by direct application of the dynamic extension algorithm would have a *dynamic* state. In this paper, the structure of the regulation problem is exploited to generate a *static*-state control law. The resulting algorithm is referred to as the *linearizing extended output stabilizing* (LEOS) control algorithm. An analysis of the internal stability of the closed-loop system generated in this manner is undertaken. To demonstrate the algorithm, an output regulating control law is designed for a kinematic model of a mobile robot.

The paper is organised into five sections. After the introduction, Section 2 deals with systems for which the input–output decoupling matrix is full-rank. The more general case, for systems where the decoupling matrix may be rank-degenerate, is dealt with in Section 3, while Section 4 considers issues associated with singularities in the control law. Section 5 presents the analysis of the design procedure applied to a kinematic model of a mobile robot, while brief conclusions are drawn in Section 6.

2. *Systems whose input–output may may be decoupled*

In this section the case of a square, nonlinear affine control system whose input–output decoupling matrix is full-rank at all points in state space is considered. In this case there exists a static-state feedback transformation of the system leading to fully decoupled input–output dynamics. Applying a simple exponentially stabilizing control law leads to a closed-loop system with asymptotically stable output dynamics. The main result of the section presents a characterisation of a set of initial conditions for which the internal dynamics of the closed-loop system are well defined for all time.

Consider a nonlinear dynamic control system of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f(x) + \sum_{i=1}^{\infty} g_i(x)u_i, \quad x(0) = x_0, \qquad y = h(x). \tag{1}$$

* Received 3 May 1995; revised 15 May 1996; revised 4 November 1996; final version received 30 January 1997. The original version of this paper was presented at the IFAC Conference on *System Structure and Control*, which was held in Nantes, France, during 5–7 July 1995. The Published Proceedings of this IFAC meeting may be ordered from: Elsevier Science Limited, The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, U.K. This paper was recommended for publication in revised form by *Associate Editor* Henk Nijmeijer under the direction of *Editor* Tamer Başar. Corresponding author Dr Robert Mahony. Tel. +61 6 279 8078; Fax +61 6 249 2698; E-mail Robert.Mahony @anu.edu.au.

† Cooperative Research Centre for Robust and Adaptive Systems, RSISE, ANU ACT 0200, Australia.

‡ Department of Electrical and Electronics Engineering, The University of Melbourne, Grattan Street, Parkville, Vic3052, Australia.

§ CESAME, Batiment Euler, 4–6 Avenue G. Lemaitre, B1348 Louvain la Neuve, Belgium.

Here $x \in \mathbb{R}^n$ is the state variable, $u = (u_1, \ldots, u_m)^T$ is an m-dimensional input variable and y is an m-dimensional output variable. Such systems are known as *square* systems. The functions f, $g_i: \mathbb{R}^n \to \mathbb{R}^n$ and $h: \mathbb{R}^n \to \mathbb{R}^p$ are assumed to be smooth. For the remainder of this section it is assumed that the drift term $f(x)$ is identically zero on the whole state space. Thus (1) can be written in the form

$$\dot{x}(t) = g(x)u, \quad y = h(x), \tag{2}$$

where the input vectors are written in matrix form $g: \mathbb{R}^n \to \mathbb{R}^{n \times m}$, with $g(x) := (g_1(x), \ldots, g_m(x))$.

The input–output decoupling matrix $A(x) \in \mathbb{R}^{m \times m}$ for the system (1) has ijth element $\{A(x)\}_{ij} = L_{g_j} L_f^{\rho_i} h_i(x)$, where $L_{g_j} h_i(x)$ is the Lie derivative of $h_i(x)$ in the direction g_j, and $\rho_1^1, \ldots, \rho_m^1$ are the characteristic numbers of the outputs $y = (h_1, \ldots, h_m)$ (Nijmeijer and Van der Schaft, 1990, pp. 247–248). Indeed, since $f(x) \equiv 0$, $A(x) = \mathrm{D}h(x)g(x)$, where $\mathrm{D}h(x)$ is the $\mathbb{R}^{m \times n}$ Jacobian matrix of partial derivatives, $\{\mathrm{D}h(x)\}_{ij} = (\partial h_i / \partial x^j)(x)$. It is known Nijmeijer and Van der Schaft, 1990, Theorem 8.9) that if $\mathrm{D}h(x)g(x) \in \mathbb{R}^{m \times m}$ is full-rank then the system can be strongly input–output decoupled by the input transformation $u(x, u') := [\mathrm{D}h(x)g(x)]^{-1}u'$, where u' is the new input. Thus the (linearizing) control law $u' := -y$, or, in terms of the original inputs and outputs,

$$u(x) = -[\mathrm{D}h(x)g(x)]^{-1}h(x) \tag{3}$$

is a candidate for regulating the system output. Indeed, using the comparison function $V(x) = 0.5h(x)^T h(x)$ and computing its time derivative along the solutions $x(t)$ of the closed-loop system $\dot{x} = g(x)u(x)$ yields $\dot{V}(x(t)) = -V(x(t))$. Certainly, the linearizing control law (3) stabilizes the system output. However, the Lyapunov function $V(x)$ provides no information on the evolution of the 'internal states' of the system—those parts of x that do not contribute to the output $h(x)$. The concept of 'internal state' is not easily defined in a rigorous manner; however, the concept of internal stability, in the context of output regulation, can be defined as follows:

Definition 1. Consider a system of the form (1) equipped with a static-state control law $u := u(x)$. The closed-loop system is said to be *internally stable with regulated output* if, for all initial conditions $x_0 \in \mathbb{R}^n$, the solution $x(t; x_0)$ of the closed-loop system exists and remains bounded for all time, and in addition the output $y(t) = h(x(t; x_0))$ converges to zero.

In practice, this requirement is too strong for many systems. Indeed, even if locally around the zero-output-level set a control of the form (3) leads to well-behaved solutions of the closed loop, this will not be the case for all initial conditions. The situation is analogous to the peaking phenomena studied by Saberi *et al.* (1990) and Sussmann and Kokotovic (1991), where initial conditions that are too far from the zero-level set can generate transients in the system that become unbounded in finite time. This leads us to propose the following definition of *weak* internal stability.

Definition 2. Consider a system of the form (1) equipped with a static-state control law $u := u(x)$. The closed-loop system is said to be *weakly internally stable with regulated output* if there exists an open neighbourhood $\Omega \subset \mathbb{R}^n$ containing the zero-level set of the output, $\{x \in \mathbb{R}^n \mid h(x) = 0\}$, as a proper subset, such that for any initial condition $x_0 \in \Omega$, the solution $x(t; x_0)$ of the closed-loop system exists and remains bounded for all time, and in addition the output $y(t) = h(x(t; x_0))$ converges to zero.

Theorem 3. Consider a square system of the form (2). Assume that $\{\mathrm{D}h(x)g(x)\}$ is full-rank for all $x \in \mathbb{R}^n$. Then the closed-loop system generated by applying the control (3) is weakly internally stable with regulated output (cf. Definition 2).

Moreover, an open neighbourhood Ω of initial conditions for which the closed-loop solutions of the system exist for infinite time can be explicitly characterized by

$$\Omega := \bigcup_{x^* \in \{x \mid h(x) = 0\}} \bigcup_{r > 0} \Omega_{(x^*, r)},$$

where, for each $x^* \in \{x \mid h(x) = 0\}$,

$$\Omega_{(x^*, r)} := \left\{ x \in B_{x^*}(\tfrac{1}{4}r) \mid g(x) \times [\mathrm{D}h(x)g(x)]^{-1} \frac{\exp(|h(x)|\,\lambda) - 1}{\lambda} < \tfrac{1}{2}r \right\}.$$

Here $B_{x^*}(r) := \{x \mid |x - x^*| < r\}$, and $\lambda := \lambda(x^*, r)$ is given by

$$\lambda = \sup_{x, y \in B_{x^*}(r)} \frac{|g(x)[\mathrm{D}h(x)g(x)]^{-1} - g(y)[\mathrm{D}h(y)g(y)]^{-1}|}{|x - y|}.$$

Finally, for any $x_0 \in \Omega_{(x^*, r)}$, for some $x^* \in \{x \mid h(x) = 0\}$ and $r > 0$, the solution $x(t; x_0)$ remains in a ball of radius $\frac{1}{2}r$ around x_0: $|x(t; x_0) - x_0| \le \frac{1}{2}r$.

Remark 4. The sets $\Omega_{(x^*, r)}$ are open subsets of $\mathbb{R}^n$, which are non-empty since $x^* \in \Omega_{(x^*, r)}$. Consequently, Ω is an open neighbourhood of $\{x \in \mathbb{R}^n \mid h(x) = 0\} \subset \Omega$.

Proof. For any x^* such that $h(x^*) = 0$ and for any $r > 0$, construct $\lambda(x^*, r)$ and $\Omega_{(x^*, r)}$ as indicated in the theorem statement. Let $x_0 \in \Omega_{(x^*, r)}$ be an initial condition for the closed-loop system

$$\dot{x} = -g(x)[\mathrm{D}h(x)g(x)]^{-1}h(x).$$

Since we assume that $g(x)$ and $h(x)$ are smooth and that $\mathrm{D}h(x)g(x)$ is full-rank, there exists a unique local solution $x(t; x_0)$, well defined on some maximal interval $[0, t^*)$. The proof proceeds by contradiction.

Assume there exists a finite time $\bar{t} < t^*$ such that $|x(\bar{t}, x_0) - x_0| \ge \frac{3}{4}r$. Let $t_1 \le \bar{t} < t^*$ be defined as

$$t_1 = \inf_{t > 0} \{t \mid |x(t; x_0) - x_0| \ge \tfrac{3}{4}r\}.$$

Observe that t_1 is defined in such a way that it follows for any $x_0 \in \Omega_{(x^*, r)}$ that $x(t; x_0) \in B_{x^*}(r)$ for $t \in [0, t_1)$.

Since, by construction,

$$\dot{y} = \mathrm{D}h(x)g(x)u(x) = -\mathrm{D}h(x)g(x)[\mathrm{D}h(x)g(x)]^{-1}h(x) = -y,$$

one has

$$h(x(t; x_0)) = h(x_0)\,\mathrm{e}^{-t}$$

on $[0, t^*)$. Thus, from the closed-loop expression

$$x(t; x_0) - x_0 = \int_0^t -g(x(\tau; x_0)) \times [\mathrm{D}h(x(\tau; x_0))g(\tau; x_0))]^{-1}h(x(\tau; x_0))\,\mathrm{d}\tau.$$

Computing the norm of this expression and approximating the integral for $t \in [0, t_1)$, one obtains

$$|x(t; x_0) - x_0| \le \int_0^t \{|g(x_0)[\mathrm{D}h(x_0)g(x_0)]^{-1}| + \lambda\,|x(\tau; x_0) - x_0|\}\,|h(x_0)|\,\mathrm{e}^{-\tau}\,\mathrm{d}\tau,$$

where the definition of $\lambda(x^*, r)$ is used along with the fact that $x(t; x_0) \in B_{x^*}(r)$. Over-bounding $\mathrm{e}^{-t} \le 1$ leaves the above inequality in a form to which the Bellman–Gronwall lemma (Sanders and Verhulst, 1985, p. 3) may be applied, yielding

$$|x(t;x_0)-x_0|\le|g(x_0)[Dh(x_0)g(x_0)]^{-1}|\frac{\exp[|h(x_0)|\lambda]-1}{\lambda}$$

for $t\in[0,t_1)$. Owing to the construction of $\Omega_{(x^*,r)}$, one now obtains $|x(t;x_0)-x_0|\le\frac{1}{2}r$, $0\le t\le t_1$, but this in turn contradicts the existence of t_1.

As a consequence, one has $|x(t;x_0)-x_0|<\frac{3}{4}$ for $t\in[0,t^*)$. Observe that $x(t;x_0)$ is a bounded solution to an ordinary differential equation on the time interval $[0,t^*)$, and consequently its limit at t^* must exist. But then classical existence results ensure that the solution is well defined for some slightly longer time interval $[0,t+\delta)$. This ensures that t^* is not a finite escape time for the system and (by contradiction) that no such finite escape time exists. The output regulation property is observed directly from the form of the output dynamics, while the other claims in the theorem statement follow immediately from the above argument restated in the knowledge that the solution $x(t;x_0)$ exists for all time. □

3. *The linearizing extended output stabilizing control algorithm*

In this section the problem of designing a static-state feedback law that exponentially stabilizes the output of a system for which the input–output decoupling matrix is rank-degenerate is considered. Many systems of this type may still be input–output decoupled using a dynamic feedback law (Nijmeijer and Van der Schaft, 1990, Section 8.2, especially pp. 263–264). This section develops an algorithm to design *static*-state feedback laws that assign linear stable dynamics to the output. We refer to this algorithm as the linearizing extended output stabilizing (LEOS) control algorithm. Here the term 'extended' refers to the similarities to the dynamic extension algorithm.

Consider a general affine nonlinear dynamic control system of the form (1). To simplify the technical details, consideration is further restricted to analytic systems, although non-analytic systems can be tackled in a piecewise fashion using the same techniques. In Section 2 (cf. (2)) it was assumed that the drift term was equivalent to zero, $f(x)\equiv 0$. In this section it is convenient to derive the desired feedback laws for non-zero drift, though the particular cases that are of interest (kinematic models of mobile robots) will all satisfy $f(x)\equiv 0$.

The following development is initially the same as the dynamic extension algorithm (Nijmeijer and Van der Schaft, 1990, Section 8.2), and is included to introduce the notation used later in the section. Consider the output equation $y=h(x)$ and let $\rho_1^1,\ldots,\rho_m^1$ be the characteristic numbers of the outputs $y=(y_1,\ldots,y_m)$ (Nijmeijer and Van der Schaft, 1990, p. 247]. Thus one may write

$$\begin{pmatrix} y_1^{(\rho_1^1+1)}\\ \vdots\\ y_m^{(\rho_m^1+1)}\end{pmatrix}=E^1(x)+F^1(x)u,$$

where $y^{(\rho_j^i+1)}$ denotes the (ρ_j^i+1)th time derivative of y, $E^1(x)=(L_f^{\rho_1^1+1}h_1(x),\ldots,L_f^{\rho_m^1+1}h_m(x))^{\mathrm{T}}$ and $F^1(x)$ is the input–output decoupling matrix with ijth entries $\{F^1(x)\}_{ij}=L_{g_j}L_f^{\rho_i^1}h_i(x)$.

Since the system considered is analytic, the rank of $F^1(x)$ is constant except on a set of measure zero in $\mathbb{R}^n$. Denote the generic rank of $F^1(x)$ (off the set of measure zero) by r_1. It is convenient to refer to the set of points at which the various rank conditions required for the construction of the output stabilizing feedback law do not hold as the set of *singular points* in $\mathbb{R}^n$. Thus the set of singular points includes all points at which rank $F^1(x)\ne r_1$ as well as other points defined in the sequel. The set of singular points will always be of zero measure in $\mathbb{R}^n$.

Given a point $x\in\mathbb{R}^n$ that is not a singular point, reorder and related the output functions $h_1,\ldots,h_m$ and the inputs u (and hence the columns of $F^1(x)$) to ensure that the upper-left $r_1\times r_1$ block of $F^1(x)$ is full-rank. Partition the output into two parts: $(h_1,\ldots,h_m)$. Correspondingly, partition the vector $E^1(x)$ into its first r_1 entries $E_1^1(x)$ and its remaining entries $E_2^1(x)$, and the matrix $F^1(x)$ into four submatrices, where the upper-left submatrix $F_{11}^1(x)\in\mathbb{R}^{r_1\times r_1}$ is square and full rank. As a consequence, the last $m-r_1$ columns of $F^1(x)$ are linearly dependent on the first r_1 columns. Consider the input transformation

$$u(x,u^1,\bar{u}^1):=\begin{pmatrix}-(F_{11}^1)^{-1}E_1^1\\ 0\end{pmatrix}+\begin{pmatrix}(F_{11}^1)^{-1} & -(F_{11}^1)^{-1}F_{12}^1\\ 0 & I_{m-r_1}\end{pmatrix}\begin{pmatrix}u^1\\ \bar{u}^1\end{pmatrix},\quad (4)$$

where the new inputs are denoted by $(u^1,\bar{u}^1)\in\mathbb{R}^{r_1}\times\mathbb{R}^{m-r_1}$. Applying this input transformation to (1) yields the output dynamics

$$\begin{pmatrix} y_1^{(\rho_1^1+1)}\\ \vdots\\ y_m^{(\rho_m^1+1)}\end{pmatrix}=\begin{pmatrix}0\\ E_2^1-F_{21}^1(F_{11}^1)^{-1}E_1^1\end{pmatrix}+\begin{pmatrix}I_{r_1} & 0\\ F_{21}^1(F_{11}^1)^{-1} & 0\end{pmatrix}\begin{pmatrix}u^1\\ \bar{u}^1\end{pmatrix}.\quad (5)$$

Thus, after applying (4), the first r_1 outputs are input–output decoupled to the new inputs $u^1\in\mathbb{R}^{r_1}$ and fully decoupled from the remaining inputs $\bar{u}^1$.

It is at this point that the present development differs significantly from the standard dynamic extension algorithm. Since our aim is simply to stabilize the output, it is possible to specify the first r_1 inputs u^1 to stabilize the first r_1 outputs explicitly. Define the inputs $u^1=(u_1^1,\ldots,u_{r_1}^1)$ as follows:

$$\begin{aligned}u_i^1(x)&=-\sum_{j=0}^{\rho_i^1}C_j^{\rho_i^1+1}\frac{d^jh_i}{dt^j}(x)\\ &=-\sum_{j=0}^{\rho_i^1}C_j^{\rho_i^1+1}L_f^jh_i(x)\quad i=1,\ldots,r_1,\end{aligned}\quad (6)$$

where $C_b^a=a!/b!\,(a-b)!$. Choosing $u_i^1(x)$ as given above ensures that $y_i(t)$ satisfies the linear homogeneous differential equation

$$\left(\frac{d}{dt}+1\right)^{(\rho_i^1+1)}y_i(t)=0,$$

which has the solution $y_i(t)=(c_i^0+c_i^1t+\ldots+c_i^{\rho_i^1}t^{\rho_i^1})\,e^{-t}$ for constants $c_i^0=y_i(0)$ and

$$c_i^p=\frac{1}{p!}\sum_{j=0}^{p-1}C_j^p\frac{d^jy_i}{dt^j}(0)\quad(p=1,\ldots,\rho_i^1).\quad (7)$$

Remark 5. The particular output dynamics specified at this point are chosen with regard to the analysis of behaviour of the control laws in the vicinity of singular points. These issues are discussed in Section 4.

Thus one may define a new (partly closed-loop) system with inputs $u'\in\mathbb{R}^{m-r_1}$ and outputs $y'\in\mathbb{R}^{m-r_1}$:

$$\dot{x}=f(x)+g(x)\left[\begin{pmatrix}-(F_{11}^1)^{-1}E_1^1\\ 0\end{pmatrix}+\begin{pmatrix}(F_{11}^1)^{-1} & -(F_{11}^1)^{-1}F_{12}^1\\ 0 & I_{m-r_1}\end{pmatrix}\begin{pmatrix}u^1(x)\\ u'\end{pmatrix}\right]\quad (8)$$

$$\begin{aligned}y'&=(y_{r_1+1}^{(\rho_{r_1+1}^1+1)},\ldots,y_m^{(\rho_m^1+1)})^{\mathrm{T}}\\ &=E_2^1(x)-F_{21}^1(x)[F_{11}^1(x)]^{-1}E_1^1(x)\\ &\quad+F_{21}^1(x)[F_{11}^1(x)]^{-1}u^1(x),\end{aligned}\quad (9)$$

where the solutions $x(t)$ of these equations satisfy $h_i(x(t))=\sum_{j=0}^{\rho_i^1}c_i^jt^j\,e^{-t}$, $i=1,\ldots,r_1$. The new inputs $u'\in\mathbb{R}^{m-r_1}$ are just the inputs $\bar{u}^1$ that were not assigned in the first step of the algorithm, while the new outputs y' are the highest-order time derivatives of the unstabilized outputs that were obtained in the previous step. Observe that the new inputs u' do not enter directly into the output equation (9).

Remark 6. It is most convenient to continue subscripting both remaining inputs and outputs as though they are parts of the full input and output vectors. Thus the new input vector is $u' = (u_{r_1+1}, \ldots, u_m)$ and the output vector is given by (9).

Obtaining a full control law is simply a matter of iterating the above procedure, starting with the newly defined system equations (8) and (9). Supserscripts l on the terms E^l, F^l (generically of rank r_l), u^l, etc. are used to indicate which iteration these quantities are associated with. Let

$$\Upsilon_i^k = \rho_i^1 + \rho_i^2 + \ldots + \rho_i^k$$

for $i = (r_1 + \ldots + r_{k-1} + 1), \ldots, (r_1 + \ldots + r_k)$. The general form for the feedback used in the kth iteration is

$$u_i^k(x) = -\sum_{j=0}^{\Upsilon_i^k} C_j^{\Upsilon_i^k+1} k^{(\Upsilon_i^k+1-j)} \frac{d^j}{dt^j} h_i(x)$$

for $i = (r_1 + \ldots + r_{k-1} + 1), \ldots, (r_1 + \ldots + r_k)$. Choosing $u_i^k(x)$ as above ensures that $y_i(t)$ satisfies the linear ODE

$$\left(\frac{d}{dt} + k\right)^{(\Upsilon_i^k+1)} y(t) = 0,$$

which has the solution

$$y_i(t) = e^{-kt} \sum_{j=0}^{\Upsilon_i^k} c_i^j t^j, \quad i = (r_1 + \ldots + r_{k-1} + 1), \ldots, (r_1 + \ldots + r_k), \tag{10}$$

for constants c_i^j given by (7). As a consequence, the kth set of outputs converge to zero strictly faster (exponentially with rate $-k$) than the previous sets of outputs. Heuristically, the motivation for htis choice is that the later outputs, more prone to numerical ill-conditioning (following from repeated singular feedback transformations), converge to zero before any ill-conditioning in the algorithm occurs. A more detailed discussion of these issues is undertaken in Section 4.

To fully determine the inputs, one continues to apply the above procedure until all the outputs have been assigned asymptotically stable dynamics. Since the system is square and analytic, and it has been assumed that a dynamic feedback law exists that input–output decouples the system, it follows that, apart from on a set of zero measure, each output can eventually be controlled by an input.

4. *Input–output singularities and internal stability*

In this section the presence of singularities in linearizing extended output stabilizing (LEOS) control laws (cf. Section 3) are considered. In general, singularities can be avoided by switching the input–output ordering used in the LEOS algorithm whenever a singular surface is approached. However, for drift-free systems of the form (2), the LEOS control law will always have singularities on the zero-output-level set, owing to the non-holonomic nature of the system equations. Careful consideration of the behaviour of the closed-loop system in the vicinity of these forced singularities provides the motivation for the particular output dynamics chosen in Section 3.

The set of singular points associated with the LEOS algorithm is defined as follows.

Definition 7. Consider a nonlinear dynamic control system of the form (1) and let $u := u(x)$ be a static-state feedback control law given by the LEOS control algorithm. The set of singular points resulting from the algorithm is referred to as the *set of singular points associated with* $u(x)$, and is denoted by S_u. This set is explicitly characterised by

$$S_u := \{x \in \mathbb{R}^n \mid \det F_{11}^k(x) = 0 \text{ for some } k \in \{1, 2, \ldots, K\}\}.$$

The presence of singularities of this nature is a fundamental limitation on any algorithm that relies on the decomposition of input–output dependence generated by the dynamic extension algorithm. On a singular surface, the degeneracy of one of the matrices $F_{11}^k(x)$ implies that certain control actions are nulled, and, equivalently, the associated output dynamics are uncontrollable. The LEOS algorithm makes no allowance for avoiding singular surfaces, and should the closed-loop system evolve to cross a singular surface, the non-zero output dynamics assigned by the algorithm will require unbounded control action to be achieved. Unbounded control of this nature will tend to generate finite-time escape dynamics in the internal states of the system. The situation, however, is not necessarily the problem that it may at first appear to be. In particular, the singular surfaces are fully algebraically characterised, and the approach of a singular surface can be monitored. By swapping the order in which the inputs and outputs are chosen in the LEOS algorithm, it is possible to alter the singular-point structure of the control law generated and often remove entirely the singular surface that is being approached. A control algorithm exploiting this technique will generate discontinuous control action at the instant when the input–output ordering is switched. It is beyond the scope of this paper to investigate the general performance of such a switching strategy in practice.

For a general affine nonlinear system of the form (1), it is unlikely that a singularity will occur exactly on the zero-output-level set. Unfortunately, a drift-free system of the form (2) has precisely the required structure (due to its non-holonomic nature) to create a singularity on the zero-output-level set. Consider a system of the form (2) where the input–output matrix is not full-rank. Applying the input transformation (4) and setting E_1^1 and E_2^1 equal to zero then it can be seen (cf. (5)) that there is a singularity (associated with $u^1 = (y_1, \ldots, y_{r_1}) = 0$) lying exactly on the zero-output-level set. To indicate how a control law generated by the LEOS algorithm remains well defined in the vicinity of singularities of this form, it is simplest to provide an example.

Consider a square system, of the form (2), with two inputs and outputs. Assume that the input–output decoupling matrix has dimension one and write the transformed output dynamics in the form (cf. (5)) $\dot{y}_1 = u_1'$, $\dot{y}_2 = F_{21}^1(F_{11}^1)^{-1}u_1'$, where (u_1', u_2') are the transformed inputs of the original system, $(u_1', u_2') = (u^1, \bar{u}^1)$ (cf. (4)). Applying the linear stabilizing control action $u_1'(x) := -h_1(x) = -y_1$, the y_2 dynamics have the form

$$\dot{y}_2 = H(x)u_1'(x), \tag{11}$$

where $H(x) := F_{21}^1(x)[F_{11}^1(x)]^{-1}$. Because of the choice of output dynamics for y_1 it follows that $\dot{u}_1^1(x(t)) = -u_1'(x(t))$. The second derivative of y_2 can be written as

$$\ddot{y}_2 = u_1'(x)\left([N(x) \quad M(x)]\begin{pmatrix} u_1'(x) \\ u_2' \end{pmatrix} - H(x)\right), \tag{12}$$

where

$$[N(x) \quad [M(x)] = DH(x)g(x)\begin{pmatrix} (F_{11}^1)^{-1} & -(F_{11}^1)^{-1}F_{12}^1 \\ 0 & I_{m-r_1} \end{pmatrix}.$$

In particular, observe that the right-hand side of (12) contains a multiplicative factor $u_1'(x) = -y_1$. This causes the singularity in the u_2' control at $y_1 = 0$.

For $y_1 \neq 0$ choose $u_2'(x)$ according to (6). Recall that while the closed-loop solution remains well defined, the output dynamics are $y_1(t) = y_1(0)e^{-t}$, and $y_2(t) = \{y_2(0) + [\dot{y}_2(0) + 2y_2(0)]t\}e^{-2t}$. Now the time evolution of y_1 ensures that $u_1'(x(t)) = -y_1(x(t)) = -y_1(0)e^{-t}$, at least while the solution $x(t)$ remains well defined. Certainly, the control u_1' will remain well defined (and indeed decrease to zero in the limit). The situation for u_2' is the crux of the matter. Substituting for the known evolution of y_1, y_2 and u_1' in the definition of u_2', one has

$$\begin{aligned} u_2'(x) &:= \frac{1}{u_1'(x)M(x)}\{[-[u_1'(x)]^2N(x) + u_1'(x)H(x) + \ddot{y}_2], \\ &= \frac{y_1(0)N(x)}{M(x)}e^{-t} + \frac{H(x)}{M(x)} + 4\frac{y_2(0) + \dot{y}_2(0)}{M(x)y_1(0)}e^{-t}. \end{aligned} \tag{13}$$

Assume that $M(x) \neq 0$ in an open neighbourhood of the zero-level set; that is, that the only singularities present at the zero-level set are those associated with the control u^1. It is necessary to show that the input $u_2'(x)$ remains well defined in the limit as $t \to \infty$. The first term of (13) is well behaved, since the numerator has a squared dependence on u_1', which dominates the effect of the $[u_1'(x)]^{-1}$ singularity. The last term is also well behaved owing to the particular choice of the y_2 dynamics, which converge like e^{-2t} while the singularity converges like e^{-t}. This analysis provides the justification for the choice of the dynamics made in the LEOS algorithm described in Section 3. The final term in the expression for u_2' is also well behaved, since, recalling (11), it follows that

$$H(x(t)) = \frac{\dot{y}_2}{u_1'(x(t))} = \frac{2[\dot{y}_2(0) + 2y_2(0)]t - \dot{y}_2(0)}{y_1(0)} e^{-t}, \quad (14)$$

at least while the solution $x(t)$ remains well defined.

The above discussion suggests that, for the systems of interest (kinematic models of mobile robots), the exponential decay of the control action could be used to obtain a result analogous to that of Theorem 3. Firstly, it is necessary to add an additional condition on the nature of $H(x)$ to ensure that the dynamics assigned by (14) can be achieved. Effectively, this is equivalent to the requirement that $H: \mathbb{R}^n \to \mathbb{R}$ is not bounded away from zero. In general, $H(x) = F_{21}^1(x)[F_{11}^1(x)]^{-1}$ is a matrix function relating the r_1 assigned inputs u^1 to the remaining $m - r_1$ outputs. In this case $e^t H(x)u^1 = -H(x(t))y_1(0)$ is the term that is crucial to the behaviour of u_2. Rather than including the initial condition in the analysis given below, we have opted to use the stronger (and much simpler) requirement of forcing $H(x)$ to the zero matrix, and consequently ensuring uniform convergence of $u_2'(x)$ for any initial condition.

Remark 8. It is important to also consider what happens for systems where three or more steps of the algorithm are needed to generate the full LEOS control law. Recall (9) and observe that for a drift-free system the output y^i will always depend linearly on the input u^1, the term $E^1(x) \equiv 0$, and the vectors $E^k(x)$ decay faster than the u^1 term owing to the nature of the output dynamics assigned. As a consequence, the singularity at the zero-level set is dominated by the u^1 term, and the analysis is analogous to that given above, though considerably more complex in notation. Unfortunately, space restrictions do not allow us to present the details of this relationship.

Remark 9. In the case of most non-holonomic systems, two iterations of the LEOS algorithm will be sufficient to design the full closed-loop controller. If one considered systems with both velocity and acceleration constraints then in general three iterations of the LEOS algorithm would be required.

Theorem 10. Consider a square analytic system

$$\frac{d}{dt}x = \sum_{i=1}^{m} g_i(x)u_i, \quad x(0) = x_0,$$
$$y_j = h_j(x) \quad \text{for } j = 1, \ldots, m,$$

where $x \in \mathbb{R}^n$ and $y, u \in \mathbb{R}^m$. Assume that the LEOS control algorithm provides a static control law $u := u(x)$, valid off a set of singular points S_u (cf. Definition 7) of measure zero. Let

$$H(x) := F_{21}^1(x)[F_{11}^1(x)]^{-1},$$

where $F_{21}^1(x)$ and $F_{11}^1(x)$ are defined in the first step of the LEOS algorithm (cf. Section 3). Define the set Γ by

$$\Gamma = \{x \in \mathbb{R}^n \mid h(x) = 0, H(x) = 0\}$$

and assume that it is non-empty. Assume further that there exists an open neighbourhood $W \subseteq \mathbb{R}^n$ of the set Γ that contains no singular points $x \in S_u$ *except those explicitly generated* by the control law at $h(x) = 0$. Then the closed-loop system is weakly internally stable with regulated output (cf. Definition 2).

Proof. This differs from that of Theorem 3 in two main ways. Firstly, instead of dealing with the entire space $\mathbb{R}^n$, one deals only with the subset $W \subseteq \mathbb{R}^n$. Thus the set Ω of stable initial conditions is constructed to be

$$\Omega = \bigcup_{\{x^* \in \Gamma\}} \bigcup_{\{r>0, B_{x^*}(r) \subset W\}} \Omega_{(x^*, r)},$$

where $B_{x^*}(r) := \{x \mid |x - x^*| < r\}$ and

$$\Omega_{(x^*,r)} := \left\{ x \in B_{x^*}(\tfrac{1}{4}r) \;\middle|\; |g(x)u(x)| \frac{\exp[C\,|h(x)|\,\lambda] - 1}{\lambda} < \tfrac{1}{2}r \right\}.$$

Here $\lambda := \lambda(x^*, r)$ is given by

$$\lambda(x^*, r) = \sup_{x,y \in B_{x^*}(r)} \frac{|g(x)u(x) - g(y)u(y)|}{|x - y|}.$$

Since W is an open neighbourhood of Γ, for all $x^* \in \Gamma$ there exists a range of r for which $B_{x^*}(r) \subset W$. It follows that Ω itself is a non-empty open neighbourhood of Γ.

Secondly, the time evolution of the output (and consequently of the control $u(x(t))$) is composed of time-dependent terms of the form $t^j e^{-kt}$. Whereas in the proof of Theorem 3 the inequality $e^{-t} \le 1$ was used, here the inequality $t^j e^{-kt} \le (j/k)^j e^{-j}$ must be employed. This is verified by observing that $t^j e^{-kt}$ is unimodal on $(0, \infty)$, with its maximum at $t = j/k$. To account for this difference, a constant C is included in the definition of $\Omega_{(x^*,r)}$, where C is taken to be the maximum of these bounds for the particular control law used. The remainder of the argument is analogous to that for Theorem 3. □

5. *The box car robot*

In this section two examples are presented that indicate the manner in which the LEOS control algorithm (cf. Section 3) is applied. The system considered is a simple model of a mobile robot commonly known as the box car robot.

Example 1. Consider the box car robot shown in Fig. 1. Denote its Euclidean position in $\mathbb{R}^2$ by (x, y) and its orientation (angle from x axis to forward direction of the robot) by α (expressed in radians). One may write the kinematic system equations for the box car robot as follows (Canudas de Wit and Sordalen, 1991):

$$\dot{x} = u_1 \cos\alpha, \quad (15)$$
$$\dot{y} = u_1 \sin\alpha, \quad (16)$$
$$\dot{\alpha} = u_2. \quad (17)$$

Writing this in the form (2) yields

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ \alpha \end{pmatrix} = \begin{pmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = g(x, y, \alpha)u.$$

The outputs used are the Euclidean coordinates (x, y). The control task is to drive the output, $h(x, y, \alpha) = (x, y)^T$ to zero while the full state (x, y, α) remains bounded.

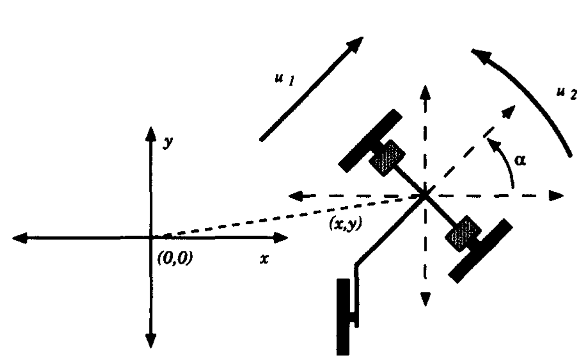

Fig. 1. The box car robot, with forward velocity u_1 and angular velocity u_2.

The system cannot be input–output decoupled using static-state feedback, since the input–output decoupling matrix

$$\mathrm{D}h(x, y, \alpha)g(x, y, \alpha) = \begin{pmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \end{pmatrix} =: F^1(x, y, \alpha)$$

is rank-deficient. Proceeding according to the LEOS algorithm, an input transformation is applied to bring the system into the form (5). As long as $\cos\alpha \neq 0$, there is no need to rearrange the order of the inputs and outputs and the matrix $F^1_{11}(x, y, \alpha) = \cos\alpha$. The singular points associated with inverting F^1_{11} are just those points where $\cos\alpha = 0$. Off the set of singular points, the input transformation is simply $u_1 = (\cos\alpha)^{-1}v_1$, $u_2 = v_2$, where v_1 and v_2 are the new inputs. According to (6), set

$$v_1 := -x \Leftrightarrow u_1(x, y, \alpha) := \frac{x}{\cos\alpha}. \tag{18}$$

The dynamics of the partially closed-loop system are now $\dot{x} = -x$, $\dot{y} = -x\tan\alpha$ and $\dot{\alpha} = u_2$. The dynamics of the output y can also be written as $\dot{y} = v_1 \tan\alpha =: v_1 H(x, y, \alpha)$. Here $H(x, y, \alpha) = \tan\alpha$ is the function that was key to the discussion in Section 4. Observe that $H(x, y, \alpha) = 0$ for $\alpha = q\pi$ and q any integer. As a consequence, the limit set Γ, defined in Theorem 10, is non-empty.

Define a new output function $h'(x, y, \alpha) = -x\tan\alpha = \dot{y}$. It is easily verified that

$$\mathrm{D}h'(x, y, \alpha)g(x, y, \alpha) = (-\tan\alpha\cos\alpha \;\; -x(1 + \tan^2\alpha)),$$

and thus $\ddot{y} = x\tan\alpha - x(1 + \tan^2\alpha)u_2$. In the notation introduced in Section 3, one has $E^2(x, y, \alpha) = x\tan\alpha$ and $F^2(x, y, \alpha) = -x(1 + \tan^2\alpha)$. The matrix $F^2(x, y, \alpha)$ is a scalar function associated with the last output to be stabilised, and must be inverted directly. The singular points associated with inverting $F^2(x, y, \alpha)$ are characterised by $x(1 + \tan^2\alpha) = 0 \Leftrightarrow x = 0$. The final set of singular points for the control law $u = (u_1, u_2)$ is

$$S_u = \{(x, y, \alpha) \mid \alpha = \tfrac{1}{2}\pi + q\pi \text{ for } q \in \mathbb{Z}, \text{ or } x = 0\}.$$

Off the set of singular points, the second input transformation of the LEOS algorithm is

$$u_2 = -\frac{1}{x(1 + \tan^2\alpha)}(w_2 - x\tan\alpha),$$

yielding the output dynamics $\ddot{y} = w_2$. The LEOS algorithm assigns second-order linear stable dynamics to the output y of the form $\ddot{y} = -4\dot{y} - 4y$. Substituting $\dot{y} = -x\tan\alpha$ and $\ddot{y} = w_2$ yields an expression for w_2. Substituting this in turn into the expression for u_2 yields

$$u_2(x, y, \alpha) := \frac{1}{x(1 + \tan^2\alpha)}(4y - 3x\tan\alpha). \tag{19}$$

The final closed-loop dynamics of the system are

$$\begin{aligned} \dot{x} &= -x, \\ \dot{y} &= 4x\tan\alpha - 4y, \\ \dot{\alpha} &= \frac{4y\cos^2\alpha}{x} - 3\sin\alpha\cos\alpha \end{aligned} \tag{20}$$

for $(x, y, \alpha) \notin S_u$.

Since the control law satisfies the conditions of Theorem 10, it follows immediately that the closed-loop system is weakly internally stable, with regulated output around the limit set $\Gamma \cap \{h(x, y, \alpha) = 0\} = \{(0, 0, q\pi) \mid q \in \mathbb{Z}\}$. In fact, further analysis yields a stronger result. While the state remains well defined, the evolution of $\alpha(t)$ is given by the solution of the non-homogeneous ODE

$$\dot{\alpha} = \frac{8y(0) + 4\dot{y}(0)}{x(0)} e^{-t}\cos^2\alpha - \tfrac{3}{2}\sin 2\alpha. \tag{21}$$

The stationary points for (21) are $\alpha = \tfrac{1}{2}\pi(2q - 1)$ for q any integer. Consequently, the solution $\alpha(t)$ of (21) will remain in some bounded interval $(\tfrac{1}{2}\pi(2q - 1), \tfrac{1}{2}\pi(2q + 1))$, where

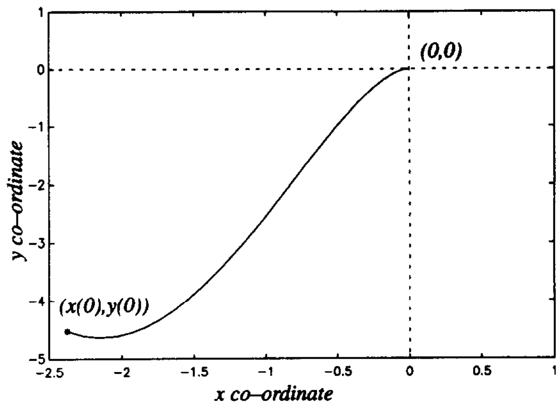

Fig. 2. Plot of the evolution of the box car robot in Euclidean 2-space for Example 1. The initial condition is $(-2.3755, -4.5254, 0.7515\pi)$.

the integer q is determined by the initial condition α_0. For any initial condition where $x \neq 0$, classical ODE theory now ensures that the solution to the closed-loop system exists for all time. The non-autonomous term in (21) is exponentially decaying with time, and the limiting dynamics will be given by the limiting dynamics of the solution of the homogeneous ODE $\dot{\alpha} = -1.5\sin 2\alpha$. It follows that $\alpha(t) \to q\pi$ as $t \to \infty$, since this is the only attractive equilibrium of the homogeneous ODE in the domain $(\tfrac{1}{2}\pi(2q - 1), \tfrac{1}{2}\pi(2q + 1))$. As required, the point $(0, 0, q\pi)$. It follows that the closed-loop system is internally stable with regulated output (cf. Definition 1).

Several simulations of the control strategy (18) and (19) have been run using the MATLAB ode45 function to integrate (20). This routine uses fourth-order Runge–Kutta routines to numerically integrate the solution trajectories while checking computational accuracy using fifth-order Runge–Kutta routines. Figure 2 displays the path of the box car robot in Euclidean 2-space for a typical example. The initial condition for this example was $(x(0), y(0), \alpha(0)) = (-2.3755, -4.5254, 0.7515\pi)$. The solution of the closed-loop system does not pass through any singular points, and the control scheme provides smooth bounded control laws.

Example 2. This example has been chosen to display the behaviour of the closed-loop system in the vicinity of singularities in the control laws. The initial condition $(x(0), y(0), \alpha(0)) = (0.0099, -1.3466, 0.5008\pi)$ was deliberately chosen to be nearly singular (both $\cos\alpha(0) \approx 0$ and $x \approx 0$). Figure 3 shows the time evolution of each component x, y and α of the state. Observe that the behaviour of the x and y coordinates is exactly as expected (despite the presence of numerical ill-condiitoning). Of course, the x coordinate is initially nearly zero, and its convergence does not show in the plot. Figure 3 also provides an excellent picture of the dynamics in the orientation α. Observe that $\alpha(t)$ remains in

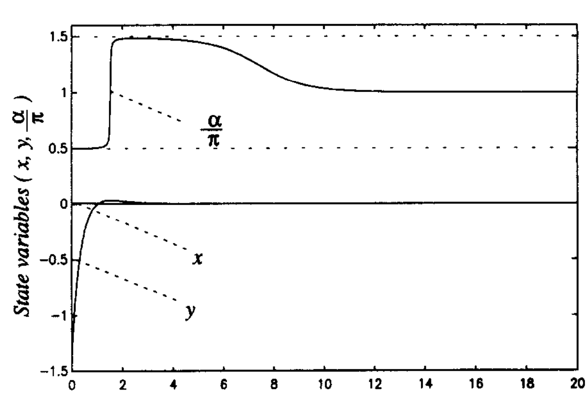

Fig. 3. Plot of the state $(x, y, \alpha/\pi)$ for Example 2.

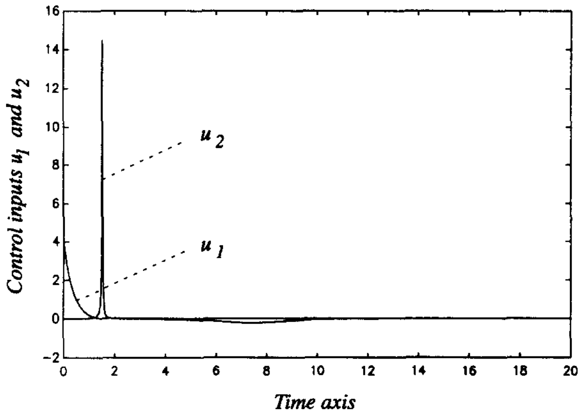

Fig. 4. Plot of the control inputs u_1 and u_2 for Example 2. The initial condition is $(0.0099, -1.3466, 0.5008\pi)$.

the interval $(\frac{1}{2}\pi, \frac{3}{2}\pi)$ for all time and that, once the non-autonomous terms of (21) have died away, $t \geq 6$, $\alpha(t) \rightarrow \pi$. At time $t = 2$ the robot appears to spin on the spot (rotating through π rad). The control input u_2 is directly linked to change in orientation of the robot and the abrupt rotation shows in Fig. 4 as a spike (of magnitude greater than +10).

6. *Conclusions*

In this paper we have discussed several issues in the design and analysis of control laws for output regulation of nonlinear systems. The main results obtained are Theorems 3 and 10 and the development of the *linearizing extended output stabilizing control algorithm* (LEOS control algorithm) presented in Section 3.

Some aspects of the control law generated by the LEOS control algorithm are summarised below.

- No assumption about the controllability of the full state for (1) is needed for the design of output stabilization control laws.
- Because of the simple structure of the algorithm, it is possible to analyse the singular structure of the algorithm. In particular, the presence of singularities resulting from non-holonomic velocity constraints is tolerated by the control law.
- With minor and obvious modifications, the case $m > p$ may be treated along similar lines. Control action that is not assigned explicitly in the algorithm can be set equal to zero or used for other purposes.

Acknowledgements—The authors would like to acknowledge the funding of the Austrllian Commonwealth Government under the Cooperative Research Centres Program. This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science Technology and Culture. The scientific responsibility rests with its authors.

References

d'Andréa-Novel, B., Bastin, G. and Campion, G. (1996) Control of nonholonomic wheeled mobile robots by state feedback linearization. *J. Robotics Res.* **4**, 543–559.

Brockett, R. W. (1983) Asymptotic stability and feedback stabilisation. In *Differential Geometric Control Theory,* ed R. W. Brockett, R. S. Millman and H. J. Sussman, pp. 181–191. Birkhäuser, Boston.

Canudas de Wit, C. and Sordalen, O. J. (1991) Exponential stabilization of mobile robots with nonholonomic constraints. In *Proc. 30th Conf. on Decision and Control,* Brighton, U.K., pp. 692–697.

Coron, J.-M. (1992) Global asymptotic stabilization for controllable systems without drift. *Math. Control, Sig. Syst.* **5**, 295–312.

Corou, J.-M. and Rosier, L. (1994) A relation between continuous time-varying and discontinuous feedback stabilization. *Math. Syst. Estim. Control,* **4**, 67–84.

Khennouf, H. and Canudas de Wit, C. (1995) On the construction of stabilizing discontinuous controllers for nonholonomic systems. In *Proc. NOLCOS-95*, Lake Tahoe, pp. 747–752.

Kolmanovsky, I. V., Reyhanoglu, M. and McClamroch, H. (1994) Discontinuous feedback stabilization of nonholonomic systems in power forms. In *Proc. 33rd IEEE Conf. on Decision and Control,* Lake Buena Vista, FL, pp. 3469–3474.

Nijmeijer, H. and Van der Schaft, A. (1990) *Nonlinear Dynamical Control Systems.* Springer-Verlag, New York.

Pomet, J.-B. (1992) Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *Syst. Control Lett.* **18**, 147–158.

Saberi, A., Kokotovic, P. V. and Sussman, H. J. (1990) Global stabilization of partially linear composite systems. *SIAM J. Control Optim.* **28**, 1491–1503.

Sanders, J. A. and Verhulst, F. (1985) Averaging Methods in Nonlinear Dynamical Systems. Springer-Verlag, New York.

Sussmann, H. (1979) Subanalytic sets and feedback controls. *J. Differential Eqns,* **31**, 31–52.

Sussmann, H. J. and Kokotovic, P. V. (1991) The peaking phenomenon and the global stabilization of non-linear systems. *IEEE Trans. Autom. Control* **AC-36**, 424–439.