Analysis and design of nonlinear control systems with the symbolic computation system MAPLE

Analysis and Design of Nonlinear Control Systems with the Symbolic Computation System MAPLE

Harm van Essen and Bram de Jager
Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands. Telephone 040-472784, Fax 040-447355. Email: jag@wfw.wtb.tue.nl

Abstract: During the last decade, progress is made in the development of analysis and design methodologies for nonlinear control systems. The non-numeric computations that are involved are very tedious. In this study the use of symbolic computation programs is treated. The treatment is restricted to an extensive and consistent discussion of the zero dynamics and the exact linearization theories. Several procedures are developed which offer computational tools and automate the analysis. Also insight in the use of symbolic computation is offered. The results established so far seem to be promising and symbolic computation is found to be a useful research tool.

Keywords: nonlinear systems, exact linearization, zero dynamics, symbolic computation, MAPLE applications

I Introduction

Design of nonlinear control systems is often based on a linear model of the system. Once a linear model is obtained, several standard methods for analysis and design of control systems are available. So, although most systems are inherently nonlinear, the majority of controller design techniques are based on linear models. These control strategies usually provide adequate performance if the system is sufficiently linear around the working point. However, if the system is highly nonlinear or its state deviates significantly from the state at which the linear model was obtained, this strategy may not be adequate. To ensure stability for a wider range of operating conditions, the controllers must be detuned and performance is degraded. In case a high performance -e.g. fast and accurate robot control, critical process control etc.- is required, controllers obtained from a linear model will not be adequate.

For highly nonlinear systems, controller design strategies using nonlinear models can be expected to provide improved performance. During the last decade progress has been made in the development of systematic design methodologies for nonlinear control systems. A comprehensive view is presented by Isidori in his book ***Nonlinear Control Systems, an Introduction*** [1] in which several problems are posed and exact (analytical) solutions are given. The methods in this book are used in our study. The non-numeric computations that are involved, have to be done by hand. Because these computations are tedious, these methods can only be applied when the computations are assisted by *Symbolic Computation* systems.

This study makes a contribution to the development of systematic design methodologies for nonlinear control systems. The contribution exists of the implementation of some analytical analysis and design strategies in a symbolic computation environment. In doing this, some specific choices have been made. First, the symbolic computation system MAPLE is used throughout this research. Second, this study is restricted to an extensive and consistent treatment of the *zero dynamics* and the *exact linearization theories*. This means in particular that stabilization and connected control objectives like tracking and attenuation will not be discussed. However, some general properties of nonlinear systems will be useful in a broader context of analysis.

The goal of this study is twofold. First, to investigate the possibilities of a modern symbolic computation program like MAPLE for research purposes and to investigate the application of this program to the development and design of (nonlinear) control systems. Second, to offer computational tools which provide facilities for analysis and design of nonlinear control systems and which will be useful in further research to employ and develop of these nonlinear control strategies.

To our knowledge, only few research is addressed in order to develop general packages for analysis and design of affine nonlinear systems. Birk and Zeitz [2], are developing a package with extensive possibilities in the Macsyma environment, but mainly directed at the design of observers. Akhrif and Blankenship [3], are known to be engaged in application of symbolic computation in nonlinear control systems.

In section II exact linearization and zero dynamics are introduced and placed in the context of control systems. Section III summarizes theoretical results and algorithms. Section IV presents the results of this study and section V contains the conclusions.

II Exact Linearization and Zero Dynamics

Some controllers are based on *exact linearization* theory. Unlike Jacobian linearization, which linearizes the system only at the nominal working point, these methods employ nonlinear transformations and nonlinear feedbacks that provide exact linearization of the model in the neighbourhood of the nominal working point. The nonlinear model can be linearized in either an *input-state* or (under some less restrictive conditions) in an *input-output* point of view. Once an exact linearized model is obtained, linear controller design techniques can be employed in an outer loop, to satisfy the control objectives.

Besides exact linearization, also other design problems for nonlinear control systems can be formulated. *Asymptotic stabilization via smooth feedback* is an interesting one. In fact, feedback stabilization is important as a preliminary step in achieving results for control problems like *(asymptotic) tracking* and *disturbance attenuation*. An important property of nonlinear (control) systems with regard to these design problems, is the notion of the *zero dynamics*. The zero dynamics are regarded as the nonlinear analog of the notion of (transmission) zeros of a transfer function (matrix) of a linear system.

The concept of zero dynamics can be viewed in two different ways, both satisfying the analogy with the (transmission) zeros of a linear system. First, the zero dynamics can be regarded as the internal dynamics imposed on the system when the constraint that the output is identically zero is enforced by an appropriate choice of input and initial state conditions. In the case of an invertible square minimal linear system, the eigenvalues corresponding to these dynamics are exactly the transmission zeros of the system transfer function matrix. The other way to approach the zero dynamics is to regard a nonlinear system rendered *maximally unobservable* via feedback. Here, the zero dynamics can be considered as the internal dynamics of the unobservable subspace/manifold of the closed-loop system. In fact these dynamics are the nonlinear analog of the "numerator dynamics" of a linear system. Again, in case of a (minimal and invertible) linear system, the corresponding eigenvalues are the transmission zeros of the system.

The two approaches to the concept of zero dynamics summarized above, are not always equivalent. There are, however, two noticeable cases in which this equivalence occurs [4]: The case of an arbitrary linear system and also the case of those nonlinear systems for which a *relative degree* can be defined.

III Nonlinear Control Theory

In this section some theoretical results concerning the analysis of nonlinear control systems are briefly summarized. First a state space description of the system that is considered will be presented. Then, the following notions will be treated: the relative degree, the normal form, the zero dynamics, exact linearization of the input-state equations, and finally exact linearization of the input-output behaviour of nonlinear systems.

Consider the multivariable nonlinear system, affine in the input u

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x) u_i \\ y_i &= h_i(x) \qquad \forall\ 1 \le i \le p \end{aligned} \tag{1}$$

Further on, to ease the presentation, $p = m$ is assumed.

The Relative Degree

A structural property of a system is the notion of the *relative degree* [1]. A MIMO system has a *vector relative degree* $\{r_1, \dots, r_m\}$ at a point x^0 of the state space if

(i) $L_{g_j} L_f^k h_i(x) = 0 \quad \forall\ x \text{ near } x^0 \quad \forall\ k < r_i - 1 \quad \forall\ 1 \le i,j \le m$

(ii) the m x m matrix A is nonsingular at $x = x^0$

$$A = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \cdots & L_{g_m} L_f^{r_1-1} h_1(x) \\ \vdots & & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(x) & \cdots & L_{g_m} L_f^{r_m-1} h_m(x) \end{bmatrix} \tag{2}$$

Note that each integer r_i is associated with the i-th output h_i of the system. When the sum of all the integers r_i is equal to the dimension n of the state space, the system is said to have a full order relative degree. Here, r_i is exactly the number of times that the i-th output $y_i(t)$ should be differentiated at $t = t^0$ in order to have at least one of the components of the input vector $u(t^0)$ explicitly appearing.

The Normal Form

By means of a change of coordinates in the state space, the system equations can be (locally) transformed to a "normal form". This normal form is of particular interest because it entails a very simple structure in the equations describing the system. In [1](prop.4.1.4) the mapping which describes the change of coordinates is presented: Suppose a system has a relative degree $\{r_1, \dots, r_m\}$ at x^0. Then

$$r_1 + \cdots + r_m \le n \tag{3}$$

Set, for $1 \le i \le m$

$$\begin{aligned} \Phi_1^i(x) &= h_i(x) \\ &\cdots \\ \Phi_{r_i}^i(x) &= L_f^{r_i-1} h_i(x) \end{aligned} \tag{4}$$

If $r = r_1 + \dots + r_m$ is strictly less than n, it is always possible to find n-r functions $\Phi_{r+1}, \dots, \Phi_n$ such that the mapping $\Phi(x)$, represented by the column

$$\mathrm{col}[\Phi_1^1(x), \cdots, \Phi_{r_1}^1(x), \cdots, \Phi_1^m(x), \cdots, \Phi_{r_m}^m(x), \Phi_{r+1}(x), \cdots, \Phi_n(x)]$$

has a Jacobian which is nonsingular at x^0 and therefore qualifies as a local coordinates transformation in a neighbourhood of x^0. The value at x^0 of these additional functions can be chosen arbitrarily. Moreover, if the distribution

$$G = \mathrm{span}\{g_1, \cdots, g_m\} \tag{6}$$

is involutive near x^0, it is always possible to choose $\Phi_{r+1}(x), \dots, \Phi_n(x)$ so that

$$\begin{aligned} &L_{g_j} \Phi_i(x) = 0 \\ &\forall\ x \text{ near } x^0 \quad \forall\ r+1 \le i \le n \quad \forall\ 1 \le j \le m \end{aligned} \tag{7}$$

It makes sense to distinguish between the first r and the last n-r new coordinates. Set:

$$\zeta^i = \begin{bmatrix} \zeta_1^i \\ \cdots \\ \zeta_{r_i}^i \end{bmatrix} = \begin{bmatrix} \Phi_1^i(x) \\ \cdots \\ \Phi_{r_i}^i(x) \end{bmatrix} \qquad \eta = \begin{bmatrix} \eta_1 \\ \cdots \\ \eta_{n-r} \end{bmatrix} = \begin{bmatrix} \Phi_{r+1}(x) \\ \cdots \\ \Phi_n(x) \end{bmatrix} \tag{8}$$

$$\zeta = (\zeta^1, \cdots, \zeta^m)$$

In this notation the normal form of the equations (1) can be written as:

$$\begin{aligned} \dot{\zeta}_1^i &= \zeta_2^i \\ &\cdots \qquad\qquad \forall\ 1 \le i \le m \\ \dot{\zeta}_{r_i-1}^i &= \zeta_{r_i}^i \\ \dot{\zeta}_{r_i}^i &= b_i(\zeta,\eta) + \sum_{j=1}^{m} a_{ij}(\zeta,\eta)\, u_j \end{aligned} \tag{9}$$

$$\dot{\eta} = q(\zeta,\eta) \tag{10}$$

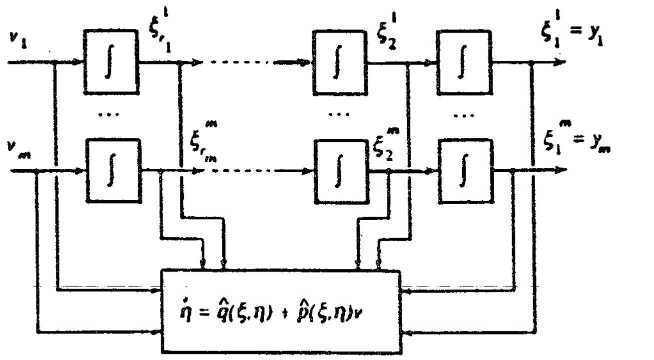
$$y_i = \zeta_1^i \qquad \forall\ 1 \le i \le m \tag{11}$$

However, if the distribution spanned by the vector fields $g_1(x)$, ..., $g_m(x)$ is not involutive or if we are not able or willing to find solutions for the set of partial differential equations (7), the inputs u_i appear affine in (10):

$$\dot{\eta} = q(\zeta,\eta) + \sum_{i=1}^{m} p_i(\zeta,\eta)u_i \tag{12}$$

In a block diagram, the special structure of the equations (9) to (11) will become clear. Here we used the static state feedback

$$v_i = b_i(\zeta,\eta) + \sum_{j=1}^{m} a_{ij}(\zeta,\eta)\, u_j \quad \forall\ 1 \le i \le m \tag{13}$$



The Zero dynamics

It is already stated that two approaches to the zero dynamics can be distinguished. Therefore, the discussion is presented in two steps:

First, the *Problem of Zeroing the Output* [1] is considered, i.e. to find all pairs of initial conditions x^0 and inputs $u^0(t)$ consistent with the constraint that the output function $y(t)$ is identically zero for all times, after which the corresponding internal dynamics of the system -which are called the zero dynamics- are analyzed.

It is not hard to deduce that if the output $y(t)$ has to be zero for all t then necessarily: The initial state (ζ^0, η^0) of the system must be set to a value such that $\zeta^0 = 0$, whereas η^0 can be chosen arbitrarily. Further the inputs $u(t)$ must be set to

$$u(t) = -[A(0,\eta(t))]^{-1} b(0,\eta(t)) \tag{14}$$

with $\eta(t)$ the solution of the differential equation

$$\dot{\eta}(t) = q(\zeta,\eta) - p(\zeta,\eta)[A(\zeta,\eta)]^{-1} b(\zeta,\eta) \tag{15}$$

and initial condition $\eta(0) = \eta^0$, where p=0 if (10) is valid.

The dynamics (15) correspond to the dynamics describing the "internal" behaviour of the system when input and initial conditions have been chosen in such a way that the output remains identically zero. These dynamics are called the *zero dynamics* of the system. Now the relevance of the equations in normal form in relation with the zero dynamics has become clear. It has been show that the zero dynamics are completely described by (10) or (12) in the coordinates η.

Second, the *Zero Dynamics Algorithm* [1] is presented. This approach is not restricted to systems having a relative degree and is therefore applicable to a broader class of systems (systems which have a relative degree are fully incorporated). For this new class of systems, the assumption of a relative degree is replaced by some milder conditions, namely the constancy of the dimensions of certain distributions and/or the rank of certain mappings. The Zero Dynamics Algorithm is an iterative procedure which constructs a locally maximal output zeroing submanifold. A physical interpretation of this algorithm is that a feedback is designed which renders the closed-loop system maximally unobservable. The internal dynamics of the unobservable part are considered to be the zero dynamics.

Exact linearization of input-state equations

In this section we show how (1) can be transformed into a linear and fully controllable system by means of static state feedback and change of coordinates in the state space. These operations are commutative.

Consider a nonlinear system having a relative degree $\{r_1, \dots, r_m\}$ satisfying $r_1 + \dots + r_m = n$. Start from the system equations in normal form (9). Obviously, the coordinates η do not appear. Introduce a new input v of the form

$$v = b(\zeta) + A(\zeta)u \tag{16}$$

The above equations can be solved for u because in the neighbourhood of $\zeta^0 = \Phi^{-1}(x^0)$ the matrix $A(\zeta)$ is nonsingular. Imposing the static state feedback

$$u = A^{-1}(\zeta)[-b(\zeta) + v] \tag{17}$$

yields a system characterized by the m sets of equations:

$$\begin{aligned} \dot{\zeta}_1^i &= \zeta_2^i \\ &\dots \\ \dot{\zeta}_{r_i-1}^i &= \zeta_{r_i}^i \\ \dot{\zeta}_{r_i}^i &= v_i \end{aligned} \qquad \forall\ 1 \le i \le m \tag{18}$$

which is clearly linear (from the new input v to the new state ζ) and controllable. The special structure of this linearized system is characterized by m chains of r_i integrators each.

From this follows that having a full order relative degree is a sufficient condition for a system to be rendered linear and controllable. In [1](lemma 5.2.1), Isidori showed this is also a necessary condition.

From the definition it is clear that the relative degree depends on the chosen output of the system. We also know that a full order relative degree is a necessary condition for the existence of a solution for the exact linearization problem. So, if we want to linearize the system we should find m output functions for which the system has a full order relative degree.

From the definition of the relative degree (2) follows that the functions $\lambda_1, \dots, \lambda_m$ we look for should be solutions of the partial differential equations:

$$L_{g_j} L_f^k \lambda_i(x) = 0 \qquad \forall\ x \text{ near } x^0 \quad \forall\ 0 \le k \le r_i - 2 \quad \forall\ 1 \le i,\ j \le m \tag{19}$$

With, as a nontriviality condition, the nonsingularity of the matrix A, and additionally the condition $r_1 + r_2 + \dots + r_m = n$. In [1], Isidori derived necessary and sufficient conditions for the existence of solutions satisfying equations (19). The conditions are stated in terms of properties of suitable distributions spanned by vector fields of the form:

$$G_0 = \text{span}\{g_1, \cdots, g_m\}$$
$$G_1 = \text{span}\{g_1, \cdots, g_m, ad_f g_1, \cdots, ad_f g_m\}$$
$$\cdots$$
$$G_i = \text{span}\{ad_f^k g_j : 0 \le k \le i, 1 \le j \le m\} \qquad i = 0, 1, \cdots, n-1$$

In [1](lemma 5.2.3) is proved: Suppose the matrix $g(x^0)$ has rank m. Then there exist a neighbourhood U of x^0 and m real-valued functions $\lambda_1, \dots, \lambda_m$ defined on U such that the system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= \lambda(x) \end{aligned} \tag{21}$$

has a relative degree $\{r_1, \dots, r_m\}$ at x^0, with

$$r_1 + r_2 + \cdots + r_m = n \tag{22}$$

if and only if:

(i) for each $0 \le i \le n-1$, the distribution G_i has constant dimension near x^0

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

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

In the proof of this result is indicated how the functions $\lambda_1(x)$, .. , $\lambda_m(x)$ can be constructed.

Exact linearization of the input-output response

In this section will be shown first what feedback linearization of the input-output mapping exactly means and then how a feedback producing a linear input-output behaviour may be designed with the so-called Structure Algorithm.

A system is said to have a linear input-output behaviour [1], if the output response can be represented in the form of a Volterra series expansion in which the first order kernels $\omega_i(t,\tau_1)$ depend only on the difference $(t-\tau_1)$ and not on x^0, and all higher order kernels are vanishing

$$y(t) = Q(t,x^0) + \sum_{i=1}^{m} \int_0^t \omega_i(t-\tau_1) v_i(\tau_1) d\tau_1 \tag{23}$$

This output response is the sum of the response under zero input and of a response depending on the input and not on the initial state, which is linear in the input itself.

Our goal is to construct a feedback in order to achieve, for a broader class of systems than those with a relative degree, such a response. Via a Taylor series expansion of $\omega_i(t,\tau_1)$ it is found [1](3.2.9) that a necessary and sufficient condition for such a response is that

$$L_{g_i} L_f^k h_j(x) = \text{independent of } x \qquad \forall\, k \ge 0 \quad 1 \le i,j \le m$$

Now the *Structure Algorithm* will be discussed. This algorithm [1] is a test for the fulfilment of the solvability condition as well as a procedure to construct a linearizing feedback. The Structure Algorithm tries to find a neighbourhood U of x^0 and a pair of feedback functions $\alpha(x)$ and $\beta(x)$ defined on U, such that

$$L_{(g\beta)_i} L_{(f+g\alpha)}^k h_j(x) = \text{independent of } x \qquad \forall\, k \ge 0 \quad 1 \le i,j \le m$$

IV MAPLE Implementation

In this section, the implementation -in the symbolic computation system MAPLE- of the algorithms mentioned in the previous section will be presented and discussed. With this implementation an attempt is made to automate the analyses for affine nonlinear systems (1). It is made possible to compute (and analyze) symbolically

- The (locally defined) relative degree.
- A change of coordinates leading to the normal form, possibly by solving the set of partial differential equations (7).
- The normal form, the zero dynamics and the corresponding unique zeroing input.
- A (locally defined) static state feedback which, if it exists, renders the system linear and controllable.
- A set of output functions which, if they exist, give the system a full order relative degree. If such an output function does not exist, an output function which maximizes the relative degree can be computed, but only for SISO systems.
- A (locally defined) static state feedback which, if it exists, renders the system linear in an input-output sense.

MAPLE is a system for symbolic mathematical computation. It has been developed since 1981 by the Symbolic Computation Group at the University of Waterloo, Canada. In essence, MAPLE is an interactive program which is designed for advanced calculus and algebra. However, MAPLE provides possibilities for programming. For this purpose an internal programming language is available. This language allows us to write our own functions and procedures, wherein all MAPLE commands and functions can be used. More information on the use and properties of MAPLE can be found in [5] and [6].

The algorithms are implemented in procedures, which are written in the MAPLE programming language and therefore only useable within MAPLE. In most cases, the procedure is a straightforward implementation of the algorithm presented and discussed. For the final implementation should be referred to [7], in which all procedures are treated extensively. (Descriptions, manual, examples, flowcharts, listings).

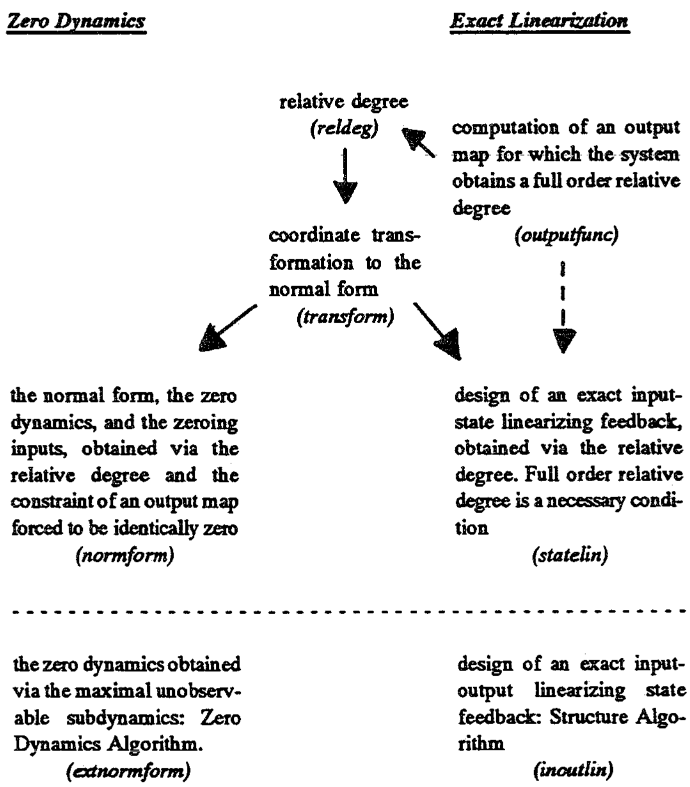
Classes in which the procedures can be ordered

In total, 19 procedures have been written. A division can be made with respect to the object of each procedure. Three classes can be distinguished.

In the first class, a number of procedures concerning basic mathematical computations like special differential operators, a test on involutivity and procedures to solve (sets) of partial differential equations are written. This is of particular interest because MAPLE does not provide facilities to solve partial differential equations and has difficulties with solving sets of ordinary differential equations. The new procedures are able to construct solutions of a special system of partial differential equations of the first order. The algorithm which solves the equations is based on the *Frobenius theorem.* In fact the solutions are constructed by means of a constructive proof of this theorem.

In the second class, a few existing (standard) MAPLE procedures, mainly concerning Linear Algebra, are adjusted and extended to fit in the present procedures. These changes are made in order to use the procedures in a broader context. The procedures are modified to handle non-rational entries of matrices (like goniometric or exponential functions). The adjustments to provide these additional possibilities are simple.

Finally, in the third class, a number of procedures concerning the analyses themselves are written, which make extensive use of the procedures mentioned in the other two parts. The procedures in this class are the main results of this study, they offer symbolic computational tools to perform analysis and design strategies for nonlinear (control) systems. The emphasis in these analyses is on an extensive and consistent treatment of the *Zero Dynamics* and the *Exact Linearization Theories*. The procedures can be ordered as in the following scheme. From this scheme, two interesting remarks can be made: In the first place the two fields of analysis can be clearly distinguished (columns left and right), but also their common foundation in the relative degree can be recognized. In the second place a clear distinction can be made (dashed line) between the analyses that are applicable on systems for which the relative degree can be defined (*normform, statelin*) and analyses applicable on a broader class of systems (*extnormform, inoutlin*). As stated, the results of both approaches are equivalent in case of a well defined relative degree.



Discussion of the use of MAPLE

Without too much difficulties, it has been possible to implement the algorithms mentioned in the MAPLE programming language. The results established so far with the procedures seem to be very promising. It is stressed, however, that additional research will be required to find the limits of the procedures and MAPLE.

Concerning the limits of the procedures, these may be in the implementation and can often be adjusted by an appropriate extension of the procedure, which provide possibilities for the special problem that is encountered. Of course, and most likely, the limits may also be in the algorithm, or in the properties of the system that is being analyzed. In most cases a suitable ERROR message is returned, provided by either MAPLE or the procedure itself. Concerning the limits of MAPLE, little can be said. The built-in possibilities of MAPLE are strictly limited (e.g. not all systems of equations can be solved).

In order to make the results of MAPLE as reliable as possible, tests are implemented which check the (intermediate) results. Inherent to the use of symbolic computation is the creation of results which contain very long, intricate expressions. The (automatic) validation of such results is seriously restricted.

V Conclusions

The collection of procedures developed in this study forms a contribution to the development of systematic design methodologies for nonlinear control systems. The procedures offer computational tools for symbolic analysis and design of nonlinear control systems. Together, the procedures provide a complete and consistent treatment of two important fields of interest in the (analytic) analysis and design of nonlinear control systems: the exact linearization theories and the zero dynamics. It is made possible to compute and analyze symbolically -for a class of (MIMO as well as SISO) systems for which the relative degree can be defined- the relative degree, the normal form, the zero dynamics, an exact linearization of the *input-state* equations, and an exact linearization of the *input-output* behaviour. Also for a broader class of systems the zero dynamics and the second forms of exact linearization can be computed symbolically. In this broader class, the systems which do have a relative degree are fully incorporated.

The procedures are well-documented in [7]. Descriptions, directions for use, examples, flowcharts, and listings are included. Also, the theoretical and mathematical backgrounds, and all relevant algorithms are included in a self-contained manner. Finally, attention is paid to (the use of) MAPLE.

The results established so far with the procedures seem to be promising, though their value should be proved in practice, when elaborate tests with realistic systems are being performed. In [7], obvious extensions of the existing procedures are proposed, in order to treat control objectives like stabilization, tracking and attenuation. It is stressed that the use of symbolic computation is not restricted to the present implementations.

As follows from the successful implementation in the MAPLE programming language, MAPLE seems to be a suitable environment for the use of analytic control strategies. A shortcoming of MAPLE is found to be the inability of solving partial differential equations. For application in the procedures developed in this thesis, these deficiencies are redressed by appropriate extension of MAPLE. Being far from perfect, it is recommended that these extensions are further analyzed and added to MAPLE in a generalized form.

VI References

[1] Isidori, Alberto; *Nonlinear Control Systems, An Introduction*, Springer-Verlag, Berlin, 2nd edition, 1989. [2] Birk, J. and Zeitz, M.; *'Anwendung eines symbolverarbeitenden Programmsystems zur Analyse und Synthese von Beobachtern für nichtlineare Systeme'*, *M.S.R*, no. 33-12, pp. 536-543, Berlin, 1990. [3] Akhrif, O. and Blankenship, G.L.; *'Computer Algebra for analysis and design of nonlinear control systems'*, *Proc. American Control Conference*, pp. 547-554, Minneapolis, 1987. [4] Byrnes, Christopher and Isidori, Alberto; *'Asymptotic Stabilization of Minimum Phase Nonlinear Systems'*, *IEEE Transactions on Automatic Control*, vol. 36, no. 10, pp. 1122-1137, October 1991. [5] Char, Bruce W. et al; *'MAPLE V Language Reference Manual'*, Springer-Verlag, New York, 1991. [6] Char, Bruce W. et al; *'MAPLE V Library Reference Manual'*, Springer-Verlag, New York, 1991. [7] Essen, Harm van; *Symbols speak louder than Numbers; Analysis and Design of Nonlinear Control Systems with the Symbolic Computation System MAPLE*, Master's thesis, report no. WFW92.061, TUE-WFW, Eindhoven, 1992.