Nonlinear Control Systems Design

NONLINEAR CONTROL SYSTEMS DESIGN

Selected Papers from the IFAC Symposium, Capri, Italy
14–16 June 1989

Edited by

A. ISIDORI

Università di Roma, "La Sapienza", Rome, Italy

Published for the

INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

by

PERGAMON PRESS

Member of Maxwell Macmillan Pergamon Publishing Corporation

OXFORD · NEW YORK · BEIJING · FRANKFURT
SÃO PAULO · SYDNEY · TOKYO · TORONTO

U.K.	Pergamon Press plc, Headington Hill Hall, Oxford OX3 0BW, England
U.S.A.	Pergamon Press, Inc., Maxwell House, Fairview Park, Elmsford, New York 10523, U.S.A.
PEOPLE'S REPUBLIC OF CHINA	Pergamon Press, Room 4037, Qianmen Hotel, Beijing, People's Republic of China
FEDERAL REPUBLIC OF GERMANY	Pergamon Press GmbH, Hammerweg 6, D-6242 Kronberg, Federal Republic of Germany
BRAZIL	Pergamon Editora Ltda, Rua Eça de Queiros, 346, CEP 04011, Paraiso, São Paulo, Brazil
AUSTRALIA	Pergamon Press Australia Pty Ltd., P.O. Box 544, Potts Point, N.S.W. 2011, Australia
JAPAN	Pergamon Press, 5th Floor, Matsuoka Central Building, 1-7-1 Nishishinjuku, Shinjuku-ku, Tokyo 160, Japan
CANADA	Pergamon Press Canada Ltd., Suite No. 271, 253 College Street, Toronto, Ontario, Canada M5T 1R5

First edition 1990

Library of Congress Cataloging in Publication Data

Nonlinear control systems design: science papers from the IFAC Symposium, Capri, Italy, 14-16 June 1989/edited by A. Isidori.— 1st ed.
p. cm.—(IFAC symposia series: 1990, no. 2)
1. Automatic control—Congresses. 2. Control theory—Congresses. 3. Nonlinear theories—Congresses. I. Isidori, Alberto II. IFAC Symposium on Nonlinear Control Systems (1989: Capri, Italy) III. International Federation of Automatic Control. IV. Series.
TJ212.2N66 1989 629.8—dc20 89–71013

British Library Cataloguing in Publication Data

Nonlinear control systems design.
1. Nonlinear control systems. Design.
I. Isidori, Aldo II. International Federation of Automatic Control III. Series
629.8'36

ISBN 0–08–037022–5

These proceedings were reproduced by means of the photo-offset process using the manuscripts supplied by the authors of the different papers. The manuscripts have been typed using different typewriters and typefaces. The lay-out, figures and tables of some papers did not agree completely with the standard requirements: consequently the reproduction does not display complete uniformity. To ensure rapid publication this discrepancy could not be changed: nor could the English be checked completely. Therefore, the readers are asked to excuse any deficiencies of this publication which may be due to the above mentioned reasons.

The Editors

Printed in Great Britain by BPCC Wheatons Ltd, Exeter

IFAC SYMPOSIUM ON NONLINEAR CONTROL SYSTEMS DESIGN

Organized by
DIS, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"
ANIPLA, Associazione Nazionale Italiana per l'Automazione

with the financial support of
CNR, Consiglio Nazionale delle Ricerche
ENEA, Comitato Nazionale per la Ricerca e lo Sviluppo dell'Energia Nucleare e delle Energie Alternative
FUB, Fondazione Ugo Bordoni
Università di Roma "La Sapienza"

Sponsored by
International Federation of Automatic Control (IFAC)
Technical Committees on Mathematics of Control, Theory, Applications

with the participation of
IEEE, Control Systems Society

International Programme Committee
A. Isidori (Chairman), Italy
J. Ackermann, FRG
H. A. Barker, UK
C. I. Byrnes, USA
H. F. Chen, PRC
J. Descusse, France
M. Fliess, France
K. Furuta, Japan
T. Glad, Sweden
B. Jakubczyk, Poland
P. Kokotovic, USA
V. Kucera, Czechoslovakia
A. J. Krener, USA
D. Q. Mayne, UK
G. Meyer, USA
S. Monaco, Italy
D. Normand-Cyrot, France
W. J. Rugh, USA
S. Sastry, USA
J. L. Speyer, USA
T. J. Tarn, USA
I. Troch, Austria
V. Utkin, USSR
A. Van der Shaft, The Netherlands
M. Vidyasagar, Canada
K. Well, FRG
A. Zinober, UK
T. Zolezzi, Italy

National Organizing Committee
S. Monaco (Chairman)
A. De Luca (Treasurer)
M. D. Di Benedetto
G. Di Pillo
A. Isidori
R. Marino
A. Schiavi

PREFACE

In the last two decades, the development of specific methodologies for the control of systems described by nonlinear mathematical models has attracted an ever increasing interest. In an earlier phase, the emphasis was mostly addressed to the understanding of basic qualitative concepts, such as controllability and observability, and to the analysis of input-output behaviour by means of functional expansions, but a relatively minor attention was addressed to problems of high importance in an engineering environment, as the design of feedback control laws. A major breakthrough occurred at the beginning of this decade, primarily due to the application of mathematical concepts derived from the field of differential geometry. Two of the main streams in the applications of these concepts to control theory are the design of feedback laws which transform a nonlinear system into an equivalent linear system (feedback linearization), and the design of feedback laws which render certain outputs independent of certain inputs (disturbance decoupling and noninteracting control). In particular, feedback linearization techniques were successfully applied to the very difficult problem of controlling an aircraft with multiaxis nonlinear dynamics. Around the mid of the decade, a further thrust was added by the appearance of new methods for the analysis of nonlinear input-output differential relations, and by the efforts to combine the results of the geometric analysis with the methods of the asymptotic analysis, primarily singular perturbations and center manifold theory. Important system theoretic concepts such as left invertibility, right invertibility, and nonlinear equivalent of transmission zeroes were successfully elaborated. This more recent stage of the development made it possible to find solutions to problems as noninteracting control with stability, asymptotic tracking and output regulation, but also made it clear how much the intrinsically more elaborate nature of a nonlinear control system could be an obstruction to the solutions of problems which are well understood in a linear environment.

Theory and applications are still gradually developing. There are limitations, open problems and unknowns. The fundamental issue of stabilizing a nonlinear system by means of state feedback continues to attract a major interest, and general conditions for the existence of stabilizing feedback laws are gradually appearing. One area of research, already initiated, which apparently will receive an increasing attention in the years to come is that of combining the design technique developed so far, which require exact knowledge of the system under control, with appropriate adaptation philosophies which could take into account unknown parameters and unmodeled dynamics. All the design methods which are presently available require, more or less, accurate knowledge of the state of the process, while no satisfactory theory for the design of nonlinear observers is available. Even a suitable nonlinear analogue of the separation principle still needs to be developed. The theory of nonlinear dynamic compensation is only in a very preliminary stage, especially if design requirements concerned with the asymptotic behaviour must be taken into account.

This Volume contains a wide selection from the papers presented at the IFAC Symposium held in Capri, 14-16 June 1989, by researchers of more than 15 different countries. The empahsis is on the methodological developments, although thirteen of the seventy selected papers are concerned with the presentation of applications of nonlinear design philosophies to actual control problems in Chemical, Electrical and Mechanical Engineering. The main subjects addressed are the following ones: Algebraic and Geometric Methods in Nonlinear Control Theory, Discrete Time Systems, Input-Output Methods of Analysis and Design, Stability and Regulation, Adaptive Control, Variable Structure Systems, Differential Inclusions, Optimal Control.

There are numerous people who helped organizing this Symposium. Particular thanks deserve the members of the International Program Committee, for their help to set up the program and all the reviewers, whose very accurate and timely responses were instrumental in paper evaluations. Under the direction of Prof. S. Monaco, the members of the National Program Committee, in particular the colleagues A. De Luca and M.D. Di Benedetto, together with C. Dollari of the Dipartimento di Informatica e Sistemistica of the Università "La Sapienza", secured an invaluable support.

Alberto Isidori
IPC-Chairman

CONTENTS

ALGEBRAIC METHODS IN NONLINEAR CONTROL THEORY

GEOMETRIC METHODS IN NONLINEAR CONTROL THEORY

DISCRETE-TIME SYSTEMS

INPUT-OUTPUT METHODS OF ANALYSIS AND DESIGN

STABILITY AND REGULATION

ADAPTIVE CONTROL OF NONLINEAR SYSTEMS

VARIABLE STRUCTURE SYSTEMS

DIFFERENTIAL INCLUSIONS

OPTIMAL CONTROL

NONLINEAR CONTROL OF CHEMICAL SYSTEMS

COMPUTER-AIDED DESIGN OF NONLINEAR OBSERVERS

J. Birk and M. Zeitz

*Institut für Systemdynamik und Regelungstechnik—Universität Stuttgart,
D-7000 Stuttgart 80, Pfaffenwaldring 9, FRG*

Abstract. Recently developed methods for the design of nonlinear observers require extensive analytical calculations which are not practicable without computer assistance. For the extended Luenberger observer the step by step design with the aid of a computer using the symbolic programming language MACSYMA is shown. In addition to the calculating functions, the developed program for the design of nonlinear observers contains some user interface and input/output functions, e.g. for an ACSL simulation of the observer equations. Corresponding to the design steps the programmed calculating functions form a hierarchy of basic mathematical, analysis, and design functions. These functions can be simply extended and used for the solution of other problems, e.g. for the nonlinear controller design. The application of the program is demonstrated for the computer-aided design of an extended Luenberger observer for a continuous stirred tank reactor.

Keywords. Nonlinear systems, observers, extended Luenberger observer, computer-aided design, symbolic calculation, MACSYMA.

INTRODUCTION

For the computer-aided analysis and design of nonlinear systems, symbolic programming languages such as MACSYMA, REDUCE, MUMATH or SMP can be used. Analytical and numerical calculations are thereby realised in a CACSD-program. First experiences with the language REDUCE for the controllability analysis of elastic robots and the stabilisation of an electrical power system have been presented by Cesaro and Marino (1984 a, 1984 b). The program system CONDENS - Control design of nonlinear systems - written in MACSYMA has been introduced by Akhrif and Blankenship (1987). CONDENS contains modules for linearization and inversion of nonlinear systems with application to a controller design.

A computer-aided design of nonlinear observers has been described by Bär, Fritz, and Zeitz (1987) for the first time; the MACSYMA program uses the two-step-transformation proposed by Keller (1987) to linearise the observer error equations and to apply an eigenvalue assignment. The nonlinear observer design program considered in this paper is based on these experiences. Special emphasis is placed on the design of extended Luenberger observers, which are applicable for a relatively wide class of nonlinear systems (Zeitz, 1987; Birk and Zeitz, 1988). In the following, the essential design steps for this observer are explained before the program and its application to a chemical reactor are described.

NONLINEAR OBSERVABILITY PROBLEM

Consider the nonlinear system

$$\dot{x} = f(x,u) \quad t > 0 \ , \quad x(0) = x_0 \ , \tag{1}$$

$$y = h(x,u) \tag{2}$$

where the state x is an n-vector, the input u is a p-vector and the output y is a q-vector. The nonlinear functions $f(x,u)$ and $h(x,u)$, and the input $u(t)$ are assumed to be real and sufficiently smooth. Introducing time t as an additional input u_{p+1} with $\dot{u}_{p+1} = 1$, Eqs. (1), (2) include also time-variable systems.

In order to investigate the nonlinear observability problem (1), (2), the observability map is used for the analysis of the relationship between the measured input/output and the unknown state (Zeitz, 1984). The nonlinear observability map

$$\bar{y} := \begin{pmatrix} y \\ \dot{y} \\ \vdots \\ \overset{(n-1)}{y} \end{pmatrix} = \begin{pmatrix} M_f^0 \\ M_f \\ \vdots \\ M_f^{n-1} \end{pmatrix} h(x,u) =: q(x,\bar{u}) \tag{3}$$

with $M_f h := \frac{\partial h}{\partial x} f + \frac{\partial h}{\partial \bar{u}} \dot{\bar{u}}$, $M_f^0 h := h$,

$$\bar{u} := \left(u^T, \dot{u}^T, \ldots, \overset{(n-1)}{u^T}\right)^T$$

consists of the first $(n-1)$ time derivatives of the output equation (2) in which the differential equation (1) is inserted. The system of nonlinear equations (3) has a unique solution x if the inverse map $q^{-1}(\bar{y},\bar{u})$ exists. In practice, this can only be proved by an actual determination of $q^{-1}(\bar{y},\bar{u})$. For a local analysis of the relation between $\bar{u}$ and $\bar{y}$ on the one side and x on the other side, the Jacobian matrix of the observability map (3) with respect to x is used (Hermann and Krener, 1977; Nijmeijer, 1981; Krener and Isidori, 1983; Bestle and Zeitz, 1983; Zeitz, 1984; Krener and Respondek, 1985; Krener, 1985)

$$Q(x,\bar{u}) := \frac{\partial q(x,\bar{u})}{\partial x} = \begin{pmatrix} N_f^0 \\ N_f \\ \vdots \\ N_f^{n-1} \end{pmatrix} \frac{\partial h}{\partial x} \tag{4}$$

$$\text{with} \quad N_f r^T := r^T \frac{\partial f}{\partial x} + f^T \frac{\partial r}{\partial x} + \dot{\bar{u}}^T \left(\frac{\partial r}{\partial \bar{u}}\right)^T ,$$

$$N_f^0 r^T := r^T \ , \quad r^T = \frac{\partial h_i}{\partial x} \ , \quad i = 1(1)q \ .$$

In comparison to the computation of $q^{-1}(\bar{y},\bar{u})$, the rank test of the observability matrix (4)

$$\text{rank} \quad Q(x,\bar{u}) = n \tag{5}$$

is simpler. Following condition (5), there exists a one-to-one correspondence of the input/output and the state x of the system in the neighbourhood of some points x and $\bar{u}$. The corresponding property of the system which is called "locally weakly observable" by Hermann and Krener (1977) and "locally distinguishable" by Krener (1985), is an essential condition for the design of a nonlinear observer.

EXTENDED LUENBERGER OBSERVER

A nonlinear observer for the system (1), (2) can be determined in a very simple way, if the system can be converted by means of a nonlinear transformation

$$x = w(x^*,\bar{u}) \ , \quad x^* = w^{-1}(x,\bar{u}) \tag{6}$$

to a general nonlinear observer form (Zeitz, 1989):

$$\dot{x}^* = Ax^* - a(y,u^*) \ , \tag{7}$$

$$y = h^*(Cx^*,u) \quad \longleftrightarrow \quad Cx^* = h^{*-1}(y,u) \tag{8}$$

$$\text{with} \quad u^* := \left(u^T, \dot{u}^T, \dots, \overset{(n)}{u}{}^T\right)^T .$$

This general nonlinear observer form is characterized by the nonlinear expressions $a(y,u^*)$, which depend only on the input and the output, and by the particular output equation (8), which can be converted to a linear form with respect to the new state variables x^* . In these coordinates, the nonlinearities $a(y,u^*)$ can be realized by an input/output injection in the observer differential equation

$$\dot{\hat{x}}^* = A\hat{x}^* - a(y,u^*) + G^*[h^{*-1}(y,u) - C\hat{x}^*] \ , \tag{9}$$

so that there are no more nonlinear expressions in the differential equation of the observer error $\tilde{x}^* = \hat{x}^* - x^*$

$$\dot{\tilde{x}}^* = (A - G^*C)\tilde{x}^* \ . \tag{10}$$

Moreover, if the matrices A and C are set up in an observer canonical form, the transient behaviour of the observer and the gain matrix G^* can be dimensioned very easily by eigenvalue assignment through the coefficients p_i , $i = 0(1)n-1$ of the characteristic polynomial (Krener and Isidori, 1983; Bestle and Zeitz, 1983; Krener and Respondek, 1985; Keller, 1987)

$$det(\lambda I - A + G^*C) = \lambda^n + p_{n-1}\lambda^{n-1} + \dots + p_1\lambda + p_0 \ . \tag{11}$$

This design method supposes that the transformation (6) and the nonlinearities $a(y,u^*)$ and $h^*(Cx^*,u)$ can be calculated by an integration of the partial differential equations of this type

$$\frac{\partial w}{\partial x^*} = \Phi_1(x^*,\bar{u}) \ , \quad \frac{\partial a}{\partial y}\frac{\partial h^*}{\partial Cx^*} = \Phi_2(x^*,u^*) \ . \tag{12}$$

The right hand sides $\Phi_1(x^*,\bar{u})$ and $\Phi_2(x^*,u^*)$ which have been derived by Zeitz (1987) depend on the inverse observability matrix (4), whose regularity must therefore be postulated. The integration of the nonlinear partial differential equations (12) is possible only in a few exceptional cases. Krener and Respondek (1985), and Keller (1987) have formulated criteria for the nonlinear functions of the observability normal form which is based on the observability map (3) in order to prove the existence of the observer normal form for a given nonlinear system. These very strong conditions for the design using the nonlinear observer canonical form are alleviated for the extended Luenberger observer (Zeitz, 1987; Birk and Zeitz, 1988)

$$\dot{\hat{x}} = f(\hat{x},u) + G(\hat{x},u^*)[y - h(\hat{x},u)] \tag{13}$$

by the following modifications.

- In canonical coordinates, the observer is set up as

$$\dot{\hat{x}}^* = A\hat{x}^* - a(\hat{y},u^*) + G^*[y - \hat{y}] \ , \tag{14}$$

$$\hat{y} = h^*(C\hat{x}^*,u) \tag{15}$$

  rather than by the observer differential equation (9). The nonlinearity $a(y,u^*)$ is not injected but realized as $a(\hat{y},u^*)$. Moreover, the observer correction depends on the difference between the measured and the computed output.

- The differential equation of the observer error formed by the difference of (14) and (7)

$$\dot{\tilde{x}}^* = A\tilde{x}^* - a(\hat{y},u^*) + a(y,u^*) + G^*[h^*(Cx^*,u) - h^*(C\hat{x}^*,u)] \ ,$$

  is linearized around the reconstructed trajectory $\hat{x}^*(t)$

$$\dot{\tilde{x}}^* = \left[A - \frac{\partial a}{\partial y}\frac{\partial h^*}{\partial C\hat{x}^*}C - G^*\frac{\partial h^*}{\partial C\hat{x}^*}C\right]\tilde{x}^* + O(|\, C\tilde{x}^* \,|^2) \ . \tag{16}$$

- Since the time variable differential equation (16) is in observer canonical form, it can be converted by an appropriate choice of G^* to a time invariable form and can be dimensioned by eigenvalue assignment.

- In the design equations for G^* and $G = \left(\frac{\partial w}{\partial x^*}\right)^{-1} G^*$ the partial differential equations (12) can be substituted without a previous integration.

In the case of single output ($q = 1$, $Cx^* = x_n^*$), these design steps for the extended Luenberger observer lead to the following dimensioning formula for the observer gain vector (Zeitz, 1987; Birk and Zeitz, 1988)

$$g(\hat{x},u^*) = \left\{\left[p_0 L_f^0 + \ldots + p_{n-1} L_f^{n-1} + L_f^n\right] s\right\} \left(\frac{\overline{\partial h^*}}{\partial \hat{x}_n^*}\right)^{-1} \quad (17)$$

$$\text{with} \quad L_f s := \frac{\partial f}{\partial x}\, s - \frac{\partial s}{\partial x}\, f - \frac{\partial s}{\partial \bar{u}}\, \dot{\bar{u}} \;, \quad L_f^0 s = s \;,$$

$$\text{and} \quad s = Q^{-1}(\hat{x},\bar{u}) \left(0,\ldots,0, \frac{\overline{\partial h^*}}{\partial \hat{x}_n^*}\right)^T \;.$$

The only condition for the application of this design method is the regularity of the observability matrix $Q(x,\bar{u})$ in the operating range of x and $\bar{u}$. The evaluation of formula (17) can be simplified by an appropriate choice of function $\overline{\partial h^*}/\partial x_n^*(x,\bar{u}) \neq 0$ such that the applications of the differential operators L_f to the so-called starting vector s lead to simple expressions. The dimensioning rule (17) contains as a special case Ackermann's formula (1983) for linear systems. If the stationary solution for an operating point x_s, u_s with $f(x_s,u_s) = 0$ is substituted into the dimensioning rule (17), a constant observer gain is received. The same result is obtained if the formula of Ackermann is applied to the system equations which are linearized around the operating point.

PROGRAM FOR THE COMPUTER-AIDED OBSERVER DESIGN

A program system has been developed for the computer-aided application of the described methods for the observability analysis and the observer design to nonlinear systems. The program system uses the symbolic programming language MACSYMA on a Symbolics-AI-Workstation and contains modules with user interface, input/output and calculating functions as shown in Fig. 1. The calculating functions, which form the basis of the program, can be divided into three categories: basic mathematical, analysis, and design functions. In every category, the calculating functions are sorted according to the corresponding problems. For example, the design functions are divided in state estimator design and transformation in canonical forms. For state estimator design, the program contains the design of the canonical form observer, the extended Luenberger observer, the operating point observer, and the extended Kalman filter. For an observability analysis of nonlinear systems, the global observability, the local weak observability, and the operating point observability can be proved analytically.

The analysis and design functions are realized by means of basic mathematical and MACSYMA functions. If there are no corresponding MACSYMA functions, necessary calculating operations are programmed as mathematical basic functions. Typical examples are the computation of a Jacobian matrix or the directional differentiations of scalars, row and column vectors in a vector field according to Eqs. (3), (4), and (17).

The hierarchy of the design, analysis, and mathematical basic functions represented in Fig. 1 corresponds to the mathematical hierarchy of the design steps for the nonlinear observer problem. Moreover, this hierarchy of calculating functions allows a simple extension of the program for other problems, e.g. the controllability analysis and the design of nonlinear controllers. A similar level structure is implemented in the already discussed program CONDENS (Akhrif and Blankenship, 1987).

In addition to the calculating functions, the program system contains user interface and input/output functions according to Fig. 1. The input/output functions, which can be called from the calculating functions as well as from the user interface functions, can be used to enter, to change, or to save system equations, to save program sequences or to plot analytical results. Moreover, the MACSYMA function, which generates FORTRAN code, is used to convert differential equations to a file, which can be directly used for the numerical simulation program ACSL. In some I/O-functions there are several functions which are realized in a second level as represented in Fig. 1. Such a second level exists for the system input, plot, and simulation functions.

The user interface of the program is realized by mouse-supported menus. The implemented calculating and I/O-functions can be called and used without detailed knowledge of the programming language and of the applied system theoretical methods, because the user is prompted in all functions for the necessary input. The lines between the modules of the program in Fig. 1 characterize the possible program sequence which can be chosen by the user. Moreover, the menu-controlled program sequence can be interrupted at every point to work interactively with MACSYMA without losing the results already calculated. In this lowest program level, elementary operations or functions not available in a higher level can be executed directly by using MACSYMA functions. The simplification of complicated expressions by factoring out or reduction is one example. After finishing the MACSYMA calculations, the program sequence can be continued at the interrupt point.

EXAMPLE OF AN OBSERVER DESIGN

The computer-aided design of an extended Luenberger observer with the developed MACSYMA program is demonstrated for the model of a continuous stirred tank reactor

$$\dot{x} = \begin{pmatrix} -a_1 x_1 + r(x) \\ -a_{21} x_2 + a_{22} r(x) + bu \end{pmatrix} \quad (18)$$

$$\text{with} \quad r(x) = [(1-x_1) + \mu(1-x_1)^\nu] k_0 \; \exp\left(-\frac{\epsilon_1}{1+x_2}\right) \;,$$

$$y = x_2 \;.$$

In these equations, the normalized concentration x_1 and the normalized temperature x_2 are used and the nonlinear reaction rate $r(x)$ is valid for an exothermal decay reaction. Fig. 2 shows the handling and the output of the program as an extract of the saved program sequence. By starting the program with the function call "Nonlinear_observer_design()" the main menu is displayed. Selecting the I/O-functions, a menu appears with the input/output functions presented in Fig. 1. By menu-controlled choice of the saved system "stirred_tank_reactor", this system is loaded and displayed in (E3) and (E4). With the I/O-function "change_system" the system could be modified. After returning from the I/O-functions, the main menu appears again, from which the function "Ext_Luenberger_observer" can be called via several menus according to the program structure in Fig. 1. The function "Ext_Luenberger_observer" computes first of all the observability matrix (4) and displays the determinant of this matrix in (E5). At this point, the user has to decide whether the state-dependent determinant is different from zero in the considered domain of x. This necessary and sufficient condition for the design of an extended Luenberger observer is satisfied for the stirred tank reactor because of $\mu > 0$, $\nu > 0$, and $x_1 \leq 1$. In order to obtain an observer gain (17) as simply as possible, the starting vector s must be made as simple as possible by an appropriate choice of the function $\overline{\partial h^*}/\partial x_n^*(x, \bar{u})$. The program proposes an expression for this function in (E6) which leads to the constant starting vector (E7). After confirmation of this proposal, the gain (17) of the extended Luenberger observer is calculated and displayed in (E8).

The observer design for this nonlinear system, which is very extensive by hand, requires a CPU-time of about 15 seconds on a Symbolics-3620-Workstation.

In order to compare the transient behaviour of the extended Luenberger observer with that of a linearly designed observer, an operating point observer is also generated using the calculating function "operating_point_observer". The transient behaviour of these observers is simulated by use of the simulation language ACSL on an IBM PC, whereby the ACSL file is automatically generated by the output function "simulation ACSL".

The simulation results of the system, the extended Luenberger observer, and the operating point observer are shown in Fig. 3. The extended Luenberger observer has a better convergence behaviour if the characteristic coefficients (11) are equal to those of the operating point observer. Further simulation studies also show a better damping of measurement noise by the extended Luenberger observer in comparison to the operating point observer. This measurement noise damping can be influenced by the observer eigenvalues.

CONCLUSIONS

The experiences obtained with the MACSYMA CACSD-program can be summarized as follows.

- A symbolic calculating program is a very important tool for the practical application of design methods for nonlinear observers. The extensive calculations are done quickly and yield correct results.
- Explicit and recursive calculations like differentiations and matrix inversions are handled without any difficulties by MACSYMA, but very large terms of solution can be produced. Implicit and non-recursive calculations, like the inversion of the observability map cannot always be handled even if there exists a solution. In such cases the knowledge of the program user is also required. The interactive facilities of such a program are therefore very important.
- The interface between the MACSYMA program and the numerical simulation language ACSL is a very feasible tool for use in the study of symbolic design results, such as comparison of different observer design methods.

REFERENCES

Ackermann, J. (1983). Abtastregelung. 2. Edition, Vol. 1 Springer-Verlag, Berlin, Heidelberg, New York.

Akhrif, O., and G.L. Blankenship (1987). Computer algebra for analysis and design of nonlinear control systems. Proceedings American Control Conference, Minneapolis, 547-554.

Bär, M., H. Fritz, and M. Zeitz (1987). Rechnergestützter Entwurf nichtlinearer Beobachter mit Hilfe einer symbolverarbeitenden Programmiersprache. Automatisierungstechnik, 35, 177-183.

Bestle, D., and M. Zeitz (1983). Canonical form observer design for non-linear time-variable systems. International Journal of Control, 38, 419-431.

Birk, J., and M. Zeitz (1988). Extended Luenberger observer for nonlinear multivariable systems. International Journal of Control, 47, 1823-1836.

Cesaro, G., and R. Marino (1984 a). On the controllability properties of elastic robots. In A.V. Balakrishnan, and M. Thoma (Eds.), Lecture Notes in Control and Information Sciences, 63. Springer, Berlin. pp. 352-363.

Cesaro, G., and R. Marino (1984 b). The use of symbolic computation for power system stabilization: An example of computer aided design. In A.V. Balakrishnan, and M. Thoma (Eds.), Lecture Notes in Control and Information Sciences, 63. Springer, Berlin. pp. 598-611.

Hermann, R., and A.J. Krener (1977). Nonlinear controllability and observability. IEEE Transactions on Automatic Control, AC-22, 728-740.

Keller, H. (1987). Non-linear observer design by transformation into a generalized observer canonical form. International Journal of Control, 46, 1915-1930.

Krener, A.J. (1985). $(Ad_{f,g}), (ad_{f,g})$ and locally $(ad_{f,g})$ invariant and controllability distributions. SIAM Journal of Control and Optimization, 23, 523-549.

Krener, A.J., and A. Isidori (1983). Linearization by output injection and nonlinear observers. Systems & Control Letters, 3, 47-52.

Krener, A.J., and W. Respondek (1985). Nonlinear observers with linearizable error dynamics. SIAM Journal of Control and Optimization, 23, 197-216.

Nijmeijer, H. (1981). Observability of a class of nonlinear systems: A geometric approach. Ricerche di Automatica, 12, 50-68.

Zeitz, M. (1984). Observability canonical (phase-variable) form for non-linear time-variable systems. International Journal of Systems Sciences, 15, 949-958.

Zeitz, M. (1987). The extended Luenberger observer for nonlinear systems. Systems & Control Letters, 9, 149-156.

Zeitz, M. (1989). Canonical forms for nonlinear systems. In this volume.

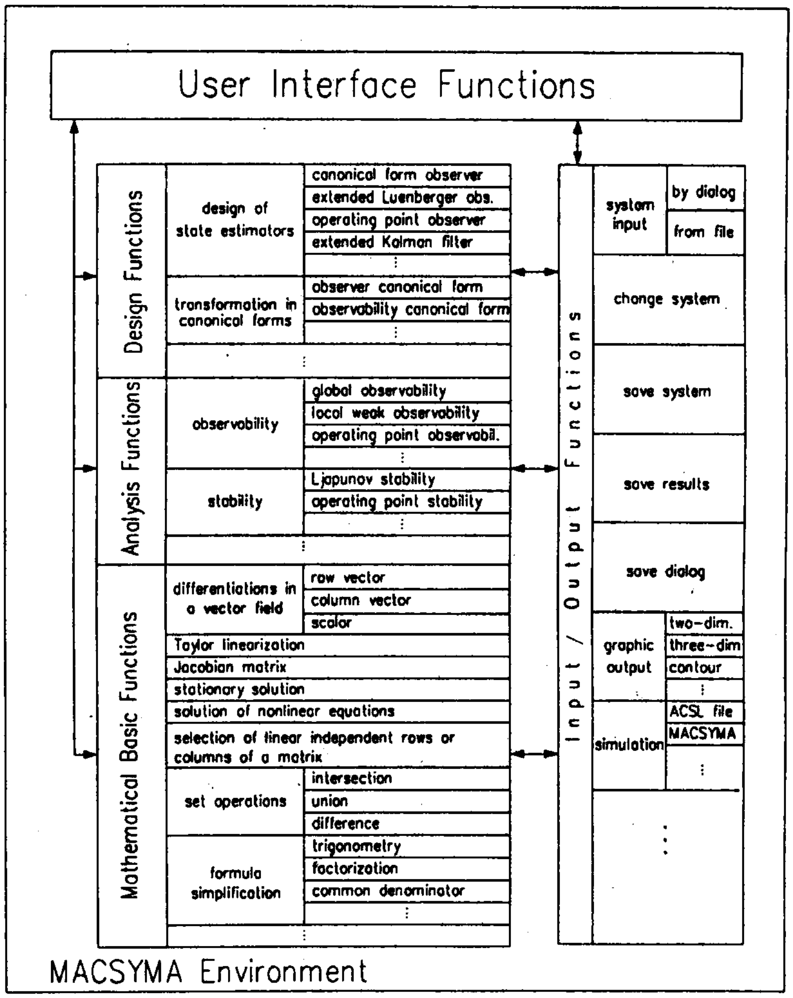


Fig. 1. Program structure with user interface, input / output, and calculating functions in MACSYMA environment.

Fig. 2. See next page.

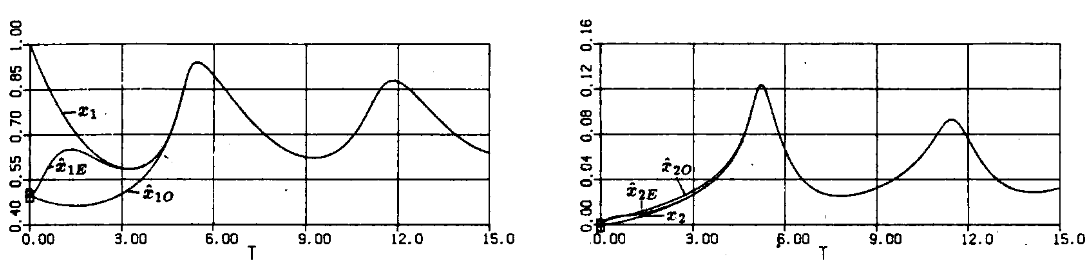
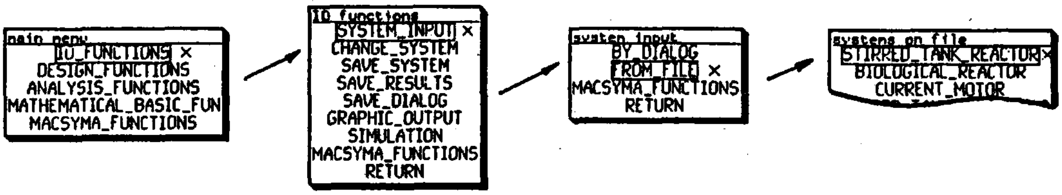


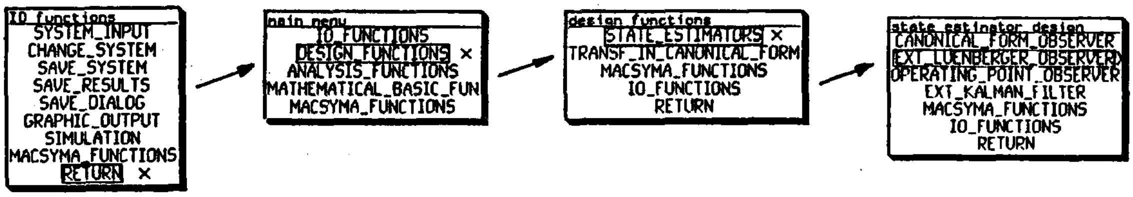
Fig. 3. Simulation results of the continuous stirred tank reactor (x), the extended Luenberger observer ($\hat{x}_E$), and the operating point observer ($\hat{x}_O$). (Parameters: $a_1 = 0.2674$, $a_{21} = 1.815$, $a_{22} = 0.4682$, $b = 1.5474$, $\mu = 0.8$, $\nu = 6.5$, $k_0 = 1.05E14$, $\epsilon_1 = 34.2894$, $u = 0$, $\lambda_1 = \lambda_2 = -2.5$).

```
(C3)  Nonlinear_observer_design();
```




```
The system  stirred_tank_reactor  is loaded.
                                                              E1
                  [                                         - ------                 ]
                  [                              NU           X2 + 1                 ]
                  [       K0 (- X1 + MU (1 - X1)    + 1) %E              - A1 X1     ]
(E3)      X_DOT = [                                                                  ]
                  [                                             E1                   ]
                  [                                           - ------               ]
                  [                            NU             X2 + 1                 ]
                  [ A22 K0 (- X1 + MU (1 - X1)    + 1) %E             - A21 X2 + B U1 ]

(E4)          Y = [ X2 ]
```




```
The structure of the ext. Luenberger observer is  dx^/dt = f(x^,u) + G(x^,u*)(y-h(x^,u)).
The computation of the ext. Luenberger obs. gain G(x^,u*) requires full rank of the obser-
vability matrix, the vector u* contains the input u and the first n time derivatives of u.
The determinant of the observability matrix reads:
                                                                   E1
                                                                - ------
                                                  NU - 1          X2 + 1
(E5)       DETQB  = - A22 K0 (- MU NU (1 - X1)          - 1) %E

The ext. Luenberger observer can be designed in these domains where DetQb is not equal to 0.

In order to simplify the observer gain, the starting vector internally used should be as
simple as possible. As a free parameter the partial derivative of the canonical output non-
linearity can be chosen arbitrarily (not equal to 0). Do you agree with the default  (Y/N)
                                                                 E1
                                                              - ------
                                                NU - 1          X2 + 1
(E6)      DHJ_STAR  = A22 K0 (- MU NU (1 - X1)        - 1) %E

                                   [ 1 ]
(E7)      STARTING_VECTOR_ELO =    [   ]
                                   [ 0 ]
YES;
The gain matrix G(x,u*) of the extended Luenberger observer is calculated as

(E8) MATRIX([(-E1*K0*(-MU*NU*(1-X1)^(NU-1)-1)*%E^-(E1/(X2+1))*(A22*K0*(-X1+MU*(1-X1)^NU+1)
     *%E^-(E1/(X2+1))-A21*X2+B*U1)/(X2+1)^2-K0*MU*(NU-1)*NU*(1-X1)^(NU-2)*%E^-(E1/(X2+1))*
     (K0*(-X1+MU*(1-X1)^NU+1)*%E^-(E1/(X2+1))-A1*X1)+(K0*(-MU*NU*(1-X1)^(NU-1)-1)*%E^-(E1/
     (X2+1))-A1)^2+PC12*(K0*(-MU*NU*(1-X1)^(NU-1)-1)*%E^-(E1/(X2+1))-A1)+A22*E1*K0^2*(-MU*
     NU*(1-X1)^(NU-1)-1)*(-X1+MU*(1-X1)^NU+1)*%E^-(2*E1/(X2+1))/(X2+1)^2+PC11)*%E^(E1/(X2+
     1))/(A22*K0*(-MU*NU*(1-X1)^(NU-1)-1))],[(A22*K0*(-MU*NU*(1-X1)^(NU-1)-1)*%E^-(E1/(X2+
     1))*(A22*E1*K0*(-X1+MU*(1-X1)^NU+1)*%E^-(E1/(X2+1))/(X2+1)^2-A21)-A22*E1*K0*(-MU*NU*(
     1-X1)^(NU-1)-1)*%E^-(E1/(X2+1))*(A22*K0*(-X1+MU*(1-X1)^NU+1)*%E^-(E1/(X2+1))-A21*X2+B
     *U1)/(X2+1)^2-A22*K0*MU*(NU-1)*NU*(1-X1)^(NU-2)*%E^-(E1/(X2+1))*(K0*(-X1+MU*(1-X1)^NU
     +1)*%E^-(E1/(X2+1))-A1*X1)+A22*K0*(-MU*NU*(1-X1)^(NU-1)-1)*%E^-(E1/(X2+1))*(K0*(-MU*N
     U*(1-X1)^(NU-1)-1)*%E^-(E1/(X2+1))-A1)+A22*K0*PC12*(-MU*NU*(1-X1)^(NU-1)-1)*%E^-(E1/(
     X2+1)))*%E^(E1/(X2+1))/(A22*K0*(-MU*NU*(1-X1)^(NU-1)-1))])

The coefficients PCij of the characteristic polynom can be chosen by pole placement. The
initial conditions of the observer x^(0) should be chosen such that h(x^(0),u(0)) = y(0).
```

Fig. 2. Dialogue for the computer-aided design of an extended Luenberger observer for a continuous stirred tank reactor (Ci – MACSYMA command line, Ei – MACSYMA output).

SHOULD THE THEORIES FOR CONTINUOUS-TIME AND DISCRETE-TIME LINEAR AND NONLINEAR SYSTEMS REALLY LOOK ALIKE?

M. Fliess

Laboratoire des Signaux et Systèmes, C.N.R.S.-E.S.E., Plateau du Moulon, 91192 Gif-sur-Yvette Cedex, France

Abstract. We claim by exhibiting several examples that the relationship between discrete-time and continuous-time systems should be re-examined. The theoretical explanations provide a renewed understanding of discrete-time state-variable description, which employs difference algebra.

I. INTRODUCTION

It is nowadays a common belief that continuous-time and discrete-time linear systems should enjoy almost identical theoretical approaches. This principle is of course also widespread for nonlinear systems, where the two most popular viewpoints for the continuous-time case, namely, differential geometry and functional expansions (cf. Isidori [18]), have been extended to discrete time [7, 15-17, 19, 20, 25, 27-32, 34-40](1). Our aim in this note is to convince the reader that this analogy may sometimes be misleading.

We recall, for example, that one of the main justifications for introducing state feedbacks is to avoid derivatives of the output. This need certainly disappears for discrete-time systems if it can be shown that derivatives are replaced by delays. A more abstract reason for questioning the usual state-space formalism, which goes back to Kalman [22], is the following.

The pure integrator $\dot{y}(t) = u(t)$ and the pure delay $y(t+1) = u(t)$ have the same transfer functions and analogous minimal state-space realizations of dimension one. However, contrary to the discrete-time case, the knowledge of the control is not enough to compute the output of the continuous-time integrator, since the initial condition must also be known. This can be explained in mathematical language by saying that the delay operator is injective, i.e., possesses a trivial kernel, but not the derivation, the kernel of which is the set of constants. Such an important feature is completely overlooked by the traditional Laplace transform and therefore by the transfer function formalism.

The preceding difficulties are overcome when employing difference algebra (Cohn [6]), which has been proved to be a most effective tool for studying discrete-time systems [8, 13] in the same manner as differential algebra helped the understanding of continuous-time systems [9, 11]. We first give an overview of our formalism, especially regarding linear and nonlinear realization. We then show with various examples how the classical realization theory leads to unsatisfactory answers from a logical viewpoint, and how to remedy this with our approach.

We must repeat what we already wrote in [13]: the complex relationship between continuous-time and discrete-time systems can only be understood by studying the connection between differential and difference algebras. A further illustration is given by looking at the feedback linearization of discrete-time dynamics and comparing it with what has been achieved in the continuous-time case [14].

Acknowledgements The author would like to thank the audience of the Control Seminar at the Ecole des Mines at Fontainebleau, which, by asking him acute questions, made him aware of the antimony between continuous-time and discrete-time systems.

II. DIFFERENCE ALGEBRA AND DISCRETE-TIME SYSTEMS

II.1. A short summary of difference algebra(2)

II.1.1. Both *differential algebra* and *difference algebra* were created between the two World Wars by the American mathematician J.F. Ritt. His aim was to provide for differential and difference equations a toolkit which would be as powerful as is *commutative algebra* for algebraic equations. It should be remembered that this was the time when commutative algebra began to mature thanks to the work of several famous German mathematicians (Dedekind, Kronecker, Hilbert, E. Noether, etc.). The books by Kolchin [24] and Cohn [6] give good surveys of these new disciplines.

II.1.2. A *difference field* K is a commutative field which is equipped with a necessarily injective morphism $\delta: K \to K$, called *transformation*, which satisfies the following rules:

$$\forall\, a, b \in K, \quad \delta(a+b) = \delta a + \delta b,$$
$$\delta(ab) = \delta a \,.\, \delta b,$$
$$\delta a = 0 \Leftrightarrow a = 0.$$

a is called a *constant* iff $\delta a = a$.

II.1.3. Let K be a difference field. Its *inversive closure* is a difference over-field K^*, which is unique up to isomorphism, such that, for any $a \in K^*$,

- $\delta^r a$ belongs to K for r sufficiently large,
- $\delta^{-1} a$ is defined.

For discrete-time systems, δ (resp. δ^{-1}) represents the backward (resp. forward) shift operator.

II.1.4. Let L/K be a difference field extension, and let K^* and L^* be their inversive closures. The *order* (resp. *effective order*) of L/K is the transcendence degree of L/K (resp. L^*/K^*). It is not difficult to prove that the effective order is less than or equal to the order.

(1) Some of Sontag's works [38, 40], which employ algebraic geometry, have points in common with our approach.

(2) See Cohn [6] for more details. We assume the reader to be familiar with the basic definitions and theorems of commutative algebra (see, e.g., Lang [26])