NONLINEAR CONTROL SYSTEMS

Y480 FINAL PROJECT

Kenneth A. Cobb
August 28, 1990
Prof. Steven Gray

TABLE OF CONTENTS

1.0 INTRODUCTION

A linear multivariable control system with m inputs and p outputs is usually described, in state space form, by means of a set of first order linear differential equations:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

in which x denotes the state vector (an element of R^n), u, the input or control vector (an element of R^m), and y, the output vector (an element of R^p). The matrices A, B, and C are matrices of real numbers of proper dimensions. The analysis of the interaction between input u and state x, and between state x and output y, has proved of fundamental importance in understanding the possibility of solving a large number of relevant control problems.

Key tools for the analysis of such interactions are the notions of reachability and observability and the corresponding decomposition of the control system into "reachable/unreachable" and "observable/ unobservable" parts.

In the case of multivariable nonlinear control systems, with m inputs or controls $u_1, \ldots, u_m$, and p outputs $y_1, \ldots, y_p$, the state space form, by means of a set of equations, is:

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x) u_i$$
$$y = h_i(x) \qquad 1 \leq i \leq p$$

The state variable x is a n x 1 column vector $x = (x_1, \ldots, x_n)$.

Two types of differential operations involving vector fields that are frequently used in the analysis of nonlinear control stystems are the Lie derivative and the Lie product. The Lie derivative, written as $L_f\lambda$, involves a real-valued function λ and a vector field f, both defined on subset U of R^n. By definition,

$$L_f\lambda(x) = \sum_{i=1}^{m} \frac{\partial\lambda}{\partial x_i} f_i(x)$$

and is called the derivative of λ along f. $\frac{\partial\lambda}{\partial x}$ is a 1 x n row vector and f(x) is a n x 1 column vector:

$$\frac{\partial\lambda}{\partial x} = [\frac{\partial\lambda}{\partial x_1}, \frac{\partial\lambda}{\partial x_2}, \ldots, \frac{\partial\lambda}{\partial x_n}]$$

$$f(x) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

The second type of operation involves two vector fields, both defined on an open subset U of R^n. The operation, which is called the Lie product, gives a new smooth vector field:

$$[f,g](x) = \frac{\partial g}{\partial x} f(x) - \frac{\partial f}{\partial x} g(x)$$

where $\frac{\partial g}{\partial x}$ and $\frac{\partial f}{\partial x}$ denote the Jacobian matrices of g and f, respectively:

$$\frac{\partial g}{\partial x} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial g_n}{\partial x_1} & & & \frac{\partial g_n}{\partial x_n} \end{bmatrix}, \quad \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_n}{\partial x_1} & \cdots & & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Another procedure that is frequently used in the analysis of nonlinear control systems is the "nonlinear change of coordinates" in the state space. Transforming the coordinates in the state space is often very useful in order to highlight some properties of interest like reachability and observability, or to show how certain control problems can be solved. A nonlinear change of coordinates can be described in the form:

$$z = \Phi(x) = \begin{bmatrix} \phi_1(x) \\ \cdot \\ \cdot \\ \cdot \\ \phi_n(x) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1, x_2, \ldots, x_n) \\ \cdot \\ \cdot \\ \cdot \\ \phi_n(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

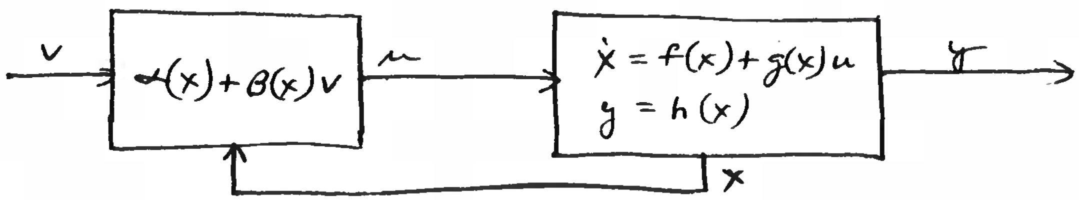
with the following properties:

1) $\Phi(x)$ is invertible if there exists a function $\Phi^{-}(z)$ such that $\Phi^{-1}(\Phi(x)) = x$ for all x in R^n .

2) $\Phi(x)$ and $\Phi^{-1}(z)$ are both smooth, such that all entries are real-valued functions with continuous partial derivatives of any order.

This type of transformation is called a global diffeomorphism on R^n . Sometimes, a transformation possessing both of these properties and defined on <u>all</u> x is difficult and the properties in question are difficult to check. Therefore, in most cases, one looks at the transformation defined only in a neighborhood of a given point. A transformation of this type is called a local diffeomorphism.

The next concept that can be used to linearize and control certain classes of nonlinear control systems is that of feedback. With single-input, single-output (SISO) systems, the state x of the system is available for measurements, and the input of the system will depend on this state and, possibly, on external referenced signals:



Thus, feedback linearization is based on three operations:

1) Change of coordinates in the state space (x→z).

2) Change of coordinates in the control (input) space (z→v).

3) Feedback.

2.0 DISCUSSION

This term paper is a review and analysis of the published paper titled:

Computer Algebra for Analysis and Design of Nonlinear Control Systems, by O. Akhrif and G. L. Blankenship, Proceedings of American Control Conference, 1987.

The first part of the published paper reviews some methods for analysis of deterministic servo-problems for nonlinear systems affine in control systems. The two methods used are the concepts of feedback equivalence and left-invertibility of nonlinear control systems. In the second part of the paper, a software package, called CONDENS, that inplements these two concepts for nonlinear control systems using a differential geometric approach, is presented. Using the theoretical concepts presented in the paper, given a nonlinear system affine in control, the CONDENS software attempts to answer the following questions:

- Is the system feedback-equivalent to a controllable linear system?
- If so, can a diffeomorphism be constructed that makes this equivalence explicit?
- Is the nonlinear system invertible?
- What is the relative degree of the system?
- If invertible, can the left-inverse be constructed?
- Given a real analytic function y(t), can y(t) appear as output of the nonlinear system?
- If so, what is the required control (input) u(t)?

A nonlinear system affine in control is said to be feedback linearizable in a neighborhood x^0 if there exists a transformation

(Φ,S) consisting of:

1) A diffeomorphism $\Phi(x)$ such that $z = \Phi(x)$, which represents a state space change of coordinates.

2) State feedback and affine change of coordinates of the control space R^m over the range of smooth functions, where

$$v = S(x,u) = \alpha(x) + \beta(x)u ,$$

such that in the new coordinates (z,v), the nonlinear system becomes a controllable linear system of the form:

$$\dot{z} = Az + Bv$$

The necessary and sufficient conditions under which such a transformation exists are stated in the published paper in a theorem by R.L. Hunt and R. Su (1983). Specifically, it states that the set of vectors $C = \{ g_1, [f,g_1], ad_f^k(g_1), \ldots, g_2, [f,g_2], ad_f^k(g_1) \}$ spans an n-dimensional space. Also, the vectors in C must involutive.

Block triangular systems are a class of systems that satisfy this theorem, and make constructing the feedback linearizable system an easy task. The solution comes down to solving a set of first order partial differential equations. Hunt and Su give a procedure to solve these equations by solving n sets of n ordinary differential equations. This algorithm can be very complex and is implemented in the CONDENS software.

A left-invertible nonlinear system is a system such that it is possible to reconstruct uniquely the input acting on the system from knowledge of the corresponding output. Invertibility of a system depends on what is defined as the <u>relative degree</u>, or relative order, of that system. The relative order of a nonlinear system is the first nonnegative integer r such that:

$$L_g L_f^j h = 0 \qquad 0 \leq j \leq r-1$$
$$L_g L_f^r h \neq 0$$

R.M. Hirschorn stated in a theorem (1979) that a nonlinear system is strongly invertible if and only if r is a finite number. A system that is strongly invertible means that there exists an open neighborhood U of x^0 such that for all x in the neighborhood, the system is invertible at x^0.

The aim when using the concept of relative degree is to solve for the control u(t) as a function of the state x(t) and the output y(t). To solve the output equation y(t) = h(x(t)) for u(t), it is necessary to solve a series of differential equations of the form:

$$\frac{d^r h}{dt^r} = L_f^r h(x) + L_f L_g^{r-1} h(x) u(t) \quad , \text{ where } L_f L_g^{r-1} h(x) \neq 0$$

The solution of this series will generate a left-inverse for the original system of the form:

$$\dot{z} = F(z) + G(z)v \qquad z(0) = x^0$$
$$w = H(z) + K(z)v$$

where F(z), G(z), H(z), and K(z) are all functions of Lie derivatives of f, g, and h.

3.0 CONDENS SOFTWARE PACKAGE

The CONDENS software package, itself, is quite interesting in that its two main features are: 1) the ability to process symbolic calculations such as differential equations and differential geometry, and 2) the automatic generation of numerical programs for solutions to problems inputted in symbolic form.

The CONDENS programs fall under three different categories. The first category includes a set of functions that perform differential geometric computations such as:

LIE(f,g) : computes the Lie bracket of vector fields f and g.

LIDEV(f,h) : computes the Lie derivative of the real valued function h along the direction vector f ($L_f h$).

JACOB(f) : computes the Jacobian matrix of f.

BTRIANG(f,g) : checks to see if the nonlinear system affine is block triangular.

RELORD(f,g,h) : computes the relative order of the SISO nonlinear system.

The second category includes the programs that address two important topics in nonlinear control systems: feedback linearization and left- and right-invertibility:

TRANSFORM(f,g): investigates the existence of a one-to-one transformation consisting of a change of coordinates and feedback, which transforms the system,

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i$$

to a controllable linear system in canonical form. If the transformation does exist, TRANSFORM will generate a set of partial differential equations.

INVERT(f,g,h) : given the same system as above, plus the output

$$y = h(x)$$

checks to see if the system is strongly invertible. If so, it computes the relative order of the system and computes the left-inverse to the system.

The third category of CONDENS presents two programs, FENOLS and TRACKS, which use TRANSFORM and INVERT to design a control law that forces the output y(t) of a nonlinear system to follow some desired trajectory:

FENOLS: Takes as input the nonlinear dynamics $f, g_1, \ldots, g_m$, the desired path $x_d(t)$, and the desired eigenvalues for the linear regulator. It then checks to see if the system is transformable to a controllable linear, and constructs the nonsingular transformation I and its inverse I^{-} .

TRACKS: Takes as input the same as above, plus $h_1, \ldots, h_m$, and the desired trajectory $y_d(t)$. It then investigates the left- and right-invertibility of the nonlinear system. It checks the left-invertibility by computing the relative order of the system and makes sure that it is finite. It also determines the right-invertibility by checking to see if the desired trajectory $y_d(t)$ is trackable by the system.

4.0 ANALYSIS

The CONDENS program is used in two examples to solve nonlinear control problems. The first example is of a basic industrial robot that has one rotational joint and a translational joint in the (x,y) plane. Once the robot is modelled, the system is described by a state space representation of the form:

$$\dot{x}1 = x_2$$

$$\dot{x}_2 = x_1 x_4^2 - x_4^2/2 + u_1$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = [\ 4x_2 x_4 \ (1 - 2x_1) + 4u_2 \ ] \ / \ (4x_1^2 - 4x_1 + 5)$$

$$y_1(t) = x_1(t) \qquad y_2(t) = x_3(t)$$

Using the exact linearization approach, CONDENS determines that the system is in block triangular form. This is because the set

$$\{ g_1, [f,g_1], g_2, [f,g_2] \}$$

spans a 4-dimensional space (n=4) and is involutive. This is the approach used in Chapter 1 of Isidori. Next, a set of first order partial differential equations is solved by solving 4 sets of 4 ordinary differential equations, where the solution of each system depends on the solution of the previous system. Thus, the new control variables are determined. The trajectory portion is then computed.

Next, using the left-inverse approach, as in Chapter 4 of Isidori, to solve the nonlinear system, the relative order or degree is computed.

If the relative order is a finite, then the system is strongly invertible. If the system is strongly invertible, then it is possible to reconstruct uniquely the input acting on the system from knowledge of the output. CONDENS determines that the relative order is 2, and computes the left-inverse to the system. In both cases, Fortran programs are provided.

Example 2 looks at a different nonlinear system and determines if it is feedback linearizable. Since it is determined that the system is not block triangular, the general method must be used. The process gets much more complex, however, the new state variables z_i are found, and subsequently, new control variables v_i are found.

5.0 CONCLUSION

I am quite impressed with the capabilities of the CONDENS software package. The fact that it can handle geometric, trigonometric, and exponential expressions is quite fascinating. Also, the ability to differentiate (Lie derivatives, Lie Product, Jacobian, etc.) these same expressions is very valuable. The Fortran programming is the simplest part (straightforward) of the whole paper.

I also feel that topics such as trajectory, right-invertibility, output tracking controllers, and desired eigenvalues may have perhaps been beyond the scope of this course and definitely beyond the scope of my current knowledge. However, I know for a fact that I learned quite a bit from reading (several times) this published paper and am even more interested in the algorithms used in the CONDENS softwware.

Finally, I would very interested in taking a Y480 course (next Summer?) that would continue on in Isidori's textbook and perhaps cover in detail some of the above topics that, because of time, were not covered in this course.

REFERENCES

1. Computer Algebra for Analysis and Design of Nonlinear Control Systems, by O. Akhrif and G. L. Blankenship, 1987.

2. Nonlinear Control Systems, by Alberto Isidori, 2nd Edition, 1989.