Computer Aided Design of Nonlinear Controllers with AppSys

WOLFGANG WERTH, NICOLAOS DOURDOUMAS
Department of Automatic Control
Technical University Graz
Krenngasse 37/II, A-8010 Graz
AUSTRIA
werth@irt-tu.graz.ac.at

Abstract: In this paper a software package - the Matlab toolbox AppSys - for the analysis and the design of nonlinear control systems is proposed. It includes both powerful tools to study the dynamical behaviour of a nonlinear system by simulation and to create appropriate approximation models. Furthermore the program proves to be an indispensable tool for the design of nonlinear state space controllers. The proposed approximation approach is known as Carleman-Linearization method and can be seen as a generalization to the common linearization technique. The software package AppSys allows to manage even complicated technical systems using the considered approximation method. The approximation process as well as the design of nonlinear state space controllers by AppSys is demonstrated for a cascaded 3-tank laboratory system.

Key-Words: controller design, nonlinear systems, Carleman Linearization, approximation, CACSD

1 Introduction

Nonlinear control theory brought out powerful methods in order to analyze dynamical systems and to design nonlinear controllers. In the case of nonlinear technical systems, the analysis and synthesis often require a lot of time consuming calculations. For these commercial software packages like Matlab/Simulink or Maple are available. Unfortunately universal mathematical software mostly cannot take into account various computational difficulties of a specific method. Furthermore the practical application of a nonlinear controller design procedure to technical systems requires simplicity in handling and implemention. Therefore an application oriented software tool can be seen as a useful (or necessary) interface between the engineer and a theoretical method.

In this paper the Matlab based toolbox AppSys is presented, which supports the engineer to calculate and verify approximation models for a nonlinear system. AppSys uses the powerful features of Matlab/Simulink and meets with the requirements of the considered approximation method. The software package provides a graphical user interface (GUI) and offers the following features:

- symbolic input of nonlinear differential equations
- automatic generation of an approximation model
- simple performing of simulation studies
- automatic controller design

On the basis of an example the benefits of the toolbox are demonstrated.

2 The Approximation method

We consider single input, single output, time invariant, continuous time systems of the form

\[ \dot{x} = a(x) + b(x)u \]
\[ y = c^T x. \]  \hspace{1cm} (1)

Let \( x \) denote the \( n \)-dimensional state vector, \( u \) and \( y \) the input and the output signal of the nonlinear system respectively. The main task of any approximation method is to generate a simple but sufficient accurate description of the considered plant. The basic idea of the Carleman-Linearization approach is the representation of the nonlinear system by a finite set of differential
The approximation of the state equation (1) is now given by

\[
\dot{x} = \sum_{k=1}^{N} A_k x^{(k)} + \sum_{l=0}^{N-1} B_l x^{(l)} u.
\]  

(3)

Note that all terms of degree greater than \( N \) are not used. To continue the determination of the Carleman representation of (1) differential equations are developed for \( x^{(2)}, \ldots, x^{(N)} \) again neglecting terms of degree greater than \( N \). Surprisingly all resulting differential equations satisfy a general form of (3). Thus, if we introduce the vector \( z \), i.e.

\[
z := \left[ x^{(1)^T}, x^{(2)^T}, \ldots, x^{(N)^T} \right]^T
\]  

(4)

then the calculated differential equations can be collected to a large bilinear state equation:

\[
\frac{d}{dt} z = A z + M z u + b u
\]  

(5)

The state equation (5) can be regarded as an approximation model of order \( N \) for the nonlinear system (1). It is plausible that the generation of the bilinear approximation model apart from first order systems does not succeed without appropriate software tools. Therefore the computer aided calculation needs to process the following steps:

- Taylor series expansion of the nonlinear terms
- calculation of \( A, M, b, c \) using some properties of the Kronecker notation
- elimination of redundancy caused by the Kronecker notation [3,9]

In addition to the above mentioned operations AppSys provides a simple symbolic input of a nonlinear state space model (1), as shown in figure 1. Note that the user has to recognize only the common Matlab syntax. Furthermore the calculation of an approximation model of any order \( N \) is done automatically. This process is demonstrated in Fig. 2. All results are stored for further use in the workspace and can be accessed in the Matlab Command Window.

Fig. 1: Input of a nonlinear system

The so-called Approximation Command Window represents the command unit of the toolbox. Besides the user can easily edit and change system parameters. Fig. 2 demonstrates also the change of the nonlinear function \( a(x) \).

Fig. 2: Approximation Command Window
3 Control problem formulation

Consider the nonlinear, time invariant, continuous time system shown in Fig. 3:

![Nonlinear control system](image)

Fig. 3: Nonlinear control system

For the given plant (1) a nonlinear state space controller \( k(x) \) has to be found to guarantee at least the local stability of a considered equilibrium point \((x_R, u_R)\) of (1).

Due to its simplicity we use the approximation model instead of the nonlinear state equation to calculate the desired state space controller, assuming that (5) represents a sufficient accurate description of the dynamic behaviour of (1). Many controller design procedures proposed in bilinear control theory can be applied to a system of (5) [3,8,9].

By means of examples for the given bilinear model (5) a nonlinear state space controller \( k_z(z) \) has to be found to minimize the following quadratic performance index

\[
J = \frac{1}{2} \int_0^\infty (z^T Q z + \rho u^2) dt \rightarrow \min,
\]

where \( Q \) and \( \rho \) are given positive definite Matrices of appropriate dimensions. Then the state space controller

\[
u = k_z(z) = -\frac{1}{\rho} (b + Mz)^T P z
\]

with

\[
A^T P + PA - \frac{1}{\rho} (b + Mz)(b + Mz)^T P + Q = 0
\]

will stabilize the bilinear control system (5). The application of the control (7) to the original system (1) requires its transformation into the real state variables \( x \), i.e.

\[
u = k(x) = k_z(z)|_{z \to z(x)}.
\]

The quadratic controller in \( k_z(z) \) leads to a control law in terms of a nonlinear state-feedback controller of a general form.

4 Example

As an illustrative example the model of a cascaded laboratory 2-tank system shown in Fig. 4 is considered.

![2-tank system](image)

Fig. 4: 2-tank system

The state variables are considered as the fluid levels \( x_i \) of the tanks. The nonlinear model of the tank system is then given by

\[
x_1 = -0.391\sqrt{x_1} + u_z
\]

\[
x_2 = -0.386\sqrt{x_2} + 0.391\sqrt{x_1}
\]

\[
y = x_2
\]

which describes the relationship between the inflow \( u_z \) and the fluid levels \( x_i \) of the tanks. The desired inflow is produced via a pump driven by an input voltage \( u \) (i.e. the control input). The task of the control system is to cause the fluid level of the second tank to follow a reference level \( r \) as closely as possible. First an approximation model (5) is calculated using the desired equilibrium point:

\[
x_R = \begin{bmatrix} 24.8 cm \\ 25.5 cm \end{bmatrix}, \quad u_R = 1.87 V.
\]

In order to achieve a simple implementation of the control law the order of the model is set to \( N = 2 \). Simulation studies which demonstrate the behaviour when the empty tanks are filled, justify the considered model order [9].