An Interactive Software for Designing Nonlinear Controllers Based on Feedback Linearization Technique

by A. Kaddouri [1], S. Blais [1], M. Ghribi [1] and O. Akhrif [2]
[1] Université de Moncton, Moncton, NB
[2] École de technologie supérieure, Montréal, QC

Abstract

This paper describes new software developed for the design of nonlinear controllers based on the well-known feedback linearization technique. NLSoft is a software package containing several symbolic manipulation modules, which includes differential geometric tools for the design and simulation of control systems. NLSoft presents a user-friendly graphical user interface (GUI) as well as a new and powerful module allowing the calculation time of linearizing control laws considering several Digital Signal Processors (DSPs) characteristics. These facilitate the real-time implementation of the control system. NLSoft is validated considering an example.

Sommaire

Le présent article décrit un nouveau logiciel (NLSoft) dédié à la conception des contrôleurs nonlinéaires basés sur la technique de la linéarisation par retour d'état. NLSoft est présenté sous forme d'une interface graphique (GUI) et possède beaucoup de modules à manipulation symbolique incluant entre autre plusieurs outils de la géométrie différentielle nécessaires pour la conception et la simulation numérique des systèmes dynamiques non-linéaires. De plus, il contient un module très puissant permettant l'évaluation du temps de calcul de la loi de commande linéarisante en considérant plusieurs processeurs numériques de signal (DSP). NLSoft a été validé en considérant un exemple.

1.0 Introduction

The nonlinearities of nonlinear dynamic systems have been traditionally handled in the literature by the use of classical linear control methods, which rely on linearized models approximating the dynamic system's equations. These methods are just valid in a small operation range. When the required operation range has to be larger, the linear controller is likely to perform poorly. Nonlinear controllers, however, may handle the nonlinearities in large range operations directly. The nonlinear control considered here is based on the feedback linearization technique (static and dynamic) which gives a good solution for tracking control problems. Several accessible references describing its constructions are now available ([1] to [4]] as examples). This well-known nonlinear control technique has been successfully applied to the control of dynamic systems, even those with high nonlinearities in their model. However, this technique needs an extensive knowledge of differential geometry which provides a very useful framework for the analysis of nonlinear control systems. Differential geometric objects (Lie brackets, Lie derivatives, etc.) are not easily manipulated by hand. In order to facilitate this process, many new CAD tools appeared in the last decade (AISYS, AP_LIN, SCILAB, CONDENS, REDUCE, PROPAC etc.). They offer the possibility of designing and/or simulating nonlinear dynamic systems. In the real-time applications, the main problem encountered with these nonlinear control techniques is that they need an important calculation time which makes implementation impossible on a general use microprocessor. Fortunately, a new class of digital signal processors (DSPs) is making this implementation possible. The software presented in this paper (NLSoft) is developed in order to design nonlinear controllers, to simulate the closed-loop systems and to give useful information for the real-time implementation considering real DSPs characteristics (TMX320F2812, ADMC401 etc.). Furthermore, it is capable to inform us about the stability of the zero dynamics if they exist. NLSoft uses several subsystems expressed as classes which permit the design of nonlinear controllers and the plotting of the results obtained from the simulation of the closed-loop system.

The block diagram of the closed-loop system to be designed and simulated using NLSoft is given in Figure 1.

2.0 Description of NLSoft

NLSoft consists of a design environment built in Visual C++. This design environment, hereafter referred to as "NLSoft Design Environment" or NODE, contains functions which can be grouped into five different subsystems. The first subsystem, the GUI, guides the user into making important design decisions at each step of the design process. The four remaining subsystems - Acquisition and Display, System Analysis, Storage and Symbolic Subsystem - contain several functions to adapt, save and recall the information provided by the user, perform analysis on a system, display and print results, launch symbolic algorithms and more.

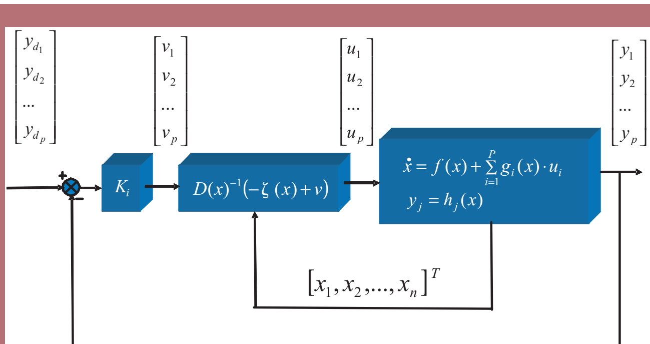


Figure 1: Block diagram of the closed-loop system

The NODE has the following characteristics:

- It contains several design packages which use the feedback linearization technique to design a control law that forces the output of a nonlinear system to follow some desired trajectory. It also contains a dynamic linearization package and a package for the analysis of the existence and the stability of the zero dynamics.
- The NODE also includes a new and powerful module permitting the calculation time of the designed nonlinear control laws. Two Digital Signal Processors (DSPs) characteristics are already included in NLSoft (TMX320F2812, ADMC401 etc.). The user may also consider other DSPs.

Finally, it permits the pole placement, the simulation of the closed-loop system and the visualization of its performance.

The NODE's architecture is given in Figure 2.

2.1 Graphical User Interface

The GUI included in the NODE presents several controls which are available with the click of a button. The GUI is modified when results are displayed at each step of the design process. Figure 3 shows the main controls of the NODE. The controls are grouped in three classes. The Data Input Controls allow the input of parameters characterizing either the nonlinear equations of dynamic systems or the DSP used to implement the designed nonlinear controllers. The data input controls are working closely with Acquisition and Display Functions to verify

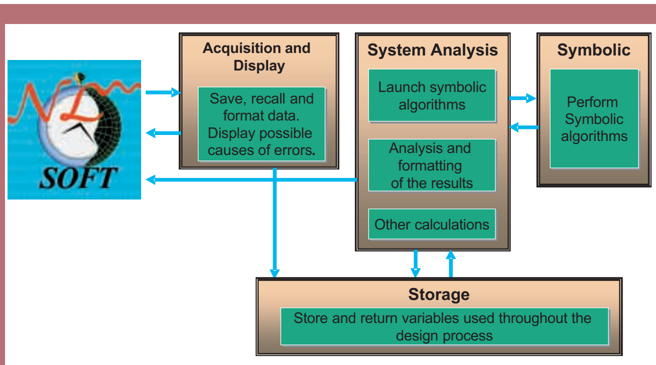


Figure 2: NODE architecture

basic syntax as data is entered. These functions prevent the launch of an erroneous algorithm in a future step of the process. The Action Controls execute a different step of the nonlinear control design process. These steps include finding the linearizing control law of a nonlinear system, calculating the corresponding calculation time required for the specified DSP to implement the linearizing control law and to simulate the linearized system with several customizable input commands. The simulation results may be shown directly or stored in a Matlab m-file for the user to plot the different graphs. The Options Control allows users to modify registered variables. The Print control allows users to print previously calculated and displayed results.

2.2 Acquisition and display subsystem

The acquisition and display subsystems contain functions that can be grouped in three categories: Syntax Functions, File I/O Functions and Formatting Functions. Firstly, the Syntax Functions allows the analysis of the input entered by users. Several syntax rules are checked, such as the closing of parenthesis and brackets, the syntax of built-in functions and more. If not properly followed, error messages are displayed with the proper syntax rules to follow. Secondly, File I/O Functions allow parameters to be saved, recalled and modified. Specifically, these File I/O functions are used for the storage of the system and DSP parameters.

2.3 System analysis subsystem

The System Analysis Subsystem contains the required functions to analyze the parameters of a system and to interpret results at all stages of the design process. It acquires data from the Storage Subsystem, generates symbolic algorithms and launches them. Upon completion of the algorithm, the results are interpreted, formatted and stored in the Storage Subsystem. The algorithms included in this subsystem are used to perform the following tasks: find the relative order of the nonlinear system, find the linearizing control law, transform the resulting system in a canonical form, find the calculation time on the designed controller and simulate the controlled closed-loop system under specific consideration. All these algorithms are executed by the NODE.

2.4 Storage subsystem

The Storage Subsystem consists of a global class used to store data required by different subsystem functions. The indexing of the stored data is made on a per function basis. The data is stored in private variables available through specific functions in order to minimize the possibility of errors.

2.5 Symbolic Subsystem

NLSoft provides a large set of symbolic functions embedded in the design environment. These functions are used by the symbolic algorithms described previously. Object-oriented programming allows the easy reuse of these functions in the different algorithms. The symbolic functions can be classified in four different groups.

- First, functions performing basic mathematical operations are developed. Amongst these are functions performing derivatives, variable replacement, parameter matching, etc.
- Second, symbolic functions for matrixes are also included. Functions finding the size of the matrix, its transpose, inverse, performing row reduction, finding the null space matrix, and more are developed.
- Third, differential geometry functions are developed. These functions reuse the functions in the first two groups to calculate their results. They include the Jacobian matrix, Lie Derivative and more.
- Finally, simplifications functions are included in the package to optimize the output results and eliminate redundancy in the output equations.

3.0 Illustration Example

The mathematical model of the salient Permanent-Magnet Synchronous Motor in the d-q synchronous reference frame [8] is given by:

$$\dot{x} = f(x) + \sum_{i=1}^{2} g_i(x)u_i = f(x) + g_1(x)u_d + g_2(x)u_q \qquad (1)$$

With:

$$f(x) = [f_1 \quad f_3 \quad f_3]^T = \begin{bmatrix} -\frac{R}{L_d} i_d + \frac{L_q}{L_d} p w_r i_q \\ -\frac{R}{L_q} i_q - \frac{L_d}{L_q} p w_r i_d - \frac{\Phi_v}{L_q} p w_r \\ \frac{1}{J}(T_{ém} - B w_r) \end{bmatrix} \qquad (2)$$

$$x = [i_d \quad i_q \quad w_r]^T \quad et \quad u = [u_d \quad u_q]^T$$

$$g_1(x) = [\frac{1}{L_d} \quad 0 \quad 0]^T \qquad g_2(x) = [0 \quad \frac{1}{L_q} \quad 0]^T$$

$$T_{ém} = \frac{3p}{2J}(\Phi_v i_q + (L_d - L_q) i_d i_q)$$

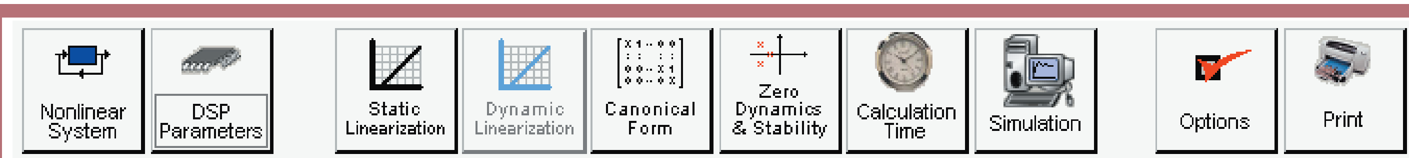


Figure 3: Main control of the NODE

where ud and uq are the d-q axis voltages, id and iq the d-q axis currents, L is the stator inductance, R is the stator resistance and wr is the rotor speed. v is the flux linkage due to the rotor magnets, p is the number of pole pairs, B is the damping coefficient and J is the rotor moment of inertia. Tém is the electromagnetic torque.

We choose the speed (wr) and the direct current (id) as output variables, then:

$$y=[y_1 \quad y_2]^T=[i_d \quad w_r]^T \tag{3}$$

The linearizing control law becomes:

$$u=D(x)^{-1}\cdot[-\zeta\,(x)+v] \tag{4}$$

where:

$$D^{-1}(x)=\begin{bmatrix} L_d & 0 \\ \dfrac{(L_q-L_d)L_q}{(\Phi_v+(L_d-L_q)i_d)}i_q & \dfrac{JL_q}{1.5p[(L_d-L_q)i_d+\Phi_v]} \end{bmatrix} \tag{5}$$

which is non-singular with: $\Phi_v \neq (L_q-L_d)i_d$

and:

$$\zeta\,(x)=\begin{bmatrix} -\dfrac{R}{L_d}i_d+\dfrac{L_q}{L_d}pw_r i_q \\ \lambda(L_d-L_q)i_q f_1+\lambda(\Phi_v+(L_d-L_q)i_d)f_2-\dfrac{B}{J}f_3 \end{bmatrix} \tag{6}$$
$$\lambda=1.5p/J$$

The total relative degree is r =1+2=3. Since the system order is 3, the system is then completely linearizable and there is no need to check the internal dynamics stability.

The results obtain by NLSoft are given in Figure 4.

The simulation results and the calculation-time of the linearizing control law are given in Figure 5. For the calculation time of the linearizing control law, we consider the TMX320F2812 DSP chip. The calculation time (90 sec.) is evaluated considering floating-point operations.

4.0 Conclusion

The paper presented a new CAD tool (NLSoft) oriented to the design and simulation of nonlinear controllers based on the feedback linearization technique. NLSoft was developed in a compact and user-friendly GUI interface. Two examples are given to validate the proposed Tool. We are working to integrate a dynamic linearization algorithm and an adaptive version to solve the problem of parameter uncertainties.

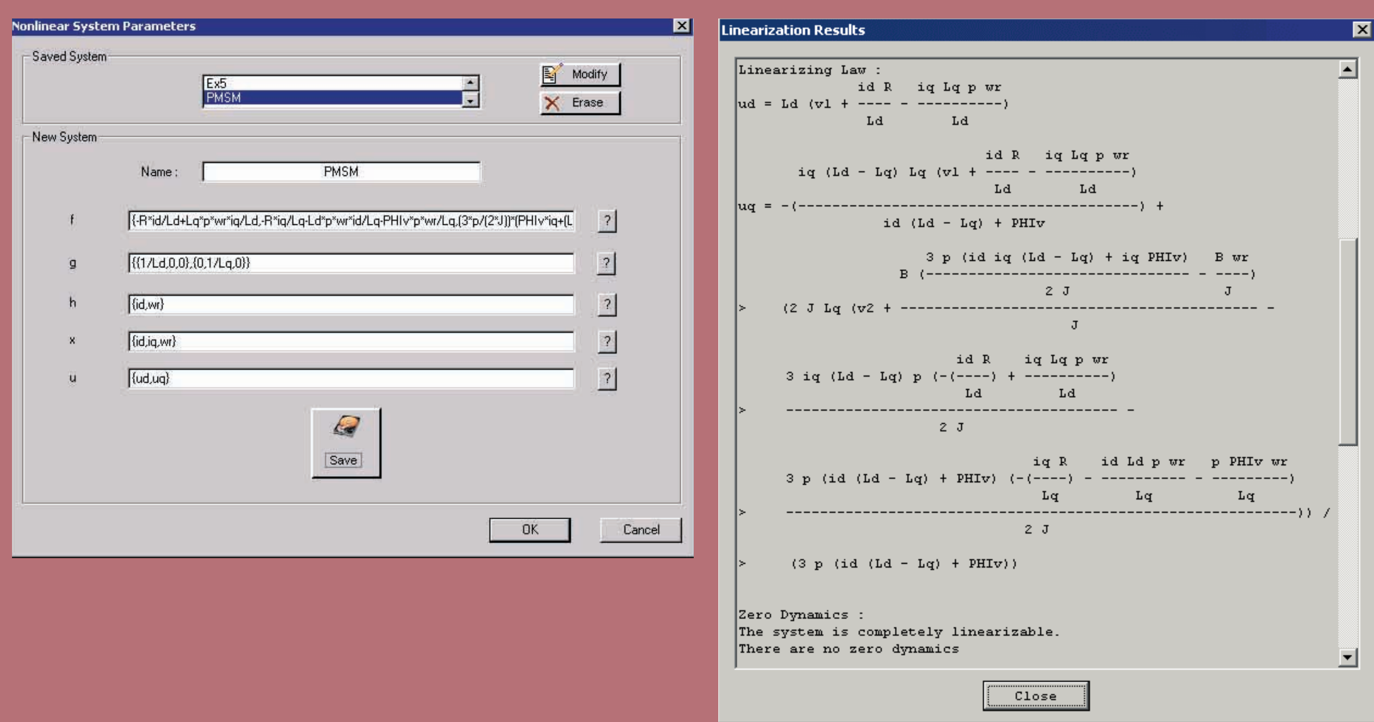


Figure 4: Results for the illustration example using NLSoft

(a) Data input of the dynamic system

(b) The Linearizing control law and zero dynamics stability check.

5.0 Acknowledgment

The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Council of Canada (NSERC).

6.0 References

[1]. A. Isidori, Nonlinear Control Systems: An Introduction (Springer-Verlag, 1989).

[2]. J. J. E. Slotine and W. Li, Applied Nonlinear Control (Prentice Hall, New Jersey, 1991).

[3]. H. G. Kwatny and G. L. Blankenship, Nonlinear Control and Analytical Mechanics: A computational Approach (Birkhäuser, 2000).

[4]. N. Munro, Symbolic Methods in Control Systems Analysis and Design (IEE Control Engineering Series 56, 1999).

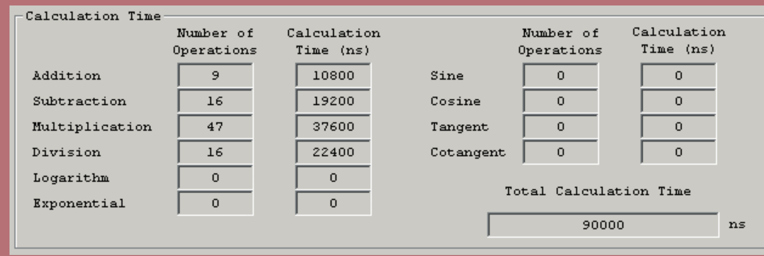
[5]. I. R. Kadiyala, APLIN: A CAD Toolbox for Nonlinear Control Design, Ph.D. thesis, Dept. of Elec. Eng., University of California, Berkeley, 1992.

[6]. D. N. Godbole et al., CAD Tools for nonlinear Control: Speed control of a Car engine, ACC'92, 1992, 317-318.

[7]. O. Akhrif, 'Nonlinear adaptive control with applications to flexible structure', Ph.D. dissertation, Malyland University, USA, 1989.

[8]. M. Bodson and al., High-Performance Nonlinear Feedback Control of a Permanent Magnet Stepper Motor, IEEE Trans. on Control Systems Technology, Vol. 1, No. 1, March 1993.

(a)



(b)

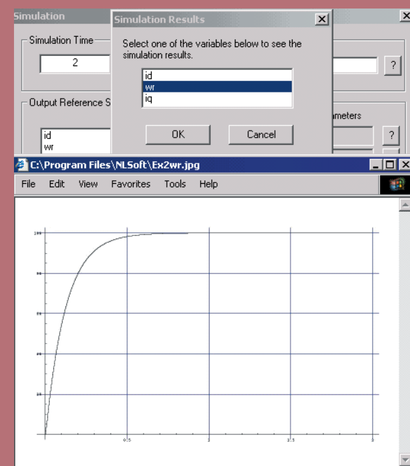


Figure 5: Output data for the example (a) Calculation time of the linearizing control law and (b) Simulation results showing the angular speed of the motor

About the authors

Azeddine Kaddouri received the Diplôme of ingénieur d'état in electrical engineering from Batna University, Algéria in 1988 and the M.Sc. and the Ph.D. degrees in electrical engineering from Laval University, Canada in 1993 and 2000 respectively. From 1993 and 1999 he was a research assistant with GREPCI research group, École de Technologie Supérieure, Montréal, Canada. In 1999, he joined the Université de Moncton, NB, Canada, where he is currently professor of Electrical Engineering. His research interests are the nonlinear control of electric drives.

Sébastien Blais received a B.Ing. in electrical engineering from Université de Moncton, Canada, in 2002. Since 2001, he has been an active member of GRET research group. Currently, he his working towards his M.Sc. degree in electrical engineering.

Mohsen Ghribi obtained his B.Sc. and the M.Sc. degrees from Université du Québec à Trois-Rivières, Canada and the Ph.D.degree from Université Laval, Canada respectively in 1987, 1989 and 1994. He has been a professor at École de Technologie Supérieure, Montréal and the École Polytechnique de Masuku (Gabon). Since 1997, he is a professor at Université de Moncton, Canada. His research interests include machines drives, control systems and real-time implementation using DSP chips.

Ouassima Akhrif received a Diplôme d'Ingénieur d'État from École Mohammadia, Rabat (Morocco) and M.Sc.A. & Ph.D. degrees from the University of Maryland, College Park, in 1984, 1987, & 1989, respectively, all in electrical engineering. During 1989-1990, she was a Visiting Assistant Professor in the Systems Engineering Department, Case Western Reserve University, Cleveland, OH. In 1992, she joined École de Technologie Supérieure, Montréal, where she currently is a Professor in the Electrical Engineering Department. Her research interests are nonlinear geometric control, nonlinear adaptive control, and their applications in electric drives, power systems, and flight control systems.