UDC 681.5

ROBOTICS APPLICATION OF SYMBOLIC DESIGN AND MODELING

Dyda A.A.[1, 2], Oskin D.A.[1,2], Krasiuk L.V.[1], Osokina E.B.[2], Yudakov A.A.[3]

[1]*Far Eastern Federal University, Vladivostok, Russia*
[2]*Maritime State University, Vladivostok, Russia*
[3]*Russian Academy of Sciences Far Eastern- Branch Institute of chemistry*

The work was supported by Ministry of Science and Education of Russian Federation, the State Contract 02G25.31.0025

ABSTRACT

This paper is devoted to the problem of the design procedure automation for nonlinear control systems on the basis of symbolic computations and such well-known technique as exact linearization via nonlinear feedback. Developed software realizes the synthesis of control laws in symbolic form that is convenient for further analysis, modeling, and practical implementation of the system. Examples of the software applications to induction motor and robot manipulator are given and discussed.

INTRODUCTION

IMPORTANT features of large-scale systems are high dimension, nonlinearity, and often uncertainty. To overcome the difficulties of large-scale systems research, the decomposition methods and appropriate computer software as effective tools are traditionally used. The notion “decomposition” can be considered, on one hand, as a procedure of splitting the large system into subsystems of lower dimensions. On other hand, decomposition can be desired result of control system design when subsystems dynamics are independent and demonstrate prescribed behavior.

A number of modern methods which are used to solve the problem of control systems synthesis exploit an idea of linearization. Among them the exact linearization of the controlled object dynamics via nonlinear feedback seems to be one of the most popular techniques. This approach, comprehensively given by A. Isidori, H. Nijmeijer and A. van der Schaft and other researchers, last two decades was extended and applied to adaptive control, variable structure control etc [1]-[4]. The main advantages of the nonlinear feedback linearization (NFL) method consist in its strong formalization and possibility to make decomposing of large system, to decouple controlled subsystems and to provide them reference (as a rule, linear) transient dynamics. Unfortunately, when the dimensions of state, control, and output vectors are large, the control system designer faces huge difficulties associated with complicated computations.

To deal with such systems, modern software for symbolic calculations can be applied. Moreover, as it will be demonstrated, the single procedure of the full system synthesis based on the NFL approach can be often implemented.

The aim of this paper is to develop the software for control law design and full control system modeling with usage of NFL method and symbolic computations.

The paper is organized as follows. In Section II a necessary background on NFL method is given. Section III briefly describes main features of algorithms and program realization of symbolic synthesis according to the NFL method. In Section IV the mathematical models of such dynamical objects as induction motor, robot manipulator, and underwater vehicle are presented. In order to demonstrate the effectiveness of approach to symbolic synthesis, developed software was applied to design the control systems for them. Results of computer synthesis are derived as symbolic expressions describing nonlinear control laws. Examples of system processes are shown. Section V contains discussion and final comments and touches possible directions of future development of the approach.

THE METHOD OF NONLINEAR FEEDBACK LINEARIZATION. BRIEF BACKGROUND

To begin with, we briefly review necessary concepts of feedback linearization, following [1]. The mathematical model of dynamical control object is supposed to be described by the equations followed:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u, \\ y &= h(x) \end{aligned} \qquad (1)$$

where

$$\dot{x} = (x_1,...x_n)^T,$$

$$\dot{u} = (u_1,...u_m)^T,$$

$$\dot{y} = (y_1,...y_m)^T$$

are the state, control, and output vectors, respectively.

According to the method of feedback linearization [1], the control should be formed as follows

$$u = A^{-1}(x)(w - \Gamma(x)). \qquad (2)$$

The decomposition matrix $A(x)$ and vector $\Gamma(x)$ are calculated by formulas

$$A(x) = \begin{bmatrix} L_{g_1}L_f^{r_1-1}h_1(x) & .. & .. & L_{g_m}L_f^{r_1-1}h_1(x) \\ L_{g_1}L_f^{r_2-1}h_2(x) & .. & .. & L_{g_m}L_f^{r_2-1}h_2(x) \\ .. & .. & .. & .. \\ L_{g_1}L_f^{r_m-1}h_m(x) & .. & .. & L_{g_m}L_f^{r_m-1}h_m(x) \end{bmatrix}, \qquad (3)$$

$$\Gamma(x) = \begin{bmatrix} L_f^{r_1}h_1(x) \\ L_f^{r_2}h_2(x) \\ ... \\ L_f^{r_m}h_m(x) \end{bmatrix}, \qquad (4)$$

where $\dot{r} = (r_1,...r_m)^T$ the relative degree vector,

$w = (w_1,...w_m)^T$ the new control input.

By the definition, the derivative of function $h_j(x)$ along vector $f(x)$ is the following

$$L_f h_j(x) = \sum_{i=1}^{n} \frac{\partial h_j(x)}{\partial x_i} f_i(x), \qquad (5)$$

The feedback control (2) transforms a nonlinear system (1) into m decoupled chains of r_i cascade-connected integrators and, as consequence, the input-output behavior of the system satisfies the linear equations

$$y_i^{(r_i)} = w_i. \qquad (6)$$

Obviously, the choice of the new control in the form

$$w_i = y_{di}^{(r_i)} + k_1 e_i^{(r_i-1)} + ... + k_{r_i} e_i, \qquad (7)$$

where $y_d(t)$ is the desirable trajectory of the system and error $e(t) = y_d(t) - y(t)$, provides asymptotically stable tracking.

If the equality $r_1 + ... + r_m = n$ takes place, the system is full-state linearizable, otherwise, it has unobservable subsystem with its own dynamics, or zero dynamics (for more details see [1], [2]).

COMPUTER SYMBOLIC DESIGN OF CONTROL LAWS

In the case when one needs to design the control system for the nonlinear dynamical object of relatively low dimension, the usage of the formulas (2)-(7) is rather simple. But with increasing of dimension which characterizes large-scale systems, number of inputs and outputs, the computational complexity soon arises.

Meanwhile, for research purposes, it is symbolic form of control that is most valuable and preferable. Naturally, symbolic expressions of control laws are convenient for further analysis, modeling, and practical realization.

Complicated work on deriving the control expressed in symbolic form and based, in particular, on the NFL method can be essentially facilitated by means of modern software systems such as Maple, MATEMATICA, MATLAB, MATHCAD etc. They have built-in functions for symbolic differentiation, vector and matrix operations, and many others.

In this research the computer program for the NFL-based symbolic synthesis and modeling of nonlinear control system was developed in the environment of Maple and MATLAB. Principal modules of the software are following:

- The description of variables (vectors of state, control and output);

- The description of the desirable system trajectory $y_d(t)$;
- The input of dynamical object model in symbolic form;
- The deriving of expressions for function $f(x)$ and matrix $g(x)$;
- The determination of relative degree vector;
- The input of coefficients which determine desirable transients;
- The symbolic calculations of decomposition matrix $A(x)$ and vector $\Gamma(x)$;
- The generation of symbolic expressions for control u and new control w;
- The forming of full closed-loop control system model by substitution of generated control laws into object model;
- The input of control object parameters;
- Control system modeling.

The user of developed program just input a description of control object in symbolic form and necessary parameters which computer demands. The program, in its turn, generates control laws that provide (at least, partial [1]) decomposition and reference dynamics fot subsystems. If one needs, the program simulates the designed control system and presents the processes in it.

EXAMPLES OF SYMBOLIC CONTROL SYNTHESIS AND SYSTEM MODELLING

To demonstrate the effectiveness of the approach and developed software, consider the examples of symbolic control synthesis and simulations for two dynamical objects. These are induction motor and robot manipulator.

Induction motor

The model of induction motor can be written as following equations [5]:

$$\begin{aligned} \dot{\phi} &= -\alpha\phi + p\omega\xi\phi + \alpha L_m I \\ \dot{I} &= \alpha\beta\phi - p\beta\omega\xi\phi - \gamma I + U/\sigma, \\ \dot{\omega} &= \mu I^T \xi\phi - T_L / J \\ \dot{\theta} &= \omega \end{aligned} \qquad (8)$$

where θ, ω are the rotor angular position and velocity; $\phi = (\phi_a, \phi_b)^T$, $I = (i_a, i_b)^T$ are the rotor flux and the stator current vectors; $U = (u_a, u_b)^T$ the control input voltage; T_L the load torque;

$$\alpha = R_r / L_r,\ \beta = L_m /(\sigma L_r),$$

$$\xi = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \gamma = \frac{L_m^2 R_r}{\sigma L_r^2} + \frac{R_s}{\sigma}$$

$$\sigma = L_s - \frac{L_m^2}{L_r} \quad \mu = \frac{3 L_m p}{2 J L_s}.$$

R_s, R_r the stator and rotor resistance; L_r, L_s, L_m the rotor, stator and mutual inductances; p the number of pole pairs.

The output and desirable trajectory of the system are chosen as following

$$y = \begin{bmatrix} \omega \\ \phi^T \phi \end{bmatrix},\ y_d = \begin{bmatrix} c_1 \sin(c_2 + vt) \\ c_3 \end{bmatrix}.$$

The results of the computer-based symbolic control synthesis are obtained (given in the Maple format), particularly,

U_1 =2*Rr*(-x2*Lm^2*Rr*Fbi-x1*Lm^2*Rr*Fai-2*x2*Rr*Lm^3*x4-2*Rr*x2^2*Ls*Lr-2*Rr*x1^2*Ls*Lr+x2*Lm*x4*Rs*Lr^2+Lm^3*x4*p*x5*x1*Lr+x1*Lm*x3*Rs*Lr^2-Lm^2*x4^2*Rr*Ls*Lr-Lm*x4*p*x5*x1*Lr^2*Ls+3*x2*Rr*Lm*x4*Ls*Lr+2*Rr*x2^2*Lm^2-Lm^3*x3*p*x5*x2*Lr-Lm^2*x3^2*Rr*Ls*Lr+3*x1*Rr*Lm*x3*Ls*Lr-2*x1*Rr*Lm^3*x3+Lm*x3*p*x5*x2*Lr^2*Ls+2*Rr*x1^2*Lm^2+Lm^4*x4^2*Rr+Lm^4*x3^2*Rr)/Lr^2/(-Ls*Lr+Lm^2);

U_2 =1/6*(4*Rr*x1*w2*J*Lr^3*Ls-4*Rr*x1*w2*J*Lr^2*Lm^2+6*Rr^2*x1^2*Lm*p*x4*Ls*Lr+6*Rr*x1^2*Lm*p^2*x3*x5*Lr^2*Ls-6*Rr*x1^2*Lm^3*p^2*x3*x5*Lr+6*Rr*x1*Lm^2*p^2*x2^2*Lr*x5-6*Rr^2*x1^2*Lm^2*p*Fbi+6*Rr*x1^3*Lm^2*p^2*Lr*x5+6*Rr*x1^2*Lm*p*x4*Rs*Lr^2+18*x2^2*p*Rr^2*Lm*x4*Ls*Lr+3*x2*p*w1*Lr^3*Ls-3*x2*p*w1*Lr^2*Lm^2+12*x2*p*Rr^2*x1^2*Lm^2+6*x2*p*Lm^4*x3^2*Rr^2+6*x2*p*Lm^4*x4^2*Rr^2+6*x2^2*p*Rr*Lm*x4*Rs*Lr^2+6*x2^2*p^2*Rr*Lm*x3*x5*Lr^2*Ls-6*x2^2*p^2*Rr*Lm^3*x3*x5*Lr-6*x2*p*Lm^2*x4^2*Rr^2*Ls*Lr-6*x2*p*Lm^2*x3^2*Rr^2*Ls*Lr-6*x2^2*p*Rr^2*Lm^2*Fbi+12*x2*p*x1*Rr^2*Lm*x3*Ls*Lr-12*x2*p*Rr^2*x1^2*Ls*Lr-12*x2*p*x1*Rr^2*Lm^3*x3-12*x2^3*p*Rr^2*Ls*Lr-12*x2^2*p*Rr^2*Lm^3*x4+12*x2^3*p*Rr^2*Lm^2)/p/Lr^2/Lm/Rr/(x1^2+x2^2);

W_1 =(C2-x1^2-x2^2)*ki11+(-2*x1*(-Rr*x1-p*x5*x2*Lr+Rr*Lm*x3)/Lr-2*x2*(-Rr*x2+p*x5*x1*Lr+Rr*Lm*x4)/Lr)*ki12;

w_2 =(C1*sin(v*t)-x5)*ki21+(C1*cos(v*t)*v-1/2*(-3*Lm*p*x3*x2+3*Lm*p*x4*x1-2*Tl*Lr)/J/Lr)*ki22-C1*sin(v*t)*v^2.

In the given formulas the following variables are used: $x1 = \phi_a, x2 = \phi_b, x3 = i_a,$

$x4 = i_b, x5 = \omega, x6 = \theta$

After deriving the symbolic expressions for controls, the program substitutes them into the object model and simulates the whole system. Some processes are shown on Fig.1. – Fig.3.

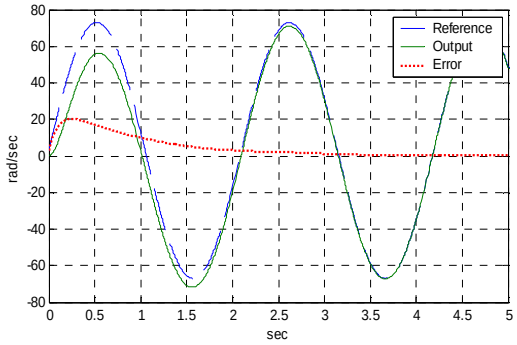

Fig. 1 The desired reference and real angular velocities.

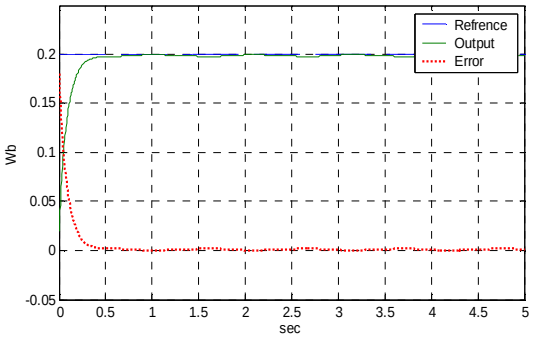

Fig. 2 The rotor flux square modulus output and reference.

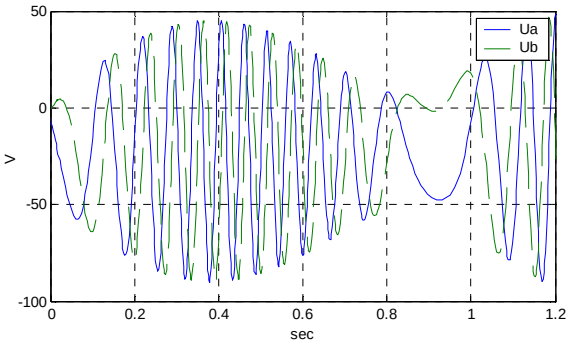

Fig. 3 Control input voltage vector

As seen, synthesized control input is bounded and provides asymptotically stable tracking of reference trajectory $y_d(t)$.

The Robot Manipulator

Next example of dynamical object to which we will apply the developed software is the robot manipulator UMS-2 [6]. Its dynamics are described by the equation

$$D(q)\ddot{q} + B(\dot{q}, q)\dot{q} + G(q) = U,$$

where $U \in R^3$, $q \in R^6$ are the control the generalized coordinates vectors. The matrices are determined as follows

$$D(q) = \begin{bmatrix} I_1 + I_2 + I_3 + m_3(q_3 + l_3)^2 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix}$$

$$B(\dot{q}, q) = \begin{bmatrix} 2m_3(q_3 + l_3)\dot{q}_3 & 0 & m_3(q_3 + l_3)\dot{q}_1 \\ 0 & 0 & 0 \\ -m_3(q_3 + l_3)\dot{q}_1 & 0 & 0 \end{bmatrix}$$

$$G(q) = [0 \quad (m_2 + m_3)g \quad 0]^T.$$

The order of the system is n=6, the number of inputs is m=3. The variables are redefined as

$$x_1 = q_1, x_2 = \dot{q}_1, x_3 = q_2, x_4 = \dot{q}_2, x_5 = q_3, x_6 = \dot{q}_3:$$

Undefined parameters are supposed to be constant.

The outputs of the system are the coordinates of the manipulator grasp in task space

$$h_1 = (x_5 + l_3)\sin(x_1),$$
$$h_2 = (x_5 + l_3)\cos(x_1),$$
$$h_3 = x_3$$

Desired trajectory $y_d(t)$ is chosen to provide the movement of the grasp along the spatial circle

$$y_{d1} = R\sin(\alpha t),$$
$$y_{d2} = c_1,$$
$$y_{d3} = c_2 - R\cos(\alpha t)$$

The program generates the following symbolic expressions for control

$$U=\begin{bmatrix}\dfrac{w_1m_3x_5^2\cos x_1+(2m_3x_5l_3+m_3l_3^2)(w_1\cos x_1-w_2\sin x_1)}{x_5+l_3}+\\ +\dfrac{(w_1\cos x_1-2x_2x_6-w_2\sin x_1)I}{x_5+l_3}\\ (m_2+m_3)(w_3+G)\\ \dfrac{m_3(x_2^2+w_2x_5\cos x_1+w_2l_3\cos x_1+2x_2^2x_5l_3}{x_5+l_3}+\\ +\dfrac{x_2^2l_3^2+x_2^2x_5^2+x_5w_1\sin x_1+l_3w_1\sin x_1)}{x_5+l_3}\end{bmatrix}$$

$$w=\begin{bmatrix}(R\sin\alpha t-(x_5+l_3)\sin x_1)k_{11}+(R\alpha\cos\alpha t-x_6\sin x_1-\\ -(x_5+l_3)x_2\cos x_1)k_{12}-R\alpha^2\sin\alpha t\\ (c_1-(x_5+l_3)\cos x_1)k_{21}+((x_5+l_3)x_2\sin x_1-x_6\cos x_1)k_{22}\\ (c_2-R\cos\alpha t-x_3)k_{31}+(R\alpha\sin\alpha t-x_4)k_{32}+R\alpha^2\cos\alpha t\end{bmatrix}$$

Fig.4 - Fig.7 show the transient processes obtained in the simulation of the whole control system.

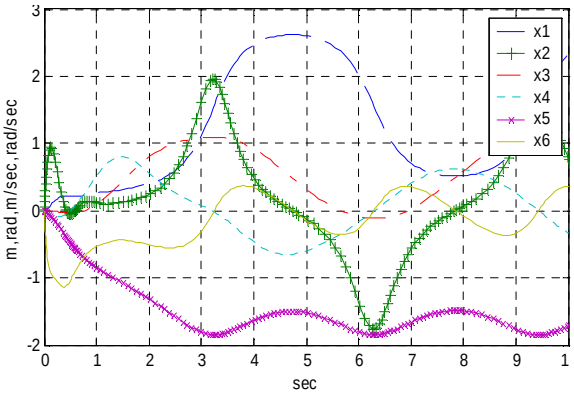

Fig. 4 The evolution of the state vector components

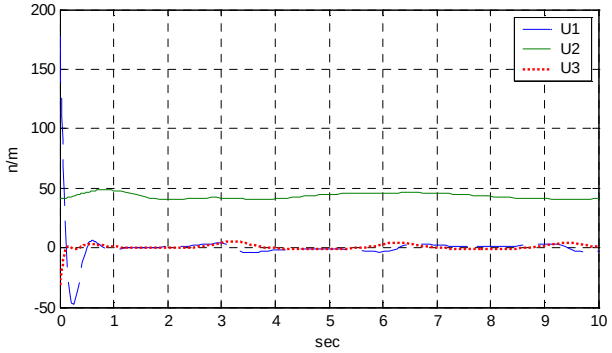

Fig. 5 The control vector components.

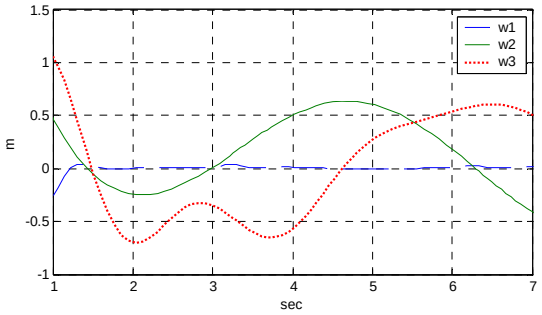

Fig. 6 The new control vector components

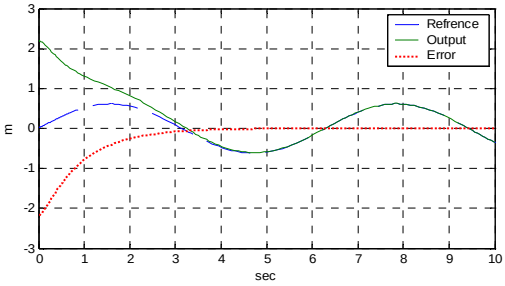

Fig. 7 First output, reference and error

To save a space in the paper, only first output, reference and error is given (Fig.7). The simulations confirm that the derived control forces the tracking errors to tend asymptotically to zero.

DISCUSSION AND FINAL COMMENTS

The software developed in this work can effectively support the procedure of control system design in symbolic form. Clearly, the bigger the dimension of the object model is, the longer and more complicated symbolic expressions will be obtained. For example, one can expect that the program will generate very large expressions describing the control laws for different flexible structures, such as given, for instance, in [8], when dynamics are approximated by a finite multi–dimensional model. The software helps to designer to solve the problem of the synthesis of nonlinear control systems and to simulate them easily. Practical implementation of derived control laws in regulators is also facilitated, because the control is expressed in explicit form and can be directly used in appropriate hardware or software.

Derived control laws can be of particular interest. Their symbolic presentation makes possible further analysis, optimization, simplification etc. If designer uses full nonlinear model of the dynamical object, evidently, the regulator of the system will be more complex than in the case of simplified model. The computer program pre-

sented here lets to designer fast to compare different regulators and to choose the perspective one.

The NFL has been taken as example to show the effectiveness of the symbolic computations. Nevertheless, other well-formalized methodologies can be considered. Future development of the approach and software for symbolic synthesis and modeling is supposed to be oriented to output feedback nonlinear, adaptive and robust control.

REFERENCES

1. A. Isidori, Nonlinear Control Systems. Berlin: Springer-Verlag, 1995.
2. H. Nijmeijer, A. van der Schaft, Nonlinear Adaptive Feedback Linearization of Systems. New York: Springer-Verlag, 1990.
3. C.I. Byrnes, F.D. Priscoli, A. Isidori, Output Regulation of Uncertain Nonlinear Systems. Boston: Birkhauser, 1997.
4. P.V. Kokotovic, I. Kanellakopoulos, A.S. Morse Adaptive Feedback Linearization of Nonlinear Systems // P.V. Kokotovic (ed.) Foundation of Adaptive Control. Berlin: Springer-Verlag, 1991.
5. B. Castillo-Toledo, S. Di Gennaro, A.G. Loukianov, J. Rivera "On the disrete-time modelling and control of induction motor with sliding modes," in Proc. American Control Conf., Boston, Massachusetts, 2004, pp. 2598-2604.
6. M. Vucobratovic, N. Kircansky, Scietific Fundamentals of Robotics Vol.1-3 . Berlin: Springer-Verlag, 1982-1985.
7. A.A. Dyda, S. Di Gennaro, "Adaptive trajectory control for underwater robot," in Proc.OCEANS-94 Osates, Brest, 1994.
8. S. Di Gennaro, A.A. Dyda, "Attitude control of a satellite with damping compensation of the flexible beam," in Proc. European Control Conf., Groningem, 1993, pp. 1656-1661.

