Nonlinear Control Design of Robot Arm

[1]*Peter ŠUSTER (3rd year)*
Supervisor: [1]Anna JADLOVSKÁ

[1]Dept. of Cybernetics and Artificial Intelligence, FEI TU of Košice, Slovak Republic

[1]peter.suster@tuke.sk, [1]anna.jadlovska@tuke.sk

***Abstract*— The aim of this paper is control algorithm design for a robot arm nonlinear model using an exact input - output feedback linearization method. In this paper is given the nonlinear model of the robot arm with two degree of freedom, which is base of the simulation model, and also a detailed description of the exact input - output feedback linearization method. The proposed control algorithm together with simulation model of the robot arm are implemented into control structure with purpose to track reference trajectory.**

***Keywords*— robot arm, exact feedback linearization method, pole placement method, integrator**

I. Introduction

The industrial robots have become a very important part of the manufacturing facilities in the various fields of industry. The robot arms are used for different kinds of welding, assembling, painting and other industrial applications. Therefore, the high level control is required for these applications, which provide the desired precision of the robot arm. Several approaches were used to control algorithm design for robot arms as robust control [2], or genetic algorithm for optimization of trajectory [3] and etc.

In this paper will be presented application of the exact input - output feedback linearization method for control algorithm design for nonlinear simulation model of robot arm with two degree of freedom with purpose of control on steady states defined by a reference trajectory. The proposed control algorithm together with robot arm simulation model are implemented into control structure, which is then verified in the Matlab/Simulink program language.

II. Exact Input-Output Feedback Linearization Method for MIMO System

This part described the exact input - output feedback linearization method for nonlinear dynamic MIMO of n - order with m - number of inputs and outputs. The basic condition for using exact linearization method is nonlinear MIMO system described in the affine form

$$\begin{aligned} \dot{x}(t) &= f(x,t) + g_1(x,t)u_1(t) + \ldots + g_m(x,t)u_m(t) \\ y_1(t) &= h_1(x,t) \\ &\ldots \\ y_m(t) &= h_m(x,t) \end{aligned} \quad (1)$$

$i = 1..m$ - ith inputs
$j = 1..m$ - jth outputs

where $x(t) \in R^n$ is state vector, $u_i(t)$ is control input, $y_j(t)$ is system output, $f(x,t)$, $g_i(x,t)$ a $h_j(x,t)$ are smooth nonlinear functions. For better overview, further will not write dependence of variables on the time t.

The essence of the exact input - output feedback linearization method is in finding a input transformation in the shape

$$u_i = \alpha_i(x) + \beta_i(x)v_i \quad (2)$$

where v_i is new input, $\alpha_i(x)$, $\beta_i(x)$ are nonlinear functions, and in defining a state transformation z_i in the shape

$$\hat{z}_i = T_i(x)\,. \quad (3)$$

The linear relationship is then created among outputs y_j and the new inputs v_i, and the interactions are removed between original inputs and outputs. That is advantage of this method, which allows to design control algorithm for each subsystems with input v_i and output y_j and independent of each other by synthesis methods for SISO systems [1].

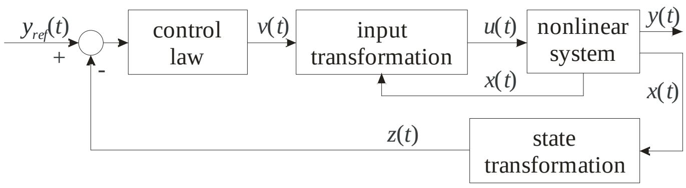

Figure 1 Control structure using exact linearization method

Principle of the exact input - output method feedback linearization method is based on repeatedly derivative of output y_j until input signals appear in the expression of derivation. The Lie derivatives are used for the calculation of individual derivatives of outputs, which are marked as $L_f h$ and $L_g h$. The first derivative has the form

$$\dot{y}_j = L_f h_j(x) + \sum_{i=1}^{m} L_{g_i} h_j(x) u_i \quad (4)$$

where:

$$L_f h_j(x) = \frac{\partial h_j}{\partial x} f(x) \ , \ L_{g_i} h_j(x) = \frac{\partial h_j}{\partial x} g_i(x)$$

If expression $L_{g_i} h_j(x) = 0$ for all i, then the inputs have not appeared in the derivation and is necessary continues derivative of the output y_j. Precondition, the number r_j represents number of derivatives, that is needed, that at least one input has appeared in the derivation $y_j^{r_j}$ i.e. $L_{g_i} L_f^{r_j - 1} h_j(x) \neq 0$ at least for one i, then the resulting derivation has shape

$$y_j^{r_j} = L_f^{r_j} h_j(x) + \sum_{i=1}^{m} L_{g_i} L_f^{r_j-1} h_j(x) u_i \quad (5)$$

This approach must be done for each output y_j. When the derivation is finished, the result are m equations, which can be written in the form

$$\begin{bmatrix} y_1^{r_1} \\ \dots \\ y_m^{r_m} \end{bmatrix} = \begin{bmatrix} L_f^{r_1} h_1(x) \\ \dots \\ L_f^{r_m} h_m(x) \end{bmatrix} + E(x) \begin{bmatrix} u_1 \\ \dots \\ u_m \end{bmatrix} \quad (6)$$

where $E(x)$ is m x m matrix of shape

$$E(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1 & \dots & L_{g_m} L_f^{r_1-1} h_1 \\ \dots & \dots & \dots \\ L_{g_1} L_f^{r_m-1} h_m & \dots & L_{g_m} L_f^{r_m-1} h_m \end{bmatrix} \quad (7)$$

If matrix $E(x)$ is regular, then it is possible to define the input transformation in the shape

$$\begin{bmatrix} u_1 \\ \dots \\ u_m \end{bmatrix} = -E^{-1}(x) \begin{bmatrix} L_f^{r_1} h_1(x) \\ \dots \\ L_f^{r_m} h_m(x) \end{bmatrix} + E^{-1}(x) \begin{bmatrix} v_1 \\ \dots \\ v_m \end{bmatrix}. \quad (8)$$

and state transformation

$$\begin{bmatrix} \hat{z}_1 \\ \dots \\ \hat{z}_m \end{bmatrix} = \begin{bmatrix} T_1(x) \\ \dots \\ T_m(x) \end{bmatrix} = \begin{bmatrix} \left[h_1(x), L_f h_1(x), \dots, L_f^{r_1-1} h_1(x)\right]^T \\ \dots \\ \left[h_m(x), L_f h_m(x), \dots, L_f^{r_m-1} h_m(x)\right]^T \end{bmatrix}. \quad (9)$$

After determination of input transformation (8) and state transformation (9) can be transformed nonlinear system (1) into linear form

$$\dot{z} = Az + bv = \begin{bmatrix} 0 & I_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & 0 & \dots & I_m \\ 0 & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} \hat{z}_1 \\ \vdots \\ \hat{z}_m \end{bmatrix} + \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 1 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \quad (10)$$

where I_j are unit matrix of the size r_j x r_j. The linear control algorithm is necessary to propose feedback control law for linear system (10) by linear method of synthesis, to ensure the desired behavior of the nonlinear system (Fig.1).

The relative order r_j exist for each subsystem in the exact input - output feedback linearization method, the resulting relative order is then defined by the sum of them such as

$$r = r_1 + r_2 + \dots + r_m \quad (11)$$

This part described the exact input - output feedback linearization method for the case when relative order r is equals to system order n.[1]

III. Second-Order Dynamic Model of Robot Arm

The robot arm model with two degrees of freedom (2DOF) is nonlinear MIMO system with two inputs (voltages for DC motors) and with two outputs (angular position of individual joint) (Fig. 2). The model is divided into two subsystems, *DC motors* subsystem, which consists two DC motors, that convert the input voltage u_j to corresponding motor torque τ_j, and *Robot arm* subsystem, which simulates the movement of robot arm joints (Fig.3).

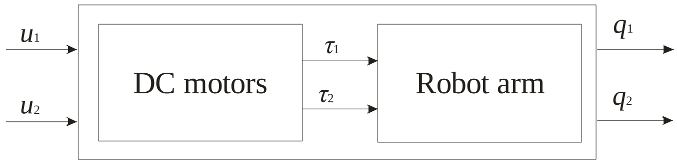

Figure 2 Model of robot arm - subsystems

The mathematical model of *DC motors* subsystem composed of two DC motors, which are described by differential equation of the shape [2]

$$R_j i_j + L_j \frac{di_j}{dt} + K_{b_j} \dot{q}_j = u_j \quad (12)$$
$$j = 1,2$$

where physical variables and parameters are:

- i_j - armature current of motor
- u_j - input voltage of motor
- $\dot{q}_j$ - angular position of rotor
- R_j - resistance of armature of motor
- L_j - inductance of armature of motor
- K_{b_j} - EMF constant

The motor output torque is then given

$$\tau_j = K_{t_j} i_j \qquad j = 1,2 \quad (13)$$

where τ_j is motor output torque of jth joint of robot arm and K_{t_j} is torque constant of jth motor. In the further, it is assumed, $K_{b_j} = K_{t_j}$ for both joints of the robot arm.

The general description of the dynamic of the robot arm with k - degree of freedom can be given by motion equation of the shape [4]

$$M(q)\ddot{q} + H(\dot{q}, q) = u \quad (14)$$

where:

- q - angular position vector [k x 1]
- $\dot{q}$ - angular velocity vector [k x 1]
- $\ddot{q}$ - angular acceleration vector [k x 1]
- $M(q)$ - matrix of inertia [k x k]
- $H(\dot{q}, q)$ - vector of damping, coriolis, centrifugal and gravitational force [k x 1]
- u - input torques vector [k x 1].

In the case 2DOF robot arm (Fig 3), which includes a *Robot arm* subsystem, can be the dynamic of the arm described follows

$$M(q) \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (15)$$

where τ_1, τ_2 are DC motors torque, q_1 and q_2 are angular position of individual joints. The matrix of inertia $M(q)$ and $H(\dot{q}, q)$ vector are in the shape

$$M(q) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} = \begin{bmatrix} (m_1 + m_2) l_1^2 & m_2 l_1 l_2 \cos(q_1 - q_2) \\ m_2 l_1 l_2 \cos(q_1 - q_2) & m_2 l_2 \end{bmatrix} \quad (16)$$

$$H(\dot{q}, q) = \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \end{bmatrix} = \begin{bmatrix} K_{q_1} \dot{q}_1 + m_2 l_1 l_2 \sin(q_1 - q_2) \dot{q}_2 \\ K_{q_2} \dot{q}_2 - m_2 l_1 l_2 \sin(q_1 - q_2) \dot{q}_1 \end{bmatrix} \quad (17)$$

where:

- m_1, m_2 - mass of arms

l_1, l_2 - length of arms

K_{q1}, K_{q2} - damping constants.

It is assumed that the total mass of the actual arm is concentrated at the end of each of them, as shows on the Fig.3.

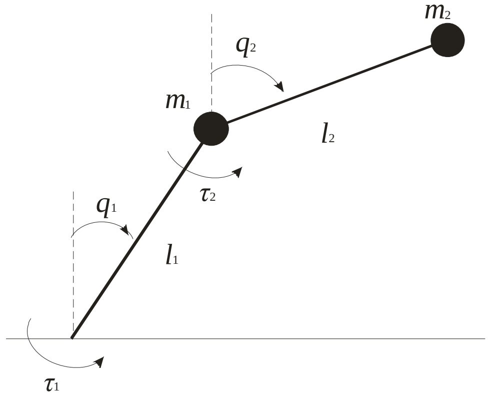

Figure 3 2DOF robot arm

Based on equations (12) to (17) was programmed simulation model of the nonlinear dynamic model 2DOF robot arm using the S - function block in the Matlab/Simulink language. The simulation model includes physical constraints and limits of the real model. The analysis of the simulation model was carried in open loop at defined input voltages u_1 and u_2 (Fig.4).

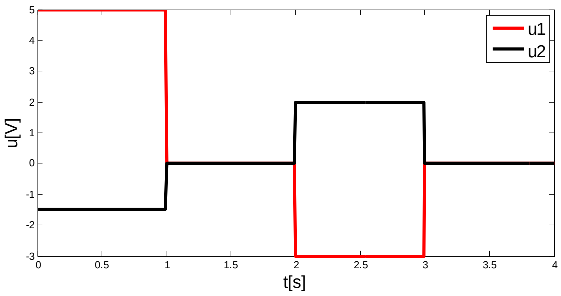

Figure 4 Analysis of 2DOF robot arm simulation model - input

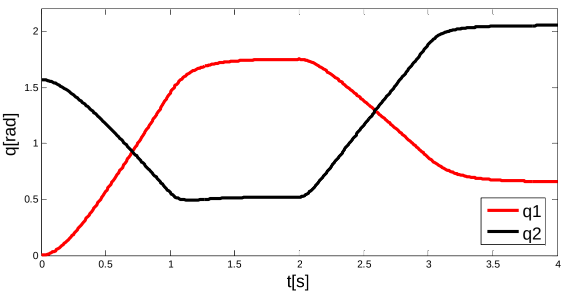

Figure 5 Analysis of 2DOF robot arm simulation model - output

The resulting graph of analysis (Fig.5) shows the possibility of further use of 2DOF robot arm simulation model in the control structure at the testing of the control algorithm using exact input - output feedback linearization method.

IV. Design of nonlinear control alforithm

The nonlinear dynamic model of the 2DOF robot arm was given in the previous part. This model is described by equations, which contain in itself a smooth nonlinear functions, the main assumption for the use of exact input - output feedback linearization method for the control algorithm design, which will deal in this part. The whole process of the control algorithm design is described in the following steps.

1. step - rewrite of the nonlinear model of the 2DOF robot arm to affine form (1).

By defining the state vector $x = (x_1, x_2, x_3, x_4, x_5, x_6) = (q_1, \dot{q}_1, i_1, q_2, \dot{q}_2, i_2)$, system input $u = (u_1, u_2)$ and system output $y = (q_1, q_2)$, then the affine form of the 2DOF robot arm nonlinear model has

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_2 \\ -invM_1 H + invM_1 \tau_1 \\ -\frac{R}{L} x_3 - \frac{K_b}{L} x_2 \\ x_5 \\ -invM_2 H + invM_2 \tau_2 \\ -\frac{R}{L} x_6 - \frac{K_b}{L} x_5 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{L} \end{bmatrix} u \tag{18}$$

$$y = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix}$$

where

$$invM = \begin{bmatrix} invM_1 \\ invM_2 \end{bmatrix} =$$

$$= \begin{bmatrix} \left[\frac{m_{22}}{m_{11}m_{22} - m_{12}m_{21}} \quad -\frac{m_{21}}{m_{11}m_{22} - m_{12}m_{21}} \right] \\ \left[-\frac{m_{12}}{m_{11}m_{22} - m_{12}m_{21}} \quad \frac{m_{11}}{m_{11}m_{22} - m_{12}m_{21}} \right] \end{bmatrix}$$

$$H = H(\dot{q}, q)$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} K_t x_3 \\ K_t x_6 \end{bmatrix}.$$

2. step - calculation of the Lie derivatives for individual outputs.

The program module has been programmed for this calculation in Matlab language using symbolic toolbox, that generated the individual derivatives for outputs y_1 and y_2. After calculation, the result are matrix of the $L_f h(x)$ derivatives and $E(x)$ matrix in the shape

$$L_f h(x) = \begin{bmatrix} L_f^0 h_1(x) & L_f^0 h_2(x) \\ L_f^1 h_1(x) & L_f^1 h_2(x) \\ L_f^2 h_1(x) & L_f^2 h_2(x) \\ L_f^3 h_1(x) & L_f^3 h_2(x) \end{bmatrix} \tag{19}$$

$$E(x) = \begin{bmatrix} L_{g_1} L_f^2 h_1 & L_{g_2} L_f^2 h_1 \\ L_{g_1} L_f^2 h_2 & L_{g_2} L_f^2 h_2 \end{bmatrix}. \tag{20}$$

After determining, that $E(x)$ matrix is regular, it was possible to define the input transformation

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -E^{-1}(x) \begin{bmatrix} L_f^3 h_1(x) \\ L_f^3 h_2(x) \end{bmatrix} + E^{-1}(x) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{21}$$

and state transformation

$$\begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \end{bmatrix} = \begin{bmatrix} T_1(x) \\ T_2(x) \end{bmatrix} = \begin{bmatrix} \left[h_1(x), L_f^1 h_1(x), L_f^2 h_1(x) \right]^T \\ \left[h_2(x), L_f^1 h_2(x), L_f^2 h_2(x) \right]^T \end{bmatrix}. \tag{22}$$

3. step - transformation of the nonlinear system (18) to linear form (10) and linear control law.

The linear control law was proposed for to ensure the desired behavior of the nonlinear model (18) using pole placement with integrator method in the shape

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = -\begin{bmatrix} K_{1-3} & K_{i1} \\ K_{4-6} & K_{i2} \end{bmatrix} \begin{bmatrix} z_{1-3} \\ \int y_{ref1} - y_1 \\ z_{4-6} \\ \int y_{ref2} - y_2 \end{bmatrix} \quad (23)$$

where:

z_{1-3}, z_{4-6} - state vectors
y_{ref1}, y_{ref2} - reference trajectories
y_1, y_2 - model outputs
K_{1-3}, K_{4-6} - vectors of gains for individual states
K_{i1}, K_{i2} - vectors of gains for integrator outputs

Therefore, it was necessary to extend the state description of the linear form (10) into following form

$$\begin{bmatrix} \dot z_1 \\ \dot z_2 \\ \dot z_3 \\ \int \dot z_1 \\ \dot z_4 \\ \dot z_5 \\ \dot z_6 \\ \int \dot z_4 \end{bmatrix} = \begin{bmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \int z_1 \\ z_4 \\ z_5 \\ z_6 \\ \int z_4 \end{bmatrix} + \begin{bmatrix} 0&0 \\ 0&0 \\ 1&0 \\ 0&0 \\ 0&0 \\ 0&0 \\ 0&1 \\ 0&0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} . (24)$$

4. step - implementation of the proposed control algorithm and his testing in the control structure.

The resulting input transformation (21) and state transformation (22) together with linear control law (23) are implemented into programmed simulation scheme for control of the 2DOF robot arm nonlinear model (Fig.6)

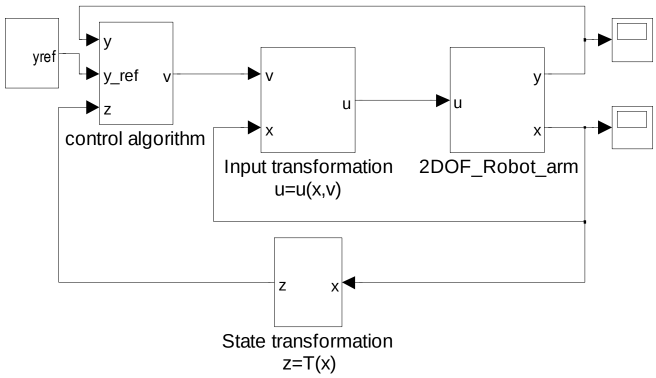

Figure 6 Simulation scheme for control 2DOF robot arm nonlinaer model using exact input - output feedback linearization method

The resulting graph of tracking refrence trajectory, which is represented the step change between steady state, when using proposed control algorithm with using exact input - output feedback linearization method is on the Fig.7.

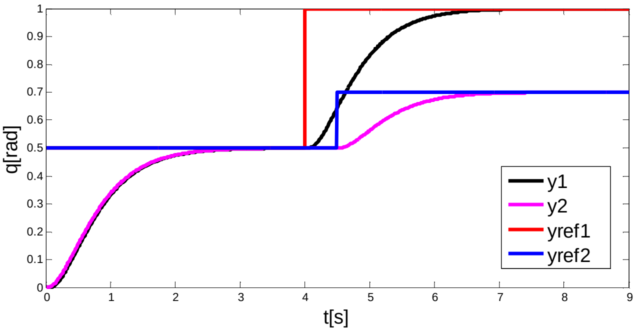

Figure 7 2DOF robot arm nonlinear mode response to track reference trajectory

V. Conclusion

This paper presented the control algorithm design for nonlinear simulation model of the 2DOF robot arm using the exact input - output feedback linearization method and pole placement with integrator method. The proposed control algorithm together with simulation model, which includes physical constraints and limits of the real model, was implemented into control structure and was verified in Matlab/Simulink program language. The resulting graph shows, that output of the model tracks step change of the reference trajectory and therefore can be considered, this approach is suitable for solution problem of control for 2DOF robot arm. The obtained knowledge from field of control algorithm design for MIMO systems using methods of nonlinear synthesis and also the proposed program module will be part of my dissertation work named Design of Effective Software Tools for Control and Analysis of Nonlinear Systems

Acknowledgment

This contribution is the results of the Vega project implementation: Dynamic hybrid architectures of the multiagent network control systems (No. 1/0286/11), supported by the Scientific Grant Agency of Slovak Republic (50%) and project implementation Development of the Center of Information and Communication Technologies for Knowledge Systems (project number: 26220120030) supported by the Research & Development Operational Program funded by the ERDF (50%).

References

[1] M. Razím and J. Štecha, “Nonlinear systems”, ČVUT, 1997, (in Czech language).
[2] G. Liu and A.A. Goldenberg. "*Robust Control of Robot Manipulators Incorporating Motor Dynamics*," Proc. of IEEERSJ International Workshop on Intelligent Robots and Systems, Japan, 1993, pp. 68-74.
[3] S. A. Mazhari and S. Kumar, “*PUMA 560 Optimal Trajectory Control using Genetic Algorithm, Simulated Annealing and Generalized Pattern Search Techniques*,” International Journal of Electrical, Computer and Systems Engineering, Vol. 2, No. 1, 2008, pp. 71-80.
[4] R. Palm, D. Driankov and H. Hellndoorn, "Model Based Fuzzy Control", Springer - Verlag Berlin Heidelberg, 1997
[5] S. Senini and P.Wolfs, "*Pole Placement Design of Integral Controller for Shunt Hybrid Active Filter*", Australasian Universities Power Engineering Conference - AUPEC 2002, 29 sept.- 02 oct. 2002, pp. 160 - 165, Melbourne, Australia