

## Application Results Identification Based on Genetic Algorithm in Nonlinear Control Design of Magnetic Levitation System

P.Šuster<sup>1</sup>, A. Jadlovská<sup>1</sup>,

<sup>1</sup> Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, TU Košice,

Letná 9, Košice

E-mail : peter.suster@tuke.sk, anna.jadlovska@tuke.sk

### Abstract:

In this paper is presented the SISO laboratory model of Magnetic levitation in terms of mathematic description, which allows verifying the different approaches of identification and control. The nonlinear simulation model of the Magnetic levitation based on the mathematical model of the Magnetic levitation system is described. The unknown parameters of the Magnetic levitation model are identifying with using the genetic algorithm or direct measurement on the laboratory model and validation the obtained model parameters is performed after the identification. In this paper are also presented the control results of Magnetic levitation simulation and laboratory model with using the optimal state control with integrator method and the exact feedback linearization input/output method.

### INTRODUCTION

In this paper is presented the modeling, identification and control algorithm design for Magnetic levitation laboratory model, which is located in the Laboratory of Cybernetics in the Department of Cybernetics and Artificial Intelligence, FEI TU of Kosice. Magnetic levitation model is an example of nonlinear, in open loop unstable SISO system with fast dynamics. The identification of unknown parameters and control algorithm design is very difficult for these properties. The problem of the identification of the Magnetic levitation model was described in the [1] and [3]. However, the proposed identification method, which are presented in the [1] for determination of the values for various model parameters required the specially prepared experiments and also is necessary often to repeat these experiments. In order to simplify the process of the identification in this paper is proposed the genetic algorithm, which is used to identify of the some model parameters of the Magnetic levitation. The other model parameters are obtained by the direct measurement of the laboratory model. Another reason for the identification of these model parameters using the genetic algorithm is also the existence the Magnetic levitation mathematical model. The identification of a nonlinear model parameters using genetic algorithm is presented in [2], [4], [5].

Some linear and nonlinear approaches were used to design control algorithm for Magnetic levitation system as a linear state control [6], an adaptive control [7] or an exact linearization [8]. In this paper are also presented the obtained results of the control of the simulation and laboratory model of the Magnetic levitation using the methods of synthesis - optimal state control with integrator method and exact feedback linearization input/output method.

The paper is organized as follows. The Magnetic levitation laboratory model and its mathematical model are shown in the first and second part. The third part describes the identification process of the Magnetic levitation simulation model and validation of the obtained parameters. In the last part is described the control algorithm design for Magnetic levitation model using optimal state control with integrator method and exact feedback linearization input/output method.

### LABORATORY MODEL OF MAGNETIC LEVITATION

The laboratory model of the Magnetic levitation (ML) is shown in the Fig. 1. It consists the education model of the Magnetic levitation and the laboratory card MF614, which is used for communication with the control PC. The essence of the whole system, with the proposed control algorithm in the control PC, is to keep levitate the steel ball in the air in the desire position by using electromagnetic force, which is produced from electric current going through the coil with soft magnetic core.

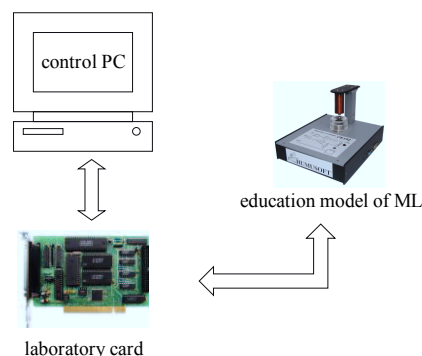


Fig. 1: CE 152 Magnetic levitation laboratory model of Humusoft

The Magnetic levitation education model is composed of the following subsystems: *power amplified, ball and coil and inductive position sensor*. The model is connected to the control PC via the A/D and D/A converters that are located on the laboratory card MF614, which is connected to PC via the PCI interface. The converters are considered as part of the model ML in this case (Fig. 2).

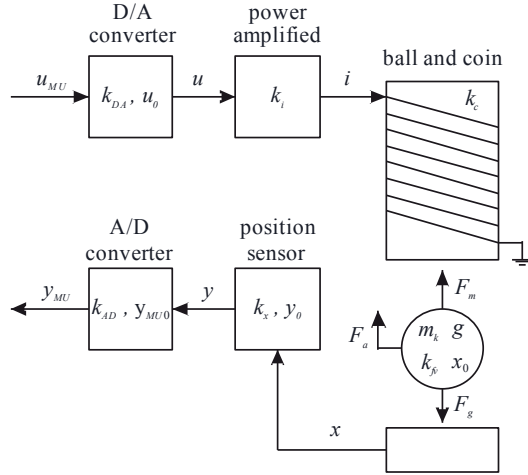


Fig. 2: The internal structure of the education model of the ML

The input is unified signal  $u_{MU}$ , which is transformed to signal  $u$  by the D/A converter, and it enters to power amplified, which converts it to current  $i$ . The current  $i$  pass through coil and thereby is generated electromagnetic force  $F_m$ , which acting to the ball in the opposite direction as gravitational force  $F_g$ . The ball will levitate between the coin and the sensor in the certain position  $x$  in the case, that the balance of these forces. The position of the ball  $x$  is converted to signal  $y$  by the inductive position sensor, which is transformed to unified signal  $y_{MU}$  by the A/D converter.

The proposed experiments for identification of the parameters of the simulation model and also verification of the designed control algorithms for the Magnetic levitation laboratory model are done in the programming language Matlab/Simulink using the Real Time Toolbox.[9]

## MATHEMATICAL MODEL OF MAGNETIC LEVITATION

The each subsystems are described by the differential and linear equations respective.

The mathematical model of the *ball and coil* subsystem is described by the second order nonlinear differential equation

$$\underbrace{m_k \ddot{x}(t)}_{F_a} + \underbrace{k_{fv} \dot{x}(t)}_{F_{fv}} = \underbrace{\frac{i(t)^2 k_c}{(x_0 - x(t))^2}}_{F_m} - \underbrace{m_k g}_{F_g} \quad (1)$$

where:  $i(t)$  - electric current [A]  
 $x(t)$  - ball position [m]  
 $m_k$  - mass of ball [kg]  
 $k_c$  - coil constant [A/V]  
 $x_0$  - coil offset [m]  
 $g$  - gravity constant [ $m/s^2$ ]  
 $k_{fv}$  - damping constant [N/m.s]  
 $F_{fv}$  - damping force [N]  
 $F_m$  - electro magnetic force [N]  
 $F_g$  - gravitational force [N]  
 $F_a$  - accelerating force [N]

Position of the ball in the magnetic field is controlled by electric current  $i(t)$ , which is generated from the power amplified. The power amplified is designed as a source of constant current, and its time constant is neglected with respect to system dynamics. The *power amplified* subsystem can be described by the following linear equation:

$$i(t) = k_i u(t) \quad (2)$$

where:  $u(t)$  - input voltage [V]  
 $k_i$  - gain of power amplified [A/V]

The *inductive position sensor* is used to determine the ball position, which is approximated by a linear equation

$$y(t) = k_x x(t) + y_0 \quad (3)$$

where:  $y(t)$  - sensor output voltage [V]  
 $x(t)$  - ball position [m]  
 $k_x$  - sensor gain [V/m]  
 $y_0$  - sensor offset [V]

The D/A converter transform the digital unified signal  $u_{MU}$  from PC to analog voltage signal  $u$  and conversely the A/D converter transform analog voltage signal  $y$  to unified digital signal  $y_{MU}$ , which is then processed in the PC. The behavior of the D/A and A/D converter can be described by the linear equations:

*D/A converter:*

$$u(t) = k_{DA} u_{MU}(t) + u_0 \quad (4)$$

*A/D converter:*

$$y_{MU}(t) = k_{AD} y(t) + y_{MU0} \quad (5)$$

where:  $u(t)$  - D/A converter output voltage [V]  
 $u_{MU}(t)$  - D/A converter input voltage [MU]  
 $k_{DA}$  - D/A converter gain [V/MU]  
 $u_0$  - D/A converter offset [V]  
 $y_{MU}(t)$  - A/D converter output voltage [MU]  
 $y(t)$  - A/D converter input voltage [V]  
 $k_{AD}$  - A/D converter gain [MU/V]  
 $y_{MU0}$  - A/D converter offset [MU]

Based on equations (1) to (5), which describe mathematical model of the Magnetic levitation system was programmed simulation scheme of the Magnetic levitation nonlinear model in the Matlab/Simulink language.(Fig. 3).

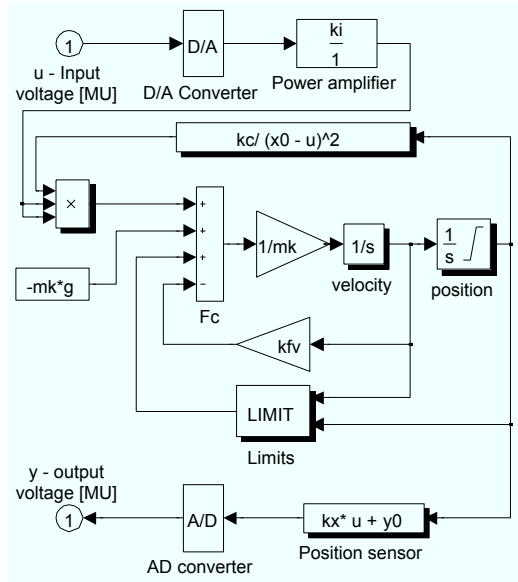


Fig. 3: The simulation scheme of the Magnetic levitation model

The ball and coil subsystem was extended about LIMITY block, because of model constants have to vary according the ball position. [1]

## IDENTIFICATION OF MATHEMATICAL MODEL PARAMETERS

In this part is given identification of the mathematical model parameters of ML (Fig.3).

The proposed simulation model of ML has 10 parameters, of which 6 parameters ( $k_{DA}$ ,  $u_0$ ,  $k_{AD}$ ,  $y_{MU0}$ ,  $k_x$ ,  $y_0$ ) are determined by direct measurement or experiments on the real model and other 4 parameters ( $k_i$ ,  $k_c$ ,  $x_0$ ,  $k_f$ ) are identified using a genetic algorithm. Subsystem parameters of the D/A and the A/D converters, which provide connection between model and control PC are directly obtained from technical parameters of the MF614 laboratory card. The D/A converter transform digital signal in the machine unit in the range of  $u_{MU} \in \langle 0;0,5 \rangle$  MU to voltage signal  $u \in \langle 0;10 \rangle$  V and therefore D/A converter gain  $k_{DA} = 20$  V/MU and offset  $u_0 = 0$ V. Similarly, the A/D converter, which transform voltage signal in the range of  $y \in \langle 0;5 \rangle$  V to machine unit signal  $y_{MU} \in \langle 0;1 \rangle$  MU, where then A/D converter gain  $k_{AD} = 0,2$  MU/V and offset  $y_{MU0} = 0$  MU.

For identification of the inductive position sensor parameters was designed experiment, in which was measured output signal  $y_{MU}$  of model. The inductive position sensor is approximated by the linear

function, it was sufficient to measure output signal values in the both of the limits of the ball position and then calculate corresponding sensor output voltage by equation (6), which was created by the substituting equation (3) into equation (5).

$$y_{MU} = k_{AD}k_x x + k_{AD}y_0 \quad (6)$$

The measured and calculated values for the identification of the sensor parameters are listed in Tab.1.

Tab. 1: Measured and calculated values

$i$	$x_i$ [m]	$y_{MU_i}$ [-]	$y_i$ [V]
1	0	0,0034	0,017
2	0,005	0,9375	4,6705

Based on the values listed in Tab. 1, it is possible to calculate sensor gain and offset values:

$$y_0 = y_1 = 0,017 \text{ V} \quad (7)$$

$$k_x = \frac{y_2 - y_1}{x_2 - x_1} = \frac{4,6705 - 0,017}{0,005} = 930,7 \text{ V/m} \quad (8)$$

The parameters of the ball and coil subsystem, mass of ball  $m_k = 8,27e-3$  kg and gravity constant  $g = 9,81$  m/s<sup>-2</sup>. [1]

## Genetic algorithm identification of parameters of model

The parameters  $k_c$  - coil constant,  $x_0$  - coil offset,  $k_f$  - damping constant for ball and coil subsystem and  $k_i$  - gain of power amplified for power amplified subsystem, which could not directly measure on the laboratory model, were identified using a genetic algorithm (GA). The identification structure for parameter estimation using GA is on the Fig. 4. at the beginning, the estimated parameters from GA are set in the simulation model, which are placed in the string  $\theta_i$ , where  $i = 1, \dots, M$ , where  $M$  denotes number of the strings in the one generation. The same signal is coming on the real model input and simulation model and then is compared output of the real model  $y(k)$  and output of the simulation model  $\hat{y}(k)$ . The  $k$  denotes the time instant  $k = jT$ , where  $T$  is constant sample period,  $j = 1, \dots, N$  and  $N$  is total number of the samples. Subsequently, based on the equation (9) and from generated errors  $e_j$  is determined performance index  $J_i$  for  $\theta_i$  string.

$$J_i = \sqrt{\sum_{j=1}^N |e_j|^2} \quad (9)$$

In this paper, GA is used to minimize the performance index  $J_i$ . For each  $i$ -th string in the one generation is then calculated its fitness function, which has the following shape:

$$fitness_i = \frac{1}{J_i} \quad (10)$$

The all string  $\theta_i$  are compared to each other according to their fitness function  $fitness_i$ , and are selected the strings to the new generation, while a genetic crossover and mutation operations are performed

over some of string. Thus, the new generation of the strings is obtained, and it must be added, that the selection of the new generation has higher probability "survive" the most successful strings, but any probability have also the less successful strings. This whole process is then repeated until required number of repetitions is fulfilled or tolerance between simulation and real model is in the desired accuracy (Fig. 4). [2], [10]

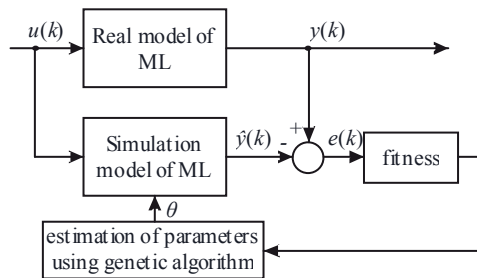


Fig. 4: Identification structure using genetic algorithm

The identification of the model parameters using GA was used the proposed *program module for GA* in the Matlab/Simulink language. The input parameters for program module are:

- number of strings in the one generation : 50
  - number of gens in the one string :  $[k_i, k_c, x_0, k_{fv}]$
  - type of selection : roulette wheel
  - type of crossover : one point
  - crossover probability : 0,8
  - mutation probability : 0,1
  - range of parameters for 1. generation  $\langle \min; \max \rangle : k_i \square \langle 0,2; 0,5 \rangle, k_c \square \langle 1e-6; 2,5e-6 \rangle, x_0 \square \langle 0,007; 0,009 \rangle, k_{fv} \square \langle 0,01; 0,06 \rangle$
  - type of completion : number of generations - 100
- The results of GA, after condition was fulfilled, are the following values for individual parameters:  $k_i = 0,3122 \text{ A/V}, k_c = 1,0175e-6 \text{ A/V}, x_0 = 0,0075 \text{ m}, k_{fv} = 0,0838 \text{ N/m.s}$ . The input signal used for identification is on the Fig. 5 and evolution process of the fitness function is shown in the Fig. 6.

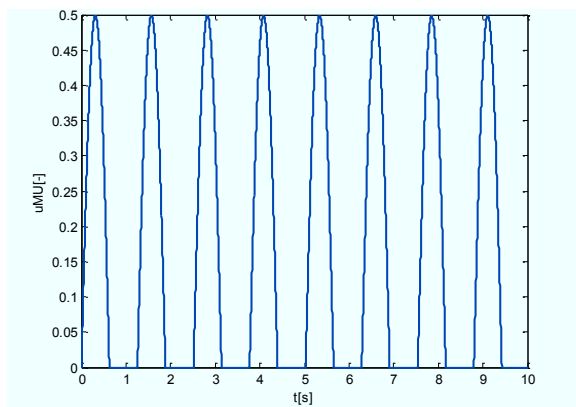


Fig. 5: Input signal for identification of ML model parameters using GA

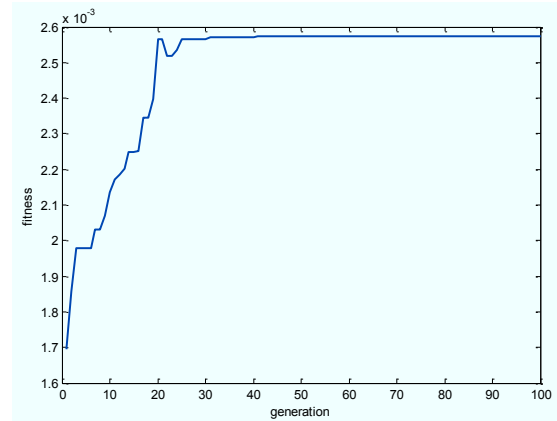


Fig. 6: Evolution process of the fitness

The resulting parameters of the simulation model, which are used in the simulation, are shown in the Tab. 2: blue - direct measurement, red - genetic algorithm, green - unidentified parameters

Tab. 2: Parameters of simulation model ML

parameter	value	dimension
$k_{DA}$	20	V/MU
$u_0$	0	V
$k_{AD}$	0,02	MU/V
$y_{MU0}$	0	MU
$k_x$	390,1	V/m
$y_0$	0,017	V
$k_i$	0,3122	A/V
$k_c$	1,0175e-6	A/V
$x_0$	0,0075	M
$k_{fv}$	0,0838	N/m.s
$m_k$	8,27*e-3	Kg
$g$	9,81	m/s <sup>-2</sup>

### Validation of ML model parameters

The validation of the identified parameters was tested in the open loop with changed input signal and also in the feedback structure using a discrete PID controller. The results are time responses of the ball position of the simulation and real model. The time response of the simulation and real model in the open loop with new input signal (Fig. 7) is shown in the Fig. 8.

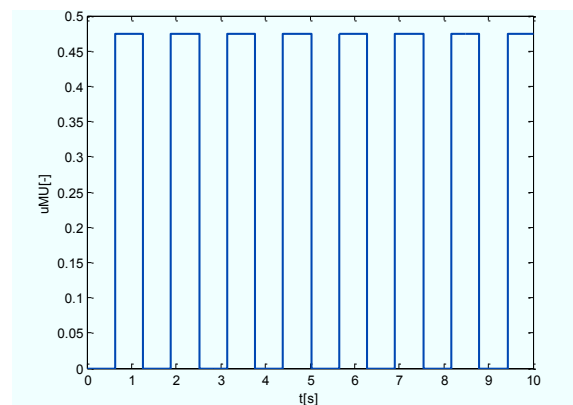


Fig. 7: Input signal for testing of identified parameters of ML model

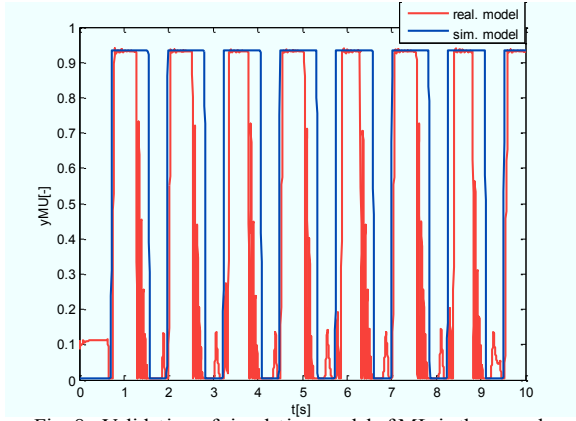


Fig. 8: Validation of simulation model of ML in the open loop - output time responses

For testing of identified parameters of ML model in the feedback structure (Fig. 9) was used with the discrete PID controller, whose transfer function is in the shape

$$G_{dPID} = K_p + K_i \frac{zT_s}{z-1} + K_d \frac{(z-1)}{zT_s} \quad (11)$$

where  $z$  is the operator  $Z$  - transformation,  $K_p$  is proportional gain of controller,  $K_i$  - integral gain of controller,  $K_d$  - derivative gain of controller and  $T_s$  is sampling period. The each gain of the discrete PID controller was designed by experiment [1] and have the following values :  $K_p = 1$ ,  $K_i = 10$ ,  $K_d = 0,03$ ,  $T_s = 0,002s$ .

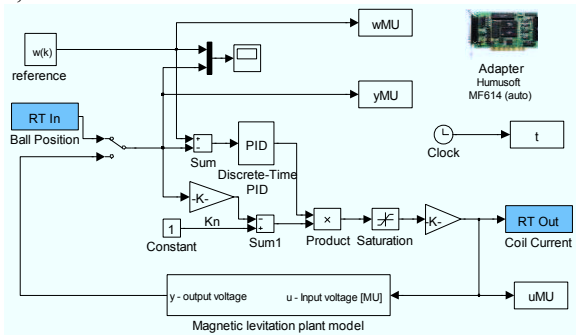


Fig. 9: Simulation scheme of feedback structure using discrete PID control for simulation and real model of ML

The following figures are shown the time responses, for testing of the identified parameters in the feedback structure.

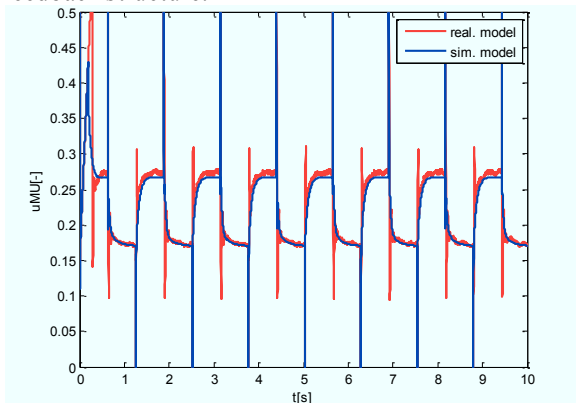


Fig. 10: Validation of simulation model of ML in the feedback structure - input time responses

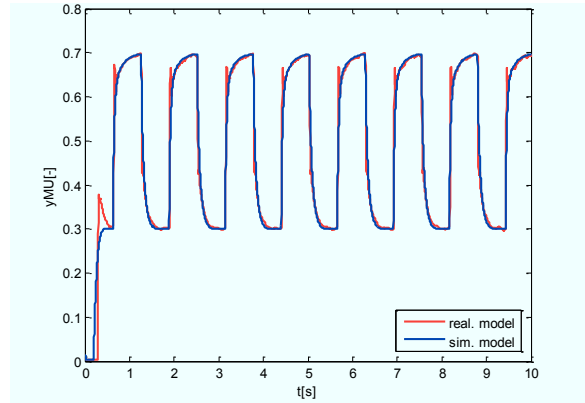


Fig. 11: Validation of simulation model of ML in feedback structure - output time responses

The resulting time responses, either in the open loop (Fig. 7, Fig. 8) or in the feedback structure (Fig. 10, Fig. 11) show, that the simulation model with identified parameters tracks the behavior of the real model with sufficient accuracy, of which show the possibility of further use of simulation model of ML in the control structures using linear and nonlinear synthesis method. The control algorithms obtained in this manner can be used directly for control of the real model, which reduce time of control design and also decreasing probability of some fault of the real model.

## CONTROL DESIGN OF ML MODEL BASED ON LINEAR/NONLINEAR METHOD OF SYNTHESIS

In this part of the paper is described the basic principle or a brief description of a control algorithms, which has been proposed and verified on the simulation model and then on the ML laboratory model. The optimal state control with integrator method and exact feedback linearization input/output method are presented for control algorithm design for Magnetic levitation model with the purpose of tracking the reference trajectories.

The both of used methods assume a model written in the state space form in the shape:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= h(x(t), t) \end{aligned} \quad (12)$$

where  $x(t) \in R^n$  is state vector,  $u(t)$  is control input,  $y(t)$  is system output,  $f(x(t))$  and  $h(x(t))$  are smooth nonlinear function. Therefore, based on equations (1) to (5) was created nonlinear state space form, which describes the dynamics of the ML laboratory model, in the shape :

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{k_i^2 k_{DA}^2 k_c u(t)^2}{m_k (x_1(t) - x_0)^2} - g - \frac{k_{fv}}{m_k} x_2(t) \end{aligned} \quad (13)$$

$$y(t) = k_{AD} k_x x_1(t) + k_{AD} y_0$$

where state vector is  $x(t) = (x_1(t), x_2(t)) = (x(t), \dot{x}(t))$ , input  $u(t) = u_{MU}(t)$  and output  $y(t) = y_{MU}(t)$ . For better overview, further will not write dependence of variables on the time  $t$ .

### Linear method of synthesis - optimal state control with integrator

The discrete state space form of the linear system is used for discrete optimal state control with integrator design (LQI control) in the shape

$$\begin{aligned} x(k+1) &= Fx(k) + Gu(k) \\ y(k) &= Cx(k) \end{aligned} \quad (14)$$

where is necessary propose control in the shape  $u(k) = -K_{LQI}x_{LQI}(k) = -K_{LQI}[x(k) \quad x_i(k)]$

which minimizes the functional form

$$J = \sum_{k=1}^M x_{LQI}^T(k) Q_{LQI} x_{LQI}(k) + u(k)^T R_{LQI} u(k) \quad (16)$$

Where  $M$  is integer,  $K_{LQI}$  is gain vector,  $Q_{LQI}$ ,  $R_{LQI}$  are weighting matrices and the output of integrator  $x_i(k)$  is calculated by equation

$$x_i(k+1) = x_i(k) + T_s(w(k) - y(k)) \quad (17)$$

where  $T_s$  is sampling period. The control structure for using optimal state control with integrator is in the Fig. 12. [11]

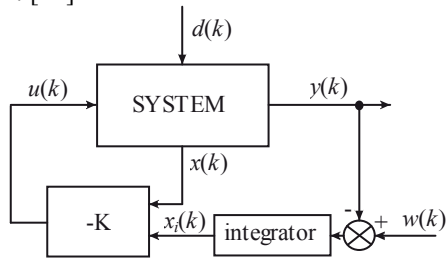


Fig. 12: The control structure for using LQI control

The transformation of the nonlinear equations (13) into Taylor series around the chosen operating point  $x_{10} = 0,0025\text{m}$ ,  $x_{20} = 0$ ,  $u_{MU0} = 0,2261\text{MU}$  was obtained the state space form of the linear model, which approximates the dynamics of the ML laboratory model. The Matlab function  $c2d()$  was used for obtain the discrete state space form (14), when sampling period was  $T_s = 0,002\text{s}$ .

The Matlab function  $lqi()$  was used for actual control algorithm design. If the input into function  $lqi()$  are matrices of the discrete state space form (14) and the weighting matrices in the shape  $Q_{LQI} = [0 \ 0 \ 0; 0 \ 0 \ 0; 0 \ 0 \ 1000]$ ,  $R_{LQI} = 10$ , then the result from function are gain vector  $K_{LQI}$  and the feedback loop roots  $root\_uro$  in the shape

$$\begin{aligned} K_{LQI} &= \begin{bmatrix} 129,2398 \\ 1,5599 \\ 8,6479 \end{bmatrix} \\ root\_uro &= \begin{bmatrix} 0,855 \\ 0,9247 + 0,0445i \\ 0,9247 - 0,0445i \end{bmatrix} \end{aligned} \quad (18)$$

The discrete Kalman estimator was used for the estimate of the ML simulation and laboratory model states, which is necessary to know for optimal state control. The discrete Kalman estimator was proposed using the Matlab function  $kalman()$ . The input into function  $kalman()$  are matrices of the discrete state space form (14) and the weighting matrices  $Q_{est}$ ,  $R_{est}$  and output of function is estimator gain vector  $L$  and estimator roots. For the proposal of the estimator were chosen the weighting matrices  $Q_{est} = 100000$  and  $R_{est} = 0,01$  and then estimator gain vector  $L$  has the following values:

$$L = \begin{bmatrix} 0,0159 \\ 5,2511 \end{bmatrix} \quad (19)$$

The control structure for testing the proposed optimal state control with integrator to control the ML simulation and laboratory model with purpose of to ensure the desired position of ball in the magnetic field of the coil is in the Fig. 13.

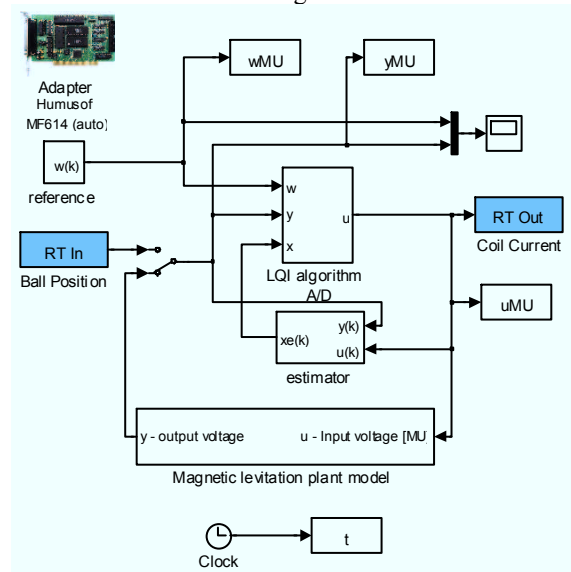


Fig. 13: Simulation scheme of control structure with LQI control for ML simulation/laboratory model

The time responses of the inputs and resulting outputs of tracking of reference trajectories with using the proposed method of the linear synthesis are in the Fig. 14 (a, b) and Fig. 15 (a, b).



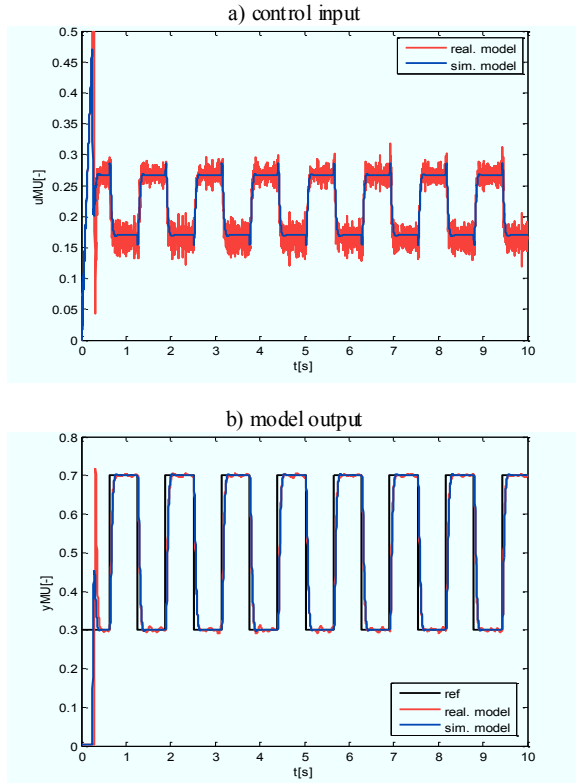


Fig. 14: Time responses of the ML simulation and laboratory model with using LQI control

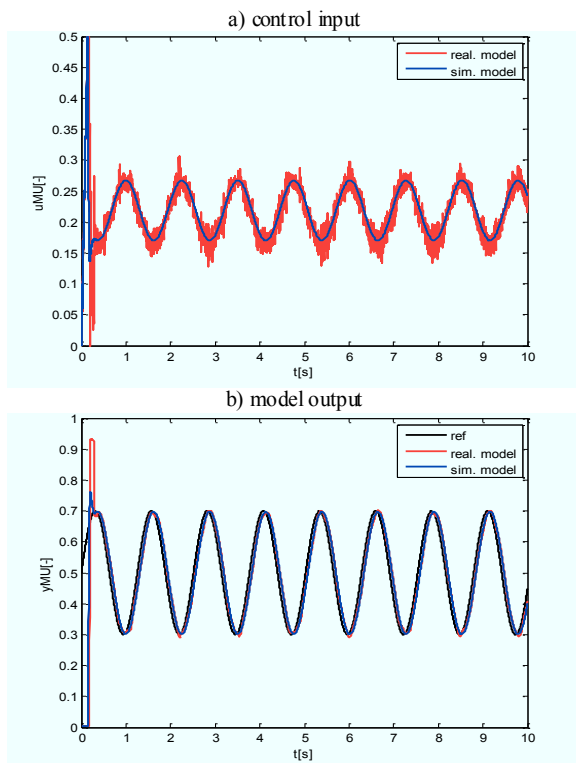


Fig. 15: Time responses of the ML simulation and laboratory model with using LQI control

### Nonlinear method of synthesis Exact feedback linearization input/output method

The exact feedback linearization input/output method is one of the structural nonlinear methods of the synthesis. This method based on the idea to compensate nonlinearities in the system (12) by adding nonlinear transformation (input and state) thus that the resulting system will be as a linear to respect to a new input  $v$  and output  $y$  and can be described by a linear state space form in the shape

$$\begin{aligned} \dot{z} &= A_{EL}z + B_{EL}v \\ y &= C_{EL}z \end{aligned} \quad (20)$$

Then, for this linear model (20) is possible to propose the control algorithm using a suitable linear method of synthesis (Fig. 16).

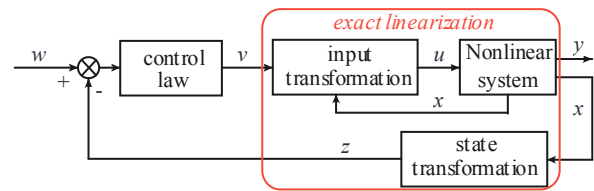


Fig. 16: Control structure for exact feedback linearization input/output method

The principle of the exact feedback linearization input/output method is based on repeatedly derivative of the output  $y$  of the nonlinear state space form (12) until a dependence on the input signal  $u$ . The number of derivation indicates a relative order of the system  $r$ . If relative order  $r$  equal the order of system  $n$ , then the transformation of system (12) is completed and is possible to define a state transformation  $z = z(x)$  in the shape

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ \vdots \\ y^{r-1} \end{bmatrix} \quad (21)$$

If the substitution  $y^r = v$ , the input transformation  $u = u(x,v)$  has the following shape

$$u = \frac{1}{\beta}(-\alpha + v) \quad (22)$$

where  $\alpha$ ,  $\beta$  are nonlinear function. The detailed description of the exact feedback linearization input/output method can be found in [12], [13].

The nonlinear state space form of the ML laboratory model (13) used for the applying of the exact feedback linearization input/output method was rewritten for better overview and has the following shape

$$\begin{cases} \dot{x}_1(t) \\ \dot{x}_2(t) \end{cases} = \begin{cases} 0 \\ 0 \end{cases} + \begin{cases} 0 \\ 0 \end{cases} \frac{x_2(t) u(t)^2}{(x_1(t) - x_0)^2} - g - B_{pom} x_2(t) \quad (23)$$

$$y(t) = C_{pom} x_1(t) + D_{pom}$$

$$\text{where } A_{pom} = \frac{k_i^2 k_{DA}^2 k_c}{m_k}, \quad B_{pom} = \frac{k_{fv}}{m_k},$$

$$C_{pom} = k_{AD} k_x, \quad D_{pom} = k_{AD} y_0.$$

The next step is derivative of the system output  $y$  until dependence on the input signal  $u$ , and then is possible to define the state and input transformation. The proposed *program module for exact feedback linearization input/output algorithm* was used for determine the required transformation in the Matlab/Simulink language [14]. After application the program module for ML model (23), the state and input transformation have the following shape

$$\begin{cases} z_1 \\ z_2 \end{cases} = \begin{cases} C_{pom} x_1 + D_{pom} \\ C_{pom} x_2 \end{cases} \quad (24)$$

$$u = \frac{(x_0 - x_1) *}{A_{pom} C_{pom}} * \frac{(A_{pom} C_{pom} (v + C_{pom} g + B_{pom} C_{pom} x_2))^{1/2}}{A_{pom} C_{pom}} \quad (25)$$

After application of the state transformation (24) and input transformation (25) is possible rewrite the nonlinear model (23) into the following linear form (20) in the shape

$$\begin{aligned} \dot{z} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \\ y &= [1 \quad 0] z \end{aligned} \quad (26)$$

Based on the state space form (26), which was transformed into the discrete form by the Matlab function *c2d()* with sampling period  $T_s = 0,002s$ , was designed the discrete control algorithm using the optimal state control with integral in the shape

$$v(k) = -K_{EL} z_{EL}(k) = -K[z(k) \quad z_i(k)] \quad (27)$$

If the weighting matrices  $Q_{EL} = [28 \ 0 \ 0; 0 \ 0,01 \ 0; 0 \ 0 \ 5500]$ ,  $R_{EL} = 5e-7$ , then the gain vector  $K_{EL}$  and feedback loop roots *root\_uro* have the following shape

$$K_{EL} = \begin{bmatrix} 0,8286e4 & 0,7733 \\ 0,0172e4 & 0,8944 \\ 8,5962e4 & 0,9713 \end{bmatrix}, \quad \text{root\_uro} = \begin{bmatrix} 0,8944 \\ 0,9713 \end{bmatrix} \quad (28)$$

The discrete Kalman estimator (19) was used for the estimate of the ML simulation and laboratory model states. The control structure for testing the proposed control using exact feedback linearization input/output method to control the ML simulation and laboratory model with purpose of to ensure the desired position of ball in the magnetic field of the coil is in the Fig. 17.

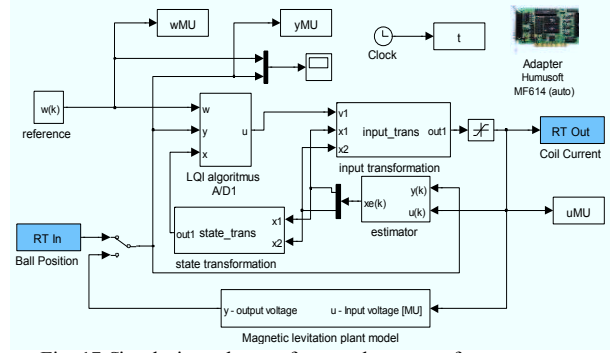


Fig. 17: Simulation scheme of control structure for exact feedback linearization input/output method for ML simulation and laboratory model

The resulting time responses of the tracking the reference trajectories with using the proposed nonlinear synthesis are in the Fig.18(a, b) and Fig. 19 (a, b).

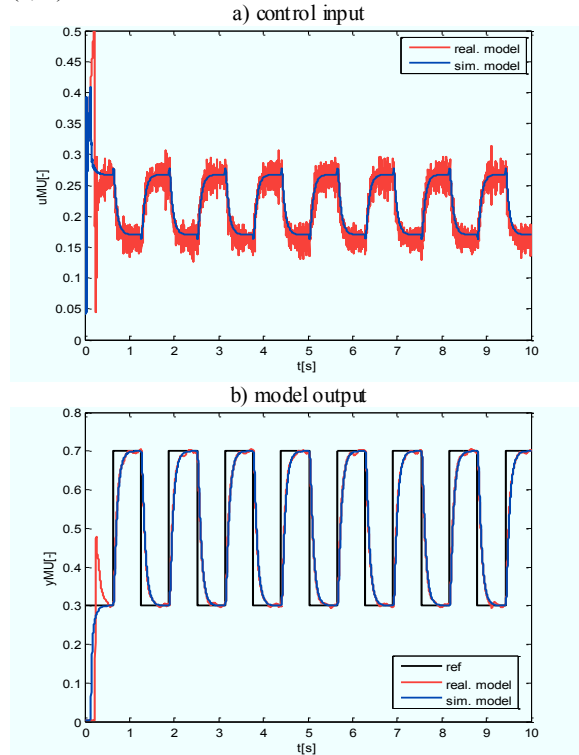
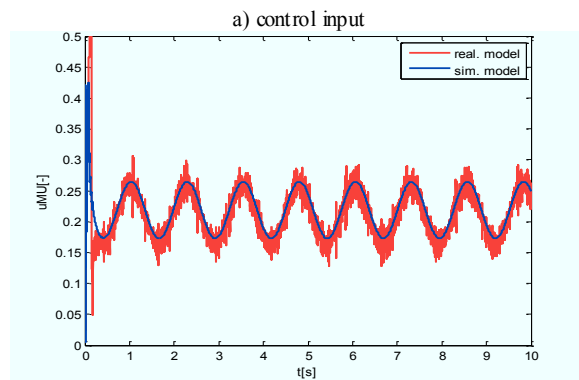


Fig. 18: Time responses of the ML simulation and laboratory model with using exact feedback linearization input/output method - square trajectory





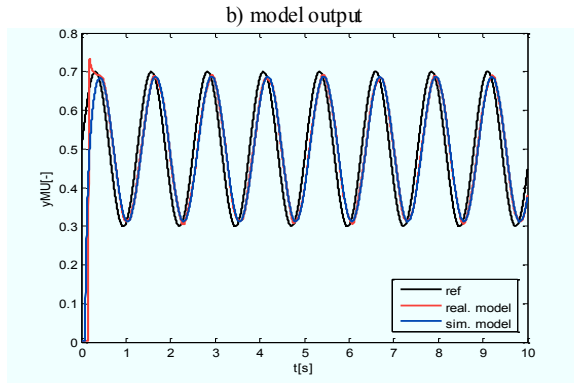


Fig. 19: Time responses of the ML simulation and laboratory model with using exact feedback linearization input/output method – sinus trajectory

## EVALUATION OF OBTAINED CONTROL RESULTS

For evaluation of results of the proposed control algorithms for control ML simulation and laboratory model was used the IAE criterion (Integral of Absolute value of Error), which is defined by the sum of errors in the shape

$$S = \sum_{k=1}^M |e(k)| \quad (29)$$

The resulting values of the chosen criterion of ML simulation and laboratory model for control algorithms (exact linearization method, optimal state control with integrator) are shown in the Tab. 3 and Tab. 4.

Tab. 3: IAE criterion – square trajectory

square trajectory	simulation ML	laboratory ML
Exact linearization	317,4156	325,1815
LQI	200,1723	212,9896

Tab. 4: IAE criterion – sinus trajectory

sinus trajectory	simulation ML	laboratory ML
exact linearization	322,5194	313,5943
LQI	207,0827	217,7103

From the resulting values from Tab. 3 and Tab. 4 and also from the resulting time responses (Fig. 14, Fig. 15, Fig. 18, Fig. 19) show, that the proposed optimal state control with integrator to ensure better tracking the reference trajectories with limits  $\langle 0,3;0,7 \rangle$  MU.

The time responses are on the Fig. 20, which were obtained of control of the ML laboratory model with using the proposed control algorithms (exact linearization method, optimal state control with integrator, PSD velocity algorithm) with purpose of

tracking the square trajectory reference with increased range  $\langle 0,08;0,88 \rangle$  MU.

Tab. 5: IAE criterion with increased range – square trajectory

laboratory ML	square trajectory
exact linearization	261,9644
LQI control	329,1399
PSD	333,4730

In the Tab. 5 are shown the resulting values of the chosen criterion for ML laboratory model, for control algorithms (exact linearization method, optimal state control with integrator, PSD velocity algorithm) and defined reference square trajectory with increased range.

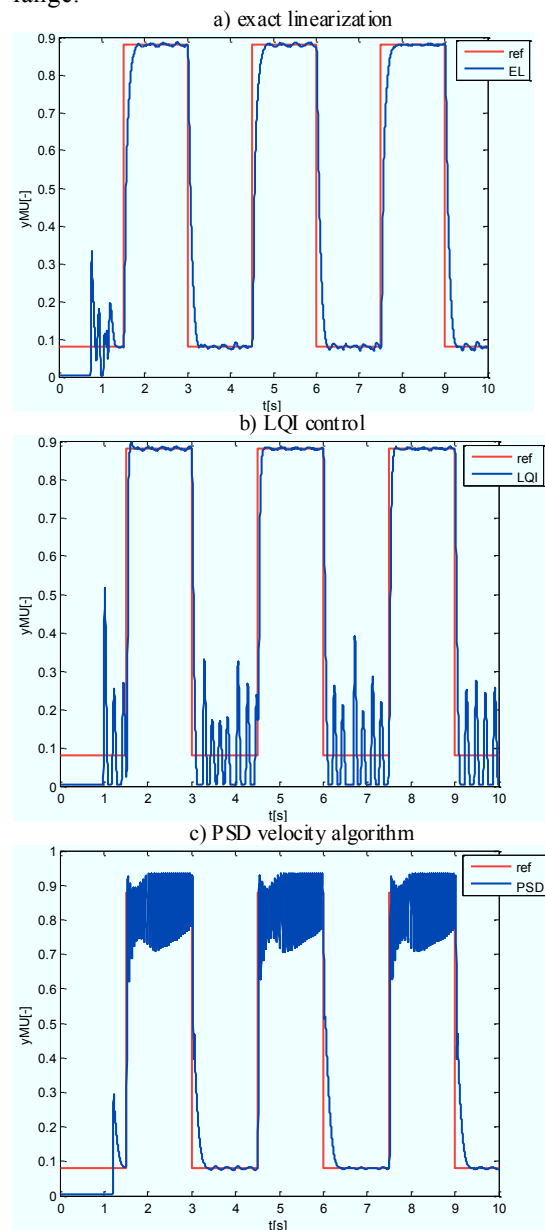


Fig. 20: Time responses of the ML laboratory model with using the proposed control algorithms - a) exact linearization, b) LQI control, c) PSD velocity algorithm.

From the resulting values from Tab. 5 and also from the resulting time responses (Fig. 20 a, b, c) show, that the proposed control algorithm with using exact feedback linearization input/output method to ensure tracking the defined reference trajectories in the both limits.

Therefore, it can be said, that the proposed control with using the exact feedback linearization input/output method has better results in the whole operating range of the laboratory model of Magnetic levitation.

The further research will focus on the control algorithm design using nonlinear methods, where will be used the obtained simulation model of the Magnetic levitation and also the obtained knowledge from identification by genetic algorithm will be used for modeling and identification of nonlinear model.

## 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

## REFERENCES

- [1] Humusoft Praha, CE 512 Education Manual Magnetic Levitation Model
- [2] Osmic, N. and Velagic, J. (2010). "Genetic Algorithm Based Identification of a Nonlinear 2DOF Helicopter Model," 2010 IEEE Mediterranean Conference on Control and Automation (MED2010), June 23-25, Marrakech, Morocco, pp. 333-338, ISBN: 978-4244-8091-3.
- [3] Pelikán, R. (2006). Magnetic levitation – modelling, simulation and control. M.Sc.thesis (in Czech), Tomas Bata University in Zlín, Czech Republic.
- [4] H.K. Kristinsson and G.A. Dumont, "System identification and control using genetic algorithms," IEEE Trans. on Systems, Man and Cybernetics, vol. 22, no. 5, pp. 1033–1046, Sep. 1992.
- [5] W. Chen, Q. Gong, T. Wang, C. Yin, and J. Yao, "A real-parameter genetic algorithm application in parameters identification for synchronous generator," in Proc. IEEE International Conference on ICIS, pp. 762 – 766.
- [6] W. Barie and J. Chiasson, "Linear and Nonlinear state-space controllers for magnetic levitation," International Journal of Systems Science, vol. 27, number 11, pp. 1153-1163.
- [7] M. Shafiq and S. Akhtar, "Inverse Model Based Adaptive Control of Magnetic Levitation System", 5th Asian Control Conference. 2004.
- [8] F. Zhang and K. Suyama, "Nonlinear Feedback Control of Magnetic Levitating System By Exact Linearization Approach," Tokyo University of Mercantile Marine, Japan, Proc. IEEE Conf. Contr. Appl., pp. 267-268 (1995).
- [9] Real time toolbox for use with SIMULINK, User's manual, Humusoft, Prague, 2007
- [10] I. Sekaj, and M. Foltin : Matlab toolbox - genetic algorithm. 11. conference MATLAB 2003. Praha, Czech republic: 25.11.2003, s. 514-519. (in Slovak)
- [11] FIKAR, Miroslav: "Dynamic optimalization process", STU Bratislava, 2010, ISBN 978-80-89316-08-3. (in Slovak)
- [12] M. Razím and J. Štecha, "Nonlinear system", ČVUT, 1997 (in Czech)
- [13] HUBA, M.: "Nonlinear system", STU Bratislava, 2003, ISBN 80-227-1908-0. (in Slovak)
- [14] ŠUSTER, Peter, JADLOVSKÁ, Anna: Modeling and control design of magnetic levitation system, In: SAMI 2012 : 10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics : Proceedings : Herľany, Slovakia, January 26-28, 2012. - Budapest : IEEE, 2012. - ISBN 978-1-4577-0195-5. - S. 295-299