Transactions of the VŠB – Technical University of Ostrava, Mechanical Series
No. 2, 2009, vol. LV,
article No. 1683

Leszek CEDRO*, Dariusz JANECKI**

IDENTIFICATION OF A MANIPULATOR MODEL USING THE INPUT ERROR METHOD IN THE MATHEMATICA PROGRAM

IDENTIFIKACE MODELU MANIPULATORU METODOU CHYBY VSTUPU POMOCÍ PROGRAMU MATHEMATICA

Abstract

The problem of parameter identification for a four-degree-of-freedom robot was solved using the Mathematica program. The identification was performed by means of specially developed differential filters [1]. Using the example of a manipulator, we analyze the capabilities of the Mathematica program that can be applied to solve problems related to the modeling, control, simulation and identification of a system [2]. The responses of the identification process for the variables and the values of the quality function are included.

Abstrakt

Použitím programu Mathematica byla řešena úloha identifikace parametrů pro robot se čtyřmi stupni volnosti. Identifikace byla provedena pomocí speciálně vyvinutých derivačních filtrů. Na příkladě manipulátoru byly analyzovány možnosti programu Mathematica při řešení úloh modelování, řízení, simulace a identifikace. Jsou uvedeny výsledky identifikačního procesu pro proměnné a také hodnoty kriteria kvality.

1 INTRODUCTION

Mathematica by Wolfram Research Inc. is a leading computation system suitable for a large variety of numeric, symbolic, or graphical input. It can be applied to solve both typically mathematical as well as technology-related problems [3].

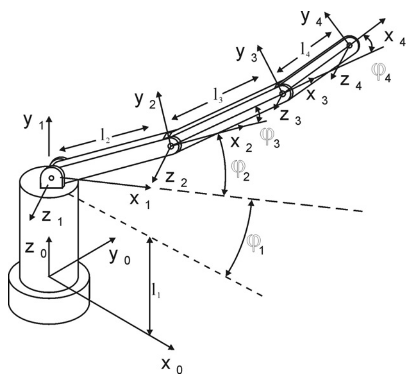

Fig. 1 Manipulator model

* DrSc., Kielce Technical University, Faculty of Mechatronics and Machine Design, Division of Computer Science and Robotics, tel/fax. +48/41/3424504, Al. Tysiąclecia PP 7, 25-314 Kielce, Poland email: lcedro@tu.kielce.pl
** Assoc. Prof., email: djanecki@tu.kielce.pl

It is a versatile environment appropriate for a wide range of computation procedures starting from basic arithmetic to the most advanced operations of further mathematics. The system can be used by students as well as research staff whenever an advanced mathematical tool is necessary [4].

The capabilities of the Mathematica program related to the modeling, control and identification of a robot system are discussed using the example of a manipulator. We consider a manipulator with four rotary joints, the structure of which is presented in Figure 1.

Let $\varphi = [\varphi_1\ \varphi_2\ \varphi_3\ \varphi_4]$ denote a vector of joint variables used as generalized coordinates, m_i - mass of the element i, and l_i - length of the element i. To simplify the calculations, we assumed a simple distribution of masses with mass concentration at the end of each element. The problems were analyzed in the following order. First, the kinetic and potential energies of the system were defined; then, using the second order Lagrange equations, it was possible to symbolically determine the equations of the robot dynamics. The identification process required developing a special function responsible for low-pass filtration and signal differentiation. A simulation was performed for a closed-loop robot system with PD controllers. The inverse model method was applied to identify the robot parameters basing on the measurement data. The results of the identification were represented graphically.

The work presented in notebook format includes the whole code necessary for the computations. Due to limited space, most of the results are hidden.

2 MODELLING

Let us determine the matrices for a rotary-type element and then the transformation matrices.

$$ob = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};\ r = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix};\ T1 = \begin{pmatrix} \text{Cos}[\varphi[1][t]] & 0 & \text{Sin}[\varphi[1][t]] & 0 \\ \text{Sin}[\varphi[1][t]] & 0 & -\text{Cos}[\varphi[1][t]] & 0 \\ 0 & 1 & 0 & l[1] \\ 0 & 0 & 0 & 1 \end{pmatrix};$$

$$T2 = \begin{pmatrix} \text{Cos}[\varphi[2][t]] & -\text{Sin}[\varphi[2][t]] & 0 & l[2]\ \text{Cos}[\varphi[2][t]] \\ \text{Sin}[\varphi[2][t]] & \text{Cos}[\varphi[2][t]] & 0 & l[2]\ \text{Sin}[\varphi[2][t]] \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};$$

$$T3 = \begin{pmatrix} \text{Cos}[\varphi[3][t]] & -\text{Sin}[\varphi[3][t]] & 0 & l[3]\ \text{Cos}[\varphi[3][t]] \\ \text{Sin}[\varphi[3][t]] & \text{Cos}[\varphi[3][t]] & 0 & l[3]\ \text{Sin}[\varphi[3][t]] \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};$$

$$T4 = \begin{pmatrix} \text{Cos}[\varphi[4][t]] & -\text{Sin}[\varphi[4][t]] & 0 & l[4]\ \text{Cos}[\varphi[4][t]] \\ \text{Sin}[\varphi[4][t]] & \text{Cos}[\varphi[4][t]] & 0 & l[4]\ \text{Sin}[\varphi[4][t]] \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};$$

The matrices required to determine the velocities of systems 1, 2, 3 and 4 are:

```
U1[1] = ob.T1;
U2[1] = ob.T1.T2; U2[2] = T1.ob.T2;
U3[1] = ob.T1.T2.T3; U3[2] = T1.ob.T2.T3; U3[3] = T1.T2.ob.T3;
U4[1] = ob.T1.T2.T3.T4; U4[2] =T1.ob.T2.T3.T4; U4[3] = T1.T2.ob.T3.T4; U4[4] =T1.T2.T3.ob.T4;
```

The generalized coordinates will be denoted by vector q.

```
q = {φ[1][t], φ[2][t], φ[3][t], φ[4][t]};
```

The kinetic energy of each element is

$$K[1] = \frac{1}{2} m[1] \sum_{i=1}^{3}\sum_{j=1}^{1} ((U1[j].r)\ (\text{Transpose}[r].\text{Transpose}[U1[j]])[[1]])[[i]][[1]]\ D[q[[j]], t]^2;$$

$$K[2] = \frac{1}{2} m[2] \sum_{i=1}^{3}\sum_{j=1}^{2} ((U2[j].r)\ (\text{Transpose}[r].\text{Transpose}[U2[j]])[[1]])[[i]][[1]]\ D[q[[j]], t]^2;$$

$$K[3] = \frac{1}{2} m[3] \sum_{i=1}^{3}\sum_{j=1}^{3} ((U3[j].r)\ (\text{Transpose}[r].\text{Transpose}[U3[j]])[[1]])[[i]][[1]]\ D[q[[j]], t]^2;$$

$$K[4] = \frac{1}{2} m[4] \sum_{i=1}^{3}\sum_{j=1}^{4} ((U4[j].r)\ (\text{Transpose}[r].\text{Transpose}[U4[j]])[[1]])[[i]][[1]]\ D[q[[j]], t]^2;$$

The inertia of each element is taken into account.

```
Table[Ko[k] = J[k] D[φ[k][t], t]^2 / 2, {k, 1, 4}]; Table[J[k] = m[k] l[k]^2 / 12, {k, 1, 4}];
```

The total kinetic energy of the system is

```
K = Sum[(K[k] + Ko[k]), {k, 1, 4}] // FullSimplify;
```

The potential energy for the particular elements is

```
V[1] = (-m[1] (0 0 g 0).T1.r)[[1]][[1]]; V[2] = (-m[2] (0 0 g 0).T1.T2.r)[[1]][[1]];
V[3] = (-m[3] (0 0 g 0).T1.T2.T3.r)[[1]][[1]]; V[4] = (-m[4] (0 0 g 0).T1.T2.T3.T4.r)[[1]][[1]];
```

The total potential energy is

```
V = Sum[V[k], {k, 1, 4}] // FullSimplify;
```

Using the expressions for kinetic and potential energy, we obtain the second order Lagrange equations

$$\frac{d}{dt}\frac{\partial K}{\partial \dot{\varphi}} - \frac{\partial K}{\partial \varphi} + \frac{\partial V}{\partial \varphi} = Mzr\,, \tag{1}$$

where Mzr is the generalized forces.

Thus, the Lagrange equations are equal.

```
Table[row[i] = (D[D[K, φ[i]'[t]], t] - D[K, φ[i][t]] + D[V, φ[i][t]] - Mzr[i][t] == 0), {i, 1, 4}];
```

3 DIFFERENTIAL FILTERS

This section, which is auxiliary in character, defines the function responsible for signal differentiation. In practice, signals contain noise resulting from measurement as well as that caused by the quantization of analogue signals. Ideal differentiation performed by means of the **D[...]** function was replaced with a function responsible for signal differentiation as well as low-pass filtration.

Let us assume that the differential filter of the *k*-th order is a series connection of a low-pass filter with boundary frequency Ω_g and a difference quotient of the *k*-th order. The low-pass filter will be responsible, firstly, for reducing the signal spectrum and, secondly, for correcting the characteristics of the difference quotient in the range of low frequencies. Thus, the filter will be called a low-pass correction filter. The desired transfer function of the low-pass filter is:

$$H_{\text{kor}}(\Omega) = \begin{cases} H_k(\Omega)/H_{\Delta k}(\Omega) & \text{for} \quad \Omega \le \Omega_g \\ 0 & \text{for} \quad \Omega > \Omega_g. \end{cases} \tag{2}$$

As a result, the transfer function of the series connection of the difference quotient and the low-pass filter in the range of low frequencies will be equal to the transfer function of an ideal differential filter.

The transfer function of the low-pass filter for $\Omega \le \Omega_g$ is equal to

$$H_{\text{kor}\,k}(n) = H_k(\Omega)/H_{\nabla k}(\Omega) = \begin{cases} \Omega/\sin\Omega & k = 1 \\ \Omega^2/2(1-\cos\Omega) & k = 2 \\ \Omega^3/(-2\sin\Omega + \sin(2\Omega)) & k = 3. \end{cases} \tag{3}$$

The filter signal response is an inverse Fourier transform of its frequency characteristics, thus:

$$h_{\mathrm{kor}\,k}(n) = \frac{1}{2\pi} \int_{-\Omega_g}^{\Omega_g} H_{\mathrm{kor}\,k}(n) e^{j\Omega n} d\Omega \,. \tag{4}$$

The proposed function, Filtr, is responsible for the low-pass filtration, while functions Filtr1 and Filtr2 are used for the low-pass filtration and differentiation of the signal. One derivative ($k = 1$) is obtained by using function Filtr1, and the other ($k = 2$) by means of function Filtr2.

```
Filtr[Mm_, Omega_, delta_, t_, x_] := Module[{}, Clear[M, Ωg, Δ];
  H0[Ω_] = 1;
  h0[n_ /; n ≠ 0] = FullSimplify[1/(2 π) ∫_{-Ωg}^{Ωg} H0[Ω] e^(I Ω n) dΩ]; h0[n_ /; n == 0] = FullSimplify[1/(2 π) ∫_{-Ωg}^{Ωg} H0[Ω] 1 dΩ];
  M = Mm; Ωg = Omega; Δ = delta; wHarris[n_] = 0.36 + 0.49 Cos[n π / M] + 0.14 Cos[2 n π / M] + 0.01 Cos[3 n π / M];
  H0Harris[Ω_] = Σ_{n=-M}^{M} h0[n] wHarris[n] e^(-I Ω n); κH1 = Σ_{n=-M}^{M} h0[n] wHarris[n]; h0p = Table[0, {2 M + 1}];
  For[i = -M, i ≤ M, i++, h0p[[i + M + 1]] = h0[i] wHarris[i] / κH1]; NN = Length[t];
  xf = Table[0, {NN - 2 M - 1}]; For[i = M + 1, i ≤ NN - M - 1, i++, xf[[i - M]] = x[[Range[i - M, i + M]]] . h0p;];
  xf]

Filtr1[Mm_, Omega_, delta_, t_, x_] := Module[{}, Clear[M, Ωg, Δ];
  H1[Ω_] = s /. s → I Ω / Δ; h1[n_ /; n ≠ 0] = FullSimplify[1/(2 π) ∫_{-Ωg}^{Ωg} H1[Ω] e^(I Ω n) dΩ];
  h1[n_ /; n == 0] = FullSimplify[1/(2 π) ∫_{-Ωg}^{Ωg} H1[Ω] 1 dΩ]; M = Mm; Ωg = Omega; Δ = delta;
  H1Harris[Ω_] = Σ_{n=-M}^{M} h1[n] wHarris[n] e^(-I Ω n); κH2 = Σ_{n=-M}^{M} h1[n] wHarris[n] (0 - n Δ); h1p = Table[0, {2 M + 1}];
  For[i = -M, i ≤ M, i++, h1p[[i + M + 1]] = h1[i] wHarris[i] (-1 / κH2)]; NN = Length[t];
  xfs = Table[0, {NN - 2 M - 1}]; For[i = M + 1, i ≤ NN - M - 1, i++, xfs[[i - M]] = x[[Range[i - M, i + M]]] . h1p;];
  xfs]

Filtr2[Mm_, Omega_, delta_, t_, x_] := Module[{}, Clear[M, Ωg, Δ];
  H2[Ω_] = s^2 /. s → I Ω / Δ; h2[n_ /; n ≠ 0] = FullSimplify[1/(2 π) ∫_{-Ωg}^{Ωg} H2[Ω] e^(I Ω n) dΩ];
  h2[n_ /; n == 0] = FullSimplify[1/(2 π) ∫_{-Ωg}^{Ωg} H2[Ω] 1 dΩ]; H2Harris[Ω_] = Σ_{n=-M}^{M} h2[n] wHarris[n] e^(-I Ω n);
  M = Mm; Ωg = Omega; Δ = delta; κH3 = Σ_{n=-M}^{M} h2[n] wHarris[n]; h0FH[n_] = 1/κH1 h0[n] wHarris[n];
  κH4 = Σ_{n=-M}^{M} (h2[n] wHarris[n] - κH3 h0FH[n]) (0 - (Δ n)^2); h2FH[n_] = -2/κH4 (h2[n] wHarris[n] - κH3 h0FH[n]);
  h2p = Table[0, {2 M + 1}]; For[i = -M, i ≤ M, i++, h2p[[i + M + 1]] = -2/κH4 (h2[i] wHarris[i] - κH3 h0FH[i])];
  NN = Length[t]; xfss = Table[0, {NN - 2 M - 1}];
  For[i = M + 1, i ≤ NN - M - 1, i++, xfss[[i - M]] = x[[Range[i - M, i + M]]] . h2p;];
  xfss]
```

4 SIMULATION

The simulation was conducted for a closed-loop robot system with PD controllers [5]. The results will be used as the data for the identification algorithm. First, the pre-determined signal is defined. It is assumed that the signal is an appropriately delayed step function (with different delay times for each arm). The function was additionally filtered using a first-order low-pass filter with a boundary frequency $\Omega_g = 0.025$ [rad].

```
Δ = 0.001; tk = 10;
syg[1] = Table[{t, UnitStep[t - 1] / 2}, {t, 0., tk, Δ}];
syg[2] = Table[{t, UnitStep[t - 3] / 2}, {t, 0., tk, Δ}];
syg[3] = Table[{t, UnitStep[t - 5] / 2}, {t, 0., tk, Δ}];
syg[4] = Table[{t, UnitStep[t - 7] / 2}, {t, 0., tk, Δ}];
tp = Table[Δ i, {i, 0, tk / Δ}];
Table[Mp[i] = Transpose[syg[i]][[2]], {i, 1, 4}];
Mf[1] = Filtr[100, 0.025, 0.001, tp, Mp[1]]; Mf[2] = Filtr[100, 0.025, 0.001, tp, Mp[2]];
Mf[3] = Filtr[100, 0.025, 0.001, tp, Mp[3]]; Mf[4] = Filtr[100, 0.025, 0.001, tp, Mp[4]];
tp1 = Δ Range[0, Length[Mf[1]] - 1];
Sygzad = {M[1] → Interpolation[Transpose[{tp1, Mf[1]}]], M[2] → Interpolation[Transpose[{tp1, Mf[2]}]],
   M[3] → Interpolation[Transpose[{tp1, Mf[3]}]], M[4] → Interpolation[Transpose[{tp1, Mf[4]}]]};
tk = tk - 0.3; Ωg =.;
```

The PD controller used for the object control is

```
Table[Mzr[i][t] = Kp[i] (M[i][t] - φ[i][t]) - Kd[i] D[φ[i][t], t], {i, 1, 4}];
```

The following parameters were assumed for the system:

```
Parametry = {l[1] → 0.4, l[2] → 0.8, l[3] → 0.6, l[4] → 0.2, g → 9.81, m[1] → 50, m[2] → 25, m[3] → 15,
   m[4] → 5, Kp[1] → 3800, Kd[1] → 159.5, Kp[2] → 40000, Kd[2] → 120, Kp[3] → 10000, Kd[3] → 50,
   Kp[4] → 500, Kd[4] → 2};
```

And the zero initial values were:

```
w1 = φ[1]'[0] == 0; w2 = φ[1][0] == 0; w3 = φ[2]'[0] == 0; w4 = φ[2][0] == 0; w5 = φ[3]'[0] == 0;
w6 = φ[3][0] == 0; w7 = φ[4]'[0] == 0; w8 = φ[4][0] == 0;
```

The equations were solved numerically:

```
roz = NDSolve[{row[1], row[2], row[3], row[4], w1, w2, w3, w4, w5, w6, w7, w8} /. Parametry /. Sygzad,
   {φ[1], φ[2], φ[3], φ[4]}, {t, 0, tk}, MaxSteps → 10000];
Table[φr[i] = φ[i] /. roz[[1]], {i, 1, 4}];
```

Figure 2 shows the responses of variables in a closed-loop system. The responses are not satisfactory from the point of view of regulation [6]. The aim of the study was to generate signals to be further applied in the identification process. In this case, it is recommended to select the appropriate predetermined signals and controller parameters so that the signals can provide us with sufficient information on the object properties.

```
Plot[Evaluate[{φr[1][t], φr[2][t], φr[3][t], φr[4][t]}], {t, 0, tk},
 PlotStyle -> {RGBColor[1, 0, 0], RGBColor[0, 1, 0], RGBColor[0, 0, 1], RGBColor[1, 1, 0], RGBColor[0, 1, 1],
   RGBColor[1, 0, 1]}, AxesLabel → {"t[s]", "φ[rad]"}]
```

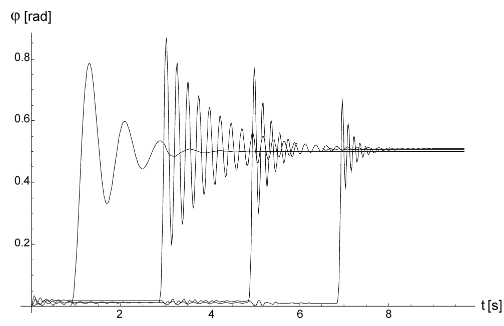

Fig. 2 Responses of variables $\varphi_1, \varphi_2, \varphi_3, \varphi_4$

5 IDENTIFICATION

The method used for identifying the robot system parameters $\theta = [m_1, m_2, m_3, m_4, l_1, l_2, l_3, l_4]$ is shown as a schematic diagram in Figure 3. It is assumed that the measurement data concerning the trajectories of the generalized variables φ and the necessary input signals Mzr are available. Using the current estimates of the object parameters, $\hat{\theta} = [\hat{m}_1, \hat{m}_2, \hat{m}_3, \hat{m}_4, \hat{l}_1, \hat{l}_2, \hat{l}_3, \hat{l}_4]$, it is possible to determine the estimate of the input signals, $Mz\hat{r}^f$.

The equations have the same structure as Eq. (1); however, the unknown parameters θ are replaced by the estimates $\hat{\theta}$, and the generalized variables φ and their derivatives $\dot{\varphi}$, $\ddot{\varphi}$, which are not measured, are replaced by the estimates of these variables, φ^f , $\dot{\varphi}^f$, $\ddot{\varphi}^f$ obtained by applying appropriate differential filters. Thus, the identification process involves determining the estimates of the parameters minimizing the quality factor.

$$J(\hat{\theta}) = \frac{1}{T}\int_0^T (Mz\hat{r}^f - Mzr^f)^2\, dt\,, \tag{5}$$

where Mzr^f is the input signal after filtering. An advantage of this method is that the identification procedure is considerably faster. It is not necessary to solve a series of differential equations at each iteration of the algorithm minimizing the quality factor. In contrast, the conventional output error method involves comparing the object input signal φ with its estimate $\hat{\varphi}$.

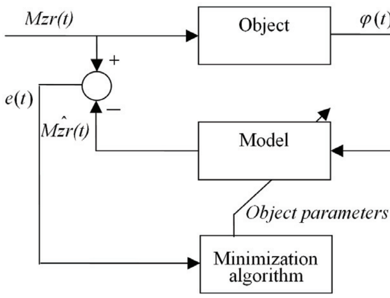

Fig. 3 Schematic diagram of the identification process

It is essential that the spectrum of the pre-determined signal be limited considerably. Despite the fact that the robot system is strongly non-linear for the filtered signals, we have:

$$Mz\hat{r}^f \cong Mzr^f \text{ for } \hat{\theta} = \theta\,. \tag{6}$$

Let us determine the values of the particular signals and their derivatives to be used in the identification process.

To reduce the number of calculations, we employ a discrete equivalent of the quality factor (5). We use signals ranging from 0 to *tk* that are sampled at different moments of time, at sampling intervals Δ .

```
tp =. ; tp = Table[Δ i, {i, 0, tk / Δ}] ;
Table[φp[i] = φr[i][t] /. t → tp, {i, 1, 4}] ;
Table[Mp[i] = M[i][t] /. Sygzad /. t → tp, {i, 1, 4}] ;
φf[1] = Filtr[100, 0.2, 0.001, tp, φp[1]] ; φf[3] = Filtr[100, 0.2, 0.001, tp, φp[3]] ;
φf[2] = Filtr[100, 0.2, 0.001, tp, φp[2]] ; φf[4] = Filtr[100, 0.2, 0.001, tp, φp[4]] ;
Mf[1] = Filtr[100, 0.2, 0.001, tp, Mp[1]] ; Mf[3] = Filtr[100, 0.2, 0.001, tp, Mp[3]] ;
Mf[2] = Filtr[100, 0.2, 0.001, tp, Mp[2]] ; Mf[4] = Filtr[100, 0.2, 0.001, tp, Mp[4]] ;
φfs[1] = Filtr1[100, 0.2, 0.001, tp, φp[1]] ; φfs[3] = Filtr1[100, 0.2, 0.001, tp, φp[3]] ;
φfs[2] = Filtr1[100, 0.2, 0.001, tp, φp[2]] ; φfs[4] = Filtr1[100, 0.2, 0.001, tp, φp[4]] ;
φfss[1] = Filtr2[100, 0.2, 0.001, tp, φp[1]] ; φfss[3] = Filtr2[100, 0.2, 0.001, tp, φp[3]] ;
φfss[2] = Filtr2[100, 0.2, 0.001, tp, φp[2]] ; φfss[4] = Filtr2[100, 0.2, 0.001, tp, φp[4]] ;
```

We replace the variables in the model equations with a list of values at specific moments of time.

```
Zamiana = {m[1] → m1, m[2] → m2, m[3] → m3, m[4] → m4, l[1] → l1, l[2] → l2,
   l[3] → l3, l[4] → l4};
Para1 = {φ[1]''[t] → φfss[1], φ[2]''[t] → φfss[2], φ[3]''[t] → φfss[3],
   φ[4]''[t] → φfss[4], φ[1]'[t] → φfs[1],
   φ[2]'[t] → φfs[2], φ[3]'[t] → φfs[3], φ[4]'[t] → φfs[4], φ[1][t] → φf[1],
   φ[2][t] → φf[2], φ[3][t] → φf[3],
   φ[4][t] → φf[4], M[1][t] -> Mf[1], M[2][t] -> Mf[2], M[3][t] -> Mf[3],
   M[4][t] -> Mf[4]};
row1 = row[1] /. Zamiana /. Parametry; row2 = row[2] /. Zamiana /. Parametry;
row3 = row[3] /. Zamiana /. Parametry; row4 = row[4] /. Zamiana /. Parametry;
```

To remove the responses of the identified parameters and the quality factors, we define the lists. Variable j denotes the number of iterations.

```
ile = 2000; j = 0; Blad = Table[0, {ile}];
m1iter = Table[0, {ile}]; m2iter = Table[0, {ile}]; m3iter = Table[0, {ile}];
m4iter = Table[0, {ile}];

l1iter = Table[0, {ile}]; l2iter = Table[0, {ile}]; l3iter = Table[0, {ile}];
l4iter = Table[0, {ile}];
```

Then, we define a function determining the quality factor.

```
Clear[m1, m2, m3, m4, l1, l2, l3, l4];
Jd[m1_, m2_, m3_, m4_, l1_, l2_, l3_, l4_] :=
  Module[{}, j = j + 1; m1iter[[j]] = m1; m2iter[[j]] = m2; m3iter[[j]] = m3;
   m4iter[[j]] = m4; l1iter[[j]] = l1; l2iter[[j]] = l2; l3iter[[j]] = l3;
   l4iter[[j]] = l4;
   Index = (Plus @@ (((row1[[1]] - row1[[2]]) /. Para1) ^2) / (Length[tp] - 2 M) +
      Plus @@ (((row2[[1]] - row2[[2]]) /. Para1) ^2) / (Length[tp] - 2 M) +
      Plus @@ (((row3[[1]] - row3[[2]]) /. Para1) ^2) / (Length[tp] - 2 M) +
      Plus @@ (((row4[[1]] - row4[[2]]) /. Para1) ^2) / (Length[tp] - 2 M));
   Blad[[j]] = Index; Index];
```

Let us assume that the initial values of the parameters to be identified are:

```
mo1 = 50.1; mo2 = 25.1; mo3 = 15.1; mo4 = 5.1; lo1 = 0.41; lo2 = 0.79; lo3 = 0.59;
lo4 = 0.19; ϵ = 0.001;
```

Subsequently, we minimize the quality factor, Jd.

```
FindMinimum[Jd[m1, m2, m3, m4, l1, l2, l3, l4], {m1, mo1, mo1 + ϵ}, {m2, mo2, mo2 + ϵ},
 {m3, mo3, mo3 + ϵ}, {m4, mo4, mo4 + ϵ}, {l1, lo1, lo1 + ϵ}, {l2, lo2, lo2 + ϵ},
 {l3, lo3, lo3 + ϵ}, {l4, lo4, lo4 + ϵ}]
{0.0223076 , {m1 → 49.5818 , m2 → 25.0003 , m3 → 15.0549 ,
  m4 → 4.96068 , l1 → 0.407419 , l2 → 0.799701 , l3 → 0.599708 , l4 → 0.200892 }}
```

The process of identification of the particular parameters is illustrated graphically in the diagrams below [7]. As can be seen, the parameter trajectories are determined after approximately 1450 iterations of the minimization algorithm. The estimates slightly depart from the values of the real parameters. The negligible differences are attributable to the system non-linearity; thus, Eq. (6) can be solved only approximately.

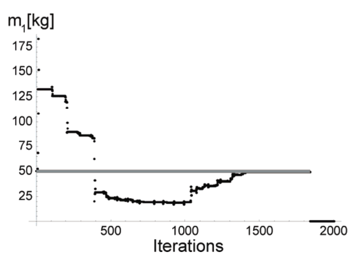

Fig. 4 Plot of the estimate of the parameter m_1

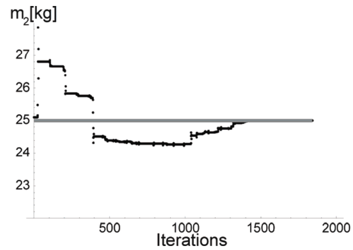

Fig. 5 Plot of the estimate of the parameter m_2

6 CONCLUSIONS

The inverse method applied for the identification of non-linear systems requires using a low-pass differential filter to limit the spectrum of the excitation signal. Spectrum limitation is an extremely significant It is not necessary to solve differential equations; it is determining appropriate derivatives only. The computation procedure may also be used for the identification of more complex problems. The data collected during the study shows that the inverse method is suitable for identifying more complex mechanical systems with more parameters to define.

REFERENCES

[1] JANECKI, D., CEDRO, L. *Differential Filters With Application To System Identification*, 7th European Conference of Young Research and Science Workers in Transport and Telecommunications TRANSCOM 2007, Žilina, Slovakia, p. 115, 2007.

[2] SODERSTROM, T., STOICA, P. *System Identification*, PHI (UK), 1994.

[3] FARANA, R. The Total Quality Management System Application to the Faculty Management System Improvement, *International Conference on Engineering Education,* Pécs, Hungary, 2008. 9. pp. ISSN 1562-3580.

[4] SMUTNÝ, L., FARANA, R., & SMUTNÝ, P. Real and Virtual Lab Stands on Web Support Experimental Education. *WSEAS Transactions on Advances in Engineering Education*. July 2005, Issue 3, Volume 2, pp. 243-250. ISSN 1790-1979.

[5] VÍTEČKOVÁ, M., VÍTEČEK, A. & KOČÍ, P. Seřízení regulátorů PI a PID pro integrační regulované soustavy. *Acta Mechanica Slovaca.* No1-A/2008. Ročník 12. Strojnická fakulta TU v Košicích. Košice. Slovenská republika, pp. 33-40. ISSN 1335-2393.

[6] VÍTEČKOVÁ, M. Simple Controller tuning and Non-oscillatory Plant Identification. *Journal of Cybernetics and Informatics*. Volume 7, 2008, pp. 26-33. ISSN 1336-4774.

[7] VÍTEČEK, A., VÍTEČKOVÁ, M. Nová metoda identifikace aperiodických soustav. In *Proceedings of 7th International Scientific – Technical Conference – PROCESS CONTROL 2006*. Pardubice : University of Pardubice, Kouty nad Desnou, Czech Republic, 13. – 16. 6. 2006, pp. R173-1 – R173-6. ISBN 80-7194-860-8.