

POSSIBILITIES OF GRAPHICAL ENVIRONMENT APPLICATION FOR DETERMINING PARAMETERS OF THE MECHANICAL SYSTEM

Marek Jaśkiewicz

*Kielce University of Technology
Department of Vehicles and Transport
1000-lecia Państwa Polskiego Av. 7, 25-314 Kielce, Poland
tel.: +48 41 3424285, fax: +48 41 3424726
e-mail: m.jaskiewicz@tu.kielce.pl*

Krzysztof Ludwinek, Jan Staszak

*Kielce University of Technology
Department of Electrical Machines and Mechatronic Systems
1000-lecia Państwa Polskiego Av. 7, 25-314 Kielce, Poland
tel.: +48 41 3424628, fax: +48 41 3424207
e-mail: k.ludwinek@tu.kielce.pl, j.staszak@tu.kielce.pl*

Tomasz Suliga

*Kielce University of Technology
Student of Electrical Engineering, Automatics and Computer Science
1000-lecia Państwa Polskiego Av. 7, 25-314 Kielce, Poland
tel.: +48 41 3424628, fax: +48 41 3424207
e-mail: tomasz.sulig@interia.eu*

Abstract

The article discusses the possibility of using LabView graphical environment for visualization and quick and easy identification of parameters of the dynamic system. For further analysis a human driver model, sitting in a car seat, fastened with seatbelts is used.

This model consists of eight rigid solids. The solids reflect parts of the human body. In the model head and neck are separate segments. The torso is divided into three parts: upper torso, middle torso and lower torso. In this model also a leg is a separate segment. The leg is divided into three parts: thigh, lower leg and foot. The individual solids are connected by torsionally vulnerable joints. Additionally a seat back was modelled too, which limits the movement of three segments of the torso. Similarly as in the standard car with the seat equipped with a three-point safety belt, in the model presented in this article, the location of the upper and lower seat belt was modelled and shown. The whole model has two frames of reference.

For this purpose, a graphical language program was developed, in which by changing the settings of analogue units representing parameters such as mass ratios, stiffness, damping, moments of inertia, it is possible to determine the impact of these parameters on the delay shapes, speed, energy, etc.

Keywords: *mechanical system parameters, the coefficient of stiffness, damping, graphical environment, LabView*

1. Introduction

The analysis of static and dynamic conditions of mechanical and electrical systems shall be based on a mathematical model [2, 5, 6]. Dynamic model equations are derived based on Lagrange equations of the second kind. Electromechanical parameters in the model are determined based on detailed knowledge of construction and materials (which are not always

known) or based on experimental studies on the basis of approximation or time-frequency characteristics obtained from measurements. The coefficients of the mathematical model (masses, moments of inertia, stiffness, damping, etc.) should be selected in such a way that the waveforms received on the basis of the model accurately reflect the waveforms obtained from the measurements [1-3, 7].

This article presents the possibility of using LabView graphical environment for quick and easy identification of mechanical parameters of the mechanical system by 10 degrees of freedom (a driver sitting in a car seat, fastened with seatbelts).

2. Dynamic model

The following assumptions have been taken into account while building a mathematical model of a dynamic man-driver, sitting in a car seat, fastened in a seat belt for computer crash test simulation:

- a flat model (movement in an XOZ car plane, further referred to as X_0OY_0 plane),
- model elements (particular solids) are treated as perfectly stiff,
- the arrangement of solids corresponds to particular parts of the body,
- the location of weight centres and moments of inertia of particular solids are known,
- solids are linked by articulated joints prone to torsion of a line stiffness and damping,
- no clearances occur among particular solids,
- the only input influencing the arrangement is V_x initial speed,
- chest bend has not been taken into consideration,
- belts are modelled as susceptible elements of Kelvin-Voigt,
- gas air bag has been modelled as a susceptible line element of Kelvin-Voigt.

The structure adopted for testing the dynamic model of the driver of 10 degrees of freedom is shown in Fig. 1. The model has 10 degrees of freedom: with 8 degrees are associated with the angular displacements $\alpha_1 - \alpha_8$ of individual blocks relative to each other, and 2 degrees are associated with the movement of the entire system in the x and y plane.

To conduct the analysis, 2 rectangular Cartesian systems of coordinates were applied, where the first one is a supplementary one:

- $X_0O_0Y_0$ – it is a motionless system of coordinates placed inside a car. X_0Y_0 plane sets a movement of an O point on a specified track, the direction of X_0 axis is parallel to a lengthwise axis of a road, whereas the direction of Y_0 axis a perpendicular one. The O point is fixed at the point of intersection of lower seating plane with support seating axis. The location of O_0 is constant and fixed to car mass.
- XOY – it is an auxiliary system of coordinates moved within X_0Y_0 of x_0 and y_0 length. Point O was established as the beginning of the system, located at the link point of trunk and thigh mass. Such system outlines rotation of particular solids over α angle against X axis. During a collision XOY system moves by x_0 and y_0 value against $X_0O_0Y_0$ system.

Motion model is described by two coordinate systems, immovable $X_0O_0Y_0$ and movable XOY which has the ability to move in the direction of the X_0 or Y_0 axis. In this model, $m_1 - m_8$ masses have the ability to move relative to each other at an angle α . This model consists of 8 blocks connected by joints. At the same time, it was assumed that the joints are vulnerable elements of linear characteristics. Starting from the top, the first block is the head, the second one is the neck and the trunk has been divided into three parts: the top one which symbolises the chest, the middle one which symbolises the belly and the lower one which is the abdomen and the hips. The leg has been divided into three parts: the thigh, lower leg and the foot. The block, that reflects the upper torso, in the assumptions, adopted more weight since the weight of hands has been added. Furthermore, it has been assumed that the only enforcement affecting the system is the V_x initial velocity.

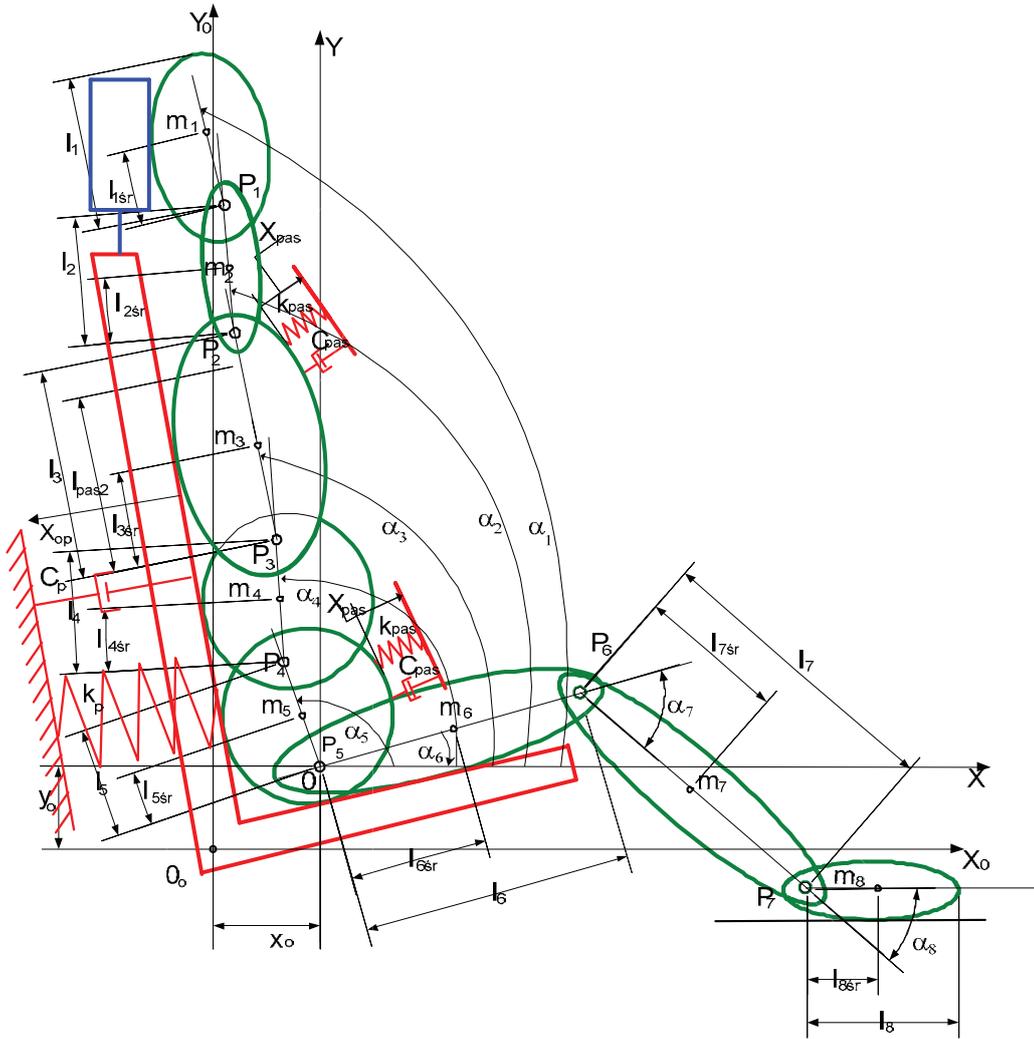


Fig. 1. The structure adopted for testing the driver model of 10 degrees of freedom

Structure of the dynamic model shown in Fig. 1 is represented by the kinematic dependencies that depend on the position of the x and y axis and the $\alpha_1 - \alpha_8$ location angle. Kinematics compounds against Y axis are described by the following relations:

$$\begin{aligned}
 y_1 &= y_{l0} + l_5 \sin(\alpha_5 + \alpha_{5p}) + l_4 \sin(\alpha_4 + \alpha_{4p}) + l_3 \sin(\alpha_3 + \alpha_{3p}) + l_2 \sin(\alpha_2 + \alpha_{2p}) + l_{1sr} \sin(\alpha_1 + \alpha_{1p}), \\
 y_2 &= y_{l0} + l_5 \sin(\alpha_5 + \alpha_{5p}) + l_4 \sin(\alpha_4 + \alpha_{4p}) + l_3 \sin(\alpha_3 + \alpha_{3p}) + l_{2sr} \sin(\alpha_2 + \alpha_{2p}), \\
 y_3 &= y_{l0} + l_5 \sin(\alpha_5 + \alpha_{5p}) + l_4 \sin(\alpha_4 + \alpha_{4p}) + l_{3sr} \sin(\alpha_3 + \alpha_{3p}), \\
 y_4 &= y_{l0} + l_5 \sin(\alpha_5 + \alpha_{5p}) + l_{4sr} \sin(\alpha_4 + \alpha_{4p}), \\
 y_5 &= y_{l0} + l_{5sr} \sin(\alpha_5 + \alpha_{5p}), \\
 y_6 &= y_{l0} + l_{6sr} \sin(\alpha_6 + \alpha_{6p}), \\
 y_7 &= y_{l0} + l_6 \sin(\alpha_6 + \alpha_{6p}) + l_{7sr} \sin(\alpha_7 + \alpha_{7p}), \\
 y_8 &= y_{l0} + l_6 \sin(\alpha_6 + \alpha_{6p}) + l_7 \sin(\alpha_7 + \alpha_{7p}) + l_{8sr} \sin(\alpha_8 + \alpha_{8p}),
 \end{aligned}
 \tag{1}$$

where:

- $y_1 - y_8$ - displacements of segments in the y direction,
- $l_1 - l_8$ - segments' length,
- $l_{1sr} - l_{8sr}$ - location of mass centres,
- $\alpha_1 - \alpha_8$ - values set of angles,
- $\alpha_{1p} - \alpha_{8p}$ - values of initial set of angles.

Kinematics compounds against Y axis are described by the following relations:

$$\begin{aligned}
 x_1 &= x_{l_0} + l_5 \cos(a_5 + a_{5p}) + l_4 \cos(a_4 + a_{4p}) + l_3 \cos(a_3 + a_{3p}) + l_2 \cos(a_2 + a_{2p}) + l_{1sr} \cos(a_1 + a_{1p}), \\
 x_2 &= x_{l_0} + l_5 \cos(a_5 + a_{5p}) + l_4 \cos(a_4 + a_{4p}) + l_3 \cos(a_3 + a_{3p}) + l_{2sr} \cos(a_2 + a_{2p}), \\
 x_3 &= x_{l_0} + l_5 \cos(a_5 + a_{5p}) + l_4 \cos(a_4 + a_{4p}) + l_{3sr} \cos(a_3 + a_{3p}), \\
 x_4 &= x_{l_0} + l_5 \cos(a_5 + a_{5p}) + l_{4sr} \cos(a_4 + a_{4p}), \\
 x_5 &= x_{l_0} + l_{5sr} \cos(a_5 + a_{5p}), \\
 x_6 &= x_{l_0} + l_{6sr} \cos(a_6 + a_{6p}), \\
 x_7 &= x_{l_0} + l_6 \cos(a_6 + a_{6p}) + l_{7sr} \cos(a_7 + a_{7p}), \\
 x_8 &= x_{l_0} + l_6 \cos(a_6 + a_{6p}) + l_7 \cos(a_7 + a_{7p}) + l_{8sr} \cos(a_8 + a_{8p}),
 \end{aligned} \tag{2}$$

where:

$x_1 - x_8$ - displacement of segments in the x direction.

Potential energy E_p of the system is described as follows:

$$\begin{aligned}
 E_p &= \frac{k_{p1}(a_1 - a_2)^2}{2} + \frac{k_{p2}(a_2 - a_3)^2}{2} + \frac{k_{p3}(a_3 - a_4)^2}{2} + \frac{k_{p4}(a_4 - a_5)^2}{2} + \frac{k_{p5}(a_6 - a_5)^2}{2} + \\
 &\quad \frac{k_{p6}(a_7 - a_6)^2}{2} + \frac{k_{p7}(a_8 - a_7)^2}{2} + \frac{k_{pas1}x_3^2}{2} + \frac{k_{pas2}x_5^2}{2} + \frac{k_{op}x_3^2}{2} + \frac{k_{op}x_4^2}{2} + \frac{k_{op}x_5^2}{2},
 \end{aligned} \tag{3}$$

where:

$k_{pas1}, k_{pas2}, k_{op}$ - stiffness coefficients of seat belt and backrest,

$k_{p1} - k_{p8}$ - stiffness coefficients of segments joints.

Rayleigh equation is described by the following relations:

$$\begin{aligned}
 R &= \frac{c_{p1} \begin{pmatrix} \bullet & \bullet \\ a_1 - a_2 \end{pmatrix}^2}{2} + \frac{c_{p2} \begin{pmatrix} \bullet & \bullet \\ a_2 - a_3 \end{pmatrix}^2}{2} + \frac{c_{p3} \begin{pmatrix} \bullet & \bullet \\ a_3 - a_4 \end{pmatrix}^2}{2} + \frac{c_{p4} \begin{pmatrix} \bullet & \bullet \\ a_4 - a_5 \end{pmatrix}^2}{2} + \frac{c_{p5} \begin{pmatrix} \bullet & \bullet \\ a_6 - a_5 \end{pmatrix}^2}{2} \\
 &+ \frac{c_{p6} \begin{pmatrix} \bullet & \bullet \\ a_7 - a_6 \end{pmatrix}^2}{2} + \frac{c_{p7} \begin{pmatrix} \bullet & \bullet \\ a_8 - a_7 \end{pmatrix}^2}{2} + \frac{c_{pas1} x_3^2}{2} + \frac{c_{pas2} x_5^2}{2} + \frac{c_{op} x_3^2}{2} + \frac{c_{op} x_4^2}{2} + \frac{c_{op} x_5^2}{2},
 \end{aligned} \tag{4}$$

where:

c_{pas}, c_{op} - damping coefficients of seat belt and backrest,

$c_1 - c_8, c_{p1} - c_{p8}$ - damping coefficients of segments joints.

Kinetic energy E_k of the system is described as follows:

$$\begin{aligned}
 E_k &= \frac{m_1 x_1^2}{2} + \frac{m_1 y_1^2}{2} + \frac{m_2 x_2^2}{2} + \frac{m_2 y_2^2}{2} + \frac{m_3 x_3^2}{2} + \frac{m_3 y_3^2}{2} + \frac{m_4 x_4^2}{2} + \frac{m_4 y_4^2}{2} + \frac{m_5 x_5^2}{2} + \frac{m_5 y_5^2}{2} + \\
 &\quad \frac{m_6 x_6^2}{2} + \frac{m_6 y_6^2}{2} + \frac{m_7 x_7^2}{2} + \frac{m_7 y_7^2}{2} + \frac{m_8 x_8^2}{2} + \frac{m_8 y_8^2}{2} + \\
 &\quad \frac{I_1 a_1^2}{2} + \frac{I_2 a_2^2}{2} + \frac{I_3 a_3^2}{2} + \frac{I_4 a_4^2}{2} + \frac{I_5 a_5^2}{2} + \frac{I_6 a_6^2}{2} + \frac{I_7 a_7^2}{2} + \frac{I_8 a_8^2}{2},
 \end{aligned} \tag{5}$$

where:

$I_1 - I_8$ - inertia moments of segments.

3. Graphical solution of dynamic model

The LabView environment is the ultimate graphical and text tool used to create virtual instruments, measurement, data processing, device control, and is an excellent mathematical environment in which it is possible to solve differential equations (including non-linear ones).

As in the case of LabView environment systems of differential equations can be calculated in two ways. The first way is to use ODE block (Ordinary Differential Equations): formula string, using the so-called. "clusters" in which the system of equations is input directly. The ODE block is available when you expand the Functions palette in Mathematics tab - Differential Equations - Ordinary Differential Equations - ODE solver (ODE VI) as shown in Fig. 2 [4].

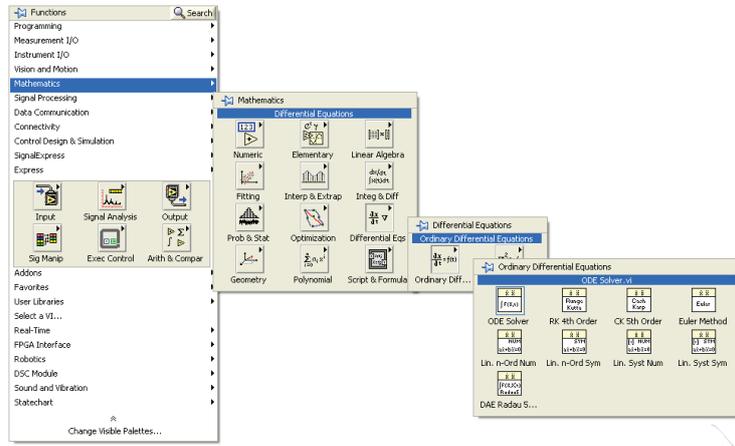


Fig. 2. Selecting a Ordinary Differential Equations bookmark - ODE solver

To transform the ODE VI into ODE formula string, click the right mouse button on the ODE VI (Solver) and then choose Select Type - ODE solver - F (x,t) is formula string, as shown in Fig. 3 [4].

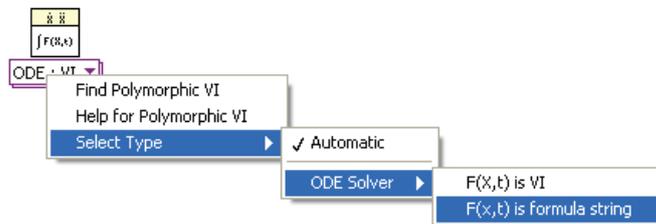


Fig. 3. Transformation of ODE VI into ODE formula string

Once the above-mentioned block has been created (*ODE formula string*) input and output of functions should be derived using the right mouse button. *ODE formula string* has six outputs, which are described below:

- The parameters under which simulation research is carried out - Simulation parameters,
- initial values - x_0 ,
- *ODE RHS*,
- times,
- Outputs
- Errors.

Inputs can be selected in two ways:

- *control* - control of parameters from the front panel in a real mode,
- *constant* reference values only possible when creating the program (without the possibility of changes during the program).

When creating a program, the order of parameter input and definition is optional. In the following example in Fig. 4 two variables $i1$ and $i2$ have been defined, and the input values are defined as *constant*. The example in Fig. 4 implements the following system of non-linear equations:

$$\frac{di1}{dt} = (-5t + 2)i1 + 2ti2, \quad (6)$$

$$\frac{di2}{dt} = (-3t + 1)i1 + 6ti2. \quad (7)$$

In the *Simulation Parameters* a dialogue box appears with parameters such as: start time, end time, tolerance, step and function. Selection of the number of initial values and equations is to stretch “cluster”.

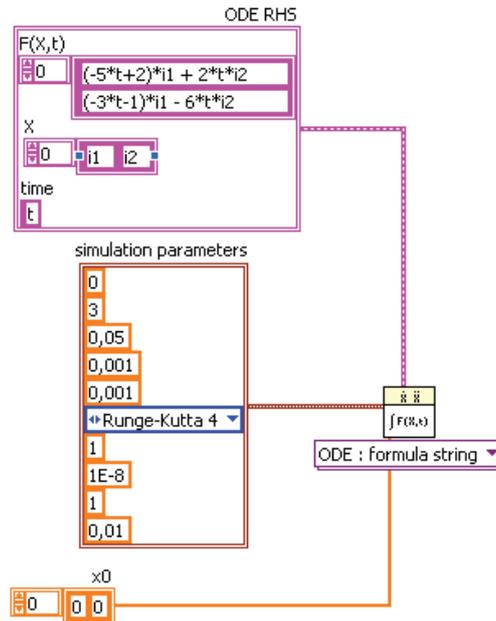


Fig. 4. Preparing the system of differential equations with two variables $i1$ and $i2$ to the solution

After saving the system of differential equations with two variables $i1$ and $i2$ state (Fig. 4) the next step is to develop the XY Graph that will show the waveforms of desired state variables. To do this, for the two state variables $i1$ and $i2$, one-dimensional arrays (with data) to “clusters” need to be replaced. Use the *Bundle* function located in the tab: *Programming - Cluster, Class & Variant*. When you replace the variables $i1$ and $i2$ into “clusters” a matrix using the *Build Array* should be developed. Fig. 5 and 6 show two windows respectively: *Block Diagram* (with a graphical program) and *Front Panel* (presenting the results of simulations), which implement the first way of solving the system of non-linear differential equations with two variables $i1$ and $i2$ state.

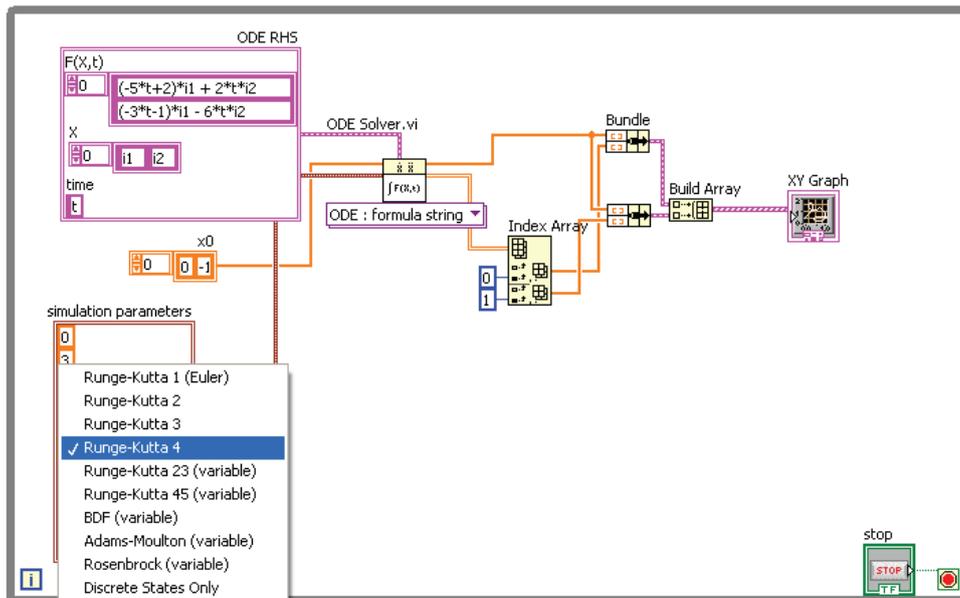


Fig. 5. The first way to solve the system of non-linear differential equations with two state variables - Block Diagram window with a graphical program

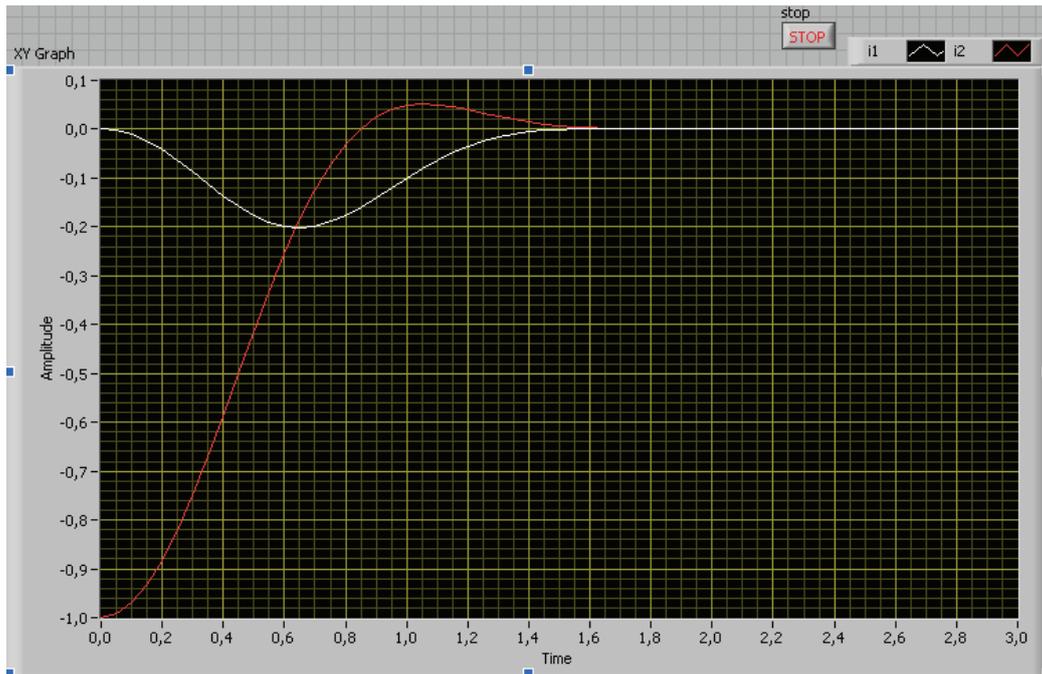


Fig. 6. The first way to solve the system of non-linear differential equations with two state variables - Front Panel window

The second way to solve the system of differential equations is to generate two vectors $i1$ and $i2$ using a sub-program calculating $F(x,t)$. Time in the program is generated using the *Ramp Pattern* function. Moreover, the created program uses for a loop and the structure of Formula Node. This way of programming using the structure of Formula Node is similar to Matlab, with the exception of the declaration of variables. Note, however, that the fault “Formula Node” block is the lack of calculations using complex numbers, which in the case of the dynamic model considered in this paper does not apply. The following steps show the second way of solving the system of non-linear differential equations with two state variables $i1$ and $i2$ together with a description of equations (6) and (7).

Figure 7 shows the structure of *Formula Node*, which during the input t time has been added, while during outputs by writing in the structure of *Formula Node* $a0$, $a1$, $a2$ and $a3$ parameters have been specified depending on t time along with the initial rhs.vi. ODE function template.

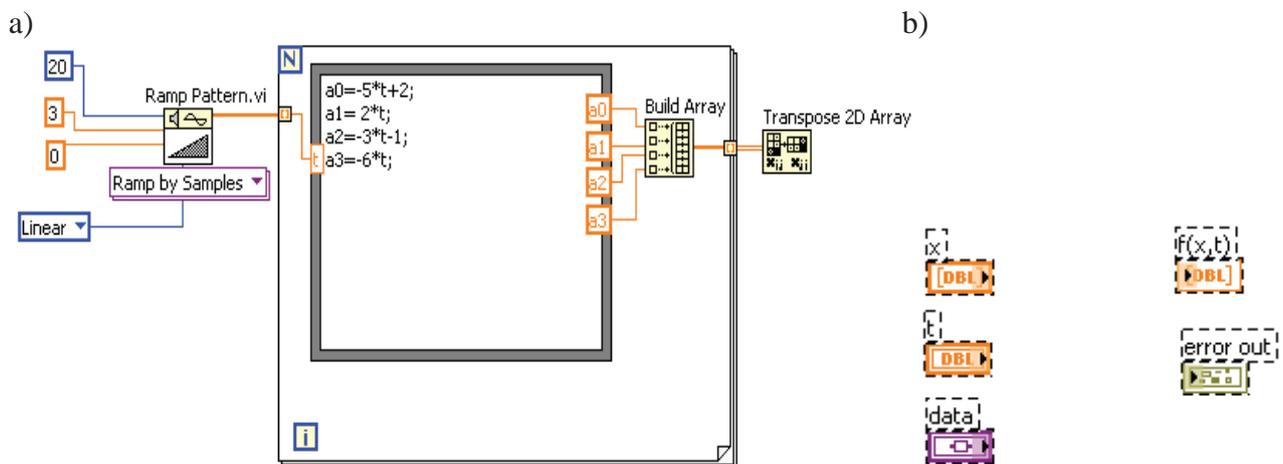


Fig. 7. Specification of a) time-dependent parameters using Formula Node structure and b) rhs.vi ODE function template

Using the *Build Array* function (Fig. 7), a matrix has been constructed, while at the output by using the *Transpose 2D Array* function a transpose matrix has been created. To check the vectors of parameters to a “cluster” form the *Bundle By Name* function needs to be used. Block *Bundle By Name* function inputs need to be set as *Constant* (as is the case in a given time step), while the outputs are connected to the *ODE VI* function. Other activities (apart from $F(x,t)$ function) are set as in the first example for *ODE: formula string*. In the case of $F(x,t)$, rhs ODE subprogram needs to be written setting up this feature in a separate window. *rhs ODE* function template can be found in LabView \ vi.lib \ gmath \ ode.llb \ “ODE rhs.vit.” Fig. 6 in addition to the structure of *Formula Node*, the template of created *rhs ODE VI* has also been presented. The subprogram, among others, uses the *date* functions. Fig. 8 shows the subprogram setting $i1$ and $i2$ derivatives as a function of $x'(t) = F(x,t)$.

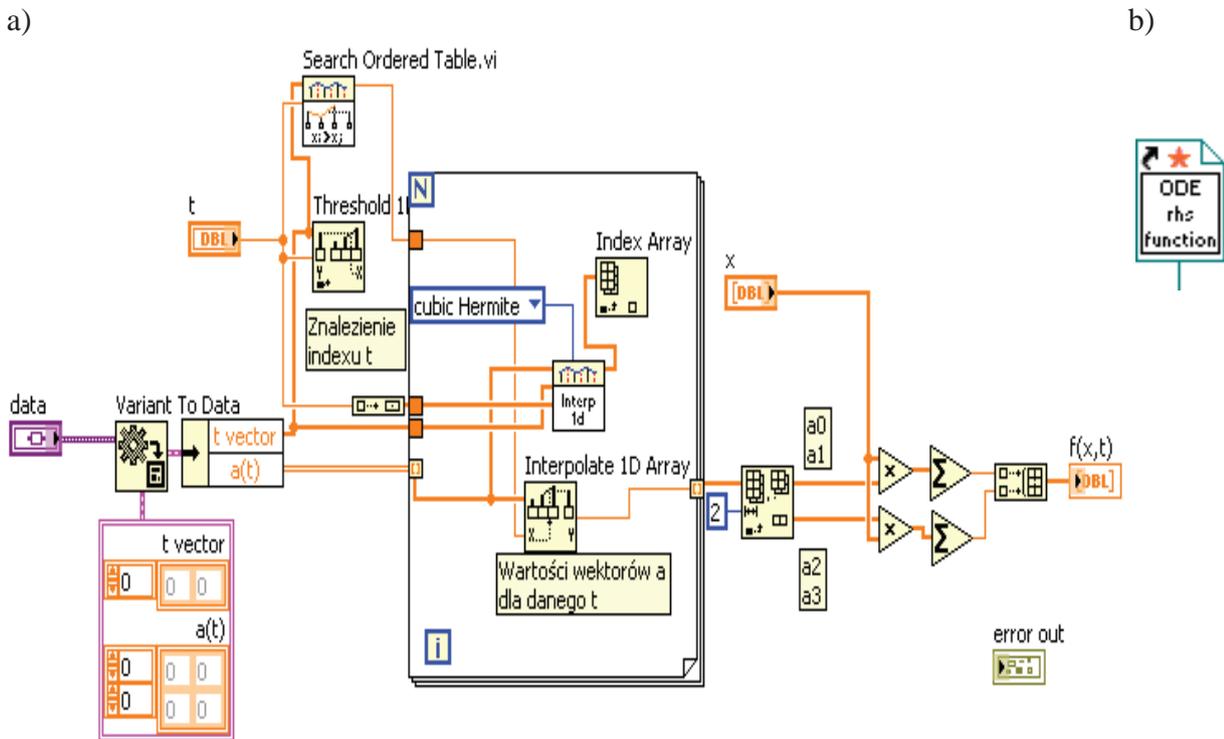


Fig. 8. Subprogram setting $i1$ and $i2$ derivatives as a function of a) the views of the Block Diagram window, and b) view of the created VI

To connect the formed *ODE rhs* block(Fig. 8) to created main program *ODE rhs* should be previously saved as VI and go to the main program. In the main program, click the right mouse button on the *ODE rhs* pad (VI setting) and go to the *Application Control* palette, and then select the *Static VI Reference*. Then there is a folder with a question mark, which should be clicked with the right mouse button and choose the path of access to the *Browse for Patch* subprogram and select created *ODE rhs* subprogram (which has been previously saved). The next step is to re-click the right mouse button on the *ODE rhs* pad and select *Strictly Typed VI Reference*. Such a ready file with the subprogram must be connected to the *ODE VI* function as shown in Fig. 9. Fig. 9 and 10 show respectively the *Block Diagram* window with the graphical program and the *Front Panel* window. These windows realise the second way of solving the system of non-linear differential equations with two variables $i1$ and $i2$ state.

Waveforms of Fig. 6 and 10 are identical. Both ways of solving non-linear differential equations can therefore be used for simulation studies to create a dynamic model of 10 degrees of freedom, which will be presented below.

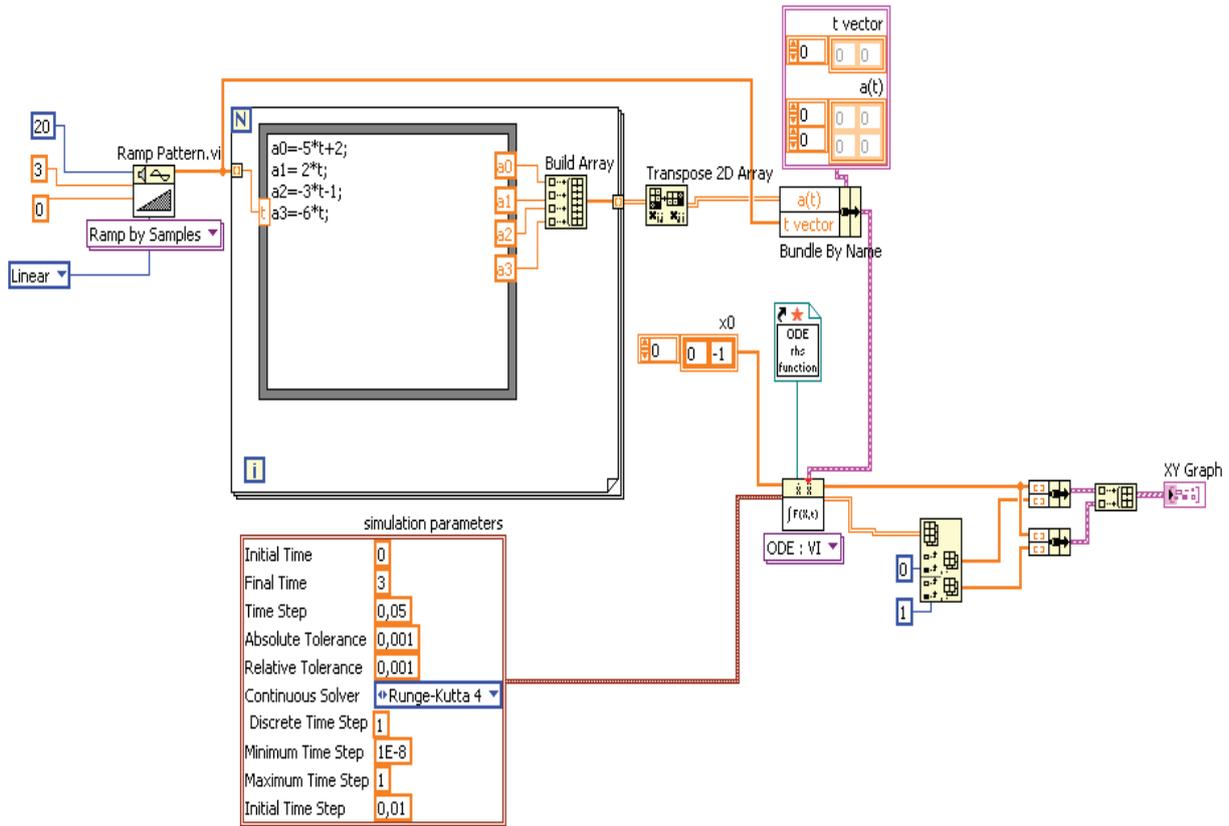


Fig. 9. The second way to solve the system of non-linear differential equations with two state variables - Block Diagram window with a graphical program, b) Front Panel window

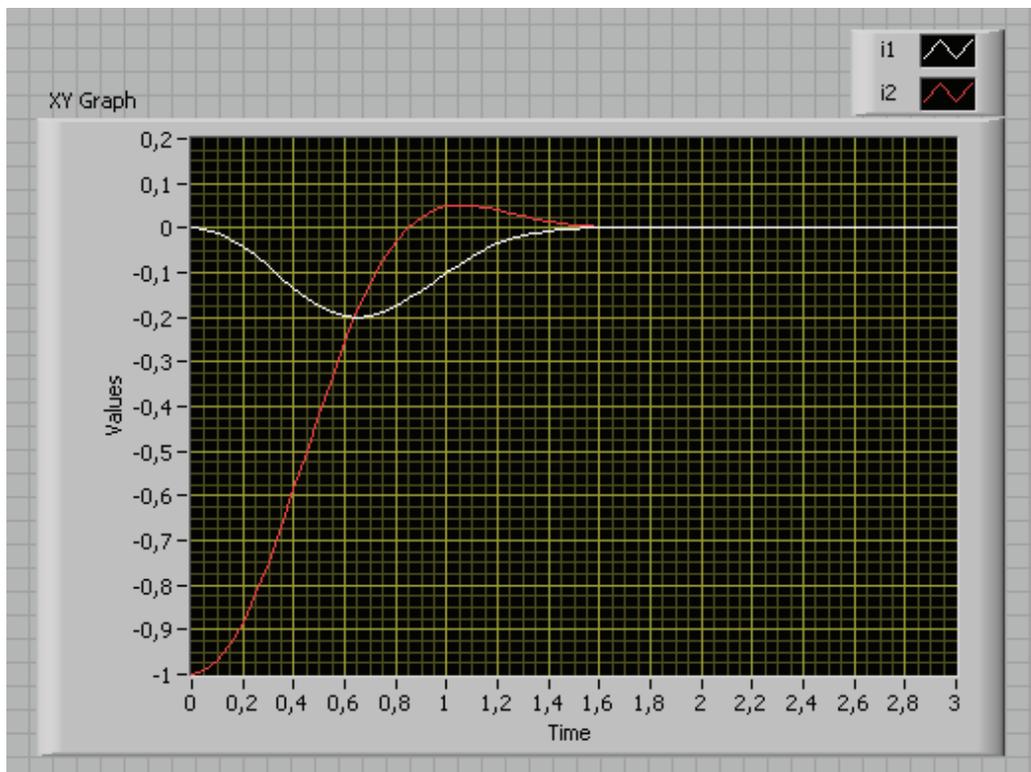


Fig. 10. The second way to solve the system of non-linear differential equations with two state variables - Front Panel window

4. Dynamic model of 10 degrees of freedom realized by using the graphical environment

For the dynamic model of 10 degrees of freedom (the driver sitting in a car seat, fastened with seatbelts) described with non-linear dependencies 1-5 graphical software in G language in LabView has been developed on the basis of which, it is possible to carry out simulation studies of the impact of various factors on the waveforms of desired state variables. Fig. 11 shows the view of the *Front Panel* window with a part of the program showing how to connect the analogue units (i.e. experimentally adjustable rates seen on the left) to the structure of *Formula Node*. The window shown in Fig. 11 enables the change of the following factors: $m_1 - m_8$ – the weight of individual blocks, $l_1 - l_8$ – the length of individual blocks, $l_{1sr} - l_{8sr}$ – the length of individual blocks from the centre of each block, $I_1 - I_8$ – moments of inertia and $\alpha_{1p} - \alpha_{8p}$ – angles of the initial position of each block.

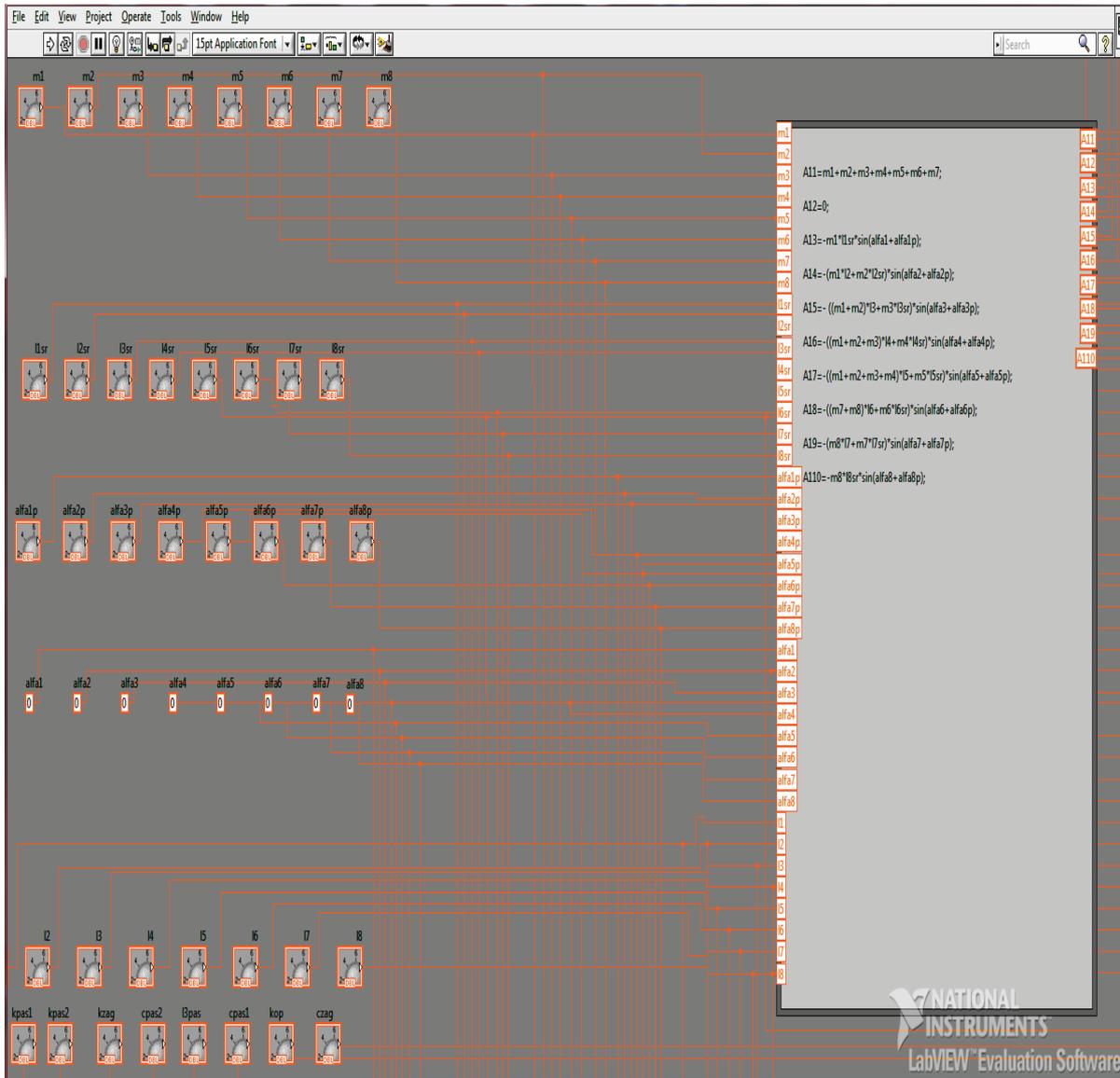


Fig. 11. A part of the created program - view of the Block Diagram window

Program part shown in the *Formula Node* window calculates the following factors:

$$\begin{aligned}
 A(1,1) &= m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7 \\
 A(1,2) &= 0 \\
 A(1,3) &= -m_1 * l_{1sr} * \sin(\alpha_{1a} + \alpha_{1p}) \\
 A(1,4) &= -(m_1 * l_2 + m_2 * l_{2sr}) * \sin(\alpha_{2a} + \alpha_{2p})
 \end{aligned}$$

$$A(1,5) = -((m1+m2)*l3+m3*l3sr)*sin(alfa3+alfa3p)$$

$$A(1,6) = -((m1+m2+m3)*l4+m4*l4sr)*sin(alfa4+alfa4p)$$

$$A(1,7) = -((m1+m2+m3+m4)*l5+m5*l5sr)*sin(alfa5+alfa5p)$$

$$A(1,8) = -((m7+m8)*l6+m6*l6sr)*sin(alfa6+alfa6p)$$

$$A(1,9) = -(m8*l7+m7*l7sr)*sin(alfa7+alfa7p)$$

$$A(1,10) = -m8*l8sr*sin(alfa8+alfa8p)$$

Figure 12 shows the view to the *Front Panel* window of the part of program showing described analogue units shown in Fig. 11.

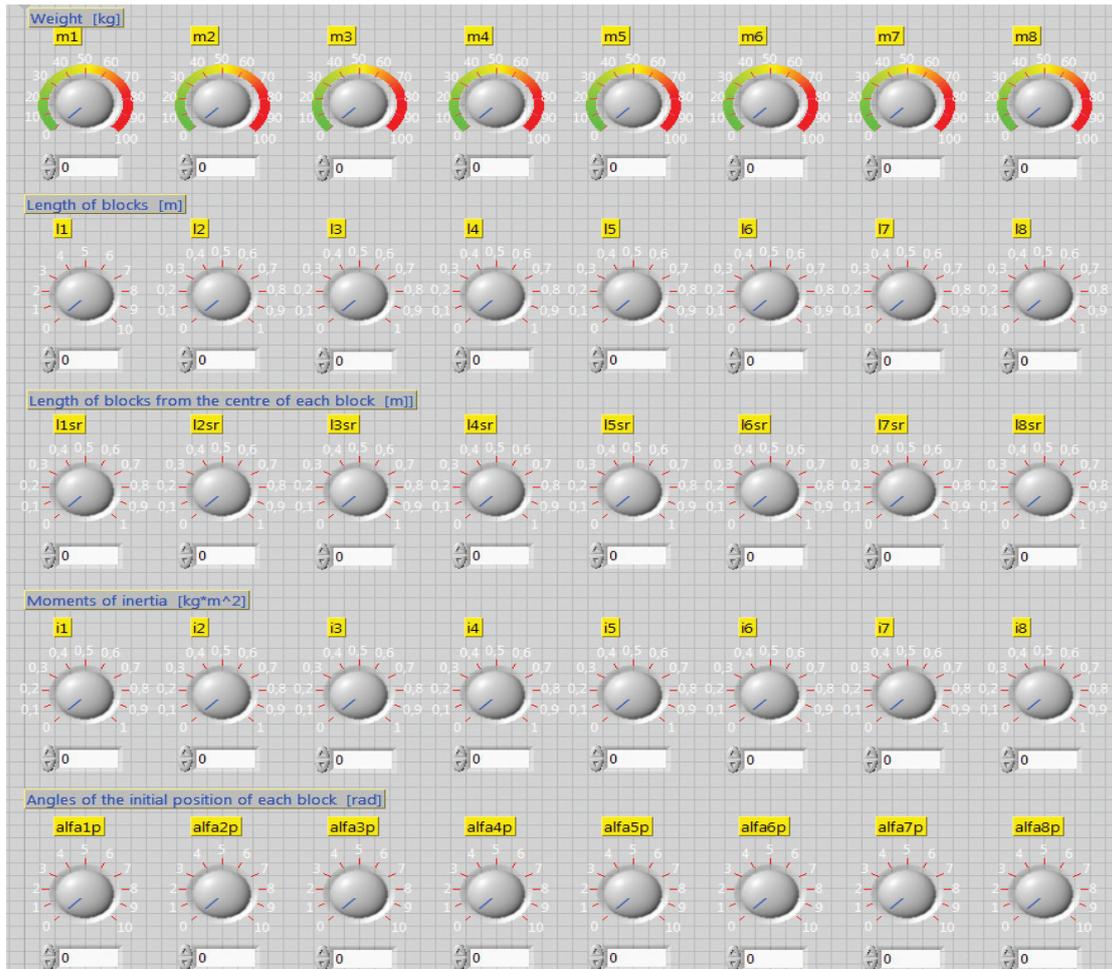


Fig. 12. View of the *Front Panel* window of the part of a program created with analogue units (from Fig. 11)

In a similar way as in Fig. 11 and 12, other factors need to be connected (k_{pas1} , k_{pas2} , k_{op} – the coefficients of stiffness of the waist and back, k_{p1} - k_{p8} – coefficients of stiffness of joints in blocks, c_{pas} , c_{op} – damping and seat belt coefficients, c_1 - c_8 , c_{p1} - c_{p8} – damping coefficients joints in blocks) to the structure of *Formula Node*. Fig. 13 shows another part of the program that uses the *Front Panel* window and two windows of *Formula Node* structure to calculate the following *B* and *Bzag* coefficients.

The structure of the *Formula Node* on the left (Fig. 13):

$$B(1,1) = 0$$

$$B(1,2) = 0$$

$$B(1,3) = (-m1*l1sr*cos(alfa1+alfa1p))^2$$

$$B(1,4) = -(m1*l2+m1*l2sr*cos(alfa2+alfa2p))^2$$

$$B(1,5) = -(((m1+m2)*l3+m3*l3sr)*cos(alfa3+alfa3p))^2$$

$$B(1,6) = -((m1+m2+m3)*l4+m4*l4sr)*cos(alfa4+alfa4p))^2$$

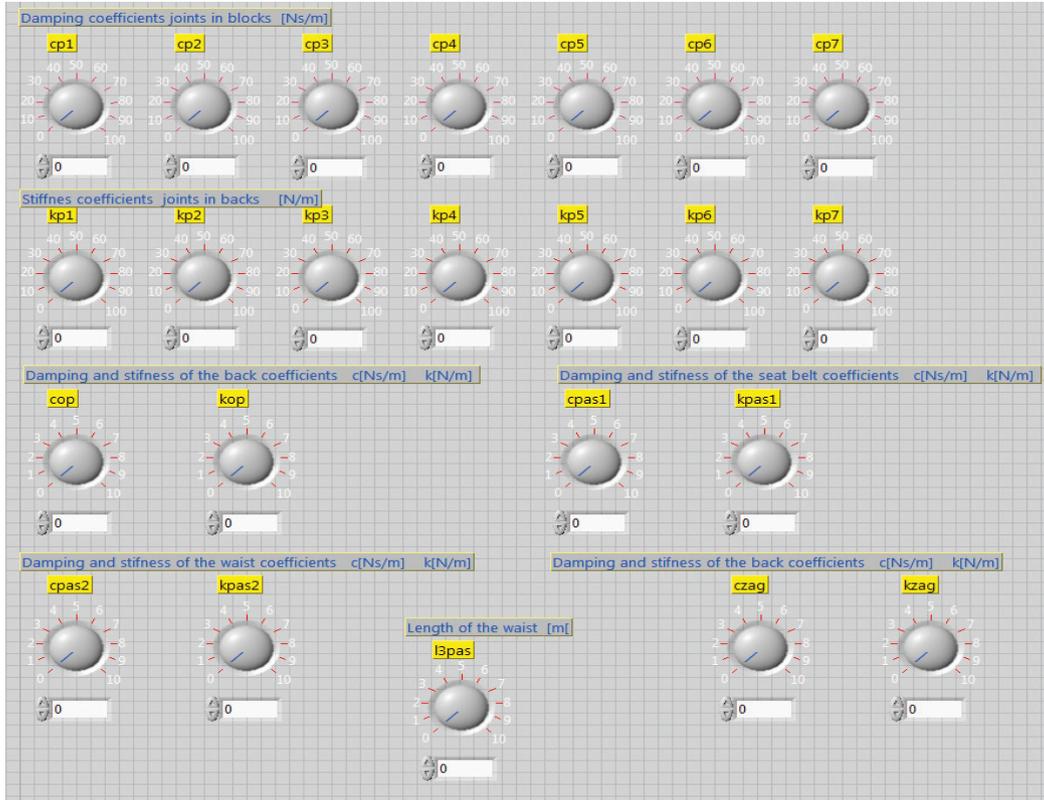
$$B(1,7) = -((m1+m2+m3+m4)*l5+m5*l5sr)*cos(alfa5+alfa5p))^2$$

$$B(1,8) = -((m7+m8)*l6+m6*l6sr)*cos(alfa6+alfa6p))^2$$

$$B(1,9) = -(m8*l7+m7*l7sr)*cos(alfa7+alfa7p))^2$$

$$B(1,10) = -(m8*l8sr)*cos(alfa8+alfa8p))^2$$

a)



b)

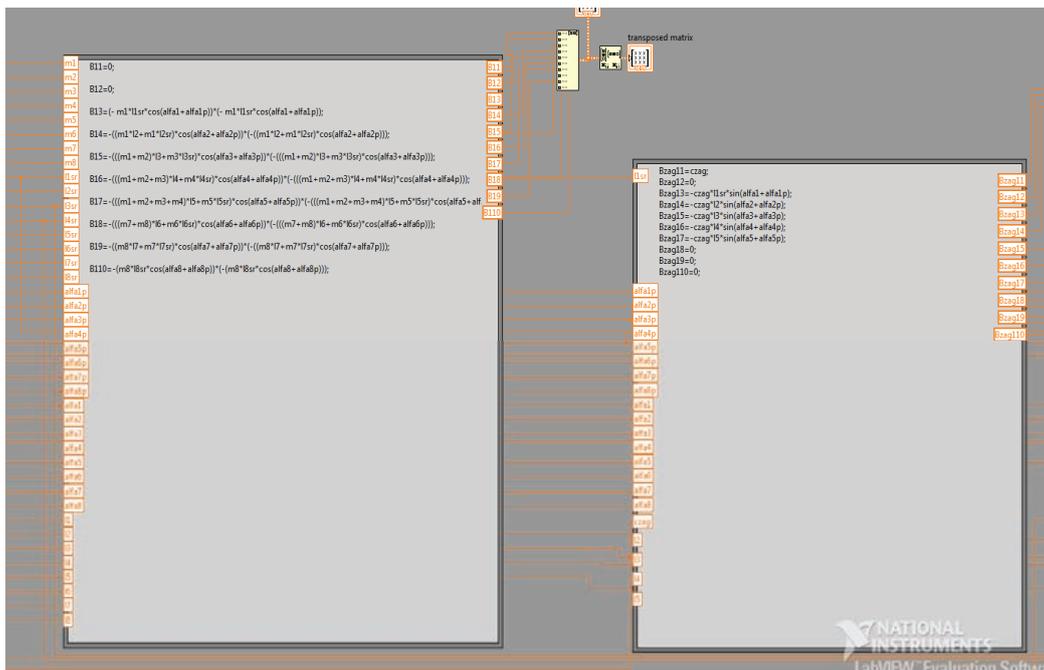


Fig. 13. Another part of the program (shown in Fig. 11) that uses the Front Panel window and two windows of Formula Node structure to calculate the following B and Bzag coefficients

The structure of the *Formula Node* on the right side (Fig. 13):

$$Bzag(1,1)=czagl$$

$$Bzag(1,2)=0$$

$$Bzag(1,3)=-czagl * l1sr * \sin(\alpha1 + \alpha1p)$$

$$Bzag(1,4)=-czagl * l2 * \sin(\alpha2 + \alpha2p)$$

$$Bzag(1,5)=-czagl * l3 * \sin(\alpha3 + \alpha3p)$$

$$Bzag(1,6)=-czagl * l4 * \sin(\alpha4 + \alpha4p)$$

$$Bzag(1,7)=-czagl * l5 * \sin(\alpha5 + \alpha5p)$$

$$Bzag(1,8)=0$$

$$Bzag(1,9)=0$$

$$Bzag(1,10)=0$$

After completing the remaining equations, graphical visualization of the impact of individual factors, and desired shape parameters for the waveforms, needs to be started with non-zero coefficient settings $m_1 - m_8$, $l_1 - l_8$, $l_{1sr} - l_{8sr}$, $I_1 - I_8$, k_{pas1} , k_{pas2} , k_{op} , $k_{p1} - k_{p8}$, c_{pas} , c_{op} , $c_1 - c_8$, $c_{p1} - c_{p8}$ and $\alpha_{1p} - \alpha_{8p}$. Then, by input of particular displacements $y_1 - y_8$, $x_1 - x_8$ and $\alpha_1 - \alpha_8$ angles it is possible to visualize the impact of various parameters on the speed waveforms, acceleration or energy, etc.

5. Conclusions

The article presents the possibility of using LabView graphical environment to build a dynamic model (the driver sitting in a car seat, fastened with seatbelts) of 10 degrees of freedom. For this purpose, a program has been developed in G graphical language (in the LabView environment) which by changing the settings of analogue units allows continuous visualization of the impact of parameters on the delay waveforms, speed, energy, etc. calculated based on the adopted model and depending on parameter values. The illustrated graphical program uses *Formula Node* structure, which is connected with analogue units representing particular coefficients and parameters of the dynamic model. The advantage of the *Formula Node* structure is that the programming method is similar to Matlab, which especially for researcher's familiar with Matlab environment greatly simplifies creating graphics software in LabView. The program presented in the following work can be used for initial graphical identification of the dynamic model parameters. Influence of parameters $m_1 - m_8$, $l_1 - l_8$, $l_{1sr} - l_{8sr}$, $I_1 - I_8$, k_{pas1} , k_{pas2} , k_{op} , $k_{p1} - k_{p8}$, c_{pas} , c_{op} , $c_1 - c_8$, $c_{p1} - c_{p8}$ and $\alpha_{1p} - \alpha_{8p}$, as well as desired $y_1 - y_8$, $x_1 - x_8$ and $\alpha_1 - \alpha_8$ on delay waveforms, speed, energy, etc. is a large spectrum of solutions and will be the subject of another work of the authors.

References

- [1] Jaśkiewicz, M., Stańczyk, T. L., *The analysis of multi-section concept for an active air headrest*, Polskie Towarzystwo Diagnostyki Technicznej, Diagnostyka 3(55), 2010.
- [2] Jaśkiewicz, M., Stańczyk, T. L., *The identification of damping and stiffness parameters of a driver model on the basis of crash tests*, Journal of KONES Powertrain and Transport, Vol. 16, No. 1, pp. 229-238, Warszawa 2009.
- [3] Jaśkiewicz, M., Stańczyk, T., *Analysis of the concept of multiple-air active head restraint*, 9th International Technical Systems Degradation Conference, pp. 36-38, Liptowsky Mikulas 2010.
- [4] *LabView. Control Design User Manual*, National Instruments, June 2009.
- [5] Ludwinek, K., Staszak, J., *Possibility of graphical environment applications for evaluating of equivalent circuit parameters and time constants*, Electrical Review (the article has been accepted for publication).

- [6] Pankoke, S., *Numerische Simulation des räumlichen Ganzkörperschwingungsverhaltens des sitzenden Menschen unter Berücksichtigung der individuellen Anthropometrie und Haltung*, Dusseldorf 2003.
- [7] Wicher, J., *Safety of cars and road traffic*, WKiŁ, Warszawa 2002.