



TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics and Mechanics

Vol. 56, Issue IV, November, 2013

## THE DYNAMIC MODELLING OF THE ROBOT MECHANICAL STRUCTURE USING THE SYMBOLIC COMPUTATION IN MATLAB

Ovidiu-Aurelian DETEȘAN

**Abstract:** An important part of modelling and simulation of a robot is the generation of the geometric, kinematic and dynamic model of its mechanical structure. The software module presented in this paper allows the modelling of the generalized mechanical structure of the robot and the automatic generation of the dynamic model equations of the considered robot, i.e. the motive generalized forces from the robot's joints. As a part of the *Robot\_Symbolic* generalized application, after *Robot\_Definition*, *Robot\_Geometry* and *Robot\_Kinematics*, *Robot\_Dynamics* is the last step to follow when the modelling of a robotic structure is intended. The application is written in MATLAB, it exploits its symbolic computation facilities, implementing the Newton-Euler's formulation. **Key words:** serial robot, dynamic modelling, symbolic computation, *Robot\_Dynamics*.

### 1. INTRODUCTION

The dynamic modelling of a robot is an important step in modelling the mechanical structure of the robot to be analyzed. The symbolic computation deals with symbolic objects, the input data being symbolic and, sometimes, numeric, while the output data are algebraic expressions. The main advantage of the symbolic computation in the robot dynamic modelling consists in direct generation of the symbolic form of the equations of the generalized forces needed to ensure the position, the orientation, the linear and angular velocity and acceleration of the end-effector, considering the action of the dynamic parameters such as mass properties – masses, mass centres, moments of inertia as well as the payload. The obtained data are useful to generate the simulation of the dynamic behaviour of the considered robot.

### 2. THE SYMBOLIC COMPUTATION IN MATLAB

MATLAB has a well defined position among many other software products for symbolic computation, being a worldwide

standard for technical computing, offering a lot of advantages, as presented in [2], [4] [5], [10], [11]: efficiency in engineering computing; the presence of its own high level programming language, with C like syntax and semantics; the presence of both the interpreter and the compiler; the portability; the interface ability with the common programming languages (C, Java) and databases management systems; the presence of libraries with hundreds of predefined functions.

MATLAB implements now a unique symbolic toolbox: the *Symbolic Math Toolbox*, including hundreds of MATLAB functions while the previous versions had the *Extended Symbolic Math Toolbox* which increased its symbolic functionality.

### 3. THE *Robot\_Symbolic* PACKAGE

The *TRR\_Symbolic* application [1], [3] was a starting point for designing a generalized application for symbolic modelling of robots with a defined number of degrees of freedom, between 1 and 6. The graphics user interface of the application is presented in Fig. 1. The application described in this paper is completely general and interactive and the

input data could be saved and reused as necessary.

The generalized modelling features of *Robot\_Symbolic* package consist in four main modules: *Robot\_Definition*, *Robot\_Geometry*, *Robot\_Kinematics* and *Robot\_Dynamics*. The whole structure of *Robot\_Symbolic* package is presented in the Table 1.

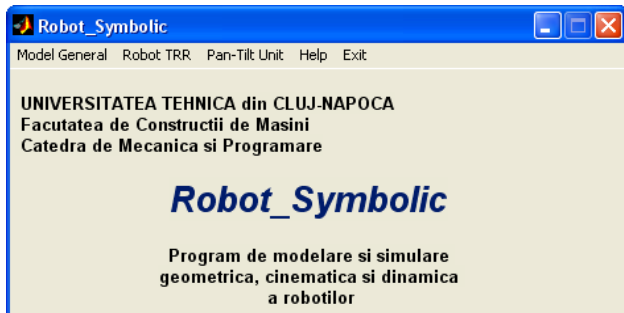


Fig. 1. The graphics user interface of *Robot\_Symbolic* application

Table 1

The structure of *Robot\_Symbolic* application

Menu options	Application files
<i>General Model</i>	
<i>Structure Definition</i>	<i>Robot_definition.m</i>
<i>Geometric Model</i>	<i>Robot_geometry.m</i>
<i>Kinematic Model</i>	<i>Robot_kinematics.m</i>
<i>Dynamic Model</i>	<i>Robot_dynamics.m</i>
<i>TRR Robot</i>	
<i>Geometric Model</i>	<i>TRR_geometry.m</i>
<i>Kinematic Model</i>	<i>TRR_kinematics.m</i>
<i>Dynamic Model</i>	<i>TRR_dynamics.m</i>
<i>Graphs</i>	
<i>Geometry/Kinematics</i>	
<i>Linear</i>	<i>graphTRR_geom_kin.m</i>
<i>Spline</i>	<i>graphTRR_geom_kin_spline.m</i>
<i>Dynamics</i>	
<i>Linear</i>	<i>graphTRR_dyn.m</i>
<i>Spline</i>	<i>graphTRR_dyn_spline.m</i>
<i>Pan-Tilt Unit</i>	
<i>Graphs</i>	
<i>Linear</i>	<i>graphsPTU_geom_kin.m</i>
<i>Spline</i>	<i>graphsPTU_geom_kin_spline.m</i>
<i>Help</i>	<i>help.m</i>
<i>Exit</i>	<i>gata.m</i>

#### 4. THE *Robot\_Dynamics* MODULE

The *Robot\_Dynamics* module, contained in the file *Robot\_dynamics.m*, uses data generated at the runtime of the robot kinematic modelling, *Robot\_Kinematics*, with the purpose of determining the symbolic dynamic model of the robot to be analyzed, using the Newton-Euler formulation [6], [7], [8], [9]. The notations used

in the program, compared to those used in [7], are presented in the Table 2.

Table 2

Notations in *Robot\_Dynamics*

Notation in [6]	Notation in the program	Notation in [6]	Notation in the program
$M_1 \dots$	M1 ...	${}^1Q_m \dots$	Qm1 ...
${}^1J_x^* \dots$	Ix1s ...	${}^1\bar{r}_{C_1} \dots$	rc11 ...
${}^4f_x \dots$	fx4 ...	${}^1I_1^* \dots$	I11s ...
${}^4\bar{f}_4 \dots$	f44 ...	${}^1\dot{v}_{C_1} \dots$	vc11dot...
${}^4n_x \dots$	nx4 ...	${}^1\bar{F}_1 \dots$	F11 ...
${}^4\bar{n}_4 \dots$	n44 ...	${}^1\bar{N}_1 \dots$	N11 ...

#### 4.1 How the Program Works

Let us consider a robot with *dof* degrees of freedom, consisting in *dof* rotation or translation kinematic joints, linked together in a serial kinematic chain defining its mechanical structure. At the beginning, the program asks for the name of the robot:

```
Robot name: <name>
The robot <name> has no kinematic model defined. To define the model, press <K>, to exit, press <X>:
```

The program checks if the robot structure with the name *<name>* has its kinematic model defined. If so, the data for the kinematic model will be loaded, from the file *name\_kin.mat*. If not, the user can opt for either launching the kinematic modelling module or the program exit. Because *Robot\_Dynamics* contains also an interactive symbolic data input stage, the program checks if the robot with that name has already a dynamic model. If yes, the user can choose either to redefine the data by pressing *<R>*, or to load the existing data by pressing *<C>*.

The (re)defined data are the mass centre position vector of the elements (*i*) with respect to the frame  $\{i\}$  origin.

In order to keep a clean workspace, the symbolic variables will be declared in a dynamic fashion, as in [2]. The following variables will be declared:

- the masses: *M1*...
- the axial and centrifugal mechanical moments of inertia: *Ix1s*..., *Iy1s*..., *Iz1s*...
- the generalized motive forces: *Qm1*...
- the link joint forces: *f11*...

- the moments of the joint forces:  $nI1...$
- the external forces:  $FII...$
- the moments of the external forces:  $NII...$
- the mass centers position vectors:  $rCII...$
- the mass centers accelerations:  $vCII\dot{...}$
- the components of the payload – the force, having the components  $fx, fy, fz$  and the moment with its components,  $nx, ny, nz$ .

The dynamic declaration code is presented as follows:

```
str_syms_d='syms ';
for i=1:dof
str_syms_d=[str_syms_d,'M',num2str(i),'...
Ix',num2str(i),'s','Iy',num2str(i),'s',...
'Iz',num2str(i),'s','Ixy',num2str(i),...
's','Ixz',num2str(i),'s','Iyx',...
num2str(i),'s','Iyz',num2str(i),'s',...
'Izx',num2str(i),'s','Izy',num2str(i),...
's','Qm',num2str(i),'n',num2str(i),...
num2str(i),'f',num2str(i),num2str(i),...
'N',num2str(i),num2str(i),'F',...
num2str(i),num2str(i),'rC',num2str(i),...
num2str(i),'x','rC',num2str(i),...
num2str(i),'y','rC',num2str(i),...
num2str(i),'z','vC',num2str(i),...
num2str(i),'dot'];
end
str_syms_d=[str_syms_d,'fx',...
num2str(dof+1),'fy',num2str(dof+1),...
'fz',num2str(dof+1),'nx',...
num2str(dof+1),'ny',num2str(dof+1),...
'nz',num2str(dof+1),'rC',...
num2str(dof+1),num2str(dof+1),'x',...
'rC',num2str(i),num2str(dof+1),'y',...
'rC',num2str(i),num2str(dof+1),'z'];
str_syms_d=[str_syms_d,'real'];
eval(str_syms_d);
```

The joint type discriminant is defined as:

```
for i=1:dof
if tip(i)=='R'
delta(i)=1;
else
delta(i)=0;
end
end
```

The generalized inertial tensor is specified by the following code sequence:

```
for i=1:dof
eval(['I',num2str(i),num2str(i),'s=[Ix',...
num2str(i),'s -Ixy',num2str(i),'s -Ixz',...
num2str(i),'s; -Iyx',num2str(i),'s Iy',...
num2str(i),'s -Iyz',num2str(i),'s; -Izx',...
num2str(i),'s -Izy',num2str(i),'s Iz',...
num2str(i),'s]']);
end
```

The useful payload is generally defined as the payload force and moment:

```
eval(['f',num2str(dof+1),num2str(dof+1),...
'=[fx',num2str(dof+1),'fy',...
num2str(dof+1),'fz',num2str(dof+1),']']);
```

```
eval(['n',num2str(dof+1),num2str(dof+1),...
'=[nx',num2str(dof+1),'ny',...
num2str(dof+1),'nz',num2str(dof+1),']']);
```

The Newton-Euler formulation comprises two main stages: (1) iterations outwards the mechanical structure and (2) iterations inwards the mechanical structure of the robot.

In the first stage, according to [1], [7], [9], the mass centres accelerations corresponding to each element are determined, as:

```
for i=1:dof
eval(['vC',num2str(i),num2str(i),...
'dot=simplify(v',num2str(i),num2str(i),...
'dot+antisym(omega',num2str(i),...
num2str(i),'dot)*rC',num2str(i),...
num2str(i),'+antisym(omega',num2str(i),...
num2str(i),'*antisym(omega',num2str(i),...
num2str(i),'*rC',num2str(i),...
num2str(i),'')]);
```

The program goes on by determining the external forces and their moments:

```
for i=1:dof
eval(['F',num2str(i),num2str(i),...
'=simplify(M',num2str(i),'*vC',...
num2str(i),num2str(i),'dot')]);
end
for i=1:dof
eval(['N',num2str(i),num2str(i),...
'=simplify(I',num2str(i),num2str(i),...
's*omega',num2str(i),num2str(i),...
'dot+antisym(omega',num2str(i),...
num2str(i),'*I',num2str(i),num2str(i),...
's*omega',num2str(i),num2str(i),'')]);
end
```

The link forces and their moments, corresponding to the second stage, are symbolically computed as:

```
for i=dof:-1:1
eval(['f',num2str(i),num2str(i),...
'=simplify(R',num2str(i+1),num2str(i),...
'*f',num2str(i+1),num2str(i+1),'+F',...
num2str(i),num2str(i),'')]);
end
for i=dof:-1:1
eval(['n',num2str(i),num2str(i),...
'=simplify(R',num2str(i+1),num2str(i),...
'*n',num2str(i+1),num2str(i+1),...
'+antisym(rC',num2str(i),num2str(i),...
')*F',num2str(i),num2str(i),'+antisym(r',...
num2str(i+1),num2str(i),'*R',...
num2str(i+1),num2str(i),'*f',...
num2str(i+1),num2str(i+1),'+N',...
num2str(i),num2str(i),'')]);
end
```

The program determines at last the generalized motive forces  $Qm1...$ , which represent the dynamic equations of the studied robot:

```
for i=1:dof
```

```
eval(['Qm',num2str(i),...
    '=simplify(delta(i)*n',num2str(i),...
    num2str(i),' '*k',num2str(i),num2str(i),...
    '+ (1-delta(i))*f',num2str(i),num2str(i),...
    ' '*k',num2str(i),num2str(i),'')]);
end
```

The symbolic data computed by this module is saved in the output file *name\_dyn.mat*, wherefrom they can be used for the subsequent simulations of the dynamic behaviour of the analyzed robot.

The *antisym()* function, used in the determination of the mass centre accelerations and of the moments of external and link forces, is a user defined function also used in [4] and its purpose is to compute the skew matrix associated to the vector transmitted as a parameter.

#### 4.2. An Use Case of *Robot\_Dynamics*: the TRR Mechanical Structure

An example to determine the symbolic equations of the dynamic model using the above described program is given as follows, for the mechanical structure of the TRR robot defined in [3], whose kinematic diagram is shown in fig. 2. The input data assumes that the mass centres are situated at half length of the links, considered as homogeneous bars and the centrifugal terms of the inertial tensors are

assumed to be zero, due to a simplifying hypothesis which says that the moments of inertia are written with respect to a frame  $\{i^*\}$  considered with the origin in the mass centre  $C_i$ , having the same orientation as the frame attached to each element ( $i$ ) of the robot.

The mass centres have the expressions:

```
rC11 =
[ 0
 0
 1/2*11]
rC22 =
[ 0
 1/2*13]
[ 1/2*12]
rC33 =
[ 0
 1/2*14]
[ 0]
```

and the inertial tensors are expressed as:

```
I11s =
[ Ix1s, 0, 0]
[ 0, Iy1s, 0]
[ 0, 0, Iz1s]
I22s =
[ Ix2s, 0, 0]
[ 0, Iy2s, 0]
[ 0, 0, Iz2s]
I33s =
[ Ix3s, 0, 0]
[ 0, Iy3s, 0]
[ 0, 0, Iz3s].
```

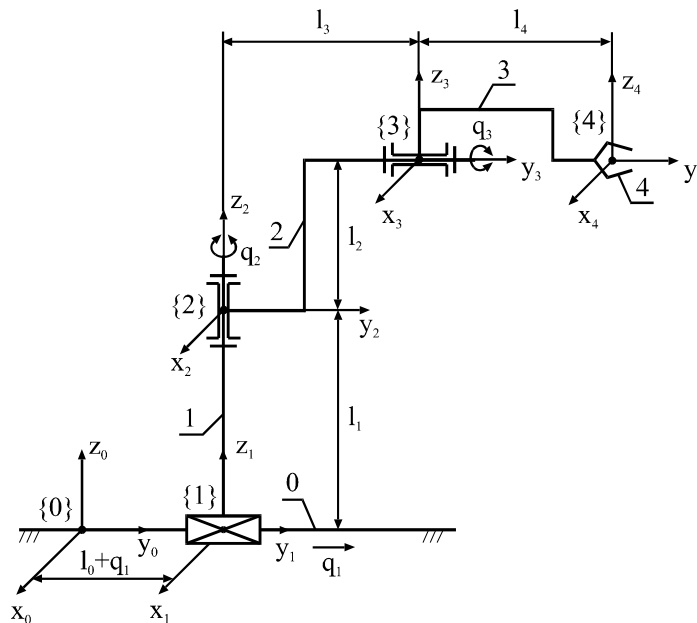


Fig. 2. The kinematic diagram of the TTR robot

The payload assumes all its six components to be non-zero:

$$\begin{aligned} \mathbf{f44} &= \\ &[ \text{fx4} ] \\ &[ \text{fy4} ] \\ &[ \text{fz4} ] \\ \mathbf{n44} &= \\ &[ \text{nx4} ] \\ &[ \text{ny4} ] \\ &[ \text{nz4} ]. \end{aligned}$$

The computed mass centres accelerations are:

$$\begin{aligned} \mathbf{vC11dot} &= \\ &[ \quad 0 ] \\ &[ \text{q1ddot} ] \\ &[ \quad \text{g} ]. \\ \mathbf{vC22dot} &= \\ &[ \sin(q2)*\text{q1ddot}-1/2*\text{q2ddot}*l3 ] \\ &[ \cos(q2)*\text{q1ddot}-1/2*\text{q2dot}^2*l3 ] \\ &[ \quad \quad \quad \text{g} ] \\ \mathbf{vC33dot} &= \\ &[ \cos(q3)*\sin(q2)*\text{q1ddot}- \\ &\cos(q3)*\text{q2ddot}*l3-\sin(q3)*\text{g}- \\ &1/2*l4*\cos(q3)*\text{q2ddot} ] \\ &[ \cos(q2)*\text{q1ddot}-\text{q2dot}^2*l3-1/2*l4*\text{q2dot}^2 ] \\ &[ \sin(q3)*\sin(q2)*\text{q1ddot}- \\ &\sin(q3)*\text{q2ddot}*l3+\cos(q3)*\text{g}- \\ &1/2*l4*\sin(q3)*\text{q2ddot} ]. \end{aligned}$$

The external forces have the expressions:

$$\begin{aligned} \mathbf{F11} &= \\ &[ \quad 0 ] \\ &[ M1*\text{q1ddot} ] \\ &[ \quad M1*\text{g} ] \\ \mathbf{F22} &= \\ &[ M2*\sin(q2)*\text{q1ddot}-1/2*M2*\text{q2ddot}*l3 ] \\ &[ M2*\cos(q2)*\text{q1ddot}-1/2*M2*\text{q2dot}^2*l3 ] \\ &[ \quad \quad \quad M2*\text{g} ] \\ \mathbf{F33} &= \\ &[ M3*\cos(q3)*\sin(q2)*\text{q1ddot}- \\ &M3*\cos(q3)*\text{q2ddot}*l3-M3*\sin(q3)*\text{g}- \\ &1/2*M3*l4*\cos(q3)*\text{q2ddot} ] \\ &[ M3*\cos(q2)*\text{q1ddot}-M3*\text{q2dot}^2*l3- \\ &1/2*M3*l4*\text{q2dot}^2 ] \\ &[ M3*\sin(q3)*\sin(q2)*\text{q1ddot}- \\ &M3*\sin(q3)*\text{q2ddot}*l3+M3*\cos(q3)*\text{g}- \\ &1/2*M3*l4*\sin(q3)*\text{q2ddot} ]. \end{aligned}$$

The moments of the external forces are calculated as:

$$\begin{aligned} \mathbf{N11} &= \\ &[ 0 ] \\ &[ 0 ] \\ &[ 0 ] \\ \mathbf{N22} &= \\ &[ \quad 0 ] \\ &[ \quad 0 ] \\ &[ I_z2s*\text{q2ddot} ] \\ \mathbf{N33} &= \\ &[ -I_x3s*\sin(q3)*\text{q2ddot}-I_x3s*\text{q3dot}*\cos(q3) \\ &*\text{q2dot}-\cos(q3)*\text{q2dot}*I_y3s*\text{q3dot}+\text{q3dot}* \\ &I_z3s*\cos(q3)*\text{q2dot} ] \end{aligned}$$

$$\begin{aligned} &[ I_y3s*\text{q3ddot}-\cos(q3)*\text{q2dot}^2*I_x3s*\sin(q3)+ \\ &\sin(q3)*\text{q2dot}^2*I_z3s*\cos(q3) ] \\ &[ I_z3s*\cos(q3)*\text{q2ddot}- \\ &I_z3s*\text{q3dot}*\sin(q3)*\text{q2dot}+\text{q3dot}*I_x3s* \\ &\sin(q3)*\text{q2dot}-\sin(q3)*\text{q2dot}*I_y3s*\text{q3dot} ]. \end{aligned}$$

The motive generalized forces have the expressions:

$$\begin{aligned} \mathbf{Qm1} &= \\ &\sin(q2)*\cos(q3)*\text{fx4}+\sin(q2)*\sin(q3)*\text{fz4}+M3 \\ &*\text{q1ddot}-\sin(q2)*M3*\text{q2ddot}*l3- \\ &1/2*\sin(q2)*M3*l4*\text{q2ddot}+M2*\text{q1ddot}- \\ &1/2*\sin(q2)*M2*\text{q2ddot}*l3+\cos(q2)*\text{fy4}- \\ &\cos(q2)*M3*\text{q2dot}^2*l3- \\ &1/2*\cos(q2)*M3*l4*\text{q2dot}^2- \\ &1/2*\cos(q2)*M2*\text{q2dot}^2*l3+M1*\text{q1ddot} \\ \mathbf{Qm2} &= \\ &I_x3s*\text{q2ddot}- \\ &I_x3s*\text{q2ddot}*\cos(q3)^2+I_z2s*\text{q2ddot}- \\ &\sin(q3)*\text{nx4}+\cos(q3)*\text{nz4}- \\ &1/2*l4*M3*\sin(q2)*\text{q1ddot}+2*\sin(q3)*I_x3s*\text{q3} \\ &\text{dot}*\cos(q3)*\text{q2dot}- \\ &2*\sin(q3)*\text{q3dot}*I_z3s*\cos(q3)*\text{q2dot}- \\ &1/2*l3*M2*\sin(q2)*\text{q1ddot}+1/4*M2*\text{q2ddot}*l3^2 \\ &- \sin(q3)*l4*\text{fz4}+I_z3s*\cos(q3)*\text{q2ddot}- \\ &\cos(q3)*l4*\text{fx4}-l3*\cos(q3)*\text{fx4}- \\ &l3*\sin(q3)*\text{fz4}+l4*M3*\text{q2ddot}*l3- \\ &l3*M3*\sin(q2)*\text{q1ddot}+1/4*M3*l4^2*\text{q2ddot}+l3 \\ &^2*M3*\text{q2ddot} \\ \mathbf{Qm3} &= \\ &\text{ny4}+I_y3s*\text{q3ddot}- \\ &\cos(q3)*\text{q2dot}^2*I_x3s*\sin(q3)+\sin(q3)*\text{q2dot} \\ &^2*I_z3s*\cos(q3). \end{aligned}$$

In the above listing, the link forces and the moments of the link forces were intentionally omitted, due to the limited space of this paper. Anyway, parts of their expressions give the generalized motive forces, formally written as:

$$\begin{aligned} Q_m^1 &= {}^4f_x s q_2 c q_3 + {}^4f_y c q_2 + {}^4f_z s q_2 s q_3 + \ddot{q}_1 \cdot (M_1 + \\ &+ M_2 + M_3) - \ddot{q}_2 \cdot \left[ M_2 \frac{l_3}{2} + M_3 \left( l_3 + \frac{l_4}{2} \right) \right] \cdot s q_2 - \\ &- \ddot{q}_2^2 \cdot \left[ M_2 \frac{l_3}{2} + M_3 \left( l_3 + \frac{l_4}{2} \right) \right] \cdot c q_2 \end{aligned} \quad (1)$$

$$\begin{aligned} Q_m^2 &= -{}^4n_x s q_3 + {}^4n_z c q_3 - {}^4f_x (l_3 + l_4) \cdot c q_3 - \\ &- {}^4f_z (l_3 + l_4) \cdot s q_3 - \ddot{q}_1 \cdot \left[ M_2 \frac{l_3}{2} + M_3 \left( l_3 + \frac{l_4}{2} \right) \right] \cdot s q_2 + \\ &+ \ddot{q}_2 \cdot \left[ M_3 \left( l_3 + \frac{l_4}{2} \right)^2 + M_2 \frac{l_3^2}{4} + {}^3J_x^* \cdot s^2 q_3 + {}^3J_z^* \cdot \right. \\ &\left. \cdot c^2 q_3 + {}^2J_z^* \right] + 2\dot{q}_2 \dot{q}_3 ({}^3J_x^* - {}^3J_z^*) \cdot s q_3 c q_3 \end{aligned} \quad (2)$$

$$Q_m^3 = {}^4n_y + {}^3J_y^* \cdot \ddot{q}_3 + ({}^3J_z^* - {}^3J_x^*) \cdot \dot{q}_2^2 \cdot s q_3 c q_3 \quad (3)$$

They express the differential equations of the dynamic model of the TRR robot.

## 5. CONCLUSION

*Robot\_Symbolic* is a generalized program that implements several algorithms for modelling a robot with up to six degrees of freedom, regardless of their complexity, using symbolic computation.

This program is useful in both research in the field of Robotics, and teaching activities, in disciplines specific to industrial robots.

The module presented in this paper, *Robot\_Dynamics*, allows the user to generate the symbolic equations of the dynamic model of the robot, further used in the simulation of the dynamic behavior of the considered robot.

## 6. REFERENCES

1. Deteşan, O.A., *Cercetări privind modelarea, simularea și construcția miniroboților*, Ph.D. Thesis, U.T.C.N., Cluj-Napoca, 2007.
2. Deteşan, O.A., *The Definition of the Robot Mechanical Structure Using the Symbolic Computation in Matlab*. In: Acta Technica Napocensis, No.51, vol. IV, 2008, ISSN 1221-5872.
3. Deteşan, O.A., *The Implementation of the TRR Robots Modelling Using the Symbolic Computation in Matlab*. In: Acta Technica Napocensis, No.51, vol. IV, 2008, ISSN 1221-5872.
4. Deteşan, O.A., Ispas, Vrg., *The Kinematic Modelling of the Robot Mechanical Structure Using the Symbolic Computation in Matlab*, Annals of DAAAM for 2009 & Proceedings of 20<sup>th</sup> DAAAM International Symposium "Intelligent Manufacturing & Automation: Theory, Practice & Education", Viena, 2009, ISSN 1726-9679.
5. Deteşan, O., Bugnar, F., *Using the Symbolic Computation in MATLAB for Determining the Geometric Model of Serial Robots*, Acta Technica Napocensis, nr. 55, vol. III, p. 683-688, Cluj-Napoca, 2012, ISSN 1221-5872.
6. Ispas, V., *Manipulatoare și Roboți Industriali*. București. Editura Didactică și Pedagogică, 2004, ISBN 973-30-1349-8.
7. Negrean, I., Vușcan, I., Haiduc, N., *Robotics. Kinematic and Dynamic Modelling*, București, Editura Didactică și Pedagogică, 1998, ISBN 973-30-5309-8.
8. Negrean, I., *Kinematics and Dynamics of Robots. Modelling, Experiment, Accuracy*, Editura Didactică și Pedagogică, 1999, ISBN 973-30-9313-0.
9. Negrean, I., Duca, A., Negrean, C., Kacso, K., *Mecanică avansată în robotică*, Cluj-Napoca, UT Press, 2008, ISBN 978-973-662-420-9.
10. Wilson, H.B., Turcotte, L.H., Halpern, D.: *Advanced Mathematics and Mechanics Applications Using MATLAB*, Boca Raton, New York, Washington D.C., Chapman & Hall/CRC, 2003, ISBN 1-58488-262-X.
11. \*\*\* *MATLAB, The Language of Technical Computing – Function Reference*, The MathWorks Inc., USA, 2004.

### MODELAREA DINAMICĂ A STRUCTURII MECANICE A ROBOTULUI, UTILIZÂND CALCULUL SIMBOLIC ÎN MATLAB

**Rezumat:** O parte semnificativă a modelării și simulării unui robot o constituie generarea modelului geometric, cinematic și dinamic al structurii mecanice a acestuia. Modulul software prezentat în această lucrare permite modelarea structurii mecanice generalizate a robotului și generarea automată a ecuațiilor modelului dinamic al robotului considerat, exprimând forțele generalizate motoare din cuplele robotului. Parte din aplicația generalizată *Robot\_Symbolic*, după *Robot\_Definition*, *Robot\_Geometry* și *Robot\_Kinematics*, *Robot\_Dynamics* este ultimul pas care trebuie urmat în vederea modelării structurii mecanice a unui robot serial. Aplicația este scrisă în MATLAB, exploatează capacitățile de calcul simbolic ale acestuia, implementând formalismul Newton-Euler.

**Ovidiu-Aurelian DETEȘAN**, Ph.D., Assoc. Prof., Eng., Technical University of Cluj-Napoca, Department of Mechanical System Engineering, Ovidiu.Detesan@mep.utcluj.ro, +40-64-401667, Cluj-Napoca, Romania.