TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics and Mechanics
Vol. 55, Issue III, 2012

USING THE SYMBOLIC COMPUTATION IN MATLAB FOR DETERMINING THE GEOMETRIC MODEL OF SERIAL ROBOTS

Ovidiu-Aurelian DETEŞAN, Florin BUGNAR

Abstract: *The software module presented in this paper allows the modelling of the generalized mechanical structure of the robot and the automatic generation of the geometrical model equations of the considered robot, i.e. the position vector and the orientation matrix of the frame attached to the end-effector, with respect to the fixed frame {0}. As part of the generalized application Robot_symbolic, after Robot_definition, Robot_geometry is the second step to be taken in consideration when the modelling of a robotic structure is intended. The application is written in Matlab and it exploits its symbolic computation facilities.* ***Key words:*** *serial robot, geometric modelling, symbolic computation.*

1. INTRODUCTION

An important step in robots modelling is the geometric modelling of the mechanical structure of the robot to be analyzed. The symbolic computation approach deals with symbolic objects, the input data being symbolic and, sometimes, numeric, while the output data are algebraic expressions. The key advantage of symbolic computation in the robot geometrical modelling resides in directly generating the symbolic form of the rotation matrices and the position vectors defining the location (position and orientation) of the frame attached to the end-effecter, with respect to the base frame {0}. The obtained data are furthermore passed to the modules of kinematic and dynamic modelling, being useful to generate the corresponding mechanical models.

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 [5], [6]: 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 two symbolic toolboxes: the *Symbolic Math Toolbox*, including over 100 MATLAB functions which access the Maple core and the *Extended Symbolic Math Toolbox* which increases its symbolic functionality.

3. THE *Robot_Symbolic* PACKAGE

The *TRR_Symbolic* application [1] was a starting point for designing a generalized application for symbolic modeling 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 introduced data could be saved and reused as necessary.

The generalized modeling 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.

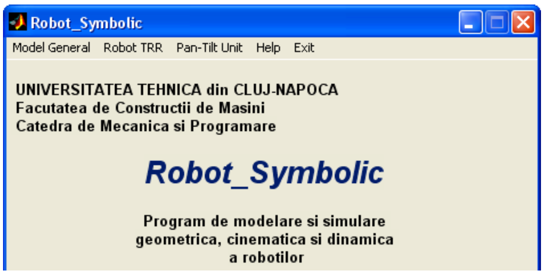

Fig. 1. The graphics user interface of Robot_Symbolic application

Table 1

The structure of Robot_Symbolic application

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

4. THE *Robot_Geometry* MODULE

The *Robot_Geometry* module, comprised in the file *Robot_geometry.m*, uses data generated at the runtime of the robot mechanical structure definition, *Robot_definition,* with the purpose of determining the symbolic geometric model of the robot to be analyzed, using the rotation matrices method [3,4]. The notations used in the program, compared to those used in [4], are presented in the Table 2.

Table 2

Notations in Robot_Geometry

Notation in [4]	Notation in the program	Notation in [4]	Notation in the program
q_1 ...	q1...	${}^{0}_{1}[R]$ ...	R10 ...
l_0 ...	l0...	$[I_3]$	eye(3)
α_z	alphaz	${}^{0}\bar{r}_1$ ...	r10 ...
β_x	betax	$\bar{p}_{10}$ ...	p10 ...
γ_z	gammaz	$\bar{p}_1$ ...	p1 ...

4.1 The Principle of Operation

Let us consider a robot with n degrees of freedom, consisting in n 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:

```
<-----GEOMETRIC MODEL----->
Robot name: <name>
The robot <name> was not structurally
defined. To define, press <D>, to
exit, press <X>:
```

The program checks if the robot structure with the name <*name*> was defined. If so, the data representing the mechanical structure will be loaded (the file *name_intro.mat*, generated by the module *Robot_Definition* [2]). If not, the user is advised either to go to the definition module or to exit the current module.

By iterative computations, the absolute rotation matrices (with respect to the frame {0}) of the frames attached to each joint i, (i = 1 ÷ n) and the absolute translation vectors corresponding to the centers of each joint, are determined, using the following code sequence:

```
disp('<-Absolute rotation...'
'matrices->');
R10
for i=2:dof+1
eval(['R',num2str(i),'0=R',...
num2str(i-1), '0*R',... num2str(i),
num2str(i-1)]);
end
disp('<--Relative transl.'...
'vectors--->');
p10=r10
for i=2:dof+1
eval(['p',num2str(i),...
num2str(i-1),'=simple(R',...
num2str(i-1),'0*r',...
num2str(i), num2str(i-1),')']);
end
     disp('<--Absolute transl.'...
```

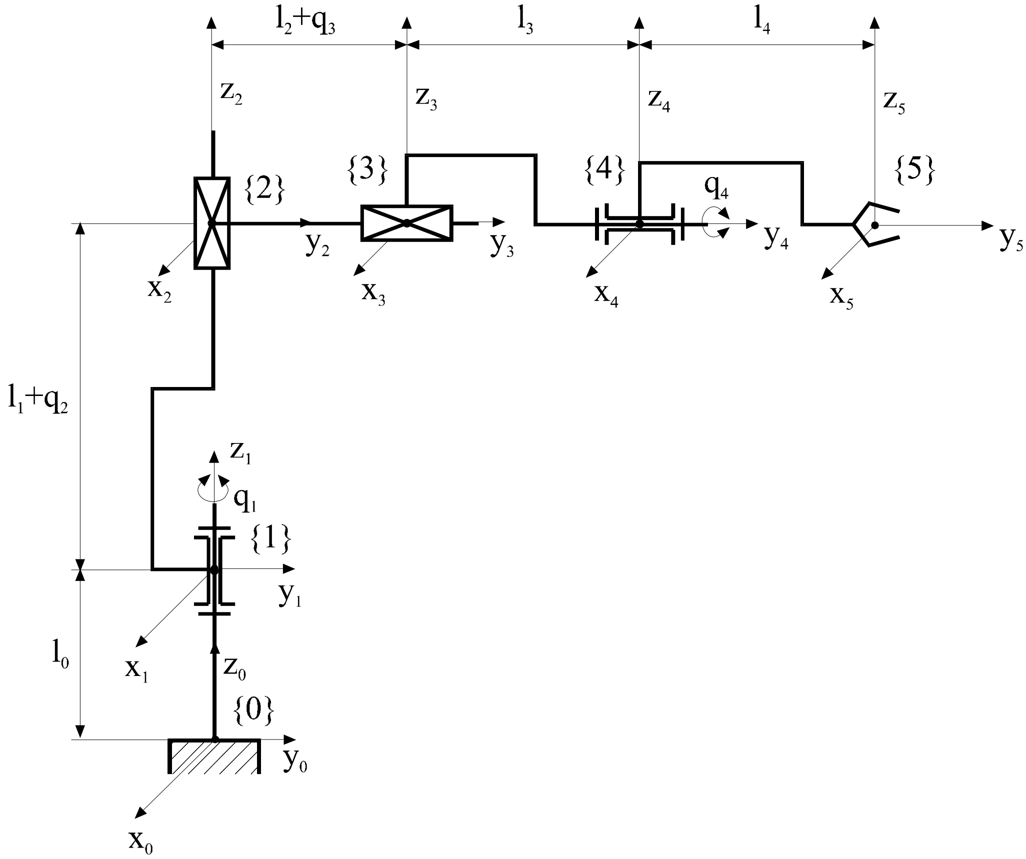

Fig. 2. The kinematic diagram of the RTTR robot

```
'vectors--->');
p1=p10
for i=2:dof+1
eval(['p',num2str(i),...
'=simple(p',num2str(i-1),...
'+p',num2str(i),...
num2str(i-1),')']);
end
```

The symbolic geometric model is defined by the equations of the position and orientation of the end-effector with respect to the fixed frame {0}.

Additionally, the Euler's set of angles *alphaz-betax-gammaz,* expressing the orientation of the end-effector, are determined, resulting by identifying the elements of the rotation matrix computed by the program, with those of the generalized rotation matrix [4], as suggested by the code sequence:

```
disp('<----- Euler's angles'...
'----->')
alphaz=eval(['simple(atan(R',...
num2str(dof),'0(1,3),-R',...
num2str(dof),'0(2,3)))'])
betax=eval(['simple(atan(R',...
num2str(dof),...
'0(1,3)*s(alphaz)-R',...
num2str(dof),...
'0(2,3)*c(alphaz),R',...
num2str(dof),'0(3,3)))'])
gammaz=eval(['simple(atan(',...'-
R',num2str(dof),'0(1,2)',...
'*c(alphaz)-R',num2str(dof),...
'0(2,2)*s(alphaz),R',...
num2str(dof),'0(1,1)*c',...
'(alphaz)+R',num2str(dof),...
'0(2,1)*s(alphaz)))'])
```

The symbolic data computed by this module are saved in the output file *name_geom.mat*, wherefrom they can be used for the subsequent kinematic and dynamic model determination.

4.2. Use Cases of *Robot_Geometry*

4.2.1. RTTR Mechanical Structure

An example of determining the symbolic equations of the geometric model using the above described program is given as follows, for the mechanical structure of the RTTR robot defined in [2], whose kinematic diagram is shown in fig. 2.

```
<-------GEOMETRIC MODEL------>
Robot name: rttr
<-Absolute rotation matrices->
```

```
R10 =
[  cos(q1), -sin(q1),        0]
[  sin(q1),  cos(q1),        0]
[        0,        0,        1]
R20 =
[  cos(q1), -sin(q1),        0]
[  sin(q1),  cos(q1),        0]
[        0,        0,        1]
R30 =
[  cos(q1), -sin(q1),        0]
[  sin(q1),  cos(q1),        0]
[        0,        0,        1]
R40 =
[ cos(q1)*cos(q4),  -sin(q1),
cos(q1)*sin(q4)]
[ sin(q1)*cos(q4),   cos(q1),
sin(q1)*sin(q4)]
[   -sin(q4),   0,   cos(q4)]
R50 =
[ cos(q1)*cos(q4),  -sin(q1),
cos(q1)*sin(q4)]
 [ sin(q1)*cos(q4),  cos(q1),
sin(q1)*sin(q4)]
[   -sin(q4),   0,   cos(q4)]
<--Relative transl. vectors--->
p10 =
[  0]
[  0]
[ l0]
p21 =
[     0]
[     0]
[ q2+l1]
p32 =
[ -sin(q1)*(q3+l2)]
[  cos(q1)*(q3+l2)]
[                0]
p43 =
[ -sin(q1)*l3]
[  cos(q1)*l3]
[           0]
p54 =
[ -sin(q1)*l4]
[  cos(q1)*l4]
[           0]
<--Absolute transl. vectors--->
p1 =
[  0]
[  0]
[ l0]
p2 =
[        0]
[        0]
[ l0+q2+l1]
p3 =
[ -sin(q1)*(q3+l2)]
[  cos(q1)*(q3+l2)]
[         l0+q2+l1]
p4 =
[ -sin(q1)*(q3+l2+l3)]
[  cos(q1)*(q3+l2+l3)]
[            l0+q2+l1]
p5 =
[ -sin(q1)*(q3+l2+l3+l4)]
[  cos(q1)*(q3+l2+l3+l4)]
[               l0+q2+l1]
<----- Euler's angles ----->
alphaz =
atan(cos(q1)*sin(q4),-sin(q1)*sin(q4))
betax =
atan(csgn(sin(q4))*sin(q4),cos(q4))
gammaz =
-1/2*csgn(sin(q4))*pi
Saving   data...   into   the   file
RTTR_geom.mat
<-END OF RTTR GEOMETRIC MODEL->
```

The following important results are: the orientation of the frame {5} axes with respect to the frame {0}, expressed by (1), the position vector of the frame {5} origin with respect to the frame {0} and the Euler's angles, representing the independent set of parameters of orientation (3).

$$ {}_5^0[R] = \begin{bmatrix} cq_1 \cdot cq_4 & -sq_1 & cq_1 \cdot sq_4 \\ sq_1 \cdot cq_4 & cq_1 & sq_1 \cdot sq_4 \\ -sq_4 & 0 & cq_4 \end{bmatrix} \qquad (1) $$

$$ \overline{p}_5 = \begin{bmatrix} -(l_2 + l_3 + l_4 + q_3) sq_1 \\ (l_2 + l_3 + l_4 + q_3) cq_1 \\ l_0 + l_1 + q_2 \end{bmatrix} \qquad (2) $$

$$ \begin{bmatrix} \alpha_z & \beta_x & \gamma_z \end{bmatrix}^T = \begin{bmatrix} \frac{\pi}{2} + q_1 & q_4 & -\frac{\pi}{2} \end{bmatrix}^T \qquad (3) $$

Thus, the symbolic equations of the geometric model of the RTTR robot is expressed either by the set of formulae (1-2) or by the set (2-3). They represent the position and orientation of the frame {5} attached to the end-effectors, with respect to the fixed frame {0}.

4.2.2. RTTRR Mechanical Structure

The RTTRR robot, whose kinematic diagram is shown in fig. 3, will be modelled geometrically using the same module *Robot_Geometry* of the application *Robot_Symbolic*. Considering its previous definition of the mechanical structure, using the module *Robot_Definition*, the file *rttrr_intro.mat* is loaded at startup. The equations of the geometrical model, as generated by the modelling module, are expressed by the equations (4), (5) and (6).

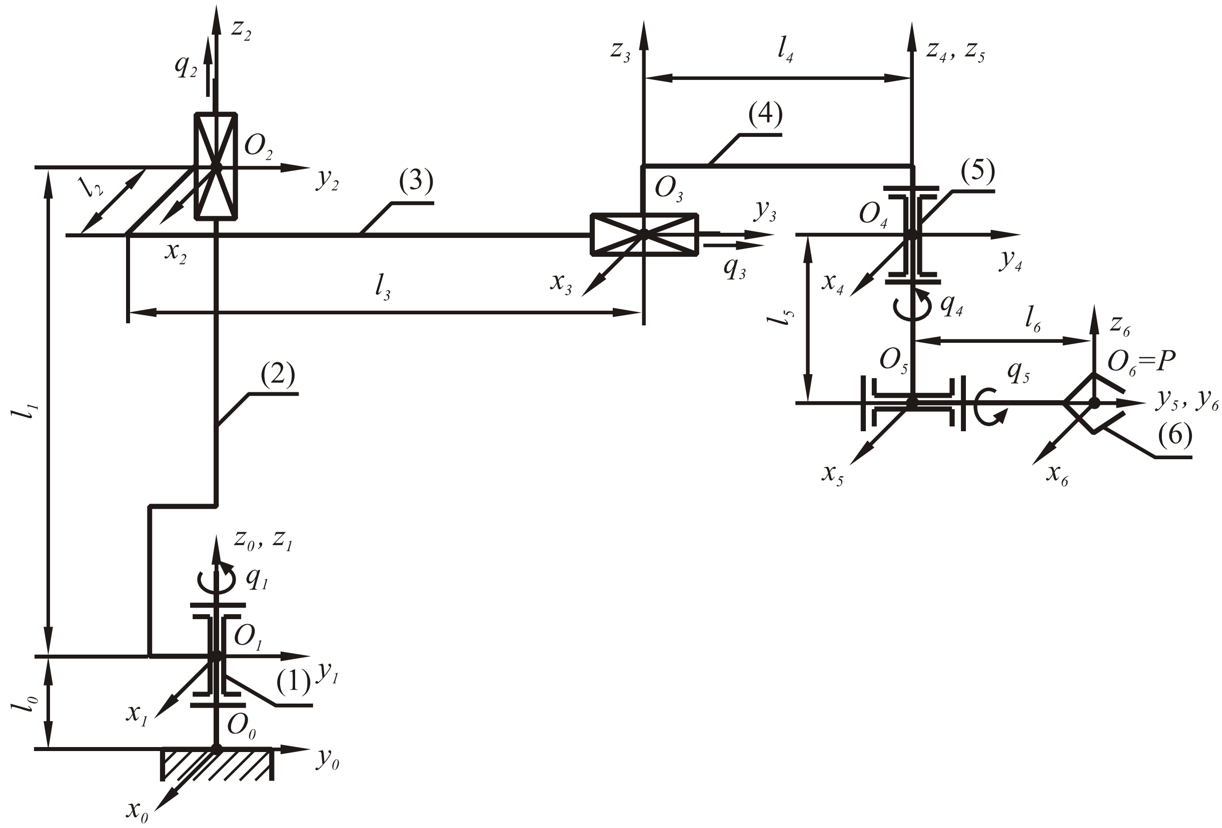

Fig. 3. The kinematic diagram of the RTTRR robot

$$
{}_6^0[R]=\begin{bmatrix} c(q_1+q_4)\cdot cq_5 & -s(q_1+q_4) & c(q_1+q_4)\cdot sq_5 \\ s(q_1+q_4)\cdot cq_5 & c(q_1+q_4) & s(q_1+q_4)\cdot sq_5 \\ -sq_5 & 0 & cq_5 \end{bmatrix}, \tag{4}
$$

$$
\overline{p}_6=\begin{bmatrix} cq_1\cdot l_2-sq_1\cdot(q_3+l_3)-sq_1\cdot l_4-s(q_1+q_4)\cdot l_6 \\ sq_1\cdot l_2+cq_1\cdot(q_3+l_3)+cq_1\cdot l_4+c(q_1+q_4)\cdot l_6 \\ l_0+q_2+l_1-l_5 \end{bmatrix} \tag{5}
$$

$$
\Psi=\begin{bmatrix} \alpha_z \\ \beta_x \\ \gamma_z \end{bmatrix}=\begin{bmatrix} \pi/2+q_1+q_4 \\ q_5 \\ -\pi/2 \end{bmatrix}. \tag{6}
$$

5. CONCLUSION

Robot_Symbolic is a generalized program that implements several algorithms for modeling 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 the industrial robots.

The module presented in this paper, *Robot_geometry,* allows the user to generate the symbolic equations of the robot, further used in the kinematics and dynamics modeling of the robot.

6. REFERENCES

1. Deteşan, O.A., *Cercetări privind modelarea, simularea şi construcţia mini-roboţilor,* 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, p. 65-68.
3. Ispas, V.: *Manipulatoare şi Roboţi Industriali.* Bucureşti. E.D.P., 2004.
4. Negrean, I., Vuşcan, I., Haiduc, N.: *Robotics. Kinematic and Dynamic Modelling*, Bucureşti, E.D.P., 1998.
5. Wilson, H.B., Turcotte, L.H., Halpern, D.: *Advanced Mathematics and Mecha-nics Applications Using MATLAB*, Boca Raton, New York, Washington D.C., Chapman & Hall/CRC, 2003.
6. *** *MATLAB, The Language of Technical Computing – Function Reference*, The MathWorks Inc., 2004.

FOLOSIREA CALCULULUI SIMBOLIC ÎN MATLAB PENTRU DETERMINAREA MODELULUI GEOMETRIC AL ROBOȚILOR SERIALI

Rezumat: Modulul software prezentat în această lucrare permite modelarea structurii mecanice generalizate a roboţilor seriali şi generarea automată a ecuaţiilor modelului geometric al robotului considerat, exprimând vectorul de poziţie şi matricea de orientare a sistemului de referinţă ataşat efectorului final, în raport cu sistemul fix {0}. Parte din aplicaţia generalizată *Robot_Symbolic,* după *Robot_Definition*, *Robot_Geometry* este al doilea pas care trebuie parcurs în vederea modelării structurii mecanice a unui robot serial. Aplicaţia este scrisă în MATLAB şi exploatează capacităţile de calcul simbolic ale acestuia.

Ovidiu-Aurelian Deteşan, 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.

Florin Bugnar, Ph.D. student, Eng., Technical University of Cluj-Napoca, Department of Mechanical System Engineering, bugnarf@hotmail.com, +40-64-401667, Cluj-Napoca, Romania.