

# GRAPHICAL ENVIRONMENT FOR SIMULATION OF THE CONTROL OF PLANAR REDUNDANT PARALLEL ROBOT

*Květoslav Belda*

Institute of Information Theory and Automation  
Academy of Sciences of the Czech Republic

**Abstract:** The most actual industrial robots and manipulators don't cope with increasing requirements on speed and accuracy. Therefore, new approaches of their construction are being found, in order to allow performance these requirements. Parallel robots seem to be one of the promising ways, how to solve this problem. The goal of this paper is to demonstrate the simulation of one such robot in environment MATLAB – SIMULINK, in which user-friendly graphical environment was programmed.

## 1. Introduction

Actual industrial robots and manipulators have a low operation speed. This is caused mainly by the limitation of the acceleration given by its construction. Parallel robots seem to be one of the promising ways, how to solve this problem. And moreover, they have several advantages over actual robots. All or almost all drives are located on the basic frame i.e. drives don't move with robot and they don't have any hold on the moving mass and stiffness. Truss construction of the robots leads to higher stiffness than in serial ones. It is advantageous for accurate machining and positioning.

On the other hand, the parallel robots have also several disadvantages. The robot's workspace is mostly smaller than for serial robots. Collisions of arms with platform are more probable than in the case of actual robots.

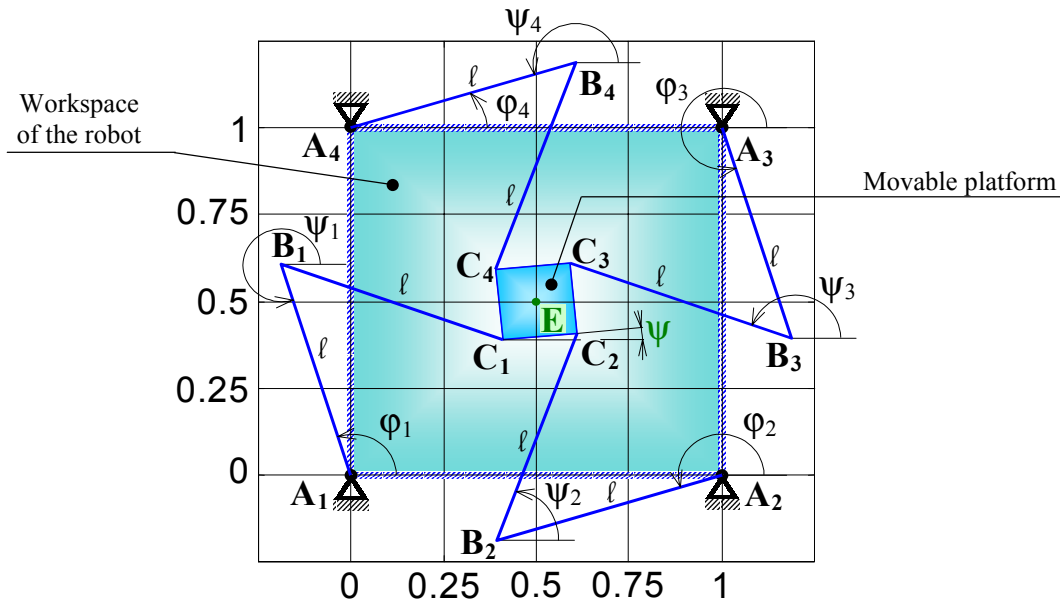


Fig. 1. Scheme of planar parallel robot with the most important geometrical description.

From mechanical point of view, the robot (fig.1), which will be presented here, has one drive and one pair of arms redundant. This redundancy helps to overcome some special positions, which ensue from construction of the robot. And moreover, it gives possibility to comply several control requirements.

From control point of view, the principal task of a robot is its movement along a planned trajectory. For this type of the robot, it is usually given in Cartesian coordinates, which are very efficacious for users.

After this, we can describe movement of the robot by nonlinear differential equation

$$\mathbf{R}^T \mathbf{M} \mathbf{R} \ddot{\mathbf{y}} + \mathbf{R}^T \mathbf{M} \mathbf{R} \dot{\mathbf{y}} - \mathbf{R}^T \mathbf{g} = \mathbf{R}^T \mathbf{T} \mathbf{u} \quad (1)$$

or nonlinear state formula

$$\begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{f}(\mathbf{X}) + \mathbf{g}(\mathbf{X}) \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{h} \mathbf{X}(t) \end{aligned} \quad (2)$$

The control should ensure the best possible compliance of the trajectory and at the same time effective cooperation between the necessary drives and one redundant drive. For need to design the control of such robot, graphical environment was constructed.

## 2. Description of the graphical environment

The graphical environment has two essential parts. The first is window for design of trajectory and the second is window for simulation of the control of the robot. These components were designed in order to simplify very often repeated planning of trajectory and testing of the control. Now they can be introduced.

### 2.1 Window for design of trajectory

As mentioned previously, general task of robots and manipulators is movement along a planned trajectory. This trajectory is known beforehand, because it is recording of planned movement of some tool or some gripper of the robot.

Design of the trajectory issues from kinematics' laws. These were programmed into ordinary MATLAB m-file. As was already stated, this process very often repeats. That is why graphical environment (GE, graphical user interface) was programmed.

This environment provides the following:

- Insertion of parameters of the trajectory.  
(key points, velocities, sampling period - time step of planned trajectory)
- Visualization of segments of planned trajectory.
- Graphical representation of kinematic characterizations of the trajectory.
- Saving of calculated data.

Note: The trajectory is saved in discrete form, i.e. time vector and appropriate vectors of positions, angles and their derivatives.

GE consists of three parts – graphical,  
– Executive,  
– visual.

Graphical part was created by using MATLAB GUI Layout Tool. Executive part, provided connection between graphical operating elements (buttons, edits, texts etc.) and visual area, executes a lot of operations.

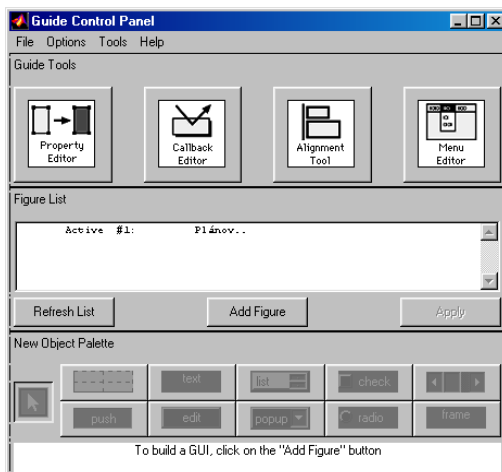


Fig. 2.: MATLAB GUI Layout Tool.

So its programming could be difficult when using Callback editor, therefore it was programmed separately from graphical part.

```

:
:
h91 = uicontrol('Parent',h80, ...
    'Units','points', ...
    'Callback','callba(91)', ...
    'ListboxTop',0, ...
    'Position',[120 15 90 15], ...
    'String','Confirm', ...
    'Tag','Pushbutton1',...
    'TooltipString','confirm values and closing the window');
:
:

```

Fig. 3.: Example of the part of graphical -file with calling execute m-file 'callba' with switch '91', (GUI – Pushbutton).

```

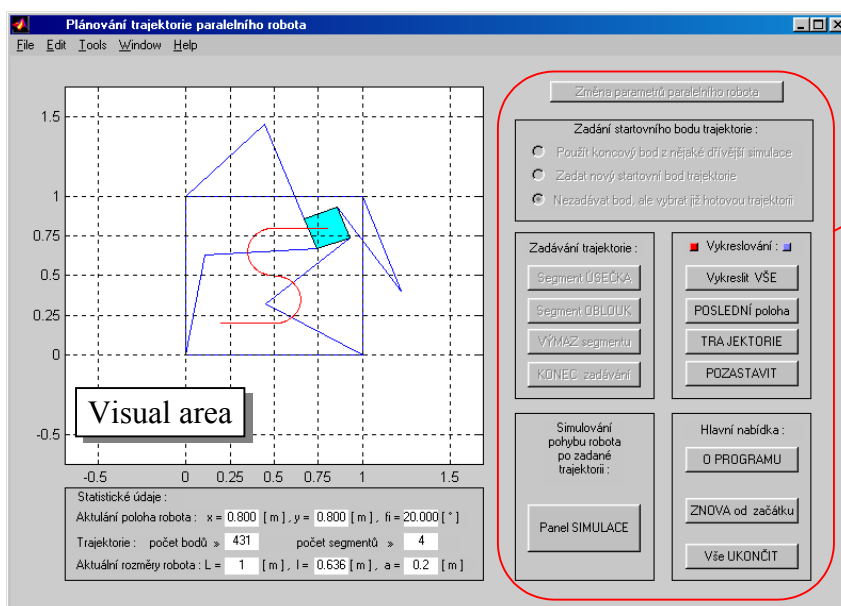
:
:
if h==91,
    global h86 h87 h88 h89 h90 xk yk fk vk wk;
    xk=str2num(get(h86,'String'));
    yk=str2num(get(h87,'String'));
    fk=str2num(get(h88,'String'));
    vk=str2num(get(h89,'String'));
    wk=str2num(get(h90,'String'));
    vzadra('u'); % call procedure vzadra with switch 'u'
end
:
:

```

Fig. 4.: Example of part of execute m-file 'calba' (saving of embedded values and calling procedure 'vzadra' with switch 'u').

Visual area was provided by SIMULINK block diagram. It calls visualize m-file, which uses usual commands (plot,line,fill with parameter 'EraseMode', 'background').

For setting of parameters into SIMULINK it is used command 'assignin('base','mh',mh);' which exports variable from function, where it is counted, to the MATLAB workspace. And SIMULINK block diagram is started by command 'sim('vsimrob');', where 'vsimrob' is a name of SIMULINK scheme. So it is running on the background.



Operating panel

Fig. 5.: The window for planning of the trajectory.

## 2.2 Window for simulation of the control of the robot

This environment provides the following:

- Choosing and loading of one planned trajectory.
- Setting of the parameters of the control.
- Visualization of the control process.
- Graphical representation of kinematic characterizations of the process.
- Saving of calculated data.

Its design is very similar as at 2.1. It uses the same principle: separate m-files, provided graphical or visual view, with callings on execute m-files. These callings are maintained across switching variable. In executive m-files, the variable is tested and from this role flows.

Actual simulation is maintained by MATLAB solver `ode45`, which solves the differential equations (`[ttt,xxx]=ode45('simdy1b',[TT(i-1),TT(i)],XXX(i-1,:))`), where 'simdy1b' is the name of the ODE file, '[TT(i-1),TT(i)]' is elementary time range and 'XXX(i-1,:)' is topical initial value. The integration is provided in parts of time steps of planned trajectory. It is very useful, because user can control the execution during all time range and can obtain some information about movement of the robot.

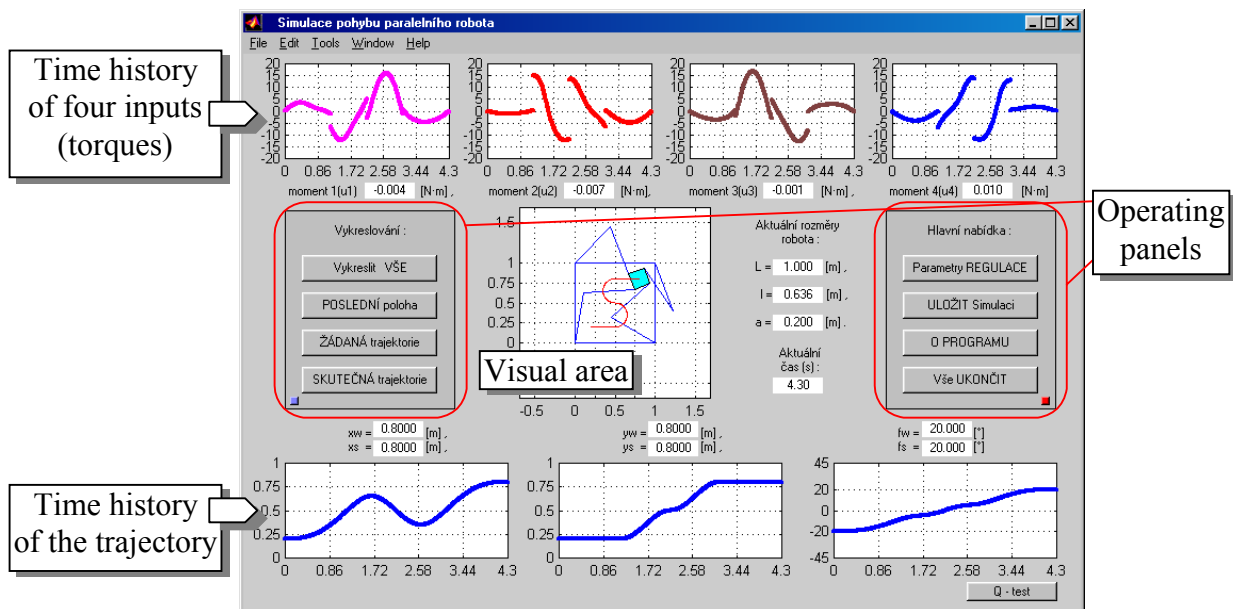


Fig. 6.: Window for simulation of the control of the robot.

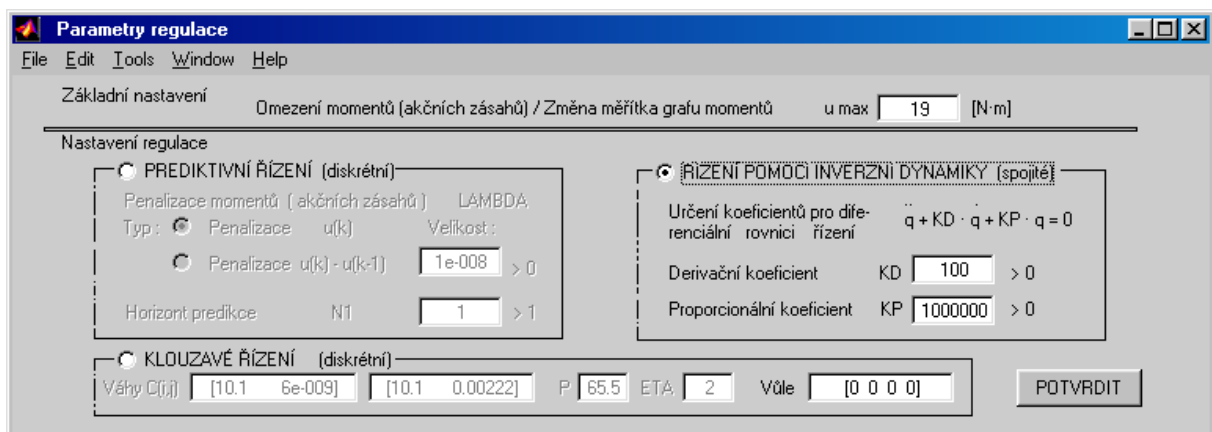


Fig. 7.: Window for choosing of the control and setting of its parameters.

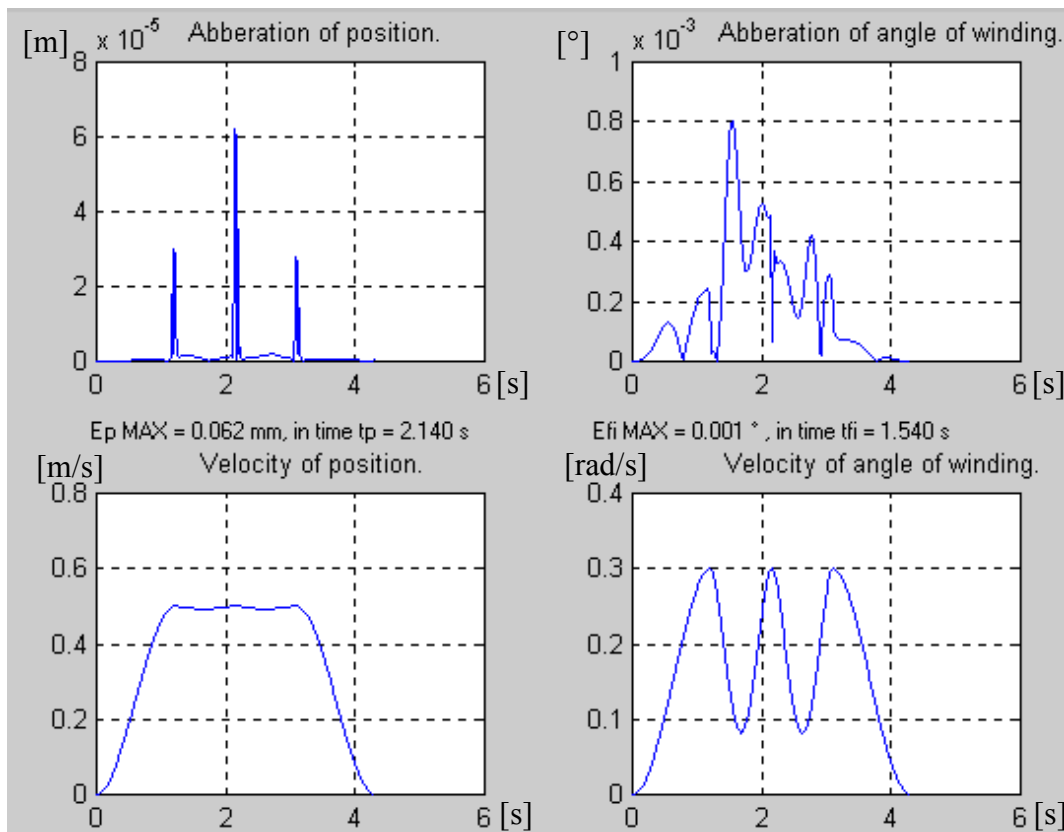


Fig. 8.: Demonstration of graphical representation of kinematic characterizations of the control process.

### 3. Result

The paper deals with graphical environment created under MATLAB. It shows, that MATLAB – SIMULINK environment is very useful not only for simulation, but it also enables to effectively make applications, which are comparable with classical windows one.

#### References

1. Online Manuals (in PDF): Building GUIs with MATLAB. The MathWorks, Inc. 1997.
2. ORDYS, A. – CLARKE, D.: A state - space description for GPC controllers. INT. J. Systems SCI., 1993, vol.24, no.9, pp. 1727 – 1744.
3. VALÁŠEK, M. – STEINBAUER, P.: Nonlinear control of multibody systems. Euromech 99 p.437.
4. SCIAVICCO, L. – SICILIANO, B.: Modeling and control of robot manipulators. McGraw-Hill Co., New York 1996.
5. ELMALI, H. – OLGAC, N.: Sliding mode control with perturbation estimation (SMCPE) : a new approach, INT. J. CONTROL 1992, vol.56, no.4, pp. 923 – 941.
6. BÖHM, J. – BELDA, K. – VALÁŠEK, M.: Design and simulation of predictive control of drives of planar redundant parallel robot. Process control 2000, p. 61 + CD ROM.

#### Acknowledgement

This research is supported by GA ČR 101/99/0729, 1999-2001: “Redundant parallel robot and its control”.

#### Contact address:

Ústav teorie informace a automatizace AV ČR  
Pod vodárenskou věží 4, 182 08 Praha 8 – Libeň  
E-mail: [belda@utia.cas.cz](mailto:belda@utia.cas.cz)