WEB CAMERA BASED CONTROL OF A MITSUBISHI MELFA ROBOTIC ARM WITH MATLAB TOOLS

R.Čermák

Západočeská univerzita v Plzni

Abstract

The contribution deals with trajectory planning of an industrial robotic arm Mitsubishi Melfa RV2-SD using MATLAB tools. The aim of this contribution is to inform about using of MATLAB toolboxes (both commercial and open-source) and non-expensive HW (webcam) to analyze a simple 2D scene, identify objects of interest in the scene, calculate joint coordinates of the robot (inverse kinematics) and after that visualize the robot posture on the display and move robot to desired position, pick up the object and move it given location (given place in a warehouse). Described problem was elaborated in the Wim Versleegers' final project [1] during his Erasmus stay at the University of West Bohemia in Pilsen in spring semester 2014 (the author of this contribution acted as a supervisor of the thesis).

1 Introduction

When teaching the principles of some problem in the university courses, usually it is not necessary to use professional (and expensive) hardware and software. The principles can be easily demonstrated with simple tools and the main accent is not given to efficiency and accuracy of the solution, but to simplicity of use and understanding. MATLAB and it's toolboxes, with the ability to connect low-cost hardware is an ideal way for that.

Described problem was elaborated in the Wim Versleegers' final project [1] during his Erasmus stay at the University of West Bohemia in Pilsen in spring semester 2014 (the author of this contribution acted as a supervisor of the thesis).

Commercial tools Image Acquisition Toolbox and Image Processing Toolbox were used for connection of the webcam to the MATLAB and to process the image. The goal was using of non-expensive hardware to capture the scene and as many open-source tools as possible for robot modeling and control. Therefore a webcam Logitech C170 was used as a visual sensor and Peter Corke's Robotic Toolbox for MATLAB [2, 3] and Mitsubishi Melfa Robot Control Toolbox for MATLAB [4] from CTU Prague were used to solve inverse kinematics, visualize the robot motion and to communicate with the Melfa robotic arm.

Mitsubishi software RT Toolbox [5] was used for offline verification of the results from the MATLAB calculations.

2 Hardware description

The hardware used in the task was very simple. We used robotic Mitsubishi Melfa RV2-SD, which is one of the smallest industrial arms available from Mitsubishi – see parameters of the robot in the next paragraph. The arm is fixed on aluminium frame – see Figure 1. On additional aluminium subframe above the workspace the webcam is located. For simplicity of image processing several conditions are set:

- the view direction is perpendicular to the workplane (2D scene is assumed),
- ideal light conditions are assumed,
- the third dimension of the objects to be picked is negligible (they don't cast any shades)
- the contrast between the object and the background is maximized (we use dark objects on the white background

The main aim was to complete the whole task from capturing the image to moving robot to desired position, with focus mainly on communication with the robot from MATLAB. Therefore the subtasks were made as simple as possible.

Figure 1: Picture of the workplace configuration

Importance of the low-cost solution was mentioned in the previous paragraph. Therefore the Logitech C170 webcam was used as a vision sensor. The webcam offers capturing pictures and video with size 1024x768 pixels (640x480 recommended). The webcam can be connected to MATLAB using Image Acquisition Toolbox – see later.

Mitsubishi Melfa RV2-SD is a small industrial arm offering small load of the tool on one side, but ability for fast movements and very precise positioning. Basic parameters of the arm can be found at the Mitsubishi or AutoCont Control Systems website.

The robotic arm can be programmed using Melfa Basic V programming language. Mitsubishi offers software RT Toolbox for online and offline programming and simulation. Connection between control unit and PC can be done using RS232, USB or TCP/IP. In our case the serial connection was used.

3 Software description

MATLAB and several toolboxes were used for capturing the image, procesing the image, solution of the inverse kinematics, simulation of the robot and visualization, and for communication with robot control unit.

Image Acquisition Toolbox was used for connection the webcam to MATLAB. The toolbox offers functions for easy connection of various hardware to MATLAB and Simulink. The image captured by the webcam was processed in the Image Processing Toolbox, which provides the user with a set of functions for video and image processing in the MATLAB environment. Both toolboxes are commercial toolboxes offered by Mathworks.

For solving of the inverse kinematics the Robotics Toolboox was used. Robotics Toolbox is one of the open-source tools created by Peter Corke [2, 3]. It offers several basic functions for robotics tasks, e.g. definition of model structure – bodies and joints, solution of forward and inverse kinematics and dynamics, coordinates transformations, etc.

In our case, we assume only low velocities. Therefore only the kinematics solution was done.

Very important part of the task was communication with the robot. We used a Mitsubishi Melfa Robot Control Toolbox for MATLAB developed at CTU Prague [4].

Offline simulation of the robot was done in MATLAB using functions from Robotics Toolbox and Mitsubishi Melfa RT Toolbox2 [5], which is a standard environment for the arm programming. See Fig.2

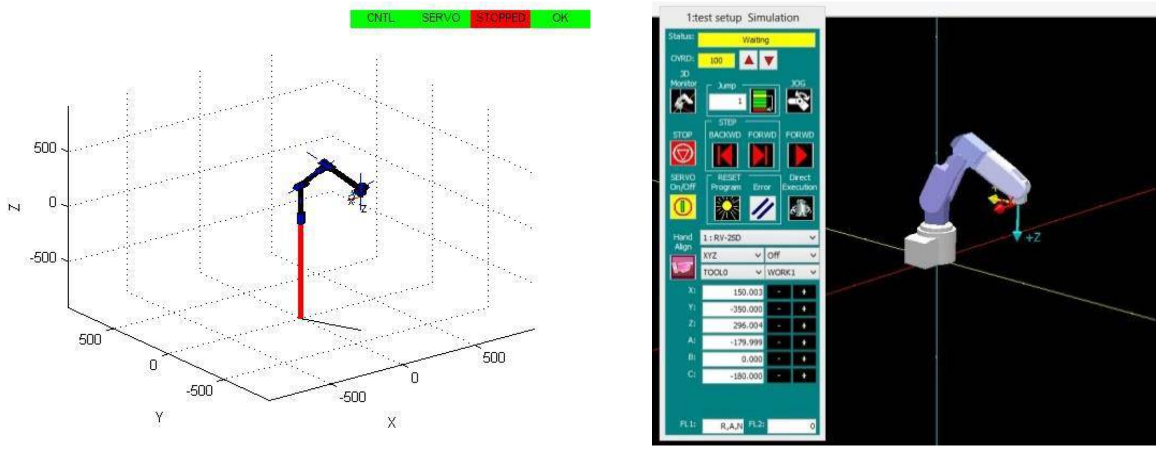

Figure 2: Visualization of the robot arm in MATLAB (with Robotic Toolbox) and in Mitsubishi RT Toolbox

4 Process description

For simplicity we analyze the whole scene, identify the object to be picked in the scene, generate trajectory, generate the code in Melfa Basic and send it to the control unit. Than we run the batch in the robot control unit, pick the object, move it into the storage and than repeat the whole process.

The overall process can be briefly described by following steps:

Capturing and processing the image (using Image Acquisition and Image Processing Tbx.)

- cropping the image
- conversion to B&W image format
- thresholding of the image, image corrections – filling gaps, closing objects
- objects identification and sorting (circular, angular)
- finding centroids of the identified objects

Robotic arm trajectory planning (using Robotics Toolbox)

- finding trajectories of the arm from start point to the object and from the object to the storage
- offline visualization

Communication with the robot (using Mitsubishi Melfa Tbx.)

- generation of code in MELFA Basic (batch processing of the whole scene)
- opening of the communication with robot via serial line
- sending the code to robot, saving the program code in the control unit
- running the program in the robot control unit
- closing the communication

Communication with the robotic arm is done by serial line. At the beginning the communication is open and confirmation as received. In the next command the program is saved in the control unit. The next command runs the program in the robot. When finished, the communication is closed. Using the serial line and batch processing of the commands is not very fast, but it is easy to be described, demonstrated and understood. Sending each command to the control unit can be a future improvement.

The communication protocol is described on Figure 3 [6]. Every command starts with ENQ. When acknowledged, the command for robot follows – see the command structure on Figure 4. The command ends with EOT.

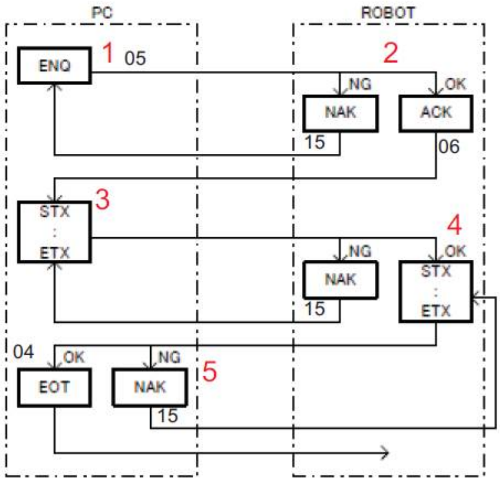

Figure 3: Communication between PC and control unit [6]

STX	ID	Sub ID	Len. H	Len. L	Robot ID	Comand	CKSUM H	CKSUM L	ETX
	D	0	0	A	R	1;1;SRVON	0	1	

Figure 4: Command structure [6]

An example of command sent to the control unit follow.

Opening of the communication command in ASCII and HEX format [6]:

ENQ	05
STX + D0ne12R1;1;OPEN=USERTOOL05 + ETX	0244303132523 13B313B4F50454E3D55534552544F4F4C303503
EOT	04

5 Conclusion

The main aim of the contribution was to inform about the possibilities of use MATLAB and it's tools (both commercial and open source) together with industrial robotic arm Mitsubishi Melfa in college courses focused on robotics fundamentals. The main advantage of this approach is in using MATLAB, which became a standard for technical computations (and not only in academia). Possibility of using numerous toolboxes available on the internet, easiness of including own code into the software, simple programming language and interactive user interface is ideal environment for experimenting with the software tools and their use in the education process.

Described problem was elaborated in the Wim Versleegers' final project [1] during his Erasmus stay at the University of West Bohemia in Pilsen in spring semester 2014 (the author of this contribution acted as a supervisor of the thesis).

References

[1] W.Versleegers. Programming a vertical articulated robotic arm with six degrees of freedom to recognize, pick up and sort geometric shapes with Matlab. Diploma work, University of West Bohemia in Pilsen, 2014

[2] P.I.Corke. Robotics, Vision & Control. Springer 2011, ISBN 978-3-642-20143-1

[3] P.I.Corke. A Robotics Toolbox for MATLAB. IEEE Robotics and Automation Magazine, Volume 3(1), March 1996, pp. 24-32

[4] Mitsubishi Melfa Robot Control Toolbox pro Matlab. Czech Technical University in Prague, https://cw.felk.cvut.cz/wiki/help/common/robot_mitsubishimelfa_toolbox

[5] RT Toolbox2/ RT Toolbox2 mini. Users Guide. Mitsubishi Electric Corporation

[6] Komunikace RV232 – Robot RV-2SD. AutoCont Control Systems

---

Roman Čermák
Západočeská univerzita v Plzni, Fakulta strojní, Univerzitní 8, 30614 Plzeň, Czech Republic
email: rcermak@kks.zcu.cz