Robot Control integrating LabVIEW and Matlab

V.S. Balashov, D.D. Skatova, S.J. Choi
Korea University of Technology and Education Department of Mechanical Engineering
Balashov.Vadim@gmail.com

Abstract
This study shows possibility of Matlab and LabVIEW integrationfor controlling of robot's manipulator. Examined approach can be used for control of complex system with intelligent control capability. Instance of Robotic System controller is described in details. Structure of control system is divided into three parts Virtual Instrument of LabVIEW, MatLab Script for solving and Matlab Simulink model for visualizing, which are explained separately. In addition, used neglects are explained and founded.

Introduction

The main task of this study is presentation of integration of two software products: National Instruments LabVIEW and Mathworks MatLab for implementation of some control system. Examined problem is controlling of robot system. Main condition is that Control system should be implemented by LabVIEW, but calculations should be produced by Matlab. Compatibility of MatLab and LabVIEW is widely used for controlling and simulating various systems, i.e., controlling of 4-wheel chair for disabled people [1], for Distance Learning [2] or for monitoring and control [3].

Problem statement

Manipulator of Robot System is classic PUMA (programmable universal manipulator for assembly) configuration of manipulator. Figure 1 presents kinematics of PUMA manipulator.

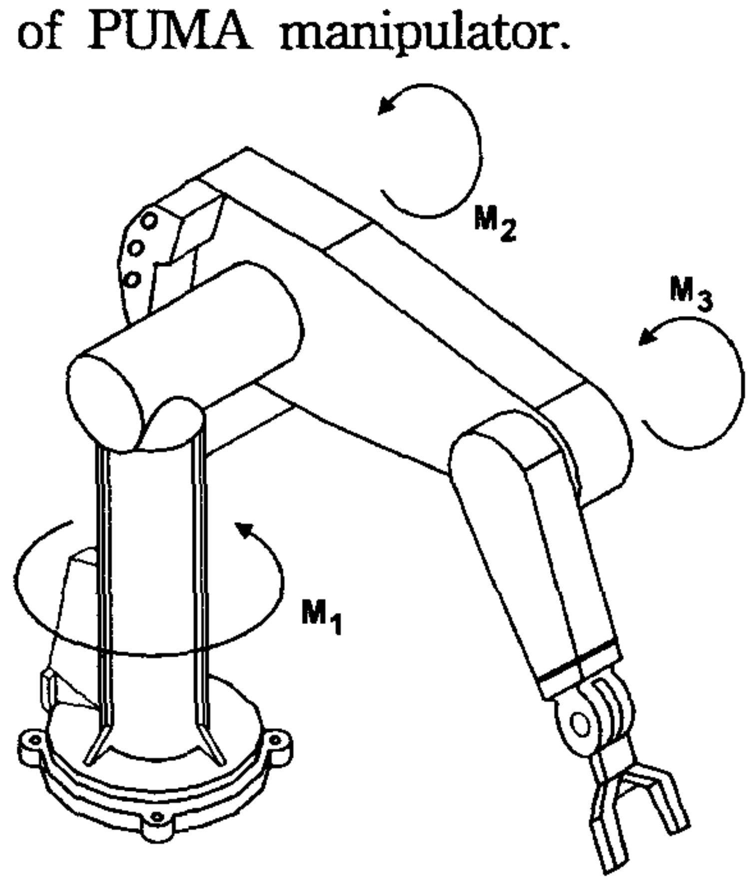

Figure 1. Kinematics of PUMA robot

Geometry schema of PUMA robot is very simple. It presented at figure 2.

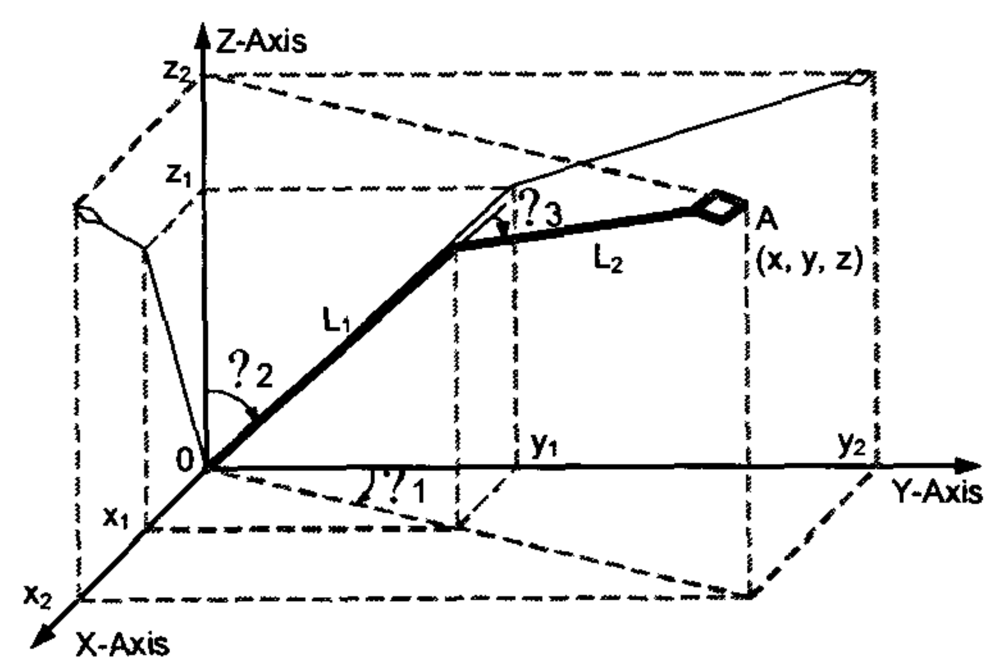

Figure 2. Geometry of PUMA manipulator

There are two main problems of robot controlling: direct problem and reverse problem.

Direct problem

Simplified expression of direct problem of manipulator's kinematics in this case is "Where actuator will be, if certain angles are applied?"

Direct problem is very simple - exact equations can be given by geometry. In case of PUMA manipulator following equations present solution of direct problem.

$$x = [l_2 \cdot \sin\alpha_2 + l_3 \cdot \sin(\alpha_2 + \alpha_3)] \cdot \sin\alpha_1$$
$$y = [l_2 \cdot \sin\alpha_2 + l_3 \cdot \sin(\alpha_2 + \alpha_3)] \cdot \cos\alpha_1$$
$$z = l_1 + l_2 \cdot \cos\alpha_2 + l_3 \cdot \cos(\alpha_2 + \alpha_3)$$

Any angles can be applied for these equations.

Inverse problem

Simplified expression of inverse problem of manipulator's kinematics in this case is "Which angles should be applied to move actuatorinto required position?"

Reverse problem is more complicated - equations should be given by equations of direct problem. Following equations present general form of reverse problem equations.

$$\alpha_1 = F_1(x, y, z)$$
$$\alpha_2 = F_2(x, y, z)$$
$$\alpha_3 = F_3(x, y, z)$$

Reverse problem equations have four different solutions. Control system should decide, which solution can be applied. Also not any coordinates of actuator destination can be applied.

Description of program

Why LabVIEW and Matlab are used?

LabVIEW is very powerful software for control of any kind of hardware. It can provide Graphical User Interface for input data and viewing results. Using of LabVIEW is most efficient way for making control systems, because programmer has no need to mention of implementations any kind of routing such as user's interface routing (controls and indicators), communication routing, files routing and so on. Main goal of programmer is implementation of control algorithm.

MatLab is very powerful software for any kind of calculation and simulation. Most of usefulequations, which can be used for engineering purpose, can be solved by MatLab. Also in most of cases Matlab can solve symbolic equations.

Integration of these software products is synergetic approach, which allows us to make powerful control system for robot's manipulator.

Implementation

Implementation of Robot's Controller consists of three parts:

- Virtual Instrument of LabVIEW: user's control panel, coordination and integration.
- MatLab Script: solving of equations.
- MatLab Simulink model: animated simulation of manipulator's movement.

Following sections describe all parts in detail.

Virtual Instrument of LabVIEW

Virtual Instrument of LabVIEW provides user's control panel and some solutions for managing of MatLab Scripts.

User's control panel of robot system in LabVIEW allows setting coordinates of actuator's destination. Screenshot of control panel is presented at figure 3.

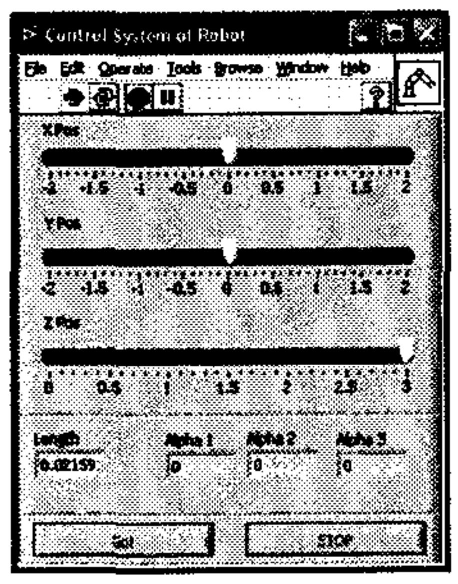

Figure 3. User's Control Panel

To operate manipulator we should set coordinates of required point and press Go! Button. After several seconds, manipulator will move to that point.

Matlab Script

Matlab Script is special box of LabVIEW. It connects to MatLab server and executes batch of commands. Figure 4 presents using of this box.

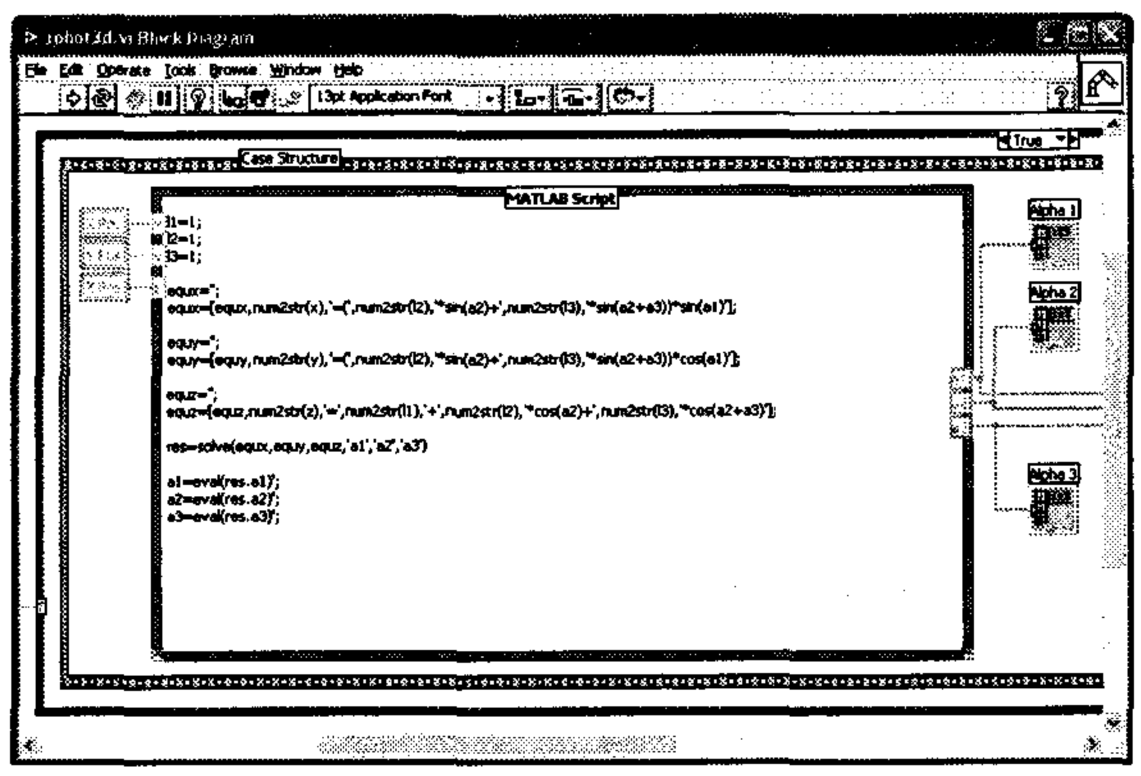

Figure 4. MatLab Script box in LabVIEW

Matlab Script consists of two parts.

First part is making of direct problem equations in special form. Angles in these equations are symbolic, but unknowns (coordinates) are numbers (because originally, these equations of reverse problem). Following Matlab commands are presented in first part of script.

```
equx = [equx,num2str(x), '=(',
  num2str(l2), '*sin(a2)+',
  num2str(l3),
  '*sin(a2+a3))*sin(a1)'];

equy = [equy,num2str(y), '=(',
  num2str(l2), '*sin(a2)+',
  num2str(l3),
  '*sin(a2+a3))*cos(a1)'];

equz = [equz, num2str(z), '=',
  num2str(l1), '+', num2str(l2),
  '*cos(a2)+', num2str(l3),
  '*cos(a2+a3)'];
```

Second part is solving of equations. MatLab provides special function for this purpose. This function uses symbolic equations and symbolic unknowns. After several seconds, it makes solution in symbolic or numeric form. Result should be divided to angles. Following Matlab commands are presented in second part of script.

```
res = solve(equx, equy, equz, 'a1',
      'a2', 'a3')
a1 = eval(res.a1)';
a2 = eval(res.a2)';
a3 = eval(res.a3)';
```

Obtained angles are sending back to LabVIEW. As described before, reverse problem equations have four different solutions. LabVIEW Virtual Instrument making analyze and decides between variants. Angles are data for control system of Robot System, which is simulated by Simulink.

MatLab Simulink model

SimMechanics is library of Simulink, which was used for simulation of manipulator. SimMechanics is very powerful simulation tool. It can simulate mechanical motions by considering of dynamics, but in this study, only kinematics of manipulator is used.

Model of manipulator consists of one ground (prop), three bodies (solid chains) and threejoints (rotary connections) between. Geometric and physical parameters of manipulator are not real, because they are not important in this study. Thus,lengths of all bodies are 1 meter (3.28 feet), and weights of them are 1 kg (2.20 lbs).

Special simple control sub-systems are used for every joint. These control sub-systems are not simulation of real control systems for rotary joints of manipulators. Main goal of these sub-systems is animated movement of actuator from initial point to final point.

Real geometrical parameters can be neglected, because all calculations in this Control System are symbolic, so calculations can be adapted for any real PUMA manipulator.

Real physical parameters can be neglected, because this study considers only kinematics.

Real properties of control system can be neglected, because described Control System isTop-Level (Command) control system, but providing of real motion is task of Low-Level (subordinate) control system. Thus, sub-control systems of manipulator can be simulated simply.

Figure 5 presents simulation model.

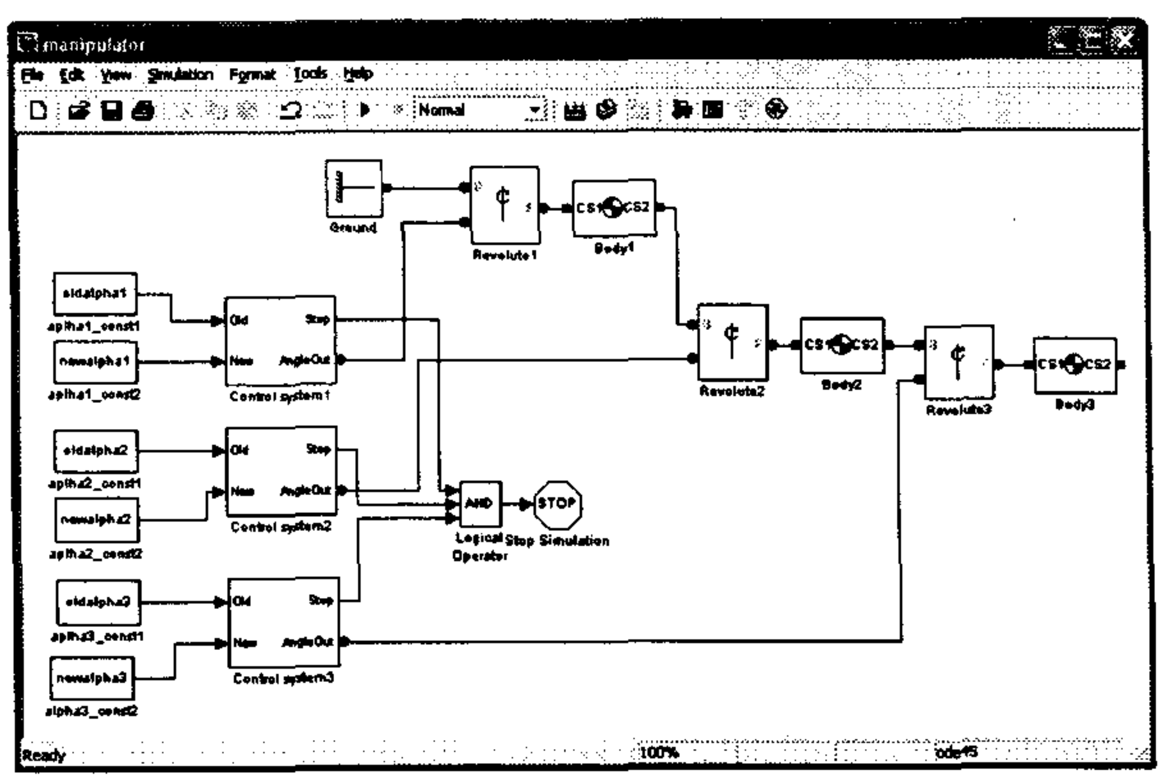

Figure 5. Simulation model of manipulator

Before running simulation, initial parameters should be applied. There are two kinds of parameters: initial and final angles for each joint.

Following figures present pictures of simulated manipulator.

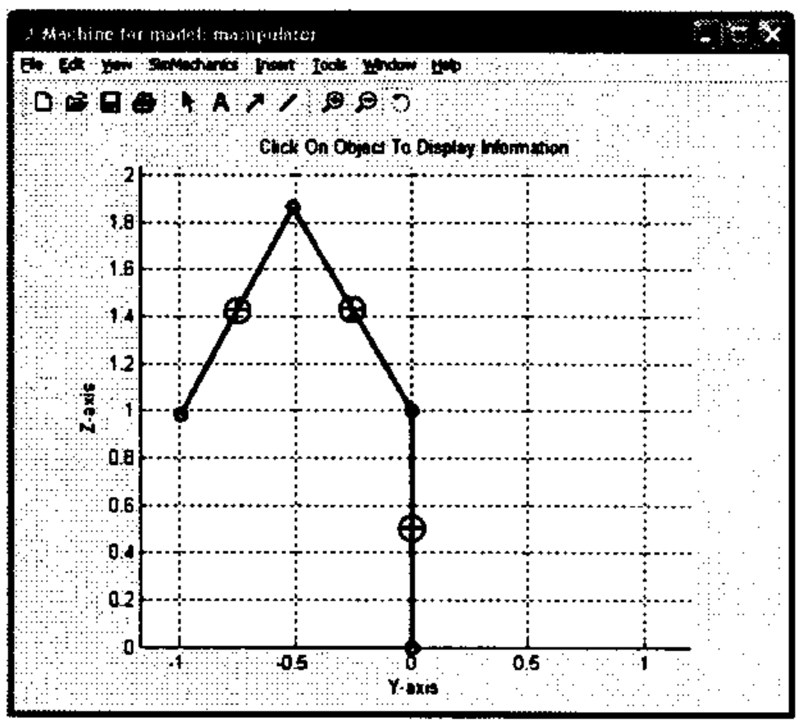

Figure 6. Plain view of simulated manipulator

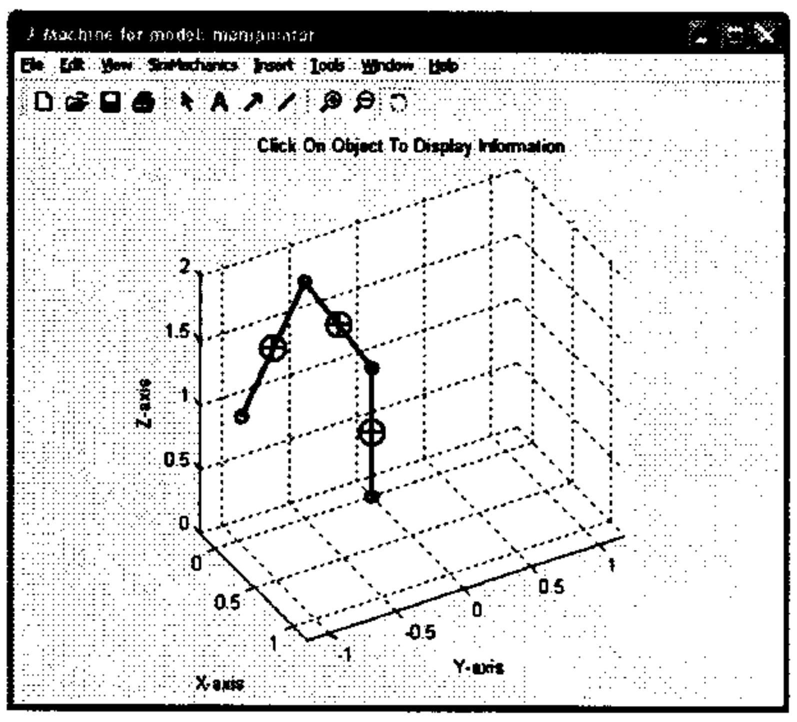

Figure 7. 3D view (oblique projection) of simulated manipulator

Simulation runs in real time, and stops automatically, when manipulator runs up to final point.

Conclusion

This study shows possibility of Matlab and LabVIEW integration. Integration can greatly increase facilitiesof these software products. This study is just example, because PUMA configuration is very simple and described control system is useless for it.

However, integration of MatLab and LabVIEW can be applied for controlling of more complicate configurations of manipulators such as manipulators with redundancy (7 and more degrees of freedom) or manipulators with parallel kinematics. In these cases, this kind of control system should be also top-level system, because it cannot be used for real-time calculation of intermediate points. In a different way, pre-calculations can be used for typical movements, which are used in manufacturing.

References

[1] Ratner D., McKerrow P. "Advancing mobile robot research by using global engineering software tools with technical support over the Internet". 2000. University of Wollongong, Australia.

[2] Salzmann Ch., Gillet D., Huguenin P. "Introduction to Real-Time Control using LabView TM with an Application to Distance Learning". 2000. Swiss Federal Institute of Technology Lausanne, Switzerland.

[3] Coito F., Almeida P., Palma L.B. "SMCRVI - a Labview/Matlab based tool for remote monitoring and control". 2005. Univ. Nova de Lisboa, Portugal.

[4] Gross B., Kozek M., Jorgl H. "Identification and Inversion of Magnetic Hysteresis Using LabVIEW and Matlab". Vienna University of Technology, Austria.

[5] Coolen, E.F.J. West, M.J. Lewis, C.P. "The use of LabView and Matlab in the development of a novelpneumotachograph" Sch. of Eng., Coventry Polytech.