NEW APPROACH OF DYNAMIC SIMULATION OF PUMA 560 IMPLEMENTED IN LabVIEW

Alexei SOKOLOV and Sándor J. TÓTH

Department of Manufacture Engineering
Technical University of Budapest
1111 Budapest, Egry J. u. 1
Hungary

Abstract

Nowadays the growing computer capacity provides a new opportunity of the simulations regarding more reliable and accurate models and fast computation. This paper presents a complete dynamic simulation of the Puma 560 robot. The simulation is based on the dynamic model of the servo drives of the joints and the robot model taking into account mass, inertia, frictions, etc. The robot model and the simulation are implemented in LabVIEW. This approach has several advantages; the opportunities of parallel computation, built in functions provide easy and fast numerical calculation, fast implementation of different models and methods. Several simulation results are demonstrated using different trajectory planning methods for one given path.

Keywords: Puma 560, dynamic simulation, LabVIEW, trajectory planning methods.

1. Introduction

The advance of high-speed technology, the growing computer capacity and the technical improvement of the control devices provide realistic opportunity for new robot control, new simulation technique and new control theories. The trend of this, the technical improvement together with the need for high-performance robots created faster, more accurate and more intelligent robots using new robot controls, new drives and advanced control algorithms. The new control algorithms, the new control devices and in most cases any robot application need simulations before the real operation. That is why the simulation is still emphasized in our research. This paper presents a complete dynamic simulation of the Puma 560 robot. All the geometrical data, mass, inertia, friction of the robot were taken into account in creating the realistic model of the Puma 560. The simulation utilises also the dynamic model of the servo drives of the joints. The robot model and the simulation were implemented using high level graphical programming language (LabVIEW). The programming in LabVIEW has several advantages; the opportunities of parallel computation, built in mathematical functions provide easy and fast numerical calculation and the main advantage is the opportunity of the fast implementation of different models and methods. The purpose of the presented simulation is to improve different trajectory planning methods that are used in the real control sys-

tem. Also the simulation can be used for tuning the parameters of the real control systems. Using the same given path different trajectory planning methods are tested from the accuracy point of view.

The following paragraphs present the models – control system (control card), amplifier, servo motor, friction model and robot hardware (links) – what the Lab-VIEW software utilises in the presented dynamic simulation of the PUMA 560 robot.

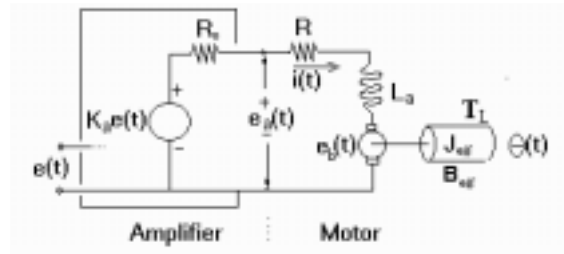## 2. The Model of the Servo Motor and the Amplifier of the Robot

Fig. 1. The model of the amplifier – DC motor combination

The simulation model of the robot utilises the simplified DC motor model, which is well-known and frequently used for simulation and control purpose. *Fig. 1* shows the equivalent-circuit model of amplifier and DC motor, where $e_a(t)$ is the input voltage of armature coil, R is the resistance of the motor armature coil, L_a is the self-inductance of the motor armature circuit, $e_b(t)$ is voltage drop (inverse electromotive force), J_{eff} is the effective inertia and Θ is the angular position of the shaft. According to the Kirchoff law for the electrical side of the DC motor one can write:

$$e_a(t) = L_a \cdot \frac{\mathrm{d}i(t)}{\mathrm{d}t} + R \cdot i(t) + K \cdot \frac{\mathrm{d}\Theta(t)}{\mathrm{d}t}. \tag{1}$$

where K is a motor constant. The model for amplifier-motor combination is easily obtained by substituting $K_A e(t)$ for $e_a(t)$ and $(R_0 + R)$ for R in *Eq.* (1)

$$K \cdot e_a(t) = L_a \cdot \frac{\mathrm{d}i(t)}{\mathrm{d}t} + (R_0 + R) \cdot i(t) + K \cdot \frac{\mathrm{d}\Theta(t)}{\mathrm{d}t}. \tag{2}$$

The torque developed on the shaft by the current – magnetic field interaction (Ampere's rule) is $T(t) = K^* i(t)$. Euler's law applied to the output shaft gives us the following equation:

$$[K \cdot i(t) - \mu \cdot T_L(t) - B_{\text{eff}}(\omega(t))] = J_{\text{eff}} \cdot \frac{\mathrm{d}^2\Theta(t)}{\mathrm{d}t^2}. \tag{3}$$

The well-planned DC motor is capable to accelerate the given robot links (J_{eff} inertia) to compensate the B_{eff} damping effect and the T_L loading torque.
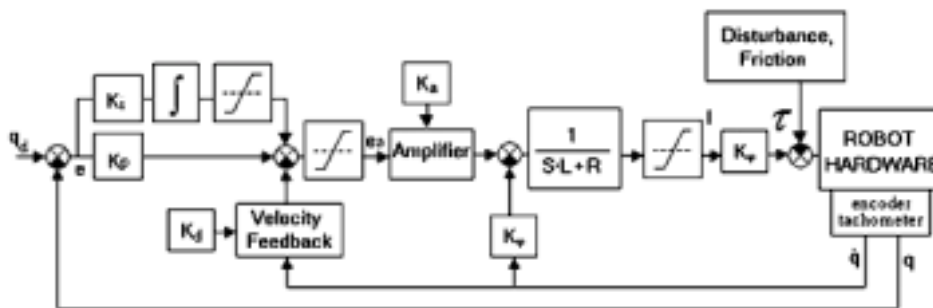
3. The Control System

Fig. 2. The block diagram of the control system

The control algorithm utilises Proportional feedback, Integral feedback and Velocity feedback based on a real robot control card developed by Compumotor [9]. *Fig. 2* describes the block diagram of the control system. The model of the control system was planned so, that the model utilises the same algorithms as the Compumotor control card in the real control system in order to be closed to the reality. The control system consists – in sequence – of the control card including the control algorithm, the amplifier, the DC motor, the friction as disturbance of the process and the robot hardware (links, joints). The parameters of the control system, the DC motor, the amplifier, the friction and the robot hardware characteristic were either measured or they were found in the literature.

4. The Friction Model

Fig. 3 describes the friction in function of the velocity applied at the joints. The two relevant parameters are $Fs_a = 0.146$ Nm and $Fs_b = 0.124$ Nm.
Fig. 4 describes the block diagram of the simulation. The first step in the simulation is the trajectory planning where the path and the velocity profile of the joints are determined. The simulation model concentrates to the first three joints of the robot in this experiment, however, the model is capable to calculate all the six joints. The simulation makes all the calculations parallel through the control algorithm the amplifier and the servo motor. Then comes the direct dynamic calculation block, which is the essential part of the simulation. Dynamic modelling of the robot manipulator consists of finding the mapping between the forces exerted on the

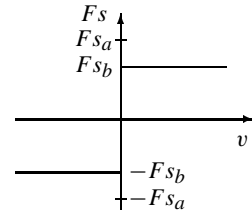| |
|---|
| if $ABS(V) = 0$ and $\tau \leq \quad Fs_a$ then $Fs = \tau$;
if $ABS(V) = 0$ and $\tau \geq -Fs_a$ then $Fs = \tau$;

if $ABS(V) = 0$ and $\tau \geq \quad Fs_a$ then $Fs = Fs_a$;
if $ABS(V) = 0$ and $\tau \leq -Fs_a$ then $Fs = -Fs_a$;

if $ABS(V) > 0$ then $F_s = Fs_b * SIGN(V)$; |

Fig. 3. The friction model

structures and the joints positions, velocities and acceleration. Two formulations are mainly used to derive the dynamic model: the Lagrange formulation and the Newton–Euler formulation.

5. Direct Dynamic Calculation

In the direct dynamic calculation block, basically, the Lagrange formulation is used, but with small changes. In that case when the Lagrange formulation is used and ($n = 6$ joints) there are 425 multiplies and 369 additions are required to take in one simulation step. The *Fig. 5* shows the block diagram of the direct dynamic calculation related to Lagrange formulation. This calculation is based on
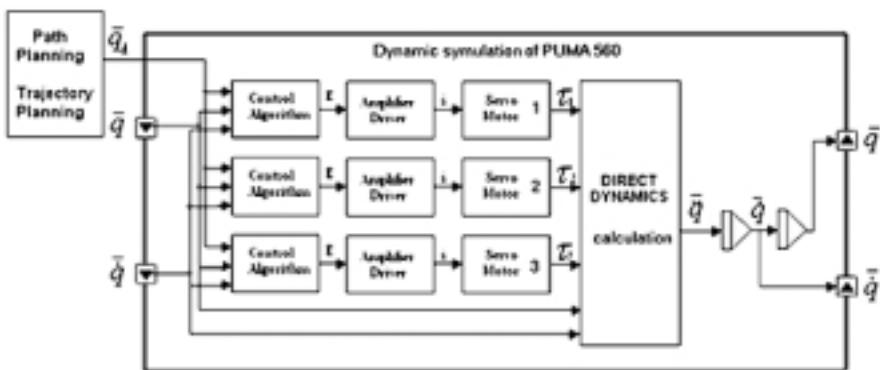

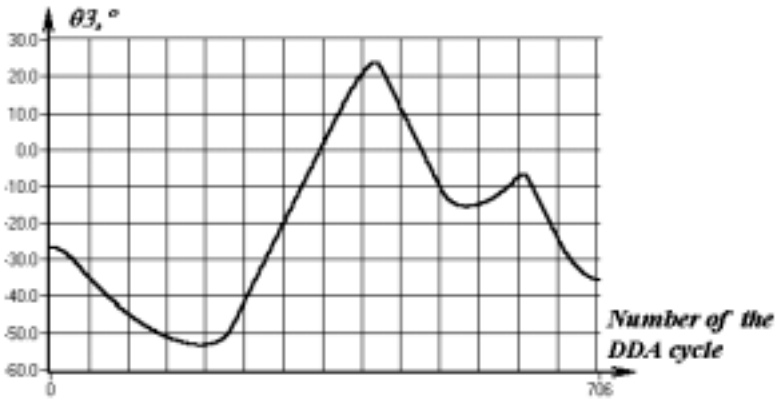*Fig. 4.* The block diagram of the simulation

Fig. 5. The block diagram of the direct dynamic calculation

the Lagrange equation, which is described by *Eq.* (4)

$$\tau = H(q) \cdot \ddot{q} + h(q, \dot{q}) \cdot \dot{q} + C(q), \tag{4}$$

where the $H(q)$ is the $n \times n$ symmetric, positive definite inertial matrix of the robot, $h(q, q)$ is the n degree vector of centrifugal, and Coriolis forces and $C(q)$ is the n degree vector of gravitational forces.

Big amount of calculations are required to produce the $H(q)$ inertial matrix, the $h(\dot{q}, q)$ and the $C(q)$ and the parameters maybe not so punctual as the simulations require. The solution that the authors propose utilises the Newton–Euler formulation to determine the H, h and C matrixes in the following way.

A. The following state is assumed in a given moment of the movement

$$\begin{array}{rcl} q &=& q_{n-1} \\ \dot{q} &=& 0; \\ \ddot{q} &=& 0; \end{array} \quad \boxed{\begin{array}{c}\text{Newton} \\ \text{Euler}\end{array}} \rightarrow \tau_{c1}.$$

Substituting these values to Newton–Euler formulation τ_{c1} is determined. Then using the Lagrange formulation and taking the assumption into consideration

$$\tau_{c1} = C(q) \tag{5}$$

the $C(q)$ function is obtained.

B. In the second step of the calculation another state is assumed in the same moment of the movement

$$\begin{array}{rcl} q &=& q_{n-1} \\ \dot{q} &=& \dot{q}_{n-1}; \\ \ddot{q} &=& 0; \end{array} \quad \boxed{\begin{array}{c}\text{Newton} \\ \text{Euler}\end{array}} \rightarrow \tau_{c2}.$$

Then using the Lagrange formulation as in the previous step

$$\tau_{c2} = h(q, \dot{q}) \cdot \dot{q} + C(q) \tag{6}$$

the $h(q, \dot{q})$ function is obtained.

C/1. In the third step of the calculation the H matrix will be determined

$$q = q_{n-1}, \qquad \dot{q} = \dot{q}_{n-1};$$

and

$$\ddot{q} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \boxed{\begin{array}{c} \text{Newton} \\ \text{Euler} \end{array}} \rightarrow \tau_{c3/1} \ \tau_{c3/1} = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} \cdot \ddot{q} + h(q, \dot{q}) \cdot \dot{q} + C(q). \tag{7}$$

C/2.

$$q = q_{n-1}, \qquad \dot{q} = \dot{q}_{n-1};$$

and

$$\ddot{q} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \boxed{\begin{array}{c} \text{Newton} \\ \text{Euler} \end{array}} \rightarrow \tau_{c3/2} \ \tau_{c3/2} = \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} \cdot \ddot{q} + h(q, \dot{q}) \cdot \dot{q} + C(q). \tag{8}$$

C/3.

$$q = q_{n-1}, \qquad \dot{q} = \dot{q}_{n-1};$$

and

$$\ddot{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \boxed{\begin{array}{c} \text{Newton} \\ \text{Euler} \end{array}} \rightarrow \tau_{c3/3} \ \tau_{c3/3} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} \cdot \ddot{q} + h(q, \dot{q}) \cdot \dot{q} + C(q). \tag{9}$$

After these calculations the Lagrange formulation can be used and based on the simulation module of the direct dynamic (*Fig. 5*) using *Eq.* (10) the $\ddot{q}$ value is obtained

$$\ddot{q} = (H(q) + Z^2 I)^{-1} \cdot (\tau - h(q, \dot{q}) \cdot \dot{q} + C(q)), \tag{10}$$

Where Z is the gear ratio and

$$I = \begin{bmatrix} Id_1 & 0 & 0 \\ 0 & Id_2 & 0 \\ 0 & 0 & Id_3 \end{bmatrix} \tag{11}$$

Id_1, Id_2 and Id_3 are the inertia of the motor shafts.

Using two times the Adams–Basforth Integrator the $\dot{q}$ and q values can be obtained. These two values are stored in shift registers for the next calculation step.

6. Simulation Results

The presented dynamics calculation method is faster than the normal use of Lagrange computation. In our case the simulation time step was $dt = 0.0005$ s and the $T = 24$ s, it means 48000 simulation steps. Using Pentium II.(300 Mhz) based PC the simulation took 337 s. *Fig. 6* describes the path of the robot and *Fig. 7* shows the arbitrary speed profile applied along the path.
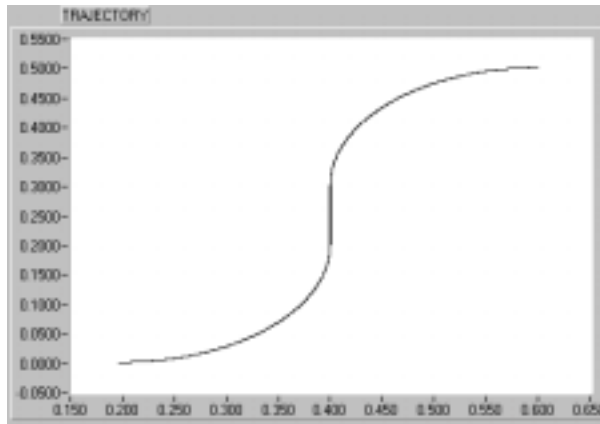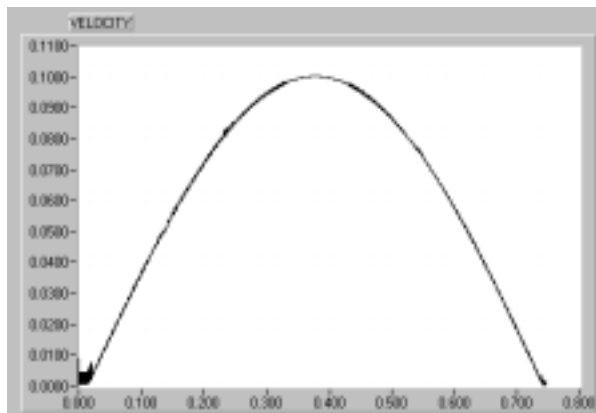

Fig. 6. The trajectory of the robot in xy plane

Fig. 7. The arbitrary speed profile along the path

In this case the maximum speed along the path is 0.1 m/s. This is a rather low value compared to the real speed values of the robots. *Fig. 9* describes the absolute error along the path.
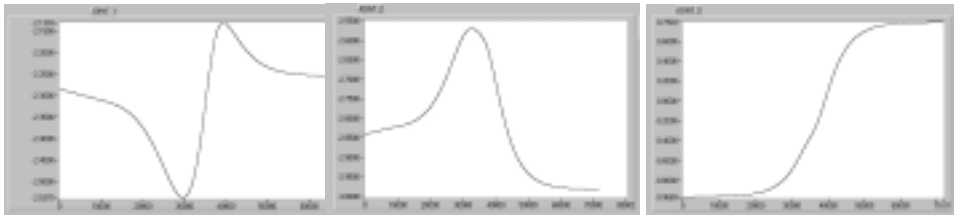
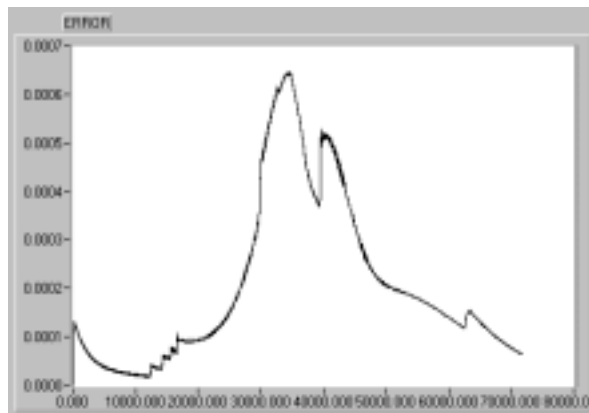*Fig. 8.* The trajectories of the first three joints

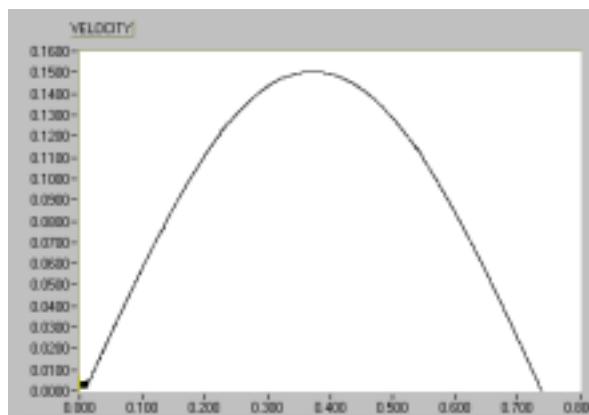*Fig. 9.* The absolute error along the path

Fig. 10. The arbitrary speed profile along the path

The maximum difference between the desired and the simulated values is 0.65 mm, which value is a reasonable one considering the low speed value along the path. By
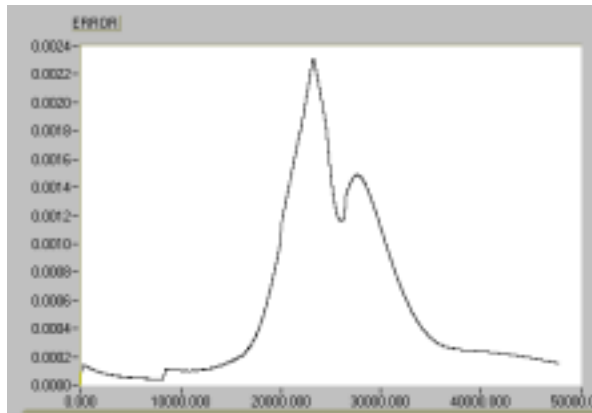
Fig. 11. The absolute error along the path

increasing the speed value and using the same arbitrary velocity profile – *Fig. 10* describes it – the maximum error value increases too – see in *Fig. 11*.
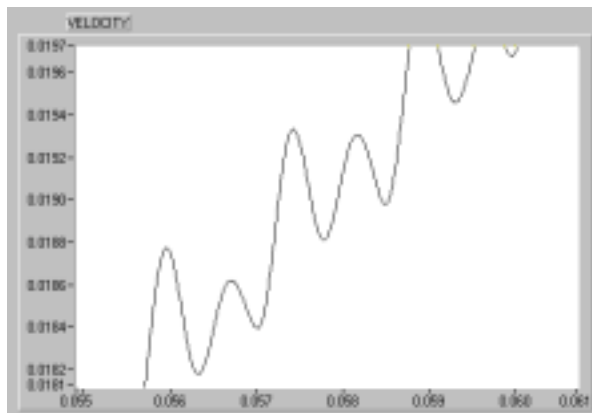

Fig. 12. The zoomed speed profile

Fig. 12 describes the part of the speed profile using the zoom function. This figure shows the dynamic behaviour of the system. The whole profile consists of small responds for step functions.

7. Conclusion

The presented dynamic simulation of the Puma560 gives reasonable results regarding to the real robot behaviour. Further study should be made to make a comparison

experiment between the real system and the model using an external measuring system. The simulation model used the first three joints of the robot in this experiment, but the dynamic calculation module takes into account all the six joints dynamics. The authors described a new way of calculation of the direct dynamics, which is faster than the usual way of use of the Lagrange formulation. The simulation model of the robot was developed using LabVIEW. This achievement provides easy and fast further development in improving the model and the trajectory planning methods. In this paper the arbitrary trajectory planning method was used to make comparison in absolute error regarding to the speed along the path.

Acknowledgement

References

[1] SHIN, K. G. – MCKAY, N. D. (1985): Minimum-time Control of Robotics Manipulators with Geometric Path Constraints, *IEEE of Automatic Control AC-30*, Vol. 6, pp. 531–541.
[2] SINGH, S. K. (1991): Trajectory Planning for Robot Control: A Control System Perspective, *Control and Dynamic System Series, Advances in Robotics System*, pp. 105–146, Academic Press Inc.
[3] PODURAJEV, J. – SOMLÓ, J. (1993): A New Approach to the Contour Following Problem in Robot Control (Dynamic Model in Riemann Space), *Mechatronics*, Vol. 3, No. 2, pp. 241–263.
[4] SOMLÓ, J. – PODURAJEV, J. (1994): Optimal Cruising Trajectory Planning for Robots, *Mechatronics*, Vol. 4, No. 5, pp. 517–538.
[5] J. TÓTH, S. (1994): Optimal Trajectory Planning for Robots, MSc Thesis, Budapest Technical University.
[6] LIPOVSZKY, GY. (1994): TUBSIM Digital Simulation Program, *http://www.rit.bme.hu/letoltheto/szamszim/modsim.html#Eleje*.
[7] ZENTAY, P. – ZOLLER, Z.: Time Optimal Trajectory Planning for Robots in LabView Programming System MicroCAD'99 (to be published).
[8] ERDŐS, G. (1988): Ipari robotok geometriai, kinematikai és dinamikai modellezésének új szimbólikus módszere, Ph.D. Thesis, Budapest.
[9] Servo Tuner User Guide, Compumotor Division Parker Hannifin Corporation, 1994.
[10] Robotics: Control, Sensing, Vision and Intelligence, K.S.Fu, R.C. Gonzalez, C.S.G. Lee, London, 1989.