

ROBOT MOTION REALISATION USING LABVIEW

Alexei SOKOLOV

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

Received: Oct. 14, 1999

Abstract

Any kind of robot motion can be defined by the path and the velocity along it. Usually the velocity is given in the form: $\dot{\lambda} = f(\lambda)$ where the parameter λ is the length of the path, $\dot{\lambda}$ is the velocity along the path. The determination of the velocities along the path is named trajectory planning.

Various kinds of motion (time optimal, motion with constant kinetic energy, process optimal, etc.) are given by different approaches.

Therefore, there is a reason to implement robot control device which can realise the motion along a given path with arbitrary velocity profile. Generally speaking, it is rather difficult to solve this task using the existing control units of industrial robots.

The LabVIEW Virtual Instrument technology gives an opportunity to build different sophisticated control systems. Such an approach makes the system rather flexible than the existing ones and allows to get information, using the LabVIEW interface, about the inner processes in convenient form. All these advantages are very important for experimental investigations of different control methods.

Using this control device time optimal robot motion was realised. It is also possible to realise robot motion along the given path with arbitrary velocity profile.

The results obtained by the LabVIEW based control system can be applied to the investigations of the robot motion and to evaluate the goodness of different control principles.

Keywords: LabVIEW programming, control device, robot control, optimal trajectory planning.

1. Introduction

At present time a big amount of industrial robot control devices exists. They accurately execute their tasks, however, they have some drawbacks. For example:

- high prices of the control device
- lack of the flexibility
- closed systems (it is very difficult to add special measuring units and to process their signals)
- restrictions in software
- big problems for getting information about processes in the system (voltage, current).

All these factors make existing systems inappropriate to laboratory researches.

In consequence of this the idea of creating of the new control system for industrial robot 'PUMA560' based on the personal computer with flexible software was suggested.

The main task which was put for designers included the design of hardware and software without above-mentioned lacks. It was also necessary to minimise the expenses.

On the first level of experimental investigations it was suggested to use only the second and the third joints of the robot. This approach gives the possibility to organise the motion in vertical plane which is sometimes enough for research purposes.

2. Control Card PCL-832

The drive control system is built on the PCL-832 card. This card is produced by Advantech Co., Ltd.

The PCL-832 3-axis Servo-motor Control Card turns IBM PC or compatible computer into a sophisticated position controller. The card's custom ASIC implementation provides high performance at an affordable price.

The PCL-832 uses digital differential analysis techniques to implement position control. Each axis has its own position control chip, allowing completely independent control of up to three servo motors. A special synchronisation circuit synchronises all three axes. The card can supply a simulated tachometer output to the servo motor driver. This signal makes a tachometer unnecessary in some applications, reducing overall system costs.

There are following features:

- Independent 3-axis servo control
- Fully continuous closed-loop P+offset controller
- Industry-standard two-phase index position encoder interface
- Single-ended or differential encoder interface inputs
- x1, x2, x4 quadrature feedback input
- 12-bit analog output with $\pm 10V$ range
- Built-in frequency/voltage converter
- Easy programming from C and other high-level languages
- Half-size AT (ISA bus) add-on card

Similar control card was used at the development of control system of the ZIM 15 (KUKA license). For details see [5].

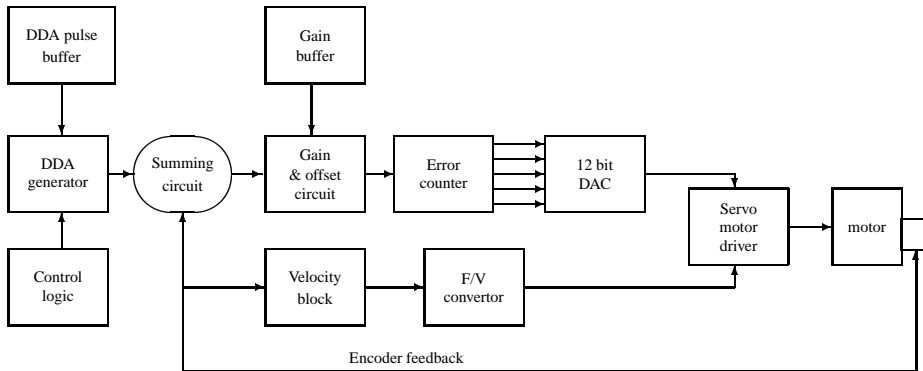


Fig. 1. The closed-loop position control

2.1. Closed-Loop Position Control

The PCL-832 uses proportional closed-loop position control to obtain reliable and accurate results. It features an internal velocity feedback loop and offset techniques to compensate for the steady-state error that is caused by using small values on the P controller.

The functional block diagram of the PCL-832 motion control card is shown below:

The DDA generates continuous command pulses through CMD+ and CMD- channels (not shown on the Fig. 1). These command pulses are fed into a summing circuit, together with the pulses generated by the servo motor encoder device. The summing circuit determines the difference between the two signals. The computed result is then fed into the P pulse-offset controller. The P pulse-offset controller, which has programmable gain (Kp), outputs pulse numbers. This pulse is fed into the error counter, which drives the DAC chip in real-time.

A velocity block is provided in the motion control chip. Its purpose is to add a velocity feedback loop in the whole system through a frequency-to-voltage (F/V) converter. This internal loop improves the motion dynamics of the servo motor system.

For more detailed information about PCL-832 see [4].

3. Servo Amplifier

In this system for attending the drives, servo amplifier model 12A8 is used. It has several features:



Fig. 2. End-effector path

- Surface-mount technology
- Small size, low cost, ease of use
- DIP switch selectable: current, voltage, velocity, IR compensation, position loop control
- Four quadrant regenerative operation

12A Series PWM (pulse width modulation) servo amplifiers are designed to drive brush type DC motors at a high switching frequency. Operating efficiencies approach 99%.

4. Programming the PCL-832

Programming the servo motor control card includes the creation of the special device driver, which is based on writing the interrupt service routines. Several special functions were written on programming language Borland C++ v.4.52 for Windows 3.1 according with the special literature recommendations and were compiled as dynamic link library.

The DDA time interrupt service routine sets numbers computed in advance in the DDA pulse buffers for every axis before the current DDA cycle finishes. During this routine operation other interrupts are disabled by the special instruction. Then the numbers are converted to special form, and the software writes numbers to the ports according to I/O address map.

Also this DLL consists of other functions which execute the following functions:

- the initialisation and closing DDA time and counter overflow interrupts;
- reset PCL-832;
- the axis gain set;
- enabling DDA cycle generator;

- setting DDA time;
- getting error counter's value;
- setting pulse number;
- getting axis status.

All of these functions use direct memory access instructions. The LabVIEW calls them from the pre-compiled DLL file.

The user's interface is built using the LabVIEW's library great possibilities. On the virtual instrument front panel there are switches, LEDs, terminals. User can control the different values. There are also push-buttons which make user's interface handy.

It is possible to define pulse numbers manually or to get it from the file or other calculations.

The VI is divided onto several modules and sub-VIs. It gives the possibility to make the software flexible and to get a module structure of it. Therefore in the variable conditions it will be enough to modernise only modules. It is the first part of the software. It directly operates with control card and uses pre-computed data. It gives all possible information about the control card's processes.

5. Optimal Motion Realisation on the Experimental Device

The main task of the designed device is the experimental investigation of different robot motion principles. To solve this problem it is necessary to have the software which can produce the calculations resulting robot motion along the path.

It will be convenient to deal only with the motion in the vertical plane which is perpendicular to the second and the third joint's axis, because only these joint drives are used.

5.1. Time-Optimal Cruising Trajectory

In many applied tasks (especially, for the technological use of robots, such as welding, deburring, grinding, milling, painting, etc.) the working actions are performed during cruising motion.

For trajectory planning, during the cruising part of the motion, the following simple approach is proposed (see at [1], [2]). At every point of the path the possible maximum velocity value should be used. This maximum, very frequently, is determined by the limit value of one of the joint velocities. The joint for which it occurs is named dominant joint.

It is supposed that the geometry of the path is known and defined in parametric form according to the PUMA560 coordinate system:

$$\begin{cases} y = y(u) \\ z = z(u) \end{cases} \quad (1)$$

where

u is arbitrary parameter;
 y, z are the world's coordinates.

In the real system these functions are defined in the discrete form. It means that path points are defined through the step:

$$\begin{cases} y_i = y(u_i) \\ z_i = z(u_i) \end{cases} \quad (2)$$

where

$i = 0, 1..n$ point index;
 n number of the points.

Let us calculate the time which is required for motion from point A to point B supposing that it is a cruising path and at least one of the joint's velocity has its limit value.

The determination of the joint coordinates which result the given world coordinates is the solution of the inverse kinematics task.

5.2. Inverse Kinematics Task

The solution is given for PUMA 560 manipulator (see: [3]). By analogy with human arm's geometry and according to the coordinates system link allocation for PUMA 560 manipulator, different configurations are defined by three configuration switches (arm, elbow, wrist). Two indicators characterise relative position of the first, the second and the third joints, but the third indicators – relative position of the fourth, the fifth and the sixth joints. There are four different solutions of the inverse kinematics six joints robot problem for the three firsts joints and there are two else for each of them for three lasts.

The two first configuration indicators allow to choose one of the four possible solutions for the first three joints. In analogy, the third indicator defines one choice from the two possible solutions for the last three joints. The user makes the configuration indicator choice before the inverse kinematics task solution is started.

The solution consists of two parts. First of all the solution for the first three joints is found. In the second place the solution for the last three joints is found.

There are different possible configuration types of PUMA 560 manipulator.

They are determined by the following principles:

RIGHT ARM	If the third joint stays in stationary position the second joint angle value increasing increases the end-effector Z_0 coordinate.
LEFT ARM	If the third joint stays in stationary position the second joint angle value increasing decreases the end-effector Z_0 coordinate.
UPPER ARM	{RIGHT/LEFT} arm relative position for elbow's coordinate system characterises by {NEGATIVE/POSITIVE} Y_0 coordinate value.
LOWER ARM	{RIGHT/LEFT} arm relative position for elbow's coordinate system characterises by {POSITIVE/NEGATIVE} Y_0 coordinate value.
WRIST DOWN	Scalar product of the gripper coordinate system unit-vector s and the coordinate system (X_5, Y_5, Z_5) unit-vector Y_5 is positive.
WRIST UP	Scalar product of the gripper coordinate system unit-vector s and the coordinate system (X_5, Y_5, Z_5) unit-vector Y_5 is negative.

According to this definition for every of these configurations the configuration indicators (ARM and ELBOW) are defined. Jointly both of them separate one from the four possible inverse kinematics problem solutions for the first three joints. For every possible manipulator configurations, which are defined by the two first indicators, the third indicator (WRIST) conditions the choice of one of the two possible inverse kinematics task solutions for the three last joints. The mentioned three joint configuration indicators can be defined in the following way:

$$\text{ARM} = \begin{cases} +1, & \text{if RIGHT arm} \\ -1, & \text{if LEFT arm} \end{cases} ;$$

$$\text{ELBOW} = \begin{cases} +1, & \text{if UPPER arm} \\ -1, & \text{if LOWER arm} \end{cases} ;$$

$$\text{WRIST} = \begin{cases} +1, & \text{if the wrist DOWN} \\ -1, & \text{if the wrist UP} \end{cases} .$$

In addition to mentioned indicators there is a switch:

$$\text{SWITCH} = \begin{cases} +1, & \text{to change wrist orientation} \\ -1, & \text{don't change wrist orientation} \end{cases} .$$

The indicator and switch values are defined by the researcher before the beginning of the inverse kinematics task solution.

Based on the robot kinematics structure it is well known that (see: [3])

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} C_1(a_2C_2 + a_3C_{23} + d_4S_{23}) - d_2S_1 \\ S_1(a_2C_2 + a_3C_{23} + d_4S_{23}) + d_2C_1 \\ d_4C_{23} - a_3S_{23} - a_2S_2 \end{bmatrix}, \quad (3)$$

where $\mathbf{p} = [p_x, p_y, p_z]^T$ is the vector which connects the initial point of the

coordinate system (x_0, y_0, z_0) with intersection point of the last three coordinate systems axis;

$C_i \equiv \cos \theta_i$; $S_i \equiv \sin \theta_i$; $C_{ij} \equiv \cos(\theta_i + \theta_j)$; $S_{ij} \equiv \sin(\theta_i + \theta_j)$.

PUMA 560 manipulator parameters are introduced in *Table 1*.

Table 1. Denavit–Hartenberg parameters of the PUMA 560 robot [3]

Joint i	θ_i	α_i	a_i , mm	d_i , mm	Limits
1	θ_1	-90°	0	0	-160° – $+160^\circ$
2	θ_2	0	431.8	149.09	-225° – 45°
3	θ_3	90°	-20.32	0	-45° – 225°
4	θ_4	-90°	0	433.07	-110° – 170°
5	θ_5	90°	0	0	-100° – 100°
6	θ_6 b	0	0	56.25	-266° – 266°

The inverse transformation equation for the first joint is (see [3]):

$$\theta_1 = \operatorname{arctg} \left[\frac{-ARM p_y \sqrt{p_x^2 + p_y^2 - d_2^2} - p_x d_2}{-ARM p_x \sqrt{p_x^2 + p_y^2 - d_2^2} + p_y d_2} \right], \quad (4)$$

where $-\pi \leq \theta_1 \leq \pi$, θ_1 is the first joint's coordinate.

For the second joint one has:

$$\theta_2 = \operatorname{arctg} \left[\frac{\sin \theta_2}{\cos \theta_2} \right], \quad (5)$$

where $-\pi \leq \theta_2 \leq \pi$, θ_2 is the second joint coordinate value.

$\sin(\theta_2)$ and $\cos(\theta_2)$ can be determined as follows. Let us introduce:

$$R = \sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}, \quad (6)$$

$$r = \sqrt{p_x^2 + p_y^2 - d_2^2}, \quad (7)$$

$$\sin \alpha = -\frac{p_z}{R} = -\frac{p_z}{\sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}}, \quad (8)$$

$$\cos \alpha = -\frac{ARM \cdot r}{R} = -\frac{ARM \sqrt{p_x^2 + p_y^2 - d_2^2}}{\sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}}, \quad (9)$$

$$\cos \beta = \frac{a_2^2 + R^2 - (d_4^2 + a_3^2)}{2a_2R} = \frac{p_x^2 + p_y^2 + p_z^2 + a_2^2 - d_2^2 - (d_4^2 + a_3^2)}{2a_2\sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}}, \quad (10)$$

$$\sin \beta = \sqrt{1 - \cos^2 \beta} \quad (11)$$

where α and β are the secondary angles.

Using the geometrical method of determination of the inverse transformations (see [3])

$$\theta_2 = \alpha + (ARM \cdot ELBOW) \cdot \beta \quad (12)$$

From these equations it is clear that:

$$\sin \theta_2 = \sin(\alpha + K \cdot \beta) =$$

$$\sin \alpha \cos(K \cdot \beta) + \cos \alpha \sin(K \cdot \beta) = \sin \alpha \cos \beta + K \cos \alpha \sin \beta \quad (13)$$

$$\cos \theta_2 = \cos(\alpha + K \cdot \beta) = \cos \alpha \cos \beta - K \sin \alpha \sin \beta \quad (14)$$

where $K=ARM \times ELBOW$.

For the third joint the following equations are valid:

$$R = \sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2} \quad (15)$$

$$\cos \varphi = \frac{a_2^2 + (d_4^2 + a_3^2) - R^2}{2a_2\sqrt{d_4^2 + a_3^2}} \quad (16)$$

$$\sin \varphi = ARM \cdot ELBOW \sqrt{1 - \cos^2 \varphi} \quad (17)$$

$$\sin \beta = \frac{d_4}{\sqrt{d_4^2 + a_3^2}} \quad (18)$$

$$\cos \beta = \frac{|a_3|}{\sqrt{d_4^2 + a_3^2}} \quad (19)$$

$$\theta_3 = \varphi - \beta \quad (20)$$

$$\sin \theta_3 = \sin(\varphi - \beta) = \sin \varphi \cos \beta - \cos \varphi \sin \beta \quad (21)$$

$$\cos \theta_3 = \cos(\varphi - \beta) = \cos \varphi \cos \beta + \sin \varphi \sin \beta \quad (22)$$

$$-\pi \leq \theta_3 \leq \pi \quad (23)$$

To simplify the task the following coordinate values are used:

$$p_x = -d_2 ; \theta_1 = \pi/2 = \text{const.}$$

It means that the first joint is not moving and the end-effector motion occurs in the vertical plane, which is in d_2 distance from the beginning of the coordinate system.

From (4) it follows that:

$$\begin{cases} p_y \geq 0 : ARM = -1; \\ p_y < 0 : ARM = 1; \end{cases}$$

The ARM*ELBOW can change its value only in the singular points.

Attention is not paid for the orientation of the gripper, therefore the inverse kinematics problem solution for the three last joints is not given here.

5.3. Time-Optimal Cruising Trajectory Realisation

As it was mentioned above, first of all it is necessary to solve the inverse kinematics task for every point of the path. To solve this task one should use the principles and equations which were given in the previous paragraph. Then, it is necessary to calculate the time which is required for the motion from point to point.

During the i -step the joint coordinate values are changing from θ^{i-1} to θ^i . The rotation angles of every link during one step are:

$$\Delta\theta_2^i \text{ and } \Delta\theta_3^i .$$

$$\Delta\theta_2^i = \theta_2^i - \theta_2^{i-1}, \quad \Delta\theta_3^i = \theta_3^i - \theta_3^{i-1}$$

From the definition of the cruising motion it follows that for time-optimal motion at least one of the velocity values should be at its limit value. The dominant joint which velocity is at maximum should be found. For this purpose it is necessary to calculate the required time for each joint when it moves with the maximum velocity. So:

$$t_2^i = \left| \frac{\Delta\theta_2^i}{\dot{\theta}_2^{\max}} \right| ; \quad t_3^i = \left| \frac{\Delta\theta_3^i}{\dot{\theta}_3^{\max}} \right| \quad (24)$$

where $\dot{\theta}_2^{\max}$ and $\dot{\theta}_3^{\max}$ are the velocity limit values of the second and the third joints, correspondingly.

Because, only the bigger of these values may be realised, comparing them, it is possible to find the maximum value and to make a conclusion about the dominant joint.

$$t_i = \max \langle t_2^i, t_3^i \rangle .$$

The joint which requires the maximum time of the motion is the dominant one. It has to move with maximum velocity and the other joint will have the velocity:

$$\dot{\theta}_j^i = \frac{\Delta\theta_j^i}{t_i} . \quad (25)$$

The required overall motion time is the sum of the times for each step:

$$t = \sum_{i=1}^n t_i. \quad (26)$$

It is necessary to find the rotation angles of each joint during every DDA cycle. First of all it has to be found the amount of the DDA cycle periods from the equation (it is supposed that DDA cycle time is known):

$$N = \text{INTEGER} \left[\frac{t}{T_{DDA}} \right], \quad (27)$$

where T_{DDA} is the time of one DDA cycle period.

If the motion time is known, it is possible to define the joint coordinates value as time-functions in discrete form:

$$\theta_2^i = \theta_2^i(t_i) \text{ and } \theta_3^i = \theta_3^i(t_i);$$

where $i = 1 \dots n$.

Using the linear interpolation method it is possible to find the function values for every j -th step of DDA cycle and prepare the data:

$$\theta_2^j = \theta_2^j(j \cdot T_{DDA}) \text{ and } \theta_3^j = \theta_3^j(j \cdot T_{DDA});$$

where $j = 0 \dots N$.

During the next step it is necessary to calculate the rotation angle for every DDA cycle time.

$$\Delta\theta_2^j = \theta_2^j - \theta_2^{j-1}, \quad \Delta\theta_3^j = \theta_3^j - \theta_3^{j-1} \quad (28)$$

where $j = 1 \dots N$. For the definition of the pulse numbers for every DDA cycle there is a need to produce the following calculations:

$$P_j = \text{INTEGER} \left| \frac{\Delta\theta^j \cdot z \cdot R}{2\pi} \right|, \quad (29)$$

where P_j the pulse number for j DDA cycle;
 z gear ratio of the drive;
 R encoder's resolution of the axis (steps per revolution);
 $j = 1 \dots N$.

For the second and the third axis the values are given in the *Table 2*.

In this way one gets the pulse number arrays for each axis. These arrays will be the input values for the servo motor control card.

Based on the proposed algorithms LabVIEW program has been developed.

Similar methodology can be applied for the realisation of the given orientations of the gripper (see: [2]).

Table 2. Parameters of the 2nd and 3rd robot drives

Axis	z , Gear ratio	R , Encoder resolution
2	107.36	800
3	53.69	1000

5.4. The Transient Parts of the Trajectory

The 6th paragraph of this paper describes some experimental results realised with the developed control system. For the determination of the motion it is necessary to determine the transient motion, too. There are theoretical results (see: [2]) available for the optimisation of transient motions. It has some similarity with the determination of optimal cruising trajectories. On the transient part of the motion maximum acceleration should be determined which is constrained by the maximum torque of one of the joints.

A simpler approach, constant realisable acceleration of the joints was used at the experiments described in the 6th paragraph.

The goal of the development of the experimental device was to be able to compare different trajectory planning aspects.

The motion along given path with arbitrary velocity profile can also be realised using the given experimental system. But, because of the lack of space details will not be given here.

6. Experiments

In the previous sections the different principles of robot motion realisation were discussed. In this section an example will be given for time optimal motion realisation.

It was supposed that the PUMA 560 robot performs working action in vertical plane. Tools can be attached to the end effector centre point (welding head or milling tool perpendicularly to the yz plane). Then, moving along the given working path, technological operations can be performed.

The working path used at the example is shown on *Fig. 3*. It consists of four parts: horizontal line, big radius arc, slopping line and small radius arc. The robot starts the motion at the beginning of the horizontal line and finishes it at the end of the small radius arc.

It was supposed that the velocity limits for the second and the third joints have the values 0.5 s^{-1} .

The value of DDA cycle time is 15 ms.

First of all, the cruising part of the path was considered. According to the described above methods the inverse kinematics task was solved. On the *Fig. 4* and *Fig. 5* there are the graphics of the functions $\theta_2 = \theta_2(t)$ and $\theta_3 = \theta_3(t)$.

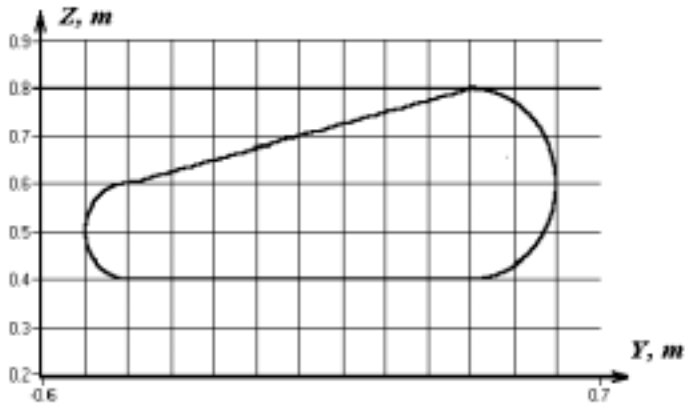


Fig. 3. The path of the motion

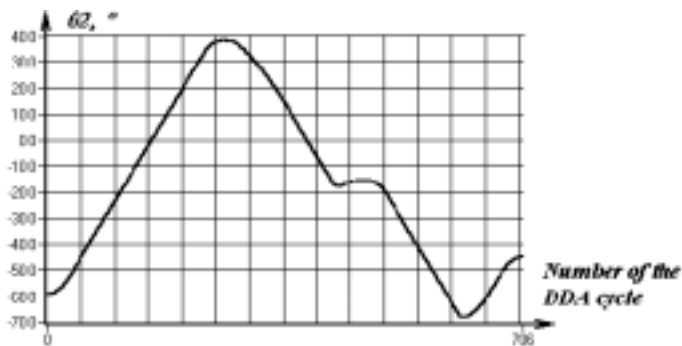


Fig. 4. The joint coordinate values for the second joint

Then the pulse number values were determined. The pulse number values are proportional to the joint velocity.

During the transient parts the joints are moving according to trapezoid velocity profile with constant acceleration or deceleration. When robot moves along the cruising part one of the joints has its limit velocity value. It should be noted that during the cruising part the dominancy is switched from joint to joint several times.

The defined trajectory is executed by the robot during the calculated time with small error values.

7. Conclusion

For experimental investigations of different types of robot motions a control device for PUMA 560 industrial robot was developed.

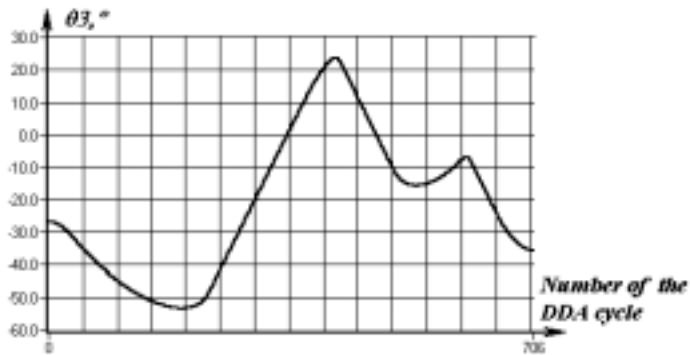


Fig. 5. The joint coordinate values for the third axis

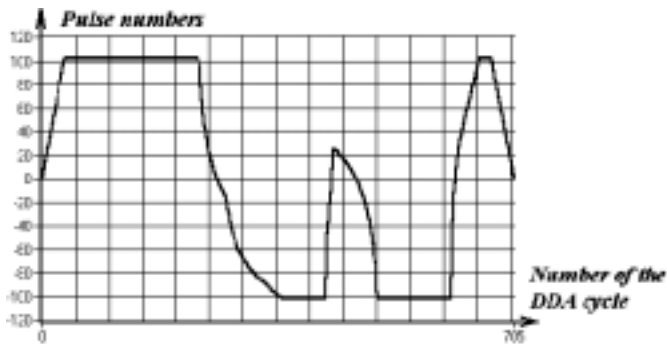


Fig. 6. The pulse numbers for the second axis

On the first level of the investigations, it was suggested to use only the second and the third joints of the robot. Using these joints, it is possible to realise the end-effector motion along paths in the vertical plane.

The control device hardware was built into a personal computer and was realised using the 3-axis servo motor control card PCL-832. The software was based on LabVIEW.

Driver program for PCL-832 was created. For this purpose, the dynamic link library was written in the program language Borland C++ 4.52. The functions are called by LabVIEW from the DLL file. The program has a modular structure and can be modernised in simple way.

Virtual instruments for LabVIEW were developed which realise the robot motion along user-defined paths. Using this software, the robot can execute time-optimal motions and the motions with arbitrary velocity profiles. The last virtual instrument gives the possibility to make experimental investigation of any motion principles.

The inverse kinematics problem of PUMA 560 robot was solved. The method

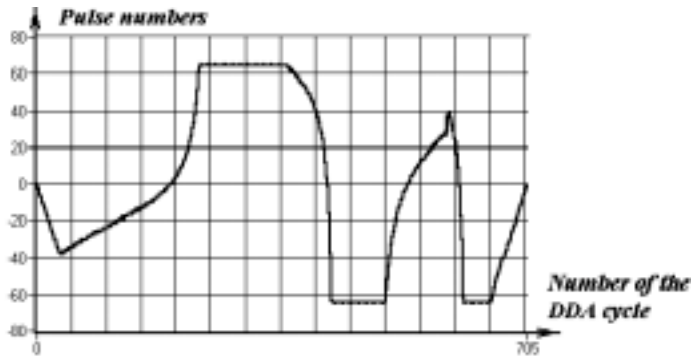


Fig. 7. The pulse numbers for the third axis

of solving is fully described in this paper. For the second and the third joints there are two possible solutions of the inverse kinematics task. The program takes a decision itself about which solution is to be used at the moment.

The results of calculations are the pulse numbers array. These values are sent to the servo motor card by the PCL-832's driver program.

Using the designed experimental device it is possible to make further investigations, to evaluate the existing approaches and to suggest new principles of the robot motion optimisation.

Acknowledgement

The device design was based on the help of the Research Fund of Hungarian Academia of Sciences (OTKA T 026407), also the theoretical work was supported by this and earlier OTKA projects, for which the author expresses sincere thank.

Author expresses gratitude to scientific leader Professor J. Somló.

References

- [1] SOMLÓ, J. – PODURAJEV, J. (1994): Optimal Cruising Trajectory Planning for Robots, *Mechatronics*, **4**, No. 5, pp. 517–538.
- [2] SOMLÓ, J. – LANTOS, B. – CAT, P.T. (1997): 'Advanced Robot Control', pp. 430, Akadémiai Kiadó, Budapest.
- [3] FU, K. S. – GONZALEZ, R. C. – LEE, C. S. G. (1989): 'Robotics: Control, Sensing, Vision and Intelligence', pp. 624, 'Mir', Moscow (in Russian).
- [4] PCL 832 Control Servo Card. User's manual.
- [5] TÓTH, S. J.: (1999) 'New PC and LabVIEW Based Control System' (to be published).