Control of OWI 535 robotic arm using C# and possibilities of future applications

[1]Jakub ČERKALA (1st year)
Supervisor : [2]Anna JADLOVSKA

[1] Dept. of Cybernetics and Artificial Intelligence, FEI TU of Košice, Slovak Republic

[1]jakub.cerkala@tuke.sk, [2]anna.jadlovska@tuke.sk

***Abstract*— In this article will be presented the control of robotics arm model OWI 535 using USB interface. Control program is designed as C# library functions and its goal is the practical use of simple robotics arm model for educational purposes. Within this article will be presented and described robotics arm, its parameters and kinematics model. Furthermore, direct communication using USB between robot's control unit and own application with graphical user interface will be described as well. In addition, the theoretical and practical options of model usage in education will be also presented.**

***Keywords*— robotic arm, C# programing language, graphical user interface, .NET library, communication via USB**

I. Introduction

Robotics and mainly robotic arms represent nowadays a wide field of science and they are often combined with mobile robotics branch and artificial intelligence methods usage [1][2][3]. There are several types of robotic arms, mainly divided by the number of degrees of freedom and use.

Robotic arms are designed mainly for industrial use and they are widely used in manufacturing, usually with robust design to allow one type of robot to be used for variety of purposes just by changing the software and tool, which is placed in the wrist of the robot. In most cases, as joints are driven by powerful servos and for this reason they are relatively expensive and less available.

This article goal is to present the possibilities of enhancement and usage of simple robotics arm for educational purposes. This paper deals with the OWI 535 robotics arm model with USB control unit, shown on Fig. 1.

II. Robotic arm model description

A. Device overview

OWI 535 itself offers an affordable robotic arm with 4 +1 degrees of freedom, which is designed for educational purposes [4]. There are various versions of mentioned model, this article describes the version, for which the robot is connected to the PC using the USB interface. The specified robotic arm model is supplied with driver for x86 and x64 operating systems and simple application for control, but this application is only an executable application without source code, which can't be modified.

Fig. 1. Maplin OWI 4 +1 DOF robotic arm with USB interface.

B. Properties and motion ranges

As mentioned earlier, the robotic arm has 4 +1 degrees of freedom, and they can be represented by subsystems: base rotation, base angle, elbow, wrist and gripper. Ranges of possible motion are listed in Table I. Robot uses independent 3V power supply from batteries and control unit is powered via USB interface. In present, model's power supply has been replaced by common PC stabilized power source to improve of motion characteristics, since batteries cannot guarantee stabile current ant voltage during whole time in use.

TABLE I
Subsystem Ranges

Subsystem	Name	Range
S1	Base rotation[1]	270 degrees
S2	Base angle	180 degrees
S3	Elbow joint	300 degrees
S4.	Wrist joint	120 degrees
S5	Gripper	Approx. 0.045 m wide grip

[1] Base can be mounted two different ways – desired rotation from control unit can be chosen.

Subsystem contains DC engines and gearboxes, which is the main weakness of the model in comparison to robots used in industry and medicine. However, gearboxes are constructed safeguard against over spin and therefore the damage resistance and durability is higher. Robotic arm uses identical gearboxes for all joints. A LED lamp is mounted into gripper, and it can be controlled like other subsystems. Robot can lift approximately 100g. Vertical reach is approx. 380 mm and horizontal reach is approx. 320 mm.

C. *Kinematic model of robotic arm*

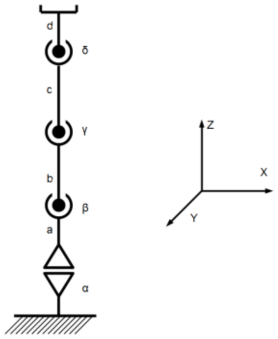

Fig. 2. Robotic arm kinematic model.

It is assumed that the robotic arm is rotated in the middle of base rotation position and the rest of the joints are rotated up and thus the arm is in maximum vertical position. After obtaining the dimensions and distances between joints, a kinematic model of the robotic arm can be built. With use of the matrixes R_Y, R_Z for rotational motion around Y axis (1) and Z axis (2) a direct kinematic model can be obtained [5], shown on Fig. 2., according to which gripper position in space for specific base rotation α and joint rotations β, γ a δ can be computed.

$$R_Y(\theta):\begin{pmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{pmatrix}, \tag{1}$$

$$R_Z(\theta):\begin{pmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 0\end{pmatrix}. \tag{2}$$

To full use of this advantage, model needs to be enhanced by robotic arm positional sensors, which can be realized as optocouplers, connected to lab card [6].

An alternative to optocouplers is the application of the potentiometers, but their installation is more complicated and they require an analogue lab card. In addition to sensing of the joint positions, a black and white camera connected to a PC through a USB interface can be added, when placed on the gripper it can increase overall possibilities of the model.

III. Communication subsystem

To be able to move the robot each engine needs to be actuated with voltage in form of correct bites pair. To do such task, robotic arm control unit can be used, because it is connected to PC via USB interface. As programming language was chosen the object oriented C# language [7].

An open source library *LibUsbDotNet.dll* is used to control the device by USB interface. There was a problem with electronically unsigned drivers supplied with robotic arm - for x64 architecture operating system, but this library already includes the ability to work with signed or unsigned drivers and own drivers as well. Such drivers have to be manually added by user in a special application that also allows the user to view the details list of a device connected to the USB. Robotic hand can be connected and selected only if it's the *Vid* - vendor ID and *Pid* - Product ID are known. Those two numbers are used to identify the any device and can be found in the above-mentioned details list.

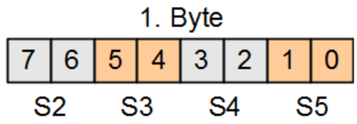

Fig. 3. Control byte pairs scheme for S2-S4 robot joint control.

Robot receives 3 control bytes in a row. The first control byte carries information for changing the orientation and movement of joints S2 to S4, specific pairs are shown in Fig. 3. The second byte is for control of the base rotation and last manages only LED lamp state on gripper. Control byte setting is based on specific bit pairs for selected subsystem, there are three possible positions – 00 means always stop, 01 is for positive rotation direction and 10 is for negative direction. LED lamp is controlled only by one bit, 0 means that the lamp is off.

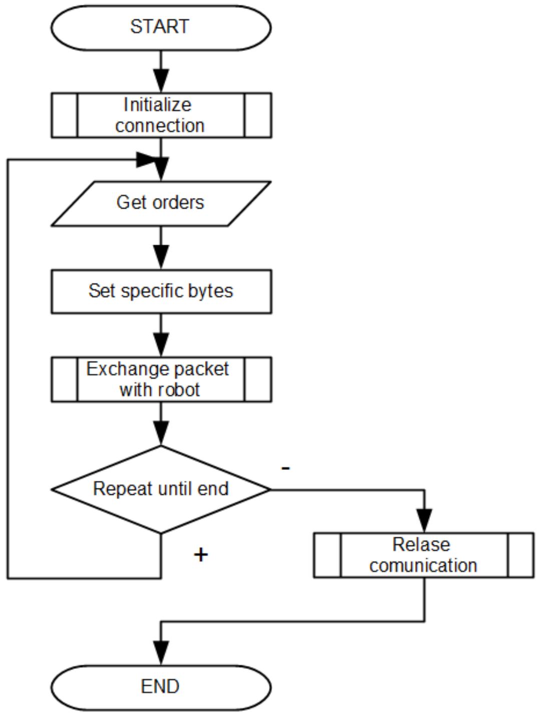

Fig. 4. Algorithm describing asynchronous communication between PC and robotic arm

Methods of created library enable sending a packet that consists of mentioned 3 bytes. A synchronous sending option sending data packets in periodic intervals was also used, but it only uses the methods for asynchronous packet sending.

Before the packet can be sent, a check if the device is connected to USB port needs to be done. Program browse the list of all connected devices and look up for robotic arm based on Vid a Pid. If such device is present in the list, the robotic arm is presumed as connected and it can receive control packets. This initialization process can be summed into block *Initialize Connection* in state-flow diagram show on Fig. 4. This state-flow diagram describes the algorithm of communication loop, namely get orders for joint rotation, set specific bits in control byte of packet and the block *Exchange packet with robot* represents sending the bytes in a row. A graphical user interface (GUI) as Windows Forms application, shown on Fig. 5, was designed for easy model use.

Fig. 5. Own graphical user interface.

IV. Graphical User Interface

GUI, that was designed allow asynchronous control of one robotic arm with commands for each subsystem. Information about subsystem state is displayed in table and status bar contains actual packet bit setting for each byte. For safety reason, there is also included a Total stop mode, which after activation immediately stops all subsystem movement and blocks sending of any next packets. Main purpose of this GUI is ability to direct control of robotic arm model by user and calibration tasks.

V. Experiments and Tests

Since this robot does not contain components that enable feedback control that are intended to be added in near future, experimental options are limited. An experiment of base and joint rotation was conducted, where the angular deviation of joint for specific time was measured. By this way it was able to determine exact time in milliseconds needed to move joint for desired angle. This experiment also revealed that the direction of movement change need to be compensated with extra time and can be counted how much time is needed. Linear characteristics for angular changes larger than 10 degrees were measured. However, the precision of this kind of measurements is insufficient and results were affected by battery power supply. This experiment will be repeated after adding the optocouplers and with stabilized power.

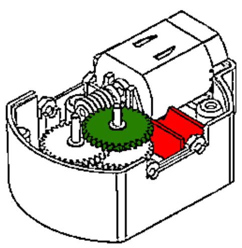

Fig. 6. Possible position of optical counters in robotic arm gearbox.

Optocouplers will be placed into gearboxes, to highlighted position shown on Fig. 6. and they will count the rotations of topmost gear, which is driven directly by DC engine. Gear will have six holes, which will suffice for desired accuracy. The rotations of this gear than can be converted into base and joint angular deviation.

VI. Conclusion

Despite high prices and limited offer, it is today possible to obtain a model of the robotic arm, which is, although it is not an industrial version, an affordable device for testing purposes of simple algorithms, and the general laws of physics. The aim of presented model is to use is options. The goal of this paper is to utilize the possibilities of the mentioned model for verification of complex control algorithms that use artificial intelligence techniques. However, before this algorithm can be practically tested, the model needs to be improved by adding sensors that enrich its options.

If the position sensors will be installed, robot could be used in trajectory tracking by direct and inverse task with use of its kinematic model [8]. As mentioned earlier, robot should have a camera mounted on gripper. It could be used for face or target tracking with use of computer vision methods. Neural networks can be used in motion and pattern recognition [9] [10].

Acknowledgment

This work has been supported by the Scientific Grant Agency of Slovak Republic under project Vega No.1/0286/11 Dynamic Hybrid Architectures of the Multiagent Network Control Systems (50%). This work is also the result of the project KEGA 021TUKE-4/2012 (50%).

References

[1] Mobile robot Khepera II - simulation approach. Center of Modern Control Techniques and Industrial Informatics [online]. [cit. 2013-03-11]. Available from: http://kyb.fei.tuke.sk/laboratoria/modely/khe.php

[2] ŠUSTER, P., JADLOVSKÁ, A.: Tracking trajectory of the mobile robot Khepera II using approaches of artificial intelligence / 2011.In: Acta Electrotechnica et Informatica. - ISSN 1335-8243. - Yr. 11, č. 1 (2011), pages 38-43.

[3] MASÁR, I. Inteligentný regulátor na sledovanie trajektórie mobilným robotom. Automatizace [online]. 2007, Yr. 50, num. 2, s. 80-85 [cit. 2013-03-10]. ISSN 0005-125X. Available from: http://www.automatizace.cz/article.php?a=1629

[4] MAPLIN ELECTRONICS. Robotic arm with USB pc interface: Assembly and Instruction manual [online]. 29 pages. [cit. 1.3.2013]. Dostupné z: http://www.maplin.co.uk/media/pdfs/A37JN_manual.pdf

[5] KROKAVEC, D. Spracovanie údajov v robotike. 1. ed. Košice: Rektorát Vysokej školy technickej v Košiciach, 1985. ISBN 85-629-85.

[6] MF624: Manuál. HUMUSOFT. [online]. [cit. 2013-03-11]. Available from: http://www.humusoft.cz/produkty/datacq/mf624/

[7] SHARP, J. Microsoft Visual C# 2008: Krok za krokem. 1. ed. Brno: Computer press, a. s., 2008, 592 s. ISBN 978-80-251-2027-9.

[8] PASTOREKOVÁ, L., MÉSZÁROS, A. BURIAN, P. Inteligentné riadenie systémov na báze inverzných neuronových modelov. AT& P PLUS. 2005, č. 7, pages. 53-58. ISSN 1335-2237.

[9] PHAM, D., LIU X. Neural networks for identification, prediction and control. 2. vyd. London: Springer-Verlag, 1995. ISBN 3-540-19959-4.

[10] KAJAN, S. Comparison of some neural control structures for nonlinear systems. Journal of Cybernetics and Informatics [online]. 2009, 11 pages [cit. 2013-03-10]. ISSN 1336-4774. Available from: http://kasr.elf.stuba.sk/sski/casopis/show_article.php?id_art=55&id_cat=9