Tomáš Škulavík

PLC-Based Fuzzy Control System for a Robotic Manipulator

Scientific Monographs in Automation and Computer Science

Edited by
Prof. Dr. Peter Husar (Ilmenau University of Technology) and
Dr. Kvetoslava Resetova (Slovak University of Technology in Bratislava)

Vol. 11

PLC-BASED FUZZY CONTROL SYSTEM FOR A ROBOTIC MANIPULATOR

Tomáš Škulavík

Universitätsverlag Ilmenau
2013

Impressum

Bibliographic information of the German National Library
The German National Library lists this publication in the German national bibliography, with detailed bibliographic information on the Internet at http://dnb.d-nb.de.

Author’s acknowledgement to Gabriela Chmelíková for translation.

This scientific monograph originated from the author's dissertation thesis defended at the Slovak University of Technology in Bratislava, Faculty of Materials Science and Technology in Trnava.

Reviewers:
Prof. Ing. Juraj Spalek, PhD.
Prof. Ing. Anton Vitko, CSc.
Prof. Ing. Alojz Mészáros, PhD.

Author’s contact address:
Ing. Tomáš Škulavík, PhD.
Slovak University of Technology in Bratislava
Faculty of Materials Science and Technology in Trnava

Technische Universität Ilmenau / University Library
Universitätsverlag Ilmenau
Postfach 10 05 65
D-98684 Ilmenau (Germany)
http://www.tu-ilmenau.de/universitaetsverlag

Production and delivery
Verlagshaus Monsenstein und Vannerdat OHG
Am Hawerkamp 31
D-48155 Münster (Germany)
http://www.mv-verlag.de

ISSN 2193-6439 (Print)
ISBN 978-3-86360-078-5 (Print)
URN urn:nbn:de:gbv:ilm1-2013100088

Titelfoto: photocase.com

Abstract

The submitted monograph deals with the design of robotic arm fuzzy control and its implementation in the PLC-based control system. In the introduction section, the robots and their control systems, as well as their properties and the possible solutions for the selected control tasks are analyzed. Then the author focuses on fuzzy control integration within the systems of the Siemens PLC company. Subsequently, the aims of the monograph and related subject matters to be investigated are defined. The main aim is to propose and implement the control for a robotic arm, whose mathematical model is unknown. The robotic arm, for which the control system has to be proposed, is described in detail. This section is followed by chapters focused on the design itself, its implementation and testing of the individual parts of the control system as a whole. The last chapter deals with analyzing the achieved results.

Key words

robotic arm, fuzzy control, PLC

Scientific contribution

The achieved results are summarized as follows:

- functional interface for the connection of the control system and the robotic arm,
- robot's control system on PLC base which determines it also for the control of more complex technologies,
- fuzzy control used in the manual control of the robot via the joystick,
- fuzzy control of the speed,
- fuzzy control of the position,
- design of the control system without using the mathematical apparatus,
- successful verification of the fuzzy control implementation into PLC system.

The proposed solution of the control system could be implemented also in real conditions, e.g. in the painting shop, where it is necessary to follow the stable speed and distance from the painted surface. Obviously, the interface to the robot's connection to PLC should be designed for the specific robot type. Similarly, the control program should be adjusted for this specific robot type. For the robotic arm with more joints the fuzzy systems should be complemented by the controllers. The author took this possibility during the design in consideration; therefore, the fuzzy controller was designed with normalized universes. By the growing number of controllers the time for computation grows as well, which can negatively influence the system dynamics. Therefore, by the design of the control system the author had to consider this issue as well. To ensure higher accuracy in the positioning the control system could be complemented by

MLTS (Multi Laser Tracker System), which could send the data on the effector’s position to the control system. Another possible implementation of the control system could be in the palletisation (or in the loading and unloading of workpieces of and into the production line or to the parts of the assembly or forming machines and devices), where not the preciseness of the effector’s motion trajectory but the final position achievement is emphasized. The designed control system is built on PLC base, which determines it to control the more complex technologies as well. The main contribution of the monograph is represented by the design and implementation of the fuzzy control of the robotic arm into the PLC based control system.

INTRODUCTION

The implementation of intelligent methods into the control mechanism has enhanced the control system design possibilities and now the use of complex mathematical apparatus only during the controlled system model development, does not have to be the only option. The monograph deals with fuzzy control and its implementation into the PLC for a robotic arm, whose mathematical model is unknown. In this case, the robotic arm model is considered to be the object of the control via which the designed fuzzy control is verified. In this introduction section, the robots and their control system, as well as their properties and possible solutions of the selected control tasks are analyzed. Then the author will focus on fuzzy control integration within the systems of the Siemens PLC company. Subsequently, the aims of the monograph and the related subject matters to be investigated are defined. The main aim is to propose and implement the control for a robotic arm, whose mathematical model is unknown. The robotic arm, for which the control system has to be proposed, is described in detail. The following chapters then focus on the design itself, its implementation and testing of the individual parts of the control system as a whole. The last chapter deals with analyzing the achieved results.

1. ROBOTS

In general, the robot is understood as a machine doing the same operations as a human, especially the motion and manipulation operations (1). Such a machine has to frequently acquire information about the environment in which it moves, as well as being capable to influence this environment both physically and mechanically. This is ensured by its motoric subsystem influencing the environment via its effectors. The effectors also ensure the robot's motion in the space. The robot has to be able to react somehow to the environment and to the changes within it which is ensured by its sensor subsystem. These systems are supervised by the cognitive subsystem, in which the decision making and main control activity take place. The subsystem hides the robot intelligence. The sensor system is divided into two parts. They are receptors reading the physical signals from the environment and converting them into suitable internal signals. The other part is represented by the data selection and processing system which selects the important information for the robot, e.g. reading the environment by a television camera and evaluating the shape and position of the object to be manipulated. The motoric system is also divided into two parts: the effectors executing the action/impact into the environment and the plans implementer, according to which the effectors are controlled, e.g. the robotic arm, servomechanisms and a control computer controlling the arm. The cognitive subsystem represents the supervising intelligent control. This subsystem carries out the deeper analysis of the information coming from the sensor system. This analysis includes perception and comprehension. The analysis requires that the robot should have an environment model built-in and the aims should be determined. Regarding this analysis, the environment model and the aim of the monograph is to ensure the robot's planning and solving of operations is carried out. The cognitive system thus closes the highest loop of K feedback necessary for the intelligent robot's behavior (2).

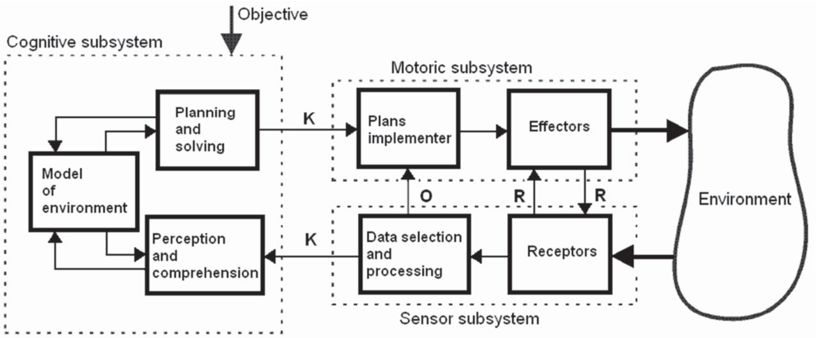


Fig. 1 *Block diagram of a robot*

There are feedback loops on the lower level between the sensor and motoric systems. It is the operation loop O, which ensures the execution of the planned task. Operation loops of robots are represented for instance by the loops of servomechanisms moving the robotic arm. The lowest control level is represented by the reflex loops R, which solves the basic simple issues similarly to the human reflexes, e.g. by burning. The loop "bumper – motor" which stops the robot approximating the obstacle can be another example. Similarly as with the humans, it is necessary, that the aforementioned subsystems are in a harmonic balance (2).

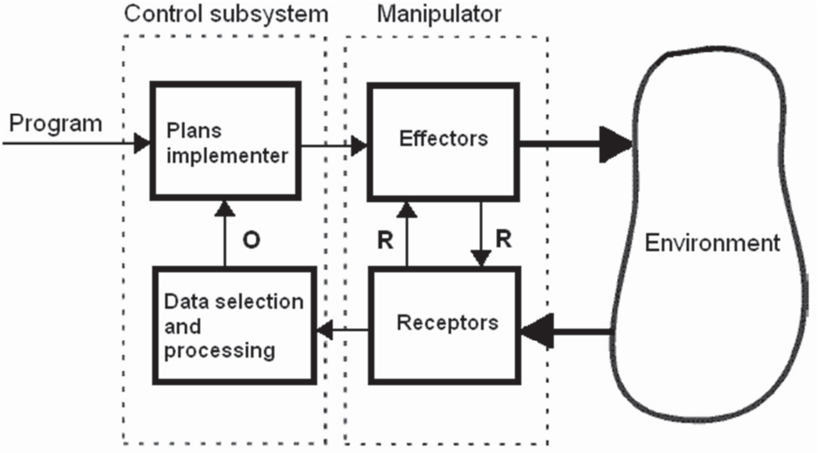


Fig. 2 Block diagram of an industrial robot

Industrial robots are not equipped with the intelligence of a cognitive robot. For the industrial robot the plan of an operation (program) is set by a human. The control subsystem (plan implementer and the block of data selection and processing) is formed mainly by the computers in various modifications. The effectors together with the receptors form the manipulator controlled by a computer (or by other electronics). In addition, all the subsystems of an industrial robot have to be harmonically balanced (2).

Classifying the robots into two groups based on their mechanical structure, is a very important parameter. The first group is represented by the robots with a stable frame – manipulators, and in the second group there are robots with a movable frame – mobile robots (1).

1.1 Industrial robots and manipulators

The mechanical part of the manipulator is most frequently formed by the arm and wrist with the gripper. The task of the manipulator is to ensure the positioning of the gripped object in the area. We know from mechanics that the position and orientation of the object is given by six data. They are more or less three values $[x, y, z]$ of the specified object reference point coordinates, in the basic Cartesian coordinate system and three angles $[\lambda, \beta, \gamma]$ of turning for a specified reference system which is firmly connected to the object regarding the basic coordinate system. The object has six degrees of freedom in the area (3). It is obvious, that the manipulator has to have at least six free and easily adjustable variables and six degrees of freedom so that the gripped object can be freely positioned. This is mechanically ensured by axes – joints which are driven (adjusted) by actuations. A lower number of joints than six also reduces' the manipulation abilities of the robot. The number of joints is thus a significant parameter of the robot (2). Figure 3 shows three basic kinematic concepts of manipulators and specifies their working area.

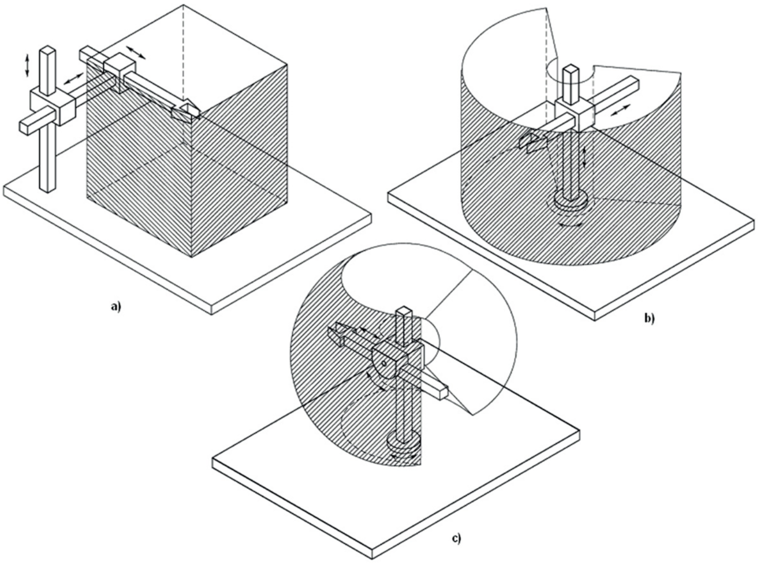


Fig. 3 *Basic kinematic concepts of manipulators a) Cartesian, b) cylindrical, c) spherical (1)*

Each arm in Fig. 3 is ended by a gripper, which can close and open. The manipulators shown in Fig. 3 have only three axes. If the task of the manipulator is to relocate and empty the glass of water, the manipulators with the concepts a) and b) would not be able to carry out the task. The manipulators with the concept c) would be able to execute the task, however, only via turning the arm over its body (2). Within the manipulation, each joint is connected with data on its setup – the joint variable. The joint variable is marked by the symbol q. The joint variables of the manipulators in Fig. 3 correspond with the coordinates of the known systems of coordinates; Cartesian, cylindrical and spherical. The aforementioned names of the concepts are derived from here. There are many kinematic concepts. Fig. 4 shows the SCARA manipulator a) and an anthropomorphic manipulator b).

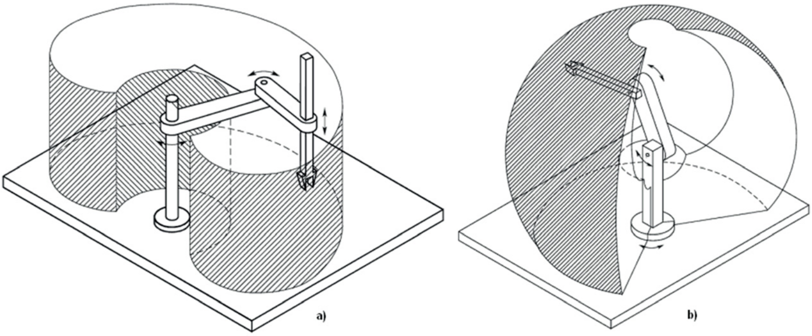


Fig. 4 *The SCARA manipulator a) and anthropomorphic manipulator b) (1)*

Different concepts have various advantages and disadvantages and influence the various practical properties of robots. More joints can significantly increase the robot's manipulation abilities. Fig. 5 shows the planar manipulator with six rotation joints.

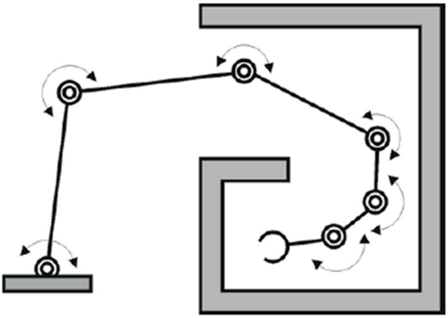


Fig. 5 *Planar manipulator with six joints (2)*

The arm can move in the plane. Obviously, due to the higher number of joints it can manipulate also beyond the obstacle. Despite the fact that it has six joints, it cannot freely manipulate with the gripped object in the area. That means that six joints is a necessity, however, it is not a sufficient condition for the complete manipulation of the object. Therefore, the correct placement of the individual joints is essential (2). Other limitations of the manipulator’s movement are caused by its limits and geometric dimensions. These limitations determine the working area of the manipulator (3).

The familiarity with the joints′ coordinates allows unambiguous determination of the manipulator end member value in the Cartesian area. If we mark the vector of joints′ coordinates by $q = [q_1; q_2; q_3; q_4; q_5; q_6]^T$ (for the robot with six joints) and the position vector of the robot’s end member, e.g. the gripper by $P = [x; y; z; \lambda; \beta; \gamma]^T$, then an unambiguous representation exists from the area of joints coordinates into the area of the Cartesian coordinates, which can be recorded as $P = f(q)$. This represents six equations, which can be constituted for the majority of manipulators via the common geometry command. The constitution of these equations is the direct task of kinematics (forward kinematics). Thus the robot movement can be programmed in the joints′ coordinates area and the robot carries out the related movement in the Cartesian coordinates. In addition, it is more natural for humans to determine and plan the movement in these coordinates. In some programming systems we investigate the task vice versa, i.e. we can calculate the joints′ coordinates values from the P position. This is an inversion task of kinematics (inverse kinematics). The task is more complex than the forward kinematics (2).

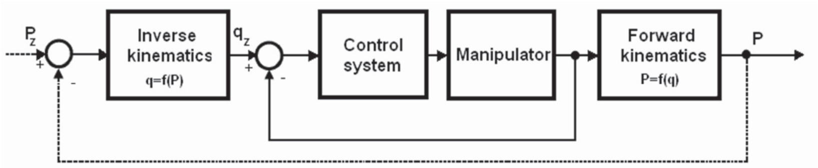


Fig. 6 *Block diagram of the industrial robot control*

The robot's trajectory is mostly programmed and saved as the joint's coordinates. Time and the data together with the required coordinates' q_z are saved in the robot's memory. In the program execution the control system ensures, that q(t) is practically identical with q_z (t). In the related course q(t) the forward kinematics is handled by the manipulator mechanism itself (2).

1.2 Techniques of industrial robots programming

The technique of robot programming or trajectory planning is crucial information in the industry. Sometimes, the data itself is referred to as the technique of robot programming. In general, there are three techniques of planning the robot trajectory (robot programming).

1 Direct programming (learning) - this can be executed in the two following ways:

a) The operator manipulates the arm and the wrist of the robot on the required trajectory at a required speed. The operator holds the working tool, which is also held in the robot's gripper and carries out the related operations with it. This should be subsequently repeated by the robot. The robot control system remembers the required movement in the format of the table of data q_z (t) and then executes the movement. There is also a disadvantage with this way of programming because the operator has to execute the programming movement perfectly accurately, since the robot repeats also the possible mistakes, e.g. the arm hitch (2).

b) The operator directs the robot into the required positions e.g. via the buttons on the programming panel. When the robot is in the required position, which can be set very precisely and independently at the time of its setting, the operator pushes the button and the robot remembers the position. The sequence of the required position data is saved in the form of a small amount of data q_1, q_2 ... q_n in the robot's memory. Before the robot activation it is necessary to complement this data with time data, and possibly also with information regarding which points should be connected in the area. This complementary data then determines how the movement will be carried out in real conditions. In any case the robot will pass the sequence of points - q_1, q_2 ... q_n. It is advantageous that the position data could be set very precisely and that there are only a few of them. On the other hand, the movement among these positions does not have to be known sufficiently (2).

Through the use of direct programming the inverse kinematics task is solved by the human together with the manipulator's mechanism via a very simple and natural technique.

2 Indirect programming – off line.

Using this technique of programming the programmed trajectory P_z (t) is in the form of curves in the area, e.g. according to the drawing. Time is the parameter of these curves and is determined by the technology procedure, e.g. in welding. The inverse kinematics is also solved via the off line technique and the data q_z (t) is used for the robot control (2).

3 Direct planning – on line.

Direct planning is similar to the previous technique with a difference in that the inverse kinematics has to be solved in real time. This method of planning is used when the robot has to carry out its movement according to the data from the sensors in the changing environment. For instance the robot has to grip the moving object with a prior unknown trajectory (2).

The techniques and control programming 1a, 2 and 3 are indicated by the abbreviation CP (Continuous Path). The technique 1b is indicated by the abbreviation PTP (Point to Point).

2. CONTROL SYSTEMS OF INDUSTRIAL ROBOTS

Fig. 7 shows the example of the robotic system configuration. The robots have an independent control unit ensuring not only their control but the communication with external systems as well.

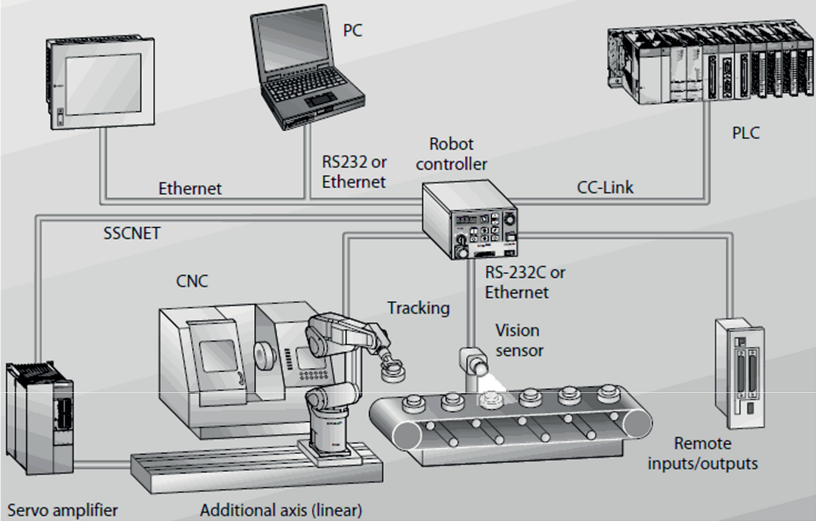


***Fig.** 7 Example of a robotic system configuration (4)*

The commercial control systems of industrial robots are special multiprocessor computer systems executing four basic processes allowing the robots´ integration into the automated systems (5).

2.1 Generating the movement trajectory and its monitoring

Each robot control system converts the digital signals from the higher level to the controlled arm movement via very precise calculations with a high distribution speed and communication of the movement commands for the individual axes executed by the servo-actuators of the joints. The robot control can be simply described as a calculation of the forces and rotations the servo-actuators has to make in order to generate successful implementation of the planned task. A number of control systems operating in real time still use the classical PID controllers and their modifications for the robot movement control. They are suitable in the situation where the robot control is based on Point-to-Point trajectory planning (PTP). However, without a significant programming effort most of them are unsuitable for the robots with Continuous Path trajectory planning (CP) (5). For their use it is necessary to understand the robot's dynamics. If for instance the robot manipulates using objects, whose weight is unknown or changes, the classical PID controllers do not meet the criteria for optimal movement and position control. In these applications the adaptive control is effective. Nevertheless, also in this case it is necessary to identify the mathematical description of the controlled system. Obviously, the algorithm technique used depends particularly on the type of activity executed by the robot. These algorithms are significantly different for the situation when the robot effector has to precisely copy the prior planned trajectory (e.g. laser welding) and also for the situation when it is only necessary to achieve the end position, where the accuracy of the executed trajectory between the initial and final points is not emphasized (e.g. palletisation) (6). Fig. 8 shows the position and speed controller with a detailed analysis described in (7).

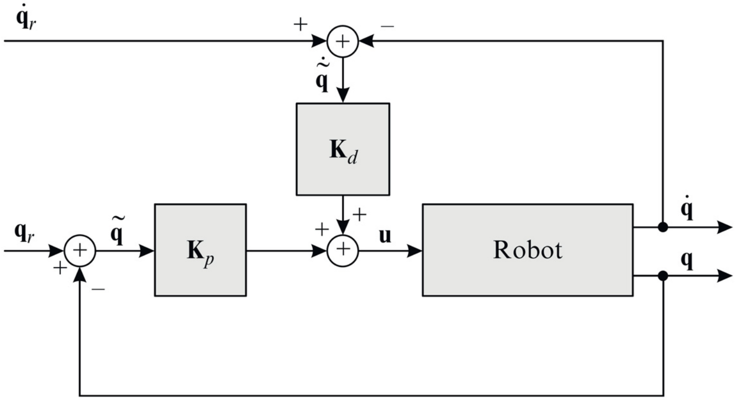


Fig. 8 *Controller of position and speed (7)*

The control algorithm can be written as follows:

$$u = K_P(q_r - q) + K_D\left(\dot{q}_r - \dot{q}\right), \qquad [1]$$

q_r is the required position,

q is the current position,

$\tilde{q}$ is the position deflection,

K_P is the position improvement,

K_D is the speed improvement,

$\dot{q}_r$ is the required speed,

$\dot{q}$ is the current speed,

$\tilde{\dot{q}}$ is the speed deflection,

u is the action impact in the form of forces or rotations the actuators have to execute.

These controllers have to be designed for each joint separately. The behavior of each controller is fully independent from the controller of the other robot joint mechanism. However, this controller does not consider the external forces caused by gravitation. Therefore, the control system is complemented by the gravitation compensation (7).

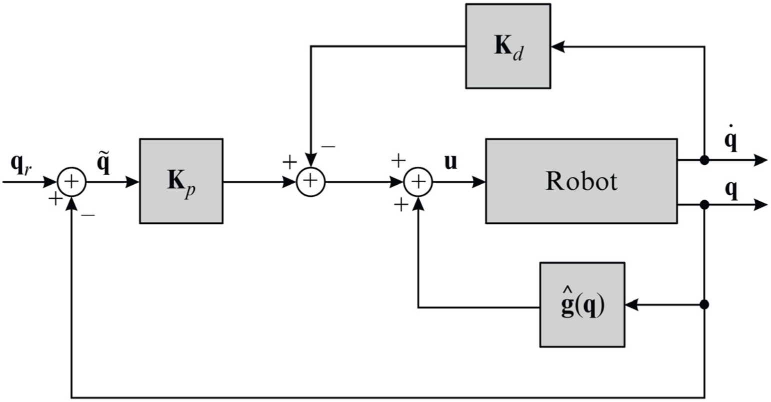


Fig. 9 *Control system with position and speed controller and gravitation compensation (7)*

The control mechanism is then as follows:

$$u = K_P(q_r - q) - K_D.\dot{q} + \hat{g}(q) \qquad [2]$$

$\hat{g}(q)$ is the model of gravitation effects.

Using gravitation compensation eliminates the load originating from error reduction caused by the gravitation effect of the controller. In this case errors in monitoring the trajectory are significantly lowered (7).

In previous cases the control was based on the internal coordinates, i.e. on joint coordinates. All positions, accelerations and speeds were recorded from the joint variables. However, mostly the end point movement of the effector is more important than the shifts of the individual joint connections. Fig. 10 shows the control system with the position and speed controller and the gravitation compensation controlled in external coordinates.

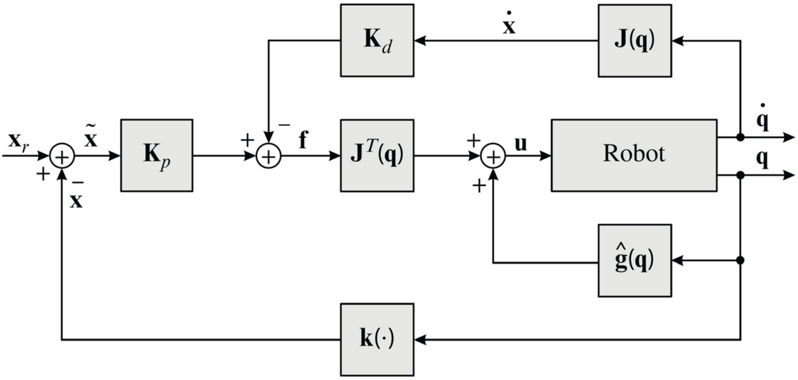


Fig. 10 *Control system with position and speed controller and gravitation compensation controlled in external coordinates (7)*

For this situation the control algorithm is as follows:

$$u = J^T(q)f + \hat{g}(q) \tag{3}$$

while

$$f = K_P \tilde{x} - K_D \dot{x} \tag{4}$$

where

J^T is a transposed Jacobian matrix describing the relationship among the joints´ moments and forces acting on the end effector,

$\dot{x}$ is the speed of the end effector,

$\tilde{x}$ is the deflection of the end effector position,

$\hat{g}(q)$ is the model of gravitation effects.

The previous control systems were mentioned as examples of the controllers in the robots´ control systems. There are many other modifications of these control systems, either with a PID controller or an adaptive or robust control system described in detail e.g. (1), (3), (5), (6), (7), (8). It is particularly effective to complement the control of robots by fuzzy logic, neural networks and genetic algorithms or their combinations. The use of these intelligent control methods are dealt in many publications, e.g. in (8), (9), (10), (11), (12), (13). They are mostly relating to theoretical outputs, which were not implemented in practice so far (e.g. in commercial manipulators´ control systems). The use of intelligent control methods has been effective mainly in the control of mobile robots (14). Fuzzy control and a classical PID control together could be effectively used in a hybrid control systems too. This kind of application was described in (15) and (16), where the hybrid fuzzy adaptive controller was used to control a pneumatic muscle actuator.

2.2 Integration into existing processes

It ensures coordination of the manipulator’s movement with process sensors or other process control devices. Process integration on the lowest level is possible by using digital inputs and outputs. For instance, the control system of the transport device can send a one-bite signal that it is ready to unload. The robot’s control system has to be able to read this information and execute the related operation. Therefore, several control systems have in-built functions of programmable logic automats. The coordination with sensors is also very important for the identification of obstacles (5).

2.3 Integration of human operation

For the robots´ control systems the interface for communication with humans is very important particularly from the point of setting and programming the robots. The control systems have mostly two types of interface. The first one is similar to the classical computer with a monitor and a keyboard for programming the robot off line. The second interface type is represented by the manual control and programming terminal (Teaching box). This terminal is portable and provides programming and controlling of the robot via buttons and lever control. These portable controls seem to be more efficient in positioning the robot since the control system can save the learned position in its memory and subsequently execute the required movement trajectory. The operator with practice can teach the robot a series of points which the robot will then use in the playback mode for the trajectory movement execution. Many applications utilizing the robots depend on the integration of human expertise, especially in the stage of programming for the correct planning and coordination of the robot's movements. These interfaces are efficient in areas with no obstacles and in the environment with no changes during the robot's programming and operation. They do not support direct human impact on the robot's movement change during its operation, or adaptation to a changing environment. More advanced interface techniques utilize behavioral programming, by which various specific robot behaviors are programmed into the control system (e.g. pick up the part; grip the part into the chuck, etc.). Then they are subsequently executed, while their specific movement parameters prescribed by the operator are determined by the supervising operator (5).

2.4 Integration of information

Many control systems support the functions of information integration via communication ports (RS232), by utilizing the in-built PC interfaces or in some cases with direct connection to the data bus of the control system. Through connection to the Internet they support the remote control and monitoring of the whole robotic cells. There are many software tools supporting these services. The graphical program environment LabVIEW produced by the company, National Instruments is one of the most used (5).

3. INTEGRATION OF FUZZY CONTROL INTO PLC SYSTEMS

The company Siemens produces the SIMATIC S7-300, a tool for designing the fuzzy controller. It is a Fuzzy Control++ software package utilizing the fuzzy logic and fuzzy control methods. The implementation of the aforementioned method in PLC devices comprises two phases. The first one is represented by the development and setting of the control system model in the environment of the related tool. The second phase is recording the saved data into the PLC device and subsequent correction of their values. The aforementioned algorithms are on the side of the PLC implemented within the functional blocks of the program being executed. For the fuzzy logic method FB30 a functional block is used (control SIMATIC S7-300/400). The aforementioned functional block has an allotted own instance data block comprising the memory elements utilized in calling the functional block, and the variables preserving the structure and parameters of the fuzzy system being modeled. Software implementation of the fuzzy functions represents an increased load for the CPU PLC module, which is reflected in the length of the program cycle duration. The time duration of executing the above mentioned functions depends mostly on the modeled system structure. They can include for example, the amounts of inputs or outputs or the base of rules size. The Fuzzy Control++ tool comprises the configuration application running in the Windows environment and in the runtime of modules, functions, and instances of data blocks for the SIMATIC S7-300 and SIMATIC S7-400 devices and libraries for WinCC visualization. With the configuration application the fuzzy system is designed and generated. The runtime modules evident by the name, process the generated data for the given system directly in the running operation of the device in the finish platform (17).

The following runtime modules are supported:

- SIMATIC S7;
- SIMATIC CFC;
- SIMATIC WinCC.

Definition of the fuzzy system comprises several steps; firstly, the definition of inputs and outputs followed by the assignment of affiliation functions and lastly, determination of the rules. By using this method we can determine the system block structure, fuzzification, inference mechanism and defuzzification. The limitations of the fuzzy model definition is determined for 8 inputs, 4 outputs, 9 affiliation functions for each input/output and 100 base rules at the most. It is necessary to add, that the larger the number of fuzzy system parameters, the longer the time for their processing in the PLC program, which can negatively influence the control dynamics. Definition of the rules is possible in three ways: via a table of rules, via a matrix of rules or directly via the fuzzy programming language FPL (Fuzzy Programming Language). The resulting direction of the control is determined by the rules and can be monitored in 3D. Once the system is developed, the generated data is recorded within the target PLC device via an MPI interface (Multi Point Interface), or via the superstructure interfaces PROFIBUS, or alternatively via an Industrial Ethernet (17). Fig. 11 shows the Fuzzy Control++ application with open windows for the controller design, 3D rules illustration, activities of the rules and an illustration of input and output signals.

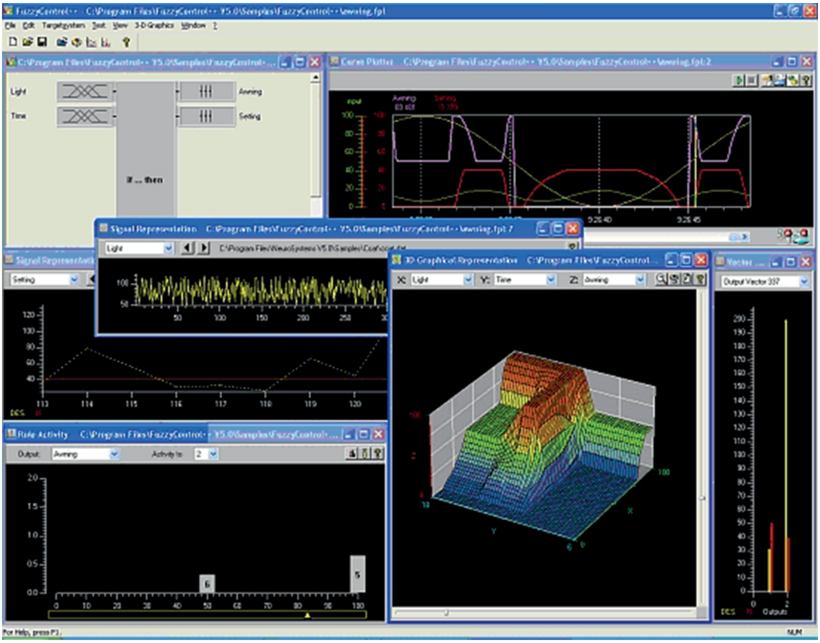


Fig. 11 Application of Fuzzy Control++ (18)

4. FORMULATION OF SUBJECT MATTER AND AIMS

In the previous chapters the author described the robots, manipulators and their control systems. The publications (1), (3), (5), (6), (7), and (8) unambiguously show that in the design of the robots´ (manipulators) control systems it is necessary to follow the mathematical description of the controlled system. Nevertheless, there can be a situation, in which such a model is not available, or the model acquisition is particularly demanding. This issue is the subject of the submitted scientific monograph.

4.1 Aims of the monograph

The aim of the scientific monograph is to design a manipulator control system with an unknown mathematical model. The notion of the investigation is to use the fuzzy controller (for the manipulator actuators control), which is advantageous since it can be designed without the precise mathematical description of the controlled system. To achieve the aim it is necessary to investigate several issues as follows:

- Analysis of the robot and selection of the system within which the robot's control system will be implemented.
- Design and implementation of the interface for connecting the robot to the control system.
- Design and implementation of the control system.
- Design and implementation of the fuzzy controller within the control system.
- Verification of the designed system via experimental measurements.

The above mentioned issues show that in order to achieve the intended aim it is necessary to investigate all the issues thoroughly as well as to include testing of partial solutions in the individual stages of the investigation.

5. FISCHERTECHNIK ROBOTIC ARM

The robotic arm for which the control system will be designed and which will also be used for its testing is produced by the company Fischertechnik. The robotic arm to be used is a manipulator model with three degrees of freedom. It is an anthropomorphic concept executed via three rotating joints. The rotation axis of the first joint is perpendicular to the axes of the other two parallel ones. The anthropomorphic structure is one of the most skilful structures due to the fact that all joints are rotary. On the other hand, the balance among the joint variables and the variables in the Cartesian coordinate system is completely lost (1). The actuation of the individual joints is ensured via three electric unidirectional small motors with a connecting voltage of 9V and current consumption of about 250mA depending on the load (350mA at maximum). In two cases the joints are connected to the gearboxes of the motors (M1 and M2) via the screw line. In the last case, the motor (M3) is connected to the joint via the cog-wheel. For each screw line the coding wheel which is the part of the incremental encoder (IRC1, IRC2 and IRC3) operating on the optical principle is fixed.
During the rotations the incremental encoder (impulse encoder, IRC) generates impulses. By calculating the impulses with a suitable calculator we are able to measure the angle of turning and via the impulses frequency measurement we can measure the rotations. These encoders have a drawback in the fact that through the connection, a loss of information regarding the current position is also lost. The setting of the outcome position of all three joints is ensured via the limit switches (S1, S2 and S3).

The robotic arm is equipped with a gripper, whose opening and gripping is driven by the unidirectional electric motor (M4) with a connection voltage of 9V and current consumption of about 200mA. On the gripper there are two limit switches for the identification of the complete opening (S4) and gripping (S5) of the empty gripper. The third switch (S6) is located on the inside edge of the gripper and provides the identification of the manipulated object that is gripped. Fig. 12 shows the working area of the robotic arm in 3D.

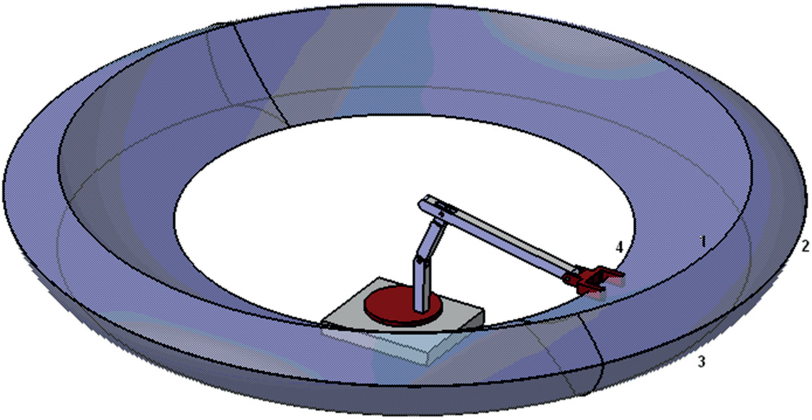


Fig. 12 *Working area of the robotic arm illustrated in 3D*

The working area of the robot was experimentally measured so that the individual joint actuators were led to the end positions and back to the outcome positions. The outcomes as well as the end positions are given by the physical limitations of the arm construction. Figures 13 and 14 illustrate the front and side view of the robot's working area together with the indicated boundary points.

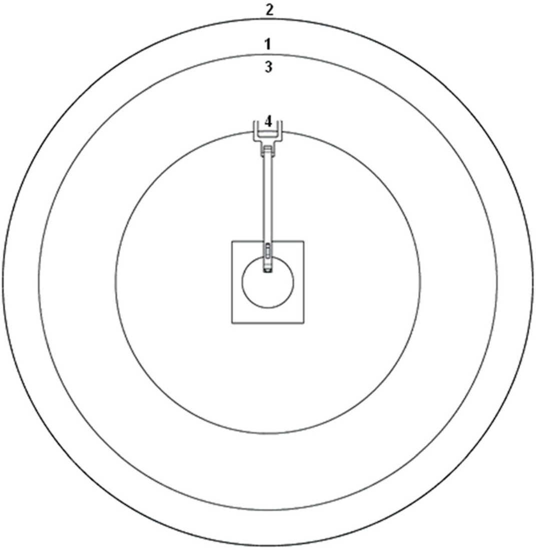


Fig. 13 *Front view of the robotic arm working area*

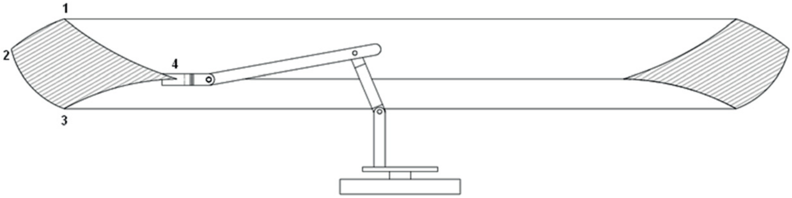


Fig. 14 *Side view of the robotic arm working area*

As previously mentioned, the outcome positions for all joint variables are recorded by the connection of related switches (S1, S2, and S3). The end positions of the individual joints are recorded via incremental encoders (IRC_M1, IRC_M2). The exception is represented by the joint driven by M3c motor (rotation around Z axis). Its outcome as well as the end positions can be recorded by the same switch (S3), since it can rotate in 360 degrees.

The incremental encoder connected to this joint is used only to investigate the current position of the joint (angle of motor rotation). Table 1 comprises the internal coordinates of the end positions indicated by numbers 1 to 4.

JOINT COORDINATES OF EFFECTOR'S END POSITIONS

Table 1

End point	Joint coordinate q1	Joint coordinate q2	Joint coordinate q3
1	0	0	0 to 600
2	186	0	0 to 600
3	186	190	0 to 600
4	0	190	0 to 600

The values of the joint coordinates from q1 to q3 correspond with the number of impulses measured from the related optical IRC (IRC_M1 to IRC_M3). The joint coordinates give the effectors position in the internal coordinates. The designed control system operates in these coordinates as well.

6. CONTROL SYSTEM PLATFORM SELECTION AND INTERFACE DESIGN FOR THE ROBOTIC ARM CONNECTION

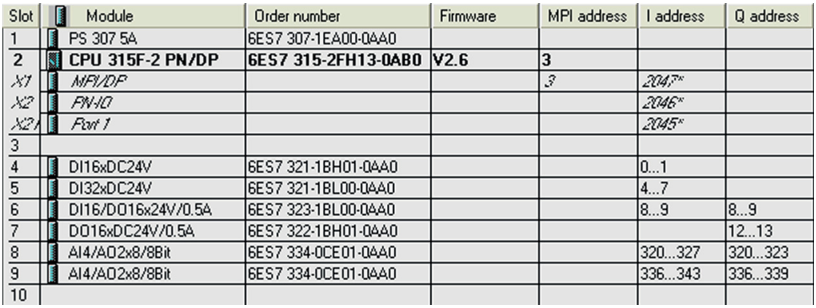
The company Fischertechnik manufactures several universal control systems for robot models. However, these systems are not suitable in the case of this study as the robots already have their own operation system which cannot be modified. Similarly, as with the commercial control systems of robots and manipulators, it is not possible to modify their operation system. It is possible to program the individual robot movements, but from the point of the motors control the system itself provides the user with no possibilities to adjust. Therefore, it was necessary to select such a control system platform, in which the whole own control system can be implemented. Programmable logic controllers have many advantages which could be useful for this purpose. They are widely used in many industrial applications (19), also in robotics and more recently, in much discussed safety systems (20), (21). The most convenient solution was to utilize the PLC series S7-300 manufactured by Siemens. Besides well modifiable hardware, it also has Fuzzy Control++ software support providing the fuzzy controller design, which is a great advantage.

6.1 PLC hardware configuration

The PLC hardware configuration conforms to the specific requirements for the selected system control. The universal solutions of the PLC series S7-300 were prepared in advance in our automation laboratory (22), (23). The available configuration of the PLC series S7-300 was adjusted so that it met the requirements of the control system hardware. These requirements conform to the given configuration of the robotic arm. The requirements are summarized as follows:

- Possibility to connect the signal from six limit switches (S1 to S6)
- Possibility to connect the signal from three optical IRC (IRC_M1 to IRC_M3)
- Possibility to control (turning on/off and direction) four DC motors (M1 to M4)
- Possibilities to control the rotations of three DC motors (M1 to M3)
- Possibility to connect the signal from switches and joystick switch (S7 to S9)
- Possibility to connect the signal from two joystick potentiometers (P1 and P2)

Fig. 15 shows the HW configuration of the PLC meeting the aforementioned requirements.



Slot	Module	Order number	Firmware	MPI address	I address	Q address
1	PS 307 5A	6ES7 307-1EA00-0AA0				
2	**CPU 315F-2 PN/DP**	**6ES7 315-2FH13-0AB0**	**V2.6**	**3**		
X1	*MPI/DP*			*3*	*2047**	
X2	*PN-IO*				*2046**	
X2	*Port 1*				*2045**	
3						
4	DI16xDC24V	6ES7 321-1BH01-0AA0			0...1	
5	DI32xDC24V	6ES7 321-1BL00-0AA0			4...7	
6	DI16/DO16x24V/0.5A	6ES7 323-1BL00-0AA0			8...9	8...9
7	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0				12...13
8	AI4/AO2x8/8Bit	6ES7 334-0CE01-0AA0			320...327	320...323
9	AI4/AO2x8/8Bit	6ES7 334-0CE01-0AA0			336...343	336...339
10						

Fig. 15 *Hardware configuration of the control system*

The order of the individual modules in the rack was followed according to the rules of HW configuration design for S7-300 stations. In the first position of the rack there is a PS 307 power supply with an output current of up to 5A, which has a sufficient reserve for the connection of the used modules as well as connected sensors. In the second position the processor is placed. In this case it is the CPU315F-2 PN/DP which has MPI/Profibus and Profinet communication interfaces. It has an advantage in that it is supported by the application of Fuzzy Control++. For communication with the PC (for the needs of programming and communication with Fuzzy Control ++) I selected the MPI communication interface with the address 3. The third position in the rack is left free. It was originally designed for the enhanced model; however, in this case it was not necessary. In the fourth position there is an SM374 module with 16 switches, which allows the simulation of inputs or outputs. In this case it was used as an input module. In the HW configuration, the SM321 module is in its place, which has 16 digital inputs with related addresses from I0.0 to I1.7. The module can operate as an HMI, where the use of switches makes it is possible to set various commands (e.g. switching between manual and automated regimes). In the fifth position there is the SM321 module with 32 digital inputs from I4.0 to I7.7. To this module the signals from the limit switches are connected (S1 to S9). The signals from the optical IRC (IRC_M1 to IRC_M3) are also connected to this module however; they are modified to 24V logic (via the interface). In the sixth position there is the SM323 module with 16 digital inputs with addresses from I8.0 to I9.7 and 16 digital outputs with addresses from Q8.0 to Q9.7. This module was left free and provides the reserve for the connection of the next technology signals. The seventh module in the rack is represented by the SM322 with 16 digital inputs with the addresses from Q12.0 to Q13.7. It provides the connection of relay coils, by which the motors turn on and off and their directions are controlled. SM334 are the last two modules, each of them has four analogue inputs with addresses from IW320 to IW327 for one module and from IW336 to IW343 for the other module. On each of these analogue modules there are also two analogue outputs with the addresses from QW320 to QW323 for one module and from QW336 to QW339 for the second module. The signals from the joystick potentiometers reading the control lever inclination are transmitted to the analogue inputs. Via three analogue outputs the rotations of three joint motors (M1 to M3) are controlled. The fourth analogue output provides the setting of the potentiometer's (references of voltage dividers) connection voltages.

The individual modules are not equipped with any special functions (e.g. high speed I/O), since they were not needed for this case. The sufficient reserve on the signal modules, which allows the system to control more complex technologies, is the biggest advantage. The basic processes described in Chapters 2.2, 2.3 and 2.4 are also ensured.

6.2 Design of interface for the robotic arm and PLC connection

Inputs and outputs of the robotic arm cannot be directly connected to the PLC signal modules due to various voltage and performance limitations on both sides. Therefore, it was necessary to design the suitable input-output interface allowing such a connection. The scheme of the interface and the printed circuits board were created using the Eagle V 5.0.0 program. The designed draft was applied to a one-sided photosensitive printed circuit board by a UV illuminating device. After subsequent chemical modifications, soldering of parts and testing, the printed circuit board was placed in the bottom section of the robotic arm model. In the interface design I had to consider the requirements for the control of the individual robot parts:

1 – Independent control of rotations/revolutions for each joint motor.

Since they are unidirectional motors with the stator from permanent magnetos, (excitation) their rotations depend on the size of the rotor connection. This type of motor cannot be controlled by a change of the exciting current. A PWM signal is one of the possibilities for controlling the rotations of these motors. The digital outputs of the signal module on the selected PLC could not generate a PWM signal with the necessary frequencies. The other possibility is represented by the use of the analogue outputs, by which it is possible to change the value of the output voltage from 0 to 10V, and thus change the motor rotations. It was necessary to ensure the PLC hardware configuration so that it had at least three analogue outputs. The performance limitation of the analogue outputs is problematic; therefore, it was necessary to design the circuit so that it was able to cover the current consumption needed (circa 350mA for 1 motor) for the motors. Fig. 16 shows the control part of the interface for one motor. Similarly, the other joint motors have connected the control part. The transistor Q20 ensures the aforementioned current coverage from the external source (V_M). Due to safety reasons, it was necessary to ensure the upper limit of the connection voltage of the motors was restricted to 9V. Therefore, the control interface circuit was designed so that by the voltage change on its input (analogue output of PLC AO4) ranged from 0 to 10V, and the voltage on its output changes from 0 to 9V (connection of X11-1 and X11-2 motors). This function is achieved by the connection of the optocoupler IL300, by which the galvanic motor and the external source are separated from the analogue input thus ensuring higher resistance to the signal module destruction.

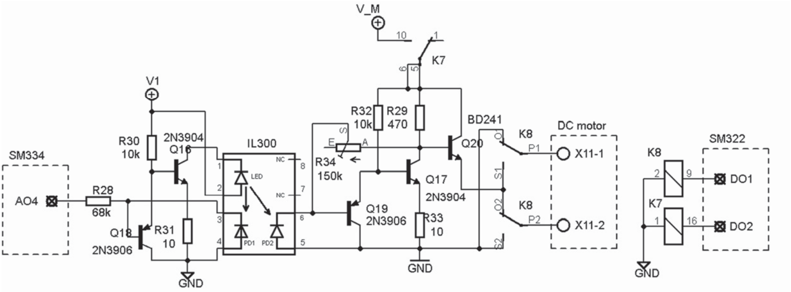


Fig. 16 *Control part of interface for motor control*

2 – Possibility of changing the rotation direction of all motors.

To be able to fully utilize all properties of the robotic arm, all motors controlling not only the joint mechanisms but also the end effector, had to have the possibility to control the direction of rotations. Since the unidirectional motors with permanent magneto are used as actuators, their direction can be changed by the change of the poles of their connection voltage. This was achieved by the utilization of the relay (K8) with two switching contacts. The K7 relay provides the connection and disconnection of the connecting voltage from the motor. Both relays are controlled by the digital outputs DO1 and DO2.

3 – Transferability of information from 5V logic to 24V logic

The robotic arm model is equipped with the sensors operating with 5V logic. To be able to process the information from the sensor by the control system, it was necessary to transfer the information into 24V logic. This was the information from the incremental encoders used in the servosystem. This requirement was met by the use of the optocouplers controlled by 5V, while their outputs switch was 24V. They are connected to the PLC digital inputs. The implementation of the optocouplers is illustrated in Fig. 17. The signal (5V logic) from the encoders is transmitted to the terminals X8 to X10 and subsequently it is amplified by the buffer (IC4A). The outputs from the amplifying buffer control the optocouplers, switching 24V to the related PLC inputs.

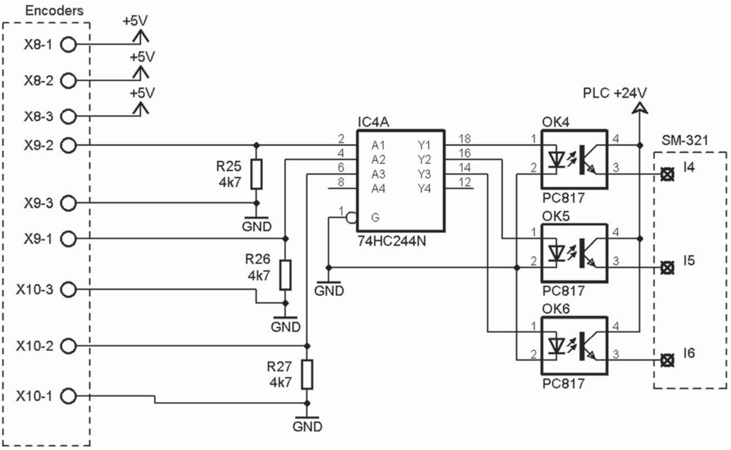


Fig. 17 *Converter of 5V logic to 24V logic*

For the switches used to detect the end positions it was not necessary to develop any control electronics, as they can directly switch 24V to the related digital PLC inputs.

4 – Possibility of manual control via a joystick

To allow convenient control of the robotic arm in the manual regime, it was necessary to add the lever controller (joystick) to the control system. The mechanical and electric construction of the joystick has to meet the requirements for controlling all the arm functions, i.e. that the joystick lever inclination in any direction has to be fluently recordable. Also, it has to be equipped with auxiliary sensors (gripper control). The selected joystick (the only available one) meets these requirements. The wipers of two potentiometers are connected to its lever. One records the lever inclination up and down, the other the inclination from left to right. The joystick is equipped with three switches however it does not have any universal interface able to connect to the PLC. Therefore, the individual elements were connected directly to the signal PLC modules. The wipers of the potentiometers were connected to the analogue inputs and the sensors to the digital inputs. Fig. 18 shows the connection of the joystick control elements to the signal PLC modules.

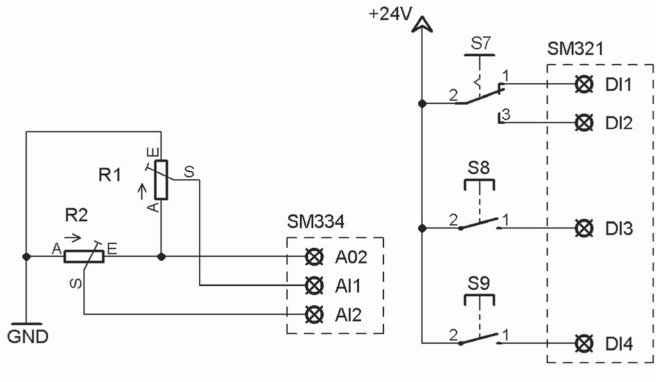


Fig. 18 *Connection of the joystick control elements to the signal PLC modules*

As shown in the figure above, the potentiometers (R1 and R2) are connected as the voltage dividers. The size of the divider connecting voltage can be set via the analogue output. In the control program this voltage was set to 10V. The outputs from the dividers are connected to the analogue inputs (AI1 and AI2). The value of the voltages on the analogue inputs then change depending on the joystick lever inclination. The sensors and one switch directly switches the 24V to digital PLC inputs.
Fig. 19 shows a block diagram of the robotic arm and the control system connection. As the HMI block the aforementioned joystick is used.

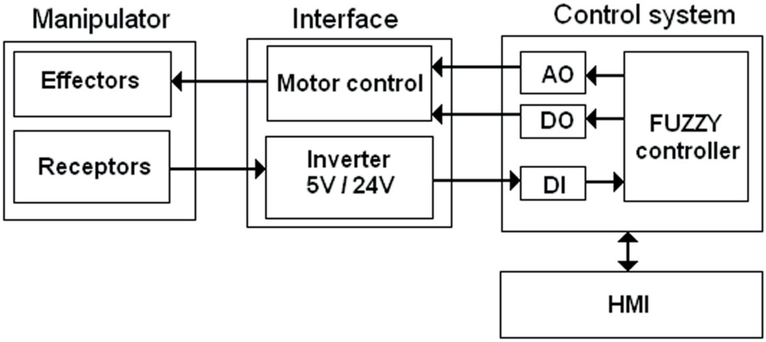


Fig. 19 *Block diagram of the robot and control system connection*

Before connecting the interface to the PLC the functions of the individual parts were tested by connecting the signals (to the interface inputs) from the laboratory connecting sources DF1731SB3A. The output signals from the interface were measured by a WENS860 two-channel oscilloscope. One of the tests was to connect at voltages within the range from 0V to 10V to the control circuit input for the motors (Fig. 16, terminal AO4) and on the output (terminals X11-1 and X11-2) the voltage change was monitored by the oscilloscope. With the correct operating connection and the K7 relay switched on, this voltage changed from 0V to 9V. The next step was to test the poles on the terminals X11-1 and X11-2. By switching on the K8 relay, the voltage on the monitored terminals changed the poles. Similarly, the other blocks and their functions were tested. The same testing ran also with the load, i.e. with the connected robotic arm. The difference was in the functionality of the individual interface blocks, which could also be confirmed visually (the correct direction of the motor rotation, speed change, stopping, gripping, etc.). These tests proved that the individual parts of the designed interface operate correctly and could be connected to the signal PLC modules. Subsequently, the interfaces together with the robotic arm were connected to the PLC. Using the application of SIMATIC Manager in the hardware configuration, similar tests were conducted with a slight difference as the outputs from the interface were monitored directly in the hardware configuration. The difference involved use of the Monitor/Modify function which allows changing of the individual values on the signal modules outputs, and at the same time allows monitoring of changes on the signal modules inputs. The verification of the joystick functionality was also one of the tests. The registered address of QW338 and the value of 7EF4 was recorded (value in HEX), which corresponds to the voltage of 10V. This voltage operates as the connecting voltage of the potentiometers connected to the joystick control lever. After the joystick lever inclination in the registers IW338 and IW336 the value changed according to the inclination direction and size. Similarly, the other functions of the interface and joystick were verified, e.g. the value in the related registers after monitoring the individually pressed switches. Once all the tests and verification of the correct interface functionality together with the connected arm and joystick to the PLC were carried out, it was necessary to design the control program.

7. DESIGN AND IMPLEMENTATION OF THE CONTROL PROGRAM

Before design of the control program it was necessary to specify the functional requirements the control program had to meet. These requirements were partially derived from the requirements for the interface (control of direction, speed of motors rotations, etc.). Other requirements for the control program were derived from the control process itself (manual mode, automated mode, positioning, etc.). The most important requirements for the control program are summarized as follows:

- The manual regime must be equipped with the possibility to control all functions of the arm via the joystick.
- The change of speed in the manual mode depends on the joystick lever inclination.
- The control program must ensure stopping of actuators in the end positions not depending on the joystick lever inclination, or calculated action impact of the controller in the automated mode (in the case that the controller is not designed correctly).
- The speed must be controlled by the simultaneous load compensation (by using the fuzzy controller).
- The position must be determined (by using the fuzzy controller).

The last two points include the use of the fuzzy controller, whose design will be described later. Fig. 20 shows a block diagram of the control system.

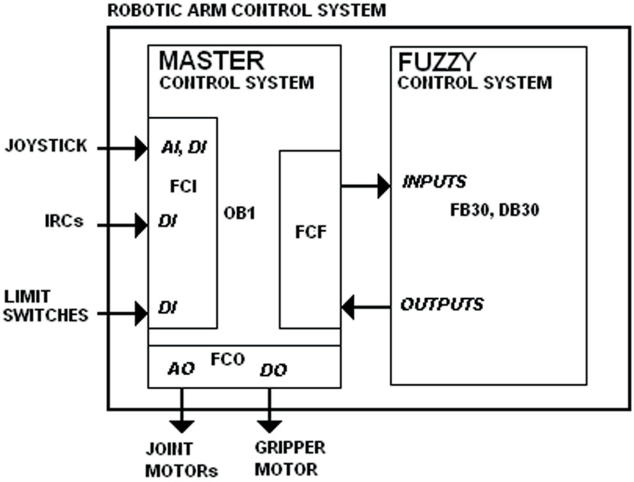


Fig. 20 Control system of the robotic arm

The SIMATIC Manager application and Step7 software were used to develop the control program. The hardware PLC configuration has already been described in Chapter 6.1. The programming device (PC) in this case uses an MPI interface for the communication with the PLC. The PLC has the address 3 and the programming device has the address 2. No other specific settings were needed. Before the programming a table of symbols was prepared (its part is illustrated in Fig. 20), in which the individual registers were assigned their symbolic titles to ease the programming and ensure the individual parts of the program were more comprehensive. In the following section, the functional part of the control program according to the aforementioned requirements and its implementation in the LAD programming language will be described. Every development phase was finalized by testing the implemented functions. Finally, the development was completed by conducting system integration tests which were carried out in accordance with e.g. (24).

7.1 Control of the gripper via the joystick

The robotic arm is equipped with a gripper driven by the M4 motor. The S5 and S4 switches provide signaling of the complete gripping and opening. The S6 switch provides the detection of the object gripping. The control program was designed so that it is possible to close or open the gripper via the lever control without the necessity to release the USE button on the joystick lever (located on the front side) at the point of achieving any of the end positions. The control program monitors the signals from the limit switches, and after achieving any of them, it turns off the gripper actuator. The repeated actuation is possible only after changing the M4 rotation direction (gripper actuator). The rotation direction is selected by moving the switch into the position CATCH or RELEASE, located on the back edge of the joystick lever. Similarly, the detection of the object gripping is operated. If the control program records gripping of the object, then the S6 button located on the internal wall of the gripper jaw turns off the gripper actuator. The gripper actuator can be repeatedly activated by a change in the M4 motor rotation direction, particularly in the case if it is necessary to release the gripped object. Fig. 21 shows the implementation technique of the gripper in LAD.

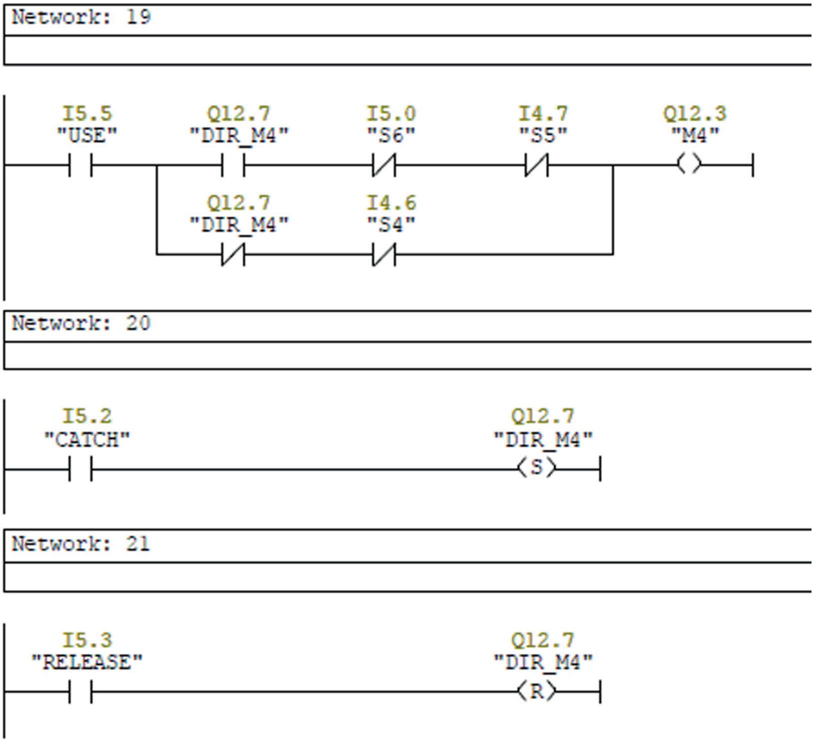


Fig. 21 *Implementation of gripper control in LAD*

7.2 Control of joint actuators with the joystick

To implement the functions for controlling the joint motors with the joystick it was necessary to solve the issue of controlling three motors with two directions of the joystick lever inclination. The issue was solved by using the remaining button (S9) located on the top of the joystick lever. By inclining the lever to the left or to the right it is possible to control the M3 motor, i.e. the arm effector rotation around the Z axis. By inclining the lever forwards and backwards, it is possible to control with the M1 motor the effector, and shift forwards or shift backwards in the X axis direction. By inclining the lever forwards and backwards and simultaneously holding the S9 button located on the top of the joystick lever, it is possible to control the M2 motor and to lift the arm effector (movement up or down) in the Z axis direction. Furthermore, it was necessary to use the calculation of the signal from the lever inclination to control the voltage of the related motor. If the joystick lever is in the balanced position (medium position) in the IW338 register, the value is 9865 and in the IW336 register, the value of 5888 is recorded. These values as well as the following recorded are introduced in the decimal format after calculation from the hexadecimal format. By shifting the inclination of the lever from the balanced (medium) position to the upper position, the IW338 register value changes from 9865 to 0. After the inclination of the lever moves to the bottom position (by pulling to self) the value changes from 9865 to 32500. By shifting the inclination of the lever from the balanced position to the left, the value in the IW336 register changes from 5888 to 0. By shifting the lever inclination to the right the value in the IW336 register changes from 5888 to 0. Since the joystick construction in the medium position is quite unstable, it was necessary to define the insensitivity zone so that the motors are not activated by the smallest touch of the lever. Fig. 22 shows the values of the IW336 and IW338 registers related to the ultimate and medium lever positions with the indicated insensitivity zone.

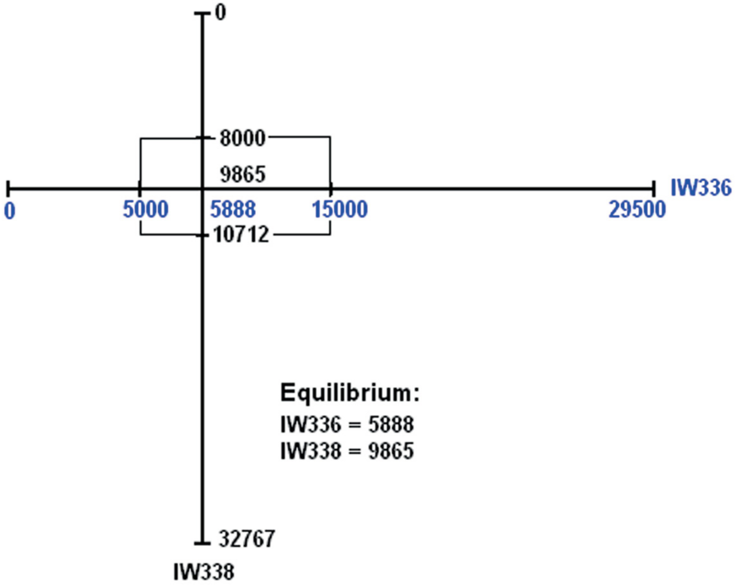


Fig. 22 *Values of the IW336 and IW338 registers with the indicated insensitivity zone*

The transmission functions of the lever inclination to the control voltage of the motors are created from the following equation:

$$y = a.x + b \qquad [5]$$

where

y is the calculated control voltage of the motor,

x is the value related to the joystick lever inclination,

a, b are transmission constants.

It was necessary to calculate the transmission constants for the individual intervals corresponding to the lever inclination from the balanced position.

Fig. 23 shows the transmission characteristics for the M1 and M2 motors by shifting the lever inclination upward. The motor selection is carried out using the S9 button.

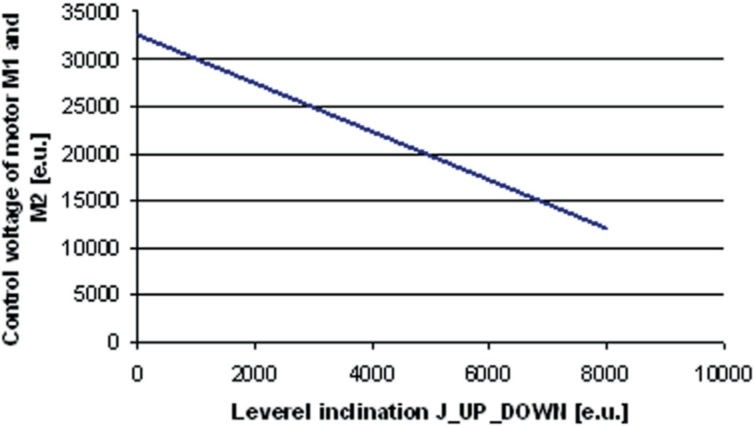


Fig. 23 *Conversion characteristics of the joystick for the upward inclination interval*

Calculation of the conversion constants by shifting the joystick inclination upward is as follows:

$$y = a.x + b;$$

for $x=0 \wedge y = 32500 \rightarrow$

$$32500 = 0.a + b$$

$$b = 32500$$

for $x=8000 \wedge y = 12000 \rightarrow$

$$12000 = 8000.a + 32500$$

$$a = -2.5652$$

For other lever inclinations, the conversion constant calculation procedure was carried out using the same technique. For the downward inclination (Fig. 24) the conversion constants have the values of a = 0.9295; and b = 2043.196.

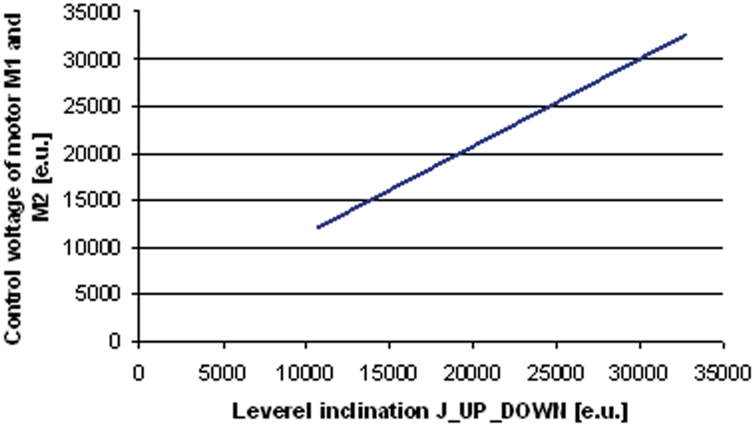


Fig. 24 *Conversion characteristics of the joystick for the downward inclination interval*

For the inclination to the left (Fig. 25) the conversion constants have the values a –4.1; b = 32500. For the lever inclination to the right (Fig. 26) the conversion constants have the values a = 1.4138; b = -9207.

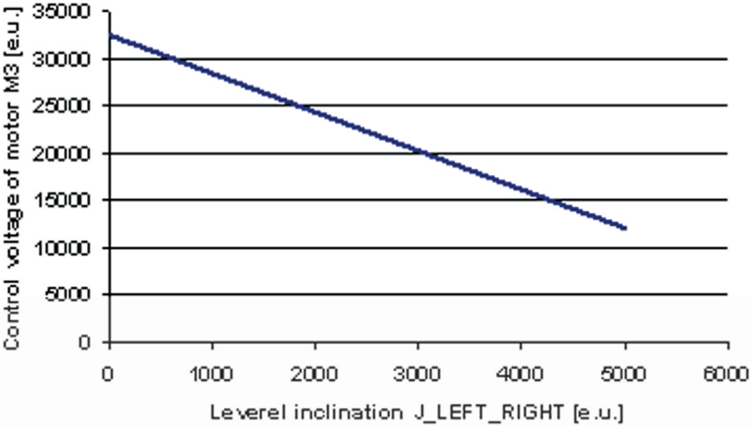


Fig. 25 *Conversion characteristics of the joystick for an inclination interval to the left*

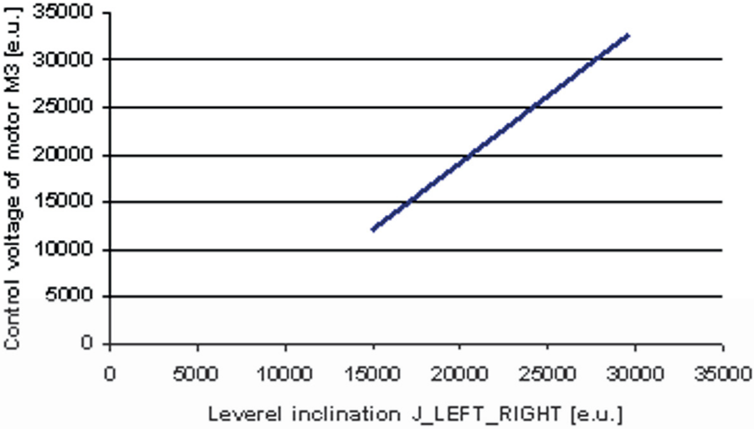


Fig. 26 *Conversion characteristics of the joystick for an inclination interval to the right*

The aforementioned calculations show that the control voltage value is in the range from 12000 to 32500. The value of 32500 corresponds to 10V. The value of 12000 was investigated by experiment; the motor at this value has the lowest rotations. This value corresponds to the voltage of approximately 4V. The calculated conversion constants a and b were used in the conversion functions implemented into the control system. As aforementioned, the conversion function is written by the relationship (5). The calculation of the joystick lever inclination to the related motor control voltage was implemented into the control system. The function inputs were formed from the a and b constants, and from the related inclination register value. The control voltage of the related joint motor is the function output. The function implementation in LAD is shown in Fig. 27.

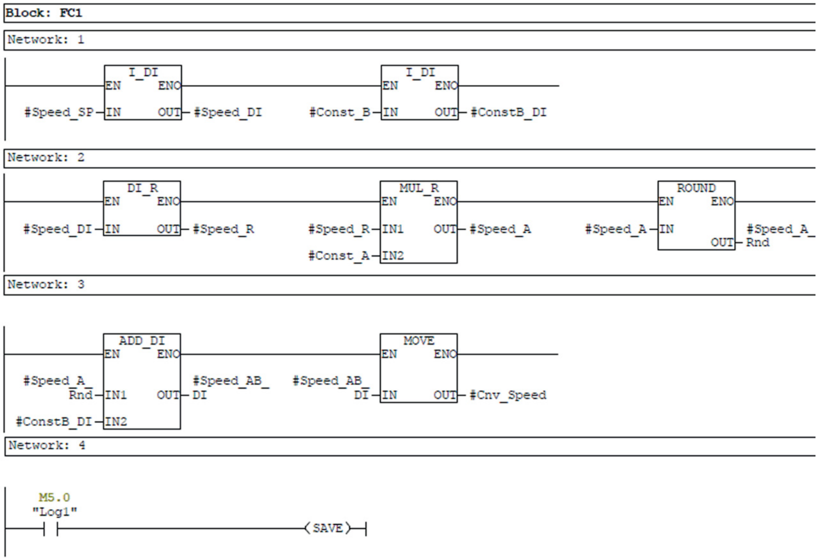


Fig. 27 *Implementation of the conversion function in LAD*

In network 1, the value of the joystick inclination is (Speed_SP) converted together with the constant b (Const_B) to the Double Integer data type. The conversions to various data types are necessary so that the functional blocks supporting only the selected data types can be used. In some cases it is essential to convert the variable through several data types until the required data type is achieved. In network 2 the multiplication of the joystick inclination value is carried out (Speed_SP converted to the data type Real) and conversion constant a (Const_A). In this network the conversions to necessary data types are carried out. In network 3, the addition of the previous network result to the conversion constant b is carried out, which results in the control voltage subsequently converting to the required data type. Network 4 saves the result value of the function operation, so that it can be processed in the main block of the control program (OB1). The described function is used for all intervals of the joystick lever inclination. The input and output functions are set according to the joystick inclination direction so that the control voltage is calculated correctly for each joint motor.

It is not only the speed of the joint motors rotation that depends upon the joystick lever inclination. The inclination also impacts upon the direction and ability to turn on and off. As a result, the control program was designed so that the motors do not activate if the joystick lever is in the balanced position or in the selected zone of insensitivity. If the lever is inclined so much that the value of any potentiometer does not fall into the zone of insensitivity, the voltage proportional to the lever inclination size is delivered to the motor. In this case, if the related outputs are indicated by the symbols M1 to M3, logic 1 is followed, which causes the switching of the relay contacts controlling the turning on and off of the motors. The voltage polarization of the motor depends on the joystick lever inclination direction. The voltage polarization of the motor is controlled by the outputs indicated as DIR_M1 to DIR_M3. If the relays controlling the turning on and off of the motors are active, and if the related voltage is delivered to the motor, then the motor rotation direction depending on the logic value, changes to the aforementioned outputs. The directions of the motors rotation corresponding to the joystick lever inclinations were programmed so that the robot effector control was intuitive, i.e. by shifting the joystick inclination away from one's self, the effector moves downwards or it shifts forward according to the selected joint motor (status of button S9). By shifting the lever inclination towards one's self, the effector moves upwards or it shifts backwards. By shifting the lever inclination to the left the arm turns around the Z axis clockwise. By shifting the lever inclination to the right the arm turns anticlockwise. Fig. 28 illustrates the control implementation of the direction and turning off and on of the motor depending on the inclination direction and joystick lever position.

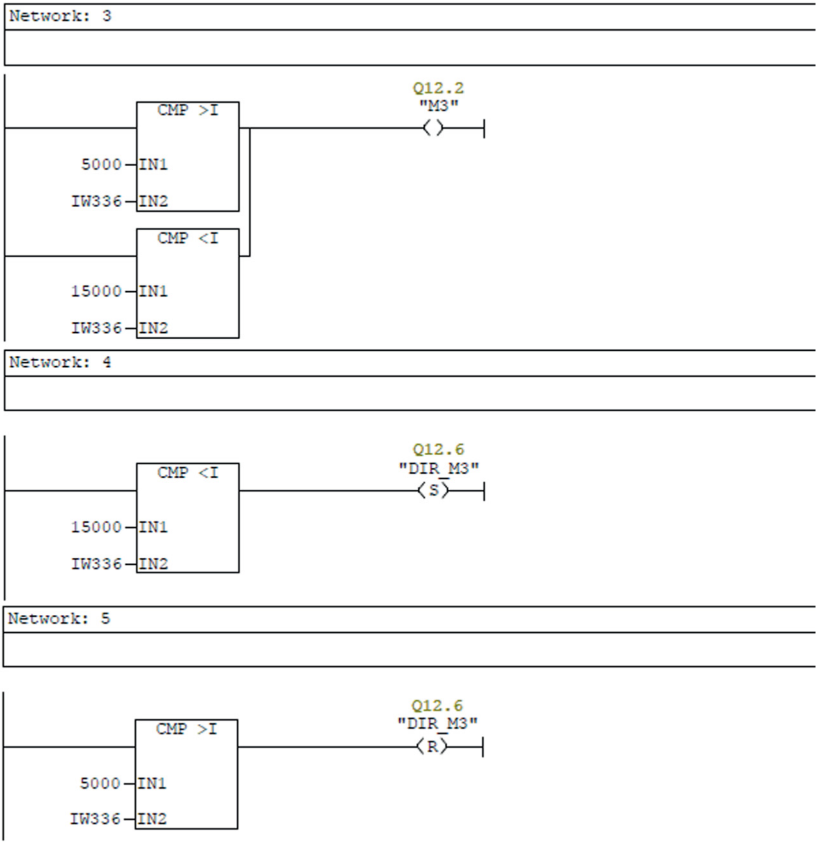


Fig. 28 *Control implementation of the motor rotation direction and turning on and off*

In network 3 the current inclination value to the left or to the right is compared to the value of the selected insensitivity zone. If the current value of the IW336 register is in the selected insensitivity zone, the relay controlling the motor’s on and off switch is deactivated. In this case, it is not possible to deliver any voltage to the motor. If this compared value is beyond the sensitivity zone, the relay controlling the motor's on and off switch is activated. If the voltage delivered to the motor is higher than 4V, the M3 motor starts to rotate. In network 4 the current inclination value of the joystick is compared to the upper border of the insensitivity zone for the inclination to the side. If this value is higher than the upper border of the insensitivity zone, it means that the joystick lever is inclined to the right and the relay controlling the direction of the M3 motor rotation is turned on. The arm rotates anticlockwise. In network 5 the current inclination value is compared to the bottom border of the insensitivity zone for the inclination to the side. If this value is lower than the bottom border of the insensitivity zone, the joystick lever is inclined to the left and the relay controlling the direction of the M3 motor rotation is deactivated. The arm rotates clockwise. Similarly, the control for the M1 and M2 joint motors was implemented.

7.3 Recording the internal joint coordinates and identifying end positions

In order to find the effector's position in the format of the internal coordinates, and thus ensure the end positions are followed, it was necessary to implement a function capable of processing the signal from the optical IRC (IRC_M1 to IRC_M3) and limit the switches (S1 to S3) into the control system. A disadvantage exists relating to the incremental encoders, as their outputs are represented by impulses, which do not assign the precise position but only the change of position. By reading them we can measure the distance passed, and thus also the current position, or the angle of rotation. Using the absolute encoders each position has its own value expressed in the format of Gray code, or BCD code. Therefore, finding the rotation angle or position is less demanding which could reflect also in the size of the function in the control program. The implemented function processing the signal from IRC_M1 and providing information about the current joint coordinate q1, is shown in Fig. 29.

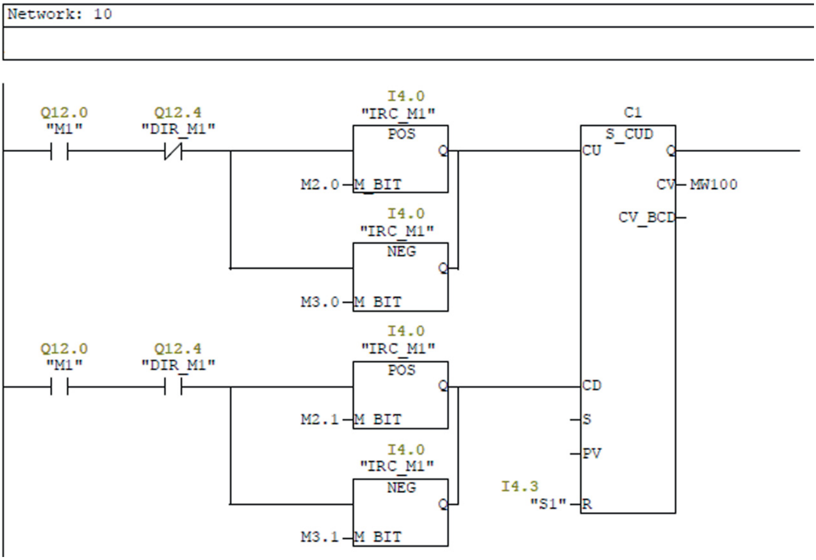


Fig. 29 *Function for finding the current value of the q1 joint coordinate*

In the function illustrated in Fig. 29, functional blocks were used to detect the rising and falling edges of the monitored signal (POS and NEG). Due to this fact, it is possible to process the rising and falling edges of the signal from IRC. The number of impulses per rotation increased and thus improved the detection of the current position (the angle of the M1 motor rotation). Such a signal adjustment is necessary only for this type of optical

IRC, as large gaps exist on the coding wheel. For commercially available IRC, this technique of using increasing impulses is not necessary. The function was introduced so that the calculator C1 is cleared with each achievement of the initial position recorded by the end sensor S1. If the arm moves in the direction to the initial position, the value of the joint coordinate saved in MW100 register increases. If the rotation direction changes, the calculator starts to subtract the rising and falling impulses and the value of the joint coordinate start to decrease. Similarly, the functions for processing the signals from IRC_M2 and IRC_M3 were implemented. The values of joint coordinates are also used in the functions providing the end positions are detected and emergency occurrence is avoided. Fig. 30 shows the implementation of the aforementioned function in LAD for the M1 joint motor.

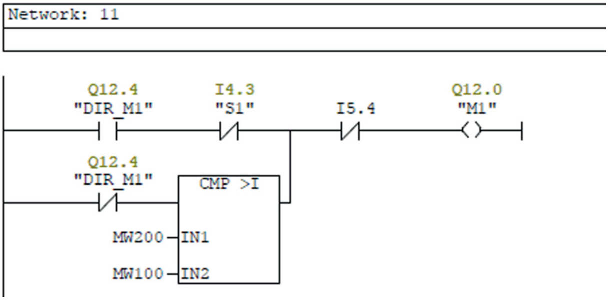


Fig. 30 *Detection of end positions and elimination of emergency occurrence*

The function shown in Fig. 30 prevents the shift backward of the arm beyond its initial position recorded by the limit switch S1. After achieving the position S1 blocks the motor M1 activation, if its rotation direction is not changed. It does not allow a shift forward of the arm beyond its value in the MW200 register. In this register the coordinate of the end position is saved (the value of 186). If the joint variable q1 achieves the value of 186, the M1 motor turns off. Turning it on is possible only by changing its rotation direction. Similarly, the functions for detecting the end positions and functions for the elimination of emergency for other joint motors were implemented (M2 and M3).

All aforementioned functions were similarly designed also for the automated regime, the only difference is that the robotic arm is not controlled by the joystick, but it executes the prior programmed movements. The functions for the manual and automated movements are very similar in terms of functionality, however the conditions are different upon which the individual functions are called. As a result, the option to select the automated or manual regime was introduced into the main programming block. The block allows switching between these two modes by means of a switch with the address of I0.0 located on the SM374 signal module.

All functional properties of the manual regime were tested after its implementation into the control system. The tests were carried out visually and by measurement. The correct rotation and the speed change of the individual joint motors during the change of the joystick lever inclination were tested visually. Similarly, the robot gripper control was verified visually. The end positions detected not only for the joint motors but also for the gripper motor were tested visually as well as by measurement. Once the individual motors achieved their end positions the voltage was measured on the individual motors using the Metex M-3270D digital multimeter. After the end position was achieved and the joystick lever position remained unchanged the motor voltage should fall to 0V. Visual control would not be sufficient in this case. If the control system would operate incorrectly and the arm exceeds the end position (even negligibly) it would stop due to physical limitations. It could be mistaken that the arm was turned off by the system after it achieved the end position. The connected voltage however, would destroy the motor after a while. Therefore, it was necessary to measure the connecting voltage on the motors as well. The functionality of the detected joint coordinates´ was also verified. The control program was switched in Step7 to the online mode and the change of values of the individual joint coordinates depending on the effector movement was monitored. Several subsequent times the individual joint motors were controlled up to the end positions, while the initial and end coordinates were also monitored. The conducted tests proved that the manual regime functionality was correct. In this phase the automated regime could not be verified satisfactorily, as the controller for the joint motors control was not yet implemented. Therefore, only the functions for detection of the end positions and emergency elimination as well as functions for the detection of joint coordinates´ values were tested. To carry out this testing the control voltage was fixed in the individual registers because in the automatic mode it is not possible to control the motors via the lever control.

8. DESIGN AND IMPLEMENTATION OF THE ROBOTIC ARM FUZZY CONTROL

Several publications deal with robot control systems and their programming, either in terms of the classical theory of the control ((1), (3), (5), (6), (7)), or in terms of the intelligent control methods ((8), (9), (10), (11), (13), (25)). The publications describe numerous possibilities for the use of controllers in the operation of robots. The controller is designed for the specific needs of operating robots. It could be used to manage speed, to regulate the actuator (actuators) or to ensure the speed of the end effectors and selected joint actuators are kept constant. The other frequent case is represented by the control of the effector's position. This is called classical positioning, when the controller calculates the action impact (or its change) to allow the joint actuators to achieve the required coordinate without deviation of the permanent control. There are numerous other controller applications for the control systems of robots and these systems are always used to control actuators, either in case of controlling the speed, trajectory or position. In this chapter the design and implementation of the fuzzy control system using the speed control and position control will be described. The fuzzy control and the fuzzy controller design are addressed in several publications, e.g. in (11), (12), (13), (26), (27), (28), (29), (30), (31), and (32). Some of the authors deal with fuzzy logic and fuzzy control in general; whereas others describe the design of fuzzy control systems in specific examples, even in the context of robotics. However, none of the publications focus specifically on the fuzzy control application within the PLC SIMATIC S7-300, or to the robotic arm control via this control system. The integration of intelligent control methods (i.e. also by the fuzzy control) into PLC systems is described in publication (17). It does not deal with the specific situation of implementing this control type therefore, it was necessary to study not only the Fuzzy Control ++ program manuals, but also to learn about the design and implementation of the fuzzy control within the PLC in the case of a less complex example. Subsequently, it was then possible to move onto the fuzzy controller design and implementation for control at the required speed and for control in the required position.

8.1 Fuzzy Linex for arm actuators control in the manual regime

Fuzzy Linex carries out the same task as the conversion functions to verify the joint actuators with the joystick in the manual regime. The notion of Fuzzy Linex implementation comes from the proposed fuzzy control advantages – simple implementation of verbally described control rules (if ...., then ....). For the design of Fuzzy Linex it is not necessary to carry out mathematical operations (development of conversion characteristics and calculation of conversion constants) as it was for the design of conversion functions described in Chapter 7.2. The implementation part is unnecessary as well, which makes the control program less difficult. The Fuzzy Linex design utilizes the inputs and outputs of the joystick behavior analysis. They are the values in the IW336 and IW338 registers for the individual inclination angles and also for the selected zone insensitivity. These values (Fig. 22) remain unchanged. The fuzzy control itself was designed using the Fuzzy Control++ application. This software is developed by Siemens and allows the design and implementation of fuzzy controllers using the PLC series S7-300 and S7-400. The fuzzy control implementation consists of two phases. The first one is represented by the development and setting of the control system model in the related tool environment. The second phase is represented by recording of the prepared data within the PLC and the subsequent correction of their values. For the PLC the stated algorithms are added to the functional blocks of the program being executed. For the fuzzy logic method the functional block FB30 was used (SIMATIC S7-300 control). The aforementioned program functional block has its own instance data block allotted (DB30) comprising the memory elements utilized in the call of the functional block and the variables preserving the structure and parameters of the fuzzy system being modeled. The first step was to define the fuzzy system inputs and outputs. Fig. 31 shows a block diagram of controlling the joint actuators in the manual regime using Fuzzy Linex.

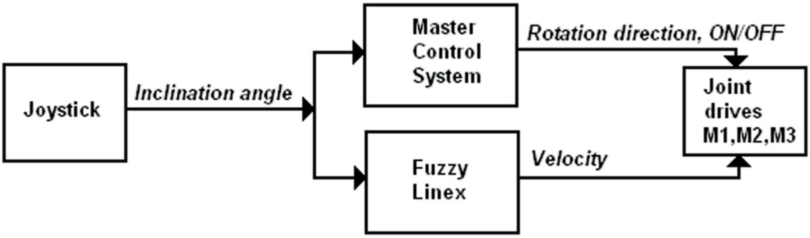


Fig. 31 *Block diagram of controlling the actuators via Fuzzy Linex*

The two inputs into the system are represented by the values from the IW336 and IW338 registers corresponding to the current angle of the joystick lever inclination in any direction. As mentioned in Chapter 7.2, the M1 and M2 motors are controlled by the same lever inclination (backward and forward from the operator). Using the S9 button the operator then selects the motor to be controlled. Fig. 32 shows the fuzzy system in the environment of the Fuzzy Control ++ application.

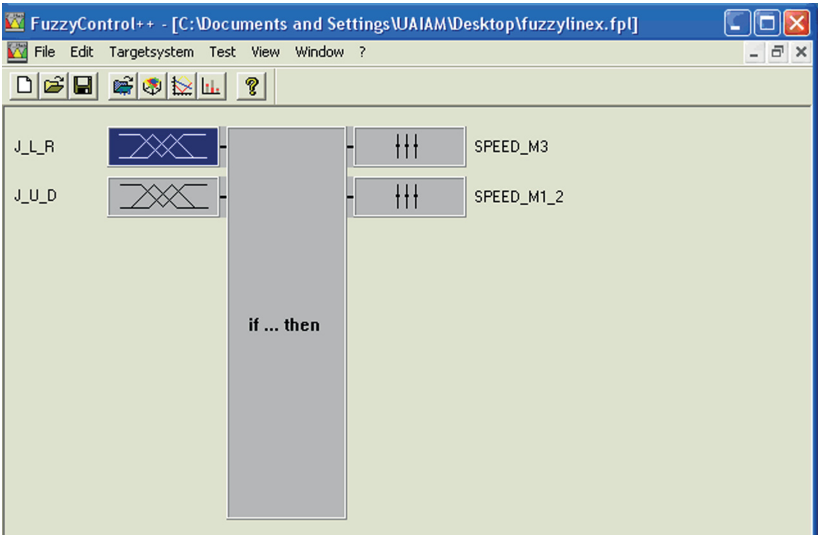
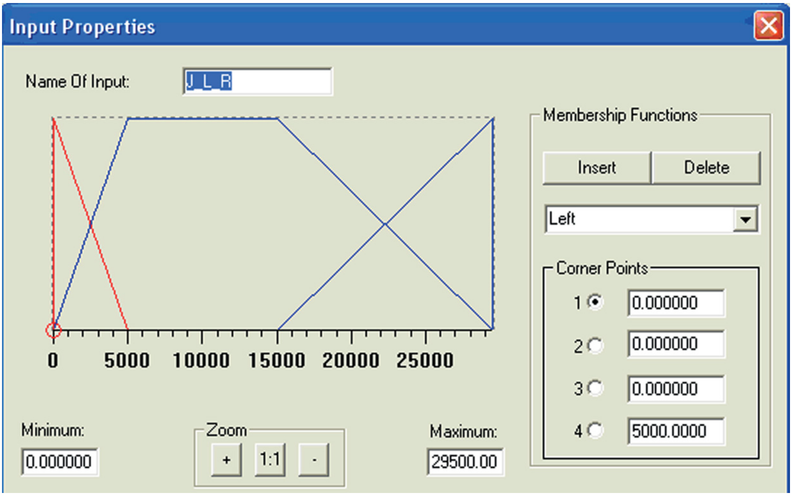


Fig. 32 *Structure of the Fuzzy Linex system in Fuzzy Control++*

The information indicating the joystick lever inclination to the left or to the right is delivered (IW336 register) to the first input (J_L_R). The information indicating the joystick lever inclination up or down is delivered (IW338 register) to the second input (J_U_D). Both inputs have the same number of membership functions; three of them. This number was selected to minimize the amount of control rules, as it ensures the control system is not complicated. Fig. 33 shows the application window of the Fuzzy Control++, in which the parameters of the first input indicated as J_L_R are set.



Name Of Input: J_L_R

Left

1 ● 0.000000

2 ○ 0.000000

3 ○ 0.000000

4 ○ 5000.0000

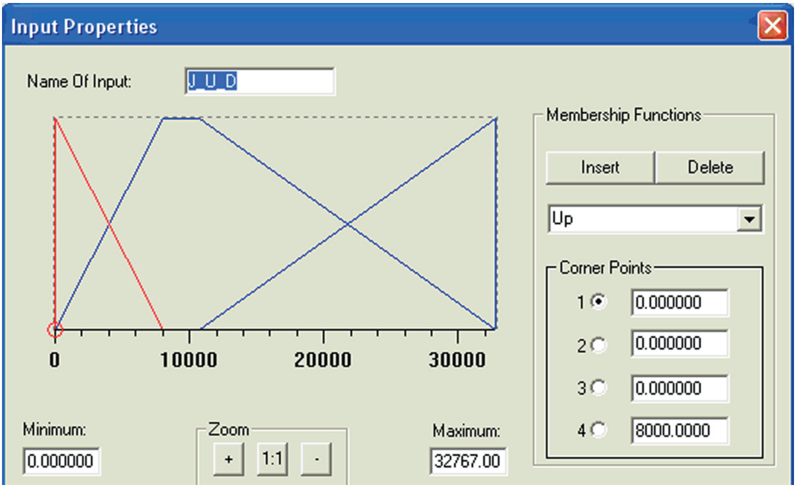
Minimum: 0.000000

Maximum: 29500.00

Fig. 33 *Membership functions for the J_L_R input*

The form and range of membership functions were set according to the values corresponding to the joystick lever inclination and the selected zone of insensitivity, illustrated in Fig. 33 was also considered. The medium membership function was indicated as Null. This corresponds to the position, in which the joystick lever is within the insensitivity zone. The two upper points were subsequently set to 5000 and 15000. As shown in Fig. 33, the boundary membership functions corresponding with the inclination to the left reaches these points (left membership function indicated as Left) and to the right (right membership function is indicated as Right). Setting of the universe range depends on the minimum and maximum values delivered to this input. As a result, the universe range for this input was set to the interval of <0;29500>. Fig. 34 shows the application window of the Fuzzy

Control++, in which the parameters for the second input indicated as J_U_D are set.



Name Of Input: J_U_D

Up

Corner Points

1 ● 0.000000

2 ○ 0.000000

3 ○ 0.000000

4 ○ 8000.0000

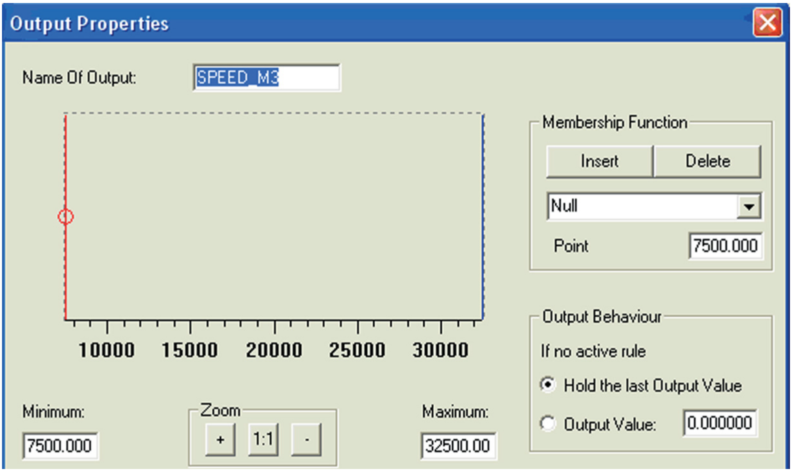
Minimum: 0.000000

Maximum: 32767.00

Fig. 34 *Membership functions for the J_U_D input*

Similarly, setting of the second input was carried out. The membership functions have indications for up, null and down. These indications correspond to the joystick lever inclination up, in the middle and down. The form and range of the membership functions in this case was set according to Fig. 34. The insensitivity zone for the inclination up or down (from 8000 to 10712) can be seen by the medium membership function. The boundary membership functions corresponding to the joystick lever inclination up and down do not infer this insensitivity zone. The universe range was selected from 0 to 32767.

In the next step it was necessary to define the fuzzy system outputs. The fuzzy system has two outputs, by which the joint motor speed is controlled. Similarly as with the inputs, one output is common for the two joint motors M1 and M2. The signal from this output is then switched to the control system output, depending on which of the motors is controlled. Fig. 35 shows the application window for Fuzzy Control++, in which the parameters of the first output indicated as SPEED_M3 are set.



Name Of Output: SPEED_M3

Null

Point 7500.000

Output Behaviour

If no active rule

● Hold the last Output Value

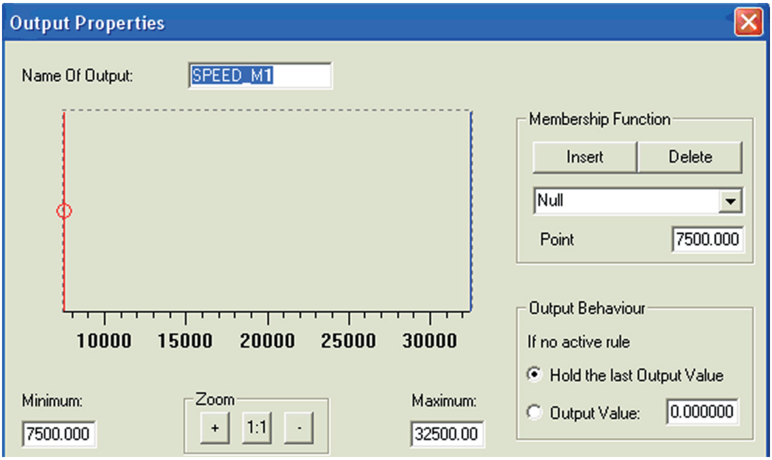
○ Output Value: 0.000000

Minimum: 7500.000

Maximum: 32500.00

Fig. 35 Membership functions for the SPEED_M3 output

The universe range was selected regarding the minimum and maximum values for controlling the voltage of the joint actuator. These values are in the range from 7500 to 32500. At the lowest value of the controlling voltage the motor does not rotate any more. The membership functions take the form of singletons. For the outputs it is not possible to set an alternative form of membership functions other than singletons. The membership functions are indicated as Null and Full, which correspond to the null and maximum rotations of the joint motor. The second output was designed in the same way. As this is the same type of motor to the SPEED_M3 output, it is evident that the universe range is set in the same way. The placement of the membership function, indicated as Null, does not have to be the same for this motor because it is a completely different joint to that of the M3 motor. By testing the joint it was found that the value of 7500 is satisfactory. Fig. 36 shows the application window of the Fuzzy Control++, in which the parameters of the second output, indicated as SPEED_M1_2 are set.



Output Properties

Name Of Output: SPEED_M1

Membership Function: Null

Point: 7500.000

Output Behaviour

If no active rule

● Hold the last Output Value

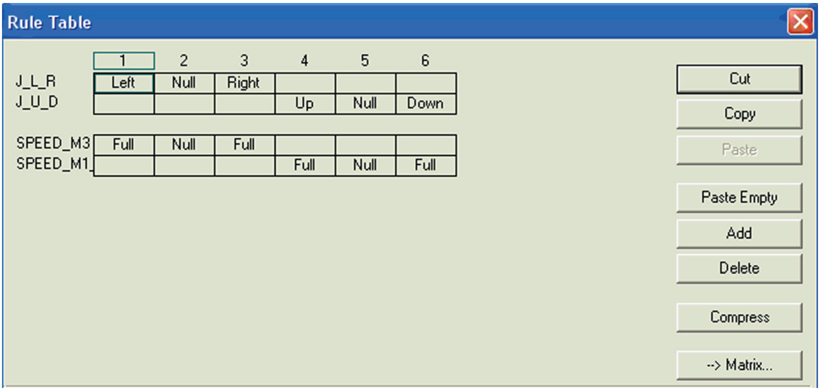
○ Output Value: 0.000000

Minimum: 7500.000

Maximum: 32500.00

Fig. 36 *Membership functions for the SPEED_M1_2 output*

In the next step the rules of control were proposed. The rules were developed from the required technique of the actuators control by using the joystick lever inclination. The higher the lever inclination angle, the higher the speed of the controlled joint motor rotation and vice versa. The rules proposed are illustrated in Fig. 37.



	1	2	3	4	5	6
J_L_R	Left	Null	Right			
J_U_D				Up	Null	Down
SPEED_M3	Full	Null	Full			
SPEED_M1				Full	Null	Full

Fig. 37 *Table of rules for the designed Fuzzy Linex system*

Fuzzy Control++ offers the possibility to create a 3D illustration depicting the basis of the rules. Fig. 38 shows a 3D representation of the rules for the SPEED_M3 output.

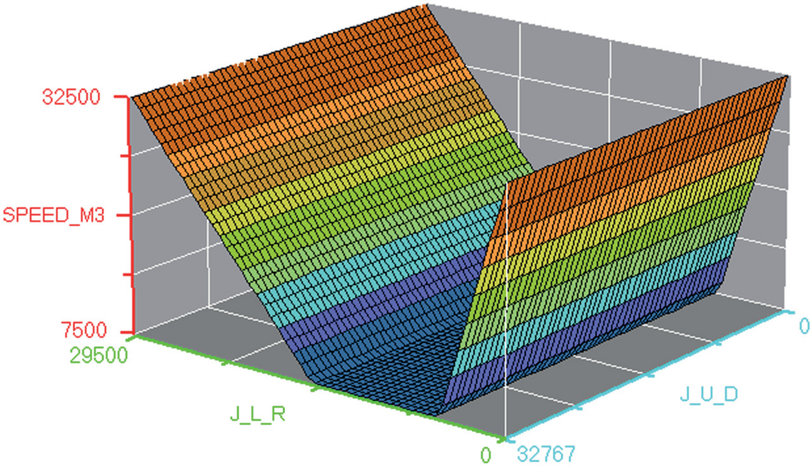


Fig. 38 *3D representation of rules for the SPEED_M3 output*

The rules were proposed so that the individual inputs do not influence each other; otherwise the Fuzzy Linex would operate incorrectly. Fig. 39 shows a 3D representation of the rules for the SPEED_M2_1 output.

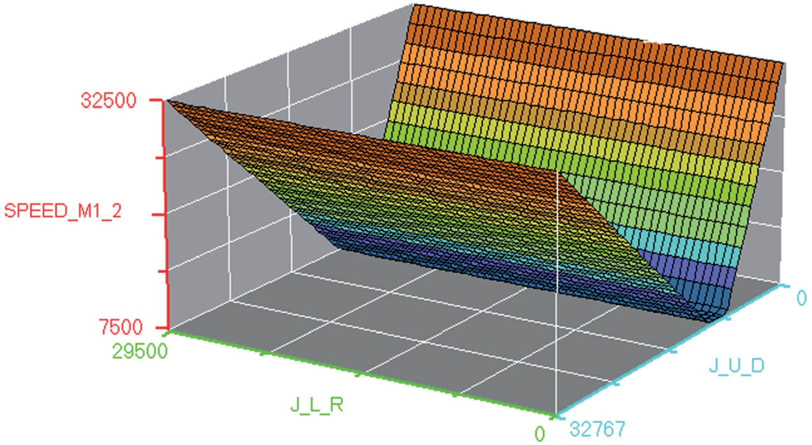


Fig. 39 *3D representation of rules for the SPEED_M2_1 output*

The fuzzy system designed by this technique must be recorded within the PLC. Before doing so, it is necessary to modify the control program so that the speed of the actuators is not calculated via the conversion functions, but via the Fuzzy Linex system. As a result, the conversion functions were eliminated from the control program, and the aforementioned functional block FB30 with the assigned database block DB30 was inserted in its place. Fig. 40 shows the FB30 functional block with inputs and outputs connected to the related registers.

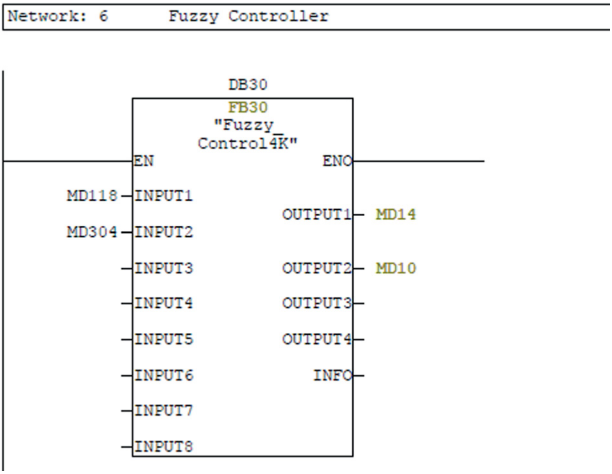


Fig. 40 *Functional block of FB30*

As shown in Fig. 40, the inputs and outputs are not directly connected to the registers comprising the values of the joystick lever inclination (IW336 and IW338), neither are they connected to the registers of outputs controlling the motors (QW320, QW322, QW336). This is caused by the fact that the inputs and outputs of the FB30 block can be connected only to real data types. Subsequently, it was necessary to carry out the conversion among the data types. The registers connected to the inputs are already comprised of converted data types. To be able to record the output data to the analogue outputs, they were converted to a suitable data type. Another reason why these outputs could not be connected directly (output 2) was represented by the fact that the controlling signal had to be switched between the registers depending on which of the motors – M1 or M2 – were being controlled. This condition results from the fact that the M1 and M2 motors are controlled by the same joystick lever inclination. The motor is selected similarly, by using the S9 button. It is very important to connect the individual registers to the inputs or outputs in the correct order, i.e. similarly as for the fuzzy system designed in the Fuzzy Control++ application. After modification of the control program it was recorded within the PLC which was connected to the programming device (PC) via an MPI communication interface. In this state the robot control system did not operate satisfactorily. It was necessary to record the proposed fuzzy system into the PLC, so that the control system was able to calculate the connecting voltage of the motors for the joystick lever inclination. The connection between the Fuzzy Control++ and the PLC was made via an MPI interface. In order to record the proposed fuzzy system, the PLC had to be in the RUN mode. Once the connection was finished, the proposed system was recorded in the PLC.

8.1.1 Testing of Fuzzy Linex

Testing of the proposed system was carried out using two techniques; visually and by measurement. The researcher monitored whether the related joint motors changed their rotations during the joystick lever inclination. During testing, the functions implemented in the previous version of the control system were emphasized (despite testing already taking place). The testing proved the correct functionality of the proposed control system with the implemented Fuzzy Linex function. The Fuzzy Control++ software is advantageous since it has a Curve Plotter function which describes the current courses of the selected signals for the fuzzy system inputs and outputs. Proof of the correct functionality of the Fuzzy Linex is shown in Fig. 41 generated via the aforementioned Curve Plotter function.

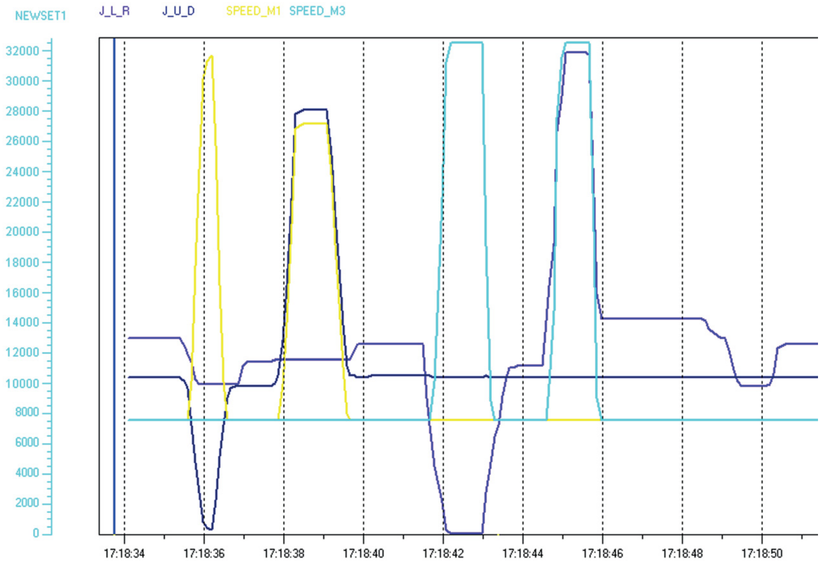


Fig. 41 *Courses of input and output signals of the Fuzzy Linex system*

Tests via the Fuzzy Plotter function comprised gradual joystick lever inclination into all boundary positions and monitoring of the related signals in the outputs. The signal in the input of J_L_R (purple curve) corresponds to the joystick lever inclination to the left and to the right. This inclination controls the M3 joint motor. The control signal controlling the M3 motor is delivered to the SPEED_M3 output (light blue curve). Fig. 42 shows that when the size of the arm inclination to the left or right is within the selected insensitivity zone, the control voltage is at the value of 7500. At this voltage the motor does not rotate. When the arm lever inclination to the left or to the right exceeds the insensitivity zone, the signal on the SPEED_M3 output increases according to the size of the lever inclination. It can be observed that the signal grows, if the lever is inclined to the right or to the left. If the lever is inclined to the middle (to the balanced position), the signal on the SPEED_M3 output will decrease. This is also the same for the signal on the J_U_D input (dark blue curve) and for the signal on the SPEED_M1_2 output (yellow curve). The signal on the J_U_D input corresponds to the joystick lever inclination up or down. The signal on the SPEED_M1_2 output corresponds to the signal controlling the M1 and M2 motors according to the button status placed on the top of the joystick lever (S9). Using the joystick it is possible to control the M3 and M1 motors (or M2) simultaneously, i.e. the joystick does not have to be always inclined only up or down, or to the left or right. Fig. 42 shows the characteristics generated using the WENS860 oscilloscope in testing the Fuzzy Linex function.

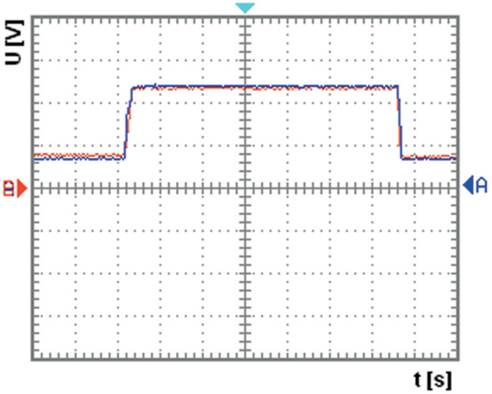


Fig. 42 Course of voltage corresponding to joystick inclination and motor voltage

Channel A (blue curve) of the oscilloscope was connected to the IW336 output. Channel B (red curve) of the oscilloscope was connected to the connecting terminals of the M3 motor. Using channel A the monitored the joystick inclination to the left or to the right, and via channel B the researcher monitored the voltage on the M3 motor. Fig. 42 shows the voltage change on the IW336 register by the inclination to the right and at the same time the voltage change on the M3 motor (QW336). In this case the lever was inclined to the right and kept in the position. Higher joystick inclination (blue curve) also results in a higher voltage on the motor (red curve). The voltage on the motor was calculated correctly by using Fuzzy Linex. Similarly, the control of other joint motors was tested. The previous tests verified the correct functionality of the control system implemented by Fuzzy Linex in the manual mode.

8.2 Fuzzy control of speed

To control the speed, the controller must calculate an action impact (or change of the action impact) so that the required rotation speed of the joint motors (or the effector's speed) is held constant. The fuzzy controller design for this situation comprised the definition of inputs and outputs, their membership functions and the definition of control rules. Fig. 43 shows the control loop of the actuators control for the speed required.

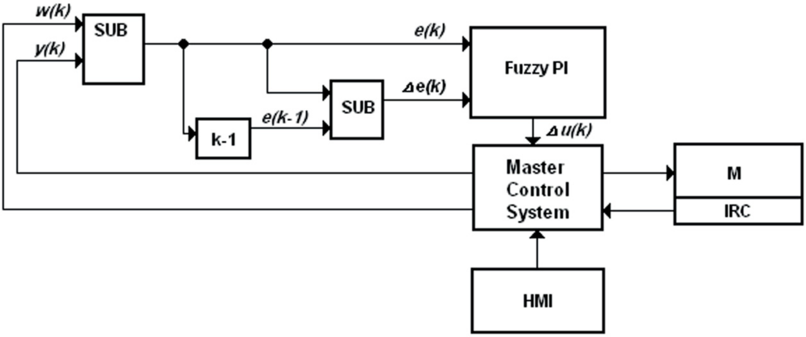
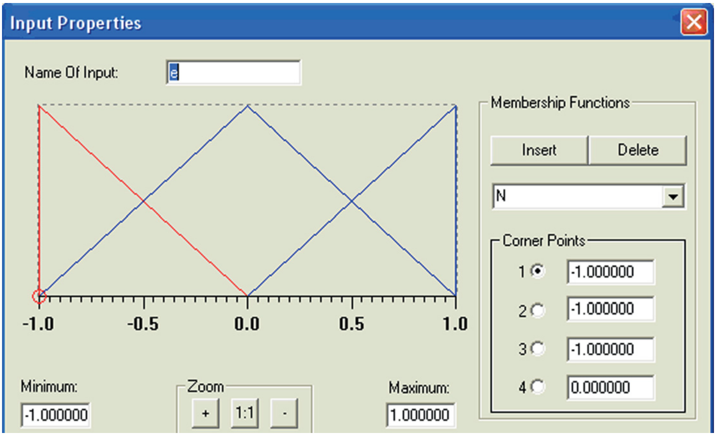


Fig. 43 *Control loop of the actuators control for the required speed*

The control deviation and its change represent the fuzzy PI controller input. The action impact represents the controller output. The master control system carries out no control tasks, but has the role of converting the data types and modifies the signals (from IRC) so that they can be processed by the fuzzy controller. It also controls the direction of the motor rotation and ensures the detection of end positions to prevent collisions. The SUB blocks and k-1 are part of the master control system. SUB blocks are the functional blocks ensuring the mathematical operation of subtraction. Block k-1 is the memory block (delay), whose output is represented by the previous input signal value. For the fuzzy controller design the researcher again used the Fuzzy Control++ software. The procedure was similar to the design using Fuzzy Linex. The first controller input is represented by the control deviation (e). Fig. 44 shows the application window of Fuzzy Control++ for setting the input parameters to which the control deviation is delivered.


Input Properties

Name Of Input: e

Membership Functions: N

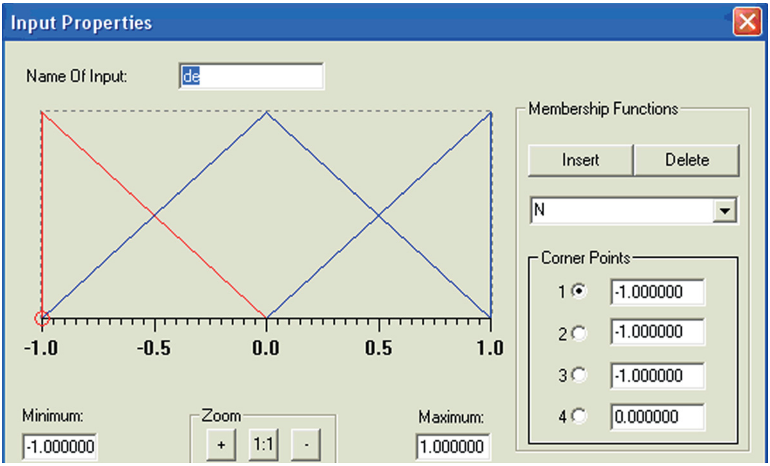
Corner Points
1 ● -1.000000
2 ○ -1.000000
3 ○ -1.000000
4 ○ 0.000000

Minimum: -1.000000

Maximum: 1.000000

Fig. 44 *Membership functions of the control deviation with normalized universe*

The input has three membership functions indicated as N, Z and P. They correspond to the negative, null and positive value of the control deviation. All three functions are of a triangular shape. The universe range was selected from -1 to 1 representing a normalized universe. Such a range was used in order to set the designed fuzzy controller more easily, or applicable also for another control task. The setting of the controller inputs will not have to be modified. The input signals are modified by the normalization coefficients in the master control system before they are delivered to the controller input. The other input into the controller, the control deviation change is delivered (de). Its settings are identical to the first controller input. Due to the universality of the designed controller, the researcher selected the universe range from -1 to 1 with three membership functions. The membership functions are indicated as N, Z and P and are of the triangular shape. Fig. 45 shows the application window of Fuzzy Control++ for setting the parameters of the input, to which the control deviation change is delivered.



Name Of Input: de

Membership Functions: N

Corner Points

1 ● -1.000000

2 ○ -1.000000

3 ○ -1.000000

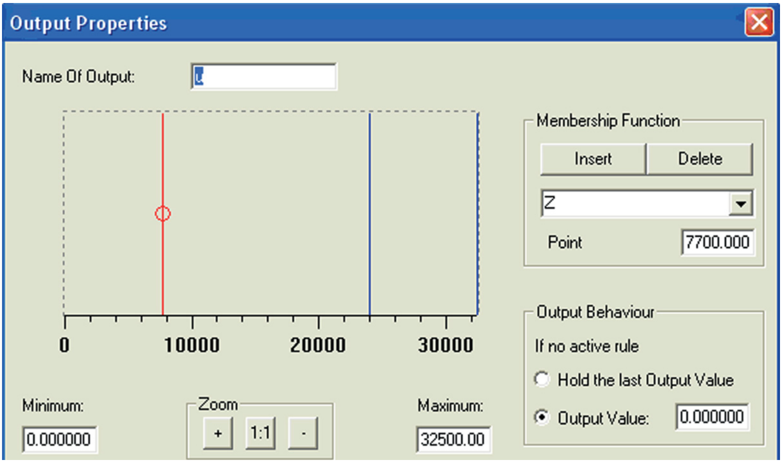
4 ○ 0.000000

Minimum: -1.000000

Maximum: 1.000000

Fig. 45 *Membership functions of the control deviation change*

Definition of the controller output was the next step. Fig. 46 shows the window of the Control++ software providing the output parameters setting.



Name Of Output: u

Membership Function: Z

Point 7700.000

If no active rule

○ Hold the last Output Value

● Output Value: 0.000000

Minimum: 0.000000

Maximum: 32500.00

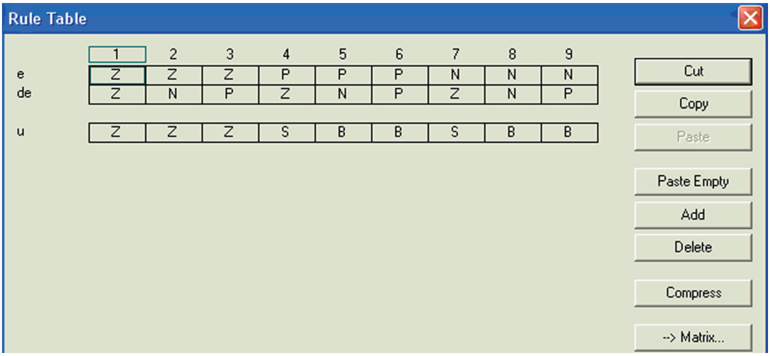
Fig. 46 *Membership functions of action impact*

The output was designed with three membership functions in the form of singletons. For the fuzzy controller outputs it is possible to set the membership functions in the Fuzzy Control++ program only as singletons. The membership functions are indicated as Z, S, and B, which corresponds to the action impact size (null, small and large). The universe was selected from 0 to 32500. Since also in the next control task the motor will be controlled by the output, it was not necessary to propose the universe with a normalized range. The membership function Z corresponding to the null action impact is shifted to the value of 7700, as this value is in the format of the control signal for the motor represents null rotations.

Control rules for the fuzzy PI controller were written as follows:

if *e* is A and *de* is B then *u* is C.

These proposed rules were recorded using the function of rule editor in the Fuzzy Control++ software. The table of control rules is shown in Fig. 47. For three membership functions for each input and output the basis of rules of the fuzzy PI controller comprises altogether 9 rules of control.



	1	2	3	4	5	6	7	8	9
e	Z	Z	Z	P	P	P	N	N	N
de	Z	N	P	Z	N	P	Z	N	P
u	Z	Z	Z	S	B	B	S	B	B

Fig. 47 *Table of control rules for the fuzzy PI controller*

The higher number of membership functions (and hence the higher number of rules) was not selected intentionally, so that the calculations did not burden the system and influence the whole control system dynamics. Fig. 48 shows a 3D representation of the basis of rules for the fuzzy controller designed.

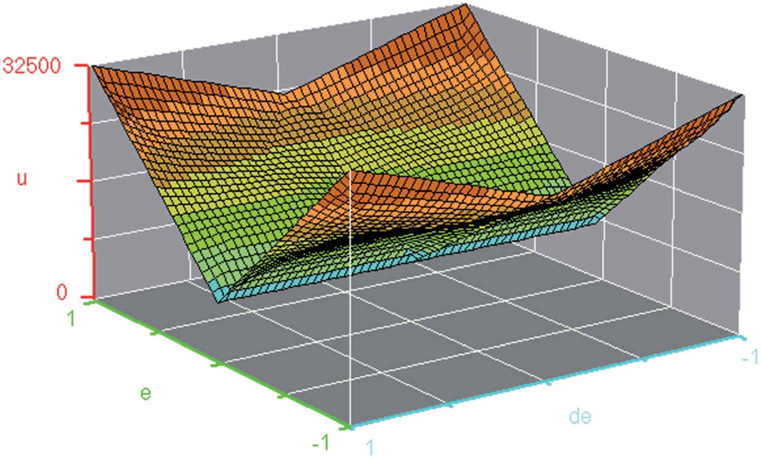


Fig. 48 *3D representation of control rules for the fuzzy PI controller*

In order to control the joint motors independently of each other, three controllers were designed, one for each joint motor. Subsequently, the fuzzy system has six inputs, three outputs and the basis of rules is comprised of 27 rules. This designed fuzzy system is shown in Fig. 49.

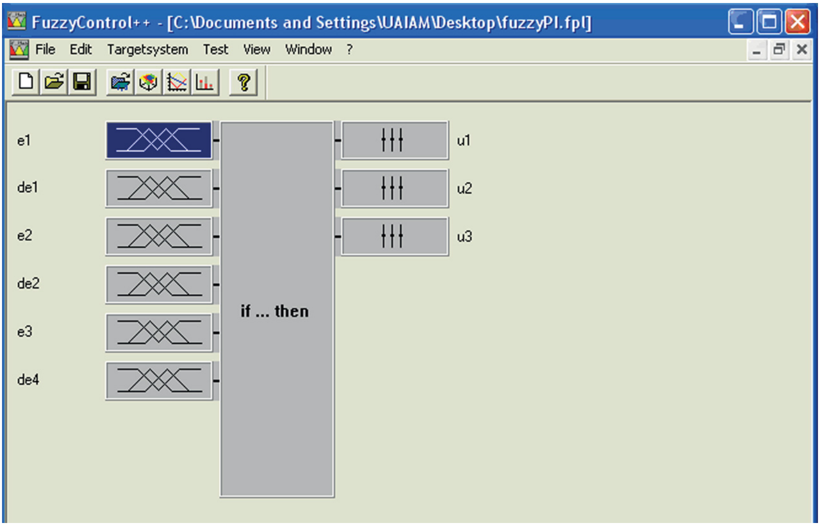


Fig. 49 Fuzzy system with three fuzzy PI controllers

Before implementing the new controller in the PLC, it was necessary to adjust the control program. The control program adjustment was represented by the addition of another functional block, the FB30. This block was connected to the second data block (DB32) as the DB30 is the data block used for Fuzzy Linex. The inputs of the block, FB30 (DB32) were implemented so that they correspond to the inputs of the fuzzy controller designed. The control program was modified so that the controller output controls the PLC output connected to the joint motor. In this situation it was also necessary to ensure the conversions among the data types both for the inputs and outputs in order to utilize the necessary functional blocks. As aforementioned, in front of the controller inputs it was necessary to include the normalization blocks to ensure the input signal (signals) in the controller was adjusted for the normalized universe. The functional blocks of the MUL_R (Multiply Real) were used as normalized blocks. Fig. 50 shows an example of the control deviation normalization.

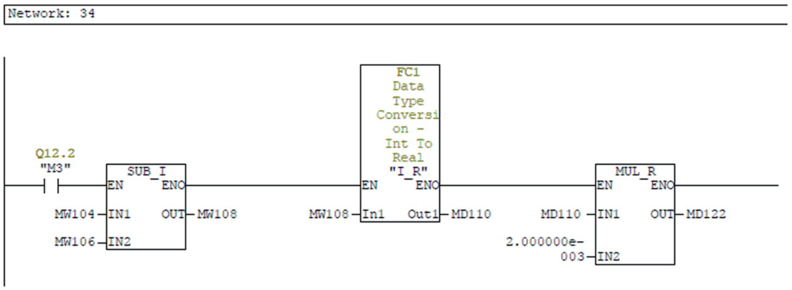


Fig. 50 *Calculation of the control deviation and its normalization*

The functional block of SUB_I provides the calculation of the control deviation. This block carries out the subtraction of the values delivered to its inputs IN1 and IN2. The set point value saved in the MW104 register is delivered to the input IN1. The process value of the controlled variable which is saved in the MW106 register is delivered to the input IN2. The output of the subtraction operation is saved in the MW108 register corresponding to the control deviation. As several times aforementioned, to use the necessary functional blocks, the researcher had to carry out conversions among the data types of the processed data. As the conversions among the individual data types were frequently used, universal functions for the data types conversions were developed. The function FC1 used in this situation converts the data type of Integer to the data type of Real. Another block is represented by the MUL_R carrying out the multiplication operation. In this case it carried out the function of the normalization block. The non-normalized value of the control deviation saved in the MD110 register is delivered to the input IN1. The value of the normalized coefficient is delivered to the input IN2. The normalized value of the control deviation on the output of the multiplication functional block is saved in the MD112 register. This value can be directly processed by the fuzzy controller. Similarly, the second fuzzy controller input is normalized. Development of the function to calculate the control deviation change was the next modification made in the control program. For this purpose the FIFO function was developed. Its input is represented by the current value of the control deviation and its output by the previous control deviation value, i.e. the function executes a one-step delay of the signal. To calculate the control deviation changes the researcher used the functional block SUB_I to execute the subtraction operation. The value from the MW108 register is delivered to the input IN1, in which the current control deviation value is saved. The value from the MW312 register is delivered to the input IN2, where the control deviation value in the previous block is saved. After subtraction of the IN1 and IN2 input values, the result value is saved in the MW314 register which corresponds to the control deviation change on the functional block SUB_I output. The control deviation change is then converted to the Real data type and normalized. Fig. 51 shows the calculation of the control deviation change and its normalization implemented in LAD.

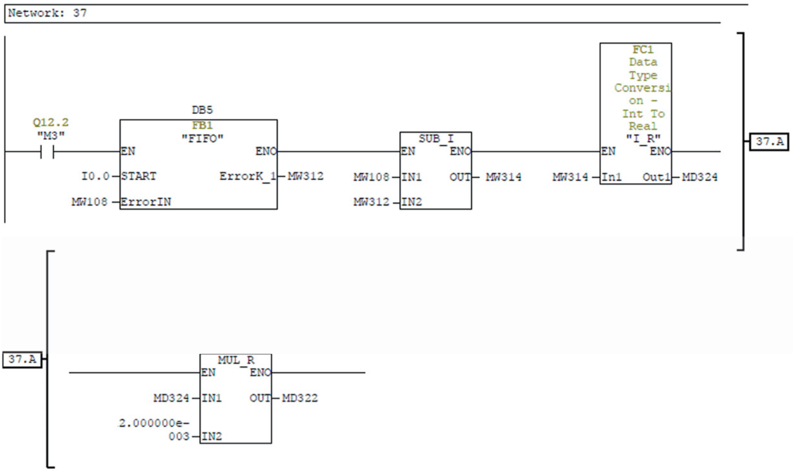


Fig. 51   *Calculation of the control deviation change and its normalization*

The control program was then modified so that the controller could be utilized both in manual and automated modes. In the manual mode the required speed is set via the joystick based on the control lever inclination. Regarding the inclination direction and status of the S9 button, the control system assesses on which joint motor the current value is in force. In the automated mode the required value of the joint motor (motors) speed is set in the control program. Its change is influenced by the type of task the robotic arm has to execute. For testing, the value was set manually directly to the MW104 register corresponding to the required value.

8.2.1 Testing of the fuzzy controller to control speed

The fuzzy controller designed for the control of the required speed was tested both in the manual and automated modes. The test was carried out similarly in both modes. The required speed of the related joint motor was selected and via the connected oscilloscope (WENS860) the researcher monitored the voltage on the motor terminals and the signal from the IRC. The difference between the manual and automated modes was in the technique of achieving the required speed selection. In the manual mode the speed was selected via the joystick lever inclination. In the automated mode the speed was set directly in the control program. In order to compare the measured courses of the signals, the first measurement was carried out without the fuzzy controller, i.e. the required speed in the form of the control voltage was delivered directly to the output controlling the motor. Once the speed was stabilized, the motor was loaded and the researcher monitored the voltage course on the motor terminals and the signal from the IRC. Fig. 52 shows the voltage course on the M3 motor and the signal from the IRC_M3 by the excluded fuzzy controller.

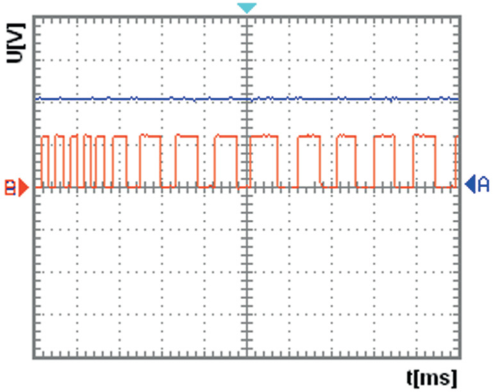


Fig. 52 *Voltage course on the M3 motor and signal from the IRC by the excluded controller*

The courses of the signals shown in Fig. 52 are recorded over time, when the voltage on the motor is stable (the required value is already selected and does not change). The blue curve illustrates the voltage course on the motor and the red curve illustrates the signal from the IRC. The signal from the IRC is in the form of impulses. The width of impulses, or the time period between two neighboring rising edges (or falling edges) informs the speed recorded, i.e. if the width of pulses (i.e. also of gaps) increases, then the motors slow down. If the width of impulses decreases, then the motor accelerates. The figure illustrates that the width of pulses starts to increase after a while. This is the moment, when the motor was loaded, i.e. its real rotations decreased. The connecting voltage remained unchanged, as the required value of the rotations did not change and the fuzzy controller was excluded all the time. In the manual mode, it is possible to achieve the required speed of the joint motor (effector) via the joystick inclination. It is the change of the value required so that the load is compensated. The operator is very important because a change in the motor speed can noticed and the speed of the lever inclination adjusted. In the automated regime, no load compensation is possible, as the size and course of the load is unknown in advance. Further measurements were carried out with the connected fuzzy controller, designed so that it compensated the load (its change) by its action impact together with keeping the required speed. Fig. 53 shows the course of the connecting voltage on the M3 motor and the IRC_M3 signal by the connected fuzzy controller.

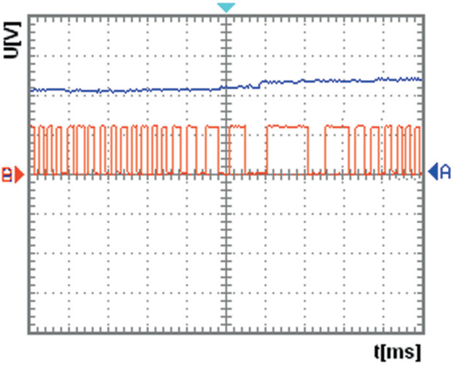


Fig. 53 *Voltage course on the M3 motor and signal from the IRC by connected controller*

The blue curve illustrates the course of voltage on the motor terminals. It is the action impact converted to the voltage. The red curve corresponds to the motor rotation speed. The test was carried out similarly as in the previous situation. Once the required value was selected and the rotations were stable the load of the motor increases. Fig. 53 shows the increase of the load measured by the width of impulses (the motor rotations decrease). The fuzzy controller reacts and starts to modify the action impact so that the motor rotations achieve the original value. The action impact change is evident from the course of the blue curve. After the achievement of the original rotations (the same width of impulses as in the beginning) the voltage on the motor stabilizes. The measurements were carried out by using the unchanged required value of the speed. In the next test it was necessary to verify the behavior of the fuzzy controller also by the load, which cannot be compensated any more due to physical reasons. The test was executed similarly as in the previous cases. However, after the rotations were stabilized, the motor was loaded to such an extent that it stopped. Fig. 54 shows the characteristics measured by the oscilloscope for the case of stopping the motor due to the overload. The red curve corresponds to the motor rotations. After the motor stops, the curve has the form of a line, i.e. the IRC generates no impulses. The red curve has no null value, as at this point, the IRC has its coding wheel in the position, reflecting logic 1 in the output. The blue curve illustrates that the controller increased the value of the action impact to the maximum level as the required value remained unchanged (the motor must rotate by the required speed).

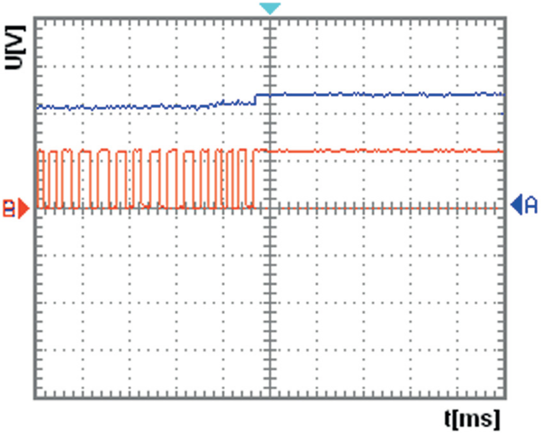


Fig. 54 *Voltage course on the M3 motor and signal from the IRC by motor overload*

The previous figure shows that the controller operates correctly; however, it could result in the motor's destruction. Therefore, protection to decrease the voltage on the overloaded motor at the correct moment was implemented into the control system. Fig. 55 shows the measured characteristics for overloading with the implemented function protecting the motor from its destruction.

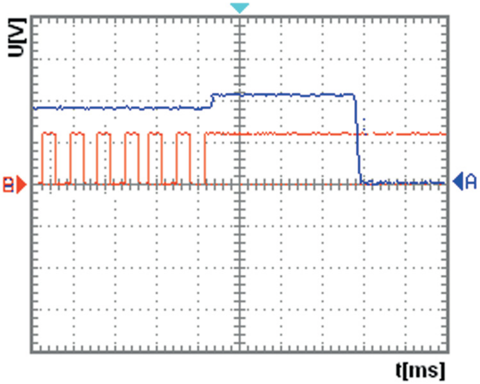
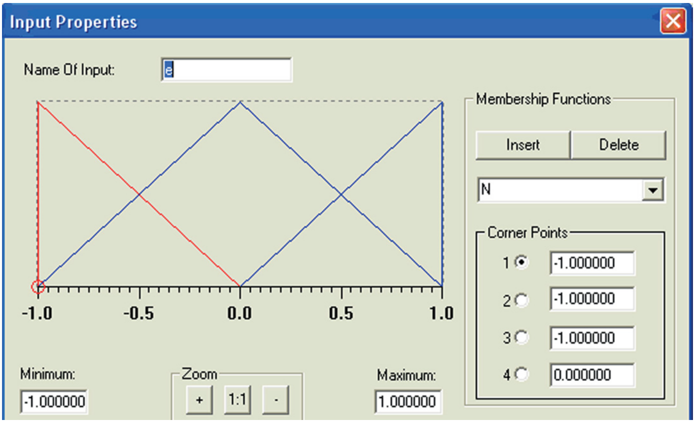


Fig. 55 *Voltage course on the M3 motor and signal from the IRC with the safety function*

Fig. 55 illustrates that when the motor rotations decreased to null, i.e. the red curve indicated the form of pulses but it was in a line, the controller increased the voltage on the motor to the maximum (blue curve). After a specific time the protection reacted and decreased the voltage in the motor terminals to the minimum (by unchanged the required value), thus preventing the overloaded motor from destruction.

8.3 Fuzzy control of the position

To achieve the required position of the robotic arm, the controller must ensure stopping of the effector in the required position without any permanent control deviation, while the effector movement must be as fast as possible. If we consider the internal coordinates, then it is necessary to ensure the stopping of the individual joint actuators in the required position. The procedure of the controller design was similar to the one in the previous case. The control loop is identical to the control loop shown in Fig. 43, illustrating the controlling of speed. The difference is only in the description of the individual signals. Here the control deviation is calculated from the difference of the required and the current positions. The change of the control deviation is represented by the difference of the current control deviation and its value in the previous step. In other words, the change of the control deviation corresponds to the distance achieved since the previous time period. The action impact (controller output) is here also represented by the control voltage. The master control system executes the same tasks as in the previous control task. In the controller design it was not necessary to pass all the design steps, as the already designed controller with normalized universes could be used and the researcher did not have to infer its internal structure. The first controller input in this case is represented by the control deviation, however, it does not correspond to the difference between the required and current speed, but to the difference between the required and current positions. The setting of input parameters, to which the control deviation is delivered, is shown in Fig. 56.



Name Of Input: e

N

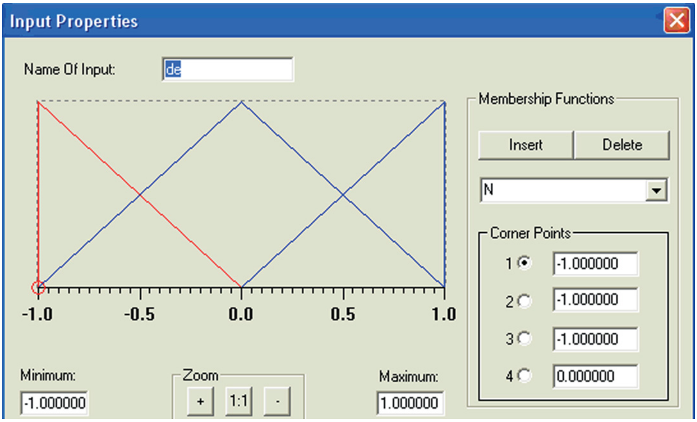
1 ● -1.000000
2 ○ -1.000000
3 ○ -1.000000
4 ○ 0.000000

Minimum: -1.000000

Maximum: 1.000000

Fig. 56 *Membership functions of the control deviation with a normalized universe*

The second input to the controller is represented by the control deviation change which here corresponds to the difference of the control deviation and the control deviation in the previous step. The setting of parameters for the second input is shown in Fig. 57.



Name Of Input: de

N

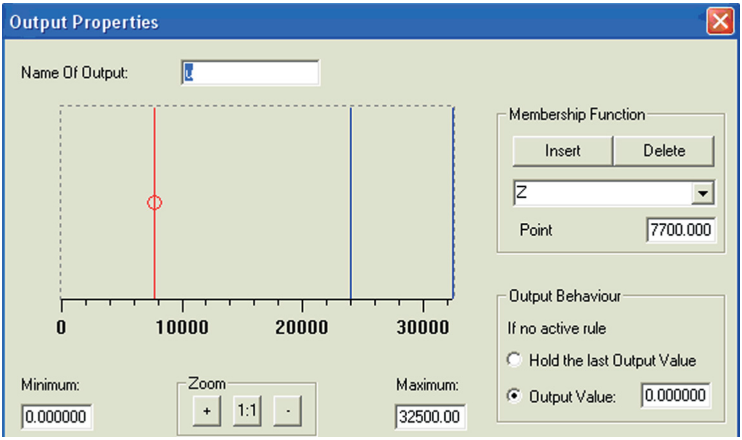
1 ● -1.000000
2 ○ -1.000000
3 ○ -1.000000
4 ○ 0.000000

Minimum: -1.000000

Maximum: 1.000000

Fig. 57 *Membership functions of the control deviation change*

The controller output is represented by the action impact corresponding to the control voltage of the joint actuators. Fig. 58 shows the parameter settings of the fuzzy controller output. In this example, the output has no normalized universe; despite this it was not necessary to infer the controller's structure due to the fact that the controller output function remained unchanged. Also in this example the joint motors are controlled via the action impact. The value of the membership function corresponding to null rotations was already set for the individual joint motors in the previous control task. It was necessary to ensure that the outputs were controlled by the same motors as in the previous example.



Output Properties

Name Of Output:

Membership Function: Z

Point 7700.000

Output Behaviour

If no active rule

○ Hold the last Output Value

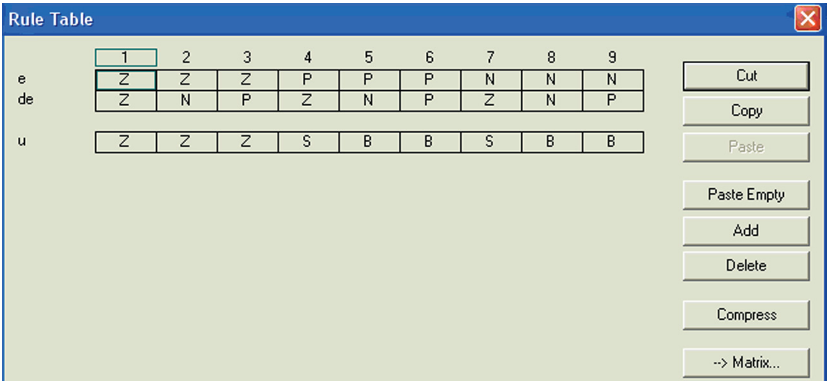
● Output Value: 0.000000

Minimum: 0.000000

Maximum: 32500.00

Fig. 58 *Membership functions of the action impact*

In this example the control deviation and its change are the inputs of the controller. The action impact represents both the controller output and the fuzzy PI controller. Therefore, it was possible to use the basis of rules from the previous controller design. The control rules are shown in Fig. 59.



	1	2	3	4	5	6	7	8	9
e	Z	Z	Z	P	P	P	N	N	N
de	Z	N	P	Z	N	P	Z	N	P
u	Z	Z	Z	S	B	B	S	B	B

Fig. 59 *Table of control rules for the fuzzy PI controller*

The previous figures illustrates that the fuzzy controller structure in the previous control task remained unchanged. It was also necessary to design the fuzzy system with three controllers so that the control independence of all joint motors was ensured. The control program had to be modified for the controller inputs. Due to the control, the signals represented the position information and its change, but not the speed delivered. It was essential therefore to modify the inputs in the fuzzy controller so that they corresponded to the inputs of the controller designed and the control deviation was delivered to the first input. Implementation of the control deviation calculation for the joint driven by the M3 motor is shown in Fig. 60.

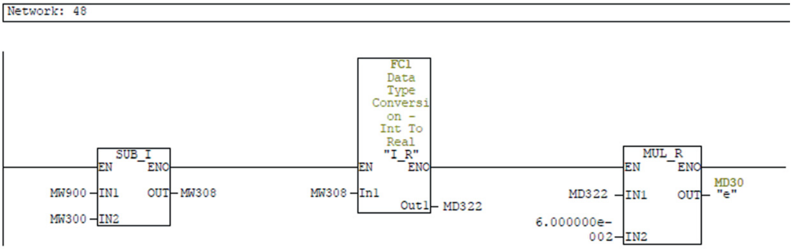


Fig. 60 *Calculation of the control deviation and its normalization in LAD*

The functional block SUB_I executes the subtraction operation of the IN2 input from the IN1 input. The required value is then saved to the MW900 register. The MW300 register is the calculator output which calculates the impulses from the IRC_M3, i.e. it saves the value of the current position. The output of the SUB_I functional block is represented by the control deviation, whose value is converted to the required data type and is subsequently normalized by using the MUL_R functional block. The output of this block when executing the multiplication operation between the control deviation and the normalization coefficient is represented by the normalized control deviation (MD30). The normalized control deviation is delivered to the first input of the fuzzy controller. Following the calculation of the control deviation the change was implemented. The change of the control deviation is calculated as the difference between the current control deviation and the control deviation recorded in the previous step. To achieve the calculation the researcher utilized the FIFO function, which was also used in the implementation process of the previous controller. Through use of the FIFO function, the value of the control deviation outlined in the previous step was acquired. Fig. 61 shows the implementation of the calculation of the control deviation change and its normalization.

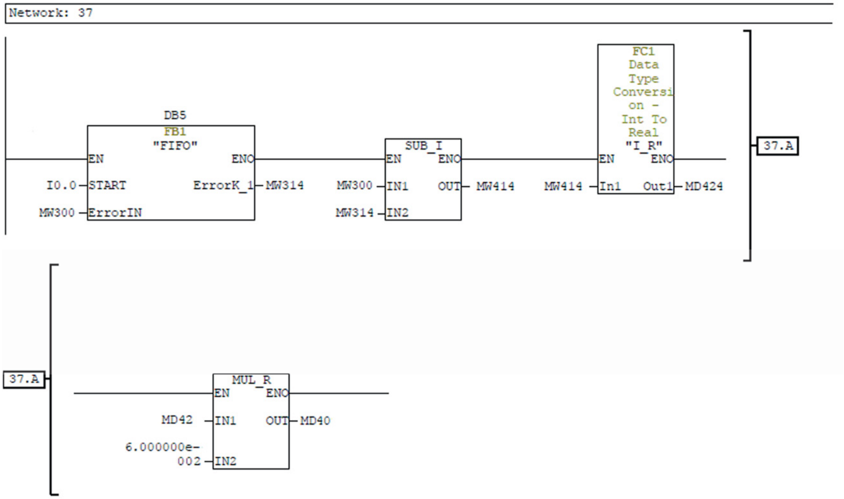


Fig. 61 *Calculation of the control deviation change and its normalization*

As aforementioned, by using the FIFO function the value of the control deviation in the previous step can be acquired (MW314). Through use of the SUB_I functional block, the value is subtracted from the current value of the control deviation (MW300). The control deviation is then converted to the required data type and multiplied by the normalization coefficient via the MUL_R functional block. The output of the MUL_R functional block corresponds to the normalized control deviation. The second part of the control system relating to the fuzzy controller did not require modification. The setting of the setpoint value was a significant modification. As the control system operates with the internal coordinates, the required position must be set in the form of internal joint coordinates. One of the options is to set the internal coordinates using the joystick. In the manual mode the operator moves the effector to the required position and then back to the initial position. The system remembers these two positions (if the operator indicates them using the command set on the switch I0.1). Subsequently, in the automated mode the effector will control movement between these two positions. In each of the positions it is possible to program the required action that the gripper must execute; however this action has no influence on the controller's function. Another option for setting the required positions is represented by the sequence of specific positions (or only one of them) directly programmed in the program.

8.3.1 Testing use of the fuzzy controller to control the position

Testing of the controller is comprised of setting the required position and then monitoring its achievement. In this example, the operator used the Fuzzy Control++ program called Curve Plotter for monitoring the courses of the individual quantities. The operator did not use the oscilloscope in the testing, as use of this technique would not allow verification of the achieved position (as it does not have an impulses counting function). Through use of the Curve Plotter function, the operator could monitor the control deviation, the current position, the change of the control deviation, and the control action and required position. Fig. 62 shows that the courses of the monitored signals using the M3 motor are controlled from the position with the coordinate of 0 to the position with the coordinate of 195. This shows that the required coordinate has the value of 195.

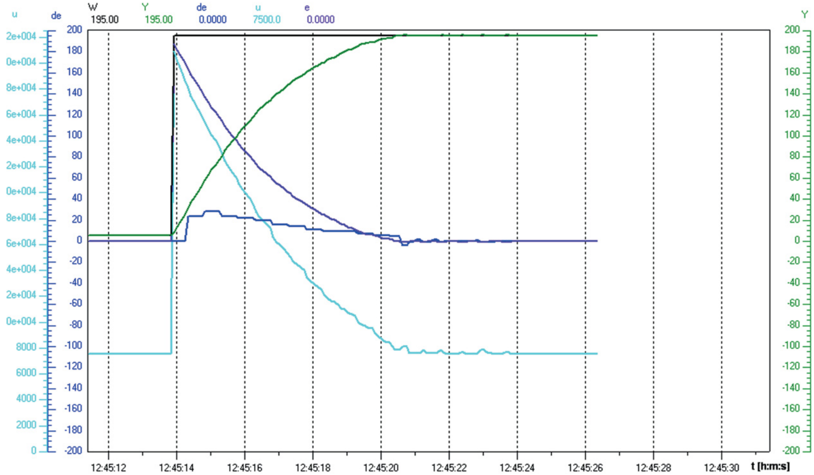


Fig. 62 *Courses of monitored signals using the M3 motor position control*

Figure 62 shows the required quantity by the black colored curve. The green curve expresses the course of the measured quantity, i.e. the course of the current position. The light blue curve illustrates the course of the action impact which corresponds with the course of the joint motor voltage control. The purple curve represents the control deviation and the dark blue curve represents the control deviation change. Fig. 63 shows the courses of the monitored signals using the M3 motor control from the position with the coordinate 195 to the position with the coordinate 5.

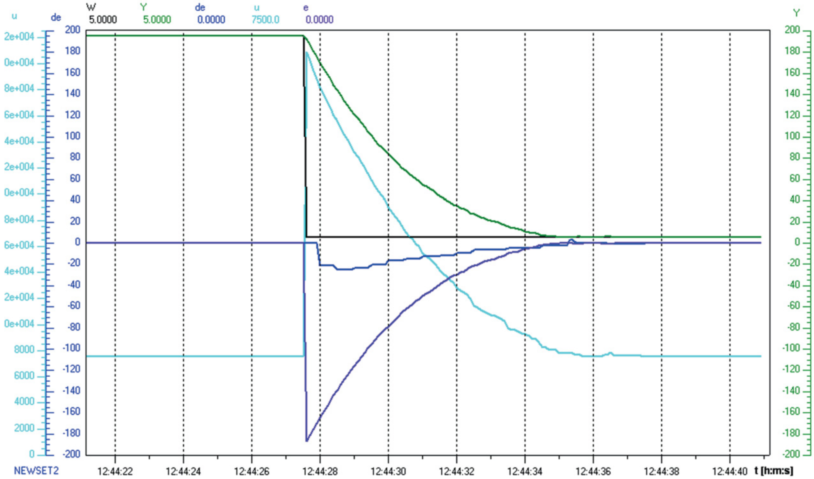


Fig. 63 *Courses of monitored signals using the M3 motor position control*

Both figures indicate that the control performs without a permanent control deviation. After achievement of the required value it is possible to monitor a gentle oscillation of the transmission characteristics, which can be caused by the oscillation of the coding wheel on the IRC.

9. ANALYSIS OF THE ACHIEVED RESULTS

In the previous chapter the author described the stage of the design, implementation and testing of the fuzzy control within the control system of the robotic arm. The functionality of the fuzzy systems designed was verified by the experiments. The experiments were proposed so that it was possible to test the functionality of the designed fuzzy systems with the available measurement systems (WENS860 two-channel oscilloscope and METEX M-3270D digital multimeter) as precise as possible. The Fuzzy Control++ Curve Plotter program was a very useful testing tool for the visualization of the signals on the inputs and outputs of the designed fuzzy system. Although the functionality and accuracy of the measured results was influenced by the production preciseness of the robotic arm and its parts, the results of the experiments proved the correct functionality of all three fuzzy systems.

In the first example using the Fuzzy Linex system, the experiment was focused on confirmation of the fuzzy system utilization not as the controller but as a transmission function. This meant that the researcher had to design the conversion functions, whose task was to calculate the control voltage of the joint motors on the basis of the inclination angle without the use of mathematical apparatus. To do so, it was necessary to make adjustments to the motor rotations depending on the inclination of the joystick lever. During the design stage, the researcher utilized the advantage of using fuzzy control as the system calculated the value of its output (outputs) on the basis of rules written in the linguistic format. They are more comprehensive and subsequently the basis of the rules can be proposed or modified more easily.

The value of the outputs was determined on the basis of three rules (three rules for each output), which was perfectly satisfactory for the system functionality. The fact that the fuzzy system outputs controlling the joint motors had only two membership functions did not harm the functionality. By using the experimental measurements the researcher verified and proved the functionality of the fuzzy system designed. The achieved results show that it is advantageous to use the fuzzy system for simple control purposes via the PLC. The design and implementation is unambiguously shorter and simpler than with the use of mathematical apparatus. It requires fewer programming skills as the fuzzy system only operates' within the determined application and in the programming area only with the functional block. However, in the design stage it is necessary to know the range of the values of the fuzzy system inputs and outputs in advance (in this case they were the values ranges for the individual joystick inclinations and ranges of motors control voltages). It is also necessary to know the behavior of the systems connected to the system inputs and outputs, as the control rules are formed particularly on its basis. Incorrectly proposed rules could reflect negatively on the overall control result.

In the second example the experiment was focused on the fuzzy system testing for the control of speed. As no measurement instrument or device to measure the robotic arm effector's speed was at the researchers' disposal, the experiment was proposed for the control of the individual joint motors rotations. As the speed of the effector depends on the rotations of the individual joint motors, the control of the speed could be focused mainly on the joint motors. The required value of the speed was not set for the robot's effector but for the individual joint motors, which from the point of the implementation was not a drawback as the control system was proposed so that it operated with the internal joint variables. For control of the individual joint motors the researcher designed the fuzzy PI controller. The fuzzy system was built so that each joint could be controlled independently of each other (for each joint an independent controller). The inputs and outputs of the controller had three membership functions which influenced the number of rules – there were nine of them. For three joint motors the designed fuzzy system had altogether 27 rules, 6 inputs and 3 outputs. This configuration does not achieve the fuzzy system possibilities (even 1000 base rules and 9 membership functions for each input and output). In the system design the researcher considered also the possibility of prolonging the period of their processing in the PLC, and hence the influence of the dynamics control. As each joint motor was controlled by its own controller, the verification measurement had to be executed also on each of the joints. As only a robotic arm model, which has physical limitations, was at the researchers' disposal, the experiments had to be carried out several times. This presented drawbacks due to the weakness of the motors which resulted in difficulties guessing the load size to ensure an overload was avoided or so that the rotation slowing was clear from the IRC signal, whose resolution was quite low (only 12 pulses per rotation, or 24 if the rising as well as the falling edge was monitored). Nevertheless, despite the limitations the achieved results showed that the controllers were designed correctly, as their task was to control the motor revolutions to the setpoint value during the load change.

In the third example the experiment was focused on testing the designed fuzzy control in the control of the position. Also in this example the experiment was built regarding the possibilities of the measurement equipment and the controlled robotic arm. If in the control of the position the effector's position is measured, the preciseness of the positioning would be strongly influenced by the construction rigidity of the arm and individual joints. By use of the model, the individual joints have large wills and the real effector's position should not correspond with the derived position from the internal joints coordinates. The other limitation was represented by the absence of the device, which could sufficiently precisely determine the effector's position in the area. Therefore, and due to the fact that the result effector's position is derived from the individual joints position, the experiment was built on the positioning of the individual joint motors. The required position of the individual joints was given in the joint coordinates, and therefore, the position of the individual joints was measured in the internal joint coordinates. The experiment was aimed not only at the verification of the fuzzy control functionality, but also at testing the applicability of the fuzzy controller with the normalized universes of inputs designed in the previous task. As aforementioned, the structure of the designed fuzzy system remained unchanged, only the signals delivered to the controller input were changed. In this example each joint was controlled by the independent controller; therefore, no impact on the fuzzy system designed in the previous task was required and the experiments results proved the fuzzy control functionality. The prerequisite that if the fuzzy system is designed universally (normalized universes) it is possible to use it also in another control task, in the case of the same controller type (in this case PI). However, it is necessary to adjust the input and output signals according to the specific control task. It is possible to combine the individual control tasks and implement more fuzzy systems in the control system if necessary.

The experiments executed in all three examples proved the functionality of the proposed solutions. If the researcher had access to better equipment, either the measurement instruments and devices or the control object itself (in this case the robotic arm model), the experiments could be carried out in more detail with a larger emphasis on testing the control system dynamics and robustness and the control process stability.

CONCLUSIONS

The use of fuzzy logic in the control processes of non-linear or mathematically complex systems, where the use of conventional controller types would be demanding or even impossible, was successfully described in many scientific publications. Its utilization was frequently implemented in practice as well, particularly in the investigation of various mutually unrelated subject matters (ABS, analysis of the portfolio within capital market investments, etc.). Fuzzy logic was also used for the system control without the mathematical model defined in this monograph. It was particularly the design of the fuzzy control for the robotic arm which was subsequently implemented into the PLC provided as the robot’s control system. Despite the fact that fuzzy control is not utilized in PLC frequently, the selected PLC type is supported by the Fuzzy Control++ software tool determined for the design and implementation of the fuzzy controller. The designed system utilizes the fuzzy control of the robotic arm via a joystick, where the task of the fuzzy system is to determine the rotation speed of joint motors on the basis of the angle size of the joystick inclination. Subsequently, the fuzzy system is used to control the speed, where it has to follow the required rotation speed of the individual joint motors independently of the load. In the case of an overloaded motor the function preventing the long-term overload and avoiding thus the motors destruction was implemented. In real operation the overload of the arm should not occur as they are selected according to their maximum load to be manipulated. Nevertheless, in this case the function was not useless, since the joint motors are weak and in the course of testing they frequently stopped due to overloading. The fuzzy system was used also in the control of the position, where it has to manage the individual joints to the required position without permanent control deviation. The fuzzy controller was designed with standardized universes, and therefore it was possible to use it without any change in the internal structure for the control of the position as well as for the control of the speed. The proposed control system has an advantage – it is built on the PLC basis. In contrast to classical control systems of the robot, it has the advantage of a higher number of I/O modules and the option of HW or SW adjustments. This is a prerequisite to control more complex technologies. From the achieved results verified by the experiments, the author can state that the aim of the monograph was met. One of the possible improvements of the control system could be found on the user interface for the planning of the trajectory or for the individual tasks of the robotic arm. Furthermore, it would be advantageous to join e.g. MLTS (Multi Laser Tracker System), to the control system, which could send the data of the effector's position to the control system. Joining of the fuzzy control and MLTS could lead to more precise positioning of the robotic arm as the effector's position would not be derived using the calculation from the position of the individual joints, but the system would operate with the real effectors position.

REFERENCES

1. SICILIANO, B., SCIAVICCO, L., VILLANI, L., ORIOLO, G. 2009. *Robotics Modelling, Planning and Control*. London: Springer, 632 p. ISBN 978-1-84628-641-4
2. ŠOLC, F. ŽALUD, L. 2002. *Robotika (Robotics*). Brno: VUT, pp. 7 – 153
3. PIRES, J. N. 2007. *Industrial Robots Programming: Building Applications for the Factories of the Future.* New York: Springer, 282 p. ISBN 978-0-387-23325-3
4. Robotické systémy MELFA (MELFA robotic systems). Mitsubishi Electric Europe B.V. 2009 [cit. 28 Jan 2011] Available on the Internet < http://www.csmtrade.sk/automation/pdf/roboty.pdf >
5. LEWIS, L. F., DAWSON, M. D., ABDALLAH, T. C. 2004. Robot Manipulator Control: Theory and Practice. New York: Marcel Dekker, Inc., 607 p. ISBN 0-8247-4072-6
6. KELLY, R., SANTIBÁÑEZ, V., LORÍA, A. 2005. *Control of Robot Manipulators in Joint Space.* London: Springer, 426 p. ISBN 978-1-85233-994-4
7. BAJD, T., MIHELJ, M., LENARCIC, J., STANOVNIK, A., MUNIH, M. 2010. *Robotics*. New York: Springer, 152 p. ISBN 978-90-481-3775-6
8. EMARA, H., ELSHAFEI, A. L. Robust robot control enhanced by a hierarchical adaptive fuzzy algorithm. [online] Motoring Applications of Artificial Intelligence 17 (2004) [cit. 28 Jan 2011]. Available on the Internet <http://www.intellcontrol.com/files/bxwx/Robust%20robot%20control%20enhanced%20by%20a%20hierarchical%20adaptive%20fuzzy%20algorithm.pdf>
9. NOVAKOVIC, B., VRANJES, B., NOVAKOVIC, D., MAJETIC, D., KASAC, J., BREZAK, D. 2002. An Adaptive Fuzzy Robot Control without a Fuzzy Rule Base. [online] 15th Triennial World Congress, Barcelona. [cit. 28 Jan 2011] <http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2002/data/content/01226/1226.pdf>

10. SONG, Z., YI, J., ZHAO, D., LI, X. 2005. *A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach*. [online] Fuzzy Sets and Systems 154 (2005). [cit. 28 Jan 2011] (Available on the Internet) <http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6V05-4FWTYM7-1-&_cdi=5637&_user=8157715&_pii=S0165011405001168&_origin=search&_coverDate=09%2F01%2F2005&_sk=998459997&view=c&wchp=dGLzVlb-zSkWA&md5=49c4df9061a02795d2531198a8e1f70c&ie=/sdarticle.pdf>
11. CHEN, G., PHAM, T. T. 2001. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems.* New York: CRC Press, 329 p. ISBN 0-8493-1658-8
12. PASSINO, K. M., YURKOVICH, S. 1998. Fuzzy Control. California: Addison Wesley Longman, Inc., 522 p. ISBN 0-201-18074-X
13. TZAFESTAS, S. G., VENEISANOPOULOS, A. N. 1994. *Fuzzy Reasoning in Infromation, Decision and Control Systems*. Netherlands: Kluwer Academic Publishers, 565 p.
14. Modi, V.J., Chan, J.K. 1991. Performance of a Flexible Mobile Manipulator. *Journal of Mechanical Design*, Volume 113, Issue 4, pp. 375-384.
15. HOŠOVSKÝ, A., NOVÁK-MARCINČIN, J., PITEĽ, J., BORŽÍKOVÁ, J., ŽIDEK, K. 2012. Model-based Evolution of a Fast Hybrid Fuzzy Adaptive Controller for a Pneumatic Muscle Actuator. *International Journal of Advanced Robotic Systems*, **9**(56), 1-11. ISSN 1729-8806
16. HAVRAN, M., PITEĽ, J. 2012. Hybridné adaptívne riadenie antagonisticky zapojených pneumatických umelých svalov. *Strojárstvo EXTRA*, **XVI**(5), 11/1 – 11/4. ISSN 1335-2938
17. GABRIEL, J. 2005. Integrovanie inteligentných metód do PLC riadiacich systémov (Integrating the Intelligent Methods into PLC Control Systems). *AT&P journal PLUS7*, pp. 23 – 27.
18. SIEMENS AG 2003 Fuzzy Control++ V. 5.0 User`s manual (Erlangen: Siemens AG, I&S IT PS 213)
19. KOPČEK, M., BEZÁK, T., ZÁVACKÝ, P., FRIDRICH, V. 2010. Inspection Stand for Monitoring of Spent Nuclear Fuel SVYP-440. In:

Process Control 2010 : 9th International Conference. Pardubice: University of Pardubice, C060a-1-5. ISBN 978-80-7399-951-3

20. RÁSTOČNÝ, K., ŽDÁNSKY, J. 2007. Applications of Programmable Logic Controllers in Safety Systems. *Advances in Transport Systems Telematics*, pp. 177-182. ISBN 978-83-917156-6-6
21. ŽDÁNSKY, J., RÁSTOČNÝ, K. 2006. Parameters Evaluation of PLC Dependability and Safety. *Advances in Electrical and Electronic Engineering*, **5**/1-2, pp. 181-184. ISSN 1336–1376
22. KOPČEK, M., KRIŽANOVÁ, G. 2008. Models Used for PID Controller and PLC Programming Teaching. In: *Process Control 2008: Proceedings of the 8th International Scientific-Technical Conference*. Pardubice: University of Pardubice. ISBN 978-80-7395-077-4
23. STRÉMY, M., ELIÁŠ, A., VAŽAN, P., MICHAĽČONOK, G. 2008. Virtual Laboratory Implementation. In: *Ecumict 2008 : Proceedings of the Third European Conference on the Use of Modern Information and Communication Technologies*. Nevelland v.z.w., pp. 385-393. ISBN 9-78908082-553-6
24. ŠPENDLA, L., TANUŠKA, P. 2011. Strategies and Techniques of System Testing. In: *Aktuaľnyje problemy i innovacii v ekonomike, upravlenii, obrazovanii, informacionnych technologijach*. Stavropoľ: Severo-Kavkazskij Gosudarstvennyj Techničeskij Universitet. ISSN 2074-1685. - Vypusk 6, Tom I: Materialy meždunarodnoj naučnoj konferencii, Stavropoľ-Kislovodsk, pp. 103-105.
25. JAGER, R. 1995. *Fuzzy Logic in Control*. Delft: Technische 37 Universiteit Delft. ISBN 90-9008318-9
26. VAŠČÁK, J. 2008. *FUZZY LOGIKA V REGULÁCII: Podrobný úvod do problematiky fuzzy riadenia (FUZZY LOGIC IN CONTROL: Detail introduction into fuzzy control*). Košice: TUKE, pp. 34 – 56.
27. PIVOŇKA, P. 2003. *Vyšší formy řízení (Higher Forms of Control)*. Brno, pp. 34 – 63.
28. MODRLAK, O. 2004. *Teorie automatického řízení II.: FUZZY ŘÍZENÍ A REGULACE (Theory of Automation Control II: FUZZY CONTROL AND CONTROL*). Liberec: KRT, 27 p.

29. ŠMEJKAL, L., URBAN, L. 2004. Fuzzy logika. In *Automatizace (Automation)*, **47**(4), pp. 224-226.
30. ŠMEJKAL, L., URBAN, L. 2004. Fuzzy logika. In *Automatizace (Automation)*, **47**(4), pp. 290-292.
31. ŠMEJKAL, L., URBAN, L. 2004. Fuzzy logika. In *Automatizace (Automation)*, **47**(5), pp. 357-359.
32. ŠMEJKAL, L., URBAN, L. 2004. Fuzzy logika. In *Automatizace (Automation)*, **47**(6), pp. 421-423.

CONTENTS

Author: Ing. Tomáš Škulavík, PhD.
Title: OPTIMAL PILOT BUS SELECTION FOR THE SECONDARY VOLTAGE CONTROL USING PARALLELISM
Published in: Ilmenau (Germany)
Edited by: Universitätsverlag Ilmenau
1st edition: 2013
Editions: Scientific monographs in Automation and Computer Science
Pages: 115 (figures 63; proofreading – Paul Wooliscroft)
Quires (20 standard pages): 5,02