Geometric and rough set approach to inverse kinematics for arm manipulator

Krzysztof Tokarz, Christian Manger

***Abstract*—-** In this paper the construction of the arm of the robot has been presented. The possibilities of different methods of calculations of reverse kinematics has been discussed. The geometric approach intended to work with the controller based on low-resource microcomputer has been presented. Details about mechanical and electronic construction of the arm of the robot has been described. Additionally, Rough Set Theory which deals with imperfect knowledge is proposed to be used to manipulate the robot arm. The information system considered in the Rough Set Theory can be obtained in a myriad of techniques usually belonging with either statistical analysis or machine learning methodologies. The approach presented in this paper is the example of the latter category.

***Keywords*—**robot, inverse kinematics, forward kinematics, rough sets, relative reduct, decision algorithm.

I. Introduction

SINCE many years people try to replace human work with machines. Machines called robots are faster and more effective than people. Many elements of robots are built with inspiration from the nature. Construction of the manipulator as the arm of the robot is based on human arm. The scope of this paper is to present the geometrical approach to calculations needed to control the arm manipulator constructed in Institute of Informatics at Silesian University of Technology. Second part of the paper presents also some details about hardware implementation of the arm and software implementation of calculations.

In the paper there is presented application of Rough Set-based methodology to the problem of robot arm manipulation. Rough Set Theory, developed by Polish scientist Zdzislaw Pawlak [1] in the early 1980s, deals with the problem of imperfect knowledge that has been studied by scientists for many years. Such imperfect or incomplete knowledge can be interpreted and manipulated in many ways, probably the most popular of which is provided by the fuzzy set theory due to Lotfi Zadeh [2].

Classical Rough Set Theory provides tools for succinct description of knowledge about the Universe by means of relative reducts and relative value reducts and resulting decision algorithms can be used in many applications with satisfying accuracy [3-6].

Manuscript received November 22, 2010.

K. Tokarz is with the Institute of Informatics, Silesian University of Technology, Gliwice, 44-100 Poland, phone: +48-32-237-2500; fax: +48-32--237-2733; e-mail: krzysztof.tokarz@ polsl.pl.

Ch. Manger is with the LG Nexera, Vienna, Austria.

II. Background

Arm consists of segments named links that represent bones, and joints. Palm is represented with the effector placed at the end of the arm. Arms of robots are built usually with servos connected together with the links. Servos can be prismatic or revolute. Prismatic servos move linearly with the effect of link extension. Revolute servos turn changing the angle between concurrent links (Fig.1.).

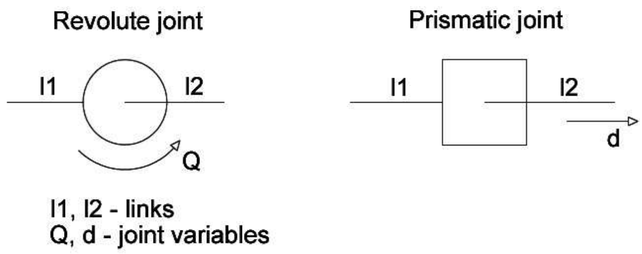

Fig.1 Types of joints

The action of joint can be described with the variable that represents the angle for revolute joints and the displacement in the case of prismatic joints. Set of joint variables determines the current position of the effector in the space. Controlling the arm placement can be done using one of two possible algorithms: forward and inverse kinematics. In forward kinematics all joint variables are known and are used to calculate coordinates of current position of the effector [7].

The question of inverse kinematics is to calculate set or sets of joint variables that allows effector to reach the chosen point in the space. Inverse kinematics calculations are much more complicated than forward kinematics. Forward kinematics always has the solution and it is always only one solution. Inverse kinematics can have the solution or not. It can give also infinite number of solutions. In some situations even if the solution exists it can be unreachable because of construction of the robot. Set of reachable points forms the working space.

The subset of this is the full operation space [8] in which points can be reachable with specific approach of the effector. To reach any point in the working space with any angle of the effector the arm should have six degrees of freedom. Any arm that has more than six degrees of freedom is called redundant– the human arm has seven degrees of freedom. Generally redundant robot possesses more degrees of freedom than is needed to execute a given task [9]. Controlling the redundant manipulator is difficult because there are more than one way to reach the position, and calculations often give infinite number of results. The algorithm commonly used in inverse kinematics calculations is Denavit-Hartenberg. It is based on transformations of matrix, and has big computational complexity [10]. Other algorithms are geometrical and use trigonometric calculations [11]. Trigonometric approach is simpler for manipulators with smaller number of degrees of freedom. For manipulators with larger number of degrees of freedom one possible solution is to split the problem into smaller parts and make calculations with geometrical approach [12,13]. It is usually done while the controller has too small computational power to make calculations with matrices.

III. Machine Learning Approaches

Machine learning algorithms are characterized by their efficiency when dealing with large data sets. Not only do they achieve high accuracy in classification tasks, but are also popularly used in feature extraction process [14].

Artificial Neural Networks are often employed in classification of the scenery for the robot arm manipulation. As long as recognition is incorrect, the network weights are adjusted until the network can properly identify scenery. Then the network can be used for recognition of unknown scenes.

The genetic algorithm approach starts out with definition of a set of rules expressing some characteristics of the robot movement. Then these rules are tested against a set of known movements presented to the program and each rule is given a fitness score basing on which some rules are disregarded, leaving only these with highest scores (selection). These are next slightly modified (mutation) and some new rules are added. The process is repeated until the rules that evolved correctly attribute movements to tasks.

Decision trees represent a special type of a classifier, which is trained by repetitive selection of individual features that stand out at each node of the tree. In classification procedure there are considered only those features that are required for the studied pattern. Quite often decision trees are binary with feature selection built in the structure which makes them suboptimal for most applications, yet they work fast.

Support Vector Machines are examples of a two-class classifier. As the criterion for optimization there is considered the margin between the decision boundaries of the two classes, defined by the distance to the closest training patterns called support vectors. The classification function is defined by these patterns and their number is minimized by maximizing the margin. The main drawback of this method is the computational complexity of the training procedure.

Rough Set Theory and its notions constitute yet another case of machine learning approaches [15] and they are discussed in more detail in the next section. Especially the modifications dedicated for continuous attribute space such as those presented in [16] and further developed in [17] or quasi-dominance rough set approach elaborated by Cyran [18] are of importance in the context of this paper.

IV. Rough Sets Foundations

The fundamental concept of Rough Set Theory (RST) is the indiscernibility relation which using available information (values of attributes A) about objects in the Universe partitions the input space into some number of equivalence classes $[x]_A$ that are such granules of knowledge, within which single objects cannot be discerned.

While in classic set theory elements are either included or not included in a set, in RST the indiscernibility relation leads to lower $\underline{A}X$ and upper approximations $\bar{A}X$ of sets, the first comprised of objects whose whole equivalence classes are included in the set, the second consisting of objects whose equivalence classes have non-empty intersections with the set. If the set difference between the upper and lower approximation of some set, called the boundary region of this set, is not empty then the set is said to be rough, otherwise it is crisp.

Sometimes among the attributes describing objects of the Universe there are distinguished two classes, called *conditional attributes C* and *decision attributes D*. Then information about the Universe can be expressed in the form of Decision Table.

A. Decision Tables

Decision Table (DT) is defined as 5-tuple

$$DT = < U, C, D, v, f > \tag{1}$$

where U, C, and D are finite sets (U being the Universe, C set of conditional attributes and D set of decision attributes), while v is a mapping which to every element $a \in C \cup D$ assigns its finite value, set V_a (domain of attribute a), and f is the information function f: $U \times (C \cup D) \rightarrow V$, where V is a union of all V_a and $f(x, a) = f_x(a) \in U$ for all x and a.

Thus Decision Table in its columns specifies all attributes defined for objects within the Universe, both conditional and decision ones, while rows provide values of these attributes for all objects. Each row constitutes also the decision rule as for specified values of condition attributes the values of decision attributes are provided.

For each decision table there is defined its consistency measure $\gamma_C(D^*)$ which answers the question whether the table is deterministic. All decision rules provided by rows of DT are compared, one by one against all others, and if there are at least two that have the same values of conditional attributes but different for decision attributes D, the table is not deterministic.

The consistency measure $\gamma_C(D^*)$ of Decision Table is equal to the C-quality of the approximation of the family D^*

$$\gamma_C(D*) = \frac{card(Pos_C(D*))}{card(U)} \qquad (2)$$

where the C-positive region of the family $D*$, $POS_C(D*)$, is defined as

$$POS_C(D*) = \bigcup_{x_i \in D*} \underline{C}D_i \qquad (3)$$

B. Relative Reducts and Relative Value Reducts

It may often happen that information contained in a Decision Table is excessive in this sense that either not all conditional attributes or not all their values are necessary for correct classification indicated by decision attributes. Rough Set Theory provides tools for finding, if they exist, such functional dependencies [19] between conditional attributes which may lead to reduction of their number without any loss of information and they involve the concept of a reduct.

A set of attributes $R \subseteq C$ is called relative reduct of C with respect to D or D-reduct of C ($RED_D(C)$) if R is the maximum independent subset of C with respect to D. If R is D-reduct then

$$POS_R(D*) = POS_R(D*) \quad and \quad C \xrightarrow{k} D \Rightarrow R \xrightarrow{k} D \quad (4)$$

Attribute $c \in C$ is redundant in C with respect to D (D-redundant) if

$$POS_C(D*) = POS_{C-\{c\}}(D*) \qquad (5)$$

otherwise the attribute c is irremovable from C with respect to D (D-irremovable).

A relative core of C with respect to D (D-core of C) is the set of all D-irremovable attributes of C.

$$CORE_D(C) = \{c \in C : POS_C(D*) \neq POS_{C-\{c\}}(D*)\} \qquad (6)$$

The relation between D-reduct and D-core is given by the following formula

$$CORE_D(C) = \bigcap_{R \in RED_D(C)} R \qquad (7)$$

Further reduction of the Decision Table is achieved by such elimination of some values of an attribute for some elements of the Universe (without eliminating the attribute itself) that does not diminish the classification abilities of DT for this set of attributes. That leads to the concept of relative value reduct (D-value reduct) and the core of value reducts (D-value core) [20].

It is said that a value of attribute $c \in C$ is D-dispensable for $x \in U$ if

$$C(x) \subseteq D(x) \Rightarrow C_c(x) \subseteq D(x) \qquad (8)$$

otherwice the value of attribute c is D-dispensable for x. If for every attribute $c \in C$ value of c is D-indispensable for x, then C is called D-independent for x.

Subset $C' \subseteq C$ is a relative value reduct of C if and only if C' is D-independent for x and

$$C(x) \subseteq D(x) \Rightarrow C'(x) \subseteq D(x) \qquad (9)$$

The set of all D-indispensable for x values of attributes in C is called the relative value core of C for x and denoted by $CORE_D^x(C)$, with the property

$$CORE_D^x(C) = \bigcap RED_D^x(C) \qquad (7)$$

where $RED_D^x(C)$ is the family of all D-reducts of C for x.

Relative reducts can be perceived as masks put on decision rules included in the decision table, indicating for each rule these attributes whose values are sufficient to perform correct classification. It is quite common that for a decision rule several distinct relative value reducts can be used and this results in the necessity of choice among them.

V. Construction of the arm

The construction of presented arm of the robot is inspired by human arm [21, 22]. Original arm has seven degrees of freedom and the same the constructed arm has. The detailed configuration is presented in the Fig.2. The mechanical arm uses 8 servos controlled by 8-bit microcomputer.

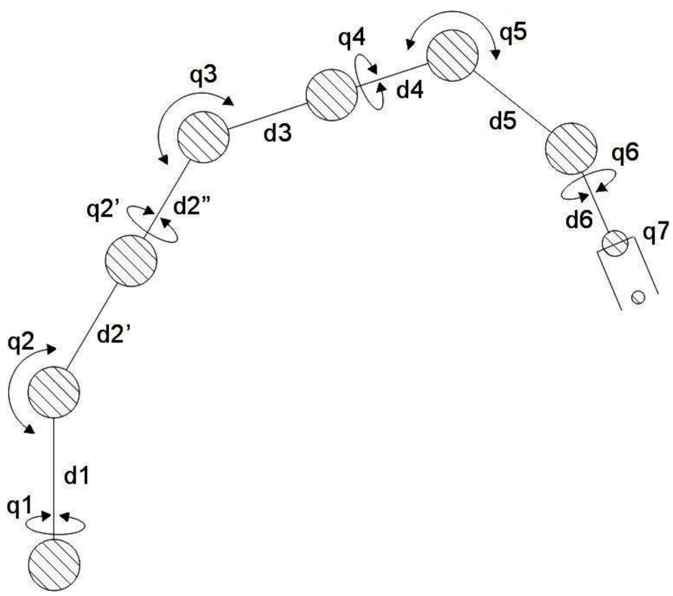

Fig. 2.. Configuration of the arm of the robot

Human arm does not allow prismatic move so all servos used to build the arm are revolute. The servo number 8 is for moving the effector, so it is out of the scope of this paper.

VI. Geometric Approach

The arm is controlled with simple, 8-bit microcontroller. Computational power of this microcontroller does not allow to perform advanced algorithms on it. Because the seven degrees of freedom manipulator needs sophisticated calculations there is a need to find simpler methods of calculations, especially for the inverse kinematics.

Decoupling inverse kinematic problem into simpler problems seems to be interesting solution. One approach is to split the arm into two parts – the arm and the spherical wrist. First part as the arm with elbow presented in the Fig.3, is responsible for locating the center of wrist. Second, as presented in Fig.4, determines the orientation of the wrist. Fist part calculation problem is named inverse position kinematics, second is named inverse orientation kinematics. It is possible to make calculations for both parts with planar geometric approach.

Calculations need the wrist to be spherical with all axes intersecting at one point. It means that position of the center of the wrist is the function of variables calculated for the inverse position kinematics. This approach with sample calculations is in details described in [23]. Calculations of inverse position kinematics presented here are based on this approach.

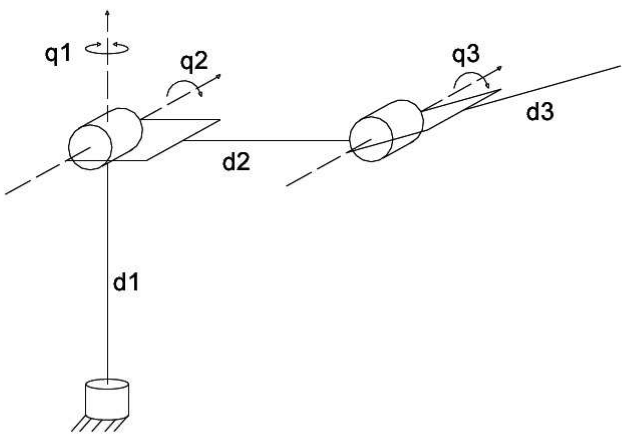

Fig. 3. Configuration for position of the center of the wrist

The point for decoupling is located at the end of the link d3 where the joint number q4 is placed. Because described arm has seven degrees of freedom split is not enough, the first part of the arm is still too sophisticated for geometrical calculations, so additional simplification has to be made. Calculations assume that the joint number q2’ is immobile in such position that links d1, d2’, d2”, and d3 are coplanar reducing to 6 number of degrees of freedom for the arm. In such situation length d2 is the sum of d2’ and d2”. Because the arm can be used also in forward kinematics mode this joint can be still used in this mode of operation.

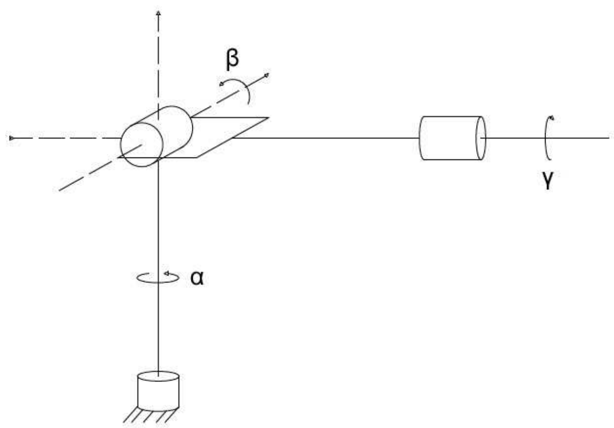

Fig. 4. Configuration of spherical wrist

Point of the split, marked as P2 in fig.5 is the center point of spherical wrist, and also the central point of the frame X1, Y1, Z1. Calculations begin with determining joint variables for servos 1, 2 and 3 representing angles q1, q2 and q3. These variables determine the position of the wrist. Second is to calculate variables specifying the orientation of the wrist.

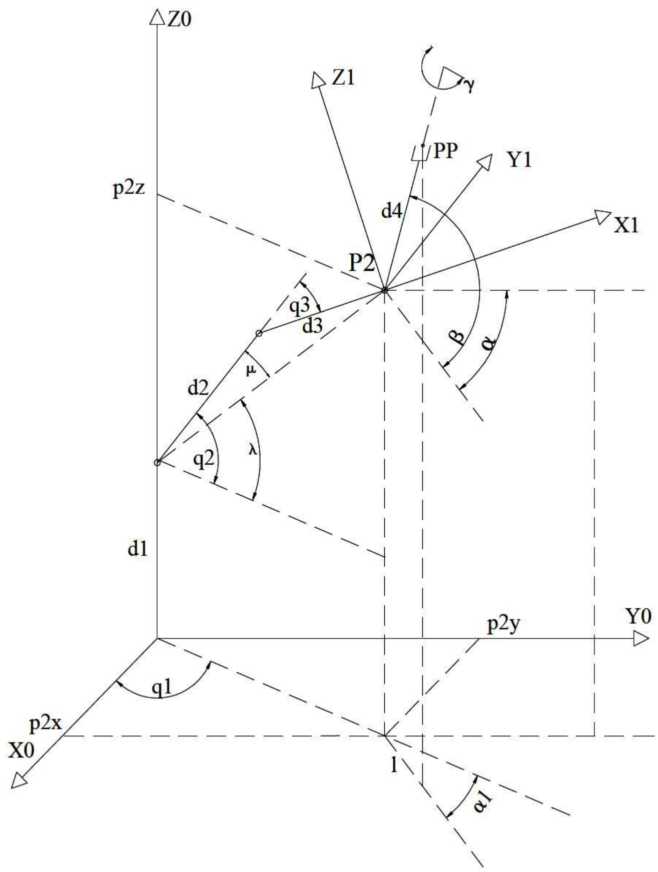

Fig. 5. Geometric schema of the manipulator

The input parameters for the calculations are coordinates of the point PP and angles α, β, and γ. Having coordinates *px*, *py* and *pz* first task is to calculate coordinates of point P2 as the center of the spherical wrist: p_2x, p_2y and p_2z.

$$p_2 x = px - l * \cos(\alpha) \tag{8}$$

$$p_2 y = py - l * \sin(\alpha) \tag{9}$$

$$p_2 z = pz - d_4 * \sin(\beta) \tag{10}$$

$$\text{where: } l = d_4 * \cos(\beta)$$

Calculation of the q1 angle is trivial.

$$\tan(q_1) = \frac{p_2 x}{p_2 y}$$
$$q_1 = \arctan(\frac{p_2 x}{p_2 y}) \tag{11}$$

If point P2 is placed on the Z0 axis equation (11) gives infinite number of results. In such situation the angle q1 can be left unchanged.

Next is to calculate the distance *d* from the joint 2 to point P2.

$$d^2 = p_2 x^2 + p_2 y^2 + (p_2 z - d_1)^2 \tag{12}$$

$$d^2 = d_2{}^2 + d_3{}^2 - 2 * d_2 d_3 * \cos(\pi + q_3) \tag{13}$$

From equations (12) and (13) it is possible to calculate angle q_3.

$$\cos(q_3) = \frac{p_2 x^2 + p_2 y^2 + (p_2 z - d_1)^2 - d_2{}^2 - d_3}{2 * d_2 d_3} \tag{14}$$

$$q_3 = \arctan(\frac{\sqrt{1 - D^2}}{D}) \tag{15}$$

for the joint 3 as shown in Fig. 5 or

$$q_3 = \arctan(\frac{-\sqrt{1 - D^2}}{D}) \tag{16}$$

for the joint 3 in the position below line d.
Now it is possible to calculate q_2 as the sum of angles λ and μ.

$$\lambda = \arctan(\frac{p_2 z - d_1}{\sqrt{p_2 x^2 + p_2 y^2}}) \tag{17}$$

$$\mu = \arctan(\frac{d_3 * \sin(q_3)}{d_2 + d_3 * \cos(q_3)}) \tag{18}$$

Of course while the joint 3 is below line d angle q_2 will be the difference

$$q_2 = \lambda - \mu \tag{19}$$

Next step is to calculate joint variables of spherical wrist. The geometric schema of the wrist is shown in the Fig. 6. Angles q1, q2 and q3 determine position of frame X1, Y1, Z1. Geometrical approach to calculations can be used also for inverse orientation kinematics i.e. for setting joint variables for the spherical wrist.
First it is to calculate two angles α_1 and β_1 as transformations of angles α and β into frame X1, Y1, Z1.

$$\alpha_1 = \alpha - q_1 \tag{20}$$

$$\beta_1 = \beta - (q_2 + q_3) \tag{21}$$

The assumption has been made that the X1 axis is coaxial to d3. The angle between planes X1Y1 and X0Y0 is q2+q3. The angle between planes X1Z1 and X0Z0 is q1.

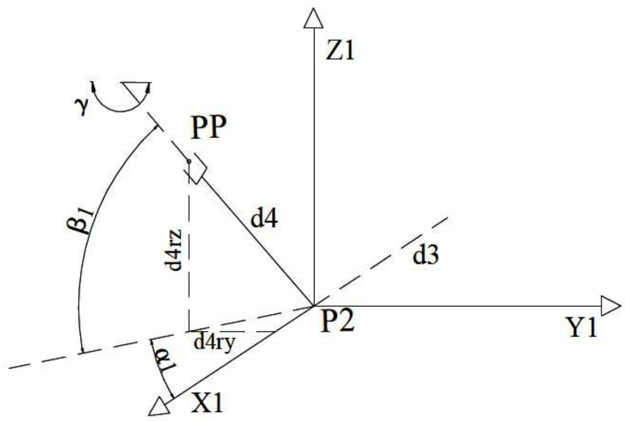

Fig. 6. Geometric schema of the spherical wrist

Now it is to calculate d4ry as the projection of d4 onto the X1Z1 plane and d4rz as the projection of d4 onto the X1Y1 plane.

$$d4ry = d4\cos(\beta_1)\sin(\alpha_1) \tag{22}$$

$$d4rz = d4\sin(\beta_1) \tag{23}$$

Having these values it is to calculate joint variables for spherical wrist.

$$q4 = \arctan\frac{d4ry}{d4rz} \tag{24}$$

$$q5 = \arctan\frac{\sqrt{d4ry^2 + d4rz^2}}{\cos(\beta_1)\cos(\alpha_1)} \tag{25}$$

Last angle q6 can be calculated with known value of q4.

$$q6 = \begin{cases} \gamma - q4 \Rightarrow \beta \in \left\langle -\frac{\pi}{2}; \frac{\pi}{2} \right\rangle \\ \pi - (\gamma - q4) \Rightarrow \beta \in \left\langle -\pi; -\frac{\pi}{2} \right\rangle \vee \beta \in \left\langle \frac{\pi}{2}; \pi \right\rangle \end{cases} \quad (26)$$

If both angles α_1 and β_1 are 0 then the q4 is unspecified and can be chosen arbitrary.

VII. Implementation

Implementation consists of two main parts, mechanical and electronic. Mechanical part has been developed with servos connected with links. Electronic part is made using 8-bit microcontroller with few additional elements.

A. Mechanical part

Mechanical implementation of the arm of the robot is based on servos. There are two groups of servos available, digital and analog. Digital servos have better stability and accuracy, but are much more expensive. Because the arm of the robot is first project of this type the less expensive analog servos have been used. Both types of servos are controlled using pulse width modulation signal. Frequency of this signal must be 50Hz that gives 20ms of the period. In typical servo during the 20 ms period the pulse of length from 0.5 ms to 2.5 ms should appear. Pulse of 1.5 ms sets the servo to neutral position equal to 90°, 0.5 ms pulse sets servo to 0° while 2.5 ms pulse sets servo to 180°. Servos are connected with links of lengths as shown in Fig 7.

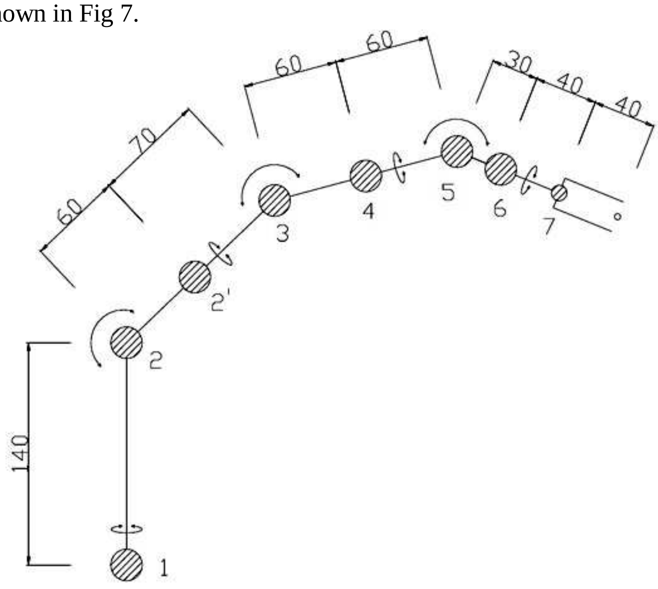

Fig. 7. Lengths of arm links (in mm)

Servos are loaded with the charge taken by the effector and by other servos. This load depends on the weight of servos and the length of links. The most loaded servo is the servo 2 which in the most stressed position must handle the load exceeding 4 kg. In Table 1 the loads between servos are presented. Servo 1 is for rotating the whole arm so it is not taken in the table. Joint 3 is on the axis of 2' so their loads are equal, the same for joints 4 and 5.

Table 1. Loads for servos [kg/cm]

No. of servo	2	2' and 3	4 and 5
2'	0,330	-	-
3	0,715	-	-
4	1,045	0,330	-
5	1,375	0,660	-
6	0,308	0,165	0,033
7	0,352	0,209	0,077
sum	4,125	1,364	0,110

Servos used in the project have parameters that allow effector to carry the load not exceeding 0,4 kg. Parameters of servos are presented in Table 2.

Table 2. Parameters of servos used

Servo		6,7	1,2',3,4,5	2
Dimensions [mm]	h	23	40.7	66
	l	12.2	19.7	30.2
	w	29	42.9	64.4
[kg/cm]		1.8	13	20
Speed [s/60˚]		0.13	0.20	0.20
Power supply [V]		4.8-6	4.8-7.2	4.8-7.2
Weight [g]		13	55	160

B. Electronic part

Electronic part of the project is based on 8-bit microcontroller Atmega 644. This integrated circuit is equipped with 64 kB memory for the program, 4 kB memory for data and 32 input/output ports. The block schema of the controller is presented in the Fig. 8.

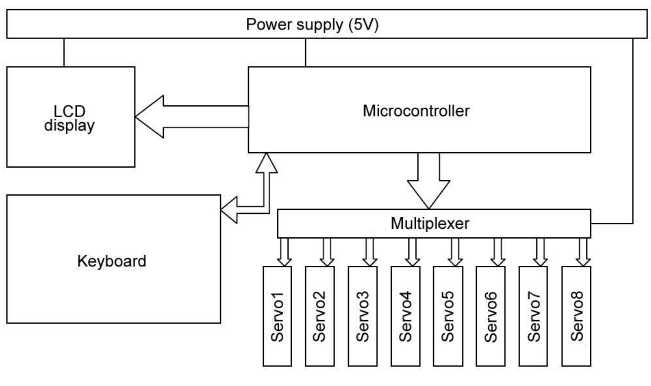

Fig. 8. Block schema of the controller

The arm of the robot can be controlled directly with keyboard. Every servo has two keys, one for right and one for left direction of the rotation. Arm can be also controlled from PC computer using serial port. Current state of the controller is shown in the alphanumerical LCD display that has 2 lines of 16 characters. Servos are connected to microcontroller with 74HTC238 multiplexer working with rotating logic one on the outputs. This multiplexer can reduce usage of microcontroller outputs from 8 to 4.

Software for microcontroller has been written in C using WinAVR programming environment. The main function of the software is sending periodically pulses to all servos. The frequency of pulses must be stable so procedures are triggered by interrupts generated by two hardware timers. Timer number 1 is used to generate 2,5 ms periods, timer number 2 generates pulses of variable length for controlling all servos. Whole cycle consists of eight 2,5 ms periods one for each servo. During each period one servo gets 0,5 – 2,5 ms pulse. Sample cycle is presented in Fig 9.

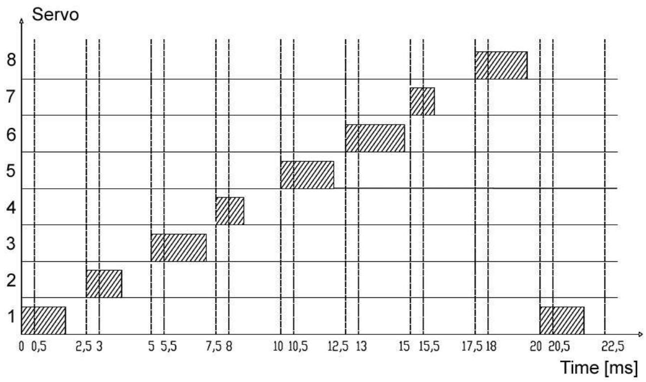

Fig. 9. Sequence of pulses

The keyboard is read during interrupt generated by timer number 1. LCD display is controlled by main function. Calculations are also done in main function, results are available for next cycle.

VIII. Summary and Conclusions

In this article the construction of the arm of the robot has been presented. The rough set theory is proposed to be applied to recognize the scenery of the robot. It should be understood that when some sample returned several partial classification verdicts from different constituent conditional clauses of the decision algorithm for the final verdict for this sample the decision was based on majority of verdicts when possible and in the case of tie verdict it is classified as undecided. While comparing coordinates of points in the input discrete space for learning and testing samples it becomes evident that the cases of incorrect or undecided classification happened for samples that were either absent or poorly represented in the training data set and thus the classifier had insufficient information for creating some decision rule with correct classification dedicated to them. In this kind of situation the Decision Algorithm certainly can fail. This observation brings immediate conclusion as to how the classification properties of rough set-based classifier can be enhanced. The higher coverage of the input space by the training samples the higher chance of correct classification. Another direction of future research is also indicated by more detailed considerations of the choice of relative reducts and relative value reducts and how this selection reflects upon classification procedure.

To make the arm moving two algorithms have been implemented. One of them, forward kinematics algorithm does not need high computational power. Second, inverse kinematics algorithm in matrix version is too sophisticated for the microcontroller used in the project. To allow 8-bit one chip microcomputer to control the arm the simplified method of calculations has been used. Geometrical approach with split the calculations into two parts is the good choice for small 8-bit microcomputers. The arm built is intended to be the base for next research on control algorithms with usage of different methods and more powerful processors.

References

[1] Z. Pawlak, “Rough Set Rudiments”, *Institute of Computer Science Report, Warsaw University of Technology, Poland*, pp. 1–47, 1996.

[2] L.A. Zadeh, “A fuzzy-algorithmic approach to the definition of complex or imprecise concepts”, *International Journal on Man-Machine Studies*, vol. 8, pp. 249-291, 1976.

[3] P. Jirava and J. Krupka, “Classification model based on rough and fuzzy sets theory”, *Proceedings of the 6th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, pp. 198–202, 2007.

[4] K. Cyran, Rough sets in feature extraction optimization of images obtained from intermodal interference in optical fiber, Proc. SPIE, 3744, pp. 241-252, 1999.

[5] K. A. Cyran, L. R. Jaroszewicz, A. Mrózek, Optical fiber and genetically optimized computer-generated hologram for force detection and classification, Proc. SPIE, 4238, pp. 234-238, 1999.

[6]K. A. Cyran, A. Mrózek, Rough sets in hybrid methods for pattern recognition, Int. J. Intell. Syst., 16 (2), pp. 49-168, 2001.

[7] M. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, Robot Motion: Planning and Control. Cambridge, MA: The MIT Press, 1982.

[8] O. Khatib, A Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation, IEEE J. Robotics & Automation, vol. 3, no. 1, pp. 43-53, 1987.

[9] D. DeMers and K. Kreutz-Delgado, “Solving Inverse Kinematics for Redundant Manipulators,” in Neural Systems for Robotics, O. Omidvar and P. v. d. Smagt ed., Academic Press, 1997.

[10] K. W. Lilly and D. E. Orin, “Alternate Formulations for the Manipulator Inertia Matrix,” International Journal of Robotics Research, vol. 10, pp. 64-74, February 1991.

[11] G. Rodriguez, A. Jain and K. Kreutz-Delgado, “A Spatial Operator Algebra for Manipulator Modelling and Control,” Int. J. Robotics Research, vol. 10, no. 4, pp. 371-381, 1991.

[12] X. Wang and J.-P. Verriest, A geometric algorithm to predict the arm reach posture of computer-aided ergonomic evaluation, J. Visualization Comput. Animat. 9, pp 33–47. 1998.

[13] M. Raghavan and B. Roth, Kinematic analysis of the 6R manipulator of general geometry, in Proceeding of the 5th International Symposium on Robotics Research, 1990.

[14] M. Doumpos and A. Salappa, “Feature selection algorithms in classification problems: an experimental evaluation”, *WSEAS Transactions on*

Information Science & Applications, vol. 2, no. 2, pp. 77–82, 2005

[15] A. Caballero and K. Yen and Y. Fang, "Classification with diffuse or incomplete information", *WSEAS Transactions on Systems and Control*, vol. 3, no. 6, pp. 617–626, 2008.

[16] K. Cyran, U. Stańczyk, Indiscernibility relation for continuous attributes: application in image recognition, Lecture Notes in Artificial Intelligence, 4585, pp. 726-735, 2007.

[17] K. A. Cyran, Modified indiscernibility relation in the theory of rough sets with real-valued attributes: application to recognition of Fraunhofer diffraction patterns, Transactions on Rough Sets IX, Lecture Notes in Computer Science, 5390, pp. 14-34, 2008.

[18] K.A. Cyran, Quasi Dominance Rough Set Approach in Testing for Traces of Natural Selection at Molecular Level, in K.A. Cyran et al. (Eds.) Advances in Intelligent and Soft Computing, Springer, 59, 163-172, 2009.

[19] R.B. Perez, A. Nowe, P. Vrancx, Y. Gomez, and D.Y. Caballero, "Using Ant Colony Optimization and rough set theory to feature selection", *WSEAS Transactions on Information Science & Applications*, vol. 2, no. 5, pp. 512–517, 2005.

[20] M.J. Moshkow, A. Skowron, and Z. Suraj, "On Covering Attribute Sets by Reducts", in M. Kryszkiewicz, J.F. Peters, H. Rybinski, and A. Skowron, eds., *Lecture Notes in Artificial Intelligence*, vol. 4585, Springer-Verlag, Singapore, pp. 175–180, 2007.

[21] K. Asano, Human arm kinematics, in Proceedings of the 5th International Symposium on Robotics Research, 1990.

[22] K. Kondo, Inverse kinematics of a human arm, J. Robot. Systems 8(2), 1991, 115–175.

[23] M. W. Spong, M. Vidyasagar "Robot Dynamics and control," Wiley, New York, 1989.

Krzysztof Tokarz, PhD – graduated computer science in 1996 at Faculty of Automatic Control, Electronics and Computer Science (FACECS) at Silesian University of Technology (SUT), currently a Head of the Unit of Microinformatics and Automata Theory in the Institute of Informatics in SUT. He is experienced academic teacher in area of construction and programming of embedded systems, microprocessor systems and wireless computer networks. Dr Krzysztof Tokarz is the author and co-author of many publications about microprocessor systems and wireless data transmission printed in journals and presented on international conferences. Some latest publications described in details methods of improvement of GPS navigation with implementation as the prototype in portable embedded device. He was a supervisor of over 20 MSc theses focused mainly on microprocessor based devices and systems. He is also the designer of embedded devices, successfully used in the industry, for example the data collecting device built for Intelix Ltd, GSM communication controller built for Proloc Ltd. He cooperates also with NFZ (Health Fund) as the expert in topic of electronic card systems.

Christian Manger, Dipl. Ing. - He finished study of computer science on the Technical University of Vienna with a diploma as "Dipl. Ing.". Currently he is a software developer and an expert in mobile applications. He has worked several years in analysis and development of financial software and software in the field of material management. After that he continued to work in the field of IT-Solutions for the health care market and also for environmental services. Currently working on a product for constructionary industries in a world wide used scenario. His interests include mobile communication systems with GPS, eLearning platforms, and robotics.