Journal of Achievements in Materials and Manufacturing Engineering
VOLUME 25
ISSUE 1
November
2007

The system for simulation and off-line, remote programming of the Mitsubishi Movemaster RV-M1 robot

J. Świder, K. Foit*, G. Wszołek, D. Mastrowski
Institute of Engineering Processes Automation and Integrated Manufacturing Systems, Faculty of Mechanical Engineering, Silesian University of Technology, ul. Konarskiego 18a, 44-100 Gliwice, Poland
* Corresponding author: E-mail address: krzysztof.foit@polsl.pl

Received 15.03.2007; published in revised form 01.11.2007

Manufacturing and processing

ABSTRACT

Purpose: of this paper: The paper presents a concept of a system for off-line programming, simulation and remote supervision of the Mitsubishi Movemaster RV-M1 robot.
Design/methodology/approach: In the software development process a robot installed in institute's laboratory has been used. The software has been coded in a high level programming language in a manner that allows future improvements.
Findings: As a result of the work, an initial version of software has been created. It has been developed for Windows platform.
Research limitations/implications: The initial version of software has been dedicated to Movemaster RV-M1 and only this model is available for simulation purposes in virtual environment. Visualization processor recognizes only selected commands of MELFA BASIC. Future aims include creation of more flexible form of the software, which could be used on different operating system platforms and can handle other types of robots.
Practical implications: Presently, the software is used internally for didactic purposes, during conduction of laboratory classes.
Originality/value: Existing commercial systems are more or less hermetic. Developed software is intended to use standards for model representation and simulation (like OpenGL API).
Keywords: Automation engineering processes; Robotics; Simulation; Remote supervision of robot; Education

1. Introduction

Nowadays robotics needs new solutions in the field of programming, control and supervision of robots. It is necessary to adjust a manipulator in its working environment and find a suitable method of control. It is connected with selection of the proper equipment and tests of a robot program. Execution of the tests in a real environment is often expensive and time consuming, but most of the work can be done using virtual reality simulation programs [1-3]. On the other hand, transcending the boundaries of industrial application, a human being is often substituted by a robot when acting in dangerous or inaccessible environments [4,5]. These robots have often limited autonomy, acting as an extension of operator's limbs and senses of sight and hearing. The issues concerning remote control of this type of robots are the subjects of telerobotics. As an example, mobile robots for debris and explosionthreaten territory inspection can be mentioned. This group also includes the robots used in outer space or in contaminated environment. In such conditions it is impossible to control the manipulator without using video supervision devices.

Telerobotics is not widely used in the industry. A production process is set-up using off-line tests, executed in virtual reality, then on-line tests, directly on manufacturing line or in robot's cell are carried out. After the tests, only basic supervision of working units is needed, often limited to simple inspection [6-14].

The aim of the work is to present the preliminary results of development of universal and modular system, that can allow remote programming and supervision of the Mitsubishi Movemaster RV-M1 robot. Such systems, being presently available on the market, are often very complex and expensive. The overall cost of their exploitations is even greater due to necessity of buying proper computer equipment and carrying out the cycle of staff training. Developed software is intended to be as simple as possible, using intuitive user interface. The manner of programming the virtual RV-M1 robot model is almost equal like for the real robot, but some functions are still not implemented. The current version of the software has been developed with participation of the student.

A similar project, developed for Mitsubishi Movemaster RV-M1, is the JRobot system [15]. It should be mentioned here, that the ideas of both systems are not connected in any way.

In the further part of the paper general assumptions and the present state of work will be presented.

2. General description of the system

2.1. Description of the Movemaster robot

During the work on software development, the attention has been paid to Mitsubishi Movemaster RV-M1 robot due to its availability in the Institute's laboratory and simplicity of programming. Because of simple communication protocol with the robot's controller, the risk of making mistakes on communication level has been minimized, then it has been easier to determine the reason of improper program's operation.

The Mitsubishi Movemaster RV-M1 robot is a typical industrial machine. It has five-axis manipulator with 1,2 kg lifting capacity. The controller of the robot is equipped with input/output and communication cards, which give the possibility to communicate with external devices and a computer. The link between the controller and a computer can be established using Centronics or RS-232 port [16]. The second one is preferred due to be a standard in communication between systems of manufacturing processes automation. The robot can be programmed using a teachbox or a computer, but using the first method is less effective due to unavailability of some advanced functions. In case of using a computer, no special software is required. The robot uses the MELFA BASIC language for program coding and due to representation of the code in the form of text, a program can be written in any text editor. It is also possible to send the code directly to the robot using almost the same manner as for sending text to printer. Robot can also work in the batch mode, when command has been executed immediately after sending it to a robot's controller. Programming in the MELFA BASIC language is not time-consuming task and allows quick creation of own communication unit in any programming language, including script interpreters. The simplest code managers could be written using only DOS batch file commands.

2.2. Project of the system

The initial idea of the system was an intention to create an application, which would generate program for the robot using a CAD drawing, in which a trajectory and gripper status have been coded [17-21]. It has been decided that a state of the gripper will be represented as a point (in the matter of an object on the drawing) placed in a coincidence point of two sections of the line representing a gripper's trajectory. The program uses its own dictionary to identify and read proper data from drawing file. The dictionary can be modified, giving possibility of adapting the application for the other type of a robot or recognizing the other input data formats. The general idea of the translation program, presented as a block scheme, is shown in Figure 1 [17-21].

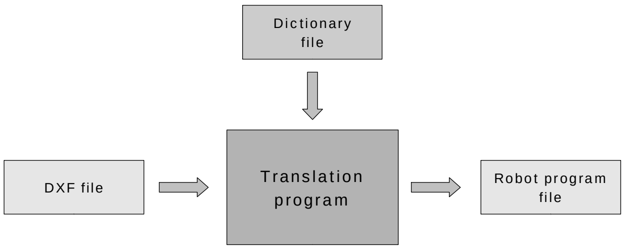

Fig. 1. The general idea of the translation program

In the second phase of development, an idea to realize the transfer of the program from a computer to the robot installed in the other room has been realized [17,18,20,21]. Due to inability to connect the robot's controller directly to the computer network, it was necessary to use an additional computer to mediate between the remote station and the robot. This computer, equipped with a network card, fulfills the role of a server, which receives the code of the robot's program from the client station and transfers it to a controller connected via RS-232 or Centronics port. Moreover, this solution has given the possibility to use a camera to observe the robot's environment and to transmit the picture to the client computer. In order to achieve the aim, two possible ways have been considered: the use of WWW server and CGI or scripting languages or use of JAVA Virtual Machine. The JAVA gives more elasticity in reference to use the system on different platforms and enables the possibility to setup the link for real-time data transfer. Schematic diagrams of both solutions are shown in figures 2 and 3. In the case of the system based on JAVA a possibility of use of a WWW browser as a client has been considered. It would require a WWW server, but on the other hand it would allow to aggregate the best properties of both of the remote control methods and the system configuration procedure could be omitted. In order to develop proper algorithms, the initial version of the software uses the real-time data transfer model, but it is not coded in JAVA.

Before sending a program to the remote robot, the code should be tested to ensure that it will be executed in a proper manner and no element on workplace would be damaged by a manipulator. The test can be carried out in a virtual reality, using a virtual robot. This conclusion has led to creation of the simulation module [17-21].

The project is still in development phase. Individual stages will be discussed in further part of this paper.

3. Phases of development

3.1. CAD TO RV-M1 translation software

The translation software has been the first result of the work. This simple application has been used to convert the trajectory coded in AutoCAD drawing file to the program written in MELFA BASIC [19]. The software has been based on a concept shown in Figure 1. It uses a dictionary file, which could be modified by a program operator. Because the application has been dedicated to the Mitsubishi Movemaster RV-M1 robot, the dictionary contains only these elements, which are needed for proper trajectory decoding. The drawing should be written in DXF text format. The translation application operates in global coordinates, so it is recommended to begin the path from the point having all coordinates set to zero (0,0,0). The trajectory is created using a line drawing tool and the subsequent gripper's positions are defined as an endpoint of individual segments of the line. The gripper status is defined by a point object inserted in coincidence point of two lines sections. Insertion of the point object means the change of the gripper status. The translation program does not have an advanced level of errors handling and has no possibility to verify drawings data in order to translate them to a proper control program. A revision of decoded parameters could be done when the program generation process is finished. An user can edit a program directly in application window. It is also possible to revise the coordinates of decoded points and change the gripper status. The main window of the CAD TO RV-M1 program is shown in Figure 4. The translation program has also a simple interface that provides fundamental communication functions. The link with the robot's control unit could be established using the RS-232 port. Access to a communication panel is possible after opening the "Program robota" ("Robot's Program") tab– compare with Figure 5 – and selecting the "Pokaż Robot Panel" ("Show Robot Panel") button. Besides the configuration of the link, the panel offers two additional functions: downloading the program or positions from a controller or sending the program code to a controller. It should be mentioned that manipulator's positions could be defined in the program code, as well as they can be entered to the memory using the robot's teachbox. In this manner, the program is completely independent on stored positions. In a case of CAD TO RV-M1 the positions are defined directly in the code. In the tab, the buttons that allow to write the program to the file or to read the code from a disk have been also located.

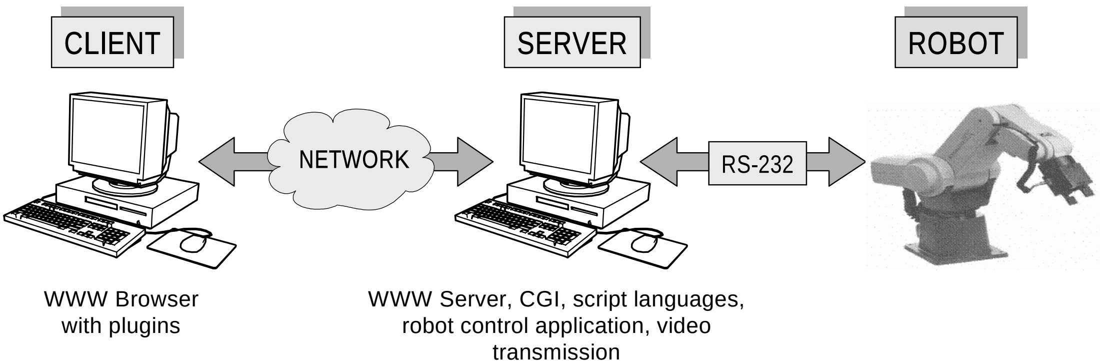

Fig. 2. The concept of the system based on WWW server and scripting

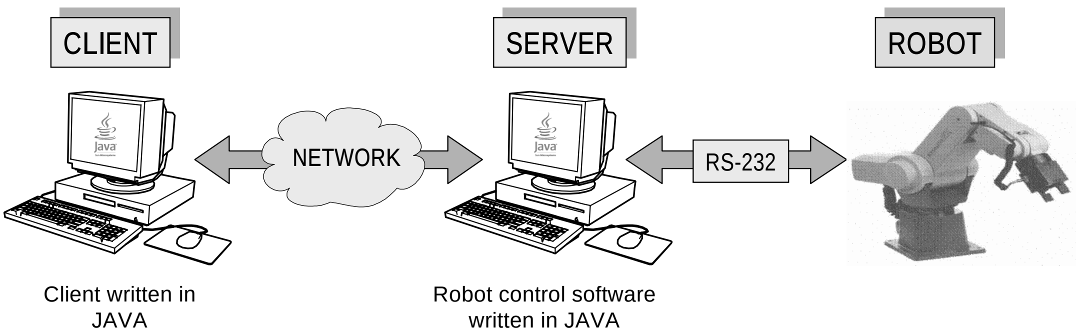

Fig. 3. The concept of the system based on JAVA

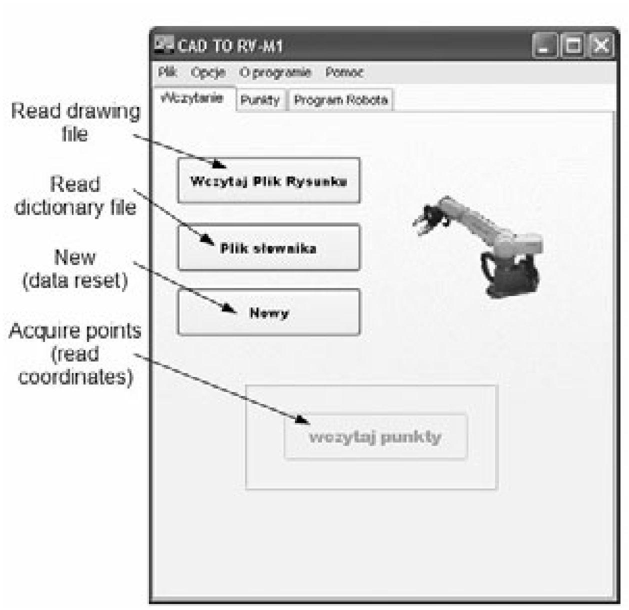

Fig. 4. The main window of the CAD TO RV-M1 program

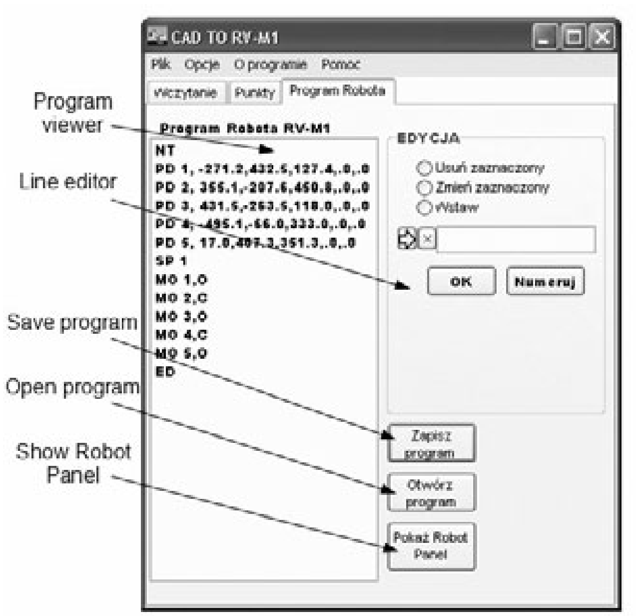

Fig. 5. The tab "Program Robota" ("Robot's Program") with generated code and control buttons

3.2. Dedicated simulation software

Execution of wrongly coded program can cause safety problems, which could lead to damages on the robot's workplace. The possibility of testing the robot's program in virtual environment considerably minimizes the risk connected with potential dangerous situations like collisions or inadequate object handling in real environment. The next stage of the project has been connected with creation of the simulation software, which has been dedicated for the Mitsubishi Movemaster RV-M1 robot [17-21]. The program has been patterned after Denford Virtual Robot software intended for educational purposes. The created application has been named ROBO and combines a virtual robot workplace with a simple control panel, a program editor and simulation management tools. The workplace has been represented as a virtual room, where robots with equipment can be placed. The visualization part of the program has been based on OpenGL graphics standard due to its implementation on varied operating systems platforms. Data needed for creation of the virtual robot model have been acquired from a CAD application. Initial version of the ROBO application had not implemented a communication module - the program code had to be created directly in the program editor or imported from DXF file, in the same manner like in CAD TO RV-M1 application.

The ROBO program has been treated as a core of further system development.

3.3. Prototype of the system

The prototype of the system consists of three applications. The main program has been based on ROBO application, which has been enhanced with additional simulation capabilities and network communication module [18-21]. The other two applications have been developed for supervision and communication needs. The first one, named KAMROB is intended to send the video stream from the camera to the client station and to mediate between ROBO application and the real robot. The robot and the camera should be connected to the same computer, which fulfills the role of a server. The second application, named SERWER, is working as an integration module between ROBO and KAMROB by means of network communication – it is a server program. Such division of tasks is assumed only to achieve modularity and flexibility of the system. The software has been developed in high level programming language and compiled on Windows platform. The system is dedicated to the Mitsubishi Movemaster RV-M1 robot. In the further part of the paper main properties of software will be described in details.

4. Description of the current version of the system

The system has been designed in a manner, which ensure maximum level of flexibility. It is also possible to use it in education process, because the simulation module can be also executed individually, without necessity to connect it to a real robot. This option is useful in situation, when only one robot is available for many client stations.

4.1. Simulation and off-line programming module (program ROBO)

The modified version of the ROBO application has retained all functions introduced in the initial version, described in section 3.2. The main novelty is a possibility to define an object of manipulation and ability of use up to five virtual robots in simulation area, which can work simultaneously (this limitation is introduced only because of graphics engine efficiency). In addition, the new elements of workspace have been introduced: tables and racks [18-21]. The position of the virtual camera could be adjusted using keyboard. The main window of the new version of the ROBO program is shown in Figure 6.

In the control panel are the sliders for setting up the position of the individual manipulator's arm. It is also possible to enter positions into text fields as numbers. At the bottom part of the control panel, a program editor is located. An user can enter the program manually or load it from the file. Application has also a function for automatic adding positions to the program – they have gained from simulation window. An additional option is an ability to read positions from a CAD drawing file, using a dictionary and to generate a program as it has been described in section 3.1.

The possibility to locate up to five robots on the workplace is connected with very important and interesting property of the program. The ROBO application can be executed on different computers in a manner that each of them controls the one of the virtual robots, placed in simulation area of other application. There is also the possibility to use a mixed mode, in which one of the virtual robots is connected with real robot. The example of such configuration is shown in Figure 7.

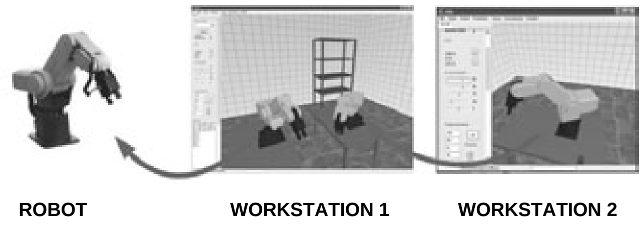

Fig. 7. The ROBO application operating in mixed mode. Workstation 1 controls the real and the virtual robot

The another property of the ROBO application is a possibility to operate a real and a virtual robot simultaneously. A program is executed in a special manner: every change of the virtual robot arms position causes an adequate displacement of the real manipulator. Moreover, the video stream from a remote camera can be shown.

Although the ROBO program could be run individually, some features require the TCP/IP protocol. It is also needed to execute the SERWER application on any of the network stations, whereas for communication with the real robot and video transmission the KAMROB program should be started.

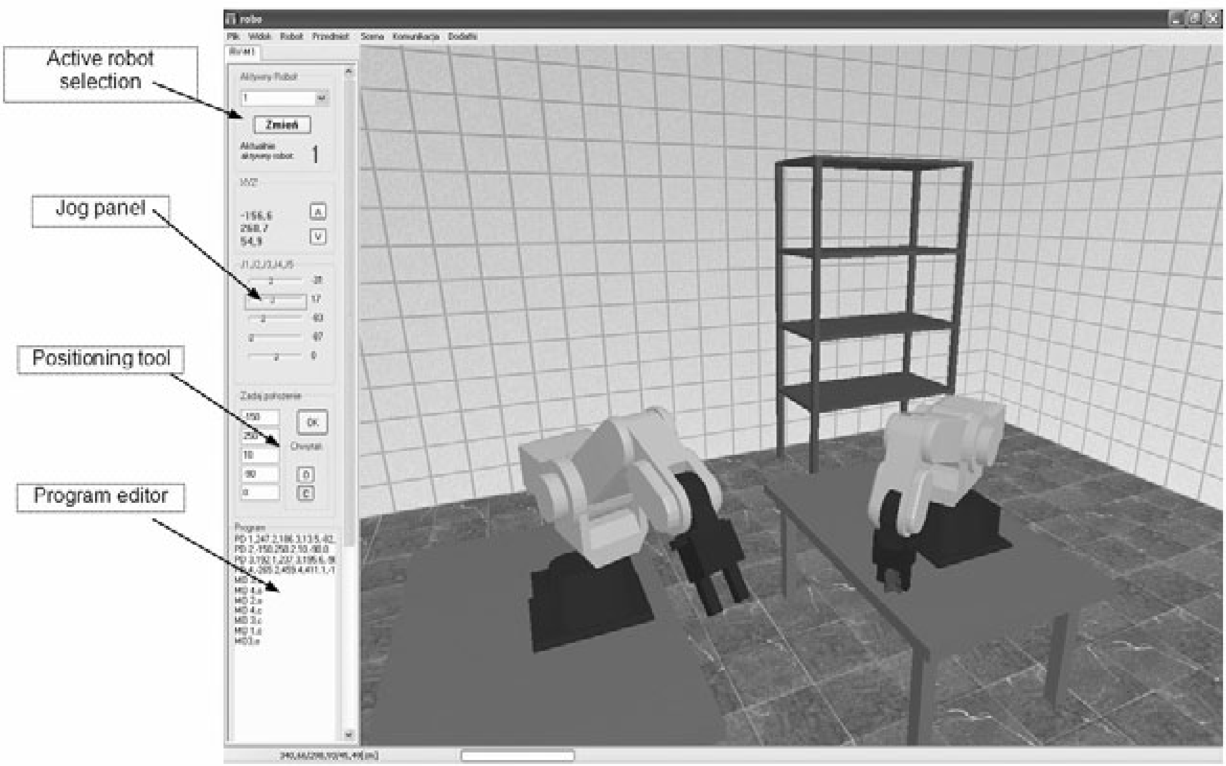

Fig. 6. The main window of the ROBO program with two robots in the simulation area

4.2. Video transmission and robot control software (program KAMROB)

The main function of the KAMROB program is to provide the communication between a server and a real robot [18-21]. Moreover, the application can control a camera and send the video stream to the client station. The KAMROB program should be executed on the computer located juxtapose to the real robot's workplace. It is necessary to connect the robot's controller via RS-232 port. The camera should be connected to the same computer. The configuration of the program is not a time consuming activity and requires only a couple of parameters. It needs to set up the serial transmission parameters, network configuration (an IP address of the computer, where the SERWER application is running – it could be loopback interface address when server is running on the same network station) and an unique identifier for the real robot, required by the ROBO application. The window of the KAMROB program, shown in Figure 8, consist of two areas: the one showing a picture from the amera and the second, where all operations are logged – including communication with a robot and a server. Using the camera requires an installation of proper drivers, which are supplied by a hardware manufacturer. It should be noticed that the video streaming function is in experimental phase and is could not be very efficient on slower computers.

The KAMROB application can be also used without a camera, acting as a simple gateway between a computer network and the robot's controller. In this mode, it can be installed on an older computer with less effective processor, because its only task is to communicate with SERWER application (Figure 9).

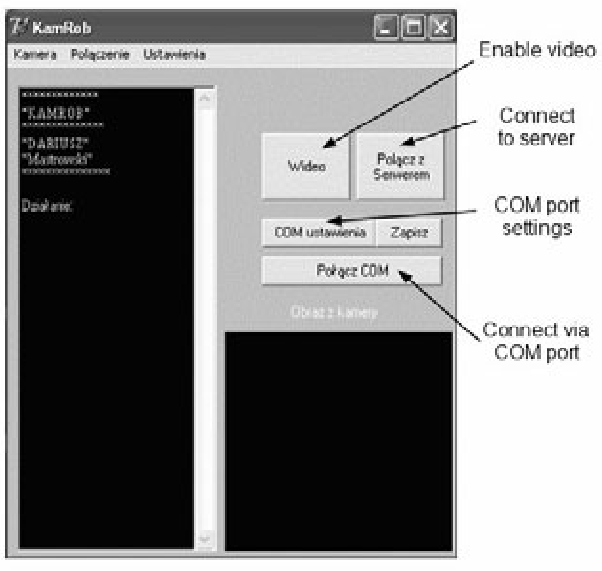

Fig. 8. The main window of the KAMROB application

4.3. Server application (program SERWER)

The SERWER application integrates the ROBO application running on different workstations by means of network communication. It also serves the transmission with the KAMROB program [18-21]. The application does not require any configuration. It uses only three network ports, which are assigned to video transmission, data exchange and diagnostic messages including errors during communication with the real

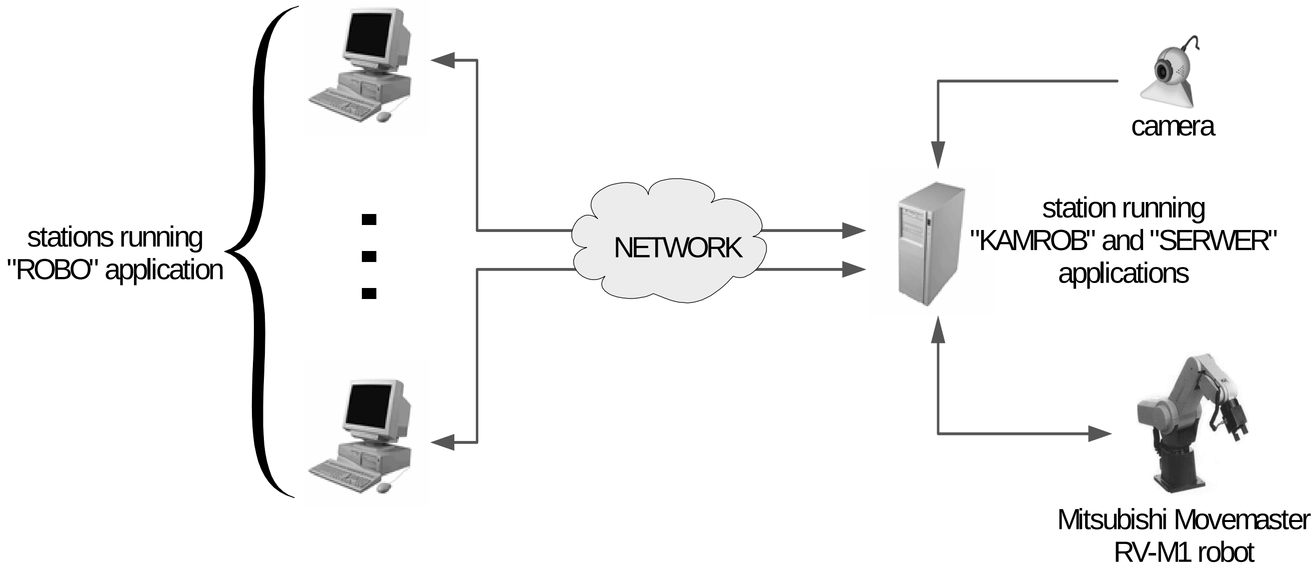

Fig. 9. The general network communication scheme of the developed system

robot. The port numbers are constant and cannot be changed by an user. Similarly to KAMROB, SERWER application has also a message area, where status of communication tasks is displayed. The main window of the program is shown in Figure 10.

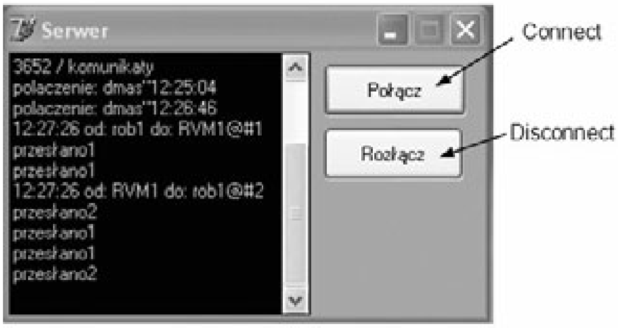

Fig. 10. The window of the SERWER module

The SERWER and KAMROB applications could be assembled into one, but the second one is needed only when the real robot is used. In the other case, two separate programs give more flexibility in network configuration.

5. Conclusions

The initial version of the system has been intended for didactic purposes. It has been tested in the Institute's Laboratory using the real robot. The plans for the future include a development of unrealized functions, mentioned in section 2. In the near time, it is scheduled to modify the software in a manner that would allow handling other types of robots.

Acknowledgements

This work has been conducted as a part of the research projects No. 4 T07C 01827 supported by the Committee of Scientific Research in years 2004-2007.

References

[1] M. Weyrich, P. Drews, An interactive environment for virtual manufacturing: the virtual workbench, Computers in Industry 38 (1999) 5-15.

[2] V.G. Duffy, G. Vincent, F.F. Wu, P.P.W. Ng, Development of an Internet virtual layout system for improving workplace safety, Computers in Industry 50 (2003) 207-230.

[3] R. Navon, A. Retik, Programming construction robots using virtual reality techniques, Automation in Construction 5 (1997) 393-406.

[4] A. Morecki, J. Knapczyk, Robotic's Basics – Theory and elements of manipulators and robots, Second Edition, WNT, Warsaw, 1994 (in Polish).

[5] M. Olszewski, Manipulators and Industrial Robots. Second Edition, WNT, Warszawa, 1992 (in Polish).

[6] T. Otrębski, Remote monitoring and control using GPRS service, Control Engineering Poland 8 (2006) 62-64 (in Polish).

[7] A. Presher, The network control of machines – in the spotlight, Design News 9 (2006) 48-51 (in Polish).

[8] S.F. Chan, R. Kwan, Post-processing methodologies for off-line robot programming within computer integrated manufacture, Journal of Materials Processing Technology 139 (2003) 8-14.

[9] G. Yasuda, Distributed autonomous control of modular robot systems using parallel programming, Journal of Materials Processing Technology 141 (2003) 357-364.

[10] G.G. Kost, R. Zdanowicz, Modeling of manufacturing systems and robot motions, Journal of Materials Processing Technology 164-165 (2005) 1369-1378.

[11] A.J. Alvares, J.C. Espindola Ferreira, WebTurning: Teleoperation of a CNC turning center through the Internet, Journal of Materials Processing Technology 179 (2006) 251-259.

[12] T.S. Mujber, T. Szecsi, M.S.J. Hashmi, Virtual reality applications in manufacturing process simulation, Journal of Materials Processing Technology 155-156 (2004) 1834-1838.

[13] Q. Peng, F.R. Hall, P.M. Lister, Application and evaluation of VR-based CAPP system, Journal of Materials Processing Technology 107 (2000) 153-159.

[14] W.B. Lee, C.F Cheung, J.G Li, Applications of virtual manufacturing in materials processing, Journal of Materials Processing Technology 113 (2001) 416-423.

[15] Jrobot, A Java interface for Mitsubishi RV-M1 Robot. http://jrobot.sourceforge.net

[16] Mitsubishi Movemaster RV-M1 User's Manual.

[17] K. Foit, J. Świder, The project of a platform-independent, off-line programming system for industrial robots, Proceedings of 7th International Scientific Conference on "Computer Integrated Manufacturing - Intelligent Manufacturing Systems" CIM'2005, Gliwice-Wisła 2005, 62-65.

[18] K. Foit, J. Świder, D. Mastrowski, The project of an off-line, remote programming system for Mitsubishi Movemaster industrial robot, Proceedings of XIII International Scientific and Technical Conference "Machine-Building and Technosphere of the XXI", 2006, Sevastopol, Ukraine, Paper Collection, 4, 252-255.

[19] D. Mastrowski, The system for remote programming, supervision and simulation of robots, cooperating with Mitsubishi Movemaster RV-M1 Robot, MSc thesis, Gliwice 2006 (in Polish).

[20] J. Świder, K. Foit, G. Wszołek, D. Mastrowski, The off-line programming and simulation software for the Mitsubishi Movemaster RV-M1 robot, Proceedings of the 12th International Scientific Conference "Contemporary Achievements in Mechanics, Manufacturing and Materials Science" CAM3S'2006, 391-394.

[21] J. Świder, K. Foit, G. Wszołek, D. Mastrowski, The off-line programming and simulation software for the Mitsubishi Movemaster RV-M1 robot. Journal of Achievements in Material and Manufacturing Engineering 20/1-2 (2007) 499-502.