Design and Implementation of a User Friendly JAVA based GUI Software for Manoeuvring Mobile Robot

T.C. Manjunath

Research Scholar (doing Ph.D), IEEE / SPIE / IOP Student Member, ISOI / SSI / ISTE Life Member,
Interdisciplinary Programme in Systems and Control Engineering, 101 B, ACRE Building,
Indian Institute of Technology Bombay, Powai, Mumbai-400076, Maharashtra, India.
Email : tcmanju@sc.iitb.ac.in   tcmanjunath@rediffmail.com ; Web : http://www.sc.iitb.ac.in/~tcmanju
Phone : +91 22 25764884 ;   FAX : +91 22 25720057

Abstract

The main objective of this work is to develop a user friendly GUI developed in JAVA language for manoeuvring a mobile robot named 'OCTAGON' with algorithms for obstacle avoidance in its path of motion. A user-friendly module is designed and successfully implemented for the operation of the entire system to do some specific operations. OCTAGON employs a sophisticated application - controlling interface created in JAVA, as it is a fantastic programming language for any application software development.

The software module application facilitates the user interaction with OCTAGON and has many in built features such as the security and authentication. When the java code is executed, the default GUI screen appears as shown in the Fig. 2 and all the motors status is displayed. For activating a particular motion, we can navigate to its page by selected the left hand menu. The software is designed for maximum robot control & working efficiency. It is so designed such that the user can have complete control over each movable part of the robot.

Also, the user can easily maneuver the robot & make it traverse a path towards the object to be picked. He or she can then manipulate the different limbs of the robotic arm so that the gripper comes closer to the object & eventually picks it up. Throughout this process the software interface guides the user through the usage of various parts of the robot & provides him with responses from the robot. Given the time on hand, the ease of programming and power of JAVA, it was an ideal choice for use in designing our application interface to the octagon.

The software is designed for maximum robot control & working efficiency. It is so designed that the user can have complete control over each movable part of the robot. Also the user can easily maneuver the robot & make it traverse in a path towards the object to be picked. He/she can then manipulate the different limbs of the robotic arm so that the gripper comes closer to the object & eventually picks it up. Throughout this process, the software interface guides the user through the usage of various parts of the robot & provides him with responses from the robot.

1. Introduction

Imagine a day in your life when you wake up in the morning and find a machine walking up to you and saying "GOOD MORNING SIR ! Have a cup of tea". How would you respond to such a situation ? With so much progress made in the filed of science, engineering and technology, this dream is absolutely realizable in the automation age. Keeping in pace with the current technology, we have designed and fabricated a mobile robot named OCTAGON as shown in Fig. 1, which is basically a moveable mobile trolley on which the articulated robotic manipulator arm is mounted to perform Pick aNd Place (PNP) operations [1].

These operations are defined by the program typed in by the user using a user friendly language specifically developed for this robot in JAVA. In this paper, we concentrate only on the software design for controlling and maneuvering the octagon. The design and fabrication was presented in the previous conference of AROB-2006. This work was taken up as a research project work by the post-graduate students of VJTI under the guidance of the author from IIT. The paper is organized as follows. First, a introduction to the designed robot is given. This is followed by the software design, control algorithms and the conclusions [2].

Fig. 1   The designed OCTAGON

2. Software Design

OCTAGON system employs a sophisticated application controlling interface created in JAVA as it is a fantastic programming language for any application software development. The developed software module for controlling the robot is shown in Fig. 5. The software module application facilitates the user interaction with OCTAGON and has many in built features [8]. The software is designed for maximum robot control & working efficiency. It is so designed that the user can have complete control over each movable part of the robot. Also the user can easily maneuver the robot & make it traverse a path towards the object to be picked [3]. He can then manipulate the different limbs of the robotic arm so that the gripper comes closer to the object & eventually picks it up. Throughout this process, the software interface guides the user through the usage of various parts of the robot & provides him with responses from it.

The features of the developed software for the robot are as follows :
- Start up screen for the software application.
- Features a complete HELP file, which can be invoked from the menu provided.
- Has a menu, which enables the user to select different options.
- Simple yet powerful interface.
- Cut - copy - paste instruction function.
- Facility to save current program.
- Facility to save current program with a different name.
- Insert function.
- Minimization of typing.
- Editing of program is possible by use of up / down buttons.
- Manual control of each motor
- Fine positioning of the base, rack, elbow and roll.
- Facility to rotate the motors clockwise and counter clockwise
- Automatic RESET function to bring the robot to the reset/home position on startup.
- Dynamic reset function to allow the user to bring the robot to the reset position at any instant.
- Auto Teach facility.

Software for octagon manoeuvring : OCTAGON has the following additional features, viz., REAL TIME MANUAL MODE CONTROL, REAL TIME RUN MODE CONTROL, PROGRAMMING INTERFACE, FACILITY TO LOOP OPERATIONS, ACCURATE POSITIONING, PARALLEL PORT INTERFACE & CONTROL, INTELLIGENT PATH DECISION MAKING. The screens used in the application are a default screen, which combines all the features of the PROGRAMMING MODE, MANUAL MODE, RUN MODE and the GRAPHICAL MODE [4].
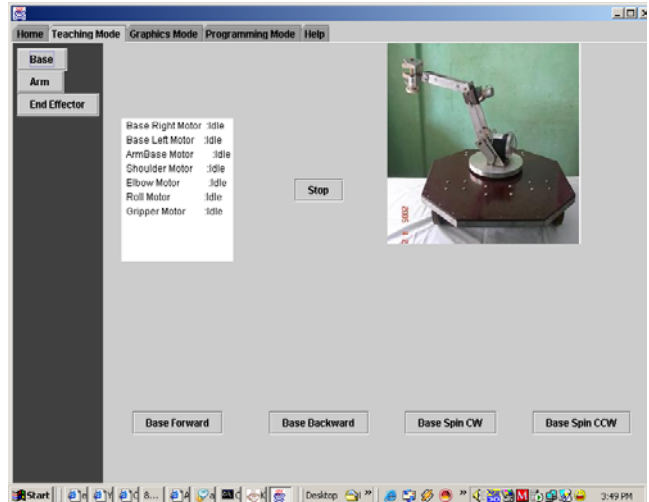

Fig. 2 General Graphical User Interface (GUI) for the robot control via JAVA environment
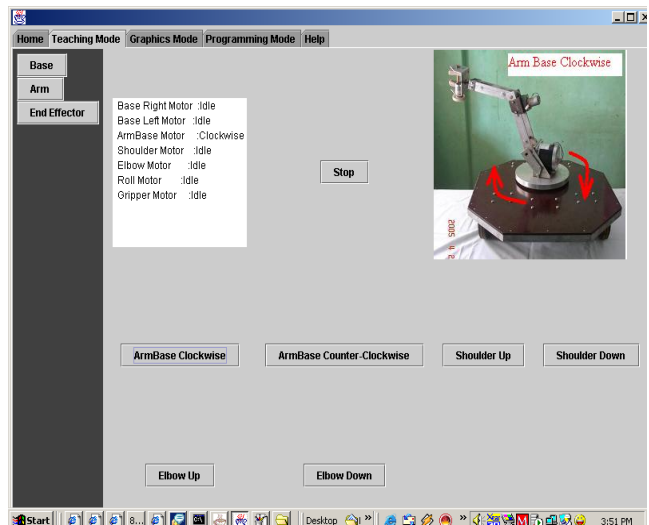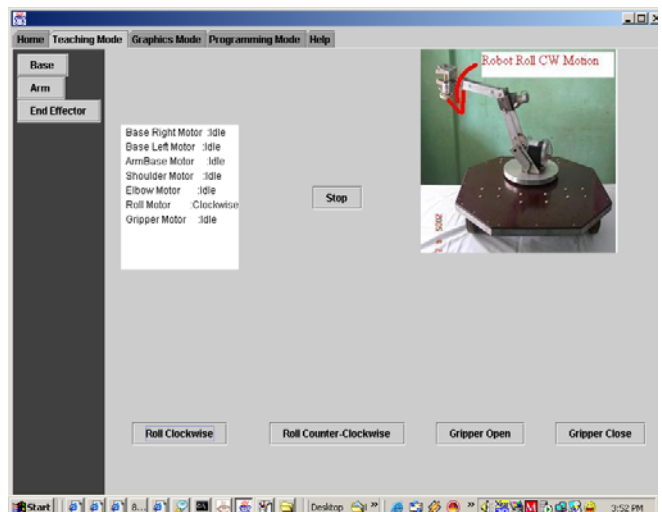

Fig. 3 GUI showing the arm base motor in operation

Fig. 4 GUI showing the roll motor in operation

When the java code is executed, the default GUI screen appears as shown in the Fig. 2 is displayed showing all the motors status. For activating a particular motion, we can navigate to its page by selected the left hand menu. The flow chart for the java code is shown below in Fig. 5.
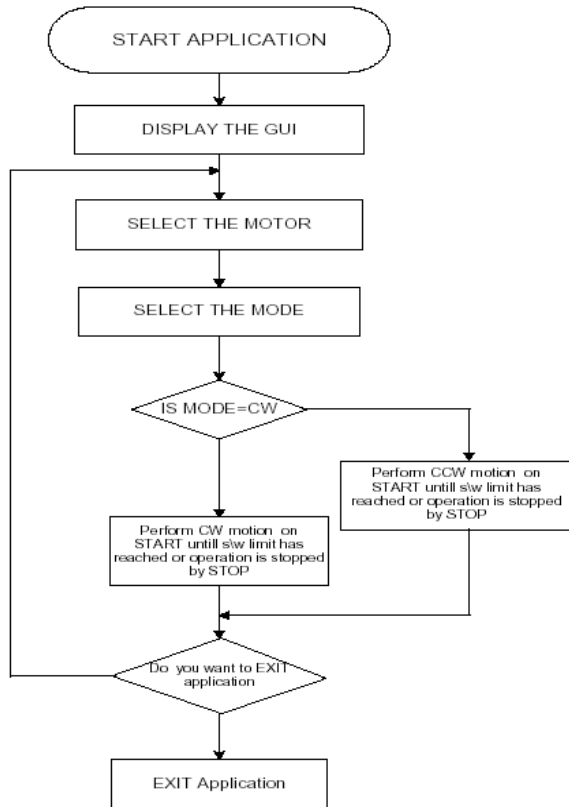

Fig. 5   Flow chart module of software module

3. Description about the GUI

The graphical user interface has check-boxes which are discussed in details as follows.

MOTOR SELECTION LIST BOX :

The screen features a listbox, where the user selects the desired motor. The user selects the motor by double-clicking on the motor name in the list box.

MOTOR SELECTED TEXT BOX :

The selected motor is then highlighted and displayed in the motor selected textbox.

DIRECTION OPTION BUTTONS :

The clockwise option button is enabled by default. The user can change the direction of rotation prior to pressing the start button. The user is required to click on the option button for selection of the direction.

START BUTTON :

The START button, when clicked, enables rotation of the desired motor. On clicking the start button, all the remaining components on the screen except the stop button are disabled as a precautionary measure [5].

STOP BUTTON :

The STOP button, when clicked, disables rotation of the desired motor by sending byte '0' to the port. On clicking the stop button, all the components disabled by the start button are re-enabled.

STATUS SIGNALS and MESSAGES :

The status signals and messages are an indication, to the user, of the motor selected, the direction of rotation and the byte sent to the port.

RESET BUTTON :

The screen features a reset button that is coded to respond to the click event. On clicking the button the software brings the robot to the HOME position or reset position. The reset is an important function, which is used in conjunction with the auto teach facility provided in the manual mode screen since it is mandatory that any teaching process be preceded by a reset, so that all calculations made by the robot software are with respect to the reset position of the robot [6].

Mobile (8 motions)	Arm (10 motions)
Forward	Arm base clockwise
Backward	Arm base anticlockwise
Forward Left	Arm shoulder up
Forward Right	Arm shoulder down
Backward Left	Arm elbow up
Backward Right	Arm elbow down
Spin Clockwise	Arm roll clockwise
Spin Anti-clockwise	Arm roll counter-clockwise
	Arm gripper open
	Arm gripper close

Table 1 : Table to show 18 motions of OCTAGON

AUTO TEACH FACILITY :

The manual mode provides the user with an auto teach facility, in which the program calculates the time delays and the associated angles, corresponding to the sequence of manually activated motors in an operation.

4. JAVA Programming Platform

JAVA is a simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language, which provides you with a complete set of tools to simplify rapid application development. We wanted to use a system that could be programmed easily without a lot of esoteric training and which leveraged today's standard practice. We could have used C, or object-oriented programming language C++. But, Java omits many rarely used, poorly understood, confusing features of C++ that in our experience bring more grief than benefit. These omitted features primarily consist of operator overloading multiple inheritance, and extensive automatic coercions [7].
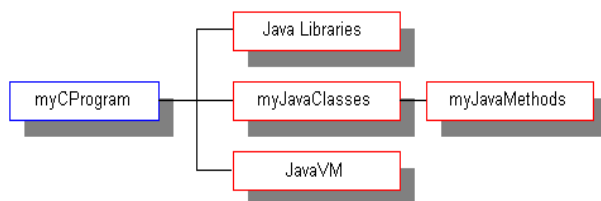

Fig. 6 Java Structure

Octagon uses JNI Technology to access parallel port using advanced JAVA. JAVA is Object-Oriented which a buzzword in industry is. Object-oriented design is very powerful because it facilitates the clean definition of interfaces and makes it possible to provide reusable "software ics". Simply stated, object-oriented design is a technique that focuses design on the data (=objects) and on the interfaces to it. To make an analogy with carpentry, an "object-oriented" carpenter would be mostly concerned with the chair he was building, and secondarily with the tools used to make it; a "non-object-oriented" carpenter would think primarily of his tools. Object-oriented design is also the mechanism for defining how modules "plug and play". Java has an extensive library of routines for coping easily with TCP/IP protocols like HTTP and FTP. This makes creating network connections much easier than in C or C++. Java applications can open and access objects across the net via URLs with the same ease that programmers are used to when accessing a local file system.

Java is intended for writing programs that must be reliable in a variety of ways. Java puts a lot of emphasis on early checking for possible problems, later dynamic (runtime) checking, and eliminating situations that are error prone. One of the advantages of a strongly typed language (like C++) is that it allows extensive compile-time checking so bugs can be found early. Unfortunately, C++ inherits a number of loopholes in compile-time checking from C, which is relatively lax (particularly method/procedure declarations). In Java, we require declarations and do not support C-style implicit declarations. The linker understands the type system and repeats many of the type checks done by the compiler to guard against version mismatch problems. The Java Native Interface (JNI) is the native programming interface for Java that is part of the JDK. By writing programs using the JNI, you ensure that your code is completely portable across all platforms.

5. Conclusions

A user friendly graphical user interface in JAVA programming language was developed and the octagon was controlled in various modes such as the manual mode, teach mode, programming mode, automatic mode. A number of pick and place operations were successfully performed by the developed robot by using the above mentioned 4 types of modes.

References

[1]. Robert, J.S. (1992), Fundamentals of Robotics : Analysis and Control, *PHI*, New Delhi.

[2]. Klafter, Thomas and Negin (1990), Robotic Engineering *PHI*, New Delhi.

[3]. Fu , Gonzalez and Lee (1995), Robotics : Control , Sensing , Vision and Intelligence, *McGraw Hill*, Singapore.

[4]. Ranky, P. G., C. Y. Ho (1998), Robot Modeling, Control & Applications", *IFS Publishers*, *Springer*, UK.

[5]. T.C.Manjunath (2005), Fundamentals of Robotics, *Nandu Publishers*, *4th Revised Edition*, Mumbai.

[6]. T.C.Manjunath (2005), Fast Track To Robotics, *Nandu Publishers*, *2nd Edition*, Mumbai.

[7]. Ranky, P. G., C. Y. Ho (1999), Robot Modeling, Control & Applications, *IFS Publishers*, *Springer*, UK.

[8]. Groover, Weiss, Nagel and Odrey (2000), Industrial Robotics, *McGraw Hill*, Singapore.