Advances in Robotics & Automation
ISSN: 2168-9695

Azad and Rashid, Adv Robot Autom 2013, 2:1
DOI: 10.4172/2168-9695.1000104

Research Article | Open Access

Design an Arm Robot through Prolog Programming Language

Aram Azad[1] and Tarik Rashid[2]*

[1]Computer Science and Information Technology, UTS: University of Technology, Sydney, Sydney, Australia
[2]Software Engineering Department, College of Engineering, Salahaddin University, Hawler, Kurdistan

Abstract

This paper aims at furnishing basic definitions for robotics in the span of man-made intelligence. Simple robotics and their faculties such as sensors and actuators are included in the definition. Moreover, intelligent and non-intelligent types of robotics are described in this research work. We generally focus on non-intelligent robotics (program-based robotics) and the setbacks designers can encounter .The methods and designs of an arm robot generally encompass three components: namely, the mechanical assembly of the arm robot, the electronic circuit, and software design employing Prolog software programming language. Our design concentrates generally on programming an arm robot to manipulate its movements. In our case study programming an arm robot we delineate probable resolutions to setbacks we encountered without going intensely into theory, and we demonstrate a stable design for resolving these setbacks via a flowchart. The design of the flowchart is next explained and subsequently coded.

Keywords: Robotics; Robot control methods; Intelligent robot; Non intelligent; Arm robot programming

Introduction

General history of robotics

The root of the word robot dates back to Karel Capek's play Rossum Universal Robots (RUR) in 1921. This actually originated from the Czech word for "corvee" [1-11]. Technologically speaking, robotics fall into two categories; these are namely: tele-manipulators, and the ability of numerical control of machines. The first types of Tele-manipulators were made of an arm and a gripper; this machine can be remotely controlled. The human gives instructions via his control device to control the arm and gripper's movements. These types of robotics are used to handle radioactive material.

However, the second type, Numeric control, is used to allow very precise control of machines in relation to a given coordinate system. This was first utilised at MIT in 1952. This type of robotics tipped to the first programming language for machines called Automatic Programmed Tools (APT) [1-11].

The amalgamation of the two types mentioned above resulted in the first programmable telemanipulators; moreover, these principles were first considered and fitted into industrial robots in 1961. Car construction plants are good examples of locations where these types of robotics remain useful. There was a desire to develop automated transportation and autonomous transport systems and production processes. Another type of robots that can move, called mobile robotics, are being built, and there is a particular type which is called insectoid robotics where the robots have many legs. These types are used for more autonomous purposes such working underwater.

In recent years wheel-driven robots have been commercially promoted and utilised for public service and purposes such as in hospitals and other places. In 1975 another type of robotics was designed called Humanoid robots. This happened when Wabot-I was offered in Japan. As matter of fact, the current Wabot-III already has some minor cognitive capabilities. In 1994, another type of humanoid robot, called "Cog", was developed in MIT-AI-Lab, and in 1999 Honda's humanoid robot came around and became widely known to the public. This type is controlled remotely by humans, and it can walk autonomously. In science fiction, robots are already human's best friends, but in reality we will only see robots for specific jobs as universal programmable machine slaves in the near future (which leads to interesting questions, see [1-11]).

History of arm robots

In 1954, the first patent for robotics was received by George Devol, and in 1956, F. Engelberger had the first company, and Devol's original patents were used for this establishment. It is worthy to mention that a Unimation robot was used for transferring different objects from a point A to a point B for less than a dozen feet. Hydraulic effectors were used and they were programming in joint coordinates. Cincinnati Milacron Inc. of Ohio was a competitor for some time in Unimation robots. However, this was not continued and in late 1970's several big Japanese conglomerates began to generate the same industrial robots. While in the US Unimation robots had been patented; nevertheless, this was not the case in Japan, who rejected being obliged to international patent laws; as a result, their design was copied [12].

The Stanford arm was created by Victor Scheinman in 1969 at Stanford University. This is an all-electric, six axis articulated robot to permit an arm solution. This gave a robot an accurate feature to follow arbitrary paths in space and widened the potential of robots to more progressive uses; e.g. assembly and arc welding. Afterward, the MIT AI Lab was created by Scheinman as a second arm, called the MIT arm. After that they sold his architecture design to Unimation, who later on, with help of General Motors, developed and then sold it as the programmable Universal Machine for Assembly (PUMA) [12].

***Corresponding author:** Tarik Rashid, Software Engineering Department, College of Engineering, Salahaddin University, Hawler, Kurdistan, Iraq, E-mail: tarikrashid4@gmail.com

Received December 29, 2012; **Accepted** June 21, 2013; **Published** June 24, 2013

Citation: Azad A, Rashid T (2013) Design an Arm Robot through Prolog Programming Language. Adv Robot Autom 2: 104. doi:10.4172/2168-9695.1000104

Citation: Azad A, Rashid T (2013) Design an Arm Robot through Prolog Programming Language. Adv Robot Autom 2: 104. doi:10.4172/2168-9695.1000104

FAMULUS is the first industrial robot which was designed by KUKA in 1973. This type of design had six electromechanically driven axes. In late 1970s the interest in industrial robotics swelled as many different companies had entered the field, including big ones such as General Electric and General Motors, who formed a joint venture with FANUC LTD of Japan, called FANUC Robotics. US start-ups included Automatix and a dept Technology, Inc. At the height of the robot boom in 1984, Unimarion was acquired by Westinghouse Electric Corporation for 107 million US dollars. Unimation was sold to Staubli Faverges SCA by Westinghouse in 1988. Articulated robots for general and clean room application were still made by Staubli in 2004, and the robotic division of Bosch in late 2004. Finally the myopic vision of the US industry was superseded by the financial resources and strong domestic market enjoyed by the Japanese manufactures. Only a few non-Japanese companies managed to survive in this market including Adapt Technology, Staubli-Unimation, the Swedish-Swiss Company and ABB IASEA [12].

Objectives

The main objectives of this paper are as follows:

I. Design and Assemble the Arm Robot: The main objective of any project is to design the arm robot that is able to carry out certain task. The revolute robotic arm is able to move similarly to a human arm. The arm is designed so it is able to rotate clockwise and counter clockwise (180 degrees), and able to pick up and place objects. The arm needs to be as light as possible in order to maximize burden. The material for the arm structure also needs to be strong and rigid. One possible material is aluminium as used in [12].

II. Design and Construct Controller Circuit Board: The second main objective is to design and engineer the controller circuit board that will be used to control the arm robot through connections to a personal computer (PC). The most important component that is used in the controller is the microcontroller. The circuit board will be connected to a serial port of a PC through a serial connector [12].

III. Design Robot Software and Interfacing: The third main objective is to design the robot interface application using Prolog programming language to control the controller circuit board to run the robot arm.

Artificial Intelligence and Robotic Definitions and Their Types

As we have stated above, robots can fall into the categories of manipulators, mobile robots, and humanoids. Often tasks are performed using physical agents called robots. Basically, robots have two faculties: sensors through which robots can receive perception from environment; and effectors through which robots are able to perform physical tasks [13-20].

Generally speaking, artificial intelligence (AI) is one of the important disciplines in computer science. AI has been considered as a theory and agents are hot topics in artificial intelligence. The most important part in the agent is the actor. This is recognised in software. It is clear that robots are created as hardware. The construction of artificial intelligence and robotics is that a software agent controls the robot that has sensors through which data can be read, and then the robot decides what to do, then guide the actuator to perform an action in the environment [13-20].

Definitions of a robot will decide the differences between two types of intelligent and non-intelligent robots. A simple example of a robot is the thermostat on a heater or in our living room. Of course, this definition might not satisfy the definition of intelligence in human terms. Intelligence, then, may be defined as an arbitrary quality considered as such by the human operator. The term robot is used in two ways: a machine that is directed by a human operator by remote control and a machine that makes limited decisions based on a computer program. Factory welding machines are programmed robots that can sense the position of the parts. Automatic drone aircraft are given goals and can fly themselves. The Russian space craft's are autonomous cars that use cameras and sensors and on-board computers to drive themselves and stay on the road. We believe robot machines cannot think as we humans do, although we keep on hoping to arrive at this point. There are two types of robots based on intelligence, these are:

Non-intelligent robots

One important type of robot is called a non-intelligent robot. These are robots that do not have sensors to receive information. Instead, the robot is only able to follow a fixed set of instructions, no matter what is happening around it. A robot like this can never become intelligent (Figure 1).

Intelligent robots

A robot is controlled by sets of instructions called a program, which is built into the controlling device. Intelligent robots must have sensors to collect information from the environment [12-15]. This information will help the robot to become intelligent. This information will be sent to the controlling program [21-29], where it is tested to decide what the robot should do next. Generally speaking, sensors can be in the form of a camera, a pressure pad, and a microphone that will allow the robot to view, touch, and hear. These sensors are tailored or built in the robot almost exactly like a human being has eyes to see, hands and fingers to hold objects or touch, and ears to listen (Figure 2).

Methods

The methods and design of robots mainly consist of three parts: namely, the mechanical construction of an arm robot, the electronic circuit, and the programming design using Prolog programming language. In this paper, our design is mainly focused on the programming of robots.

The Mechanical construction of an arm robot

This is the main part of the arm robot which needs three different structures; these are shoulder, elbow and wrist. The challenge in this particular part is how to affix the motor to the piece or joint to achieve the preferred turning degree level. Another challenge, notably, the

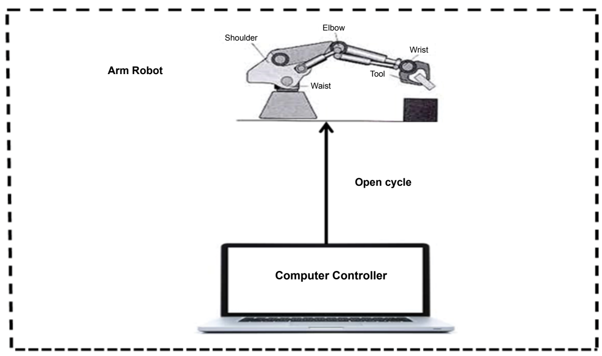

Figure 1: Shows a non-Intelligent Robot.

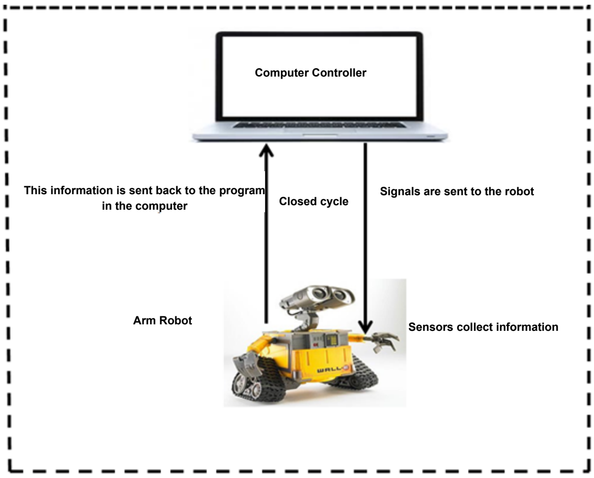

Figure 2: Shows an intelligent robot.

intended degree of freedom, can be achieved by the perfect combination of the different arm components. The mechanical design is achieved using the software Autodesk Inventor; this software is used to build and design the arm robot. Dimensions for mechanical design are a crucial issue and it is regarded as the core issue in the arm robot. They have to be very accurate, and there is no room for minor error, since a minor error in the dimensions will cause a major problem in the assembly and function of the arm structure.

Electronic circuit design

This is equally as important as the mechanical part; this part is dedicated to provide control over the movement of the arm robot. A microcontroller is used to provide this purpose.

Design the programming part using prolog language

Design a software program that enables the movement of arm robot; there must also be an interface with the software to control the arm robot.

The design of the programming part using prolog language is divided into two parts:

Plan and design: There is an arm robot with four other objects in this case: Square, Semi-Square, Circle, and Triangle. The arm is able to hold the objects and put them on each other and on the floor (Figure 3); the user of the program can control the hand robot by using some commands on a screen or menu. Additionally, the program will end whenever the user pushes on the exits command. There should be a user interface which shows the commands to the user, to wait for a user to say what to do. These important commands are:

1) Hold object
2) Put object
3) Free hand
4) Display
5) Exit

We have restricted a user with some rules that the arm robot must obey when performing tasks. These rules are clearly defined below:-

1) The arm robot can put a semi-square on a square
2) The arm robot can put a triangle on a square
3) The arm robot can put a circle on a semi-square
4) The arm robot should be free before taking hold of an object

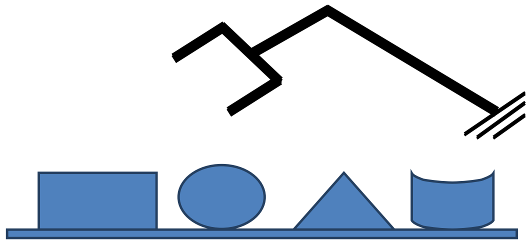

Figure 3: Shows an arm robot with the four different objects.

5) The arm robot should be full before commanded to free arm
6) The object should be free in order to hold it
7) A blocked object cannot be held

When the user chooses the hold command, the program will ask the user about the name of the object to hold; before that, the program should check if the arm is free, if the arm is free, then the arm continues; otherwise, it writes a message to tell the user that the arm is full, and returns back to the welcome screen. Then it should check if the object is free, in other words, is not blocked or held.

When the user chooses the put command, the program should check if the arm is full or not. If the arm is full, then it continues, or else it returns back to the welcome screen; then the program should ask about the name of the target object to put it on, then the program checks if the target object is blocked or not. If the object is free then it continues, or else shows an error message to the user of "enter another target." When the user enters the free command, the program should check the arm if it is already free or not; if it's free, then the error message will be displayed to the user that says "The arm already is free!" and then return to welcome screen. If the arm is full, then free the arm by placing object that is already held, and then return back to the welcome screen.

When the user enters the Display command, the program should display the status of each object and also the arm status; it should also show which objects are blocked, and it should say which objects are already on them.

In an ideal case all the objects and arm status statements are "free".

You need a header statement here that tells your reader the following phrases are the status statements

Arm (free/full), Object (free/hold/block/another Object)

Both Arm and Objects had status, as declared above; you can see arm has two statuses, and each object has 4 types of status; the fourth one "another Object" means it is under that object, if I put triangle over square the square status will become "triangle". The flowchart is designed for the arm robot with all possibilities and functionalities indicated, (Figure 4).

Coding and implementation: After completing the design and the flowchart of the arm robot, the flowchart was implemented and coded in the prolog programming language [30-35]. Below is the full programming code for a complete arm robot system

Domains
x=symbol.
y=integer.
s=string.
Predicates

Citation: Azad A, Rashid T (2013) Design an Arm Robot through Prolog Programming Language. Adv Robot Autom 2: 104. doi:10.4172/2168-9695.1000104

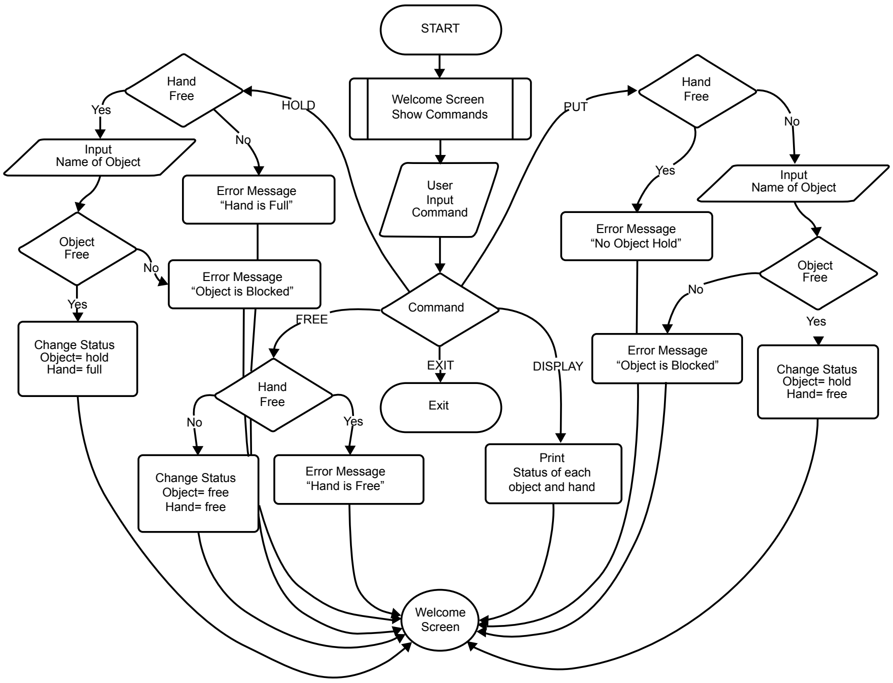

Figure 4: Shows a complete flowchart design for an arm robot system.

```
% nondetermrobothold(s,s,s,s,s,s).
% nondetermrobotput(x,x,x,x,x,s,s).
nondeterm check(s,s).
nondeterm robot(s,s,s,s,s,s,s,y).
% nondeterm call(y).
% nondeterm valid(x).
Clauses
check("triangle","square").
check("semisquare","square").
check("circle","semisquare").

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,0):-
nl,
write("Press 1 to hold"),nl,
write("Press 2 to put"),nl,
write("Press 3 to free"),nl,
write("Press 4 to exit"),nl,
write("Press 7 to print"),nl,
write("Enter Number:"),
readint(N),
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,N).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,N):-
N<1,
write("Enter valid input 1,2,3,4,7: "),
readint(M),
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,M).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,N):-
N>7,
write("Enter valid input 1,2,3,4,7: "),
readint(M),
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,M).

robot(_,_,_,_,_,_,_,4):-
write("Good Bye"),nl.

robot(Square,SemiSquare,Triangle,Circle,Hand,_,Target,1):-
nl,
Hand="free",
write("Enter shape to hold 'sqaure,triangle,semisquare,circle': "),
readln(Shape),
robot(Square,SemiSquare,Triangle,Circle,"full",Shape,Target,6).

robot(Square,Semisquare,Triangle,Circle,"full",Shape,Target,6):-
Square="free",
Shape="square",
robot("hold",Semisquare,Triangle,Circle,"full",Shape,Target,0).

robot(Square,SemiSquare,Triangle,Circle,"full",Shape,Target,6):-
Semisquare="free",
Shape="semisquare",
robot(Square,"hold",Triangle,Circle,"full",Shape,Target,0).

robot(Square,SemiSquare,Triangle,Circle,"full",Shape,Target,6):-
Triangle="free",
Shape="triangle",
robot(Square,SemiSquare,"hold",Circle,"full",Shape,Target,0).

robot(Square,SemiSquare,Triangle,Circle,"full",Shape,Target,6):-
```

```
Circle="free",
Shape="circle",
robot(Square,SemiSquare,Triangle,"hold","full",Shape,Target,0).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,1):-
Hand="full",
nl,write("Error The hand is not free"),nl,
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,0).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,7):-
nl,
write("Square: "),write(Square),nl,
write("SemiSquare: "),write(SemiSquare),nl,
write("Triangle: "),write(Triangle),nl,
write("Circle: "),write(Circle),nl,
write("Hand: "),write(Hand),nl,
write("Source: "),write(Source),nl,
write("Target: "),write(Target),nl,
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,0).

robot("hold",SemiSquare,Triangle,Circle,_,_,_,3):-
robot("free",SemiSquare,Triangle,Circle,"free","free","free",0).

robot(Square,"hold",Triangle,Circle,_,_,_,3):-
robot(Square,"free",Triangle,Circle,"free","free","free",0).

robot(Square,SemiSquare,"hold",Circle,_,_,_,3):-
robot(Square,SemiSquare,"free",Circle,"free","free","free",0).

robot(Square,SemiSquare,Triangle,"hold",_,_,_,3):-
robot(Square,SemiSquare,Triangle,"free","free","free","free",0).

robot(Square,SemiSquare,Triangle,Circle,"free",Source,Target,3):-
nl,write("The hand already is free "),nl,
robot(Square,SemiSquare,Triangle,Circle,"free",Source,Target,0).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,_,2):-
Hand="full",
nl,write("Enter shape to put on 'sqaure,triangle,semisquare,circle' :"),
readln(Shape),
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Shape,5).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,2):-
Hand <> "full",
nl,write("You should hold a shape before you put"),nl,
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,0).

robot(Square,Semisquare,Triangle,Circle,_,Source,Target,5):-
check(Source,Target),
Semisquare="free",
Circle="hold",
Source="circle",
Target="semisquare",
robot(Square,"circle",Triangle,"free","free","free","free",0).

robot(Square,SemiSquare,Triangle,Circle,_,Source,Target,5):-
check(Source,Target),
Square="free",
Triangle="hold",
Source="triangle",
Target="square",
robot("triangle",SemiSquare,"free",Circle,"free","free","free",0).

robot(Square,SemiSquare,Triangle,Circle,_,Source,Target,5):-
check(Source,Target),
Square="free",
Semisquare="hold",
Source="semisquare",
Target="square",
robot("semisquare","free",Triangle,Circle,"free","free","free",0).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,5):-
nl,write("The target already blocked"),nl,
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,0).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,5):-
Target="circle",
nl,write("Error its not possible"),
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,2).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,5):-
Target="semisquare",
nl,write("Error its not possible"),
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,2).

robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,5):-
Target="triangle",
nl,write("Error its not possible"),
robot(Square,SemiSquare,Triangle,Circle,Hand,Source,Target,2).

Goal
robot("free","free","free","free","free","free","free",0).
```

Conclusion

This paper provides a basic definition and background about robotic systems in the area of artificial intelligence. The paper refers to robotic systems with their basic facilities such as sensors and actuators. Two types of robotics are explained: these are intelligent and non-intelligent robotic systems. In this paper we have considered the non-intelligent robotic in which programming was our base to manipulate the system (program based robotics). A case study of designing an arm robot proposal is presented in which we describe possible solutions and functionalities to those problems without going deeply into theory anda firm design for solving the problem is suggested via a flowchart.

The design of flowchart is then implemented and coded. The language of the program used to design the arm robot is prolog.

References

1. Niemueller T, Widyadharma S (2003) Artificial Intelligence-An Introduction to Robotics.
2. Carlson D (1998) Synchro Drive Robot Platform.
3. Tong A (1995) Star Trek Episode: The Measure of a Man.
4. Singh S, Kumar S, Singh V, Sain C (2012) Robotic Analysis And Programming. International Journal of Advanced Scientific Research and Technology.
5. Wolf JC, Hall P, robinson P, Culverhouse P (2008) Bioloid Based Humannoid Soccer Robot Design.
6. Floreano D, Godjevac J, Martinoli A, Mondada F, Nicoud JD (2005) Design, Control And Application of Autonomous Mobile Robots. Advances in Intelligent Autonomous Systems 18: 159-186.
7. Udengaard M, Legnemma K (2007) Design of an Omnidirectional Mobile Robot for rough Terrain. IEEE International Conference on Robotics & Automation.
8. Salih JFM, Rizon M, Yaacob S, Adom AM, Mamat MR (2006) Designing Omni-Directional Mobile Robot with Mccanum Wheel. American Journal of Applied Science 3: 1831-1835.
9. Doroftei L, Grosu V, Spinu V (2007) Omnidirectional Mobile Robot Design And Implementation. Bioinspiration and Robotics: Walking and Climbing Robots.
10. Smart WD, Kaelbling LP (2008) Effective Reinforcement Learning For Mobile Robots. IEEE International Conference on Robotics & Automation 4: 3404-3410.
11. Murphy D, Challacombe B, Khan MS, Dasgupta D (2006) Robotic Technology In Urology. Postgrad Med J 82: 743-747.
12. Omar MNB (2007) Pick and Place Robotic Arm Controlled By Computer. Robotic and Automation.
13. Leang K (1999) Minibot Sonar Sensor How to.
14. Dellaert F, Fox D, Burgard W, Thrun S (1999) Monte Carlo Localization For Mobile Robots. IEEE International Conference on Robotics and Automation 2: 1322-1328.
15. De Giacomo G, Lesperance Y, Levesque H, Reiter R (2001) Indigolog Overview.
16. Grabowski B (2003) Small Robot Sensors.
17. Görz G, Rollinger CR, Schneeberger J (2000) Handbuch Der Künstlichen Intelligenz, 3. Auflage. Oldenbourg Verlag München Wien.
18. Lego (2003) Lego Mindstorms Tutorial on Correcting Course.
19. Trimble Navigation Limited (2003) Trimble-What Is Gps?
20. Helsinki University Of Technology (2001) Moving Eye-Virtual Laboratory Exercise On Telepresence, Augmented Reality, and Ball-Shaped Robotics.
21. Röbke-Doerr P (2003) Navigation Mit Satelliten. C't Magazine Fur Computer Technik 1:150-151.
22. Russell S, Norvig P (1995) Artificial Intelligence. A Modern Approach. Prentice Hall.
23. Russell S, Norvig P (2003) Artificial Intelligence. A Modern Approach. (2nd Edn), Prentice Hall.
24. Robocup Team (2003) Robocup.
25. RWTH Aachen (2003) Robocup Team. Allemaniacs.
26. Thrun S (2003) Robotic Mapping: A Survey. Robotics.
27. Weisstein EW (1999) Voronoi Diagram. Math World.
28. Welch G, Bishop G (2003) An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill.
29. Nelly D, Cees VB (2007) Introduction of Robotics in Science Lessons. Comlab Conference.
30. Delson JN, West H (1994) Robot programming by human demonstration: The use of human inconsistency in improving 3D robot trajectories. Massachusetts Institute of Technology, Dept. of Mechanical Engineering, MA.
31. Edsinger A (2007) Robot Manipulation in Human Environments. Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, MA.
32. Hoang H (2005) Automated construction technologies: analyses and future development strategies. Master thesis, Massachusetts Institute of Technology, Dept. of Architecture, MA.
33. Flynn AM, Brooks RA (1987) MIT Mobile Robots-What's Next? (Working Paper 302). Massachusetts Institute of Technology, Artificial Intelligence Laboratory 1: 611-617.
34. Yoon Y (2006) Modular robots for making and climbing 3-D trusses. Massachusetts Institute of Technology, Dept. of Mechanical Engineering, MA.
35. Schatz David (1983) The strategic evolution of the robotics industry. Massachusetts Institute of Technology. Sloan School of Management.