INTERNATIONAL JOURNAL OF ADVANCED INNOVATIVE TECHNOLOGY IN ENGINEERING

Published by Global Advanced Research Publication House Journal

Home page: www.ijaite.co.in

Robotic Devices with Artificial Intelligence

[1]Prof. Uvaraj V. Mane, [2]Prof. Satchidanad S. Choure

[1,2]Department of Mechanical Engineering, Bharati Vidyapeeth IOT Karghar, Navi Mumbai, Maharashtra, India

[1]uvarajmane11858@gmail.com, [2]sachinschoure@gmail.com

Article History

Received on: 01 April 2022

Revised on: 15 April 2022

Accepted on: 16 May 2022

Keywords: Robot, Degree of freedom (DOF), Monte Carlo Localization (MCL), Artificial Intelligence (AI), Voronoi diagrams

e-ISSN: 2455-6491

Production and hosted by
www.garph.org

©2021|All right reserved.

ABSTRACT

Today, robotics covered a broad sector of economic activities from the automotive and electronics industries. Up to now, however, robot automation technologies have mainly been deployed in large-volume manufacturing, which results in costly and complex robot systems, which often cannot be used in small and medium-sized manufacturing. New branches of robot automation that emerge nowadays such as food, logistics, recycling, etc. require new designs of robot systems. This paper gives a short introduction to the basics of robotics in the field of Artificial Intelligence. It also gives an overview of the robotic history, some basic problems encountered in modern robotics, and possible solutions to those problems. The problems introduced are perception, basic pose description, transition and sensor models, localization as a special case of perception (Monte Carlo Localization), representation of the environment (workspace and configuration space), path planning (cell decomposition, skeletonization, Voronoi diagrams), movement of robots, and some real-life examples.

1. INTRODUCTION

The term "Robot" can be traced back to Karel Capek's play "R.U.R. Rossum's universal robots" (in 1921) which comes from the Czech word for "corvee". The research and development efforts in robotics will strongly contribute to the creation of new opportunities towards European employment and growth.
Robots are physical agents that perform tasks by manipulating the physical world. They are equipped with sensors to perceive their environment and effectors to assert physical forces on it.
Robotics is based on two enabling technologies:
i) The Telemanipulators are remotely controlled machines that consist of an arm and a gripper. The movements of the arm and gripper follow the instructions the human gives through his control device.
ii) Numeric control allows controlling machines in relation to a given coordinate system.
The combination of both of these techniques leads to the first programmable telemanipulator. The first industrial robot using these principles was

installed in 1961. The development of mobile robots was driven by the desire to automate transportation in production processes and autonomous transport systems. The former driverless transport systems were used on factory floors to move objects to different points in the production process in the late seventies. New forms of mobile robots have been constructed lately like insectoid robots with many legs modeled or autonomous robots for underwater usage. For a few years, wheel-driven robots are commercially marketed and used for services like "Get and Bring" (in hospitals).
Humanoid robots are being developed since 1975 when Wabot-I was presented in Japan. The current Wabot-III already has some minor cognitive capabilities. Another humanoid robot is "Cog", developed in the MIT-AI-Lab in 1994. Honda's humanoid robot became well known to the public when presented back in 1999. Although it is remote-controlled by humans, it can walk autonomously on the floor and stairs. 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 [17].

2. Robotics And Artificial Intelligence (AI)

Artificial intelligence is a simulation of human intelligence in machines. It is realized in software. Robots are manufactured as hardware. The control of the robot is a software agent that reads data from the sensors which decides what to do the next and then directs the effectors to act in the physical world.
Practical robotics had its origins inside factories manufacturing products on assembly lines, where speed, precision, and reliability were supreme. Thus, both human tedium and imprecision were done away with improving the uniformity of a quality product that could come off the assembly line around the clock. Precision machine tools are considered the inspiration for such a development. However, before robots could be programmed using a computer, it was not possible to easily change any detail of their repetitive function. With the flexibility of programming and the training supplied by a skilled operator, robots could carry out very complex, human-like, and repetitive tasks; however, the task could be changed at short notice by using a different program. As robots migrated underwater, in the air, and on the ground, where many future applications could be imagined. A complementary range of sensors and considerable artificial intelligence would be needed to achieve autonomy. This striving for autonomy in complex, unstructured, and unpredictable environments, sometimes coexisted by humans and has given rise to the field of 'intelligent robotics' where perception, reasoning, and actuation are highly coupled to achieve useful tasks with little human guidance shown in Figure 1.

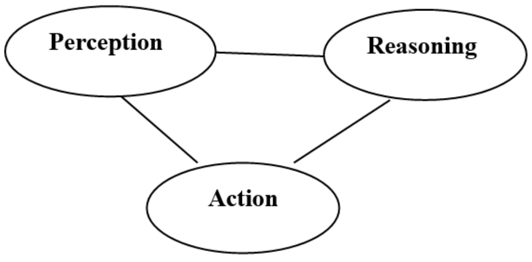


Figure 1: Intelligence Robotics

3. Robot Hardware

A. Sensors

Sensors are the perceptual interface between robots and their environment. We have passive sensors like cameras, which capture signals that are generated by other sources in the environment. On the other hand, we have active sensors (for example sonar, radar, and laser) which emit energy into the environment. This energy is reflected by objects in the environment. These reflections can then be used to gather the information needed. Generally, active sensors provide more information than passive sensors. But they also consume more power. This can lead to a problem for mobile robots which need to take their energy with them in batteries. We have three types of sensors that record distances to objects or generate an entire image of the environment or measure a property of the robot itself.
Many mobile robots use range finders, which measure the distance to nearby objects. A common type is the sonar sensor [6]. Alternatives to sonar include radar and laser. Some range sensors measure very short or very long distances. Close-range sensors are often tactile sensors such as whiskers, bump panels, and touch-sensitive skin. The other extreme is long-range sensors like the Global Positioning System (GPS) [8, 10].

The second important classes of sensors are imaging sensors. These are cameras that provide images of the environment that can then be analyzed using computer vision and image recognition techniques. The third important class is proprioceptive sensors. These inform the robot of its own state. To measure the exact configuration of a robotic, joint motors are often equipped with shaft decoders that count the revolution of motors in small increments. Another way of measuring the state of the robot is to use force and torque sensors. These are especially needed when the robot handles fragile objects or objects whose exact shape and location are unknown.

B. Effectors

The robots manipulate the environment, and move and change the shape of their bodies through the effectors. To understand the ability of a robot to interact with the physical world, we will use the abstract concept of a degree of freedom (DOF). We count one degree of freedom for each independent direction in which a robot or one of its effectors can move. As an example, let's contemplate a rigid robot like an autonomous underwater vehicle (AUV). It has six degrees of freedom, three for its x, y, z location in space and three for its angular orientation. These DOFs define the kinematic state of the robot. This can be extended with another dimension that gives the rate of change of each kinematic dimension. This is called a dynamic state. Robots with no rigid bodies may have additional DOFs. For example, a human wrist has three degrees of freedom; it can move up and down, side to side, and can also rotate. Robot joints have 1, 2, or 3 degrees of freedom each.

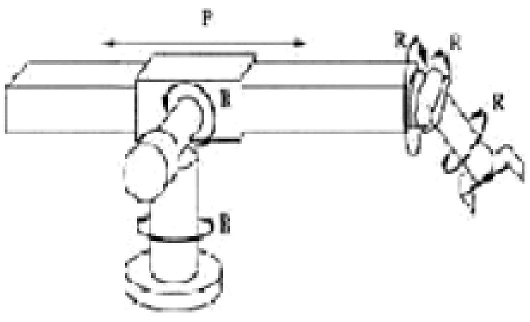


Figure 2: The Stanford Manipulator

The manipulator shown in Figure 2 has exactly six degrees of freedom, created by five revolute joints (R) and one prismatic joint (P). Revolute joints generate rotational motion while the prismatic joints generate sliding motion. Manipulators are easier to control than robots having only the minimum number of DOFs. The number of degrees of freedom does not need to have corresponding actuated elements. For example, a car can move forward or backward, and it can turn, giving it two DOFs. A robot is nonholonomic if it has more effective DOFs than controllable DOFs and holonomic if the two numbers are the same. Holonomic robots are easier to control than nonholonomic. But holonomic robots are mechanically more complex. Most manipulators and robot arms are holonomic and most mobile robots are nonholonomic.

C. Movement

For mobile robots, a special group of effectors is the mechanisms. The robot uses for locomotion, including wheels, tracks, and legs. The differential drive consists of two independently actuated wheels; one on each side. If both wheels move at the same velocity, the robot moves in a straight line. If they move in opposite directions, the robot turns on the spot. An alternative is the synchro drive3, in which each wheel can move and turn around its own axis.

D. Power Sources

Robots need a power source to drive their effectors. The most popular mechanism for both manipulator actuation and locomotion is the electric motor. Other possible ways are pneumatic actuation using compressed gas and hydraulic actuation using pressurized fluids.

E. Bits and Pieces

Most robots have some kind of digital communication like wireless networks. Especially today, those modules get cheaper. They can be used for communication between robots or for some kind of backlink to the robot's home station. Finally, we need a body frame to hang all the bits and pieces.

4. Robotic Perceptions

A robot receives raw sensor data from its sensors. It has to map those measurements into an internal representation to formalize this data. This process is called robotic perception. This is a difficult process since in general the sensors are noisy and the environment is partially observable, unpredictable, and often dynamic.

A. *Localization*

A very generic perception task is localization. It is the problem of determining where things are. Localization is one of the most penetrating perception problems in robotics. There are three increasingly difficult types of localization problems-

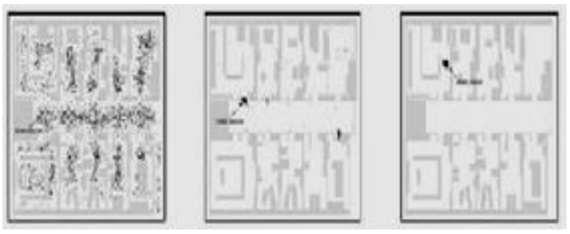
i) Tracking: If the initial state of the object to be localized is known we can just track this object.

ii) Global localization: In this case, the initial location of the object is unknown. We first have to find the object. After that we found it, and this becomes a tracking problem.

iii) Kidnapping: This is the most difficult task. We take the object the robot is looking for and place it somewhere else. Kidnapping is often used to test the robustness of a localization technique under extreme conditions.

B. *Monte Carlo Localization (MCL)*

MCL is essentially a particle filter algorithm. The requirements area map of the environment shows the regions, the robot can move to and an appropriate motion model and sensor model. We assume that the robot uses range scanners for localization. The algorithm takes the given map and creates a population of N samples by a given probabilistic distribution. Then we start a continuous update cycle. The localization starts at time t = 0. Then the update cycle is repeated for each time step.
Each sample is propagated forward by sampling the next state value Xt+1 given the current value Xt for the sample and using the transition model given. Each sample is weighted by the new evidence, P(et+1| xt+1). The population is resampled to generate a new population of N samples. Each new sample is selected from the current population. The probability that a particular sample is selected is proportional to its weight. The new samples are unweighted.



(a)Initialization (b)Unsure due to symmetry (c) Localized

Figure 3: An example of a Monte Carlo Localization

So as the robot gathers more knowledge about the environment by analyzing the range scanner data it resamples the population and the particles concentrate at one or more points. That is the location of the robot in the field. There are some problems we can encounter. For example, if we have a completely symmetric map like in Figure 4, a robot can walk around those rooms and has always more cluster points that could be the location of the robot. Now we want to study the example shown in Figure 3. First, the robot initializes the samples. Since it does not know anything, the particles look uniformly distributed. The robot has created the population of N samples from a given probability distribution.

• Robot	

Figure 4: Symmetric map

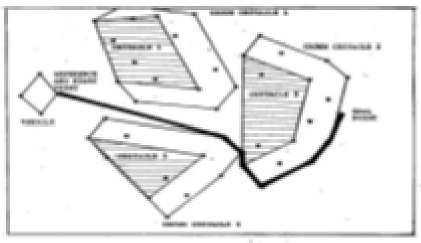
The robot starts to move in the northeast direction and gets into the door. This means that the robot first pipes all of its particles through its transition model. So, all particles move about the same way on the map as the robot moves. Then it weighs all particles. Some particles may have moved into a wall, so they will get a very small weight. Now, the robot again does a range scan. It detects that it is in a door with a slim wall on one side and a bulky one on the other. This configuration appears twice on the map, so the particles in these positions get very high weights. Since the probability that a specific sample is taken into the new population is proportional to its weight almost all new samples are from those two areas shown in Figure 3(b). But there is still uncertainty about the location. The robot yet has to move on and gather more data to be sure about its location. The robot moves further and creates a new population. Now, it has enough data so that only the particles moved in the west of the map get high. The particles in the east of the map get very low weights. So, now all particles are in one cluster point. The robot has localized its position.

C. Multi-Object Localization – Mapping

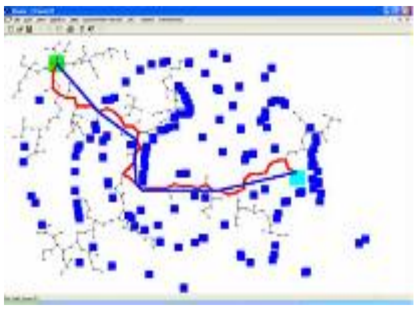
We only discussed the localization of a single object, but often one seeks to localize many objects. Now, we will give a short introduction to how robots can do the same. This problem is often referred to as simultaneous localization and mapping (SLAM). The robot does not only construct a map, but it must do this without knowing where it is. A problem is that the robot may not know in advance how large the map is going to be. The most widely used method for the SLAM problem is EKF. It is usually combined with landmark sensing models and requires that the landmarks are distinguishable. The robot now starts to move and discovers more and more landmarks. The uncertainty about the location of the landmarks and themselves increases with time. When the robot finds one landmark, he already discovered again the uncertainty of its position and landmarks [12] and [16].

5. PATH PLANNING

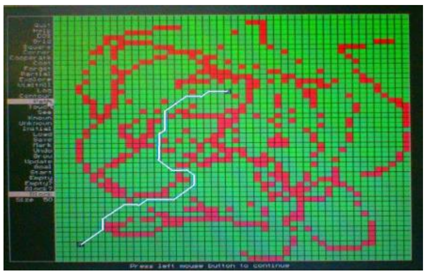
Once the localization and mapping aspects are resolved, the dual requirements of path planning and obstacle avoidance for a mobile robot to navigate from a starting position to a nominated goal point via an efficient collision-free path. There are a large number of path planners described in the literature. A representable set is the A* methodology, Rapidly Exploring Random Trees (RRT) and Distance Transforms (DT). The distance transform has the advantage of accommodating initially unknown environments, time-varying environments, variations in terrain navigability, and explorations of unknown spaces, covering all accessible territory and planning overt paths. Some examples are shown in Figure 5.



(a) A* Path Planning



(b) RRT Path Planning



(c) DT Path Planning

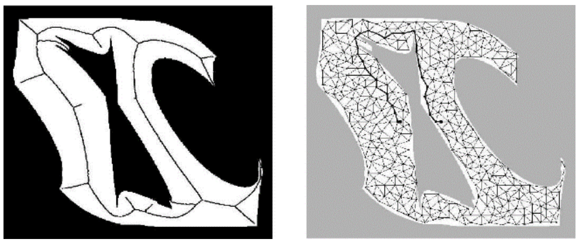
Figure 5: Path planning examples

A. Cell Decomposition

One approach toward planning a path in configuration space is to simply map the free space onto a discrete raster which reduces the path planning problem within each of the cells. The simplest cell decomposition matches the free space onto a regularly spaced grid but more complex methods are possible. Cell decomposition has the advantage of easy implementation.

B. Skeletonization Methods

The second major path planning algorithm to reduce the complexity of the free space to a one-dimensional representation is Voronoi Diagrams. The graph shown in Figure 6 is a Voronoi graph of the free space [18]. A robot using this method would first change its configuration to a point on the skeleton, which is due to the nature of a Voronoi diagram with a straight-line movement, then follow the skeleton to the point closest to its destination and make another straight line move to the final position. This method again reduces the original path planning problem to a discrete graph.



(a)Voronoi diagram (b)Probabilistic roadmap

Figure 6: Skeletonization methods

6. Moving

In general, a robot cannot simply set its effectors to a certain position but needs to exert forces to put them to the desired new position.

For generating the better motion plans of the robot, dynamic state models would be used. Such models are typically expressed through differential equations. A common technique is to use a separate controller to keep the robot on a path referred to as a reference controller. The most basic controller would be a proportional Controller (P controller) which applies a force proportional to the deviation amount to try to compensate for it. The control at generated by a P controller,

at = KP(y(t)- xt)

Where, xt is the state of the robot at time t and KP is the gain factor which determines, how strong the controller tries to correct the discrepancy between y(t) (the desired state) and xt (the actual state). Lower KPvalues only slow down the oscillating behavior but do not solve the issue.

Figure 7: Types of controllers

The simplest strictly stable controller is the proportional derivative controller (PD controller) which is described by the following equation:

$$a_t = K_P(y(t) - x_t) + K_D \partial(y(t) - x_t)/\partial t$$

Here, the gain factor K_D counteracts the proportional term reducing the overall response. PD controllers may fail to regulate an error down to zero. The solution to this problem lies in adding the integrated error overtime to the equation:

$$a_t = K_P(y(t) - x_t) + K_I\int(y(t) - x_t)dt + K_D \partial(y(t) - x_t)/\partial t$$

Again, K_I is the components gain factor, the long-lasting deviation from the reference path resulting in a growth of 'which makes the controller react stronger, thus correcting the path. This kind of controller is called a PID controller. They are widely used in industry for a variety of control problems.

CONCLUSIONS

The overall process for the development of intelligent robotics is very positive but more evolutionary with a steady penetration into the industrial and domestic worlds at affordable prices on the near horizon. It would be expected that, as the price per unit reduces through mass production as it has for automobiles and personal computers, there will become a time when simple robotic devices for use in the home will become standard peripherals (like printers), and many consumer items from washing machines and lawnmowers, to automobiles and entertainments systems, will have intelligent robotics capabilities built-in. The case when technologies once regarded as mind-blowing become commonplace and merely a matter of affordability and compliance with legal systems. The substitution of sensor-based intelligence for speed and precision to accommodate uncertainty and unstructuredness will remain the main driving force, being the essential paradigm shift.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

FUNDING SUPPORT

The author declares that they have no funding support for this study.

REFERENCES

[1] Doug Carlson. Synchro drive robot platform.http://www.visi.com/~dc/synchro/index.htm, 1998.

[2] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In IEEE International Conference on Robotics and Automation (ICRA99), May 1999.

[3] G. De Giacomo, Y. Lesperance, H. Levesque, and R. Reiter. Indigolog overview.http://www.cs.yorku.ca/~alexei/igoaa/indigolog.htm, 2001.

[4] Bob Grabowski. Small Robot Sensors. http://www.contrib.andrew.cmu.edu/~rjg/websensors/robot_sensors2.html, 2003.

[5] G. Görz, C.-R. Rollinger, and J. Schneeberger (Hrsg.). Handbuch der künstlichen Intelligenz, 3. Auflage. Oldenbourg Verlag München Wien, 2000.

[6] Kam Leang Minibot Sonar Sensor,http://www.atsemi.com/article/Howto.htm, 1999.

[7] Lego mindstorms tutorial on correcting course.http://mindstorms.lego.com/eng/community/tutorials/tutorial.asp?tutorialid=project6, 2003.

[8] Trimble Navigation Limited. What is GPS?http://www.trimble.com/gps/what.html, 2003.

[9] Helsinki University of Technology. Moving Eye - Virtual Laboratory Exercise on Telepresence, Augmented Reality, and Ball-Shaped Robotics. http://www.automation.hut.fi/iecat/moving_eye/home.html, 2001.

[10] Peter Röbke-Doerr. Navigation mit Sateliten. c't, 01/2003:150–151, Jan 2003.

[11] Stuart Russel and Peter Norvig. Artificial Intelligence. A Modern Approach. Prentice-Hall, 1995.

[12] Stuart Russel and Peter Norvig. Artificial Intelligence. A Modern Approach, 2nd Edition. Prentice-Hall, 2003.

[13] Robocup Team. Robocup. http://www.robocup.org, 2003.

[14] Robocup Team. Robocup 2003. http://www.robocup2003.org, 2003.

[15] RWTH Aachen Robocup Team. AllemaniACs. http://robocup.rwth-aachen.de, 2003.

[16] Sebastian Thrun. Robotic Mapping: A Survey. Technical report, School of Computer Science –Carnegie Mellon University, 2003.

[17] Andrew Tong. Star Trek TNG Episode: The Measure Of A Man. http://www.ugcs.caltech.edu/st-tng/episodes/135.html, 1995.

[18] Eric W. Weisstein. Voronoi diagram. http://mathworld.wolfram.com/VoronoiDiagram.html, 1999..