ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 10, April 2016

An Introduction to Robotics in Artificial Intelligence

Dr.I.Lakshmi,
Assistant Professor, Stella Maris College, Catherdal Road, Chennai-60086

Abstract: This document gives a short introduction to the basics of robotics in the context of artificial intelligence. It describes the very basics of robotics like sensors and effectors, gives an overview on robotic history, and introduces some basic problems encountered in modern robotics. It describes possible solutions to those problems without going deeply into theory.

I. INTRODUCTION

A. Capek and his Robots

The term "Robot" can be traced back to Karel Capek's play "R.U.R. Rossum's universal robots" (in 1921) that comes from the Czech word for "corvee".

B. A brief History of Robots

Robotics is based on two enabling technologies: Telemanipulators and the ability of numerical control of machines. Telemanipulators are remotely controlled machines which usually consist of an arm and a gripper. The movements of arm and gripper follow the instructions the human gives through his control device. First telemanipulators have been used to deal with radio-active material. Numeric control allows controlling machines very precisely in relation to a given coordinate system. It was first used in 1952 at the MIT and lead to the first programming language for machines (called APT: Automatic Programmed Tools). The combinations of both of these techniques lead to the first programmable telemanipulator. The first industrial robot using these principles was installed in 1961. These are the robots one knows from industrial facilities like car construction plants.

The development of mobile robots was driven by the desire to automate transportation in production processes and autonomous transport systems. The former lead to driver-less transport systems 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 after examples nature gave us or autonomous robots for underwater usage.

Since a few years wheel-driven robots are commercially marketed and used for services like "Get and Bring" (for example 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 since 1994. Honda's humanoid robot became well known in 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 friend but in reality we will only see robots for specific jobs as universal programmable machine slave in the near future (which leads to interesting questions, see [17]).

C. Definition: What is a Robot?

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 (covered in more detail in next section). As mentioned before Robots can be put into three main categories: manipulators, mobile robots and humanoid robots.

D. Robotics and AI

Artificial intelligence is a theory. The base object is the agent who is the "actor". It is realized in software. Robots are manufactured as hardware. The connection between those two is that the control of the robot is a software agent that reads data from the sensors, decides what to do next and then directs the effectors to act in the physical world.

II. THEORY AND APPLICATION

A. Robot Hardware

Sensors

Sensors are the perceptual interface between robots and their environment.1 On the one hand 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 [1]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 on mobile robots which need to take their energy with them in batteries. We have three types of sensors (no matter whether sensors are active or passive). These are sensors that either ² record distances to objects or ² generate an entire image of the environment or² measure a property of the robot itself. Many mobile robots make use of range finders, which measure distance to nearby objects. A

[1] For an overview of available sensor hardware especially for small robots see [4].

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 10, April 2016

common type is the sonar sensor (see [6] for an example). Alternatives to sonar include radar and laser (see Figure 1). 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 are long-range sensors like the Global Positioning System (GPS, see [8, 10]). The second important class of sensors is imaging sensors. These are cameras that provide images of the environment that can then be analyzed using computer vision and image recognition techniques. [1]For an overview of available sensor hardware especially for small robots see [4].[2] The third important class are 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 is unknown. Imagine a ton robot manipulator screwing in a light bulb.

Effectors

Effectors are the means by which robots manipulate the environment, move and change the shape of their bodies. 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 lets 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 (also known as yaw, roll and pitch). 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 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. Six degrees of freedom are required to place an object, such as a hand, at a particular point in a particular orientation.

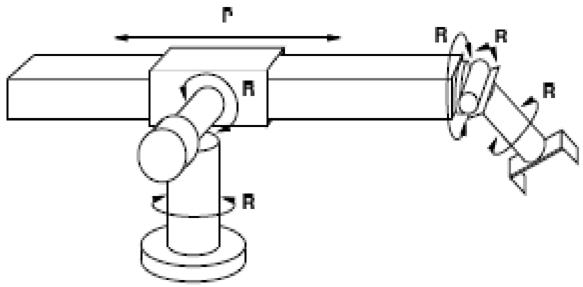

Fig 1: Manipulator

The manipulator shown in Figure 2 has exactly six de- Figure 2: The Stanford Manipulator [12] gees 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.

If you take your arm as an example you will notice, that it has more than six degrees of freedom. If you put your hand on the table you still have the freedom to rotate your elbow. Manipulators which have more degrees of freedom than required to place an end effector to a target location are easier to control than robots having only the minimum number of DOFs. Mobile robots are somewhat special. The number of degrees of freedom does not need to have corresponding actuated elements. Think of a car. It can move forward or backward, and it can turn, giving it two

DOFs. But if you describe the car's kinematic configuration you will notice that it is three-dimensional. On a flat surface like a parking site you can maneuver your car to any (x;y) point, in any orientation. You see that the car has 3 effective DOFs but only 2 controllable DOFs. We say a robot is non-holonomic 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 (think of parking a car: it would be much easier to be able to move the car sideways). But holonomic robots are mechanically more complex. Most manipulators and robot arms are holonomic and most mobile robots are nonholonomic.

Movement

For mobile robots a special group of effectors are 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 on a straight line[2]. If they move in opposite directions, the robot turns on the spot. An alternative is the synchronic drive[3], in which each wheel can move and turn around its own axis. This could easily lead to chaos. But if you assure the constraint that all wheels always point in the same direction and move with the same speed your robot is save. Both differential and synchronic drives are nonholonomic. Some more expensive robots use holonomic drives, which usually involve three or more wheels and can be oriented and moved independently.

Power Sources

Robots need a power source to drive their effectors. The most popular mechanism for both manipulator

[2]If you have ever tried to implement that (for example with a Lego Mindstorm) you know this can become hard especially with cheap hardware, see [7]

[3]Staying with the Mindstorm example you might want to have a look at [1].

actuation and locomotion is the electric motor. Other possible ways are pneumatic actuation using compressed gas and hydraulic actuation using pressurized fluids. They have their application niches but are not widely used.

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 back link to the robots home station. Finally you need a body frame to hang all the bits and pieces.

B. Robotic Perception

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. Good representation should meet three criteria: They should

- contain enough information for the robot to make a right decision
- be structured in a way that it can be updated efficiently
- be natural, meaning that internal variables correspond to natural state variables in the physical world

Filtering and updating the belief state is not covered here as it was covered in earlier presentations. Some topics are Kalman filters and dynamic Bayes nets. Information can also be found in [12].

Where am I? (Localization)

A very generic perception task is *localization*. It is the problem of determining where things are. Localization is one of the most pervasive perception problems in robotics. Knowing the location of objects in the environment that the robot has to deal with is the base for making any successful interaction with the physical world – manipulators have to know where the objects are they have to manipulate and mobile robots need to know where they are in order to find their way to a defined target location. There are three increasingly difficult flavors of localization problems:

- *Tracking* – If the initial state of the object to be localized is known you can just track this object.
- *Global localization* – In this case the initial location of the object is unknown you first have to find the object. After you found it this becomes a tracking problem.
- *Kidnapping* – This is the most difficult task. We take the object the robot is looking for and place it somewhere else. Now the robot again has to localize the object without knowing anything about the object's location. Kidnapping is often used to test the robustness of a localization technique under extreme conditions.

C. *Planning to Move*

After landing safely on the surface of a new planet, our autonomous ScoutBot somehow decides, that it should analyze a strange stone formation a couple hundred meters south. Wheels are set in motion and the robot finally reaches the new location. This is the most basic movement problem called *point-to-point motion problem*, which is to determine the right motions to change the position of the robot or its effectors to a new configuration without collision with known obstacles.

Furthermore it needs to take several examples from the stones. It uses a built-in drill to extract some stone fragments. This is a more complicated problem called the *compliant motion problem*. It comprises the motion of the robot while in contact with an obstacle.

Before an algorithm for one of the above problems an be developed, we need to find a representation which enables us to properly describe our problems and find solutions for them. The intuitive attempt would be to describe each movable piece of a robot by its Cartesian coordinates. This method is known as *workspace representation* because the robot and its environment share the same coordinate system. The problem with this representation is, that even in an obstacle free environment, a robot can never reach all possible workspace coordinates because some parts are fixed to each other and cannot move independently but have to obey certain *linkage constraints*. A solution to solve motion problems defined over workspace coordinates would have to generate paths that take these constraints into account which is not easy, as the state space is continuous and the constraints are nonlinear. Another, more practical approach is called *configuration space representation* where the state of the robot is represented by the configuration of its joints instead of Cartesian coordinates. The example robot has two joints. We can represent the state of the robot by just the two angles □ 1 and □ 2. If there are no obstacles present, one could simply connect start and end configuration with a straight line and let the robot rotate every joint to its new configuration with a constant speed. A configuration space can always be decomposed into two parts. The *free space* which comprises all attainable configurations and its complement called *occupied space*, which covers all configurations blocked by obstacles or restricted joint capabilities. This leads to a problem of configuration spaces. As it is in general easy to convert configuration space to workspace coordinates (which is called *kinematics*) the inverse operation (*inverse kinematics*), to generate a configuration from working space coordinates, is a difficult task as for certain configurations the solution is seldom unique therefore the practical approach is to generate a configuration and apply it to the robot, to see if it is in free space.

D. Moving

Now, as we may have found a path to reach a certain target, we need to actually move the robot or its effectors there. This can be trickier than one would expect, because robots are physical agents. They cannot follow arbitrary paths, except at arbitrary slow speeds. In general, a robot cannot simply set his effectors to a certain position, but needs to exert forces to put them to the desired new position.

Dynamics and Control

Better motion plans could be generated if, instead of basing them on kinematic models of the robot, *dynamic state models* would be used. Such models are typically expressed through *differential equations*. Unfortunately the complexity of dynamic space is greater than that of kinematic space rendering motion planning based on them for more than simplest robots impossible. Thus in practice simpler kinematic path planners are common. To overcome their limitations however a common technique is to use a separate *controller* to keep the robot on track. A controller which tries to keep a robot on it is preplanned *reference path* is referred to as reference controller.

Potential Field Control

Potential fields, which we already used as an additional cost function in motion planning, can also be used to directly generate robot motion. To do this, we need to define an attractive potential field, which pushes the robot towards its destination, and another field, which pushes it away from obstacles. The goal becomes the global minimum of the resulting potential; all other points have values based on their distance to the target configuration and their proximity to obstacles. Potential fields can be efficiently calculated for any configuration and scale very good with the dimensionality of the configuration space. Unfortunately the path can eventually be trapped in a local minimum. Nevertheless this method is great for local robot control but still requires global planning. Another point is, that it is a kinematic method and might fail if the robot is moving quickly.

Reactive Control

In cases, where we cannot get accurate models of our environment or where noise and lack of computational power might prevent application of the methods known so far another approach must be made. In some cases a *reflex agent* (reactive control) is more appropriate. The controller bases his behavior not on a model of the environment, but on immediate feedback (like short range proximity scanners or even reacting on being stuck). This interplay of a (simple) controller and a (complex) environment is often referred to as *emergent behavior*. As no model is perfect, this has to be seen in a wider sense. Other methods rely upon training data, rather than a model of their environment.

III REALITY

A. Types of Robots

Now we will see what robots are mainly used for nowadays.

Hard working Robots

Traditionally robots have been used to replace human workers in areas of difficult labor, which is structured enough for automation, like assembly line work in the automobile industry (the classical example) or harvesting machines in the agricultural sector. Some existing examples apart from the assembly robot are:

- Melon harvester robot
- Ore transport robot for mines
- A robot that removes paint from large ships
- A robot that generates high precision sewer maps

If employed in suitable environment robots can work faster, cheaper and more precise than humans.

Transporters

Although most autonomous transport robots still need environmental modifications to find their way they are already widely in use. But building a robot which can navigate using natural landmarks is probably no more science fiction. Examples of currently available transporters are:

- Container transporters used to load and unload cargo ships
- Medication and food transport systems in hospitals
- autonomous helicopters, to deliver goods to remote areas

Fig 2: Types of Robot

Insensible Steel Giants

As robots can be easily shielded against hazardous environments and are somewhat replaceable, they are used in dangerous, toxic or nuclear environments. Some places robots have helped cleaning up a mess:

- In Chernobyl robots have helped to clean up nuclear waste
- Robots have entered dangerous areas in the remains of the WTC
- Robots are used to clean ammunition and mines all around the world

For the same reasons robots are sent to Mars and into the depth of the oceans. They explore sunken ships or walk the craters of active volcanoes.

Servants and Toys

Robots may not yet be a common sight in our world, but we already encounter them in many places. Many modern toys like the Sony Ambo are conquering today's children's life. Robots are developed that will help older people to have a better and more secure life (ball-robot Rollo, see [9]). Nowadays, they start to come to us as toys or household helpers. Their time has just begun.

B. Showcase: AllemaniAC Robocop Robots

As a real life example we will take a look at the robotic football team of the RWTH Aachen which participates in the Robocop tournament (see [13]).

Sensors

The part that is currently worked on is localization: The robot has to view of an AllemaniAC know where it is, where its team mates and opponents are and where the ball and goal are. The robot uses a laser range scanner to detect obstacles like its opponents. It is mounted about 30 cm above the floor. This way the ball is not detected by the range scanner. For this purpose the robot has a camera.

Fig 3: Schematic side

It is also used to track the ball after the robot detected it and find to the goal and landmarks that are placed in the corners of the field for supporting the localization. There are plans to remove those landmarks as well in the future. The robot uses a form of MCL for localization. The WLAN module is used to tell team mates if the robot can see them or if they see the ball. This is like real players shouting where other players should move or where the ball is. Right now the robots do not communicate directly to each other but they have a central controlling computer besides the field that acts like a message preprocessor and command repeater.

Effectors

The AllemaniAC uses a differential drive with shaft decoding for movement. The motor is very powerful with 2.5 kW moving the robot with a speed of 12 km/h. Of course a football player has to interact with the ball. Therefore it has a "shoot effector" that consists of two electro-magnetic spools. These spools enable the robot to control the strength that it kicks the ball with. This is currently a unique feature of the AllemaniAC (as far as we know).

CPU and Communication

The robot has two computers on board with a Pentium III (933MHz) each. One is used for image recognition and the other for higher functions like localization and communication with base station. The robots cannot communicate with each other yet. On tournaments there is usually too much traffic on all WLAN channels from other teams and the folks around. Since the robots still use 802.11b there is not enough bandwidth available to work properly and reliable under such conditions.

Software

Although we did not cover software architectures used for robot construction in this talk we want to present some principles of the underlying software of the AllemaniAC robot. The AllemaniAC is based on ReadyLog, an enhanced version of IndiGolog (see [3]). ReadyLog extends IndiGolog with continuous actions (tracking the ball must be done all the time while the robot is moving) and actions with uncertain outcome (if the robots plays the ball towards the goal he cannot be sure that the ball hits the goal. An opponent might intercept it).

Current State

As seen in a demonstration[4] there is still a lot of work that has to be done before the robots really play as a team and do not interfere with each other. But that is why this is science and not a commercial product. We will see how well the AllemaniACs does perform in this years Robocup in Padua (see [14]). For more information about the AllemaniACs see [15].4You may find several videos and more information on [15] in the documents section.

[4]You may find several videos and more information on [15] in the documents section

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. Indigo log overview. http://www.cs.yorku.ca/~alexei/ig-oaa/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 Howto. http://www.atsemi.com/article/Howto.htm, 1999.

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

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

[9] Helsinki University of Technology. Moving Eye - Virtual Laboratory Excercise 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.

[19] GregWelch and Gary Bishop. An Introduction to the Kalman Filter. Technical report, Department of Computer Science – University of North Carolina at Chapel Hill, 2003.