IJSRD - International Journal for Scientific Research & Development| Vol. 6, Issue 07, 2018 | ISSN (online): 2321-0613

Artificial Intelligence in Robotics

Saikrishna L. Boddula[1] Vaibhav S. Andhale[2] Vipul S. Dalal[3] Ashwini D. Padekar[4]
[1,2,3]Student [4]Assistant Professor
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]MGM's College of Engineering & Technology, Kamothe, Navi Mumbai, India

Abstract— Since Robotics is the field concerned with the connection of perception to action, Artificial Intelligence must have a central role in Robotics if the connection is to be intelligent. Artificial Intelligence addresses the crucial questions of: what knowledge is required in any aspect of thinking; how that knowledge should be represented; and how that knowledge should be used. Robotics challenges AI by forcing it to deal with real objects in the real world. Techniques and representations developed for purely cognitive problems, often in toy domains, do not necessarily extend to meet the challenge. Robots combine mechanical effectors, sensors, and computers. AI has made significant contributions to each component. We review AI contributions to perception and object oriented reasoning.

Key words: Manipulators, Degree of Freedom, Monte Carlo Localization, Skeletonization

I. Introduction

Robots are physical agents that perform tasks by manipulating the physical world. To do so, they are equipped with effectors such as legs, wheels, joints, and grippers. Effectors have a single purpose: to assert physical forces on the environment. Robots are also equipped with sensors, which allow them to perceive their environment. Present day robotics employs a diverse set of sensors, including cameras and lasers to measure the environment, and gyroscopes and accelerometers to measure the robot's own motion. Most of today's robots fall into one of three primary categories. Manipulators, or Robot Arms are physically anchored to their workplace, for example in a factory assembly line or on the International Space Station. Manipulator motion usually involves a chain of controllable joints, enabling such robots to place their effectors in any position within the workplace.

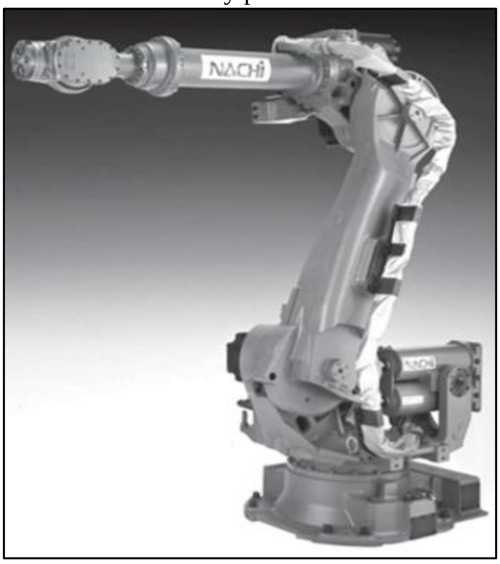

Fig. 1: An Industrial Robotic Manipulator for Stacking Bags on a Pallet. Image Courtesy of Nachi Robotic Systems

The second category is the Mobile Robot. Mobile robots move about their environment using wheels, legs, or similar mechanisms. They have been put to use delivering food in hospitals, moving containers at loading docks, and similar tasks. The third type of robot is a Mobile Manipulator. Humanoid robots mimic the human torso. Mobile manipulator scan applies their effectors further afield than anchored manipulators can, but their task is made because they don't have the rigidity that the anchor provides.

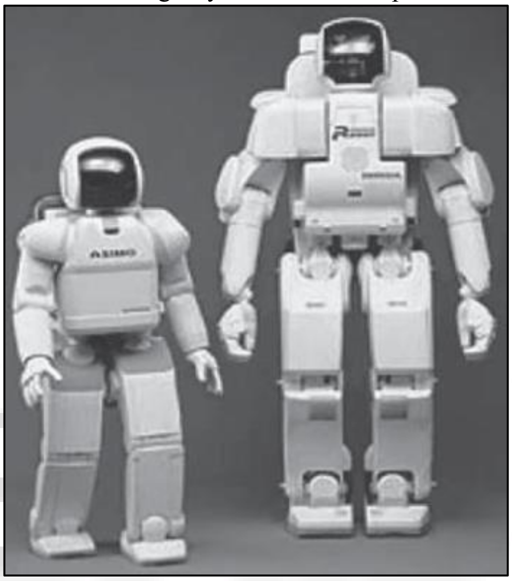

Fig. 2: Honda's P3 and Asimo Humanoidrobots

II. Robot Hardware

A. Sensors

Sensors are the perceptual interface between robot and environment. It includes Active sensors, Passive sensors, Range Finders, Stereo Vision, Tactile Sensors, Location sensors, Proprioceptive sensors, inertial sensors, Force sensors and Torque sensors.

B. Effectors

Effectors are the means by which robots move and change the shape of their bodies. Effectors are the means by which robots move and change the shape of their bodies. To understand the design of effectors, it will help to talk about motion and shape in the abstract, using the 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.

III. Robotic Perception

Perception is the process by which robots map sensor measurements into internal representations of the environment. Perception is difficult because sensors are noisy, and the environment is partially observable, unpredictable, and often dynamic.

A. *Localization & Mapping*

Localization is the problem of finding out where things are—including the robot itself Knowledge about where things are is at the core of any successful physical interaction with the environment. For example, robot manipulators must know the location of objects they seek to manipulate; navigating robots must know where they are to find their way around. Localization using particle filtering is called Monte Carlo localization, or MCL. A Monte Carlo localization algorithm using a range-scan sensor model with independent noise.

B. *Other Types of Perception*

Not all of robot perception is about localization or mapping. Robots also perceive the temperature, odors, acoustic signals, and so on. Many of these quantities can be estimated using variants of dynamic Bayes networks. All that is required for such estimators are conditional probability distributions that characterize the evolution of state variables over time, and sensor models that describe the relation of measurements to state variables. Probabilistic techniques outperform other approaches in many hard perceptual problems such as localization and mapping. However, statistical techniques are sometimes too cumbersome, and simpler solutions may be just as effective in practice. To help decide which approach to take, experience working with real physical robots is your best teacher.

C. *Machine Learning in Robot Perception*

Machine learning plays an important role in robot perception. This is particularly the case when the best internal representation is not known. One common approach is to map high dimensional sensor streams into lower-dimensional spaces using unsupervised machine learning methods. Such an approach is called low-dimensional embedding. Machine learning makes it possible to learn sensor and motion models from data, while simultaneously discovering a suitable internal representation. Another machine learning technique enables robots to continuously adapt to broad changes in sensor measurements. Picture yourself walking from a sun-lit space into a dark neon-lit room. Clearly things are darker inside. But the change of light source also affects all the colors: Neon light has a stronger component of green light than sunlight. Yet somehow we seem not to notice the change. If we walk together with people into a neon-lit room, we don't think that suddenly their faces turned green. Our perception quickly adapts to the new lighting conditions, and our brain ignores the differences.

IV. PLANNING TO MOVE

All of a robot's deliberations ultimately come down to deciding how to move effectors. The point-to-point motion problem is to deliver the robot or its end effector to a designated target location. A greater challenge is the compliant motion problem, in which a robot moves while being in physical contact with an obstacle. An example of compliant motion is a robot manipulator that screws in a light bulb, or a robot that pushes a box across a table top. We begin by finding a suitable representation in which motion planning problems can be described and solved.

A. *Configuration Space*

The space of robot states defined by location, orientation, and joint angle, is a better place to work than the original 3D space.

B. *Path Planning*

The path planning problem is to find a path from one configuration to another in configuration space. the complication added by robotics is that path planning involves *continuous* spaces. There are two main approaches: cell decomposition and skeletonization. Each reduces the continuous path-planning problem to a discrete graph-search problem.

V. MOVING

Our plans particularly those produced by deterministic path planners—assume that the robot can simply follow any path that the algorithm produces. In the real world, of course, this is not the case. Robots have inertia and cannot execute arbitrary paths except at arbitrarily slow speeds. In most cases, the robot gets to exert forces rather than specify positions. This section discusses methods for calculating these forces.

- Dynamics and control
- Potential-field control
- Reactive control
- Reinforcement learning control

VI. ROBOTIC SOFTWARE ARCHITECTURES

A methodology for structuring algorithms is software called a software architecture. An architecture includes languages and tools for writing programs, as well as an overall philosophy for how programs can be brought together. Modern-day software architectures for robotics must decide how to combine reactive control and model-based deliberative planning. In many ways, reactive and deliberate techniques have orthogonal strengths and weaknesses. Reactive control is sensor-driven and appropriate for making low-level decisions in real time. However, it rarely yields a plausible solution at the global level, because global control decisions depend on information that cannot be sensed at the time of decision making. For such problems, deliberate planning is a more appropriate choice. Consequently, most robot architectures use reactive techniques at the lower levels of control and deliberative techniques at the higher levels. We encountered such a combination in the discussion of PD controllers, where we combined a (reactive) PD controller with a (deliberate) path planner. Architectures that combine reactive and deliberate techniques are called hybrid architectures.

- Subsumption architecture.
- Three-layer architecture.
- Pipeline architecture.

VII. APPLICATION DOMAINS

- Industry and Agriculture.
- Transportation.
- Robotic cars.
- Health care.

- Hazardous environments.
- Exploration.
- Personal Services.
- Entertainment.
- Human augmentation.

References

[1] Chetan Sakharam Tirgul, Mangesh Raghunath Naik, "Artificial Intelligence And Robotics" ISSN:2278 – 1323 Volume 5, Issue 6, June 2016

[2] Antonio Chella, Luca Iocchi, Irene Macaluso, "ARTIFICIAL INTELLIGENCE AND ROBOTICS", Dec 19, 2013

[3] Artificial Intelligence Robotics https://www.cs.umn.edu/research/research_areas/robotics-and-artificial-intelligence

[4] Artificial Intelligence Robotics https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_robotics.htm