Unifying Undergraduate Artificial Intelligence Robotics: Layers Of Abstraction Over Two Channels

Frederick L. Crabbe
Computer Science Department
United States Naval Academy
572M Holloway Rd Stop 9F
Annapolis, Maryland 21402

Abstract

From a Computer Science and Artificial Intelligence perspective, Robotics often appears as a collection of disjoint, sometimes antagonistic sub-fields. The lack of a coherent and unified presentation of the field negatively impacts teaching, especially to undergraduates. The paper presents an alternative synthesis of the various sub-fields of Artificial Intelligence robotics, and shows how these traditional sub-fields fit in to the whole. Finally, it presents a curriculum based on these ideas.

Introduction

Modern Artificial Intelligence robotics education treats the field as a collection of overlapping subfields. An examination of the current robotics textbooks (McKerrow 1991; Arkin 1998; Dudek & Jenkin 2000; Murphy 2000; Niku 2001; Siegwart & Nourbakhsh 2004; Craig 2005; Choset *et al.* 2005) indicates that these subfields are: traditional planning based robotics, behavior-based robotics, probabilistic robotics, mobile robotics, and engineering robotics. Reading these texts gives the impression that each of these fields is overlapping, yet distinct, except for engineering robotics, which many Computer Science/Artificial Intelligence instructors consider to be an entirely separate field.

This fragmentation of fields likely derives from the sometimes fractious relations between the fields as they competed for primacy. "The whole idea of plan execution and the runtime maintenance of something called a 'plan' is misguided." (Brooks 1986) "This development follows a much broader trend in mobile robotics, where probabilistic techniques are commonly the method of choice over more ad hoc approaches, such as behavior-based techniques." (Thrun 2002) While such competition is natural in a research setting, it makes it difficult to present these multiple fields coherently to an undergraduate. Exacerbating the situation, many robotics instructors present topics chronologically according to historical development. This approach, while interesting to practitioners, fails to put the areas in their *technical* context.

This paper argues for a perspective that unifies all the competing robotics sub-fields into a single framework for instruction. By using the organizing principle that robotic systems are best understood as layers of abstractions over input and output channels, it results in a more natural order of topics and emphasizes their relations rather than differences. We will begin by presenting the layered framework, followed by an example AI Robotics curriculum that emphasizes the similarities and encourages a cohesive big-picture understanding of the field. We will then compare the curricular themes presented in other robotics textbooks. Finally we will discuss the sometimes surprising implications of this view in how the various robotics sub-fields relate to each other.

Layers of Abstraction

The application of layers of abstraction in Computer Science is a well known technique used either prescriptively to coordinate standards development, or descriptively to make sense of complicated processes and ease comparison of apparently conflicting ideas. The classic example of the former is the OSI network layer system (ISO 1994) which specifies an organization for computer networking (Figure 1, left). On the other hand, layers of abstraction are used as a general guideline for the understanding of complicated systems throughout Computer Science and Engineering, such as the organization of computer hardware, operating systems or large pieces of software (Figure 1, right). The claim made here is that the technique of applying layers of abstraction as a descriptive tool is useful for understanding robotics systems.

Layers of abstraction are a natural way to characterize intelligent robotics, in which low-level perceptions are converted to high-level decisions and actions and then converted back down to low-level motor movements. Every system does this by: processing multiple sensor inputs; combining the input into increasingly higher levels of abstraction until an action decision can be made; and breaking down the decision into increasingly more specific information until it can be executed as motor commands. Intelligent Robotics as a field is best seen as two information channels (input and output) crossing multiple layers of abstraction from physical signals to sophisticated symbols (such as multiple term logical representations) . All of the major paradigms fit into and can be interrelated by this paradigm. Figure 2 shows the framework presented here with two channels and the layers of abstraction through which information is processed. The top row is the input channel, starting with physical signals on the left, and passing through multiple abstractions as

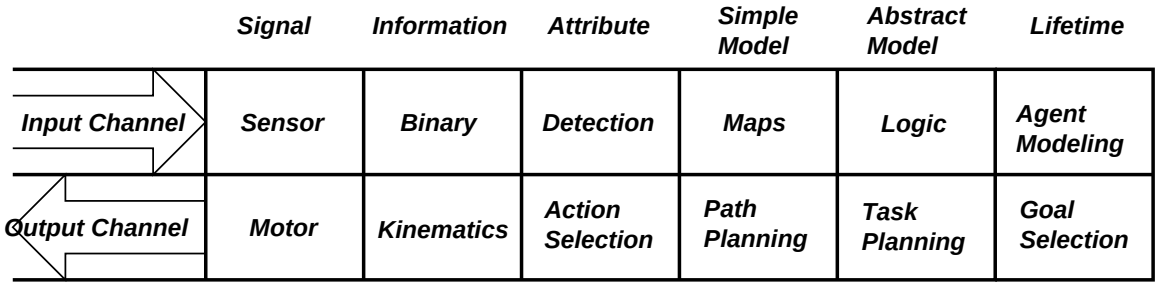

Figure 2: The Layers of Abstraction framework. Input flows from left to right in the figure; output flows from right to left. Higher levels of abstraction are to the right.

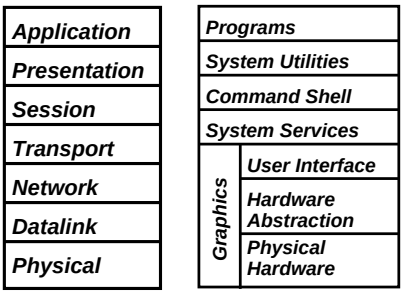

Figure 1: Two standard layered models of abstraction. On the left is the ISO/OSI network standard, an example of prescriptive model describing how a system should be designed. On the right is a general layered operating system model as a layered architecture (Drawn from Tanenbaum (Tanenbaum 1992) and Solomon (Solomon 1998)). It is an example of a layered model describing how systems are currently designed.

it moves to the right. The bottom row is the output channel, moving from high-level symbols on the right to motor signals on the left. Each input channel can be made up of multiple pathways along which individual pieces of information can travel. For instance, input from multiple sensors would all travel in the input channel, but until that information is fused, it would travel along multiple pathways. All of the common robot architectures can be seen in this view as variations on how many layers of abstraction are passed through, and at which layer (or layers) information *crosses over* from the input channel to the output channel.

The Layers

Figure 2 describes six distinct layers. Because this model is descriptive there is considerable room for adjustment in both the numbers of layers and where they are divided. These particular layers were identified because they correspond to major robotic architectures in the literature and have proved useful in teaching. They are the signal, information, attribute, simple model, abstract model, and lifetime layers.

The lowest layer is the *signal layer*. Information at this layer takes the form of electrical impulses from sensors and to motors. While all electronic robots have activity at this layer (otherwise they would never move), few perform any crossover from input to output at this layer. The classic exceptions to this are the Braitenberg Vehicles (Braitenberg 1984) which directly connect sensors to motors. Figure 3 shows the information flow in the framework of a Braitenberg Vehicle.

The next layer up is the *information layer*, where analog input and output signals are handled digitally; this layer is the interface between the world of physics and the world of information. In the input channel, the electrical signals are converted to bits in memory, typically with an analog to digital converter. In the output channel, the information layer performs kinematic analyses to convert positions in robot geometry into motor positions and thus motor current outputs. Input to the output channel of this layer is a desired position in space. The process of inverse kinematics is to take that position and convert it into a set of motor position that would place the robot in that location. From there, these motor positions are converted directly to electrical signals to the motors themselves.

It is at this layer where much of what might be called engineering robotics takes place (Figure 3). Sensor inputs are used to generate the desired robot positions that are then converted to motor movements, such as a camera directing an articulated arm to grasp an object[1].

In the *attribute layer* the input channel generalizes the information input by recognizing simple environmental states such as *landmark-detected* or *goal-detected*. These differ from the treatment of the raw data in the layer below in that there is some processing required, often integrating sensor information over short periods of time from multiple sensors. In the output channel collections of possible actions (by action we mean 'desired position in the environment') are weighed, and an action is selected, resulting in a suitable specification to pass down to the information layer. The attribute layer is where most of the action takes place in behavior-based architectures (Figure 3). Collections of independent modules fuse input to recognize simple environ-

[1]This characterization is not exactly accurate, as what many researchers might consider "engineering robotics" extends well into the model layer (Latombe 1991; McKerrow 1991). There are, however, perspectives that do primarily limit engineering robotics to these lower layers (Craig 2005). This will be discussed in greater depth below.

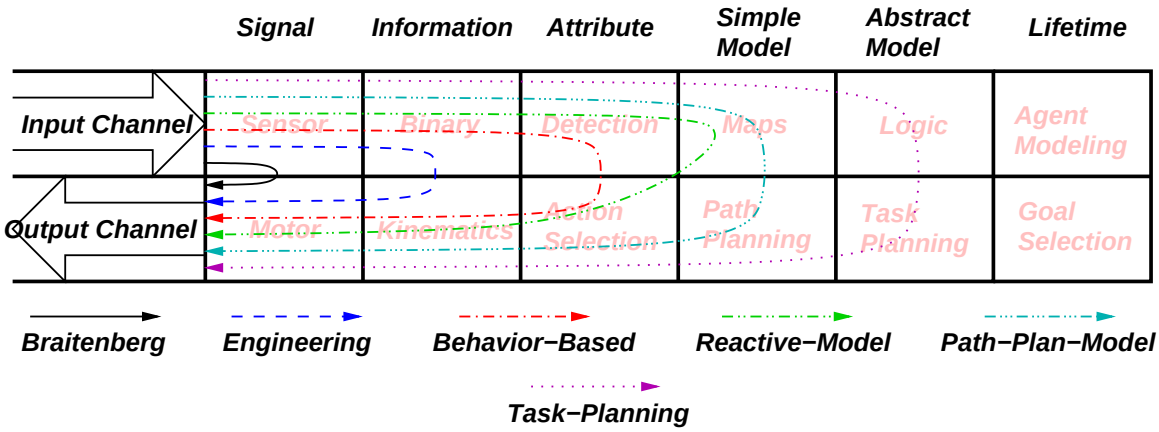

Figure 3: Information flow in different robotics architectures.

ment properties and make action recommendations, which are then selected from to result in a new desired robot position.

In the *simple model layer*, the input channel begins to build concrete and explicit models of the external world. For mobile robots, these take the form of maps of the navigable space, while for manipulators, these models take the form of a configuration space. For example, probabilistic occupancy grid maps (Moravec & Elfes 1985; Moravec 1988) are low-level, simple abstractions, as they make no serious attempt to cluster the occupied cells into objects. The construction of the simple models is performed by collecting the detected input in the attribute layer into obstacles and free space. These can be further processed into more abstract map models, such as generalized Voronoi diagrams (Aurenhammer 1991; Choset & Burdick 2000). Note that occupancy grids and Voronoi diagrams both apply to mobile robots navigating through real space *and* to manipulator arms navigating through configuration space. With the transition to the output channel at the simple model layer, comes path planning, in which a path is found through free space to get the robot from one position to another (we refer to this as a path-plan model architecture). Again, the techniques are the same in real space and configuration space. Once a path is determined, the commands can be passed down to the action selection mechanism in the attribute layer. The simple model layer accounts for the rest of engineering robotics and most of mobile robotics (Figure 3).

From the map models, path planning typically occurs, but sometimes, instead of path planning, the robot behaves reactively toward that model, skipping the planning part altogether (Yamauchi, Schultz, & Adams 1998). We call this the reactive-model architecture.

The *abstract model layer* takes the representation of space from the simple model layer, and generates more sophisticated models, such as propositions of relations about the world state. These are usually described with a logic, such as predicate logic, or an AI planning system (McDermott 1992). These logic sentences are built out of the information in the lower layers by grouping together data from the maps and other detected information from the layers below. In the output channel, a full-fledged task planning system can use these representations to generate "good old fashioned AI"-style plans for achieving goals that may involve more than just navigation. Steps in the plan, such as "go to the door" can be converted into lower level actions by the path planning systems of the lower layer. Traditional planning based systems also perform most of their computation at the model layer (Figure 3).

In all of the paradigms presented up to this point, input information is abstracted up to some layer where it crosses over to the output channel and converted to motor commands. There is no reason that information cannot cross from input channel to output channel at multiple layers. This is precisely what happens in the planning/behavior hybrid systems (Pell *et al.* 1998, e.g.). In these systems, longer term information is modeled and planned over, which is used to direct the reactive decisions being made in the attribute layer (Figure 4).

In the top *lifetime layer*, decisions are made about the longer term behavior of the robot. It is here that other agents are modeled in order to coordinate collective behavior, and the robots are able to consider what tasks they will pursue over their lifetimes. Actual robot systems that operate at this level are rare, especially those that perform true goal selection. Activity in multiple robot coördination has resulted in some modeling of other agents, but most multi-robot systems encode assumptions about other robot behavior rather than perform true modeling (Matarić 1992; 1997).

There are two observations to be made in examining the layers from left to right. First, both the amount of state required and the time window over which that state integrates sensor input increases. For instance, when comparing the input processing at attribute and the model layers, processing at the attribute layer is both less intense than at the model layer and it requires fewer input samples in order to generate output. The model layer needs to integrate many input samples before it can build a model suitable for use in decision making. The second observation is, moving from left to

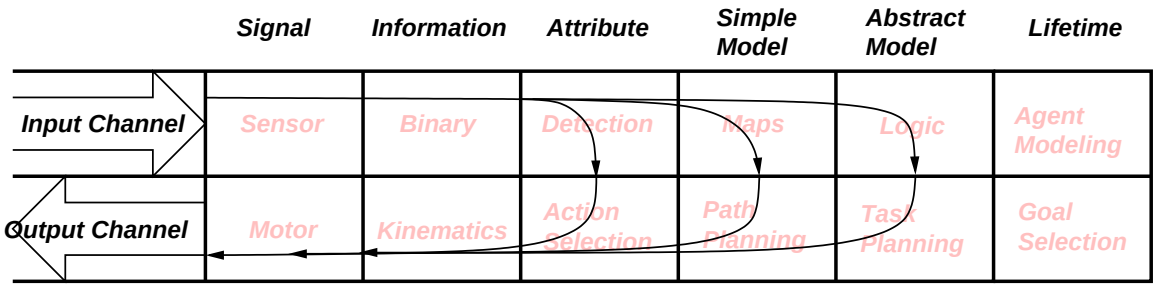

Figure 4: Information flow in hybrid systems.

right, the number of pathways used in the input channel decreases as information is consolidated into the abstractions. For example, input from multiple sensors is typically combined in order to recognize properties and smooth out errors at the attribute layer. These detections are further combined into maps at the simple model layer. This process reverses as information travels from right to left down the output channel, where an action is broken into the multiple motor commands necessary for completion.

The Curriculum

The two-channel layered model described above suggests a particular curriculum based on alternately presenting the workings of each layer, followed by one or two example architectures that focus on that layer. This enables the students to implement an example of that architecture on a robot suited to that layer, thus experiencing the power of architectures at that layer while simultaneously seeing the connections to the layers below. We have developed this curriculum in a four hour per week robotics course taken primarily by twelve to sixteen senior Computer Science majors who have not necessarily taken an Artificial Intelligence course. In this course we try to present an AI robotics perspective while emphasizing how each robot architecture relates to the others in the context of the layered model.

For the signal layer, we present the basic operation of the sensors and electric motors found on most robots. As an example robot architecture, we present Braitenberg's architecture as described in *Vehicles*, chapters one through five (Braitenberg 1984). The students can experiment directly with these ideas using simple commercial robots such as the BYO-bot (Miller 2005) or some of the BEAM robotics kits (Hrynkiw & Tilden 2002). These robots provide some of the standard functions of the Braitenberg Vehicles, but are not programmed by the student directly connecting sensors and motors with wires, and are limited in the number of ways it can be programmed. To remedy this, we designed a similar low-cost robot that the students can program by connecting wires. The sensors and motors have pluggable connectors into which the students can insert wires to attach the sensors to the motors so that they resemble the diagrams in the Vehicles book. Our initial design was developed out of a laboratory assignment in a robot engineering class (Bishop & Wick 2005) (Figures 5 and 6), but designs in development now also include potentiometer knobs to adjust connection

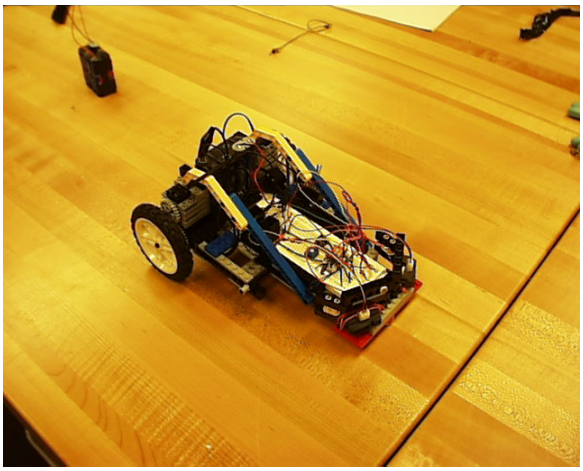

Figure 5: Photograph of a student-designed Braitenberg robot prototype.

strength.

For the information layer, we present the mathematical models of engineering robotics, with emphasis on kinematics in the Denavit-Hartenberg system (Denavit & Hartenberg 1955) for rigid arms and extend it to the integral based system for mobile robots used in Dudek and Jenkin (Dudek & Jenkin 2000). In keeping with emphasis on similarities and cohesion, we derive the results for mobile robots using both systems to show their equivalence. The students perform both forward and inverse kinematics on both a rigid arm and a mobile robot. For the rigid arm we use the Robix Rascal (Advanced Design Inc. 2005) a rugged kit suitable when high accuracy is not required (Figure 7). For the mobile robot we use the aging but still viable Rug Warrior (A K Peters, Ltd. 2005) (Figure 8).

In presenting the attribute layer, we discuss basic techniques for identifying properties in input streams, sensor fusion, and action selection mechanisms. For example, we discuss the combination of multiple sensors such as infrared and vision to determine the distance and bearing to obstacles. We also discuss the combination of multiple sensor readings over time as a separate form of sensor fusion. From this we present behavior-based architectures, how behaviors are designed and actions are selected, emphasizing how this

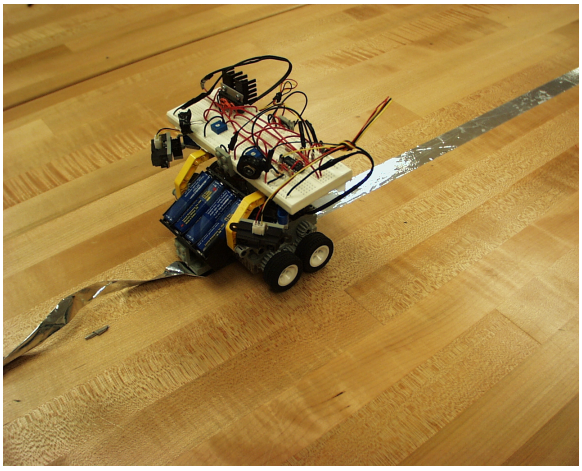

Figure 6: Photograph of an alternative Braitenberg robot prototype.

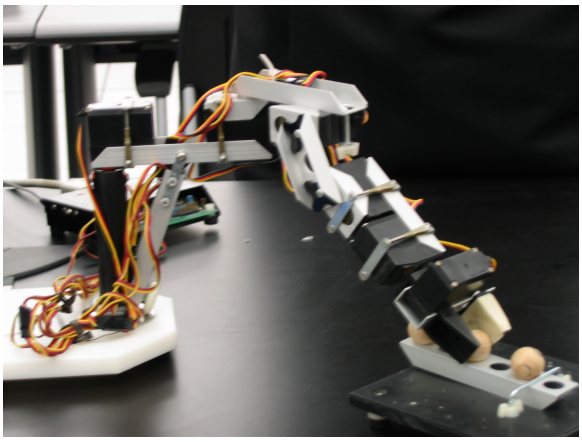

Figure 7: Photograph of a Robix Rascal Arm.

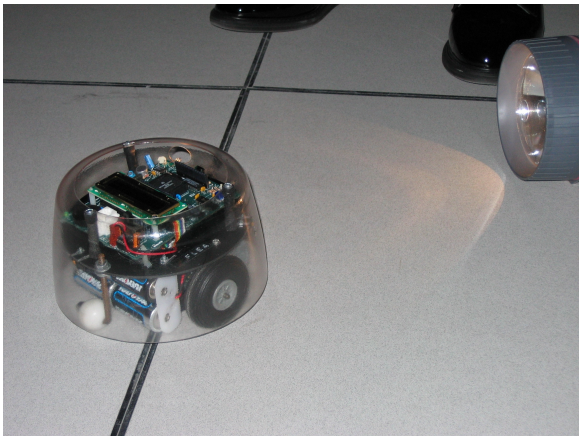

Figure 8: Photograph of a Rug Warrior performing ded. reckoning.

leads to robust behavior in chaotic environments. The corresponding student assignment is to implement a behavior-based system for a chaotic multi-robot task, such as playing a game of capture the flag. These sorts of task emphasize the benefits of more reactive systems in the face of real-time unexpected situations but not requiring more sophisticated reflection and planning to solve the task. Depending on the nature of the task, we use Rug Warrior or Khepera (K-Team S.A. 2005) robots.

At the simple model layer, we present the basics of occupancy grid maps and how they can be used as a representation of the configuration space of either a manipulator or a mobile robot[2]. This is followed by topographical maps, and how they could be built out of occupancy grids by identifying objects, and constructing Voronoi diagrams or other roadmaps. As we shift over to the output channel, we present first how reactive modules in the attribute layer can use maps such as occupancy grids, by treating the input from the map as if it came directly from the environment (a technique some authors refer to as "virtual sensors" (Asada 1990)). Then we show how paths can be found by searching either the grid-like spaces of occupancy grid maps, or the graphs of the Voronoi diagrams. Once paths are calculated they can be passed down to the attribute layer, where the first step in the path is determined (and weighed against other possible courses of action).

In the abstract model layer, we discuss logical representations of the world and how they can be gathered both from properties in the identification layer and from the maps built in the simple model layer. For the output channel, we discuss task-planning and how that can use logical descriptions to solve hard problems in the world. Because many of the students have not yet taken Artificial Intelligence, we present the AI problems solely as finding paths in abstract graphs of plan-space and leave the algorithmic details to the AI course. In our model, steps in the task-plan are actions that require path-planning to solve, such as "go-to-red-door". Each step in the plan is passed down to the simple model layer, where a path for the action is calculated. The action for manipulator plans, such as "grasp-hammer," are passed down to the path planning system in the same manner. The student's final project assignment is to use a Khepera robot equipped with a gripper to move blocks (Figure 9).

The balance of the class discussions focus on the lifetime layer, where the highest level operations take place. It is at this layer that much of the cutting edge research is performed and thus it is difficult to give a clear picture of where these issues are headed. We typically concentrate on multi-agent systems, the coördination of robots to perform a task, and how fully autonomous robots might select which goals they want to pursue and which they do

[2]Technically, the configuration space of a mobile robot would include both its *pose* (x,y position plus orientation) and the angular position of the wheels or legs. Since most often, only the pose is interesting, the configuration space is defined ignoring the configuration of the locomotors. This results in the configuration space being a nearly direct representation of the physical space of the surrounding area.

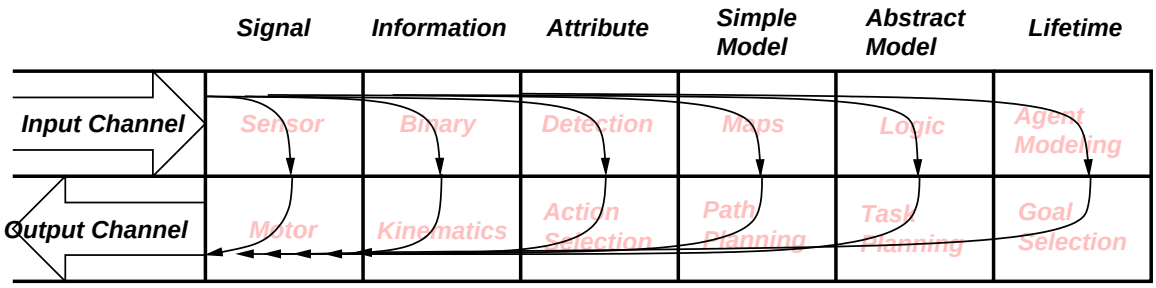

Figure 10: The logical conclusion.

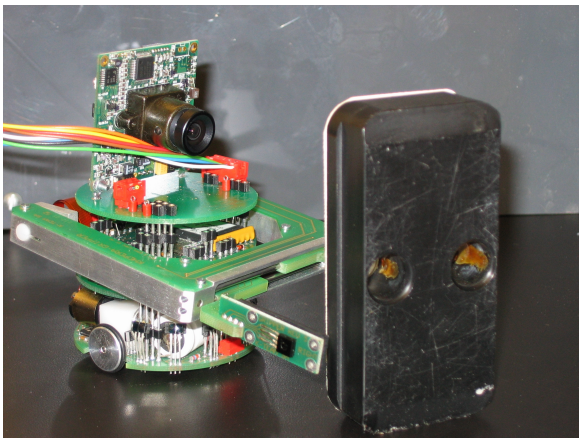

Figure 9: A Khepera robot maneuvering to lift a block.

not. Although these issues apply throughout the layers, this is a good place to discuss the ethics of fully autonomous robotics, in terms of worker displacement, liability, and of course, independence (the "Terminator" issue). We also point out that the logical conclusion of the layers and the hybrids is a model where information crosses over all the layers (Figure 10). While there may not be any examples of such an artificial system today, it does echo some current psychological models of animals and humans (Gallistel 1980; 1990). Reflex reaction corresponds to the signal layer, while brain-stem and cerebellum behavior correspond to the information and attribute layers. While the physiological mechanisms are less well understood at higher layers, psychological evidence indicates that there is pathway crossover at all layers in human behavior.

At the conclusion of the class we can re-emphasize that all of the styles of robotics share the same general architecture, and only vary on how many levels of abstraction are applied and at what levels information crosses from the input to the output channel. We also re-emphasize that just as with good software engineering practices, high levels of abstraction reuse implementations of lower levels.

Getting It All To Fit

A curriculum as presented here has more material than might be covered in other robotics courses. We feel that to give an accurate assessment of the whole of AI robotics, it is important to look at the whole of robotics and how these pieces fit together. Fortunately, due to the nature of the formalism, the material need not be overwhelming. The modular layered approach means that there is less repetition of similar material in different architectures. For instance, when presenting behavior-based and task-planning architectures separately, the lower layers are essentially presented twice. By using a bottom-up approach in the layered model we advocate here, this duplication is avoided because the explanation of the behavior-based layers is the foundation for the task-planning layers.

Even given the efficiency of this approach, courses move at different speeds from semester to semester, depending on both the students and the instructor. Fortunately, the approach makes it easy to leave out a module without further repercussions. Any particular module can be treated as a black box, and all that needs to be presented is the specification and interface, what it does rather than how it works. In this manner, any particular section can be jettisoned when it comes time.

Finally, the projects we've presented here seem to add up to a great deal of programming, implementing two inverse kinematics systems, a behavior-based system and a task-planning system. Our approach to alleviating this is to provide as many of the program modules as possible, especially from the lower layers. For example, in the planning project, the students are given all the tools at the lower layers to perform the sensory processing, and take the actions proposed by their planning system. The development of this library of tools was done over several years by the students themselves. As the class was being built, assignments were made with the intent that they would contribute to the suite of libraries for later assignments. This process is not complete, and for each assignment, one group is given a different assignment from the others for addition to the library.

Other Approaches

This section attempts to compare our framework and themes for teaching AI robotics to undergraduates to those themes and frameworks presented in current textbooks. When comparing the approach described here to that described (or at least adhered to) in the available textbooks, it is important to note the difference in purpose. The motivation of our approach is to present a principled perspective of the entirety of

robotics, suitable for an undergraduate survey course with an AI bent. Most of the existing books explicitly narrow their focus to a particular area in robotics, as would be appropriate for a graduate course. Even so, most of these texts do have perspectives and themes that can can be discussed.

Artificial Intelligence: A Modern Approach

Russell and Norvig's influential *Artificial Intelligence: A Modern Approach* (Russell & Norvig 2003) is of course not actually a robotics textbook, but the second edition does have a chapter on robotics written mostly by Sebastian Thrun. The overall theme of the Russell-Norvig text is that of agents sensing and acting in an environment. That theme is essentially redundant when it comes to robotics, as all robots are agents by definition and application of the theme is not helpful in differentiating between the various architectures. If there could be said to be a secondary theme, it could be that "[t]he literature on robotics research can be divided roughly into two parts: mobile robots and stationary manipulators."(p. 940) While accurate, the ontological position that it takes (there are manipulators, mobile robots, and hybrids of the two) appears to emphasize the differences rather than the similarities between these two types of robots. Within the chapter, the sequence of the sections is: hardware, perception, motion planning, motion planning under uncertainty, moving, software architectures, and applications. There is a particular emphasis on probabilistic approaches. The section on software architectures briefly discusses layered architectures as a combination of deliberative and reactive processes, and mentions that most modern software systems contain a variant of the architecture, hinting at the formalism we have presented here.

Interestingly, what a lay person might consider intelligent behavior in a robot is never actually mentioned. As far as specific robot capabilities are concerned, the chapter ends after path planning. Of course, high level issues such as task planning and natural language are covered at great length in others areas of the book, but all these techniques have special concerns when applied to robots and are missing from this chapter. This disconnect is present in most of the intelligent robotics books, and appears to reflect the AI robotics community's current research emphasis on effective spatial reasoning over higher-level task composition.

Introduction to AI Robotics

Introduction to AI Robotics by Robin Murphy (Murphy 2000) is, by topic area, the most appropriate text for an intelligent robotics course. The book itself is organized into two (overlapping) mini-books, the first on software architectures, and the second on navigation. Manipulators are not discussed to any depth.

Prof. Murphy comes from the behavior-based robotics tradition, and follows the party line from that perspective. "Shakey-style" robotics are presented and then criticized, leaving behavior-based robotics as the only sane solution (or perhaps if the problem is especially complex, a hybrid of the two is what is called for). Notably missing is a discussion of where high level representations might come from. How does a robot take a mass of noisy sensor data, integrate it over space and time, and generalize to logical style descriptions such as those in a planning language. This would go a long way toward truly reconciling the approaches. Much like Russell-Norvig's mobile versus manipulator paradigm, planning vs. behaviors emphasizes, to our mind, too many of the differences, and too few of the similarities. Further, it doesn't leave much room for other sorts of robotics (such as engineering) that don't fit neatly into the dichotomy.

The second mini-book on navigation is almost entirely stand-alone, making little direct reference back to the software architectures. Aside from the lost opportunity to talk deeply about configuration space, the split has the effect of leaving the student perplexed about where the navigation portion fits into the various architectures explained in book one.

Mobile Robot Texts

Siegwart and Nourbakhsh's *Autonomous Mobile Robots*† (Siegwart & Nourbakhsh 2004) and Dudek and Jenkin's *Computational Principles of Mobile Robotics* are both solely about mobile robots, and as such, buy in to the Russell-Norvig partition. By the authors' admissions, both books focus on the problem of getting a robot to move through space in support of some other task. This slice of the larger robot problem calls for a less encompassing framework than presented in this paper, and so makes direct comparison difficult.

Dudek and Jenkin, however, do set out a broad ranging theme that "[w]hat sets *mobile robotics* apart from other research areas such as conventional manipulator robotics, artificial intelligence, and computer vision is the emphasis on problems related to the understanding of *large-scale space*."(p. 1) This position appears to be the antithesis to the view presented here, though Dudek and Jenkin may maintain that they are highlighting the differences in order to differentiate their narrow topic from the rest of robotics.

Siegwart and Nourbakhsh make fewer broad thematic claims, but appear to be in fairly strong agreement with Dudek and Jenkin on the scope, breadth, depth, and even order of topics in a mobile robotics textbook. They do place more of an emphasis on the layered nature of robotics, even going so far as to present a diagram (p. 10) nearly identical to figure 11. They do however, restrict information flow to just one particular kind of architecture.

Because of the emphasis in these books on the mobile over the manipulator nature of robots, their robots rarely do anything besides navigate, forgoing traditional AI-style tasks (when they say "planning", they mean almost exclusively "path-planning"). Much like Russell-Norvig, they address intelligent robotics only in the sense that navigation can be considered a task requiring intelligence. Without coverage elsewhere of higher level topics such as task planning, the mobile robots texts implicitly endorse the idea that traditional AI plays no role in current mobile robotics.

†Texts marked with a dagger (†) are ones that we have not used as texts in our robotics class. We have examined the text, and in most cases incorporated some material into the course.

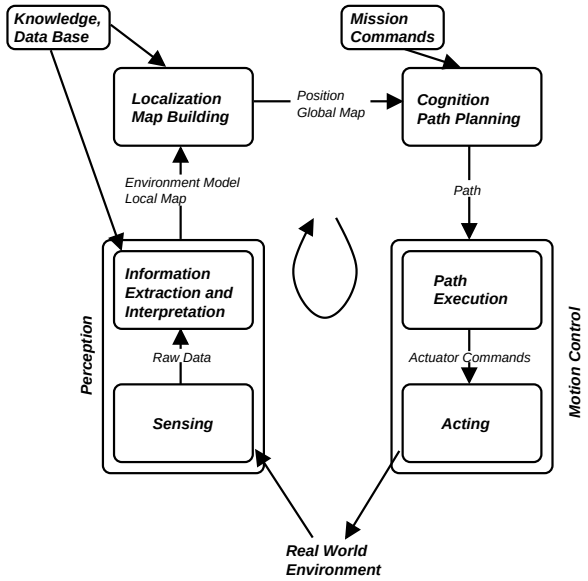

Figure 11: Reference control scheme from Siegwart and Nourbakhsh, (Siegwart & Nourbakhsh 2004).

As an aside, Dudek-Jenkin discusses configuration space with respect to mobile robot pose without recourse to the manipulator model. We wish they wouldn't; this has caused confusion with students who end up believing that configuration space is merely a direct representation of real space, degrading the term and causing confusion among not just students, but also some other textbook authors. Siegwart and Nourbakhsh get this right.

Robot Motion

A different view on some of the same problems faced in mobile robotics is offered by the texts on "robot motion". From this perspective, all robot movement problems are mapped to configuration space, where path planning can be performed. In mobile robots, configuration space is close (if not identical) to real space, whereas in fixed base manipulators, configuration space is the set of positions (configurations) in which the manipulator can reside. A collision-free path in a mobile robot's configuration space takes the mobile robot from one location to another, whereas a collision-free path in a manipulator's configuration space takes the robot from one position to another. Either way, the main problem for robot motion is finding paths through the appropriate configuration space. In this view, outside of differences in hardware and the associated kinematic differences, the primary difference between manipulator robots and mobile robots is the latter's additional problems of localization and mapping. Thus it is not that surprising that mobile robot texts and robot motion texts might share many properties.

Jean-Claude Latombe's classic *Robot Motion Planning*†(Latombe 1991) is still a definitive reference when it comes to moving either a mobile robot or a manipulator. The material is dense, light on motivation, heavy on the analytic geometry, thus is unlikely to be appropriate as an undergraduate text. The format is an introduction of configuration space and how it is constructed, followed by an in-depth survey of how paths can be found in such spaces. The later chapters look at complications on the problem, such as multiple moving obstacles. The book shows its age in the chapter on "uncertainty in action," which approaches the problem something to be engineered around, as opposed to embraced with probabilistic techniques. As with the mobile robot textbooks above, "planning" means "path-planning." Unlike the mobile robot texts, it considers the problem almost entirely in an abstract mathematical space, only occasionally referring to idealized robots and not at all to real-world platforms.

The recent *Principles of Robot Motion*†(Choset *et al.* 2005), team written by seven authors, could easily be viewed as an update of Latombe's text. Like *Robot Motion Planning*, it focuses on the idea of configuration space as a representation for mobile robots and articulated arms alike. It presents many of the same topics as Latombe, such as potential fields, roadmaps, and cell decompositions. Of course, each of these areas has been updated with the new results from the intervening fifteen years. There is also new material based on some more recently developed approaches, especially with respect to probability. Most of this material fits smoothly into the text, such as sampling approaches to roadmaps for path planning. On the other hand, some of the new material is not directly about motion at all, but rather about the problems of sensation specific to mobile robots, such as approaches to simultaneous localization and mapping (SLAM).

Due to the mathematical difficulty and limited scope of both these texts, neither is a good choice for the course under discussion here, but the unifying philosophy is aligned with the spirit of our approach, and the material they contain is an important piece of the larger picture.

Introduction to Robotics. . .

Two books with similar names, *Introduction to Robotics: Mechanics and Control*† by Craig (Craig 2005) and *Introduction to Robotics: Analysis, Systems, Applications*† by Niku (Niku 2001) present the standard view of engineering robotics. In fact, the first few chapters differ only in explanation style, covering the same material in pretty much the same order. Both focus almost exclusively on manipulator arms, covering: kinematics, velocities, dynamics, manipulator design, and control. Niku presents fuzzy logic control while Craig focuses on more traditional linear and non-linear differential equation based control. Neither author pays any attention to the higher level issues that would characterize an AI style course, but are good references for instructors who wish to include coverage of these topics.

Discussion

The idea of abstraction in computing is not new, nor is the bottom up approach to teaching. However, by examining the texts currently available in robotics, it appears that undergraduate robotics is rarely taught in that manner. Textbooks that focus on one robotics sub-field usually ignore all of the other sub-fields. Even books that have broad coverage, often fail to relate the sub-fields to each other. For each major

sub-field in robotics, examining it from the perspective of the layers of abstraction framework provides additional insight.

At the signal layer, Braitenberg architectures are often presented as an interesting but unrelated thought experiment in robotics. Instead, they fit naturally into the framework and make an excellent place to introduce robots.

Behavior-based robots have been traditionally presented as an antidote to and a radical departure from the traditional planning systems. According to this layered framework they are neither unrelated nor antithetical to planning systems; they are a difference of opinion on how much abstraction is necessary to perform various tasks. Furthermore, the framework highlights that hybrids of behavior-based and planning architectures are a natural combination of crossing information from the input to the output channel at multiple layers, thus taking advantage of the time differences at the various layers.

Mobile robotics is commonly presented as its own subfield, with its own kinematics and own high-level issues, primarily navigation. Some authors explicitly present them as orthogonal to manipulators, as if mobility and manipulation is thought of like oil and water: they can be mixed with vigorous shaking, but on the local level, they are distinct. By examining mobile robotics in terms of the layer framework one can see that the system organizational issues for mobile robotics are identical to those of other robotics systems. They can been viewed as requiring slight modifications of details, such as using a variant on the kinematic formulations, but otherwise are the same as stationary articulated arms. In fact, two of the more elegant and interesting results come from this fact: the equivalence of the integral kinematic method for mobile robots with the Denavit and Hartenberg methods; and the equivalence of path planning in manipulator configuration space with path planning in mobile robot physical space.

"Probabilistic robotics" is an imprecise phrase, at its root meaning that there are some probabilities in the system somewhere. In practice, it typically is used to mean probabilities exist in either sensation in the simple model layer for localization and mapping, or accounting for uncertain movements in the path planning problem through the use of partially observable Markov decision processes (Thrun 2000). If architectures are defined by the number of layers through which information is passed and where information crosses channels, then probabilistic robotics are not a particular architecture at all. The use of probability is a specific approach to solving the problem of a particular layer and channel. For instance, Bayesian classifiers can be used in the detection module for behavior-based systems. Further, a probabilistic robot might use any combination of probabilistic modules with non-probabilistic mechanisms in other areas, such as probabilistic localization with a deterministic path planner. From the perspective of the layers of abstraction, the degree to which probabilities are involved in a system is orthogonal to the architecture of the system.

Advocates of probabilistic robotics maintain that it can bestow the fast reaction time and robustness to noise benefits of behavior-based robotics over logic predicate based task-planning robotics, while being more principled than said behavior-based systems. The validity of this claim depends on what systems are being compared. For instance, when comparing a probabilistic path planner to a task planning robot, some of the robustness of the path-planner is due to only abstracting up to the simple-model layer. When path planning is performed in these systems, it is not the same sort of problem being solved as would be by a task-planning robot. Similarly, a system that uses probabilities in the input channel for map making but uses modules that are reactive towards the maps built (what we've called reactive-model architectures) would owe some of their robustness to behavior-based architectures. Yet, the probabilistic path planning system that has no reactive component is just as vulnerable to surprise occurrences as other planning systems.

Because probability is not an architecture, but a technique to be used within many architectures, comparison to individual architectures is not immediately elucidating. On the other hand, the success of probability does suggest a general principle. While more abstraction reduces robustness through the discard of information, the introduction of probabilistic techniques at a particular layer can ameliorate the brittleness of abstraction while maintaining its advantages. It will be interesting to see if researchers will succeed in applying probabilistic techniques to the higher layers of the abstract model, and whether this principle will continue to hold true.

It is common to find discussion of AI planning robotics without any explanation of where the logic representation comes from, or how actions output by the planner end up as motor movements. It is similarly common to find discussions of the use of topographical maps without discussion of their origin. In the layered model, these connections are emphasized at each stage, providing the student with the context to understand these topics.

Many authors like to describe the level of autonomy in a robot system as a spectrum, from fully autonomous to teleoperated. While strictly speaking the level of autonomy is orthogonal to the architecture, it is interesting to note that as autonomy increases, it is nearly always introduced at the lowest layers of the system first. As systems become progressively more autonomous, the designers add autonomy layer by layer from the lower layers on the left to the higher layers on the right. But this need not proceed in both channels, for instance, there can be little or no autonomy in the input channel, with hand-build maps provided by a controller, yet the robot is fully autonomous in the output channel. Similarly, the autonomy could all be in the input channel, where the result of the processing is presented to the controller who then makes decisions and prescribes the robot's actions directly. By describing the level of autonomy in this manner, by showing *where* autonomy is added, it provides a more specific concept than just that of a simple spectrum.

Finally, it is interesting to note that because this frame of reference encompasses all of the standard robotics sub-fields it can act as an evaluation tool of the claims of novel architectures. By examining new proposals from the perspective of this framework, a researcher can determine relations to

existing systems and evaluate uniqueness claims. For example, the layers of abstraction model points out that there appears to be a disconnect between artificial intelligence and mobile robotics as it is currently practiced by the AI community. AI texts place the field of AI in the upper layers of this taxonomy, abstract models, task planning, with some path planning. In contrast, most mobile robotics efforts spend their time building simple models; there is almost no consideration of what the robot does when it gets to its destination. This is not a criticism– after all a robot has to be at the job site before it can perform a job, but rather point where the connection can be made.

Conclusion

This paper presents and argues in favor of a perspective for teaching AI robotics that looks at the problem as a two-channel set of layers of abstraction. This perspective is useful for teaching an undergraduate course that focuses on the broad spectrum of subfields in robotics. It covers all of the major issues while providing a unifying context for what sometimes seem disparate approaches to robotic problems. In addition, the technical focus of these layers of abstraction adds perspective to some of the historical developments that have led AI and Robotics to the stature that they enjoy today. When AI robotics is only a small portion of a much larger AI course, this framework effectively and efficiently connects the topics to the much broader fields on which it depends.

Acknowledgments

We would like to thank Lynn Stein, who by providing a venue for these ideas, first forced us to coalesce them into a coherent system. This paper owes a great debt to editors, especially Zach Dodds and Lloyd Greenwald, for providing guidance, insight, and well turned phrases. We would also like to thank the anonymous reviewers who also made many good suggestions. Rebecca Hwa and Chris Brown were helpful in reading drafts and providing sounding boards. This work was supported in part by a grant from the Office of Naval Research, N0001403WR20241.

References

A K Peters, Ltd. 2005. Rug warrior pro. *http://www.akpeters.com/book.asp?bID=39*. Web page accessed Mar 2005.

Advanced Design Inc. 2005. Robix rascal. *http://www.robix.com/*. Web page accessed Mar 2005.

Arkin, R. 1998. *Behavior-Based Robotics*. Cambridge, MA: MIT Press.

Asada, M. 1990. Map building for a mobile robot from sensory data. *IEEE transactions on Systems, Man, and Cybernetics* 37(6).

Aurenhammer, F. 1991. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys* 23(3):345–405.

Bishop, B., and Wick, C. 2005. Analog computation for mobile robotics education. In *Proceedings of the 2005 American Society for Engineering Education Annual Conference and Exposition*.

Braitenberg, V. 1984. *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press.

Brooks, R. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* RA-2:14–23.

Choset, H., and Burdick, J. 2000. Sensor based motion planning: The hierarchical generalized voronoi graph. *International Journal of Robotics Research* 19(2):96–125.

Choset, H.; Lynch, K. M.; Hutchinson, S.; Kantor, G.; Burgard, W.; Kavraki, L. E.; and Thrun, S. 2005. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press.

Craig, J. 2005. *Introduction to Robotics: Mechanics and Control*. Upper Saddle River, NJ: Perason Prentice Hall.

Denavit, J., and Hartenberg, R. S. 1955. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics* 22:215–221.

Dudek, G., and Jenkin, M. 2000. *Computational Principles of Mobile Robotics*. Cambridge, UK: Cambridge University Press.

Gallistel, C. R. 1980. *The Organization of Action: A New Synthesis*. Hillsdale NJ: Earlbaum.

Gallistel, C. R. 1990. *The Organization of Behavior*. Cambridge, Ma: MIT Press.

Hrynkiw, D., and Tilden, M. 2002. *JunkBots, Bugbots, and Bots on Wheels: Building Simple Robots With BEAM Technology*. New York: McGraw-Hill.

ISO. 1994. *Standard 7498-1*. International Organization for Standardization.

K-Team S.A. 2005. Khepera II robot. *http://www.k-team.com/robots/khepera*. Web page accessed Mar 2005.

Latombe, J. 1991. *Robot Motion Planning*. Norwell, MA: Kluwer Academic Publishers.

Matarić, M. J. 1992. Minimizing complexity in controlling a mobile robot population. In *IEEE International Conference on Robotics and Automation*, 830–835.

Matarić, M. J. 1997. Using communication to reduce locality in multi-robot learning. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 643–648.

McDermott, D. 1992. Robot planning. *AI Magazine* 13(2):55–79.

McKerrow, P. J. 1991. *Introduction to Robotics*. Sydney: Addison Wesley.

Miller, D. P. 2005. Byo-bot. *http://www.kipr.org/robots/byobot.html*. Web page accessed Mar 2005.

Moravec, H. P., and Elfes, A. 1985. High resolution maps from wide angle sonar. In *Proc. IEEE In. Conf. Robotics and Automation*, 116–121.

Moravec, H. P. 1988. Sensor fusion in evidence grids for mobile robots. *AI Magazine* 61–74.

Murphy, R. R. 2000. *Introduction to AI Robotics*. Boston: MIT Press.

Niku, S. 2001. *Introduction to Robotics Analysis, Systems, Applications*. Upper Saddle River, NJ: Prentice Hall.

Pell, B.; Bernard, D. E.; Chien, S. A.; Gat, E.; N.Muscettola; Nayak, P. P.; Wagner, M. D.; and Williams, B. C. 1998. An autonomous spacecraft agent prototype. *Autonomous Robots* 5(1):29–52.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence. A Modern Approach*. Cambridge, MA: MIT Press, second edition.

Sheth, P. N., and Uicker, J. J. 1971. A generalized symbolic notation for mechanisms. *Journal of Engineering for Industry* 93(70-Mech-19).

Siegwart, R., and Nourbakhsh, I. 2004. *Introduction to Autonomous Mobile Robots*. Cambridge, MA: MIT Press.

Solomon, D. A. 1998. *Inside Windows NT*. Microsoft Press, second edition edition.

Tanenbaum, A. S. 1992. *Modern Operating Systems*. Englewood Cliffs, NJ: Prentice Hall.

Thrun, S. 2000. Probabilistic algorithms in robotics. *AI Magazine* 21(4):93–109.

Thrun, S. 2002. Robotic mapping: A survey. In Lakemeyer, G., and Nebel, B., eds., *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.

Yamauchi, B.; Schultz, A.; and Adams, W. 1998. Mobile robot exploration and map-building with continuous localization. In *IEEE International Conference on Robotics and Automation*, 3715–3720.