MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

AI Memo 1148

October, 1989

Battling Reality

Anita M. Flynn and Rodney A. Brooks

Abstract

In the four years that the MIT Mobile Robot Project has been in existence, we have built ten robots that focus research in various areas concerned with building intelligent systems. Towards this end, we have embarked on trying to build useful autonomous creatures that live and work in the real world.

Many of the preconceived notions entertained before we started building our robots turned out to be misguided. Some issues we thought would be hard have worked successfully from day one and subsystems we imagined to be trivial have become tremendous time sinks. Oddly enough, one of our biggest failures has led to some of our favorite successes. This paper describes the changing paths our research has taken due to the lessons learned from the practical realities of building robots.

Support for this research was provided in part by the University Research Initiative under Office of Naval Research contract N00014-86-K-0685, in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014-85-K-0124 and in part by a gift from Siemens Corp. The Robot Talent Show was funded by the Systems Development Foundation.

1 Introduction

The MIT mobots come in a number of shapes and sizes, utilize an assortment of different sensors, take advantage of a few types of locomotion strategies and are implemented in several varieties of computer hardware. Central throughout however, is a common methodology for organizing the sensors, actuators and computational elements to effectively control complexity.

A basic tenet of our research of the last four years is that it is important to build complete systems that exist in the real world with all its noise. This avoids the trap of building abstraction barriers that never quite get bridged, but which provide cracks into which all the hard problems slip. Another key component has been the realization that the hardest part in building robots is not so much in building the intelligent decision system per se, but in *achieving perception*. Extracting useful information from noisy sensors requires extensive effort, and the success of our subsumption architecture framework is primarily due to the manner with which it handles sensor fusion rather than control.

Finally, through *doing it all*, or being forced to write our own languages and compilers, designing our own sensors, building our own computers, even manufacturing our own motors, comes a thorough understanding of how much computation is actually needed for a given level of behavior. The primary result is that if the computation is well understood and the subcomponents of the system are organized correctly, then the actual silicon area needed to implement the control for a mobile robot becomes quite small. This realization leads to some very interesting prospects for one day creating microminiature robots that hold the promise of useful behavior at very low cost.

2 Theme

The general problem we set out to solve four and a half years ago was how to organize the intelligence system of a robot in order to answer the question of what it would take to build something that we would consider clever. What were the essential components that would be needed to create an intelligent entity and how should those components be put together?

The ideas we started with took a route that was different from the traditional thinking in Artificial Intelligence at that time. Namely, our approach emphasized:

- that there would be no traditional notion of planning
- that no central representation of the world was needed
- that notions of world modeling are impractical and unnecessary
- that biology and evolution were good models to follow in our quest
- that we insist on building complete systems that existed in the real world so that we would not trick ourselves into skipping hard problems

To encapsulate these ideas and to address the real-world, real-time issues, we developed a general methodology for the organization of the control system, which we call the subsumption architecture. The subsumption architecture is a parallel and distributed computational formalism for connecting sensors to actuators in robots. The underlying ideas of the architecture and experimental runs have been well documented in the literature **[Brooks 86]**, **[Brooks, Connell and Ning 88]**, **[Brooks 89]**.

The two distinguishing aspects of the subsumption architecture are that (1) it imposes a layering methodology in building intelligent control programs and (2) within each layer, the finite state machines give the layer some structure and provide a repository for state.

A key observation from our experiments has been that using this subsumption-style organization, very small amounts of computation are needed to generate intelligent behaviors.

2.1 The MIT Mobots

A variety of experiments with the subsumption architecture have been performed on a number of mobile robots built in the Mobot Lab. Figure 1 shows the family portrait at this time of writing. The different robot projects focused in on different theoretical issues or experiments with new sensors and hardware.

The first robot, Allen, **[Brooks 86]**, **[Brooks and Connell 86]** was the initial testbed for the subsumption architecture; it simulated the parallel layered processes on an offboard Lisp Machine. Allen had only a ring of simple sonar sensors to achieve behaviors such as obstacle avoidance, wall following and door finding.

After Allen, Herbert was begun to extend the experiments with the subsumption architecture by adding more interesting sensors and actuators **[Brooks, Connell and Ning 88]**, **[Connell 88]**. Herbert has an arm, a

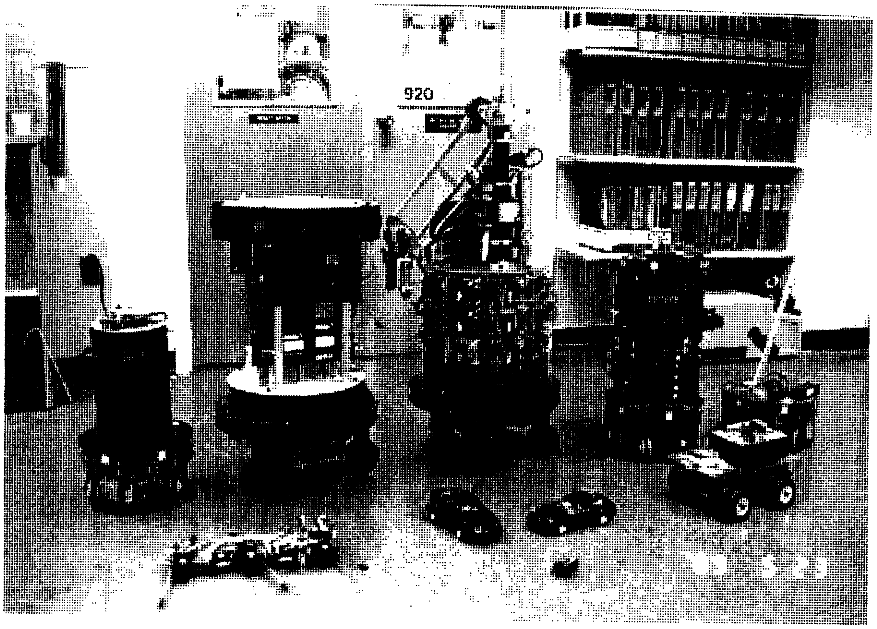

Figure 1: The MIT Mobile Robots include, in the back row, left to right; Toto, Allen, Herbert, Seymour and Tito. In the middle row are Genghis, Tom and Jerry, and Labnav. Squirt, although rather hard to see, is down in front.

light striper vision system for recognition, and infrared proximity sensors for obstacle avoidance. In addition, all computation was moved onboard into a network of simple microprocessors with only a patch panel of physical wires implementing the interconnections, to make an explicit statement showing the lack of central control. Herbert has wandered through a sequence of rooms, found a soda can, picked it up and returned it to the place where it was switched on. The subsumption architecture implementation in this system achieves all those behaviors without retention of any state for more than *three seconds*, and without any communication between processes. The key idea that enables accomplishing useful tasks without recourse to world models is the notion of *communicating through the world* **[Connell 89]**.

The idea here of communicating through the world, is that Herbert's intelligence architecture is organized as a collection of independent agents which compete for the resources of the body. Different behaviors communicate by watching how other behaviors change the environment rather than by receiving control signals from them explicitly. For instance, the vision system can find a soda can (a cylindrical object to be exact) in a general cluttered scene. The arm by itself, is capable of searching and finding a soda can (with infrared sensors and touch switches on the hand) whenever a can is present somewhere in its workspace. However, instead of the vision system explicitly telling the arm to move to coordinates X,Y,Z, what the vision system does is simply direct the base to servo itself such that the soda can is seen in the center of the field of view of the camera, and then stop the base. Similarly, the arm does not wait for explicit control signals from the vision system, but merely monitors the base. Whenever the base is stopped, the arm realizes that other behaviors have set up the world such that a soda can is somewhere within its workspace. This observation of the base being stopped triggers the arm to initiate its manipulation task. By organizing the behaviors in this manner of communicating through the world, the issue of fusing data from many sources becomes pushed back to the level of behavior fusion instead of sensor fusion. That is, communication through the world has circumvented the problem of calibrating the vision system to the arm. By alleviating the need to fuse data from the camera system with data from the arm sensors, the final system becomes much less computationally intensive. This observation has inspired a number of later projects.

While Herbert's development was in progress, Tom and Jerry **[Connell 87]** were built to demonstrate just how little computational hardware was needed to implement a subsumption architecture. Two small cars with only four near-infrared proximity sensors, Tom and Jerry exhibited behaviors roughly equivalent to Allen's, yet the entire brain was compiled down to a single programmable array logic chip. Without any need for sensor fusion or world model representation, the silicon area needed to implement the control system can be very small.

Genghis **[Brooks 89]** is a six legged robot which walks under subsumption control and has, like Herbert, an extremely distributed control system. While the control system is composed of 57 finite state machines, only 5 have any central role over more than one leg. That is, each leg has a notion of how to behave when placed in various circumstances. Walking, balancing and climbing fall out by merely sequencing the lifting of the six legs. The robot successfully walks over rough terrain using 12 motors, 12 force sensors, 6 pyroelectric sensors, one inclinometer and 2 whiskers. Walking behavior falls out from very simple autonomous behaviors assigned to each leg, and as more sensors and more layers of control are added, Genghis can be clearly seen to walk better. The first few layers of finite state machines initiate stand-up and simple walking talents. Incremental layers above those respond to different types of sensors which instigate balancing, turning and people-following behaviors.

Seymour is a new robot we are building now with all onboard processing to support vision processing of 8 to 10 low resolution cameras **[Brooks and Flynn 89]**. All of these cameras will feed into different subsumption layers which will act upon those aspects of the world they perceive. For instance, one layer might use a camera to follow a moving blob **[Horswill and Brooks 88]**. Another might use motion stereo to detect obstacles in the path **[Brooks, Flynn and Marill 87]**. Yet another can locate ceiling edges to determine sizes and shapes of rooms, as has been prototyped on the robot Tito **[Sarachik 89]**. Passive infrared motion sensors are configured to detect and range to humans **[Viggh and Flynn 88]**.

Squirt is the smallest robot we have ever built **[Flynn, Brooks, Wells and Barrett 89a]**. It packs sensors, actuator, computer and power supply in a 1.3 cubic inch volume. Using two microphones and a photoresistor, Squirt (shown in figure 2) is programmed to act as a "bug"; spiraling around searching for dark corners to hide in, venturing out only after hearing loud noises some time after the noises have disappeared. The end effect is that Squirt gravitates towards the center of action (i.e. ends up under the conference table). The entire control system for Squirt fits in 1300 bytes of code.

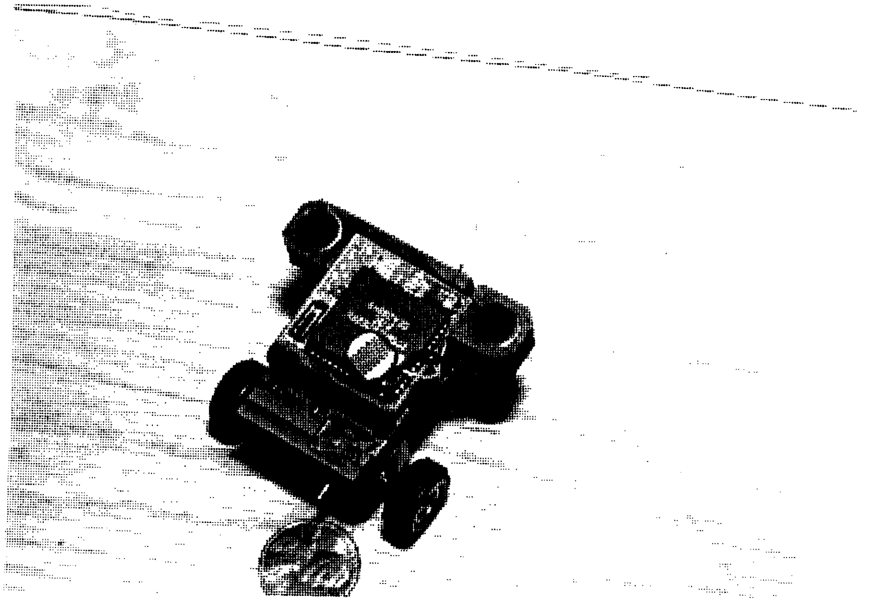

Figure 2: Squirt, the smallest robot we've built to date, packs motor, batteries, microcomputer, interface electronics and three sensors into a volume slightly larger than one cubic inch.

3 Reflections on the Last Four Years

What have we learned in the last four years of building robots? Some lessons have reinforced earlier convictions while others have turned us away from the original paths we went charging down.

3.1 Experiences with the Subsumption Architecture

When we started, we focused our efforts on the design and organization of an intelligent control system, as we felt the bulk of the research would lie in organizing the software. We came up with the subsumption architecture approach and have experimented with it in many ways. We have made a few improvements to it over the years, most notably incorporating the strategy of communicating through the world rather than allowing higher levels to twiddle signals internal to lower levels. There have also been some evolutions in programming styles, such as the new format of making suppressing signals act like monostables rather than as specified delays. The monostable form of suppression implies that the suppressing module repeatedly sends its suppressing packet for as long as it is activated, rather than sending one message specifying that it should be suppressing for some specific amount of time. We've also designed new behavior languages for specifying subsumption programs in a higher level abstraction than finite state machines **[Brooks and Flynn 89]**. Mechanisms in the behavior language allow for experiments in learning by spreading activation signals throughout the network to increase the likelihood that certain behaviors will fire.

However, there have been lots of other issues that have gained equal importance. The driving force behind subsumption was to achieve robustness and distributed control so that the system would not become saturated, but the real problems we ran into involved perception.

Time and time again, for every new robot or for every new behavior that we thought of creating, we had to go back and invent or engineer a sensor for the job, (in the spirit of Nature's matched filters **[Wehner 87]**), so that our control system could have something interesting to process. We have built sonar rings, laser stripers, infrared proximity detectors, tilt sensors, force sensors, whiskers, digital compasses, pyroelectric ranging systems and even home-brewed cameras.

The inseparablity of perception and the intelligence system now seems to be our major thrust in AI. *Sensing* is not *perceiving*. It is one thing for a transducer to convert optical energy into an electrical signal. It is even fine to print out the pattern and show it to a human who will have no trouble understanding the image. However, having a robot extract useful information that lies hidden somewhere in a large array of bits is much harder.

It is interesting to observe the trend towards sensing research in the dozens of papers that our group has written in the last four years. Early papers dealt with subsumption and ideas for control, but the majority of the papers in recent years have focused in on perception problems of various sorts. It is very difficult to instill environmental awareness into a mobile robot in the midst of real world noise. The human system is amazingly good and we take so much of it for granted, that the *a priori* notion in building a robot is that seemingly simple perceptual feats should be trivial. However, we have spent far more time in the last four years battling noise in near-infrared proximity sensors than in writing any piece of subsumption code.

The subsumption architecture has not changed much - and maybe that is because it was a good idea.

3.2 Sensor Fission

Our verdict is that subsumption was a good idea — but not only for the original reasons. While subsumption was successful in the way it dealt with multiple goals, control arbitration and robustness, it was even more successful in the manner in which it dealt with the issue of sensor fusion. Sensor fusion is the notion that if you can combine information from multiple sensors into one model, you can achieve better perceptual acuity [**Flynn 88b**]. The problem with sensor fusion however, is that it is computationally expensive and it is not clear that it is useful in carrying out tasks.

Our experience indicates that the best way to deal with sensor fusion is to ignore it, and this is the way subsumption handles the issue.

Instead of sensor fusion, our control systems use a form of sensor fission where different sensors trigger different behaviors and arbitration is done at the actuator stage rather than the sensor stage. By organizing the intelligence system in this way, such that various sensors inject themselves at various levels of the control system to initiate distinct behaviors, there is no need for maintaining a model of the world or for having to make judgements about which sensor data to believe should there be conflicts. The issue of arbitration takes the form of behavior fusion rather than data fusion. Circumventing all the overhead of sensor fusion and calibration means we can build the brains for our robots with trivial amounts of computational hardware.

3.3 Engineering All the Pieces Led to Insight

The fact that subsumption style control systems can compile down into very lean code has been one of the most pleasant surprises in our research. We did not realize it for a long time though, and in fact, it took one of our greatest failures to bring us around to that point. When we first built Allen, we ran his initial three-layer control system remotely on a lisp machine. We used a serial cable to listen for sonar data and send down base commands. The parallel processes of the subsumption architecture were simulated sequentially on the uniprocessor and there were many layers of lisp machine system software between our robot and his brain. We always planned on replacing his serial tether with a radio link and adding cameras and a high bandwidth TV channel, but it never worked. Undergraduate after undergraduate took a crack at Ham TV, police walkie-talkies and cellular telephones, but we found that the 9th floor of the AI Lab produced far more electrical noise than any mobile robot deserved. We realized we were sinking a lot of time into technology that was not our strength and also was not at the center of our research problem.

Consequently, for all robots after Allen, we put computers onboard. It is possible to squeeze plenty of computation onto small boards these days and furthermore, using CMOS technology, they can run off trivial levels of power. In addition, digital electronics is much easier to deal with than RF or analog circuitry. Computer scientists building computers - that seemed much more reasonable. Of course, building our own computers meant we needed assemblers, debuggers and software tools, but that was perfectly reasonable also. The necessity of targeting subsumption code down to robot-based microprocessors rather than the luxury of leaving it in Lisp and running on a lisp machine forced us to write compilers for the subsumption architecture. In turn, that exercise forced us to completely understand the computations that were undertaken.

It turned out that in Allen's brain, there really was not much happening. From that realization Tom and Jerry were spawned; small cars with only single-chip PALs which produced all the functionality of Allen. In the robots that followed, the subsumption compiler, which was retargetable, was used to program brains made out of 6301s, 68000s, 6811s and programmable logic devices.

In choosing to build mobile robots (from the argument that mobility tended to predominate in evolution in creatures of general intelligence), we were forced to do extensive engineering. We found we could not buy systems of sensors or computers off the shelf very easily. Either they often did not exist, as in the case of sensors, or they were too large and power hungry, as in the case of computers. If we incorporated a MicroVax and accompanying disk drive and fan on our mobile base, we would very likely end up weighing it down and would need larger motors. Larger motors would require more batteries, but more batteries would just increase the weight of the base. That would mean we would need larger motors again. As we looked around at mobile robot projects across the country, we noticed this general trend. It seemed mobile robots were half motors and batteries, and the pieces we were interested in for our research, the sensing and control, fit into a very small volume.

3.4 Connector Nightmares

Mobile robots are half motors and batteries — and the other half is connectors.

We noticed that there was a tremendous amount of overhead in putting the entire robot together. This overhead usually came in the form of interconnections. With motors from one vendor, batteries from another, computers from yet another and sensors geographically dispersed, the ensuing connectors form a Gordian knot of wires. Herbert, for instance, has roughly 500 electrical cabling connectors, 400 mechanical fasteners and over 10,000 Scotchflex insulation displacement points for the prototype wiring between chips. Genghis, only one foot long and weighing just one kilogram, has 72 connectors. In fact, we spent so much time mucking around in catalogs looking for small simple plugs and sockets in order to make modules easy to take apart and debug, that we dubbed robotics just a "Deep Theoretical Study in Connectors".

By insisting on building mobile robots in order to study intelligence, we neatly backed ourselves into the corner of having to *do it all* as nothing off the shelf was ever quite light enough or energy conscious enough. We ended up building our own arms, designing our own bases, inventing new sensors, laying out computer boards and writing custom debuggers, assemblers and compilers. The dive headfirst into engineering took a large investment of time, but looking back we can see that it was well worth the effort as we now have an infrastructure perfectly fitted to our needs and more importantly, we have a clear vision of just what is necessary to build a robot.

Getting frustrated with connectors led to an interesting idea. When the VLSI community first got started, they found that often, the biggest constraint on how much functionality could be put into a chip was the limit imposed by the package and the allowable number of pins. If a chip had too many signals that needed to go off chip and onto another chip, there were serious problems in both packaging and the ensuing overhead of raising the signal levels to drive the pads. The solution, and it turned out to be easier than solving the connector problem, was to combine the functionality of second chip onto the same die as the first, thus omitting any overhead for interconnections.

Why not do the same with a robot?

3.5 Gnat Robots - The New Look and Feel

Just about two years ago, when our group was in the thick of groping our way through Herbert's jungles of connectors, some new research ideas surfaced from the microsensor community concerning putting motors on chips. The idea was to use silicon micromachining techniques to etch freely movable structures onto a silicon chip and to electrify them, creating electrostatic actuators **[Bart, Lober, Howe, Lang and Schlecht 88]**. Although they would not be electromagnetic and would be no larger than the diameter of a human hair, they filled the bill for some of the problems we were wrestling with.

The whole escalating problem of large motors requiring larger batteries in turn necessitating larger motors and so on, could be eliminated if we went in the reverse direction. By scaling down and using smaller motors which could get by on smaller power supplies, we could gain a tremendous advantage. As most of the componentry we are interested in for our research can fit on a small silicon chip anyway, why not scale everything down and integrate an entire robot on a chip? The real advantage of putting everything (motors, batteries, sensors and intelligence) on one chip though, is that now we eliminate connectors!

Self-contained, completely autonomous chip robots (we call them gnat robots) give us a completely new image for robotics **[Flynn 87]**, **[Flynn 88a]**, **[Brooks 87]**. At first you might ask what a motor with fractions of milliwatts of power would be good for, but if the only requirement is to push the chip on which it is built, then there might be something worthwhile there. As many mobile robots are used solely to collect information, such as security robots or planetary explorers, there really is no advantage to lugging around extra bulk. A gnat robot would be perfectly sufficient since software and data (as compared to the motors and batteries on large robots) take up almost no space.

Such a new look and feel for robotics of the future requires new perspectives on how we solve problems. One good model to keep in mind is to think of parallel processing computers vs. traditional sequential uniprocessors. Programming an algorithm on a parallel computer requires standing on your head and thinking sideways in comparison to traditional ways of implementing algorithms, but if the algorithm solution is well matched to the structure of the parallel computer, then there can be tremendous advantages in speed. Lots and lots of very simple processors can work together to outperform a goliath uniprocessor.

Similarly, we can match gnat robots to many real world tasks and solve problems in better, albeit different, ways. By using many simple robots in place of one large complicated expensive robot, we can do work in the environment at a fraction of the cost. By integrating an entire robot onto a chip, we can essentially print robots by the millions, just as we print integrated circuits, and take advantage of the drastic reductions in cost per level of performance that integrated circuits have demonstrated over the past twenty years.

[Flynn, Brooks and Tavrow 89] provides detailed proposals for avenues around some of the technology barriers and gives examples of applications for gnat robots, from mowing the lawn one blade at a time with a yardful of miniature gnat robots to machines that get into hard to reach places and do useful work, such as cleaning lenses of space telescopes. Many of the ideas for thinking about how to solve problems with societies of gnat robots come from ideas in earlier work of multiple agents within one brain communicating through the world to compete for the resources of the robot's body **[Connell 89]**. One shouldn't think of swarms of gnat robots as machines which are told what to do, but rather as autonomous creatures that when turned on, do what is in their programmed nature to do, in the spirit of **[Brooks and Flynn 89]**.

3.6 Squirt—The Bare Essentials

Although the idea of building robots on a chip was primarily driven by economics and the thought that robots could be much more useful to society if they cost less per unit level of talent than they do today, it turns out that we can reap tremendous cost savings now using strictly off the shelf technology. We decided to build a robot using the smallest off the shelf motors and batteries we could find. The aim was to build the world's smallest robot. Squirt was the end product of our attempt at building a one cubic inch robot, although he ended up slightly larger.

The interesting result from Squirt was that it showed what was the minimal amount of hardware needed for building a robot for doing research in connecting perception to action. The spinoff was that we realized that we could multiply our research machines and provide graduate students with just enough hardware to allow them to make the connection, allowing them to focus on developing algorithms for intelligence (without recourse to simulation) without undue cost, overhead or complexity. Although Squirt was a stepping stone exercise towards gnat robots - with the long term goal of providing robotic technology cost-effectively, in turned out that even at the macroscopic dimensions of Squirt, robots became inexpensive.

3.7 The Robot Talent Show

The fallout of this realization was a Robot Talent Show which the AI Lab held in January, 1989 **[Flynn, Brooks, Wells and Barrett 89b]**. Not only could each graduate student in the Mobot Group have their own robot to work on, but now every graduate student in the AI Lab could have their own mechanism to connect perception to action. The catch was that everybody had to build their own.

We gave out kits of parts (sensors, motors, connectors, batteries, wire and prototyping equipment) and made up printed circuit boards with Squirt-style computers and distributed a robot building manual **[Flynn et al 88]**. We ported the subsumption architecture programming environment to general lab workstations and participants could work in their offices developing algorithms and downloading code to their creatures. **[Flynn 89]** gives a pictorial account of the final outcome. A few of the robots are shown in figure 3. Over 60 people participated and the resulting robots came in a wide assortment and showed reams of creativity and ingenuity. There were autonomous blimps, tanks that chased toys cars, robots that learned to run away, and cars that played laser tag. The Robot Talent Show gave students a hands-on feel for what the hard problems are in designing intelligent creatures.

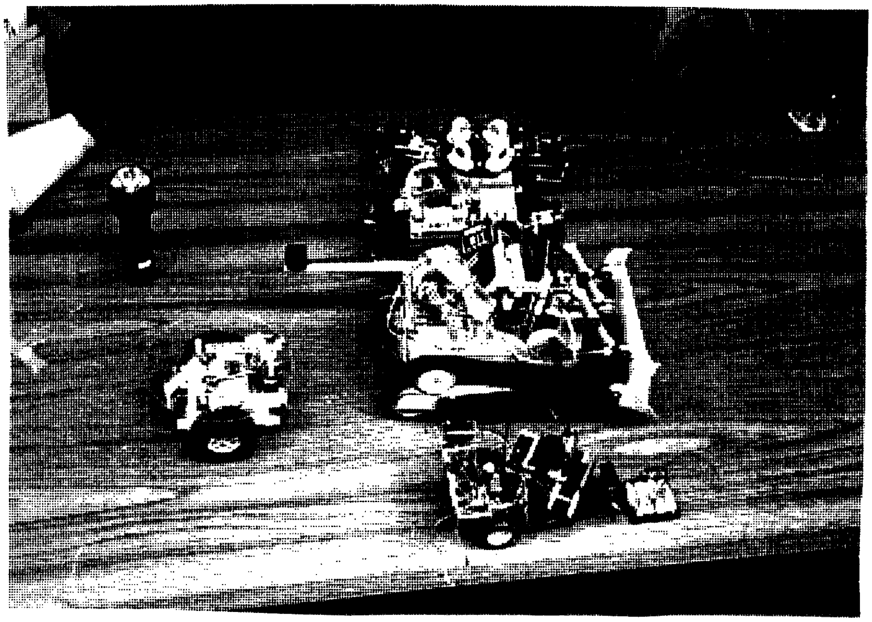

Figure 3: A few of the robots that appeared in the AI Lab's Robot Talent Show are shown here: (left) an omnidirectional vacuum cleaner; (top to bottom) a laser tag competition entry, a tank that chased sports cars, a bulldozer that scooped up small moving toy animals and an obstacle avoiding car with onboard video camera.

4 Revelations in Building Things

Building things has forced us to stay on track and to keep focused on solving the problems that need to be addressed in creating entities which we would consider clever. Before one builds something, one can ascribe all sorts of complex structures to the thought processes in one's robot, and one can hypothesize complicated networks, special architectures, the need for lots of memory and all sorts of computrons to connect perception to action. Sometimes though, one can be pretty far off in some of those preconceptions.

Unless you design, build, experiment and test in the real world in a tight loop, you can spend a lot of time on the wrong problems, such as simulating complex control schemes while the real hard problems of perception are abstracted away and ignored.

One of the great advantages to building things from scratch instead of reverse engineering someone else's work, is that you can see exactly how much machinery is required. Oftentimes, the *a priori* intuition about what will be needed is completely wrong. Then you can look at an already-built system (such as Nature's) with greater insight and an ability to discern the extraneous from the essential.

4.1 Tongue in Cheek

Before we built Squirt, we did some experiments with different designs for a locomotion mechanism. First, we built the prototype base, Squirt Jr., shown in figure 4, which consisted of two wheels, two motors and a battery. Each motor drove one wheel directly and the batteries were connected straight to the motors without any intermediate circuitry except a power switch.

We now have a running joke in our lab, because to our surprise, when we first powered up this base and long before we added the processor or sensors, Squirt Jr. seemed to have a mind of his own and acted as if he had a number of subsumption-style behaviors already compiled in.

At his lowest level, he displays a random wander behavior, as one wheel invariably slightly drags and causes the vehicle to turn. However, when higher levels are triggered, the random walk is subsumed. For instance, whenever it hits an obstacle, it stops!

Upon closer inspection however, it is easy to see how the system works. Obviously, it has a two-state transmission, as when in the *wedged-state* (and one wheel is stuck) it rotates about that wheel. In the *unwedged-state*, it goes straight forward. The most sophisticated subsystem onboard Squirt Jr.

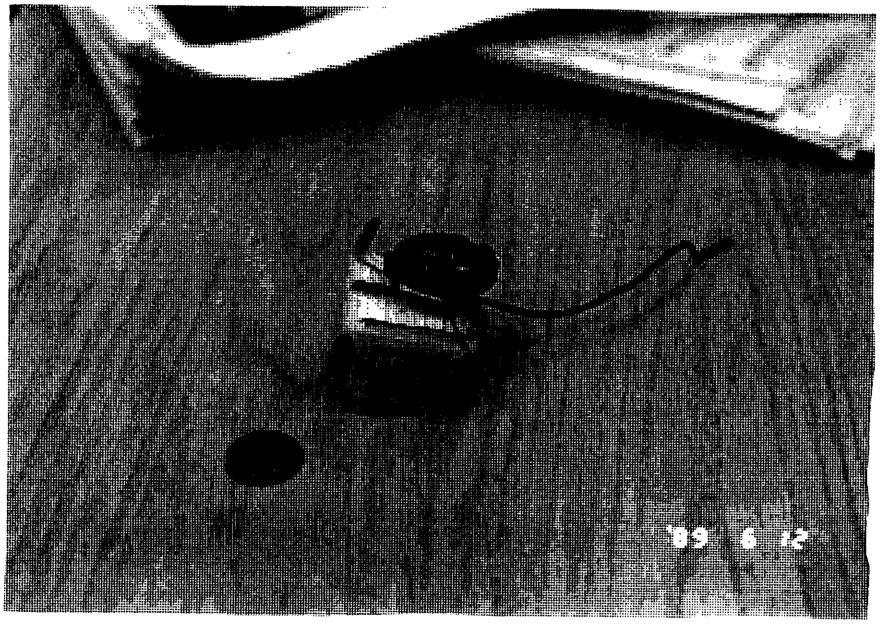
'89 6 12

Figure 4: Squirt's prototype had no control system but still seemed intelligent to näive observers.

however, is his visual apparatus. A higher level behavior utilizes an edge finder so that whenever Squirt Jr. sees the "edge" of a table, he falls off!

The behavior is extremely robust and works 100% of the time. He's obviously been carefully programmed to act according to the laws of physics. Note that all this computation has been compiled down into a *very* small amount of silicon.

4.2 The Lesson

The lesson to be learned from Squirt Jr. is that building things can give insight into how things are built, and that our hypothesized models, while attempting reverse engineering, may attribute all sorts of complex structures to systems that are really only very simple. We often say that artificial intelligence is the study of the computations that connect perception to action, but here is a case in which *no* computations are being performed. Actually, maybe we should say the computations are done mechanically.

4.3 Let The Physics Do The Walking

Mechanical logic may be utilized far more often in Nature than we would at first like to admit. In fact, mechanical logic may be used far more in our own robots than we realize. When we built Genghis, we had some higher level behaviors that monitored forces on the motors and we programmed in compliance, or force balancing, in an effort to keep the robot level. For instance, suppose there was no programmed compliance; if the robot was in a standup position such that three legs on one side were on the ground and the middle leg on the other side was on a rock, then the other two legs on that side would be hanging in the air. However, since the motor on the middle leg is not infinitely strong, it sags a little bit, and the front and back legs are then really not as high up in the air as they might be. Programmed-in compliance makes the middle motor sag even more until the front and back legs reach the ground. For small rocks however, the behavior with and without the programmed compliance is indistinguishable because the mechanical compliance does enough. Explicit software we originally envisioned to be essential was unnecessary.

Genghis provides a further lesson of physics in action. One of the main reasons he works at all is because he is small. If he was a large robot and he put his foot in a crack and then torqued his body over, he would break. Larger walking machines usually get around this problem by carefully scanning the surface and thinking about where to put their feet. However, Genghis just scrabbles and makes forward progress more through persistance than any explicit mechanism. He doesn't need to build models of the surface over which he walks and he doesn't think about trying to put his last foot on the same free spot an earlier foot was placed. Instead, Genghis happens to be a creature of circumstance. Because as creatures are scaled down and volume scales as the cube, whereas surface area scales as the square, smaller creatures become relatively stronger and can get by with sticking their feet in cracks. They think nothing of it.

5 Conclusion

One real robot is worth a thousand simulated robots. Building robots that are situated in the world crystalizes the hard issues. We have found that perception is the key problem while intelligent control is relatively easy in comparison.

The essence of good research is figuring out the right problem to work on. This is especially true in mobile robotics where it is ever so easy to mis-hypothesize the dynamics of existence in various scenarios and to mis-introspect our own internal workings. Understanding the environment and truly discovering the constraints on cognition are more readily done by building one robot than by thinking grand thoughts for a long time. Real sensors deliver a different image of reality than the highly processed impressions from our own sensors.

6 References

[Bart, Lober, Howe, Lang and Schlect 88] "Design Considerations for Microfabricated Electric Actuators", Steven F. Bart, Teresa A. Lober, Roger T. Howe, Jeffrey H. Lang and Martin F. Schlect, *Sensors and Actuators*, 14(3):269-292, July.

[Brooks 86] "A Robust Layered Control System for a Mobile Robot", Rodney A. Brooks, *IEEE Journal of Robotics and Automation*, RA-2, April, 14–23.

[Brooks 87] "Micro-brain for Micro-brawn: Autonomous Microbots", Rodney A. Brooks, *IEEE Micro Robots and Teleoperators Workshop*, Hyannis, MA, November.

[Brooks 89] "A Robot that Walks: Emergent Behavior from a Carefully Evolved Network", Rodney A. Brooks, *Neural Computation*, 1:2.

[Brooks and Connell 86] "Asynchronous Distributed Control System for a Mobile Robot", Rodney A. Brooks and Jonathan H. Connell, *SPIE Vol. 727 Mobile Robots*, Cambridge, MA, November, 77–84.

[Brooks, Connell, and Ning 88] "Herbert: A Second Generation Mobile Robot", Rodney A. Brooks, Jonathan H. Connell and Peter Ning, *MIT AI Memo 1016*, January.

[Brooks and Flynn 89] "A Robot Being", Rodney A. Brooks and Anita M. Flynn, *NATO Workshop on Robotics and Biological Systems*, Tuscany, Italy, June.

[Brooks, Flynn and Marill 87] "Self Calibration of Motion and Stereo for Mobile Robot Navigation", Rodney A. Brooks, Anita M. Flynn, and Thomas Marill, *MIT AI Memo 984*, August.

[Connell 87] "Creature Building with the Subsumption Architecture", Jonathan H. Connell, *IJCAI-87*, Milan, August, 1124–1126.

[Connell 88] "A Behavior-Based Arm Controller", Jonathan H. Connell, *MIT AI Memo 1025*, June.

[Connell 89] "A Colony Architecture for an Artificial Creature", Jonathan H. Connell, *MIT PhD Thesis in Electrical Engineering and Computer Science*, September.

[Flynn 87] "Gnat Robots (And How They Will Change Robotics)", Anita M. Flynn, *IEEE Micro Robots and Teleoperators Workshop*, Hyannis, MA, November.

[Flynn 88a] "Gnat Robots: A Low-Intelligence, Low-Cost Approach", Anita M. Flynn, *IEEE Solid-State Sensor and Actuator Workshop*, Hilton Head, SC, June.

[Flynn 88b] "Combining Sonar and Infrared Sensors for Mobile Robot

Navigation", Anita M. Flynn, *International Journal of Robotics Research*, December.

[Flynn, Brooks, Wells and Barrett 89a] "The World's Largest One Cubic Inch Robot", Anita M. Flynn, Rodney A. Brooks, William M. Wells and David S. Barrett, *Proceedings IEEE Micro Electro Mechanical Systems*, Salt Lake City, Utah, February, 98–101.

[Flynn, Brooks, Wells and Barrett 89b] "Squirt: The Prototypical Mobile Robot for Autonomous Graduate Students", Anita M. Flynn, Rodney A. Brooks, William M. Wells and David S. Barrett, *MIT AI Memo 1120*, July.

[Flynn, Brooks and Tavrow 89] "Twilight Zones and Cornerstones: A Gnat Robot Double Feature", Anita M. Flynn, Rodney A. Brooks and Lee S. Tavrow, *MIT AI Memo 1126*, July.

[Flynn et al 88] "The Olympic Robot Building Manual", Edited by Anita Flynn, with contributions from Colin Angle, Rodney Brooks, Jon Connell, Ian Horswill, Maja Mataric, Henry Minsky, Peter Ning, Paul Viola and William Wells, *MIT AI Lab Manual*, December.

[Flynn 89] "The Official Photograph Album of the 1989 Robot Olympics", Edited by Anita M. Flynn, *MIT AI Lab Manual*, April.

[Horswill and Brooks 88] "Situated Vision in a Dynamic World: Chasing Objects", Ian D. Horswill and Rodney A. Brooks, *AAAI-88*, St. Paul, MN, August, 796–800.

[Sarachik 89] "Characterising an Indoor Environment with a Mobile Robot and Uncalibrated Stereo", Karen B. Sarachik, *Proceedings IEEE Robotics and Automation*, Scottsdale, AZ, May, 984–989.

[Viggh and Flynn 88] "Infrared People Sensors for Mobile Robots", Herbert E. M. Viggh and Anita M. Flynn, *SPIE Conference on Mobile Robots*, Cambridge, MA. November.

[Wehner 87] "'Matched Filters' - Neural Models of the External World",

Rüdiger Wehner, *Journal of Comparative Physiology A 161*, 511–531.