Editor: Richard Doyle
Jet Propulsion Lab
rdoyle@jpl.nasa.gov

Creating Autonomous Roboticists

David P. Miller and Cathryne Stein, KISS Institute for Practical Robotics

KISS Institute's Botball Program is a national program, designed for middle and high school students, that uses the activities of robot building and programming to engage students in science, technology, engineering, and math. Botball puts robotics equipment into schools, then trains students and teachers in that equipment's use. The program is organized into Spring Regional Tournaments and a National Botball Tournament held at the Annual National Conference on Artificial Intelligence, most recently in Austin, Texas, last August.

How we got here

Before 1985, "intelligent" robots were big and slow. The sense-plan-act architecture then in vogue with most AI practitioners not only put high computational demands on the simplest of acts, but required a set of programming skills that only a few could master.

Editor's Perspective

At last year's event at the National Conference on Artificial Intelligence in Austin, Texas (AAAI-2000), I was privileged to be a judge for the AAAI-sponsored Botball competition among high schools from across the country. Everyone who witnessed this event was caught up in the sheer energy, the can-do attitude of the students, the engineering and programming brilliance of their inventions, and the mix of healthy rivalry and team spirit that defined the scene. No doubt to the consternation of the vendors in the exhibition hall, the Botball event by far drew the largest crowd. But I'm sure the vendor representatives who wandered over got caught up in the excitement too.

The inspiration I felt is very much at the heart of what the KISS Institute is aiming to achieve with their Botball education program. This program is playing an important role in creating and inspiring the next generation of AI and robotics technologists, many of whom might very well be the engineers and computer scientists who design and build sophisticated, highly autonomous rovers and other space platforms for future NASA missions of exploration. The inspiration cuts both ways.

—Richard Doyle

After 1985, everything changed. Several new control architectures emerged, the best known being Rodney Brooks' subsumption architecture.[1] Subsumption's key concept was that sensing, combined with small control loops operating in parallel, could operate faster, would demand less computational resources, and was closer to biological models of embodiment than was the largely sequential and deliberative sense-plan-act model.

Other architectures, inspired in part by subsumption, carried these ideas further, creating new tools and examples to ease the process of creating capable robots. This work eventually led to the control system of the Sojourner rover on the Mars Pathfinder mission[2] and the Remote Agent technology experiment on the Deep Space One mission,[3] but there were other spin-offs as well.

In 1988, Brooks and others at the MIT AI Lab put together a robot-building manual and a set of boards built around the Motorola 68HC11 CPU.[1] The AI Lab had a robot festival where many of the faculty and graduate students built robots. It was perhaps the largest collection of autonomous robots yet assembled—but not for long.

Years earlier, Woody Flowers, a mechanical engineering professor, had been running the famous 2.270 undergraduate design course at MIT. In this course, students received a bag of junk and had to design a remotely controlled device to play a game against the other devices created by their classmates. It was an extremely popular course and has been exported to other universities. There is even a version of this design competition at JPL, sponsored by the Mechanical Systems Division. Several computer science and electrical engineering students saw the AI Lab's robot manual as the key for creating their own version of Flowers' design class.

The course 6.270 was formed in the late '80s as the MIT EE/CS Department's version of 2.270. In this student-taught class, given during the MIT Winter intercession, students receive a bag of specific chips, circuit boards, sensors, wire, a soldering iron, and some LEGOS. They also get a game specification and must create an autonomous robot to play the game against their classmates.

After a few more years of improving the tools, this idea was ready for the next step. Botball continues the trend of

Figure 1. Botball kits include two microprocessors, about 20 sensors, 10 motors, and almost 2,000 LEGO pieces.

more sophisticated equipment and tasks being used by younger and less experienced students. In Botball, middle and high school students create teams of autonomous robots to play against other teams from other schools.

The Botball program

So what exactly is Botball? The Botball Program (www.botball.org) challenges students to design, build, and program (in C) a team of small mobile robots to compete in a highly charged double-elimination tournament, pitting robot against robot in a game of skill, speed, and strategy. Students work in a team, each team starting with a specialized kit containing microcontrollers, motors, sensors, lots of LEGO, and plenty of other goodies (see Figure 1). All the robots are autonomous. There is no communication except for the occasional byte of information passed between robots.

Each Regional Botball Program kicks off with a hands-on tutorial for the lead teachers, where they receive their team's official robot kit. After the tutorial, teachers take the kit back to their teams, and the students have approximately six weeks to design, build, and program the robot (see Figure 2). (Considerable reconstructing, deconstructing, debugging, rewriting, and arguing with the hardware people is usually a major part of the agenda as well.)

In addition to the robot building, students also participate in a Web site competition as part of the Botball program. They receive a specific robotics challenge and must then use the Internet to do research, design a solution, and create a Website—using whatever tools they wish—detailing their solution to the assignment. These challenges tend to be space-related, such as "Design a robot to sample the ice from the lunar poles" or this year's "Design a robot to visit an Apollo landing site." Students must consider such factors as how to power the robot, how long it will operate, where to land, how to protect the site, and so forth.

At the end of six weeks, students bring their robots—along with family, friends, teachers, cheering squads, and assorted hangers-on—to the Regional Botball Tournament, which features seeding rounds and a double elimination competition.

For Botball, we stress that all robots must be student-built, and to help ensure that requirement we do not allow any adults in the pit areas, except tournament officials (see Figure 3). After the Botball Tournament, we host an informal celebration, featuring awards for top robots in several categories, as well as for the Website competition.

Educational goals

There are several educational goals of the Botball program. Four of the most important are:

- teaching basic engineering principles,
- teaching team leadership and participation skills,
- applying math to robotics, and
- promoting awareness and teaching basic skills in computer programming.

Teaching basic engineering principles

In the Botball program, students learn by doing. However, simply giving students a problem and expecting them to solve it when they are lacking the tools will teach them nothing but frustration.

In engineering, as in many other professions, we learn best by experience—primarily the experience of others who have come before. Engineering is in many ways the ultimate example of case-based reasoning. It is a discipline that is not only well illustrated but best taught by example. This is especially true with robotics. Robots are a conglomeration of many types of basic systems: mechanical, electrical, control, engineering,

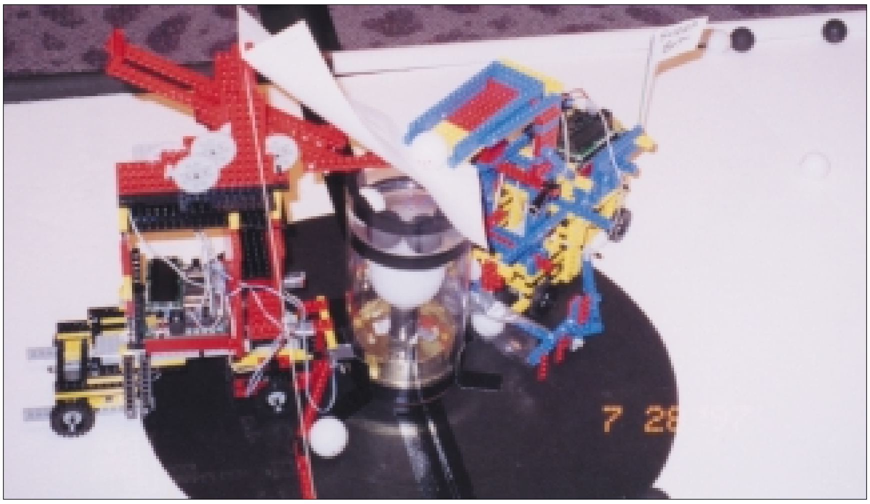


Figure 2. Botball robots can be designed with devious goal-blocking devices.

Botball this year

The public is invited to attend this year's National Botball Tournament, which will be held in conjunction with the 17th International Joint Conference on Artificial Intelligence in Seattle, at the Washington State Convention and Trade Center on 7 and 8 August. This year's Botball program is supported in part by NASA Robotics Education Program, AT&T, and LEGO Mindstorms.

Current Botball Regional Tournaments are held in Silicon Valley, Southern California, Boston, Dallas, Oklahoma, Florida, Pittsburgh, Indiana, and the Greater DC region. Every year new regions join. If you are interested in bringing Botball to your community, please contact KIPR as soon as possible at botball@kipr.org.

and computer science. Well-designed robots also have some human factors and aesthetic engineering in them as well.

When we give a Botball teachers' tutorial, we present the participants with one solution to that year's challenge. We spend considerable time developing this solution because the subsystems should provide useful examples, but the teachers should not solve the problem for the students. We make sure everyone realizes that we have chosen a nonoptimal solution as an example to avoid having teams simply duplicate our design.

We also go through several exercises to create modules and subsystems that could prove useful in their solution, leaving the participants with the skills and experience to quickly create and integrate the necessary pieces that will let them build a robot to solve the challenge. Usually, they find that they must adapt some of the modules we gave them into wholly new structures—adding to their corpus of engineering solutions.

Teaching team leadership and participation skills

Most of a student's academic career is an individual effort. Yet most research, engineering, government, and industrial projects are team efforts. Botball gives students a team experience in an academic setting (as opposed to athletics, which is often the only team-effort exposure students get).

Botball simply has too much to do in too short a time for one person to do it all. This is by design. Students learn to work together or learn the consequences of noncooperation. Both are equally important, though the former is more pleasant.

As part of the teacher training that kicks off the Botball program, the participating teachers spend significant time on team organization. There is no one correct way to do this, but teachers who have participated in previous years discuss what has worked and what has not. They also discuss promising ideas from other regions, and review management techniques such as DeBono Hats and brainstorming.

As a teaming exercise, Botball works. We have received numerous reports from teachers about students who finally realized that other students' contributions to the project were not only valid, but necessary for the group to succeed. The hard deadlines and diverse and extensive work to be done really drives home the value of teamwork and cooperation.

Figure 3. Matt Dobbs and Carol Sonleitner, student technicians at KISS Institute, put finishing touches on Firefly Catcher, a demo-bot that can "catch" a light.

Applying math to robotics

It is probably fair to say that many students have no idea why they are learning math other than the seemingly arbitrary requirements of their school's system. Botball gives students a chance to use mathematics at a variety of levels and in a variety of ways.

All teams will use basic arithmetic in their programs to control their robots. Many teams will also have their robot perform ded-reckoning. To do this, the students must make use of plane geometry and general problem-solving skills. Many teams will also create proportional control loops as part of their robot control. With their exposure to closed-loop control, not only will they be exposed to algebra and limits, but they'll also get some real-life exposure to Xeno's paradox as their robot slows and slows and never quite gets there. Ideally, they'll also figure out how to program around this problem.

More ambitious teams might try to define the kinematics of the manipulators on their robot. This will require solving simultaneous equations and opens the door to linear and matrix algebra.

Those who want to refine their control loops might get into PID control and thus numerical solutions to simple derivatives and integrals. More important than the effect a PID loop will have on their robot's performance is the insight that the students get on what derivatives and integrals really are. Much of the mystery of calculus goes away (and the appreciation for its elegance grows) when a student sees how to approximate and integral through a summation loop.

Mathematics is greatly underappreciated, probably because many students feel that it appears to have no practical use. Botball gives students a chance to use math in an area where they are personally invested—where their knowledge of math can directly help in their team's performance. The examples provided with the Botball kit are in no way designed to replace any part of a traditional math education. Rather, Botball helps to motivate students to use what they already know, in terms they can better appreciate.

Promoting awareness and teaching basic skills in computer programming

While most elementary and high schools now offer computer courses, these courses usually do not teach computer science or computer programming. Instead, they concentrate on basic skills such as keyboarding and on using standard (or what some people would like to assume are standard) applications such as Excel or PhotoShop. Some students come out of these courses believing that they know all there is to know about using computers.

Botball requires students to do some real computer programming. Botball robots are autonomous. A robot that has not been programmed will never leave the starting box. To assist students in what is often their first exposure to computer programming, the kit includes the Interactive C programming environment. This software lets the programmer execute individual C statements, query and set the values of global variables, and incrementally load files and libraries.

Teams get a variety of programming examples, libraries, and tutorials similar to what is provided for the mechanical aspects. Examples show how to use the different types of sensors, motors, servos, and other I/O devices on the robots. A (very) nonoptimal program that plays the Botball game is also provided as a starting point.

Most people have difficulty facing a blank page. We provide templates and examples as a way for them to edit rather than create from scratch. Not only is this easier for most people, but it instills the idea that software, like hardware, is not created instantly from a whole cloth, but instead is an incremental process of improvement and testing.

Tools

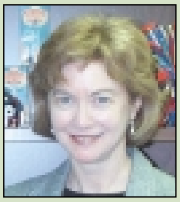
Botball robots are programmed using a multithreaded, interpreted version of C called Interactive C. The students receive a wide variety of library tools for doing such varied tasks as playing musical notes on the robot's speaker to sending infrared packets from one robot to another. The multithreading lets several feedback loops execute seemingly in parallel. The subsumption architecture is straightforward to implement using these tools, and many teams use a variation on subsumption for programming their robots.

Botball has eliminated the need for craftsmanship but maintained the requirement for the students to perform problem solving, design, and aesthetics. LEGO eliminates the need for a machine shop. All the sensors and motors are outfitted with plug-and-play connectors and drivers, so there's no wiring to do nor any assembly language to write. Nevertheless, more than sufficient challenge remains.

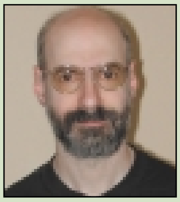
Robotics is one of the great integrating disciplines. Mechanical, electrical, and computer engineering all come into play in obvious ways, as do physics and mathematics. Control theories such as behavior control and cooperating robots use biological systems as their inspiration. Robotics in manufacturing has huge economic and social implications with a potential that is comparable to the advent of the Industrial Revolution. And, as robots become more common, the aesthetics of the devices, along with the natural-language interfaces, will need dramatic improvement. So, robotics touches on almost every technical discipline.

Additionally, intelligent robots and space are inextricably linked. The space environment is one of the few applications of robotics where autonomy is not desired—it is required. Space is also an incredible educational motivator. Although simpler than most Botball robots in both its mechanical capabilities and its software sophistication, Sojourner is a larger inspiration to Botball students than Deep Blue or Shakey ever will be. And so, today's Botball students might build and program the future's rovers and deep space probes. ■

The Authors



Cathryne Stein is a cofounder and current director of KISS Institute for Practical Robotics. Under her direction, KISS Institute has developed several educational programs, including KISS Institute's Botball and Robots in Residence Programs, through which several thousand students have participated in some form of robotics activity. She enjoys teaching robotics to young students, and has a particular interest in the factors surrounding girls and technology. Contact her at KISS Inst. for Practical Robotics, 1818 W. Lindsey Dr., Norman, OK 73069; cstein@kipr.org; www.kipr.org.



David P. Miller is the technical director of KISS Institute for Practical Robotics, as well as the Wilkonson Professor for Intelligent Systems in the Mechanical Engineering Department at the University of Oklahoma. He received his PhD in computer science from Yale and since that time has taught computer science at Virginia Tech, helped to develop Mars Rover technology at the Jet Propulsion Laboratory, and helped found IS Robotics as well as KISS Institute for Practical Robotics. He has worked with dozens of robots and thousands of students, teaching courses for kindergartners through PhDs. Contact him at KISS Inst. for Practical Robotics, 1818 W. Lindsey Dr., Norman, OK 73069; dmiller@kipr.org; www.kipr.org.

References

1. R.A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE J. Robotics and Automation*, vol. 2, no. 1, Mar. 1986, pp. 14–23.
2. D.P. Miller et al., "Reactive Navigation Through Rough Terrain: Experimental Results," *Proc. Nat'l Conf. Artificial Intelligence*, AAAI Press, Menlo Park, Calif., 1992, pp. 823–828.
3. B. Pell et al., "An Autonomous Spacecraft Agent Prototype," *Proc. First Ann. Workshop on Intelligent Agents*, AAAI Press, Menlo Park, Calif., 1997.
4. *The Olympic Robot Building Manual*, A. Flynn et al., eds., MIT AI Lab. Memo 1230, MIT, Cambridge, Mass., Dec. 1988.