



1 **The Robotic Autonomy Mobile Robotics Course: *Robot Design, Curriculum***
2 ***Design and Educational Assessment***

3 ILLAH R. NOURBAKHS

4 *The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA*

5 KEVIN CROWLEY

6 *Learning Research and Development Ctr, Univ. of Pittsburgh, Pittsburgh, PA 15260, USA*

7 AJINKYA BHAVE AND EMILY HAMNER

8 *The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA*

9 THOMAS HSIU

10 *Gogoco, LLC, Sunnyvale, CA 94086, USA*

11 ANDRES PEREZ-BERGQUIST

12 *The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA*

13 STEVE RICHARDS

14 *Acroname, Inc. Boulder, CO 80303, USA*

15 KATIE WILKINSON

16 *Learning Research and Development Ctr, Univ. of Pittsburgh, Pittsburgh, PA 15260, USA*

17 **Abstract.** Robotic Autonomy is a seven-week, hands-on introduction to robotics designed for high school students.
18 The course presents a broad survey of robotics, beginning with mechanism and electronics and ending with robot
19 behavior, navigation and remote teleoperation. During the summer of 2002, Robotic Autonomy was taught to twenty
20 eight students at Carnegie Mellon West in cooperation with NASA/Ames (Moffett Field, CA). The educational
21 robot and course curriculum were the result of a ground-up design effort chartered to develop an effective and
22 low-cost robot for secondary level education and home use. Cooperation between Carnegie Mellon's Robotics
23 Institute, Gogoco, LLC. and Acroname Inc. yielded notable innovations including a fast-build robot construction
24 kit, indoor/outdoor terrainability, CMOS vision-centered sensing, back-EMF motor speed control and a Java-based
25 robot programming interface. In conjunction with robot and curriculum design, the authors at the Robotics Institute
26 and the University of Pittsburgh's Learning Research and Development Center planned a methodology for evaluating
27 the educational efficacy of Robotic Autonomy, implementing both formative and summative evaluations of progress
28 as well as an in-depth, one week ethnography to identify micro-genetic mechanisms of learning that would inform
29 the broader evaluation. This article describes the robot and curriculum design processes and then the educational
30 analysis methodology and statistically significant results, demonstrating the positive impact of Robotic Autonomy
31 on student learning well beyond the boundaries of specific technical concepts in robotics.

13 Keywords:**14 1. Introduction**

15 Robots have been playing an active role in education
 16 since the advent of the LOGO Turtle (Papert and Harel,
 17 1991). Both as project foci in laboratory coursework
 18 and as team challenges in national contests, the pro-
 19 cess of designing, building and programming robots
 20 has served to excite students across a broad age range.
 21 The current field of robotic educational endeavors is
 22 extremely large and diverse; see Fong et al. (2002) and
 23 Druin and Hendler (2000) for an overview.

24 We have had two primary goals in designing and
 25 executing a new robotics course. First, we planned to
 26 explicitly evaluate the educational impact of robotics
 27 on secondary level students. We were particularly in-
 28 terested in quantifying lessons learned in service of
 29 robotics that are broadly applicable to learning in gen-
 30 eral. Second, we hoped to collect data covering a far
 31 longer span of time than can be afforded based on a
 32 weekend robotics course. Not only would the planned
 33 course need to fill a summer; but the students should be
 34 able to continue their explorations at home after course
 35 completion.

36 To enable our basic goal—the educational assess-
 37 ment of a long-term course of study in robotics—the
 38 authors and others developed, taught and evaluated
 39 *Robotic Autonomy*, a seven-week introductory hands-
 40 on robotics course as part of Carnegie Mellon West's
 41 NASA-Ames campus in Mountain View, California
 42 (RASC, 2003). The research surrounding this effort
 43 included robot design, curriculum design and ongoing,
 44 long-term educational evaluation. Although we and
 45 other authors recognize and study the role of robotics
 46 in education (Beer et al., 1999; Druin, 2000; Kumar
 47 and Meeden, 1998; Murphy, 2000; Nourbakhsh, 2000a,
 48 2000b; Wolz, 2000), this work is notable in that all as-
 49 pects of the robot mechanism, electronics, software and
 50 educational curriculum were subject to ground-up, co-
 51 ordinated design. A total of 30 Trikebot robots were
 52 built and used during this program. They continue to
 53 be used by graduates of the course at home.

54 This article begins with a brief overview of the
 55 *Robotic Autonomy* curriculum, then presents the ed-
 56 ucational robot design process in Section 3, including
 57 mechanical considerations, control electronics and the
 58 student robot programming interface. Section 4 then
 59 presents the educational analysis methodology in de-

tail. A discussion of results follows, with statistically
 significant learning demonstrated over a number of
 coded themes, including Teamwork and Problem Solv-
 ing, as well as an analysis of gender differences. The re-
 sults strongly support the contention that robotics cur-
 riculum not only meets specific instructional goals but
 can also provide meaningful student engagement for
 general interest, skills and confidence for promoting
 future success in technology education.

2. Course Overview

A sufficiently competent mobile robot is not available
 commercially at a reasonable price for long-term stu-
 dent robot interaction. Thus short-term robotic educa-
 tional efforts often turn to Lego building blocks, usually
 designing curriculum both around robot morphology
 and construction as well as robot programming and in-
 teraction (Stein, 2002; Wolz, 2000). Another success-
 ful approach has been the integration of research robots
 and field robot prototypes into curriculum, where time
 with the robot is rare and therefore valuable (Coppin
 et al., 1999, 2002; Maxwell and Meeden, 2000). We
 were particularly interested in focusing on a course that
 would concentrate on robot behavior and robot algo-
 rithm rather than robot morphology. In order to provide
 every graduate of *Robotic Autonomy* with such a rich,
 programmable robot that would be robust to hundreds
 of hours of use, we chose to design and produce a new
 educational robot (Hsiu et al., 2003).

Robotic Autonomy was taught over a seven-week
 period in the summer of 2002 at the Carnegie Mellon
 West campus, located within NASA/Ames Research
 Center (Mountain View, California). The top-level goal
 for this course was straightforward: to provide selected
 high school students with an immersive exploration
 of mobile robotics using leading-edge technologies.
 Course graduation was intended to mark, not the com-
 pletion of these educational activities, but a launching
 point: every student would take home a robust, pro-
 grammable mobile robot system for continued explo-
 ration for months and years. Although robotics would
 be the focus of this curriculum, we hoped that lessons
 learned would encompass important concepts reaching
 well beyond just robotics.

103 2.1. Organization

104 The Robotic Autonomy course was aimed at students
 105 entering their senior year of high school, and spec-
 106 ified one prerequisite: the successful completion of
 107 any introductory programming course. Following the
 108 application and acceptance process, student composi-
 109 tion ultimately included 18 students attending under
 110 full scholarship and 10 paying students (see Fig. 1).
 111 The scholarship students were from various underpriv-
 112 ileged backgrounds, and were primarily Hispanic. The
 113 course was comprised of 8 girls and 20 boys.

114 The course structure depended primarily on team-
 115 work. Principles governing effective teamwork were
 116 explicitly discussed, as shown by the curriculum be-
 117 low. Students self-organized into teams of three dur-
 118 ing the first day, with the constraint that single-gender
 119 teams be created whenever possible. Based on previ-
 120 ous experience teaching robotics courses at the under-
 121 graduate level, we felt that single-gender female teams
 122 would be more likely to encourage active participa-
 123 tion by all members of the team, especially in the case
 124 of shy female students. Throughout the seven weeks,
 125 all team members shared joint responsibility to meet
 126 course challenges, with all members of the team re-
 127 ceiving the same grade on each week's activities. In
 128 order to tackle weekly assignments, each team used
 129 just one of their three robots in early weeks, but by the

first month's end made use of all three team robots in
 cooperative robot team exercises. 130 131

The course was taught by a single principal instruc-
 tor and four teaching assistants. The teaching assis-
 tants ranged from graduate to undergraduate students. 132 133 134
 There were also two instructors who took the course
 as a way of becoming trained to teach it in the future. 135 136
 They provided some teaching assistance as well. One 137
 of the teaching assistants was female, and the rest of 138
 the instruction team was male. 139

2.2. Curriculum 140

Robot-based curricula is used today across diverse 141
 age groups and with a broad variety of purposes. At 142
 the informal learning extreme, after-school programs 143
 based on annual contests have become popular with the 144
 advent of national contests including Botball (2004) 145
 and Stein (2002) and US FIRST (Hobson, 2000; US 146
 FIRST, 2004; Yim et al., 2000). These contests de- 147
 mand that teams of students together design, fabri- 148
 cate and iterate to present robotic solutions that of- 149
 ten perform in head-to-head exercises against competi- 150
 tors. Botball and US FIRST are foremost team-based 151
 physical design challenges, and there is evidence that 152
 such competitive design exercises draw upon cross- 153
 disciplinary skills in a powerful manner (Manseur,



Figure 1. The Robotic Autonomy 2002 students.

4 *Nourbakhsh et al.*

154 2000). Yet the competitive aspect of these contests can
 155 be troubling in terms of gender disparity; for instance
 156 (Milto et al., 2002) utilizes student surveys to conclude
 157 that males find the contest format more interesting than
 158 women do.

159 Nevertheless contests have permeated a significant
 160 proportion of robotics curricula at all age levels.
 161 Kolberg and Orlev (2001) presents cumulative weekly
 162 exercises designed for 5th and 6th grade levels, build-
 163 ing incrementally sophisticated behaviors and culmi-
 164 nating in a final contest. Numerous courses at the under-
 165 graduate level, including (Archibald and Beard, 2004)
 166 present robotics in the framework of a simplified Robot
 167 Soccer contest, with students working in teams to de-
 168 sign and build soccer-playing robots that then compete
 169 for glory. At the Swiss Technical Universities of EPFL
 170 and ETHZ (Siegwart, 2001) has documented robot de-
 171 sign and fabrication courses for undergraduates also
 172 culminating in a final contest that changes annually.

173 A number of robot courses bring existing robot plat-
 174 forms to bear, shifting the intellectual emphasis from
 175 robot design to robot algorithm design, as in the under-
 176 graduate setting (Maxwell and Meeden, 2000; Billard
 177 and Hayes, 1997; Billard, 2003; Nourbakhsh, 2000b;
 178 Murphy, 2000; Kumar and Meeden, 1998). For ex-
 179 ample since 1994 (Nourbakhsh, 2000b) has taught a
 180 mobile robot programming course at Stanford and at
 181 Carnegie Mellon University in which students incre-
 182 mentally program Nomad mobile robots, culminating
 183 in a team-based final competition.

184 Common to virtually all of the robotics curricula sur-
 185 veyed is the focus on challenge-based, hands-on and
 186 bottom-up learning. The bottom-up approach, which
 187 maximizes exploration and self-discovery, is inspired
 188 by Constructionism (Papert and Harel, 1991). The
 189 curriculum designed for Robotic Autonomy is most
 190 similar in spirit to that of Kolberg and Orlev (2001), pre-
 191 senting a series of incremental exercises at a weekly pe-
 192 riod. However, Robotic Autonomy is a full-time, eight
 193 hour per day course, demanding a rich set of exercises
 194 that can stimulate students of varying robot aptitude.
 195 We wished to focus on robot algorithm development
 196 rather than physical robot design specifically because
 197 of the paucity of such coursework. Our hope was that
 198 sufficient attention to robot algorithms, mated with a
 199 highly competent robot platform, would lead to truly
 200 sophisticated robot projects by course's end. We also
 201 wished to avoid the potential gender bias of a contest-
 202 based focus, and indeed wanted to ensure that self-
 203 motivation and inquiry would be paramount because of

the important of such skills for home exploration with
 the Trikebot. Thus our curriculum diverges from that
 of design-centered coursework such as Archibald and
 Beard (2004) and Siegwart (2001) and is more similar
 to the incremental programming curricula of Kumar
 and Meeden (1998), Nourbakhsh (2000b) and Kolberg
 and Orlev (2001).

Robotic Autonomy is designed around a one-week
 unit length, with an intra-week repeating structure to
 provide a familiar weekly trajectory. Each Monday and
 Tuesday is spent presenting new material and posing
 a new, open-ended challenge for each team to tackle.
 Wednesday is Challenge Day, including extensive test-
 ing of the challenge submissions of every team. In
 addition, a portion of this day is set aside for each
 team to document their weekly solutions, including
 source code, prose, pictures and videos to be placed
 on a specially configured team website. On Thurs-
 day morning, teams receive the details of an end-of-
 week contest, which apply the concepts learned for that
 week's challenge in an enjoyable and competitive for-
 mat. Thus Thursday is spent preparing carefully for the
 next day's contest. Friday is Contest Day, with invited
 guests (parents, administrators and visitors) watching
 and cheering as team robots engage in games such as
 line-following races, bomb defusing contests, musical
 chairs, *et cetera* (see Fig. 2). While contests are thus not
 completely eliminated, Friday contests are designed
 carefully as significantly simpler tasks than the chal-
 lenges due two days earlier, on Wednesdays. As the
 end of the course approached, Friday contests were
 replaced by team-designed Exhibitions, making the
 full transition from mediated learning to self-directed
 exploration.

In summary, new concepts are largely presented
 early in the week, with the most difficult bar set by the
 Wednesday challenge. Following this intellectual apex,
 the Friday contest offers a chance for students to reuse
 lessons learned that week in an enjoyable and playfully
 competitive atmosphere. In addition to the direct lec-
 tures and challenges, weekly guest speakers are brought
 in on Mondays and Tuesdays to provide one-hour dis-
 cussions on their areas of expertise. These speakers
 provide both an outside perspective on robotics and a
 window into the lifestyle of career roboticists.

The outline below shows the challenges and con-
 tests associated with each week of Robotic Auton-
 omy, together with the underlying concepts learned in
 that week. Also noted are prepared speeches and guest
 speakers' topics. The complete curriculum for Robotic



Figure 2. Several examples of Friday contests: Robot Theater (left); Bomb Squad (center); Outdoor Jogging (right).

254	Autonomy as well as all student web sites are available	Controlling robot speed and position using time	288
255	for download (RASC, 2003).	Testing and tuning ded-reckoning, servo and speed calibration	289
256	Week 1	Trapezoidal speed profiles	291
257	<i>Challenge</i>	Programming the Trikebot	292
258	Stand-alone Java timer and calculator programs	Testing ded-reckoning error using geometric scripted motions	293
259	Build Trikebots with unique outfits	Sequential scripted motions	294
260	<i>Contest</i>	Website documentation	296
261	Capture the Flag (remote-control operation)	iPAQ connection diagnostics: problem solving without instructor assistance	297
262	<i>Concepts</i>	Elements of a good robotic theater performance	299
263	Using hand tools	Designing and implementing functionality for GUI buttons in teleoperation	300
264	Using buttons and textfields in the Java GUI: javac and java		301
265	How joints, servos, and motors work	<i>Talks</i>	302
266	Kinematics: the Instantaneous Center of Rotation	Thomas Hsiu: talk on robotics in special effects & Hollywood	303
267	Introduction to electronics: batteries, power, PWM motor control, servos, wiring, plugs, connectors, polarity		304
268	Using the iPAQ to directly test the Trikebot		305
269	Using the Java Trikebot UI for direct motor control	Week 3	306
270	How to use the iPAQ: network configuration	<i>Challenge</i>	307
271	<i>Talks</i>	Touch-free racing (signaling to the robot via the rangefinder)	308
272	Thomas Hsiu: talk on Mechanical design considerations	Autonomous wandering and exploring	310
273		<i>Contest</i>	311
274		Escape (crossing an obstacle field)	312
275		Musical chairs (mixed autonomous wander and remote operation)	313
276		<i>Concepts</i>	314
277		Downloading firmware to the iPAQs without instructor assistance	315
278	Week 2	The role of sensing in autonomy	317
279	<i>Challenge</i>	Survey of rangefinding sensors	318
280	Ded-reckoning primitives for timed robot moves	Accessing the Sharp Rangefinder readings on the Trikebot using Java	319
281	Autonomous, choreographed, robot dance	Creating sensor-driven robot control code	320
282	<i>Contest</i>		321
283	Robot theater (choreographed autonomy)		322
284	Robot soccer (button-based remote operation only)		
285	<i>Concepts</i>		
286	Physical robot sources of error: wheel-floor interactions, backlash, slippage		
287			

6 *Nourbakhsh et al.*

323	Open sourcing robot software: how to make a code repository and why	Outdoor jogging (high-speed visual following)	373
324	Adjustable autonomy: mixing autonomy with remote control	<i>Contest</i>	374
326	Proprioception in humans and achieving this in robots	Navigator (autonomous navigation using visual fiducials)	375
327	Back-EMF based DC Motor speed sensing: principles and execution	Robot exhibition design	376
330	Motor acceleration and current: relationships	<i>Concepts</i>	377
331	Teamwork: evaluating the effectiveness of teams; communication; best practices	Designing your own team exhibition: starting the process	378
332	Techniques for maintaining a sense of direction: sensing versus physical manipulation	Active visual tracking and following	379
333	<i>Talks</i>	Picking good colors for CMUcam: using DumpFrame for diagnostics	381
337	Tom Lauwers: talk on Starting a Robotics Club of your own	Providing tracking information as feedback to neck servos to center head	382
338	Illah Nourbakhsh: talk on Innovative Mechanism: Gyrover, Bowleg hopper	Considerations for designing hardware and software for 1 robot to follow another	383
339		Videotaping your robots: open-source value; pointers	384
340		Localization and navigation: designing navigating systems; landmarks; heading, termination	385
341		Adjustable autonomy; modes of interaction with semiautonomous navigators	386
342	Week 4		387
343	<i>Challenge</i>		388
344	Martian Explorer (Video-based, high-latency teleoperation)		389
345	Go Home (Teleoperation-based localization)		390
346	Visual tracking challenge (maximum tracking distance)		391
347	<i>Contest</i>		392
348	Bomb Squad (team-based bomb disposal)		393
349	Outdoor visual control (vision-based cues for head-to-head races)		394
350	<i>Concepts</i>		395
351	Human vision: anatomy, color sensing and object recognition in the brain	Week 6	396
352	CMOS-based vision sensors: background photonics and limitations	<i>Challenge</i>	397
353	CMUcam: electronics overview and introduction	Mobot (a simple line-following race)	398
354	Attaching and using CMUcam (hardware, EE, UI)	Security Patrol (motion/intruder detection)	399
355	Intelligent Teleoperation: research survey	<i>Contest</i>	400
356	Color tracking with CMUcam: pitfalls and representations of color	Mazeworld (mixed teleoperation and autonomous line-following)	401
357	Autogain, auto white-balance, and other visual feedback loops in CMUcam	Exhibition 2.0 (second chance to refine and practice custom exhibit)	402
358	<i>Talks</i>	<i>Concepts</i>	403
359	Anthony Rowe (CMUcam inventor): talk on designing and using the CMUcam	Detecting and performing line-following: vision and feedback control, convergent control, oscillations, considerations for vision outdoors	404
360		Robotics and Social Responsibility: broad discussion of technology and impact	405
361		Control and state: Zero-state functional systems; reactive systems; state machines	406
362		Navigation as map representation and state representation	407
363		<i>Talks</i>	408
364		Liam Pedersen: talk on Robotics at NASA and NOMAD in Antarctica	409
365		Jonathan Buford: talk on muscle wire on a robotic spider	410
366			411
367			412
368			413
369	Week 5		414
370	<i>Challenge</i>		415
371	Duckling (autonomous robot visual tracking and following)		416
372			417
			418
			419
			420
			421
			422

423 *Concepts*

- 424 Navigation: path planning techniques commonly
- 425 used, a survey
- 426 Public speaking and presentations: pointers and tips
- 427 Robot demonstrations and exhibit design

428 *Talks*

- 429 Illah Nourbakhsh: talk on The Personal Rover
- 430 Project
- 431 Steve Richards: talk on Acroname's robots; over-
- coming ded-reckoning error

433 **3. Robot Design**

434 The robot design goals are informed by the intended
 435 target audience for the educational course: high school
 436 students between their junior and senior year. Prerequ-
 437 isites include basic mechanical dexterity (e.g. sim-
 438 ple assembly and fabrication) and knowledge of a
 439 programming language (e.g. *Introduction to Program-*
 440 *ming*). Significantly, each student would be slated to
 441 take a Trikebot home to program and use at will after
 442 course completion. Thus, the Trikebot would need to
 443 be designed not only for the beginning robotics stu-
 444 dent but for the continuing, sophisticated user. In other
 445 words the robot would need to have sufficient expres-
 446 siveness and capability to serve as an educational and
 447 exploratory tool beyond the confines of a seven-week
 448 course.

449 Design and production of a new educational robot,
 450 the Trikebot, was a costly and time-consuming step in
 451 the execution of Robotic Autonomy. After surveying
 452 state-of-art educational robot hardware alternatives,
 453 we concluded that this design and fabrication process
 454 would yield a significantly more desirable solution; the
 455 reasons behind this decision are worth amplification.

456 The Robotic Autonomy course was an intensive but
 457 short-term course, providing students with seven weeks
 458 in a formal learning environment with the hope of
 459 sparking self-directed further exploration into *robot al-*
 460 *gorithms* with the robotic platforms after graduation. To
 461 this end six qualities were paramount in the selection
 462 of an educational robot for Robotic Autonomy.

- 463 1. *Mechanical Empowerment*. Even in an algorithm-
 464 focused course, a broad introduction to robotics de-
 465 mands inclusion of electromechanical aspects of
 466 robot design as well as robot programming. Fur-
 467 thermore, because students would keep the robots
 468 following graduation, they must be sufficiently em-
 469 powered to be able to repair their robots in the near-
 470 certain case of eventual physical malfunction and

breakage. Our strategy for meeting this need is to
 ensure that the educational robot arrives in kit form:
 students construct each robot, which has solely off-
 the-shelf life-limited parts, and are thereafter able
 to replace such parts.

- 2. *Behavioral Richness*. The desire for open-ended,
 project based exploration leads to a requirement for
 sufficiently rich robot-world and human-robot in-
 teraction as to engage students during weeks and
 months of programming. This relatively vague re-
 quirement is made concrete by way of two hard
 constraints: the robot platform must have visual
 competence (i.e. ability to track fiducials, follow
 lines, detect visual environmental changes) and
 must be richly programmable using a high-level pro-
 gramming language (e.g. C++, Java, etc.).
- 3. *Robustness*. As is the case with all electromechan-
 ical course products, an educational robot must be
 robust to the numerous accidents which occur with
 great frequency in the initial few days of a mobile
 robotics course. Furthermore, because we aimed for
 student projects for which robots may move for
 an hour or more autonomously, mechanical robust-
 ness should extend temporally over more than a few
 minute of run-time.
- 4. *Maneuverability*. In keeping with (2) Behavioral
 Richness and to engage and challenge students in
 their own natural world, we stipulate that the robotic
 platform should be capable of maneuvering at fast
 walking speed in both indoor and outdoor environ-
 ments, including sidewalks, short grass and gravel.
 Such breadth of application environments opens the
 field in terms of team and individual robot program-
 ming challenges throughout classroom areas and the
 field.
- 5. *Wireless Scaling*. Robotic Autonomy planned for up
 to 40 students at one time, and therefore a hard con-
 straint is that all wireless robot control be scalable
 to at least 40 simultaneous robots. This requirement
 alleviates the unnecessary logistical burden of time-
 sharing robot execution among multiple teams and
 robots.
- 6. *Price Point*. Given a fixed budget and the desire
 to award every Robotic Autonomy graduate with
 a high-competence mobile robot, we established a
 hard limit of \$2,000 total cost per robot platform,
 with a bias for the least expensive possible solution.

A number of existing robotic platforms satisfy a
 subset of the criteria noted above. Table 1 provides
 comparison data for several popular educational robot

Table 1. A comparison of educational robot platforms and kits in view of six Robotic Autonomy robot constraints. Rows include the basic Lego Mindstorms kit (Martin et al., 2000); the Lego kit augmented by a Handyboard microprocessor (Botball, 2004); Amigobot (ActivMedia, 2004); Khepera (K-Team, 2004); ER1 (Evolution Robotics, 2004); Garcia (Acroname, 2003); and the authors' Trikebot solution.

	Mechanism	Behavior	Robustness	Maneuvering	Scalability	Price point
Lego RCX	Y	N	N	N	N	\$200
Lego-Handy	Y	Y	N	N	N	\$1000
Amigobot	N	Y	Y	N	N	\$2500
Khepera	N	Y	Y	N	N	\$2000
ER1	Y	Y	Y	N	Y	\$300
Garcia	N	Y	Y	N	Y	\$1700
Trikebot	Y	Y	Y	Y	Y	\$1200

521 packages. Kits such as the Lego Mindstorm are the
 522 basis of a number of successful robot courses (Fagin,
 523 2003; Gage and Murphy, 2003; Kumar, 2001; Schu-
 524 macher et al., 2001; Wang, 2001; Wang and Wang,
 525 2001). When used in conjunction with the Lego RCX,
 526 such a solution satisfies our price point and mechani-
 527 cal empowerment constraints only. Lack of RAM and
 528 ROM space on the RCX obviates rich programmabil-
 529 ity, as does a lack of vision-based on-board sensing.
 530 Addition of a more sophisticated microprocessor such
 531 as the Handyboard (Botball, 2004; Nagchaudhuri et al.,
 532 2002) allows for more sophisticated sensors, actuators
 533 and algorithms.

534 A second popular approach even eschews the me-
 535 chanical modularity of Lego, preferring to empower
 536 students to fabricate robots of their own design us-
 537 ing metal, wood, plastic and other rapid prototyping
 538 materials (Heer et al., 2002; Siegwart, 2001). While
 539 such robot design projects have real educational bene-
 540 fits, such a focus on robot construction is often at the
 541 expense of time spent exploring sophisticated robot
 542 programming.

543 Existing, commercially available educational robots,
 544 as shown in Table 1, satisfy only a subset of our con-
 545 straints (K-Team, 2004; ActivMedia, 2004; Evolution
 546 Robotics, 2004; Acroname, 2003). To be fair, such
 547 commercial solutions achieve far higher levels of over-
 548 all robustness than the Trikebot; however, when fail-
 549 ures do occur, such systems can be extremely difficult
 550 to repair in the home due to their lack of off-the-shelf
 551 parts and their mechanical complexity. In terms of price
 552 point, existing commercial products appear to be more
 553 expensive than Trikebot. This is surprising given that
 554 such commercial products are created in higher vol-
 555 umes than the Trikebot. There are two reasons for this
 556 disparity. First commercial products must include suf-

ficient markup for long-term viability while Trikebot's
 price essentially represents Cost of Manufacture. Sec-
 ond the Trikebot benefits in price from significant parts
 donations and price reductions by commercial vendors.
 Finally, in terms of the maneuverability feature, the
 level of terrainability desired for Robotic Autonomy is
 not available in existing products to our knowledge.

In comparison to existing robot platforms the Trike-
 bot occupies a point in design space that is particularly
 well-suited to the nature of Robotic Autonomy. The
 chassis consists of durable plastic pieces fitted together
 via a slot and tab design. All degrees of freedom are
 actuated by off-the-shelf hobby servomotors available
 on-line. The on-board IPaq PDA is also off-the-shelf,
 and provides scalable 802.11b wireless connectivity
 to an off-board portable laptop, which itself enables
 high-level programming in Java. The CMUcam vision
 sensor, capable of line following and object tracking
 (via color) paves the way for relatively engaging and
 rewarding robot behavior, all of which can be executed
 both indoors and outdoors because of the Trikebot's
 large diameter wheels and ground clearance.

Sections 3.1, 3.2 and 3.3 describe design objectives
 and solutions for robot mechanism, control electronics
 and the student programming interface respectively.

3.1. Robot Mechanism

The Trikebot chassis has three primary functions. It is a
 camera platform for the CMUcam (Rowe et al., 2002),
 it provides mobility over a variety of indoor and outdoor
 terrains, and it can carry a relatively large payload. In
 addition to these functions, the Trikebot is meant to be
 assembled and serviced by students with few special-
 ized tools. Most of the related design decisions were
 driven by these requirements.

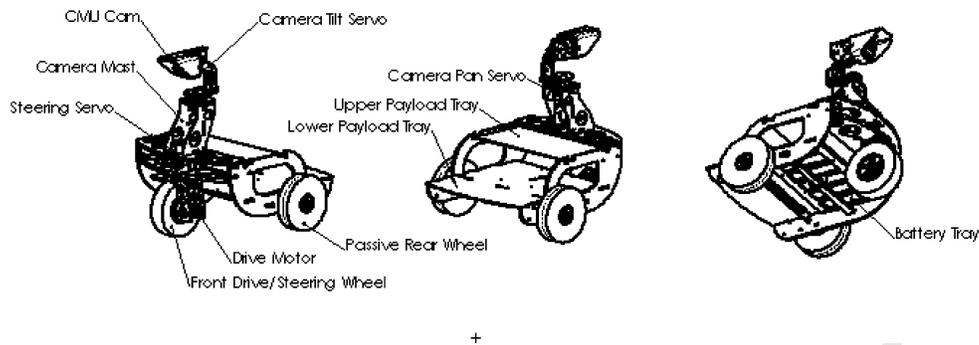


Figure 3. Trikebot chassis overview.

591 As a camera platform, our goal was to place the camera
 592 at least 18 inches above the ground plane. This
 593 was part of a decision to make the Trikebot a floor-
 594 based robot with which students could interact more
 595 dynamically than a smaller table-top size robot. Putting
 596 the camera relatively high off the ground both gives the
 597 camera a wider effective field of view and encourages
 598 students to interact with the robot at an eye-to-eye level.
 599 Camera placement is important both for teleoperation
 600 modes of control and for autonomous robot operation.
 601 The pan and tilt mechanism is critical for *diagnostic*
 602 *transparency* and affection; it enables the robot to
 603 clearly indicate direction of gaze and widens the field
 604 of view further (Fong et al., 2003).

605 Because we expect the Trikebot to operate not only
 606 indoors but also on relatively flat outdoor areas such
 607 as parking lots, sidewalks and lawns, it must be able
 608 to overcome minor obstacles such as electrical cables,
 609 door thresholds and gravel. The robot's ground clear-
 610 ance and wheel size enable such locomotion. To fac-
 611 ilitate mobility in closed quarters, we required Trike-
 612 bot to turn in-place within a 24 inch circle. Finally, to
 613 encourage student-robot interaction, the top speed of
 614 the Trikebot was specified as comparable to a person's
 615 medium speed walk, roughly 30 in/sec.

616 As a worst-case payload requirement, the Trikebot is
 617 designed to carry a laptop computer, six 7.2 V Remote
 618 Control (RC) car battery packs and various onboard
 619 electronics. This payload objective would turn out to
 620 be an overestimate primarily because our final archi-
 621 tecture enabled an off-board laptop to communicate via
 622 802.11b with the Trikebot, as described in Section 3.2.

623 Being assembled and maintained by students in a
 624 general classroom environment required that the ma-
 625 jority of the components of the Trikebot be assembled
 626 using simple hand tools and that they be robust enough
 627 to handle rough treatment. Of course, cost is always an

issue, so appropriate manufacturing techniques were
 chosen for the quantities of parts used. This dictated the
 look and feel of the individual components designed.

Together, all of the above objectives influenced the
 final design of the Trikebot. The proceeding chassis
 overview is followed by descriptions of how the various
 elements of the Trikebot chassis meet these objectives.

The Trikebot chassis is a three-wheeled mobile robot
 base in a tricycle-like configuration, with a single
 driven steerable wheel and two fixed passive wheels.
 Its major physical features are a tall camera mast with
 a pan and tilt mechanism and two large, flat payload
 areas, one low in the chassis and another smaller shelf
 above it (Fig. 3). Altogether the Trikebot has 4 control
 degrees of freedom—drive motor, steering, camera pan
 and camera tilt.

The *tread width*, or distance between wheel center-
 lines as viewed from the front or back, is 15.8 inches
 and the *wheelbase*, or distance between wheel axes
 as viewed from the side, is 10.9 inches (Fig. 4). The
 wheels of the Trikebot are 6 inches in diameter, sup-
 porting a ground clearance of 2.2 inches. The nominal
 camera height is 18.3 inches and it can pan approxi-
 mately $\pm 90^\circ$ and tilt $+90^\circ/-45^\circ$. Overall, the mechani-
 cal chassis alone, minus batteries and electronics but in-
 cluding servos and drive motor, weighs approximately
 10.5 lbs.

Wheel Configuration. A tricycle configuration with
 a single driven steering wheel gives the Trikebot very
 good agility using a single gearmotor as its drivemotor
 and a single high power servo for steering. The servo
 can steer the driven wheel through 180° allowing the
 robot to turn nearly in place, well within a 24 inch
 circle. This allows the Trikebot to turn completely around
 within a confined space such as a doorway.

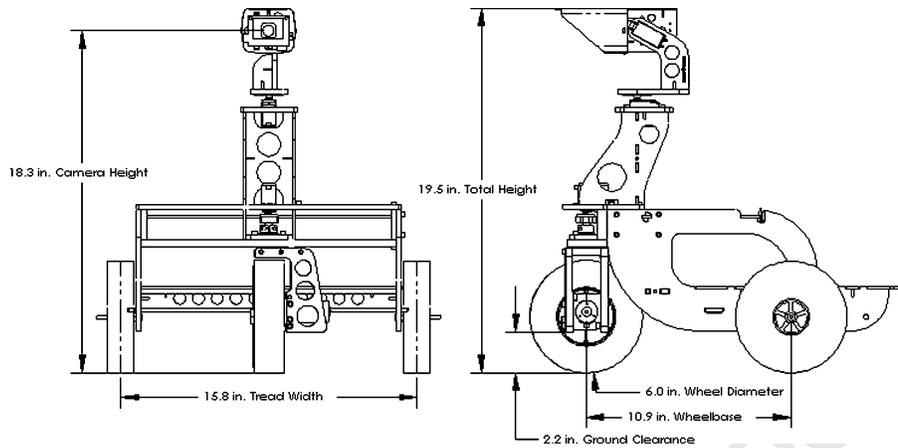


Figure 4. Trikebot chassis dimensions.

663 We chose the tricycle design in lieu of the other
 664 common three-wheeled configuration, with two driven
 665 wheels and one trailing caster wheel, to avoid several
 666 problems generated by that configuration. One problem
 667 is that the trailing caster wheel can restrict the freedom
 668 of movement of the robot in certain situations. For instance,
 669 when reversing direction of travel, the action of
 670 the caster reversing to trail the direction of motion can
 671 force the robot to deviate its course or cause wheel slip
 672 in the robot's drive wheels. Furthermore, two driven
 673 wheels must match their speeds exactly in order for
 674 the robot to travel in a straight line. This generally requires
 675 additional motor encoders to achieve sufficient accuracy.
 676 The tricycle design eliminates both of these
 677 issues.

678 The single wheel forward arrangement was chosen
 679 for agility over obstacles. The driven wheel can more
 680 easily grip and climb over an obstacle at slow speeds,
 681 subsequently dragging the rear wheels over the obstacle.
 682 The tradeoff is less stability during maneuvers at
 683 high speeds, but it was expected that most of the Trikebot's
 684 maneuvers would not be at full speed.

685 One final advantage of a three-wheeled design is
 686 lowered torsional stress on the chassis. In a four
 687 wheeled chassis, a single wheel can be raised above
 688 the others when traversing uneven terrain. This causes
 689 torsional stress on the chassis which can twist the chassis
 690 (and its payload) unless it is strong enough to resist
 691 the twisting. A three wheeled chassis undergoes much
 692 less twisting, meaning the chassis can be both simpler
 693 and lighter.

694 **Wheels.** With wheel diameters of 6 inches and a
 695 ground clearance of 2.2 inches, the Trikebot can drive

over obstacles such as power cords, uneven sidewalks,
 and even gravel paths. The traction element of the
 wheels consists of closed-cell foam rubber tires. These
 tires provide adequate stiffness and traction, yet are still
 light and help absorb shocks. The rear passive wheels
 and front wheel hub are stock RC model airplane parts
 and car parts, utilized to minimize cost.

The Drivetrain. The drivetrain consists of a 19.5:1
 gearmotor directly coupled to the drive wheel. The
 gearmotor's output bearings are adequate for the loads
 expected to be delivered by the Trikebot and direct drive
 provided the simplest design. Together with a motor
 clamp and motor support structure, the drive wheel and
 gearmotor comprise a drive wheel assembly (Fig. 5).

The drive wheel assembly turns about a kingpin
 which is centered above the center of the drive wheel
 and mechanically fixed to the drive wheel assembly.
 By positioning the steering axis directly above the center
 of the drive wheel, no steering torque is generated
 when the drive motor is engaged. The kingpin is supported
 by two sets of ball bearings pressed into the main
 chassis. These bearings carry the load of the chassis on
 the drive wheel assembly, allowing the robot to steer

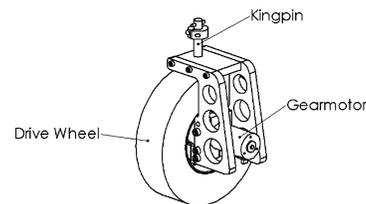


Figure 5. The drive wheel assembly.

719 with minimal friction. A high-torque RC servo directly
720 drives the kingpin, providing steering control.

721 **Camera Mast and Pan and Tilt.** The camera mast
722 incorporates a pan and tilt mechanism and elevates the
723 camera to its desired 18 inch minimum height. The
724 positioning of the mast to the front of the chassis al-
725 lows the camera to scan slightly in front of the front
726 wheel while looking down. This facilitates activities
727 such as line following or object-in-path detection. The
728 pan and tilt mechanism was also designed to maximize
729 the camera's potential field of view. The camera is cen-
730 tered above the camera's pan axis and the camera's
731 centerline passes through the tilt axis. This simplifies
732 the analysis of the camera's view relative to the robot.
733 One design compromise was to reduce the camera total
734 pan angle from a panoramic 360° to 180°. This allowed
735 both the pan and tilt to be directly controlled by stock
736 RC servos, simplifying the design and reducing costs.

737 **Payload Area.** The Payload areas of the Trikebot are
738 positioned low and to the rear of the camera mast in
739 order to place the fully loaded robot's center of gravity
740 as low as possible and roughly 1/3 of the wheelbase be-
741 hind the front wheel. A low center of gravity maximizes
742 the stability of the Trikebot and placing the center of
743 gravity 1/3 of the way behind the front wheel helps
744 provide traction to the front driven wheel. The battery
745 racks are located below the lower payload tray, again to
746 lower the center of gravity and to provide easy access
747 to the batteries.

748 The lower payload tray is sized to accommodate a
749 standard laptop computer with the screen closed and
750 the upper payload tray tilts up to allow easier access to
751 the front of the lower tray.

General Construction. Most of the Trikebot chassis 752
is constructed of lasercut acetal (Delrin) sheets (Fig. 6). 753
Laser cutting provides maximum flexibility for the rel- 754
atively small number of parts produced for this project 755
while being more economical than traditional machin- 756
ing. Aluminum machined parts were used for a few 757
items, such as the drive hubs and motor clamps, but 758
machining was minimized as it costs over ten times to 759
produce comparable parts over lasercutting. However 760
lasercutting has its drawbacks, allowing only cuts per- 761
pendicular to flat sheets of material like paper or plastic 762
(or metal for higher power laser cutters). To accom- 763
modate this, the Trikebot's parts fit together with tabs 764
and slots, not unlike paper or cardboard models. Self- 765
tapping screws wedged into slots hold the plastic parts 766
together. While these fastening methods increase the 767
design time, they minimize secondary machining oper- 768
ations such as drilling and tapping holes, ultimately 769
saving cost. This also allows most of the Trikebot to be 770
assembled by the students themselves using hand tools. 771
When mechanical repairs or adjustments are needed, 772
the students have been able to perform these tasks them- 773
selves. Using rapid manufacturing technologies such 774
as lasercutting, combined with using stock parts such 775
as RC servos and wheels, enables the Trikebot to be 776
produced economically and quickly in the quantities 777
required for this project, while fulfilling the desired 778
design objectives. 779

3.2. Control Electronics 780

The role of the control electronics was to create a clean 781
interface between the physical robot layer and the high- 782
level Java programming interface the students would 783
use to program the robot. The electronics abstract away 784
most of the communication overhead, interface control



Figure 6. The unassembled components of one Trikebot; 30 assembled Trikebots (right).

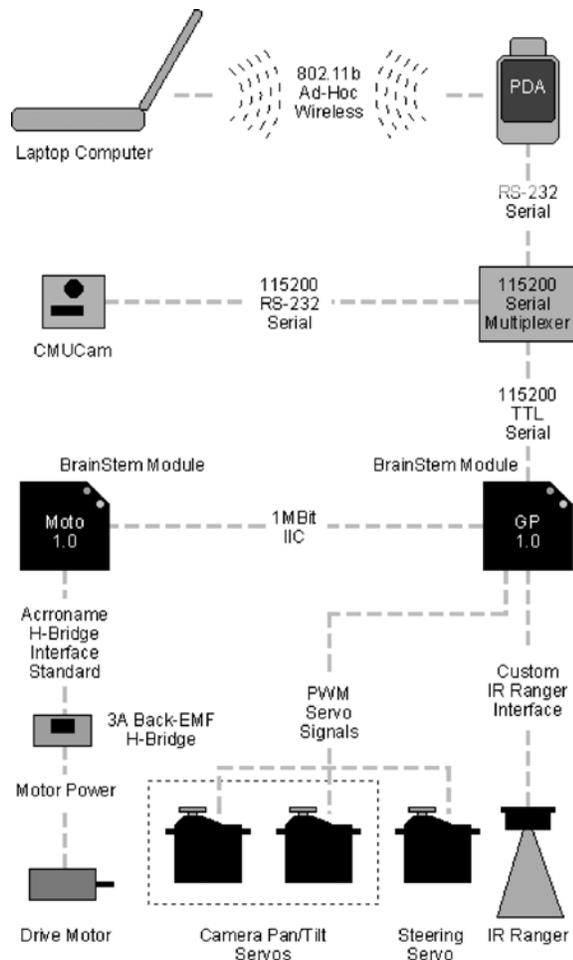


Figure 7. The Trikebot control electronics' connectivity.

785 and motion control aspects of the Trikebot. Our solu-
 786 tion accomplished this abstraction while allowing flex-
 787 ibility for expansion, lower level control and design
 788 modularity.

789 Figure 7 depicts the connectivity of the Trikebot's
 790 control electronics. An iPAQ 3650 serves as the
 791 802.11b wireless link between the robot electronics
 792 and the students' laptops. This ARM processor has
 793 sufficient power to be the Trikebot's main server
 794 computer but lacks an interface for easy programma-
 795 bility by the student. Laptop to iPAQ communication
 796 is achieved over TCP/IP, with the resulting serial
 797 stream multiplexed between the CMUCam, which
 798 provides visual perception services, and the Brainstem
 799 network, which provides motion and sensing
 800 control.

3.2.1. **BrainstemTM Architecture.** In the Trikebot, 801
 the BrainStem network is primarily a slave controller. 802
 The student's laptop performs high-level decision mak- 803
 ing and sequencing, in turn requesting control outputs 804
 and inputs from the Brainstem network using a Java 805
 API. The BrainStem architecture offers rich I/O capa- 806
 bilities in slave mode but can also function indepen- 807
 dently via TEA (tiny embedded application) programs 808
 which use ANSI C syntax to run on small virtual 809
 machines located within the BrainStem module's 810
 controller (Acroname, 2003). Writing TEA programs 811
 or setting up reflexes offers more control capacity and 812
 can serve as an expansion option for Trikebots. TEA 813
 can also offer failsafe behavior handling when the wire- 814
 less link or iPAQ encounters trouble. 815

The Trikebot's steering and camera pan/tilt servos 816
 are driven by the BrainStem GP 1.0 module. This board 817
 also supports the Sharp GP2D02 IR distance ranger. 818
 Both of these tasks are managed by the GP 1.0 module 819
 which encapsulates the serial clocking of data from the 820
 digital IR sensor, dampens the motion input to the ser- 821
 vos, and manages the servo ranges and offsets. Once 822
 configured, simple commands can be sent to the GP 823
 1.0 module for neck position, steering and distance 824
 ranging. 825

This GP 1.0 module also acts as a serial to I2C router 826
 to communicate with the other BrainStem Module, the 827
 Moto 1.0 board. This approach allows all commands to 828
 be sent to the BrainStem I2C network via a single serial 829
 connection. The Moto 1.0 module handles the closed- 830
 loop motion control of the Trikebot's motor. This mo- 831
 tion control is performed using PWM (pulse width 832
 modulation) output to an H-Bridge daughterboard. 833

3.2.2. **Back-EMF Based Speed Control.** One unique 834
 ability of the H-Bridge and Moto 1.0 module used in 835
 the Trikebot is Back-EMF speed measurement. This 836
 approach uses the natural characteristics of a spinning 837
 motor to derive a feedback voltage that is linearly pro- 838
 portional to the speed of the motor. 839

Most precision robotics applications use motors with 840
 optical or magnetic encoders offering quadrature posi- 841
 tion sensing. This approach is effective but the combi- 842
 nation of the precision encoders and quadrature decod- 843
 ing chips on the motion controller make this approach 844
 expensive. Using Back-EMF control allows feedback- 845
 based PID speed control while using a simple gearmo- 846
 tor with no encoder. 847

The basic idea behind Back-EMF speed control 848
 is that while a motor is being driven, the H-Bridge 849

850 windings that actually offer the connection to the drive
 851 current for the motor can be “floated” or left discon-
 852 nected. When this occurs, the induction developed in
 853 the windings of the motors quickly collapses and the
 854 motor transitions to a generator of current due to the
 855 residual inertia in the mechanical drive system. This
 856 takes place in only a few milliseconds. Once this tran-
 857 sition has occurred, the output voltage from the motor
 858 is directly related to the speed of the motor.

859 The Back-EMF circuit built into the 3A H-Bridge
 860 used in the Trikebot’s Moto 1.0 board measures the
 861 voltage from the motor and converts it to a logic voltage
 862 centered at 2.5 volts. When the motor is running full
 863 speed in one direction, the voltage drops to ~0.0 volts
 864 and when the motor is running full speed in the other
 865 direction, the voltage rises to ~5.0 volts. This is read
 866 by a 10-bit analog input on the Moto 1.0 module and
 867 used as the feedback for the PID equation driving the
 868 duty cycle of the motor. Once the A/D measurement is
 869 taken, the motor is switched back on and driven via the
 870 PWM output.

871 **3.2.3. iPAQ Robot Server.** The iPAQ ARM-based
 872 processor serves as both an 802.11b to serial bridge
 873 and a real-time sensorimotor controller on-board the
 874 robot. Together with the Brainstem components, this
 875 unit completes the on-board electronics of the Trikebot
 876 (Fig. 8). There are a number of reasons to avoid placing
 877 the student laptop directly onboard the Trikebot. First,
 878 reducing the payload requirements enables a longer
 879 running time for the robot and reduces the chances of
 880 robot damage in the case of collisions. By the same
 881 token, the laptop is kept out of harm’s way while pro-
 882 viding direct diagnostic feedback to the student, even
 883 during program execution. Finally, an off-board lap-
 884 top can serve as a teleoperation input device. Given
 885 the NASA collaboration in this project, such teleoper-



Figure 8. Control electronics located on the Trikebot.

ation was particularly relevant for curriculum exercises 886
 involving simulation of Mars Rover type activities. 887

The fundamental problem of removing the laptop 888
 and thus the high-level control program from the Trike- 889
 bot concerns communication latency. Even in the best 890
 of cases, roundtrip communication latency via 802.11b 891
 can easily exceed 150 ms. Although this is acceptable 892
 for high-level commands involving steering and speed 893
 decisions for the Trikebot, this is unacceptable for fast- 894
 feedback control loops such as visual pan-tilt tracking 895
 of moving objects using the Trikebot’s CMUcam. 896

Figure 9 summarizes the functional layers of the 897
 iPAQ firmware. Using a checksum-based message- 898
 passing protocol, the off-board laptop communicates 899
 high-level vision commands and robot I/O commands 900
 to the iPAQ. The iPAQ controls the serial multiplexer 901
 state and formats and handles dialogue with both the 902
 CMUcam and the Brainstem Architecture. 903

In addition to providing communication services to 904
 each downstream electronic device, the iPAQ serves 905
 three other functions. When the laptop requests an im- 906
 age dump from CMUcam, the iPAQ acts as an inter- 907
 mediate image buffer to collect and send that infor- 908
 mation. Because CMUcam can take up to 2 sec. to 909
 deliver a complete image at 115200 baud, this pro- 910
 cess must be asynchronous and thus the image data is 911
 transmitted back to the laptop via a dedicated TCP/IP 912
 image socket. Second, the iPAQ can serve as a pan- 913
 tilt feedback controller, utilizing CMUcam to measure 914
 the visual displacement of a tracked object, then com- 915
 manding the pan and tilt servos via Brainstem to visu- 916
 ally center the object being tracked. Once again, this 917
 feedback loop must be performed asynchronously and 918
 provides feedback to the laptop regarding the tracked 919
 object and the Trikebot’s neck position using a separ- 920
 ate TCP/IP socket. Third, the iPAQ takes advantage 921
 of the fact that all three servoed joints in the Trikebot 922
 are statically stable to save power. By running separate 923
 timers for each joint, the iPAQ is able to power down 924
 each servo once it has reached the commanded position 925
 (for the steering servo this is plausible only when the 926
 robot is not moving). This strategy significantly lowers 927
 power requirements when the Trikebot is idling. 928

3.3. Programming Interface 929

As the interface between student and robot, the laptop 930
 environment is critical for students to learn success- 931
 fully and enjoyably. One objective is that the environ- 932
 ment enable the student to directly control the robot’s 933

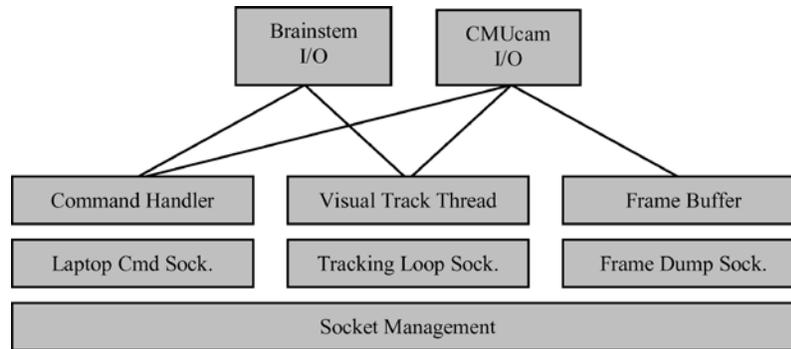


Figure 9. The functional layers of the iPAQ firmware.

934 motion (i.e. teleoperation) as easily and quickly as possible
 935 possible “out of the box.” A second objective is that, assum-
 936 ing basic knowledge of some programming language,
 937 a student should be able to program the Trikebot for
 938 autonomous motion with as shallow a learning curve
 939 as possible. The goal is thus to rapidly surmount the ob-
 940 stacles of learning machine-specific programming and
 941 compilation details, instead devoting the majority of in-
 942 tellectual effort to exploring the space of autonomous
 943 and interactive robot behaviors. Finally, closing the
 944 loop, the third objective is that the interface provide
 945 maximal diagnostic transparency during program execu-
 946 tion so that the student is empowered to improve the
 947 performance of the Trikebot (Nourbakhsh, 2000c).

948 The Trikebot UI is both the teleoperative interface
 949 and the program execution and monitoring interface
 950 and is described in Section 3.3.1. The subsequent
 951 section describes the programming interface, through
 952 which the students write JAVA code to control the
 953 Trikebot interactively and autonomously.

954 **3.3.1. Control and Diagnostic UI.** The Trikebot UI,
 955 shown in Fig. 10, enables direct teleoperation of the
 956 Trikebot. Direct teleoperation is not only important
 957 as a novelty; it is critical to the ongoing diagnostic
 958 process of students being able to shift their point of
 959 view to that of the robot. By dumping images from the
 960 Trikebot’s CMUcam, for example, students can visu-
 961 ally inspect the quality of the video signal on which
 962 they are attempting computer vision operations. By
 963 manually moving the robot using a keyboard *joystick*,
 964 students disambiguate the locomotive limitations of the
 965 robot from the behavioral limitations of their programs.

966 The UI is subdivided into multiple windows, both for
 967 screen real estate adaptability and to logically separate
 968 functionality so that each individual form of human-

robot interaction is focused and simple. At the control
 969 level, the UI enables the student to drive the Trikebot
 970 directly, control the head’s pan/tilt position and dump
 971 images from CMUcam. During each of these control
 972 operations, the interface displays and continuously up-
 973 dates the same sensor values that students use dur-
 974 ing programming: motor speed and current values and
 975 rangefinder distance readings. Coupling this sensor
 976 feedback to the teleoperation screen further reinforces a
 977 student’s ability to operate the Trikebot from the robot’s
 978 point of view, observing and reacting to sensor mea-
 979 surements directly. 980

The *Tracking* window within the Trikebot UI
 981 (Fig. 11) enables students to launch, observe and mod-
 982 ify the same high-level visual tracking routines in the
 983 Trikebot’s iPAQ that they can use programmatically.
 984 This window is critically important when using CMU-
 985 cam because it enables students to select, test and fine-
 986 tune vision parameters to ensure that the Trikebot will
 987 successfully track its visual targets. 988

The Trikebot UI was implemented outside of any
 989 high-overhead IDE, ensuring that the finished product
 990 can be compiled and executed using simple command-
 991 level calls in Java 1.4 or beyond. This ease of compil-
 992 ability is key to the *User Controls* window that is also
 993 part of the UI (Fig. 10). This window provides the stu-
 994 dent with a series of buttons and input/output textfields
 995 so that, without spending any time on GUI develop-
 996 ment, the student can launch their programs, observe
 997 Trikebot state during program execution and even halt
 998 their programs from the UI. This coupling of the teleop-
 999 eration and control UI to the buttons and fields used to
 1000 interact with student code is a critical aspect of the suc-
 1001 cess of the Trikebot as an educational, programmable
 1002 robot. The complete JAVA source fileset is available at
 1003 TRIKEBOT (2003). 1004



Figure 10. The Trikebot laptop UI.



Figure 11. The CMUcam Tracking window enables iPAQ visual tracking code to be launched.

1005 **3.3.2. Programming Interface.** Although the JAVA
 1006 client for the Trikebot UI spans a large number of source
 1007 files, the *User Controls* panel is implemented as a sepa-
 1008 rate source file. In order to change the labels of buttons
 1009 and text fields, students modify only a single contigu-
 1010 ous block of one file. In order to write the JAVA func-
 1011 tions that are triggered when those buttons are pressed,
 1012 the students modify a second contiguous block in one
 1013 other file (Fig. 12). Students are thus able to program
 1014 the Trikebot by making direct modifications to two files
 1015 using a text editor such as JEXT (2003), then compil-
 1016 ing and executing from a command line using javac and

java. This programming process removes the complex- 1017
 1018 ity of teaching students how to use an elaborate IDE 1019
 such as FORTE. 1020

The use of JAVA as the programming language of 1021
 choice deserves some discussion. In previous work the 1022
 authors have taught robot programming using LISP, C, 1023
 C++ and JAVA. The most effective language among 1024
 this list was LISP, not only because of its functional 1025
 nature, similar in spirit to more recent robot program- 1026
 ming languages such as GRL (Horswill, 1999), but also 1027
 because of the existence of a Listener Window and, 1028
 thus, the interactive ability to call any parameterized, 1029

```

private void Action1() //dumb wander
{
    int refreshes = 0;
    String Debugging;
    theWindow.quit = false;
    // Set up anything before the control loop
    while (theWindow.quit == false)
    {
        trikebot.RefreshState();
        //get the state variables
        if(trikebot.state.Range() <= 150) {
            trikebot.Drive(20,0);
            //wander forward at a slow speed
        } else {
            trikebot.KillMotor();
        }
    }
    // Do any needed cleanup
}

```

Figure 12. An example of a student code fragment from the summer 2002 course.

1029 defined function at will. This ability to execute a por-
 1030 tion of the robot code in order to diagnose surprising
 1031 robot behavior is extremely important in robotics, and
 1032 this same purpose is served somewhat by the Trikebot
 1033 UI's teleoperation capabilities.

1034 The least effective languages in robot programming
 1035 tend to be those which open up the field of possible
 1036 programming errors unrelated to the robot. For this
 1037 reason, C is particularly poor due to the virtually un-
 1038 bounded ability of a novice programmer to wreak mem-
 1039 ory space havoc via innocently written C code. JAVA
 1040 serves as a practical, modern compromise in that it of-
 1041 fers greater safety than C in a package that is relatively
 1042 popular as compared to languages such as PASCAL and
 1043 LISP.

1044 4. Educational Analysis Methodology

1045 We assessed impact of the course experience on two
 1046 levels. First, we conducted a broad evaluation of all
 1047 the students' experiences in Robotic Autonomy. This
 1048 evaluation was intended to provide both formative and
 1049 summative information about whether the course was
 1050 connecting with students at the appropriate level and
 1051 making progress toward the broad instructional goals.
 1052 Second, we conducted an in-depth study of one week
 1053 of the course. This study, focusing on the experience of
 1054 two teams of students, was intended to identify some of
 1055 the micro-genetic mechanisms of learning that might
 1056 inform patterns of change described in the broader
 1057 evaluation.

4.1. Data Collected: Whole-Course Evaluation 1058

At the broadest level, four classes of data were used to
 1059 evaluate the educational effectiveness of the Robotic
 1060 Autonomy class. First, students completed anonymous
 1061 surveys about what they were learning throughout the
 1062 course. On the first day of class, students completed
 1063 an initial survey of 14 questions covering their techno-
 1064 logical backgrounds, their expectations for what they
 1065 would learn in the course, and their plans for college
 1066 and beyond. Each Monday throughout the course, stu-
 1067 dents also completed a written survey asking them to
 1068 reflect on the prior week's activities. Students rated
 1069 their team's performance, described any discoveries
 1070 they had made or hard problems they had encountered,
 1071 and indicated how useful they had found specific course
 1072 activities. During the last week of class, students com-
 1073 pleted a final survey that included similar content to the
 1074 initial survey, but also asked specific questions about
 1075 whether and how students had learned about the core
 1076 themes and content of the course. The survey forms can
 1077 be downloaded at RASC (2003). 1078

Second, in addition to the weekly written feedback,
 1079 an on-site ethnographer conducted on-camera inter-
 1080 views with each team. These interviews usually lasted
 1081 about ten minutes and were flexible in format. The
 1082 teams were asked about their progress on the assign-
 1083 ments and whether anything particularly notable had
 1084 occurred that week. A total of 9 hours of weekly team
 1085 interviews were collected, with approximately 1 hour
 1086 of interview time per team. Interviews were conducted
 1087 at different times throughout the week, although an
 1088 attempt was made to do most of the data collection
 1089 mid-week. 1090

Third, students were required to open-source and
 1091 document their challenge programs on the class web-
 1092 site. The format included an explanation of what the
 1093 program did and how to use it, an analysis of its perfor-
 1094 mance and limitations, suggestions for future improve-
 1095 ments, and photographs and videos of the robot per-
 1096 forming *in situ*. Each team created seven open-source
 1097 robotics websites to fulfill this requirement. Also asso-
 1098 ciated with each weekly challenge was a grade assigned
 1099 by the instructor using both quantitative and qualitative
 1100 grading criteria. The student documentation and grades
 1101 enabled us to analyze the "output" of student learning
 1102 over the span of the course. 1103

Finally, after completion of the course, follow-on
 1104 data was collected in the form of monthly online sur-
 1105 veys (RASC, 2003). These surveys asked students 1106

1107 about their attitudes toward robotics, science, and en-
 1108 gineering; their activities with respect to robotics over
 1109 the past month; and their future robotics and career
 1110 plans. In the first 6 months following the end of class,
 1111 monthly surveys were consistently collected from more
 1112 than two-third of course graduates.

1113 4.2. *Data Collected: One Week in-Depth Evaluation*

1114 In addition to the overall evaluation of the Robotics
 1115 Autonomy class, an intensive, one-week study of two
 1116 of the nine teams was conducted to develop a more de-
 1117 tailed description of the learning and problem solving
 1118 that occurred in the course on a minute-to-minute basis.
 1119 The in-depth study focused on the fifth week of class.
 1120 This week was particularly interesting because teams
 1121 had mastered the basics of working with the robots and
 1122 were, for the first time, learning how to work with true
 1123 robot autonomy. Prior to the fifth week, students used
 1124 remote control and ded-reckoning to navigate the robot.
 1125 In week five, the core problem for students was how to
 1126 enable the robot to do its own navigating through color
 1127 tracking. Based on his experience teaching robotics,
 1128 the instructor considered this transition to autonomous
 1129 navigation to be one of the hardest challenges for stu-
 1130 dents to overcome.

1131 Out of the nine teams in the Robotic Autonomy
 1132 course, we chose to follow two teams—one all fe-
 1133 male group, Powerpuff Girls, and one all male group,
 1134 Snagglepuss. We purposely did not choose the high-
 1135 est or lowest performing groups, aiming instead for
 1136 groups who were making progress but were still likely
 1137 to face substantial challenges in making the transi-
 1138 tion to working with adjustable autonomy. We based
 1139 our selection of the two groups on the students' on-
 1140 line descriptions of their challenge programs, weekly
 1141 team video interviews, and teacher opinions of the
 1142 teams. The Powerpuff Girls were chosen over the other
 1143 all female group, the FemmeBOTS, because Femme-
 1144 BOTS contained a college freshman majoring in Elec-
 1145 trical and Computer Engineering and it was thought
 1146 that she might provide a disproportionate advantage.
 1147 The instructors also thought that the Powerpuff Girls
 1148 worked together more effectively as a team. Snaggle-
 1149 puss was chosen because the team had a good group
 1150 dynamic and also appeared to be very creative. All
 1151 three members of Snagglepuss and two members of
 1152 Powerpuff Girls attended the Robotic Autonomy pro-
 1153 gram through scholarships from National Hispanic
 1154 University.

Each team spent approximately four hours a day en- 1155
 gaged in group work leading up to the contest and chal- 1156
 lenge problems. The one-week ethnographer video- 1157
 taped these problem-solving sessions. As there was 1158
 only one ethnographer, every moment the group spent 1159
 together was not recorded. However, each group was 1160
 videotaped for about 10 hours, including several two to 1161
 three hour problem-solving blocks. No set schedule of 1162
 data collection was followed; a team was videotaped 1163
 until they seemed to come to the end of a problem solv- 1164
 ing session or were all working independently. Snag- 1165
 glepuss frequently divided the problem into parts and 1166
 worked independently more often than did the Power- 1167
 puff Girls. Also one member of Snagglepuss was ab- 1168
 sent for medical reasons for two and a half days of the 1169
 five day data collection. Class lectures during the focus 1170
 week were also videotaped. 1171

To support the interpretation of the tapes, the ethno- 1172
 grapher wrote nightly reflections detailing her impres- 1173
 sion of the day's activities and how students worked 1174
 together as a group. Each reflection began with a gen- 1175
 eral impression about how successful the day had been 1176
 for the class as a whole. Then, for each team, the ethno- 1177
 grapher recorded impressions of the team as a whole, 1178
 and then each member of the team individually. In con- 1179
 structing these interpretations we explicitly sought to 1180
 expand on areas that would help to interpret the activity 1181
 she had recorded, aided by written notes that she had 1182
 taken while videotaping. 1183

4.3. *Development of Learning Themes and Definitions of all Six Themes* 1184 1185

In order to facilitate the evaluation of learning in the 1186
 students, it was important to partition expected learning 1187
 into a set of learning themes for which data would then 1188
 be quantitatively coded. We hypothesize that six learn- 1189
 ing themes were particularly well suited to the learn- 1190
 ing taking place in an interdisciplinary program such as 1191
 Robotic Autonomy. The themes chosen were: Mechan- 1192
 ics, Programming, Teamwork, Problem Solving, Robot 1193
 Point of View (Robot POV), and Self-Identification 1194
 with Science and Technology (ID with Technology). 1195
 The first two themes, Mechanics and Programming, 1196
 encompass obvious lessons garnered from direct inter- 1197
 action with building and programming robots. 1198

The remaining four themes represent important 1199
 additional opportunities for learning. These themes 1200
 (Teamwork, Problem Solving, Robot POV, ID with 1201
 Technology) represent the types of broader learning 1202

1203 goals popular in curriculum design. Although popular as design goals, such broad categories rarely yield demonstrable gains, particularly in short-term programs such as Robotic Autonomy.

1207 **Mechanics**

1208 *Sensors, motors, iPAQ, back-EMF, wiring, Trikebot, etc.*

1209

1210 Mechanics embodies the interrelationship between various kinematics substructures of the robot and the kinematics of the overall robot. This includes an understanding of mechanical components and the manner in which all these components function together as a deterministic whole system. Basic mechanisms (servos, motors, chassis, suspension, bearings) and electronics (motor controllers, micro-processors, range-finding sensors, the vision system, the iPAQ) comprise this category. Because Robotic Autonomy students began the course by constructing the Trikebot rover using a fast-build kit, we hypothesized significant learning in the area of Mechanics, particularly in the early weeks of the course.

1224

1225 **Programming**

Java, debugging, documenting, compiling, etc.

1226

1227 Programming includes learning how to write commands and scripts that control the robot using, in this case, the Java programming language. The programming skills learned extend well beyond robotics, encompassing code generation/code writing, debugging, documenting, and commenting. Because the Robotic Autonomy challenges posed to the students were primarily challenges for the behavior of the Trikebot, we anticipated that a great deal of the direct learning with respect to overcoming daily challenges would fall in the category of Programming.

1238

1239

1240 **Teamwork**

Communication, importance of teamwork, etc.

1241

1242 Learning how to work effectively in teams is a crucial ingredient for success in many endeavors. Specific skills within teamwork include generating and vetting new ideas; assigning roles and responsibilities; and co-constructing knowledge through observation, imitation, conversation and other socio-cognitive processes. Thus learning progress relative to teamwork would be an important focus of any educational evaluation. In Robotic Autonomy

all students worked in teams of three on every phase of project completion. The Robotic Autonomy teams were formed in the first week and left intact throughout the seven-week curriculum.

1254

1255

1256

Problem Solving

Patience, perseverance, learning a new method of problem solving, etc.

1258

Robots such as the Trikebot are extremely complex machines. As such, the process of understanding and refining solutions using the Trikebot requires mastery of problem solving methodologies. Such skills include developing effective strategies for solving the problems that arose throughout the course: setting appropriate subgoals, using feedback from the robot to effectively identify weaknesses in current strategies, knowing when to abandon ineffective approaches, etc.

1268

1269

Robot Point of View

Autonomy, integration of hardware and software, control of robot with programming, robot diagnosis, etc.

1273

This relatively focused learning theme relates to a critical skill in the understanding of a robot's operating sphere of influence. Robots are extremely limited, in that their sensory and effectory systems are highly constrained relative to that of a human. By *robot point of view* we mean the ability to "see" through the robot's eyes and thus understand the sensor limitations and action constraints under which the robot must operate. It is only by assuming an appropriate robot point of view that a robot designer can begin to discern the space of possible behaviors that are feasible from those that are impractically ambitious.

1286

1287

Self-Identification with Science and Technology

Self-confidence, robotics community, career/experience, ethics/open sourcing, etc.

1290

This extensive learning theme encompasses broad empowerment with respect to science and technology. This includes developing an interest in technology, confidence in one's ability to work with technology, and interest in pursuing education and future careers in science and technology. In short, this theme considers students coming to see themselves as people who enjoy and are capable of technological explorations.

1299

1300 4.4. Theme Coding Process

1301 Two reviewers collaborated to code the learning
1302 themes. Each of the six themes was divided into gen-
1303 eral and specific subcategories. For example, for the
1304 Programming learning theme, a response that simply
1305 said “programming” would be put in the General Pro-
1306 gramming subcategory, while a response that said “pro-
1307 gramming in Java” would be coded under the specific
1308 subcategory of Java or Other Programming Language.
1309 The following written survey questions were coded
1310 for the six learning themes:

1311 Initial Survey:

1312 *What is this course about?*

1313 *What made you want to take this course?*

1314 *What do you expect to learn in this course?*

1315 Final Survey:

1316 *Five things I learned from this course were:*

1317 *What was your favorite part of this course? Why?*

1318 *What was the one thing you most wanted to change
1319 about this course? Why?*

1320 *Please describe three plans you’ve made to work
1321 with your Trikebot.*

1322 *Please write any additional comments that you have
1323 for us.*

1324 Weekly Surveys:

1325 *This week I made a big discovery or leap. (yes or no)*

1326 *What was it?*

1327 *There was something that took me a long time to get*

1328 *or that I missed. (yes or no) What was it?*

1329 What students would change about the course and the
1330 additional comments were initially coded but were not
1331 used in the final learning theme analysis, because the
1332 majority of the responses were unrelated to student
1333 learning. For example, most of the additional com-
1334 ments were about how much the students liked the
1335 class, professor, and teaching assistants. The major-
1336 ity of the responses about what students would change
1337 said “nothing” or were a comment on a specific course
1338 challenge or contest. These two questions were how-
1339 ever used for overall evaluation of the course.

1340 Of the 452 responses coded in the Initial, Fi-
1341 nal, and Weekly surveys, only 5 did not fit into the
1342 learning themes. That 98.9% of the responses fit the
1343 learning themes supports the validity of the coding
1344 scheme.

1345 Once the themes were coded we calculated the pro-
1346 portion of times each student said each specific cate-
1347 gory. The formulas are below:

Initial Survey 1348

Teamwork and Problem Solving: Number of times 1349
mentioned in “What is the course about” and “What 1350
do you expect to learn” questions \div 2. 1351

*Programming, Mechanics, ID with Technology, and 1352
Robot POV:* Number of times mentioned in all three 1353
initial questions \div 3. 1354

Final Survey 1355

Teamwork and Problem Solving: Number of times 1356
mentioned in “Five things learned” and “Favorite 1357
part of class” questions \div 2. 1358

*Programming, Mechanics, ID with Technology, and 1359
Robot POV:* Number of times mentioned in “Five 1360
things learned”, “Favorite part of course”, and 1361
“Three plans for your Trikebot” questions \div 3. 1362

The same proportions were calculated for the whole 1363
class using first a sum of the total mentions of a theme 1364
and then a count of the number of students who men- 1365
tioned a theme. Since the sums and counts turned out 1366
to be very similar, counts were used for the statistics 1367
so that the percentage of students that said something 1368
could be extrapolated. ANOVAs for each subcategory 1369
were run. Few differences were seen, so we ran theme 1370
totals (collapsing all categories) as well as specific 1371
theme vs. general theme. 1372

5. Whole Course Evaluation Findings 1373

To describe student experiences in the course, we first 1374
present analyses of the initial surveys, weekly surveys, 1375
and final surveys. The surveys were used in two ways: 1376
to track the success of the course, and also to track what 1377
students thought they were learning about each of the 1378
six core themes in the course. 1379

5.1. Overall Success 1380

In terms of success, responses indicated that the course 1381
kept the students’ interest and that the curriculum se- 1382
quence was effective. Every week students were asked 1383
to anonymously rate how much they enjoyed the week 1384
on a scale of 1 to 5, with 5 being the highest. All weeks 1385
except for the fifth week were given a mean rating of 1386
4 or above. Ratings for the fifth week, which was the 1387
week when autonomous navigation was presented, av- 1388
eraged 3.4. Consistent with the overall ratings of en- 1389
joyment, students found the contests and challenges 1390

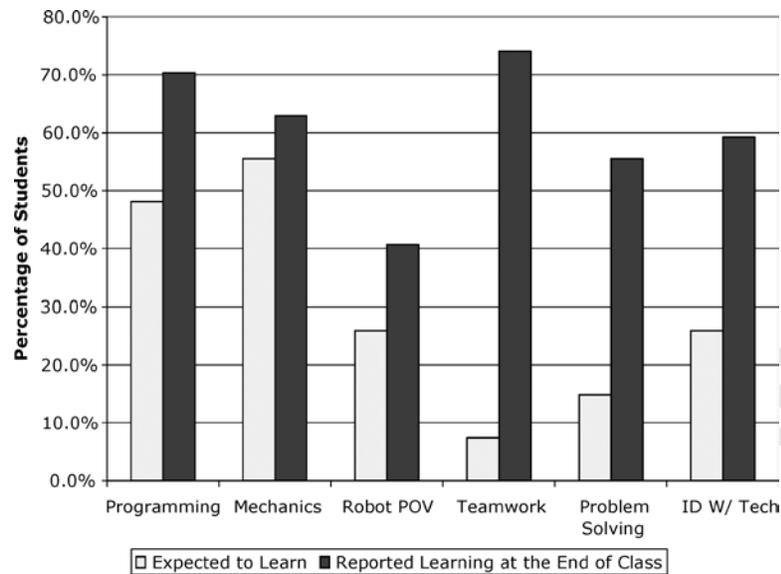


Figure 13. Student self-reports of learning opportunities for each of the core themes in the course. Students were coded for a theme if they mentioned it at least once in response to the survey question.

1391 to be increasingly motivating and engaging. On each
 1392 weekly survey, students were asked whether the chal-
 1393 lenges and contests for that week were their favorite
 1394 so far in the course. At least 33% of the students each
 1395 week reported that it had been their favorite week thus
 1396 far. As the course progressed, students consistently re-
 1397 ported high mean levels of learning each week (3.7 and
 1398 above).

1399 On the final survey student responses also suggested
 1400 that they had been engaged appropriately by the over-
 1401 all course experience. Students rated instructor effec-
 1402 tiveness at a mean of 4.9 on a 5-point scale. Students
 1403 thought the pacing of the course had been appropriate,
 1404 rating pacing at 3.6 on a scale from 1 (“Too Slow”)
 1405 to 5 (“Too Fast”). The guest speakers were appreci-
 1406 ated (4.7 out of 5) with every student agreeing that
 1407 speakers should be included if the course is taught
 1408 again.

1409 When asked on the final survey what should be
 1410 changed about the course when it is offered again, 11
 1411 of the 27 students said that nothing should be changed,
 1412 6 students wanted the course to be longer or cover more
 1413 material, and 5 students gave random responses, such
 1414 as the course should be held at a better location. Only
 1415 5 students wrote down a specific course criticism, for
 1416 instance that a certain contest should be redesigned or
 1417 that the course should have allowed more mixed gender
 1418 student teams.

5.2. Learning the Core Themes 1419

We first asked the question of how students’ under- 1420
 standing of their own learning changed from the begin- 1421
 ning to the end of the course. Students’ expectations for 1422
 their learning of each of the six themes were coded from 1423
 their responses to the initial survey question: *What do* 1424
you expect to learn in this course? On the final survey, 1425
 students understanding of their learning of each of the 1426
 themes was coded from their responses to a question 1427
 that asked them to list the main things they had learned 1428
 in the course. 1429

As shown in Fig. 13, students developed different 1430
 ideas about learning opportunities from the beginning 1431
 to the end of the course. First, consider what students re- 1432
 ported about the three themes that are the most specific 1433
 to the technical aspects of robotics. At the beginning 1434
 of the course, 56% of students expected to learn about 1435
 Mechanics while, at the conclusion of the course, 63% 1436
 reported Mechanics as one of the important things they 1437
 learned. Similarly, 48% of students expected to learn 1438
 about Programming and 70% reported that they had, 1439
 in fact, done so. These findings do not strike us as re- 1440
 markable; after all, a course about autonomous robots 1441
 would certainly include the mechanical and program- 1442
 ming aspects common to all robotics. 1443

What are more interesting are the larger differences 1444
 seen in self-reported learning of Teamwork, Problem 1445

1446 Solving, and ID with Technology. While 7% of the stu-
 1447 dents initially expected to learn about Teamwork, that
 1448 theme turned out to be the most commonly reported
 1449 learning outcome at the end—74% of the students listed
 1450 it as something they had learned. Similarly, Problem
 1451 Solving and ID with Technology were commonly re-
 1452 ported as learning outcomes at the conclusion of the
 1453 course, although they had been infrequently mentioned
 1454 as possible outcomes at the beginning. These findings
 1455 suggest that the course was successful at meeting the
 1456 deeper goals of developing domain-general interest and
 1457 skills that would prepare students for success in broader
 1458 technology and science education in college.

1459 A caveat deserves mention regarding the results
 1460 shown in Fig. 13. The initial survey question preambled
 1461 a single blank block for an answer; and therefore many
 1462 students responded with a single learning expectation.
 1463 The final survey offered five blank lines for answers
 1464 to the same question, and therefore students always re-
 1465 sponded with many themes. Although this structural
 1466 difference has impact on the absolute response fre-
 1467 quency, distribution data across themes is informative;
 1468 it is in this change in distribution that the increased em-
 1469 phasis on Teamwork, Problem Solving and ID w/ Tech
 1470 can be seen.

1471 In addition to coding whether students mentioned
 1472 learning opportunities for each of the themes, we coded
 1473 relevant questions from the initial and final surveys to
 1474 track how much specific detail students reported when
 1475 they described learning opportunities around specific
 1476 themes. Although students mentioned Mechanics and
 1477 Programming a similar number of times in the ini-
 1478 tial and final surveys, they provided significantly more
 1479 specifics about each theme on the final survey. For
 1480 instance, while students mentioned vague statements
 1481 about “robot technology” on the initial survey, they
 1482 were more likely to mention specific technologies such
 1483 as “IR sensors” or “back-EMF” on the final survey, $F(1,$
 1484 $52) = 5.47, p < .05$. While they mentioned “learning
 1485 to program” on the initial survey, they were more likely
 1486 to talk about “states in programming” or “Java” on the
 1487 final survey, $F(1, 52) = 8.61, p < .01$. Thus, student de-
 1488 scriptions of their own learning became more specific
 1489 and grounded in the curriculum content.

1490 How students talked about the themes of Teamwork
 1491 and Problem Solving also changed to include more
 1492 specifics by the end of the course. Students originally
 1493 said they would “learn teamwork” or “work in teams of
 1494 three”. In the final surveys comments like “Teamwork
 1495 is hard especially with varying levels of skill and dif-

ferent personalities, [it] can be rewarding only through
 compromise” and “teamwork leads to victory” were
 more common, $F(1, 52) = 15.91, p < .001$. Similarly,
 from a few general statements about “learning how to
 solve problems” on the initial survey, student state-
 ments changed to specific observations such as learn-
 ing to “really pay attention to what I am doing and try
 to solve it first before asking for help”, $F(1, 52) =$
 $12.00, p < .001$.

We now turn to an analysis of the weekly surveys
 students completed each Monday. Two of the key ques-
 tions on the survey asked students to reflect on whether
 they had, in the preceding week, made a breakthrough
 or discovery and whether they had struggled to under-
 stand anything. Responses for all weeks and students
 were summed for analysis. There was a possibility for
 162 responses to each question, but not every student
 reported a struggle and breakthrough every week. For
 all six surveys given there were 51 reported struggles,
 between five and thirteen per week, and 87 break-
 throughs, between nine and seventeen per week.

As shown in Fig. 14, student struggles were mostly
 around two themes: Programming and Mechanics. This
 is not surprising, because those topics are most directly
 tied to the challenges. Typical responses are shown
 below.

- “Our program had a bug which turned out to be a miss-
 ing zero.”
- “There were long time delays between commands.”
- “Robots need to be tested in the same conditions as
 where they will perform.”

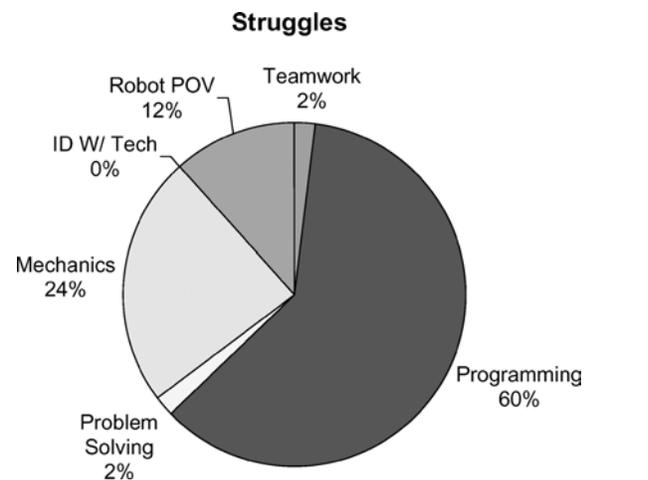


Figure 14. Percent of reported struggles by learning theme.

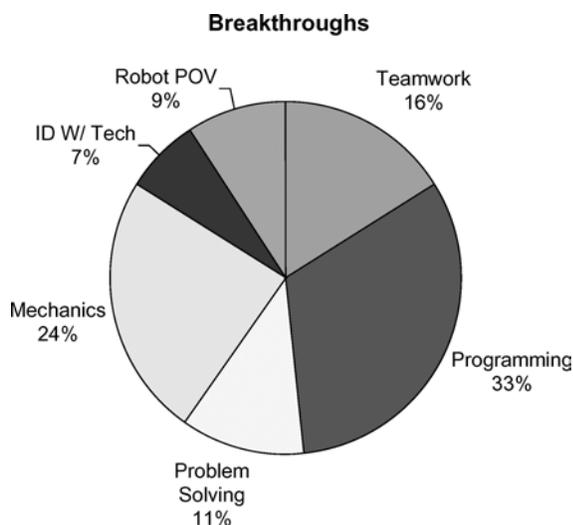


Figure 15. Percent of reported breakthroughs by learning theme.

1527 In contrast, student breakthroughs occurred widely
 1528 among the six themes. Mechanics and Programming
 1529 were still mentioned most often, but breakthroughs
 1530 coded as involving Teamwork, Problem Solving, Robot
 1531 POV, and ID with Technology were also common (see
 1532 Fig. 15).

1533 Examples of these breakthroughs include:

1534 *Programming*: “New programming languages are eas-
 1535 ier to understand than I thought.”

1536 *Mechanics*: “Understanding how sensors are so won-
 1537 derful and yet so error prone.”

1538 *Teamwork*: “The big discovery was that if I try hard, by
 1539 working with my teammates, we could make a lot of
 1540 things happen.”

1541 *Problem Solving*: “Don’t ever leave anything at the end
 1542 or else you will be struggling to finish it on time.”

1543 *Robot POV*: “Robots are babies.”

1544 *ID with Technology*: “I made the discovery that building
 1545 a robot could be very exciting instead of hard.”

1546 5.3. Gender Differences

1547 Finally, we analyzed the student self-report data for
 1548 potential gender differences. Although we began the
 1549 project with no particular expectations that girls and
 1550 boys would have different experiences, we were sen-
 1551 sitive to the historical problem that computer science
 1552 has had in attracting girls to engage in advanced study.
 1553 We were also acutely aware of the fact that the majority
 1554 of the students were boys, all of the outside speakers

were men, and that the instructor and all but one teach- 1555
 ing assistant were men. As the robot course was one 1556
 of the first intensive advanced technology experiences 1557
 for most of the students, we were aware that it had 1558
 the potential to work against or in support of existing 1559
 stereotypes regarding girls and technology. Thus, we 1560
 were particularly interested in whether the experience 1561
 was successful and positive for the eight girls enrolled. 1562

For most of our findings, there were no differences 1563
 between girls and boys, suggesting that the course 1564
 provided a supportive and interesting environment for 1565
 both. We did observe three differences. First, on the 1566
 weekly surveys girls were more likely to report having 1567
 struggled with Programming, $F(1, 25) = 9.12, p = .01$. 1568
 Second, girls also entered the class reporting less con- 1569
 fidence with technology than boys, $F(1, 25) = 9.72, 1570$
 $p = .01$. Third, girls’ confidence with technology in- 1571
 creased more than boys’ by the end of the course, $F(1, 1572$
 $25) = 14.58, p = .001$. Thus, despite our initial con- 1573
 cern, the course appeared to welcome and support the 1574
 participation of girls. 1575

In summary, findings on student reported learning 1576
 suggest that the course was successful in meeting its 1577
 specific instructional goals of teaching the technology 1578
 of autonomy and also its general goals of support- 1579
 ing meaningful student engagement with technology to 1580
 build general interest, skills, and confidence that could 1581
 promote future success with technology education. 1582

5.4. Post-Course Survey Results 1583

Educational evaluation of the Robotic Autonomy 1584
 course has identified learning mechanisms and pat- 1585
 terns within the scope of the seven-week course time- 1586
 line. We implemented a periodic follow-up survey with 1587
 course graduates in order to comprehend the longevity 1588
 of those learning results. The follow-up survey was im- 1589
 plemented as a web-based form sent to all course grad- 1590
 uates once per month for six months. The web form was 1591
 comprised of twenty topic questions, designed to probe 1592
 ongoing self-identification with technology, quantita- 1593
 tive self-reports regarding time spent with the Trikebot 1594
 at home, and future career plans. 1595

Student participation in the monthly survey was ini- 1596
 tially high, with 15 or more results each month for 1597
 months one through four, with a significant drop-off 1598
 in participation thereafter. While the dataset size obvi- 1599
 ates statistical evaluation, some instructive qualitative 1600
 trends and results can still be developed, as discussed 1601
 below. 1602

1603 Self-identification with technology is a significant
 1604 theme based on analyses of Robotic Autonomy learn-
 1605 ing patterns and further because it has potential life-
 1606 long impact on attitudes and focus on technology lit-
 1607 eracy. The follow-up survey scored “I am familiar
 1608 with robotics” and “I am comfortable with technology”
 1609 (among twenty total questions) using five-point scales
 1610 in order to establish the longevity of increased comfort
 1611 and identification with technology during Robotic Au-
 1612 tonomy. Familiarity with robotics consistently scored
 1613 3.60 or higher, with zero survey respondents trending
 1614 lower month over month. Familiarity with technology
 1615 as a whole scored much more highly, 3.90 to 4.60, and
 1616 18% of respondents trended higher month over month.
 1617 The stability of these self-report results is encouraging
 1618 because this suggests that gains in technology con-
 1619 fidence over the course of Robotic Autonomy were
 1620 not sacrificed in the months following graduation. The
 1621 particular strength of “comfort with technology” sug-
 1622 gests that, at the broad topic level, gains are not un-
 1623 dermined following course graduation and may even
 1624 be more tenacious than specific technological lessons
 1625 such as robotics. This is further evidenced by a down-
 1626 ward trend on “I will use my Trikebot at least once
 1627 next month,” which trended down month over month
 1628 for 33% of respondents (the remaining 66% of respon-
 1629 dents reported the same score month over month). As
 1630 the specific robotics tool is pushed to the background,
 1631 we are encouraged that the broad technology literacy
 1632 lesson lives on.

1633 The theme of teamwork was tracked via a general
 1634 question, “I like working in teams” and a specific com-
 1635 munication question, “I will keep in touch with my
 1636 RASC classmates.” Results again showed robustness
 1637 of general learning in spite of narrow loss of interest,
 1638 natural with the passage of time. With regard to remain-
 1639 ing in contact with Robotic Autonomy classmates, 31%
 1640 of respondents trended downward month over month,
 1641 with an average overall score of 3.40. Yet with regard
 1642 to enjoying working in teams, there was no downward
 1643 trend, with an average score of 4.40. While specific so-
 1644 cial relationships with Robotic Autonomy peers fades
 1645 due to the passage of time, we hypothesize that team
 1646 problem-solving skills gained during the course can be
 1647 retained through other activities.

1648 In quantitative terms, average reported hours of
 1649 Trikebot usage per student per month for months one
 1650 through five were: 13.5 hours; 4.8 hours; 7.3 hours;
 1651 8.3 hours; 2.3 hours, respectively. We hypothesize
 1652 that as senior year high school responsibilities grew,

time for robotics exploration decreased by December. 1653
 Yet between 30% and 57% of respondents reported 1654
 participation in other robotics activities each month, 1655
 with a slight upward trend month over month. This is 1656
 once again encouraging because skills acquired during 1657
 Robotic Autonomy, particularly confidence with tech- 1658
 nology, will be of value in enabling participation in 1659
 such projects. 1660

In summary the follow-up survey, while not yield- 1661
 ing statistically significant conclusions, supports the 1662
 contention that lessons learned during Robotic Auton- 1663
 omy are not transient, and that comfort with technology 1664
 and a willingness to participate in technology-related 1665
 projects may be the key long-term benefits of such an 1666
 educational robotics program. A surprising quantita- 1667
 tive result is that each respondent used their Trikebot 1668
 robot at home for an average of 34 total hours in the 1669
 four months following Robotic Autonomy graduation. 1670

6. Conclusions 1671

The overarching goal of this work has been the demon- 1672
 strate end-to-end design and execution of a mobile 1673
 robotics educational course. The educational focus of 1674
 this assessment has been to characterize the impact 1675
 of this hands-on robotics course using formal tech- 1676
 niques. Our prior experiences with robotics education 1677
 suggested that relatively broad forms of learning may 1678
 be demonstrable, and this hypothesis has been vali- 1679
 dated. Learning about the coded themes of Mechanics 1680
 and Programming is to be expected in a robotics course. 1681
 Quantitative results based on self-reports supported this 1682
 expectation. More surprising were large jumps from 1683
 expectation to reported learning along the themes of 1684
 Problem Solving, Teamwork and ID with Technology. 1685
 This suggests that the course was able to meet deeper 1686
 goals of developing domain-general interest and skills 1687
 that can prepare students for broad success in technol- 1688
 ogy and science education. 1689

Coding for the level of detail in student comments re- 1690
 garding learning themes led to statistically significant 1691
 increases in specificity. Significant trends were mea- 1692
 sured for “robot technology,” Programming, Teamwork 1693
 and Problem Solving. These results suggest that stu- 1694
 dents learned concrete lessons for each theme, digging 1695
 below the surface of abstract concepts to a functional 1696
 level of detail. 1697

Evaluation of self-reported struggles and break- 1698
 throughs supported the above conclusions. Student

1699 struggles were reported mainly around two themes:
1700 Programming and Mechanics. But, student break-
1701 throughs were reported across a broad range of themes,
1702 including Teamwork, Problem Solving, Robot POV
1703 and ID with Technology. Once again the inclusion of
1704 non-technological themes reported as breakthroughs
1705 suggests that, during the course, learning extended
1706 beyond the content of technical challenges and into
1707 broader scientific and social lessons.

1708 Finally, analysis of student self-report data for gen-
1709 der differences was intended to identify the effect of this
1710 advanced technology course on existing stereotypes re-
1711 garding girls and technology. Thus a critical question
1712 would be the degree to which Robotic Autonomy was
1713 a positive and successful experience for the girls en-
1714 rolled. Three significant results summarize conclusions
1715 on this query. First, girls were more likely to struggle
1716 with Programming. Second, girls entered the course
1717 reporting less confidence with technology than boys.
1718 But third, girls' confidence in technology increased
1719 throughout the course significantly more quickly than
1720 the boys'. Thus the course appeared to support the par-
1721 ticipation of the girls and was able to compensate some-
1722 what for the initial differences between girls' and boys'
1723 comfort with technology.

1724 Acknowledgments

1725 The Robotic Autonomy program was funded in part
1726 by NASA/Ames Research, National Hispanic Univer-
1727 sity and Intel Corporation. The back-EMF speed con-
1728 trol solution was first proposed by Randy Sargent and
1729 demonstrated at the Swiss EPFL AMR-DST in the
1730 laboratory of Roland Siegwart. Thanks for the follow-
1731 ing individuals who made significant contributions to
1732 Robotic Autonomy: Adriana Cardenas, Daniel Clancy,
1733 John D'Ignazio, Maylene Duenas, Neeti Malhotra and
1734 Raj Reddy. Thanks also to the anonymous reviews and
1735 editorial staff of Autonomous Robots for their valu-
1736 able suggestions that have improved this article signifi-
1737 cantly.

1738 References

1739 Acroname. 2003. Web reference: <http://www.acroname.com>.
1740 ActivMedia. 2004. Web reference: <http://www.mobilerobots.com>.
1741 Archibald, J.K. and Beard, R.W. 2004. Goal! robot soccer for un-
1742 dergraduate students. *Robotics & Automation Magazine, IEEE*,
1743 11(1).

Beer, R., Chiel, H., and Drushel, R. 1999. Using autonomous
robots to teach science and engineering. *Communications of the
ACM*. 1744
Billard, A. 2003. Robota, clever toy and educational tool, Special
Issue on *Socially Interactive Robots, Robotics and Autonomous
Systems*. 1747
Billard, A. and Hayes, G. 1997. Learning to communicate through
imitation in autonomous robots. In *Proceedings of the 7th Inter-
national Conference on Artificial Neural Networks*. 1750
Botball. 2004. Web reference: <http://www.botball.org> 1753
Cooper, M. et al. 1999. Robots in the classroom—tools for accessible
education. In *Proceedings of the 5th European Conference for the
Advancement of Assistive Technology*. 1755
Coppin, P. and Wagner, M.D. 2002. EventScope: A telescience inter-
face for internet-based education. In *Proceedings of the Workshop
on Telepresence for Education, IEEE International Conference on
Robotics and Automation*. 1757
Coppin, P., Morrissey, A., Wagner, M.D., Vincent, M., and Thomas,
G. 1999. Big Signal: Information interaction for public telerobotic
exploration. In *Proceedings of the Workshop on Current Chal-
lenges in Internet Robotics, IEEE International Conference on
Robotics and Automation*. 1761
Druin, A. 2000. *Robots for Kids: Exploring New Technologies for
Learning*. The Morgan Kaufmann Series in Interactive Technolo-
gies, Morgan Kaufmann, pp. 158–164. 1762
Druin, A. and Hender, J. 2000. *Robots for Kids: Exploring New
Technologies for Learning*. The Morgan Kaufmann Series in In-
teractive Technologies, Morgan Kaufmann. 1763
Evolution Robotics. 2004. Web reference: <http://www.evolution.com> 1764
Falcone, E., Gockley, R., Porter, E., and Nourbakhsh, I. 2003. The
personal rover project, Special Issue on *Socially Interactive
Robots, Robotics and Autonomous Systems*. 1766
Fagin, B. 2003. Ada/Mindstorms 3.0. *Robotics & automation mag-
azine, IEEE*, 10(2). 1767
Fong, T., Nourbakhsh, I., and Dautenhahn, K. 2003. A survey of so-
cially interactive robots. *Robotics and autonomous systems, spe-
cial issue, Socially Interactive Robots*, 42(3/4), 143–166, in print. 1778
Fong, T., Nourbakhsh, I., and Dautenhahn, K. 2002. A survey of
socially interactive robots: Concepts, design and applications.
Technical Report CMU-RI-TR-02-29, Carnegie Mellon Univer-
sity, Pittsburgh, PA, USA. 1781
Gage, A. and Murphy, R.R. 2003. Principles and experiences in using
legos to teach behavioral robotics. In *Proceedings of the 33rd
Frontiers in Education Conference*, Vol. 2. 1785
Heer, D., Traylor, R., and Fiez, T.S. 2002. Tekbots/sup TM/: Creating
excitement for engineering through community, innovation and
troubleshooting. In *Proceedings of the 32nd Frontiers in Education
Conference*, Vol. 2. 1788
Hobson, R.S. 2000. The changing face of classroom instructional
methods: Service learning and design in a robotics course. In
Proceedings of the 30th Frontiers in Education Conference,
Vol. 2. 1789
Horswill, I. 1999. Functional programming of behavior-based sys-
tems. In *Proceedings IEEE International Symposium on Compu-
tational Intelligence in Robotics and Automation*. 1790
Hsiu, T., Richards, S., Bhave, A., Perez-Bergquist, A., and
Nourbakhsh, I. 2003. Designing a low-cost, expressive educational
robot. In *Proceedings of IROS 2003*, Las Vegas, USA. 1791
JEXT. 2003. Web reference: <http://www.jext.org> 1800
K-Team. 2004. Web reference: <http://k-team.com> 1801

- 1804 Kolberg, E. and Orlev, N. 2001. Robotics learning as a tool for integrating science technology curriculum in K-12 schools. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 1. 1834
- 1805 Kolberg, E. and Orlev, N. 2001. Robotics learning as a tool for integrating science technology curriculum in K-12 schools. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 1. 1835
- 1806 Kolberg, E. and Orlev, N. 2001. Robotics learning as a tool for integrating science technology curriculum in K-12 schools. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 1. 1836
- 1807 Kumar, A.N. 2001. Using robots in an undergraduate artificial intelligence course: An experience report. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 2. 1837
- 1808 Kumar, A.N. 2001. Using robots in an undergraduate artificial intelligence course: An experience report. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 2. 1838
- 1809 Kumar, A.N. 2001. Using robots in an undergraduate artificial intelligence course: An experience report. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 2. 1839
- 1810 Kumar, D. and Meeden, L. 1998. A robot laboratory for teaching artificial intelligence. In *Proc. of 29th SIGCSE Symposium on Computer Science Education*. 1840
- 1811 Kumar, D. and Meeden, L. 1998. A robot laboratory for teaching artificial intelligence. In *Proc. of 29th SIGCSE Symposium on Computer Science Education*. 1841
- 1812 Kumar, D. and Meeden, L. 1998. A robot laboratory for teaching artificial intelligence. In *Proc. of 29th SIGCSE Symposium on Computer Science Education*. 1842
- 1813 Mansour, R. 2000. Hardware competitions in engineering education. In *Proceedings of the 30th Frontiers in Education Conference*, Vol. 2. 1843
- 1814 Mansour, R. 2000. Hardware competitions in engineering education. In *Proceedings of the 30th Frontiers in Education Conference*, Vol. 2. 1844
- 1815 Mansour, R. 2000. Hardware competitions in engineering education. In *Proceedings of the 30th Frontiers in Education Conference*, Vol. 2. 1845
- 1816 Martin, F., Mikhak, B., Resnick, M., Silverman, B., and Berg, R. 2000. *Robots for Kids: Exploring New Technologies for Learning*. The Morgan Kaufmann Series in Interactive Technologies, Morgan Kaufmann, pp. 10–32. 1846
- 1817 Martin, F., Mikhak, B., Resnick, M., Silverman, B., and Berg, R. 2000. *Robots for Kids: Exploring New Technologies for Learning*. The Morgan Kaufmann Series in Interactive Technologies, Morgan Kaufmann, pp. 10–32. 1847
- 1818 Martin, F., Mikhak, B., Resnick, M., Silverman, B., and Berg, R. 2000. *Robots for Kids: Exploring New Technologies for Learning*. The Morgan Kaufmann Series in Interactive Technologies, Morgan Kaufmann, pp. 10–32. 1848
- 1819 Martin, F., Mikhak, B., Resnick, M., Silverman, B., and Berg, R. 2000. *Robots for Kids: Exploring New Technologies for Learning*. The Morgan Kaufmann Series in Interactive Technologies, Morgan Kaufmann, pp. 10–32. 1849
- 1820 Maxwell, B. and Meeden, L. 2000. Integrating robotics research with undergraduate education. *IEEE Intelligent Systems*. 1850
- 1821 Maxwell, B. and Meeden, L. 2000. Integrating robotics research with undergraduate education. *IEEE Intelligent Systems*. 1851
- 1822 Milto, E., Rogers, C., and Portsmore, M. 2002. Gender differences in confidence levels, group interactions, and feelings about competition in an introductory robotics course. In *Proceedings of the 32nd Frontiers in Education Conference*, Vol. 2. 1852
- 1823 Milto, E., Rogers, C., and Portsmore, M. 2002. Gender differences in confidence levels, group interactions, and feelings about competition in an introductory robotics course. In *Proceedings of the 32nd Frontiers in Education Conference*, Vol. 2. 1853
- 1824 Milto, E., Rogers, C., and Portsmore, M. 2002. Gender differences in confidence levels, group interactions, and feelings about competition in an introductory robotics course. In *Proceedings of the 32nd Frontiers in Education Conference*, Vol. 2. 1854
- 1825 Milto, E., Rogers, C., and Portsmore, M. 2002. Gender differences in confidence levels, group interactions, and feelings about competition in an introductory robotics course. In *Proceedings of the 32nd Frontiers in Education Conference*, Vol. 2. 1855
- 1826 Murphy, R. 2000. *Introduction to AI Robotics*. MIT Press. 1856
- 1827 Nagchaudhuri, A., Singh, G., Kaur, M., and George, S. LEGO robotics products boost student creativity in precollege programs at UMES. In *Proceedings of the 32nd Frontiers in Education Conference*, Vol. 3. 1857
- 1828 Nagchaudhuri, A., Singh, G., Kaur, M., and George, S. LEGO robotics products boost student creativity in precollege programs at UMES. In *Proceedings of the 32nd Frontiers in Education Conference*, Vol. 3. 1858
- 1829 Nagchaudhuri, A., Singh, G., Kaur, M., and George, S. LEGO robotics products boost student creativity in precollege programs at UMES. In *Proceedings of the 32nd Frontiers in Education Conference*, Vol. 3. 1859
- 1830 Nagchaudhuri, A., Singh, G., Kaur, M., and George, S. LEGO robotics products boost student creativity in precollege programs at UMES. In *Proceedings of the 32nd Frontiers in Education Conference*, Vol. 3. 1860
- 1831 Nourbakhsh, I. 2000a. When students meet robots. Essay in *IEEE Intelligent Systems and Their Applications*, 15(6), 15. 1861
- 1832 Nourbakhsh, I. 2000a. When students meet robots. Essay in *IEEE Intelligent Systems and Their Applications*, 15(6), 15. 1862
- 1833 Nourbakhsh, I. 2000b. Robotics and education in the classroom and in the museum: On the study of robots, and robots for study. In *Proceedings Workshop for Personal Robotics for Education*, IEEE ICRA. 1863
- Nourbakhsh, I. 2000c. Property Mapping: A simple technique for mobile robot programming. In *Proceedings of AAAI 2000*. 1837
- Papert, S. and Harel, I. 1991. *Situating constructionism*, In *Constructionism*, Ablex Publishing Corp. 1838
- RASC. 2003. Web reference: <http://www.cs.cmu.edu/~rasc> 1840
- Rowe, A., Rosenberg, C., and Nourbakhsh, I. 2002. A low cost embedded color vision system. In *Proceedings of IROS 2002*. 1841
- Rowe, A., Rosenberg, C., and Nourbakhsh, I. 2002. A low cost embedded color vision system. In *Proceedings of IROS 2002*. 1842
- Schumacher, J., Welch, D., and Raymod, D. 2001. Teaching introductory programming, problem solving and information technology with robots at West Point. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 2. 1843
- Schumacher, J., Welch, D., and Raymod, D. 2001. Teaching introductory programming, problem solving and information technology with robots at West Point. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 2. 1844
- Schumacher, J., Welch, D., and Raymod, D. 2001. Teaching introductory programming, problem solving and information technology with robots at West Point. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 2. 1845
- Schumacher, J., Welch, D., and Raymod, D. 2001. Teaching introductory programming, problem solving and information technology with robots at West Point. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 2. 1846
- Siegrwart, R. 2001. Grasping the interdisciplinarity of mechatronics. *Robotics & Automation magazine, IEEE*, 8(2). 1847
- Siegrwart, R. 2001. Grasping the interdisciplinarity of mechatronics. *Robotics & Automation magazine, IEEE*, 8(2). 1848
- Stein, C. 2002. Botball: Autonomous students engineering autonomous robots. In *Proceedings of the ASEE Conference*. 1849
- Stein, C. 2002. Botball: Autonomous students engineering autonomous robots. In *Proceedings of the ASEE Conference*. 1850
- TRIKEBOT. 2003. Source code download site. Web reference: <http://www.cs.cmu.edu/~rasc/RA/TrikeBackg.htm> 1851
- TRIKEBOT. 2003. Source code download site. Web reference: <http://www.cs.cmu.edu/~rasc/RA/TrikeBackg.htm> 1852
- US FIRST. 2004. Web reference: <http://www.usfirst.org> 1853
- US FIRST. 2004. Web reference: <http://www.usfirst.org> 1854
- Wang, E. 2001. Teaching freshman design, creativity and programming with LEGOs and Labview. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 3. 1855
- Wang, E. 2001. Teaching freshman design, creativity and programming with LEGOs and Labview. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 3. 1856
- Wang, E. and Wang, R. 2001. Using Legos and RoboLab (LabVIEW) with elementary school children. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 1. 1857
- Wang, E. and Wang, R. 2001. Using Legos and RoboLab (LabVIEW) with elementary school children. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 1. 1858
- Wang, E. and Wang, R. 2001. Using Legos and RoboLab (LabVIEW) with elementary school children. In *Proceedings of the 31st Frontiers in Education Conference*, Vol. 1. 1859
- Woloz, U. 2000. Teaching design and project management with Lego RCX robots. In *Proc. SIGCSE Conference*. 1860
- Woloz, U. 2000. Teaching design and project management with Lego RCX robots. In *Proc. SIGCSE Conference*. 1861
- Yim, M., Chow, M., and Dunbar, W. 2000. Robots for kids: Exploring new technologies for learning. The Morgan Kaufmann Series in Interactive Technologies, Morgan Kaufmann, pp. 245–290. 1862
- Yim, M., Chow, M., and Dunbar, W. 2000. Robots for kids: Exploring new technologies for learning. The Morgan Kaufmann Series in Interactive Technologies, Morgan Kaufmann, pp. 245–290. 1863
- Yim, M., Chow, M., and Dunbar, W. 2000. Robots for kids: Exploring new technologies for learning. The Morgan Kaufmann Series in Interactive Technologies, Morgan Kaufmann, pp. 245–290. 1864