Evaluation of Learning in Small Robotics Platforms

M. Farmer, J. Pippine, C. Sullian, C. Valdez, A. Watson

Abstract - **This paper describes the development of techniques used to evaluate the effectiveness of adaptive technology algorithms applied to robotics platforms in the execution of two Defense Advanced Research Projects Agency (DARPA) programs. It also describes how these procedures were modified and improved over the course of 4 years to adapt to continually improving robotic performance. Exploration begins with early multi-team experiments on wheeled platforms exhibiting special recognition by moving through sensed environments. Techniques described include selection of testing scenarios, choice of a cooperative testing scheme, design and maintenance of testing environment, performer/evaluator interactions, and evaluation metrics. These procedures are then transferred to new multi-performer experiments involving legged platforms and modified to meet the needs of this markedly different and more challenging testing environment. Key changes addressed include identification, selection and design of terrain scenarios, shifting from a sensed to a ground-truth environment, moving to a competitive testing scheme, design and construction of a unique testbed, a more streamlined and rigorous performer/evaluator interaction, and changes in focus of evaluation metrics. The key outcome of this process is that it has effectively enabled the Agency to analyze and assess the results from the two sets of experiments as it plans for future robotics programs.**

I. Introduction

Adaptive technologies are, in general, processes that allow a system to adapt to the environment it is in even if it is radically different from the environment the system was initially trained in. In robotics this is a highly valuable trait as it allows autonomous systems to function in a variety of situations without the need for a constant re-supply of user input. The Defense Advanced Research Projects Agency (DARPA) Information Processing Techniques Office (IPTO) has been fostering these types of adaptive technologies under the umbrella description of "learning." Two such programs are Learning Applied to Ground Robotics (LAGR) and Learning Locomotion (L2).

Manuscript received June 16, 2009.
M. Farmer is with System Planning Corporation, Arlington, VA 22201 USA (e-mail: mfarmer@sysplan.com).
J. Pippine is with System Planning Corporation, Arlington, VA 22201 USA (e-mail: jpippine@sysplan.com).
C. Sullivan is with System Planning Corporation, Arlington, VA 22201 USA (e-mail: csullivan@sysplan.com).
C. Valdez is with System Planning Corporation, Arlington, VA 22201 USA (e-mail: cvaldez@sysplan.com).
A. Watson is with System Planning Corporation, Arlington, VA 22201 USA (703-351-8626; e-mail: awatson@sysplan.com).

As critical as the technological advancements put forth by the variety of performers in these programs are, so are the techniques utilized by the government to assess the quality of these advancements. This paper is primarily concerned with these testing and analysis techniques as they were developed in early wheeled platform testing, culminating in the LAGR program, and how they have been adapted and re-applied to the radically different environment of legged platforms, namely the L2 program.

II. Wheeled Platforms

A. Background

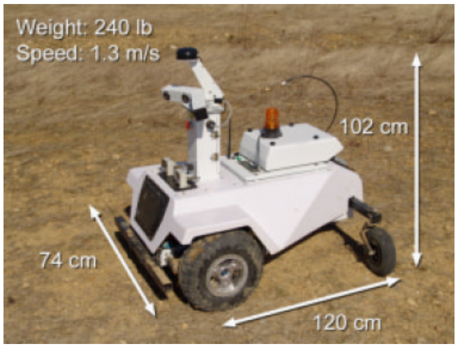
Beginning in 2000, DARPA began investigating off-road ground robotics with the start of the Perception for Off-Road Robotics (PerceptOR) program and the Unmanned Ground Combat Vehicle (UGCV) program. While UGCV studied the development of new unmanned ground vehicles, PerceptOR looked at the development of algorithms to improve the capability of unmanned vehicles driving in off-road, unstructured terrains. From a robotic field testing point of view, this proved to be quite demanding. During the three year effort, the Government held six month-long experiments at four unique test sites. While effective at improving the state of the art by 60-70%, systems would often prove fragile when introduced to a new environment. However, this exposure to new environments was key to producing algorithms that could be deployable. In 2004, as PerceptOR ended, DARPA looked to create a new model for evaluation that would test more frequently in continually changing environments, which led to the LAGR program.

B. DARPA Program: Learning Applied to Ground Robotics (LAGR)

1) Inspiration

The new approach for LAGR was to incorporate adaptive learning techniques into established navigation algorithms to see if this could increase the rate at which progress was being made in the field. This "learning" ability included many approaches, the most important of which was long-range scene interpretation. The initial belief was that long range (defined as outside stereo range) planning could not be done with specific hard coded programs. The desired end result, when paired with learning ability, would be navigation that could use more intelligent abilities, thus pulling 3D information from 2D cues.

2) Platform: HHerminator



The onboard sensor packages included two stereo camera pairs (Point Grey Bumblebees), an infrared proximity sensor, and bumper-activated switches for obstacle impact detection. Navigation capabilities included an XSENS three-axis gyro inertial measurement unit, compass, accelerometer, and a WAAS-enabled Garmin 16A global positioning system receiver.

Mobility governance was set at a maximum speed of 1.3 m/s. Independently driven 24V DC motors supplied power to the front wheels. The rear wheels were two simple casters, which are free to rotate and pivot.

Computational capabilities included one embedded low power VIA processor for low level control such as control and motion of the wheels. Three dual core Pentium M based boards were imbedded onboard with one for each of the two stereo eyes, and the third for route planning. Finally, a gigabit Ethernet card provided the infrastructure for communication between onboard machines, while external communication was handled by a 802.11b wireless router.

The agreed upon standard was to utilize RedHat Linux as the base OS of choice for the project. Although, teams were allowed to install and run the compiler of their choice as well as program in the language of their choice. The primary constraint was that any libraries in use by their code must be contained only on their planner flash disk. This came as an attempt to constrain the teams so that no calls could be made to machines other than the main OS installed on their flash disk. As a result, implemented code had restricted access to the libraries used by the planner code stored on their eye machines.

3) Testing

The overall approach in the LAGR program was to foster cooperation among the research teams. It was decided in the infant stages of the program that the down selects would be purely metric based with no predetermined limits on the number of teams proceeding to subsequent phases. The lack of a competitive down select allowed the teams involved to share ideas amongst themselves to rapidly pass over hurdles that might slow progress in an isolated environment.

Teams were tested in real environments selected by the government team on a monthly basis. One of the most important test strategies was to provide a general "theme" for testing (learn by example, GPS denied environment, etc.) while holding back enough situational information to prevent hard coding for a specific environmental feature set. Teams knew which element of navigation was to be explored during a test but not the specifics associated with the site.

Once the test site was determined, general procedure was to provide the team code with a waypoint file which listed the goal point of that particular test.

All teams were run on an identical course and wherever possible all elements of the course were reset to initial locations. The starting point was reproducible to within 5cm and roughly 5 degrees orientation on each run. Although weather and lighting conditions were outside the realm of testing team control, run times were randomized to prevent any systematic error they might produce. While not on site with us in person, access to the test was made available to teams in the form of multiple live video streams. At any given time during the test, all teams could pull streams from 1-3 on site cameras, all of which had pan, tilt, and zoom capabilities so they could closely follow the vehicle. An additional video stream was provided that showed the output of the operator control station (OCS); the OCS provided information on the real time state of the robot during runs. Teams were also encouraged to call in to a central conferencing number during their specific time slot and discuss their performance with the government team in real time as well as provide insight into what the government team was seeing.

Various test facilities around the country were used, to provide a variety of terrain, texture, and foliage:

- Ft. Belvoir, Springfield, VA

  Primary terrain consisted of sparse grass with medium to large open areas surrounded by thick sets of trees. Used when doing basic tests or when the government team had a specific course modification in mind, such as a man made maze.
- Cold Regions Research and Engineering Laboratory (CRREL), Hanover, NH

  This location provided lush and green vegetation, ideal for poor GPS coverage, challenging elevation changes/grades, and generally below average vehicle traction. Given that one of the techniques teams used to determine vehicle location was observing the wheel motor encoders, the often wet and slippery ground provided unique challenges as well.
- South West Research Institute (SWRI), San Antonio, TX

  This location had a generally aridness with mostly brown vegetation. Trees in the area were generally thin which made them difficult to reliably detect using the robots onboard stereo vision. Although GPS coverage in the area was excellent, there were areas with a thick enough canopy to produce a GPS restricted environment.

- Eglin AFB, Pensacola, FL
  A combination of wide and open tarmacs and sparse to dense vegetation provided a perfect environment to focus on long range planning and obstacle avoidance.
- Ft. Bliss, El Paso, TX
  This location was chosen to examine the effects of degraded traction and motion control on the robot and planning algorithms. The overly sandy and loose terrain exposed limits the vehicle capabilities and flaws in the code used to compensate for slip conditions.

Data was recorded real time to two 40GB onboard log disks on the robot. These logs contained all data from the run, stereo camera feeds, and GPS data. The logs were extensive enough that the run could be duplicated completely in a software environment. The rate at which logs were taken was proportional to vehicle speed; i.e. the faster the robot was moving, the higher the data rate the logs would record at. After the runs the disks were swapped out from team to team and copied to two 1TB storage disks. These disks were then duplicated and a single copy was forwarded to each team. Also, all on site video was recorded during all runs and the video and OCS streams were included on the 1TB storage disks forwarded to every team as well.

4) Evaluation and Metrics

Results were reported as a score that was tabulated by comparing the performance of the performer code to that of a baseline system which was run on the same course and under the same conditions. In our case, the key performance metric was speed, because as a system learned about the terrain it was traversing is should able to maximize it's efficiency in reaching the goal. Three trials were run, with each being capped at ten minutes of runtime due to battery life and logistics of testing numerous performers. The two fastest trials were then averaged together to give an overall time score for a performer.

The baseline software used simple obstacle avoidance and path planning and did not take into account the type of object it was avoiding or do any other scene interpretation. The baseline could "see" objects up to 10m out and plan around them, always trying to get to the goal as defined by GPS.

The performer's software was then loaded onto the same platform that the baseline code was executed on, was given the same start and goal points, and was expected to exceed the baseline's performance by a factor of at least two. This would show the definitive impact that learning had when compared with the baseline system.

5) Summary

A few things were learned from the testing techniques in the LAGR program. Foremost is that interactions in real time with teams during testing are crucial. After action reports are standard while providing moderate feedback, but having teams online when the experiment is happening provided invaluable commentary and analysis from performers and test directors alike. An added serendipitous result of the live teleconferences were that often times asking teams for thoughts on the spot lead to inadvertent breakthroughs. In the first few tests, the video streams and conference calls were considered optional; they very quickly became tools the teams and the government couldn't live without. Secondly, hardware standardization and maintenance became a top priority. With limited sensing capabilities, the vehicle relied heavily on a single part functioning in a predictable manner every run. A minor mechanical shock to the mast of the robot that houses the stereo cameras could cause erratic performance in subsequent runs, causing the teams to search for code based performance issues that just were not there. As the vehicles experience more and more wear, calibration and maintenance schedules became critical. Although, given the size of the vehicle and limited available spare vehicles, this lead to difficulties if there was a mid-test mechanical failure. It was found to be advantageous to cycle robots in and out of service, frequently sending the hardware back to the producer for refurbishing whether defects were known or not. In addition, calibration algorithms were created to promote more accurate camera to robot location transforms. With more frequent calibration, mechanical impacts with the robots weren't as large a concern.

The project concluded in the spring of 2008 after 26 tests over four years. The threat identification and path planning code developed during this effort set a new state of the art in the field and has been transitioned into other unmanned platforms.

III. Legged Platforms

A) Background

For the greater part of the last decade, an increased investigation has been put into the benefits of biped and quadruped locomotion. Made famous by the efforts of Sony's AIBO dogs, Honda's ASIMO, and DARPA's Big Dog prototype, the feasibility of reliable legged vehicles have been shown to be possible. Although legged robotics have many advantages over their wheeled counterparts, they have issues dealing with mobility over terrain while maintaining their balance. After further investigation of these balance and mobility challenges, legged platforms have proven to have higher potential for direct troop support on the ground as the limits of wheeled locomotion limitation are not present.

B. DARPA Program: Learning Locomotion (L2)

1) Inspiration

Even after the successes of the UGCV and PerceptOR programs, two challenges remained for robotics: perception and locomotion. As LAGR focused on long range perception and path planning of a sensed environment and not the mobility of the platform, the other side of the equation was to focus on locomotion and mobility techniques while holding the perception of the terrain constant. The result was the start of DARPA's L2 program.

The driving question through this effort was, "with perfect knowledge of terrain and surroundings, what is the best way to traverse obstacles?"

2) Platform: LittleDog

The LittleDog platform, developed by Boston Dynamics, is a four-legged small walking platform. Each Leg has three degrees of freedom, giving the overall robot twelve degrees of freedom. Each leg contains three electric motors and a three-axis force sensor in the foot. The two Li-ion batteries supply enough power for approximately 30 minutes of continuous use. Top speeds of 25+ cm/s have been demonstrated, using a trotting gait on flat terrain with this platform.

Positioning is determined using a combination of an onboard IMU and a six-camera Vicon motion capture system. IMU and foot sensor data are relayed to a host computer, via 802.11g wireless, where it is combined with the position data from the motion capture system.

All the high-level processing is performed on a Pentium quad core desktop computer. All the performers use the Red Hat OS, and are free to use the programming language of choice, which have included MATLAB, C, and Java. All commands are given to the robot through the API developed by Boston Dynamics, limiting the performers to the amount of control they have over the joints, as well as limiting them to an execution speed of 100 Hz on the robot.

3) Testing

Influenced heavily by the LAGR program, it was determined that the most effective way to evaluate learning systems in a tight timeframe and keep the performers focused, was to do regularly scheduled testing. Like LAGR, a monthly test was scheduled with several intentions in mind: maintain an aggressive technology development, continue to test on a varied assortment and increasingly difficult terrains obstacles, and look for a rapid turnaround to a difficult end of phase metric objectives. Unlike the LAGR cooperative effort, the competitive style of the L2 program limits the open communication between teams. Nevertheless, the L2 program does allow for the continued development of several different styles and approaches.

The end of phase metrics of this testing approach is static and a down select of performer teams occurs for those who do not successfully meet the required performance objectives. With a required down select process, an added pressure is put on the teams to not only develop the best code they can, but also to make significant progress when compared to the other performers. The difference in testing styles has its benefits and drawbacks, but for the structure of the respective programs, each style is appropriate for the sought after goals.

The initial method of test execution was adopted from the LAGR methodology. Initially, each performer would ship a 250GB hard disk drive to the government team a few days before the test. Upon arrival, each hard drive would be installed and booted to make sure everything worked correctly. Once testing was complete, post processing included creating a duplicate image of the drive. This copy was stored at the government team's test site for archival purposes as well as for comparisons of old performances to recent trials. Once imaging was complete, each drive was repackaged and shipped back to the performers. The entire process proved time consuming, and highly inefficient.

An evaluation was made of test procedures and determined a need for successful test executions while efficiently keeping the performer teams remote. The current method of submitting code takes advantage of advanced software revision control programs (i.e. SVN, CVS, etc) coupled with high-speed internet connections and cheap storage. The L2 Government Team (LLGT) now has two hard drives that are kept on site for each performer: a drive used for the open tests (disk A), and a drive that is used for the closed tests (disk B). The day before each monthly test, the LLGT allows the performer to remotely login, modify, and update the software on disk A to the code base they plan on using for the upcoming test. After each team has updated the software, a quick shake-out trial is run. This trial is usually on a simple version of the upcoming terrain, or just across a flat surface, to verify that all the updates have been applied correctly and there are no major differences in the behavior the performer sees on their hardware as compared to the behavior we see at our site with our hardware.

By cutting out the time involved in shipping drives, making images, and other pre- and post-test procedures involved with the LAGR inspired method of software updates, teams have several more days per month to develop their software as well as increased the amount of time for the LLGT to test each month. The LLGT is now able to perform trials for all performers, on three separate terrains, whereas initially only one terrain was attempted in the same time frame.

Phase 1 boards focused on rocky, "real-world" terrain styles. Though several iterations were required before an acceptable difficulty was discovered, the final test featured a board that was designed by fastening real rocks to a MDF base; this resulted in the most natural board at that point in the program. In Phase 2, our focus shifted to geometric terrains (such as rectangular barriers, steps, and slopes) attempting to remove some of the haphazardness involved in the path planning and execution. Currently, Phase 3 has seen a continuation of geometric terrains, made to be much more challenging by increasing the number and variety of natural terrains, making the program more real-world applicable.

During each test, the LLGT collects the automatically generated log files, videos from two high-definition cameras, along with high-speed video as needed. All of the collected data and video are made available to the performers as soon as possible, usually within 24 hours of the test. Test reports are generated by the LLGT and disseminated to the government and all performer teams outlining the purpose of each test, end of phase metrics to be sought after, and current and ongoing team progress.

4) Evaluation and Metrics

Learning Locomotion's key quantitative metric was speed. The final criterion of success for the program is to have an average speed of 7.2 cm/s across several distinct and complex terrains that the performer has minimal prior knowledge of, as well as be able to navigate terrains that are up to 10.8 cm high (e.g. jersey barrier). Since this is a novel platform, and legged locomotion is still in it's infancy, we choose our metric based off of the average speed of a fully laden solider traversing difficult terrain (1 mph) and scaled it down to the leg length of the LittleDog platform.

Evaluation has been done over three phases, each with a different expected metric that increased with the capabilities of the performer. The phase 1 metric was 1.6 cm/s over a single rocky terrain, unseen by the performers, which was achieved by all performers.

For phase 2, many more terrains were generated (7) in many different styles, forcing the performers to create more robust software and greatly increasing the capabilities of the platform. The metric speed was set at 4.2 cm/s, which was determined to be a realistic goal for the performers to reach in the 9-month timeframe while still encouraging progress towards the final metric speed for phase 3. A competitive down select was also incorporated into the evaluation metric for phase 2. Five of the six performers exceeded the metric speed, allowing them to continue work into phase 3.

During phase 3, we have continued the terrain trend started in phase 2, expanding a total of 13 different classifications of terrains, reincorporating more natural terrains. This phase of the program is still on going and no results are available for publication at this time.

5) Summary

Since the beginning of L2, there have been several important lessons that we've learned thus far, the first of which is based in the refinement of the code updates and log storage. After considering the availability of remote updates, online access, and readily available storage devices, shipping physical drives back and forth was determined to be too inconvenient and impractical. Additionally, as enthusiasm builds to test the limits of the robot across varying terrain schemes and geometries, it is important to keep in mind the end of program goals as well as maintain the foresight of scaling such a platform up to a level adequate for war fighter support. As the upper limits of static gaits are reached, the platform physical's limitations have limited breakthroughs in high-speed dynamic behaviors. This is discouraging in the short term, but yields potential for further insight and development in the future for legged robotics.

Currently, the third and final phase of the L2 program is underway. Focusing on more realistic terrains than that of prior phases and coupled with more aggressive speed and height metrics, the code developed out of this program will set a gold standard for quadruped locomotion. As additional legged robotic platforms arise, the developed L2 code has proven robust and adaptive and will be scalable to field-able, larger, platforms.

IV. Collaborative vs. Competitive Testing

Throughout the entire LAGR program and the first phase of Learning Locomotion, to the focus had been on fostering a collaborative effort among the performers. They were encouraged to talk amongst themselves, discuss, and share ideas and software. Starting with phase 2 of Learning Locomotion this was changed, enacting a competitive down select where only the fastest performers were allowed funding for phase 3.

The collaborative testing scheme really shined in LAGR because it not only gave performers the opportunity to work on what they are best at (e.g. long distance planning), but also allowed them to incorporate components that another performer had developed (e.g. stereo object detection). This approach also allowed adjunct performers to join the program late without having to start from scratch since they could go to the other performers and build off their established code base. The biggest downside experienced while testing in such a manner was the lack of motivation for the performers to deliver high caliber software as there was no threat of losing funding for performing poorly or getting outpaced by their peers. Late into the second phase, software was still extremely buggy month after month, and the performers were not taking the extra effort to fix it.

The competitive testing scheme in L2 started in phase 2, following the success of other DARPA programs like the Grand and Urban Challenges. This style of testing forced the performers to compete against each other with only the best moving forward within the program, and penalizing them for submitting poor or underachieving software. It was determined that this idea did just that; the software submitted for each monthly test was highly refined and was making leaps and bounds toward meeting and even surpassing performance metrics. It also led to a wide variety of techniques, since each performer was working inside a vacuum during the majority of the program. The downside to this approach was that it forced the performers to focus too much on the singular metric of speed which led to software that could go fast, but was not necessarily robust or adaptable to other platforms or terrains.

There are definite pluses and minuses to both methodologies of testing, and choosing the appropriate style depends on what is trying to be evaluated and accomplished. In a collaborative test environment there still must exist some motivation to encourage continuing increases in performance and foster technological breakthroughs. For a competitive approach, metrics have to be well defined and ideally multiple metrics should form the score on which the teams are ranked rather than a single metric that forces the performers down one track.

V. Conclusions

Over the course of evaluating two DARPA robotics platform programs, the development of procedures used to evaluate applied learning techniques has continued to evolve. It has optimized team to team and team to government communication, software dissemination, test scheduling, and team evaluation. Starting by addressing limitations of perception and locomotion in autonomous systems, tests began with early multi-team experiments on wheeled platforms, which displayed learning techniques by moving through a number of varied environments. These techniques were successfully translated into multi-team autonomous legged platform experiments with by building upon past test trials via a lessons-learned approach.

The desire to conduct tests every six weeks instead of four, has been expressed by both academic and industry performers. This is generally because the performance requirements have escalated and it is necessary to have additional time to adapt algorithms to new terrains and prepare new features. However, the aggressive test plan has been viewed by the government test team and its sponsor as a success in that it has effectively led to the development of new algorithms in learning and locomotion.

A reliable and proven testing scheme has been established which will serve as the foundation for future robotic platform experimentation, testing, and evolution.

References

[1] L.D. Jackel, E. Krotkov, M. Perschbacher, et al. (2006). The DARPA LAGR program: Goals, challenges, methodology, and phase I results *Journal of Field Robotics*. 23(11-12), 946-793.