RoboCup-99: A Student's Perspective

Jacky Baltes **Nick Hildreth**
Centre for Imaging Technology and Robotics
University of Auckland
Private Bag 92019, Auckland 1, New Zealand
j.baltes@auckland.ac.nz nhil002@ec.auckland.ac.nz

Abstract

One of the reasons for organizing robotic games is that they allow researchers to evaluate their systems and approaches on a level playing field. This evaluation is important in a quickly developing field such as robotics with few real world applications. This paper investigates through a case-study how much participating at the RoboCup-99 competition has benefited a MSc. student at the University of Auckland. Although the participation was certainly stimulating, its influence on the research was indirect. The paper makes a number of suggestions that will make it easier to quantitatively evaluate research at these competitions and thus influence research more directly.

1 Introduction

Robotic games are extremely popular. Apart from their entertainment value, the ability to evaluate research progress and compare one's own approach in a competitive environment against that of other teams is often cited as one of the reasons for organizing competitions.

However, there is only anectodal evidence of the impact of robotic games on research programmes.

This paper describes the experiences of a student, Nicholas Hildreth, participating in the RoboCup robotic soccer competitions and how they influenced his research in path planning for mobile robots.

Section 2 gives a short introduction to the All Botz, the University of Auckland RoboCup team. The adaptive path planner is briefly introduced in section 3. Section 4 discusses how the robotic games were used to support the empirical evaluation of the adaptive path planner. Section 5 summarizes the paper and suggests some ways in which robotic games could have better supported the evaluation of our research.

2 History of the All Botz

Nick's first experience with robotics was when he took the graduate course on "Intelligent Active Vision" at the University of Auckland starting in February of 1998. The course description of this paper reads as follows:

> Control of autonomous mobile agents in a realistic, dynamic, and uncertain environment. It covers a variety of different areas including robotics, planning and machine learning.

The paper was offered for the first time and things were a bit chaotic. We used remote controlled toy cars for the practical work in the paper, since there was insufficient funding to purchase a "real robot." We designed and built parallel port interfaces to the remote control transmitters so that the cars could be controlled from a computer. Position and orientation information for the robots are provided by a global vision system mounted on the ceiling.

It was a real eye opener to see how difficult it was to do even simple things, such as driving a straight line, with these toy cars. However, the course turned out to be very motivating in the end. The culmination of the project was a race amongst different groups of students, the so-called Aucklandianapolis. To complete this challenge successfully, a team of students had to implement a complete mobile robot system including video processing, path planning, and path tracking control. Figure 1 shows our first mobile robot competition.

After the course finished in June 1998, Nick started a project on the design of an agent architecture for a mobile robot. The project resulted in some preliminary work. However, at the same time, we learned about the PRICAI-98 RoboCup competition. Since we had had good success at the Aucklandianapolis

Figure 1: The First Aucklandianapolis Competition

competition, we decided to extend the current architecture to create a team to enter the competition. After having implemented a striker and a goal keeper, we thought we were ready for the competition.

Two weeks before the competition, we found out that our toy cars were too large. However, with luck we managed to buy some used smaller toy cars that operated on the same frequency and could be controlled with the parallel port interfaces that we had built.

After recalibrating the system for the new cars, we managed to be ready for our first international competition. Our team consisted of only two players: a striker and a goal keeper.

At the competition, we realized that our approach to the problem was different from that of other teams. The difference in hardware became apparent. The All Botz were the only team that used non-holonomic robots and a camera that is mounted on the side. All other teams used holonomic robots and mounted the camera directly overhead. However, not all teams were following this trend and had developed their own approach. For example, the CIIPS glory team from UWA, Perth, was the only local vision team at the competition. It was interesting to see the wide variety of approaches. Some teams focuses on the mechanics on their robots, putting a lot of effort into the selection of motors and tires. Other teams focused on the embedded controller and the electronics. The main emphasis for these teams are additional sensors, such as ultrasound, infrared, or even local vision. yet other teams emphasize the cooperation amongst the individual players of a team.

At the competition, we noticed our path planner, an extension of Bicchi's non-holonomic path planner ([1]), did not perform well in the dynamic environment. Our robots would spent most of their time planning, but before the planner would finish, an object would move in the domain and the path planner hat to be restarted.

One of the disappointing aspects of the games were the amount of set plays that were used by some of the teams. Some teams spent a long time manually positioning their robots for free kicks and other stoppages in play. This was disappointing, since our main goal was to advance our research, not to take part in a game of robot chess.

3 Adaptive Path Planning

The experiences at the PRICAI-98 RoboCup competition motivated Nick to work on his Master's thesis and he chose the topic: *"the problem of path planning for car-like mobile robots in highly dynamic environments"* [3]. This section is a brief introduction into the basic idea behind adaptive path planning and the methodology used for evaluating this research.

At the PRICAI-98 competition, we used a version of Bicchi's path planner that was optimized for the RoboCup domain. This planner proved to be too slow for a soccer game. On a Pentium 200MMX running the Linux operating system, the planner would take more than one second to create a path. The level of the competition in Singapore was such that the ball would rarely sit still for more than a second, so that our path planner would spent a lot time replanning. In few cases would it actually start executing a plan.

The main motivation for the adaptive path planner that he developed in his thesis was the realization that:

- Path planning is an expensive operation, so the result of this work should be reused if possible.
- The result or output of a path planner is a path
- Assuming that changes in the domain are small between individual planning episodes, the current plan will be structurally similar to the plan for the new situation.

This motivation is similar to Hammond's case-based planning ([2]) with some important differences. Firstly, case-based planning assumes that a plan database exists with previous plans and that the most similar plan to the current situation can be found by using a similarity metric. Secondly, the database of previous plans needs to be maintained; new plans need to be added so they can be reused in the future or old plans that are not useful must be removed. So, a lot of work on case-based planning focuses on the design of suitable similarity metrics and on database policies.

In adaptive path planning, we assume the existence of a albeit slow static path planner that can be used to create an initial plan. The previous plan is the most similar one to the current situation and that therefore, there is no need to maintain a plan database.

3.1 Path Representation

At the heart of any path planner is the plan representation. A plan consists of a sequence of path segments. Most other path planners use a representation with different segment types such as straight lines, turns to the left, etc. This representation makes it difficult to adapt a path, since the adaptations will need to be specific for a given segment type.

To simplify the adaptation, the adaptive path planner uses a uniform representation for all path segments. Each segment contains the following information:

- start point I
- initial bearing α
- length of the segment L
- radius of the segment R
- time limit to traverse the segment T
- A possibly empty attachment A. An attachment is used to attach an object to a path segment, so that if the object moves, all attached path segments will move as well.

This representation, shown in Fig. 2, proved very useful, because plans in this repesentation can be easily adapted to compensate for movements of objects or goal locations in the domain.

Objects are attached and detached from path segments dynamically. If an obstacle moves too close to a path, the path is split at the closest point to the obstacle and a new segment is inserted which is attached to the object. As the object continues to move the attached segment will move as well. Once the object is too far from the path, the object is detached from the path segment.

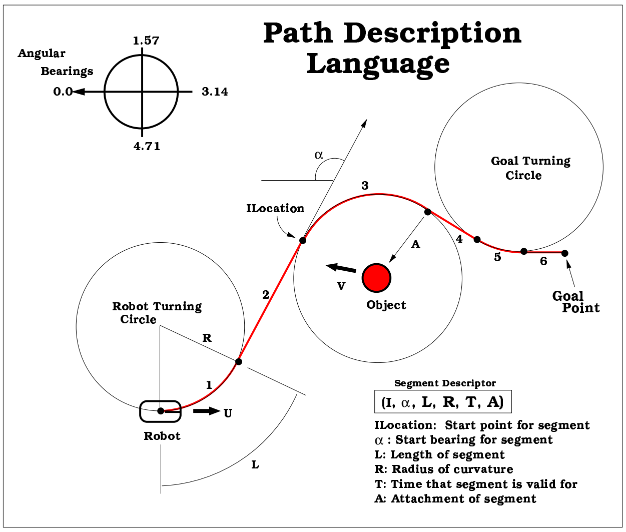

Figure 2: Path Description Language

3.2 Repair Strategies

Any movement in the domain results in attached path segments being moved as well, and thus may create a discontinuity, a so-called disjunction, in the path (see Fig. 3. The assumption is that repairing these disjunctions is cheaper than creating a new path from scratch.

There are two types of disjunctions: distance disjunctions (break in the path) and angle disjunctions (continuous path, but discontinuity in the first order derivative).

The adaptive path planner starts with the largest disjunction and attempts to adapt the plan to fix the disjunction. In this process, new, but smaller disjunctions may be introduced into the path. The repair methods are applied using a standard A^* search algorithm that uses the complexity of the repair as a heuristic function.

To fix these disjunctions, the adaptive path planner contains a library of path repair methods. These path repair methods are highly specific repairs that have a high chance of success and of reducing the size of the disjunction.

The following classes of repair methods are implemented in the adaptive path planner:

- **Positional Adjustment:** Changes in the start and end position and bearing of a segment. By themselves these repair strategies are not very useful, but in combination with others (e.g., shape

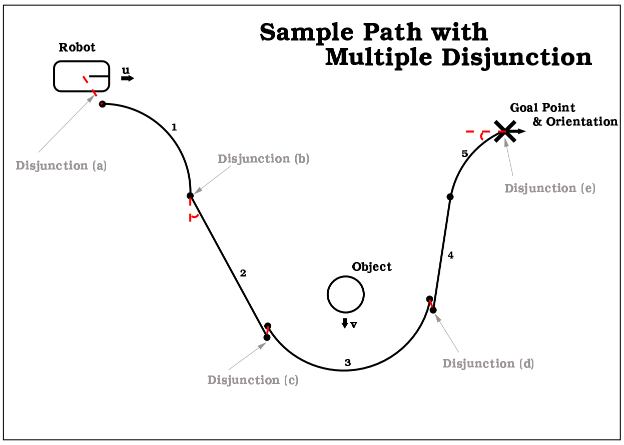

Figure 3: An Object Movement that Results in a Disjunction

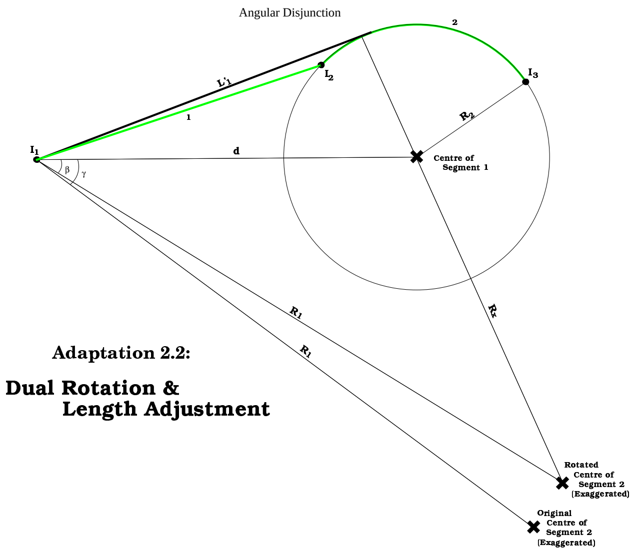

Figure 4: A Shape Adjustment Adaptation

adjustment) they can fix many plans.

- **Shape Adjustment:** These repair methods change the length or curvature of a segment. For exmaple, a tight turn can be converted into a gentler turn.
- **Type Adjustment:** These repair strategies change the sign of the curvature, so a left turn can be converted into a straight line or a right turn. Also, the traversal direction of a circle can be swapped.
- **Segment Structure Adjustment:** These repair methods use segment insertion, segment breaks, and segment deletion to change the structure of the plan.

A simple shape adjusting adaptation is shown in Fig. 4. An angular disjunction is fixed by simultaneously rotating and stretching the line segment and shortening the circle.

3.3 Evaluation Methodology

Although an intuitive argument can be made for the advantages of the adaptive path planner, to quantitatively evaluate the performance of such a path planner is difficult. For example, it is also intuitive that an adaptive path planner will perform poorly if there is no relationship between the current state and the next state of the world.

Firstly, the adaptive part of the path planner is not *complete*, that is there are situations in which the adaptive planner will find a suitable combinations of adaptations to create a valid path, but such a path exists in practice. In this case, the static path planner is called to create a new plan from scratch. This usually occurs when the world state changed dramatically from the current one. The worst case run-time of the adaptive path planner is thus worse than that of the static path planner. The motivation is that for sustained planning, the adaptive path planner can avoid calling the static path planner often.

It is true, that in principle any adaptive path planner that can delete and insert segments into a plan can be converted into a complete path planner. In the worst case, the planner would delete all segments from the path and then start creating a new path from scratch. However, if this is done, the worst case complexity would be similar to that of other static path planners. This would be missing the point of an adaptive path planner completely. One would replace one search problem with another one in a similar space. The main point is that there is a small set of highly specific repair strategies that can adapt a path to the new situation quickly. The success rate of these repair strategies must be high and they planner must be able to apply them quickly.

This argument then relies on a sequence of path planning problems that reflect the dynamics of the application environment.

Through an empirical evaluation of a simulated world and a number of case-studies, the evaluation of the adaptive planner shows that this set of adaptations is sufficient to fix a plan in most cases, yet small enough to not increase the run-time of the path planner significantly. The repair strategies have specific pre-conditions, which when met result in the new path having a high chance of being able to be completed.

However, the method used to create the random path planning problems and case studies is under control of the designer of the algorithm to be tested. Even with his best intentions, it is difficult to prevent subconscious assumptions from entering both the design of the synthetic as well as the path planner. It would be far more convincing to have a standard environment for mobile robots that could be used to compare the different path planning approaches on realistic problems.

There are few real world applications for mobile robots that are convincing and general enough to allow different robots to take part in them. The state of the art in mobile robots is currently a vacuuming robot that can maneuver through an unstructured environment.

Therefore, it was decided that the research should be evaluated at the RoboCup competition. The motivation was that a complex robotic competition is the next best thing to an unbiased environment.

4 Evaluation at RoboCup-99

A prototype version of this path planner was finished just in time for RoboCup-99. It was the intention, that we would use the competition as an opportunity to test the new path planner under real world conditions.

However, this turned out to be much more difficult than expected. The main problem is that every team has only a few games and there was no opportunity for practice games. The first few games were very important, so we used our best path planner. For a proper empirical evaluation, we needed to play two sessions against each team, one with the adaptive path planner and one with a standard path planner.

Another disappointment was that the promised video tapes of our games were not made available, so we had no possibility to view our games afterwards.

Nevertheless, the competition proved valuable, since we noticed some recurring patterns, which the prototype path planner was not able to deal with efficiently. After returning from RoboCup-99, we redesigned part of the path planner and completed its implementation paying attention to the patterns that emerged during the competition.

Figure 5 shows a case study of a situation that was

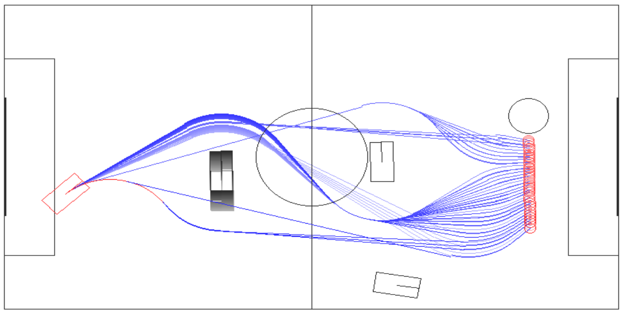

Figure 5: Case Study Derived From Observed RoboCup Competition

inspired by the RoboCup competitions. It shows how the adaptive path planner changes the path to compensate for a moving ball and an interfering object.

Furthermore, Nick gained some important information by talking to other researchers at the competition and at the workshop. For example, Nick learned about practical potential field path planning through informal discussions with other teams. The most impressive performance was that of the Robotis team from Korea, which showed the best control. The Robotis robots moved about three times faster than the other teams.

Flexibility and robustness, obviously very important characteristics for a robot to survive in the real world, have taken a back seat to special purpose solutions. The RoboCup committee suggested that instead of every team mounting their own camera, a high quality camera would be provided by the organizing committee. All teams except the All Botz refused to even try the idea. One team forced the organizing committee to repaint all playing fields, since the center line was 2mm too wide and they could not compensate for this change in their software. Also, most teams did not switch sides at half time, since their robots performed significantly better on one half of the field than on the other. Even during the RoboCup-99 finals, teams remained on their own side.

5 Conclusion

This paper describes a student's involvement in the RoboCup initiative and the influence that the participation at the robotic competitions had on his research.

The main impact that the competition had on his research were two-fold. Firstly, experiences gained in the soccer games provided a better understanding of the path planning problem in real-world situations. Secondly, it gave Nick an excellent opportunity of discussing with researchers that were extremely familiar with the issues.

The robotic games did not provide sufficient opportunity to evaluate quantitatively the performance of the adaptive path planner. It would have been a good idea to setup a special playing field so that teams can hold short practice games.

Nick Hildreth has now completed his MSc. and is considering doing a PhD in Computer Science.

Acknowledgments

This work was supported by AURC grant XXX/9343/3414107 from the University of Auckland.

References

[1] Antonio Bicchi, Giuseppe Casalino, and Corrado Santilli. Planning shortest bounded-curvature paths for a class of nonholomic vehicles among obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1349–1354, 1995.

[2] Kristian J. Hammond. *Case Based Planning*. Academic Press Inc., 1989.

[3] Nicholas Hildreth. An adaptive path planning system for car-like mobile robots. Master's thesis, University of Auckland, February 2000.