A Tool for Mobile Robotics Research

Indratmo[1], Anies Hannawati[2]
[1]Electrical Engineering Department, Faculty of Industrial Technology, Widya Mandala University
e-mail: indratmo@mail.wima.ac.id
[2]Electrical Engineering Department, Faculty of Industrial Technology, Petra Christian University
e-mail: anies@petra.ac.id

Abstrak

Makalah ini memperkenalkan TeamBots sebagai suatu alat penelitian di bidang robotik. TeamBots memungkinkan para peneliti untuk mengembangkan sistem kontrol robotik dan mengevaluasi sistem kontrol tersebut dengan simulasi atau perangkat robot yang sebenarnya. Simulasi memainkan peranan yang sangat penting dalam penelitian di bidang robotik. Interaksi yang terjadi antar robot dalam sistem multi-robot dapat menghasilkan fenomena yang menarik. Sangatlah sulit untuk memperkirakan efektivitas dari suatu sistem kontrol tanpa mensimulasikannya di suatu lingkungan kerja. Makalah ini menyajikan sebuah studi kasus dan membandingkannya dengan penelitian serupa. Hasil eksperimen menunjukkan bahwa perbedaan lingkungan kerja mempengaruhi kinerja suatu sistem kontrol. Sistem kontrol yang terbukti terbaik di lingkungan kerja tertentu belum tentu menjadi sistem kontrol yang terbaik di lingkungan kerja yang berbeda.

***Kata kunci**: simulator, robot, gangguan, sistem multi-agen..*

Abstract

This paper introduces TeamBots as a tool for mobile robotics research. TeamBots enables researchers to develop robotic control systems and test the control systems in simulation as well as on physical robots. Simulation plays important roles in multi-robot research. Interaction among robots in multi-robot systems can produce interesting phenomena. It is hard to predict the effectiveness of a control system without simulating it in a multi-robot environment. This paper presents a case study and compares it to closely related work. The experiment result shows that even for the same task (i.e. foraging) a control system that performs best in a particular environment may not be the best control system in different environments.

Keywords : simulator, mobile robots, interference, multi-agent systems.

Introduction

Analyzing multi-robot systems requires different approaches than those used for analyzing a single robot. A single robot does not interact with other robots. When evaluating a single robot, we only have to focus on the robot's performance in completing the designated task. On the other hand, it is hard to predict the performance of a robotic control system in multi-robot systems. We need to simulate/implement the control system in a multi-robot setting to get and evaluate the results. However, building physical robots is complex and expensive.

To cope with this problem, researchers have developed simulators to facilitate multi-robot research. Simulation is a cost-effective means for evaluating multi-robot systems. Using a simulator, we can test a robotic control system before deploying it to physical robots. These simulators, however, are only suitable for specific domains. We must choose an appropriate tool for our domain.

This paper introduces a software tool for mobile robotics research. We describe the features of this tool and provide a case study to show that many interesting research issues can be examined using this tool in a relatively easy way.

Background

1. TeamBots: The Tool

TeamBots [7] is a software tool for mobile robotics research. It consists of a collection of Java classes for designing and simulating robotic control systems. TeamBots can run control systems on physical robots (e.g., Nomad 150s) and in simulation environments (Figure 1). The

Catatan: Diskusi untuk makalah ini diterima sebelum tanggal 1 Mei 2002. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 2, nomor 2, September 2002.

simulation environments are very flexible. We can create a simulation environment, with walls, obstacles, and targets in it, by modifying an easy-to-understand description file (a text file with extension "dsc"). In addition, the simulation supports multiple robots with different control systems. Thus, we can simulate different control systems in a multi-robot setting and observe the robots' performances.

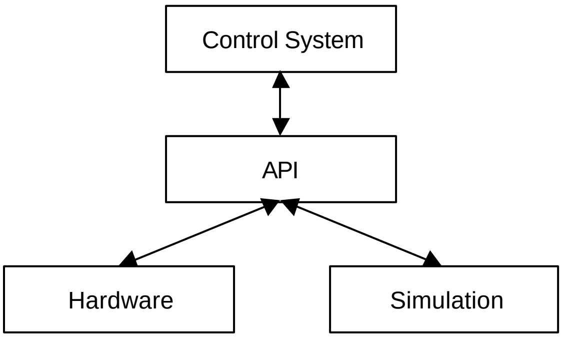

Figure 1. Control systems can run on several supported robot hardware and in simulation through a single API as the interface [2].

Some highlights of TeamBots's features based on JavaBots[2]:

- **Platform independent:** TeamBots is written in Java. Thus, it is not dependent on certain platforms or operating systems. Researchers can use TeamBots in their favorite platforms.
- **Faster development time:** Java has removed some programming features that are known as common bugs in other programming languages (e.g., pointer arithmetic in C/C++). This simplicity helps programmers deliver working code more quickly.
- **Modularity:** TeamBots provides integrated packages that support "reinforcement learning, motor schema-based navigation, vision, hazard sensing and manipulation" [2].

In addition to the features above, TeamBots includes various examples that help its new users get started using the tool. Examples range from a simple foraging task to a complex soccer game [7]. Moreover, TeamBots provides a non-graphic mode simulation that runs faster than the graphic mode for collecting extensive data.

2. Foraging Tasks

A lot of robotics research uses foraging as the experimental domain. A foraging task usually requires a team of robots to look for, collect, and deliver attractors (pucks) to a designated location. During this process, the robots may interfere with one another. This interference is caused by *resource competition* (e.g. space, information, or objects) or *goal competition* [5]. For example, suppose two robots are delivering pucks to the same home base at the same time. At one time, only a robot can occupy the home base. Thus, the two robots compete for space to complete the delivery task. Interference may degrade a team's performance; hence, we must consider this factor when designing multi-robot control systems.

Much research has been conducted to investigate interference in multi-robot systems. Mataric [6] introduced an approach called *social rules* to minimize interference in a multi-robot system. In her experiments (i.e. foraging), the robots learn to yield and share information about the puck locations that is beneficial to the whole group.

In other work, Fontan and Mataric [3] investigate a territorial approach to reduce interference in multi-robot foraging. Their studies show that there are an optimal number of robots for completing the given task. If the size of this team is reduced or increased, then the overall performance decreases.

Goldberg and Mataric [4] use interference to evaluate multi-robot systems. In their experiments, they implement three different strategies (i.e. homogeneous, pack and caste arbitration) for a foraging task and measure the interference characteristic of each strategy as well as its performance.

Balch [1] evaluates performance of three robot teams on the multi-foraging task where each team is behaviorally different from one another. The first team consists of homogeneous robots that all look for attractors and deliver them to the designated locations. The second team consists of a "sorter" and "pickers". The sorter stays near the home base area to deliver attractors that are dropped on the boundary of the home base by the pickers. In the third team, the robots are specialized in only collecting either blue or red attractors. In this experiment, the homogeneous teams out perform the other two teams.

The Experiment

1. Simulation Setup

This section describes an experiment done using TeamBots as a case study. Using this case study, we want to show that simulation is very useful in

multi-robot research. Simulation can give quantitative data to analyze. Without such data and simulation, it is hard to observe many interesting phenomena in multi-robot environments.
There are two main factors examined in the experiment:

- Impacts of puck density on multi-robot foraging performance
  Balch [1] and Fontan and Mataric [3] have shown the impacts of robot density on performance in multi-robot foraging. Balch shows that the overall performance of a multi-robot team increases along with the increased number of robots in the team. In contrast, Fontan's experiment shows that the increased number of robots, after a certain number, degrades the overall performance of the team. In both experiments, they all concern about the density of robots but not the density of pucks.
- Impacts of behavioral diversity on multi-robot foraging performance
  The performance of two robot teams is evaluated at each density of pucks. The first team consists of homogeneous robots that wander, pick up, and deliver pucks to the home base. In the other team, the robots have behavioral diversity. One of them acts as a sorter while other robots act as a picker. The sorter wanders near the home base, picks up, and delivers pucks that are dropped by the pickers. The pickers' behavior is similar to the homogeneous teams' except that the robots drop pucks on the boundary of the home base.

One factor that makes this experiment is different from other work is the variety of puck density in the environments tested.

Figure 2 shows the simulation setup. The density of pucks varies from 20 to 100 pucks/100 m2. The number of robots varies from one to eight in the homogeneous systems and two to eight in the heterogeneous systems. Each configuration is run 10 times, giving 750 total trials. The performance is measured by the number of pucks collected in 10 minutes.

2. Configuration of Robots

This experiment uses two configurations of motor schema based control systems. One is for the sorter; another one is for the pickers. These configurations are a modified version of a sample program in TeamBots ("forage.java"). The configuration of the control system in "forage.java" is not suitable for this experiment because the robots are stuck easily in the home base due to the high level of interference that occurs when the robots delivers pucks at the same time.

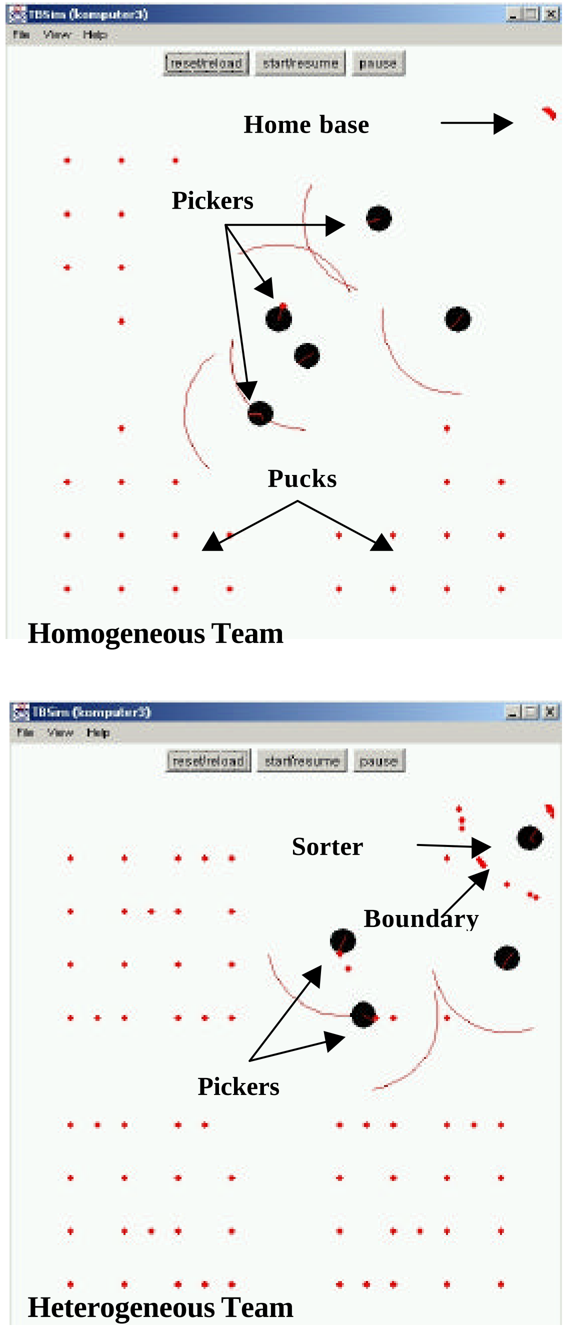

Figure 2. Homogeneous vs. Heterogeneous team. In the heterogeneous teams, a robot (sorter) stays near the home base to deliver pucks that are dropped by other robots (pickers) on the boundary.

The robots have three different states, i.e. *wander*, *acquire*, and *deliver* (Figure 3). First, the robots wander around the environment. When a robot finds pucks, it moves toward the closest one and acquire the puck. After picking up the puck, the robot moves toward the home base and drops the puck in the home base area (homogeneous teams) or on the boundary (heterogeneous teams). The sorter's behavior is similar to the pickers'. The main difference is in the sorter control system there is a motor schema that keeps the robot stay near the home base.

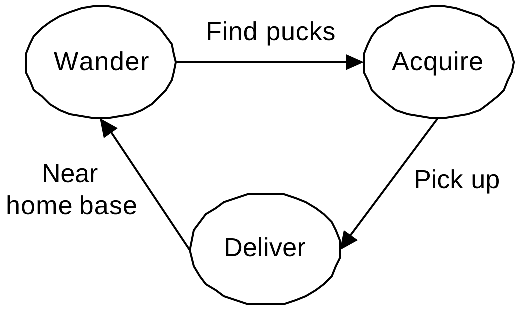

Figure 3: The behavioral states of the robots

Results and Analysis

Table 1 shows the comparison of performance between homogeneous and heterogeneous teams in various puck densities. The results show that in all densities tested, small homogeneous teams (up to four robots) out perform heterogeneous teams. Heterogeneous teams, however, perform better than homogeneous teams if the team is relatively large and the density of pucks is high.

Table 1. The performance of homogeneous teams (Ho) vs. heterogeneous teams (He). The sign "X" indicates teams with better performance.

		Density of pucks/100 m2									
		20		40		60		80		100	
		Ho	He	Ho	He	Ho	He	Ho	He	Ho	He
Number of robots	2	X		X		X		X		X	
	3	X		X		X		X		X	
	4	X		X		X		X		X	
	5	X		X		X		X			X
	6	X			X		X		X		X
	7	X			X		X	X			X
	8	X			X		X		X		X

These results are different from [1]. In [1], homogeneous teams perform best over the color specialization (i.e. acquiring red or blue pucks only) and the territorial specialization (i.e. sorter and picker robots). The difference is mainly caused by the home base location. In [1], the home base is located at the center of the environment. Thus, the interference among the robots is not as high as in this experiment.

The puck density and the number of robots in a team affect the teams' performance. In an environment where the puck density is low, the robots spend most of their time for wandering around the environment. Therefore, in such environments, homogeneous teams have better performance because all members participate in looking for pucks. In homogeneous teams, a robot (the sorter) only wanders near the home base; hence, it cannot help other team members find pucks. On the contrary, when the puck density is high, the robots can find pucks easily. Therefore, the robots deliver pucks to the home base simultaneously. In this case, having a sorter is beneficial because it reduces the level of interference near the home base area.

The level of interference contributes to the uncertainty factor. For example, in the density of 100 pucks/100 m2 with eight homogeneous working robots, the best performance was 33 pucks collected while the worst was only nine pucks delivered to the home base. This fact shows that the heterogeneous teams' performance is more consistent than the homogeneous teams' in all environments tested. However, the heterogeneous teams' performance is limited by the capability of the sorter. The pickers may drop many pucks on the boundary of the home base but the sorter may not be able to deliver all of those pucks to the home base in the given time.

Conclusion

Simulation plays important roles in multi-agent robotics research. Using a simulator, we can observe many interesting phenomena that are resulted from interactions among robots in a team. Furthermore, simulation can give massive quantitative data quickly and inexpensively.

This paper introduced TeamBots as a suitable tool for mobile robotics research. Section 2 presented a brief description and highlighted some of the TeamBots's features. Moreover, we presented a case study to show that we can examine many interesting research issues in mobile robotics using this tool in a relatively easy way. In term of the number of lines, the modified code for this experiment has only about 50 lines more than the original code ("forage.java").

References

[1] Balch, T., "The Impact of Diversity on Performance in Multi-robot Foraging", Autonomous Agents (Agents '99), Seattle, WA, May 1999.
[2] Balch, T. and Ram, A., "Integrating Robotic Technologies with JavaBots", Working Notes of the AAAI 1998 Spring Symposium, Stanford, CA, 1998.
[3] Fontan, M. S. and Mataric, M. J., "Territorial Multi-Robot Task Division", IEEE Transactions on Robotics and Automation, 14(5), October 1998.
[4] Goldberg, D. and Mataric, M. J., "Interference as a Tool for Designing and Evaluating Multi-robot Controllers", Proceedings of AAAI-97, Providence, Rhode Island, July 1997, pp 637-642.
[5] Mataric, M. J., "Issues and Approaches in Design of Collective Autonomous Agents", Robotics and Autonomous Systems, 16(2-4), December 1995, pp 321-331.
[6] Mataric, M. J., "Learning Social Behaviors", Robotics and Autonomous Systems, 20, 1997, pp 191-204.
[7] TeamBots. Available at: http://www.Teambots.org