The 2003 Ohio University Small Size RoboCup Team

David Chelberg, Carl Bruggeman, Doug Lawrence, Jae Lew, Jundong Liu, Cynthia Marling, Maarten Uijt De Haag, J. Jim Zhu, Andy Dick, Jason Kendrick, Yong Liu, Carmen Mazzant, Ted Smith, Mark Tomko

Russ College of Engineering and Technology
Ohio University
Stocker Center
Athens, OH 45701, U.S.A.
`http://www.ohiou.edu`

Abstract. This paper describes the Robocats, Ohio University's entry in the small-size robot league. Our team represents research efforts in computer science, vision, mechanical engineering, electrical engineering, and control theory. We discuss some of our achievements in each area here.

1 Introduction

The Ohio University Robocats small-size robot team consists of a team of five physical robots under the guidance of an intelligent, distributed, multi-agent software system. The Ohio University RoboCup effort began in the Ohio University School of Computer Science as a testbed for real-time, intelligent, and distributed systems. Since then, development efforts have expanded to include students and faculty in the electrical engineering, industrial technology, and mechanical engineering departments [2]. Students in all three areas can participate in RoboCup as partial fulfillment of their curriculum. The Robocats competed internationally for the first time at RoboCup-2002 in Fukuoka, Japan. This paper describes the Robocats soccer team and highlights the developments efforts since RoboCup-2002.

2 Robots

Our robots utilize a three wheel, omni-directional design intended to achieve maximum flexibility of movement [4, 1, 6]. Each robot is roughly cylindrical with wheels spaced 120 degrees apart. The wheels allow powered rotation in one direction and free rotation in the orthogonal direction. The front of each robot is equipped with dribbling and kicking mechanisms.

Our efforts since last year have focused on improving the locomotion and kicking systems. We are currently investigating a number of smaller design modifications as well.

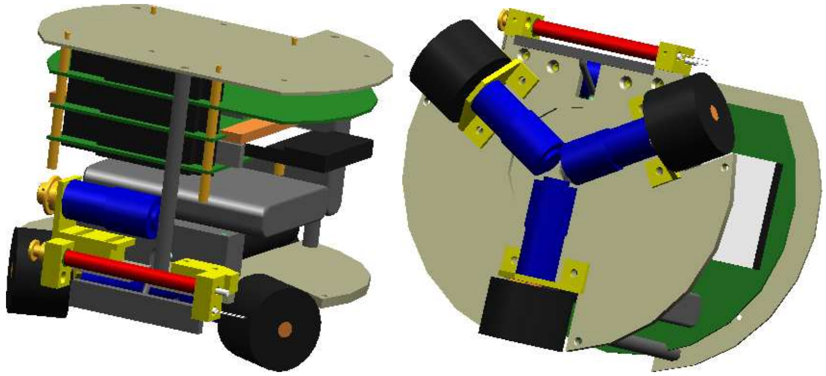

Fig. 1. OU Robot Design

This year, we have simplified the drive train by directly coupling the motors to the wheels. Directly driving the motors has a number of advantages over the geared drive system we employed in 2002. Removing the spur gears and associated components decreases the overall system weight and increases the amount of space available for other components. In addition, the direct-drive system eliminates wobble caused by the gearing mechanisms.

We have developed a new pneumatic kicking mechanism to replace the old system, which relied on two large solenoids mounted beneath the robot. In addition to producing a considerably more powerful kick, the new system uses a voltage-sensitive orifice proportional solenoid valve, or pneumatic motor, which allows the robot to alter the strength of the kick. As a result, the robots can shoot and pass at different speeds. This should greatly enhance the cooperative abilities of our robots on the field.

In addition to the new drive train and kicking mechanisms, we are developing an improved dribblebar, lowering the center of gravity of the robots, and investigating a soft shroud to protect the robots.

3 Electronics

Our onboard electronics consist of a 100 MHz 486 CPU on a PC104 with 64 MB disk on chip and 64 MB of onboard RAM. We use 802.11b to communicate with the off-board computer systems. Each PC runs a customized Linux system using the 2.2.19 kernel. The PC serves as a local interface between the strategy and control systems. In addition to the PC, each robot carries a control board responsible for distributed control and inertial navigation.

The next generation control board improves on a number of the features of the previous design and includes several new features. Additions for this year include ball sensor circuitry, a dual-axis accelerometer, a gyroscope, battery hot swapping capability, and brown out protection circuitry. Improvements have also been made to the board layout, efficient and fault-tolerant use of the microcontrollers, and a more robust method of quadrature decoding on the axial wheel measurements.

The ball sensor circuitry uses an infrared LED mounted in front of the dribble bar and kicker. By detecting changes in signal intensity, the microcontroller (PIC16F877) can determine if the ball is present to avoid futile kicking attempts. The PIC can set threshold levels to adapt to different field lighting conditions. The addition of an accelerometer (ADXL202E) allows the controller board to measure dual-axis accelerations between -2g and +2g. The PC can use acceleration data to improve the fidelity of its control.

Other new features include the ability to hot swap the batteries and brown out detection circuitry. We can now replace batteries without shutting down the computer system. Since we can detect brown outs, the onboard computer can shut down safely before the battery is depleted below a safe level.

4 Control

We have developed an omni-directional mobile robot controller using a nonlinear control method [5]. The robot controller consists of an inner and outer loop. The outer loop calculates the desired robot speed according to the robot trajectory command and the position error. The inner loop adjusts the motor voltages to achieve the desired robot speed. The inner loop controller compensates for the motor dynamics, which helps to achieve better acceleration and deceleration control. Both the outer loop and inner loop control are nonlinear controllers based on linearization along a nominal trajectory. This method is known as trajectory linearization control (TLC). TLC combines nonlinear dynamic inversion and linear time-varying eigenstructure assignment in a novel way, which provides robust stability and performance along all feasible trajectories without interpolation of controller gains. Our tests have shown our controller to be both effective and robust.

5 Strategy

We have developed a distributed, multi-agent architecture for our AI software in order take advantage of available off-field computing resources [3]. Each agent consists of a hierarchy of *schemas*, which implement strategies for achieving a desired world state. High-level schemas perform large tasks or implement deliberative planning, while lower level schemas perform less complex or reactive tasks. The schema hierarchy yields a hybrid of deliberative and reactive agents that allow long term planning to occur as computing resources and time become available, while using reactive behaviors to respond to changes in the world state in real time.

The hierarchical arrangement of schemas also means that a high-level schema can consist of a number of reusable lower lever schemas. A high-level schema chooses lower-level schemas to execute based on the desirability of the world state it achieves. Desirability of a schema is determined by the perceived utility of the schema and the feasibility of performing the action that would achieve that outcome. Since utility is context-dependent, higher-level schemas determine the utilities of component lower-level schemas. A schema must contain some measure of its own feasibility based on the current world state, since a high-level schema differentiates among its component schemas purely by the goals they achieve.

Our architecture yields flexible strategy that represents a balance between long term, deliberative planning, and short term, reactive planning.

6 Summary

The Ohio University RoboCup project has continued to produce innovations since its first international competition in 2002. Our research has yielded a number of new and improved features in our entry for RoboCup-2003. We feel that these design elements will represent a substantial improvement over last year's system.

References

1. Brian Carter, Matt Good, Mike Dorohoff, Jae Lew, Robert L. Williams II, and Paolo Gallina. Mechanical design and modeling of an omni-directional robocup player. In *Proceedings RoboCup 2001 International Symposium*, Seattle, WA, August 2001.
2. David M. Chelberg, Matthew Gillen, Qiang Zhou, and Arvind Lakshmikumar. 3D-VDBM: 3D visual debug monitor for RoboCup. In *IASTED International Conference on Computer Graphics and Imaging*, pages 14–19, November 2000.
3. M. Gillen, A. Lakshmikumar, D. Chelberg, C. Marling, M. Tomko, and L. Welch. A hybrid, hierarchical schema-based architecture for distributed autonomous agents. In *AAAI symposium on Intelligent, Distributed and Embedded Systems*, Stanford University, Stanford CA, March 2002.
4. R.L. Williams II, B.E. Carter, P. Gallina, and G. Rosati. Dynamic model with slip for wheeled omni-directional robots. *IEEE Transactions on Robotics and Automation*, accepted.
5. Yong Liu, Xiaofei Wu, J. Jim Zhu, and Jae Lew. Omni-directional mobile robot controller design by trajectory linearization. In *Proceedings of American Control Conference*, June 2003.
6. Lance Wilson, Craig Williams, Justin Yance, Jae Lew, Robert L. Williams II, and Paolo Gallina. Design and modeling of a redundant omni-directional robocup goalie. In *Proceedings RoboCup 2001 International Symposium*, Seattle, WA, August 2001.