CMDragons 2008 Team Description

Stefan Zickler, Douglas Vail, Gabriel Levi, Philip Wasserman, James Bruce,
Michael Licitra, and Manuela Veloso

Carnegie Mellon University
`{szickler,dvail2,jbruce,mlicitra,mmv}@cs.cmu.edu`
`{glevi,pwasserm}@andrew.cmu.edu`

Abstract. In this paper we present an overview of CMDragons 2008, Carnegie Mellon's entry for the RoboCup Small-Size League. Our team builds upon the research and success of RoboCup entries in previous years. Technical improvements for 2008 include a new physics-based simulator, as well as an experimental role-recognition approach based on feature selection.

1 Introduction

Our RoboCup Small-Size League entry, CMDragons 2008, builds upon the world champion team CMDragons 2007, as well as the ongoing research used to create the CMDragons team (1997-2003,2006-2007) and CMRoboDragons joint team (2004, 2005). Our team entry consists of five omni-directional robots controlled by an offboard computer. Sensing is provided by two overhead mounted cameras linked to frame-grabbers on the offboard computer. The software then sends driving commands to the individual robots. The first section describes the robot hardware and the offboard control software required to implement a robot soccer team. The second section highlights some advances introduced in this year's team. The paper then finishes with concluding remarks.

2 System Overview

Our team consists of seven homogeneous robot agents, with five being used in a game at any point in time. In Figure 1, an example robot is shown with and without a protective plastic cover. The hardware is the same as used in RoboCup 2006 and 2007. We believe that our hardware is still highly competitive and allows our team to perform close to optimal within the tolerances of the rules. Thus, similar to the previous year, we will not introduce any major changes to the robotic hardware. Instead, we focus our efforts on improving the software to fully utilize the robots' capabilities instead.

2.1 Robot Hardware

Each robot is omni-directional, with four custom-built wheels driven by 30 watt brushless motors, each featuring a reflective quadrature encoder. The kicker is

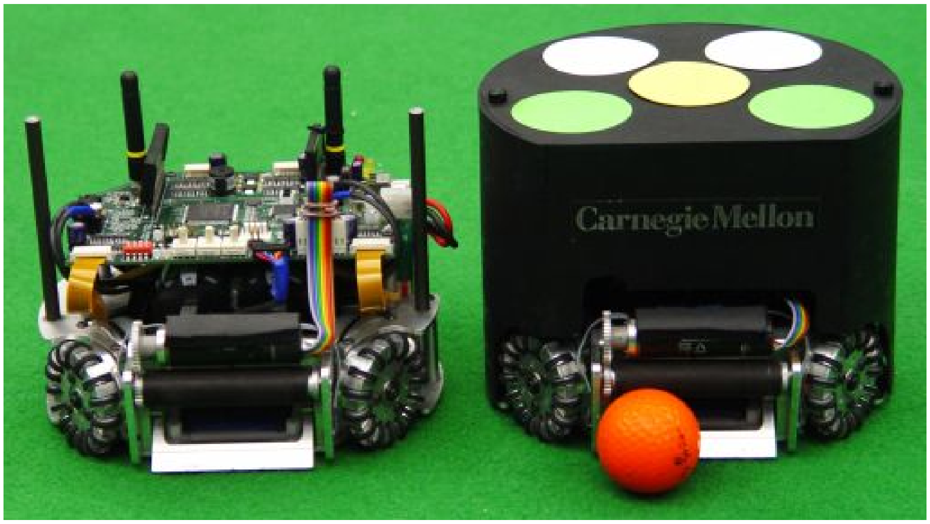

Fig. 1. A CMDragons robot shown with and without protective cover.

a large diameter custom wound solenoid attached directly to a kicking plate. It is capable of propelling the ball at speeds up to $15 m/s$, and is fully variable so that controlled passes can also be carried out. The CMDragons robot also has a chip-kicking device, implemented by a custom-made flat solenoid located under the main kicker, which strikes an angled wedge visible at the front bottom of the robot. It is capable of propelling the ball up to $4.5 m$ before it hits the ground. Both kickers are driven by a bank of three capacitors charged to $200 V$. Ball catching and handling is performed by a motorized rubber-coated dribbling bar which is mounted on an hinged damper for improved pass reception. A more detailled description of the robot's design and electronics can be found in [1].

Our robot is designed for full rules compliance at all times. The robot fits within the maximum dimensions specified in the official rules, with a maximum diameter of 178mm and a height of 143mm. The dribbler holds up to 19% of the ball when receiving a pass, and somewhat less when the ball is at rest or during normal dribbling. The chip kicking device has a very short travel distance, and at no point in its travel can it overlap more than 20% of the ball due to the location of the dribbling bar. While technically able to perform kicks of up to $15 m/s$, the main kicker has been hard-coded to never exceed kick-speeds of $10 m/s$ for full rule compliance.

2.2 Software

The software architecture for our offboard control system is shown in Figure 2. It follows the same overall structure as has been used in the previous year, outlined in [1]. The major organizational components of the system are a server program which performs vision and manages communication with the robots, and two

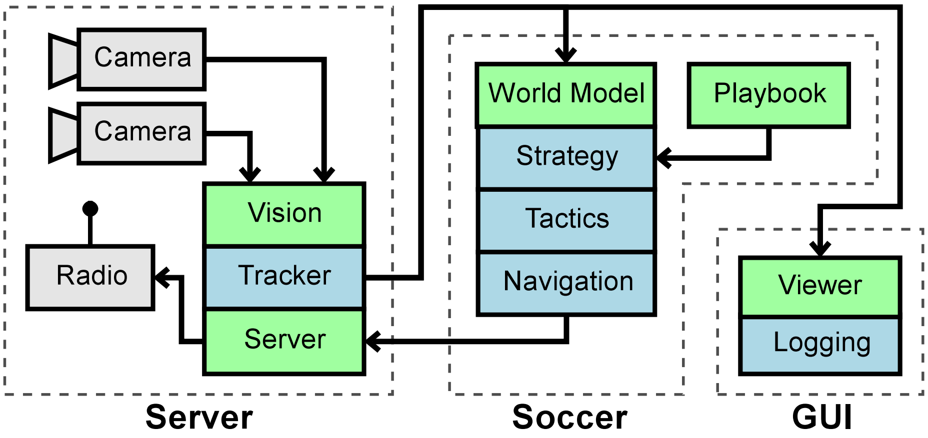

Fig. 2. The general architecture of the CMDragons offboard control software.

client programs which connect to the server via UDP sockets. The first client is a soccer program, which implements the soccer playing strategy and robot navigation and control, and the second client is a graphical interface program for monitoring and controlling the system.

The server program consists of vision, tracker, radio, and a multi-client server. The vision system uses CMVision2 for low-level image segmentation and connected region analysis [2, 3]. On top of this system lies a high-level vision system for detecting the ball and robot patterns. Our robot pattern detector uses an efficient and accurate algorithm for multi-dot patterns described in [4]. Tracking is achieved using a probabilistic method based on Extended Kalman-Bucy filters to obtain filtered estimates of ball and robot positions. Additionally, the filters provide velocity estimates for all tracked objects. Further details on tracking are provided in [5]. Final commands are communicated by the server program using a RS232 radio link.

The soccer program is based on the STP framework [5]. A world model interprets the incoming tracking state to extract useful high level features (such as ball possession information), and act as a running database of the last several seconds of overall state history. This allows the remainder of the soccer system to access current state, and query recent past state as well as predictions of future state through the Kalman filter. The highest level of our soccer behavior system is a strategy layer that selects among a set of plays [6, 7]. Below this we use a tree of tactics to implement the various roles (attacker, goalie, defender), which in turn build on sub-tactics known as skills [5]. One primitive skill used by almost all behaviors is the navigation module, which uses the RRT-based ERRT randomized path planner [8–10] combined with a dynamics-aware safety method to ensure safe navigation when desired [11]. It is an extension of the Dynamic Window method [12, 13]. The robot motion control uses trapezoidal velocity profiles (bang-bang acceleration) as described in [14, 5].

3 Significant Developments

The evolution of the Small Size League has brought several new challenges with it. The increasing complexity and quality of teams requires new approaches to achieve adaptive robot strategies. Additionally, new techniques are needed to accurately model the constantly improving ball-manipulation abilities of robots. We present two improvements to our system that attempt to deal with these issues: a simulator based on rigid-body dynamics, and a role recognition system based on feature selection.

3.1 Rigid-Body Dynamics Simulator

Being able to accurately simulate robot behaviors is an integral part of our team's development cycle. Simulation allows rapid testing of new code without the need to impose drain on our robotic hardware. Additionally, it allows us to simulate scenarios which we are unable to recreate by our limited number of physical robots, such as full five-on-five RoboCup games. In recent years, our ball-manipulation capabilities and behaviors have become very sophisticated. Our strategies often rely on complex dynamics interactions such as deflecting a ball off from opponents. The introduction of chip-kicks has added the additional requirement of three-dimensional simulation. Finally, many of our most recent behaviors rely heavily on our robots' ball-dribbling capabilities, and thus should also be accommodated in a simulated model.

In order to model such behaviors, we introduce a new simulator that is able to compute complex rigid body dynamics. Various free and commercial simulation engines exist to perform this task. We chose Ageia PhysX as our simulation engine of choice due to its performance and ease of use, but other candidates such as e.g. the Open Dynamics Engine (ODE) might be equally suited. The concept of such simulation engines is that, given a scene of rigid-bodies and a set of initial positions and velocities, we can apply forces and torques to arbitrary objects in the scene. The engine will then integrate the scene forward in time, automatically resolve any rigid-body interactions (such as collisions) using basic Newtonian physics, and finally provide us with a new state of positions and velocities. Our simulator is able to act as a full replacement of the standard server depicted in figure 2, thus processing the soccer-system's commands and returning the newly observed state of the world.

Our robots are modeled using a convex triangulated shell in combination with other convex rigid-body primitives. The golf-ball is approximated by using a simple sphere. In order to achieve accurate dynamics simulations, it is important that we have an adequate model of our robot's ball-handling capabilities. One particularly challenging issue is to correctly model the robots' dribbler-bar. One approach is to model the dribbler implicitly, by defining a motorized, rotating, high-friction cylinder which is connected to the robot by using a rotary joint. In practice however, this approach failed to accurately model the dribbling behavior, possibly due to the accumulation of joint-error and an insufficient simulated torque transfer between the bar and the ball. The approach we took instead

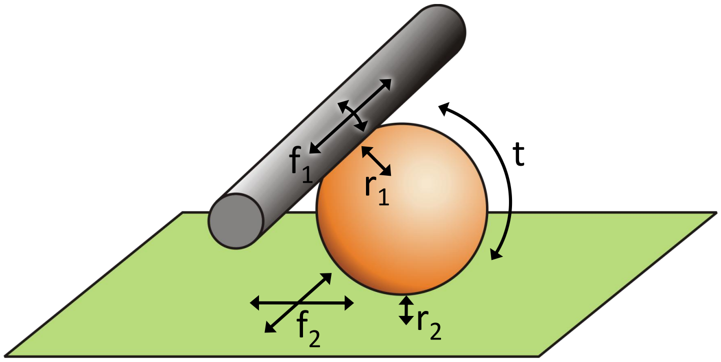

Fig. 3. A diagram of the simulated dribbling model and its parameters.

was to model the dribbling action explicitly: if the ball is in contact with the dribbler-bar then we directly apply a torque to the ball facing towards our robot. The exact dribbling behavior can be controlled by modifying the various variables that affect the ball's dynamics. A diagram of the dribbling model is shown in figure 3. The applied torque is depicted as t. Variables that affect our ball, are the friction in relation to the floor f_2 and in relation to the dribbler bar f_1. The dribbler bar uses an anisotropic friction model, defining low friction against the vertical direction of the bar (thus, roughly simulating a freely spinning bar), and higher friction against the horizontal direction of the bar (thus modeling the high friction of the bar's rubber). In addition to friction, we can control the ball's coefficient of restitution against the dribbler and the floor respectively. Naturally r_1 will have a fairly low coefficient, thus roughly modeling the dribbler's softness and dampening, whereas r_2 will have slightly higher coefficient to model the bounciness of the carpet. Additional constants in our physics model are angular and linear damping of both the robot and ball, thus simulating air-drag. The robot's motions are simulated explicitly as well. Instead of modeling the wheels which will impose friction on the floor, we model the robot as having a flat, low-friction base-plate. We then directly apply forces and torques to the robot to simulate its motions. Kicks and chip-kicks are simulated in a similar fashion, by exerting linear impulses directly on the ball. A simulated sequence of one of our robots performing a "dribble and fling"-maneuver can be found in figure 4.

It should be noted that currently, all of the variables have been tweaked experimentally to obtain a relatively accurate model of the robot. In the future, it will be interesting to employ supervised learning techniques to automate this task.

3.2 Feature Selection for Role Recognition

Robot soccer is a domain where activity recognition has the potential to make a large contribution. One particularly interesting application is to classify the

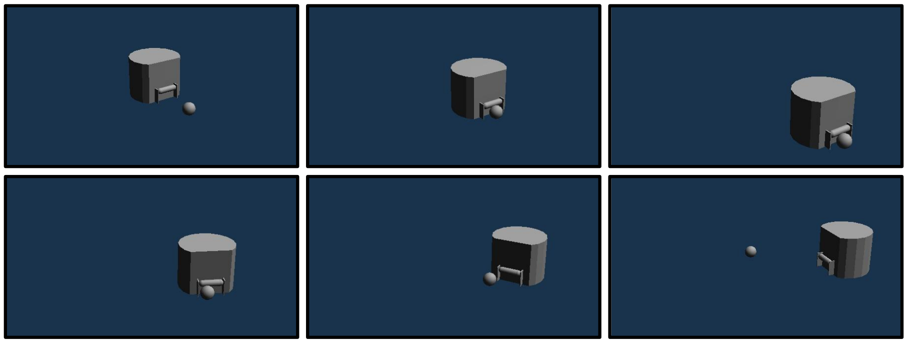

Fig. 4. A simulation sequence of a “dribble-and-fling” behavior.

roles of our opponents and base strategic decisions upon that classification. Activity recognition is a temporal classification problem. We map from a temporal sequence of observations to the roles of the robots on the opposing team. It is a challenging problem because the observations, i.e. the positions of the ten robots and the ball, do not directly map to roles; we must infer roles from low level position information rather than higher level, more abstract observations. To address this challenge, we train classifiers to map from *features* of the observations to roles rather than directly from observations to roles. Features are functions of the observations that inject domain knowledge into the classification by transforming the observations into a more useful form for the classifier. As an example, the distance between the ball and the robot that is closest to the ball is an important feature. Rather than inputing the coordinates of the robot and the ball to the classifier, we would input the actual distance instead.

Choosing an appropriate set of features is important for accurate activity recognition. As humans designing features, we can easily define prototypes for good candidate features. For example, in soccer, a feature that tests if the distance between two objects is less than a threshold is useful. Instantiating this prototype with different pairs of objects and different thresholds results in a large pool of candidate features, particularly our multi-robot soccer domain where the number of objects and relationships between objects is large. We use *feature selection* to choose a small subset of the candidate features to include in the final model. Reducing the number of candidate features is important to reduce over-fitting, which would reduce the accuracy of the final model, and to reduce the computational cost of classification so that roles may be recognized and responded to online.

For this year’s team, we will investigate feature selection for activity recognition using conditional random fields [15]. In the past, we have shown that conditional random fields are well suited to activity recognition in robot domains [16]. We have also investigated feature selection in conditional random fields, specifically, using ℓ_1 regularization [17]. Our ultimate goal is to train a classifier that can accurately recognize opponent roles in real time. We plan to take the output of this classifier into account when making strategic decisions about how to respond to the opponent team.

4 Conclusion

Competition	Result
US Open 2003	1st
RoboCup 2003	4th
RoboCup 2004	4th [1]
RoboCup 2005	4th [1]
US Open 2006	1st
RoboCup 2006	1st
China Open 2006	1st
RoboCup 2007	1st

Table 1. Results of RoboCup small-size competitions for CMDragons from 2003-07

This paper gave a brief overview of CMDragons 2008, covering both the robot hardware and the software architecture of the offboard control system. The hardware has built on the collective experience of our team and continues to advance in ability. The software uses our proven system architecture with continued improvements to the individual modules. The CMDragons software system has been used in three national and seven international RoboCup competitions, placing within the top four teams of the tournament every year since 2003, and finishing 1st in 2006 and 2007. The competition results since 2003 are listed in table 1. We believe that the RoboCup Small-Size League is and will continue to be an excellent domain to drive research on high-performance real-time autonomous robotics.

References

1. Bruce, J., Zickler, S., Licitra, M., Veloso, M.: CMDragons 2007 Team Description. Technical report, Tech Report CMU-CS-07-173, Carnegie Mellon University, School of Computer Science (2007)
2. Bruce, J., Balch, T., Veloso, M.: Fast color image segmentation for interactive robots. In: Proceedings of the IEEE Conference on Intelligent Robots and Systems, Japan (2000)
3. Bruce, J.: CMVision realtime color vision system. The CORAL Group's Color Machine Vision Project `http://www.cs.cmu.edu/~jbruce/cmvision/`.
4. Bruce, J., Veloso, M.: Fast and accurate vision-based pattern detection and identification. In: Proceedings of the IEEE International Conference on Robotics and Automation, Taiwan (May 2003)

---

[1] Provided software component as part of a joint team with Aichi Prefectural University, called CMRoboDragons

5. Browning, B., Bruce, J.R., Bowling, M., Veloso, M.: STP: Skills tactics and plans for multi-robot control in adversarial environments. In: Journal of System and Control Engineering. (2005)
6. Bowling, M., Browning, B., Veloso, M.: Plays as effective multiagent plans enabling opponent-adaptive play selection. In: Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'04). (2004)
7. Bruce, J.R., Bowling, M., Browning, B., Veloso, M.: Multi-robot team response to a multi-robot opponent team. In: Proceedings of the IEEE International Conference on Robotics and Automation, Taiwan (May 2003)
8. Bruce, J.R., Veloso, M.: Real-time randomized path planning for robot navigation. In: Proceedings of the IEEE Conference on Intelligent Robots and Systems. (2002)
9. LaValle, S.M., James J. Kuffner, J.: Randomized kinodynamic planning. In: International Journal of Robotics Research, Vol. 20, No. 5. (May 2001) 378–400
10. James J. Kuffner, J., LaValle, S.M.: RRT-Connect: An efficient approach to single-query path planning. In: Proceedings of the IEEE International Conference on Robotics and Automation. (2000)
11. Bruce, J.R., Veloso, M.: Safe multi-robot navigation within dynamics constraints. Proceedings of the IEEE **94** (July 2006) 1398–1411
12. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robotics and Automation Magazine **4** (March 1997)
13. Brock, O., Khatib, O.: High-speed navigation using the global dynamic window approach. In: Proceedings of the IEEE International Conference on Robotics and Automation. (1999)
14. Bruce, J.R.: Real-Time Motion Planning and Safe Navigation in Dynamic Multi-Robot Environments. PhD thesis, Carnegie Mellon University (Dec 2006)
15. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 282–289
16. Vail, D.L., Veloso, M.M., Lafferty, J.D.: Conditional random fields for activity recogntion. In: AAMAS. (2007)
17. Vail, D.L., Lafferty, J.D., Veloso, M.M.: Feature selection in conditional random fields for activity recognition. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2007)