Opponent-Driven Planning and Execution for Pass, Attack, and Defense in a Multi-Robot Soccer Team

Joydeep Biswas, Juan P. Mendoza, Danny Zhu, Benjamin Choi, Steven Klee, Manuela Veloso

joydeepb@ri.cmu.edu, jpmendoza@ri.cmu.edu, dannyz@cs.cmu.edu, benchoi@cmu.edu, sdklee@cmu.edu, veloso@cmu.edu

Robotics Institute and Computer Science Department, Carnegie Mellon University, Pittsburgh, USA

ABSTRACT

We present our Small Size League (SSL) robot soccer team, CMDragons, which performed strongly at the RoboCup'13 competition, placing second out of twenty teams after a prolonged final match ending in penalty shoot-outs. We briefly present the robots' hardware and individual skills, and then focus on our multi-robot passing, attack, and defense planning and execution in the challenging SSL adversarial multi-robot environment. We introduce a pass-ahead behavior, as well as a new dynamic two-stage planner, Coerce and Attack, which explicitly considers opponent defense to hypothetical attack patterns. The Coerce stage generates a coerce attack formation to coerce the opponent robots into leaving strategic openings. The Attack stage modifies the coerce attack pattern in a fluid manner to exploit openings in the defense using pass-ahead to attempt to score. We further present our threat-based defensive multi-robot algorithm which identifies potential threats based on the opponent positioning, and plans the defense accordingly. We present the performance of CMDragons at RoboCup'13 in terms of metrics that evaluate the effectiveness of the low-level skills as well as the high-level defense and offense.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms

Keywords

Artificial Intelligence, Robot Soccer, Adversarial Domains

1. INTRODUCTION AND RELATED WORK

The RoboCup Federation was founded in 1997 with the ultimate goal of having, by the year 2050, a team of fully autonomous soccer-playing robots defeat the World Cup human soccer champions. We are clearly very far from this goal, but significant progress has been made. RoboCup has multiple soccer leagues, which evolve every year, each addressing incremental technical challenges of a multi-robot team competing in an uncertain adversarial environment. The work in this paper is carried out within the context of the RoboCup Small Size League (SSL). Each league is designed to contribute towards the scientific progress of the different challenges of robot soccer, and the SSL contributes fast-paced gameplay and multi-robot teamwork.

The SSL has evolved considerably since its inception in 1997, and in 2013 consists of teams of 6 robots each, with predefined color-coded patterns on the covers, playing on a green carpeted field of size $6\,\text{m} \times 4\,\text{m}$. Cameras mounted overhead capture images of the field, which are then processed by a common vision system, SSL-Vision [1], which reports the poses of all the robots and the position of the ball to both teams' behavior controllers. Robots are commanded via radio by each team's centralized controller. The robots are designed and built by each team independently, subject to regulation size constraints of a maximum diameter of 180 mm and a maximum height of 150 mm.

The robots and the ball in a soccer game in the SSL today move at high speeds, with the robots travelling in excess of $3\,\text{m/s}$ and shooting on the goal at $8\,\text{m/s}$. Consequently, teams face a number of challenges in playing soccer in the SSL, namely: (i) The ball is rarely stopped, and instead constantly moving and passed between robots at high speeds, thus making it difficult to intercept; (ii) while passing between teammates, opponents can easily block passes if they are made to stopped robots; (iii) most goals scored in the SSL are made after high-speed passes, making it impractical for the defense to function simply by following the ball; and (iv) due to the wide variation in the defense strategies of opponents, it is impractical to cover all possible cases with a fixed set of pre-scripted attack strategies. The CMDragons team, as we present in this paper, aims to address these varied challenges. We performed strongly at RoboCup'13, placing second out of twenty teams, after a prolonged final match ending in penalty shoot-outs. To address the challenges of the SSL, in addition to robot skills to intercept fast moving balls, we introduce:

1. A pass-ahead coordination behavior for passing between moving teammates with minimal delay and setup time,

2. A novel Coerce and Attack planner that detects opponent team roles, coerces the opponents to leave strategically advantageous attack locations open, and then

Appears in: *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), Lomuscio, Scerri, Bazzan, Huhns (eds.), May, 5–9, 2014, Paris, France.*

This research was partially supported by the National Science Foundation award number NSF IIS-1012733, and by DARPA award number FA8750-12-2-0291. The views in this document are those of the authors only.

exploits these openings to attack on an unsuspecting defense, and

3. A threat-based defense that explicitly considers passes between opponents to block them.

The state of the art in robot soccer strategies rely exclusively on human-generated plans. There are several approaches to reasoning about *which* plans to select, and *when*. Case-Based Reasoning [2] selects such plans by matching the world state, via a similarity metric, to previously observed states, or cases. Skills, Tactics, and Plays (STP) [3] is an alternative approach to multi-robot coordination where predefined Plays are used to select roles, or Tactics, to be used for specific preconditions of the game. There has also been some work on adapting the plays selected online [4] in response to success and failures of specific plays. Today, most of the SSL teams, including three of the top 4 teams from RoboCup 2012, Team A [5], Team B [6] and Team C [7] use either the STP architecture, or a similar play-based variant. The Coerce and Attack Planner, which we introduce in this paper is novel as a high-level online strategy planning algorithm in the SSL.

To compute *where* to pass to, the most common approach is to use potential fields based on feasible regions [8] or clear path based on the attack plan [9]. In contrast, we introduce a pass evaluation that models the probability of scoring a goal by passing to different locations and then shooting on the goal. Passing actions previously introduced include one-touch passing [8], multi-step pass-and-shoot strategies [10], and physics-based dribbling [11]. We introduce a new method of passing, called pass-ahead, which, unlike these previous approaches, involves passing to a planned *future* location of the receiver rather than its current location.

2. THE ROBOTS AND BASIC SKILLS

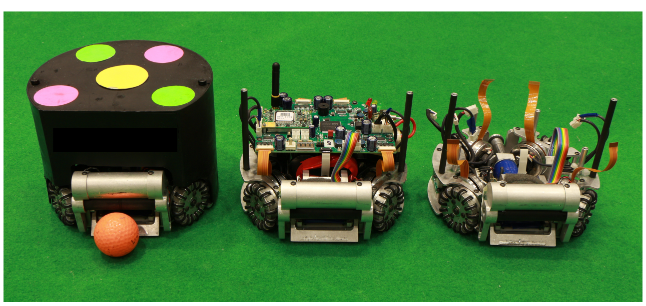

Figure 1: The CMDragons robots, showing (left) a robot with the ball, (middle) without the cover, and (right) without the electronic main board.

The CMDragons team comprises 12 identical robots based on the designs of the SSL robots of CMDragons from 2006. We replicated the mechanical designs, and replaced the electronics main board with a newer design to create a team of 12 robots. While the actual games at RoboCup are played by teams of 6 robots, having 12 robots allowed us to test our software in the lab in full games prior to the competition. Figure 1 shows the internals of the robots, including the electronic board and driving and kicking mechanisms.

One of the basic skills for robots in the SSL is the ability to drive to a target location, starting from an arbitrary start location with an arbitrary starting velocity. We use a near-time optimal trajectory planner [12], implemented as the function $\langle t^*, \mathbf{V}^* \rangle = \text{CalcMotion2D}(\mathbf{x}_s, \mathbf{v}_s, \ \mathbf{x}_f)$ to compute the sequence of velocity commands $\mathbf{V}^*$ and the total time t^* required to navigate from initial location $\mathbf{x}_s$ and initial velocity $\mathbf{v}_s$, to a final location $\mathbf{x}_f$ and zero final velocity.

Using the near-time optimal trajectory planner, we can plan for intercepting moving balls. The problem of dynamic ball interception requires computation of *where* along the trajectory of the ball a robot can intercept it, and *how* to intercept it. The question of where to intercept the ball is determined by where the robot can drive to sooner than the ball can reach, and the question of how to intercept it is governed by the relative location of the intercept with respect to the kicking target location. The computation of the optimal ball intercept location is complicated by the fact that the function CalcMotion2D does not have an analytic form, so the optimal interception location can only be evaluated numerically. For a future ball location *p_ball* along the trajectory of the ball, we can compute

1. the ball travel time to reach *p_ball* based on the carpet model: $t\_ball(p\_ball)$,

2. the robot intercept location based on the kicking target location: $intercept(p\_ball)$,

3. the robot travel time: $t\_robot(intercept(p\_ball))$,

4. the slack time: $slack = t\_ball - t\_robot$, and hence

5. whether the intercept will be successful: $slack \geq 0$.

This sequence of computations is performed for discrete future locations of the ball along its trajectory to compute the optimal intercept location by a linear search. There are two types of ball intercept locations, the *minimum time intercept*, given by

$$\underset{intercept}{\arg\min}(t\_ball) : slack \geq 0, \tag{1}$$

and the *maximum slack intercept*, given by

$$\underset{intercept}{\arg\max}(slack) : slack \geq 0. \tag{2}$$

The minimum time intercept is the location where the robot could intercept the ball fastest, whereas the maximum slack intercept is the location where interception will be most robust to execution errors due to the available slack time. Therefore, the minimum time intercept is used for cases where the cost of failure is low, like an attacker opportunistically trying to intercept and shoot on the goal, while the maximum slack intercept is used for cases where the cost of failure is high, like the primary defense trying to block an opponent's shot on the goal.

3. PASSING

Since SSL rules forbid robots from dribbling the ball for more than 50 cm, passing the ball between teammates is by far the most common method for creating goal opportunities

Thanks to Michael Licitra, mikelicitra@gmail.com, for designing the mechanical designs, and for designing and fabricating the electrical designs for the robots.

in a controlled way. Currently in the SSL, robots pass to each other through *flat passes*: kicking forward at ground level, *chip passes*: kicking forward and up in the air, or *yanking*: imparting back-spin on the ball and then releasing it to roll backwards. This section describes how CMDragons performs these passes with extensive coordination between the passing robot P and the receiving robot R. First, all potential receivers search for locally optimal locations $\boldsymbol{x}^* \in \mathbb{R}^2$ in the field to receive a pass from P, as described in Section 3.1. Then, P evaluates all potential receivers and chooses the best to perform a coordinated pass. Finally, P and the chosen receiver R coordinate their passing and receiving maneuvers to minimize the opponents' opportunity to prevent a successful pass, as described in Section 3.2.

3.1 Pass Location Selection

When searching for the best location to receive a pass, our algorithm attempts to maximize the probability of scoring a goal if passer P were to pass to R. That is, we define $\boldsymbol{x}^*$ as:

$$\boldsymbol{x}^* \equiv \arg\max_{\boldsymbol{x}\in\mathbb{R}^2} [P(\text{goal} \mid \boldsymbol{x})] \tag{3}$$

Notice that we can divide the probability on the right into two factors: the probability of R successfully receiving the ball at location $\boldsymbol{x}$, and the probability of R successfully scoring a goal from $\boldsymbol{x}$ given that it has successfully received the ball:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathbb{R}^2} [P(\text{receive} \mid \boldsymbol{x}) P(\text{goal} \mid \text{receive}, \boldsymbol{x})] \tag{4}$$

Since the SSL domain is high-dimensional, highly dynamic, and adversarial, it is unrealistic to expect to compute the two probabilities above exactly. However, the function we actually maximize attempts to approximate these probabilities in a computationally feasible way. We define a set of important conditions c_i that must be true for R to receive a pass at location $\boldsymbol{x}$ and successfully score on the goal. We also assume the events to be independent to simplify computation. The approximating function is thus defined as:

$$\hat{P}(\text{receive} \mid \boldsymbol{x}) \equiv \prod_i \hat{P}(c_i|\boldsymbol{x}). \tag{5}$$

For R to successfully receive a pass, all c_i need to be true, and $\hat{P}(c_i|\boldsymbol{x})$ is an approximation to the probability that c_i will be true given $\boldsymbol{x}$. The conditions c_i we consider are:

- **c_1: No opponent can reach $\boldsymbol{x}$ faster than R can.** $\hat{P}(c_1|\boldsymbol{x}) \sim 0$ when an opponent can navigate to $\boldsymbol{x}$ faster than R; $\hat{P}(c_1|\boldsymbol{x}) \sim 1$ otherwise.
- **c_2: No opponent intercepts the pass.** $\hat{P}(c_2|\boldsymbol{x}) \sim 0$ when an opponent can navigate to a point along the line between the origin $\boldsymbol{x}_0$ of the pass and its destination $\boldsymbol{x}$ faster than the ball can get from $\boldsymbol{x}_0$ to that point, considering passing speed; $\hat{P}(c_2|\boldsymbol{x}) \sim 1$ otherwise, as visualized in Figure 2.
- **c_3: The pass is long enough for R to react and receive the pass robustly.** $\hat{P}(c_3|\boldsymbol{x}) \sim 0$ when the time the ball would take to travel from $\boldsymbol{x}_0$ to $\boldsymbol{x}$ is less than a minimum reaction time $t_{\min}$; $\hat{P}(c_3|\boldsymbol{x}) \sim 1$ otherwise.
- **c_4: The pass is short enough to be performed accurately.** $\hat{P}(c_4|\boldsymbol{x}) \sim 0$ when $|\boldsymbol{x}-\boldsymbol{x}_0|$ is greater than a maximum distance $d_{\max}$; $\hat{P}(c_4|\boldsymbol{x}) \sim 1$ otherwise.

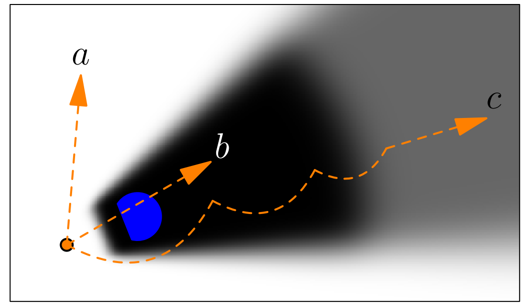

Figure 2: The estimated probability that a pass from the ball location (orange circle) will not be intercepted. This probability is high for locations with ball trajectories that pass far from opponents, such as a, and low for those with ball trajectories that pass close, such as b. Some passes, such as c, pass close to the opponent but can still be successful using chip passes, although the prior success probability for those is lower than for regular passes, as indicated by the gray region to the right.

- **c_5: Location $\boldsymbol{x}$ is reliable for pass reception.** $\hat{P}(c_5|\boldsymbol{x}) \sim 0$ when $\boldsymbol{x}$ is too close to the defense area, entrance into which is forbidden by the rules of SSL, to the boundary of the field, where passes run the risk of going out of bounds, or to other teammates, where teammates could interfere with R; $\hat{P}(c_5|\boldsymbol{x}) \sim 1$ otherwise.

An analogous approximation is computed for the probability of scoring a goal from location $\boldsymbol{x}$ given that the pass has been received. In this case, $\hat{P}(\text{goal} \mid \text{receive}, \boldsymbol{x})$ is a product of probabilities of the following conditions c_i':

- **c_1': Shots from $\boldsymbol{x}$ can reach the opposing goal faster than their goalkeeper can block them.** $\hat{P}(c_1'|\boldsymbol{x}) \sim 0$ if the shot time is greater than the time t_g the opposing goalkeeper takes to block an arbitrary point on the goal; $\hat{P}(c_1'|\boldsymbol{x}) \sim 1$ otherwise.
- **c_2': There is a wide enough open angle θ_g from $\boldsymbol{x}$ to the opposing goal.** $\hat{P}(c_2'|\boldsymbol{x}) \sim p_{g\,\min}$, for a constant prior $p_{g\,\min}$, when $\theta_g = 0$ ($p_{g\,\min} > 0$ because an angle may open up as robots move), and $\hat{P}(c_2'|\boldsymbol{x}) \to 1$ as $\theta_g \to \theta_{\max}$. When $\theta_g > \theta_{\max}$, $\hat{P}(c_2'|\boldsymbol{x}) = 1$, indicating that beyond a certain threshold, the value of θ_g has no influence on the probability of scoring. Figure 3 shows a visualization of $\hat{P}(c_2'|\boldsymbol{x})$.
- **c_3': R will have enough time to take a shot before the opponents block the shot.** $\hat{P}(c_3'|\boldsymbol{x}) \sim 1$ for locations where R can do a one-touch shot on the goal, while $\hat{P}(c_3'|\boldsymbol{x}) \sim p_{turn} < 1$, for some constant prior probability p_{turn}, when the robot needs to receive the ball, turn, and then shoot (only a two-touch shot is possible).
- **c_4': R will have enough time to take a shot before opponents steal the ball.** $\hat{P}(c_4'|\boldsymbol{x}) = 1$ for locations inside the opponents' defense area, where their defenders are not allowed to enter, while $\hat{P}(c_4'|\boldsymbol{x}) = p_{out} < 1$, for some constant prior probability p_{out}, when $\boldsymbol{x}$ is outside of the opponents' defense area.

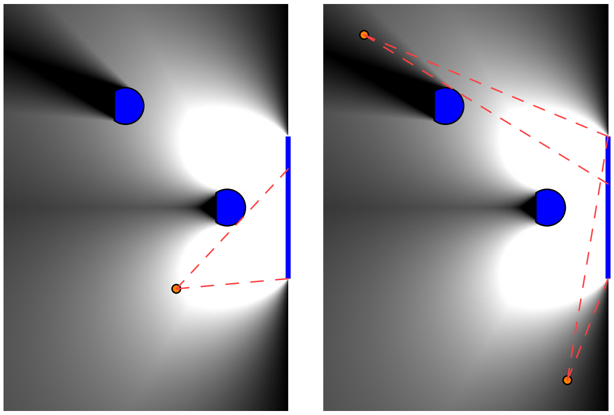

Figure 3: The estimated probability that a given location $\boldsymbol{x}$ has a wide enough open angle on the opponents' goal (blue line) to score a goal. The left image shows a location with a wide open angle, ideal for a shot. The right image shows two locations with relatively small open angles, one due to obstruction by a robot and distance from the goal, and the other because of its location near the corner of the field.

Equation 5 and its analogue for $\hat{P}(\text{goal} \,|\, \text{receive}, \boldsymbol{x})$ provide a value function for all potential receiving locations; the search for $\boldsymbol{x}^*$ is simply conducted by random sampling and evaluation of points. This is feasible since the space to search is a relatively small 2D space. Furthermore, at each time step, only locations close to the previous optimal are searched, to avoid big jumps in the target destination of R.

3.2 Pass-ahead Coordination

Following the receiver robot's selection for location $\boldsymbol{x}^*$, Passer P and receiver R coordinate the pass so that R arrives at $\boldsymbol{x}^*$ at approximately the same time the passed ball arrives at $\boldsymbol{x}^*$. P thus *passes ahead* to where R will be, rather than passing to where R is. The purpose of this coordination is to minimize the window of time in which the opponents can predict and block threats from the chosen location $\boldsymbol{x}^*$. Algorithm 1 describes the process of coordination in detail.

Algorithm 1 Pass-ahead coordination algorithm. Given a pass location $\boldsymbol{x}^*$, decides when P and R should start maneuvering.

```
T_r ← receiver navigation time to receive location x_r ≈ x*
T_s ← passer navigation time to kicking location x_p ≈ x_0
T_b ← minimum ball traversal time from source x_0 to x*
T_p ← T_s + T_b
if T_p ≥ T_r then
    P starts moving to x_p
end if
if T_r ≥ T_p then
    R starts moving to x_r
end if
if P is at x_p and T_r ≈ T_p then
    P shoots ball to x*
end if
```

This coordination can be clearly visualized using a Simple Temporal Network (STN) [13]. Figure 4 illustrates the STN with the time constraints for passing ahead. Note that there are some constraints where a robot could potentially wait indefinitely. However, we are interested in the lowest achievable time bounded by these constraints. In pass-ahead planning, we use time constraints as part of our multi-agent plan representation [14].

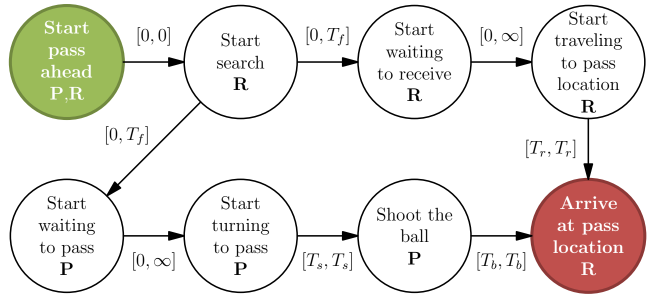

Figure 4: A Simple Temporal Network (STN) for pass-ahead. The letters inside each node indicate whether the passing agent (P) or receiving agent (R) is involved in that event. T_f indicates a maximum allowable time to search for a pass location.

Computations for Algorithm 1 and Figure 4 rely on the ability to accurately estimate robot navigation time to a given location and orientation, as described in Section 2. They also require an accurate estimate of the time the ball will take to traverse a specified distance when it is imparted with a specific initial velocity. This computation is based on a two-phase (sliding then rolling) model of the ball's trajectory. These models provided the necessary accuracy to succeed at coordinated passing ahead; such low-level coordination, combined with the higher-level planning of Section 4, led to the success of our multi-agent attacks in RoboCup 2013.

4. THE COERCE AND ATTACK PLANNER

We introduce a novel Coerce and Attack Planner (CAP) to plan attack sequences during free kicks. Robot soccer in the SSL, just like human soccer, involves free kicks during games, which are awarded for minor offenses such as losing the ball off the field, or for dribbling the ball for more than 50 cm. To initiate a free kick, the referee first issues a "stop" command to the teams to command all robots to stay at least 50 cm away from the ball while he/she places the ball where the offense occurred. The referee then waits until he/she deems that both teams are ready, and then sends the free kick signal. The team that is awarded the free kick then has up to 10 seconds to take the free kick by kicking the ball, and until the ball is kicked, all opponent robots must stay at least 50 cm away from the ball.

Since the opponents are not permitted to get closer than 50 cm to the ball until it is kicked, and since the free kick taker has 10 s to kick the ball, free kicks provide a convenient scenario for the team that is awarded the free kick to *plan* a progression of gameplay that might lead to a goal being scored. Additionally, there are certain characteristics of the defense that can be exploited to influence the plan. In general, there are two types of defending roles that the opponent robots may assume: "ball-following" roles and "robot-following" roles. The ball-following roles defend against direct shots on the goal from the ball, and hence are positioned as a function of the ball's location. The robot-following roles, on the other hand, attempt to block passes, and hence follow the attacking robots.

The CAP relies on these characteristics to plan a coordinated attack when awarded a free kick. The CAP *coerces* robot-following opponents into positions that leave strategically advantageous openings, allowing a teammate to *attack* by moving into the opening, receiving a pass, and shooting into the opponent's goal. The CAP interleaves planning, execution, and monitoring in the following seqeunce:

1. **Detect Opponent Tactics** (Monitoring): Tactic detection estimates the ball-following and robot-following tactic that each opponent robot is running.

2. **Compute Optimistic Attack** (Planning): Based on the detected tactics, the CAP computes an "optimistic attack" plan to score on the goal, considering only the ball-following opponents detected.

3. **Compute Coerce Plan** (Planning): Based on the detected tactics and the optimistic attack, the CAP computes a "coerce plan", placing attacking robots to coerce opponents away from the optimistic attack.

4. **Execute Coerce Plan** (Execution): The coercing robots are moved into the planned positions.

5. **Verify Tactic Models** (Monitoring): The placement of the opponents in response is observed.

6. **Compute Attack Plan** (Planning): If the actual positioning of the opponents differs from the expected positions of the coerce plan, then a new "attack plan" is computed, else the previously computed optimistic attack is used as the attack plan.

7. **Execute Attack Plan** (Execution): The CAP then commands the robots to execute the attack plan.

During the free kicks, out of the team of 6 robots, one must be the goalkeeper, and one is required to take the free kicks. Hence, the CAP must reason about how many of the remaining 4 robots should be assigned to the coerce plan, and how many to the attack plan. This allocation varies based on the opponent tactics detected, and in some cases, robots may be re-used for the coerce plan as well as the final attack plan, as we now explain.

In step 3, the CAP uses the number of robot-following opponents detected from step 1, to allocate as many robots to the coerce plan. The remaining robots are allocated to the optimistic attack plan. If there are insufficient remaining robots to allocate exclusively to the attack plan, then robots allocated to the coerce plan are re-used during the attack. We illustrate the allocation of robots by the CAP in two example scenarios.

Example 1. If the CAP detects 3 robot-following opponents in Step 1, it allocates 3 robots to coerce them away during Step 4 from the optimistic attack plan. The CAP then allocates the remaining 1 robot on the team to execute the attack plan during Step 7 using pass-ahead.

Example 2. If the CAP detects 4 robot-following opponents in Step 1, it allocates all 4 robots to coerce the 4 robot-following opponents away during Step 4 from the optimistic attack plan. The CAP then reuses one of these 4 coercing robots to execute the attack plan during Step 7.

4.1 Tactic Detection

In order to determine which of the opponents are running ball-following tactics and which are running robot-following tactics, the CAP estimates the tactic controlling each opponent robot. The tactics that the CAP detects are:

- **Goalkeeper**: a ball-following tactic that stays exclusively within the defense area to block direct shots on the goal,

- **Primary Defense**: a ball-following tactic that that stays on the perimeter of the defense area and always moves to cover the angle between the ball and the goal,

- **Mark**: a robot-following tactic that follows attacking robots to prevent them from receiving passes or shooting on the goal, and

- **Wall**: a robot-following tactic that stays as close as possible to the free kick taker to prevent it from passing to its teammates.

For every tactic t, given a world state W consisting of the locations of all the robots and the ball on the field, the model of the tactic M_t is used to compute the probability $P(p|M_t, W)$ that a robot positioned at location p would be running tactic t, modelled by M_t. In our work, $P(p|M_t, W)$ is computed analytically using assumptions of SSL-specific tactic behaviors and the rules of the SSL. A possible alternative would have been to estimate the probabilities numerically from logs [15, 16]. Figure 5 shows the probability distributions for the models of the tactics listed above. Let R be the set of opponent robots, and p_r denote the location of an opponent robot $r \in R$. The robots R_t running a tactic t for the current world state W are thus detected to be those robots in the set $R_t = \{r \in R \,|\, P(p_r|M_t, W) > \alpha_t\}$, where $\alpha_t \in [0, 1)$ is a threshold defined for detected tactic t. This means that the detected tactic for an opponent is the tactic that best explains its position on the field for the current world state.

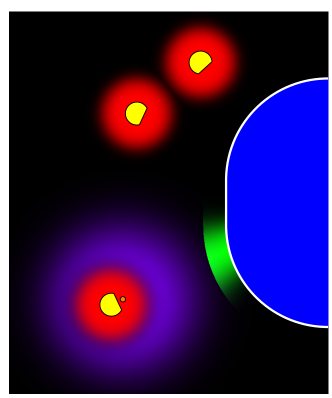

Figure 5: The probability distributions given by the models for various tactics opponent robots might run, including Mark (red), Wall (purple), Primary Defender (green), and Goalkeeper (blue). The defense area line is shown in white, our robots in yellow, and the ball in orange.

4.2 Computing Attack Plans

There are two steps of the CAP that involve computing attack plans: step 2, when computing the optimistic attack, and step 6, when computing the final attack plan. When computing the optimistic attack plans, the only opponents taken into account are the ones detected (during step 1, opponent tactic detection) to be running ball-following tactics. When computing the final attack plan, all opponents are taken into account.

An attack plan consists of a pass from the free kick taker to a pass receiver at a specific location on the field, and a subsequent shot on the goal by the pass receiver. The possible locations of the pass receiver are evaluated by discrete sampling on a grid of size 6 cells by 4 cells spanning the entire field. While this discrete sampling could be coarser or finer, we empirically evaluated this discretization to be an acceptable tradeoff between computational complexity and the granularity of the resulting plans. Each cell on the grid is evaluated for a possible pass location as discussed in Section 3.1. The cell with the highest probability of a goal being scored from it is then chosen as the pass location for the attack plan.

4.3 Computing The Coerce Plan

Once the optimistic attack plan is computed, the coerce plan is computed on the grid to place robots to coerce robot-following opponents away from the optimistic attack plan. For the coerce plan, every cell on the grid is evaluated as follows:

1. Consider placing one attacking robot in the cell.
2. Based on the detected robot-following opponents, estimate where the robot-following opponents will drive to, in response to our robot being placed in this cell.
3. Evaluate the "interference likelihood", defined as the likelihood of these robot-following opponents intercepting either the pass (Section 3.1) or the shot on the goal from the optimistic attack.

After these steps are performed for all possible cells, the coerce plan is then the set of those cells with the smallest values of interference likelihood. By sending attacking teammates to the cells in the coerce plan, the robot-following opponents are thus coerced into marking our robots, consequently leaving the optimistic attack plan free of interference.

4.4 Results

The CAP proved to be extremely useful at RoboCup 2013, and accounted for a large number of our passes during the games. To evaluate the effectiveness of the CAP, we reviewed the logs from the games, and counted the number of times the CAP was used, and the number of times successful passes were made resulting from the CAP. Table 1 lists the number of times that the CAP was used in the games to attempt a pass, the number of times a robot from the coerce plan was reused for the attack, and the number of successful passes made.

The games against Team D and Team E did not involve much use of the CAP and hence are not listed in this table. Robots in the coerce plan could only be reused for the actual attack if they were not marked by robot-following opponents. For the games with Team C and Team F, often at least one of the robots in the coerce plan was unmarked, and was therefore reused for the attack plan.

The success rate of passes during the games against Team G and Team B was lower than usual, as their defense strategies

Opponent	Attempted Passes	Successful Passes	Coerce Robot Reuses
Team G	11	6	0
Team F	6	5	4
Team C	16	13	6
Team H	11	11	0
Team B	4	2	0

Table 1: Results of passing from free kicks using the Coerce and Attack Planner during actual games at RoboCup 2013.

were more frequently able to respond swiftly to changes in the attack pattern from the coerce step to the attack step.

Based on the performance of the CAP, one promising direction of future work would be to model the expected motion of the opponents likely to occur *after* a pass is made in the final attack pattern. This could potentially result in the CAP preferring attack patterns that are more difficult to defend against, due to the required motion of the defense.

5. THREAT-BASED DEFENSE

The threat defense evaluator considers *threats*, which are computed based on the locations of the ball and opponent robots, and chooses locations to place defenders to defend against each of them. There are two kinds of threats: one *first-level threat* and multiple *second-level threats*.

Three distinct tactics work together to form a coordinated defense. The *goalkeeper* remains within the defense area, staying near the goal and defending against the first-level threat. *Primary defenders*, of which there are at most two at any given moment, always move along the edge of the defense area. They guard against the first-level threat if all of them are needed to do so, but one may guard against second-level threats if only one is needed for the first-level threat. *Secondary defenders* are placed away from the defense circle to guard against second-level threats.

5.1 First-level Threat

The first-level threat represents the location of the most immediate threat of a shot on our goal. It is defined to be either the location of the ball or, when the defense evaluator judges that a pass is imminent (as defined below), the location of one of the opponent robots.

A pass is defined to be imminent when the ball's speed is above a certain threshold, its velocity is not pointed toward our goal, and the defense evaluator judges that it may be headed toward an opponent robot which might be able to receive it soon. The determination of whether an opponent is in position to receive is made using a heuristic function based on the velocity of the ball and the vector from the ball to the robot. More precisely, for each opponent, its "risk of receiving" is given by

$$-\frac{||d||}{||v||} \cdot (1 + c \cdot (1 - \cos\theta)), \tag{6}$$

where c is an adjustable parameter, v is the velocity of the ball, d is the vector from the ball's location to the opponent's location, and θ is the (unsigned) angle between v and d. This expression is greater for positions near the ball than ones far away, and for positions which are in front of the ball's motion than for ones which are not. Examples of this evaluation are shown in Figure 6. If the highest of any opponent's risk of receiving is above a threshold, then the evaluator judges that the opponent is in position to receive.

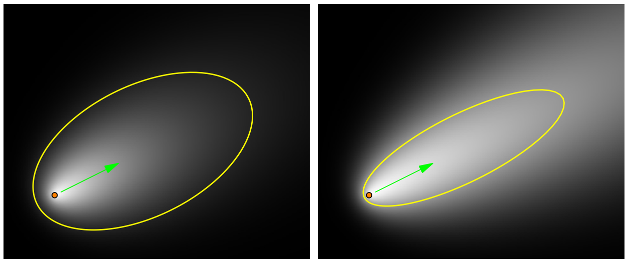

Figure 6: A The risk of receiving as a function of robot position, for a given ball position (orange circle) and velocity (green arrow), as computed by Eq. 6 with $c = 5$ (left) and $c = 20$ (right). The yellow ellipse represents an isocontour of constant risk.

When this happens, the first-level threat is the location of the opponent with highest risk of receiving. This means that when the opponent team makes a pass, it is possible to anticipate where it will be received, and immediately defend against that location, rather than continuously following the ball as it moves, which would result in slower responses.

Once the location of the first-level threat is computed, the defense evaluator decides how to position the goalkeeper and primary defenders to block all open angles on our goal. Primary defenders guarding against the first-level threat always have their target locations along the edge of our defense area. There are three cases that need to be considered:

- If one defender can block the entire open angle by itself: one defender stands along the bisector of the open angle, and the goalkeeper in front of the center of the goal.
- If one defender and the goalkeeper can block the open angle: one defender stands just inside the line from the threat location to the nearer corner of the goal; the goalkeeper stands along the bisector of the remaining open angle, as far back as possible.
- If two defenders and the goalkeeper are needed to cover the open angle: the goalkeeper stands along the bisector of the open angle, leaving two smaller open angles to either side of it; one defender stands along the bisector of each of these smaller angles. Figure 7 demonstrates these computations.

5.2 Second-level Threats

The second-level threats are the opponents which might be able to receive the ball from the first-level threat. For every such opponent, two potential defense locations are computed:

- A point on the line from the opponent to the center of our goal. The point is based on latency and acceleration, chosen so that if the opponent starts accelerating, our robot can respond fast enough so that the line to the goal is always blocked. This defends against passes between opponents.

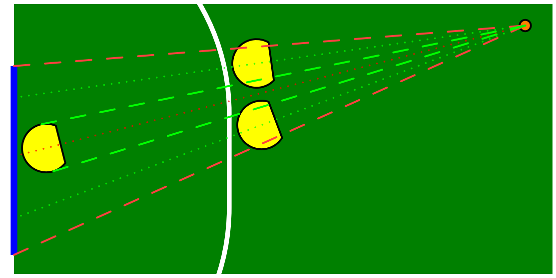

Figure 7: Placement computation for the primary defenders when two are required. The goalkeeper is placed along the bisector (red dotted line) of the angle from the ball to the goal (red dashed lines). The defenders are placed along the bisectors (green dotted lines) of the two remaining smaller angles (green dashed lines).

- The midpoint of the line segment from the opponent to the first-level threat. This defends against shots on our goal.

Once the set of positions is computed, they need to be assigned to defenders.

The positions are ranked according to the following criteria, given in decreasing order of priority (where "opponent" refers to the opponent robot which caused a position to be generated):

- Opponents which are closer to our side of the field than a configurable threshold are ranked higher than those which are not.
- Positions which block shots (as opposed to passes) are ranked higher.
- Opponents which have a larger available open angle on the goal are ranked higher. All angles larger than a configurable threshold are treated as equal.
- Opponents which will be able to shoot on the goal sooner are ranked higher. The time to shoot is given by the passing time plus the shot time.

The highest-ranked positions are then assigned to the remaining defenders, with each one greedily assigned to the nearest defender.

An exception to the assignments is when there is a "held" task. This occurs when a defender is blocking a goal shot from a second-level threat, and then the ball is passed toward that opponent, making it the first-level threat. In this case, the defender which is blocking the goal shot continues to block that shot until the primary defenders have moved into place to guard the new first-level threat.

6. PERFORMANCE AT ROBOCUP 2013

At RoboCup 2013, CMDragons played 7 games in total, and won all but the final game. We scored a total of 27 goals during regular gameplay and 7 goals from penalty kicks, while only 1 goal was scored on us during regular gameplay and 6 goals from penalty kicks. In addition to goals scored in each game, we can evaluate performance during the games using a number of metrics that evaluate the effectiveness of the defense and offense strategies of CMDragons. These metrics include:

1. **Offense Ratio**: the ratio of the game time that the ball was on the opponent's half of the field, to the game time that the ball was on our half of the field.

2. **Attack Ratio**: the ratio of the number of times our team attempted to shoot towards their goal, to the number of times the opponent attempted to shoot towards our goal.

The offense ratio indicates how often we were on the offensive rather than the defensive, and the attack ratio indicates how often we exploited opportunities to attempt to make shots on goal, compared to our opponents. Table 2 lists the scores and performance metrics for each of the games played, including the Round Robins (RR), Quarter Finals (QF), Semi-Finals (SF) and Finals (F). The game against EMEnents during the round robins was played against an empty field, and resulted in a winning score of 10 : 0 for CMDragons, so we do not include it in the table.

Opponent	Stage	Score	Offense Ratio	Attack Ratio
Team G	RR	2:0	1.38	1.82
Team F	RR	10:0	1.48	3.14
Team C	RR	2:1	2.57	1.9
Team E	QF	2:0	1.78	1.55
Team H	SF	2:0	1.13	1.75
Team B	F	2(4):2(5)	1.63	1.09

Table 2: Game scores and performance for the games played by CMDragons at RoboCup 2013. Scores are in the form CMDragons:Opponent.

The performance metrics from the logs of the RoboCup 2013 games reveal a number of interesting features. The offense ratios for all the games were greater than 1.0, indicating that the majority of the game times was spent attacking rather than defending. The attack ratios for all the games except for the finals were significantly greater than 1, indicating that our offense was more aggressive at attempting shots on the opponent's goal than the opponents' were on ours. The strategies of the opponents varied significantly across games. Team G [17] and Team B [6] had defense strategies that were very swift at responding to changes in our attack formations, particularly when transitioning from the Coerce step to the Attack step of the CAP. The Team F [18] attack strategy included a number of opportunistic attempts on our goal, which our defense intercepted and deflected to their goal. Thanks to the new dynamic ball interception skill (Section 2) and the strategic placements of the secondary defenders (Section 5.2), we successfully intercepted many passes between opponents, some of which even resulted in goals in the games against Team F [18], Team E [19] and Team B [6].

7. CONCLUSION

We introduced several contributions for our multi-robot team in the adversarial robot soccer domain, in particular a coordinated pass-ahead behavior, a Coerce and Attack planner, and a threat-based defense. We empirically demonstrated the combined effectiveness using several performance metrics over the logs of actual games at RoboCup 2013. Our future work includes focusing on additional opponent model learning, and incorporating direct input from a human. Furthermore, we look forward for our and others' investigations of the Coerce and Attack planner as a general technique in other adversarial scenarios, in which the planner explicitly drives the opponent to a state that enables a successful plan to be open for execution.

8. REFERENCES

[1] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso. SSL-vision: The shared vision system for the RoboCup Small Size League. In *RoboCup 2009 Symposium*, pages 425–436.

[2] R. Ros, M. Veloso, R.L. De Màntaras, C. Sierra, and J.L. Arcos. Retrieving and reusing game plays for robot soccer. In *Advances in Case-Based Reasoning*, pages 47–61. 2006.

[3] B. Browning, J. Bruce, M. Bowling, and M. Veloso. STP: Skills, tactics, and plays for multi-robot control in adversarial environments. *Proceedings of the IME, Part I: Journal of Systems and Control Engineering*, 219(1):33–52, 2005.

[4] M. Bowling, B. Browning, A. Chang, and M. Veloso. Plays as team plans for coordination and adaptation. In *RoboCup 2003 Symposium*, pages 686–693.

[5] T. Panyapiang et al. Skuba 2013 Extended Team Description. In *RoboCup 2013*.

[6] Y. Wu et al. ZJUNlict Extended TDP for RoboCup 2013. In *RoboCup 2013*.

[7] S.M. Mohaimanian Pour et al. Parsian Extended TDP for Robocup 2013. In *RoboCup 2013*.

[8] J. Bruce, S. Zickler, M. Licitra, and M. Veloso. Dynamic passing and strategy on a champion robot soccer team. In *ICRA 2008*, pages 4074–4079.

[9] A. Tews and G. F. Wyeth. Multi-robot coordination in the robot soccer environment. In *ACRA 1999*, pages 90–95.

[10] R. Nakanishi, J. Bruce, K. Murakami, T. Naruse, and M. Veloso. Cooperative 3-robot passing and shooting in the robocup small size league. In *RoboCup 2006 Symposium*, pages 418–425.

[11] S. Zickler and M. Veloso. Efficient physics-based planning: sampling search via non-deterministic tactics and skills. In *AAMAS 2009*, pages 27–33.

[12] T. Kalmár-Nagy, R. D'Andrea, and P. Ganguly. Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. *Robotics and Autonomous Systems*, 46:47–64, 2004.

[13] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1-3):61–95, 1991.

[14] P. Riley and M. Veloso. Planning for distributed execution through use of probabilistic opponent models. In *AIPS 2002*, pages 72–81.

[15] K. Han and M. Veloso. Automated robot behavior recognition. In *Robotics Research: The Ninth International Symposium, 2000*, pages 249–256.

[16] C. Erdogan and M. Veloso. Action selection via learning behavior patterns in multi-robot domains. In *IJCAI 2011*, pages 192–197.

[17] K. Yasui et al. RoboDragons 2013 Extended Team Description Paper. In *RoboCup 2013*.

[18] Ö.F. Varol et al. BRocks 2013 Team Description. In *RoboCup 2013*.

[19] S. Rodríguez et al. STOx's 2013 Team Description Paper. In *RoboCup 2013*.