Selectively Reactive Coordination for a Team of Robot Soccer Champions

Juan Pablo Mendoza, Joydeep Biswas, Philip Cooksey, Richard Wang,
Steven Klee, Danny Zhu and Manuela Veloso
School of Computer Science. Carnegie Mellon University
5000 Forbes Avenue. Pittsburgh, PA, 15213

Abstract

CMDragons 2015 is the champion of the RoboCup Small Size League of autonomous robot soccer. The team won all of its six games, scoring a total of 48 goals and conceding 0. This unprecedented dominant performance is the result of various features, but we particularly credit our novel offense multi-robot coordination. This paper thus presents our Selectively Reactive Coordination (SRC) algorithm, consisting of two layers: A coordinated opponent-agnostic layer enables the team to create its own plans, setting the pace of the game in offense. An individual opponent-reactive action selection layer enables the robots to maintain reactivity to different opponents. We demonstrate the effectiveness of our coordination through results from RoboCup 2015, and through controlled experiments using a physics-based simulator and an automated referee.

1 Introduction

The RoboCup 2015 robot soccer Small-Size League (SSL) consists of teams of six autonomous robots playing on a field of 9m×6m, with overhead cameras that observe the pose of the twelve players and of the orange golf ball used to play. These observations (gathered at 60Hz) are passed to each team's computer (Zickler et al. 2010), which runs the planning algorithms to choose actions for each individual team robot. Such actions are sent by radio (at 60Hz) to the robots for execution. The RoboCup SSL is a very complex multi-robot planning problem with clear goals to achieve, in a fast-paced adversarial environment, with inevitably non-deterministic real physical sensing and execution.

Many researchers, present authors included, have worked on this research problem (Veloso, Stone, and Han 2000; Veloso, Bowling, and Stone 2000; D'Andrea 2005; Bruce et al. 2008; Sukvichai, Ariyachartphadungkit, and Chaiso 2012; Li et al. 2015), making contributions in real-time sensing (Bruce and Veloso 2003) and control (Behnke et al. 2004), planning (Zickler and Veloso 2009), and teamwork (Stone and Veloso 1999), which have enabled the current RoboCup SSL games to be a fascinating demonstration of effective AI multi-robot planning algorithms under significant uncertainty. Every year, teams improve and the game conditions and rules change to increase the difficulty of the problem and challenge the autonomous planning algorithms (Weitzenfeld et al. 2015). This year, the CMDragons (see Figure 1), composed of the same robot hardware for the last 10 years, won the competition, scoring 48 goals and suffering 0 goals in 6 games. This level of performance had never been reached before in the league. While various defense and offense algorithms contributed to our result (Mendoza et al. 2015), we strongly credit our new coordinated, aggressive, and continuous attack, which we present here.

Figure 1: All the CMDragons 2015 robots (6 play at a time), champions of the RoboCup Small Size League of robot soccer. *(Hardware designed and built by Mike Licitra, and carefully maintained by Joydeep Biswas and Richard Wang.)*

In this paper, we contribute our Selectively Reactive Coordination (SRC) approach to tractably and effectively solve the coordinated soccer offense problem. Our SRC algorithm is composed of two layers: The *coordinated opponent-agnostic* layer enables the team to conduct potentially expensive optimizations offline to find multi-robot team plans that generally perform well. The *individual opponent-reactive action selection* layer is highly reactive to opponents within the constraints imposed by the coordination layer plans, and thus enables the team to adapt appropriately to opponent behavior.

While robot soccer is a specific planning problem, we believe many of the ideas we present generalize to other dynamic multi-robot domains –e.g., capture-the-flag (Atkin, Westbrook, and Cohen 1999), keepaway (Stone et al. 2006), rescue planning (Jennings, Whelan, and Evans 1997), and team patrolling (Agmon et al. 2008)– in which robots balance executing an agreed-upon team plan with reacting to changes in the environment. We hope that this paper inspires others to pursue the robot soccer problem, or to apply or extend our algorithms to other dynamic multi-robot domains.

2 Multi-Robot Offense Coordination: Statement and Overview

One of the core challenges of planning for a team of soccer robots consists of representing and reasoning about the opponent team. The level of opponent-reactivity of team plans can vary, as exemplified by two extremes: (i) The *purely reactive* team, which positions its robots completely in reaction to the adversary, is unable to carry out plans of its own and is susceptible to coercion (Biswas et al. 2014); (ii) The *open loop* team, which positions its robots ignoring the opponent's state, is unable to appropriately react to opponent behavior. In this work, we introduce a novel intermediate Selectively Reactive Coordination (SRC) algorithm that creates team plans of its own while also responding to the opponent. SRC combines an *opponent-agnostic team coordination* layer with an *opponent-reactive individual action evaluation and selection* layer. This section provides an overview of our SRCalgorithm, while Sections 3 and 4 describe in detail the two layers in the context of the CMDragons.

Coordination via Zones and Guard Locations. The SRC creates a skeleton of an opponent-agnostic multi-robot plan $\mathcal{P}$ to be followed by the team, composed by a set of *roles* $R = \{r_1, \ldots, r_n\}$ for a team of n robots. The roles capture *what* the team members should be doing and constrains *how* the robots should do it to adhere to the plan. The coordination layer performs two main functions: (i) selects a plan skeleton based on the state of the game, and then (ii) matches each robot ρ_i to a role r_j.

The CMDragons 2015 offense has two types of roles: one *Primary Attacker* (PA), and $(n-1)$ *Support Attackers* (SAs). The PA role is completely opponent-and-situation-driven, and is thus unconstrained by $\mathcal{P}$. The SAs move to maximize the estimated probability of the team scoring. Plan $\mathcal{P}$ constrains the behavior of each SA_i by (i) bounding its to motion a *zone* $\boldsymbol{z}_i \subset \mathbb{R}^2$, and (ii) assigning it a default target *guard location* $\boldsymbol{p}_i^0 \in \boldsymbol{z}_i$. Each element of a zone set $\boldsymbol{Z} = \left\{(\boldsymbol{z}_1, \boldsymbol{p}_1^0), \ldots, (\boldsymbol{z}_{n-1}, \boldsymbol{p}_{n-1}^0)\right\}$ is assigned to a SA in the team. A plan can consist of a single zone assignment $\mathcal{P} = \boldsymbol{Z}$, or of a sequence of such steps $\mathcal{P} = [\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_k]$. These plan-skeletons encode strategies that work well generally against various opponents. We search for effective plans offline, using extensive data and human knowledge.

Individual Action Selection. SRC considers the opponents and ball in the positioning of the PA and the SAs, leading to an opponent-reactive action selection layer for each robot, that maximizes the estimated probability of scoring a goal. Formally, we have our n offense robots $\mathcal{R} = \{\rho_1, \ldots, \rho_n\}$ and a team of adversary robots $\mathcal{R}^o = \{\rho_1^o, , \ldots, \rho_m^o\}$. We assume full knowledge of the observable state of the world $\boldsymbol{x} \in \boldsymbol{X}$ consisting of: Each of our robots' pose and velocity state $\boldsymbol{x}_i^\rho$, the opponent robots' pose and velocity state $\boldsymbol{x}_i^o$, the ball state $\boldsymbol{x}^b$, and the state $\boldsymbol{x}^g$ of the game, with information such as time left and score.

Table 1 shows the higher-level actions of the individual robots (as opposed to low-level actions, such as apply current to the wheel motors, or to the kicker), as used in the coordination approach. In our formulation, the PA executes *active* actions that manipulate the ball, while the SAs perform *passive* actions that do not involve ball manipulation.

Action	Effect	Type
$\mathrm{move}(\boldsymbol{p})$	Move to location $\boldsymbol{p}$	Passive
$\mathrm{getBall}$	Move to intercept ball	Active
shoot	Shoot ball to opponent's goal	Active
$\mathrm{pass}(\boldsymbol{p})$	Pass ball to location $\boldsymbol{p}$	Active
$\mathrm{dribble}$	Dribble ball to hold possession	Active

Table 1: Actions available to each robot. Active actions manipulate the ball, while Passive actions do not.

Each Support Attacker SA_i moves either to its guard location $\boldsymbol{p}_i^0 \in \boldsymbol{z}_i$, or to a location $\boldsymbol{p}_i^* \in \boldsymbol{z}_i$ to receive a pass from the PA, depending on whether the PA is ready to pass to SA_i. Pass location $\boldsymbol{p}^*$ is computed via optimization as the location within $\boldsymbol{z}_i$ that maximizes the probability of successfully receiving a pass from the PA and then scoring a goal by taking a shoot action.

The PA selects the optimal action $\boldsymbol{a}^*$ among the set of possible active actions A^a.The PA only considers passing to the $(n-1)$ locations $\boldsymbol{p}_i^*$ that the SAs have chosen. Therefore, the action space for the PA is $(\mathrm{getBall}, \mathrm{shoot}, \mathrm{dribble}, \mathrm{pass})$, which can be fully explored and evaluated to estimate the probability of scoring a goal for each action, and choose the optimal action $\boldsymbol{a}^*$.

Complete Overview of SRC algorithm Algorithm 1 presents the complete algorithm and refers to the rest of the paper. First, the algorithm *jointly* computes $(n-1)$ zones to assign to each SA. Then, each of the n fully-instantiated roles (one PA and $(n-1)$ SAs with zones) is *jointly* optimally assigned to each robot. Finally, each robot plans its actions *individually* within the constraints of its role.

Algorithm 1 Selective Reactive Coordination for Offense.
Input: State of the world $\boldsymbol{x}$.
Output: Individual robot actions.

function $\mathrm{PlanAction}(\boldsymbol{x})$
Instantiate roles r_i with zones $\boldsymbol{z}_i$ (Section 3)
$\{(\boldsymbol{z}_i, \boldsymbol{p}_i^0)\}_{i=1}^{n-1} \leftarrow \mathrm{ComputeZones}(\boldsymbol{x})$
$\{r_i\}_{i=1}^{n} \leftarrow [\mathrm{SA}(\boldsymbol{z}_1, \boldsymbol{p}_1^0), \ldots, \mathrm{SA}(\boldsymbol{z}_{n-1}, \boldsymbol{p}_{n-1}^0), \mathrm{PA}]$
Optimally assign roles (Section 3)
$\{(\rho_i, r_i)\}_{i=1}^{n} \leftarrow \mathrm{OptAssign}\left(\{r_i\}_{i=1}^{n}, \boldsymbol{x}\right)$
Choose actions individually (Section 4)
for i in $[1, 2, \ldots, n]$ **do**
$\boldsymbol{a}_i^* \leftarrow \mathrm{IndividualAction}(\rho_i, r_i, \boldsymbol{x})$
end for
end function

This layered joint-individual algorithm maintains tractability: $\mathrm{ComputeZones}$ is $O(n)$, $\mathrm{OptAssign}$ is $O(n^3)$, and $\mathrm{IndividualAction}$ is $O(n+m)$ for each robot, where m is the number of opponents. As the size of the team grows, the $\mathrm{OptAssign}$ step might need to be modified to maintain real-time planning.

(a) Coverage-zones for 3 SAs (b) Dynamic-zone assignment, step i (c) Dynamic-zone assignment, step $i+1$

Figure 2: Coordinated zone assignments for Support Attacker robots. White dashed lines show the zone boundaries; white and orange circles show our SAs and PA respectively. A pass from the PA in (b) triggers a change in zones to those in (c).

3 Role Assignment to Zones

This section addresses the problem of selecting zones and assigning robots to be PA or SAs based on such zones. We explore two selection approaches: *coverage-zones* and *dynamic-zones*. Throughout this paper, we assume that the number n of offense robots is known. Balancing offense and defense, a complex problem on its own, is beyond the scope of this paper, which focuses on offense coordination.

Coverage-zone Selection Our coverage-zone approach is based on an offline definition of zone sets $\boldsymbol{Z}_i$, each of which cover the opponent's half of the field. Online, the team chooses the right coverage set $\boldsymbol{Z}$ based on features of the state of the game, such as possession and ball position. This approach follows a long tradition in robot soccer of building upon human knowledge of the game of soccer to reason about zones and formations (Stone and Veloso 1999).

Figure 2a shows a set of Coverage-zones for a four-robot offense used at RoboCup 2015. These predefined zone sets partition the offensive half of the field, giving much freedom to the individual SAs to search for the optimal positioning within these large zones, while ensuring a well-distributed opponent-agnostic formation along the field.

Dynamic-zone Selection The other approach we employed in RoboCup 2015 relies more heavily on coordinated zone selection to determine the flow of actions, leaving little positioning choice to the individual robots. The algorithm coordinates the team of robots to move, as the play progresses, in *sequences of zones* $\mathcal{P} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_k]$, each of a smaller size than the coverage-zones.

Each plan $\mathcal{P}_i$ is selected from a set $\mathbb{P}$ of possible plans. To create $\mathbb{P}$, we first created a set $\mathbb{P}_0$ of candidate plans, leveraging human knowledge and intuition about the game. Then, we ran extensive simulation tests to find the best-performing plans from $\mathbb{P}_0$, according to various performance metrics, such as goals scored and pass completion. Automated plan generation and selection is a subject for future work.

Each of the plans in $\mathbb{P}$ has a set of applicability conditions. For instance, in RoboCup 2015, the set $\mathbb{P}$ was divided into defense, midfield, and offense plans. In general, the goal of these plans is to move the ball toward the opponent's goal. Transitions from $\boldsymbol{Z}_i$ to $\boldsymbol{Z}_{i+1}$ are triggered when either a robot kicks the ball or a timeout period expires. Figures 2b shows a set of dynamic zones for a 3-robot offense, which evolves to that of Figure 2c when the PA passes the ball.

Optimal role assignment Once the set of roles has been fully instantiated with zones, our algorithm assigns each robot to a role. This assignment is performed optimally, given an appropriate cost function $C_i(\rho_j)$ of assigning role $r_i \in R$ to robot $\rho_j \in \mathcal{R}$: The optimal assignment is a bijection $f : R \to \mathcal{R}$, such that the total assignment cost $\sum_i C_i(f(r_i))$ is minimized. This optimal assignment can be computed in $O(n^3)$ time (Bertsekas 1981).

The definition of the cost function C_i is crucial to achieve the right assignments. The optimal assignment is the one that maximizes the probability of scoring a goal. In our algorithm, we approximate this by a cost function that represents the *time* $t_i(\rho_j)$ that it would take robot ρ_j to fulfill role r_i, multiplied by an importance factor w_i:

$$C_i(\rho_j) = w_i t_i(\rho_j) \tag{1}$$

Our robots are homogeneous, and thus there is no intrinsic benefit of choosing one over another for a specific role. Thus, the assignment that maximizes the probability of scoring is the one that minimizes the probability of the opponent disrupting our plans. This probability highly correlates with the time taken to perform a plan, and thus we minimize the total completion time, giving a higher importance w_i to the PA than to the SAs.

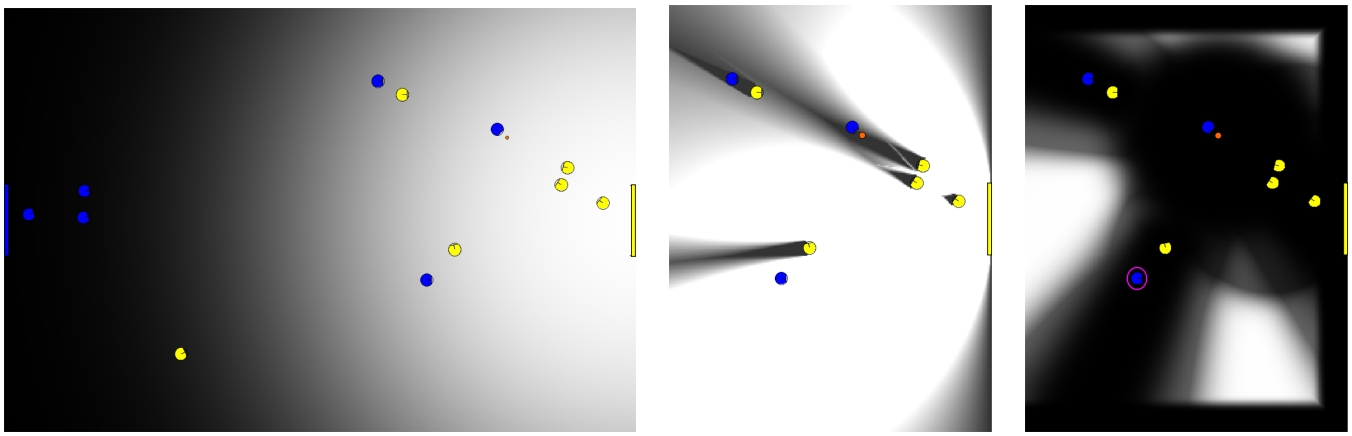
The cost of assigning the role of Primary Attacker to robot ρ_j is thus the time that it will take for ρ_j to drive to location $\boldsymbol{p}^*_{\text{PA}}$ computed to be the best ball interception location for ρ_j (or 0 if ρ_j is already in possession of the ball). Similarly, the cost for the Support Attackers is computed as the time it will take for robot ρ_j to drive to location $\boldsymbol{p}^*_{ij}$ evaluated to be the best location within zone $\boldsymbol{z}_i$ to support the PA.

4 Opponent-Reactive Individual Action Evaluation and Selection

Once the coordinated team has assigned each robot a fully instantiated role (see Section 3), each robot performs its role individually, with limited communication with the rest of the team (Browning et al. 2005). This individualization enables our algorithms to scale tractably with the number of robots.

4.1 Primary Attacker (PA)

The PA is the most complex role, as it requires the robot to make various decisions and use several different skills. For clarity of presentation, we present the PA at a level of



(a) $P(\text{near}\,|\boldsymbol{p})$ (b) $P(\text{open}\,|\boldsymbol{p},\boldsymbol{x})$ (c) $P(\text{receive}\,|\boldsymbol{p},\boldsymbol{x})$

Figure 3: Estimated probabilities for individual decision-making of the blue robots. Lighter gray indicates higher probability points. Probability that $\boldsymbol{p}$ (a) is near enough to the yellow goal and (b) has a wide enough angle on it, to shoot and score, and (c) Probability that the highlighted SA can receive a pass at different locations $\boldsymbol{p}$ from the PA holding the orange ball.

abstraction suitable for this paper; for example, even though the PA has various skills for intercepting a moving ball, here we join them into a single getBall action.

The goal of the PA is to manipulate the ball to maximize the probability of scoring a goal. When the PA does not have possession of the ball, it executes the getBall action. When the PA has possession, it chooses among three action types: shoot on goal (shoot), pass to a Support Attacker SA_i (pass_i), or individually dribble the ball (dribble). We briefly explain the dribble action, new to the CMDragons 2015.

Individual Dribbling. To hold possession of the ball, the PA uses a rotating dribbler bar that imparts back-spin on the ball, making it roll toward the robot. Furthermore, the PA drives with the ball to keep it away from opponents, while driving to the location $\boldsymbol{p}_t$ of the most promising pass or shoot option (as computed in Equation 4 or 3 below). Thus, the PA, with location $\boldsymbol{p}_{\text{PA}}$, balances the goal of driving in the direction $\vec{u}_t = \boldsymbol{p}_t - \boldsymbol{p}_{\text{PA}}$ of its target, and avoiding the closest opponent with direction $\vec{u}_o = \boldsymbol{p}^o - \boldsymbol{p}_{\text{PA}}$, if its distance $d_o = |\boldsymbol{p}^o - \boldsymbol{p}_{\text{PA}}|$ is smaller than a threshold $D_{\min}$. Figure 4 shows a diagram of these quantities.

The robot aligns with $\vec{u}_t$ by rotating its heading $\vec{u}_b$ towards $\vec{u}_t$ along the smaller angle ϕ^- unless there exists a *turning threat* $\text{threat}(\vec{u}_b, \vec{u}_o, d_o)$ within ϕ^-, in which case it rotates along the larger angle ϕ^+. A *turning threat* exists if the opponent is within ϕ^-, and closer than $D_{\min}$:

$$\begin{aligned}\text{threat}(\vec{u}_b, \vec{u}_o, d_o) = (d_o \le D_{\min}) \wedge \\ ((\vec{u}_b \times \vec{u}_o)(\vec{u}_b \times \vec{u}_t) \ge 0) \wedge \\ ((\vec{u}_t \times \vec{u}_o)(\vec{u}_t \times \vec{u}_b) \ge 0)\end{aligned} \quad (2)$$

Once aligned, the robot dribbles that ball towards $\boldsymbol{p}_t$, while avoiding obstacles along the way (Bruce and Veloso 2006).

Primary Attacker Algorithm. Algorithm 2 shows the procedure for choosing the optimal PA action. The PA estimates the probability that each one of these actions will lead to a goal, given the location of the ball $\boldsymbol{p}_b$ and the state of the world[1] $\boldsymbol{x}$. The probability of scoring a goal by shooting is estimated as the probability that the ball is close enough to the opponent's goal for a shot to be effective *and* that the robot has a wide enough angle on the goal:

$$P(\text{goal}\,|\,\text{shoot}, \boldsymbol{p}_b, \boldsymbol{x}) = P(\text{near}\,|\boldsymbol{p}_b) P(\text{open}\,|\boldsymbol{p}_b, \boldsymbol{x}). \quad (3)$$

Figures 3a and 3b illustrates these two functions, treated as independent for simplicity.

The probability of scoring a goal by first passing is estimated as the probability that the pass will successfully reach its target robot *and* that the robot will subsequently successfully shoot on the goal from the estimated world state $\boldsymbol{x}'_i$ after the pass, obtained from forward predictions of our own robots, and assuming the opponents are static. This probability is highly dependent on the location $\boldsymbol{p}^*_i$ at which robot SA_i decides to receive the pass:

$$\begin{aligned}P(\text{goal}\,|\,\text{pass}_i, \boldsymbol{p}^*_i, \boldsymbol{x}) = P(\text{receive}_i\,|\boldsymbol{p}^*_i, \boldsymbol{x}) \times \\ P(\text{goal}\,|\,\text{shoot}, \boldsymbol{p}^*_i, \boldsymbol{x}'_i).\end{aligned} \quad (4)$$

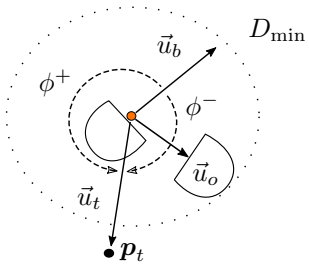


Figure 4: Variables used to execute the dribble action. The PA holding the orange ball decides how to drive to $\boldsymbol{p}_t$ while avoiding the opponent in direction $\vec{u}_o$.

[1]While the location of the ball is part of $\boldsymbol{x}$, we state it explicitly for clarity of explanation in the remainder of the paper.

Algorithm 2 Primary Attacker action-selection algorithm.
Input: Robot ρ_i, instantiated role PA, world state $\boldsymbol{x}$
Output: Chosen individual action $\boldsymbol{a}^*$.

function INDIVIDUALACTION(ρ_i, PA, $\boldsymbol{x}$)
  if not in possession of the ball **then**
    $\boldsymbol{a}^* \leftarrow \text{getBall}$
  else
    $A \leftarrow \{\text{shoot}, \text{dribble}, \text{pass}_1, \ldots, \text{pass}_{n-1}\}$
    $\boldsymbol{a}^* \leftarrow \arg\max_{\boldsymbol{a} \in A} [P(\text{goal} \,|\boldsymbol{a}, \boldsymbol{x})]$
  end if
  return $\boldsymbol{a}^*$
end function

This approximation is a one-step lookahead that assumes the receiving robot will shoot on the goal. Estimating further pass success probabilities is computationally expensive and inaccurate in a highly dynamic adversarial domain (Zickler and Veloso 2010; Trevizan and Veloso 2012). However, as these probabilities are estimated at every timestep, multiple passes emerge naturally. The estimated pass success probability $P(\text{receive}_i \,|\boldsymbol{p}_i^*, \boldsymbol{x})$ itself is a composition of various probabilities, as described in Section 4.2.

The probability of scoring by dribbling (to be followed by a pass or shot) is estimated by a constant value k_d:

$$P(\text{goal} \,|\, \text{dribble}, \boldsymbol{p}_b, \boldsymbol{x}) = k_d \tag{5}$$

This simple estimate provided significant results during RoboCup 2015: Our PA used dribble 42 times in the semi-final and 60 times in the final, giving our team 47 and 146 seconds of additional ball possession time, respectively.

4.2 Support Attackers (SAs).

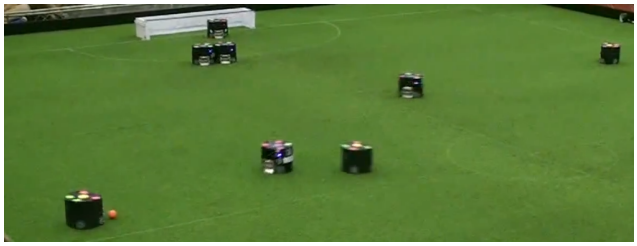
The task of each Support Attacker SA_i is to maximize the probability of the team scoring by supporting the PA from within its assigned zone $\boldsymbol{z}_i$. Each SA_i thus (i) searches for the location $\boldsymbol{p}_i^*$ that maximizes the probability of receiving a pass and then scoring a goal (Equation 4) and then (ii) moves to $\boldsymbol{p}_i^*$ at the right time to receive a pass from the PA. We now describe these two components and the resulting algorithm.

Optimal Pass Location Search. To find the location $\boldsymbol{p}_i^*$ that maximizes Equation 4, we must compute estimates of the two factors in it. Section 4.1 addresses the computation of $P(\text{goal} \,|\, \text{shoot}, \boldsymbol{p}, \boldsymbol{x}')$. To estimate the probability $P(\text{receive}_i \,|\boldsymbol{p}, \boldsymbol{x})$ of successfully receiving a pass at location $\boldsymbol{p}$, we compile a set of conditions c_k that must all be true for a pass to be received, and combine their individual probabilities assuming independence (Biswas et al. 2014):

$$P(\text{receive}_i \,|\boldsymbol{p}, \boldsymbol{x}) = \prod_k P(c_k | \boldsymbol{p}, \boldsymbol{x}). \tag{6}$$

Examples of these individual factors include the probability of the pass not being intercepted by any opponent, and the probability of not losing the ball by attempting to receive too close to the field boundary. Figure 3c shows an example of the resulting map from location on the field to probability of success. Because the optimization space is only 2-dimensional, we effectively employ random sampling to find the optimal location $\boldsymbol{p}_i^*$ that maximizes Equation 4.

Pass-ahead Computation After choosing $\boldsymbol{p}_i^*$, SA_i and PA execute the pass using a *pass-ahead* procedure: Only once SA_i has calculated that its own navigation time $t_\rho(\boldsymbol{p}_i^*)$ is comparable to the time $t_b(\boldsymbol{p}_i^*)$ that the pass will take to get there, SA_i moves to $\boldsymbol{p}_i^*$. Until then, SA_i moves to (or stays at) its guard location $\boldsymbol{p}_i^0$. Thus, we combine an opponent-agnostic default location $\boldsymbol{p}_i^0$ that enables our robots to move the world to a less dynamic and thus more predictable state, with an opponent-reactive pass location $\boldsymbol{p}_i^*$ that enables our robots to adapt appropriately. This pass-ahead coordination (Biswas et al. 2014) has been crucial in the high pass success rate of the CMDragons, as it makes the task of marking our SAs significantly more difficult. Figure 5 illustrates a pass-ahead maneuver leading to a goal in RoboCup 2015.



(a) Pass initial configuration

(b) Pass final configuration, immediately preceding a goal

Figure 5: Pass-ahead maneuver leading to a goal in RoboCup 2015. The figure shows the initial and final world configurations, and the motion of the ball and pass receiver.

Secondary Attacker Algorithm. Algorithm 3 describes the procedure for choosing the optimal SA action. While robots choose their individual actions independently, we enable limited communication to avoid computation redundancy. For example, SA robots communicate to the PA robots their computed values for $\boldsymbol{p}_i^*$, $P(\text{goal} \,|\, \text{pass}_i, \boldsymbol{p}_i^*, \boldsymbol{x})$, $t_b(\boldsymbol{p}_i^*)$ and $t_\rho(\boldsymbol{p}_i^*)$.

5 RoboCup 2015 and Simulation Results

Our layered coordinated offense approach proved extremely successful during the RoboCup 2015 competition. Here, we discuss the CMDragons' performance in the tournament, and conduct simulation experiments to further support SRC.

CMDragons Performance in RoboCup 2015 The RoboCup 2015 SSL tournament consisted of 17 teams from various universities in the world. The CMDragons played three games during the Round Robin stage (RR1, RR2, RR3), one Quarter-Final (QF), one Semi-Final (SF),

Algorithm 3 Support Attacker action-selection algorithm.
Input: Robot ρ_i, instantiated role $\mathrm{SA}(\boldsymbol{z}_i, \boldsymbol{p}_i^0)$, world state $\boldsymbol{x}$
Output: Chosen individual action $\boldsymbol{a}^*$.

function INDIVIDUALACTION($\rho_i, \mathrm{SA}(\boldsymbol{z}_i, \boldsymbol{p}_i^0), \boldsymbol{x}$)
  $\boldsymbol{p}_i^* \leftarrow \mathrm{FindOptLoc}(\rho_i, \boldsymbol{z}_i, \boldsymbol{x})$
  if $t_b(\boldsymbol{p}_i^*) \leq t_\rho(\boldsymbol{p}_i^*)$ **then**
    $\boldsymbol{a}^* \leftarrow \mathrm{move}(\boldsymbol{p}_i^*)$
  else
    $\boldsymbol{a}^* \leftarrow \mathrm{move}(\boldsymbol{p}_i^0)$
  end if
  return $\boldsymbol{a}^*$
end function

and the Final (F). We won all 6 games, scoring 48 goals, and conceding 0. Table 2 summarizes goal, shot, and pass statistics for our team in each game.

Game	Shots			Passes		
	Scored	Missed	Succ.%	Compl.	Missed	Succ.%
RR1	6	11	35.3	32	9	78.1
RR2	10	5	66.7	14	4	77.8
RR3	10	15	40.0	30	7	81.1
QF	15	25	37.5	38	4	90.5
SF	2	29	6.5	51	12	81.0
F	5	15	25.0	29	15	65.9
Total	48	100	32.4	194	51	79.2
Avg	8	16.7		32.3	8.5	

Table 2: Statistics for each CMDragons game in RoboCup 2015. RR2 and RR3 ended early due to a 10-goal mercy rule, after 10:20 and 18:25 minutes respectively (normally, games last 20 minutes).

Our team demonstrated an effective level of coordination: with an average of 32.3 passes completed per game, our team had a 79.2% pass completion rate[2]. Furthermore, most of the team's goals were collective efforts: 22 goals were scored directly after 1 pass, 11 directly after 2 consecutive passes, and 1 after 3 consecutive passes.

Simulation Validation We complement our real-world results from RoboCup with experiments run on a PhysX-based simulator, with an automated referee (Zhu, Biswas, and Veloso 2015) that enabled extensive testing without human intervention. We test our novel SRC algorithm against two alternate highly competitive versions of our team described below. We tested each condition for over 500 minutes of regular game-play, with no free kicks, fitting the focus of this paper. While it is always difficult to clearly dominate over other versions of our team by changing a single aspect of the team, the SRC algorithm showed a significant improvement over the alternatives.

First, we tested our team using SRC against a team in which each SA_i moves to its Individually-Optimal Location $\boldsymbol{p}_i^*$ computed over the entire field (team IndivOpt). Thus, IndivOpt has the advantage of globally-optimal individual positioning, while SRC has the advantage of team coordination. Table 3 shows that the SRC offense outperforms team IndivOpt in terms of offensive statistics.

Team	Goals Scored	Blocked (Goalie)	Blocked (Other)	Total Shots
SRC	59	128	1636	1823
IndivOpt	35	85	1381	1501
SRC	67	165	2045	2227
ExactPlan	53	142	2453	2648

Table 3: Results of simulation experiments. Our SRC algorithm outperforms a team that positions robots in their Individually-Optimal Location (IndivOpt), and a team that positions them according to an Exact Team Plan (ExactPlan)
.

Then, we tested SRC against a team in which the coordination layer, instead of creating only a skeleton of a team plan, creates an Exact Team Plan (team ExactPlan); this plan assigns robots to specific locations, rather than zones as in SRC. Thus, ExactPlan has the advantage that effective complete plans can be specified in advance, but it lacks the reactivity of SRC. Similar exact plan strategies have been successfully applied by highly competitive teams in SSL (Zhao et al. 2014). Table 3 shows that SRC outperforms ExactPlan: even though ExactPlan shoots more, SRC shoots past the defense more and scores more. We hypothesize that ExactPlan shoots more in general because its PA is less likely to find good passing options, and decides to shoot instead.

6 Conclusion

This paper presents a Selectively Reactive Coordination (SRC) approach to the problem of offensive team coordination in the context of robot soccer. This approach achieves a tradeoff between the ability to create team plans independently of the opponents, and the ability to react appropriately to different opponent behaviors.

An opponent-agnostic coordination layer creates skeletons of team plans that are generally effective against various opponents. These plans are encoded by zones to be assigned to the Support Attacker robots, while the Primary Attacker robot is unconstrained by the plan. Team plans can be generated offline, and can therefore leverage human knowledge or extensive computation. During the game, plans are selected based on efficiently-computable conditions.

Given the constraints imposed by the selected team plan, each robot plans its actions individually. At this level, decisions are highly reactive to the behavior of the opponents.

We present empirical support for our approach through statistics of our unprecedented performance in RoboCup 2015, and through controlled experimental results obtained using a physics-based simulator and an automated referee.

Acknowledgments This research was partially supported by ONR grant number N00014-09-1-1031 and AFRL grant number FA87501220291. The views and conclusions contained in this document are those solely of the authors.

[2]For anecdotal reference, the pass completion rates of the human teams in the 2014 World Cup final were 71% and 80%.

References

Agmon, N.; Kraus, S.; Kaminka, G.; et al. 2008. Multi-robot perimeter patrol in adversarial settings. In *Proceedings of the International Conference of Robotics and Automation (ICRA)*, 2339–2345.

Atkin, M. S.; Westbrook, D. L.; and Cohen, P. R. 1999. Capture the flag: Military simulation meets computer games. In *Proceedings of AAAI Spring Symposium Series on AI and Computer Games*, 1–5.

Behnke, S.; Egorova, A.; Gloye, A.; Rojas, R.; and Simon, M. 2004. Predicting away robot control latency. In *RoboCup 2003: Robot Soccer World Cup VII*, 712–719. Springer.

Bertsekas, D. P. 1981. A new algorithm for the assignment problem. *Mathematical Programming* 21(1):152–171.

Biswas, J.; Mendoza, J. P.; Zhu, D.; Choi, B.; Klee, S.; and Veloso, M. 2014. Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 493–500.

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

Bruce, J., and Veloso, M. 2003. Fast and Accurate Vision-Based Pattern Detection and Identification. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Bruce, J. R., and Veloso, M. M. 2006. Safe multirobot navigation within dynamics constraints. *Proceedings of the IEEE, Special Issue on Multi-Robot Systems* 94(7):1398–1411.

Bruce, J.; Zickler, S.; Licitra, M.; and Veloso, M. 2008. CMDragons: Dynamic passing and strategy on a champion robot soccer team. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 4074–4079.

D'Andrea, R. 2005. The Cornell RoboCup Robot Soccer Team: 1999–2003. In *Handbook of Networked and Embedded Control Systems*. Springer. 793–804.

Jennings, J.; Whelan, G.; and Evans, W. 1997. Cooperative search and rescue with a team of mobile robots. In *Proceedings of the International Conference on Advanced Robotics (ICAR)*, 193–200.

Li, C.; Xiong, R.; Ren, Z.; Tang, W.; and Zhao, Y. 2015. ZJUNlict: RoboCup 2014 Small Size League Champion. In *RoboCup 2014: Robot World Cup XVIII*. Springer. 47–59.

Mendoza, J. P.; Biswas, J.; Zhu, D.; Wang, R.; Cooksey, P.; Klee, S.; and Veloso, M. 2015. CMDragons2015: Coordinated Offense and Defense of the SSL Champions. In *Proceedings of the International RoboCup Symposium*.

Stone, P., and Veloso, M. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence* 110(2):241–273.

Stone, P.; Kuhlmann, G.; Taylor, M. E.; and Liu, Y. 2006. Keepaway soccer: From machine learning testbed to benchmark. In *RoboCup 2005: Robot Soccer World Cup IX*. Springer. 93–105.

Sukvichai, K.; Ariyachartphadungkit, T.; and Chaiso, K. 2012. Robot hardware, software, and technologies behind the SKUBA robot team. In *RoboCup 2011: Robot Soccer World Cup XV*. Springer. 13–24.

Trevizan, F., and Veloso, M. 2012. Trajectory-Based Short-Sighted Probabilistic Planning. In *Proceedings of NIPS-12*.

Veloso, M.; Bowling, M.; and Stone, P. 2000. The CMUnited-98 Champion Small-Robot Team. *Advanced Robotics* 13(8).

Veloso, M.; Stone, P.; and Han, K. 2000. The CMUnited-97 Robotic Soccer Team: Perception and Multiagent Control. *Robotics and Autonomous Systems* 29 (2-3):133–143.

Weitzenfeld, A.; Biswas, J.; Akar, M.; and Sukvichai, K. 2015. RoboCup Small-Size League: Past, Present and Future. In *RoboCup 2014: Robot World Cup XVIII*. Springer. 611–623.

Zhao, Y.; Xiong, R.; Tong, H.; Li, C.; and Fang, L. 2014. ZJUNlict: RoboCup 2013 Small Size League Champion. In *RoboCup 2013: Robot World Cup XVII*.

Zhu, D.; Biswas, J.; and Veloso, M. 2015. AutoRef: Towards Real-Robot Soccer Complete Automated Refereeing. In *RoboCup 2014: Robot World Cup XVIII*. Springer. 419–430.

Zickler, S., and Veloso, M. 2009. Efficient Physics-Based Planning: Sampling Search Via Non-Deterministic Tactics and Skills. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Zickler, S., and Veloso, M. 2010. Variable Level-of-Detail Motion Planning in Environments with Poorly Predictable Bodies. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*.

Zickler, S.; Laue, T.; Birbach, O.; Wongphati, M.; and Veloso, M. 2010. SSL-vision: The shared vision system for the RoboCup Small Size League. In *RoboCup 2009: Robot Soccer World Cup XIII*. Springer. 425–436.