Opponent-Aware Ball-Manipulation Skills for an Autonomous Soccer Robot

Philip Cooksey(✉), Juan Pablo Mendoza, and Manuela Veloso

Carnegie Mellon University, Pittsburgh, PA, USA
pcooksey@andrew.cmu.edu

Abstract. Autonomous robot soccer requires effective multi-agent planning and execution, which ultimately relies on successful skill execution of individual team members. This paper addresses the problem of ball-manipulation for an individual robot already in possession of the ball. Given a planned pass or shoot objective, the robot must intelligently move the ball to its target destination, while keeping it away from opponents. We present and compare complementary ball-manipulation skills that are part of our CMDragons team, champion of the 2015 RoboCup Small Size League. We also present an approach for selecting the appropriate skill given the state of the world. To support the efficacy of the approach, we first show its impact in real games through statistics from the RoboCup tournament. For further evaluation, we experimentally demonstrate the advantages of each introduced skill in different sub-domains of robot soccer.

1 Introduction

The RoboCup Small-Size League (SSL) is a multi-robot domain consisting of teams of six robots that play soccer in a highly dynamic and adversarial domain. Overhead cameras track the positions of the ball and each robot, which are fed into a centralized computer shared by both teams [8]. Each team must autonomously coordinate their robots and manipulate the ball to score more goals than the opponent and win. Both team coordination and individual skills are important aspects of this problem. This paper addresses the latter, focusing on planning and execution for opponent-aware ball manipulation.

To plan tractably in a domain as complex and time-sensitive as robot soccer, one can separate the team planning aspect of the problem –e.g., to whom and where the robot controlling the ball should pass– from the execution of the plan –e.g., how to pass/move the ball to the chosen teammate– [3]. This paper focuses on the individual's plan and the execution of it: we address the problem of a robot that is tasked with moving the ball to a specific target location under opponent pressures. This location is assumed to be given by a separate team planner, but the robot can evaluate different methods of achieving its task, given its ball-manipulation skills.

To manipulate the ball, most teams in SSL have converged to similar mechanisms: a kicker to impart momentum on the ball, and a dribbler bar to dribble

S. Behnke et al. (Eds.): RoboCup 2016, LNAI 9776, pp. 84–96, 2017.
https://doi.org/10.1007/978-3-319-68792-6_7

Fig. 1. CMDragon's robot, with a ball touching its dribbler bar (horizontal black cylinder). The dribbling bar can be spun to put backspin on the ball for semi-control.

the ball (see Fig. 1). Their ball-manipulation skills thus depend on these mechanisms, and the optimal skill depends on the state of the opponents: kicking the ball to its target is a highly accurate method of moving the ball, provided no opponents are nearby to steal the ball before the kick, or to intercept it before it reaches its destination. Alternatively, the robot can dribble the ball to a better location before kicking, which is less reliable in the absence of opponents, due to the risk of losing the ball while dribbling, but may be better than directly kicking it if there are opponents nearby.

This paper illustrates the effect of different skills, by using the mechanisms above in different complementary ways to create opponent-aware ball-manipulation plans. First, we specify four macro-skills that the robot can take: align to shoot the ball, align to shoot using the dribbler, move the ball to a more beneficial location using the dribbler, and kick the ball. We define in detail the algorithms and physical limitations of these skills. Then, we use a skill decision algorithm to select among these skills depending on the state of the opponents.

We provide evidence of the efficacy of our approach using two methods: statistics gathered from the RoboCup 2015 competition, and in-lab experiments. The RoboCup statistics provide evidence of the effectiveness of the approach in real competitive games. To collect experimental data in a more controlled setup, we ran repeated experiments of various soccer scenarios that illustrate the advantages of each of the defined skills. These experiments show that the various skills are successful in different scenarios, which supports the need for an opponent-aware decision process among the skills, as well as within each skill.

Research into accurate dribbling has been previously studied as a way of maintaining control over the ball while navigating. Researchers have used modified potential fields to avoid non-moving obstacles along with constraints on motion to dribble in the RoboCup Middle-Size League [4]. Similar to our omni-directional soccer robots, researchers have analyzed the kinematics and control needed for dribbling a ball along a path [6]. The only research that models the dynamics of a multi-body environment uses a physics-based robot motion planner [9]. The downfall of this approach is the enormous computational cost of modeling and predicting every robot in the environment. Our work is unique in that it focuses on developing a method of ball-manipulation with opponent awareness while still being computationally feasible in real-time.

2 Problem

In the robot soccer problem, each robot in the team must be able to effectively perform the individual skills selected by the team. In this paper, we assume that a robot ρ at location $\boldsymbol{p}_\rho$, currently in possession of the ball, must move the ball from its current location $\boldsymbol{p}_b$ to a target location $\boldsymbol{p}_t$, chosen by a separate team planner [7]. Thus, the robot needs to decide how to best move the ball to $\boldsymbol{p}_t$.

Our robots can manipulate the ball via two mechanisms: (i) a *kicker* enables the robots to impart momentum on the ball and thus perform shots or passes, and (ii) a *dribbler bar* enables the robots to impart backspin on the ball, and thus drive while maintaining contact with the ball. Kicking the ball enables the robots to move the ball quickly, but without protecting it. Dribbling the ball enables them to move the ball while guarding it; however, this method of moving the ball is slower, and it sometimes fails due to the robot losing control of the ball.

Due to the hardware design of the robots, the robot must have the ball immediately in front of it to be able to dribble it or kick it –i.e., the robot must face in direction $\boldsymbol{\phi} = (\boldsymbol{p}_b - \boldsymbol{p}_\rho)$, and be at a distance of approximately $r_\rho + r_b$ from the ball, where r_ρ and r_b are the radius of the robot and ball, respectively. Furthermore, the robots are only capable of kicking in the forward direction $\boldsymbol{\phi}$. Thus, to execute a pass or a shot, the robot must be facing both the ball and the target location $\boldsymbol{p}_t$.

To intelligently decide how to move the ball to $\boldsymbol{p}_t$, the robot must know (i) how to use its dribbler and kicker effectively, and (ii) how to evaluate the probability of success of different ways of using them. The use of the kicker and evaluating how likely it is for a pass or a shot to be successful has been researched previously [2], and we use similar techniques here. The following sections focus on our approach to using the dribbling bar effectively, and how to best choose among different dribbling and kicking skills.

3 Individual Skills

This section covers the opponent-aware ball manipulation skills. First, possession and alignment are defined, which formulate the robot's requirement to remain in control of the ball and to align to the target. Next, we describe the Skill Decision Algorithm, which is an opponent-aware algorithm that implements the skills in an intelligent way. Lastly, we detail the two new dribbling skills used in the Skill Decision Algorithm.

3.1 Possession and Alignment

Robot, ρ, has two conflicting objectives when in possession of the ball: aligning with the ball towards its target (*alignment*), and maintaining possession of the ball from the opponents (*possession*). In the previous CMDragon team, the focus was purely on *alignment* which we define as

$$\mathbb{A} = \frac{\boldsymbol{\phi}}{|\boldsymbol{\phi}|} \cdot \frac{(\boldsymbol{p}_t - \boldsymbol{p}_\rho)}{|\boldsymbol{p}_t - \boldsymbol{p}_\rho|} < \epsilon_a \wedge |\boldsymbol{p}_b - \boldsymbol{p}_\rho| < \epsilon_d \tag{1}$$

such that it is aligned with the target angle by less than $\cos^{-1}(\epsilon_a)$ and close enough to the ball within some ϵ_d [1].

However, opponents introduce a threat that removes any guarantee on *possession*, and, by only considering *alignment*, the ball is often stolen. This can be contributed to two factors: the arc travel time of ρ and the opponent's proximity to the ball. Figure 2 demonstrates where the arc distance to *alignment* can take longer than the opponent's distance. In our simplified example, the opponent is very close in proximity to ρ and has an easy opportunity to gain *possession* of the ball by heading directly to it.

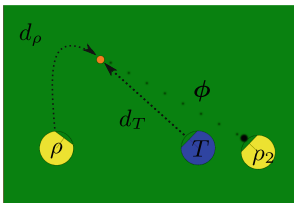

Fig. 2. ρ drives to a position near the ball that aligns to pass to ρ_2 while T drives directly to take the ball.

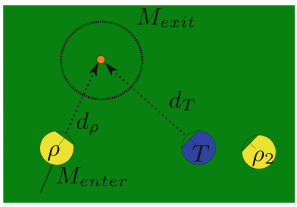

Fig. 3. Variables used to determine if ρ should drive directly to the ball since T is threatening to take possession.

We define the objective of *possession* by describing d_ρ and d_T as our robot's and the closest opponent robot's distance to the ball respectively, M_{enter} as a proportional gain added to d_ρ, and M_{exit} as a constant distance from the ball. Shown in Fig. 3, a *possession threat* ($\mathbb{P}$) is then defined to be true if

$$\mathbb{P} = d_\rho + (d_\rho * M_{enter}) > d_T \vee d_T < M_{exit} \tag{2}$$

Our approach always maintains *possession* before considering *alignment*. If there is a *possession threat* then ρ drives directly to the ball and dribbles the ball. ρ is free to just align itself if there is no threat.

3.2 Skill Decision Algorithm for Individual Skills

Based on the robot's manipulation mechanisms, we have created four skills that can be used to move the ball to its target:

Kick (K): Kicks the ball to $\boldsymbol{p}_t$. The quickest method of moving the ball to the target.

Align-non-dribbling ($A_{\neg D}$): aligns behind the ball by moving to the location $\boldsymbol{p}_b + \frac{(\boldsymbol{p}_b - \boldsymbol{p}_t) r_\rho}{|\boldsymbol{p}_b - \boldsymbol{p}_t|}$, where r_ρ is the radius of the robot. Shown in Fig. 4.

Dribbling-rotate (D_R): Dribbles the ball by approaching the closest location $\boldsymbol{p}_b + \frac{(\boldsymbol{p}_\rho - \boldsymbol{p}_b) r_\rho}{|\boldsymbol{p}_\rho - \boldsymbol{p}_b|}$ while facing the ball. It then quickly rotates to align to $\boldsymbol{p}_t$. Shown in Fig. 5.

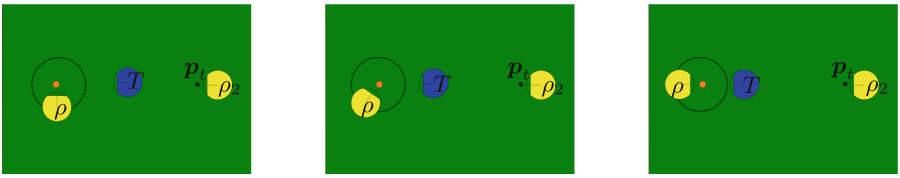

Fig. 4. Align-non-dribbling: drives around the ball to align to pass to target, $\boldsymbol{p}_t$, defined by the circle.

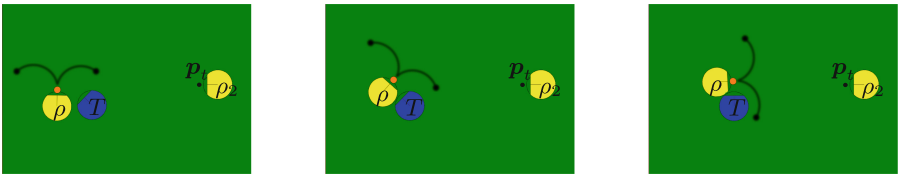

Fig. 5. Dribbling-rotate: dribbles the ball and then turns to align to pass target, $\boldsymbol{p}_t$

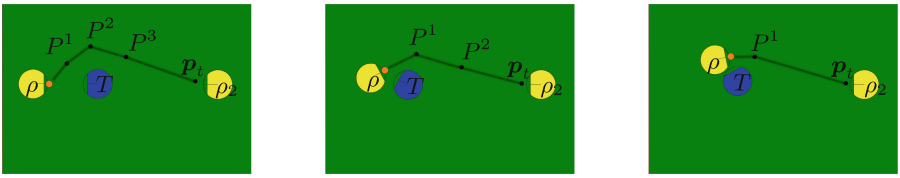

Fig. 6. Dribbling-move: dribbles the ball and pushes it along the path to $\boldsymbol{p}_t$ while avoiding obstacles.

Dribbling-move (D_M): Dribbles the ball by approaching it directly, and then moves the ball by pushing it toward $\boldsymbol{p}_t$, while avoiding obstacles. Shown in Fig. 6.

The Skill Decision Algorithm (SDA), used by ρ at each time step, is shown in Algorithm 1. The team-goal's evaluation, $E(\boldsymbol{p}_t)$, is determined by multiple factors including: open angle, opponent interception time, and pass/shoot distance, which combines to create a probability of succeeding [2]. If $E(\boldsymbol{p}_t)$ is less than or equal to a threshold δ then the skill D_M is chosen to improve the current state, i.e., to improve the $E(\boldsymbol{p}_t)$. If greater than δ then SDA checks alignment, $\mathbb{A}$, with $\boldsymbol{p}_t$. If $\mathbb{A}$ then SDA kicks the ball to $\boldsymbol{p}_t$. Otherwise, SDA checks if there is a *possession threat*, $\mathbb{P}$, since aligning might lose the ball. If $\mathbb{P}$ then SDA uses the skill D_R to grab the ball, protecting it, while still quickly aligning itself to kick to $\boldsymbol{p}_t$. Otherwise, SDA uses the more robust $A_{\neg D}$ skill to align itself around the ball.

Algorithm 1. Skill Decision Algorithm. **Input:** Team goal success probability, aligned with ball and target, and possession threat. **Output:** Skill.

```
function SDA(E(p_t), 𝔸, ℙ)
    if E(p_t) <= δ then                          ▷ Low success probability
        s = D_M                                  ▷ Move the ball to target location
    else                                         ▷ Higher success probability
        if 𝔸 then                                ▷ Checking alignment
            s = K                                ▷ Kick if aligned
        else
            if ℙ then                            ▷ Check if an opponent is near
                s = D_R                          ▷ Dribble the ball before an opponent steals it
            else
                s = A_¬D                         ▷ We have time to align nicely
            end if
        end if
    end if
return s                                         ▷ Skill to execute
end function
```

3.3 Dribbling-Rotate

D_R's priority is to align to $\boldsymbol{p}_t$ as quickly as possible. D_R dribbles the ball while maintaining an inward force to compensate for the centrifugal forces of the ball in order to maintain control as shown in Fig. 7. It faces in the direction ϕ that provides the necessary centripetal force to maintain the ball on the dribbler of the robot: facing slightly inwards while turning provides a component of the normal force from the robot that always points towards the center of the circumference. Given the robot drives forward with speed s while gradually changing its orientation with speed ω, forming a circle of radius R, the constraint $s = \omega R$ holds in this case. The necessary angle offset ϕ can be obtained analytically by noticing that all the forces in Fig. 7 need to cancel out in the rotating reference frame. Therefore, we obtain the pair of equations:

$$\begin{aligned} |\boldsymbol{f}_N| \sin\phi &= |\boldsymbol{f}_C| \\ |\boldsymbol{f}_N| \cos\phi &= |\boldsymbol{f}_F| \end{aligned} \tag{3}$$

Then, given the acceleration of gravity g, the coefficient of friction of the carpet μ, and the mass of the ball m (which cancels out in the end), we obtain:

$$\begin{aligned} |\boldsymbol{f}_N| \cos\phi &= m\omega^2 R \\ |\boldsymbol{f}_N| \sin\phi &= \mu m g \end{aligned} \tag{4}$$

Solving these equations for ϕ gives the result for the desired heading:

$$\phi = \tan^{-1}\left(\frac{\omega^2 R}{\mu g}\right) \tag{5}$$

We estimated μ by starting from measurements of when the robot kicks the ball. Then we locally optimized to the value that gives the best dribbling performance.

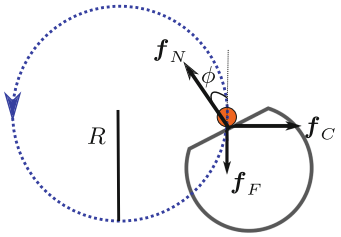

Fig. 7. Robot dribbling ball while facing slightly inwards. There exists an angle ϕ for which the forces are balanced.

3.4 Dribbling-Move

D_M's priority is to keep *possession* while driving towards $\boldsymbol{p}_t$ and avoiding all opponents and teammates. The priority of *alignment* naturally occurs as ρ drives toward $\boldsymbol{p}_t$ as shown in Fig. 6. In Algorithm 2, D_M determines if ρ has the ball by checking: (i) if it is in front of ρ, B_{front}, (ii) close to ρ, B_{close}, and (iii) located somewhere on ρ's dribbler, $B_{ondribbler}$. If ρ loses the ball then D_M drives directly to the ball to regain possession. The path used to drive to $\boldsymbol{p}_t$ is generated by a Rapidly-exploring Random Tree (RRT) where the opponents and teammates are obstacles, as defined in [5]. The path is made of multiple intermediate locations, $(P^1, P^2, \ldots, P^{n-1}, \boldsymbol{p}_t)$. After any point P^n, D_M is always slightly turning ρ's forward direction towards the next point P^{n+1} by some empirically tuned γ.

Algorithm 2. Dribbling-Move. **Input:** State of the world, robot, ball, and target. **Output:** Location and angle.

function $Dribbling - Move(W,\ R\ B,\ \boldsymbol{p}_t)$
$R_b = B_{loc} - R_{loc}$
$B_{front} = R_b.x > 0$
$B_{close} = R_b.x < MaxRobotRadius + (2 * BallRadius)$
$B_{ondribbler} = |R_b.y| \leq DribberWidth$
$B_{lost} = \neg(B_{front} \vee B_{close} \vee B_{ondribbler})$
$\{P^1, P^2, ..., \boldsymbol{p}_t\} = RRT(\boldsymbol{p}_t, W)$
$T = TurningThreat(W)$
  if B_{lost} ***then***
    $\{P^1, P^2, \ldots, \boldsymbol{p}_t\} = RRT(B, W)$
  else if $|R_{angle} - P^1_{angle}| > \alpha \vee |R_{loc} - P^1| < D_{min}$ ***then***
    $\theta = P^1_{angle}$
    if T ***then***
      $\theta = P^1_{angle} + 180°$
    end if
    return $\{R_{loc}, \theta\}$
  end if
$\theta = R_{angle} + (R_{angle} - P^1_{angle}) * \gamma$
return $\{P^1, \theta\}$
end function

This maintains control of the ball while dribbling and moving. If the turning angle goes beyond a threshold, α, then D_M stops and rotates in place with the ball. α was empirically tuned by testing the limits of turning before the dribbler lost the ball ($\alpha = 40°$ for our experiments). If there is a *turning threat* such that an opponent, in close proximity, is in the direction ρ is turning, then it turns in the opposite direction to protect the ball from being stolen [7]. D_M is complete once it arrives at $\boldsymbol{p}_t$.

4 Results

4.1 RoboCup

In this section, we analyze the semi-final and the final game of the 2015 RoboCup Small-Size league, shown in the Table 1. We used the new skills (D_R, D_M) and SDA during these games in the tournament. The data was collected by analyzing the log files of the games. For our purposes, we defined a successful pass if it got to its intended target, and a $\mathbb{P}$ as described earlier in this paper. In the tables, the second column is the number of times a skill succeeded in passing the ball while the third column is the success of those passes actually reaching the teammate. This distinction is important since the skill might pass around the $\mathbb{P}$ but the teammate fails or the ball is intercepted.

The semi-final game is divided into two parts since for roughly half the game only $A_{\neg D}$ was used. In the first part, there were 14 passes with $\mathbb{P}$ and only 3 were successful using only $A_{\neg D}$. In the second part, we used SDA with the rule that dribbling was only allowed on the offensive side of the field. Therefore, on the defensive side, $A_{\neg D} + \mathbb{P}$ was used 6 times, succeeding 1 time. On the offensive side, D_R passed 11 times and succeeded 4 times with no clear improvement. D_M did improve with 7 successful passes out of 10 times. Interestingly, D_M was never used when not under pressure by an opponent, which was the major cause of the low value of $E(\boldsymbol{p}_t)$ (< 0.1). Therefore, D_M started in a situation with a vastly low probability of success and under $\mathbb{P}$, but still it succeeded 7 times in getting away from the opponents and finding a better pass.

Table 1. Semi-final and final game in 2015 RoboCup small-size league. Statistics for the three maneuvering skills.

Semi-Final game against STOx's		
Skill	**(Success/Total)**	**(Success/Total)**
First Part	**Total # of Uses**	**$\mathbb{P}$ + Pass**
$A_{\neg D}$	17/32	3/14
Second Part		
$A_{\neg D}$	9/28	1/6
D_R	11/15 ($\forall\mathbb{P}$)	4/11
D_M	10/17 ($\forall\mathbb{P}$)	7/10

Final game against MRL		
Skill	**(Success/Total)**	**(Success/Total)**
	Total # of Uses	**$\mathbb{P}$ + Pass**
$A_{\neg D}$	10/29	3/13
D_R	23/36($\forall\mathbb{P}$)	11/18
D_M	10/23 ($\forall\mathbb{P}$)	6/10

In the final game, we used SDA for the entire game with the same offensive restriction to dribbling. Again, we see poor performance for $A_{\neg D} + \mathbb{P}$ with 3

successful passes out of 13. D_R performed much better in this game with 11 successful passes out of 18, and D_M performed well again with 6 successful passes out of 10.

Real games only provide sparse amounts of information on the benefit of the added skills because they are short and unreproducible. Still, they provide evidence on the algorithm's performance in real-world conditions against unknown opponents for which they were designed to handle. The results show that $A_{\neg D}$ is very unsuccessful when there is a *possession threat*, and by implementing more intelligent ball-manipulation skills we could improve the success rate against unknown opponents. Based on our review of the competition games there were clear times when D_R was better than D_M and vice versa. To better understand our analysis of the game, we introduce *passing with marking* to challenge our robot with situations often found in soccer, specifically those with *possession threats*.

4.2 Passing with Marking

Passing with marking is a sub-domain of soccer that uses marking to induce a state where the probability of successfully passing is lowered due to the proximity of the opponent(s), i.e., a *possession threat*. The domain starts with one robot ρ being marked by a close opponent Taker, T, at some distance d_T. ρ is placed closest to the ball while T blocks the initial pass. As T's distance to the ball, d_T, decreases, it is more likely to gain possession or block the pass. The objective is for ρ to pass to ρ_2 before T steals the ball or kicks it out of bounds. We define stealing the ball as when T has the ball within a robot radius plus a ball radius for at least 1 second. This constraint ends stalemates where both robots are driving into the ball and not moving. The domain is defined by a bounded area, and the teammate ρ_2 moves within this area to get open for the pass defined by its own team objective [7].

We devised two scenarios of the *passing with marking*: EXP1 where ρ is facing the ball and T, shown in Fig. 8, and EXP2 where ρ faces away from T with the ball near its dribbler, shown in Fig. 9. We ran both EXP1 and EXP2 in

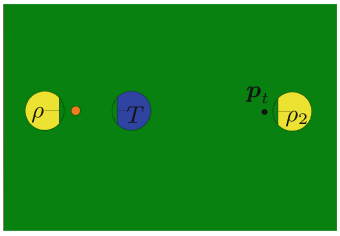

Fig. 8. (EXP1) Used in the simulation evaluation, it is a passing with marking domain where T is facing off against ρ, who must kick to ρ_2.

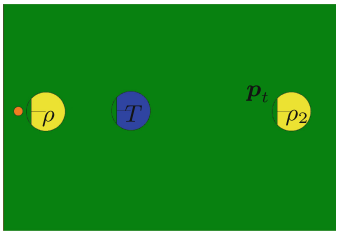

Fig. 9. (EXP2) Used in the simulation evaluation, it is a passing with marking domain where ρ must get around the ball to align itself to kick to ρ_2.

a physics-based simulation. For each test, we used only one of the approaches to see if the skill could pass the ball to ρ_2 using only that approach. We also devised two opponents that change the performance of the approaches. The *Drive to ball* opponent heads for the ball and tries to grab a hold of it. The *Clear ball* opponent attempts to kick the ball out of bounds, which usually involves it heading towards either the right or left side of ρ to kick it away.

EXP1 induces a state where T is blocking the initial pass and as d_T decreases it has a higher chance of stealing the ball away from ρ. This is clearly demonstrated in Table 2 where for both opponents the non-dribble approach $A_{\neg D}$ often fails to pass to its teammate. However, $A_{\neg D}$ does surprisingly better than D_R against the *Drive to ball* for two reasons. First, D_R fails at this task because as ρ approaches the ball so does T, and they often get stuck in a stalemate as D_R's forward velocity pushes against T. Second, $A_{\neg D}$'s success is due to luck as it kicks the ball immediately off of T and on occasion can get the rebound and pass to ρ_2. As d_T increases, the ball bounces less and $A_{\neg D}$ does not get as lucky as shown in Table 2. D_M performs the best against *Drive to ball* since when it gets into the stalemate position it can sometimes rotate in place with the ball and move to a better passing position. The rotation in place allows D_M to succeed where D_R failed. Both D_R and D_M did very well against clear ball because the same stalemate position did not arise as often since T is trying to get to the side of the ball in order to kick it out of bounds. This gave our approaches the opportunity to dribble without getting stuck.

Table 2. Passing with marking for 100 episodes on each approach where the experiment EXP1 is shown in Fig. 8.

Physics-based simulation EXP1 (success/100)					
	Opponents				
Approach	Drive to ball				Clear ball
	$d_T = \{260, 360, 460, 860\}$ mm				$d_T = 260$ mm
$A_{\neg D}$	15	11	2	1	3
D_R	4	2	0	14	89
D_M	35	48	53	97	84

EXP2 induces a state where ρ has to get around the ball in order to align itself to pass to ρ_2, while T puts pressure from the back as it tries to steal the ball. In Table 3, $A_{\neg D}$ did the worst out of the three approaches, and its small success against *Drive to ball* is because T would sometimes get stuck behind ρ and remain behind it. D_R and D_M both did well against *Drive to ball*. D_M was the best because it simply rotated in place first to align itself and it did so in the opposite direction of T. This meant that T was often circling around on the backside of ρ, giving it a clear pass. However, the rotating in place was D_M's downfall against *Clear ball* since T would stay on facing direction until ρ rotated to a side and then T would kick it away. The reason D_R did the best against

Table 3. Passing with marking for 100 episodes on each approach where the experiment EXP2 is shown in Fig. 9. Total is success plus failed.

Physics-based simulation EXP2 (success/100)			
	Opponents		
Approach	Drive to ball	Clear ball	
	$d_T = 590\,\text{mm}$	$d_T = \{590, 690\}\,\text{mm}$	
$A_{\neg D}$	21	4	8
D_R	84	48	85
D_M	96	22	18

Table 4. Passing with marking using real-robots with 10 episodes each.

Real-robot experiment (success/10))	
	Opponents
Approach	Clear ball
	$d_T = 355\,\text{mm}$
$A_{\neg D}$	0
D_R	2
D_M	5

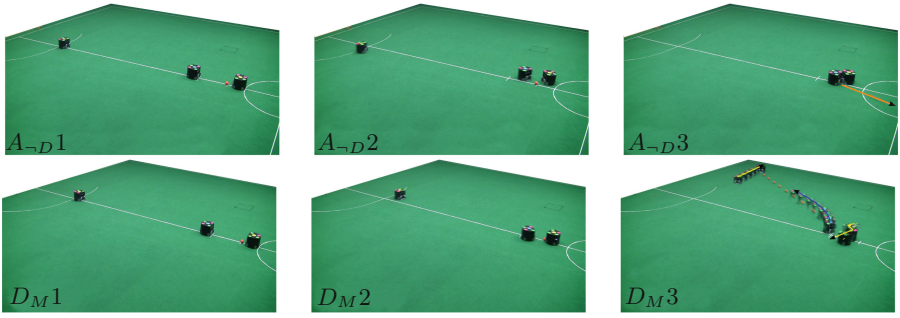

Fig. 10. ($A_{\neg D}$): ρ rotates around the ball to pass but loses it as the opponent kicks it away. (D_M): ρ, using Algorithm 2, goes directly to the ball, slides past the opponent to the right, and passes to the teammate. ρ was 75 mm and T was 355 mm away from the ball in their initial positions. T was running *Clear ball*.

Clear ball was because, as it circled around, it created more distance between itself and T. For the same reason, it performed better as d_T increased where D_M remained relatively the same.

We do not have a way of automating experiments on real-robots, and robots wear with use so it is not cost effective to run hundreds of experiments on the real-robots. We did run the EXP1 with the *Clear ball* opponent on our real-robots. Each approach was tested 10 times, and the results in Table 4 are slightly

different than our simulated experiment. $A_{\neg D}$ did very poorly as predicted by simulation but the performances of D_R and D_M were not as expected. This difference can be attributed to factors on the complexity of executing skills in a stochastic environment with noisy actuators and perception. D_M still performs well on passing to the teammate. D_R would often continuously circle while T blocked the pass and the ball was eventually stolen. An example run of $A_{\neg D}$ and D_M is shown in Fig. 10.

5 Conclusion

We set our goal of improving the CMDragon team by better understanding the failing points of the previous team's approach to ball-manipulation. *Possession threats*, or lack of opponent awareness, stood out as a major downfall as purely aligning to the target led to opportunities for opponents to steal. The second major issue was passing even when the probability of success given to the robot, from the planner, was low. Since the previous team only had one option, the best they could do was to kick the ball and hope for luck.

Our solution to *possession threats* was to drive directly to the ball and dribble it. If the probability of success was high enough then we used D_R to quickly turn to the $\boldsymbol{p}_t$ and pass. For the low probability, our solution was to use D_M to move the ball to the target location to get a better evaluation for a pass. We then combined these skills into the Skill Decision Algorithm that defined when each skill should be chosen in order to best serve the individual and the team.

The results show an improvement in pass success using our method in both the real games and simulation experiments. Future work includes learning the possession threat parameters for individual teams during the game. There might also be more sophisticated parameters for choosing which skill to execute in the Skill Decision Algorithm. This can be seen in our simulated experiments, which showed that there exists certain scenarios where one approach outperforms the other. Further, our results indicate that one skill is not enough to solve the complexity of outmaneuvering an opponent, and that a combination is best suited to solve the problem.

The problem with having the planner plan every skill for each robot is not feasible for many real-time domains. The planner gives its best goal with what it can feasibly plan, but the robot itself must execute the skill to accomplish the goal. We demonstrate in this paper that increasing the sophistication of the individual robot skills improves the performance of the team. The robot can choose to execute the team goal differently based on the current state of the world. And by changing its skill, the robot can improve its probability of success. For multi-robot teams in adversarial environments, the individual robot must have sophisticated skills that can handle different scenarios with complex opponents. A team is therefore successful when the individual robot can choose skills that improve the team goal and/or improve itself in the environment.

References

1. Biswas, J., Cooksey, P., Klee, S., Mendoza, J.P., Wang, R., Zhu, D., Veloso, M.: CMDragons 2015 extended team description paper. In: Robocup 2015 (2015)
2. Biswas, J., Mendoza, J.P., Zhu, D., Choi, B., Klee, S., Veloso, M.: Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team. In: Proceedings of AAMAS 2014, the Thirteenth International Joint Conference on Autonomous Agents and Multi-agent Systems, Paris, France, May 2014
3. Browning, B., Bruce, J., Bowling, M., Veloso, M.: STP: skills, tactics and plays for multi-robot control in adversarial environments. J. Syst. Control Eng. **219**, 33–52 (2005). The 2005 Professional Engineering Publishing Award
4. Damas, B.D., Lima, P.U., Custódio, L.M.: A modified potential fields method for robot navigation applied to dribbling in robotic soccer. In: Kaminka, G.A., Lima, P.U., Rojas, R. (eds.) RoboCup 2002. LNCS, vol. 2752, pp. 65–77. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45135-8_6
5. Lavalle, S.M., Kuffner Jr., J.J.: Rapidly-exploring random trees: progress and prospects. In: Algorithmic and Computational Robotics: New Directions, pp. 293–308 (2000)
6. Li, X., Wang, M., Zell, A.: Dribbling control of omnidirectional soccer robots. In: 2007 IEEE International Conference on Robotics and Automation, pp. 2623–2628. IEEE (2007)
7. Mendoza, J.P., Biswas, J., Zhu, D., Cooksey, P., Wang, R., Klee, S., Veloso, M.: Selectively reactive coordination for a team of robot soccer champions. In: Proceedings of AAAI 2016, the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, February 2016
8. Zickler, S., Laue, T., Birbach, O., Wongphati, M., Veloso, M.: SSL-vision: the shared vision system for the RoboCup small size league. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS (LNAI), vol. 5949, pp. 425–436. Springer, Heidelberg (2010). doi:10.1007/978-3-642-11876-0_37
9. Zickler, S., Veloso, M.: Efficient physics-based planning: sampling search via non-deterministic tactics and skills. In: Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Budapest, Hungary, pp. 27–33, May 2009