Journal of Intelligent & Robotic Systems (2025) 112:2
https://doi.org/10.1007/s10846-025-02334-0

REGULAR PAPER

Designing Offensive Robot Soccer Strategies from Principles in Invasion Team Sports

Hao Liang Chen[1] · Ruben Beumer[1] · Elena Torta[1] · Herman Bruyninckx[1,2,3] · René van de Molengraft[1]

Received: 4 March 2025 / Accepted: 30 October 2025 / Published online: 26 November 2025
© The Author(s) 2025

Abstract

RoboCup soccer robots seldom use offline defined strategies to proactively coordinate their passing and movement actions in a soccer game. Human team sports do that abundantly, especially in Invasion Team Sports (ITS). This manuscript uses literature on ITS strategy, with a focus on futsal because of its similarity to RoboCup soccer, to analyze the tactical principles – imagined as abstract guidelines, e.g., "to threaten defensive lines" – behind their coordination patterns. Our main contribution is in the structured design of robot soccer strategies via an explicit composition of human-inspired tactical principles. Our approach provides robot soccer strategists with clearer insight in applying best practices of human strategy design, serving as a practical starting point for designing and adjusting robot soccer strategies; this is currently missing in robot soccer literature, as it does not delve deeper into *why* the coordination patterns exist. Our contribution is realized by performing the following steps: 1) the proposition of an initial guideline to apply a composition of tactical principles in the design of robot soccer strategies, 2) the formalization of tactical principles into robot action and perception constraints, and 3) the design of a 5-on-5 robot soccer strategy with the insights from the previous two steps. The strategy is implemented in the Skill, Tactics, and Plays (STP) software as used by the Tech United Eindhoven soccer robots. The experimental results validate the applicability of our designed strategy. The first validation, in the form of a sensitivity analysis, is performed on the Tech United Eindhoven simulator, and the second is realized in a real game against Robotsports.

Keywords Soccer robots · Strategy · Futsal · Invasion team sports · Adversarial environment · Knowledge engineering

1 Introduction

The ultimate goal of RoboCup soccer is to beat the human world champions by 2050 [1]. At this moment, humans surpass robots not only in ball handling skills, but also in coordinated movements that arise from their strategies. This manuscript explores concepts of strategy design within human Invasion Team Sports (ITS) [2], applied to robot soccer. Special attention is paid to small-sided court games like futsal, in which the 5-on-5 format matches best with current settings in RoboCup leagues such as the Humanoid League[1] and the Middle Size League (MSL)[2].

Ruben Beumer, Elena Torta, Herman Bruyninckx, and René van de Molengraft contributed equally to this work.

✉ Hao Liang Chen
h.l.chen@tue.nl

Ruben Beumer
r.m.beumer@tue.nl

Elena Torta
e.torta@tue.nl

Herman Bruyninckx
herman.bruyninckx@kuleuven.be

René van de Molengraft
m.j.g.v.d.Molengraft@tue.nl

[1] Mechanical Engineering, Robotics, Eindhoven University of Technology, De Zaale, Eindhoven 5612 AZ, Noord Brabant, Netherlands

[2] Mechanical Engineering, KU Leuven, Celestijnenlaan 300, 3001 Leuven, Belgium

[3] Flanders Make, Gaston Geenslaan 8, 3001 Leuven, Belgium

[1] https://humanoid.robocup.org/wp-content/uploads/RC-HL-2025-Rules.pdf

[2] https://msl.robocup.org/wp-content/uploads/2024/12/Rulebook_MSL2025_v26.0.pdf

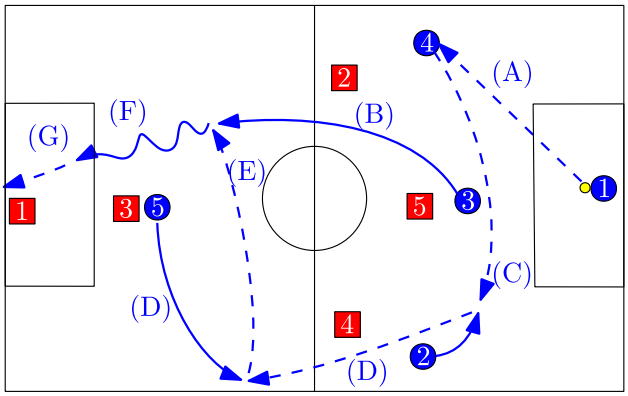


Fig. 1 An offensive strategy provided by the FC Eindhoven futsal team. The blue circles represent the attacking team, the red squares represent the defending team, and the yellow circle represents the ball. The dashed arrows represent a shooting/passing action, the solid arrows represent a movement action, and the squiggly solid arrow represents a dribble action. All figures in this manuscript maintain the same visualization style, unless explicitly stated otherwise, with the blue offensive team starting from the right, and the red defensive team from the left. The letters indicate action sequences, for example, (A) is a pass from blue 1 to blue 4, followed by (B) which is a movement by blue 3

1.1 Implicit Intentions in Human Strategies

To our knowledge, no guideline currently exists in robot soccer literature on best practices for strategy design. As a result, the roboticist does not know to what extent a strategy needs to be designed. ITS literature design strategies according to *Tactical Principles* (TPs), which can be understood as abstract guidelines that define proper team behavior under specific circumstances. For example, "a team should change the point of attack when the defense hinders its current progression". The *strategy* relates specific TPs to the spatio-temporal properties of the soccer game to obtain some long-term vision [3]. The result is a blueprint of *sketched actions* that the team can follow. Figure 1 reports an offensive strategy of the FC Eindhoven futsal team[3], in which sketched actions are visualized as passes (blue dashed arrow) and movements (blue solid arrow).

We want to emphasize that the TPs that govern these sketched actions are not immediately apparent, thus making it unclear *under which circumstances* a sketched action should be initiated, is executed sufficiently, or needs to be aborted. One can only speculate that, e.g., passes (C) and (E) are executed to switch the point of attack. Without an understanding of the underlying TP(s), a roboticist can only mimic the geometric properties of such a sketch; this is unreliable given the abstract nature of sketches [4]. A similar sentiment is shared in the human sports literature [5], where the authors emphasize the abundance of performance analysis literature that *"are all outcome-focused, reporting what happened in a game, but do not reveal how or why some action or event occurred"*. Examples of analyzed variables are, e.g., goal shots, but these do not inform a coach how to *create* these goal shots.

[3] https://fc-eindhoven.nl/fce-international/

1.2 Robot Soccer Coordination Gap

We observe two main categories of coordination approaches in robot soccer literature, where more details can be found in Sec. 2. The *reactive approach* (Fig. 2 left side) focuses on formulating static Cost Optimization Problems (COP). Each robot is assigned some role, for example, "attacker" and "defender", and static TPs are defined according to that role, e.g., "the attacker needs to pressure the penalty area". The TPs are translated into mathematical constraints/cost functions to formulate a COP, and a solver provides iteratively an optimal output [6]. That output can be executed by the robot, e.g., to go to a specific coordinate (x,y). The *proactive approach* (Fig. 2 right side) assigns a sequence of (sketched) actions to each robot role, e.g., the attacker needs to go to area "X" and then to area "Y" [7]. Essentially, each robot role has its own dynamic COP, that changes as the strategy progresses. However, in the robot soccer literature, strategies are still designed mainly using spatial and temporal data [8].

Current soccer robots cannot properly realize human strategies as in Fig. 1. Reactive approaches struggle to enforce specific coordination patterns. Proactive approaches based on spatial and/or temporal data have difficulty in properly capturing the intentions behind the coordination patterns,

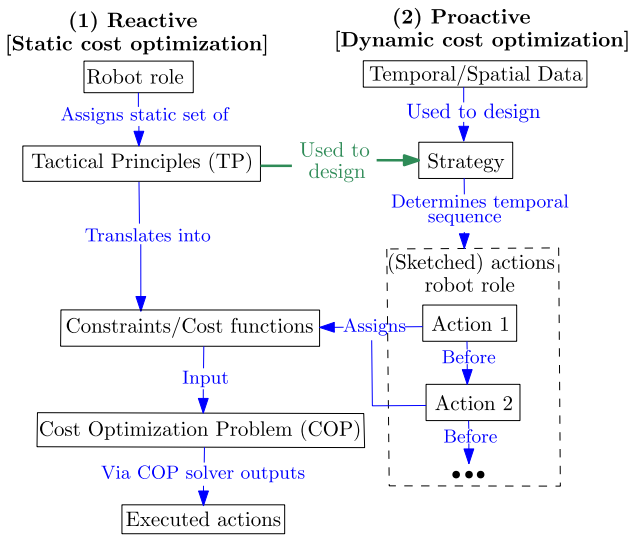


Fig. 2 Illustration of the procedures of the two main coordination approaches for robot soccer. On the left, the reactive approach generates a static COP for each robot role to solve. On the right, the proactive approach allows each robot role's COP to dynamically change. The blue arrows represent the relation between software components. The green arrow represents the additional relation that is introduced in this manuscript

i.e., for what reason actions may have been executed. We argue that roboticists need to understand the TPs underlying strategy design to properly design, execute, and adapt human-like strategies. This is represented as the green arrow in Fig. 2 and is the focus of this manuscript.

1.3 Contribution

Our main contribution is to design robot soccer strategies in a structured manner via an explicit composition of human inspired TPs. That is done by formalizing (decoupled) abstract principles such as "to threaten defensive lines" and "to be able to reset the attack" into robot interpretable constraints. The effects of that formalization are not necessarily observable in the strategy performance, but mainly in the strategy design procedure by facilitating the incorporation of non-robotic expert knowledge; experts of who we cannot expect to be able to formalize their knowledge in robot software [9]. Making such semantic drivers (in the form of TPs) explicit and relating them to robot constraints is not only beneficial in the context of this manuscript. The authors of [10] show that structural behavior design can be realized by linking semantic areas to motion control constraints in the context of single-robot navigation. Humans themselves also rely on semantics for coordination, e.g., via the usage of traffic rules and signs that enable decentralized coordination in the traffic system [11]. We argue that robot behavior design can always benefit from understanding the semantics in (human-made) environments.

Our contribution is realized by performing the following steps:

1. The proposition of an initial guideline in applying a composition of TPs – based on insights from ITS – in the design of robot soccer strategy (Sec. 3).
2. The formalization of the TPs into robot action and perception constraints (Sec. 4 and Appendices A and B).
3. The design of a 5-on-5 robot soccer strategy with insights from the above steps (Sec. 4).

We show how TPs, even when defined in a simple and abstract way, can be translated into formal models (e.g., state machines and motion control constraints). Our approach provides robot soccer strategists with clearer insight in applying best practices of human strategy design, serving as a practical starting point for designing and adjusting robot soccer strategies.

The designed strategy is implemented into the Skills, Tactics, and Plays (STP) framework as used by the Tech United Eindhoven soccer robots. The strategy is validated on simulated and real-world experiments (Sec. 5). The first validation, in the form of a sensitivity analysis, is performed on the Tech United Eindhoven simulator, and the second is realized in a real game against Robotsports[4].

2 Related Work

2.1 Reactive Soccer Robots

Reactive approaches emphasize adaptability and an immediate response to the current opposition behavior. If an optimal set of TPs can be defined that properly reflects the principles behind human strategy design, then robots may behave similarly to humans. A strategist does not have to manually program coordination patterns for every possible configuration of the opposition, as optimal behavior emerges at every time instance. This emergent behavior design is its greatest strength, and resources can be allocated to improve other aspects of robot soccer such as individual skills [12].

The survey paper [12] distinguishes three software components to enable these reactive behaviors. The first component is an optimization of the (online) decision-making of the ball holder in relation to its own teammates. The output of the optimizer is: a decision to pass to a teammate [13], a sequence of passing actions resulting in a final goal scoring attempt [14], or a dribble, passing and shooting sequence [15].

The second component focuses on positioning to facilitate passing and shooting actions. For example, the 2022 RoboCup Small Size League (SSL) champions (TIGERs Mannheim) used a force field to position toward the opponent's side while staying away from the opponent players [16]. Utility maps [6] optimize the robot position on a gridded field representation, with cost functions, or TPs, such as "do not enter the goal areas" and "receive the ball at least 2 meters away from the opponent". Additional constraints restrict optimization to specific zones in the field [7]. Those zones can be related to formation knowledge, such as the diamond formation in futsal [14].

The third component focuses on player-role assignments *to guarantee* that the appropriate robot occupies a certain zone in the field. Typically, this assignment is based on spatial or temporal criteria, such as: the distance between a player and a zone [14], the time to occupy a certain zone [7], or distances to a target and to a possible collision [17]. The combination of these three components leads to emergent behavior.

2.1.1 Limitations Reactive Approach

The main limitation of reactive approaches is to consistently (re)produce behavior aligned with human strategies.

[4] https://www.robotsports.nl/index.php/nl/

First, it is difficult to find a static set of TPs that is applicable to all (steps in the) strategies. Even in ITS literature [18–20], there is no consensus on the types of TPs that should exist; each author proposes its own set of TPs with varying terminology. In addition, it seems more likely that each step of the strategy focuses on its own set of TPs. For example, blue 3 in Fig. 1 may initiate movement (B) with a deceptive purpose to mislead the opposition, while at dribble (F) it moves to threaten the penalty area. This implies that the Cost Optimization Problem (COP) needs to be adjusted according to the spatial and temporal evolution of the strategy.

Second, it is difficult to force robots to behave in a specific manner in specific scenarios. No opponent is the same, and a good strategist should adjust the strategy accordingly. In reactive approaches, behavior adjustments occur by adjusting the COP before matches. For example, the MSL champion Tech United Eindhoven would add/remove cost-functions or adjust scalar weighing factors. These modifications do not provide reliable predictions on how the precise behavior will eventually emerge. This is problematic for ITS strategies, e.g., in futsal [20], as these strategies rely on coordinated movement patterns.

Third, it is difficult to capture long-term strategic visions in a (disjoint) set of cost functions. Reactive approaches tend to be short-term oriented. The authors of [21] mention the "price of anarchy" in basketball, where the optimal long-term solution for the team does not necessarily emerge from a sequence of optimal short-term decisions by the individual team members.

2.1.2 Example Non-optimal Coordination

An example of non-optimal coordination is observed in Fig. 3. In the top image, the attacking team is observed with the blue bounding box 1 robot in ball possession. That robot initiates a pass to the blue bounding box 2 robot in the middle image. Only *after* the pass is initiated, does the blue bounding box 3 robot in the top image start its movement to support the new ball holder. It takes a full five seconds for blue bounding box 3 robot to arrive at the bottom image, while the defenders allow the blue bounding box 2 robot to maintain ball possession all that time. The rationale behind this behavior is that a fixed set of roles are assigned to the attacking robots, for which only one non-ball-holding robot can provide offensive support. Additionally, the cost functions for blue bounding box 2 robot should not have been designed to allow it to move that far away in the middle image. One could add an additional cost function, e.g., "to stay within a certain distance of team members", but that could lead to undesired situations in other circumstances.

2.2 Proactive Soccer Robots

Proactive approaches emphasize consistency and repeatability. An offline plan of movement and passing patterns are defined, and conditions are formulated to when such patterns need to be respected. The main strength is that the strategist can exert a greater influence on how the attack will develop, in contrast to reactive approaches in which the outcome emerges by solely solving a static COP. That strength is also the drawback of this approach, as it requires more effort and expertise from the strategist in designing proper coordination patterns. A badly designed strategy may in that case even perform worse than reactive approaches. Currently, no guidelines exist in the state-of-the-art on designing robot soccer strategies. Consequently, strategy design may seem to be an overwhelming procedure as one does not understand what to pay attention to. This manuscript addresses this gap.

2.2.1 Software Frameworks

In robot soccer literature, one can find many software frameworks that allow the incorporation of such proactive approaches [22–26]. The authors of [22] introduce an architecture to facilitate flexibility in team structures and inter-robot communication, which was deployed on robots in the RoboCup Small Size League (SSL) [27]. The architecture is based on so-called "Locker Room Agreements" (LRA). These LRAs contain prior knowledge such as "positioning constraints" and "role assignment conditions". Multi-step multi-agent plans can then be formulated by an external expert to be executed in certain situations, e.g., corner kicks.

Setplays [23], developed by the Simulation League, allow the formulation of multi-step plans applicable to any number of robots. Each step contains an action and involves an arbitrary number of roles that are assigned to available players. A step can be initiated or aborted when certain conditions are met. Each Setplay allows parameters and player roles to be instantiated online. To facilitate the creation of Setplays, the graphical interface SPlanner was created by [28] to draw sketches similar to typical team strategy sketches (Fig. 1); the number of sketchable actions was increased in [29], to allow the insertion of actions involving opponents.

The Skills, Tactics, and Plays (STP) framework [24] was introduced for centralized multi-robot coordination. Tech United Eindhoven adjusted this framework to also be applicable in distributed systems [30]. Each play is a team plan containing applicability ("pre") and termination ("post") conditions involving a random number of robots. Each robot executes a sequence of tactics, where each tactic is parameterized and represents a robot behavior, e.g., "make a run"

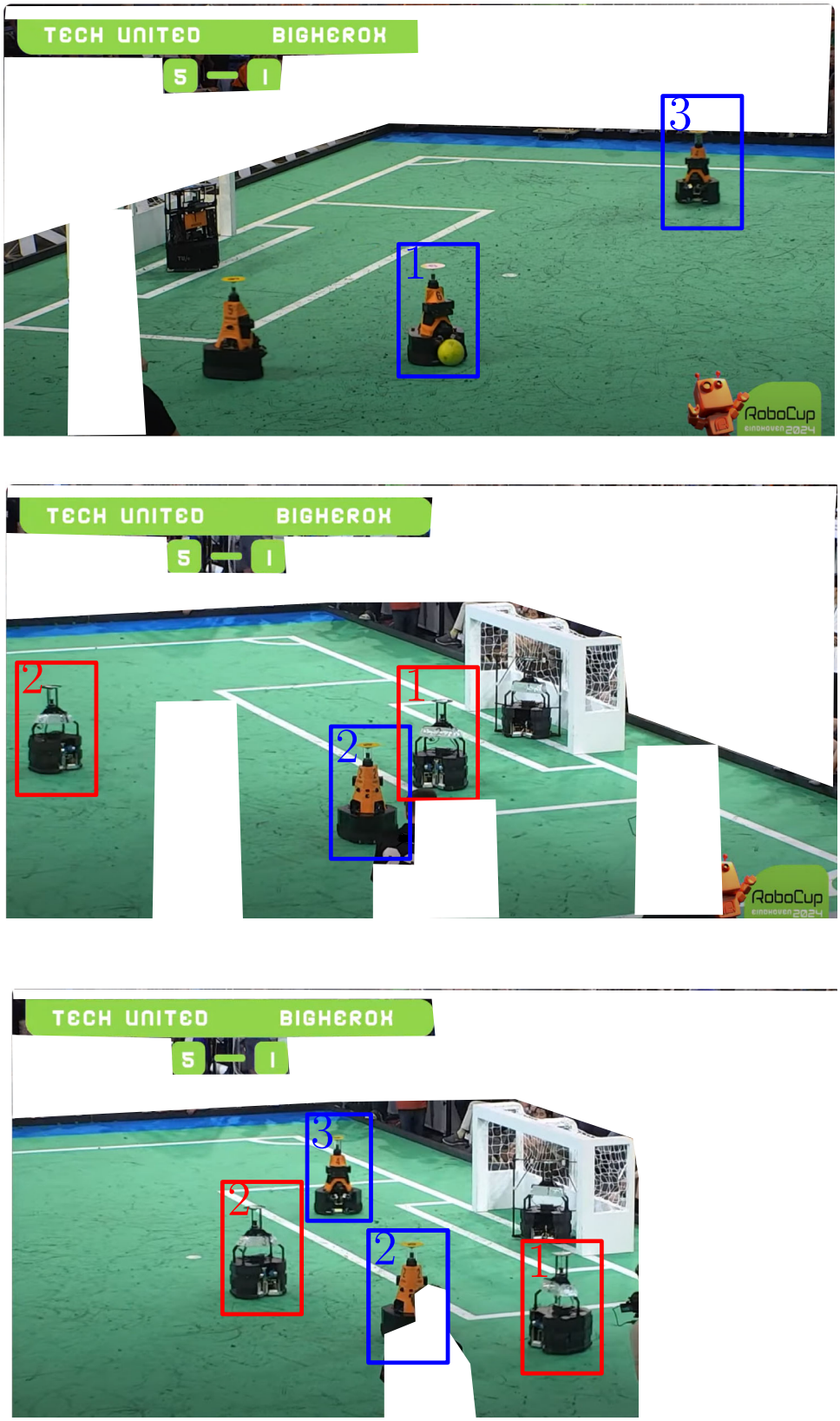


Fig. 3 Example of an attack performed by the Tech United Eindhoven robots in the MSL final of RoboCup 2024. (The pictures come from video.) The top picture is taken at 6:22:42 (hours:minutes:seconds) where the blue bounding box 1 robot has the ball. The middle picture is taken at 6:22:47 where a long pass is initiated to the blue bounding box 2 robot. The two defending robots are indicated with a red bounding box. The bottom picture is taken at 6:22:51, when a defensive gap is created when the red bunding box 1 defender moves away. The white polygons are for privacy reasons, and no robot is occluded by it

or "pass to robot role 2". Each tactic is represented as a state machine, with in each state a specific execution of low-level control/optimization algorithms (i.e., the parameterized skills, e.g., planning and following a trajectory to some waypoint in the field) and an evaluation condition to check whether the tactic is completed. For example, waypoint generation via utility maps [6] can be part of a tactic. The introduction of movement patterns can be done by restricting the zones in which this utility map is applied [7, 30].

The literature on formal soccer theory [25] is also a source of inspiration for translating qualitatively sketched strategies from an expert into formal mathematical representations that can be understood and executed by robots. The basic primitives were extracted and the behaviors were specified in Readylog [31]. Qualitative spatial relations were used in combination with a semantic map to quantify human instructions such as "left midfield". Three types of passing patterns were distilled from the literature on soccer theory to build up an offense: long passes, diagonal passes, and deep and back passes. The authors could specify strategies involving those passing patterns from human soccer into robot programmable code via Readylog.

The authors of [26] allow the expert coach to adjust the plans of the soccer robots through natural language processing. The text or voice commands are translated into Pure-Past Linear Temporal Logic (PPLTL) rules to express temporal goals by matching them to prior defined templates. Those rules can then be used to formulate a plan specified in the Planning Domain Definition Language (PDDL) [32] that can be executed by soccer robots. The authors of [33] realize coordination between quadruped robots by deploying a Petri Net Plan.

2.2.2 Lack of Expert Knowledge

The responsibility of designing proper strategies is assigned to the user of the above-mentioned software frameworks, e.g., a soccer coach having expert knowledge. There are two problems with this way of thinking. First, the TPs that a coach knows are abstract descriptions, and it is unclear how to formalize that into one of these software frameworks. For example, it is not immediately apparent how to realize the TP of "to threaten defensive lines" [20]. Second, experts, such as soccer coaches, cannot always be available or willing to provide good strategies for the software frameworks mentioned above. That observation was already made with the traditional expert systems [9]. One of the reasons could be that there is little incentive or time available for a coach to disclose their knowledge to a robot soccer team. Even the eight-time MSL champion Tech United Eindhoven has no active soccer coach involved in strategy design. Therefore, strategies are often designed empirically. For example, the authors of [7] would design a sequence of zones based on their effectiveness in simulation. Data-driven approaches, such as learning-for-demonstration in [8, 34], rely on an expert to provide strategy input to make them more generically applicable. However, as the tactical principles behind that strategy input are missing, such approaches cannot generate new strategies that differ too much from the expert-provided input.

This manuscript thus attempts to make the tactical principles behind strategy design explicit and applicable to robot programming. That allows robot soccer teams to design better strategies by themselves without needing to consult an external expert. Essentially, the approach proposed in this manuscript can be seen as a hybrid of proactive and reactive approaches, similar to [7], where strategy confines the search space of a COP to explicit zones on the soccer field. In our case, that zone selection is based on tactical principles from invasion team sports.

3 Invasion Team Sports Tactical Principles Analysis

This section provides background knowledge on the TPs from ITS literature that influence the design of our robot soccer strategy. An initial strategy guideline on the usage of these TPs is proposed at the end of this section. Section 4 discusses the formalization of these TPs into robot-interpretable commands.

This manuscript adopts four categories of TPs introduced in [18]: *mobility*, *opportunity*, *deception*, and *reserve*. This selection was made because not all categories of TPs involve strategy design, e.g., the *economy* principle which focuses on strategy selection in relation to scoreline, nor were they all needed to illustrate the applicability of TPs in robot soccer strategy design. In Sec. 3.5, we elaborate on how the four proposed categories can capture the essence of futsal strategies and propose an initial guideline on its usage. Table 1 summarizes the TPs that are considered in this manuscript. The selected TPs are inspired by common principles in ITS literature (e.g. [18–20]) and are not intended as an exhaustive or definitive taxonomy. Next, we elaborate on the TPs in Table 1. Note that in all our analyses we exclude the goalkeeper as it does not actively contribute to the attack.

3.1 Mobility

Definition 1 Principle of mobility: "positional attack, based on continuity of play, requires preparatory maneuvers before the attack can be launched. Through fast shifting and good circulation of the ball, attackers may induce, in a given spot of the attack zone, a breaking down point in the state of equilibrium between the two teams and thus facilitate a shot on goal" [18, p. 16]

Table 1 The left column represents the Tactical Principle (TP), the middle column the descriptions of the TPs, and the right column the category to which the TP belongs

Tactical Principle (TP)	Description	Category TP
($TP_{m,1}$) Pass and move	Every pass is accompanied by a (supporting) player movement, who proactively attempts to create new (future) opportunities.	Mobility
($TP_{m,2}$) Spread players	Spreading players increases the effective play and movement area.	Mobility
($TP_{o,1}$) Threaten defensive lines	A desire to position and receive the ball in the area behind the defender(s) towards their goal.	Opportunity
($TP_{o,2}$) Switch play	A continuation of the attack on the opposite side of the field.	Opportunity
($TP_{o,3}$) Pressure penalty area	The penalty area is a vulnerable area for the defending team.	Opportunity
($TP_{o,4}$) Threaten unmarked space	Supporting players move into areas that are unmarked by the defense, preferably towards the goal.	Opportunity
($TP_{d,1}$) Offensive support	At least one offensive support needs to be present at all times for the ball holder to increase ambiguity of continuity of attack.	Deception
($TP_{r,1}$) Reset attack	The team should be able to recover the (initial) attacking formation in a timely manner	Reserve
($TP_{r,2}$) Safe pass option	At all times, the ball holder should have a safe option (backwards) to pass the ball	Reserve

The first subscript of TP indicates the category (m=mobility, o=opportunity, d=deception, r=reserve)

This category of principles focuses on continuous attack progression. That aligns well with the concept of a *rotation* in futsal [20], quoted as *"a set of player movements aimed at retaining possession and moving the ball towards the opposition's goal"*. A similar sentiment is echoed in basketball [35], where the authors emphasize the phrase *"move or die"*. The core idea is that continuity in the attack leads to delay of the opponent's ability to physically reorganize its defense.

$\mathbf{TP}_{m,1}$ is inspired by analyzing futsal strategies [20]. In particular, the well-known butterfly rotation in Fig. 4 where a pass (A) is sequenced by movements (B). The second part continues in a mirrored fashion, allowing the attacking team to infinitely repeat the rotation with a "pass and move" sequence. This principle can also be observed in Fig. 1, where similarly pass (A) is followed by movement (B).

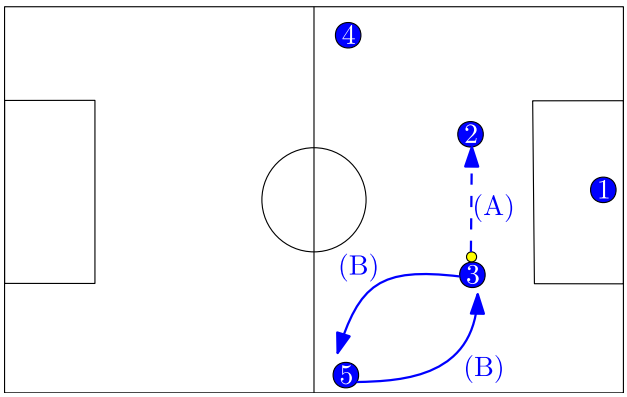


Fig. 4 Visualization of the first part of the "butterfly rotation" based on [20]. Two actions are visualized: (A) a pass is made from blue 3 to blue 2, and (B) a rotating movement is executed between blue 3 and blue 5

$\mathbf{TP}_{m,2}$ improves ball circulation by allowing greater use of open space movements and ball passes. The authors of [19] describe a similar principle under the category of "width and length". This principle can also be observed in Fig. 1, where the players maintain spread by changing from a 1-3-1 formation to a 1-2-2 formation.

3.2 Opportunity

Definition 2 **Principle of opportunity**: "taking advantage of the opponents' mistakes is such an obvious principle that it does not deserve further development. Still, one must see the mistake and seize the occasion" [18, p. 16].

This category of principles provides the intention behind attack progression. The desired outcome is to get the ball closer to the opposition's goal, while creating (local) numerical superiority. The categories of "penetration" and "depth mobility" in [19] overlap to some extent with this category.

$\mathbf{TP}_{o,1}$ is repeatedly mentioned in [20]. The forward movement of blue 3 in Fig. 1 could have been performed for this exact reason. Our interpretation of a "defensive line" (dl) is visualized in Fig. 5. Numerical superiority is determined by counting the number of attackers and defenders to the left of a particular defensive line. One could hypothesize that the intention behind blue 3's initial movement is to pass the defensive line "dl2" and create numerical superiority together with blue 5.

$\mathbf{TP}_{o,2}$ advocates to switch the attack to the opposite side of the field if the opposition disrupts the ongoing attack. This principle is mentioned in strategy design in [19, 20]. In Fig. 1, this principle becomes apparent in the passes (C) and (E).

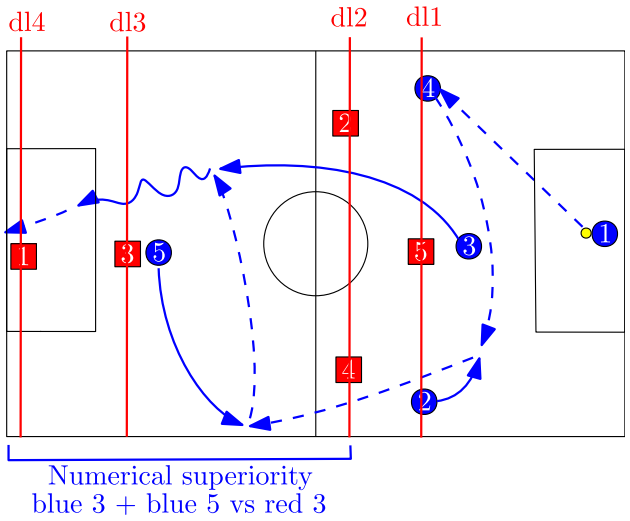


Fig. 5 The defensive lines (dl) drawn on Fig. 1. In case the opposition does not react fast enough, numerical superiority may be obtained in the area behind defensive line 2 when blue 3 passes it

$\mathbf{TP}_{o,3}$ advocates to pressure the penalty area, which is deemed to be a vulnerable area for the opposition. A similar principle is stated in the category of "penetration" in [19]. In Fig. 1, this principle becomes apparent via dribble (F) that brings the ball holder into the penalty area.

$\mathbf{TP}_{o,4}$ advocates to exploit areas that are (currently) undefended. The authors of [36] emphasize on the importance of increasing the effective space of the attacking team. For simplicity, this manuscript focuses on three predefined spaces (Fig. 6) to study the occupation of opposition space, rather than considering all possible open spaces. That distinction is made as strategies within [20] showed attacks to conclude often in one of these three spaces. In Fig. 1, one could hypothesize that another reason behind movement (B) is to take advantage of the undefended "TOP" space.

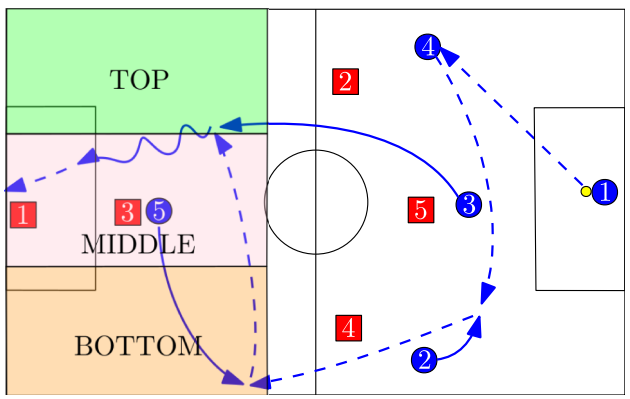


Fig. 6 The monitored spaces to progress attack of Fig. 1. The three rectangular spaces are "TOP" (green), "MIDDLE" (pink), and "BOTTOM" (orange) which have an equal width in vertical direction and some length in the horizontal direction. If no opponent is in one of those spaces, then we define that space to be not occupied

3.3 Deception

Definition 3 Principle of deception: "it means trying to deceive and bring the opponent to make a mistake, a bad move. In team sports, this principle is used at all levels, on a collective or individual basis" [18, p. 15]

This category of principles focuses on misleading the opposition, which is widely adopted in team sports [37]. The authors of [38] have shown that increasing the unpredictability of attacking movements benefits the attacking team. The authors of [39] describe ITS as a complex adaptive system in which two teams exchange information to increase or decrease entropy. Entropy is related to the stability of the opposition's defense, in which the attacking team's goal is to destabilize it by providing false information.

$\mathbf{TP}_{d,1}$ is our interpretation of creating confusion. It is based on the definition of deception [40] as *"an intentional attempt conducted by an agent (A) to induce or reinforce in another individual a belief that, in A's opinion, is false"*. The opposition thus needs to be convinced that the attack may progress in a different direction. In this case, we realize that by always having an extra offensive support available. The authors of [19] mention a similar principle of *"supporting the player in possession by providing options to give sequence to the play"*. In Fig. 1, we can observe at multiple moments that the ball holder has offensive support, e.g., at the moment blue 5 obtained the ball with blue 3 as support.

3.4 Reserve

Definition 4 Principle of reserve: "a support player is this player by whom the attack may be immediately restarted when a maneuver has failed. In soccer, having the forwards carry the ball makes it possible to distribute other players and constitute a reserve along the longitudinal axis of play" [18, p. 17]

This category of principles focuses on bringing stability in the attacking team's organization. That is necessary, as every time a teammate seeks an opportunity, it will move away from its initial position in the formation. In Fig. 4, that is apparent with movements (B) where blue 3 moves forward, leaving empty space in the attacker's formation. That empty space is covered by blue 5 moving backward in that same time instance [20]. In a sense, one should aim not to increase the entropy of the attacking team too much [39].

$\mathbf{TP}_{r,1}$ advocates for team members to think about reorganizing their attack in case the attack seems to be interrupted. The team member seeking an opportunity thus needs to understand when to stop that movement. That is also apparent in the butterfly rotation, where movements (B) in Fig. 4 can be understood as two sub-movements (B1) and (B2) in Fig. 7. Movements (B1) represent the duration in which

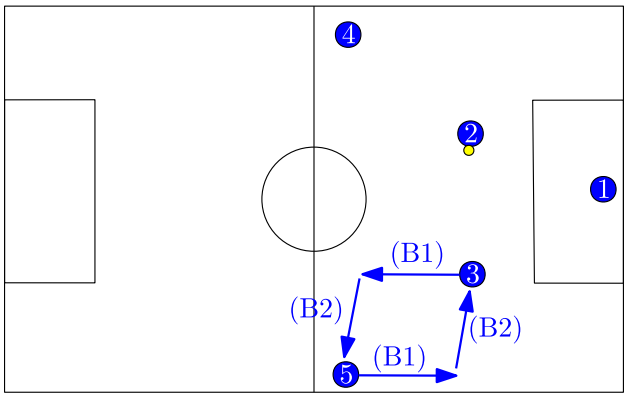


Fig. 7 Our interpretation of movements (B) in Fig. 4 as a decomposition of 1) movements (B1) for seeking opportunities, and 2) movements (B2) for resetting the attack

the opportunity-seeking movement can be tested. At the end of (B1), a decision needs to be made to either recover the formation via movements (B2), or commit to the opportunity-seeking movement, e.g., by letting blue 3 move further along the direction of (B1). That allows the team to "safely" perturb the opposition's defense while keeping tactical mechanisms intact to avoid self-disruption. We can hypothesize that such resetting mechanisms are also hidden in Fig. 1. For example, blue 3 could similarly abort mid-way movement (B) the attack to recover the 1-3-1 formation as in Fig. 8.

$\mathbf{TP}_{r,2}$ advocates safe passing options to maintain ball possession. A similar principle is advocated in the category of "width and length" in [19]. This principle is also apparent in Fig. 1, where blue 2 and blue 4 remain at the back.

3.5 Proposed Guideline on Using TPs

Figure 9 shows how the butterfly rotation can incorporate the four TP categories at each step of the strategy. During

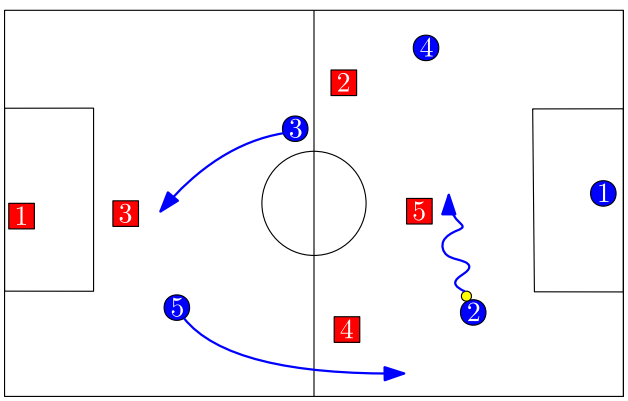


Fig. 8 The situation in Fig. 1 where 1) pass (C) is just completed, and 2) blue 3 is mid-movement (B) and blue 5 mid-movement (D). The attack is called off, and visualized are hypothetical movements to reset the attack

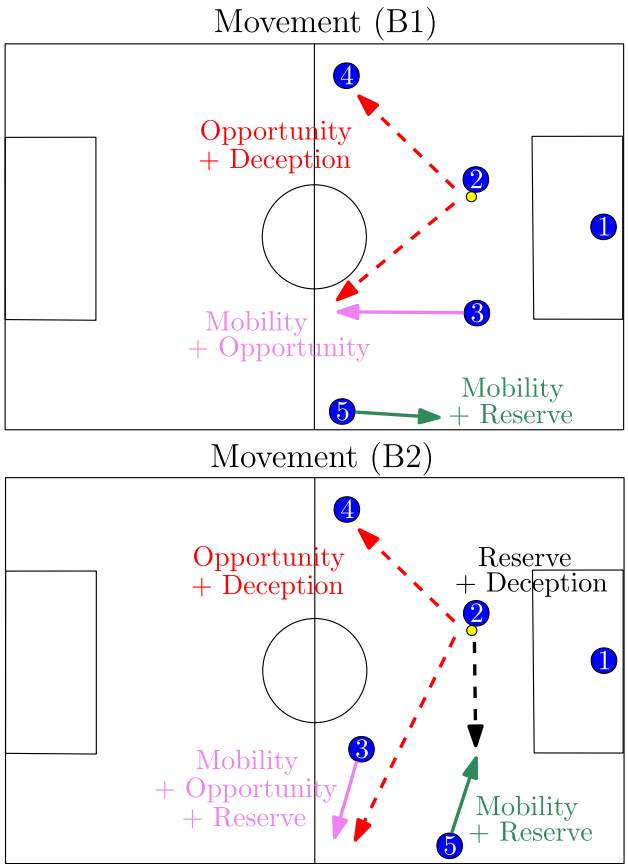


Fig. 9 Situation at movements (B1) and (B2) of Fig. 7. The visualization explicitly illustrates how the proposed four categories of TPs can be applied at each step of the strategy. Passes are visualized with dashed lines, and movements with solid lines. The color coding corresponds to the composition of TPs that can be present behind the action

movement (B1) (top), blue 3 and blue 5 move according to the mobility principle. The former does this according to the opportunity principle, while the latter does this according to the reserve principle. The deception principle is present, as blue 2 has multiple options to continue the attack, either by dribbling further, passing to blue 4, or passing to the hypothesized future position of blue 3. A similar reasoning can be applied on movement (B2) (bottom).

Based on the above observation, we outline an initial guideline for designing each step of the strategy, i.e., Strategy Guidelines (SGs):

- (necessary) **SG1**: incorporate $\text{TP}_{m,1}$ when progressing the attack.
- (optional) **SG2**: $\text{TP}_{m,2}$ increases flexibility in strategy execution. This is optional, as it may disrupt the ongoing attack by reducing local numerical superiority around the (future) ball holder.
- (optional) **SG3**: at least one of the opportunity-seeking movements of $\text{TP}_{o,1}$-$\text{TP}_{o,4}$ should be present. This is optional as 1) opportunity-seeking movements are not

always committed as seen in Fig. 7 with movement (B1), and 2) $TP_{d,1}$ ensures that some progress in the attack is present.
- (necessary) **SG4**: satisfy $TP_{d,1}$ to always have at least one team member present from which we may continue the attack.
- (necessary) **SG5**: $TP_{r,1}$ allows for stability in the attacking team and to re-initiate the attack.
- (optional) **SG6**: $TP_{r,2}$ improves the chance of maintaining ball possession, but may disrupt the ongoing attack. This is optional as there is always a keeper present.

The full strategy is then realized by sequencing each of the designed steps, resulting in a multi-step plan; that aligns well with ITS strategy design [41]. Section 4.2 elaborates further on this multi-step plan. Examples of the application of these SGs on our designed strategy (Fig. 11) is given in Appendix B.2.

To our knowledge, no explicit guideline of this nature exists in the ITS literature. That absence is understandable, given the diverse number of strategies that exist. Our aim is not to provide an optimal or universal approach to strategy design, but rather a structured starting point that can assist roboticists in formalizing their strategy in a manner that aligns better with ITS principles.

4 Robot Soccer Strategy Design

This section discusses a robot soccer strategy (Fig. 11 and Fig. 12) designed with the insights from Sec. 3. We discuss the formalization of TPs into 1) motion constraints and 2) perception constraints. Motion constraints are realized by restricting the path planning target to specific zones in the field. Perception constraints are realized by the design of specific transition conditions between steps in the strategy. Both the motion and perception constraints suggested in this manuscript solely serve as an initial starting point, and more optimal representations are reserved for future work. The formalization details are found in Appendices A.2 and B to keep the manuscript concise and readable. We conclude the section by elaborating on the rationale behind our designed strategy. Section 5 continues by experimentally validating the designed strategy.

4.1 Formalization of Tactical Principles

4.1.1 Motion Constraints

Robot soccer [7, 25] and human soccer [42, p. 107] commonly command players via a (sequence of) zone(s) on the field. One can imagine this as constraining the search space of a COP. That approach is also taken in this manuscript. For simplicity, the uniformly discretized field of 6×4 zones "Z#" in Fig. 10 is considered. That discretization is sufficient for the strategy design in this manuscript, while more extensive discretization is reserved for future work. Each zone "Z#" has four corner points $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$, with $x \in \mathbb{R}$ along the width of the field and $y \in \mathbb{R}$ along the length of the field.

Most of the TPs in Table 1 are formalized as constraints on zones to which the players should move towards, e.g., $\mathbf{TP}_{o,3}$ is enacted by constraining the motion target of a player to be within either Z2 or Z3. That formalization is described in further detail in Appendix B.1 with Table 4.

4.1.2 Perception Constraints

A strategy can be envisioned as a multi-step plan with transition conditions between them. Opportunity principles ($TP_{o,1}$–$TP_{o,4}$) suggest not only motion constraints, but also transition conditions for determining whether the opportunity-seeking movement needs to be initiated. For example, a movement to threaten the "TOP" space (by moving to Z1) should be done if the robot can perceptually observe that the "TOP" space remains (sufficiently) undefended (by checking opposition on Z1, Z2, Z5, Z6, Z9, and Z10). Appendix A.2 provide further details about the formalization of these opportunistic transition conditions.

The reserve principle $TP_{r,1}$ suggests a transition condition for determining when an attack needs to be reset. The blue highlighted areas in Fig. 11 determine "how long" the opportunity-seeking movement needs to be done (Fig. 12); that blue area essentially represents the stopping point of (B1) and starting point of (B2) in Fig. 7. Appendix B.2 elaborates further on our chosen blue areas.

Additionally, the reserve principle $TP_{r,1}$ suggests to not move "too far away"(ZC_7; Table 4) from a hypothetical reset-

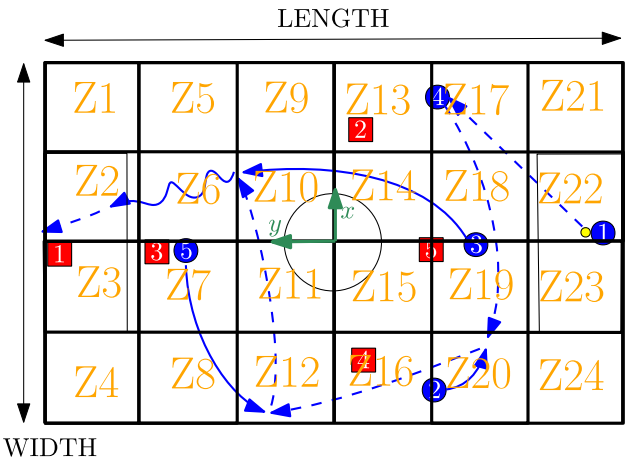


Fig. 10 A field segmentation of equal zones in a 6×4 grid applied on top of the soccer field in Fig. 1. Each zone is labeled with an orange label "Z#"

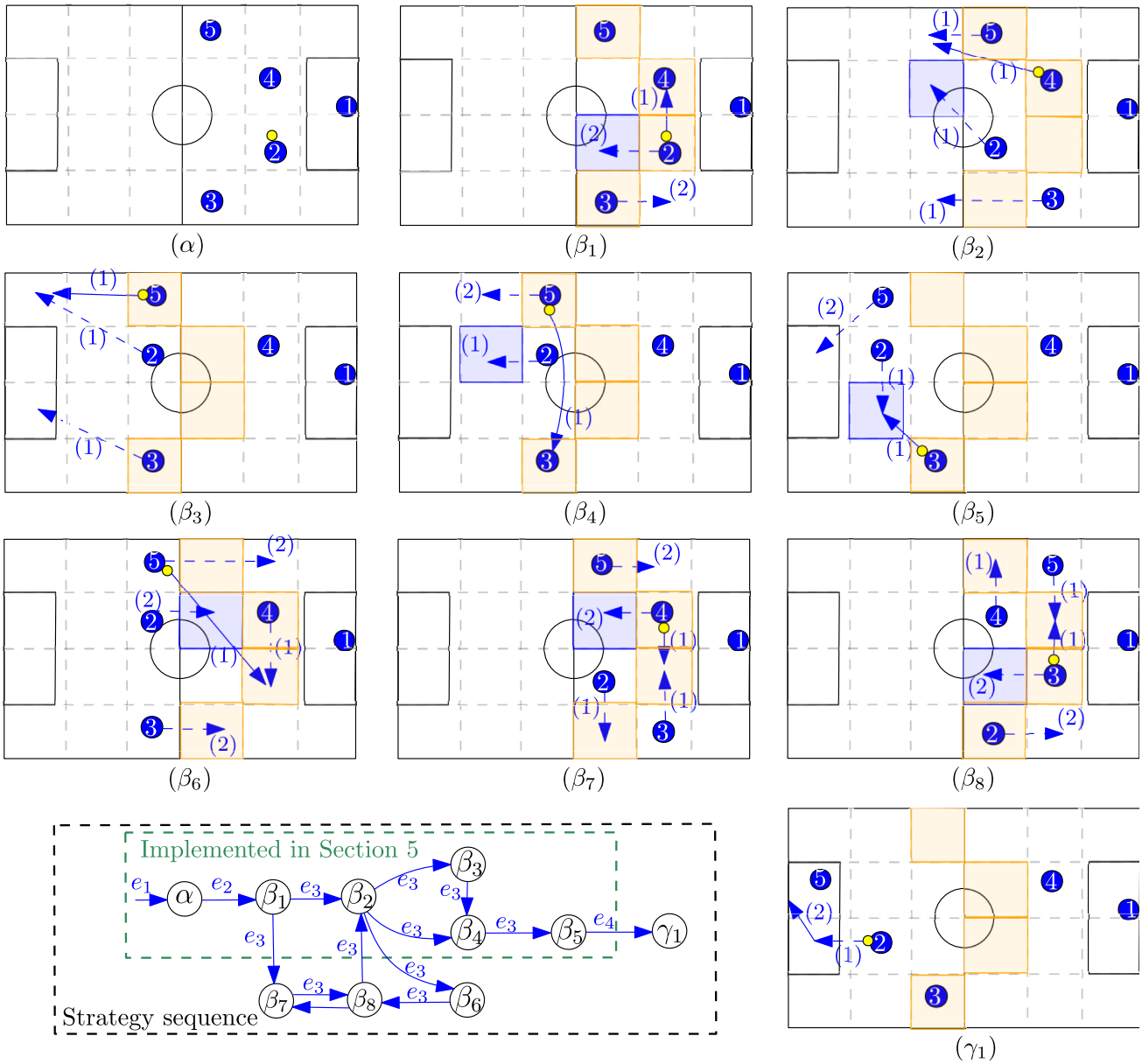


Fig. 11 Illustration of the designed strategy. The blue highlighted area within each state represents the halfway zone at which a decision needs to be made to commit or reset the attack ($TP_{r,1}$). The yellow highlighted areas in each state represent the formation to reset the attack ($TP_{r,1}$). The numbers in bracket indicate which actions occur simultaneously or sequentially. At the bottom left, the state diagram is visualized to show the state dependencies. Only parts within the green dashed square is implemented and tested in Sec. 5

ting formation; represented as the occupation of the yellow highlighted areas in Fig. 11. For example, in β_1 players reset the attack by moving to the formation in α. The resetting formation can shift "forward" (as seen via the yellow areas in β_5) if sufficient progress has been made, as suggested by [20]. Our rationale was to shift the formation forward if 1) three players are in the opposition's half of the field, and 2) the intention is to progress the attack further; β_6 is excluded as the intention is to reset the attack.

4.2 Executable Action Framework

$\mathbf{TP}_{m,1}$ is achieved by constraining the sequence of actions in the strategy. Figure 13 illustrates an initial framework to implement the suggested "pass and move" structure (blue dashed rectangle). This framework only serves to encode executable actions, where the combination of motion constraints (Table 4) and perception conditions (e; Table 2) is used to encode the intentions behind the strategy in Fig. 11.

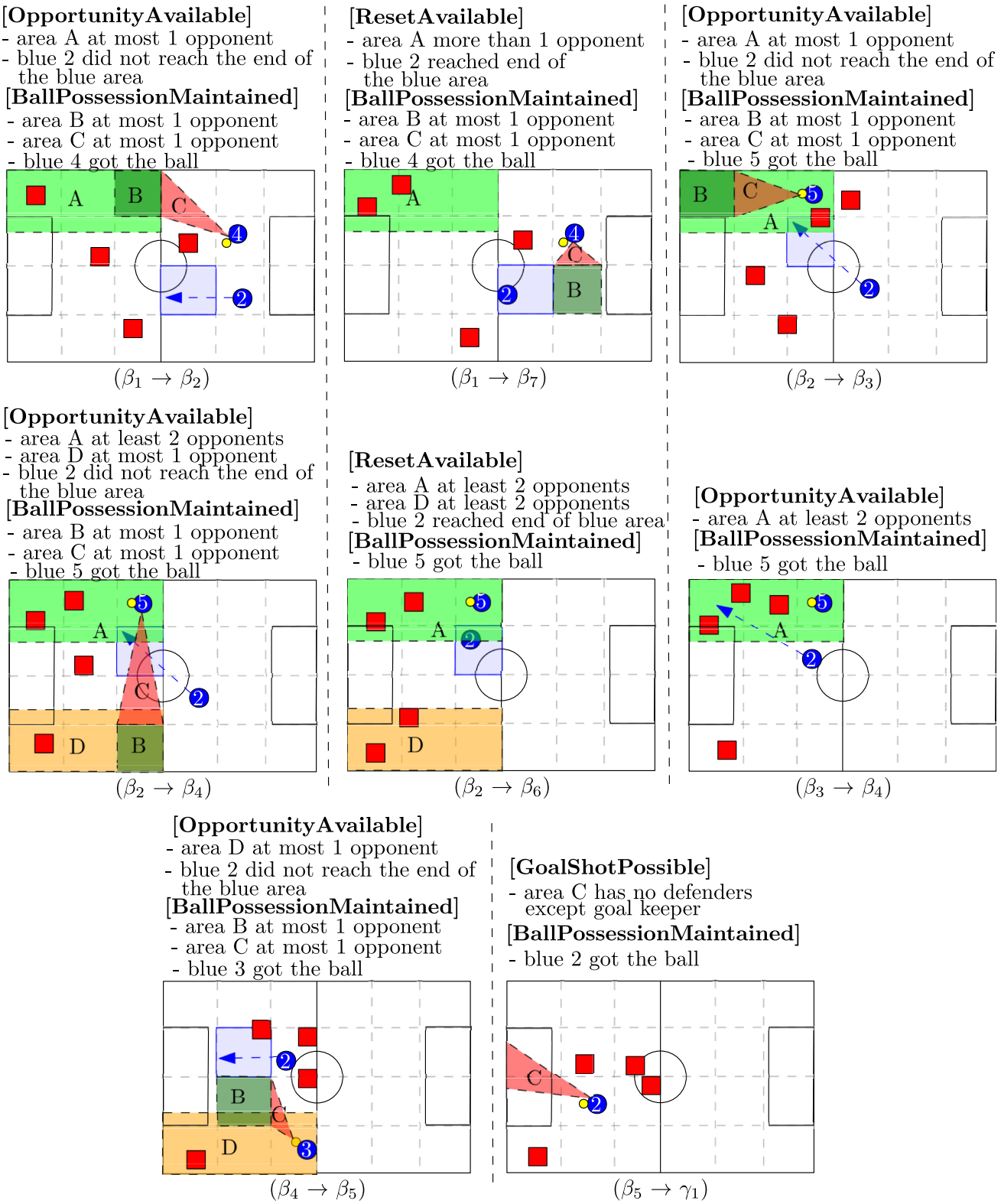


Fig. 12 Illustration of the implemented transition conditions of the strategy in Fig. 11. Each transition visualizes the situation on the field for which the transition is valid. In brackets the abstract transition condition of Table 2 is formulated, and directly below the specific implemented conditions (distinguished via dashes) that needs to be satisfied to fulfill the corresponding abstract condition. Every mentioned condition needs to be satisfied for a transition to occur. Area A is light green, area B is dark green, area C is red, and area D is orange

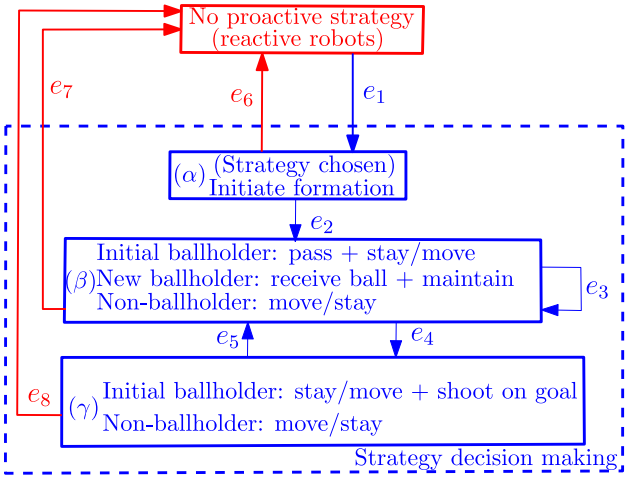


Fig. 13 The sequence of proposed actions that would satisfy the tactical principle $TP_{m,1}$ in a designed strategy (blue dashed square). The blocks represent states, and the arrows the transitions conditions between them (Table 2). The red colors are related to reactive robot behavior, while the blue colors are related to the strategy design

By default, we assume that the soccer robots do not use a proactive strategy and are solely *reactive* (Sec. 2.1), as it is challenging to define strategies to address all possible scenarios.

Table 2 uses abstract conditions as `OpportunityAvailable`, where its implementation can vary (Appendix A). Generic descriptions of these abstract conditions are:

- `FormationRealized`: a description of the field to determine when the non-ball-holding players have "sufficiently" realized the desired formation.
- `BallPossessionMaintained`: a description of the field to determine whether ball possession can be maintained currently and/or in the future.
- `OpportunityAvailable`: a description of the field that allows the strategy to progress the attack by capitalizing on some opportunity. It uses the constraints described in Sec. 4.1.2.
- `ResetAvailable`: a description of the field that is necessary to allow a reset of the attack. Generally, this implies that no `OpportunityAvailable` is possible.
- `GoalShotPossible`: a description of the field to determine whether it is reasonable to attempt a goal shot.

We kept these conditions abstract, as no consensus exists in literature on what kind of conditions should be used to evaluate the field. For example, the authors of [43, 44] analyze dominant regions, while the authors of [45] analyze opponent formation. Future work is thus reserved on designing more precise transition conditions.

Figure 12 visualizes the implemented transition conditions of each step. The implemented conditions are listed below the abstract condition (in brackets) to which it belongs; all these conditions need to be satisfied for a transition to occur. These implemented conditions are selected and designed based on 1) common-sense reasoning – for example, we expect the ball holder to have sufficient skills to deal with one defender –, and 2) experiences described in ITS literature [18–20]. We want to remark that the strategy in Fig. 11 is not claimed to be perfect. Transitions can be added, e.g., an alternative option could exist after β_4 to pass back to blue 5. The lack of completeness is due to 1) conciseness of the manuscript, as the authors of [41] have shown that an offensive basketball strategy can exceed 100 state transitions, and 2) the absence of an established framework in designing the spatio-temporal evolution of a strategy; this manuscript formalizes generic principles for designing robot soccer strategies, and leaves formalization on strategy spatio-temporal evolution to future work.

Appendix B provides details about the application of the guideline (Sec. 3.5) in the design of the strategy and the formalization of its transition conditions, while Appendix C

Table 2 The left column describes the edges of Fig. 13, and the right column describes the conditions that needs to be satisfied for these edges to be applicable

Edges	satisfaction of conditions
e_1	Strategy dependent initialization.
e_2	`FormationRealized` AND `BallPossessionMaintained`.
e_3	(`OpportunityAvailable` OR `ResetAvailable`) AND `BallPossessionMaintained`.
e_4	`GoalShotPossible`.
e_5	(`OpportunityAvailable` OR `ResetAvailable`) AND (not `GoalShotPossible`) AND `BallPossessionMaintained`.
e_6	not `BallPossessionMaintained`
e_7	(not `OpportunityAvailable` AND not `ResetAvailable`) OR (not `BallPossessionMaintained`)
e_8	(not `OpportunityAvailable` AND not `ResetAvailable` AND not `GoalShotPossible`) OR (not `BallPossessionMaintained`).

`FormationRealized`, `BallPossessionMaintained`, `OpportunityAvailable`, `ResetAvailable`, and `GoalShotPossible`, represent abstract conditions, where its implementation can vary. Possible implementations are discussed in Appendix A

provides details about robot software implementation. We will clarify the rationale behind the designed strategy next.

4.2.1 Initialization (α)

The strategy has the butterfly rotation in Fig. 4 as a starting point. The strategy is initiated when 1) any player obtains the ball on Z19, and 2) no proactive strategy is currently active. The other non-ball-holding players will move to occupy Z13, Z16, and Z18. `BallPossessionMaintained` is realized via two conditions: 1) the ball holder can retain the ball, and 2) the suggested pass from blue 2 to blue 4 in the sequential state β_1 can likely be executed, thus maintaining ball possession in the future. The edge e_2 to β_1 is then initiated when conditions `FormationRealized` and `BallPossessionMaintained` are satisfied. If `BallPossessionMaintained` cannot be satisfied, the team can decide to default back to the reactive behavior via edge e_6.

4.2.2 Pass and Move (β)

The states β_1–β_5 are used to progress the attack, while the states β_6–β_8 are used to reset the attack. The transitions between β's occur via edge e_3 which checks: 1) whether ball possession can be maintained and/or a safe pass can be performed to the player suggested in the sequential state (`BallPossessionMaintained`), and 2) whether the opportunity-seeking movement is feasible (`OpportunityAvailable`); if not, under which conditions we should reset the attack (`ResetAvailable`). A transition to reactive robot behavior is possible via edge e_7 if either 1) no transitions exist to leave the state, i.e., the strategy is not further defined anymore, or 2) there is a too high risk to lose the ball. Next, we explain the β's:

- β_1, β_7, and β_8: these states are designed according to the butterfly rotation (Fig. 4). An infinite loop can occur between β_7 and β_8, requiring a player label reassignment. For example, blue 4 becomes blue 5 when transitioning from β_8 to β_7. For clarity, we have not drawn the opportunity-seeking transitions from β_7 onward.
  The transition $\beta_1 \rightarrow \beta_2$ in Fig. 12 requires that the "TOP" space (area A) has at most one opponent in it to allow us to capitalize on it in the future in β_3. To increase the chance of a successful pass in β_2, a check is performed that both area B (zone of the future ball holder) and area C (passing area) each have at most one opponent. The requirement of "one opponent" is based on our expectation that the player's skills should be sufficient to deal with one opponent. The choice of attacking the "TOP" space is because that is furthest away from the ball holder's position (blue 2) when initiating β_1, thus making our attack less predictable.
  The resetting transition $\beta_1 \rightarrow \beta_7$ requires that the above attack progressing transition remains invalid while blue 2 moves towards the end of the blue area.
- β_2: a continuation of the attack from β_1 or β_8 with the intent to attack the "TOP" space. The movement of blue 2 to Z10 is intended to provide local numerical superiority to the future ball holder blue 5. That movement is inspired by [20], which demonstrated its effectiveness in enabling the attack to bypass a hypothetical opposition pressing blue 5 in two ways. One of them is visualized in β_3 with blue 2 moving to Z1. An alternative is that blue 2 stops at Z10 to receive a pass from blue 5, after which blue 5 moves to Z1 to receive the ball back. The movement of blue 3 is to maintain pressure on the opposite side of the field.
  The transition $\beta_2 \rightarrow \beta_3$ requires that the "TOP" space remains sufficiently undefended. In contrast, the transition $\beta_2 \rightarrow \beta_4$ requires a sufficiently undefended "BOTTOM" space; we hypothesize that the opposition rather expects an attack on "TOP" space as the new ball holder (blue 5) is within that area.
  The resetting transition $\beta_2 \rightarrow \beta_6$ requires that neither of the two above mentioned transitions become valid while blue 2 moves to the end of the blue highlighted area.
- β_3: a continuation of β_2 with the intent to further attack the "TOP" space. Blue 3 offers a passing option to blue 2 in the penalty area on the opposite side of the field in case a goal shot cannot be attempted.
  The transition $\beta_3 \rightarrow \beta_4$ is an attempt to continue the attack to the "BOTTOM" space in case the team has misjudged the situation in transitioning to β_3. This is often

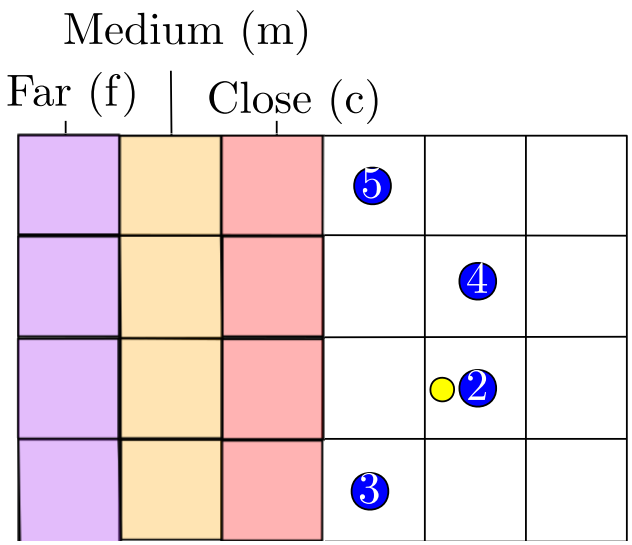


Fig. 14 The initial configurations of the simulated experiments. The gridded field representation aligns with Fig. 10. The attacking team configuration (right) does not change. The defending team occupies either the close (c) red areas, medium (m) orange areas, or far (f) purple areas

Table 3 The results of the simulated experiments

Id	Selected attack	Success	Last state	Stolen ball	Lost ball self
c_1	Too early	-	β_2	X	-
c_2	C	X	β_5	-	-
c_3	Too early	-	β_1	-	X
c_4	C	-	β_5	X	-
c_5	A	X	β_3	-	-
c_6	A	-	β_3	X	-
c_7	B	-	β_5	-	X
c_8	A	-	β_3	X	-
c_9	A	X	β_3	-	-
m_1	A	X	β_3	-	-
m_2	Too early	-	β_1	-	X
m_3	A	-	β_3	-	X
m_4	A	X	β_3	-	-
m_5	B	-	β_5	X	-
m_6	Too early	-	β_2	X	-
m_7	C	X	β_5	-	-
m_8	C	-	β_5	-	X
m_9	A	X	β_3	-	-
f_1	A	-	β_3	-	X
f_2	A	X	β_3	-	-
f_3	A	X	β_3	-	-
f_4	Too early	-	β_1	-	X
f_5	Too early	-	β_2	X	-
f_6	B	X	β_5	-	-
f_7	Too early	-	β_1	-	X
f_8	A	X	β_3	-	-
f_9	A	X	β_3	-	-

Details are found in Appendix D.2 (and Supplementary video material). "Id" refers to the defense configuration in Fig. 14. "Selected attack" implies the executed attack, with the output "Too early" implying strategy failure at β_1 or β_2. "Success" implies a successful strategy execution where the final ball holder receives the ball in the intended zone. "Last state" implies the last executed state. "Stolen ball" implies ball loss due to the opponent. "Lost ball self" means the intended receiver did not catch the ball

due to blue 2 being blocked from reaching Z1, increasing the time of blue 5 maintaining ball possession and delaying the attack. No checks for pass success are performed (areas B and C) in order to determine the consequence of a "looser" transition condition; section 5.1.3 shows that good robot skills can still ensure a good strategy execution.

- β_4: a continuation of the attack from β_2 with the intent to attack the "BOTTOM" space. Both blue 2 and blue 5 move forward to maintain pressure on the opposite side of the field.
  The transition $\beta_4 \rightarrow \beta_5$ requires the "BOTTOM" space to remain sufficiently undefended.
- β_5: a continuation of β_4 with the intent to further attack the "BOTTOM" space. Blue 2 moves to Z7 to achieve local numerical superiority with blue 3. Blue 2 receives a pass to get the ball closer to the goal, and therefore increase the chance of scoring a goal. Blue 5 threatens the penalty area on the opposite side of the field.
  The transition from $\beta_5 \rightarrow \gamma_1$ requires that the area C towards the goal remains undefended.
- β_6: to reset the attack from β_2.

4.2.3 Finish (γ)

State γ_1 continues on β_5 by dribbling to the penalty area and shooting on goal, as inspired by Fig. 1. Blue 5 positions itself to create ambiguity of attack progression and increase the chance of local numerical superiority. The edge e_4 checks if a goal attempt seems likely.

The edge e_5 is enabled in case 1) the goal shot opportunity disappears during the execution of γ_1, 2) state transitions exist to continue the strategy, and 3) the ball can likely be maintained. The edge e_8 is enabled in case 1) the ball cannot

be maintained, or 2) no goal shot opportunity remains and the strategy cannot be continued.

5 Experimental Validation

Experiments evaluate the strategy confined in the green dashed box (Fig. 11 left-bottom). That limitation exists, as the attacking robots' software cannot reliably execute the long sequence of passes required for resetting attack transitions, e.g., via β_7; for the real-world experiments, additional hardware related problems were present. The strategy is implemented into the Skills, Tactics, and Plays (STP) framework used by Tech United Eindhoven (see Appendix C for details). Section 5.1 includes a preliminary sensitivity analysis to evaluate the strategy in varying conditions in simulation. Section 5.2 illustrates the strategy applicability on real robots. Both types of experiments deploy the Tech United Eindhoven software[5] on the attacking team. A distributed multi-robot team is considered in which each robot maintains its own estimate of the world state and determines its own strategy [30]. A voting system is in place to achieve consensus, e.g., on what role the robot should take in the strategy, though this approach does not guarantee unanimous agreement at all times (Sec. 5.1.4).

5.1 Simulated Experiments

The Tech United Eindhoven simulator[6] is used to perform the simulations. The defending team also uses the Tech United Eindhoven software and behaves reactively, with each robot iteratively optimizing a set of weighted cost-functions[7]. The simulation uses two hardware units, each independently calculating the team actions that are primarily limited by CPU performance. The defending team runs on an Ubuntu 20.04.6 LTS laptop with an Intel Core i7-9750H CPU, while the attacking team runs on an Ubuntu 20.04.6 LTS desktop with an AMD Ryzen 5600X CPU.

A preliminary sensitivity analysis is included to demonstrate the robustness of our approach. Figure 14 illustrates the three evaluated initial configurations, each tested 9 times for which the results are summarized in Table 3. The hypothesis behind this experimental design is that putting defenders closer to the attackers reduces the effective play area, thus negatively influencing the strategy execution. An exhaustive sensitivity analysis is reserved for future work, as the main focus of this manuscript is on the formalization of human tactical principles into robot soccer strategies, rather than optimizing strategies beyond the state-of-the-art; section 6.1 discusses this further. During the strategy execution, the attacking team decides at run-time to advance the attack in three directions: attack A ($\beta_1 - \beta_2 - \beta_3$), attack B ($\beta_1 - \beta_2 - \beta_4 - \beta_5$), or attack C ($\beta_1 - \beta_2 - \beta_3 - \beta_4 - \beta_5$). The baseline of attacks A and B without interaction with the defense is visualized in Fig. 15. The baseline of attack C is not visualized as that is dependent on the interaction with the defense. A "success" in Table 3 implies whether 1) attack A finishes with blue 2 receiving the ball in Z1 (β_3), or 2) attacks B and C finish with blue 2 receiving the ball in Z7 (β_5).

Strategy robustness is further tested by the reactive defenders, whose behavior can vary even with similar initial conditions. Figure 16 visualizes the attacker and defender trajectories for two experiments initialized with the same conditions. Although the defensive team's behavior varies, one can observe that the attacking team's behavior corresponds well with the baseline of attack A with (slight) variations in the continuous motion. The zone constraints as

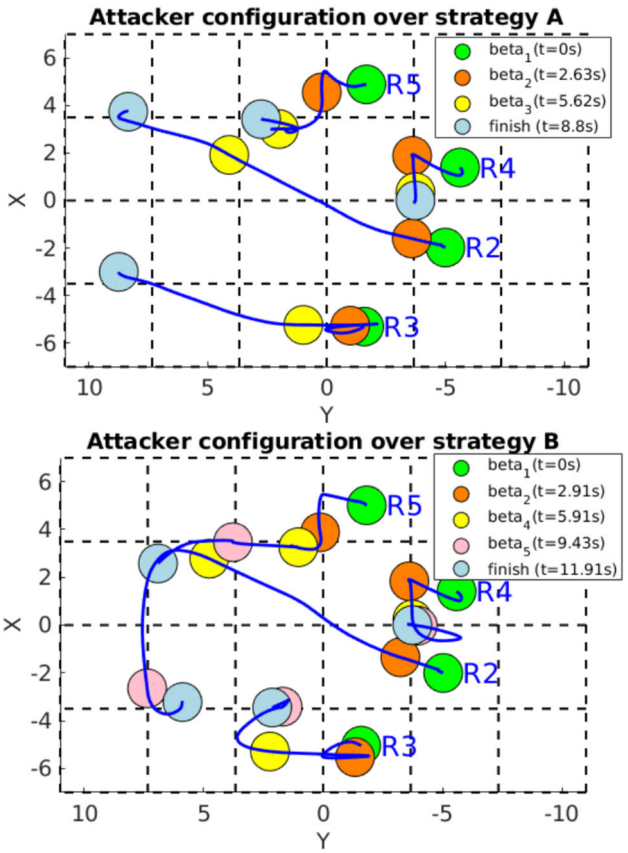


Fig. 15 Baseline trajectories (blue lines) of all the attackers for attack A (top) and attack B (bottom), plotted on the robot soccer field with a gridded representation consistent with Fig. 10. Each attacking robot is labeled "R#" in correspondence with Fig. 11. Colored circles indicate the time-stamps when 1) a new state in the strategy is initiated (green = β_1, orange = β_2, yellow = β_3/β_4, pink = β_5), or 2) when the strategy is finished/aborted (light blue)

[5] https://gitlab.tue.nl/tech-united-eindhoven/Turtle3/

[6] https://gitlab.tue.nl/tech-united-eindhoven/Turtle3/-/tree/master/Simulator

[7] https://gitlab.tue.nl/tech-united-eindhoven/Turtle3/-/blob/master/Strategy/src/mu_positioning_lib/muFields/mu_field_defensive.c

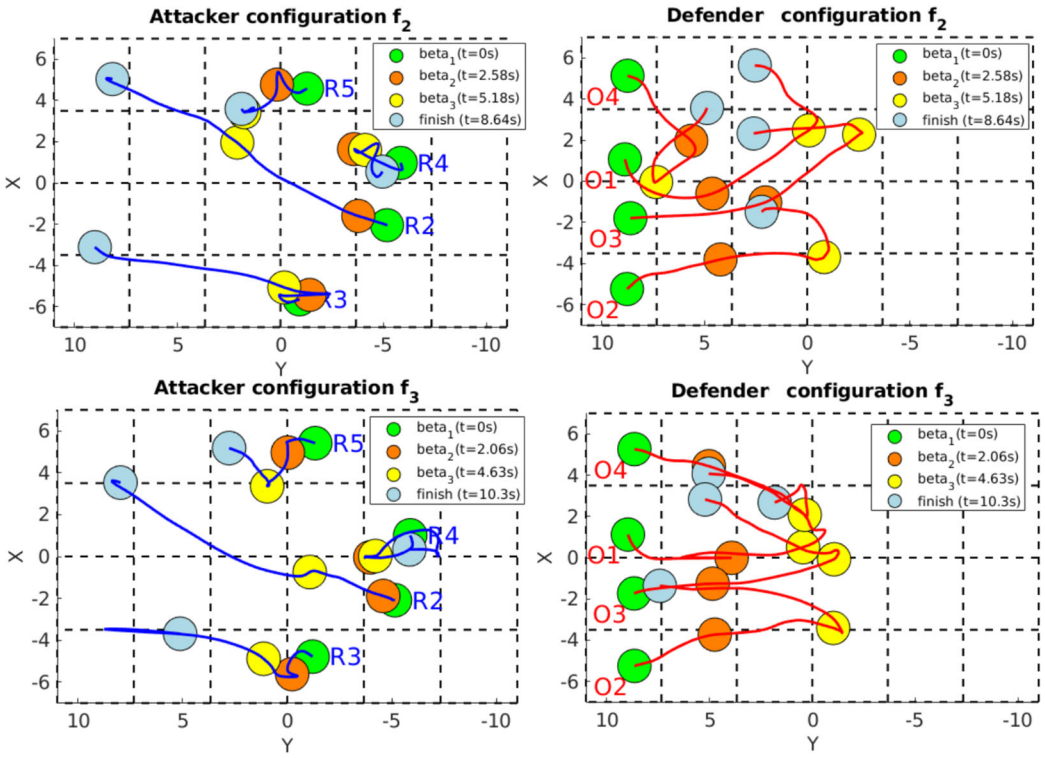


Fig. 16 Illustration of the attacker and defender behavior on the soccer field for experiments f_2 (top) and f_3 (bottom). In both cases, attack A is executed. Both experiments are started with identical initialization conditions. On the left, the attackers trajectories are visualized (blue lines), and on the right the defenders trajectories are visualized (red lines). The colored circles, indicating the moments at which the strategy changes, are identical for both the attacker and defender team, where the color coding adheres to the one described in Fig. 15

suggested in Fig. 11 are (mostly) respected. This highlights the robustness of our approach even in varying (dynamic) conditions. Next, we provide qualitative observations of the results of Table 3.

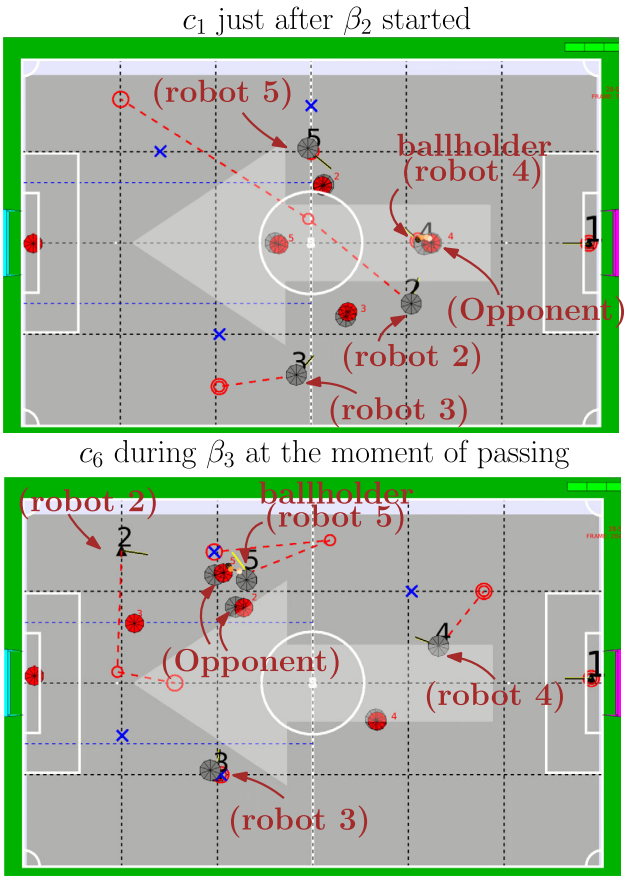
5.1.1 Pressure on Mid-field

The results in Table 3 suggest that stolen ball events increase whenever the defenders are positioned closer to the attackers. That supports our hypothesis that decreasing the attacking team's effective play area is disadvantageous. Figure 17 visualizes situations in c_1 and c_6 just before the ball has been stolen by the marked opponent(s).

Another consequence of positioning the defense closer is the disruption of robot 2's movement during β_2. That is expected, as the defense can easier maintain pressure around the mid-field zones (Z10, Z11, Z14, and Z15) which robot 2 wants to cross. Figure 18 visualizes the player configurations at the end of β_2 for three different experiments enacting attack A. The ideal situation is visualized on top, where robot 2 can follow its diagonal trajectory without disruption. That allows the pass from robot 5 to robot 2 to occur in only 1.8 seconds and corresponds well with baseline attack A. The situation in the middle, however, shows two defenders blocking robot 2's path. Robot 2 slows down and takes a detour to avoid collision with the defenders. That causes robot 5 to pass to robot 2 with a delay of 3.8 seconds which allows the defense to intercept the pass. The situation at the bottom shows the attacking team taking advantage of a pressured mid-field, where robot 2 blocks the marked opponents from reaching robot 5.

The disruption of robot 2's movement also negatively affects attacks B and C in Fig. 19. At the top, the defense pushed robot 2 "above" robot 5, deviating attack B from its baseline. Robot 2's movement towards the bottom-left of the field is disrupted by both the defense and robot 5. As a result, robot 2 is off-position for a future pass in β_5. At the bottom, a closeby opponent blocks robot 2's path. Consequently, the defense can react more quickly to the future pass of robot 3 to robot 2 in β_5.

Observation 1 The strategy may need to include some "expected completion time" constraint for the separate states. If the attacking team does not expect to complete the strategy in time, e.g., due to disruption of specific player's movement, then a reset/abort of the attack should be suggested.



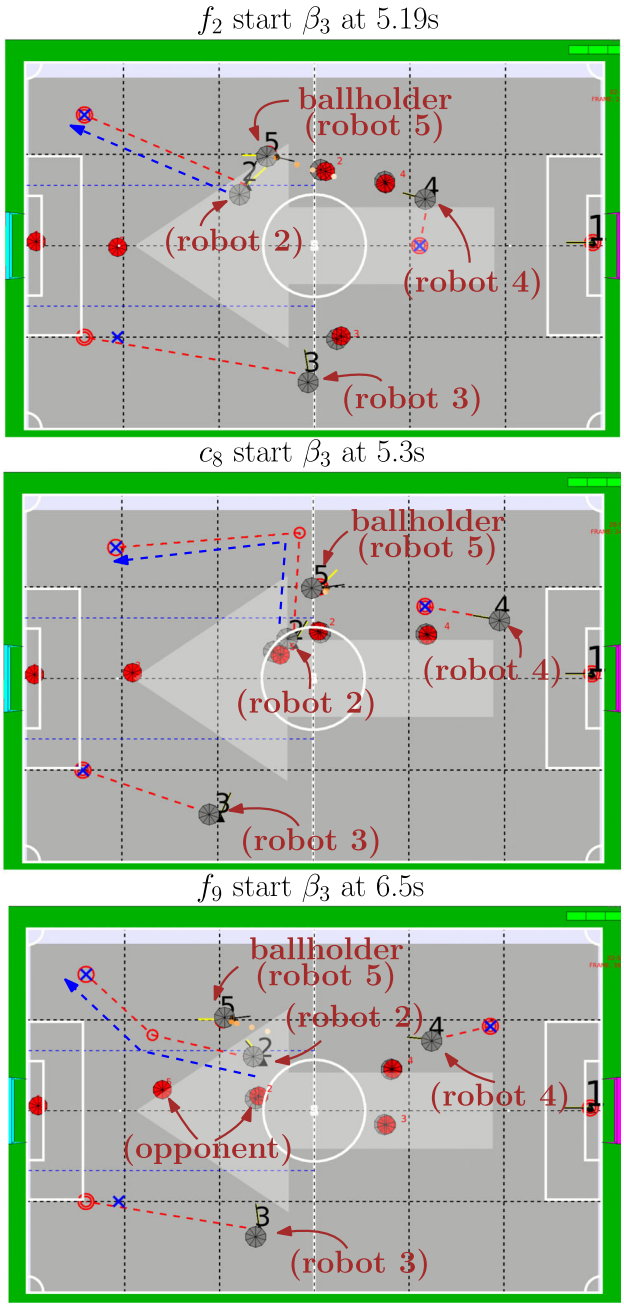


Fig. 17 Visualization of two soccer fields in the Tech United Eindhoven simulator. The gridded representation of the soccer field corresponds to Fig. 10. Experiment c_1 (top), and c_6 (bottom) are visualized at moments when the opponent pressures the ballholder. The attacking team starts on the right side (gray circles), while the defending team starts on the left side (red circles). The red dashed lines represent the output of the individual player's motion planner. Explicit brown labels are assigned to the attacking (and defending) team robots in correspondence to Fig. 11

Fig. 18 Visualization of three simulated soccer fields. Experiment f_8 (top), c_8 (middle), and f_9 (bottom) are visualized directly at the start of β_3. The trajectory of robot 2 is highlighted with a blue dashed arrow

5.1.2 Timing of State Transitions

The strategy is designed with quick state transitions in mind. Experimental results show that delaying the pass in β_2 can bring benefits to later states of the attack. Figure 20 shows the player configurations for a delayed transition from β_2 to β_4 for attack B. Robot 4 kept the ball for a long time before initiating the pass to robot 5. Although β_4 for c_7 started at a later time than β_4 for m_5 (Fig. 19 top), we observe 1) a more favorable position of robot 2 to move undisturbed to the left-bottom of the field, and 2) a more favorable spread of the defense away from robot 3. The latter is a direct consequence of the former, as a deeper progression of robot 2 into the "TOP" space alerts the defense to its presence.

The strategy needs to quickly make use of the favorable situation. A transition from β_4 to β_5 occurs in 2.7 seconds (baseline 3.52 seconds), and β_5 would have been completed in 3 seconds (baseline 2.48 seconds) if the ball had been caught. More examples of a delayed β_2 can be found in experiments m_4, f_6, and f_8. Figure 21 visualizes the situation in f_6 with robot 2 on position at the moment robot 5 obtains the ball.

Observation 2 The initial part of an attack can be executed slower, focusing on spreading the opposition and creating favorable spaces for the continuation of the attack, whereas the latter part of an attack needs to make quick use of that created space by executing faster.

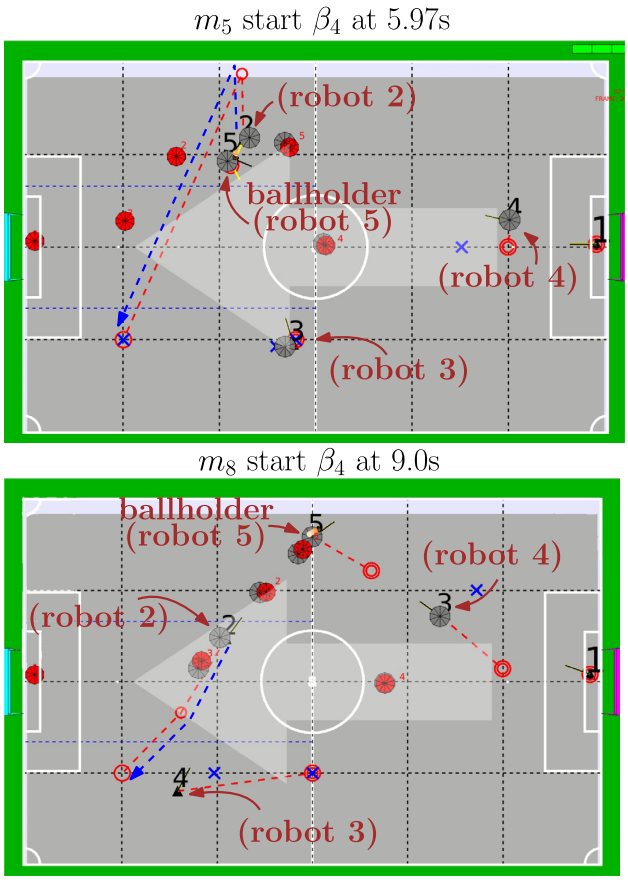


Fig. 19 Visualization of two simulated soccer fields. Experiment m_5 (top), and m_8 (bottom) are visualized directly at start of β_4. The trajecory of robot 2 is emphasized by an additional blue dashed arrow. Experiment m_8, due to software issues, wrongly assigned the robot 3 and 4 roles initially (Sec. 5.1.4)

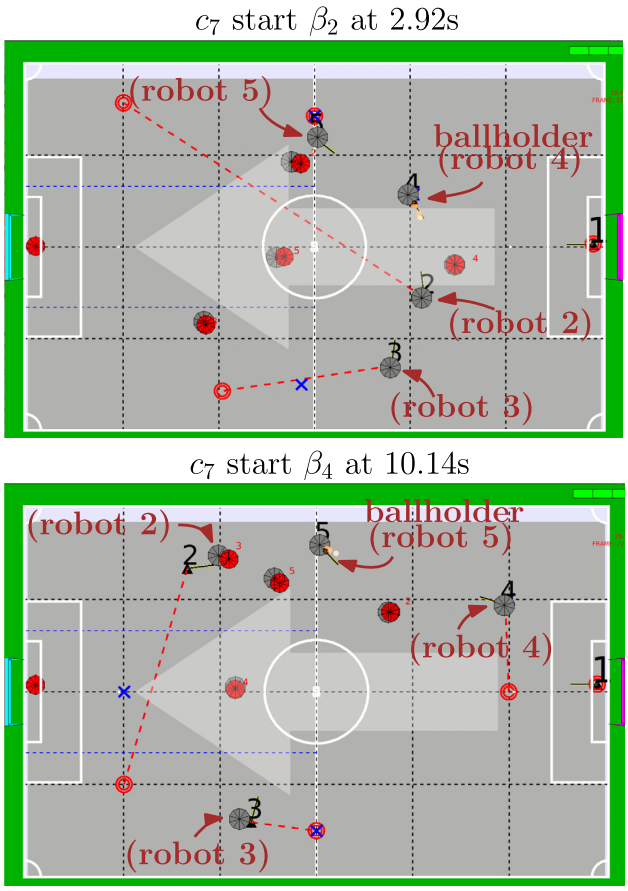


Fig. 20 Visualization of two simulated soccer fields. Experiment c_7 is visualized at the start of β_2 (top) and at the start of β_4 (bottom)

5.1.3 Robot Skills Influence Strategy

Figure 22 visualizes some challenging situations during the execution of experiment c_2. The intention of β_4 is to provide a pass from robot 5 to robot 3. The situation at top shows three defenders blocking this passing line. Robot 5's skills are good enough to dribble past the defenders before initiating a pass to robot 3. The intention of β_5 is to provide a pass from robot 3 to robot 2. The situation at the bottom shows three defenders surrounding robot 2. Yet, the pass succeeds and therefore attack C is completed.

Observation 3 The outcome of a strategy is dependent on a lot of nuances, e.g., the low-level skills of the attacking and defending robots.

5.1.4 Miscommunication Distributed Multi-robot System

We may attribute most of the failed experiments in Table 3 to the distributed multi-robot software architecture, and not the strategy itself. Figure 23 visualizes the situation before a pass is given from robot 2 to robot 4. Both robots are under no pressure, as the opposition is positioned far on the left side. However, robot 2 passes too hard for robot 4 to receive the ball. Most likely, a miscommunication is present on the time of arrival of robot 4 on the pass target. Appendix D.1

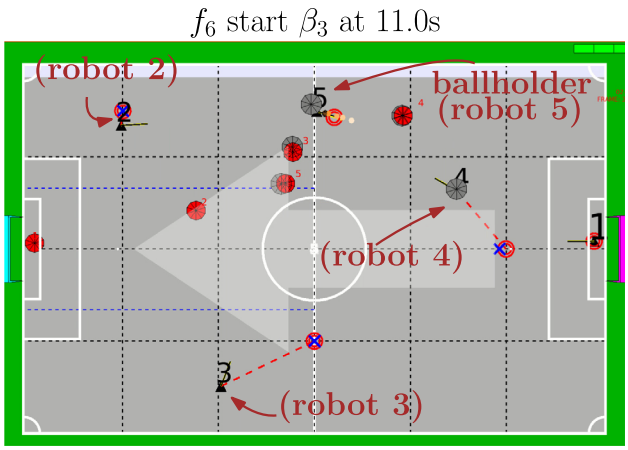


Fig. 21 Visualization of one simulated soccer field. Experiment f_6 is visualized at the start of β_3

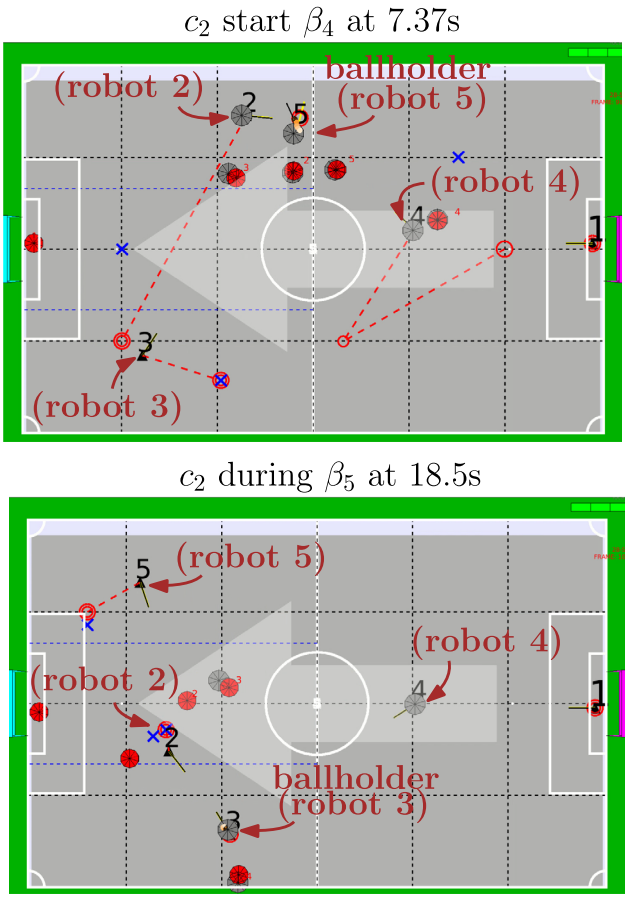


Fig. 22 Visualization of two simulated soccer fields. Experiment c_2 is visualized at the start of β_4 (top) and the moment a pass is given at β_5 (bottom)

elaborates further on the failed experiments with the "Lost ball self" indication; suggesting that c_3, m_2, m_3, f_1, f_4, and f_7 likely stem from software problems.

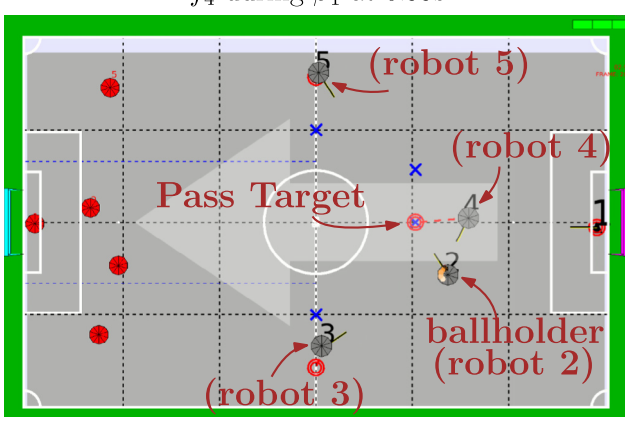


Fig. 23 Visualization of one simulated soccer field. Experiment f_4 is visualized at the start of β_1. Robot 2 wants to pass to the pass target, while robot 4 wants to move to it

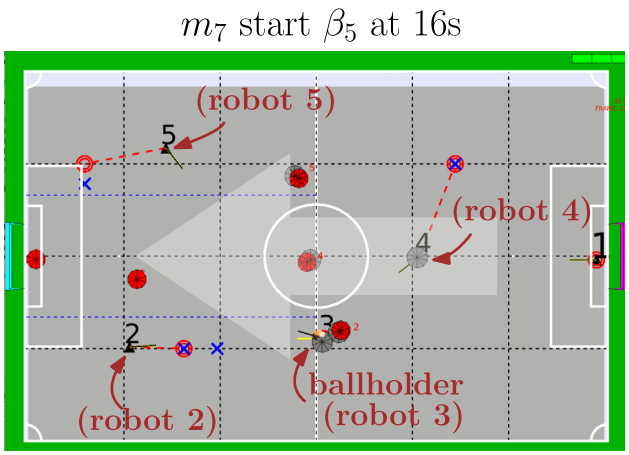


Fig. 24 Visualization of one simulated soccer field. Experiment m_7 is visualized at the start of β_5

Observation 4 Evaluating and executing a strategy is challenging whenever low-level skill components of the attacking robots cannot be executed reliably.

5.1.5 Restrictiveness of Strategy

Figure 24 visualizes the situation at which robot 3 just obtained the ball and β_5 is initiated. Although attack C is completed, robot 2 cannot make a goal attempt due to the nearby defenders. An alternative option in which robot 3 passes the ball to robot 5 instead may have been better, as 1) the defenders are positioned further from it, 2) it is closer to the goal, and 3) the passing line towards robot 5 seems undefended. The currently designed strategy, however, does not have such an option.

Similarly, the formulated transition conditions (Fig. 12) could be improved. For example, experiment c_2 (Fig. 22 top) executes a transition from β_3 to β_4 as the defenders have occupied the "TOP" space of the field. That transition may have been done prematurely, as the passing line between robot 5 and robot 2 remains undefended.

Observation 5 A good strategy design depends on how capable the strategist is in predicting the defender's behavior, and designing transition conditions that properly capture their behavior.

5.2 Real World Results

Figure 25 shows the real-world experiment in which a slightly modified attack A (Fig. 15) was applied[8] against Robotsports[9]. The strategy was flipped 180 degrees as the attacking

[8] supplementary material video_1

[9] https://www.robotsports.nl/index.php/nl/

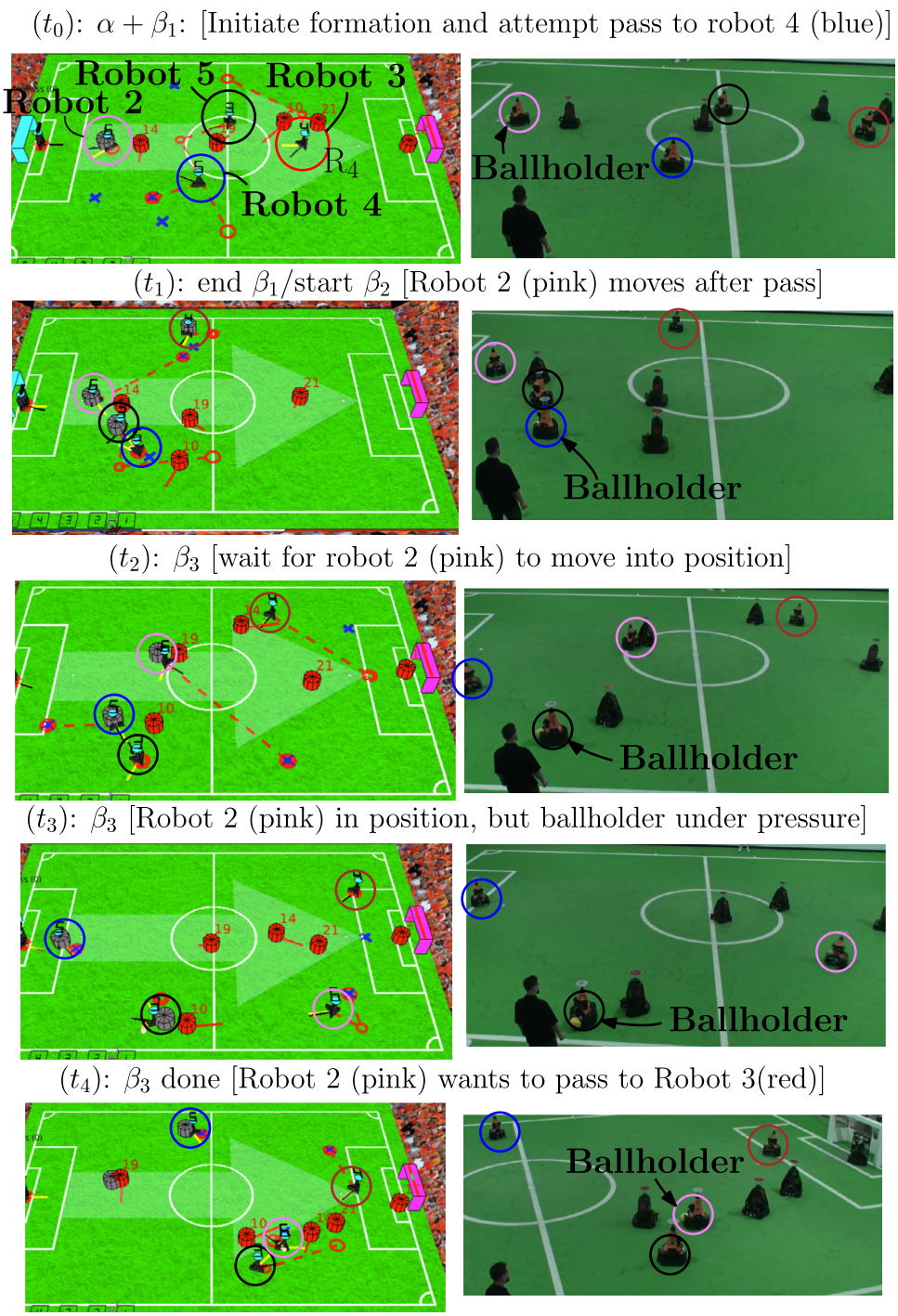


Fig. 25 Internal world model of the robots on the left, and the real soccer field on the right. In contrast to visualizations before, the attacking team Tech United Eindhoven is on the left side and the defending team Robotsports on the right side. A 180 degrees flipped version of attack A in Fig. 15 is applied. In the world model, the black triangles are the attacking team and the red cylinders the defending team. The intended trajectory of the robots are visualized with the dashed red lines with circles along this line representing waypoints. The role of each robot is clarified on both sides by encircling them with a color corresponding to the roles

team starts on the left side. In addition, some changes were made to the assignment of zones as it involved a previous iteration of the designed strategy of Fig. 11. Appendix D.3 summarizes the changes, but the overall intentions behind the strategy remain the same: taking advantage of the "TOP" space.

Robot 2 (pink) has the ball at t_0. State α is initiated to move the players into formation and after that execute the suggested pass in β_1 to robot 4 (blue). This pass is successful at t_1, but robot 5 (black) is out of position due to the defender blocking its path which causes a delay of β_4. This is visible at t_0 where an angle is seen in the path (red dashed line) connected to robot 5 due to the red cylinder defender moving left towards robot 2. Robot 5's path planner keeps adjusting the trajectory accordingly to prevent collisions, resulting in robot 5 being pulled with the defense at t_1.

That delay would be less troublesome if robot 2 had reached its intended position at t_2 at the moment robot 5

got the ball. Instead, robot 5 needs to wait for robot 2, which collides with a defender around the mid-field. Robot 2 finally reaches its position at t_2, but cannot immediately receive a pass due to pressure on robot 5. Robot 2 receives a delayed pass at t_4 which completes attack A, but at this moment the defense surrounded robot 2. The reactive algorithm cannot take advantage of the favorable position of robot 3 near the goal. Instead, several passes are made with eventually robot 3 scoring a goal. Notice that by not constraining robot 4 to Z18, as instructed in Fig. 11, it was able to complement the formation well by occupying Z16 at t_4.

Observation 6 It may not be necessary to constrain all robots strictly to a single zone. Instead, it may be beneficial to allow some robots (outside of the active play) to position with less zone constraints.

5.2.1 Reflection on the Real World Results

In the simulated experiments of Sec. 5.1, α was implicitly satisfied as the attackers already started in the correct formation. In the real-world experiment, we allow the other non-ball-holding robots to start anywhere on the field, e.g., with robot 3 starting on the defender's half of the field. This causes a delay in executing β_1 and β_2, which results in the defense to move towards the attacker's half of the field at t_1. This is not undesired, as the defender's half of the field remains undefended. The problem, however, is that at the moment β_3 starts at t_2, robot 2 is still on its own half of the field. Compare this to the situation at the top of Fig. 18, where robot 2 has well progressed into the defender's half of the field. The disruption of robot 2's movement causes the defense to 1) pressure the ball holder robot 5, and 2) reorganize itself to cover robot 2 as seen in t_4. Observation 2 fits well: the initial "slowness" of the attack caused defense mispositioning. The attack, however, failed to capitalize on that mispositioning fast enough. That failure is mainly related to the low-level motion and passing skills which are not sufficiently optimized for the quick nature of our strategy. Optimizing those low-level skills is beyond the scope of this manuscript.

6 Conclusion and Discussion

6.1 Optimal Strategy Design

In this manuscript, a robot soccer strategy is designed with human tactical principles. The applicability of that strategy is evaluated in Sec. 5 for varying (dynamic) conditions. A comprehensive sensitivity analysis can help in improving that strategy, but it remains unclear 1) what quantifiable performance criteria should be measured, and 2) what quantifiable input parameters needs to be adjusted. Therefore, this comprehensive sensitivity analysis is left for future work. An intuitive performance criterion might be the number of goals scored, but that does not provide direct insight into the quality of the strategy [5].

According to the authors of [3], the responsibility of a strategy is to create "favorable situations". Quantifying performance as the creation of "favorable situations" remains challenging, but we understand it as the attacking team actively enforcing/capitalizing on the defending team's mispositioning. Choosing proper sensitivity input parameters will likely require an analysis of inter-team relations. For example, varying the timing of state transitions (Sec. 5.1.2) might be best captured with some (dynamic) threshold distance value between the attacking and defending team, instead of a static time-delay parameter. The input parameter can also be strategy specific; the analysis of mid-field pressure (Sec. 5.1.1) is specific to our designed coordination patterns. Additionally, intra-team relations may also require investigation, e.g., by varying the probability of successful passes or movement speed of players. That can provide feedback for teams that cannot reliably execute low-level skills.

6.2 Intentions Behind Coordination Patterns

In our opinion, the main benefit to formalizing TPs is that it provides better insight into *how* and *why* some action or event occurred. That can lead to the design of evaluation criteria that can better estimate situations to initiate, abort, or divert the strategy. For example, the choice to move to some location (x,y) may align better with the rationale "to capitalize on undefended space", rather than a statistical rationale of "70 percent of the time a goal is scored from this position".

We think the insights of this manuscript are therefore complementary to decision making procedures in literature, e.g., decision making in adversarial games [46], as it offers an alternative perspective on parameter choices. These parameters can also complement strategy generation procedures as in [8].

TP formalization can also benefit decentralized coordination, a complex process where information (entropy) is exchanged intra- and inter-team [39]; a balance needs to be sought between making your future action apparent to team members and ambiguous to the opposition. Players aware of TPs may then better estimate each other's intentions. Reactive approaches (Sec. 2.1) are further disadvantaged in this case: they may increase ambiguity to the opposition, but at the cost of also increasing ambiguity to your team members.

6.3 Motion Control Constraints

The strategy can be imagined as a sketch of coordination patterns, where its realization depends on the player's configuration at runtime. This manuscript realizes that sketch by confining the output of a motion planner to some predefined zone sequence on the field. The continuous variations on the

field are then tackled by the motion planner, whereas discrete changes are handled by the strategy, e.g., observing an undefended space on the field. That approach allows the attacking team to consistently execute the strategy, even under varying conditions on the field (Sec. 5.1).

Future research needs to investigate to what extent strategy should constrain the motion planner. That can be role-dependent, as demonstrated by the real-world experiment in Sec. 5.2, which showed that not all players need to be constrained at all times. Having some players constrained less, allowed the team to still benefit from ineffectively executed strategies.

Another improvement is on adding constraints on the planned motion trajectory. The experimental results in Sec. 5.1 suggest that path planning influences strategy execution. For example, a path can be planned to block opponents, reducing pressure on another team member, or a path can be planned that minimizes contact with defenders, increasing the player's own speed.

Supplementary information

Appendices are present to support the main text. Additionally videos are present that capture the experimental results.

Appendix A: Transition Conditions

This appendix elaborates on the abstract transition conditions in Table 2: `FormationRealized`, `BallPossessionMaintained`, `OpportunityAvailable`, `ResetAvailable`, and `GoalShotPossible`. Suggestions are provided to realize the above-mentioned conditions, but they are not definitive nor exhaustive; Appendix B.2 shows their usage.

A.1 Positioning Conditions

`FormationRealized` can be realized by using (a composition of) the queries in Fig. 26. The moment until `OpportunityAvailable` is monitored – the blue highlighted area in Fig. 11 – can be similarly defined with one of the below queries, e.g., "until blue 2 passes left of Z15" described as `Left(blue 2, Z15)`. The following three queries are implemented in this manuscript:

- `ContainedIn(player,zone)`: if it is sufficient for players to be in a zone.
- `DistanceLess(player, target, value)`: if we require players to be positioned with respect to some (x,y) coordinate.
- `Right/Left/Above/Below(player, zone)`: the opportunity-seeking movement can be performed until this condition is satisfied, checking whether a player has passed the zone in a particular manner.

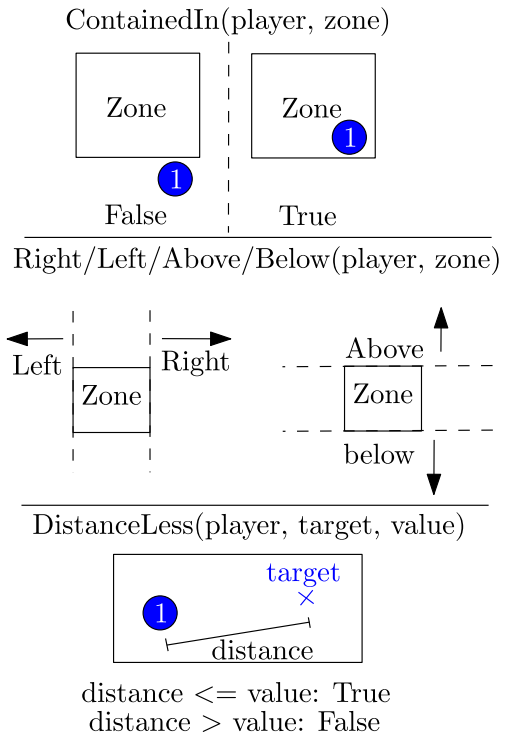


Fig. 26 Visualization of the positioning queries deployed in this manuscript. On top, the zone containment query. In middle, a zone orientation query. In bottom, a distance query

A.2 Opportunistic Conditions

The `OpportunityAvailable` and `ResetAvailable` conditions can be implemented by using (a composition of) the queries below. `ResetAvailable` is (mostly) realized as a negation of `OpportunityAvailable`, executing a resetting motion if none of the opportunities remain.

A.2.1 Defensive Lines

$TP_{o,1}$ can be monitored via the query `DefLine(threatening_player)`. Two outputs are provided:

- `NumberOfAttackers` $\in \mathbb{N}$: the number of team members behind the defensive line (Fig. 5) that is just passed by the `threatening_player`.
- `DifferenceToOpponent` $\in \mathbb{N}$: the difference between the number of attacking and defending players, excluding the goal keeper, in the area behind the defensive line.

The first output allows the execution of particular strategies requiring a certain number of attacking players. The second output concludes (local) numerical superiority if its value is greater than zero. Assuming the defensive

lines in Fig. 5 remain unchanged, then at the moment blue 3 passed "dl1", we conclude numerical inferiority: `DefLine(blue 3) = (NumberOfAttackers = 2, DifferenceToOpponent = -1)`. If blue 3 passed "dl2", we conclude conclude numerical superiority: `DefLine(blue 3) = (NumberOfAttackers = 2, DifferenceToOpponent = 1)`.

A.2.2 Open Spaces

$TP_{o,2}$–$TP_{o,4}$ can all be monitored by checking zone occupation on the soccer field. $TP_{o,4}$ focuses on the open spaces visualized in Fig. 6. We query the occupancy of one of the three spaces via `ContainOpponentsInSpace(SpaceType, amount, operator)`:

- `SpaceType`: is either "BOTTOM","MIDDLE", or "TOP" with some length parameter, e.g., `TOP(0.33×FLD_LENGTH)`.
- `amount`: the number of opponents.
- `operator`: can be `LessEqualThan` (≤), `Equal` (==), or `MoreEqualThan` (≥).

Numerical superiority in the "BOTTOM" space can for example be checked by querying `ContainOpponentsInSpace(BOTTOM(0.33×FLD_LENGTH),0,Equal) == True`, thus checking whether the "BOTTOM" space contain no defenders. One can design similar queries for $TP_{o,2}$ and $TP_{o,3}$ focused on particular zone occupation(s) in Fig. 10. For example, checking whether Z2/Z3 is occupied for $TP_{o,3}$.

A.3 Goal Attempt

`GoalShotPossible` can be evaluated via the query `GoalRemainFree(player)`. This is illustrated by the blue triangle in Fig. 27 and evaluates whether any opponent other than the goalkeeper is present in it. The `player` argument determines from which player's perspective the triangle is analyzed. A relatively simple monitor is chosen as an estimate of goal shot success, as the highly dynamic environment would make it difficult to model this goal shot opportunity accurately. Especially if a pass still needs to be given to the player that needs to shoot on goal.

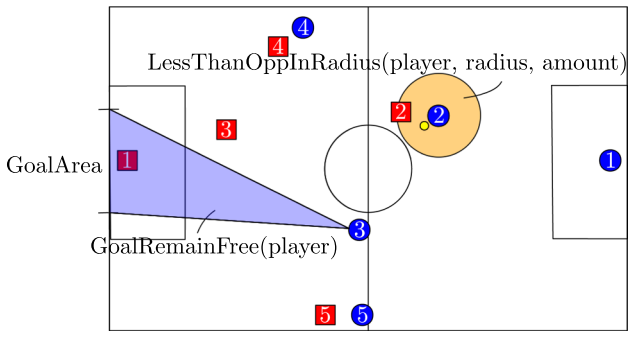


Fig. 27 Visualized are the GoalRemainFree() and LessThanOppInRadius() queries

A.4 Ball Retention Conditions

`BallPossessionMaintained` can be realized by using (a composition of) the queries below.

A.4.1 Pressure Ball Holder

The pressure on the ball holder is queried via `LessThanOppInRadius(player, radius, number)`, illustrated as the orange circle in Fig. 27.

- `player`: the ball holder.
- `radius`: the size of the circle.
- `number`: the number of opponents that would satisfy the *less than* relation.

A.4.2 Pressure Ball Receiver

To estimate the success of a future pass, the query `AllowedPressure(Zone, player, numberIn, numberBetween)` is evaluated:

- `Zone`: represents the green area in Fig. 28 to which the ball holder wants to pass in the future
- `player`: the ball holder that gives the pass.
- `numberIn`: the allowed number of opponents within the green area in Fig. 28.
- `numberBetween`: the allowed number of opponents in the yellow area in Fig. 28.

This query only gives a rough estimate of whether the pass will succeed, as the receiving robot is likely not on the

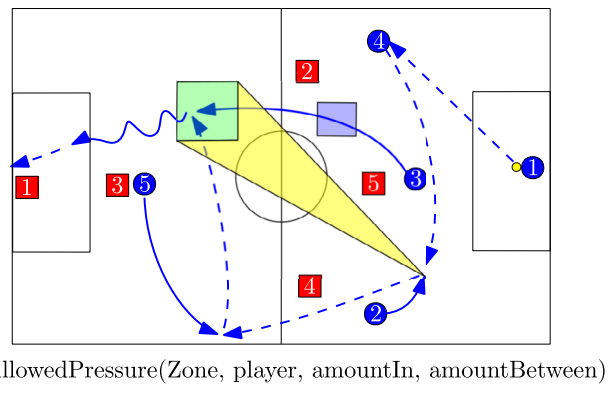


Fig. 28 The monitored spaces for checking the success of a pass. The blue area represents the end of the opportunity-seeking movement, the green area the intended passing area of the subsequent attack, and in yellow the area between the ball holder and the passing area

intended zone yet. For example, in Fig. 28 the decision to commit to the green zone needs to be made when blue 3 reaches the blue zone. When the robot actually reaches the green zone, we assume that the low-level passing module is sufficiently fast to calculate online the pass success probability. If the pass is deemed infeasible, the resetting states within the strategy can still recover from this undesired situation.

A.4.3 Specific Player has the Bal

Certain strategy transitions require that a specific player has the ball. That can be realized by the query `PlayerGotBall(player)` which returns either `True` or `False`. The argument `player` refers to the specific player that needs to have the ball.

Appendix B: Strategy Design Details

This appendix provides further detail on the strategy design. First, the mentioned TPs (Table 1) are formalized as Zone Constraints (ZCs) (Table 4) in Appendix B.1; these ZCs are rules to confine the motion planner target to specific zones in Fig. 10. We do not claim that these ZCs are the optimal representation of these TPs, but rather serve as an initial baseline for further refinement. Second, we apply Sec. 3.5's guideline on Fig. 11 in Appendix B.2; we formalize the used motion constraints (ZCs) and transition conditions (Appendix A).

B.1 Tactical Principles as Zone Constraints

We provide extra explanation of the ZCs in Table 4, according to the TP category.

B.1.1 Mobility Constraints

The maximum number of players of $\mathbf{ZC}_0$ and $\mathbf{ZC}_1$ is currently determined by the strategy in Fig. 10. Examples of "equal width value" zones are Z3 and Z7, while "equal length value" zones are Z3 and Z2. The currently formulated $\mathbf{ZC}_0$ and $\mathbf{ZC}_1$ allow for both the 1-3-1 and 1-2-2 formations to exist.

B.1.2 Opportunity Constraints

For $\mathbf{ZC}_2$, the chosen zone is based on the (expected) opposition configuration at the start of the opportunity-seeking movement. For example, in Fig. 5 blue 3 could have chosen any of the zones Z1–Z12 to seek numerical superiority past "dl2". That static zone selection is used as a baseline, as the opponent's behavior cannot be reliably predicted offline. In addition, resetting mechanisms (Sec. B.1.4) exist to recover from failed opportunity-seeking movements.

The lengthwise restriction of $\mathbf{ZC}_3$ is to ensure attack progression. The restriction on the width is to ensure sufficient ball displacement along the width of the field; to let the opposition expend more effort in disrupting the attack. If the ball holder is present at Z13, then the non-ball-holding player can be positioned anywhere on Z3, Z4, Z7, Z8, Z11, Z12, Z15, and Z16.

For $\mathbf{ZC}_4$, zones Z2 and Z3 approximate the RoboCup MSL penalty area the best.

For $\mathbf{ZC}_5$, one of the three spaces in Fig. 6 must remain (sufficiently) unoccupied by the defense. A zone should be chosen that intersects with that unoccupied space. Additionally, that zone cannot be lengthwise behind the ball holder

Table 4 The left column represents Zone Constraints (ZCs), the middle column the descriptions of the ZCs, and the right column the TP (Table 1) from which the ZCs are derived

Zone Constraints (ZC)	Description	TPs
(ZC_0) Spread width	At most two players have a target zone of equal width value.	$TP_{m,2}$
(ZC_1) Spread length	At most three players have a target zone of equal length value.	$TP_{m,2}$
(ZC_2) Threaten defensive line	Move towards a zone that is behind a defensive line that would result in numerical superiority given the current configuration.	$TP_{o,1}$
(ZC_3) Switch play	A zone is assigned that cannot be lengthwise behind the ball holder's zone and needs to be at least one zone width apart.	$TP_{o,2}$
(ZC_4) Go penalty area	The supporting player moves to either zones Z2 or Z3.	$TP_{o,3}$
(ZC_5) Occupy unmarked space	A zone is assigned that cannot be lengthwise behind the ball holder's zone and intersects with an unmarked space of Fig. 6.	$TP_{o,4}$
(ZC_6) Offensive support	At least one non-ball-holding player has a zone assigned that is not lengthwise behind the ball holder's zone.	$TP_{d,1}$
(ZC_7) Reset formation	Each player's target zone is valid as long as the (dynamic) formation can be recovered by having each player at most move 2 zones.	$TP_{r,1}$
(ZC_8) Time to reset formation	The zone at which the opportunity-seeking movement is aborted or continued, is halfway the trajectory that would have been realized if the opportunity remains.	$TP_{r,1}$
(ZC_9) Safe pass option	At least one non-ball-holding player has a zone assigned that is not lengthwise in front of the ball holder's zone.	$TP_{r,2}$

to ensure attack progression. For example, a strategy that wants to threaten the "TOP" space with the ball holder on Z8 requires a supporting player to move to one of the following zones: Z1, Z2, Z5, or Z6.

B.1.3 Deception Constraints

$\mathbf{ZC}_6$ will likely be satisfied via either ZC_2-ZC_5. This constraint is added to ensure that not all non-ball-holding players are lengthwise behind the ball holder. Otherwise, a similar situation can be realized as in the middle of Fig. 3, where the opposition can immediately cover the ball holder.

B.1.4 Reserve Constraints

For $\mathbf{ZC}_7$, the concept of a *dynamic formation* is inspired by [20]. They suggest that during the reset of an attack, it is not always necessary to let the players move back to their initial position. If the ball has sufficiently progressed, then one could "move" the initial formation forward. For example, in Fig. 8 the players could reset the formation by occupying Z3, Z13, Z15, and Z16. The choice of 2 zone displacement is chosen to allow player spread. Diagonal movement is allowed, e.g., Z8 to Z11 would count as 1 zone displacement.

$\mathbf{ZC}_8$ essentially formalizes the zone at the end of movement B1 in Fig. 7. If there is no evidence at the end of B1 that some opportunity remains, then the resetting movement B2 is initiated. The zone at the end of B1 is set as the halfway point of the hypothetical trajectory in case the opportunity-seeking movement would have succeeded. For example, if the player starts in Z19 and the target zone of the opportunity-seeking movement is Z6, then we allow that movement to be performed until the robot is past, e.g., Z14/Z15.

$\mathbf{ZC}_9$ is the opposite of ZC_6 and focuses on creating backup options.

B.2 Formalized Strategy

B.2.1 β_1, β_7, and β_8

We elaborate on β_1, with most of the explanation also applying to β_7 and β_8 as they are similar.

- **SG1** satisfied: blue 2 performs a pass to blue 4, after which blue 2 and blue 3 move ($\mathbf{TP}_{m,1}$).
- **SG2** satisfied: we evaluate the end-configuration of the state as we want to evaluate the result of our action patterns, i.e., Z13 (blue 5), Z15 (blue 2), Z18 (blue 4), and Z20 (blue 3). No zones are considered equal width ($\mathbf{ZC}_0$). At most two zones are considered equal length ($\mathbf{ZC}_1$); blue 2 and blue 5, or blue 3 and blue 4.
- **SG3** not satisfied: no direct threat exists ($\mathbf{ZC}_2$-$\mathbf{ZC}_5$) as blue 2 only moves slightly forward.
- **SG4** satisfied: when blue 2 has the ball, blue 3 and blue 5 provide offensive support ($\mathbf{ZC}_6$). When blue 4 has the ball, blue 2 and blue 5 provide offensive support.
- **SG5** satisfied: the resetting motion (β_7) connected to β_1 focuses on recovering the default formation of α. Evaluation is done at the end-configuration of the state as we want to evaluate the result of our action patterns. Blue 2 and 3 recover the formation by displacing one zone ($\mathbf{ZC}_7$). Blue 2 aborts its opportunity-seeking movement roughly halfway distance if it would had directly commited to its end-target in β_2 ($\mathbf{ZC}_8$)
- **SG6** satisfied: When blue 2 has the ball, blue 4 remains as safe passing option ($\mathbf{ZC}_9$). When blue 4 has the ball, blue 3 remains as safe passing option.

A transition to β_2 (see Fig. 12) requires satisfaction of the four conditions below:

- `ContainOpponentsInSpace(TOP(0.5× FLD_LNGTH), 1, LessEqualThan == True.`
- `Left(blue2,Z15) == False`; to maintain speed in the opportunity seeking robot (blue 2) for the subsequent state, we initially command it to move past the blue-highlighted zone (Z15) (Fig. 12). This is also done for other states involving such a blue-highlighted zone.
- `AllowedPressure(Z9, blue4, 1, 1) == True.`
- `PlayerGotBall(blue4) == True.`

A transition to β_7 requires satisfaction of the four conditions below:

- `ContainOpponentsInSpace(TOP(0.5× FLD_LNGTH), 1, LessEqualThan == False.`
- `Left(blue2,Z15) == True.`
- `AllowedPressure(Z19, blue4, 1,1)==True.`
- `PlayerGotBall(blue4) == True.`

B.2.2 β_2

- **SG1** satisfied: blue 4 performs a pass to blue 5, while simultaneously blue 2, blue 3, and blue 5 move ($\mathbf{TP}_{m,1}$). It is not necessary to sequence a "pass" and "move" action (as with β_1) when the initial ball holder (blue 4) does not move.
- **SG2** satisfied: we evaluate the end-configuration of the state, i.e., Z9 (blue 5), Z10 (blue 2), Z18 (blue 4), and Z12 (blue 3). At most two zones are considered equal width ($\mathbf{ZC}_0$); blue 2 and blue 4. At most three zones are considered equal length ($\mathbf{ZC}_1$); blue 2, blue 3, and blue 5.
- **SG3** satisfied: blue 2 moves to Z10 to occupy the "TOP" space ($\mathbf{ZC}_5$), while blue 3 moves to Z12 for a switch play ($\mathbf{ZC}_3$).

- **SG4** satisfied: when blue 4 has the ball, blue 2 and blue 5 provide offensive support ($\mathbf{ZC}_6$). When blue 5 has the ball, blue 2 and blue 3 provide offensive support.
- **SG5** satisfied: the resetting motion (β_6) connected to β_1 focuses on recovering the default formation of α. Evaluation is done at the end-configuration of the state. Blue 2, blue 3, and blue 5 recover the formation by backtracking ($\mathbf{ZC}_7$). Z10 for blue 2 can serve as the final target for a "TOP" space opportunity-seeking movement, but could also serve as roughly the half-way point for another opportunity-seeking movement; if the strategy continues in β_3, then Z10 could be imagined as the half-way point of the trajectory Z15–Z1 ($\mathbf{ZC}_8$).
- **SG6** satisfied: When blue 4 has the ball, blue 3 remains as safe passing option ($\mathbf{ZC}_9$). When blue 5 has the ball, blue 4 remains as safe passing option.

Transition to β_3 requires satisfaction of the four conditions below:

- `ContainOpponentsInSpace(TOP(0.5×FLD_LNGTH), 1, LessEqualThan == True.`
- `Left(blue2, Z10) == False.`
- `AllowedPressure(Z1, blue5, 1, 1)==True.`
- `PlayerGotBall(blue5) == True.`

Transition to β_4 requires satisfaction of the five condition below:

- `ContainOpponentsInSpace(TOP(0.5×FLD_LNGTH), 1, LessEqualThan == False.`
- `ContainOpponentsInSpace(BOTTOM(0.5×FLD_LNGTH), 1, LessEqualThan == True.`
- `Left(blue2, Z10) == False.`
- `AllowedPressure(Z12, blue5, 1, 1)==True.`
- `PlayerGotBall(blue5) == True.`

A transition to β_6 requires satisfaction of the four conditions below:

- `ContainOpponentsInSpace(TOP(0.5×FLD_LNGTH), 1, LessEqualThan == False.`
- `ContainOpponentsInSpace(BOTTOM(0.5×FLD_LNGTH), 1, LessEqualThan == False.`
- `Left(blue2, Z10) == True.`
- `PlayerGotBall(blue5) == True.`

B.2.3 β_3

- **SG1** satisfied: blue 5 performs a pass to blue 2, while simultaneously blue 2 and blue 3 move ($\mathbf{TP}_{m,1}$).
- **SG2** satisfied: we evaluate the end-configuration of the state, i.e., Z9 (blue 5), Z1 (blue 2), Z18 (blue 4), and Z3 (blue 3). At most two zones are considered equal width ($\mathbf{ZC}_0$); blue 2 and blue 5. At most two zones are considered equal length ($\mathbf{ZC}_1$); blue 2 and blue 3.
- **SG3** satisfied: blue 2 moves to Z1 to occupy the "TOP" space ($\mathbf{ZC}_5$), while blue 3 moves to Z3 to threaten the penalty area ($\mathbf{ZC}_4$).
- **SG4** satisfied: when blue 5 has the ball, blue 2 and blue 3 provide offensive support ($\mathbf{ZC}_6$). When blue 2 has the ball, blue 3 provides offensive support.
- **SG5** partially satisfied: no resetting motion is defined, but we move the default formation of α forward as three players penetrated the opponent's half of the field. The yellow formation (Fig. 11) can be recovered within 2 zone displacements by blue 2 moving to Z9, blue 5 moving to Z14, blue 4 moving to Z15, and blue 3 moving to Z12 ($\mathbf{ZC}_7$). $\mathbf{ZC}_8$ is not satisfied as no sequential β is defined.
- **SG6** satisfied: when blue 5 has the ball, blue 4 remains as safe passing option ($\mathbf{ZC}_9$). When blue 2 has the ball, blue 4 and blue 5 remains as safe passing option.

Transition to β_4 requires satisfaction of the two conditions below:

- `PlayerGotBall(blue5) == True.`
- `ContainOpponentsInSpace(TOP(0.5×FLD_LNGTH), 2, MoreEqualThan == True.`

B.2.4 β_4

- **SG1** satisfied: blue 5 performs a pass to blue 3, while simultaneously blue 2 moves ($\mathbf{TP}_{m,1}$).
- **SG2** satisfied: we evaluate the end-configuration of the state, i.e., Z5 (blue 5), Z6 (blue 2), Z18 (blue 4), and Z12 (blue 3). At most two zones are considered equal width ($\mathbf{ZC}_0$); blue 2 and blue 4. At most two zones are considered equal length ($\mathbf{ZC}_1$); blue 2 and blue 5.
- **SG3** satisfied: blue 2 moves to Z6 and blue 5 moves to Z5 to occupy the "TOP" space ($\mathbf{ZC}_5$).
- **SG4** satisfied: when blue 5 has the ball, blue 2 and blue 3 provide offensive support ($\mathbf{ZC}_6$). When blue 3 has the ball, blue 2 and blue 5 provide offensive support.
- **SG5** satisfied: a similar resetting formation is considered as β_3. The yellow formation can be recovered within 2 zone displacements by blue 2 moving to Z14, blue 5 moving to Z9, blue 4 moving to Z15, and blue 3 moving to Z12 ($\mathbf{ZC}_7$). Z6 for blue 2 can serve as the half-way point for another opportunity-seeking movement; if the strategy continues in β_5, then Z6 could be imagined as the half-way point of the trajectory Z10–Z7 ($\mathbf{ZC}_8$).

- **SG6** satisfied: when blue 5 has the ball, blue 4 remains as safe passing option ($\mathbf{ZC}_9$). When blue 3 has the ball, blue 4 remains as safe passing option.

Transition to β_5 requires satisfaction of the four conditions below:

- `ContainOpponentsInSpace(BOTTOM(0.5× FLD_LNGTH), 1, LessEqualThan == True`.
- `Below(blue2, Z6)==False`.
- `AllowedPressure(Z7, blue3, 1, 1)==True`.
- `PlayerGotBall(blue3) == True`.

B.2.5 β_5

- **SG1** satisfied: blue 3 performs a pass to blue 2, while simultaneously blue 2 moves ($\mathbf{TP}_{m,1}$). Blue 5 delays its movement to the penalty area to make it non-apparent.
- **SG2** satisfied: we evaluate the end-configuration of the state, i.e., Z2 (blue 5), Z7 (blue 2), Z18 (blue 4), and Z12 (blue 3). At most two zones are considered equal width ($\mathbf{ZC}_0$); blue 4 and blue 5. No zones are considered equal length ($\mathbf{ZC}_1$).
- **SG3** satisfied: blue 2 moves to Z7 to occupy the "BOTTOM" space ($\mathbf{ZC}_5$). Blue 5 moves to Z2 to attack the penalty area from the opposite side ($\mathbf{ZC}_3$/$\mathbf{ZC}_4$).
- **SG4** satisfied: When blue 3 has the ball, blue 2 and blue 5 provide offensive support ($\mathbf{ZC}_6$). When blue 2 has the ball, blue 5 provides offensive support.
- **SG5** satisfied: a similar resetting formation is considered as β_3. The yellow formation can be recovered within 2 zone displacements by blue 2 moving to Z12, blue 5 moving to Z9, blue 4 moving to Z14, and blue 3 moving to Z15 ($\mathbf{ZC}_7$). Z7 for blue 2 can serve as the half-way point for another opportunity-seeking movement; if the strategy continues in γ_1, then Z7 could be imagined as the half-way point of the trajectory Z6–Z7–Z3 ($\mathbf{ZC}_8$).
- **SG6** satisfied: when blue 3 has the ball, blue 4 remains as safe passing option ($\mathbf{ZC}_9$). When blue 2 has the ball, blue 3 and blue 4 remain as safe passing option.

Transition to γ_1 requires satisfaction of the two conditions below:

- `GoalRemainFree(blue2) == True`.
- `PlayerGotBall(blue2) == True`.

B.2.6 γ_1

- **SG1** satisfied: blue 2 moves into the penalty area before shooting ($\mathbf{TP}_{m,1}$).
- **SG2** satisfied: we evaluate the end-configuration of the state, i.e., Z2 (blue 5), Z3 (blue 2), Z18 (blue 4), and Z12 (blue 3). At most two zones are considered equal width ($\mathbf{ZC}_0$); blue 4 and blue 5. At most two zones are considered equal length ($\mathbf{ZC}_1$); blue 2 and blue 5.
- **SG3** satisfied: the penalty area is threatened ($\mathbf{ZC}_4$).
- **SG4** satisfied: when blue 2 has the ball, blue 5 provides offensive support ($\mathbf{ZC}_6$).
- **SG5** partially satisfied: a similar resetting formation is considered as β_3. The yellow formation can be recovered within 2 zone displacements by blue 2 moving to Z12, blue 5 moving to Z9, blue 4 moving to Z14, and blue 3 moving to Z15 ($\mathbf{ZC}_7$). $\mathbf{ZC}_8$ is not satisfied as no sequential β is defined.
- **SG6** satisfied: when blue 2 has the ball, blue 3 and blue 4 remain as safe passing option ($\mathbf{ZC}_9$).

B.2.7 β_6

- **SG1** satisfied: blue 5 performs a pass to blue 4, and after the pass blue 2, blue 3, and blue 5 move ($\mathbf{TP}_{m,1}$).
- **SG2** satisfied: we evaluate the end-configuration of the state, i.e., Z17 (blue 5), Z14 (blue 2), Z19 (blue 4), and Z16 (blue 3). No two zones are considered equal width ($\mathbf{ZC}_0$). At most two zones are considered equal length ($\mathbf{ZC}_1$); blue 2 and blue 3, or blue 4 and blue 5.
- **SG3** not satisfied: during a resetting motion, it is not necessary to seek an opportunity.
- **SG4** satisfied: when blue 5 has the ball, blue 2 and blue 3 provide offensive support ($\mathbf{ZC}_6$). When blue 4 has the ball, blue 2 and blue 3 provide offensive support.
- **SG5** satisfied: the resetting motion (β_8) connected to β_6 focuses on recovering the default formation of α. Evaluation is done at the end-configuration of the state. Blue 2 and blue 5 recover the formation by moving one zone ($\mathbf{ZC}_7$). The resetting motion is aborted by blue 2 at roughly halfway the distance of the zone sequence Z10–Z14–Z13 ($\mathbf{ZC}_8$; the trajectory to go from β_6 to β_8 unhindered. Note that blue 2 switches role to blue 4 at β_8.
- **SG6** satisfied: when blue 5 has the ball, blue 4 remains as safe passing option ($\mathbf{ZC}_9$). When blue 4 has the ball, blue 5 remains as safe passing option.

Transition to β_8 requires satisfaction of the two conditions below:

- `AllowedPressure(Z18, blue4, 1,1)==True`.
- `PlayerGotBall(blue4) == True`.

Appendix C: STP Framework Implementation

The STP framework [24] contains three important aspects:

- Skills: "*encode low-level single-robot control algorithms for executing a complex behavior to achieve a short-term focused objective*" [24].
- Tactics: "*encapsulate what the robot should do, in terms of executing skills, to achieve a specific long-term goal*" [24].
- Play: "*encode how the team of robots should coordinate their execution of tactics in order to achieve the team's overall goals*" [24].

This manuscript focuses on the "Tactics" and "Play" aspect, where we re-use "Skills" of the Tech United Eindhoven software framework. For example, the communication between the pass giver and receiver is done according to the pass handler[10] and details about the continuous motion control can be found in the motion folder[11].

The "pass and move" principle ($\mathbf{TP}_{m,1}$) forms the basis for the design of our "Plays" and "Tactics". Three tactics are designed:

- T_{BP}: the ball holder needs to maintain ball possession and provide a pass if desired.
- T_{MZO}: the non-ball holder needs to move to a designated zone for offensive support.
- T_{MZB}: the non-ball holder needs to move to a designated zone to receive a pass.

T_{BP} is based on the tactic `advanced_attack()`[12] and $\mathrm{T}_{\mathrm{MZO}}$ and $\mathrm{T}_{\mathrm{MZB}}$ on `task_position_mu()`[13]. Our main modifications are in deducing the path planner targets within $\mathrm{T}_{\mathrm{MZO}}$ and $\mathrm{T}_{\mathrm{MZB}}$.

The current path planner target is chosen as a grid cell (x, y) on a pre-segmented soccer field. The optimal grid cell is obtained by optimizing the utility function

$$u(x, y) = \sum_{i=0}^{k} w_i \cdot f_i(x, y), \tag{1}$$

with $w_i \in \mathbb{R}$ a scalar weight value, $k \in \mathbb{N}$ the number of objective functions, and $f(x, y) \in \mathbb{R}$ an objective function providing a value to the grid cell (x, y). An example of an $f(x, y)$ is "to position closer to the opposition's goal". We re-use the functions described in `mu_field_attacker_adv()` and `mu_field_walk-free()` in the `mu_positioning_lib` folder[14], with 1) slight modifications in w_i and/or $f_i(x, y)$, and 2) a confinement of the solution (x, y) to one (or multiple) zone(s) in Fig. 10.

The "Play" is formulated by 1) defining applicability conditions, 2) assigning each robot a (sequence of) "Tactic(s)", and 3) defining "Tactic" completion conditions that lead the players to collectively transition to the next "Tactic"/"Play". According to the "pass and move" structure, we define a "Play" as two "Tactics"; one before the pass is given and one after the pass is given. Below, an example is seen of β_2 (details in Appendix B.2) formulated in a "Play" structure. Note that we elongate the executed trajectory of blue 2 past Z5 to Z12 to maintain speed in subsequent state transitions; otherwise blue 2 may slow down when it reaches Z5.

- **Applicability conditions**:
  - `ContainOpponentsInSpace(TOP(0.5× FLD_LNGTH), 1, LessEqualThan == True.`
  - `Left(blue2,Z15) == False.`
  - `AllowedPressure(Z9, blue4, 1, 1) == True.`
  - `PlayerGotBall(blue4) == True.`
- **Tactics blue 2**
  - Tactic 1: execute T_{MZO} to Z5.
  - Tactic 2: execute T_{MZO} to Z5.
- **Tactics blue 3**
  - Tactic 1: execute T_{MZO} to Z12.
  - Tactic 2: execute T_{MZO} to Z12.
- **Tactics blue 4**
  - Tactic 1: execute T_{BP} and give pass to blue 5.
  - Tactic 2: execute T_{MZO} to Z18.
- **Tactics blue 5**
  - Tactic 1: execute T_{MZB} to Z9.
  - Tactic 2: execute T_{BP} to maintain ball.
- **Tactic 1 completed**: after pass is given by blue 4.
- **Tactic 2 completed**: 1) the ball is received by blue 5, and 2) if subsequent state transitions exist to check applicable transitions.

[10] https://gitlab.tue.nl/tech-united-eindhoven/Turtle3/-/tree/master/Strategy/src/pass_handler

[11] https://gitlab.tue.nl/tech-united-eindhoven/Turtle3/-/tree/master/Motion

[12] https://gitlab.tue.nl/tech-united-eindhoven/Turtle3/-/blob/master/Strategy/src/STP/tasks/tasks_gameplay.c

[13] https://gitlab.tue.nl/tech-united-eindhoven/Turtle3/-/blob/master/Strategy/src/STP/tasks/tasks_positioning.c

[14] https://gitlab.tue.nl/tech-united-eindhoven/Turtle3/-/tree/master/Strategy/src/mu_positioning_lib

Appendix D: Experiments

D.1 Distributed Multi-robot System Errors

Experiments c_3, m_2, and f_4 failed by robot 2 passing too hard. Figure 29 illustrates situations just before a pass given. Most likely, a miscomunication occurs between the ball holder and receiver.

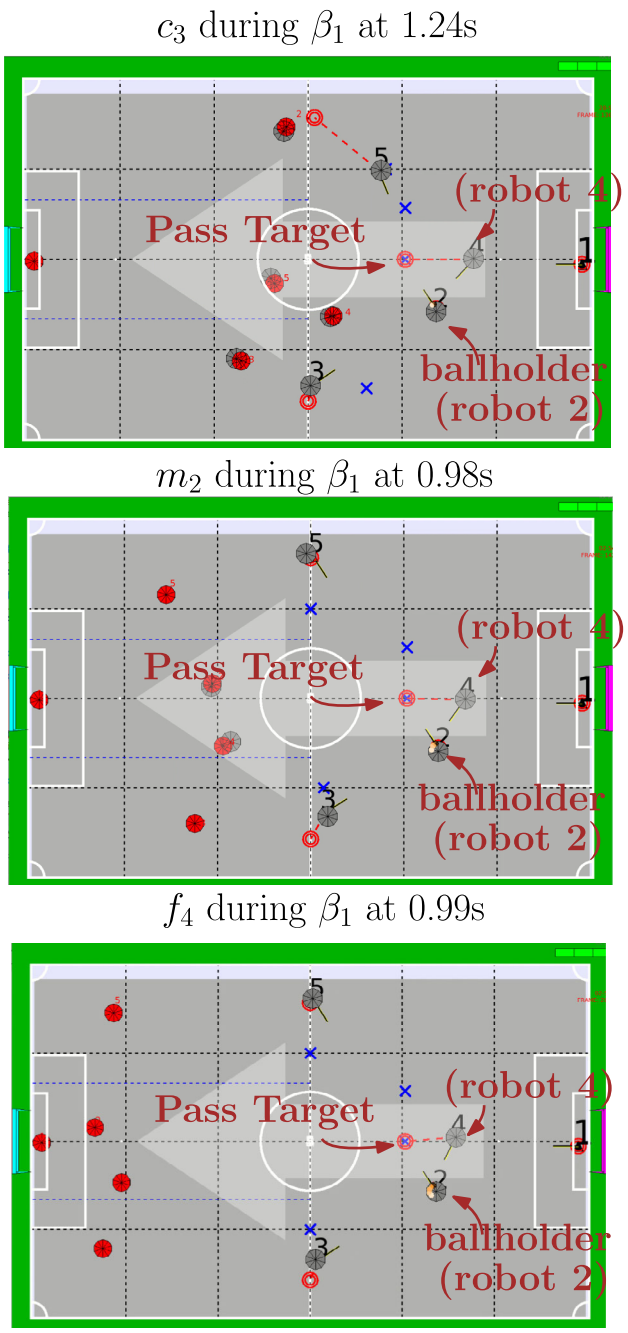


Fig. 29 Visualization of three simulated soccer fields. Experiments c_3 (top), m_2 (middle), and f_4 (bottom) are visualized during β_1 just before a pass is initiated from robot 2 to robot 4 to the pass target

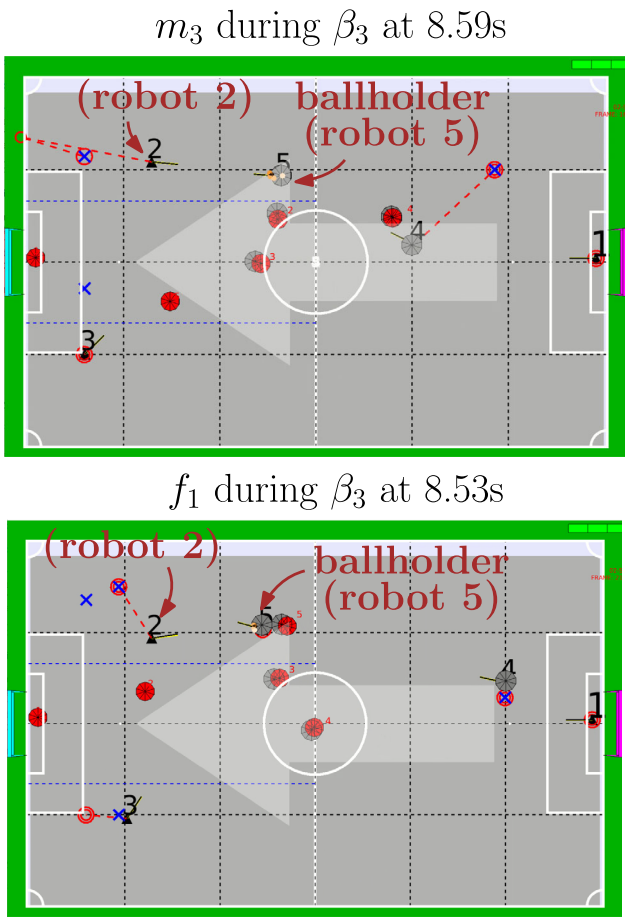


Fig. 30 Visualization of two simulated soccer fields. Experiments m_3 (top) and f_1 (bottom) are visualized during β_3 just before a pass is initiated from robot 4 to robot 2

Experiments m_3 and f_1 failed at the last pass in β_3 due to either problems in robot 5's aiming algorithm or a pass target miscommunication. Figure 30 illustrates situations just

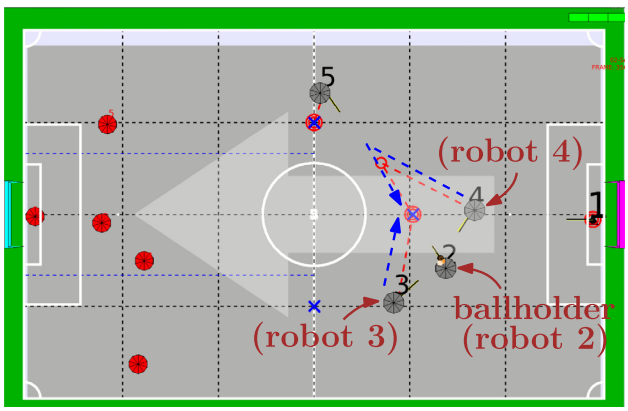


Fig. 31 Visualization of one simulated soccer field. Experiment f_7 is visualized during β_1 just before a pass is initiated from robot 2 to robot 4

before a pass is given. In both cases, the ball just misses robot 2 slightly below the pass target (blue cross).

Experiment f_7 failed due to an erroneous role assignment. Figure 31 illustrates robot 3 wrongly thinking it has the role of robot 4 by moving to the same target. A similar phenomenon is observed in experiment m_8.

D.2 Experimental Results Trajectories

Figure 32 illustrates the attacker trajectories, and Fig. 33 illustrates the defender trajectories. The illustrations follow the design of Fig. 15, with the defender trajectories visualized in red.

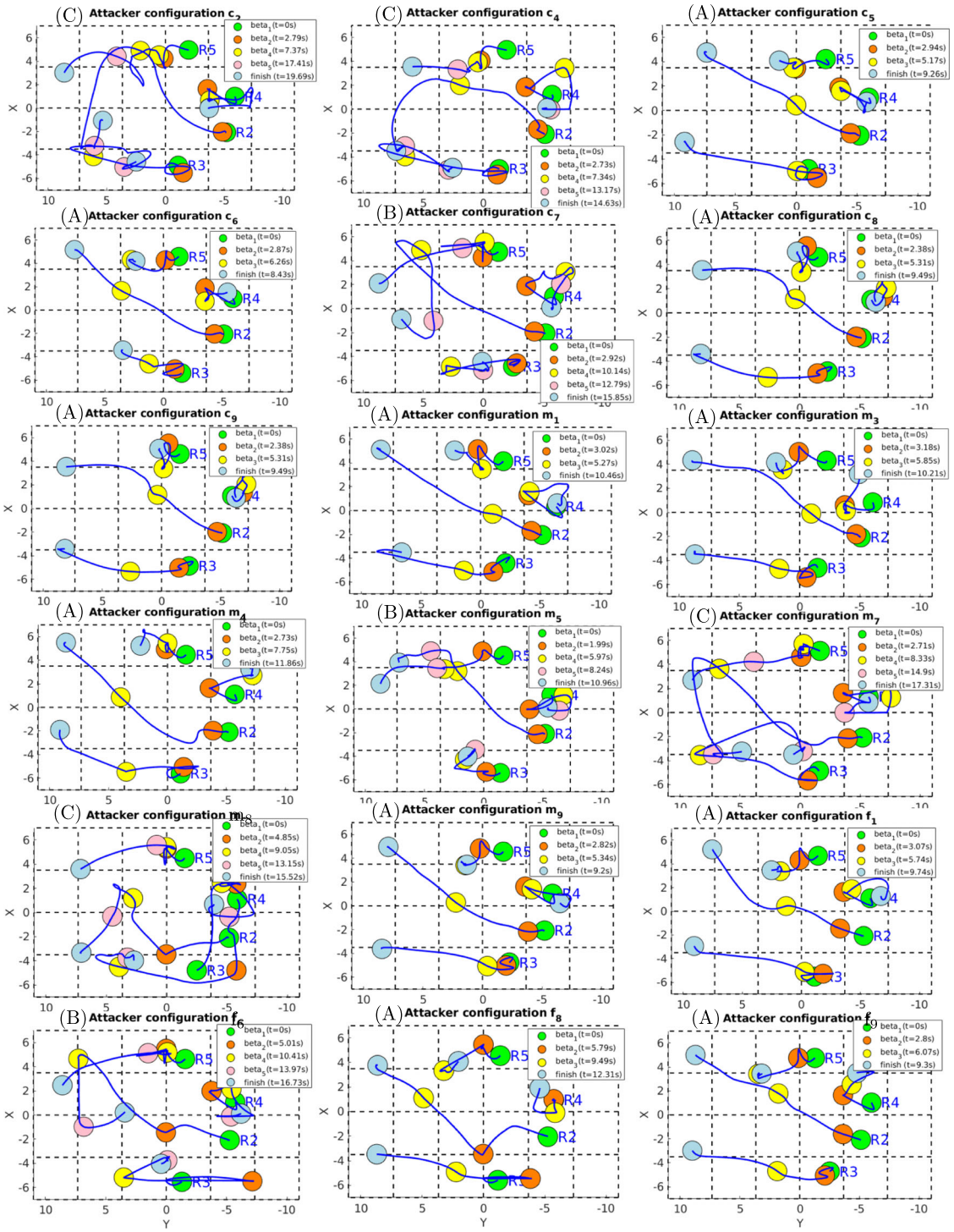


Fig. 32 Attacker trajectories (Table 3), excluding "Too early" attacks and f_2 and f_3. Attack type in brackets

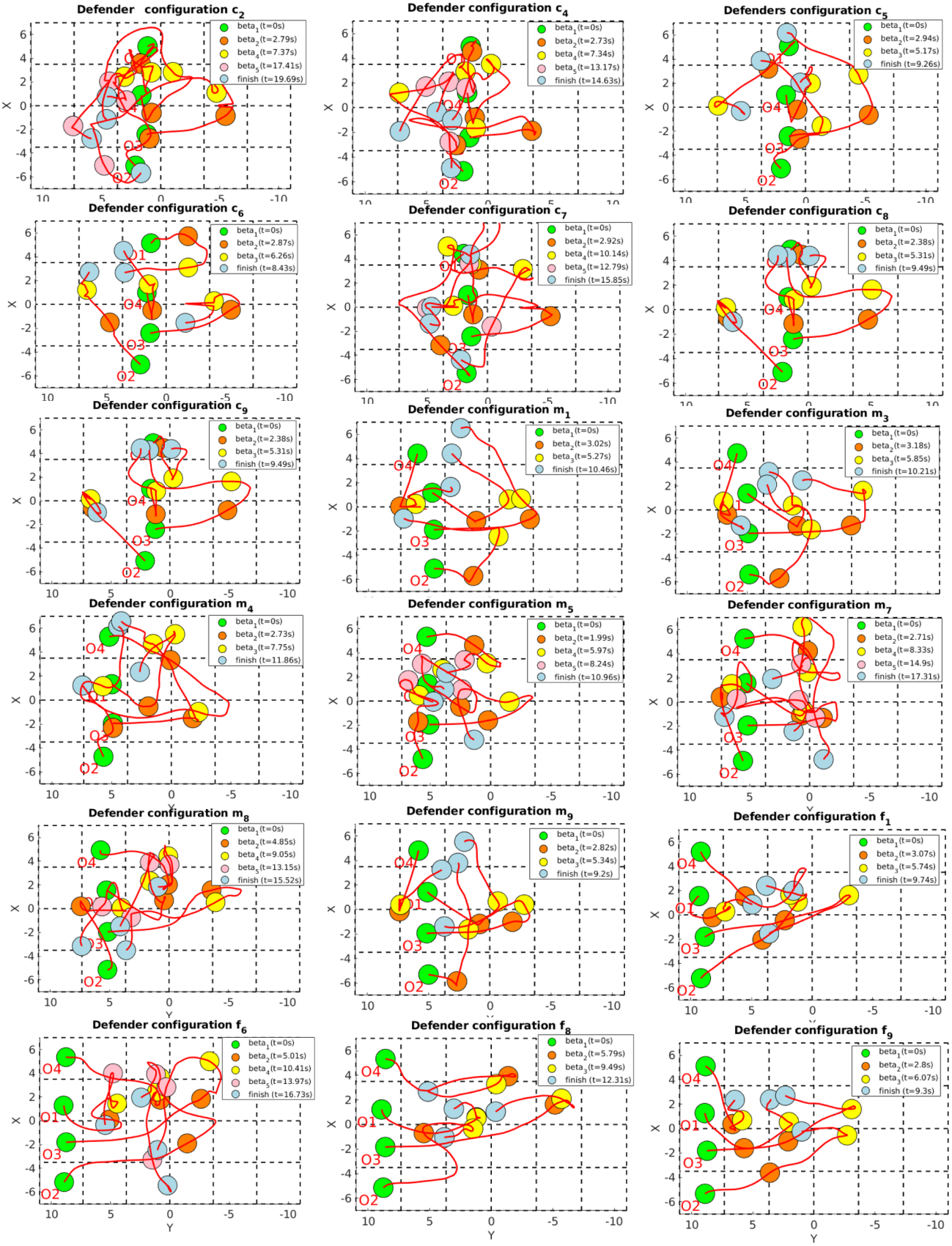


Fig. 33 Defender trajectories (Table 3), excluding "Too early" attacks and f_2 and f_3

D.3 Zone Adjustments Real World Experiment

The real-world experiment zone assignments mostly resemble those in Fig. 11. Only small changes are made as it involved a previous strategy iteration:

- Robot 4 at β_2 and β_3 was allowed to occupy any zone on the attacker's side of the field, i.e., Z13–Z24.
- Robot 3 at β_2 would move towards zone Z9.
- Robot 2 at β_3 was assigned a target zone at Z5 instead of zone Z1.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10846-025-02334-0.

Acknowledgements We would like to thank the FC Eindhoven futsal for allowing us to illustrate and discuss one of their strategies. We would like to thank MSL team Robotsport for participating in our real-world experiment. We would like to thank the members of Tech United Eindhoven robot soccer team for their cooperation. Finally, we would like to thank our reviewers and editor for providing thorough feedback that has led to a significant improvement of the manuscript.

Author Contributions All authors contributed to the conceptualization of the study. HC (Hao Liang Chen) and HB (Herman Bruyninckx) contributed to the design of the study. HC and RB (Ruben Beumer) contributed to the experimental setup of the study. HC wrote the draft of the manuscript. RB, ET (Elena Torta), HB, and RM (René van de Molengraft) provided critical review of the method and manuscript. All authors contributed to the article and approved the submitted version.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability All materials to support the implementation of the approach is explained within the supplementary materials as appendices. Within these appendices, links to the Tech United Eindhoven robot soccer team repository is provided.

Materials Availability Not applicable.

Code Availability Not applicable.

Declarations

Conflicts of interest/Competing interests The authors have no relevant financial or non-financial interests to disclose.

Ethics approval and consent to participate No ethical approval is required as no humans or animals were actively involved in our experiments.

Consent for publication We blurred/occluded humans that could be identifiable on our pictures/videos. For the real world experimentations, consent was given by the referee to show his back.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Rossi, A., Paetzel-Prüsmann, M., Keijsers, M., Anderson, M., Anderson, S.L., Barry, D., Gutsche, J., Hart, J., Iocchi, L., Kokkelmans, A., Kuijpers, W., Liu, Y., Polani, D., Roscon, C., Scheunemann, M., Stone, P., Vahl, F., Molengraft, R., Stryk, O.: The human in the loop Perspectives and challenges for RoboCup 2050. Auton. Robot. **48**(2), 8 (2024). https://doi.org/10.1007/s10514-024-10159-3
2. Gudmundsson, J., Horton, M.: Spatio-temporal analysis of team sports. ACM Comput. Surv. **50**(2), (2017). https://doi.org/10.1145/3054132
3. Buekers, M., Montagne, G., Ibáñez-Gijón, J.: Strategy and tactics in sports from an ecological-dynamical perspective: What is in there for coaches and players? Mov Sport Sci/Sci Mot **108**, 1–11 (2020). https://doi.org/10.1051/sm/2019026
4. Schwering, A., Wang, J., Chipofya, M., Jan, S., Li, R., Broelemann, K.: Sketchmapia: Qualitative representations for the alignment of sketch and metric maps. Spat. Cognit. Comput. **14**, 220–254 (2014). https://doi.org/10.1080/13875868.2014.917378
5. Lord, F., Pyne, D.B., Welvaert, M., Mara, J.K.: Methods of performance analysis in team invasion sports: A systematic review. J. Sports Sci. **38**(20), 2338–2349 (2020). https://doi.org/10.1080/02640414.2020.1785185
6. Neves, A.J.R., Amaral, F., Dias, R., Silva, J., Lau, N.: A new approach for dynamic strategic positioning in robocup middle-size league. In: Pereira, F., Machado, P., Costa, E., Cardoso, A. (eds.) Progress in Artificial Intelligence, pp. 433–444. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23485-4_43
7. Mendoza, J.P., Biswas, J., Cooksey, P., Wang, R., Klee, S., Zhu, D., Veloso, M.: Selectively reactive coordination for a team of robot soccer champions. Proc. AAAI Conf. Artif. Intell. **30**(1), (2016). https://doi.org/10.1609/aaai.v30i1.10415
8. Simões, M., Nogueira, T.: Towards setplays learning in a multiagent robotic soccer team. In: 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), pp. 277–282 (2018). https://doi.org/10.1109/LARS/SBR/WRE.2018.00058
9. Bell, M.Z.: Why expert systems fail. J. Oper. Res. Soc. **36**(7), 613–619 (1985). https://doi.org/10.1057/jors.1985.106
10. Chen, H.L., Hendrikx, B., Torta, E., Bruyninckx, H., Molengraft, R.: Behavior adaptation for mobile robots via semantic map compositions of constraint-based controllers. Front. Robot. AI **10**, 917637 (2023). https://doi.org/10.3389/frobt.2023.917637
11. U.S. Department of Transportation, Federal Highway Administration: Manual on Uniform Traffic Control Devices: For Streets and Highways, 11 edn. U.S. Department of Transportation, Federal Highway Administration, Washington, D.C. (2023). 11th Edition. https://mutcd.fhwa.dot.gov/pdfs/11th_Edition/mutcd11thedition.pdf
12. Antonioni, E., Suriani, V., Riccio, F., Nardi, D.: Game strategies for physical robot soccer players: A survey. IEEE Trans. Games **13**(4), 342–357 (2021). https://doi.org/10.1109/TG.2021.3075065
13. 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 the 2014 International Conference on Autonomous Agents and Multi-Agent Systems. AAMAS '14, pp. 493–500. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2014). https://dl.acm.org/doi/10.5555/2615731.2615812
14. Ribeiro, A.F.A., Lopes, A.C.C., Ribeiro, T.A., Pereira, N.S.S.M., Lopes, G.T., Ribeiro, A.F.M.: Probability-based strategy for a football multi-agent autonomous robot system. Robot. **13**(1), (2024). https://doi.org/10.3390/robotics13010005
15. Akiyama, H., Nakashima, T., Fukushima, T., Zhong, J., Suzuki, Y., Ohori, A.: Helios2018: Robocup 2018 soccer simulation 2d league champion. In: Holz, D., Genter, K., Saad, M., Stryk, O. (eds.) RoboCup 2018: Robot World Cup XXII, pp. 450–461. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-27544-0_37
16. Geiger, M., Ommer, N., Ryll, A.: Robocup 2022 ssl champion tigers mannheim - ball-centric dynamic pass-and-score patterns. In: Eguchi, A., Lau, N., Paetzel-Prüsmann, M., Wanichanon, T. (eds.) RoboCup 2022: Robot World Cup XXV, pp. 276–286. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-28469-4_23
17. MacAlpine, P., Price, E., Stone, P.: Scram: Scalable collision-avoiding role assignment with minimal-makespan for formational positioning. Proc. AAAI Conf. Artif. Intell. **29**(1), (2015). https://doi.org/10.1609/aaai.v29i1.9424
18. Gréhaigne, J.-F., Godbout, P., Bouthier, D.: The foundations of tactics and strategy in team sports. J. Teach. Phys. Educ. **18**(2), 159–174 (1999). https://doi.org/10.1123/jtpe.18.2.159
19. Costa, I., Greco, P.J., Costa, I., Manuel, J., Silva, G., Greco, P., Mesquita, I.: Tactical principles of soccer: concepts and application tactical principles of soccer. Motriz. Revista de Educação Física **15**, 657–668 (2009)
20. Sturgess, P.: Futsal: Training, Technique and Tactics. Bloomsbury Publishing, (2017)
21. Skinner, B.: The price of anarchy in basketball. J. Quant. Anal. Sports **6**(1), (2010). https://doi.org/10.2202/1559-0410.1217
22. Stone, P., Veloso, M.: Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. Artif. Intell. **110**(2), 241–273 (1999). https://doi.org/10.1016/S0004-3702(99)00025-9
23. Mota, L., Reis, L.P., Lau, N.: Multi-robot coordination using set-plays in the middle-size and simulation leagues. Mechatron **21**(2), 434–444 (2011). https://doi.org/10.1016/j.mechatronics.2010.05.005. Special Issue on Advances in intelligent robot design for the Robocup Middle Size League
24. Browning, B., Bruce, J., Bowling, M., Veloso, M.: Stp: Skills, tactics, and plays for multi-robot control in adversarial environments. Proc. Inst Mech. Eng., Part I: J. Syst. Control Eng. **219**(1), 33–52 (2005). https://doi.org/10.1243/095965105X9470
25. Dylla, F., Ferrein, A., Lakemeyer, G., Murray, J., Obst, O., Röfer, T., Schiffer, S., Stolzenburg, F., Visser, U., Wagner, T.: Approaching a formal soccer theory from behaviour specifi cations in robotic soccer. WIT Trans. State-of-the-art in Sci. Eng. **32**, (2008)
26. Musumeci, E., Suriani, V., Antonioni, E., Nardi, D., Bloisi, D.D.: Adaptive team behavior planning using human coach commands. In: Eguchi, A., Lau, N., Paetzel-Prüsmann, M., Wanichanon, T. (eds.) RoboCup 2022: Robot World Cup XXV, pp. 112–123. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-28469-4_10
27. Veloso, M., Stone, P.: Individual and collaborative behaviors in a team of homogeneous robotic soccer agents. In: Proceedings International Conference on Multi Agent Systems (Cat. No.98EX160), pp. 309–316 (1998).https://doi.org/10.1109/ICMAS.1998.699074
28. Cravo, J., Almeida, F., Abreu, P.H., Reis, L.P., Lau, N., Mota, L.: Strategy planner: Graphical definition of soccer set-plays. Data Knowl. Eng. **94**, 110–131 (2014). https://doi.org/10.1016/j.datak.2014.10.001
29. Simões, M.A.C., Nobre, J., Sousa, G., Souza, C., Silva, R.M., Campos, J., Souza, J.R., Nogueira, T.: Strategy planner: Enhancements to support better defense and pass strategies within an lfd approach. In: 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 46–52 (2020). https://doi.org/10.1109/ICARSC49921.2020.9096188
30. Koning, L., Mendoza, J.P., Veloso, M., Molengraft, R.: Skills, tactics and plays for distributed multi-robot control in adversarial environments. In: Akiyama, H., Obst, O., Sammut, C., Tonidandel, F. (eds.) RoboCup 2017: Robot World Cup XXI, pp. 277–289. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00308-1_23
31. Ferrein, A., Fritz, C., Lakemeyer, G.: Using golog for deliberation and team coordination in robotic soccer. Künstliche Intell. **19**(1), 24 (2005)
32. Haslum, P., Lipovetzky, N., Magazzeni, D., Muise, C., Brachman, R., Rossi, F., Stone, P.: An Introduction to the Planning Domain Definition Language vol. 13. Springer, (2019). https://doi.org/10.1007/978-3-031-01584-7
33. Ziparo, V.A., Iocchi, L., Lima, P.U., Nardi, D., Palamara, P.F.: Petri net plans. Auton. Agent. Multi-Agent Syst. **23**(3), 344–383 (2011). https://doi.org/10.1007/s10458-010-9146-1
34. Simões, M.A., Silva, R.M., Nogueira, T.: A dataset schema for cooperative learning from demonstration in multi-robot systems. J. Intell. Robot. Syst. **99**(3), 589–608 (2020). https://doi.org/10.1007/s10846-019-01123-w
35. D'Amour, A., Cervone, D., Bornn, L., Goldsberry, K.: Move or die: How ball movement creates open shots in the nba. In: Sloan Sports Analytics Conference (2015). https://www.lukebornn.com/papers/damour_ssac_2015.pdf
36. Taki, T., Hasegawa, J., Fukumura, T.: Development of motion analysis system for quantitative evaluation of teamwork in soccer games. In: Proceedings of 3rd IEEE International Conference on Image Processing, vol. 3, pp. 815–818 (1996). https://doi.org/10.1109/ICIP.1996.560865
37. Güldenpenning, I., Kunde, W., Weigelt, M.: How to trick your opponent: A review article on deceptive actions in interactive sports. Front. Psychol. **8**, (2017). https://doi.org/10.3389/fpsyg.2017.00917
38. Fewell, J.H., Armbruster, D., Ingraham, J., Petersen, A., Waters, J.S.: Basketball teams as strategic networks. PLoS ONE **7**(11), 1–9 (2012). https://doi.org/10.1371/journal.pone.0047445
39. Corrêa, U.C., Alegre, F., Freudenheim, A.M., Santos, S., Tani, G.: The game of futsal as an adaptive process. Nonlinear Dyn. Psychol. Life Sci. **16**(2), 185–203 (2012)
40. Sip, K.E., Lynge, M., Wallentin, M., McGregor, W.B., Frith, C.D., Roepstorff, A.: The production and detection of deception in an interactive game. Neuropsychol. **48**(12), 3619–3626 (2010). https://doi.org/10.1016/j.neuropsychologia.2010.08.013
41. Lamas, L., Barrera, J., Otranto, G., Ugrinowitsch, C.: Invasion team sports: Strategy and match modeling. Int. J. Perform. Anal. Sport **14**(1), 307–329 (2014). https://doi.org/10.1080/24748668.2014.11868723
42. Athletic, T.-T.: How To Watch Football: 52 Rules for Understanding the Beautiful Game, On and Off the Pitch. particular books, (2022)
43. Fujimura, A., Sugihara, K.: Geometric analysis and quantitative evaluation of sport teamwork. Syst. Comput. Japan **36**(6), 49–58 (2005). https://doi.org/10.1002/scj.20254
44. Nakanishi, R., Maeno, J., Murakami, K., Naruse, T.: An approximate computation of the dominant region diagram for the real-time analysis of group behaviors. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009: Robot Soccer World Cup XIII, pp. 228–239. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11876-0_20

45. Fukushima, T., Nakashima, T., Akiyama, H.: Online opponent formation identification based on position information. In: Akiyama, H., Obst, O., Sammut, C., Tonidandel, F. (eds.) RoboCup 2017: Robot World Cup XXI, pp. 241–251. Springer, Cham (2018).https://doi.org/10.1007/978-3-030-00308-1_20
46. Li, X., Meng, M., Hong, Y., Chen, J.: A survey of decision making in adversarial games. SCI. CHINA Inf. Sci. **67**(4), 141201 (2024). https://doi.org/10.1007/s11432-022-3777-y

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Hao Liang Chen obtained his M.Sc. degree (2018) in Mechanical Engineering and his Ph.D. degree (2025) from Eindhoven University of Technology. His Ph.D. research focused on modeling robot behavior using semantic world model knowledge. He is currently a postdoctoral researcher at the Eindhoven University of Technology in the Robotics group, where his current research is focused on the design of digital twin systems for robotic education.

Ruben Beumer received the M.Sc. in Mechanical Engineering from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2021. He is currently pursuing a PhD in the Robotics section of the same department, doing research on decision-making and world modelling for robots in agriculture. He is also a member of the Tech United RoboCup team, working on robots participating in the MSL and (in the near future) the humanoid league.

Elena Torta is an associate professor in Robotics at the Mechanical Engineering Department of the Eindhoven University of Technology. Her research addresses control and world modelling problems in human-robot and robot-robot collaboration. She holds a PhD cum laude from the Eindhoven University of Technology and is experienced and fascinated by research problems that cross disciplinary boundaries. She holded a position as software architect at ASML B.V. for five years. She has performed several human-robot interaction studies that lead to computational models actionable by robotics control algorithms.

Dr. Herman Bruyninckx obtained the Masters degrees in Mathematics (Licentiate, 1984), Computer Science (Burgerlijk Ingenieur, 1987) and Mechatronics (1988), all from the KU Leuven, Belgium. In 1995 he obtained his Doctoral Degree in Engineering from the same university. He is full-time Full Professor at the KU Leuven, and part-time at the Eindhoven University of Technology. The research focus in both places is on the composability of the most advanced, knowledge driven algorithms for the dynamics of motion control of complex robotics applications, with distributed sensor processing and resource monitoring. The complementary objectives are to realize such systems with the least amount of resource costs, with "good enough" quality, and with full "explainability".

René van de Molengraft received the M.Sc. degree (cum laude) in mechanical engineering and the Ph.D. degree from Eindhoven University of Technology, Eindhoven, The Netherlands, in 1986 and 1990, respectively. In 1991, he returned to the department of Mechanical Engineering, after having fulfilled his military service. He is currently a Full Professor in Robotics and chair of the Robotics Section in the department. He founded the Tech United RoboCup Team, and gained eight world champion titles in the MSL robot soccer competition, and two world champion titles in the @Home competition for domestic service robots.