2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Circular Field Motion Planning for Highly-Dynamic Multi-Robot Systems with Application to Robot Soccer

Fabrice Zeug*, Marvin Becker* and Matthias A. Müller

***Abstract*— The rise of autonomous driving in everyday life makes efficient and collision-free motion planning more important than ever. However, multi robot applications in highly dynamic environments still pose hard challenges for state-of-the-art motion planners. In this paper, we present a new iteration of a reactive circular fields motion planner with the focus on simultaneous control of multiple robots in robotic soccer games, which is able to operate omnidirectional robots safely and efficiently despite high measurement delays and inaccuracies. Our extension enables the definition and effective execution of complex tasks in soccer specific problems. We extensively evaluated our planner in several complex simulation environments and experimentally verified the approach in realistic scenarios on real soccer robots. Furthermore, we demonstrated the capabilities of our motion planner during the successful participation in the RoboCup 2022 and 2023.**

I. INTRODUCTION

Research on motion planning for mobile robots is one of the most intensively studied subfields of robotics and remains a very active topic [1]. Many approaches have been developed that are able to operate safely in complex environments [2]–[4]. However, highly dynamic scenarios that involve multiple robots operating in the same environment pose hard challenges that many traditional approaches are not able to overcome [5]. One of these settings is robot soccer.
In recent years robot soccer has gained popularity in robotics research. It provides a well-suited environment to study and overcome the prevailing challenges of modern mobile robotics. A major part of research in this field is initiated and motivated by the RoboCup [6]. The goal of RoboCup is to push state-of-the-art robot soccer by offering a platform for robotics teams to compete against each other in various leagues. The approach that we present in this paper is developed for the Small Size League [7], in which multiple mobile robots autonomously compete as a team. As both teams have to operate without human input, algorithms have to be developed that make strategic decision and autonomously control the robot to the desired targets without colliding with other robots. The high measurement delays and uncertainties and the high velocities of robots of up to $3\,\mathrm{m\,s^{-1}}$ are serious challenges that have to be overcome.
In this paper, we focus on the collision avoidance and motion planning problem of the multi-agent robot team and present the implementation and modifications of a new reactive motion planner in this scenario.

*These authors contributed equally. This work was partially funded by the Region Hannover in the project *roboterfabrik*. The authors are with the Institute of Automatic Control, Leibniz University Hannover, Germany, zeug@stud.uni-hannover.de, {becker,mueller}@irt.uni-hannover.de

II. RELATED WORK

Motion planners in the field of robot soccer are often split into a global path planning unit and a local control unit, which follows the generated global path while simultaneously avoiding moving obstacles [5], [8], [9]. Global path planners are capable of finding feasible paths in complex environments. However, path planning comes with high computational costs and is not well suited for online control of robots [10]. In contrast, local planners adapt quickly to changing environments but are often not capable of finding solutions in complex scenarios [4]. It is crucial to react immediately to changes in the environment and avoid obstacles for the safe operation of robots in dynamic applications [11]. The combination of both approaches results in a potent, flexible solution for many applications [4], [10], [12].
A seminal work in local motion planning is the artificial potential fields (APF) approach from [13]. APFs generate a force on the robot by using the negative gradient of a potential, which has a component that is proportional to the distance to the goal and inverse proportional components to the distance to obstacles. A major flaw of APFs is that they result in local minima, which disqualifies them for global application. The authors in [14] use virtual obstacles to repel the robot from local minima, while [15] presents a method which involves finding random close configurations to escape from local minima. APFs have been further enhanced by other authors in [16]–[18]. The approach in [17] completely avoids local minima by defining a circular field (CF) which is derived from forces that apply on a moving electrical charge in a magnetic field. The force is inverse proportional to the distance to an obstacle and always applied perpendicular to the robots' direction of movement. The authors of [19] show that in planar environments with static obstacles, such a CF approach guarantees collision avoidance.
Bypass directions for obstacles are not addressed in [17], despite being crucial for efficient and successful robot motion. In [20] a similar collision avoidance approach is presented, where the direction of motion is used to determine the bypass direction around the obstacle. The CF approach in [21] uses the vector to the goal and the vector to the obstacles' center of mass to determine the avoidance direction around each obstacle individually in every control loop cycle. Here, the planner always chooses the bypass direction that deviates least from the straight line to the goal. Globally, this method only provides suboptimal results and it does not take the movement of other robots into account. The approach was extended in [22] by using a virtual multi-agent framework

to find the best set of avoidance directions for the entire environment. Compared to the method presented in [21], this approach requires significantly more computational resources especially in applications with many obstacles, but provides superior global solutions. Moreover, results from the multi-agent approach will not lead into dead ends, which could happen in clustered scenarios with the approach from [21]. The main contribution of this paper is the implementation, enhancement and evaluation of the reactive CF approaches from [21] and [22] on real soccer robots. In this context, the capabilities of our planner were demonstrated during the international robot soccer competitions at the RoboCup 2022 and 2023, in which we secured the third and second place in the Small Size League in Division B, respectively [7]. Furthermore, we extended and enhanced the planner focusing on the application in robot soccer by implementing support for multiple simultaneous goals and by introducing anti-goals to avoid certain areas. Our proposed planner is designed for achieving low computational cost, fast reaction times, and enabling multi-robot cooperative planning. Moreover, we compare the performance of our planner against several state-of-the-art motion planners in three challenging simulation environments and demonstrate the proficiency of our planner in two real world applications with realistic scenarios. Finally, our complete robot soccer software framework is made available to the community[1].

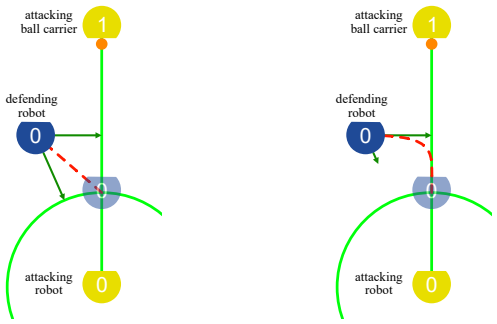
III. SETUP AND FOUNDATION

In the RoboCup Small Size League [7], teams of six (or eleven) robots compete against each other in a soccer-like game. The robots have a diameter of $180\,\mathrm{mm}$ and are capable of driving omnidirectionally. A regular golf ball is used for playing. The robots can interact with the ball by using a special roller that gives the ball a backspin towards the robot, so the ball stays with the robot while moving. Furthermore, the robots are equipped with a kicking device to kick the ball with up to $6.5\,\mathrm{m\,s^{-1}}$. Both robot teams are tracked with a shared vision system which is mounted over the field. The average refresh-rate of the vision system is $60\,\mathrm{Hz}$ and the latency can reach up to $45\,\mathrm{ms}$. All robots from one team are controlled using a single computer, which sends velocity commands to the robots. To improve the noisy position and velocity information, the measurement from the vision system is filtered by an adapted discrete Kalman filter [23]. Our adaption uses a second prediction step to account for the vision delay (details are described in our accompanying repository [24]).

IV. PLANNER

Motion planning in robot soccer places high demands on the planning algorithm, as the problems frequently involve complex tasks such as *defend the goal against a specific opposing robot* and go beyond typical point-to-point movements. To meet these high requirements, we distinguish in our approach between three different object types: goals, anti-goals, and obstacles, which influence the behavior of the robot differently and enable us to solve complex tasks without colliding with obstacles. Fig. 1 exemplarily illustrates this for the case of using two goals that encompass the task of defending against a pass.

[1]https://github.com/luhsoccer/cf_soccer_icra2024



(a) Both goals have the same weight

(b) The line goal has a higher weight than the circular goal

Fig. 1. Illustration of ally robot 0 (blue) defending against a pass from enemy robot 1 to enemy robot 0 (yellow). Two goals (light green) guide the ally: a line between enemies and a circle for the intercept distance from enemy robot 0. The goal force is displayed in dark green. In examples (a) and (b), the goals have different weights, affecting the robot's movement strategy (red).

A. Goal Forces

In contrast to many motion planners, our planner is capable of handling multiple goals at once. The planner always tries to satisfy every goal simultaneously. In this context, we prioritize each goal i by assigning a corresponding weight ${}^{i}w_g$. In our definition, goals can be described as geometric shapes instead of simple goal poses. It is worth mentioning that, if goals do not intersect, the planner will lead the robot to a position between the goals. Our implementation for the RoboCup includes points, lines, circles (filled and not filled), arcs, and rectangles. Furthermore, it is possible to combine multiple shapes into a composed shape. Note that every point of a goal is weighted equally and that the planner always tries to reach the closest point on the shape of the goal. Using these modifications, our combined goals are able to portray higher-level tasks in robot soccer, e.g., blocking and intercepting passes. Fig. 1 shows the exemplary task of defending an enemy robot against a pass and the corresponding geometric primitives. To steer the robot towards goals, we use an adapted version of the velocity limiting controller (VLC) from [13]. We adapt the VLC to the design of our task to handle multiple moving goals. Every goal i generates a desired velocity:

$$ {}^{i}\dot{\boldsymbol{x}}_d \;=\; {}^{i}w_g \frac{k_{G,G}}{k_{G,V}} ({}^{i}\boldsymbol{x}_g - \boldsymbol{x}) + {}^{i}\dot{\boldsymbol{x}}_g, \tag{1} $$

where $k_{G,G}, k_{G,V} > 0$ are scaling constants, the position of the robot is $\boldsymbol{x}$ and ${}^{i}\boldsymbol{x}_g$ describes the closest point on the goal. In contrast to the original approach, our planner takes the goal velocity ${}^{i}\dot{\boldsymbol{x}}_g$ into account when reaching the goal position. Before a control force is determined, the desired

velocity is combined with the desired velocity from anti-goals.

B. Anti-Goal Forces

Anti-goals describe areas from which the robot is constantly repelled until the maximum influence distance $d_{ag,\max}$ is reached. They are used to, e.g., become available for passes by moving away from borders, enemies, or other unfavorable areas. The same geometric shapes from goals can be used to describe anti-goals. They also generate a desired velocity:

$$ {}^{k}\dot{\boldsymbol{x}}_{ag,d} = \begin{cases} 0 & \text{if } \| {}^{k}\boldsymbol{x}_{ag} - \boldsymbol{x} \| > d_{ag,\max} \\ {}^{k}w_{ag} \frac{{}^{k}\boldsymbol{x}_{ag} - \boldsymbol{x}}{\| {}^{k}\boldsymbol{x}_{ag} - \boldsymbol{x} \|^2} & \text{otherwise} \end{cases}, $$

with ${}^{k}\boldsymbol{x}_{ag}$ being the closest point on the anti-goal and ${}^{k}w_{ag}$ representing a weight factor. A total desired velocity is deducted from all goals and anti-goals and limited to the maximum desired velocity $\dot{x}_{\max}$ with:

$$ \dot{\boldsymbol{x}}^{*}_{d,\text{total}} = \sum_{i=0}^{n_{\text{goals}}} {}^{i}\dot{\boldsymbol{x}}_{d} + \sum_{k=0}^{n_{\text{anti-goals}}} {}^{k}\dot{\boldsymbol{x}}_{ag,d}, \tag{2} $$

$$ \dot{\boldsymbol{x}}_{d,\text{total}} = \dot{\boldsymbol{x}}^{*}_{d,\text{total}} \min\left(1, \frac{\dot{x}_{\max}}{\| \dot{\boldsymbol{x}}^{*}_{d,\text{total}} \|}\right). \tag{3} $$

Finally, the total force from goals and anti-goals is:

$$ \boldsymbol{F}_{g} = k_{G,V}(\dot{\boldsymbol{x}}_{d,\text{total}} - \dot{\boldsymbol{x}}). \tag{4} $$

C. Obstacle Forces

Obstacles describe areas that the robot should avoid during the execution. In contrast to anti-goals, the robot is not actively repelled if no collision is imminent. All enemy robots as well as the ally ones are considered as obstacles, to avoid collisions between them. The forces generated by obstacles are derived from the approach presented in [22] with the main difference being that robots are considered as perfectly round obstacles. As a result of that, only the point ${}^{j}\boldsymbol{x}_{c}$ on the obstacle that is closest to the robot is considered for the force generation. An artificial current ${}^{j}\boldsymbol{c}$ is defined by ${}^{j}\boldsymbol{c} = {}^{j}\boldsymbol{d} \times {}^{j}\boldsymbol{b}$, where ${}^{j}\boldsymbol{d} = {}^{j}\boldsymbol{x}_{c} - \boldsymbol{x}$ is the closest distance to obstacle j and ${}^{j}\boldsymbol{b}$ is the magnetic field vector. It defines in which direction the obstacle is bypassed and can be pointing in positive or negative Z-direction. In section IV-D we will present different methods to determine the direction of these magnetic field vectors. The current vector is then used to determine the CF force ${}^{j}\boldsymbol{F}^{*}_{CF}$ using a virtual magnetic field ${}^{j}\boldsymbol{B}$ similar to the force generated by a moving electrical charge in a magnetic field:

$$ {}^{j}\boldsymbol{B} = \frac{k_{CF}}{\| {}^{j}\boldsymbol{d} \|^2} {}^{j}\boldsymbol{c} \times \frac{{}^{j}\dot{\boldsymbol{x}}}{\| {}^{j}\dot{\boldsymbol{x}} \|}, \tag{5} $$

$$ {}^{j}\boldsymbol{F}^{*}_{CF} = \frac{{}^{j}\dot{\boldsymbol{x}}}{\| {}^{j}\dot{\boldsymbol{x}} \|} \times {}^{j}\boldsymbol{B}, \tag{6} $$

with k_{CF} being a constant gain to adapt the magnitude of the force and ${}^{j}\dot{\boldsymbol{x}}$ being the relative velocity between obstacle and robot. Furthermore, ${}^{j}\boldsymbol{F}^{*}_{CF}$ is only used when certain conditions are met:

$$ {}^{j}\boldsymbol{F}_{CF} = \begin{cases} 0 & \text{if } \| {}^{j}\boldsymbol{d} \| \geq d_{o,\max} \\ 0 & \text{if } \| {}^{j}\boldsymbol{d} \| \geq \| \boldsymbol{x}_{g,wm} \| \\ 0 & \text{if } {}^{j}\boldsymbol{d} \cdot \dot{\boldsymbol{x}} < 0 \\ {}^{j}\boldsymbol{F}^{*}_{CF} & \text{else} \end{cases}, \tag{7} $$

where $d_{o,\max}$ is the maximum influence distance of obstacles and $\| \boldsymbol{x}_{g,wm} \|$ is the norm of the weighted average distances to all goals:

$$ \boldsymbol{x}_{g,wm} = \frac{\sum_{j=0}^{n_{\text{goals}}} {}^{j}w_{g}({}^{j}\boldsymbol{x}_{g} - \boldsymbol{x})}{\sum_{j=0}^{n_{\text{goals}}} {}^{j}w_{g}}. \tag{8} $$

As a result of (7) the influence of an obstacle is ignored if the robot moves away from the obstacle, the obstacle is further away than the maximum influence distance $d_{o,max}$ or further away than the mean of the weighted goal distances. This avoids that obstacles that currently are not relevant for completing the task influence the behavior of the robot.
The forces from goals and anti-goals $\boldsymbol{F}_{g}$ and all obstacles forces ${}^{j}\boldsymbol{F}_{CF}$ are then combined to a total force

$$ \boldsymbol{F}_{t} = w_{1,2}\boldsymbol{F}_{g} + \sum_{j=0}^{n_{\text{obstacles}}} {}^{j}\boldsymbol{F}_{CF}, \tag{9} $$

with $w_{1,2}$ representing the combination of two relaxation factors. They reduce the influence of goals and anti-goals when the robot operates close to obstacles. In particular, $w_{1,2}$ is determined as follows:

$$ w_{1,2} = \max(w_{1}w_{2}, w_{\min}), \tag{10} $$

$$ w_{1} = 1 - e^{\frac{-\| \boldsymbol{d}_{o,\min} \|}{\alpha r_{\text{robot}}}}, \tag{11} $$

$$ w_{2} = 1 - \frac{\boldsymbol{d}_{g,\max} \cdot \boldsymbol{d}_{o,\min}}{\| \boldsymbol{d}_{g,\max} \| \| \boldsymbol{d}_{o,\min} \|}, \tag{12} $$

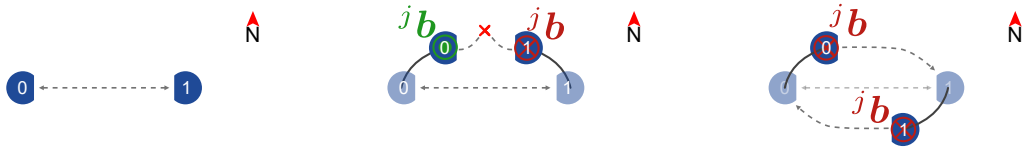
where $\boldsymbol{d}_{o,\min}$ is the vector from the robot to the closest obstacle, $\boldsymbol{d}_{g,\max}$ the vector from the robot to the furthest goal, and $w_{\min}$ a minimum relaxation factor. In (11) α describes a constant gain and r_{robot} the radius of the robot. The gain w_{1} reduces the influence of goals when the robot is close to obstacles, while w_{2} reduces the influence when the closest obstacle blocks the direct movement towards the goal and the robot has to avoid it. The gain w_{2} also enhances the force when the obstacle is behind the robot. Both w_{1} and w_{2} are derived from the goal relaxation factors presented in [22]. The sum of all forces is converted into a velocity command $\dot{\boldsymbol{x}}_{c}$, which is then sent to the corresponding robot:

$$ \dot{\boldsymbol{x}}_{c} = \dot{\boldsymbol{x}} + \boldsymbol{F}_{t}\Delta t, \tag{13} $$

with Δt being the inverse of the planner frequency.

D. Determination of Avoidance Direction

The key to a successful and optimal execution of the given task is the determination of the magnetic field vector ${}^{j}\boldsymbol{b}$ for every obstacle j, as they determine the global path that the robot follows. In 2D only two bypass directions are possible, and thus only two magnetic field vectors are valid. We



(a) start of the scenario (b) both robots try to bypass to the north (c) robot 0 bypasses to the north, robot 1 bypass to the south

Fig. 2. The behavior of two planners with given magnetic field vectors in an example scenario. Two ally robots (blue) have to switch places (see Fig. (a)). When both planners choose different magnetic field vectors, both robots move to the north (or south) and block each other (see Fig. (b)). If both robots choose the same vector one robot moves towards the north and one towards the south. Here the conflict is resolved instantaneously (see Fig. (c)).

calculate the magnetic field vectors ${}^j\boldsymbol{b}$ similar to [21]. Our planner is further enhanced in two points. First, if obstacles are closer to each other than a maximum distance d_{og}, they are considered as a group and always get the same magnetic field vector assigned. This prevents the planner from trying to lead the robot through bottlenecks that it cannot pass through. Therefore, d_{og} is chosen as to the robot's diameter including a safety margin. We calculate the angle between two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^2$ from the difference of the heading of both vectors to preserve the sign of the heading difference[2]:

$$\sphericalangle^*(\boldsymbol{v}_1, \boldsymbol{v}_2) = \operatorname{atan2}(v_{1,y}, v_{1,x}) - \operatorname{atan2}(v_{2,y}, v_{2,x}), \quad (14)$$

$$\sphericalangle(\boldsymbol{v}_1, \boldsymbol{v}_2) = \begin{cases} \sphericalangle^*(\boldsymbol{v}_1, \boldsymbol{v}_2) - 2\pi & \text{if } \sphericalangle^*(\boldsymbol{v}_1, \boldsymbol{v}_2) > \pi \\ \sphericalangle^*(\boldsymbol{v}_1, \boldsymbol{v}_2) + 2\pi & \text{if } \sphericalangle^*(\boldsymbol{v}_1, \boldsymbol{v}_2) \leq -\pi \\ \sphericalangle^*(\boldsymbol{v}_1, \boldsymbol{v}_2) & \text{otherwise} \end{cases}.$$

We use the following equations to determine the magnetic field vector ${}^j\boldsymbol{b}$:

$${}^j\boldsymbol{b} = \begin{cases} \boldsymbol{u}_z & \text{if } \sphericalangle(\boldsymbol{x}_{g,wm} - \boldsymbol{x}, \boldsymbol{x}_{og} - \boldsymbol{x}) > 0 \\ -\boldsymbol{u}_z & \text{else} \end{cases}, \quad (15)$$

where $\boldsymbol{u}_z = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top$, $\boldsymbol{x}_{og}$ is the center of an obstacle group and $\boldsymbol{x}_{g,wm}$ as defined in (8). The second enhancement is an extension to multiple robots described in section IV-E.

E. Extension to multiple robots

In our approach, one independent planner instance is executed for every robot on the field. Each can be given a separate task. Every planner considers all other robots as moving, unpredictable obstacles. Furthermore, we introduce cooperation between planner instances to solve local conflicting plans. It uses the principle that two ally robots should always assign the same magnetic field vector to each other. The effect of this strategy is explained in Fig. 2 and also works with more than two local conflicting robots as demonstrated in the accompanying video. If the ally planner has not yet assigned a magnetic field vector for the own robot, a magnetic field vector is determined by the original strategy (cf. Eq. (15)). The result is then forwarded to the ally planner, so it can use the same vector in its planning. The overall information about avoidance directions of allied robots is stored in the Magnetic Field Vector (MFV) matrix $M \in \mathbb{B}^{n \times n}$, where $\mathbb{B} = \{-1, 0, 1\}$ and n is the number of robots in one team. In particular, the values -1 and 1 indicate negative and positive magnetic field vectors, respectively, whereas 0 represents an undefined vector. Hence, the entry M_{ij} of M specifies the avoidance direction that robot i assigns to its ally robot j. At the beginning, all entries are undefined. The planner that first tries to assign a magnetic field vector to bypass another robot decides which MFV is used by both robots for the other robot (implying $M = M^\top$).

[2]In (14), we use the atan2 function [25], where $\operatorname{atan2}(v_y, v_x)$ gives the unique angle between the positive x-axis and a vector $v = (v_x, v_y)$ in the x-y-plane.

F. Rotational Motions

In our motion planning framework, the command for the rotational velocity is determined irrespective of the translational motions. The robot is omnidirectional, has three degrees of freedom and thus is able to perform rotational and translational movements simultaneously. The rotation is also controlled with a VLC using the equations presented in section IV-A. In our implementation, it is possible to turn towards a defined position, i.e., the ball, or to a given angle.

V. RESULTS

We evaluate the performance of the proposed circular fields for robot soccer (CFRS) motion planner and compare it against other reactive and optimizing planning approaches. In particular, we compare our planner against the timed elastic band (TEB) planner from [26], the APF planner from [13], the magnetic field inspired (MFI) planner from [20], the gyroscopic (GS) planner from [18], the CF planner from [21] and the circular fields predictions (CFP) planner from [22]. Videos of all simulations and experiments can be found in [24] and are summarized in the accompanying video.
First, we investigate the performance of our planner in the official league simulation [27], where every robot is modeled using point mass kinematics with friction. The simulation takes vision delays and inaccuracies as well as collisions into account. We use the settings of the online RobCup 2021 where the vision delay was $45\,\mathrm{ms}$, the refresh rate was $60\,\mathrm{Hz}$, and the robot position standard deviation was $0.0013\,\mathrm{m}$. The planners and the simulation are executed on an ordinary laptop (AMD Ryzen 5700U, 16GB RAM).

A. Single Robot Performance

In the first scenario, we evaluate the single robot performance in a random static environment. The robot has the task of driving through a randomly generated cluster of six robots. Every planner is evaluated in the same twenty

clusters. The start and goal positions are $6\,\mathrm{m}$ apart. One exemplary scenario of the twenty clusters including the paths of all planners is shown in Fig. 3.

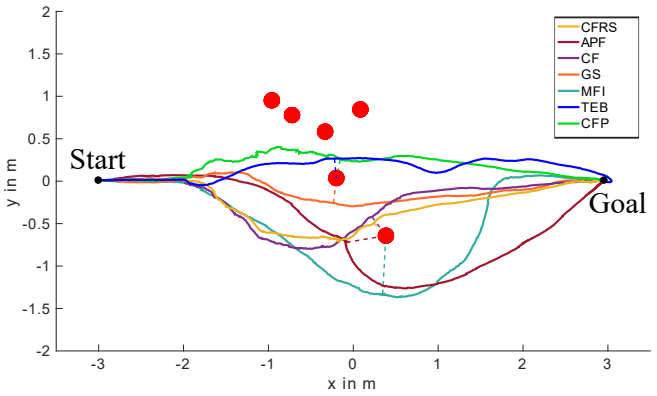


Fig. 3. Example robot cluster (red) and paths of the planners. The closest obstacle distance is marked with dashed lines.

In Table I the quantitative results of the simulations are shown. The CF planner stands out with an average duration of only $2.66\,\mathrm{s}$, which is significantly faster than all other approaches. However, our planner scores second place in terms of duration and manages to keep a $39\,\%$ higher minimal obstacle distance. This can be traced back to the grouping that is done in our approach. In clustered environments, our planner more often decides to bypass the whole cluster at once while the CF planner goes right through the cluster every time which is problematic when small gaps are present. Also, it can be observed that the avoidance directions are frequently altered when using the CFP planner. This can be explained by improved predictions, which become accessible only after a certain computational duration and once the robot has started its movement. It is noteworthy that the APF planner is not able to complete all the scenarios due to the existence of local minima. In 4 of 20 scenarios, the robot stopped moving before the goal was reached. The TEB planner has a significantly longer duration than all other planners which can lead to substantial disadvantages in games of the small size league. The TEB planner is also the only planner of our investigations that does not belong to the category of reactive planners. The MFI planner also performs well with an even higher minimal obstacle distance but with longer paths than the CFP approach.

The time required to calculate velocity commands (last column of Tab. I) is very similar for every planner except for the TEB planner, which takes significantly longer. This can be traced back to the fact, that the TEB planner determines a path to the goal in every step, unlike the other planners. Moreover, the first step of the TEB planner takes on average $(20.13 \pm 3.33)\,\mathrm{ms}$. The CFP also requires calculating a global path to determine the best rotation vectors. However, the global paths explored by the predictive agents are computed in parallel to the control loop and avoidance directions are only updated in the latter once new global paths have been evaluated (compare [22] for details).

All planners except the TEB (too long computation time) and APF (possible local minima) planner produce feasible results that can be used in the considered scenarios. However, in robot soccer applications enemy or ally robots rarely stand still, thus in the following we investigate the performance when multiple robots are moving at the time.

TABLE I

PLANNER PERFORMANCE IN RANDOM ROBOT CLUSTER

Name	Duration [s]	Length [m]	Min. Obstacle Distance [m]	Avg. Velocity [$\mathrm{m\,s^{-1}}$]	Step Comp. time [ms]
CFRS	3.13 ± 0.62	6.47 ± 0.45	0.53 ± 0.32	2.07 ± 0.25	0.24 ± 0.02
APF	4.20 ± 0.82	7.67 ± 0.85	0.57 ± 0.17	1.82 ± 0.27	0.24 ± 0.01
CF	2.66 ± 0.22	6.18 ± 0.22	0.38 ± 0.23	2.32 ± 0.13	0.33 ± 0.02
CFP	3.66 ± 0.97	6.90 ± 0.62	0.63 ± 0.36	1.89 ± 0.30	0.24 ± 0.02
GS	4.64 ± 0.39	6.26 ± 0.18	0.20 ± 0.17	1.35 ± 0.07	0.25 ± 0.01
MFI	3.54 ± 0.33	7.09 ± 0.27	0.70 ± 0.27	2.00 ± 0.12	0.24 ± 0.01
TEB	5.90 ± 1.50	6.24 ± 0.29	0.35 ± 0.09	1.06 ± 0.28	7.39 ± 1.45

B. Multi-Robot Performance

In this scenario, two robots from the same team have to perform movements in proximity to each other including crossing paths. Here, it is insufficient to only react on fast-moving obstacles. Instead, both planner instances have to agree on a bypass direction to solve the conflicting movement tasks. The first setup includes two robots that have to switch places similar to Fig. 2. The initial distance of both robots is $4\,\mathrm{m}$. We executed the same scenario 50 times with every planner. The results are shown in Tab. II. We define a successful execution of the scenario if the minimal obstacle distance is higher than $0.25\,\mathrm{m}$ measured from the centers of the robots. This gives a safety margin of $0.07\,\mathrm{m}$ which should be kept due to measurement inaccuracies.

TABLE II

PLANNER PERFORMANCE IN SWITCH SCENARIO

Name	Duration [s]	Min. Obstacle Distance [m]	Success Rate [%]
CFRS	1.85 ± 0.02	0.364 ± 0.041	100
APF	2.94 ± 0.16	0.176 ± 0.077	0
CF	1.90 ± 0.29	0.330 ± 0.061	88
CFP	2.12 ± 0.68	0.335 ± 0.076	84
GS	2.38 ± 0.53	0.179 ± 0.009	0
MFI	2.72 ± 0.68	0.185 ± 0.014	0
TEB	3.87 ± 0.91	0.450 ± 0.120	88

It stands out that only the planners that use the CFs and the TEB planner are able to perform successful executions of the scenario. This can be traced back to the fact that the relative velocity ${}^{j}\dot{\boldsymbol{x}}$ between a robot and an obstacle is used in (5) and (6) to calculate the force originating from obstacles in CFs. Unlike CFs and the TEB planner, the remaining planners do not consider the velocity of obstacles. As a result, they lack the ability to evade other robots that are approaching. While the CF and CFP planner manage to reach a success rate of $88.12\,\%$ and $83.72\,\%$, respectively, our planner has a success rate of $100.00\,\%$. The TEB planner has a success rate of $88.89\,\%$, but is significantly slower than the CFs planners.

C. Challenging Switch Scenario

To further investigate the performance of the three CF-based planners, we adapted the scenario with an offset of

the goal positions. The execution of this scenario of the three planners is shown in Figure 4. In this scenario, our planner manages to keep the success rate of $100.00\,\%$ in the 50 simulations while both other planners did not manage to execute the scenario successfully at all. In the other approaches, both robots try to bypass each other on the top side, because it would be more optimal for their respective tasks. As a result, the planner instances do not agree on a bypass direction and collide. Only our approach is capable of agreeing on a bypass direction. One robot has to bypass the other on the lower side which takes longer, but the overall result is superior. With this form of cooperation, the planner is able to completely avoid collisions in this scenario.

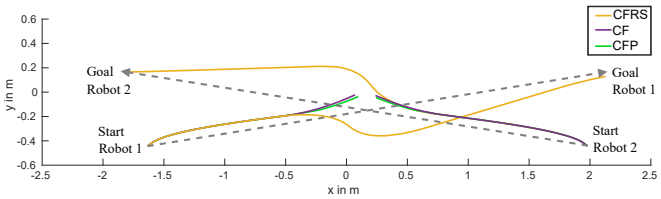


Fig. 4. Switch scenario with offset (gray) and path of CFs planners.

D. Real-World Setup

We also implement and verify our planner on real-world robots. In a real-world setup, the planner is able to perform movements in proximity to other robots and avoid obstacles. Fig. 5 and 6 show the execution of two scenarios that implement common strategies for defending against threats in games of the small size league. In the first scenario (see Fig. 5) three robots build a wall at the penalty area to defend one attacking robot. All robots are controlled with our proposed planner and the three robots in the wall cooperate. The main objective of the defending robots is to keep up with the movements of the attacking robot and avoid gaps between the robots.

The second scenario (see Fig. 6) portrays a defending situation, where the defending robots try to stay between an enemy robot and the ball to prevent a pass to that robot. In this scenario, the attacking robots move on a constant path while both defending robots are controlled by our proposed planner and cooperate to avoid collisions. Note that the defending robots position themselves slightly ahead of the direct line between the opposing robot and the ball, intending to intercept passes in the robot's motion direction.

In addition to these laboratory experiments, we proved the feasibility of CFs for the small size league at the RoboCup 2022 and 2023. In 2022, we used the CFP planner and were able to score third place in our division. However, one major downside of the CFP planner is the computational costs of the planner in the presence of many obstacles, which led to the fact that a powerful desktop workstation computer was needed to control the six robots at once. Therefore, in 2023 we used our proposed improved planner and scored second place. The planner was intentionally tuned to be more aggressive and allow physical contact with enemy robots to compete with other teams in robot soccer. Consequently, the

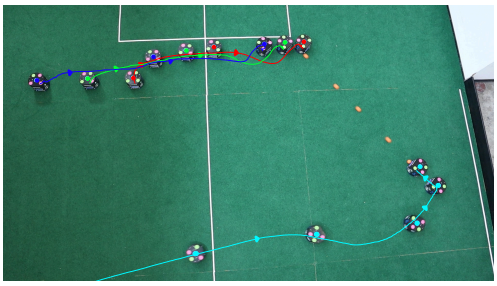


Fig. 5. Wall building scenario executed in the real live setup. The turquoise robot attacks the goal by driving to the ball and shooting it. The other three robots build a wall at the penalty area.

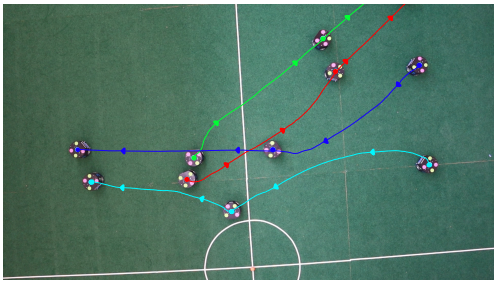


Fig. 6. Defending scenario where the blue and green robots move constantly on a predefined path. The red and turquoise robots defend the enemy robots against the ball which is placed in the center of the field.

planner did not avoid collisions completely, but it was able to avoid yellow cards due to collisions which are given if the collision velocity is $1.5\,\mathrm{m\,s^{-1}}$ or higher.

VI. CONCLUSION

In this paper, we presented a new approach to reactively controlling multiple robots simultaneously in a robot soccer application. We successfully implemented and adapted the CFs approach on real-world robots despite high measurement delays and low rate updates. Compared to previous implementations of CF-based planners, our planner is able to steer robots effectively and avoid collisions with other robots with high velocities and a low control rate of only $100\,\mathrm{Hz}$. Furthermore, it is able to locally solve conflicting plans from allied robots while needing significantly less computational resources than the CFP planner. The extensions to multiple moving goals and anti-goals with various shapes allow to precisely define the planners' behavior and describe high-level tasks. The capabilities of the planner were extended from executing simple point-to-point movements to actively executing all common tasks of robot soccer. However, the solution to local conflicting plans is always in favor of the planner that first detects the conflict and thus can be suboptimal for the whole team. We plan to extend our approach by implementing more sophisticated multi-agent strategies to solve these conflicts more effectively.

ACKNOWLEDGMENT

This research project would not have been possible without the support of the student initiative *luhbots* of the Leibniz University Hannover. We would like to thank all members of the initiative who contributed to the success of this project.

REFERENCES

[1] K. M. Lynch and F. C. Park, *Modern robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[2] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[3] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, 2012, pp. 1–6.

[4] B. P. Gerkey and K. Konolige, "Planning and control in unstructured terrain," in *ICRA workshop on path planning on costmaps*. Citeseer, 2008.

[5] L. C. Wang, L. S. Yong, and M. H. Ang, "Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment," in *Proc. IEEE Int. Symposium Intell. Control*, 2002, pp. 821–826.

[6] RoboCup, "Robocup federation official website," Aug. 2022, accessed: 2022-08-09. [Online]. Available: https://www.robocup.org

[7] SSL, "Small size league | robocup soccer," Aug. 2022, accessed: 2022-08-09. [Online]. Available: https://ssl.robocup.org/

[8] B. Lee, S. Lee, and G. Park, "Trajectory generation and motion tracking control for the robot soccer game," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1999, pp. 1149–1154.

[9] Q. Zhang, D. Chen, and T. Chen, "An obstacle avoidance method of soccer robot based on evolutionary artificial potential field," *Energy Procedia*, vol. 16, pp. 1792–1798, 2012.

[10] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The field d* algorithm," *J. Field Robotics*, vol. 23, no. 2, pp. 79–101, 2006.

[11] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[12] D. Kappler, F. Meier, J. Issac, J. Mainprice, C. G. Cifuentes, M. Wüthrich, V. Berenz, S. Schaal, N. Ratliff, and J. Bohg, "Real-Time Perception Meets Reactive Motion Generation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1864–1871, 2018.

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1985, pp. 500–505.

[14] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, 2003.

[15] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Proc. IEEE Int. Symposium on Industrial Electronics*, vol. 3, 2001, pp. 1530–1535 vol.3.

[16] J. Sfeir, M. Saad, and H. Saliah-Hassane, "An improved artificial potential field approach to real-time mobile robot path planning in an unknown environment," in *Proc. IEEE Int. Symposium on Robotic and Sensors Environments*, 2011, pp. 208–213.

[17] L. Singh, H. Stephanou, and J. Wen, "Real-time robot motion control with circulatory fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 1996, pp. 2737–2742.

[18] G. Garimella, M. Sheckells, and M. Kobilarov, "A stabilizing gyroscopic obstacle avoidance controller for underactuated systems," in *Proc. IEEE Conf. Dec. and Contr.*, 2016, pp. 5010–5016.

[19] M. Becker, J. Köhler, S. Haddadin, and M. A. Müller, "Motion planning using reactive circular fields: A 2D analysis of collision avoidance and goal convergence," *IEEE Transactions on Automatic Control*, 2023, early access. [Online]. Available: https://doi.org/10.1109/TAC.2023.3303168

[20] A. Ataka, H. Lam, and K. Althoefer, "Reactive magnetic-field-inspired navigation for non-holonomic mobile robots in unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 6983–6988.

[21] S. Haddadin, R. Belder, and A. Albu-Schäffer, "Dynamic motion planning for robots in partially unknown environments," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6842–6850, Jan. 2011.

[22] M. Becker, T. Lilge, M. A. Müller, and S. Haddadin, "Circular fields and predictive multi-agents for online global trajectory planning," *IEEE Robot. Autom. Lett.*, vol. 6, pp. 2618–2625, Apr. 2021.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960.

[24] F. Zeug, M. Becker, and M. A. Müller, "Dataset: Circular fields for highly-dynamic multi-robot applications likerobot soccer," 2023. [Online]. Available: https://doi.org/10.25835/i5vzko1b

[25] "IEEE standard for floating-point arithmetic," *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, 2019.

[26] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142 – 153, 2017.

[27] ER-Force, "Software framework of the SSL-team ER-Force," Apr. 2022, accessed: 2022-09-27. [Online]. Available: https://github.com/robotics-erlangen/framework