A Sampling-Based Algorithm for Multi-Agent Coordination in RoboCup Soccer

Filippo A. Spinelli[1,3], Nicole Ospelt[1], Chang He[1], Emilio Palma[1], Samuel Montorfani[1], Raffaele Soloperto[1,2], John Lygeros[1,2]

***Abstract*—Multi-agent decision making for autonomous robot soccer presents unique challenges, including real-time coordination, adaptive strategies, and efficient collaboration under dynamic conditions. In this work, we introduce a novel sampling-based algorithm designed to enable a team of humanoid robots to cooperatively coordinate during a soccer match. Our method solves via distributed optimization a mathematical representation of the problem, computing velocity references from generalized cost functions via gradient descent. It enables dynamic team coordination, adaptive offensive and defensive strategies, and quick responses to changing scenarios. Our method generalizes to any number of robots and has shown significant improvements in game-play and agent cooperation compared to standard state machine approaches. Simulation and hardware deployments further demonstrated robust performance with limited information and low-latency updates. We believe our work should lay the foundations for reliable and interpretable decision-making algorithms for multi-robot cooperation in real world.**

***Index Terms*—Multi-Robot Coordination, Robot Planning, Collective Behavior**

I. Introduction

To achieve autonomy in the domain or RoboCup (RC) Soccer, three main components are needed: *i)* the perception module, gathering information about the robot state and the environment, *ii)* the behavior module, processing this information in order to enable high-level decision-making, and *iii)* the motion control module, converting the high-level commands into joint trajectories. Our work focuses on the behavior component of such a complex software stack. This is traditionally addressed via State Machines (SMs) as shown in [1]: driven by pre-defined transitions, the robot's behavior evolves through different states, each associated with a set of actions. While being effective and easy to implement, such a solution restricts the ability of the robot to deal with unforeseen scenarios, and results particularly limiting for multi-agent decision-making. Alternative solutions based on distributed optimization exist [2], [3], able to include global information [4] and intra-robot communication [5] for better cooperation. Our work builds on top of the Feedback Equilibrium Seeking (FES) algorithm [6], [7], which extends Feedback Optimization (FO) [8], [9] to drive multi-robot systems. This method solves the optimal problem iteratively, in closed loop with the physical system. Compared to other approaches, this guarantees increased robustness to sensor noise, model uncertainty and environment disturbances.

This work is supported by the NCCR Automation.

Raffaele Soloperto is supported by the European Research Council under the H2020 Advanced Grant no. 787845 (OCAL).

[1]The authors are with ETH Zurich, Zurich, Switzerland.

[2]The authors are with the Automatic Control Laboratory, ETH Zurich.

[3]The author is with the Robotic Systems Lab, ETH Zürich.

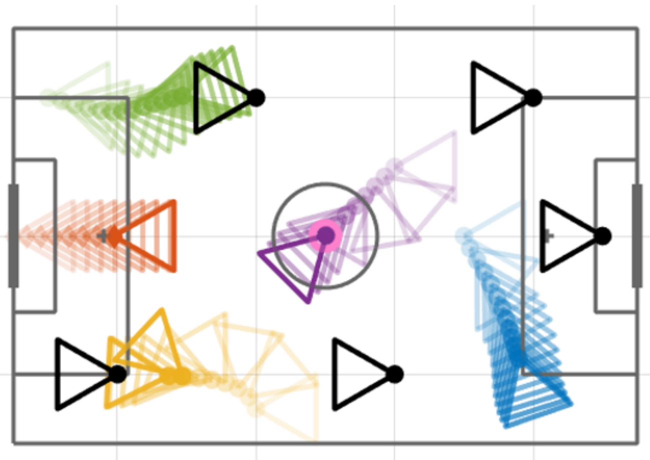

Fig. 1. Optimized robot trajectories. The goalkeeper (red) protects the goal based on the current ball position. The purple robot follows the shortest path to the ball. The other agents optimize the full set of costs, with defensive robots attracted by the opponents, while offensive ones aiming for free ranges.

Contributions

In this work, we propose a Gradient Descent (GD)-based algorithm controlling the poses of a robotic team in both offensive and defensive scenarios, steered to ensure optimal poses based on the current players' configuration on the field. Making use of FES, it provides velocity references leveraging both local and global information.

II. Problem formulation

Given an agent $i \in \{1, \ldots, N\}$, and its state $x_i[k] = \left[p_i^x[k], p_i^y[k]\right]^\top$, we define its discrete time dynamics as a point mass with input $u_i[k] = \left[v_i^x[k], v_i^y[k]\right]^\top$. Each agent beside the goalkeeper is dynamically assigned with a different role $R_i[k] \in R = \{\text{BF}, \text{FC}, \text{BS}\}$ accounting for to the global team state. Robots' roles are chosen among Ball Follower (BF), Field Covering (FC) and Ball Searcher (BS) at run time. Offensive and defensive tasks are assigned based on the current position only. We order all the active agents into time-varying teammates $A_{tm}[k]$ and opponents $A_{op}[k]$ index sets with the related state and role sets:

$$X_{A_{tm/op}}[k] = \left\{x_i[k] \mid i \in A_{tm/op}[k]\right\},$$
$$R_{A_{tm}}[k] = \left\{R_i[k] \mid i \in A_{tm}[k]\right\}.$$

III. Proposed approach

By using the FES algorithm [7] we compute $u_i[k]$ for each agent i with GD:

$$u_i[k] = -\alpha \nabla_{x_i} U_i\big(R_i[k], X_{A_{tm}}[k], X_{A_{op}}[k], x_{ball}[k]\big), \quad (1)$$

where U_i is the agent-specific cost, conditioned on the robot role R_i, the robots' states and the ball position x_{ball}. Figure 1 shows an exemplary result of the closed-loop behavior. The agent-specific cost at each step k is computed as the sum of:

Collision Avoidance: To discourage collisions with $I = \{A_{op}, A_{tm}\}$, we use a Gaussian penalty:

$$U_{CA}(x_i, x_I) = \frac{\alpha_{CA}}{\sqrt{2\pi}\sigma_{CA}} \sum_{j \in I, j \neq i} exp\Big(-\frac{\|x_i - x_j\|^2}{2\sigma_{CA}^2}\Big). \quad (2)$$

Ball Following: The BF role is assigned to the agent "closest in time" to the ball, using an MPC optimization. For this robot only, we penalize the distance from the estimated ball position, using the Huber-like cost (H), as proposed in [10], to mitigate local minima effects:

$$\begin{aligned} U_{BF}(x_i, x_{ball}) &= H(x_i - x_{ball}) \\ &= \alpha_{BF}\Big(\sqrt{\|x_i - x_{ball}\|^2 + \beta_{BF}} - \sqrt{\beta_{BF}}\Big). \end{aligned} \quad (3)$$

Field Coverage: A good strategy is to explore the field as much as possible. This is achieved by defining a Gaussian coverage function at each field position r, centered on each robot x_i:

$$g(r, x_i) = \frac{1}{\sqrt{2\pi}\sigma_{FC}} exp\Big(-\frac{\|r - x_i\|^2}{2\sigma_{FC}^2}\Big), \quad (4)$$

and integrating their differences over the field to penalize agents too close to each other:

$$\begin{aligned} &U_{FC}(x_i, x_{A_{tm}}) = \\ &-\alpha_{FC} \int_{x_{lb}}^{x_{ub}} \int_{y_{lb}}^{y_{ub}} \sum_{j \in A_{tm}, j \neq i} g(r, x_i) - g(r, x_j) dr. \end{aligned} \quad (5)$$

This cost inherently encodes repulsion from the field boundaries $(x_{lb}, x_{ub}, y_{lb}, y_{ub})$. It is computed by sampling discrete locations on the field, then used for the GD optimization.

Cooperative Behavior: To cooperatively attack or defend, the distance between the active teammates' center of mass and the ball position is penalized:

$$U_{CB}(x_{A_{tm}}, x_{ball}) = \alpha_{CB} \Big\| \frac{\sum_{i \in A_{tm}} x_i}{|A_{tm}|} - x_{ball} \Big\|^2. \quad (6)$$

This aggregative term inspired by [4], coupled with U_{FC}, allows the robots to strategically position on the field.

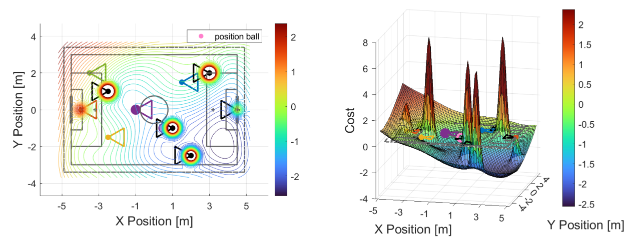

Fig. 2. 2D (left) and 3D (right) cost functions for the purple robot. We show the full cost for a 5vs5 scenario.

Offensive and Defensive Bias: Inspired by football tactics, we aim at tailoring the behavior to the field location. To let attackers pursuing the free space and defenders preventing goals, we include a Gaussian cost whose peak's sign and bias depend on the opponent's position:

$$\begin{aligned} &U_B(x_i, x_{A_{op}}) = \\ &-\frac{1}{\sqrt{2\pi}\sigma_B} \sum_{j \in A_{op}} \alpha_B(x_j) exp\Big(-\frac{\|x_i - x_j - \beta_B(x_j)\|^2}{2\sigma_B^2}\Big). \end{aligned} \quad (7)$$

If the opponent is in our half, the peak $\alpha_B(x_j)$ is negative and the 2D shift $\beta_B(x_j)$ is on the segment connecting the opponent and the goal center. Otherwise the peak is positive and the shift is zero. The overall cost can be seen in fig. 2.

IV. Results

The method, initially evaluated in a custom 2D Matlab simulation, has been deployed in a RC 3D simulator and on real robots. In the 3D simulator, we played matches against a baseline SM positioning behavior. The optimal coordination allowed to win on 75% of the scenarios, with an average controlled field, described via Voronoi regions [11], of 63%. It also considerably reduced the distance covered by each robot, hence limiting potential falls. The algorithm demonstrated good sim-to-real transfer, with adaptability to the changing environment. Operating in closed loop with the real system, the approach was robust to state estimation noise, delayed communication and temporary failures.

V. Conclusion

We presented a sampling-based algorithm which enables humanoid robots to perform team coordination during a football match. By optimizing velocity commands through GD on generalized cost functions, the method promotes adaptive strategies and adjustment to changing scenarios. The approach generalizes to any number of robots and maintains its properties even with limited information access and low-latency updates. It demonstrated improvements over standard state-machine approaches in both simulation and hardware implementations. Despite being more generalizable than SM, this method relies on cost tuning in order for the desired behavior to emerge. While a Multi-Agent Reinforcement Learning (MARL) method with sparse rewards may potentially alleviate this problem, we believe model-based optimization provides more guarantees and a more interpretable behavior for deployment in real scenarios.

References

[1] M. Risler and O. von Stryk, "Formal behavior specification of multi-robot systems using hierarchical state machines in xabsl," in *AAMAS08-workshop on formal models and methods for multi-robot systems*. Estoril, Portugal, 2008, p. 7.

[2] H. Jaleel and J. S. Shamma, "Distributed Optimization for Robot Networks: From Real-Time Convex Optimization to Game-Theoretic Self-Organization," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1953–1967, Nov. 2020.

[3] R. Yang, L. Liu, and G. Feng, "An overview of recent advances in distributed coordination of multi-agent systems," *Unmanned Systems*, vol. 10, no. 03, pp. 307–325, 2022. [Online]. Available: https://doi.org/10.1142/S2301385021500199

[4] G. Carnevale, A. Camisa, and G. Notarstefano, "Distributed Online Aggregative Optimization for Dynamic Multirobot Coordination," *IEEE Transactions on Automatic Control*, vol. 68, no. 6, pp. 3736–3743, Jun. 2023.

[5] A. Nedic and A. Ozdaglar, "Distributed Subgradient Methods for Multi-Agent Optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[6] G. Belgioioso, D. Liao-McPherson, M. H. de Badyn, S. Bolognani, J. Lygeros, and F. Dörfler, "Sampled-Data Online Feedback Equilibrium Seeking: Stability and Tracking," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Dec. 2021, pp. 2702–2708, iSSN: 2576-2370.

[7] G. Belgioioso, D. Liao-McPherson, M. H. de Badyn, S. Bolognani, R. S. Smith, J. Lygeros, and F. Dörfler, "Online feedback equilibrium seeking," *IEEE Transactions on Automatic Control*, 2024.

[8] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-Varying Convex Optimization: Time-Structured Algorithms and Applications," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2032–2048, Nov. 2020.

[9] A. Hauswirth, Z. He, S. Bolognani, G. Hug, and F. Dörfler, "Optimization algorithms as robust feedback controllers," *Annual Reviews in Control*, vol. 57, p. 100941, 2024.

[10] G. P. Meyer, "An alternative probabilistic interpretation of the huber loss," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2021, pp. 5261–5269.

[11] F. Aurenhammer and R. Klein, "Voronoi diagrams." *Handbook of computational geometry*, vol. 5, no. 10, pp. 201–290, 2000.