Context-based Coordination for a Multi-Robot Soccer Team

Francesco Riccio, Emanuele Borzi, Guglielmo Gemignani, and Daniele Nardi

Dept. of Computer, Control, and Management Engineering,
Sapienza University of Rome, via Ariosto 25, 00185, Rome, Italy
{riccio,borzi,gemignani,nardi}@diag.uniroma1.it

Abstract. The key issue investigated in the field of Multi-Robot Systems (MRS) is the problem of coordinating multiple robots in a common environment. In tackling this issue, problems concerning the capabilities of multiple heterogeneous robots and their environmental constraints need to be faced. In this paper, we introduce a novel approach for coordinating a team of robots. The key contribution of the proposed method consists in exploiting the rules governing the scenario by identifying and using "contexts". The robots actions and perceptions are specialized to the current context to enhance both single and collective behaviors. The presented approach has been largely validated in a RoboCup scenario. In particular, we adopt a soccer environment as a testing ground for our algorithm. We evaluate our method in several testing sessions on a simulator representing a virtual model of a soccer field. The obtained results show a substantial improvement of the team adopting our algorithm.

1 Introduction

In recent years, researchers managed to develop intelligent robots for a wide variety of environments outside research labs. Nowadays, robotic deployment reaches a broad range of fields, finding several applications such as in human-dangerous environment explorations, surveillance, or health care assistance. In such environments, often multiple robots are required to cooperate to achieve a higher effectiveness or carry out tasks that otherwise could not be completed.

In fact, Multi-Robot Systems (MRSs) present many advantages with respect to single robot systems. A MRS is to be considered more robust, with respect to system failures; scalable, depending on the environment specification and task requirements; and more efficient in performing a given task. Multi-Robot Systems have been widely studied in the framework of RoboCup competitions.

The purpose of this work is to present new approaches and methodologies to coordinate multiple autonomous robots and to guarantee their effective behaviors. Specifically, we develop our work in the soccer RoboCup scenario where a team of robots is required to coordinate to effectively play a soccer game. In this scenario, we present an algorithm that exploits the high level information of occurring situations to obtain a specific behavior in response of multiple environmental stimuli. The aim of this work is to provide a high level of knowledge about the current state of the world, allowing a team of robots to have a more effective way of perceiving the environment and the entities in it. The key contribution of this paper is an approach for modeling the context features of a particular environment and an algorithm for integrating different coordination techniques for a team of robots. The approach has been deployed on several simulated and real robots, including a team of humanoid NAOs. On these robots, we carried out multiple experiments to evaluate the effectiveness of our contribution.

In the remainder of the paper, we first present an overview of related work, focusing on past research on multi-robot coordination. Next, we describe our approach to coordination highlighting all of our contributions thoroughly. Then, we present an application of the approach to the case of a team of humanoid robots in a soccer scenario. This setting is then used to quantitatively evaluate the proposed approach. Finally, we conclude with a discussion of our contribution and remarks on future work.

2 Related Work

The coordination of multiple robots has been broadly studied during the last few years, considering multiple scenarios and heterogeneous agents that need to operate in a specific environment. In particular this problem has been broadly studied in the RoboCup soccer community where a team of robots needs to autonomously play a soccer game against another robotic team.

One of the first attempts to coordinate a team of heterogeneous robots was proposed by a joint project of seven different Italian universities. The ART-Azzurra-99 Team [6], later extended in [3], developed a coordination system able to efficiently coordinate heterogeneous agents in a team. Their approach relied on a task assignment technique. The algorithm automatically distributes tasks that the team need to accomplish, based on auction techniques.

An alternative algorithm is given in [2] where an asynchronous distributed system for task allocation which either relies on the perception of each robot or on a token passing approach in order to allocate the robots within the team.

Lou et al. [4] propose an improved algorithm for task allocation based on an auction system. They divide the set of possible tasks in subgroups and assign a task to each robot without violating precedence constraints among tasks.

Wiegel et al. in [9] propose a task allocation for the soccer middle-size league[1] based on utility estimations. First, they define the set of preferred poses for the team depending on the current situation, and then compute the utility values with respect to generate set of reference poses.

More recently, MacAlpine *et al.* proposed a more advanced form of robot coordination [5]. In this article, the authors introduce a formation system algorithm that is exploited within the 3D simulation league. The algorithm computes a global world model that is shared between the agents and is locally evaluated. After each evaluation each robot broadcasts the obtained result. The team is split in an offensive and in a defensive group and the role of each member is assigned depending on the ball position and the distance from specific positions.

Finally, additional solutions to the heterogeneous robot coordination challenge were proposed in [1, 7, 8]. These solutions respectively rely on an estimation of the world-state, an estimation of a mapping function between robots and tasks or between robots and roles. In these works the authors employ generic

[1] http://www.robocup.org/robocup-soccer/middle-size/

world-state evaluations [1], specific world rules [7] or utility estimation functions and artificial potential fields to position the robots within the environment [8].

Considering the analyzed approaches, we notice that they focus either on sharing encoded information among the team (*local estimation*) or on reconstructing a suitable interpretation of the world with respect to each single robot (*distributed world knowledge*). Conversely, our method focuses on integrating such approaches according to the environment model. In fact, we propose a coordination algorithm, based on both a distributed world knowledge and task-role assignments, as described in the next section.

3 Approach

Our approach relies upon two well known methods for coordinating a team of robots: *distributed task assignment* and *distributed world modeling.* In order to coordinate a team, distributed task assignment relies on the exchange between robots of meaningful task-related values. Generally, such task-related values are utility estimations with respect to a given task. Conversely, distributed world modeling exploits the direct exchange between robots of their internal world representation. The proposed approach aims at combining the robustness of the two approaches. In the rest of this section, we describe the two main components on which our approach relies upon, namely the Coordination System and the Context System.

3.1 Coordination System

The coordination system is in charge of generating a suitable mapping function between the set of acting robots R and the set of tasks T. We conceptually separate the coordination system in two main steps. First, we update a *distributed world model* in accordance with the *events* occurring during the game. Then, we exploit the generated world model to compute utility estimations that are used to assign tasks among the robots.

Distributed World Modeling The Distributed World Model (*DWM*) is defined as a dynamic global world knowledge about the current state of the environment and status of the task. The *DWM* is formalized by considering a set of partial models, each of which is a local representation of the world state for the i-*th* robot. Thus, given a team of robots $R = \{r_1, r_2, ..., r_n\}$, a *distributed world model DWM* is defined as the knowledge of the world reconstructed from a set of partial models $LM = \{LM_1, LM_2, ..., LM_n\}$. Formally, if for each robot r_i we define $LM_i(t)$ as the local model of the robot r_i at a particular time t, then we can define the distributed world model of the team as

$$DWM(t) = f(LM, t) \tag{1}$$

where f is a *reconstruction function* generating the distributed world model considering the partial models of each individual agent. The reconstruction function needs to be specified depending on the environment constraints and task specification. However, exchanging local models has an high computational cost and it is time consuming. Moreover, it assumes a reliable network condition which is hardly verified in real applications.

To overcome this issue, we design an event-based system which allows the robots to infer the local model for the i-*th* robot at time t by evaluating $LM_i(t-1)$ and the occurring events. Hence, we define the *model update function* $\psi(\cdot)$ which takes as input environment dependent events E and an estimation of the previous local models $\overline{LM}(t-1)$, returning the updated local models $\overline{LM}(t)$:

$$\overline{LM}(t) = \psi(E(t-1), \overline{LM}(t-1)) \quad (2)$$

where $E(t-1)$ are the events occurred at t-1.

Accordingly, we reformulate the *reconstruction function* as:

$$DWM(t) = f(\overline{LM}(t), t) \quad (3)$$

Task Allocation Depending on the status of the global world model DWM, we adapt the utility function of the team to maximize the performance with respect to the common goal. The key idea is that a static, unique utility function cannot fulfill the requirements imposed by the game in every situation. To achieve this flexibility, we develop a task assignment routine based on utility estimations, and as we will explain in the next section, is *context-dependent*. This routine is an instance of a marked based technique which evaluates at any time the configuration of the robots within the environment, generating the best association between robots and tasks. More specifically, given a set of tasks $T = \{\tau_1, \tau_2, ..., \tau_m\}$ for a team of robots R $= \{r_1, r_2, ..., r_n\}$, a *utility estimation vector* (*UEV*) can be defined as vector containing a list of estimations of "how good" a particular robot is for each task τ_i at a certain time. In other words, if we define $b_{i,j}(t)$ the estimation that the robot r_i computes for the role τ_j at time t, the UEV for such a robot can be expressed as:

$$UEV_i(t) = \left[b_{(i,1)}(t), \ \ \dots \ \ , \ b_{(i,m)}(t)\right] \quad (4)$$

Consequently, we can define *utility estimation matrix* (*UEM*) a matrix where each row i is the *UEV* for each robot r_i. This matrix is computed individually by each robot and it is built by gathering the *UEV*s coming from all the teammates. Formally, this matrix will have the following form:

$$UEM_i(t) = \left[UEV_1(t), \ \ \dots \ \ , \ UEV_n(t)\right]. \quad (5)$$

By considering the score of each robot and the current configuration of the distributed world model, given the score of each robot $b_{(i,j)}$, we can define a *coordination mapping function* Φ, which assigns to the task τ_j the robot with the highest score $b_{(i,j)}$, breaking ties randomly.

3.2 Context System

Our aim is to use contextual knowledge to increase the robot performance in accomplishing a given task. In our scenario, contextual knowledge is used to help the robots to evaluate the events and their effects: how they are triggered; how they modify the environment; how long their effects persist in time. In order to consider contextual information, we introduce in our approach a representation of contexts. Contexts, can be thought as specific configurations of the operational environment. For example, in a soccer game a context could be when the ball rolls out of the field and it needs to be put back into the game, or when the robots need to coordinate in environments with low-bandwidth for communication.

Our approach exploits the output of the contextual system to handle the events occurring during the game. More specifically, the context system outputs *contextual features*, which are used to weight events and their effects on the distributed world model. Formally, we characterize the context system as a function *CS* that takes in input sensory data D, internal robot states S, and external environment dependent information I. *CS* outputs contextual weights related to the notified events. Formally *CS* can be defined as:

$$CS : \; [\; D \;\times\; S \times\; I \;] \rightarrow C$$

where the vector of context weights $C \in \mathbb{R}^k$, with $k = \|E(t)\|$.

In our formulation such contextual features C are used to influence the regular operation of the coordination system in order to improve the efficiency in executing a task. Accordingly, at any time t the set of events is weighted as:

$$E_w(t) = C \cdot E(t). \tag{6}$$

At this point, by considering Eq. 2, we can influence the coordination system using each weighted event E_w, resulting in:

$$\overline{LM}_i(t) = \psi(E_w(t), LM_i(t)). \tag{7}$$

The context system influences the robot behavior by specializing their actions according to the current contexts. The key insight is that a more specialized and informed agent improves its performance. We exploit this concept in a RoboCup scenario to enhance the capabilities of a soccer robot team, but it could be applied to any coordination system.

4 Coordinating in the RoboCup Soccer Scenario

Our approach to coordination has been developed in the RoboCup Standard Platform League scenario. The approach has been deployed on a team of NAOs, which are commercial, autonomous, 25-DOFs humanoid robots. Such robots are equipped with a wide variety of sensors and actuators, including two CMOS cameras, multiple proximity sensors, four micro-phones, and two speakers. In the chosen scenario, a team of robots needs to coordinate in a 9x6 meters soccer field of the RoboCup Standard Platform League.

In this section, we describe in detail our approach applied to the RoboCup Standard Platform League scenario. Accordingly, we first introduce the modeling of context information related to the soccer scenario. Next, we illustrate how these contexts can be recognized during a soccer game. Finally, we describe how these contexts can be used to improve the coordination of a team of robots.

4.1 Representing Contexts

Defining and representing contexts and context information is a non-trivial task even in a simplified scenario such as a soccer game. To overcome this issue we propose a hierarchical structure, used to recognize possible contexts occurring in the soccer scenario.

In this setting, we formalize two different layers for properly representing contextual information, namely the *task-related* and the *environmental* layer. In the task-related layer, we encode a set of three basic contexts called *task-related contexts* (C_T):

- **Playing**: the robots know the current location of the ball, and the robot coordinate according to the default task-space comprising the common task in a soccer scenario, i.e. striker, defender, supporter and second supporter;
- **Search for ball**: the robots do not know the ball position and cooperate in order to minimize the time in locating the ball;
- **Throw-in**: the robots are searching for the ball but can modify their search strategy by exploiting particular rules governing the soccer scenario.

In the environmental layer, we instead characterize the world depending on the network reliability that allows us to define another two contexts, called *environmental contexts* C_E:

- **Network up**: the robots are in a suitable network condition, i.e. the messages exchanged among the robots are received in a fixed amount of time;
- **Network delayed**: the current network condition does not allow a reliable communication among the robots.

The environmental contexts do not affect the task-space, but actively influence the coordination system. For instance in a *network-delayed* setting the robots modify coordination parameters such as the *role-persistence*. This parameter is used to control how roles are swapped among active robots. Specifically, each robot waits a given amount of time before releasing the role and assuming the new role. This is used to avoid too frequent role swappings and to allow for a more robust task allocation during a game. Since the role-persistence is defined as a time interval, it is crucially important that messages coming from other robots are evaluated depending on quality of the network in order to consider possible delays and, consequently, misleading information. Accordingly, when the network communication is limited, the robots can adaptively change the amount of messages exchanged among the active robots to limit the traffic in the communication.

It is worth remarking that contexts within the same layer are mutually exclusive. For example, if the ball has been seen (i.e. we are in the Playing context), the team can be either in a network-up or in a network-delayed contexts. Therefore, in this scenario we define *context information* as a tuple of two elements, one for each layer, namely

$$C = \langle c_T,\ c_E \rangle$$

where $c_T \in C_T$ and $c_E \in C_E$ represent the context for the task-related layer and the environmental layer, respectively. Fig. 1 illustrates our multi-modal hierarchy for representing context information.

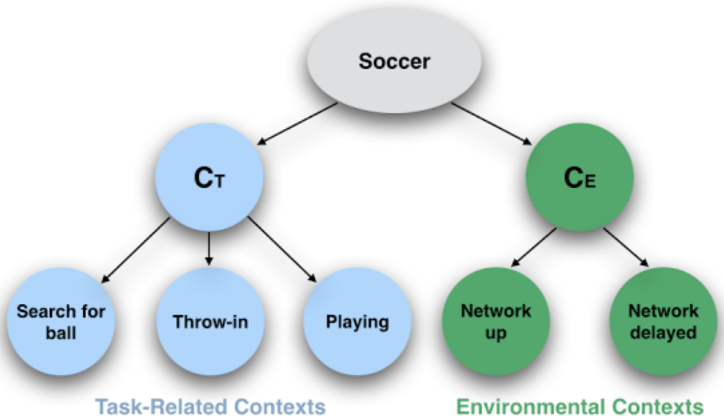

Fig. 1. Multi-Modal Context hierarchy in a soccer scenario.

4.2 Recognizing Contexts

As described in the previous section, during a soccer game we distinguish five different contexts. The three Task-Related Contexts (*Throw-In*, *Ball Lost*, and *Playing*) are recognized through the perceptions of the robots and the messages sent by the Game Controller to all the players. In particular, Throw-In considers the scenario in which the ball has rolled out of the field and is flagged by the Game Controller. Ball Lost is recognized only through the perceptions of the robots which have lost the sight of the ball. Finally, Playing considers the scenario in which the game is normally played and is assumed as the default context.

In multi-robot coordination the robots share useful information via wireless communication. However, a reliable network support is not always guaranteed in real applications. RoboCup competitions are not an exception. To this end, the two Environmental Contexts (*Network up* and *Network delayed*) are recognized based on multiple measurements locally and periodically performed by each robot. Specifically, each robot evaluates the current state of the network at each time step by considering the Round Trip Time (RTT) of the sent packages. The RTT is defined as the time elapsed between the moment in which a package is sent from a source A to a source B and the moment in which B acknowledges to A that the package has been received.

During the game we periodically compute the RTT among the robots pairwise. This measurement allows us to have an understanding of the quality of the network for single channel of communication (i.e., between two robots of the team). Additionally, by considering all the communications within the team, we are able to understand the *global network level* (GNL_{RTT}) by averaging the single channel estimations. This percentage measures the quality of the network depending on the RTT. Such a measurement is set to 100% if in average the RTT is smaller than a given threshold, or it is decremented if the acknowledge of a given package is never returned.

Finally, we define ranges of network reliabilities over the GNL_{RTT} in order to influence the robots' behavior. We manually set a discriminative value $\alpha =$ 50% for determining environmental contexts (Fig.1). We estimated the value of α on several testing sessions, noticing that a lost of packages associated to a $GNL_{RTT} \leq 50\%$ heavily affects the performance of the underneath coordination system. Accordingly, we define two ranges that result into the two different environmental contexts:

- Network up: the $GNL_{RTT} > \alpha$
- Network delayed: the $GNL_{RTT} \leq \alpha$

As a consequence, we influence the coordination system considering the contextual features generated by the two contexts. We identify a set of important parameters which are strongly network dependent, namely the rate of the packages sent, the weight of the events notified by a robot with a poor individual GNL_{RTT}, and the role-persistence used to switch roles among active robots. For instance, if the team realizes to be in the Network Delayed context, then we decrease the amount of exchanged data towards robots with a poor RTT and decrease the reliability of the information coming from them, as described in the following section.

4.3 Using Contexts

In this work we focus on the use of contextual information to more effectively coordinate a team of robots. Our main contribution is to demonstrate that by properly formalizing context and their influence within on the coordination system, we can decrease the number of variables to be optimized in a task allocation process, and most importantly, we decrease their operative numerical domain.

For instance, in our coordination system, we have a set of twelve roles that each robot can assume. Considering that each role cannot be selected by two robots simultaneously, without being able to recognize contexts we would have to search in a space composed by *12!* states. By using a contextual categorization of such roles, instead, we can split them in 3 different groups. Now, the search space for the task allocation within each context is composed by *4!* states. For large search spaces this can yield a big reduction in the complexity of the problem.

In the following, we describe how contextual information is used to shape the coordination system, both influencing the distributed world representation, and generating utility estimations, for the task assignment routine, that better satisfy the current context requirements.

Distributed World Modeling For the considered scenario, we adopt an occupancy grid model to represent the environment and reason about it. This structure is suitable to encode meaningful information and synchronize the robots views. More precisely, each cell of the grid encodes information such as teammate and opponent positions, the estimated ball position, and an estimation of the wireless signal level.

For instance, in the *search context* the grid is used to locate the next-most-likely position of the ball and to minimize the time needed to recover it. In this setting, we divide the field into cells with a fixed granularity assigning to each cell a probability score, which represents the likelihood of finding the ball in that particular area. The robots are displaced within the field in accordance with the top scoring cells, and coordinate to minimize the time spent in reaching a given area to be explored. Such areas represent the targets of the coordination algorithm and are constantly updated, while the players are roaming.

More precisely, while searching, the robots lower the score of the near cells and rerank the most promising ones. This procedure automatically generates clusters of *explored* and *unexplored* cells. The former is used to synchronize the distributed world model, while the latter is used to define the targets of the coordination system. The robots share the centroid of explored clusters to exclude recently controlled areas from the search. In this context, the reconstruction function of Eq. 1 is defined as

$$\forall robot_i \quad cell_j = \arg\min_i \{cell_{ij}\} \tag{8}$$

where $cell_j$ is the score of the j-*th* cell in the DWM, which represents the minimum score for the j-*th* among all the partial local models of the robots LM_I. Such a score has a default value for each context and cell, which is accordingly modified during the execution of the searching task.

Task Allocation The robots are coordinated using a DTA based on utility estimation. Therefore, we define a utility function, according to Eq. 5, that given a robot and a task, it returns a score representing how suitable the input robot is for the input task. Further, depending on the current context, the utility function changes to fulfill the requirement of the environment and the current task according to Eq. 7.

Let us assume to be in a playing context. In this case, the utility function takes into account the euclidean distance between the ball and the robot; the distance between the robot and a target position; the robot orientation; the elapsed time since the current role was assigned; and a bias term for each robot (such bias is used to solve ambiguous situations). This function is defined as:

```
uv(i,j) = penalty * playing_utility_function(i,j)
```

where $uv(i,j)$ is the utility score for the j-*th* role with respect to the i-*th* agent and *penalty* is a parameter used to prioritize the most important tasks. Each weight has been empirically calculated after several testing sessions and the *penalty* variable is used to assign negative rewards to fallen or penalized robots. The players share their utilities, and once their utility matrices are complete,

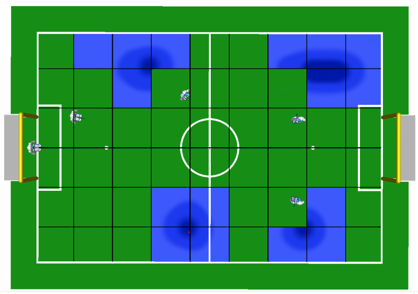

Fig. 2. Distributed World Model in the soccer scenario.

they can decide the best role to assume. The robots check the matrix column-wise and assign to the j-*th* task the robot with the maximum score.

Similarly, in the searching contexts the robots use exactly the same criterion to allocate tasks with respect to the areas to look for the ball. The utility function takes the following form

```
uv(i,j) =  searching_utility_function(i,j)
```

where $uv(i,j)$ is the utility score for the j-*th* centroid with respect to the i-*th* agent. Due to the new utility function, we can minimize the time for exploring most promising areas.

In order to guarantee more robustness to the task allocation, a role-time-persistence has been introduced. However, this choice imposes the usual trade-off between stability and reactivity. To efficiently adapt the role-persistence, we exploit the environmental context. In particular, we adapt the role-persistence relying upon the current configuration of the network.

5 Experimental Results

The algorithm has been extensively tested in a virtual RoboCup-dedicated environment developed by the B-human Team[2] and it has been deployed on a team of Nao robots. In order to show the improvements of our coordination algorithm, we set up several experiments with different configurations. Our aim is to highlight specific features of the algorithm, and to test the thesis behind our coordination approach (i.e. the fact that a properly informed coordination can considerably enhance the performances).

In the first setting, we demonstrate the improvement of our system with respect to a *non-context-aware* approach. In this configuration, we deploy two teams that share the same high-level behaviors, one featuring the context-based coordination (*blue team*) and the other modeling a unique utility estimation for the whole testing session (*red team*). Specifically, the red team does not modify its utility estimations, if the ball is not seen or if the ball rolled out of the field.

In this setting, we measured the cumulative time during which the ball was not seen by the team in 10 minutes of the game. In Fig. 3 (a) we can notice that the recovering time is considerably reduced when deploying the proposed approach. Further, it is worth highlighting the importance of different levels of information in the context system. In a Throw-In context, in fact, the ball is subject to specific rules and the performance can be further enhanced.

Instead, Fig. 3 (b) highlights the robustness of our approach with respect to the robots' field of view. In this setting, we again measured the cumulative time during which the ball was not seen by the team during a 10 minutes game. In this experiment, we varied the robots' field of views to 2, 3 and 9 meters. It is worth noticing that both teams improve their performance. However, the context-coordination preserves a better profile in all the configurations.

Finally, in order to show the effectiveness of our approach in managing network *contexts* we report the stability of the role switching with respect to the quality of the global network level GNL_{RTT} introduced in the previous section. To this end, we traced the network GNL_{RTT} in a real gaming session, and then

[2] http://www.b-human.de/

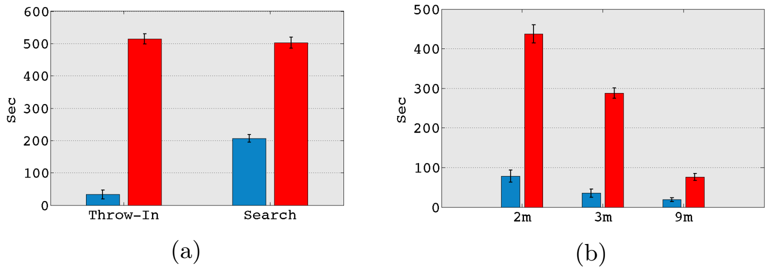

Fig. 3. The blue team features the context-coordination, while the red team is not context aware. On the y-axis, the averaged cumulative sum of the time interval in which the robots do not see the ball is shown, while on the x-axis, we report the results for two different contexts (a), and three depth values for the robots field of view (b).

we run several experiments by varying the simulated network stability with respect to the logged GNL_{RTT}. This, allows us to have more reliable simulated testing session and have a realistic profile of the GNL_{RTT} during a game. In such testing sessions, we report the error in meters that each robot has with respect to the assigned role. In this setting, both the blue and the red team features the context-coordination, however, the blue team role-persistence that controls the role switching is changed depending on the GNL_{RTT}, while the red team switches roles according to a fixed role-persistence threshold. Our goal is to generate a more robust behavior when the network has a poor reliability, and simultaneously, a more reactive role mapping when the team is experiencing a good communication setting. As in the previous experiments, the tests have been carried out in multiple sessions of 10 minutes of an SPL game. Fig. 5 shows the results obtained in a testing session. We report, for each role, the error in meters by computing the average of the error that each robot has during the simulation, with respect the default position of the assigned role.

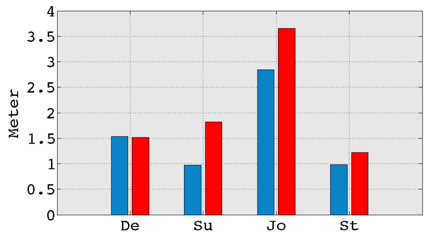

Fig. 4. The y-axis shows the error between the robots and the default position for the assigned role, while the x-axis reports the set of roles: Defender (De), Supporter (Su), Jolly (Jo) and Striker (St). The blue team adapts its role-persistence depending on the GNL_{RTT}, while the red team adopts a static role-persistence time threshold.

In this setting the two team feature the same coordination system and the performance is only conditioned by the role-persistence policy. Also in this configuration, we notice an improvement in formalizing environmental contexts as the network status. In our opinion, such context-related information is a starting point for handling network issues, and it needs to be better investigated in order to further improve team coordination.

6 Conclusion

In this paper we have presented a novel method for coordinating a team of robots. Starting from the idea that a more informed team will eventually show improved performance during task execution, we presented an approach that integrates utility estimations and a distributed world knowledge to come up with a mapping of robots to roles. The proposed method presented extracts context information to influence and modify the coordination rules and select the most suitable configuration in accordance to the current situation. Given the previously described experimental results, we are able to state that the Context-based coordination provides the expected improvements. Indeed, the specialization and the adaptation of the coordination algorithm significantly increases the performances of a team of robots.

Considering the results obtained with this approach, our intent is to generalize the method for addressing the problem of *"Multi-Robot coordinated search and target localization"*. The main idea is to deploy a coordinated team of robots to localize multiple targets (e.g. lost objects, control malfunction infrastructures, victim assessments) in an arbitrary environment and to improve the execution of the current robots' tasks by exploiting any kind of information that can help the robots to specialize their search and to improve their performances.

References

1. Abeyruwan, S., Seekircher, A., Visser, U.: Dynamic role assignment using general value functions
2. Farinelli, A., Iocchi, L., Nardi, D., Ziparo, V.A.: Task assignment with dynamic perception and constrained tasks in a multi-robot system. In: (ICRA). pp. 1535–1540. Barcelona, Spain (2005)
3. Iocchi, L., Nardi, D., Piaggio, M., Sgorbissa, A.: Distributed coordination in heterogeneous multi-robot systems. Autonomous Robots 15(2), 155–168 (2003)
4. Luo, L., Chakraborty, N., Sycara, K.: Multi-robot assignment algorithm for tasks with set precedence constraints. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. pp. 2526–2533. IEEE (2011)
5. MacAlpine, P., Barrera, F., Stone, P.: Positioning to win: A dynamic role assignment and formation positioning system. In: Chen, X., Stone, P., Sucar, L.E., der Zant, T.V. (eds.) RoboCup-2012: Robot Soccer World Cup XVI. Springer Verlag, Berlin (2013)
6. Nardi, D., et al.: ART-99: Azzurra Robot Team. pp. 695–698 (1999)
7. Stone, P., Veloso, M.: Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. Artif. Intell. pp. 241–273 (1999)
8. Vail, D., Veloso, M.: Multi-robot dynamic role assignment and coordination through shared potential fields. In: A. Schultz, L.P., Schneider, F. (eds.) Multi-Robot Systems. Kluwer (2003)
9. Weigel, T., Gutmann, J.S., Dietl, M., Kleiner, A., Nebel, B.: Cs freiburg: coordinating robots for successful soccer playing. Robotics and Automation, IEEE Transactions on 18(5), 685–699 (2002)