SWITCH! Dynamic roles exchange among cooperative robots *

Carlos E. Agüero, Vicente Matellán, José M. Cañas, and Víctor M. Gómez

Universidad Rey Juan Carlos
Tulipán s/n, 28933, Móstoles (Madrid) - Spain
{caguero,vmo,jmplaza,vmanuel}@gsyc.escet.urjc.es ,
WWW home page: http://www.robotica-urjc.es

Abstract. In this article we propose a dynamic role exchange algorithm for mobile robots. The heuristics used to alternate among the different roles is based on utility functions. We have utilize the Sony Aibo ERS-7 for testing our design and implementation. The RoboCup *4-legged* competition environment has been chosen for proving our work.

1 INTRODUCTION

RoboCup is a very challenging and motivating robotic competition where under a simple soccer match, are hidden lots of research in several areas.

In this paper we will focus on the 4-legged league [1]. Teams in this league are made by four Aibo robots [2]. In this competition, communication among robots is allowed (since 2002 edition held in Fukuoka, Japan). Cooperative behaviors can benefit from communications in many ways: Improving accuracy to locate objects by adding a global shared model [3], to allocate dynamically player roles [4].

Dynamic multi-robot coordination had already been explored by other teams. Early work on cooperation and coordinated positioning in RoboCup was proposed in [5]. Cooperation among heterogeneous robots has been explored in [6]. In [7] the ball position is determined by a probabilistic integration of all ball perceptions coming from the players. [8] suggested a potential field approach for managing robots constructed with shared information. Newcastle university team [9], finalist in Osaka 2005, shares ball location helping robots to find the ball quickly.

In this paper we present our work in this area focusing on task allocation. Dynamic task allocation allows a team to divide its main objective in a couple of sub-objectives more specialized and adapted to the location of each of the teammates and the strategy of the team. Section 2, 3 and 4 details our contribution to role allocation problem and strategy decisions for roles in RoboCup. Section 5 analyzes experiments and results we have got. Finally, section 6 summarizes our conclusions using coordination in robot teams and using our algorithms.

* This research has been partially sponsored by grants *DPI2004-07993-C03-01, DPI2004-07993-C03-02* and *S-0505/DPI/0176* corresponding to *Ditpa* and *Robocity 2030* projects respectively

2 ROLES SPECIFICATION

Several teams [10], [9] attending RoboCup events in the last years have used a set of roles assigned to each robot instead of using a single role for all players of the team. The use of a collection of roles allows a team to organize more easily all responsibilities involved in the task to resolve.

The roles can be assigned to each robot in an statical or in a dynamical way. Dynamic role allocation benefits, from example, from opportunistic situations like fast ball changes along the field, penalized players[1], or failures in some robot.

We have considered four main roles in the RoboCup domain: *Goal-keeper*, *Striker*, *Defender*, and *Supporter*. *Goal-keeper* is the only role assigned statically. The main reason to have a single player to the *Goal-keeper* role is that the rules do not allow that players enter in its own goal area (like in the hand ball). The rest of roles can be exchanged among robots according to game conditions.

Next, we are going to describe the objectives of each role and the advantages we have obtained using dynamic role allocation:

- ***Goal-keeper:*** Its goal is to protect its own goal from shots by the other team players. Also, it should rest in its own area.
- ***Striker:*** It tries to get the ball and to carry it, or to kick it, towards the net. When the other team have got the ball, it tries to recover it actively (going after the ball).
  None of the other roles are devoted to get the ball. This approach has one implicit advantage: It avoids *pushing* among players of our team, that is explicitly penalized in the rules. If a team does not have a ball booking mechanism [11] or similar, all robots will finish bumping into each other, and its players will be penalized.
- ***Defender:*** Its goal is to intercept the ball if an opponent kicks it to its net. Furthermore, it should stand in the way of the opponent and should try to hide the net preventing the opponent to kick the ball.
  Another implicit consequence of the *Defender* role is that one robot of the team always remains in a position near its own net. This fact is very useful taking into account that the ball quickly moves from side to side. We have always one robot covering its defending half of the field.
- ***Supporter:*** The function of this role is to assist the striker in its path, and to cover the maximum amount of field in case the ball will be kicked in the wrong way.
  The main contributions of this role are to recover the ball if the kicks made by the striker do not go in the good direction, and also to maintain a good position for future passing kicks.

3 UTILITY AND HEURISTIC FUNCTIONS

In this section we are going to introduce some concepts needed to explain how the roles are dynamically assigned. Our role allocation algorithm will be based

[1] Penalized players are removed from the field for 30”

in heuristic functions. Those functions evaluates some parameters like the ball distance, localization, etc. and obtains a value for each role. We will call this value *utility*. Utilities will be calculated periodically and roles will be assigned to robots in a priorized way according to the values obtained.

A general definition for *utility* is “value to estimate the cost of executing an action”. In our approach, utility is employed to evaluate the degree of adaptation of one role to one robot in a particular game conditions.

In our proposal *utilities* will be individually computed by each robot as the weighted sum of several factors: Distances to ball, nets, etc. In order to describe those factors, we will use the formalism described in [4]:

- Let $I_1, ..., I_n$ be the set of n robots.
- Let $J_1, ..., J_n$ be the set of n priorized roles and $w_1, ..., w_n$ their relative weights.
- Let U_{ij} the nonnegative utility of robot I_i for role J_j, $1 \leq i, j \leq n$.

1. Update Utilities
2. Look for robot with larger utility for *Striker* role
3. Look for robot with larger utility for *Defender* role
4. Look for robot with larger utility for *Supporter* role

Update Utilities step computes a matrix with all combinations among robots and roles according to the heuristic described below. The next steps selects the more suitable robot for the *Striker, Defender and Supporter* roles in this order.

The heuristic functions used to allocate the roles developed are the following:

- $U_{i,Striker} = D_{I_i,Ball} + (\theta_{I_i-Net} - \theta_{Ball-Net}) + REC$
- $U_{i,Defender} = D_{I_i,Own\_net} + REC$
- $U_{i,Supporter} = D_{I_i,Opp\_net} + REC$

$D_{I_i,Ball}$, D_{I_i,Own_net} and D_{I_i,Opp_net} are the distances to the ball, own net, and opponent net. θ_{I_i-Net} is the difference in orientation between the local orientation of the robot and the orientation needed for focusing the net. $\theta_{Ball-Net}$ is the angle between the ball and the opponent net. REC (Role Exchange Cost) is also added to prevent excessive roles exchanges if only small changes have happened in the environment. This factor provides some kind of hysteresis to the system.

Next are the equations to select the appropriate robot for each role. In this paper we suppose that the order in roles assignment is *Striker*,*Defender* and *Supporter*.

1. $Utility_{striker} = min(U_{i,striker})$, $\forall i \in (1..n)$
2. $Utility_{defender} = min(U_{i,defender})$, $\forall i \in (1..n) \wedge i \neq Robot_{striker}$
3. $Utility_{supporter} = min(U_{i,supporter})$, $\forall i \in (1..n) \wedge i \neq Robot_{striker}, Robot_{defender}$

Every robot updates its utilities periodically, and broadcasts this information to its teammates. We will refer to this information as *coordination information*. The information sent is its own location, and an estimation of its distance to the ball. *Coordination information* is sent at 5Hz.

When one robot receives *coordination information* it updates data associated to the corresponding robot in its global model. This global model stores position of the teammates and, combined with the position of the ball is used to calculate utilities functions.

4 ROBOT POSITIONING

In the previous section we have explained our mechanism for selecting roles according to the game situations. Other interesting issue is what should a robot do if it is running *Defender* or *Supporter Role* (*Striker* always goes for the ball). In our proposal, *Defender* and *Supporter* go to a specific position. In this section we describe how the players calculate their "ideal position" according to the roles they have been assigned.

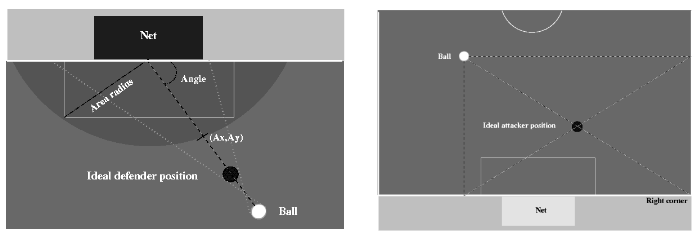

Fig. 1. Detailed points of interest calculating ideal posititions for roles

4.1 *Defender* positioning

Defender should stay in a position near its own net in order to defend its goal. We have developed an approach that places *Defender* between the ball and the middle of its own net. If an opponent hits the ball and the ball travels in the net direction, the defender must be in its trajectory.

$$\begin{aligned} angle &= arctan(ownNet_y - ball_y, ownNet_x - ball_x) \\ Ax &= -cos(angle) * areaCorner \\ Ay &= \frac{FieldLength}{2} - sin(angle) * areaCorner \\ ball2aproxArea &= distance(A, Ball) \end{aligned}$$

Illegal defense must be taken into account when the team is defending. This penalty is applied when a defending robot goes into its own area. In order to prevent our *Defender* from entering in its own area, we have approximated the area penalty to a semi-circumference.

The *Defender* ideal position will be the center of the segment delimited by the ball and point A (see figure 1 left). This position (DEF_x,DEF_y) is calculated using the next equations:

$$DEF_x = -cos(ang) * (areaCorn + \frac{ball2Area}{2})$$
$$DEF_y = \frac{FldLgth}{2} - sin(ang) * (areaCorn + \frac{ball2Area}{2})$$

4.2 *Supporter* positioning

Ideal position is the middle point of the quadrant delimited by the ball and the fairest corner of the opponent side (as shown in figure 1 right) where we can see this quadrant). This can be calculated as:

$$D_{lateral} = Max(dist(ball_x, lat_{left}), dist(ball_x, lat_{right}))$$
$$D_{goalLine} = \frac{FieldLength}{2} + ball_y$$
$$SUPPORT_x = ball_x \pm \frac{D_{lateral}}{2}$$
$$SUPPORT_y = ball_y - \frac{D_{goalLine}}{2}$$

This position guarantees that *Supporter* will never stand in the way of striker and the opponent net. Its goal is to support the striker and, in case a kick does not go in the right direction, to reduce the time to go for the ball.

5 EXPERIMENTS

TeamChaos [12] code is composed by the source code of the program that runs into the robots and a suite of tools. One of this tools is the team simulator. We have tested what are the differences if our team uses *Switch coordination* or not.

5.1 Playing with *Switch coordination* in simulator

Using dynamic role allocation, experiments show that the game is more dynamic. The main reason is that we give the biggest priority to take the ball. In simulator we have seen that always the better placed robot wants to go for the ball. If we drag the ball to different positions on the field, role reallocation acts quickly and robots change their positions. In consequence, the ball is take it more quickly than experiments without *Switch coordination*.

In order to test the real advantages of the dynamic role allocation, we have made six matches in our simulator, three of them with coordination and the others without *switch*. During the simulation there are not any opponents and each match has 10 minutes. Every time a player scores, the ball is taken to the middle of the field. In table 1 we can see the results of the simulations.

Match #	Goles with *Switch*	Goles w/out *Switch*
1st	3	2
2nd	4	3
3th	4	3

Table 1. Results of matches without opponents

Non cooperative team has a clear problem: All players go to the ball. There are not any advantage playing with one player or playing with three players. In this case, more players involve more obstructs between them.

Coordinated team offers a very more ordered play with the three attacker players spreaded on the field. With less robots standing in the way of others, there are more time to carry the ball inside the net and if the pushing rule would be applied they would have been removed from the field.

Although there is not a very large improvement in goals, the overall behavior of the team is much more organized. When the code of our *Striker* will be more tuned, we hope the results with and without coordination will drastically emerge.

5.2 Playing with *Switch coordination* with real robots

We have also tested *Switch coordination* in real robots. We have used a team of three Sony AIBO ERS-7 using the framework of the *TeamChaos* RoboCup team [12]. First, we have tested the operation of the assigning protocol for the *striker* role.

We have hold down the three robots in boxes. We want to show that only the robot with best utility and only one goes for the ball. If we put the ball in front of one of the robots, only the closest robot goes for it. If we move the ball and put it close to another player, this one takes the role of *striker* and it moves its legs for catching the ball. Note that for this experiment, *defender* and *supporter* roles are running a still behavior. You can see a movie for this experiment in *http://gsyc.es/c̃aguero/Switch-Pre.avi*.

6 CONCLUSIONS

We have developed a basic coordination mechanism among members of a multi-robot team. Localization and local ball estimation are the elements shared. Combining periodically the information received with local information, each robot updates a global model of the environment. Using coordination we have also got a very good way to identify the rest of the team members. We can not do previously because all teammates look alike.

Using the global map, an strategy layer uses the shared information to allocate roles. We have used a simulator to choose several heuristic functions and factors. We have observed the behavior in opportunistic situations dragging the ball and releasing it in other point. Robots quickly exchanges their roles according to the new scenario. However, when doing little changes in the environment, robots are able to cushion them maintaining their roles and improving the dynamism of the system.

Finally, ideal positions for each role are one option chosen by our team that has worked very well in our tests. Movement of the *Defender* robot is very smooth without big jumps between positions, preventing this critical player from moving constantly.

Supporter robot is less critical so it moves more than *Defender* covering the attacking field. Tests have shown that if a kick fails, *Supporter* goes for the ball quicker than previously implementations without coordination. Before using cooperation, the three players were closer all together and now they are expanded in a more intelligent way around the field.

References

1. Legged Robocup Federation: Robocup Four-Legged League. http://www.tzi.de/4legged/bin/view/Website/WebHome (2006)
2. Sony: Sony Global - AIBO Global Link. http://www.sony.net/Products/aibo/ (2006)
3. Roth, M., Vail, D., Veloso, M.: A world model for multi-robot teams with communication. In: IROS-2003. (2003)
4. Gerkey, B., Mataric, M.: On role allocation in robocup. In: RoboCup 2003: Robot Soccer World Cup VII, 2004, Springer-Verlag (2004) 43–53
5. Stone, P., Veloso, M., Riley, P.: The CMUnited-98 champion simulator team. Lecture Notes in Computer Science **1604** (1999) 61–75
6. Castelpietra, C., Iocchi, L., Nardi, D., Piaggio, M., Scalzo, A., Sgorbissa, A.: Communication and coordination among heterogeneous mid-size players: ART99. Lecture Notes in Computer Science **2019** (2001) 86–95
7. Weigel, T., Gutmann, J.S., Dietl, M., Kleiner, A., Nebel, B.: CS Freiburg: Coordinating Robots for Successful Soccer Playing. IEEE Transactions on Robotics and Automation **18(5)** (2002) 685–699
8. Vail, D., Veloso, M.: Dynamic multi-robot coordination. In: Multi-Robot Systems. Kluwer (2003)
9. Laboratory, N.R.: The NUbot 2005 Team Report. www.robots.newcastle.edu.au (2005)
10. GermanTeam: http://www.germanteam.org/ (Team report, 2005)
11. Agüero, C.E., Martín, F., Martínez, H., Matellán, V.: Communications and basic coordination of robots in teamchaos. In: Actas VII Workshop de Agentes Físicos. (2006) 3–9
12. TeamChaos: Team Chaos roboCup Team. http://www.teamchaos.es (2006)