Cooperative Behaviour of specific tasks in multi-agent systems and robot control using dynamic approach

Fernando Ribeiro, Ivo Moutinho, Nino Pereira, Fernando Oliveira, José Fernandes, Nuno Peixoto, Antero Salgado

Grupo de Automação e Robótica, Departamento de Electrónica Industrial, Universidade do Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
fernando.ribeiro@dei.uminho.pt, ivo@aeiou.pt, ninopereira@clix.pt, fremaxeei@gmail.com, joseluisfernandes@gmail.com, nunopeixoto35242@hotmail.com, anterosalgado@aeiou.pt

Abstract. In order to foster research and development in a multi-agent robotic environment three fundamental improvements on the robots need to be carried out: a) a very reliable and robot control which works at high speeds and a dynamic approach is described in this work; b) Cooperative behaviour is very important since without it there is no ball pass, and that is becoming more and more necessary; c) Upwards kick, since traditional horizontal kickers are already very common. Other improvements were carried out in the robots but due to lack of space in this paper are not described. This paper describes how these three issues were tackled by the MINHO team and shows their next directions.

1. Introduction

These robots are completely built in Minho Lab, and improved every mechanically, electronically and in terms of software. They are first designed using a CAD system to check its assemblability (see **Fig. 1**). This work describes the three most important issues to improve game play quality. The robot control needs to be very reliable because robots are moving faster and faster and they have to avoid fast moving obstacles and obey orders immediately. A dynamic approach developed by Estela Bicho [1] is adapted for these robots and described in this paper. Since the teams are putting more players on the field, there is a trend to make ball pass rather than only dribbling the opponent and a description is made. Rules allow a kick upwards and Minho team developed a new solution in terms of mechanics and electronics.

Fig. 1. CAD drawing of the new robot

2. Robot control using a dynamic system

2.1 Introduction

The robot control motion uses a dynamic system that calculates the navigation direction and velocity. This system has as main objective making the robot able to move in a dynamic way to the target located in any point of the field. The target position can change along the time in most cases, like the ball, and this target is defined by the game strategy.

With this system it is possible to instruct the robot to catch the ball, and take it to the goal or to go to a known strategic position, being only necessary to determine that position according to the game strategy. After that, the dynamic system is responsible for making the objectives accomplished. Moreover, adding the obstacles to the system, the robot is able to avoid any obstacle until getting to the target.

2.2 Navigation direction control

The rotation movement around itself is generated by continually giving values to the navigation direction. One range of values $f_{robot}(t)$, is generated by a dynamic system formulated by one differential equation where the state variable is the robot navigation direction.

$$\frac{dj_{robot}}{dt} = f_{t\,\arg et}\left(j_{robot}\right) \qquad \textbf{(1)}$$

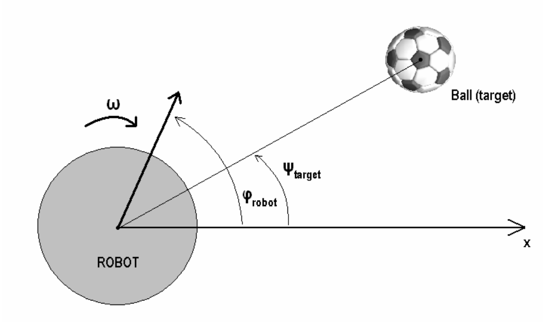


Fig. 2. Target dependent robot navigation direction

In **Fig. 2**, the black arrow points the robot front side. There is no dependence of external references, since the differential equation represents the difference between f_{robot} and $?_{target}$,in other words, the angle between the target and the robot.

The vectorial field in this dynamic system is built by one force that gives the desirable value to the navigation direction. The force has three parameters:

- Attractor, value that specifies the navigation direction, ? target.

- Attraction intensity.
- Values range of Navigation direction under what the force takes effect.

The differential equation that defines the dynamic system of the angular velocity control is:

$$w(t) = \frac{dj_{robot}}{dt} = -w_{max.} \cdot \tanh\left(\left(j_{robot} - y_{target}\right) \cdot g\right) \qquad \textbf{(2)}$$

? – Robot rotation angular velocity
f $_{robot}$ – Robot navigation direction
? $_{max.}$ – Robot rotation maximum speed
? $_{target}$ – Target direction
? – Target attraction intensity

Note: $\Delta a = j_{robot} - y_{target}$

Like the chart in **Fig. 3**, the objective is that the difference between the navigation direction angle and the target angle becomes zero, and the angular velocity becomes zero, which means that the robot is stopped and turned to the target.

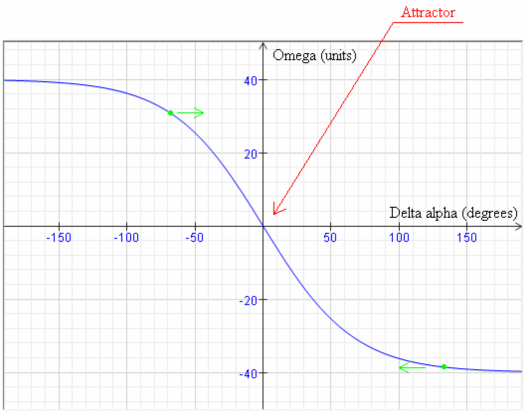


Fig. 3. Robot navigation direction graphical representation

The navigation direction or the target directions are irrelevant. Only a difference between them will generate one positive or negative angular velocity that will decreases until the direction is the same (target and the navigation direction). In that case, the system stabilises and the robot stops.
The angular velocity marked in the graphic (omega), is an input parameter to the function of robot traction motors and have a range from 0 to 40 (no specific unit).
One very important parameter is the **?**, because it specifies the attraction intensity to the target, allows defining which speed changes the angular velocity, the robot response time. But this value must not be very high, because the system will be instable. The equation implemented in the robot that can be seen in **Fig. 3**, was:

$$w(t) = -40 \cdot \tanh(\Delta a \cdot 0.015) \quad \textbf{(3)}$$

2.3 Linear velocity control

The robot linear velocity is continuously generating values to its position, since starting point in a straight line until the target. This series of values, $x_{robot}(t)$, are generated by a dynamic system formulated by the follow differential equation:

$$\frac{dx_{robot}}{dt} = f_{t\arg et}(x_{robot}) \quad \textbf{(4)}$$

The differential equation that defines the dynamic system of the linear velocity control is as follows:

$$v(t) = \frac{dx_{robot}}{dt} = -v_{\max.} \cdot \tanh\left(\left(\left(x_{robot} - x_{t\arg et}\right) - s\right) \cdot l\right) \quad \textbf{(5)}$$

v –Robot linear velocity
$\mathbf{x_{robot}}$ – Robot position in a straight line to the target
$\mathbf{v_{max.}}$ – Robot maximum linear velocity
$\mathbf{x_{target}}$ – Target position in a straight line to the robot
s – Security distance to the target
? – Attraction intensity to the target

Note: $\Delta d = x_{robot} - x_{t\arg et}$, (distance to target)

As it can be seen in the following chart **Fig. 4**, when the difference between the robot and target position are equal to the safety distance the robot must stop.

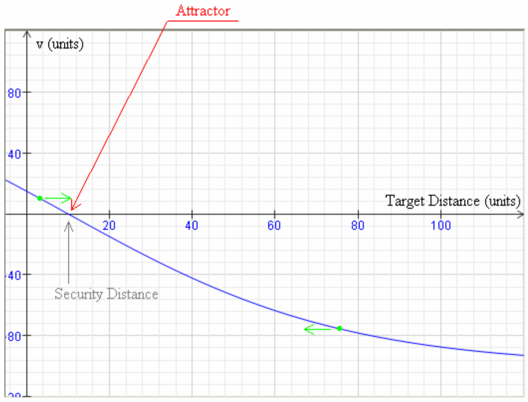


Fig. 4. Robot linear velocity graphical representation

The linear velocity indicated on the chart (v), is an input parameter to the function of robot traction motors and ranges from 0 to 100 (there is no specific unit). The distance to the target also ranges between 0 and 100 values.
One important parameter is **?**, because it allows specifying the attraction intensity to the target, meaning the speed change ratio (acceleration) which also represents the robot response time. But this value must not be very high otherwise the system becomes unstable.
The equation implemented on the robot and represented on the chart in figure 3 is:

$$w(t) = -100 \cdot \tanh\left(\left(\Delta d - 10\right) \cdot 0.015\right) \qquad (\mathbf{6})$$

3. Cooperative Behaviour in general and specific tasks

3.1 Information Acquisition

Each robot is a completely autonomous machine which takes its own decisions based on the information it gets from surrounding environment (using sensors and camera) and team-members. When game entity raw information (ex: ball position) is received from different sources, then a trustful weight is associated with it based on the confidence of the source. A statistical algorithm is applied to eliminate wrong values and enhance the accuracy of the result.
This treated information is then used to accomplish cooperative behaviour among all robots on the field.

3.2 Dealing with specific tasks – Making a pass as example

In RoboCup football games there are many different situations to deal with, therefore behaviour is not static. Robots must change roles, adapt to new situations and develop team work in a synergistic way.
A particular and challenging situation is created every time the game is stopped and there is a free kick, a throw in, a corner or a game restart. In these cases one team must re-position in defensive way and the other in an attack manner. These cases have been discussed earlier [2]. A particular issue arises when it concerns a ball pass between two players. Taking advantage from flexible hardware design, providing that the strength of the kick can be specified according to distance, a pass algorithm was implemented and demonstrated.
This algorithm is presented in a simplified version in **Fig. 5**. This scheme involves two robots. The one possessing the ball or most closed to it takes the responsibility to make the pass. Another robot moves to an empty space with a free path to the ball and sends a signal to the 1st robot indicating position and attendance for receiving the ball.

When all conditions are met to guarantee success, pass is carried out. After that, both robots make an evaluation for the success of the accomplished task. This will permit a sort of re-enforcement learning applied to specific tasks.

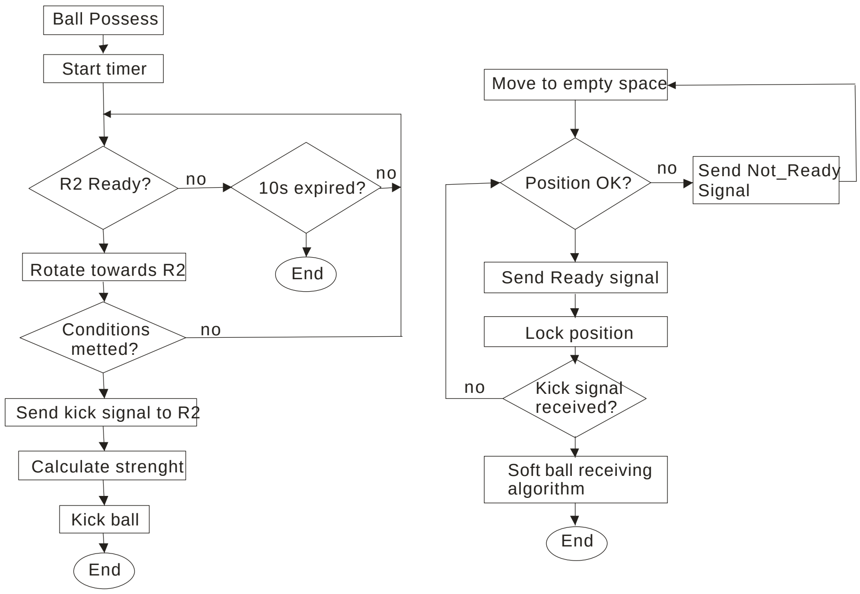


Fig. 5. Algorithm for effective ball passing

4. Upwards Kicker

The kicking system used in past competitions proved to be very efficient, the robots could execute pass with precision and powerful shoots. Basically, it is based on electromagnetic laws and consists of an electric coil and a movable iron core.

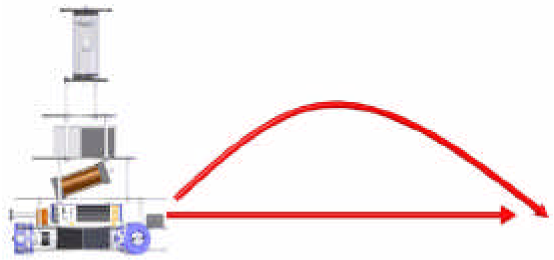
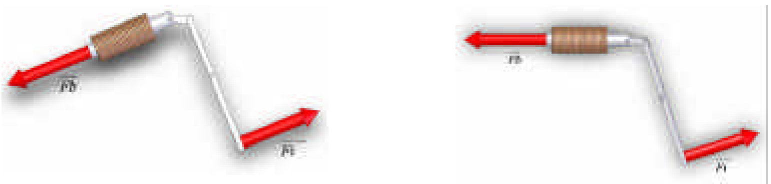


Fig. 6. Horizontal and Upwards Kicker

In order to enhance even more de capabilities of robot kicking, the team developed a new kicking system that allows throwing the ball not only in the horizontal direction but also upwards. Kicking upwards is important because it reduces probability of opponent interception.

The design and development of the upwards mechanical kicking solution was based on the basic principles which describe a projectile launch. According to these, it is possible to control height and maximum reach of the projectile (in this case the ball), by varying force and angle of impact.
Using the know-how acquired with the development of the horizontal kick a few years ago and the physics basic principles, a new electro-mechanical device which permits to apply a controlled force on the ball, according to an impact angle calculated to optimize the relationship of the ball maximum reach/height.
According to the lever basic principles, to maximize the kick it is necessary that at the time of the impact the angle of the force exerted by the coil (Fb) must be equal to the angle of the exerted force on the ball (Fi).



$$\overrightarrow{Fi} \approx \overrightarrow{Fb} \qquad \overrightarrow{Fi} \approx \overrightarrow{Fb} * \cos a$$

Fig. 7. Mechanical Kicker impact angle

The changes to get an upwards kick are not only mechanic changes but also on Kick control hardware. The old controller could only command one of the electromagnetic coils, and therefore the robot could only kick in one direction.

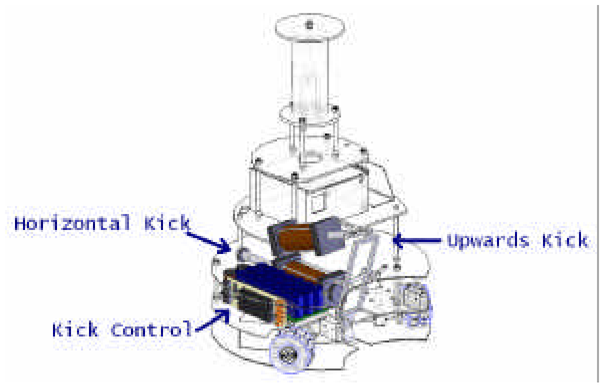


Fig. 8. Mechanical construction of the Kicker

After assembling the kicking system, some tests were carried out in order to check the kick performance and reliability and they proves very successful.

5. Conclusions

The robots proved very successful in ball pursuing. They are very fast and very responsive without loosing control. This technique will be merged with obstacle avoidance in a near future.
The ball pass algorithm proved also that it works fine in many cases. Sometimes with obstacles and moving opponents the result is not very visible.
The kick upwards is very strong and hardly misses the target. The Electronics needed to be changed from previous year to adapt 2 cores and these are selected by software when a decision is made to kick.
With these three important improvements the robots play much better than before.

References

[1] "Dynamic approach to behavior-based robotics: design, specification, analysis, simulation and implementation", E. Bicho, Shaker Verlag, Aachen, 2000, ISBN 3-8265-7462-1.
[2] "Optimization of fast moving robots and implementation" - Fernando Ribeiro, Pedro Silva, Ivo Moutinho, Vítor Silva, Nino Pereira