

# Development of Anytime Algorithms for Pass Implementation in Robocup Simulation League

Borislav Novikov<sup>[0000-0001-6283-720X]</sup> and Michail Panteleyev<sup>[0000-0002-6077-760X]</sup>

Saint Petersburg Electrotechnical University, ul. Professora Popova 5, 197376 St. Petersburg, Russian Federation  
{bnovikov012, MPanteleyev}@gmail.com

**Abstract.** Creating intelligent agents that can carry out team interaction is one of the most important tasks in the field of artificial intelligence. For experimental verification of the effectiveness of the proposed theoretical approaches to its construction, virtual football championship held annually as a part of the Robocup world championship. One of the important tasks in building the intelligence of a robot football player is the implementation of the pass. As part of this task, the agent must be able to correctly understand the situation, determine the possibility of performing a pass, and calculate the parameters of the ball strike. The article will consider the method of determining the area where the ball will be passed without the possibility of interception of the ball during its flight by an opponent and with a known advantage of the receiver within its limits. This area is calculated by determining the receivers, interceptors, and calculating special areas, including the Voronoi diagram. The proposed algorithm also meets the requirements for anytime algorithms, which allows you to achieve a result whose quality is proportional to the time spent on calculations, which is extremely important in such a rapidly changing environment as virtual football. Experimental testing confirmed the algorithm's performance as an anytime algorithm, namely the ability to execute the algorithm faster due to the quality of the result.

**Keywords:** RoboCup · Intelligent agents · Anytime algorithm · Virtual soccer · Pass implementation.

## 1 Introduction

The creation of intelligent agents that would be capable of solving problems in groups when opposing another team is currently one of the central problems in artificial intelligence [1]. The Robot World Cup (RoboCup) [2] championship is held annually to experimentally examine the multi-agent counteraction models in virtual soccer. One of the important particular tasks in the construction

---

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

of intelligence of the football player agent is to determine the possibility and parameters of the pass implementation in a given situation.

In [3] an approach to ball interception based on introduced concepts of qualitative and relative velocity (quick to left, neutral, quick to right) are proposed and investigated. An experimental comparison of several methods was carried out: an exact numerical calculation of the interception point, a naive method with movement directly to the ball, a method based on reinforcement learning, and method with qualitative velocities. The experiments have shown the numerical method and the learned behavior to be the best, while the numerical method is slightly preferable.

In [4] a simple and robust algorithm for the interception of a moving ball by an omnidirectional robot for RoboCup Small Size League is proposed. The heuristic algorithm requires minimal knowledge of robot dynamics and based on two key ideas: (i) consideration of ball motion via transition to a reference frame where the ball is static and (ii) planning the motion of the robot in such a reference frame from the geometric viewpoint. Experiments conducted in a real SSL environment confirmed successful interception in a variety of scenarios, characterized by different directions of ball motion and the positional relationships between the ball, robot and goal.

The solution for a ball interception behaviour developed as part of the CAMBADA team from University of Aveiro for RoboCup Middle Size League is presented in [5]. CAMBADA solution is based on a uniformly accelerated robot model, that takes into account a number of parameters, in particular ball velocity, robot current  $v$

In [6] is considered an approach based on the calculation of the dominant area of one agent team with respect to an opponent team. Inside the calculated area, players have an advantage, namely, each point of this area is closer to them than to the opponent. It means that the implementation of the pass along the trajectory located inside this area will be made under the assumption that the opponent will not be the first on the ball.

An important feature of the virtual football environment is the need to make decisions in real time [7]. According to that, the allowable thinking time can vary significantly depending on the situation. It follows that algorithms for solving specific problems should be constructed as anytime algorithms (ATA) [8], which allow the agent exchange the computation time for the quality of the result.

## 2 Problem statement

The game involves two teams of robot agents, 11 players on each side:  $T = T_i, O = O_i$ , where  $i = \overline{1, 11}$ .

Each player has a set of static parameters, which include the radius  $r$  of the body, the area of possession of the ball *KICKABLE\_AREA*, the maximum running speed  $V_{max}$ , etc. In addition, the current state of the player is characterized by dynamic parameters, such as coordinates  $(x, y)$ , current speed  $V$ , direction of movement  $w$ , etc.

The players can perform commands: Kick - kick the ball (the ball must be in the *KICKABLE\_AREA* of the player); Dash - accelerate the run; Catch - capture the ball (only for the goalkeeper), etc [9].

To implement the pass, the player must execute the Kick command so that the ball begins to move towards the partner. Let us consider the situation when the player  $T_i$  decided to give the pass to partner  $T_j$ . Agent needs to determine the direction of movement and the initial speed of the ball so that the partner  $T_j$  would reach the ball at the desired point and at the same time no players of the other team (opponents) can intercept the ball, i.e. reach it earlier than  $T_j$ .

Let's consider the problem under the worst case assumption, when all opponents are already moving along the shortest path to the ball trajectory at maximum speed  $V_{max}$  to intercept the ball.

In addition, we assume that the agent knows the exact (cleared of server-generated noise) global (Cartesian) coordinates of the ball and all players involved in the episode.

Under these assumptions, the task boils down to the definition of a "Safety Pass Area" (SPA), each point of which satisfies the above conditions.

Algorithm of SPA calculation will be developed as an anytime algorithm.

### 3 Calculating the Safe Pass Area

Let's denote the necessary concepts: "The Game Area" - GA, "The Influence Area" - IA, "The Reach Area" - RA, "The Safe Catching Area" - SCA.

Then, SPA can be calculated by finding the intersection of these sets as follows:

$$SPA = GA \cap IA \cap RA \cap SCA \quad (1)$$

#### 3.1 The Game Area

It is represented by the four coordinates that define the area of the possible location of the objects in the game. It is the constant parameters set by the server [9].

#### 3.2 The Influence Area [10]

To find the area of influence, you need to build a Voronoi diagram[11]. This diagram is based on a set of points  $P$  of the plane  $S$ . It is a partition of this plane, in which each point  $p_i$  of the set  $P$  is mapped to the area of this partition  $s_i$ . Each area contains all the points of the plane that are closer to the point of this area  $p_i$  than to other points  $p_j$ , where  $j \neq i$ . the area  $s_i$  Mapped to point  $p_i$  is a polygon and is called a locus.

If for the number of points required to build a diagram the coordinates of players is taken, and for plane - game area, the locus of the graphs will be considered as areas of influence of each player (see Fig. 1).

This diagram can also be used for solving other tasks, such as analyzing the situation on the field, which makes it useful to calculate it.

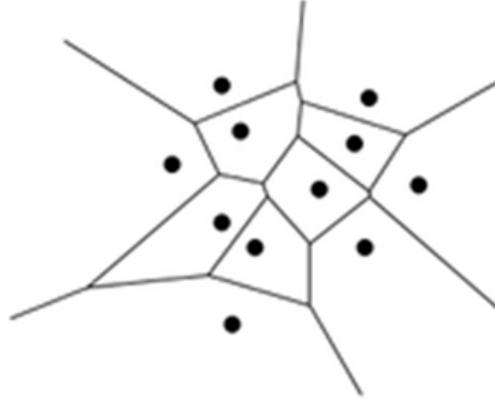


Fig. 1: The Influence Area

### 3.3 The Reach Area

This area is a set of points in which the ball can be after the Kick command execution. If the calculated area does not have any shared points with IA, then a safe pass cannot be executed.

We need to calculate the maximum flight length  $L$  of the ball, which becomes the radius of the circle with center in the coordinate of the ball (see Fig 2). To find it, the agent needs to calculate the value of the *power* parameter of the kick command.

Let the real value of the force be *act\_pow*. Then, we calculate it by using the formula 2 [12]. Its value depends on the distance *dist\_diff* between the player and ball and the angle *dir\_diff* between the direction of the body and ball. The *power* value will be equal to 100, because it is a maximum value of the *power* parameter of the kick command. The *kickable\_margin* parameter is set by the server and is pre-known [9].

$$act\_pow = power * (1 - 0.25 * \frac{dir\_diff}{180} - 0.25 * \frac{dist\_diff}{kickable\_margin}) \quad (2)$$

Then, putting the obtained value into the formula 3 allows us to find the absolute of the applied acceleration vector, where *kick\_power\_rate* is the parameter set by the server [9]:

$$|\vec{V}_0| = |\vec{a}| = act\_pow * kick\_power\_rate \quad (3)$$

Thus, we know the maximum initial velocity of the ball  $V_0 = |\vec{V}_0|$ . Knowing the ball speed deceleration parameter *DECAY* set by the server *DECAY* [9],  $L$  can be obtained as follows:

$$L = \frac{V_0}{1 - DECAY} \quad (4)$$

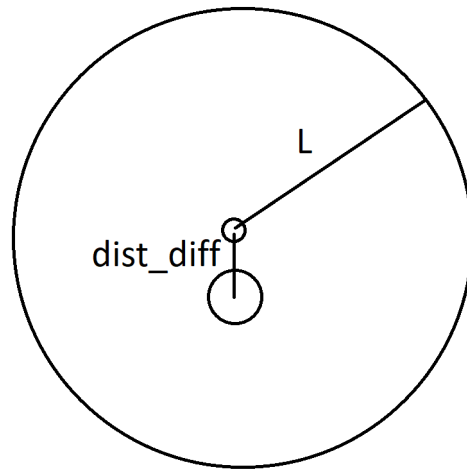


Fig. 2: The Reach Area

### 3.4 Marking the Receivers

The receiver is within the reach area or can reach it earlier than anyone else, at least at one point.

To find them, you need to calculate the reach area and the area of influence of each player.

We consider potential receivers as players who have at least one vertex of the influence area within the reach area.

### 3.5 The Order of Receivers

It is advisable to consider the Receivers in a non-random order.

Three options can be offered at least for creating an order:

- those who are closer to the goal go first
- those who are closer to the ball go first
- those who border strongly with my dominant area and weakly with enemies go first

The amount of receivers to be considered can be decreased.

### 3.6 Marking the Interceptors

We need limit the amount of potential trajectories of the ball. To begin, we find two points of DA, which are at the minimum angle  $\alpha$  and maximum angle  $\beta$  relative to the coordinate of the ball.

Then, let the interceptors be players-opponents who are closer to the ball than target-player (the player to whom the pass is given), and at an angle in the range  $[\alpha - 90; \beta + 90]$  relative to the ball. Let's mark them as  $E = E_i$  (see Fig. 3.), where  $i = \overline{1, p}$ .

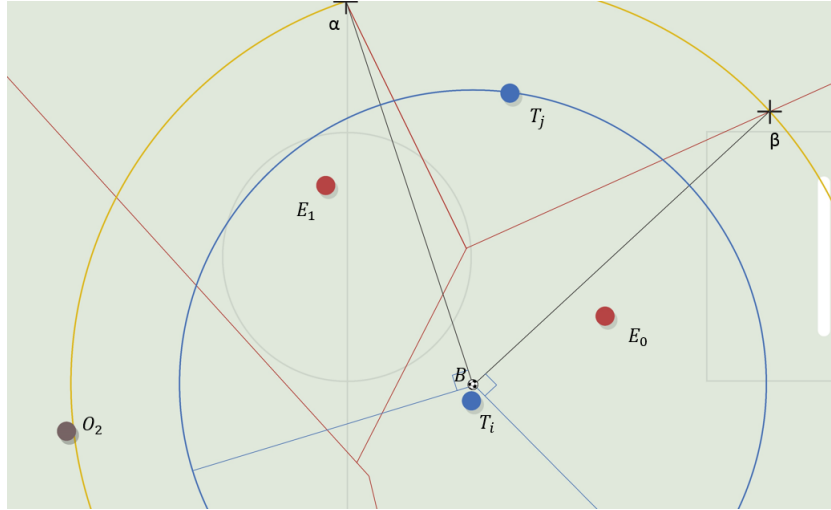


Fig. 3: Marking the Interceptors

### 3.7 The Order of Interceptors

It is advisable to consider the interceptors also in a non-random order.

Three options can be offered at least for creating an order:

- those who are closer to the receiver go first
- those who are closer to the ball go first
- those who border strongly with receivers dominant area go first

The amount of interceptors to be considered can be decreased.

### 3.8 The Flight Trajectories

Potential flight trajectories of the ball will be considered as a set of rays casted in the range of previously found angles. The number of possible angles is limited and equal to amount of trajectories. ATA's requirements force agent to check only some of them. To ensure the optimal result the simple order of checking the angles consider as non-optimal.

One of the possible approaches to choose the order is to begin with checking of the extreme angles and then at each step check the angle, which is between the already checked neighbor angles and divides the greatest interval.

Also, there is an approach in which the agent have the point of interest. For example, the task is to push the ball closer to the goal. So, it might help if the angle's probability of checking depends on the distance to the target angle.

### 3.9 The Area Calculation

Further, let  $B, E_i, T_i$  be the coordinates of the objects. We cast in the range of previously found angles  $[\alpha; \beta]$   $M$  rays  $R = R_i$ , where  $i = \overline{1, m}$ , which are vectors of unit length. Then,  $\overrightarrow{(B + R)}$  - the set of possible trajectories of the ball.

For each  $i$  and  $j$  in  $\overrightarrow{(B + R_i)}$  and  $E_j$  we find the intercept point  $I_{ij}$ . As mentioned earlier, we assume that the opponent  $E_j$  chose the shortest path to the trajectory of the ball, namely, the perpendicular line to  $\overrightarrow{(B + R_i)}$ . Then, to calculate the point  $I_{ij}$  find the projection of the vector  $\overrightarrow{(B + BE_j)}$  on the ray  $R_i$ , which is the length of the vector  $\overrightarrow{BI_{ij}}$ , by the formula (5). So, the intercept point can be calculated by the formula (6):

$$|\overrightarrow{BI_{ij}}| = \text{proj}_{\overrightarrow{R_i}} \overrightarrow{B - BE_j} = \frac{\overrightarrow{B - BE_j} \cdot \overrightarrow{R_i}}{|\overrightarrow{R_i}|} \quad (5)$$

$$I_{ij} = B + \overrightarrow{R_i} * |\overrightarrow{BI_{ij}}| \quad (6)$$

For each  $E_j$  time  $t_{ij}$  of reaching  $I_{ij}$  can be obtained by the formula (7). Regarding the initial assumption that the opponent moves at the constant speed  $V_{max}$  follows:

$$t_{ij} = \frac{|\overrightarrow{E_j I_{ij}}|}{V_{max}} \quad (7)$$

We calculate a point  $X_{ij}$ , which is the farthest point, passing to which the ball will be intercepted. Firstly, by the formula 8 we find the value of the initial velocity  $V_{0_{ij}}$ , at which it during  $t_{ij}$  will pass the distance  $|\overrightarrow{BI_{ij}}|$ . Take into account *act\_pow*, calculated by the formula (2). Secondly, we calculate the distance  $S_{ij}$ , which the ball will pass with  $V_{0_{ij}}$ , by the formula 9.

$$V_{0_{ij}} = \frac{|\overrightarrow{BI_{ij}}| * (1 - DECAY)}{(1 - DECAY^{t_{ij}}) * \text{act\_pow}} \quad (8)$$

$$S_{ij} = \frac{V_{0_{ij}}}{(1 - DECAY)} \quad (9)$$

Then, the point  $X_{ij}$  (see Fig. 5.a) will be:

$$X_{ij} = B + \overrightarrow{R_i} * S_{ij} \quad (10)$$

Next, we select for each  $i$  of  $X_{ij}$  the most farthest point from  $B$  and call it  $G_i$ . The curve obtained by connecting the points  $G_i$  (see Fig. 5.b) is the boundary of the safe pass area from below and the boundary of the interception area from above.

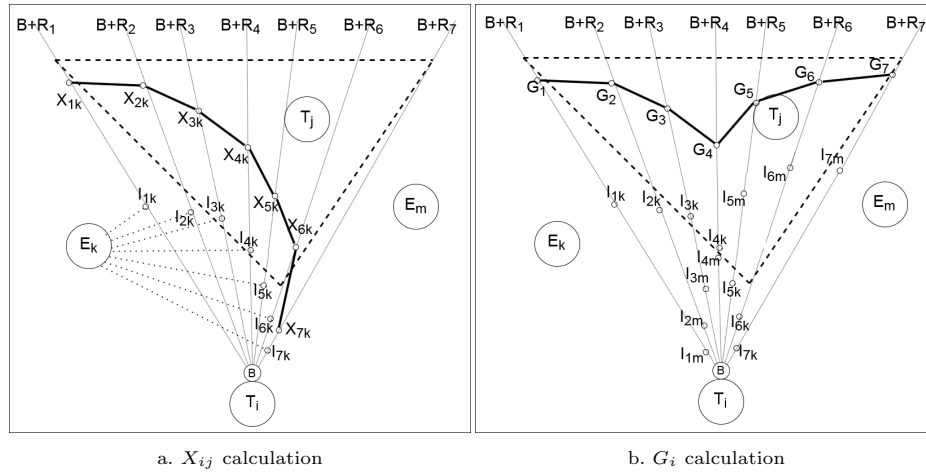


Fig. 5: SCA Calculation

### 3.10 The Point Selection.

The selection of the specific point from the SPA can be performed based on various criteria such as:

- distance to the goal
- distance to the target player
- distance to the edge of the field

### 3.11 The Actions of the Receiver

The player must correctly determine that he was selected as the receiver and start moving immediately to the assumed point of receiving the pass. For correct calculation of this point, it is necessary to take into account that he will be able to see the ball flying in his direction only after one frame.

## 4 The SPA Calculation Algorithm

The pseudocode of the algorithm will be offered below:



**Data:**  $T_i$  - coordinate of the player with ball;  
 $T_j$  - coordinate of the receiver;  
 $B$  - coordinate of the ball;  
 $T = T_i$ , where  $i = \overline{1, n}$  - teammates' coordinates;  
 $O = O_i$ , where  $i = \overline{1, n}$  - opponents' coordinates;  
 $M = const$  - the amount of trajectories;  
**Result:**  $SPA = SPA_i$ , where  $i = \overline{1, m}$  ;  
 $RA \leftarrow RA\_Calculation(T_i, B)$ ;  
 $IA \leftarrow IA\_Calculation(GA, T, O)$ ;  
 $R \leftarrow Receivers\_Marking(RA, IA)$ ;  
 $Receivers\_Ordering(R)$ ;  
 $R \leftarrow Receivers\_Marking(RA, IA)$ ;  
 $SPA \leftarrow \emptyset$   
**for**  $r \in R$  **do**  
    **if** *need to interrupt* **then**  
        | interrupt;  
    **end**  
     $alpha, beta \leftarrow Caculate\_A\_B(RA, IA)$ ;  
     $Tr \leftarrow Calculate\_Trajectories(alpha, beta)$ ;  
     $Trajectories\_Ordering(Tr)$ ;  
     $I \leftarrow Interceptors\_Marking(alpha, beta, IA)$ ;  
     $Interceptors\_Ordering(I)$ ;  
    **for**  $i \in I$  **do**  
        **if** *need to interrupt* **then**  
            | interrupt;  
        **end**  
         $X \leftarrow \emptyset$  ;  
        **for**  $tr \in Tr$  **do**  
            **if** *need to interrupt* **then**  
                | interrupt;  
            **end**  
             $x \leftarrow Calculate\_X(tr, B, i)$ ;  
        **end**  
         $Add\_To\_X(X, x)$ ;  
    **end**  
     $G \leftarrow Calculate\_G(X)$ ;  
     $spa \leftarrow Calculate\_SPA(G, IA, RA)$ ;  
     $Add\_To\_SPA(SPA, spa)$ ;  
**end**  
 $PASS\_POINT \leftarrow Select\_Point(SPA)$ ;

## 5 The Results of Experiments

In this section, we apply our algorithm to the RoboCup Simulation League and evaluate how well it works.

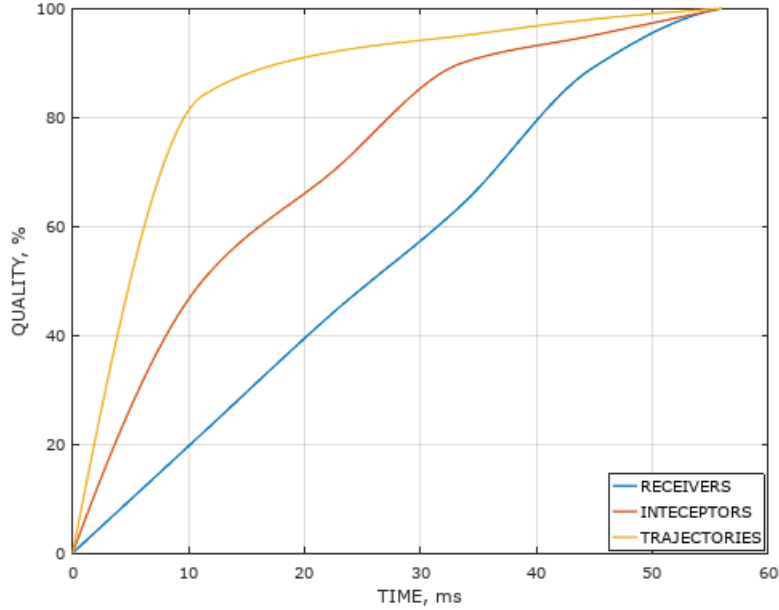


Fig. 6: Experiments Results

The graph above shows how the algorithm execution time depends on the quality of its result (see Fig. 6). The algorithm works as the anytime algorithm, as you can see. There are three curves on the graph: the receiving curve, the interceptor curve, and the trajectory curve.

The experiment was carried out with a separate decrease in the number of considered receivers, interceptors, and trajectories. Reducing several parameters at the same time can result in an even greater reduction of the algorithm execution time.

### 5.1 Reducing the amount of receivers considered

Table 1: Comparison of the quality and execution time of the algorithm for different values of the number of considered receivers

Amount, %	Quality, %	Time, ms
20	22	11.2
40	44	22.4
60	64	33.6
80	89	44.8
100	100	56

The estimated parameter Quality was calculated as follows:

$$Quality = Calculated\_SPA\_Area * 100 / SPA\_Rael\_Area \quad (11)$$

A decrease of the number of considered receivers strongly affects the result time (see Fig. 6).

## 5.2 Reducing the number of interceptors considered

Table 2: Comparison of the quality and execution time of the algorithm for different values of the number of considered interceptors

Amount, %	Quality, %	Time, ms
20	51	11.2
40	72	22.4
60	91	33.6
80	96	44.8
100	100	56

The estimated parameter Quality was calculated as follows:

$$Quality = Calculated\_SCA\_Area * 100 / SCA\_Real\_Area \quad (12)$$

A decrease of the number of considered interceptors does not significantly affect the result time (see Fig. 6).

## 5.3 Reducing the number of trajectories considered

Table 3: Comparison of the quality and execution time of the algorithm for different values of the number of considered trajectories

Amount, %	Quality, %	Time, ms
20	84	11.2
40	92	22.4
60	95	33.6
80	99	44.8
100	100	56

The estimated parameter Quality was calculated as follows in equation (12).

A decrease of the number of considered interceptors has a little effect on the result time (see Fig. 6).

## 6 Conclusion Remarks

In this paper we presented a variety of methods to calculate the possibility and parameters of the pass in the virtual environment of the RoboCup Simulation League. We introduced an approach to ball passing that has quality which depends on the time of calculation.

We introduced an approach to ball passing that has quality which depends on the time of calculation.

## References

1. Pantelev M.G., Puzankov D.V. Intelligent agents and multiagent systems, SPbGETU "LETT" Publishing, SPb (2015)
2. RoboCup. The Robot World Cup Initiative. Robocup officiale site. Available at: <http://www.robocup.org>
3. Stolzenburg, F., Obst, O., Murray, J. Qualitative velocity and ball interception. In: Jarke, M., Lakemeyer, G., Koehler, J. (eds.) KI 2002. LNCS (LNAI), vol. 2479, pp. 283–298. Springer, Heidelberg (2002).
4. Makarov P.A. et al. A Model-Free Algorithm of Moving Ball Interception by Holonomic Robot Using Geometric Approach. In: Chalup S., Niemueller T., Suthakorn J., Williams MA. (eds) RoboCup 2019: Robot World Cup XXIII. RoboCup 2019. LNCS, vol. 11531, pp. 166-175. Springer, Cham
5. Cunha J., Lau N., Rodrigues J. Ball Interception Behaviour in Robotic Soccer. In: T. Rofer et al. (Eds.): Robot Soccer World Cup XV, LNCS vol. 7416, pp. 114–125, Springer-Verlag Berlin Heidelberg (2012)
6. Nakanishi R., Bruce J., Murakami K., Naruse T., Veloso M.M. Cooperative 3-robot passing and shooting in the RoboCup small size league. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X. LNCS (LNAI), vol. 4434, pp. 418–425. Springer, Heidelberg (2007)
7. Pantelev M.G. A formal model of proactive iterative action planning for real-time intelligent agents [In Russian]. Proc. of the 14 National Conference on Artificial Intelligence with International Participation KII-2014. - Kazan: Fizmatlit. - V 1. - pp.323-334. (2014)
8. Zilberstein S. Using Anytime Algorithms in Intelligent Systems. AI Magazine 17(3), 73-83 (1996)
9. Chen M, Dorer K. RoboCup Soccer Server, 150 p. (2003)
10. Dashti H.T. et al. Dynamic Positioning Based on Voronoi Cells (DPVC). In: Breidenfeld A., Jacoff A., Noda I., Takahashi Y. (eds) RoboCup 2005: Robot Soccer World Cup IX. RoboCup 2005. Lecture Notes in Computer Science, vol 4020. Springer, Berlin, Heidelberg
11. Dobrin A. A review of properties and variations of Voronoi diagrams. Whitman College (2005)
12. Remco de Boer, Jelle Kok, Frans Groen. UvA Trilearn 2002 team description – Faculty of Science, University of Amsterdam (2002)