# Models and Methods of Real-Time Action Selection in Virtual Soccer

Borislav Novikov, Daniil Kivarin, Michail Panteleyev
Saint Petersburg Electrotechnical University 'LETI'
Saint Petersburg, Russia
{bnovikov012, dkivarin, mpanteleyev}@gmail.com

*Abstract*—**The paper is devoted to the problem of action selection by an intelligent agent in real-time systems. This is one of the key tasks in the field of multi-agent systems in respect of the existing time constraints and incomplete perception of the surrounding world by agents. As an example, the virtual soccer environment, which is a highly dynamic system with a rapidly changing environment, is selected for illustration. The article analyzes and builds models of basic actions from virtual soccer. As a basis for these models, an approach is proposed based on the step-by-step implementation of the task under consideration with a constant assessment of the usefulness of possible options for action. The proposed algorithm also meets the requirements for anytime algorithms, which allows agent to achieve a result which quality is proportional to the time spent on calculations, which is extremely important in such a rapidly changing environment as virtual soccer. 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.**

## I. Introduction

Multi-agent worlds or systems are sets of interacting intelligent agents (IA) performing purposeful actions in a dynamic environment [1]. Currently, MAS are used for many applications in many fields including transportation, logistics, graphics, robotics and others. One of key tasks of intelligent agents in such worlds is real time planning in a constantly changing multi-agent environment. Depth and completeness of the analysis of possible actions in dynamic environment are limited and must adapt to current time limitations [2], [3].

The activity of the IA is aimed at achieving certain goals, but often the same goal can be achieved in various ways. Moreover, these ways will differ among themselves in many terms (for example, in terms of security or speed). In order for IA to make a rational choice between the possible ways to achieve its goals, the concept of utility is introduced. Utility is the preference of the considered state of the world in comparison with other possible ones. The utility function maps the world state to a real number, which indicates the appropriate satisfaction level of the agent. Consideration of utility during planning allows the IA to solve the problem of uncertainty when choosing from a variety of possible further actions so that the goal is achieved with maximum efficiency.

Considering that activities of IA performed in conditions of time limitations, algorithms underlying its actions should be organized as anytime algorithms (ATA) [4]. As opposed to standard artificial intelligence algorithms, which can take a lot of time to get results, ATA are designed to work with a small amount of available time. The quality of the result in such algorithms increases with increasing time of their execution. The use of ATA allows the IA to rationally distribute the available time resources during the next actions planning.

Currently, the most popular platform for the study of MAS and IA is the virtual soccer server Robocup Soccer Server [5]. This platform allows to simulate teams counteractions scenarios in real time in conditions of limited perception of the world. To study models and decision-making methods in real time, a specific task related to virtual soccer was selected, namely, the ball pass to a teammate. This example is an important basic operation, as it is one of the main methods of interaction between agents in a team. This task also consists of many sub-tasks that need to be solved, the result of which can be useful in the context of other complex actions, such as a shot on goal or dribbling.

In [6] 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 [7] 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 [8]. 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 [9] 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.

The purpose of this article is to represent models and methods for ball pass based on the calculation of the utility assessment of possible actions and analyze their work in typical situation from virtual soccer.

## II. Algorithm Development

### A. Basis Concepts

The game involves two teams of robot agents, 11 players on each side [10], [11]: $T = T_i, O = O_i$, 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 [10], [11]. 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 [10], [11].

The players can perform commands: Kick - kick the ball; Dash - accelerate the run; Catch - capture the ball (only for the goalkeeper), etc [10], [11].

To implement a pass, the player must execute the Kick command so that the ball begins to move towards the partner. Player-receiver needs to determine the direction of movement and the initial speed of the ball and reach the ball at the desired point. At the same time all players of the other team (opponents) must not have a chance to intercept the ball, i.e. reach it earlier than receiver.

As mentioned earlier, it is crucial to spend time effectively so the algorithm will be developed as an anytime algorithm.

### B. Algorithm Description

The considered algorithm implements the player ball passing in a virtual soccer match. During the planning of the pass, the player needs to analyze and choose a partner for the pass, the ball flight path and the strength of the ball kick, which comes down to choosing one of the many options of these factors combinations. Also worth noting that planning occurs in conditions of limited time, therefore, it is not possible to consider all possible options for a difficult situation on the field.

At the input, the algorithm receives information about the environment, which the player perceives at the current moment of time, and information that was previously perceived, and now it is already stored in his memory. Such information includes positions of partners and opponents on the field, angles of their views, directions and values of their movement speed, as well as the current position of the player and the ball. As a result of its work, the algorithm gives the trajectory along which it is necessary to pass the ball, and the force that must be applied to the ball during the kick.

In the algorithm, the following main steps of work can be distinguished:

1) Identification of receivers
2) Identification of interceptors
3) Determination of pass trajectories
4) Determination of kick power

At the end of each of these steps, an utility assessment of the new parameters obtained and a reassessment of the utilities already calculated in the previous steps are performed. Calculation of these assessments allows to correct the way of working of the next algorithm step so that the most useful options for achieving the player goals are considered.

This algorithm is adapted to the possibility of time constraints, so it can be completed at any time and at the same time it gives the best result achieved during operation. As the execution time increases, the accuracy of the result of the algorithm work also increases, which corresponds to the concept of an anytime algorithm.

### C. Identification of Receivers

Receivers - a set of allied players considered for the purpose of passing the ball to them. Later in the article we will denote them as R.

*1) Identification:* To find them, we need to calculate two special areas: Influence Area (IA) [12] and Reach Area (RA).

Each point of IA can be reached by the player who owns this area faster than by anyone else.

For this purpose we need to build a Voronoi diagram [13] This diagram of a finite set of points on a plane represents a plane partition in which each region of this partition forms a set of points that are closer to one of the elements of the set than to any other element of the set.

If you take the coordinates of the players as the set of points for plotting the diagram, and the area of the game as the plane, then each area of the diagram will correspond to the area of influence of a particular player (see Fig. 1).

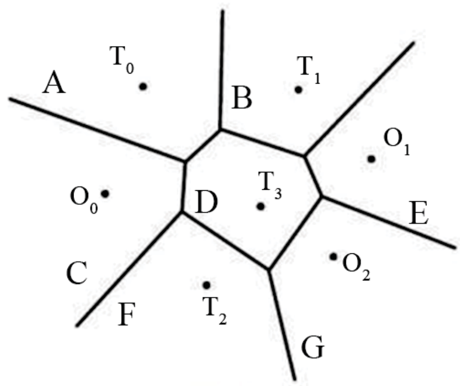


Fig. 1. Example of Influence Area. Each polygon of this area corresponds to the player within. Areas A, B, D, F - teammates' IAs, C, E, G - opponents IAs.

Reach Area represents the area of points where the ball may end up after kicking it.

To find it, we need to determine the maximum current force of impact on the ball using Equation 1, the speed transmitted to the ball using equation 2, and use it to calculate the final length of the ball's flight with equation 3.

The power parameter in equation 1 is passed by the player and taken as 100, which corresponds to the maximum impact force, kickable_margin, DECAY, kick_power_rate in equation 1, equation 2, equation 3 - constant parameters set by the game server [10], [11]. dist_diff - distance to the ball and dir_diff - angle between agent's view and the ball.

$$act\_pow = power * (1 - \frac{dir\_diff}{4 * 180} - \frac{dist\_diff}{4 * kickable\_margin}) \tag{1}$$

$$V_0 = act\_pow * kick\_power\_rate \tag{2}$$

$$L = \frac{V_0}{1 - DECAY} \tag{3}$$

The resulting value L becomes the radius of the circle centered at the point where the ball is located and represents RA (see Figure 2).

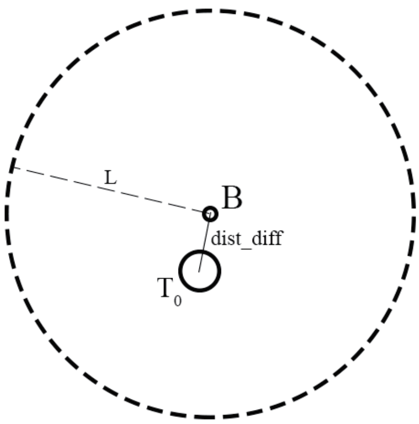


Fig. 2. Example of Reach Area. Dotted-line circle represents the Reach Area.

We will consider receiver as an ally whose area of influence the ball can reach and at least one vertex of its area of influence is within reach area.

*2) Utility Assessment:* At this step of the algorithm, for each of identified potential receiving players, the primary value of the utility assessment $U_R$ is calculated. The utility is value, expressed as a percentage, showing how profitable is the ball pass to this partner for achieving the goals of the considering player.

The utility assessment $U_R$ can be calculated based on many different factors, the number and variety of which varies depending on the considering situation and current goals that the player pursues. For example, at this stage of the ball pass, the distance from the receiver to the ball, the angle of rotation of receiver body and the angle of his view, the speed and direction of his movement, the number of vertex of his area of influence which are within reach area, and the distance between the receiver and the opponent's goal can be taken into account.

Factors which are considered in utility assessment form set of assessment criteria. Each element in this set is a formal description of the criterion $factor\_value$ and its weight $factor\_weight$, showing the significance of criterion influence on the utility value.

The formal description of the criterion $factor\_value$ is a normalized numerical value, which is calculated by the equation (4).

$$factor\_value = factor\_value_{cur} / factor\_value_{max} \tag{4}$$

Where $factor\_value_{cur}$ is the value of the parameter defining the criterion for the current receiver; $factor\_value_{max}$ is the maximum value of this parameter among the entire set of receivers. The numerical value calculated by the equation (4) is taken without changes in the case when the maximum value of the parameter is necessary to achieve the goals. On the other hand, if the minimum value is necessary, the inverse of calculated numerical value is taken.

Weights of the criteria are directly set in the range from zero to one before the algorithm starts, so that their total sum for all criteria is equal to one.

The primary value of the utility assessment is calculated based on formal description $factor\_weight_i$ and weight $factor\_weight_i$ of each criterion according to the equation (5).

$$U_R = \sum_i factor\_weight_i * factor\_value_i \tag{5}$$

To optimize the further work of the algorithm, identified potential receivers are ordered depending on the calculated assessments $U_R$ so that the players who have the greatest utility value are considered first.

The step-by-step calculation of utility assessments allows to organize the ball passing algorithm as an anytime algorithm. Therefore, the player at any time in planning has a priority partner available for the pass. However, at this step, the trajectory and kick power obtained as a result of the algorithm will not take into account the possibility of an interception by the opponent team, which negatively affects the real efficiency of the pass.

It is advisable to present combinations of player actions at each step in the form of a ball passing graph, at the vertices of which the possible actions are contained. This graph at the current step of algorithm is presented on figure 3. In this figure vertex $S$ represents considered ball pass situation. Vertices $R$ generated from $S$ are identified receivers with defined utility assessment $U_R$ for each one.

### D. Identification of Interceptors

Interceptors - a set of opponents considered as interference when passing the ball to a particular receiver. Further in the article we will refer to interceptors as I.

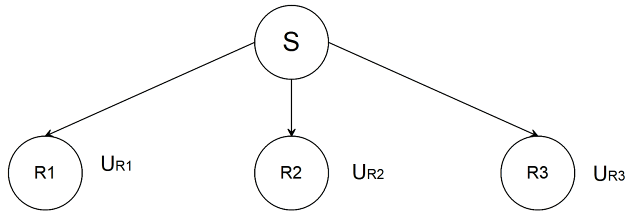


Fig. 3. Graph of possible pass parameters combination with identified receivers

*1) Identification:* To find them, we need to calculate the angles in the range of which the ball can be kicked. Let's consider two points of intersection of the receiver's area of influence and the area of reach that are at the minimum $\alpha$ and maximum $\beta$ angles relative to the ball's coordinate. The angles at which these points are located will be the angles we need.

Then, the interceptor will be considered as an opponent who is closer to the ball than the receiver and as well as in the range of angles $[\alpha - 90, \beta + 90]$ (see Fig. 4). The first restriction comes from the fact that the interceptor, located further from the ball than the receiver, will not reach it faster. The second restriction is based on the assumption that the interceptor moves perpendicular to the flight path of the ball, so even when considering the extreme possible trajectories, namely $\alpha$ and $\beta$, the opponent will try to reach the ball at its current position, which makes it impractical to consider it, because the ball will immediately leave this point after the kick.

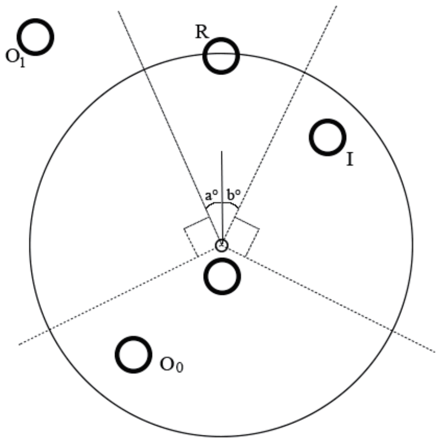


Fig. 4. Identification of Interceptors. This is an example of identification. Interceptors marked as I. Non-interceptor opponents marked as O. Receiver marked R. As you can see identification divided opponents onto two groups.

*2) Utility Assessment:* In order to optimize the further algorithm work, for each potential interceptor found, the value of the utility assessment $U_I$ is calculated. An assessment of the interceptor utility is a percentage value that shows how likely it is that the passing ball will be intercept by him before reaching the receiving player.

During the calculation of this utility assessment, same as the calculation of receivers utility $U_R$, the calculated value depends on a number of factors that were given before the algorithm start. Factors for utility assessment of interceptors include the angle of rotation of their bodies and the angle of their view, the speed and direction of their movement, as well as the distance between them and the ball.

These factors are included in the new set of criteria for interceptors, on the basis of which the value of the utility assessment $U_I$ is calculated by equation (5).

Based on the obtained values, the further order of consideration of players from the set of interceptors is determined. The most useful interceptors will be considered before others.

*3) Utility Reassessment:* After identification of the set of possible interceptors, new parameters appear that can correct the previously calculated utility of receivers $U_R$, therefore, at this step of the algorithm, the value $U_R$ is recalculated.

During reassessment of receivers utility, factors such as the number of identified interceptors and the inverse average of their utility can be considered.

Based on these factors, a set of new criteria for receivers is formed, based on which, according to equation (5), the corrective value of the receivers utility $U'_R$ is calculated.

Reassessment of the receiving players utility $U_R$ is calculated according to the equation (6).

$$U_R = (U_R + U'_R)/2 \tag{6}$$

Ball passing graph at the current step of algorithm is presented on figure 5. At this step each receiver vertex $R$ generates set of interceptor vertices $I$ with own utility assessment $U_I$.

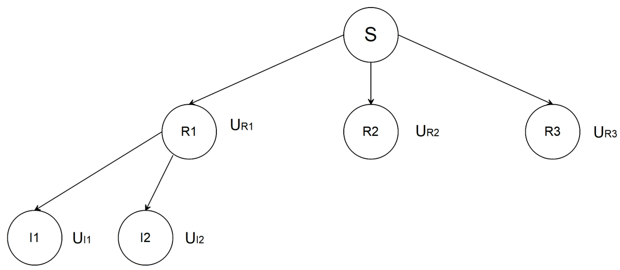


Fig. 5. Graph of possible pass parameters combination with identified receivers and interceptors

## E. Determination of Pass Trajectories

Trajectories - a set of possible directions of the ball's flight from the point of its current location. Represent a finite set of vectors casted in the range of previously found ALPHA and BETA angles. Further we will denote the trajectories as T.

*1) Amount of Trajectories:* This step of the algorithm involves determining the number of trajectories considered. The number of trajectories depends on the previously obtained utility assessment of the receiver under consideration $U_R$. Since the algorithm works under time-constrained conditions, it will be advisable to save time by viewing fewer trajectories for less useful interceptors in order to devote it to processing more priority players. The relationship between the number of trajectories and the utility value is calculated by equation (7).

$$k_T = (\beta - \alpha) * U_R \tag{7}$$

*2) Utility Assessment:* For each of considered trajectories, it is also necessary to calculate the value of the utility assessment $U_T$. For a trajectory, the percentage value of $U_T$ shows how safe it is for passing the ball to the receiver.

The number of potential interceptors near the trajectory, the angle between the trajectory and the nearest interceptor, the angle between the trajectory and the receiver, as well as the angle between the trajectory and the player point of interest can influence this utility assessment.

The utility of the trajectory is calculated depending on the selected criteria according to equation (5).

The obtained utility values are used in further ordering of the trajectories for consideration in such a way that the most useful trajectories are considered in priority.

*3) Utility Reassessment:* The receiving player with the most number of trajectories useful for ball passing should be considered in priority in view of the limited time for the player to make a decision. Therefore, after determining the utility assessments of the considered trajectories, a second recalculation of the $U_R$ value is performed.

This reassessment is based on the utility values of the trajectories $U_T$.

The corrective value of the utility of the receiver $U'_R$ is calculated by equation (8).

$$U'_R = \sum U_T / n \tag{8}$$

Where $n$ is the number of trajectories for considered receiver.

Reassessment of the utility of receiver players $U_R$ is calculated according to equation (6).

Ball passing graph at the current step of algorithm is presented on figure 6. In addition to available interceptor vertices $I$ receiver vertices $R$ generates set of trajectories vertices $T$ each of which also has defined utility assessment $U_T$.

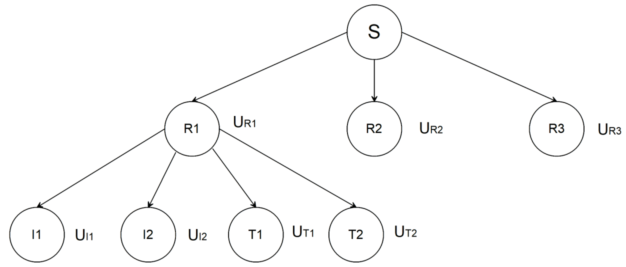


Fig. 6. Graph of possible pass parameters combination with identified receivers, interceptors and trajectories

### F. Determination of Kick Parameters

The kick parameter is the force P and direction T of the ball kick, transmitted by the agent to the server.

*1) Amount of Interceptors:* In order to improve the speed of the algorithm, it is advisable for each trajectory to consider the influence of only part of the possible interceptors. Therefore, we introduce the value of minimum utility (equation (9)), which limits the set of possible interceptors from below.

$$U_{ITmin} = U_{Imin} + U_T \tag{9}$$

Where $U_{Imin}$ is the minimum value of the utility assessment among all possible interceptors for the considered trajectory; $U_T$ is the value of the utility assessment of the trajectory in question.

*2) Determination:* The purpose of this stage is to calculate the parameters of the kick on a given trajectory, namely the required speed transmitted to the ball and, as a result, the force of impact on the ball. Moreover, these parameters must be calculated taking into account the counteraction of the possibility of intercepting the ball by the interceptor.

To find these parameters, we will proceed from the assumption that the interceptor moves along the most effective, namely the shortest path to the selected trajectory of the ball, that is, along a perpendicular. This assumption allows you to calculate the point of interception $X$ of the ball (see Fig. 7) using following equations (10), (11), as well as the time $t$ when the interceptor will reach this point using equation (12).

$$|\overrightarrow{BX}| = proj_{\overrightarrow{T}} \overrightarrow{B - BI} = \frac{\overrightarrow{B - BI} \cdot \overrightarrow{T}}{|\overrightarrow{T}|} \tag{10}$$

$$X = B + \overrightarrow{T} * |\overrightarrow{BX}| \tag{11}$$

$$t = \frac{|\overrightarrow{IX}|}{V_{max}} \tag{12}$$

Knowing the time and distance to the intersection point, as well as the deceleration parameter of the ball's speed $DECAY$ and the previously calculated $act\_pow$, we can determine the maximum speed transmitted to the ball at which it will be intercepted $V_{min}$). Having found it using equation (13), we can assume that by passing the ball a speed greater than that found earlier, we ensure that it is impossible to intercept it.

$$V_0 = \frac{|BX| * (1 - DECAY)}{(1 - DECAY^t) * act\_pow} \tag{13}$$

Next, after receiving $V_{min}$, we must take into account the limit of the speed transmitted to the ball $V_{0,max}$, that was calculated earlier. It will give us the first upper limit $V_{max} = V_{0,max}$ and the range of possible transmitted speed $[V_{min}; V_{max}]$.

The next restriction will be made by the area of influence of the receiving player. We need to find the intersection points

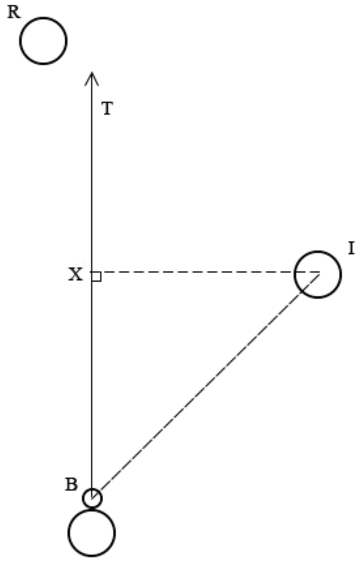


Fig. 7. Determination of Interception Point. X - interception point, R - receiver, I - interceptor, B - ball, T - trajectory

of the trajectory of this area. Since the area of influence is a convex polygon and the trajectory necessarily intersects this area (the range of $\alpha$ and $\beta$ angles), there will be either one intersection point (if the trajectory is at an $\alpha$ or $\beta$ angle), or two. Let's call the point closest to the ball $K_a$, and the point further - $K_b$ (in the special case, with one point, let $K_a = K_b$). Then, if we put $K_a$, $K_b$ in the equation (14) as an $S$ parameter, we can get $V_{k_a}$, $V_{k_b}$:

$$V = \frac{S}{1 - DECAY} \tag{14}$$

If the value of $V_{k_a}$ is greater than the current $V_{min}$, then $V_{min} = V_{k_a}$. If the value of $V_{k_b}$ is less than the current $V_{max}$, then $V_{max} = V_{k_b}$.

The impact force will be calculated using the following formula:

$$P = \frac{V * 100}{V_{0,max}} \tag{15}$$

Then, knowing $V_{min}$ and $V_{max}$ by equation (15), we can calculate the range of possible impact force for this trajectory $[P_{min}; P_{max}]$.

*3) Utility Reassessment:* The utility of the trajectory also depends on the safe receiving zone, which is determined after calculating the optimal kick power. In view of this new criterion, at this step of the algorithm, the $U_T$ value is recalculated.

Reassessment of the utility of trajectories is calculated taking into account the normalized size of the safe receiving zone according to equation 16:

$$U_T = (U_T + safe\_zone/safe\_zone_{max})/2 \tag{16}$$

Where $safe\_zone$ is the size of the safe receiving zone for the current trajectory, and $safe\_zone_{max}$ is the maximum value of this zone among all possible trajectories for the receiver in question.

*4) Final Utility Reassessment:* After updating the utility assessments for the set of considered trajectories, it is necessary to perform a final recalculation of the $U_R$ value, which includes all possible factors on which the utility of the receiving player depends.

The corrective value of the utility of receivers $U'_R$ is calculated by equation (8).

Reassessment of the utility of received players $U_R$ is carried out according to equation (6).

Received updated utility values determine the final order of consideration of players from the set of possible receivers.

Final ball passing graph is presented in figure 8. In this figure new power vertices $P$ appear on graph. Power vertex is generated by combination of trajectory vertex $T$ and vertices of interceptors $I$ considered in this step of algorithm for each trajectory.

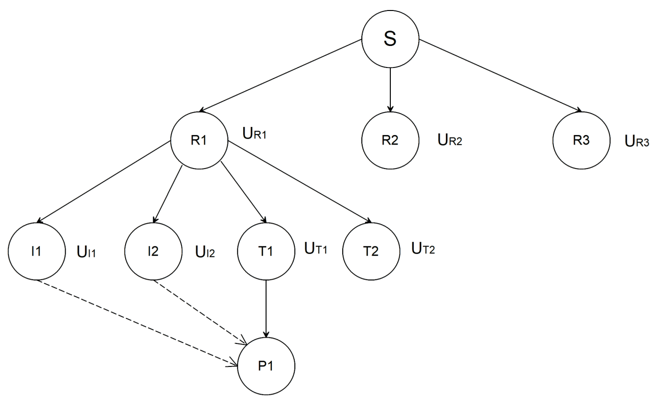


Fig. 8. Final graph of of possible pass parameters combination

## G. Action Selection

After the last reassessment of the utility of receivers $U_R$, it is necessary to go through the graph in depth, at each stage choosing the next most useful vertex as the next step.

As a result of this walk-through, the best values of the trajectory and kick power will be selected, on the basis of which the player in question will perform ball passing.

The presence of utility assessments at each algorithm step ensures the operation of the algorithm under the condition of time constraints.

In the case of finishing of the algorithm before the possible ball transfer trajectories are calculated, the direction to the receiving player with the highest value of the utility value is taken as the resulting trajectory.

In the case of finishing of the algorithm before the optimal kick power is calculated for the trajectory, the maximum possible kick power for the player in question is taken as the result.

### *H. Pseudo-code of Algorithm*

The pseudo-code of the proposed algorithm is shown below.

**Algorithm 1** Pass Implementation

$IA \leftarrow CalculateInfluenceArea(O, T)$
$RA \leftarrow CalculateReachArea(T\_0, B)$
$R \leftarrow FindReceivers(T, IA, RA)$
$U\_R \leftarrow Estimation(R)$
$Ordering(R, U\_R)$
**for** $r \in R$ **do**
  $Alpha, Beta \leftarrow CalculateAlphaBetaAngles(r, IA, RA)$
  $I \leftarrow FindInterceptors(Alpha, Beta, R, B)$
**end for**
$U_{Ir} \leftarrow Estimation(I)$
$Ordering(I, U_{I_r})$
$CorrectEstimation(U_R, U_{I_r})$
$Ordering(R, U_R)$
**for** $r \in R$ **do**
  $AmountTrajectories \leftarrow FindAmountTrajectories(Alpha, Beta, U_R)$
  $T \leftarrow FindTrajectories(AmountTrajectories)$
**end for**
$U_{Tr} \leftarrow Estimation(T)$
$Ordering(T, U_{T_r})$
$CorrectEstimation(U_R, U_{T_r})$
$Ordering(R, U_R)$
**for** $r \in R$ **do**
  **for** $t \in T$ **do**
    $AmountInterceptorsOnTrajectorie \leftarrow FindAmountInterceptorsOnTrajectorie(U_{Tr})$
    **for** $t \in T$ **do**
      $P_{t_i} \leftarrow CalculateParameter(t, i, B, RA, IA)$
    **end for**
    $CorrectEstimation(U_{Tr}, U_{P_{t_i}})$
    $CorrectEstimation(U_R, U_{T_r})$
  **end for**
**end for**
$ActionSelection(R, T, I, P, U_R, U_T, U_I, U_P)$

## III. Results of Experiment

The presented algorithm was developed for agents participating in the Robocup Simulation League Championship, i.e. in a virtual environment. However, it is important to understand the specifics of the championship, namely, how the participation of agents in the game is organized. This happens by connecting a client program located on a separate computer to the game server. The parameters of the machines that will run agent programs are defined and do not differ for participating teams. This fact determines how fast the agent performs actions on the hardware platform and allows testing the algorithm not only for its performance, but also for its applicability in practical conditions, within the real time limits for planning actions.

An experimental study for the situation discussed above was carried out on a computer with an AMD A4-9120 processor, clock frequency 2.20 GHz. These characteristics are similar to the characteristics of computers used in the championship.

The algorithm was programmed in C++ on a computer running OS Ubuntu 18.04.

The goal of the experiment is to make sure that it works as an anytime algorithm, and also to make sure that the algorithm with the utility assessment of each stage works better than the algorithm without the estimation.

### *A. Anytime Algorithm*

The performance of the algorithm as an anytime algorithm can be estimated by getting a graph of the algorithm's performance curve (see Figure 9), which increases as the algorithm execution time increases (figure).

The efficiency of the algorithm is estimated as the number of successful passes on the total number of attempts (in considered measurement equal to 30):

$$efficiency = \frac{successful\ passes * 100}{amount\ of\ attempts} \tag{17}$$

Measurements were performed repeatedly with different amounts of time allocated for the algorithm to work. In total, ten measurements were taken with 5 ms step for available algorithm work time.

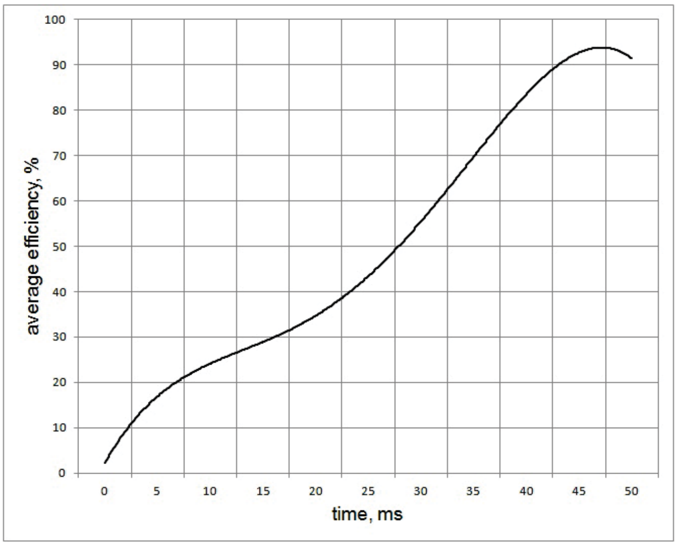


Fig. 9. The graph of the efficiency of the algorithm. The curve grows as the execution time increases.

The graph clearly shows that the algorithm works correctly as an anytime time algorithm.

### *B. Utility Assessment*

The advantage of utility assessment of each stage can be verified by comparing the efficiency curves of the algorithm under the same conditions when the utility assessment is running and changing the order of consideration of receivers, interceptors, and trajectories, as well as when the algorithm is running without these stages (see Figure 10).

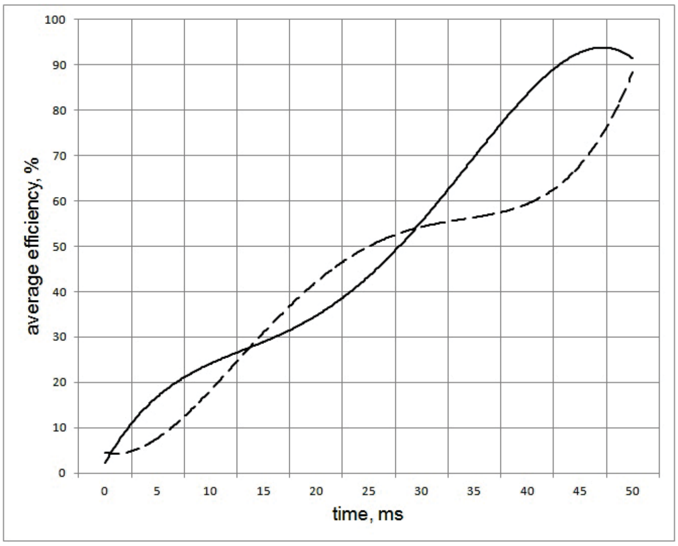


Fig. 10. Comparison of the effectiveness of algorithms with and without utility assessment. Black curve - with estimation. Dotted line - without estimation.

As you can see by looking at the graph, at the stage with a small amount of available time an algorithm with a utility assessment works better than an algorithm without it. With a further increase in time, a decrease in the efficiency growth occurs due to the calculation of utility assessments by the algorithm. However, because of an optimized approach to considering the number of trajectories and interceptors per trajectory, the algorithm with assessments soon gets the advantage again.

## IV. Conclusion

In the course of the research, models and methods of action selection in real-rime were considered: determination and selection of the pass receiver, determination and selection of the limit on the amount of opponents considered as interceptors, determination and selection of the pass trajectory, determination and selection of parameters for kicking the ball command. The results of the study were also used to build the anytime algorithm for pass implementation. the results of the experiment showed that anytime algorithms allow to increase the efficiency of the time used by the agent. The results also showed that the utility assessment at each stage of the algorithm can increase its efficiency.

We plan to improve the efficiency of this algorithm: an improvement of the performance of the algorithm can be achieved, for example, by improving the part with the building of the Voronoi diagram, which takes quite a long time, an improvement of the results of the algorithm results can be achieved by applying more complex criteria for utility assessment or a combination of them. We also plan to use some parts of the research to build algorithms for solving other problems, such as the goal-kicking problem, which is quite similar to pass problem.

The obtained experimental results are focused on the environment of virtual soccer. However, the presented approach, based on dividing one main task of an intelligent agent into sub-tasks and further assessments of their implementation utility for choosing the best way to achieve considered goals, can be extended to other applications of agent systems, which constitutes the direction of further research.

## References


[1] Russell S., Norvig P. Artificial Intelligence: A Modern Approach. - M.: Williams Publishing House, 2007. - 1410 p.

[2] Panteleev M.G., Puzankov D.V. Intelligent agents and multiagent systems, SPbGETU "LETI" Publishing, SPb (2015)

[3] Panteleev 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)

[4] Zilberstein, S. (1996). Using Anytime Algorithms in Intelligent Systems. AI Magazine, 17(3), 73.

[5] RoboCup. The Robot World Cup Initiative. Robocup officiale site. Available at: http://www.robocup.org

[6] 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).

[7] 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

[8] 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)

[9] 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)

[10] Chen M, Dorer K. RoboCup Soccer Server, 150 p. (2003)

[11] Remco de Boer, Jelle Kok, Frans Groen. UvA Trilearn 2002 team description – Faculty of Science, University of Amsterdam (2002)

[12] Dashti H.T. et al. Dynamic Positioning Based on Voronoi Cells (DPVC). In: Bredenfeld 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

[13] Dobrin A. A review of properties and variations of Voronoi diagrams. Whitman College (2005)