A Model-Free Algorithm of Moving Ball Interception by Holonomic Robot Using Geometric Approach

Pavel A. Makarov[0000-0002-1984-556X], Tolga Yirtici, Nurullah Akkaya, Ersin Aytac, Gorkem Say, Gokhan Burge, Berk Yilmaz, Rahib H. Abiyev[0000-0002-3085-6219]

Near East University, Nicosia 99138, TRNC
pavel.makarov@neu.edu.tr

Abstract. In this paper, one common problem for the teams competing in the RoboCup Small Size League (SSL) is addressed, namely the interception of a moving ball at an arbitrary aspect angle relative to the direction of the shot. We present a simple, robust and efficient algorithm for the interception of a moving ball by an omnidirectional SSL robot. The algorithm, designed on the basis of a heuristic approach, requires minimal knowledge of robot dynamics and relies on two key ideas. The first idea is the consideration of ball motion via transition to a reference frame where the ball is static, and the second one is planning the motion of the robot in such a reference frame from the geometric viewpoint. Experiments conducted in a real SSL environment confirmed the beneficial properties of the algorithm: it provides successful interception in a variety of scenarios, characterized by different directions of ball motion and the positional relationships between the ball, robot and goal.

Keywords: interception skill, pass and shoot, mobile robot, holonomic motion, model-free methods.

1 Introduction

Efficient control in a dynamic environment without precise data characterizing the environment and the agent is strongly required for a mobile robot to accomplish various missions concerned with a certain motion scenario. With a sufficiently accurate model of robot motion, the objective is attainable using the methods of optimal control theory, once the mathematical formulation for the “most desirable” motion is given in terms of cost function. In the absence of such a model, the machine learning approach, which is rather trendy nowadays, appears to be a reliable and flexible solution. However, machine learning usually needs a huge amount of training data, and significant computational resources. Another potential hurdle is the fine tuning of the resulting controller, as the approach features almost no transparency. Alternatively, heuristic methods can be applied to obtain a fast and practical solution for the problem. The main drawback of the heuristic approach is its weak universality, *i.e.* the necessity to design a separate algorithm for each particular objective. Nevertheless, a well-designed heuristic algorithm can provide near-optimal performance of the robot within the scope of the objective, with minimal development and computational costs.

In this paper we address a common problem experienced by the teams competing in the RoboCup Small Size League (SSL), namely the interception of a moving ball at an arbitrary aspect angle relative to the direction of the shot. Recently, the "NEUIslanders" SSL team, whom the authors are with, has significantly improved in this area, using the aforementioned heuristic approach to create the relevant control algorithm without the knowledge of a robot motion model.

SSL teams use omnidirectional four-wheel (rarely – three-wheel) cylinder-shaped robots, equipped with a single kicker. Computer renderings of the "NEUIslanders" soccer robot are depicted in Fig. 1. As a rule, the translational motion of the robots is controlled via set velocity (two components). Both robot and ball tracking is provided by common vision system relying on several ceiling-mounted cameras. Commands for all robots of a team are generated on a single off-board computer and transmitted to the agents. Therefore, measurement errors together with the delays in the delivery of vision data and commands are typical factors that should be taken into account in the design of any control algorithm.

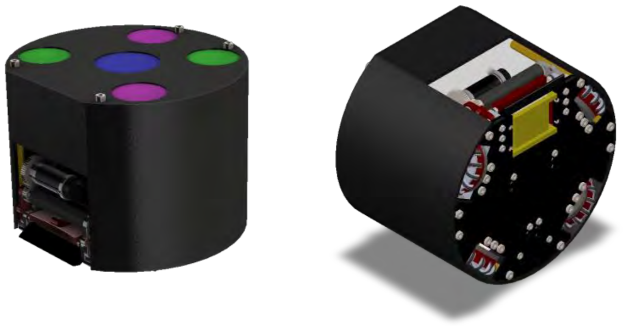

Fig. 1. Soccer robot of the "NEUIslanders" SSL team and its omnidirectional chassis.

The problem of ball interception is reduced to the alignment of the intercepting robot's kicker with the ball at a certain moment of time in such a way that robot orientation at this moment corresponds to the aimed direction of the shot. For particularity only, we will assume that the robot aims at the adversary's goal. A natural requirement for efficient interception is to minimize the time elapsed until the shoot.

Unfortunately, few works on the problem have been reported in literature. Three different techniques – precise computation of the interception point, reinforcement learning and the method of qualitative velocity – are considered in [1], but all within the framework of the RoboCup soccer simulator (Soccer Server), where the major part of real-world adverse factors is ignored. The same limitation concerns [2]: the authors suggest a combination of reinforcement learning with the off-line computation of an optimal trajectory for the agent, and validate their approach through computer simulations only. As for the teams of the RoboCup soccer community, the prediction of future ball location and the relevant planning of interceptor trajectory are primarily used, according to the available releases [3, 4, 5]. It should be noted that the prediction-based approach is not free of some inherent shortcomings, emanating, on the one hand, from the limited prediction accuracy owing to measurement errors for the instantaneous ball speed, combined with the uncertainty of friction deceleration [6], and, on the other hand, from the necessity to estimate the required time for the robot to attain a certain position.

The algorithm presented in this work has been developed with the aim of providing a robust, reliable and easy-tunable interception skill, avoiding the drawbacks of prediction-based methods, to our robots. Optimal performance in terms of rapidity was not demanded, but the ability to align the kicker with the ball (static or rolling) regardless of the direction of ball motion and the positional relationship between the ball, robot and goal was a strict requisite. Examples of nontrivial interception scenarios are schematically depicted in Fig. 2: we will refer to these cases as head-on, cross and overtake interception.

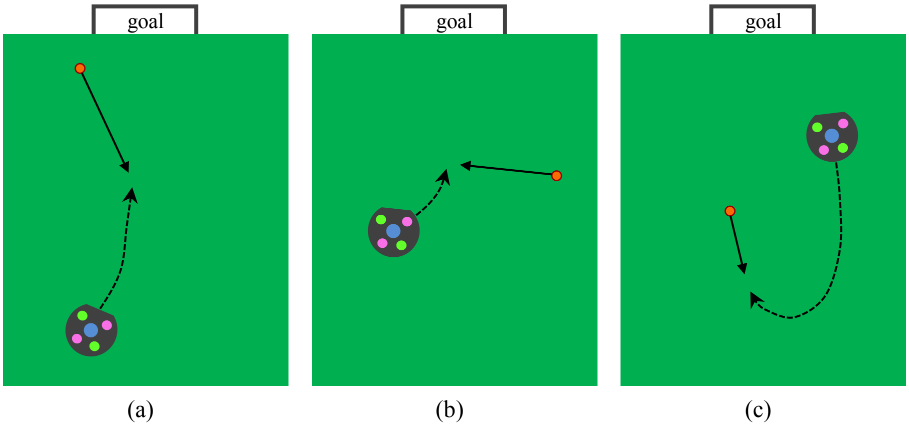

Fig. 2. Ball interception scenarios: (a) head-on, (b) cross and (c) overtake interception.

2 Interception Algorithm

2.1 Overview

The inputs for the proposed algorithm are the current position vectors $\mathbf{r}^{\mathrm{r}}$ and $\mathbf{r}^{\mathrm{b}}$ of the intercepting robot and the ball accordingly, ball velocity $\mathbf{V}^{\mathrm{b}}$, and polar angle φ specifying the desired shooting direction. Vectors $\mathbf{r}^{\mathrm{r}}$, $\mathbf{r}^{\mathrm{b}}$, $\mathbf{V}^{\mathrm{b}}$ are obtained from the tracking system (a Kalman filter is conventionally used for this purpose). The algorithm produces set velocity $\mathbf{V}^{\mathrm{set}}$ for the intercepting robot as the output, and runs at each iteration of the control loop independently, avoiding the necessity to store any specific data in memory.

The first key idea of the algorithm is to consider robot motion in the reference frame, where the ball is static. Formally, we determine the *relative* set velocity $\mathbf{W}^{\mathrm{set}}$ for the robot with respect to the moving ball first, and then convert it into the absolute set velocity $\mathbf{V}^{\mathrm{set}} = \mathbf{W}^{\mathrm{set}} + \mathbf{V}^{\mathrm{b}}$. This concept, on the one hand, ensures robot alignment with the moving ball under condition that $\mathbf{W}^{\mathrm{set}}$ is produced by a reliable algorithm of static ball interception, and, on the other hand, serves as a good alternative to prediction-based techniques.

The second key idea, related to static ball interception, is prompted by the lack of knowledge about robot dynamics. Assuming that only a few reference characteristics of robot motion are known, such as maximal speed and admitted acceleration, we have chosen a geometric approach as the most suitable for the synthesis of the algorithm. Namely, a family of desired paths of approach to the ball, where each path depends on the initial position of the interceptor, is introduced, and the field of set velocities $\mathbf{W}^{\mathrm{set}}(\mathbf{r}^{\mathrm{b}})$ is set in order to guide the robot along the desired path, avoiding significant deviations from it. By "significant deviations" we mean discrepancies that can lead to the failure of interception.

2.2 Detailed Description

Relative set velocity $\mathbf{W}^{\mathrm{set}}$ is representable by its magnitude and direction. Unit vector $\mathbf{n}^{W}$, specifying the direction of $\mathbf{W}^{\mathrm{set}}$, is determined as follows. First, we compute the target robot position (robot coordinates are defined as the coordinates of its center) with respect to the current ball location as $\mathbf{r}^{\mathrm{trg}} = \mathbf{r}^{\mathrm{b}} - d \cdot \mathbf{n}^{\mathrm{shoot}}$, where $\mathbf{n}^{\mathrm{shoot}}$ is the unit vector pointing in the direction of the shot, and d denotes the distance from the center of the robot to the ball when the latter is aligned with the kicker. It is assumed that robot orientation corresponds to $\mathbf{n}^{\mathrm{shoot}}$. Then, two virtual circles of radius ρ are introduced around the ball, as depicted in Fig. 3a: these circles are located symmetrically about the line joining $\mathbf{r}^{\mathrm{trg}}$ and $\mathbf{r}^{\mathrm{b}}$ (shoot line), and intersect at $\mathbf{r}^{\mathrm{trg}}$ with the angle 2α between their tangent lines. The circles should serve as some boundaries to be rounded by robot trajectory.

The following simple logic underlies the computation of $\mathbf{n}^{W}$:

- if the robot is behind the ball (relative to shooting direction) within the 2α-segment, it can move directly to the target position;
- if the robot position does not fall into the 2α-segment, and the robot is outside the virtual circles, it is directed along the tangent line to one the circles, dropped through the current position of the robot, so that the circle will be bent round. The choice of the circle depends on which side from the shoot line the robot is located;
- if the robot occurs inside one of the circles (which is quite possible when it bends a circle round), the direction is set as tangent to the closest point of the circle and turned slightly to push the robot outside. The "deeper" is the robot, the stronger is the turn of the vector n^{W}.

Omitting the formulas describing such logic, we illustrate it by Fig. 3b, where the field of relative set velocity is plotted on a rectangular grid of robot positions.

The found vector n^{W} is multiplied by certain set speed W: $\mathbf{W}^{\mathrm{set}} = \mathbf{n}^{\mathrm{set}} \cdot W$. The latter is equated to maximal robot speed $V_{\max}$ when the robot is far enough from the ball, and is decreased as it comes closer. The idea of decreasing the relative set speed is twofold: on the one hand, it allows the robot to bend the circles round more accurately, reducing its acceleration; on the other hand, it ensures better kicker alignment under the adverse conditions (measurement errors and delays) typical in real-world operation. In our implementation we use the formulas

$$W = \min\{W_0, V_{\max}\},\quad W_0 = W_{\text{close}} + \frac{\left|\mathbf{r}^{\text{r}} - \mathbf{r}^{\text{trg}}\right|}{D_{\text{close}}}(V_{\max} - W_{\text{close}}),$$

where W_{close} stands for the relative speed when the robot is close to the ball, and D_{close} is the distance at which we start to decrease speed.

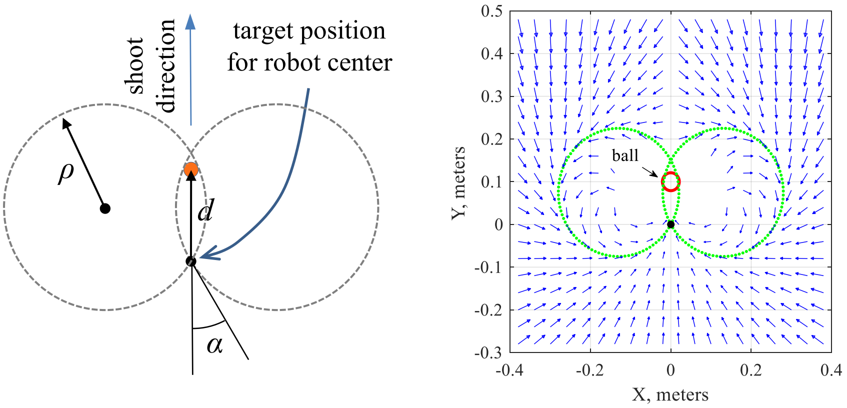

Fig. 3. (a) The geometry used to determine the direction of relative set velocity, and (b) relative set velocity calculated for different robot positions by the proposed algorithm.

Radius ρ is specified with respect to some admitted robot acceleration a_{adm}. According to the centripetal acceleration formula, one should assign this radius as $\rho = W^2/a_{\text{adm}}$, where the reasonable choice for W, under the presumption that the virtual circles are rather small, is W_{close}. However, we should take into account that the intercepted ball is not stationary, hence *absolute* robot speed might significantly differ from W_{close}. Owing to a greater load on the electric motors, the behavior of the robots at the disposal of "NEUIslanders" (and presumably the same problem is faced by all omnidirectional robots with limited motor power) is less stable at high speeds, which confines robot maneuverability in case of rapid motion. Concerning the algorithm, it means that the radius of turn should be increased in order to suppress unpredictable deviations from the nominal trajectory. We have come to the following *ad-hoc* formulas assigning the radius of turn:

$$\rho = \min\{\rho_{\max}, \max\{\rho_{\min}, \rho_0\}\},\quad \rho_0 = \frac{\left(\kappa\left|\mathbf{v}^{\text{b}}\right| + W_{\text{close}}\right)^2}{a_{\text{adm}}},$$

with $\rho_{\min}$ and $\rho_{\max}$ confining the range of ρ, and coefficient κ introduced to consider the increase of robot speed due to ball motion.

Once $\mathbf{W}^{\text{set}}$ is computed, we determine the output set speed $\mathbf{V}^{\text{set}}$ as

$$\mathbf{V}^{\text{set}} = \mathbf{W}^{\text{set}} + \mathbf{V}_0^{\text{b}}, \quad \mathbf{V}_0^{\text{b}} = \mathbf{V}^{\text{b}} \cdot \min\left\{1, \frac{|\mathbf{v}^{\text{b}}|}{V_{\max}}\right\}.$$

In these formulas, $\mathbf{V}_0^{\text{b}}$ is ball velocity, clipped by the maximal robot speed – this operation was introduced to avoid unnecessary back motion of the robot when the ball is rapidly rolling towards it.

Table 1 summarizes all parameters present in the interception algorithm and their assigned values in its current implementation by the “NEUIslanders”. The majority of the listed values were selected empirically – due to the lack of time we had no opportunity to create a tool, which could automate the search of the optimal ones.

Table 1. Parameters of ball interception algorithm.

Notation	Description	Value
d	Distance from robot center to the ball aligned with the kicker	10 cm
$\rho_{\min}$	Minimal radius of virtual circle	15 cm
$\rho_{\max}$	Maximal radius of virtual circle	40 cm
α	Half-angle between the tangents to virtual circles	30 degrees
a_{adm}	Nominal admitted acceleration of robot	2.25 m/sec^2
$V_{\max}$	Maximal robot speed	2.8 m/sec
W_{close}	Assigned relative speed in the vicinity of the ball	0.35 m/sec
D_{close}	Distance to the target point, when the approach speed is reduced	1.5 m
κ	Coefficient for ball speed consideration	0.7

3 Experimental Results

The experiments conducted in the real SSL environment (soccer field, vision system, *etc.*) confirmed the robustness and efficiency of the presented algorithm with parameters according to Table 1. Visually, the behavior of the intercepting robot resembles the motion to a predicted point, and the performed trajectories do not look redundant for any of the interception scenarios, with the possible exception of the final stage, when the robot has to be aligned (in velocity) with the moving ball.

The figures below present the observed interceptor behavior in dynamics, with the time step of 0.15 sec. The coordinates used for presentation were obtained from raw vision data, and interpolated to the appropriate time grid. Small vectors laid from the center of the robot indicate the set velocity generated by the algorithm at each moment of time. Fig. 4 depicts the head-on scenario; Fig. 5, Fig. 6 and Fig. 7 – different cases of cross interception, and Fig. 8 – overtake interception. One can clearly see that in all cases the robot is able to shoot towards the goal at the time of the final snapshot.

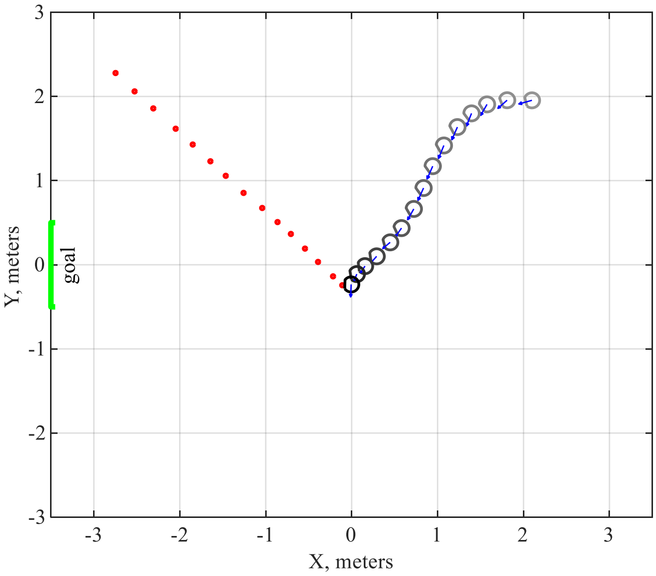

Fig. 4. Robot and ball motion: head-on interception.

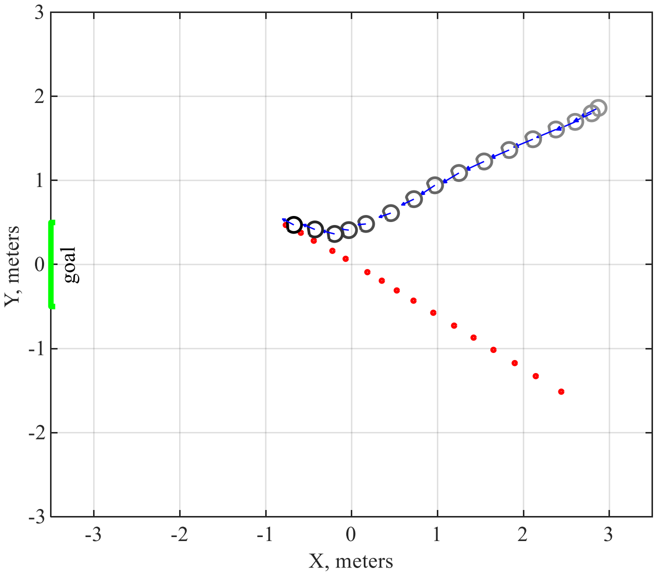

Fig. 5. Robot and ball motion: semi-cross interception.

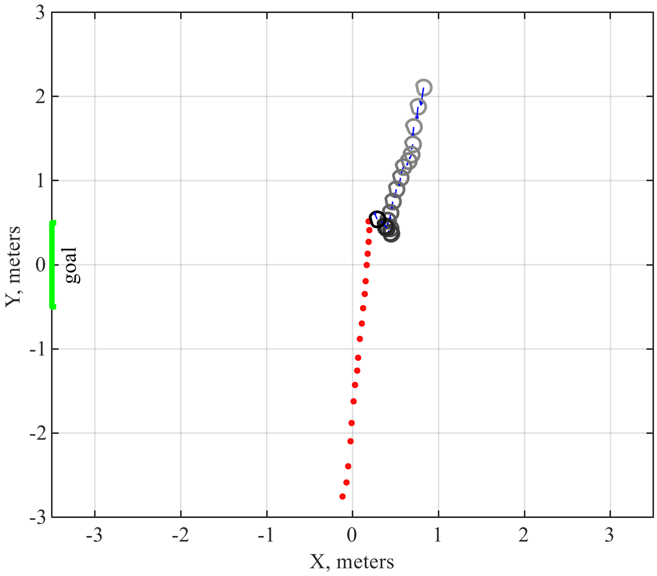

Fig. 6. Robot and ball motion: cross interception towards the ball.

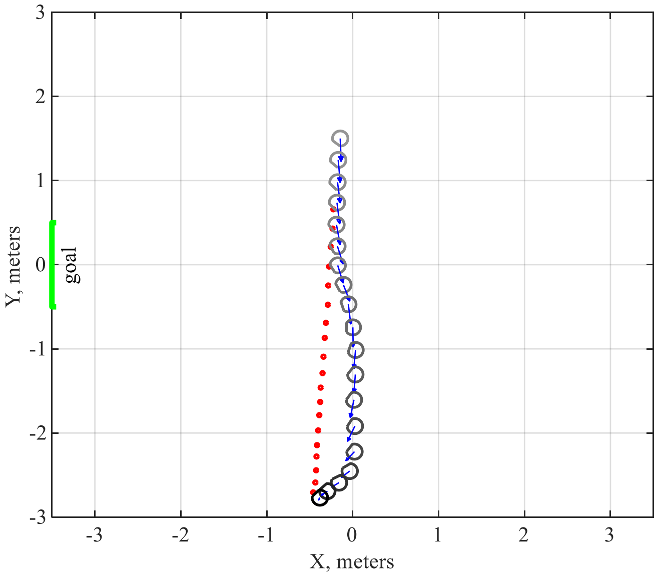

Fig. 7. Robot and ball motion: cross interception with overtaking.

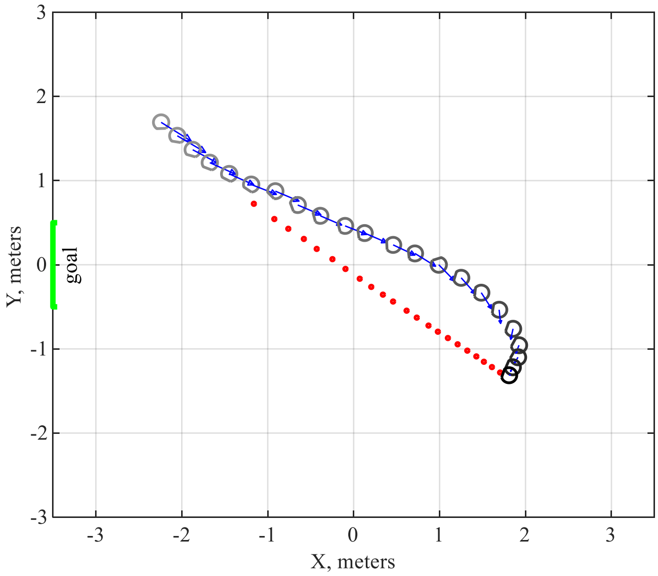

Fig. 8. Robot and ball motion: overtake interception.

It should be noted that the change of the direction of robot motion, observed in Fig. 5 and Fig. 6, is explained by the above-mentioned velocity alignment as the robot and the ball approach to each other. Another noticeable feature is a considerable distance between the trajectories of the ball and the robot in overtake mode (see Fig. 8). This behavior results from the increase of the virtual circle radius for the maneuver when the interceptor has to move fast.

4 Conclusion

Using a heuristic approach, a simple, robust and efficient algorithm for the interception of a moving ball by an omnidirectional SSL robot was developed. The algorithm requires minimal knowledge of robot dynamics and relies on two key ideas. The first idea is the consideration of ball motion via transition to a reference frame where the ball is static, and the second one is planning the motion of the robot in such reference frame from the geometric viewpoint. Experiments conducted in a real SSL environment confirmed the beneficial properties of the algorithm: namely, it provides successful interception in a variety of scenarios, characterized by different directions of ball motion and the positional relationships between the ball, robot and goal.

References

1. Stolzenburg F., Obst O., Murray J.: Qualitative Velocity and Ball Interception. In: Jarke M., Lakemeyer G., Koehler J. (eds.) KI 2002: Advances in Artificial Intelligence, vol. 2479, pp. 283–298. Springer, Berlin, Heidelberg (2002).
2. Maire F., Taylor D.: A Quadratic Programming Formulation of a Moving Ball Interception and Shooting Behaviour, and its Application to Neural Network Control. In: Stone P., Balch T., Kraetzschmar G. (eds) RoboCup 2000: Robot Soccer World Cup IV. RoboCup 2000. Lecture Notes in Computer Science, vol 2019, pp. 327–332. Springer, Berlin, Heidelberg (2001)
3. Maeda K., Kohketsu A., Takahashi T.: Ball-Receiving Skill Dependent on Centering in Soccer Simulation Games. In: Asada M., Kitano H. (eds.) RoboCup-98: Robot Soccer World Cup II, pp. 152–161. Springer, Berlin, Heidelberg (1999).
4. Bowling M., Veloso M.: Motion Control in Dynamic Multi-Robot Environments. In: Veloso M., Pagello E., Kitano H. (eds.) RoboCup-99: Robot Soccer World Cup III, pp. 222–230. Springer, Berlin, Heidelberg (2000).
5. M. M. Rahimi, M. M. Shirazi, M. Arfaee, M. A. Najaf Gholian, A. H. Zamani, H. Hosseini, F. Hashemi Chaleshtori, N. Moradi, A. Ahsani, M. Jafari, A. Zahedi, P. Abdollahi, A. Zolanvari, M. A. Khosravi: Parsian Extended Team Description for RoboCup. Robocup SSL, Nagoya, Japan (2017).
6. R. H. Abiyev, P. Makarov, A. Cagman, E. Aytac, G. Burge, A. Turk, N. Akkaya, T. Yirtici, G. Say, B. Yilmaz: NEUIslanders Team Description Paper RoboCup 2019. Robocup SSL, Sydney, Australia (2019)