STOx's 2016 Extended Team Description Paper

Saith Rodríguez, Eyberth Rojas, Katherín Pérez, Carlos Quintero, Oswaldo Peña, and Juan Calderón

Faculty of Electronics Engineering
Universidad Santo Tomás
Bogotá, Colombia
{saithrodriguez,eyberthrojas,kathyperez,carlosquintero,
oswaldopena,juancalderon}@stoxs.org
http://www.stoxs.org

Abstract. This paper shows the current state of the STOx's team after its last participation in the latest RoboCup world championship. Among the most important characteristics we see that STOx's has achieved a number of goals converted per game of 2.71, while only received 0.43 goals per game, the lowest compared to previous participations. Also, we show two important tools developed and implemented in our team's software platform: one that allows us to identify and reconstruct the parabolic path followed by the ball after a chip kick with an average error of the estimated landing point of less than 6cm. The other tool is used to calculate a set of key points to construct dynamic plays after an indirect free kick. The developed algorithm provides locations that are good candidates for receiving the pass and shoot to the goal with high chances of scoring and is based on a constrained optimization problem.

1 Introduction

Our team has been participating in the RoboCup initiative since 2011 [1] with its fifth participation in the latest RoboCup version in 2015 [2]. During the experience lived in the RoboCup events, the relationships and talks with other participants and the TDPs of the most experienced teams [3] have allowed us to assemble a team that has grown and improved over time.

The evolution of the main technical aspects of the league has provided the teams with the possibility to explore a large variety of new cooperative techniques [4, 5], faster games [6], more dynamic plays and a highly attractive show for the general audience and soccer fans. Specifically, the extension of the field dimensions has given new room for the fast robot agents to perform more creative, free and dynamic games. The decisions in terms of accurately predicting the behavior and actions of the most competitive teams have become also an interesting and yet open challenge [7].

The robots have had little changes since the incorporation of the chip kickers, more powerful traction motors of some teams and some customized modifications that certain teams have added. However, new major changes that add functionalities to the current state of the robots may be included in the near future in order to keep evolving the league. At the end of the day, the SSL is a challenge that pushes forward the state of the art, not only in the field of artificial intelligence, but also in robotics.

In this paper, we start discussing the current state of the STOx's team, specifically by showing its overall performance in the latest RoboCup championship and highlighting some of their most important characteristics. Afterwards, we show the development of a technique that has allowed us to predict the trajectory of the ball after it has been kicked by a chip kicker. Finally, we describe a highly important framework used by our team in order to discover points with high scoring chances and creating dynamic attack plays. In all sections we show descriptions, simulations and results of our techniques.

2 Current state of the STOx's team

With only its fifth participation in the RoboCup world championship, the STOx's team still a young team in the league and there are still large amounts of possibilities to improve the team's technical abilities. Since we started our participation in 2011, every year, major changes were required to significantly increase the team's performance, especially in the robot's electronics and mechanics [8, 9]. During these 5 years, we have designed and built 3 different generations of robots that strive for more robust, accurate, fast and reliable agents; a key factor in the development of a successful SSL team. After profound changes from one year to the next in the robots we have finally achieved a state where only minor changes and maintenance are required between consecutive tournaments. This is a highly important state for young teams to start the creation and development of more interesting and dynamic AI techniques.

For RoboCup 2016, the general mechanics and electronics of the robots will remain the same as the last year. Only a minor set of modifications and maintenance actions are currently being carried out. One that may be worth to mention is the reinforcement and protection of the dribbler motors since many of them ended broken after the last competition. For a more detailed view of our robot's main characteristics see [2, 8].

With a set of new dynamic attack plays and heavily reinforced defense strategies we were able to achieve in RoboCup 2015 our best participation in a world championship event: enter the top 4 teams in the league. The path to this achievement consisted on 7 official games: 3 in the round robin, 1 in lucky loser, 1 in quarter finals, 1 in the semifinal and 1 for third place.

Table 1 shows a set of statistics extracted from the game logs in RoboCup 2015 that aims at describing the main STOx's attributes in attacking and defense.

The first thing to notice is the amount of goals received per played game (0.43) which is the lowest in all of our participations in the RoboCup championships. The amount of converted goals, 19 and normalized by the number of games 2.71 is also by far, the largest achieved by our team.

Table 1. Matches statistics - STOx's RoboCup 2015

	G1	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
Goals received	0	0	0	0	0	2	1
Goals saved by goalie	1	0	0	0	0	5	3
Goals saved by defenders	2	0	1	0	6	19	6
Intercepted passes	2	0	0	4	3	6	2
Goals converted	0	10	5	3	1	0	0
Shots attempted	7	7	2	8	9	1	2
Passes attempted	0	0	1	1	3	0	0

Table 2 and Table 3 show a summary of the attacking and defense most important statistics during the championship. Our team's conversion rate (35%) shows how effective is our attack to convert a goal after we are able to shoot to goal.

In the defense strategies, we count shots from the opponent that aimed at our goal that were saved by our goalie or by our defenders. The first thing to notice is the high amount of saved goals which shows a strong characteristic of our team's participation in RoboCup 2015: its strong defense. The goals saved by the goalkeeper, although much less than those saved by the defenders, happened on highly difficult games (G6 and G7) under very fast and closed plays.

Table 2. Summary of the statistics for attacking strategies of the STOx's team in RoboCup 2015

Shots attempted	55
Goals converted	19
Shots missed or blocked	36
Conversion rate	35%

Table 3. Summary of the statistics for defense strategies of the STOx's team in RoboCup 2015

Goals saved	40
By Goalie	9
By Defenders	31
Blocking rate	93%

3 Automatic identification and reconstruction of a chip kick

Since 2005, many of the teams participating in the SSL adopted a kicking device for their robots that added one new capability; that of making passes and shots to the goal by lifting the ball from the ground. With these chip kickers the teams did not require to find a clear path to perform a pass or shoot to goal, widening the game possibilities and hence making it hard for the opponents to block these passes or potential goals. Furthermore, the vision system of the SSL creates a mapping from a 3D real world to a virtual world made out of, mainly, 2D positions and velocities, making the chip kicks even more difficult to detect.

Today's gameplay in the SSL is even faster, more dynamic and more intelligent and the chip kicks are a natural and fundamental piece of this elegant game. Modern AI strategies of the participating teams require fast computation and prediction of the opponent team's behavior in order to take appropriate actions. This accurate prediction is a key factor in many of the current team's sophisticated AI techniques[5, 7].

Concretely, the accurate prediction of the landing point of the ball after a chip kick is a very important task, for both, defending and attacking plays. It could be used to decide whether a defender should release a mark or not or if the goalkeeper should try to block a shoot or not. Also, if the prediction is not correctly calculated, the pass of a team mate may fail, leaving the control of the ball in the opponent's possession. A dangerous pass of the opponent team may not be correctly blocked or a defender robot may not have enough time to block an opponent's pass.

In all cases, this prediction should be made as fast as possible since it may be useless if the ball already finished its trajectory. The following sections show our approach to accurately identify and predict a chip kick as fast as possible.

3.1 Identification of a parabolic flight

The identification of a chip kick or parabolic flight when using one camera (or set of cameras) in general position may be not straightforward. From the system's perspective, the ball has only a two-dimensional position in each frame with no clue of its elevation with respect to the ground, whatsoever.

A simple observation yields to the conclusion that the apparent ball trajectory (i.e., the one detected by the camera) when there is a chip kick looks in the vision system as a curve path. However, the noise in the vision system and the amount of possible curves that exist when chipping from different places in the field to other different locations make it hard to come up with a simple thumb rule for detecting whether the given shoot was a chip kick or not. Fig. 1 shows a representation of how a chip kick looks in the vision system and the real path the ball follows.

The first step is to identify the approximate moment when the robot kicked the ball. This is easy achievable by tracking the derivative of the ball's velocity

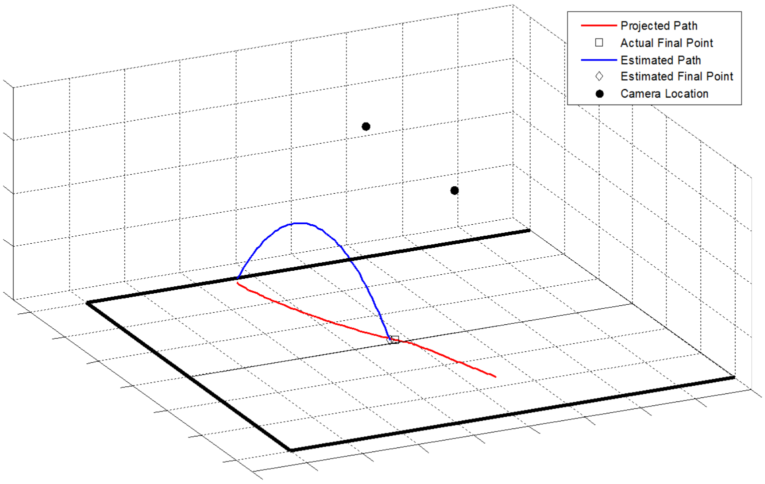
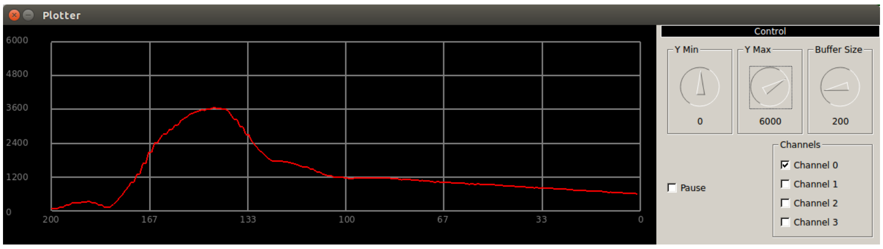


Fig. 1. Representation of the projected path followed by the ball in a chip kick, how the real parabolic path may look like and the actual and estimated ball's landing points

from frame to frame. When the magnitude of the velocity's derivative shows a peak, then a new shot has been kicked. In other words, we start counting a shot when the velocity of the ball suddenly increases. Fig. 2 shows the velocity profile of a ball that has just been kicked.



Y Min 0
Y Max 6000
Buffer Size 200
□ Pause
Channels
☑ Channel 0
□ Channel 1
□ Channel 2
□ Channel 3

Fig. 2. Typical ball's velocity profile after it has been kicked

We have implemented a simple learning machine capable of identifying whether a given set of frames, corresponding to a new shot is a parabolic flight or not based on previous examples. We need first to extract a set of features for each new shot with the information required to identify a chip kick from a different type of kick. Based on this, we used for each kick, the nine angles formed between its first ten consecutive apparent ball positions, as features to predict whether it is a parabolic flight or not. These angles look like those in Fig. 3

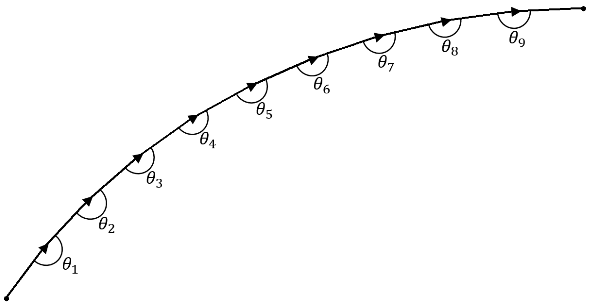


Fig. 3. Angles between ten consecutive apparent ball positions in a parabolic shot

According to the figure, each shoot is represented as a 9–dimensional vector, i.e., $x_i \in \mathbb{R}^9$. We have collected data from the RoboCup 2015 game logs and manually labeled each shot in the logs as chip kick or not chip kick, obtaining 270 chip kicks and 300 not chip kicks. This dataset is then divided into two disjoint sets namely the training set and the testing set to perform cross validation: this is, the training process used to build the classifier is performed using only the data in the training set whereas the data in the testing set is only used to assess the generalization capability of the classifier, i.e., its classification performance on new data.

For this task, we have selected a support vector machine (SVM) binary classifier with gaussian kernel. SVMs are recognized to be very powerful learning machines for general problems capable of finding highly complex non-linear classification boundaries [10]. The idea behind SVMs consists on finding a hyperplane that separates instances of two different classes with maximum distance between the hyperplane and the closest training data point. More formally, the SVM algorithm solves the following optimization problem:

$$\arg\min_{w,\zeta,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\zeta_i \tag{1}$$

subject to $y_i(wx_i - b) \geq 1 - \zeta_i$, where $\zeta_i > 0 \forall i = 1, \ldots n$ are slack variables, w is the vector that defines the classification surface and C is a regularization parameter that avoids overfitting to the training data.

The nonlinear capabilities of the SVMs come from the kernel trick [10]. The idea is to find a new representation of the original data into a new space where a hyperplane suffices to separate the new transformed data. This transformation is usually achieved through a special function known as the kernel without the need to explicitly mapping the data. The gaussian kernel is one of the most general and widely used kernel functions:

$$k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2) \text{ for } \gamma > 0 \tag{2}$$

The process of cross validation previously described is useful also to select the parameters involved in the learning procedure of a learning machine, namely the value of the regularization parameter C and the value of the kernel parameter γ. The general procedure consists on performing a grid search in the space of the parameters and perform the training using a specific combination of the parameters. Each combination of the parameters is valued according to its performance on the testing set to finally select the combination that yields to the lowest generalization error.

We have trained a SVM with the characteristics described above using the LibSVM library [11] and attained 97.4% of accuracy in the testing set which allows us to automatically identify whether a new shot during a game is a chip kick or not after its 10 initial frames. After trained, our classifier can be easily plugged into our AI software to identify whether a given shoot is a chip kick or not using only the support vectors.

3.2 Reconstruction of the parabolic path

The reconstruction of the parabolic path of the ball when kicked by a robot using its chip kicker by using only a few initial observations is the final task to solve.

For this task, we have implemented a method based on [12] that uses the 3 initial frames to directly reconstruct the parabolic path of the ball by solving a system of linear equations. The first step consists on transforming the coordinates of an object captured by the camera (camera coordinates) to the world coordinates by using the rotation matrix R and the translation vector l. If the point where the parabolic flight starts is (x_0, y_0, z_0) and the velocity of the ball during its trajectory is $v = (v_x, v_y, v_z)$, then, the camera's coordinates of the path is:

$$(x, y, z) = \left(x_0 + v_x t + \frac{1}{2} g_x t^2, y_0 + v_y t + \frac{1}{2} g_y t^2, z_0 + v_z t + \frac{1}{2} g_z t^2 \right) \tag{3}$$

where t is the time since the shoot started and (g_x, g_y, g_z) are the components of the earth gravity, which must be transformed into the camera coordinates using the rotation matrix R. For details, see [12].

If m points $(x_i, y_i, z_i), \forall i = 1, 2, ..., m$ are taken at times $t_i, \forall i = 1, 2, ..., m$ respectively, then, the projection of the position of the ball in the image plane of the camera is:

$$\left(\frac{x'_i}{z'_i}, \frac{y'_i}{z'_i} \right) = \left(\frac{x_0 + v_x t_i + \frac{1}{2} g_x t_i^2}{z_0 + v_z t_i + \frac{1}{2} g_z t_i^2}, \frac{y_0 + v_y t_i + \frac{1}{2} g_y t_i^2}{z_0 + v_z t_i + \frac{1}{2} g_z t_i^2} \right) \tag{4}$$

If $\alpha_i = \frac{x'_i}{z'_i}$ and $\beta_i = \frac{y'_i}{z'_i}$, then:

$$\alpha_i = \frac{x_0 + v_x t_i + \frac{1}{2} g_x t_i^2}{z_0 + v_z t_i + \frac{1}{2} g_z t_i^2}$$

$$\begin{aligned}\alpha_i z_0 + \alpha_i v_z t_i + \alpha_i \frac{1}{2} g_z t_i^2 &= x_0 + v_x t_i + \frac{1}{2} g_x t_i^2 \\ \alpha_i z_0 + \alpha_i v_z t_i - x_0 - v_x t_i &= -\frac{1}{2} g_z \alpha_i t_i^2 + \frac{1}{2} g_x t_i^2 \qquad (5)\end{aligned}$$

$$\begin{aligned}\beta_i &= \frac{y_0 + v_y t_i + \frac{1}{2} g_y t_i^2}{z_0 + v_z t_i + \frac{1}{2} g_z t_i^2} \\ \beta_i z_0 + \beta_i v_z t_i + \beta_i \frac{1}{2} g_z t_i^2 &= y_0 + v_y t_i + \frac{1}{2} g_y t_i^2 \\ \beta_i z_0 + \beta_i v_z t_i - y_0 - v_y t_i &= -\frac{1}{2} g_z \beta_i t_i^2 + \frac{1}{2} g_y t_i^2 \qquad (6)\end{aligned}$$

(5) and (6) form two equations with six variables. At least 4 more equations are required to find a solution for the $6th$ order linear equation system, requiring at least 2 more samples for a total of 3. However, due to noise in the vision system, the more samples are included in the estimation of the parabolic flight, the higher the accuracy of the reconstruction. We have implemented this technique on 60 different chip kicks made from different locations in the field.

Fig. 4 shows the reconstructed parabolic paths computed by the algorithm using $3, 4, ..., 60$ samples. It is possible to confirm that the reconstructed paths that use more samples are closer to the actual parabolic flight than those that use less.

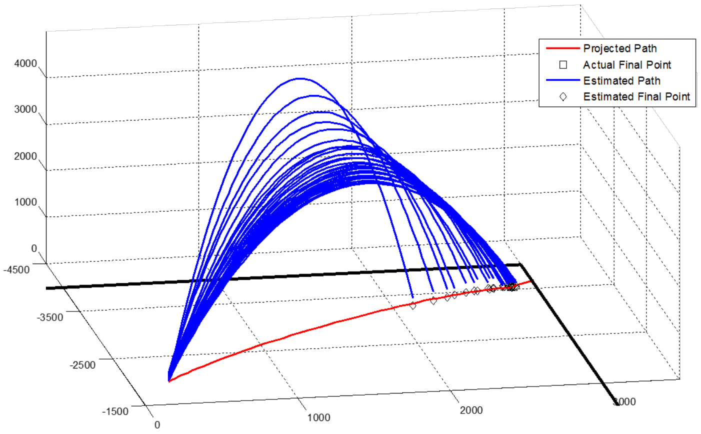


Fig. 4. Reconstructed parabolic flights computed by the algorithm using a different amount of samples

Also, Fig. 5 shows the average error of the estimated final point for all 60 kicks for a different number of samples used to reconstruct each path. The average error decreases as expected and shows that it is already fairly low for more than 10 samples.

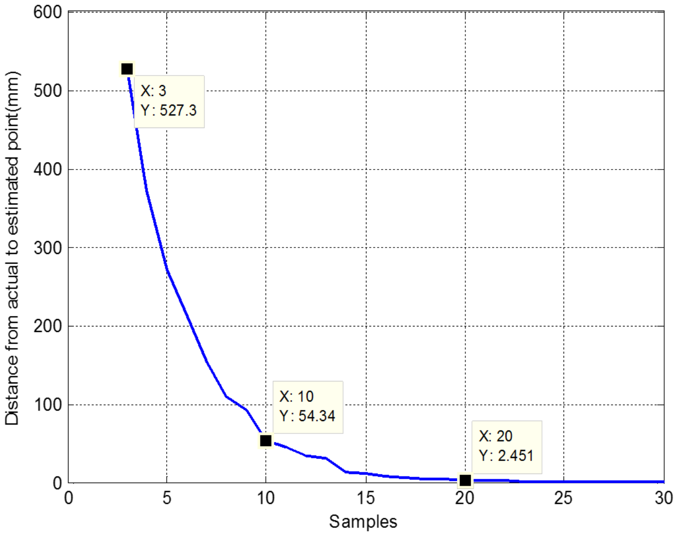
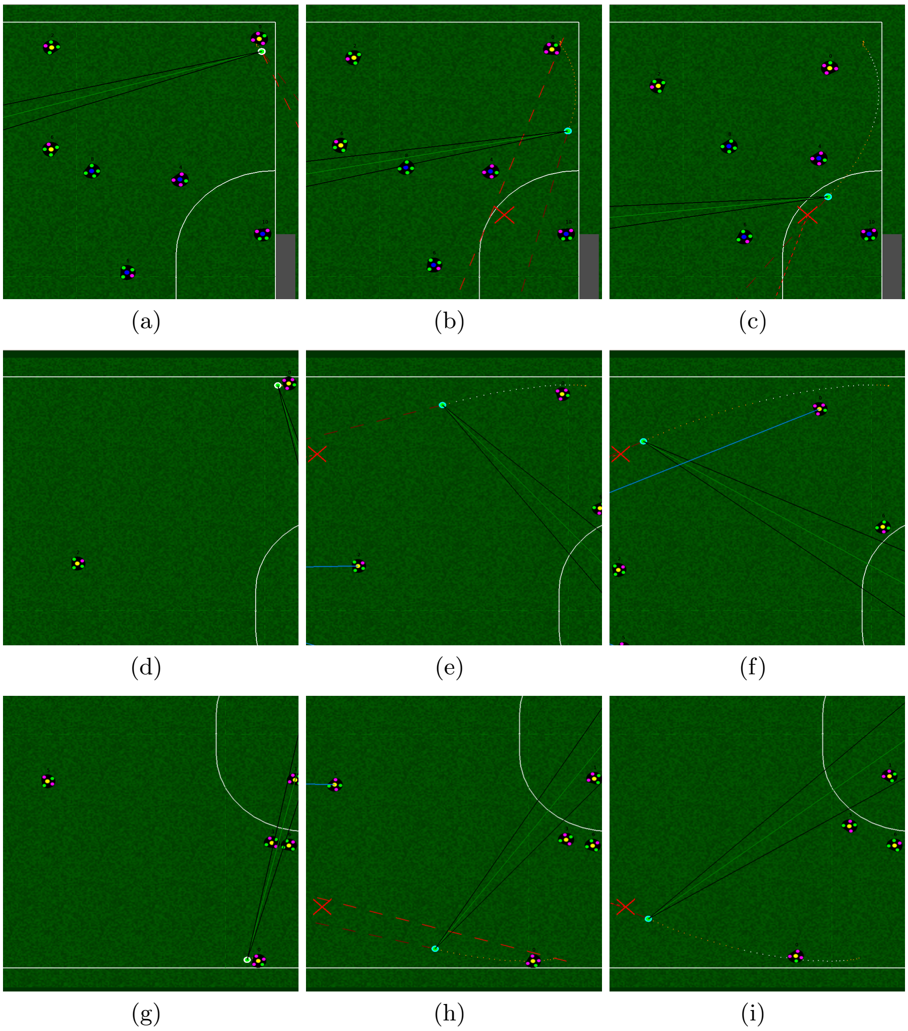


Fig. 5. Average error of the estimated final point for all 60 kicks for a different number of samples used to reconstruct the parabolic paths.

These results show that we can successfully identify and reconstruct the parabolic flight of a ball kicked by a robot during a game using no more than 10 samples. First, a new shoot is detected by monitoring sudden changes in the ball's velocity. At that moment, we compute the angles between the first 10 ball positions and use such information to detect whether the shoot was a chip kick or not. If the answer is yes, we use such 10 stored positions to reconstruct the parabolic flight and predict the landing point. Fig. 6 shows three scenarios of such predictions, after the ball has been kicked in real games, implemented in the STOx's software environment.

Fig. 6(a), Fig. 6(d) and Fig. 6(g) show the initial setup just before the servicing robot kicks the ball. Then, in Fig. 6(b), Fig. 6(e) and Fig. 6(h) it is possible to see the ball trajectory a few frames after it has been kicked together with the predicted landing point (red cross). In these figures, at least 10 frames has passed and hence the reconstruction is already done. It is also possible to clearly see the difference in the curvature paths captured by the vision system depending on the direction and location of the chip kick with respect to the camera position. Finally, Fig. 6(c), Fig. 6(f) and Fig. 6(i) show the ball close to fall near the predicted landing point.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6. a) Field setup before kicking a chip kick for scenario 1. b)Projection of the ball after it has been kicked and prediction of the landing point (red cross) for scenario 1. c) The ball almost finishing its parabolic path and getting close to the predicted point for scenario 1. d) Field setup before kicking a chip kick for scenario 2. e)Projection of the ball after it has been kicked and prediction of the landing point (red cross) for scenario 2. f) The ball almost finishing its parabolic path and getting close to the predicted point for scenario 2. g) Field setup before kicking a chip kick for scenario 3. h)Projection of the ball after it has been kicked and prediction of the landing point (red cross) for scenario 3. i) The ball almost finishing its parabolic path and getting close to the predicted point for scenario 3.

4 Calculation of key points for the construction of dynamic plays

Our team, together with the league, has been evolving from simple predefined set plays to more complicated, dynamic and harder to predict plays when attacking. In this section, we will show fundamental advances in our team's AI that aims at providing our robots with a more intelligent gameplay by calculating a set of locations within the field where our robots might get high chances of scoring a goal. In this work we only show examples of this calculation for creating plays that follow a free kick. However, the concept can be seamlessly used in situations of free game.

4.1 Defining an objective

Given the current state of our robots and their capabilities, we have defined "good locations" for making a pass, when a free kick has been issued, as locations in the field where a clear pass can be achieved, free of opponents marks, close to the opponent's goal, with clear shot to goal and hard to predict for the opponents. In this scenario, we are considering that the robot receiving the pass will shoot at goal immediately with high chances of success.

In our general setup, we assume that a kickoff has been issued. Then, one robot prepares to make a pass and the remaining attackers will be positioned in different field locations to create diversion. The pass is made to only one of the attackers who will receive the ball and shoot at goal. Under this scenario, the most important design issues for us are:

- That the robot that receives the pass can shoot right away, without turning
- That the robot that receives the pass is close enough to the opponent's goal

We have modeled this scenario as the following optimization problem:

$$\arg\min_{(x,y)} w_1\theta_s(x,y) + w_2 d_g(x,y) \tag{7}$$

where $\theta_s(x,y)$ is the angle in the location (x,y) between the direction where the pass is coming and the direction of the opponent's goal, $d_g(x,y)$ is the distance from the (x,y) location to the goal and w_1, w_2 are weights that need to be tuned. The idea is to find a location (x,y) with smallest weighted sum between distance to goal and shot angle.

4.2 Constraints and simplifications

For the solution of the optimization problem (7) we have included a set of constraints and simplifications that allow us to find an improved solution that fulfills all the conditions for a "good location" and that can still be solved in short time. The following are the steps taken to further simplify the problem at hand:

- **Create a coarse grid of the field:** We have created a coarse grid of the game field in order to consider only a finite set of possible solutions to the problem. The granularity of the grid controls the complexity of the solution to the optimization problem and may vary according to the computational power capabilities. For larger granularities of the grid we would get larger solution times.
- **Discard locations with the presence of opponents within a given radius:** This constrain guarantees that the selected locations will be free of opponents and furthermore that it will take a while for nearby opponents to reach such locations.
- **Discard locations that are too close to the ball:** In general, it is a good idea to take the opponent unaware and especially unprepared. By including this constraint we are motivating solutions that include fairly long passes with the hope that the opponent will be as unprepared as possible
- **Discard locations with $\theta_s(x, y) > \pi/4$:** Although the objective function already encourages solutions with small values of $\theta_s(x, y)$, we have concluded experimentally that shots to goal after a direct pass (without the robot receiving the ball and turning) are possible up to a value of $\theta_s(x, y)$ smaller than $\pi/4$
- **Discard locations inside the opponent's penalty area:** Our attackers can not go inside the opponent's goal area and thus these locations must be excluded.

The procedure described above is performed when a new free indirect kick is issued. After that, the optimization procedure returns as result the location (x, y) with minimum value of the objective function. Nevertheless, in practice, we do not only retrieve this optimum value, but we also find a set of (x_i, y_i) locations that fulfill all the constraints and that are organized by their value of the objective function. From this set of "good locations" we finally choose 1, 2 or 3 to locate our attackers depending on the amount of available attackers for the play. The pass will be performed to one of these robots standing on these optimized locations.

4.3 Finding a location with high chances of scoring

The final step in the creation of these dynamic plays is the decision about which of the attackers will receive the pass. For this, we have created a model that allows us to discriminate among a set of "good locations", the one with highest chances of scoring. The idea is to quantify the scoring chance of a robot located at point (x, y), according to the position of the goalkeeper.

The idea for this quantification process is quite simple: find the area uncovered by the goalkeeper by projecting all possible shot directions from the attacking robot to the goal line.

This quantification process will exclusively depend on the position of the attacking robot and the position of the opponent's goalkeeper, an attribute that had not been taken into account in the optimization problem. Even though this final criteria does not take into account the distance between attacker and goal line, we can safely guarantee that the selected point will be close enough to the goal line since this is one of the criteria of the optimization problem (7).

4.4 Example and results

After careful tuning of the parameters related to the whole process we were able to implement our procedure for servicing free kicks in real games. Fig. 7 show an example of a sequence where the described strategy is implemented and how does it look in our STOx's software framework.

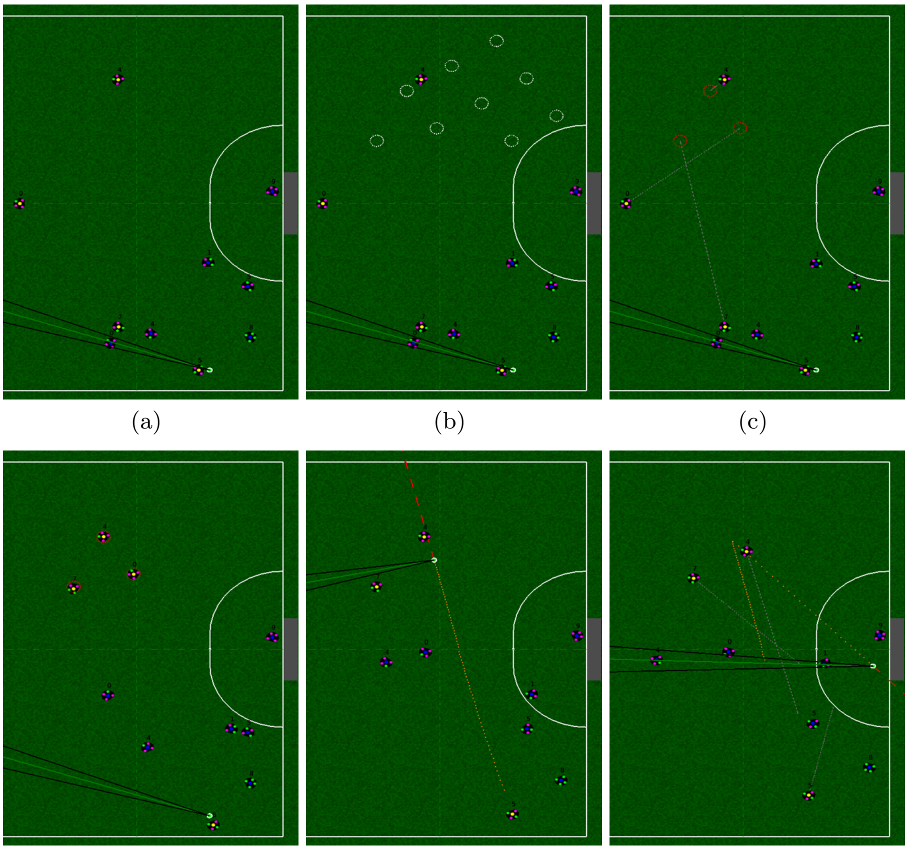
Fig. 7(a) shows the initial setup for servicing an indirect free kick where STOx's is the yellow team. Fig. 7(b) shows the locations with lower values of the objective function (7) after the optimization procedure is performed. These locations also fulfill all constraints and simplification described above. In Fig. 7(c), only a subset of these locations are selected and the three attacking robots are sent, one to each location (Fig. 7(d)). The evaluation step is performed and the pass is made to robot number 4 (Fig. 7(e)) who finally is capable of shooting to goal (Fig. 7(f)).

5 Conclusions

In this paper we have shown two important techniques in the STOx's team software development. On one hand, the reconstruction of the parabolic flight has shown to have an error of less than $6cm$ when predicting the landing point of the ball after a chip kick. We have also developed a learning machine with 97% of generalization accuracy capable of recognizing whether a given shot is a chip kick or not. We have shown 3 examples from real games where this technique successfully predicts the parabolic flights of the ball in different locations and orientations in the field. On the other hand, we have shown a procedure to find locations in the field with high chances of scoring after an indirect free kick has been issued. The technique is based on a constrained optimization problem with specific simplifications that has shown to be effective in a real game.

References

1. S. Rodríguez and E. Rojas. Diseño e implementación de un equipo Small Size League para la RoboCup, Universidad Santo Tomas, 2010.
2. S. Rodríguez, E. Rojas, K. Pérez, J. López, C. Quintero, O. Peña, J. Giraldo and J.M. Calderón. STOx's 2015 Extended Team Description Paper. Hefei, China 2015. Available for download in http://wiki.robocup.org/wiki/File:Small_Size_League_-_RoboCup_2015_-_ETDP_STOx%E2%80%99s.pdf
3. T. Panyapiang, K. Chaiso, K. Sukvichai and P. Lertariyasakchai. Skuba 2012 Extended Team Description. Mexico D.F, Mexico 2012. Available for download in http://robocupssl.cpe.ku.ac.th/_media/robocup2012:skuba_etdp_2012.pdf



(a) (b) (c)

(d) (e) (f)

Fig. 7. a) Initial setup for servicing a free kick for the STOx's team. b) Locations found by the described algorithm are shown in white circles on the field. c) Three selected locations shown in red circles to send the three available attackers. d) The three attackers has been placed in to the three selected locations. e) The pass is made to the attacker with higher scoring chance. f) The attacker receives the pass and shoots at goal, finally scoring.

Fig. 8. Picture of STOx's team members in RoboCup 2015

4. J.P. Mendoza, J. Biswas, P. Cooksey, R. Wang, S. Klee, D. Zhu, and M. Veloso. Selectively reactive coordination for a team of robot soccer champions. In Proceedings of the Association for the Advancement of Artificial Intelligence Conference (AAAI), to appear 2016.
5. J.P. Mendoza, J. Biswas, D. Zhu, R. Wang, P. Cooksey, S. Klee, and M. Veloso. CMDragons 2015: Coordinated offense and defense of the SSL champions. In Proceedings of the International RoboCup Symposium, to appear 2016.
6. S. Rodríguez, E. Rojas, K. Pérez, J. López, C. Quintero, J.M. Calderón. Fast Path Planning Algorithm for the RoboCup Small Size League. In Chapter RoboCup 2014: Robot World Cup XVIII. Vol 8992 of the series Lecture Notes in Computer Science pp 407–418. May of 2015.
7. C. Quintero, S. Rodríguez, K. Pérez, J. López, E. Rojas, J. Calderón. Learning Soccer Drills for the Small Size League of RoboCup. In Chapter RoboCup 2014: Robot World Cup XVIII. Vol 8992 of the series Lecture Notes in Computer Science pp 395–406. May of 2015.
8. S. Rodríguez, E. Rojas, K. Pérez, J. López, C. Quintero, and J.M. Calderón. STOx's 2014 Extended Team Description Paper. Joao Pessoa, Brazil 2014. Available for download in http://robocupssl.cpe.ku.ac.th/_media/robocup2014:etdp:stoxs_2014_etdp.pdf
9. R. Fosler. Generating High Voltage Using the PIC16C781/782. Application note, Microchip Technology Inc, (2005).
10. C. Cortes,V. Vapnik: Support vector networks. Machine Learning. 273–298, (1995).
11. Chang, C.C., Lin, C.J.: LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 27:1-27:27, (2011)
12. R. Rojas, M. Simon and O. Tenchio. Parabolic Flight Reconstruction from Multiple Images from a Single Camera in General Position. In Chapter RoboCup 2006: Robot World Cup X. Vol 4434 of the series Lecture Notes in Computer Science pp 183–193, 2007.