

CYRUS Soccer Simulation 2D Team Description Paper 2022

Nader Zare¹, Arad Firouzkouhi⁴, Omid Amini⁵, Mahtab Sarvmaili¹, Aref Sayareh⁶, Saba Ramezani Rad⁴, Stan Matwin¹², and Amilcar Soares³

¹ Institute for Big Data Analytics, Dalhousie University, Halifax, Canada

² Institute for Computer Science, Polish Academy of Sciences, Warsaw, Poland

³ Memorial University of Newfoundland, St. John's, Canada

⁴ Amirkabir University of Technology, Iran

⁵ Qom University of Technology, Iran

⁶ Shiraz University, Iran

{nader.zare, mahtab.sarvmaili}@dal.ca,
{arad.firouzkouhi, saba.ramezani}@aut.ac.ir
{arefsayareh, omidamini360}@gmail.com,
stan@cs.dal.ca, amilcarsj@mun.ca

Abstract. Soccer Simulation 2D League is one of the major leagues of RoboCup competitions. In a Soccer Simulation 2D (SS2D) game, two teams of 11 players and one coach compete against each other. The players are only allowed to communicate with the server that is called Soccer Simulation Server. This paper introduces the previous and current research of the CYRUS soccer simulation team, the champion of RoboCup 2021. We will present our idea about improving Unmarking Decisioning and Positioning by using Pass Prediction Deep Neural Network. Based on our experimental results, this idea proven to be effective on increasing the winning rate of Cyrus against opponents.

Keywords: Pass Prediction · Deep Neural Network · Soccer Simulation.

1 Introduction

RoboCup has been holding robotic soccer competitions annually since 1997. The idea of robotic soccer games had been proposed as a novel research topic in 1992. The main goal of this competition is implementation of robotic soccer games as a research area to improve the robotics, therefore it consists of several leagues such as rescue, soccer simulation, and standard platform soccer to advance every aspect of Robotics.

Soccer simulation 2D (SS2D) league is one of the oldest RoboCup leagues. CYRUS team has been participating in RoboCup competitions since 2013 and won the championship of SS2D league in RoboCup 2021[22]. This team has gained the first, second, third, fourth, and fifth places in RoboCup 2021, 2018, 2019, 2017, 2014, respectively. Also, CYRUS won first place in IranOpen 2021, 2018, and 2014, RoboCup Asia-Pacific 2018, and second place in JapanOpen

2020. CYRUS uses Agent2D as its base code [1]. In this paper, we briefly explain a novel idea about Unmarking and Positioning using Pass Prediction Neural Network.

1.1 Related Work

In recent years, SS2D teams improved their performance by implementing novel ideas, such as Player’s MatchUp. Helios has developed an algorithm called Player’s MatchUp for exchanging players’ positions during a game [5]. They took advantage of the distributed representation of actions and players to find similarities among teams’ behavior [4]. FRA-United applied reinforcement learning algorithm in the context of SS2D to improve marking behavior [6, 7]. Oxsy used the RSA accumulator (Real-World Performance of Cryptographic Accumulators) to model opponents strategy by using the observation coach. Oxsy members have implemented an idea called adaptive offside trap [8, 9]. Razi improved their offensive performance by updating chain action evaluator function and gravity strategy [10, 11]. Persepolis has implemented a Path Planning algorithm to improve its offensive strategy [12]. Hades improved their marking algorithm by evaluating the opponent player’s position [13].

1.2 Previous Work

CYRUS was established in 2013 and participated in many competitions since then. In this section, we briefly explain about the history of our research. We have started working on the passing behavior of team, unmarking strategy and team’s formation [14, 15]. Then after, we focused on the defensive behavior [16], shooting algorithms and communication [17] of our team. In 2018 we developed an opponent behavior prediction module [18]. Since 2018, we have mainly focused on the development of Pass prediction module and Data Generator framework which are elucidated in [19–21]. In recent years, we have released some parts of our research such as CYRUS 2014 Source Code, PYRUS - Python SS2D Base Code, SS2D Log Analyzer.

CYRUS 2014 Source Code One of the major contribution of Cyrus to SS2D community was publicly releasing the Cyrus 2014 source code. The code is available on our GitHub⁷.

PYRUS - Python Soccer Simulation 2D Base Code Due to the high performance of C++, majority of SS2D teams’ base code is written in C++ such as Helios Base [1], Gliders Base [2], WrightEagle Base [3]. Despite its processing time, the implementation of novel AI and machine learning algorithms are very much complicated, and requires lots of coding and debugging. This would be a tedious and time consuming task for the teams. Python on the other hand is simple and very well supported in terms of AI and machine learning libraries such as Scikit learn [25], PyTorch [24] and TensorFlow [23]. Therefore we have decided to develop a Python 2D base accessible for all of the SS2D community.

⁷ Cyrus 2014 Source <https://github.com/naderzare/cyrus2014>.

This base is available in Cyrus Github⁸, covering all features in the Agent2D when server runs on FULL-STATE mode.

2 Pass Prediction

As in real soccer, passing is one of the possible actions that can lead the ball to the goal. Predicting the pass target player, from the point of view of the ball owner, has many benefits in defensive and offensive algorithms. For example, opponent’s pass prediction can enhance marking decision algorithm, and teammate’s pass prediction can increase the efficiency of the unmarking strategy. In this paper, the ball owner is the player who can kick the ball in the current cycle or receive the ball in the future cycles. To predict the behavior of (our) ball owner, we were required to create the dataset of game states from this player point of view. For this purpose, we embedded a Data Extractor module in each one of the players and then we recorded the features of game states and their corresponding label. The label shows the uniform number of the player who is target of the best pass.

This module contains two sub-modules, Features Extractor and Label Generator (Figure 1). The Features Extractor sub-module perceives the state of the game, and generates features of the ball, opponent players, and our players by using the observation of the ball owner. The Label Generator module uses the output of the Chain-Action module to assign the corresponding action related to the features as a label by the Features Extractor sub-module. After generating the data set, the Player Sorter module sorts the features by using x of players’ positions for opponents and the uniform numbers of players for our players. The structure of the Data Extractor module is presented in Figure 1, and more details are available in our recent papers[20, 22, 21].

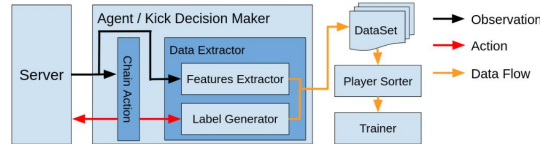


Fig. 1. Overview of the Data Extractor module.

3 Unmarking

Unmarking is an offensive action that players of the offender team use to escape from opponent marking and find a good spot for receiving a pass (when they are not the ball owner). The player who wants to unmark is called Unmarker. Unmarking includes two main parts, "Decisioning" and "Positioning". "Decisioning" finds the pass sender, so in this part, the player discovers one

⁸ Pyrus Base Code Source [git@github.com:Cyrus2D/Pyurus2D.git](https://github.com/Cyrus2D/Pyurus2D.git).

teammate that will send a pass to itself in the future. The positioning parts look for the best spot to receive the future pass.

3.1 Unmark Decisioning by using Pass Prediction module

When one of our players posses the ball, the rest of players should assist it for obtaining the best chain action to move the ball forward. For this purpose, each player of CYRUS tries to simulate a tree that contains the passes and game states generated in the future. This tree includes the probable passes and their outcome states in the future. Each node of the tree contains a state of the game so that one of our players is the ball owner in the state. Each edge of the tree shows a probable pass in the future.

Pass Predictor To create the tree, we developed a module called Pass Predictor to predict the target player of the pass in a given state. This module receives a state and a list of players called "Ignored Players". The Ignored Players presents players who can not be the pass receiver in the given state. To find the best passes, the Pass Predictor module generates features from the given state and feeds them to the trained pass prediction DNN and then produces the probability of receiving the pass for all players. There after, it finds the players who have the maximum probability among players that are not in the "Ignored Players" list. Finally, it creates an object called Pass-State-Value containing pass, its outcome state, and state value. The state value stands to show the probability of the pass. Eventually, this module returns two passes who can be the pass receiver in the state, their result state, and state point.

Ignored Player list The Ignored Player list contains players who are ball owners in nodes of the tree. Therefore, a player can be the ball owner in one node of the tree.

Root node The root node of the tree is created by using the current state of the game if one of our players can kick the ball right now. However, if one of our player receives the ball in the next few steps, the unmarker predicts the cycle of intercepting the ball and the position of ball receiver, then it updates the root node of pass predictor tree.

After creating the root node, the unmarker player feeds the state of the root node and the Ignored Players list to the Pass Predictor module and then it receives the list of two Pass-State-Value objects. At this step, the Ignored Players will be updated to include the ball owner. After receiving the Pass-State-Value objects, the unmarker inserts them into a list called Candidate List. To grow the tree, we use the Best First Search algorithm in which, at the next step, the unmarker selects the best object from the Candidate List with highest value, and then it added the selected object to the tree. Also, the ball owner of the state is added to the Ignored Players list. This procedure continues until the number of tree nodes is equal to ten. After termination of this procedure, the unmarker agent looks for its corresponding node in the tree, then it chooses the parent player as the "Passer" for the unmarking procedure in order to receive the pass from it in the future.

Figure 2 shows an overview of the Unmarking Decisioning.

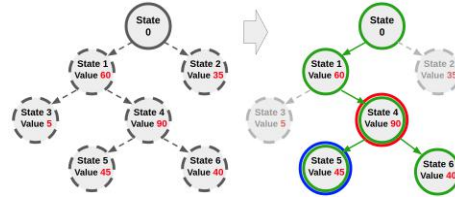


Fig. 2. Overview of the Unmark Decisioning algorithm. The left tree shows the result states and their points. The green circles in the right tree presents the nodes have been selected as best node from the Candidate list. The blue circle shows the node that the unmarker is ball owner in its state, and its parent node is shown by red circle.

3.2 Positioning in unmark

After finding the "Passer" player, the unmarker should find a position to receive the pass. For this purpose, the unmarker generates a set of points around its current position in three levels that respectively have 2, 4, 8 meters as radius. Each level contains 8 points. Selecting the best position contains two steps. First, the unmarker simulates a pass from the Passer Player to each position, if the unmarker can receive the ball in the position before the opponent players the position is a validated point. Then, among the valid points, the unmarker selects the closest position to the opponent's and it tries to move to the best position in the future cycles.

Figure 3 shows an example of the Unmark Decisioning and Unmark Positioning parts in SS2D field.

4 Results

To obtain the data set, we ran 10,000 games between CYRUS and YuShan2021, and we obtained 3,000,000 data instances. We split data into two subsets, 80% for training and 20% for testing. The prediction model (DNN) has three layers of 350, 250, and 11 neurons, with RELU activation function and a softmax function at the last layer. The validation accuracy of the trained neural network on the test data was 73%.

Three unmarking algorithms are available in CYRUS, Pass Prediction Unmarking, Hard-Coded Unmarking, Voronoi Unmarking. The Pass Prediction Unmarking uses Deep Neural Network for decisioning, and searching around the player for positioning. Hard-Coded algorithm, the ball owner is passer if the ball owner is closer than 20 meters to the unmarker, otherwise the selected passer is a player that is close to the line between ball owner and the agent. In the Voronoi Unmarking, players uses Voronoi diagram to find the best position to escape from opponent's marking. In CYRUS2021, we used Hard-Coded and Voronoi Unmarking algorithms, and if the first algorithm does not find a position, then the second algorithm tries to find a new position.

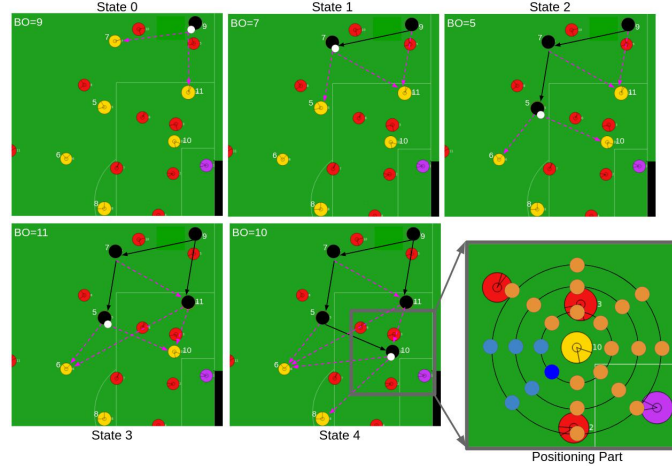


Fig. 3. Overview of the Unmarking algorithm. Player 10 would like to unmark, and Player 9 is the current Ball Owner in the current cycle of the game. States show the future state of the game. Pink dashed arrows present candidate passes, and black arrows show the selected best passes. Black circles present players in Ignored Player list. The best targets for Player 9 are Player 7 and Player 11 in state 0. Player 7 is selected to be the ball owner, and the best targets are Player 5 and Player 11 in the next step. The pass from Player 7 to Player 5 is selected in state 2. In the next step, Player 11 is selected as the receiver, and in the state for Player 10 is the ball receiver from Player 5. The black arrows in State 4 show that the unmarker player should find a position to receive a pass from Player 5. The Positioning Part presents all position candidates for unmarker. Blue circles are validated positions and the dark blue circle is the best position for unmarking. Orange circles shows positions that the Player 10 can not receive a pass there.

To test the benefits of Pass Prediction Unmarking algorithm, we created 6 versions of CYRUS. In the first version, we just used the Pass Prediction Unmarking. Pass Prediction and Voronoi Unmarking algorithms were used in the second version. Four other versions are presented in Table 1. To test the benefits of the new algorithm, we ran 1000 games against YuShan2021 for each versions. All of the results are presented in Table 1.

The obtained results proves the significance of our proposed method in the improvement of statistics against YuShan2021. For example, the new algorithm increased our expected win-rate for 5%, also it improves our ball possession percentage and number of shoots.

5 Conclusion

In this paper we proposed an novel idea for the unmarking strategy of CYRUS by Pass Predicting. To model our teammates' passing behavior in the game, we exploited the generated features and trained a Deep Neural Network called Pass Predictor. To select a player as Passer for another agent, we used the Pass

Versions	Pass Prediction	Hard-Coded	Voronoi	Goals Scored	Goals Conceded	Goal Difference	Win-rate	Expected-Win-rate	Pass Numbers	Pass Accuracy	Possession	Shoot Number
V1	Y			1.13	0.70	0.43	45.75	66.42	313.54	0.90	62.18	3.28
V2	Y		Y	1.10	0.67	0.43	45.80	67.55	315.90	0.90	62.26	3.26
V3 (CYRUS2021)		Y	Y	1.04	0.78	0.26	39.80	61.42	307.35	0.89	61.36	3.15
V4			Y	0.98	0.70	0.28	41.70	60.79	308.30	0.89	61.44	3.11
V5			Y	1.08	0.76	0.32	43.40	63.27	287.69	0.88	59.86	3.07
V6				0.92	0.70	0.22	39.70	59.79	283.46	0.88	59.69	2.85

Table 1. The impact of different unmarking approaches on Cyrus performance

Predictor module to create a tree when the current state of the game is its root node. To search and grow the tree, we used the Best First Search algorithm. Finally, we explain an idea to find the best position for unmarking.

6 Future Work

We intend to explore other challenging aspects of SS2D and unmarking behavior in Soccer Simulation 2D and Unmarking. First, we would like to use the Pass Predictor module to find the best position for catching a pass by relocating the position of unmarker and feeding it to the DNN. Also, we are willing to create a model with more generality on the wider range of teams.

References

1. Akiyama, H., Nakashima, T.: Helios base: An open source package for the robocup soccer 2d simulation. In Robot Soccer World Cup 2013 Jun 24 (pp. 528-535). Springer, Berlin, Heidelberg.
2. Prokopenko, M., Wang, P.: Gliders2d: Source Code Base for RoboCup 2D Soccer Simulation League. CoRR abs/1812.10202 (2018)
3. Bai, A., Chen, X., MacAlpine, P., Urieli, D., Barrett, S. and Stone, P., 2011, July. Wrigteagle and UT Austin villa: RoboCup 2011 simulation league champions. In Robot Soccer World Cup (pp. 1-12). Springer, Berlin, Heidelberg.
4. Akiyama, H., Nakashima, T., Fukushima, T., Suzuki Y., Ohori, A.: HELIOS2019: Team Description Paper. In: RoboCup 2019 Symposium and Competitions. Australia, Sydney (2019).
5. Yamaguchi, M., Kuga, R., Omori, H., Fukushima, T., Nakashima, T., Akiyama, H.: Helios2021: Team description paper. In: RoboCup 2021 Symposium and Competitions, Worldwide (2021)
6. Gabel, T., Sommer, F., Breuer, S., Godehardt, E.: FRA-UNited-team description 2019. In: RoboCup 2019 Symposium and Competitions: Team Description Papers. Sydney, Australia (2019).

7. Gabel, T., Kloppner, P., Eren, Y., Sommer, F., Breuer, S., Godehardt, E.: FRA-UNited — Team Description 2021. In: RoboCup 2021 Symposium and Competitions, Worldwide (2021).
8. Marian, S., Luca, D., Sarac, B., Cotarlea, O.: OXSY 2018 Team Description. In: RoboCup 2018 Symposium and Competitions: Team Description Papers. Montreal, Canada (2018).
9. Marian, S., Luca, D., Sarac, B., Cotarlea, O.: OXSY 2018 Team Description. In: RoboCup 2021 Symposium and Competitions: Team Description Papers. Worldwide (2021).
10. Noohpisheh, M., Shekarriz, M., Barzegar, S., Borzoo, D., kariminia, A.: Razi Soccer 2D Simulation Team Description Paper 2018. In: RoboCup 2018 Symposium and Competitions: Team Description Papers. Montreal, Canada (2018).
11. Noohpisheh, M., Shekarriz, M., Bordbar, A., Liaghat, M., Salimi, A., Borzoo, D., Zarei, A.: Razi Soccer 2D Simulation Team Description Paper 2019. In: RoboCup 2019 Symposium and Competitions: Team Description Papers. Sydney, Australia (2019).
12. Noohpisheh, M., Shekarriz, M., Zaremehrijardi, F., Khademi Ardekani, F., Khor-sand, S. A.: Persepolis Soccer 2D Simulation Team Description Paper 2021. In: RoboCup 2021 Symposium and Competitions: Team Description Papers. World-wide (2021).
13. Akhondi, F., Esmaelifar, S., Esmaelifar, S., Rokni, S. R., Rajabi, A., Hasanpour, G.: Hades2D soccer2D simulation Team Description paper. In: RoboCup 2021 Sym-posium and Competitions: Team Description Papers. Worldwide (2021).
14. Zare, N., Zolanvar, H., Mahoor, P., Fatehi, M., Karimi, M., Tekrar, F., Asali, E., Khayami, R.: Cyrus Soccer 2D Simulation Team Description Paper 2013. In: RoboCup 2013. Netherlands, Eindhoven, 2013.
15. Khayami, R., Zare, N., Karimi, M., Mahor, P., Afshar, A., Najafi, M. S., Asadi, M., Tekrar, F., Asali, E., Keshavarzi, A.: CYRUS 2D simulation team description paper 2014. In: RoboCup 2014. Joao Pessoa, Brazil, 2014.
16. Zare, N., Karimi, M., Keshavarzi, A., Asali, E., Ali Poor, H., Aminian, A., Be-heshtian, E., Mola, H., Jafari, H., Khademian, M. J.: Cyrus Soccer 2D Simulation Team Description Paper 2015. In: RoboCup 2015. Hefei, China, 2015.
17. Zare, N., Keshavarzi, A., Beheshtian, S. E., Mowla, H., Akbarpour, A., Jafari, H., Arab Baraghi, K., Zarifi, M. A., Javidan, R.: Cyrus 2D Simulation Team Description Paper 2016. In: RoboCup 2016. Leipzig, Germany, 2016.
18. Zare, N., Sadeghipour, M., Keshavarzi, A., Sarvmali, M., Nikanjam, A., Agha-yari, R., Firouzkouhi, A., Abolnejad, M., Elahimanesh, S., Akhgari, A.: Cyrus 2D Simulation Team Description Paper 2018. In: RoboCup(2018), Montreal, Canada (2018).
19. Zare, N., Sarvmali, M., Mehrabian, O., Nikanjam, A., Khasteh, S.-H., Sayareh, A., Amini, O., Barahimi, B., Majidi, A., Mostajeran, A.: Cyrus 2D Simulation 2019. In: RoboCup(2019).
20. Zare, N., Sayareh, A., Sarvmali, M., Amini, O., Soares, A., Matwin, S.: CYRUS 2D Soccer Simulation Team Description Paper 2021. In: RoboCup 2021 Symposium and Competitions, Worldwide (2021)
21. Zare, N., Sayareh, A., Sarvmali, M., Amini, O., Matwin, S., Soares, A.: Engineering Features to Improve Pass Prediction in 2D Soccer Simulation Games. In: RoboCup 2021: Robot World Cup XXIV, Springer (2021)
22. Zare, N., Amini, O., Sayareh, A., Sarvmali, M., Firouzkouhi, A., Soares, A., Matwin, S.: Improving Dribbling, Passing, and Marking Actions in Soccer Simu-

- lation 2D Games Using Machine Learning. In: RoboCup 2021: Robot World Cup XXIV, Springer (2021)
23. Martín Abadi, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
 24. Paszke, A. et al., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32.
 25. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, pp.2825-2830.