Ball Path Prediction for Humanoid Robots: Combination of k-NN Regression and Autoregression Methods

Yasaman Mirmohammad*
Bio-Inspired System Design Lab
Amirkabir University of Technology
Tehran, Iran
ys.m@aut.ac.ir

Shayan Khorsandi*
School of Computer Engineering
Iran University of Science and Technology
Tehran, Iran
sh_khorsandi@comp.iust.ac.ir

Mohammad Navid Shahsavari*
Bio-Inspired System Design Lab
Amirkabir University of Technology
Tehran, Iran
navid.shahsavari@aut.ac.ir

Behnam Yazdankhoo
School of Mechanical Engineering
University of Tehran
Tehran, Iran
byazdankhoo@ut.ac.ir

Soroush Sadeghnejad
Bio-Inspired System Design Lab
Amirkabir University of Technology
Tehran, Iran
s.sadeghnejad@aut.ac.ir

***Abstract*—In this paper, we propose a method for predicting the path of the ball on the soccer field for the humanoid robots. A cost-function-based k-nearest neighbor regression method is first proposed to account for the part of the prediction which is based on previously observed data. Next, the autoregression method is utilized in order to carry out the prediction based on the current ball path. Finally, these two methods are combined to form the final prediction model. Moreover, two different schemes are introduced based upon the proposed model: fixed and adaptive schemes. In fixed scheme, the prediction is made once during the initial steps of the motion and is used throughout the whole ball movement. However, in adaptive scheme, autoregression method coefficients are updated in fixed predefined steps during the motion. This is beneficial to robustify the prediction against an externally applied disturbance on the ball path. Our proposed method is tested by simulation and practical implementation and the results demonstrate a high precision rate.**

***Index Terms*—Ball path prediction, Humanoid robots, Autoregression, K-nearest neighbor regression.**

I. Introduction

Object path prediction is a well-known and important task from both research and industrial point of view. Various scientific papers have considered path prediction in recent years which reflects the significance of this task in a broad sort of applications. The important point is, however, that various mathematical approaches have been adopted in each of these related works.

For instance, flight path prediction is investigated in [1] which is based on applying a stochastic model named Hidden Markov Model (HMM). Ship route prediction is another application proposed in [2] in which a k-nearest neighbor classifier is applied as a model for predicting routes in the waterways. Ma *et al.* used an LSTM-based method for agent predictions in an analysis regarding traffic, so that the autonomous vehicle can make appropriate navigation decisions to control traffic [3].

More specifically, path prediction of a thrown body has become important in human-behavioral-based tasks in robotics, which is also reflected in different robocup competitions such as soccer and basketball [4]. Therefore, crucial tasks which can be influenced by path prediction are shooting and passing in soccer-playing humanoids [5]–[8], or ping-pong-playing humanoids such as [9]–[11] in which a Kalman filter based estimation method is used to predict the status of the ball. Throwing-and-catching is another robotic task which has been considered in recent years, for instance [12] utilizes a feedforward neural network model for estimating the position based on the intentions of the human partner during the preparatory motion. Also tracking is one of the crucial tasks for many robot-based applications which could benefit from prediction, as [13] addressed the predictive tracking issue.

To mention broader examples, [14] forms a multi-Robot system in which we will face a task assignment problem for a large number of tasks and soccer-playing robots. Under these circumstances, task allocation and path planning for reaching an optimum point become necessary. Ball Path prediction can also help motion planning algorithms such as [15] in order for humanoids to generate optimal motions during a soccer match, for instance.

Some tasks may inherently show a time series behavior, where the next position of the object is mainly a function of the previous positions. Several numerical approaches can be applied in this regard, such as neural networks [16]–[18]. Implementation of the time series approach can be viewed in [19] which proposed a deep learning approach to estimation and prediction by a bidirectional LSTM, and [20]

*Authors contributed equally to this work which presented a time-delay neural network (TDNN) based approach for the path prediction of a thrown body.

Other methods such as k-nearest neighbor have also been applied to path prediction problem, for instance [21] and [22] represented a nearest neighbor regression method in which the predicted trajectory is a linear combination of selected trajectories.

In our previous paper [23] we analyzed and compared different methods of ball path prediction for humanoid robots. In this work, with specific concentration on soccer player humanoid robots, we aim at predicting the ball path on a soccer field. One of the major drawbacks of the previous works is that they rely on merely either current position data or the previously gathered dataset. To address this problem, in this paper we combine k-nearest neighbor regression (hereinafter, offline) and autoregression (hereinafter, online) methods in order to account for both current path and previously learned paths.

In fact, a weighted average of online and offline methods is used in this work to form the final prediction model of the ball path. This model is then divided into two schemes: fixed and adaptive. In fixed prediction scheme, after some initial steps which are reserved for the required calculations, the ball path is predicted only once and that prediction is considered valid throughout the whole ball motion. In adaptive scheme, however, the online method is updated in predefined intervals during the prediction which, in turn, updates the remaining points of the prediction thereafter. The adaptive scheme will be particularly fruitful for prediction when the ball path is suddenly changed during the motion, an issue which has not been addressed in the related works.

Therefore, our contributions include: Firstly, combining k-nearest neighbor regression and autoregression methods in order to achieve more accuracy in prediction. And secondly, proposing an adaptive prediction scheme in order to robustify the prediction against external disturbances.

This paper is organized as follows. In part II, different parts of the proposed path prediction method, along with the final model, are theoretically explained. The results obtained from simulation and experimental tests are then described in part III. And finally, part IV is dedicated to conclusions and suggestions for future contribution.

II. PREDICTION METHOD

A. Overall Approach

Prediction of the ball path is investigated from two different perspectives in this work. On one hand, the robot should be able to carry out the prediction task by means of learning from the previously observed paths. On the other hand, the current motion itself could be conceived of as a time series prediction problem without regard to path dataset.

These two aspects of the prediction will be discussed in the following sections and will be finally integrated to result in the ultimate prediction approach.

B. Offline Portion

Offline portion is responsible for taking into account the learning part of the prediction. To this end, here we adopt the k-nearest neighbor (k-NN) regression method utilized in [21] and then in [22], in both of which k is set to 2.

Input of the predictor is reference measurement of the current path $C(1:m) = \{P_c(1), P_c(2), ..., P_c(m)\}$ where m is number of frames until now and P denotes the position in the cartesian space. The database includes N paths $S_1, S_2, ..., S_N$ where each path $S_i(1:n) = \{P_i(1), P_i(2), ..., P_i(n)\}$ and $n > m$ is the total number of frames. The model predicts the path $C(f:n) = \{P_c(f), P_c(f+1), ..., P_c(n)\}$ where $f = m+1$. In order to calculate this path, two guiding paths $A(1:n) = \{P_a(1), P_a(2), ..., P_a(n)\}$ and $B(1:n) = \{P_b(1), P_b(2), ..., P_b(n)\}$ are taken from the database which the measured points of $C(1:m)$ lie higher than corresponding points of path $B(1:m)$ and lower than corresponding points of path $A(1:m)$.

In this work, two factors are used in order to select $A(1:m)$ and $B(1:m)$: distance and motion slope. The distance is defined as a summation of euclidean distances between the corresponding points of the paths.

$$D(A,C) := \sum_{i=1}^{m}(A(i) - C(i)) \tag{1}$$

However, unlike [21], in which initial velocities were chosen as another factor for selecting path A and B, here we introduce the motion slope factor which is defined as the difference between the slope of path $A(s:m)$ and path $C(s:m)$ where $1 \leq s < m$. The slope of path $A(s:m)$ is defined by

$$M_A(s:m) := \frac{y_A(m) - y_A(s)}{x_A(m) - x_A(s)} \tag{2}$$

and $M_C(s:m)$ is calculated in a similar manner. The motion slope between path $A(s:m)$ and $C(s:m)$ is then computed by

$$M(A,C) := M_A(s:m) - M_C(s:m) \tag{3}$$

In order to select the proper $A(1:m)$, a cost-function-based approach similar to what has been proposed in the literature for k-NN or kernel regression is adopted here [24]. For this aim, we design a cost function which best fulfills our requirements by linearly combining the two previously introduced factors (namely: distance and the motion slope) as follows

$$J_A := \rho\frac{|D(A,C)|}{|D_{max}(A,C)|} + \frac{|M(A,C)|}{|M_{max}(A,C)|} \tag{4}$$

where $\rho > 0$ determines the relative impact of the two factors in the cost function, and subscript max denotes the maximum value. J_A is then computed for all the candidate paths for $A(1:n)$ and the one with the minimum cost function is selected as the final $A(1:n)$. Equations (1) to (4) are used in an exactly similar manner for selecting the path $B(1:n)$.

Finally, the predicted path $C_{off}(f:n)$ is calculated as a linear combination of $B(f:n)$ and $A(f:n)$

$$C_{off}(f:n) = wA(f:n) + (1-w)B(f:n) \tag{5}$$

where $0 \leq w \leq 1$. In [22], w is considered as 0.5, but here is computed by following expression.

$$w = \frac{D(B,C)}{D(A,C) + D(B,C)} \tag{6}$$

C. Online Portion

Online portion is regarded as the time series part of the prediction. In other words, the current path is utilized for predicting the future positions of itself. To this end, the autoregression method is used. Due to space limitation, only the main formulas are mentioned hereunder. More details are presented in the main paper [23]. According to this method the future position is computed as follows:

$$x(f) = (2+\psi)x(f-1) + (-1-2\psi)x(f-2) + \psi x(f-3) \tag{7}$$

$$\psi = \frac{r_1}{r_0} \tag{8}$$

$$r_k := \frac{c_k}{c_0} \tag{9}$$

$$c_k := E[x(i)x(i-k)] \tag{10}$$

where $k = 0, 1$ and $E[.]$ denotes mathematical expectation and i is defined in (1).

In order to predict the ball position in the next q steps, the following expression is used:

$$x(f+q) = (2+\psi)x(f+q-1) + (-1-2\psi)x(f+q-2) + \psi x(f+q-3) \tag{11}$$

where $q = 1, ..., n-f$. In an exactly same manner, (7) to (11) are used for predicting position in y direction. Now $C_{on}(f:n)$ is defined by the following expression

$$C_{on}(f:n) := \Big(x(f:n), y(f:n)\Big)^T \tag{12}$$

D. Combination of Offline and Online Portions

In this paper, we design a combination of offline and online methods in order to use both the prior knowledge and the current position data. Due to low efficiency of the offline method in the lack of sufficient dataset size, and also reducing the calculations performance in the presence of a huge dataset because of the high time constraints, using offline method alone seems inefficient. On the other hand, online method suffers from problems such as being unable to predict the stop position of the ball and lack of learning from the previous ball paths.

As a consequence, each method can cover the disadvantages of the other by being combined together. Our proposed method is introduced by the following expression

$$C(f:n) = \alpha C_{off}(f:n) + (1-\alpha)C_{on}(f:n) \tag{13}$$

where $0 \leq \alpha \leq 1$. Equation (13) is called the *fixed prediction scheme*, in which both online and offline portions of the prediction are determined merely once at the beginning of the motion.

However, if an unexpected change in ball movement due to an external disturbance occurs in the middle of the motion, the above-mentioned scheme fails to accurately predict the ball path. To tackle this issue, we propose a novel approach which is called *adaptive prediction scheme* in which the online prediction method is updated in predefined steps. In this method $n-f$ steps ahead is divided into j equal parts and online method equations (7) to (11) are repeated for each part. In other words, online portion coefficients are recalculated for the first $f + j'n'$ steps, where $n' = \frac{n-f}{j}$ is the length of each part and $j' = 1, ..., j$, and then prediction is made for the remaining points. Therefore we have:

$$C_{on}(f:j'n') = \Big(x(f:f+n'), y(f:f+n'), ..., x(f+(j'-1)n':f+j'n'), y(f+(j'-1)n':f+j'n')\Big)^T \tag{14}$$

Now the new $C_{on}(f:n)$ is put into (13) and the path prediction is conducted while the online portion coefficients are updated in fixed steps during the actual path. It should be noted that here j or n' are chosen such that for both of them we have $j \in \mathbb{N}$ and $n' \in \mathbb{N}$.

III. Results

A. Description

Two scenarios are designed in order to carry out the simulation and experiment:

- Scenario 1: The ball moves normally in an x-y direction.
- Scenario 2: The ball initiates a normal movement, but an unknown disturbance is applied to the ball during its path in an unknown moment.

Note that the initial speed of the ball in both the simulations and the experiments is random and less than $2m/s$.

B. Simulation Results

Gazebo is used as our simulation environment. It is selected because of its integrity with ROS, the framework that is utilized in the robots. Gazebo's world is consisted of two models; Robocup 3D soccer simulation field and Robocup 3D soccer simulation ball with their corresponding friction force in order to achieve the real world conditions and ball movements. Our dataset is collected by simulating the ball movement multiple times with different linear and angular velocities for being used in offline prediction. Ball position in 2D axis is captured each 40ms, 100 times to occupy the whole ball movement in 4 seconds. The 2D position of the ball is then sent to ROS for capturing data. We collected 139 sample ball paths for the dataset. Gazebo simulation environment is depicted in Fig. 1.

For scenario 1, $\alpha = 0.9$, $m = 10$, $\rho = 5$ and $s = 10$ and the fixed prediction scheme is utilized. The result for this scenario is shown in Fig. 2.

For scenario 2, $\alpha = 0.1$ and other parameters are as same as scenario 1. Both fixed and adaptive schemes are adopted for this scenario. In the adaptive scheme, the online method is updated every 15 steps ($n' = 15$). The results for this scenario are illustrated in Fig. 3 and Fig. 4. Note that the actual path is the same in these two figures.

C. Experimental Results

Experiments are carried out utilizing soccer artificial grass field and standard soccer ball as shown in Fig. 5. A fixed camera is placed above the soccer field, which is shown in Fig. 6. The Shared Vision System for the RoboCup Small Size League [25] is used in order to implement detection and localization of the ball. 166 sample ball paths were collected for the dataset. The ball is kicked by a human from a fixed starting point in different directions, and each path consists of samples from position of the ball relative to the starting point.

For scenario 1, $\alpha = 0.9$, $\rho = 1$ and the other parameters are the same as simulation. Also the fixed prediction scheme is adopted. The result for this scenario is shown in Fig. 7.

For scenario 2, $\alpha = 0.1$ and other parameters are the same as scenario 1. Both fixed and adaptive schemes are used in this scenario. In the adaptive scheme, the online method is updated every 15 steps ($n' = 15$). Fig. 8 and Fig. 9 show the results for this scenario. Note that the actual path is the same in these two figures.

D. Analysis of Results

In scenario 1, the fixed scheme with only offline portion (i.e. $\alpha = 1$) can be efficient, but this approach strongly depends on the size and variation of the dataset and because of this dependency, it is possible that it does not lead to the best prediction. Hence, by combining it with the online method, the position data of the current path can also be involved in the prediction. However, the online method weakly predicts the position at which the ball stops. For this reason, it seems better to use a low coefficient for the online method.

As can be seen in Fig. 2 and Fig. 7, both path and stop position are predicted well (The final position at which the ball stops is the end of the actual path). The root mean square errors (RMSE) are $1.00m$ and $0.36m$, respectively, and the normalized root mean square errors (NRMSE) are 3.96% and 10.27%, respectively. It should be noted that NRMSE is calculated by dividing RMSE by the length of the actual path.

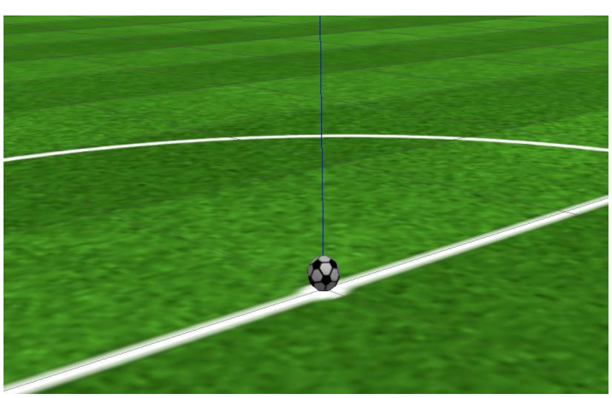

Fig. 1. Gazebo simulation environment.

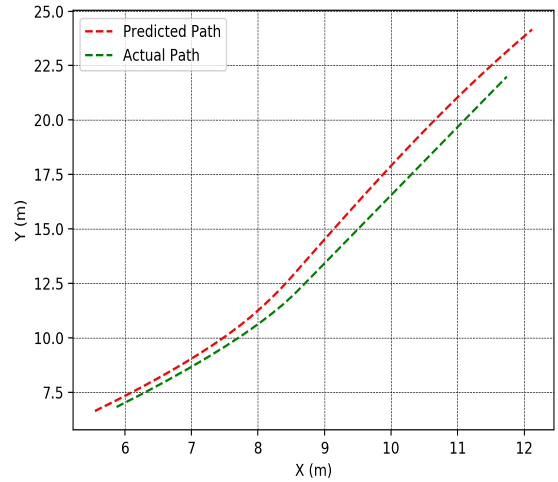

Fig. 2. Simulation result for scenario 1 using fixed prediction scheme.

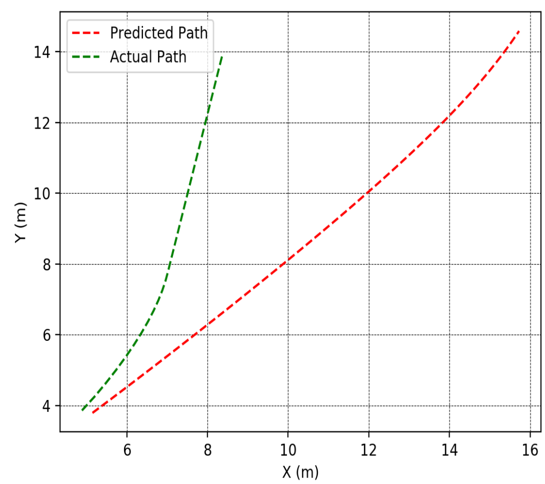

Fig. 3. Simulation result for scenario 2 using fixed prediction scheme. Fixed prediction is not able to predict the ball path properly due to the external disturbance.

However, for scenario 2, the fixed scheme can not predict the disturbance effect on the ball path as can be seen in Fig. 3 and Fig. 8. The RMSE for these two figures are $5.38m$ and $0.45m$ and NRMSEs are 31.59% and 10.86%, which seem relatively high. As a result, the adaptive scheme is used in this scenario with the same actual path in order to account for the sudden change in the ball direction and decrease the prediction error. Fig. 4 and Fig. 9 show that this approach can effectively update the primary prediction. The RMSE for these two figures are $0.61m$ and $0.08m$ and NRMSEs are 3.59% and 2.07%. Since the actual paths are the same, these errors show significant improvement in the prediction process, both in simulation and experiment.

Note that the discontinuities and overlaps in the prediction

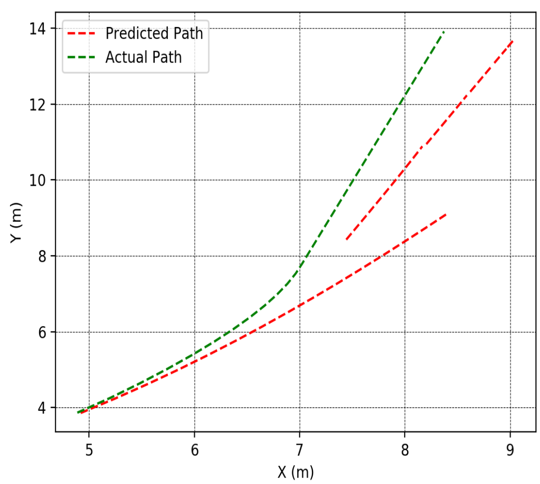

Fig. 4. Simulation result for scenario 2 using adaptive prediction scheme. The actual path is the same as Fig. 3. Adaptive prediction is able to predict the ball path despite the external disturbance.

Fig. 5. Ball and soccer grass field used for experimental setup.

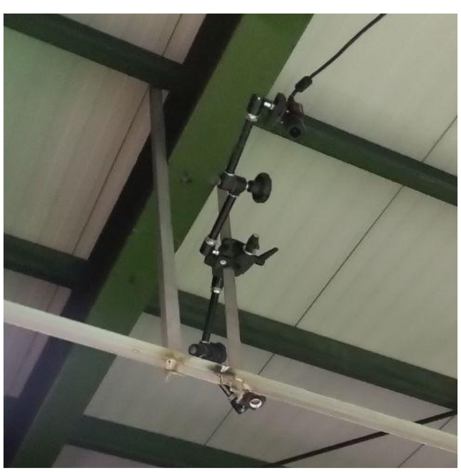

Fig. 6. Camera configuration used for experimental setup.

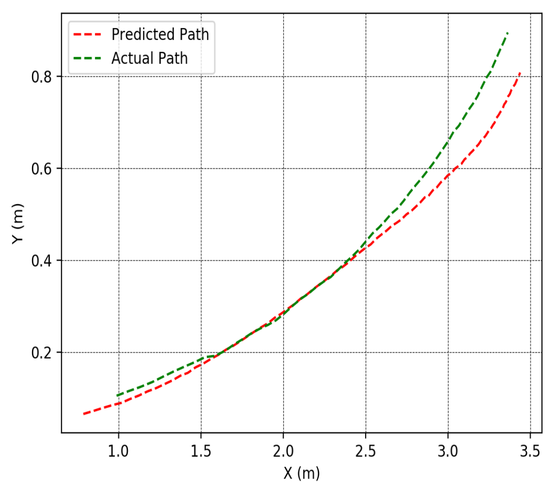

Fig. 7. Experimental result for scenario 1 using fixed prediction scheme.

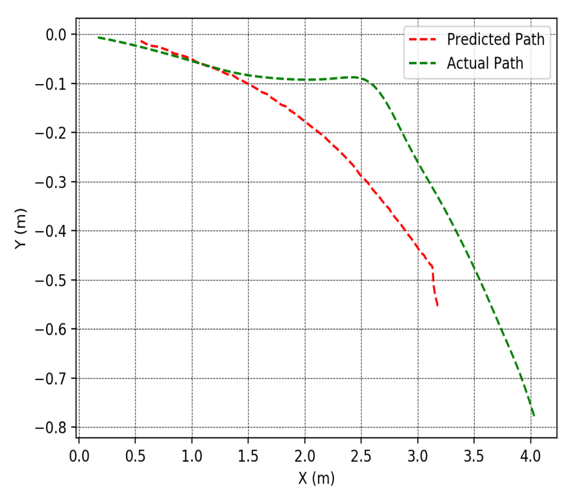

Fig. 8. Experimental result for scenario 2 using fixed prediction scheme. Same as Fig. 3, Fixed prediction is not able to predict the ball path properly due to the external disturbance.

curves in Fig. 4 and Fig. 9 are due to the updates made at the corresponding steps. In other words, they do not indicate lack of or duplicate predictions at the specified positions, they are simply due to the step-wise corrections during the update procedure in order to modify the predictions made at the previous steps.

Finally, it should be pointed out that it is important to choose the coefficient of the online method (i.e. $1-\alpha$) considerably larger than the coefficient of the offline method in the adaptive scheme, because the online method is responsible for updating the prediction.

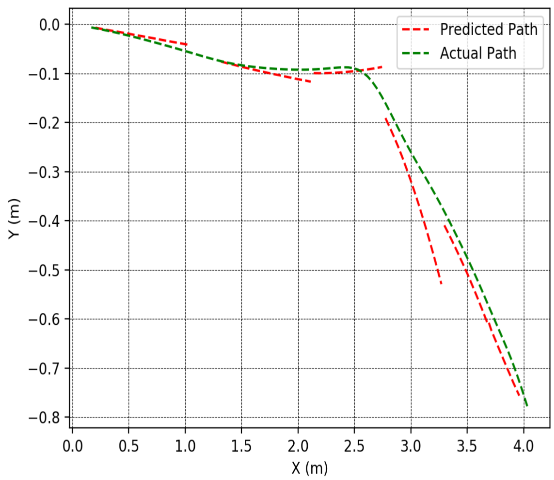

Fig. 9. Experimental result for scenario 2 using adaptive prediction scheme. The actual path is the same as Fig. 8. Same as Fig. 4, Adaptive prediction is able to predict the ball path despite the external disturbance.

IV. Conclusion and future works

In this paper, we aimed at predicting the path of the ball for humanoid robots. First, we used k-NN regression in combination with autoregression method in order to benefit from both previously gathered knowledge and current ball path. Furthermore, in order to be able to predict an unexpected change in the ball movement due to an external disturbance, we proposed an adaptive prediction scheme in which the online method coefficients are updated in fixed intervals.

Simulation and experimental results showed that the fixed prediction scheme can predict the path in a proper way, but in the presence of an external disturbance this method can not carry out the prediction task accurately. However, it could be deduced from the results that the adaptive scheme can robustify the prediction against an external disturbance which alters the preliminary path and, thus, can make the prediction more accurate.

For the future contributions, a suggestion is to consider more neighbours in offline method for the k-NN algorithm. Another suggestion to mention is to calculate the parameters adaptively instead of setting them as constant values. Moreover, a study on the effect of the initial speed of the ball could be considered as a future contribution.

References

[1] S. Ayhan and H. Samet, ”Aircraft trajectory prediction made easy with predictive analytics”, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 21-30, 2016.

[2] A. L. Duca, C. Bacciu, and A. Marchetti, ”A K-nearest neighbor classifier for ship route prediction”, OCEANS 2017-Aberdeen, pp. 1-6, 2017.

[3] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, ”Trafficpredict: Trajectory prediction for heterogeneous traffic-agents,” Proceedings of the AAAI Conference on Artificial Intelligence, pp. 6120-6127, 2019.

[4] J. Baltes, K.-Y. Tu, S. Sadeghnejad, and J. Anderson, ”HuroCup: competition for multi-event humanoid robot athletes”, The Knowledge Engineering Review, vol. 32, pp. 1-14, 2017.

[5] R. Gerndt, D. Seifert, J. H. Baltes, S. Sadeghnejad, and S. Behnke, ”Humanoid robots in soccer: Robots versus humans in RoboCup 2050”, IEEE Robotics & Automation Magazine, vol. 22, pp. 147-154, 2015.

[6] T. A. Shangari, F. Shamshirdar, B. Azari, M. Heydari, S. Sadeghnejad, and J. Baltes, ”Real-Time Ball Detection and Following Based on a Hybrid Vision System with Application to Robot Soccer Field”, in: Robot Intelligence Technology and Applications 4, ed: Springer International Publishing, pp. 521-527, 2017.

[7] S. Lengagneua, P. Fraisse, and N. Ramdani, ”Planning and fast re-planning of safe motions for humanoid robots: Application to a kicking motion”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 441-446, 2009.

[8] M. Gomez, Y. Kim, and E. T. Matson, ”Iterative learning system to intercept a ball for humanoid soccer player”, 6th International Conference on Automation, Robotics and Applications (ICARA), pp. 507-512, 2015.

[9] Y. Zhao, R. Xiong, and Y. Zhang, ”Model based motion state estimation and trajectory prediction of spinning ball for ping-pong robots using expectation-maximization algorithm”, Journal of Intelligent & Robotic Systems, vol. 87, pp. 407-423, 2017.

[10] Y. Zhang, R. Xiong, Y. Zhao, and J. Chu, ”An adaptive trajectory prediction method for ping-pong robots”, International conference on intelligent robotics and applications, pp. 448-459, 2012.

[11] Y. Liu and L. Liu, ”Accurate real-time ball path estimation with onboard stereo camera system for humanoid ping-pong robot”, Robotics and Autonomous Systems, vol. 101, pp. 34-44, 2018.

[12] D. Carneiro, F. Silva, and P. Georgieva, ”The role of early anticipations for human-robot ball catching”, IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 10-16, 2018.

[13] J. J. LaViola, ”Double exponential smoothing: an alternative to Kalman filter-based predictive tracking”, Proceedings of the workshop on Virtual environments, pp. 199-206, 2003.

[14] F. Janati, F. Abdollahi, S. S. Ghidary, M. Jannatifar, J. Baltes, and S. Sadeghnejad, ”Multi-robot Task Allocation Using Clustering Method”, in: Robot Intelligence Technology and Applications 4, ed: Springer International Publishing, pp. 233-247, 2017.

[15] J. Baltes, J. Bagot, S. Sadeghnejad, J. Anderson, and C.-H. Hsu, ”Full-Body Motion Planning for Humanoid Robots using Rapidly Exploring Random Trees”, KI - Knstliche Intelligenz, vol. 30, no. 3, pp. 245-255, 2016.

[16] W. Remus and M. OConnor, ”Neural networks for time-series forecasting”, in Principles of forecasting, ed: Springer, pp. 245-256, 2001.

[17] T. Kolarik and G. Rudorfer, ”Time series forecasting using neural networks”, ACM Sigapl Apl Quote Quad, pp. 86-94, 1994.

[18] I. A. Gheyas and L. S. Smith, ”A neural network approach to time series forecasting”, Proceedings of the World Congress on Engineering, pp. 1-3, 2009.

[19] Y. Zhao, R. Yang, G. Chevalier, R. C. Shah, and R. Romijnders, ”Applying deep bidirectional LSTM and mixture density network for basketball trajectory prediction”, Optik, vol. 158, pp. 266-272, 2018.

[20] K. Mironov and M. Pongratz, ”Applying neural networks for prediction of flying objects trajectory”, Newsletter UGATU, vol. 17, no. 6, pp. 28-32, 2013.

[21] K. Mironov, I. Vladimirova, and M. Pongratz, ”Processing and forecasting the trajectory of a thrown object measured by the stereo vision system”, IFAC-PapersOnLine, vol. 48, pp. 28-35, 2015.

[22] K. Mironov and M. Pongratz, ”Fast kNN-based Prediction for the Trajectory of a Thrown Body”, 24th Mediterranean Conference on Control and Automation (MED), pp. 512-517, 2016.

[23] B. Yazdankhoo, M. N. Shahsavari, S. Sadeghnejad, and J. Baltes, ”Prediction of a Ball Trajectory for the Humanoid Robots: A Friction-Based Study”, in RoboCup 2018: Robot World Cup XXII, ed: Springer International Publishing, pp. 387-398, 2019.

[24] S. Klanke and H. Ritter, ”Variants of unsupervised kernel regression: General cost functions”, Neurocomputing, vol. 70, no. 7-9, pp. 1289-1303, 2007.

[25] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso, ”SSL-Vision: The Shared Vision System for the RoboCup Small Size League”, in RoboCup 2009: Robot Soccer World Cup XIII, ed: Springer Berlin Heidelberg, pp. 425-436, 2010.