



Soccer Goalkeeper Task Modeling and Analysis by Petri Nets

Azadeh Gholami^{*}, Bahram Sadeghi Bigham

Computer Sciences and IT Department, IASBS, Zanjan, Iran

Received 12 August 2017; revised 30 January 2018; accepted 18 February 2018; available online 15 March 2018

Abstract

In a robotic soccer team, goalkeeper is an important challenging role, which has different characteristics from the other teammates. This paper proposes a new learning-based behavior model for a soccer goalkeeper robot by using Petri nets. The model focuses on modeling and analyzing, both qualitatively and quantitatively, for the goalkeeper role so that we have a model-based knowledge of the task performance in different possible situations. The different primitive actions and behaviors as well as the events to switch between them, and also environment models were designed and implemented. For this purpose, a modeling and analysis framework based on Petri nets is used, which enables modeling a robot task, analyzing its qualitative and quantitative properties and using the Petri net representation for actual plan execution. The proposed model building blocks and some tasks of robot are detailed. The novelty of approach is considering some alternatives through tasks execution, which are implemented by conflicts in their Petri net models, and also Q_learning employment in these decision points in order to learn the best policy. Therefore, the execution of actions in different tasks will be controlled effectively. The results of theoretical analysis of some case studies show impressive performance improvement in goalkeeper task execution.

Keywords: Goalkeeper, Robot task, Petri nets, Modeling, Analysis, Q_learning, Task planning.

1. Introduction

Robot task planning in the past was performed by a limited set of instructions that was programmed to it for ad-hoc applications. Therefore, having a systematic methodology to design and planning of robot task absorbed many attentions, and this reason was the great motivation for defining new powerful methods for designing and analyzing the robot task plans and behaviors. New methods have capability to present well and richer task plans which can be formally checked for performance properties. One of the formal approaches is Discrete Event Systems (DES), with one important powerful formalism: Petri nets [2, 17, 20]. Petri nets are graphical and mathematical modeling

tool applicable to many systems as a widely popular used formalism for DESs modeling and also in robotics field.

An interesting case for studying and developing of task plans is the role of Goalkeeper. Goalkeeper is a very important and challenging role in soccer, human or robotic, that it should perform correct actions and tasks, also coordinated with the other teammate robots. The main purpose of the goalkeeper is to defend goal from the opponent team attacks by performing proper behaviors in each situation it may encounter, like: covering the goal line, leaving goal and go ahead to stop opponent forward, intercepting the ball and so on.

This paper focuses on goalkeeper robot and uses a Petri net based framework which applies formal methods for

^{*} Corresponding author. Email: azadehgholami85@yahoo.com

modeling, execution and analysis of robot tasks. Optional ways in tasks executions, for the first time in literature, which lead to have decision points, provides alternatives in tasks which cause to more powerful, general and flexible model. Applying learning to decision points of the nets is one other major advantage of this work that allows to achieve more desirable and effective performance for goalkeeper.

Several works have been done in the field of modeling by Petri nets and also their performance improvement. Some of the more important and related works are briefly mentioned here. Lacerda used Petri nets to model discrete event systems (DESs) to present a methodology for performing supervisory control of DESs, based on linear temporal logic [13]. Marked Ordinary Petri Nets (MOPNs) as the simplest form of Petri nets, are used to model and synthesis deadlock free plans for a multi-agent environment by King et al., [8]. A representation framework based on Petri nets named Petri Net Plan (PNP) is presented by Ziparo and Iocchi [26] for describing robot and multi-robot behaviors. In Kim and Chung work [7] Generalized Stochastic Petri nets are used to model and analysis (both qualitatively and quantitatively) of a tour-guide robot task. Moreover, in many works, [10, 11, 12, 19, 22], Petri net based approaches are used to deal with different problems and issues of robot task planning, coordinating and analyzing.

Lausen et al., proposed a goalkeeper robot for the Middle-Size League of RoboCup [9] that its tasks implemented by a set of primitive tasks and behaviors that is coordinated by a 2-level hierarchical state machine [14]. Fuzzy logic approach is used for selection mechanism by Ramos et al. [23], using some behaviors and actions. An integrated framework is introduced in Costelha and Lima work [4], which is used in this paper, providing a tool to model, analysis and execution of mobile (multi-)robot task by Petri nets. The closest work to this paper work is of martins [16], which provides a behavior modeling for soccer goalkeeper robot.

In the field of Petri net optimal controlling several works have been done. A method to construct a supervisor based on reinforcement learning is proposed by Yamasaki and Ushio [25]. In their proposed method, specifications are given by rewards, and an optimal supervisor is derived by considering rewards for the occurrence of events and disabling events. A methodology and a set of algorithms

are proposed in Matos Pedro work, [18], for Generalized Semi-Markov Processes (GSMP) learning, using model checking techniques for verification and to propose new approaches for testing DES. Chang and Kulic presented a learning method that automatically creates Petri nets from observation of human demonstrations to model the tasks [3], in which the learned Petri nets are capable of generating action sequences to allow a robot to imitate the task. Leonetti had combined planning with Petri nets and reinforcement learning [15]. Kim et al., discussed about a multi-agent system for robot-soccer that consists of a supervisory controller, and controllers for attack, defense and goalie (goalkeeper) robots, using real time vector path planning, Petri net theory and Q-learning techniques [6].

The rest of this paper is organized as follows: next section introduces preliminary concepts and theoretical requirements. The new model for goalkeeper is introduced in Section 3. The model analysis process and task learning method is explained in Section 4. Section 5 includes the experimental results and analysis consequents. Finally, Section 6 covers some conclusion remarks and suggests some fields of study for future works.

2. Preliminaries

In this section the theoretical concepts and methods used in this work are described.

2.1. Petri Nets

Petri nets as a powerful modeling formalism which have origin from Carl Adam Petri dissertation, in 1962 [20], are able to provide modeling aspects of systems such as concurrency, parallelism, synchronization, stochastic features and decision making. A Petri net is a particular kind of directed graph, together with an initial state called the initial marking, M_0 . The underlying graph N of a Petri net is a directed, weighted, bipartite graph consisting of two kinds of nodes, called places and transitions, where arcs are either from a place to a transition or from a transition to a place. In graphical representation, places are drawn as circles, transitions as bars or boxes [17]. Generalized Stochastic Petri Net (GSPN) is a general form of Petri nets where includes timed and immediate transitions, enabling analysis of performance and functional behavior of a system. The following is the formal definition of GSPN [2, 7].

Definition: A GSPN is an eight-tuple like, $PN = \langle P, T_I, T_E, I, O, M_0, R, W \rangle$, where:

- $P = \{p_1, \dots, p_n\}$ is a finite, non-empty set of places;
- $T = \{t_1, \dots, t_m\}$ is a finite, non-empty set of transitions;
- $T_I \subset T$ denotes the set of immediate transitions;
- $T_E \subseteq T$ denotes the set of exponential transitions;
- $I: P \times T \rightarrow \mathbb{N}_0$ represents the arc connections from places to transitions, such that $i_{I_j} = 1$ or $i_{I_j} = 0$;
- $O: T \times P \rightarrow \mathbb{N}_0$ represents the arc connections from transitions to places, such that $o_{I_j} = 1$ or $o_{I_j} = 0$;
- $M_j = [m_1(j), \dots, m_n(j)]$ is the marking of the net at time j . M_0 is the initial marking of the net.
- $R: T_E \rightarrow \mathbb{R}^+$ is a function, $R(t_{E_j}) = \lambda_j$, where λ_j is the firing rate of the exponential transition;
- $W: T_I \rightarrow \mathbb{R}^+$ is a function, $W(t_{I_j}) = \omega_j$, where ω_j is the weight of the immediate transition;

The marking of the Petri net that shows the state of system, will change according to the firing rules. A transition is enabled if all its input places are marked at least with one token. If it fires, it removes one token from each input place and creates one on each output place; If there is a set of enabled transitions, both immediate and exponential, the former ones have higher priority and fire before exponential ones; If there is a set of enabled exponential transitions, $T = \{t_{E_1}, \dots, t_{E_n}\}$, the firing probability of each transition is given by

$$P(t_{E_j}) = \frac{\lambda_j}{\sum_{k=1}^n \lambda_k} \quad (1)$$

If there is a set of enabled immediate transitions, $T = \{t_{I_1}, \dots, t_{I_n}\}$, the firing probability of each transition is given by

$$P(t_{I_j}) = \frac{\omega_j}{\sum_{k=1}^n \omega_k} \quad (2)$$

2.2. Q_Learning

Q-Learning is one of the reinforcement learning algorithms [24] in which instead of assigning value to the states, it assigns a Q-value to a pair of (state, action) and updates these Q-values which are evaluations for state-action pairs. Q-values are updated as follows, which is known as Bellman equation:

$$\hat{Q}(s, a) \leftarrow r(s, a) + \gamma \max_{a'} \hat{Q}(s', a') \quad (3)$$

Where, $Q(s, a)$ is the estimation of the expected discounted total rewards when a learning agent takes an action a at state s , γ is a learning rate that ($0 < \gamma < 1$). The Q-values converges with probability 1 to a true value if γ decays appropriately and update of the Q-values is done an infinite number of times at each state-action pair.

2.3. Petri Net Based Framework for Modeling and Analysis

At the end of modeling goalkeeper, a framework is used that is developed by Costelha and Lima [4]. This framework introduces a modular and layered designation of tasks that allows to model from lower layer to the upper separately, and then composes these models to the single, full Petri net. All of the knowledge in this methodology is discretized and is shown based on logical predicates. Main components of framework methodology are different types of places, layers and single Petri net generator. There are Predicate, Action and Task places in this methodology and each named properly with a prefix according to its meaning. Because of the modular design process in this model, different layers are considered, as shown in Figure 1. Environment Layer (including environment changes), Action Executor Layer (including models of actions), and Action Coordinator Layer (including models of tasks).

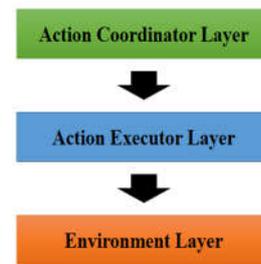


Fig. 1. Different layers of modeling framework

After modeling all layers, to compose them to a single full Petri net of the whole system, an algorithm is used that merges all environment, action and task Petri net models to one full Petri net. In merging of nets, place labels have important role because of distinguishing types of places by using them. By using full Petri net generator algorithm introduced in the framework [4], the single full Petri net is obtained that can be analyzed using techniques for calculating performance properties and also, the Reachability Graph and Algebraic Analysis can be used to achieve the transient analysis [17], as well as the qualitative and quantitative properties. By analysis matrices, transient stationary analysis to obtain, Qualitative Properties

(Boundedness, Liveness and deadlock) and Quantitative Properties (Probability that a condition holds, Probability of having a number of tokens in a place, expected number of tokens in a place and transition throughput rate) can be done [1,4,17] can be done.

3. Proposed Goalkeeper Model

To model actions, tasks, and respective environment based on the introduced methodology, it is necessary to do some preliminaries first. We need to discretize the game field as it is demonstrated in Figure 2. *In Goal* (1: The goal area and inside of the goal). *Close Goal* (2: The area that is close to the goal as shown). *Penalty Area* (3: Represents the penalty area in soccer field). *Off Goal* (4: The area off the penalty area and out of it). As said before, all of knowledge in this modeling methodology is defined as logic predicates. The predicates that have been considered for modeling goalkeeper are included in various kinds like position and motion related predicates like (*GKIN Goal*, *GK In Close Goal*, *Ball In Penalty Area*, *Ball In Off Goal*, *GK Stopped*, *BallMoving2Goal*), game (*Game Start*, *Game Over*), belief and knowledge of the goalkeeper (*GK See Ball*, *GK Has Ball*, *GK Close 2 Ball*), and so on. Note that ‘predicate.’ and ‘predicate. NOT_’ are the labels for positive and negative predicates, respectively.



Fig. 2. Field discretization

3.1. Goalkeeper Task Petri net Model

In this section, goalkeeper task components including environment, primitive actions, tasks or behaviors and overall role of goalkeeper are modelled. Note that all of the nets are created and handled by PIPE [21]. Moreover, to set the stochastic transition rates, the rates are set based on the knowledge of the transition meaning. E.g., for motion transitions the rates are calculated based on the physical properties obtained from field and other phenomenon. Note that ‘GK’ stands for Goal Keeper continuously.

3.1.1.Environment Modeling

In this layer, the ball and goalkeeper position models and also other information of environment are modelled, e.g., *Ball Position*, which includes changing the ball position between different areas under the effect of other phenomenon in the environment. The environment model for the ball position changing is presented in Figure 3. Moreover, *GK Position*, *GK Close 2 Ball*, *GK Has Ball* and other models are involved in this layer. All Petri net places in this layer are predicate places.

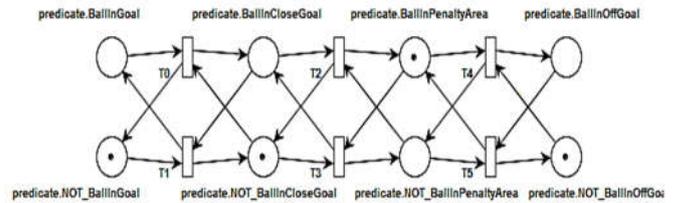


Fig. 3. Ball Position model

3.1.2.Primitive Actions Modeling

In this layer, action models are described based on their running-conditions and desired-effects. Each action needs some Running-conditions (conditions that need to be true for the action to be executed and produce any effect and change) and then effects on the environment and changes it, to obtain its Desired-effects (conditions that are the action execution results and action aims to make them true). These are summarized in Table 1, and are briefly defined in the following. Note that all places in this layer are predicate places.

Table. 1. The system and simulation specifications

Actions	Running-Conditions	Desired-Effects
Stop GK	-	GK Stopped
Go 2 Home Position	NOT_GK In Position	GK In Position
Defend On Goal Line	GK See Ball, GK In Position	GK In Goal, Ball Stopped
Go Towards The Ball	GK See Ball	GKClose2Ball
Close The Shot Angle	GK See Ball	GK Close 2 Ball, Ball Stopped
Catch Ball	GK See Ball, GK Close 2 Ball	GK Has Ball
Kick Ball	GK See Ball, GK Has Ball	NOT_GK Has Ball

Stop GK: Goalkeeper should be stopped (the velocity will be zero).

Go 2 Home Position: Represents the action of moving from any point of the field to the middle of the goal that is on the goal line, with orientation towards the center of the field.

Defend on Goal Line: This action includes moving on the goal line (or parallel to goal line) for following the ball and goal covering, when goalkeeper sees the ball.

Go Towards the Ball: This primitive action aim is to go towards the ball when it sees the ball and under some other situations.

Close the Shot Angle: This action includes moving forward to minimize the shot angles. i.e., goalkeeper aims at locating in a position that minimizes the angle between each post (sides of the goal), ball and the goalkeeper.

Catch Ball: This primitive action aims at catching the ball and getting the possession of the ball. To do this action and getting the ball, goalkeeper should first approach the ball.

Kick Ball: This action purpose is to move ball or kicking the ball to move it by moving against the ball in order to move it and remove it from the goal neighborhood.

3.1.3. Task Plan Models

In this section, task plans that are consist of compositions of the primitive actions of goalkeeper are modelled. It is considered five tasks and one general overall task for goalkeeper, in which their net places can be predicates, actions or tasks, as follows:

GK Stop Task: Executing the primitive action Stop GK to stop the goalkeeper.

GK Defend Task: The main task of goalkeeper to defend, whether it sees the ball or not.

Leave Goal Task: Leaving the goal to move towards the ball and stop the opponent.

Ball Interception Task: The ball interception when the goalkeeper sees the ball.

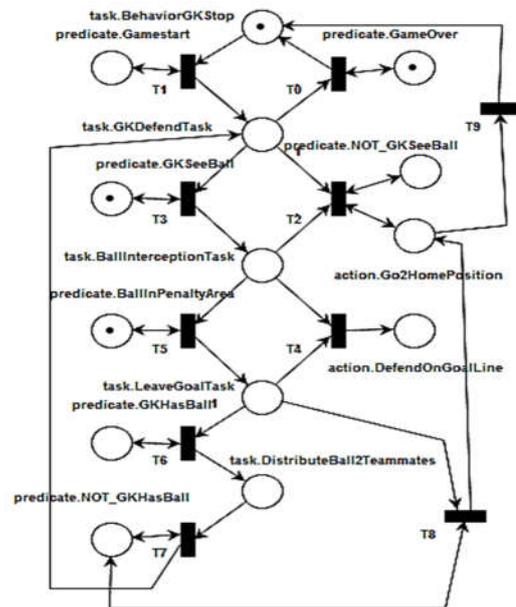


Fig. 4. GK Task, goalkeeper overall task plan model

DistributeBall2Teammates: When the goalkeeper has the ball, it should pass it to teammates to continue the game and maybe start a new attack.

GK Task: The main purpose of goalkeeper, i.e. goal defending via executing some tasks and behaviors and switching between them, to complete the role. All of this process is shown in model of overall task plan depicted in Figure 4.

4. Model Analysis and Task Learning

After constructing the various layers, in order to apply Petri net analysis methods to the proposed model, the validity and correctness of model should be evaluated by qualitative analysis, and then some performance evaluations are done to show the quantitative properties of the model.

4.1. Qualitative Properties Analysis of Model

To say briefly about qualitative properties of our model, by considering an initial state for each net, the boundedness of them has been checked, i.e., by considering the reachability tree, for each marking the number of tokens is determined. Also, for all Petri nets with a given initial marking, all of the transitions are live, meaning that all of them are fired in some reachable state. A deadlock state is a

state where none of the transitions are able to fire. Deadlock property had been checked for the model. Also, each of the actions and task Petri nets are determined to be safe, having at most one token per place.

4.2. Quantitative Properties and Performance Analysis

To do analysis on the performance of our model, we select the *Leave Goal Task* and *Ball Interception Task* as the base tasks to be expanded and to be analyzed through the experiments, and before that we describe these tasks more detailed.

Leave Goal Task: Through performing this task, goalkeeper first executes the *Close the Shot Angle* action and then under some situations, it can execute two primitive actions based on the value of predicate *GKClose2Ball*, which is the desired effect of *Close the Shot Angle*. When this predicate is true then a decision point or (conflict point with random switches) will occur and goalkeeper should decide to do one of these two primitive actions: *Catch Ball* or *Kick Ball* (to remove it from the goal neighborhood). So goalkeeper selects one of these actions to do, i.e., goalkeeper will execute one of these actions randomly or by setting random switches. Therefore, when the predicate *GKClose2Ball* is false, goalkeeper should execute the primitive action *Close the Shot Angle*, and then it reaches to the decision point as shown in Figure 5.

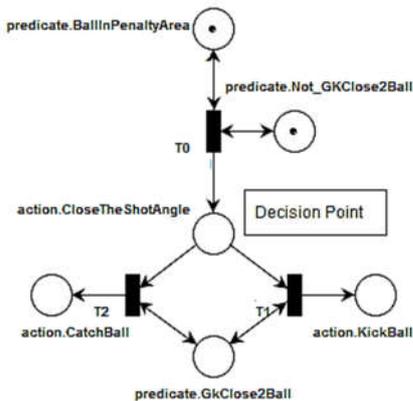


Fig. 5. *LeaveGoalTask* model

Ball Interception Task: One can consider two situations for this task execution: one, when the ball is close to the goal, and the other, when the ball is far from the goal. For the first situation, i.e., the ball is in penalty

area and so the predicate *Ball in Penalty Area* has true value, goalkeeper should execute the task *Leave Goal Task*. In the second situation, that ball is far from goal, and predicate *Ball in off Goal* is true, goalkeeper should cover the goal by following the ball movements and orientation. The goalkeeper should be positioned close to the goal and cover the goal line properly to defend it. So, goalkeeper should execute the primitive action *Defend on Goal Line*. Thus, a task and an action that are included in this task model are *Leave Goal Task* and *Defend on Goal Line*. In order to switch between these two elements of this task, there exists a conflict or decision point that will be checked in experiments and results will be analyzed.

This task Petri net model is depicted in Figure 6.

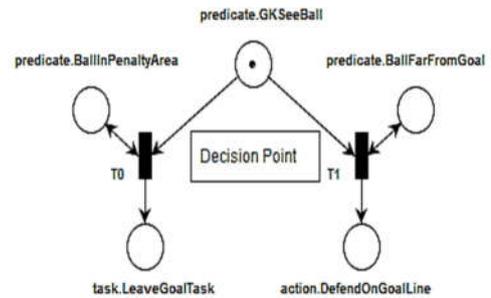


Fig. 6. *BallInterceptionTask* model

Now, it is needed to expand the selected task plans to the full Petri net. After generating single full Petri net of *Leave Goal Task* plan, as the first analysis example, by starting from a given initial marking, one wants to have a goalkeeper to decide in each situation of the positioning in the goal, to kick the ball or catch it, and do analysis on how this decision will influence on its overall performance in defending of goal. By considering an initial marking for the full Petri net of selected task plan, the reachability graph of this net will be obtained. It is considered that goalkeeper always sees the ball and at the first, the ball is in penalty area and goalkeeper is positioned in own goal.

As shown in Figure 5, by setting weights of transition T1 and T2, we can control this decision point. To decide to do *Catch Ball* or *Kick Ball*, the results of applying this weights to the transitions of the task plan, is diagrammed in Figure 7.

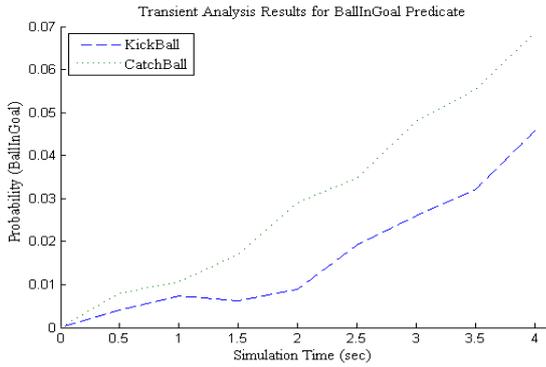


Fig. 7. Probability of scoring goal in two manners

As can be seen, the probability of scoring goal in own goal, i.e., having token in Ball In Goal predicate, is evaluated by applying different weights, when goalkeeper does the action Kick Ball by probability one and Catch Ball by probability zero, is less than the time that goalkeeper does the action Catch Ball only, i.e., the transition T2 with probability one and T1 with probability zero.

As another experiment, we analyze the predicate *GK Has Ball*, i.e., the predicate of having the possession of the ball, through executing the *Leave Goal Task*, and the result is shown in Figure 8, as is diagrammed, when goalkeeper does the action *Catch Ball*, it is more probable to receive the possession of the ball comparing with the time it does the action *Kick Ball*. Note that this probabilities and numerical results is achieved by averaging from values obtained of doing several times simulations on the full Petri Net of the selected task to be performed.

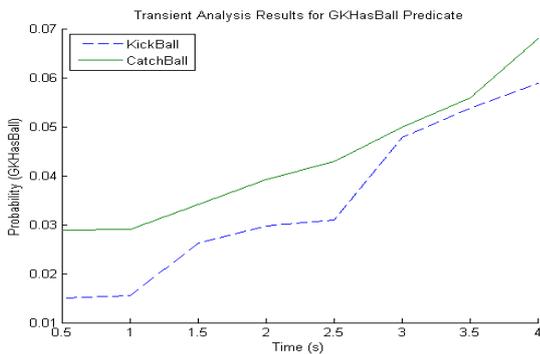


Fig. 8. The probability of having the possession of the ball

As another task analysis example, we want to evaluate the goalkeeper performance in defending goal by two alternatives, in Ball Interception Task model. When the goalkeeper just stays in goal and the other when it leaves

the goal and goes forward to stop the opponent player. After several running of the full Petri Net of the *Ball Interception Task* and first, by assigning weight one to T0 and zero to T1, i.e., doing the *Leave Goal Task* by probability one and the action *Defend On Goal Line* by probability zero, and then vice versa, i.e., assigning weight zero to T0 and one to T1, i.e., doing the *Defend On Goal Line* by probability one and the task *Leave Goal Task* by probability zero, the probability of having token in *Ball In Goal* is diagrammed in Figure 9. As shown in diagram, the probability of goal scoring is lower when the goalkeeper just stays in goal with respect to the time it leaves the goal and goes forward.

GKQ_learning: In conflict points of the nets, goalkeeper should decide to do one of the alternatives. This can be done through assigning weights in a desirable manner to the transitions involved in these points, based on an effective policy obtained from a learning process. Therefore, we propose GKQ_learning and describe its components as follows:

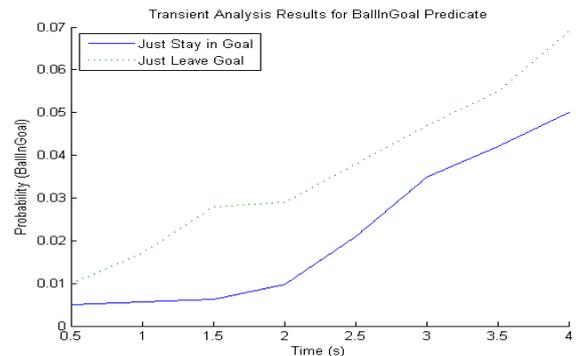


Fig. 9. The probability of scoring goal comparing two behaviors

– Learning model: We consider the full Petri net of Ball Interception Task model. Assume the motion of goalkeeper in the field that it should decide to stay or move forward or backward, in each areas that it will be positioned. For example, at the first, when goalkeeper is located in goal, it should decide to move forward and leave the goal to stop the opponent player, or just stay in the goal and on the goal line to defend goal. So, we consider this process as a Markov Decision Process (MDP), with respect to observation of the goalkeeper, and we will apply learning to this process.

– State and action spaces: The set of states are the markings that are reachable, starting from a given initial marking of the full Petri net of the base task, Ball

Interception Task. Actions that we consider are a set consisting of three elements, i.e., ‘stay’, ‘move forward’ and ‘move backward’.

– Q-Table and reward system: By considering the mentioned MDP, we use a Q-table that its elements are the approximation values for the Q-values of the learning agent, which can be initialized with all zeros (for instance). The learning agent iteratively identifies the current state and do an action like a . The result of this action is the reward $r(s, a)$ for the agent and it will see the new state $s' = \delta(s, a)$. The elements of the Q-table will change according to the previously mentioned Bellman equation. We consider the set {GK Has Ball, Ball Stopped} as the absorbed goals of goalkeeper, and define the reward values. When these two predicate places have one tokens, meaning that these conditions are hold, and the reward is considered as the $100 \times (1)$ or $100 \times (2)$, when one of these predicates is true, and when both of them have tokens, respectively. So the values of the reward table for feasible states changings, are 100 or 200.

5. Experimental Results

In this section we want to provide the results of applying the GKQ_learning to Ball Interception Task, to achieve desired policies for goalkeeper behavior. We assume goalkeeper always sees the ball, which is also the running-condition of some of the actions that it performs. Also, the goalkeeper and the ball position in the initial state are In Goal and In Penalty Area, respectively. By the considerations already said and the initial marking for the full Petri net of Ball Interception Task, after simulation of this net, the reachable markings are obtained. By analyzing the token transformation between places in all reachable markings a scenario of goalkeeper movements and actions will be generated. By applying GKQ_learning to this scenario, by constructing the rewards of each step and obtaining the Q-Table at the end, the goalkeeper will learn to decide the best choice in each decision point. Now to show the effect of learning in goalkeeper task improvement, by considering the state transformation and goal state (s) represented before, and the purpose which is to reach a policy, actually, the sequence of actions, under related decision makings, to find the best and optimum way to defend the goal from scoring goal by opponent team, we investigate the probability of having token in Ball In Goal predicate place to show the promotion of goalkeeper

performance in goal defending. After repeating 10 times, and running the algorithm for 10000 episodes with different learning rates, which also results to use $\gamma=0.8$, we finally reached to averaged convergence values of Q-Table. According to the convergence Q_values that have been reached, our goalkeeper model uses these values to set the weights of the transitions in full Petri net of the selected base task to have the best decision about doing actions stay, go forward or go backward, and by tracing the sequence of states, it computes the maximum Q-value for that state, by finding action which satisfies the mentioned value. The results of average probability of scoring goal (having token in Ball In Goal), was calculated via 10 times execution of the full net, comparing before and after learning, diagrammed in Figure 10. As expected and results show, this probability is decreasing after learning with respect to before learning. This is an expected result proving that GKQ_learning leads to great improvement in goalkeeper task execution. Therefore, the knowledge of goalkeeper will be promoted through applying GKQ_learning, which can be used in runtime, to change the weights of the transitions involved in conflict points, according to the reached policy.

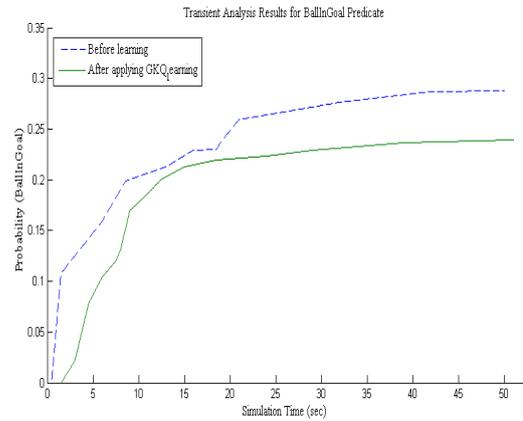


Fig. 10. Probability of scoring goal before and after learning

6. Conclusion

A new learning-based model for soccer goalkeeper robot was proposed using Petri nets, where the knowledge of the robot itself and its surrounding environment is approximated through logic predicates. Considering decision points in nets and applying a proposed learning process to the decision points of the model, to achieve desired behavior policy, as the first time in literature, is a

great advantage of this work to promote goalkeeper's performance in different situations it may encounter. The model analysis that was performed completely shows the improvement.

Note that, this work mainly focuses on a complete modeling process, description of theoretical analysis of the model, and also proposing an effective method for performance improvement with obtaining to a desired policy for robot, so the implementation of the work in real robot was not the case, and is prospective work of the authors. Designing and analyzing other soccer robot players maybe other future field for work. Using of other systematic methodologies in modeling process can be done in future. Also, applying other learning methodologies will be interesting field of study.

Acknowledgment

We wish to thank Professor Pedro U. Lima, who helped us through this work, and also special thanks to Hugo Costelha for his support and help in this work.

References

- [1] Bause, F.; Kritzinger, P. S., "Stochastic Petri Nets: An introduction to the theory", 2nd ed, Vieweg Verlag, Vol. 12, (2002).
- [2] Cassandras, CH. G.; Lafortune, S., "Introduction to Discrete Event Systems", 2nd ed, Springer US, ISBN 978-0-387-33332-8, (2008).
- [3] Chang, G.; Kulic, D., "Robot Task Learning from Demonstration Using Petri Nets", 22nd IEEE International Symposium on Robot and Human Interactive Communication, Gyeongju, Korea, August (2013).
- [4] Costelha, H.; Lima, P., "Robot task plan representation by Petri nets: modelling, identification, analysis and execution". *Autonomous Robots*, pp. 337-360, (2012).
- [5] Damas, B.; Lima, P., "Stochastic discrete event model of a multi-robot team playing an adversarial game". *Proceeding Of 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles – IAV2004*, (2004).
- [6] Kim, D.; Kim, Y.; Kim, K.; Kim, J.-H.; Vadakkepat, P., "Vector Field based Path Planning and Petri Net based Role Selection Mechanism with Q-learning for the Soccer Robot System", in *Int. J. Intelligent Automation and Soft Computing*, (2000).
- [7] Kim, G.; Chung, W., "Navigation Behavior Selection Using Generalized Stochastic Petri Nets for a Service Robot". *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(4): pp. 494-503, July (2007).
- [8] King, J.; Pretty, R. K.; Gosine, R. G., "Coordinated execution of tasks in a multiagent environment", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, pp. 615-619, (2003).
- [9] Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; Osawai, E.; Matsubara, H., "Robocup: A challenge problem for ai and robotics". *RoboCup-97: Robot Soccer World Cup I*, Springer-Verlog, pp. 1-19, July (1997).
- [10] Kloetzer, M.; Mahulea, C., "A Petri net based approach for multi-robot path planning". *Discrete Event Dynamic Systems*, Volume 24, Issue 4, pp. 417-445, (2014).
- [11] Kloetzer, M.; Mahulea, C.; Colom, J. M., "Petri net approach for deadlock and collision avoidance in robot planning". Technical report, (2013).
- [12] Kloetzer, M.; Mahulea, C.; Colom, JM., "Petri net approach for dead-lock prevention in robot planning", 18th international Conference on Emerging Technologies and Factory Automation (ETFA 2013), (2013).
- [13] Lacerda, B. F. A., "Supervision of Discrete Event Systems Based on Temporal Logic Specifications". Dissertation, INSTITUTO SUPERIOR TECNICO, (2013).
- [14] Lausen, H. H.; Nielsen, J. B.; Lima, P. U., "Model and behavior-based robotic goalkeeper", *RoboCup 2003: Robot Soccer World Cup VII*, pp. 169-180, (2004).
- [15] Leonetti, M., "Reinforcement Learning in Plan Space, Ph.D". Dissertation, Sapienza University of Rome, Department of Computer and System Sciences, (2010).
- [16] Martins, C., "Goalkeeper robot behavior design and coordination in soccer robotics", MSc, Department of Electrical and Computer Engineering, Lisbon University, (2010).
- [17] Murata, T., "Petri Nets: Properties, analysis and applications", *Proceeding of the IEEE*, Vol. 77 (4), pp. 541-580, (1989).
- [18] Matos Pedro, A. de, "Learning and testing stochastic discrete event systems", Master's thesis, Universidade do Minho, Portugal, December (2011).
- [19] Pereira, J. N.; Silva, P.; Lima, P. U.; Martinoli, A., "Formalization, implementation, and modeling of institutional controllers for distributed robotic systems", *Artificial Life*, Vol. 20, num. 1, pp. 127-141, Cambridge: MIT Press, (2014).
- [20] Petri, C. A., "Kommunikation mit automaten", Institut fur Instrumentelle Mathematik, Bonn, Technical report, English translation, (1966).
- [21] Akharware, N., "PIPE2: Platform Independent Petri net Editor", MSc Degree in Computing Science thesis, Department of Computing, Imperial College of Science, Technology and Medicine (University of London), pipe2.sourceforge.net, (2005).
- [22] Qin, Y.; Xu, R., "GSPN-based Modeling and Analysis for Robotized Assembly System", *IEEE International Conference on Robotics and Biomimetics*, 21-26, (2009).
- [23] Ramos, N.; Lima, P. U.; Sousa, J., "Robot behavior coordination based on fuzzy decision-making", *ROBOTICA 2006- 6th Portuguese Robotics Festival*, (2006).
- [24] Sutton, R. S.; Barto, A. G., "Reinforcement Learning: An Introduction", Vol. 1, No. 1, Cambridge: MIT Press, (1998).
- [25] Yamasaki, T.; Ushio, T., "Supervisory Control of Partially Observed Discrete Event Systems based on a Reinforcement Learning", *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3 (200m 3).
- [26] Ziparo, V. A.; Iocchi, L., "Petri Net Plans (PNPs), *Proceeding of the Fourth International Workshop on Modelling of Objects, Components, and Agents*" (MOCA'06), pp. 267-290, (2006).