Advances on Simulation Based Selection of Actions for a Humanoid Soccer-Robot

Heinrich Mellmann and Benjamin Schlotter
Adaptive Systems Group, Department of Computer Science,
Humboldt-Universität zu Berlin,
Unter den Linden 6, 10099 Berlin, Germany
{mellmann,schlottb}@informatik.hu-berlin.de
http://www.naoth.de

Abstract*—This paper introduces a method for making fast decisions in a highly dynamic situation, based on forward simulation. This approach is inspired by the decision problem within the RoboCup domain. In this environment, selecting the right action is often a challenging task. The outcome of a particular action may depend on a wide variety of environmental factors, such as the robot's position on the field or the location of obstacles. In addition, the perception is often heterogeneous, uncertain, and incomplete. In this context, we investigate forward simulation as a versatile and extensible yet simple mechanism for inference of decisions. To evaluate an action, the outcome is simulated based on the estimated state of the situation. The simulation of a single action is split into a number of simple deterministic simulations –** ***samples **– based on the uncertainties of the estimated state and of the action model. Each of the samples is then evaluated separately, and the evaluations are combined and compared with those of other actions to inform the overall decision. This allows us to effectively combine heterogeneous perceptual data, calculate a stable decision, and reason about its uncertainty. This approach is implemented for the kick selection task in the RoboCup SPL environment and is actively used in competitions. In simulated experiments we validate the new scheme and evaluate different strategies.**

***Index Terms*—decision making, action selection, forward simulation**

I. Introduction

A highly dynamic environment requires a robot to make decisions quickly and with limited information. In the RoboCup scenario, the robot that is in possession of the ball needs to take action as quickly as possible before the opponent players get a chance to interfere. However, the particular situation might be very complex and many aspects, like the robot's position on the field, as well as the positions of the ball and obstacles, need to be taken into account. This makes inferring a decision a complicated task.

In this work we propose an inference method based on forward simulation to handle this complexity and ensure short reaction times at the same time. We focus in particular on the RoboCup scenario where the robot has to choose the best kick from several different possibilities, which provides the motivation for our approach.

Outside of RoboCup, internal forward simulation has already been successfully used as an inference method in robotics. In [1], forward models are used to predict the noise a robot produces by intended motor actions. In [2], the authors investigate navigation of robots in a dynamic environment. They use a simulation approach to envision movements of other agents and pedestrians to enable avoiding dynamic obstacles while moving towards a goal. In [3] the authors introduce a pancake baking robot which is planning its actions using a full physical simulation of the outcome of possible actions. In [4] the authors use a physics based action selection scheme to generate and select robot actions to maximize the motion of the articulated object and thus learn a better model of the object. [5] introduces Imagination-Augmented Agents to complement a RL algorithm which solves puzzle games like sokoban. To estimate the state after an action a simulation-based approach is used.

In the RoboCup community, there have already been several attempts to implement similar methods. In particular [6], [7] and [8] focus on a very similar task – the selection of the optimal kick. In [6], a probabilistic approach is used to describe the kick selection problem which is then solved using Monte Carlo simulation. In [7], the kick is chosen to maximize a proposed heuristic *game situation score* which reflects the goodness of the situation. In [8], the authors use an instance based representation for the kick actions and employ Markov decision process as an inference method. In [9] the authors find that projection of the intention of other players can significantly improve the performance of path planning algorithms.

The work presented in this document is a continuation of the simulation based approach for selection of actions presented in [10]. The presented simulation based inference method has been implemented for the NAO-robot and actively used by the team Berlin United at the RoboCup competitions for several years. Evaluation based on labeled video and log data from real RoboCup competitions was presented in [10]. The results show a significant improvement in comparison to our previous methods.

For an effective decision, data from heterogeneous sources (e.g., visual percepts, ultrasound) needs to be combined. Often different filtering/modeling techniques are used for state estimation, which adds to the difficulty of the inference of decisions. In particular, representation and propagation of uncertainty is problematic. As we will show, the simulation based approach can provide a straightforward way of dealing with uncertainty.

The intuition behind a simulation-based approach is to *imagine* (or simulate) what would happen as a result of the execution of a particular action and then choose the action with the optimal (imagined/simulated) outcome. A potential issue with this approach is that the quality of the decision depends on the quality of the simulation, i.e., the model of the environment. For example in [3], [11], the robots use complete fine-grained physical simulations for their decision-making. In contrast, we argue that the simulation itself can be quite coarse. To compensate for errors in the simulation, it is executed a number of times with varying initial conditions sampled according to the estimated state of the situation. Each of these realizations is evaluated individually and the overall decision for an action is then based on the distribution of the particular evaluations of the simulation. This is repeated for all possible actions (kicks) and the action with the best outcome distribution is chosen for execution.

In this paper we analyze the simulation scheme, discussed in [10], in more detail. In particular, we focus on generalization of the structure and formalization of the algorithm, and extension to parametric actions with parameters in continuous spaces. We also empirically analyze the stability of decisions depending on the number of predictions, and analyze the performance of different decision schemes involving parametric actions in comparison to purely discrete action selection, as presented in [10], in a simulated RoboCup scenario.

Our results show that a small number of predictions are already sufficient for a stable decision, making the algorithm suitable for running on a robot with limited resources, such as the NAO. The results also indicate that optimizing action parameters can lead to better-performing strategies. The algorithm involving parametric actions was able to score a goal on an empty field in significantly less time than the version without optimization.

The remainder of the paper is structured as follows. In the next section we discuss the action selection problem within the RoboCup domain on a simplified example and derive the general scheme of the algorithm. In the Section III we introduce a method for evaluation of actions based on stochastic forward simulation. In the Section IV we introduce a decision scheme for parametric actions and discuss different decision strategies. Our experimental findings are discussed in Section V. Finally we conclude our findings in Section VI.

II. Decision making in Robot Soccer – an Introductory Example

To illustrate the task of making a decision in RoboCup consider a basic example as shown in Figure 1 (top left). The shown situation contains only the robot, the ball and the opponent goal. For simplicity, we assume the robot is able to shoot the ball in any direction. The task is then to choose a direction to shoot the ball to.

In a situation like the one shown in Figure 1 (top left) this problem can be solved geometrically in a straight forward manner. For simplicity we choose the direction towards the center of the opponent goal. If we move the ball to the side closer to the goal line as shown in the Figure 1 (top right), this simple strategy will obviously not work as expected. There is a significant chance for the ball to leave the field due to imprecise execution of the kick or disturbances of the field surface. The ball is also very likely to collide with the goal post or goal side and not actually enter the goal.

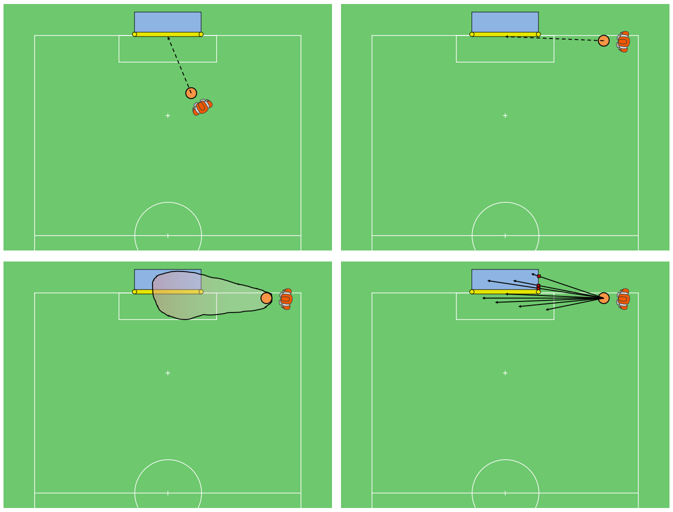

Fig. 1. A basic example illustrating the decision problem in robot soccer and a solution approach based on simulation.

To address these aspects we consider a model for the uncertainty of the ball's trajectory after the kick. Such model can be trained on the robot. We call this model the *action model*. The distribution of the possible ball paths might then look as illustrated in Figure 1 (bottom left). With such a model we could try to estimate the chances of the ball leaving the field, colliding with the goal, and entering the goal. Based on these estimations we could, for instance, choose a direction minimizing the chance of leaving the field and maximizing the chance of scoring the goal.

The major challenge for estimating these probabilities for particular outcomes is the complexity of the environment, which makes it difficult to directly calculate the corresponding integrals. Instead, we approximate the probabilities of particular outcomes by sampling the action model; in particular, we sample the possible trajectories the ball could take after the kick Figure 1 (bottom right). For a particular trajectory we can explicitly calculate whether the ball collides with the goal, leaves the field or ends up in the goal. In case that the likelihood of a ball entering the goal is very low, the different paths can be compared based on the distance to the goal. Then the best action could be chosen based on the mean distance to the opponent goal.

This example considerations lead to the following general scheme for a decision algorithm:

1) Simulate - the possible outcomes for each action;
2) Evaluate - each of the simulated situations;
3) Select - an action based on integrated evaluations of each action in consideration (Gain vs Loss)

In the above example we considered a very limited scenario. A real RoboCup game poses a much more complex situation, due to incomplete and noisy estimation of overall state, as well as the presence of other agents, both from own and opponent teams. Coping with this complexity requires more complex action models or a finer-grained simulation, or a more elaborate decision process. However, the overall approach remains true. In the following section we will see how this scheme can be generalized and applied to more complex situations.

III. Evaluating Actions with Stochastic Forward Simulation

With the example from Section II in mind, the robot can make a decision based on the expected value of an action. In this section we discuss how such value of an action can be estimated in a straightforward manner with the help of forward simulation. In the following subsections we briefly describe the particularities for the implementation of the state estimation, action models, simulation, and evaluation processes. Some aspects of these processes have been presented in more detail in our previous work [10].

A. State Estimation

In our case, the state of the situation consists of the robot's position on the field, the position of the ball relative to the robot, the positions of the teammates, and the obstacles in close proximity and the corresponding uncertainties. Each of these factors is modeled by a different independent probabilistic algorithm; specifically, a particle filter for self localization and a multi-hypothesis extended Kalman filter for the ball.

Let $\mathcal{S} \subset \mathbb{R}^m$ be the set of all possible states and s_t the probability variable describing the estimated state of the situation at the time t with it's uncertainty expressed by the probability distribution$P(x_t)$. Let $\mathcal{A} \subset \mathbb{R}^n$ be a set of all possible actions and $\hat{\mathcal{A}} = \{a_i | i = 1, .., k\}$ the set of random variables describing the set of actions available to the robot. The effect of an action $a \in \hat{\mathcal{A}}$ is described by an action model $P(s_{t+1}|s_t, a)$, the probability distribution of the transition from state s_t to state s_{t+1} after the execution of the action a.

B. Action Models

In order to simulate the result of an action, we need models for the effect of the action on the state of the situation, for the dynamics of particular objects and for interactions between the objects. As mentioned before, we assume the robot being able of following three kicks: short kick right, short kick left, and a long kick forward. Before the execution of a kick the robot may turn around the ball in order to improve the outcome of the kick. So each action consists of two parts: turning around the ball, and executing one of the kicks.

The effect of a kick can be described by the resulting movement of the ball. In particular, by its initial direction and the initial velocity after the kick. The resulting trajectory of the ball is then simulated by a simple rolling friction model as described in [10]. We assume that the action is over when the ball comes to a halt. This can happen either due to friction on the floor or due to a collision with the goal, another player, or an obstacle. Thus, we describe the outcome of a kick through the distribution of the expected stopping locations of the ball.

With this, an action can be described as a parametric tuple $a(\rho) := (\rho, v_0, \alpha) \in \mathbb{R}_+ \times [-\pi, \pi) \times [-\pi, \pi)$, with the rotation around the ball ρ, and the kick described by the initial velocity v_0 and the direction α of the resulting ball movement. We assume the direction of the ball motion α and the initial velocity v_0 of the kick behaving independently in accordance with the normal distribution. For a particular kick the parameter v_0, α and the corresponding standard deviations σ_v, σ_α are fixed. They are estimated empirically. The rotation around the ball before the kick ρ is a free parameter and is simulated without additional noise.

C. Simulation

The task of the simulation process is to predict the state of the situation after the execution of a kick. In general, an exhaustive physical simulation is a complicated and resource consuming process. To reduce complexity, we make several assumptions. We focus only on simulating aspects involved in the action, i.e., the motion of the ball and its potential collision with obstacles and goals. We furthermore assume that all objects excluding the ball remain static. Though this is obviously not true, the velocity of the ball is usually much higher than that of the robots, which makes it a viable assumption in this case. To model collisions with obstacles, especially goals, we assume a fully nonelastic collision, where the ball's trajectory ends at the point of contact.

More generally, we use simulation to estimate the action model $P(s_{t+1}|s_t, a)$. The dependency between the states and the action can be very complex in general, to approach this, we assume there exists a causality function $F : \mathcal{S} \times \mathcal{A} \longrightarrow \mathcal{S}$ with $s_{t+1} = F(s_t, a)$, mapping a particular state and an action to the resulting state. The simulation process approximates the function F. With this we can define a *hypothesis* for an action $a \in \mathcal{A}$ as a set of $N \in \mathbb{N}$ samples drawn from the model distribution of the action a and the state estimate s_t:

$$\mathcal{H}_a := \{F(\hat{s}_i, \hat{a}_i) | (\hat{s}_i, \hat{a}_i) \sim P(s_t, a), i = 1 \dots N\} \subset \mathcal{S} \quad (1)$$

Figure 2 illustrates the hypotheses (resulting positions of the ball) for the three possible kicks without prior rotation around the ball. Most of the samples for the right kick left the field, while the samples for the left kick are completely inside the field. The samples of the forward kick illustrate how complicated the resulting distribution can be: a partition of the samples is inside the goal, another large part is inside the field and a few have left the field.

D. Evaluation

With the simulation process described in the previous section, the likelihood $P(s|s_t, a)$ of the occurrence of an event

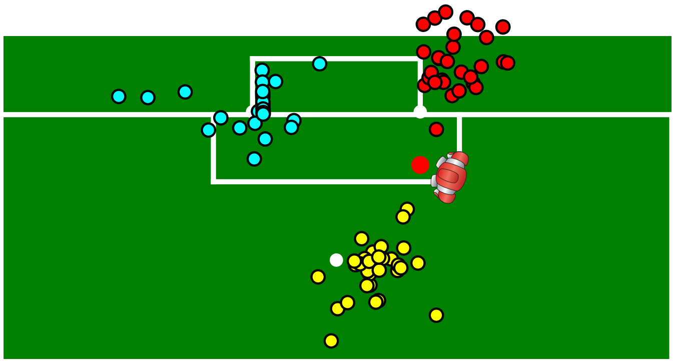

Fig. 2. Simulation of three different kicks: sampled distributions of the possible ball positions after a left (yellow) and right (red) sidekick, and the forward kick (cyan).

$s \subset \mathcal{S}$ as a result of the action a in state s_t can be statistically estimated from the hypothesis $\mathcal{H}_a$ as:

$$P(s|s_t, a) \sim \frac{|\mathcal{H}_a \cap s|}{|\mathcal{H}_a|}. \tag{2}$$

We define a set of target states $S_{goal} \subset \mathcal{S}$, consisting of situations in which the ball is inside the opponent goal. We also define a set of situations $S_{out} \subset \mathcal{S}$ containing situations where the ball is inside the own goal or outside of the field, and $S_{infield} \subset \mathcal{S}$ consisting of all situations where the ball is inside the field area. We can use this sets to evaluate and compare simulated future states. The states in S_{goal} are desirable and those in S_{out} should be avoided. For the comparison of the neutral states $S_{infield}$ we introduce a scalar value function $v : \mathcal{S} \longrightarrow \mathcal{R}_+$ assigning each state $s \in \mathcal{S}$ a value $v(s) \in \mathcal{R}_+$. The function v is encoding the team strategy. The value function introduced in [10] is a manually created potential field optimizing two criteria: positions closer to the opponent goal get a better potential and positions further away from the own goal are also better.

To ensure meaningful values outside of the field area, we assume that the value function assumes its maximal value inside the opponent goal $v|_{S_{goal}} = v_{max}$, and vanishes outside of the field $v|_{S_{out}} = 0$.

With this we can define the utility of an action $a \in \mathcal{A}$ in state s_t through

$$u(a, s_t) := \int_{s \in \mathcal{S}} v(s) \cdot P(s|s_t, a) ds. \tag{3}$$

The utility of an action can be estimated as

$$u(a, s_t) \sim \frac{1}{|\mathcal{H}_a|} \sum_{\hat{s} \in \mathcal{H}_a} v(\hat{s}). \tag{4}$$

Applying the special values for S_{goal} and S_{out} we get

$$u(a, s_t) = v_0(a) \cdot p_{infield} + v_{max} \cdot p_{goal} \tag{5}$$

with $v_0(a) = \frac{1}{|\mathcal{H}_a \cap S_{infield}|} \sum_{\hat{s} \in \mathcal{H}_a \cap S_{infield}} v(\hat{s})$, $p_{infield} := P(S_{infield}|s_t, a)$, and $p_{goal} := P(S_{goal}|s_t, a)$. Thus, we only need to evaluate the function v for the states inside the field.

In order to explicitly control the maximal likelihood of the ball leaving the field, we introduce the threshold $T_{out} > 0$ and define the trimmed utility function

$$\hat{u}(s_t, a) := \begin{cases} 0, & \text{if } \sigma_{risk}(a) > T_{out} \\ u(a, s_t), & \text{otherwise} \end{cases} \tag{6}$$

Note, that for each evaluation of the utility function $\hat{u}$ a full stochastic simulation, as described above, has to be performed.

IV. Decision Algorithm

With the example from Section II in mind, we consider the situation where a robot has already approached the ball and needs to choose the best kick action. Before the kick, the robot may adjust its rotation around the ball to improve the effectiveness of the kick.

In this section we discuss how the best action can be selected. We assume that we have a finite number of parametrizable kick actions as described in Section III-B. The general approach is to determine an optimal parametrization for each of the actions in the first step, and then select one of the optimal actions, which is promising the best outcome. Thus, the main decision algorithm consists of two steps:

1) optimize the parameters of each action separately;
2) select the best action from the set of optimized actions;

The overall decision has to take into account the trade-off between possible risks, e.g., ball leaving the field, and possible gains, e.g., scoring a goal, weighted by the likelihood of their occurrence. The estimation of those risks and gains can be done based on the individual ratings of the particular simulation results, i.e., samples.

In our scenario we assume that the robot can turn around the ball in order to improve the outcome of a kick. Thus, before selecting a kick we estimate the rotation around the ball which would be necessary to for each kick to be optimal. One criteria for the optimality is the value of the action. The other is the risk of the ball being taken over by the opponent if the robot spent too much time at the ball. Note that these risks arising due to optimization of each action (turning around the ball) is not encoded as part of the utility function $\hat{u}$ introduced in Section III-D.

This leads to a classical saddle-point problem: maximizing the chance of scoring a goal while minimizing the chance of own goal. We can split this problem and optimize different aspect in both steps of the decision scheme, which leads to different strategies. For example, in the first step each individual action might be optimized to have minimal rotation time around the ball. Then in the second step we could select the action with the higher chances to score a goal.

Note, that both steps of the above decision scheme are relying internally on the basic simulate-evaluate-decide-scheme introduced in Section II.

A. Strategies

To evaluate the presented decision scheme, we introduce four different strategies.

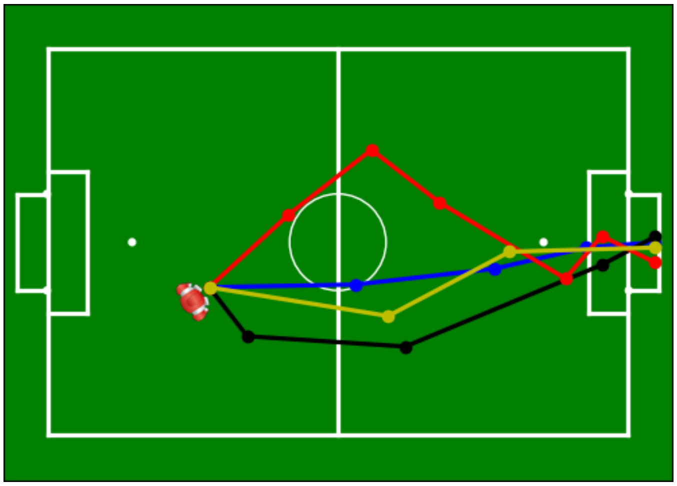

Fig. 3. Visualization of the path the ball takes from on position to the goal using different strategies. The red lines represents the *fast* strategy. The blue path represents the *optimal one* strategy. The black path represents the *optimal all* strategy. The *optimal value* strategy is represented by the yellow path.

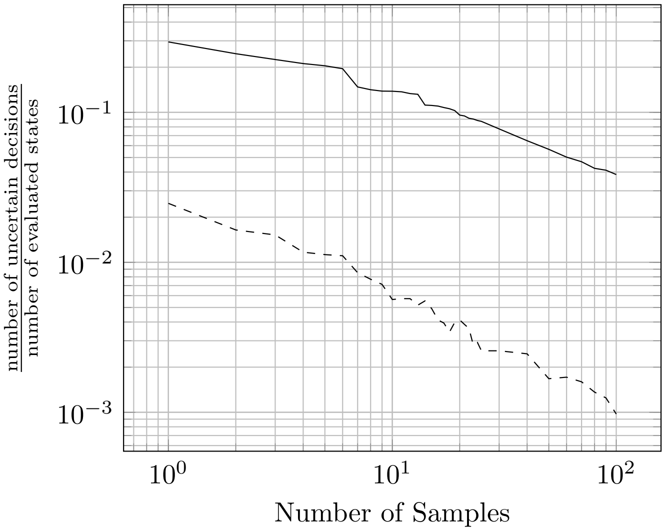

Fig. 4. Log-Log plot of the variation in decision depending on the number of simulations used to make a decision. The variation of the decisions depends linearly on the number of simulations.

The *fast* minimizes the rotation in both steps of the algorithm. As the result, this strategy does not turn around the ball except when all kick actions are considered as *too risky* or no kick can improve the ball position. In this case the robot turns towards the goal, until one action becomes acceptable.

The *optimal value* strategy maximizes the utility function $\hat{u}$ in both steps of the algorithm. As the result an action is chosen, which maximizes the likelihood to score a goal.

The *optimal all* strategy maximizes the utility $\hat{u}$ of each action in the first step, and selects the one with the minimal rotation around the ball, in the second step. Thus, this strategy represents a trade off between the strategies *fast* and *optimal value*.

The *optimal one* strategy assumes that only the forward kick can be used. In this strategy the robot will always turn in the direction of the gradient of the potential field. Many team in SPL use a similar strategy. It was included to evaluate the benefits of additional actions like sidekicks.

In Figure 3 one example of the kick sequence is shown for each of the strategies.

V. Experimental Results

A. Stability of Decisions Depending on Number of Simulations

A central aspect of the simulation based decision approach is the evaluation of samples and not the full underlying probability density functions as described in **??**. To make this algorithm usable in real time on the NAO robot platform, only a small number of samples can be used. To test if using more samples results in a significant qualitative improvement, multiple experiments were conducted using an abstract 2D simulator. We discretized the field in 30cm steps and in 5° rotation steps. For each of these positions the decision was calculated 100 times each with different number of samples. The highest column in each resulting decision histogram represents the corresponding most likely decision. In Figure 4 the number of robot poses in which the most likely decision is chosen more than 80 % of the time and more than 50 % is visualized. We can see that sing 20 samples or more guaranties that only in 10 percent of all states the decision chosen have a likelihood of 20 % or less. In [10] 30 samples were used. This analysis shows that using 30 samples to estimate the result of a kick is appropriate as increasing the number of samples has only a small effect on the uncertainty.

B. Strategy Evaluation

To evaluate the four different strategies described in Section IV-A we performed multiple experiments in an abstract 2D simulator. All strategies lead to either a goal or to ball out. We evaluated how the strategies perform on an empty field regarding how much the robot had to move until scoring a goal or shooting out. We assume the following simple robot behavior. If the robot position is not the ball position the robot turns towards the ball and then walks straight towards it. If the kick action and strategy demand a different rotation at the ball the robot turns accordingly. To simulate consecutive action we make the assumption that the kick action models accurately model the result of a kick action. So the ball position after the kick is determined by drawing one sample from the distribution of the selected kick action.

In Figure 3 the ball positions after a kick are visualized as colored circles. Here we are interested in how much the robot needs to turn around the ball. Ideally the robot should do it as little as possible so opponent players can't take control of the ball before we execute a kick.

Figure 5 shows the evaluation of turning times and number of kicks from 10.000 experiments with random start positions on the field. The logarithmic scaled histogram in Figure 5 (top) shows how often the robot had to turn around the ball by a specific amount. The *fast* strategy turns the least, it never

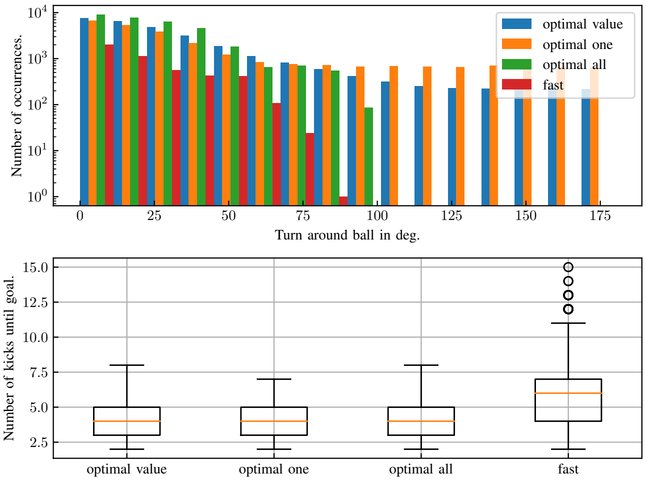

Fig. 5. The histogram was calculated for 10000 random start positions. Top: How often, how much the robot needed to turn around the ball. Bottom: Number of kicks until a goal was scored.

turns more than 90°. The *optimal one* and *optimal value* perform very similar and turn the most – up to 180°. The *optimal all* strategy is a compromise between less turning and optimizing according to the value function. It turns more than *fast*, but never more than 100°. Note, that the histogram in Figure 5 (top) does not contain turns by 0°.

Figure 5 (bottom) visualizes the number of kicks until a goal was scored or it was out for each strategy. All the optimal strategies perform significantly better than the *fast* strategy. This is not surprising since those strategies maximize the likelihood of scoring the goal while the *fast* strategy was designed to perform a kick immediately as long as the ball position would improve. The *optimal all* strategy minimizes both the number of kicks and the amount of turning around the ball and thus should be preferred over the other strategies.

VI. CONCLUSIONS

We presented an action selection algorithm based on forward simulations. We discussed the algorithm in general terms and in its application in the scenario of the kick selection problem in RoboCup. The existing algorithm presented in [10] was extended by parametric actions. This analysis shows that optimizing the kicks lead to less time spending turning around the ball before a kick. This significantly reduces the risk of getting the ball stolen by opponent players before the kick is executed.

Furthermore the stability of decisions depending on the number of predictions was analyzed. We found that using 30 samples to estimate the result of one kick is good trade-off between computation time and accuracy.

Our current effort focuses in particular on stepwise extension to simulating the ball approach and more dynamic evaluation. For instance, the potential field might reflect the influence regions of the own teammates based on their position, which would favor the kicks towards these regions and enable emergent passing.

At the present state the implemented method is limited to the selection of the kicks only. We believe that the true potential of the forward simulation can only unfold if extended to all areas of decision making like role decision, passing, positioning etc.

VIDEO

The following video illustrates the simulation based decision algorithm in a real RoboCup game. https://www2.informatik.hu-berlin.de/~naoth/media/video/hrs17-action-simulation.mp4

REFERENCES

[1] A. Pico, G. Schillaci, V. V. Hafner, and B. Lara, “How do i sound like? forward models for robot ego-noise prediction,” in *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Sept 2016, pp. 246–251.

[2] A. Bordallo, F. Previtali, N. Nardelli, and S. Ramamoorthy, “Counterfactual reasoning about intent for interactive navigation in dynamic environments,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 2943–2950.

[3] L. Kunze and M. Beetz, “Envisioning the qualitative effects of robot manipulation actions using simulation-based projections,” *Artificial Intelligence*, vol. 247, no. Supplement C, pp. 352 – 380, 2017, special Issue on AI and Robotics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370214001544

[4] C. Eppner, R. Martín-Martín, and O. Brock, “Physics-based selection of actions that maximize motion for interactive perception,” in *RSS workshop: Revisiting Contact - Turning a problem into a solution*, 2017. [Online]. Available: http://www.robotics.tu-berlin.de/fileadmin/fg170/Publikationen_pdf/EppnerMartinMartin17_RSS_WS.pdf

[5] T. Weber, S. Racanière, D. P. Reichert, L. Buesing, A. Guez, D. Jimenez Rezende, A. Puigdomènech Badia, O. Vinyals, N. Heess, Y. Li, R. Pascanu, P. Battaglia, D. Silver, and D. Wierstra, “Imagination-Augmented Agents for Deep Reinforcement Learning,” *ArXiv e-prints*, Jul. 2017.

[6] R. Dodds, P. Vallejos, and J. Ruiz-del Solar, “Probabilistic kick selection in robot soccer,” in *Robotics Symposium, 2006. LARS ’06. IEEE 3rd Latin American*, Oct 2006, pp. 137–140.

[7] P. Guerrero, J. Ruiz-del Solar, and G. Díaz, *Probabilistic Decision Making in Robot Soccer*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 29–40. [Online]. Available: https://doi.org/10.1007/978-3-540-68847-1_3

[8] M. Ahmadi and P. Stone, *RoboCup 2007: Robot Soccer World Cup XI*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ch. Instance-Based Action Models for Fast Action Planning, pp. 1–16. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-68847-1_1

[9] M. Nieuwenhuisen, R. Steffens, and S. Behnke, “Local multiresolution path planning in soccer games based on projected intentions,” *RoboCup 2011: Robot Soccer World Cup XV*, pp. 495–506, 2012.

[10] H. Mellmann, B. Schlotter, and C. Blum, “Simulation based selection of actions for a humanoid soccer-robot,” in *RoboCup 2016: Robot Soccer World Cup XX*, 2016, to appear. [Online]. Available: http://www.ais.uni-bonn.de/robocup.de/2016/papers/RoboCup_Symposium_2016_Mellmann.pdf

[11] A. F. T. Winfield, C. Blum, and W. Liu, *Advances in Autonomous Robotics Systems: 15th Annual Conference, TAROS 2014, Birmingham, UK, September 1-3, 2014. Proceedings*. Cham: Springer International Publishing, 2014, ch. Towards an Ethical Robot: Internal Models, Consequences and Ethical Action Selection, pp. 85–96. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10401-0_8