Robotic Billiards: Understanding Humans in Order to Counter Them

Thomas Nierhoff, Konrad Leibrandt, Tamara Lorenz, and Sandra Hirche

Abstract—**Ongoing technological advances in the areas of computation, sensing, and mechatronics enable robotic-based systems to interact with humans in the real world. To succeed against a human in a competitive scenario, a robot must anticipate the human behavior and include it in its own planning framework. Then it can predict the next human move and counter it accordingly, thus not only achieving overall better performance but also systematically exploiting the opponent's weak spots. Pool is used as a representative scenario to derive a model-based planning and control framework where not only the physics of the environment but also a model of the opponent is considered. By representing the game of pool as a Markov decision process and incorporating a model of the human decision-making based on studies, an optimized policy is derived. This enables the robot to include the opponent's typical game style into its tactical considerations when planning a stroke. The results are validated in simulations and real-life experiments with an anthropomorphic robot playing pool against a human.**

Index Terms—**Cognitive robotics, decision making, dynamic programming, human-robot interaction, intelligent robots, maximum likelihood estimation.**

I. Introduction

As ROBOTIC technologies advance, robots are no longer only suitable for automated processes with predefined motion sequences but may accomplish tasks in dynamically changing environments and even in the interaction with humans. Seamless human–robot interaction is one of the most challenging topics in current robotic research with a tremendous application potential ranging from manufacturing via care to training/education. Interaction between humans and robots can be of different type, for example on a physical basis [1]–[3], using gestures, or social interaction [4]–[6]. The same accounts for the level of interaction [7], ranging from reactive control strategies [8], [9] to learning-based agents [10], enabling them to predict future actions of the interaction partner and to act accordingly. On an even higher level one is interested in the intention of a participating subject, going beyond the set of observable action toward figuring out the underlying action goals [11], either in collaborative scenarios [12]–[14] or in competitive scenarios [15], [16].

Competitive games are an appealing test scenario for human–robot interaction for various reasons. On the one hand, most people encounter them on a daily basis whenever doing some kind of competitive sports. On the other hand, they provide an ideal test bed [17]–[20] as game rules are well defined and evaluation is easy—if you win, you are better than your opponent. Due to its multifaceted aspects combining motor performance and planning capabilities, billiards (pool) is used as a representative scenario in this paper.

For mastering a competitive game its entity, different factors must be taken into account to master a competitive game. When facing a human opponent, being aware of his individual preferences and limitations can help to improve oneself's game play. Once the weak spots of an opponent are known, they can be countered to gain a tactical advantage. At the same time, a player must be aware of his own flaws to consider them properly when planning ahead. All existing pool planners, e.g., PoolFiz [21], rely on a model-based approach and use a pool simulator based on the underlying pool physics [22]–[25]. To evaluate their performance in a competitive setup, the international computer games association computational pool tournament was hold three times so far in 2005, 2006, and 2008. Most of the participating planners densely sample the most likely strokes, thus creating a search tree with a large branching factor in order to predict the outcome of every stroke [26], [27]. A different method is presented in [28] and [29] where a gradient-based optimization is used to pocket balls and to take position play into account. Besides, there exist various planners based on fuzzy logic [30], [31] and deeper theoretical considerations about the existence of Nash equilibria at pool [32]. Yet all those planners assume that both players act according to the same underlying optimality principles and do not account for individual preferences which can occur when facing a human with a pool robot [33]–[36]. In combination with the specific hardware limitations of robotic systems, the question thus becomes

Manuscript received May 18, 2015; accepted June 15, 2015. This work was supported in part by the European Research Council Starting Grant "Control Based on Human Models (con-humo)" under Grant 337654, and in part by the Deutscher Akademischer Austauschdienst (german academic exchange service) (www.daad.de). This paper was recommended by Associate Editor H. Qiao.

T. Nierhoff is with the Chair of Information-Oriented Control (ITR), Faculty of Electrical Engineering, Technische Universität München, Munich D-80333, Germany, and also with the Department of Mechano-Informatics, University of Tokyo, Tokyo 113-8656, Japan (e-mail: tn@tum.de).

K. Leibrandt is with the Hamlyn Centre for Robotic Surgery, Imperial College London, London SW7 2AZ, U.K. (e-mail: k.leibrandt12@imperial.ac.uk).

T. Lorenz is with the Chair of Information-Oriented Control (ITR), Faculty of Electrical Engineering, Technische Universität München, Munich D-80333, Germany, and also with the Experimental Psychology Unit, Ludwig-Maximilians University Munich, Munich 80539, Germany (e-mail: tlorenz@tum.de).

S. Hirche is with the Chair of Information-Oriented Control (ITR), Faculty of Electrical Engineering, Technische Universität München, Munich D-80333, Germany (e-mail: hirche@tum.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2457404

whether a robot is capable of overcoming its physical drawbacks through extensive planning and proper incorporation of the opponents preferences.

The contribution of this paper is the consideration of the human decision model to improve the success rate of a robotic system at a competitive game. The presented approach based on a Markov decision process (MDP) is able to take both hardware limitations and a detailed model of the human decision-making process into account. Thus the robot can predict the human's actions and adapt to them by deriving an optimized policy. Due to the competitive nature of the game, the approach resembles a expectiminimax tree. Experiments are both simulation-based with thousands of games evaluated and hardware-based using a real-anthropomorphic robot playing pool against a human. They show how the robot is able to improve its success rate significantly by planning ahead while incorporating the human model. To the best of our knowledge, this is the first paper explicitly improving the robot's game play at pool by considering a human-specific decision-making model.

The remainder of this paper is organized as follows. Section II presents both a detailed problem description and the general planning framework. In Sections III and IV, it is described how hardware limitations and the human model are included in the planner. Experimental results are presented in Section VI. Finally, Section VII concludes with a summary and possible expansions for the future.

II. Problem Statement and Conceptual Approach

A. Problem Statement

Following the definition in [37], "eight-ball" billiards (from now on called "pool") can be seen as an competitive interaction scenario where a robot faces a human in a turn-based manner. In order to win the game, every player has to pocket all balls of his own color and finally the black ball before the other player. At each turn, the player in charge can pocket balls by executing a stroke with the cue. A model-based approach is pursued in this paper to select an optimal stroke by predicting the outcome of the next few turns. If it is the robot's turn, the challenge is to find an optimized stroke (action) A_r from the action set $\mathcal{A}_r$ also incorporating the pool state $S \in \mathcal{S}$ and possible actions $A_h \in \mathcal{A}_h$ of the human. Using the depth variable t accounting for the number of actions to plan ahead, the problem is reformulated as follows:

$$A_{0,r} = \operatorname*{argmax}_{A_{t,r}} Q\big(S_t, A_{t,\{r,h\}}\big), \quad t = 0, 1, \dots, n \tag{1}$$

with the yet unknown cost function Q encoding the quality of the next n actions. Through proper choice of Q, maximizing the term in (1) results in an optimal tactic to win the game. Note that we only optimize over $A_{t,r}$, i.e., the actual and future action of the robot, because the strokes $A_{t,h}$ of the human cannot be directly influenced.

Differing from simulations, the robot must consider three important aspects in real life. The first aspect, an unknown human decision-making, makes it difficult to predict the executed human stroke $A_{t,h}$ for a given situation S_t. Yet this knowledge is necessary as the optimal stroke of the robot $A_{0,r}$ also depends on the strokes of the human according to (1). Hence, studies must be conducted to derive an approximate model of the human decision-making to evaluate (1) precisely. The second aspect, a limited motor skill of robot and human, refers to the deviation between the planned stroke $A^*_{\{r,h\}}$ and the executed stroke $A_{\{r,h\}}$. Caused by limits in perception, uncertainties in motor control, and imperfect internal models of both players, the executed stroke is in general not similar to the planned stroke. This turns the deterministic game in simulation into a probabilistic game in real-world situations. When representing both factors well enough with a probability distribution f_n, it is modeled as follows:

$$A_{\{r,h\}} = f_n\Big(A^*_{\{r,h\}}\Big). \tag{2}$$

The last aspect, robot kinematic constraints, limits the space of actions $\mathcal{A}_r$ of the robot as follows:

$$\mathcal{A}_r \subset \mathcal{A}. \tag{3}$$

As the human kinematic constraints are negligible, it is assumed that $\mathcal{A}_h = \mathcal{A}$.

All three aspects must be quantified and represented by precise models for incorporating them in (1). Focusing in this paper on the human behavior, it is investigated how well the human decision-making can be modeled and to what extent a good model improves the win rate of the robot.

B. General Framework

This section develops a mathematical framework using an MDP to model the game of pool in its entity and consider all aspects listed in Section II-A. Pool can be represented as a sequential stochastic game with a continuous action space and a continuous state space. It is sequential due to the turn-based stroke execution switching between both players and stochastic due to the nondeterministic outcome of every stroke caused by the limited skill of each player. Differing from games with a finite set of states such as chess or backgammon, there is an infinite number of possible combinations for the balls to be placed on the table. The same holds for the actions. When discretizing the state and action space, the game of pool is modeled as a MAXPROB MDP (MAXPROB MDP) (see [38]). It is based upon a standard MDP defined by tuples of the form $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{G}, s_0 \rangle$ where $S \in \mathcal{S}$ is the set of states (position of all balls on the table plus current game situation determined by the rules), $A \in \mathcal{A}$ is the set of actions (strokes), $T \in \mathcal{T}$ is a transition function $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0; 1]$ denoting the probability of moving from state S_i to S_j by executing the action A, $R \in \mathcal{R}$ is a mapping $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ specifying action rewards, and S_0 is the start state.

For an MDP, the general goal is to find an optimal policy π^* maximizing the expected reward when traversing the states of the MDP. By introducing the discount factor $\gamma \in [0; 1]$ and the value function $V : \mathcal{S} \to \mathbb{R}$, the value function $V^\pi(s)$ denoting the expected reward for the policy π starting in state S is defined as follows:

$$V^\pi(S) = \mathbb{E}^\pi\left[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t)\right]\Bigg|_{S_0=S}. \tag{4}$$

In addition, the action-value function $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ in (1) is formalized as follows:

$$Q^{\pi}(S, A) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R(S_t, A_t)\right]\Bigg|_{S_0=S, A_0=A}. \tag{5}$$

The optimal policy π^* is the policy that maximizes the value function V^{π} for all possible policies π as follows:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, V^{\pi} \tag{6}$$

resulting in the optimal value function $V^*(S)$ and action-value function $Q^*(S, A)$. Both value function and action-value function are closely related, as such the optimal value function $V^*(S)$ is obtained by finding the action A that maximizes the optimal action-value function $Q^*(S, A)$ as follows:

$$V^*(S) = \max_{A} Q^*(S, A). \tag{7}$$

Note that the used formulation is consistent with the existing literature on the topic (see [39]).

MAXPROB MDPs are a special class of MDPs with the goal of maximizing the probability of reaching a goal state; that is the probability of winning a game. They are based upon stochastic shortest path MDP (SSP MDP) introducing a set of absorbing goal states $G \in \mathcal{G}$. Such a goal state is defined by $\mathcal{T}(G, A, G) = 1, \forall A \in \mathcal{A}$ and $\mathcal{R}(G, A) = 0, \forall A \in \mathcal{A}$, allowing only self-transitions in G and accumulating no reward. The only two conditions for SSP MDPs are that for every state S at least one proper policy exists that reaches a goal state with probability $P = 1$. Additionally, every improper policy has a reward of $-\infty$. For pool, SSP MDPs are still not applicable as they only feature goal states (robot wins) but no dead end states (human wins). In case of dead end states, it is not guaranteed anymore that a goal state can be reached from any other state with probability $P = 1$. MAXPROB MDPs on the other hand have a similar structure as SSP MDPs but explicitly account for dead ends by judging the quality of every policy not based on the expected reward but based on the probability of eventually reaching a goal state. By assigning a reward $R = 1$ to every action that reaches a goal state and $R = 0$ to all other actions, the resulting optimal value function $V^*(S)$ reflects the probability of reaching a goal state (e.g., winning the game) when starting in state S.

In theory, an optimal policy can be obtained then for an arbitrary start state S_0 using heuristic search algorithms [40], [41]. Unfortunately, due to the large state space, those algorithms cannot be used for pool. Instead one has to use approximation algorithms resulting in a near-optimal value function $\hat{V}^{\pi} \leq V^{\pi^*}$ by evaluating only the most promising subset of state-action pairs. In general, those algorithms turn the probabilistic MDP into a deterministic planning problem resulting in a search tree with S_0 being the root and solve it using a deterministic planner [42].

Having obtained a policy, an individual action-value function can be assigned to each stroke specifying the expected reward. Then a player simply has to execute the stroke with the highest expected reward. A method to find an action-value function that also considers the human motor skill and decision-making is developed in the following sections.

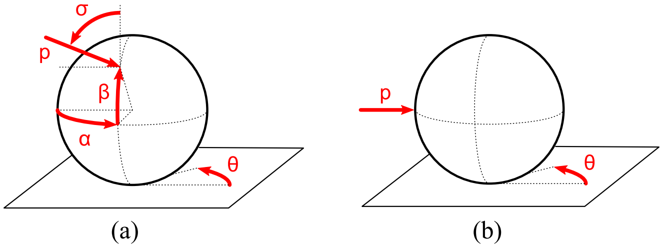


Fig. 1. Stroke parameter overview. (a) All possible stroke parameters. (b) Considered stroke parameters in this paper.

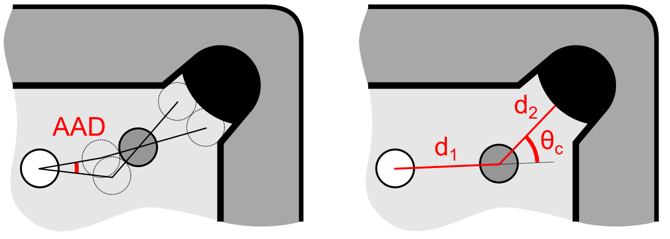
III. Modeling Human/Robot Motor Skill and Robot Kinematic Constraints

As explained in Section II-A, the limited motor skills of both players let the executed stroke differ from the desired or planned stroke, thus requiring a measure to describe the percentage of success of a desired stroke. This is described by the pocket probability which incorporates the limited motor skill in the presented planning framework. As it is based upon the stroke difficulty measure, the latter is introduced first.

A. Measure of Stroke Difficulty

The expected reward in (4) is closely related to the stroke difficulty. When playing pool, the expected reward depends on the number of pocketed balls. Intuitively the more difficult it is to pocket a ball, the lower the reward is. When executing a stroke, the player must not hit the ball he wants to pocket (called "object ball") directly. Instead, he has to hit the white ball (called "cue ball") with the cue. From a mathematical perspective, a stroke can be described by five parameters α, β, γ, θ, and p [43] for a given 2-D cue ball position on a flat table Fig. 1. Stroke intensity p and stroke angle θ have to be considered even by novice players in order to pocket balls. On the other hand, the angles α, β, and σ are mainly varied by advanced players to induce spin on the cue ball for a better position play. A central assumption for the remainder of this paper is a well defined spin during the stroke. As such the cue ball is supposed to be hit centrally and with no side spin (e.g., $\alpha = 0$, $\beta = 0$, and $\sigma = \pi/2$). As a result, the parameter space can be reduced from 5-D to 2-D (θ and p).

Different measures are used in this paper to define the stroke difficulty. For direct shots where the cue ball hits the object ball without any other ball–ball or ball–cushion collisions, the stroke difficulty can be calculated using basic geometry. Assuming a negligible spin transfer during ball–ball collisions, the stroke intensity p and stroke angle θ being independent of each other, and the stroke intensity p being large enough to pocket a ball, the difficulty of a direct shot is solely specified by the range of values $[\theta_l; \theta_u]$ (see [44]). The value $\theta_m = 0.5(\theta_l + \theta_u)$ denotes the mean angle for which the object ball is pocketed in the middle of the pocket. Intuitively, the smaller the interval $[\theta_l; \theta_u]$ or allowed angular deviation (AAD) becomes the more difficult the stroke will be [see Fig. 2(a)]. The same technique can also be used for combination shots including more than one object ball.



(a) (b)

Fig. 2. Visualization of two methods to determine the difficulty of a stroke. (a) AAD stroke difficulty based upon the maximal angular deviation for which the object ball is just pocketed. (b) Expert stroke difficulty based upon the distance d_1 between cue and object balls, the distance d_2 between object ball and hole and cutting angle θ_c.

Another method to determine the stroke difficulty is based upon expert knowledge [45], [46]. It reveals that the human perception of stroke difficulty depends only on a few parameters: the distance d_1 between cue ball and object ball, the distance d_2 between object ball and pocket, and the cutting angle θ_c [see Fig. 2(b)]. Various approaches in this paper try to determine an expression for the difficulty of a stroke based on this knowledge [26], [30], one of the most prominent ones is [28]

$$\kappa = \frac{\cos(\theta_c)}{d_1 d_2}. \tag{8}$$

Analogously to the AAD stroke difficulty, a small value corresponds to a tricky stroke and a large value represents an easy one. The AAD stroke difficulty can be linked in a straightforward fashion to the pocket probability, implicitly considering the parameters θ_c, d_1, d_2, and partially blocked paths due to other balls on the table. The expert stroke difficulty on the other hand suggests that the human perception of difficulty deviates from the AAD stroke difficulty due to a different perspective, thus requiring a new difficulty measure when modeling the human decision-making.

Robot kinematic constraints can be incorporated using the AAD stroke difficulty. It is assumed that robot kinematic constraints limit the range of admissible θ-values to the interval $[\theta_l^k; \theta_u^k]$. Then the resulting interval of stroke angles that are both executable by the robot and pocket a ball is given by the intersection $[\theta_l; \theta_u] \cap [\theta_l^k; \theta_u^k]$. Fig. 3 illustrates the influence of the robot kinematic constraints. Shown is a simplified, yet true-to-scale collision model of the used pool robot and the pool table, approximated by bounding boxes. The robot is limited by two constraints. First, it has a stiff body (dark gray rectangle in Fig. 3) which collides with the table if the white ball is placed too far away from any cushion. Second, its right endeffector does not lie on the table but on the cushion, which causes problems if the white ball is placed too close to the cushion. The angle θ_r specifies the range of possible stroke angles for a given position of the white ball on the table. An angle $\theta_r = 2\pi$ means the white ball can be shot in any direction whereas an angle $\theta_r = 0$ stands for an unreachable white ball. Moreover, do gray circles mark the intervals $[\theta_l^k; \theta_u^k]$ of stroke angles for specific positions of the white ball.

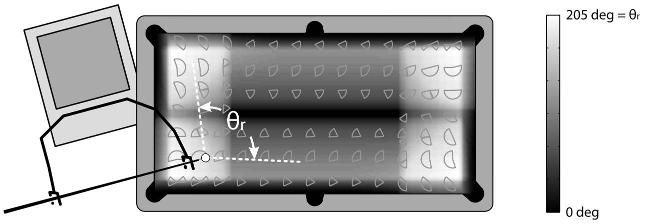


Fig. 3. Kinematic constraints of the robotic system. Due to collisions with the pool table, the robot cannot execute a stroke in an arbitrary direction for a given ball position. The brighter the area is, the larger the range of possible stroke directions is. Gray-circled segments mark the possible stroke direction for certain positions of the white ball.

B. Measure of Pocket Probability

This section derives the pocket probability $P_s \in [0; 1]$. It links the stroke difficulty presented in Section III-A to the individual motor skill of each player as described mathematically in (2).

It is assumed that the stroke precision of robot and human is represented by normal distributions for the stroke intensity p and stroke angle θ as $\mathcal{N}_{p,\{r,h\}}(\mu_p, \sigma^2_{p,\{r,h\}})$ and $\mathcal{N}_{\theta,\{r,h\}}(\mu_\theta, \sigma^2_{\theta,\{r,h\}})$. Here, μ_p and μ_θ encode the desired (planned) stroke A^*. The variances $\sigma^2_{p,\{r,h\}}$ and $\sigma^2_{\theta,\{r,h\}}$ represent the individual skill level of robot and human. Novice players are characterized by large variances whereas for professionals the variance is close to zero. Equation (2) is thus reformulated using the 2-D normal distribution $\mathcal{N}_{\{r,h\}}$ as follows:

$$A = f_n(A^*) = \mathcal{N}_{\{r,h\}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{9}$$

with $\boldsymbol{\mu} = [\mu_p, \mu_\theta]^T$, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma^2_{p,\{r,h\}}, \sigma^2_{\theta,\{r,h\}})$.

The limited motor skill of human and robot is encoded by the variances $\sigma^2_{p,\{r,h\}}, \sigma^2_{\theta,\{r,h\}}$ (see [44]). When defining the stroke difficulty using the AAD method, the stroke difficulty can be converted for every stroke into a pocket probability $P_s \in [0; 1]$. Similar to the stroke angle, we assume that a ball will be pocketed for stroke intensities within the interval $[p_l; p_u]$. For stroke intensities $p < p_l$, the ball is not fast enough to be pocketed and values $p > p_u$ cannot be achieved due to a limited player strength. With $f_{\{r,h\}}$ as the probability density function of $\mathcal{N}_{\{r,h\}}$, the pocket probability can be described by

$$P_s = \int_{p_l}^{p_u} \int_{\theta_l}^{\theta_u} f_{\{r,h\}} d\theta dp. \tag{10}$$

We further assume that $\sigma_p \ll p_u - p_l$, $p_l \ll \mu_p \ll p_u$, i.e., the pocket probability is mostly unaffected by the choice of μ_p and depends primarily on the stroke angle θ. Then (10) is approximated with $f_{\theta,\{r,h\}}$ as the probability density function of $\mathcal{N}_{\theta,\{r,h\}}$ by

$$P_s \approx \int_{\theta_l}^{\theta_u} f_{\theta,\{r,h\}} d\theta. \tag{11}$$

The formulas also hold for combination shots but must be extended if two or more object balls shall be pocketed with a single stroke.

IV. Modeling Human Decision-Making

This section focuses on the human decision-making to include it in the general planning framework. Differing from a robot for which the assumptions in Section III hold and where the pocket probability can be calculated based upon the AAD stroke difficulty, the expert stroke difficulty indicates that humans select their stroke based on different criteria. When modeling human decision-making, three subproblems are identified.

1) *Compute the Human Stroke Difficulty κ_h:* How difficult a human thinks it is to pocket a ball.
2) *Compute the Human Pocket Probability P_h:* How difficult it is for a human to pocket a ball.
3) *Compute the Human Discount Factor γ_h:* How far a human is planning ahead.

Differing from a rationally acting robot who will pocket the ball with the highest action-value function, unknown human likings or limitations may favor a different stroke. As it is also unknown how these preferences affect the pocket probability, one has to specify not only the human stroke difficulty κ_h but also the human pocket probability $P_h(\kappa_h)$ as a function of the human stroke difficulty (see [47]).

1) Human Stroke Difficulty: In this section, an algebraic expression of the human stroke difficulty κ_h based on two psychological experiments is derived. Similar to the stroke difficulty in Section III-A, it measures the perceived human difficulty of each stroke and allows to rank them accordingly. In the first experiment, the most influential factors on the human stroke difficulty are figured out. The result is then used in combination with the second experiment to determine a precise model of the human stroke difficulty.

For the first experiment, 25 participants (15 males, 21–31 years, Ø25 years) are shown 24 pool scenarios on a real-pool table representing a wide variety of possible strokes and AAD values. In every scenario, two playable object balls are placed on the table. For 18 scenarios, the AAD stroke difficulty is similar for the two object balls whereas the three parameters d_1, d_2, and θ_c are varied. For the other six scenarios, one of the two object balls is easier to pocket according to the AAD stroke difficulty, but either more difficult to reach (the player has to lean more over the pool table) or partially blocked by a third ball, reducing the interval of admissible stroke angles $[\theta_l; \theta_u]$. Each participant has to judge which ball seems easier to pocket and is asked to specify freely one or more reasons for this decision which are categorized afterwards. The most frequently stated reasons with their number of occurrence for the first 18 scenarios are shown in Table I. The answers show that the variables d_1, d_2, and θ_c are the most influential factors. The effect of a cushion close to cue ball or object ball is indefinite as it helps some players at aiming whereas others feel disturbed. For the six remaining scenarios, in 70% the easier stroke according to the AAD stroke difficulty is selected even if it is more tricky to reach. In 60%, the easier stroke is selected even if it is partially blocked by another ball. Similar to the answers regarding the advantage of a cushion nearby, some people state that the ball helps at aiming; however, other people feel distracted by it.

TABLE I
Influence of Different Factors on the Human Stroke Difficulty

occurrence	reason
264×	A smaller cutting angle θ_c
75×	A smaller distance d_2
31×	A smaller distance d_1
29×	The cushion nearby helps at aiming
24×	The cushion nearby disturbs at aiming
12×	Intuition
9×	Anatomic reason
8×	A smaller distance $d_1 + d_2$
17×	Very specific reason
$\sum$ 469×	

For the second experiment, 12 images are created showing pool scenarios of different AAD stroke difficulty with one cue ball and one object ball. To compare every possible pair of images, six sets of six images were shown to 23 participants (13 males, 18–51 years, Ø25 years). The participants had to arrange every individual set of images according to their perceived difficulty. Using these sequences, both a relative ranking $\alpha_k,\ \alpha, k \in \{1, 2, \dots, 12\}$ and an absolute ranking $\beta_k,\ k \in 1, 2, \dots, 12,\ \beta \in [1, 12]$ based on the mean ranking are obtained for every single scenario.

In order to map the result of both experiments to a human stroke difficulty, five representative measures describing the human stroke difficulty κ_h are considered. The first two measures are based upon additive linear models to account for the increasing stroke complexity for high values of d_1, d_2, and θ_c. The next two models include the term $\cos(\theta_c)$ to account for the increasing difficulty at high cutting angles. The last term is an extension of the existing measure (8). They are

$$\begin{aligned}
\kappa_1 &= a_1 d_1^{c_1(d_1)} + a_2 d_2^{c_2(d_2)} + a_3 \theta_c^{c_3(\theta_c)}\\
\kappa_2 &= a_1 d_1^{c_1(d_1)} + a_2 d_2^{c_2(d_2)} + a_3 \theta_c^{c_3(\theta_c)} + \kappa_p\\
\kappa_3 &= a_1 d_1^{c_1(d_1)} + a_2 d_2^{c_2(d_2)} + a_3 \cos(\theta_c)^{c_3(\theta_c)}\\
\kappa_4 &= a_1 d_1^{c_1(d_1)} + a_2 d_2^{c_2(d_2)} + a_3 \cos(\theta_c)^{c_3(\theta_c)} + \kappa_p\\
\kappa_5 &= \frac{\cos(\theta_c)^{c_3(\theta_c)}}{d_1^{c_1(d_1)} d_2^{c_2(d_2)}}
\end{aligned} \tag{12}$$

with $\mathbf{a} = [a_1, a_2, a_3]$ as constant coefficients and $\mathbf{c} = [c_1, c_2, c_3]$ with c_1 and c_2 as constant coefficients and c_3 as polynomial of degree one. A two-staged optimization is used to find optimized values for $\mathbf{a}$ and $\mathbf{c}$. The cost function Γ_1 of the first stage consists of two terms related to the first and second psychological experiments. Minimizing the first term corresponds to matching the predicted decision $\hat{\Delta}_j(\mathbf{a}, \mathbf{c})$ to the dominant decision Δ_j of participants in the first experiments. The factor δ_j acts as a penalty term if the predicted decision does not match the decision from the first experiment. By minimizing the second term, one matches the predicted relative ranking $\hat{\alpha}_k(\mathbf{a}, \mathbf{c})$ to the relative rank α_k of each scenario

TABLE II
OPTIMIZED PARAMETERS WHEN APPROXIMATING THE HUMAN STROKE DIFFICULTY

measure	a_1	a_2	a_3	c_1	c_2	c_3	Γ_2
κ_1	4.6	7.1	1.2	0.10	0.10	$4.5-4.7\theta_c$	0.02
κ_2	1.3	6.5	2.0	1.0	0.21	$3.8-4.9\theta_c$	0.008
κ_3	-0.53	1.3	-4.8	-1.1	4.6	$0.24+2.2\theta_c$	0.06
κ_4	0.81	-1.4	-4.8	1.9	-0.75	$2.8+1.9\theta_c$	0.08
κ_5	-	-	-	0.33	0.38	$4.1-2.7\theta_c$	0.002

in the second experiment

$$\Gamma_1 = \sum_{j=1}^{18} \delta_j(\mathbf{a},\mathbf{c})^2 + \sum_{k=1}^{12} \left[\hat{\alpha}_k(\mathbf{a},\mathbf{c}) - \alpha_k\right]^2$$

$$\text{where } \delta_j = \begin{cases} 2 & \text{if } \hat{\Delta}_j(\mathbf{a},\mathbf{c}) \neq \Delta_j \\ 0 & \text{else.} \end{cases} \tag{13}$$

The cost function Γ_2 of the second stage is a measure how much the predicted absolute ranking $\hat{\kappa}_k(\mathbf{a},\mathbf{c})$ of each scenario in the second experiment differs from the absolute ranking β_k. To suppress outliers, the 25% trimmed mean $\beta_{k,25}$ is used instead of β_k. By minimizing the function Γ_2, a measure of the resulting stroke difficulty is obtained

$$\Gamma_2 = \sum_{k=1}^{12} \left[\hat{\kappa}_k(\mathbf{a},\mathbf{c}) - \beta_{k,25}\right]^2. \tag{14}$$

Final results are displayed in Table II. Because the function κ_5 has the lowest cost Γ_2, it is the best approximation to the real-human stroke difficulty. Thus the human stroke difficulty κ_h is determined as follows:

$$\kappa_h = \frac{\cos\theta_c{}^{4.1-2.7\theta_c}}{d_1^{0.33} d_2^{0.38}}. \tag{15}$$

2) Human Pocket Probability: This section develops a model of the human pocket probability $P_h \in [0; 1]$ to measure the pocket probability of the human player depending on the human stroke difficulty as follows:

$$P_h = f_p(\kappa_h) \tag{16}$$

thus linking the two measures in a similar way to the AAD stroke difficulty and the pocket probability. To model P_h precisely over a wide variety of (θ_c, d_1, d_2)-triplets, a Monte Carlo approach is used that evaluates random strokes occurring during a series of conducted pool games.

In total, four participants (3 males, 24–36 years, Ø29 years) took part in the third experiment. Two players are intermediate amateurs who frequently play a few pool games per week, and two are novice players with hardly any experience. Participants were grouped into dyads of equal playing skill. Their task was to play multiple games while sticking to direct strokes and hitting the cue ball centrally. The stroke results are split into 11 intervals depending on their stroke difficulty. For comparing the influence of AAD stroke difficulty, expert stroke difficulty, and human stroke difficulty on the human pocket probability, all measures are normalized to the [0; 1] interval. For every interval, the mean pocket probability and mean stroke difficulty are calculated. Pearson's correlation coefficient between stroke difficulty and pocket probability for the three stroke difficulty measures is shown in Table III. For small values of d_1 and d_2, both human stroke difficulty and expert stroke difficulty become very large and cannot be represented properly by a linear model. Thus the correlation coefficient has been recalculated as shown in Table IV, neglecting the interval with the highest stroke difficulty values. The linear relation when neglecting very easy strokes shows that humans act indeed rationally when selecting the stroke with the lowest human stroke difficulty as it is the one with the highest chance to pocket a ball.

TABLE III
CORRELATION COEFFICIENT BETWEEN STROKE DIFFICULTY AND POCKET PROBABILITY WHEN CONSIDERING ALL STROKES

player	evaluated strokes	correlation coefficient		
		κ	AAD	κ_h
one	216	0.46	0.61	0.89
two	182	0.65	0.72	0.86
three	164	0.54	0.58	0.88
four	152	0.76	0.68	0.92

TABLE IV
CORRELATION COEFFICIENTS BETWEEN STROKE DIFFICULTY AND POCKET PROBABILITY WHEN NEGLECTING VERY EASY STROKES

player	evaluated strokes	correlation coefficient		
		κ	AAD	κ_h
one	169	0.79	0.85	0.94
two	152	0.87	0.77	0.91
three	134	0.64	0.52	0.88
four	121	0.89	0.82	0.91

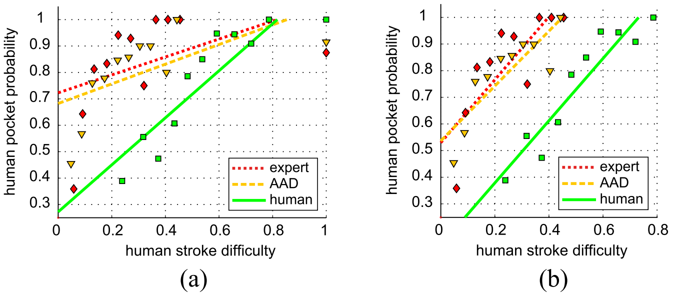


Fig. 4. Plotted results of the normalized stroke difficulty versus pocket probability for player one with corresponding regression lines. (a) All strokes considered. (b) Without easy strokes.

Fig. 4 displays the obtained results for player one, showing the intervals and the computed linear regression function for the mapping from human stroke difficulty to human pocket probability. The three different relations correspond to the expert stroke difficulty, the AAD stroke difficulty, and the human model developed in this section. When using the expert stroke difficulty or AAD stroke difficulty, a nonlinear relation between pocket probability and stroke difficulty is observable for small stroke difficulty values. This results in lower correlation coefficients for κ, AAD compared to κ_h as calculated in Tables III and IV. Based on the results, the relation between P_h and κ_h can be determined individually for each player.

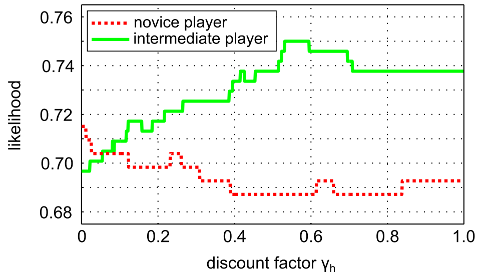


Fig. 5. MLE of the human discount factor γ_h for a novice player and an intermediate player. Only the maximum of each plot is of interest. A discount factor $\gamma_h = 0$ stands for a player not planning at all whereas a discount factor $\gamma_h = 1$ accounts for good planning capabilities.

For player one, it becomes

$$P_h = f_p(\kappa_h) = \begin{cases} 0.36\kappa_h + 0.27 & \forall\, \kappa_h \in [0; 2.02] \\ 1 & \forall\, \kappa_h \in [2.02; \infty]. \end{cases} \tag{17}$$

3) Human Discount Factor: So far only the humans' skill is discussed, specifying how well they can pocket balls. Another aspect are planning capabilities, defining how much they think about future states when planning their next stroke. Similar to (4), it is modeled by a discounted reward factor γ_h. Whereas a small γ_h value implies that the player focuses more on the current situation, a large discount factor γ_h indicates that he also takes future table states into account. The data from the third experiment can be used to compute an individual discount factor γ_h for every participant by a maximum likelihood estimation (MLE) maximizing the percentage of correctly predicted stroke decisions Ψ for every player over the parameter γ_h as follows:

$$\hat{\gamma}_h = \underset{\gamma_h}{\operatorname{argmax}}\, \mathcal{L}(\gamma_h|\Psi). \tag{18}$$

Results of the MLE for two dyads for a planning depth of 1 are displayed in Fig. 5 and [47]. While the intermediate player attested to think about the outcome of a stroke ($\gamma_h > 0$), the novice player confirmed that he solely focuses on the easiest stroke ($\gamma_h = 0$). This corresponds to the displayed results, with $\hat{\gamma}_h \approx 0.55$ for the intermediate player and $\hat{\gamma}_h \approx 0$ for the novice player.

V. Pool-Specific Adaption

This section describes how the general framework presented in Section II-B is modified and combined with the stroke difficulty and pocket probability measures in Sections IV and IV to fit pool-specific needs. The two main challenges that must be overcome are a high computational complexity when solving the MAXPROB MDP and the derivation of an optimized policy.

Solving the MAXPROB MDP involves the creation of a search tree. Expanding the search tree means to run a pool simulation that predicts the outcome of a stroke for a given state on the table. Yet running a pool simulation is a time-costly operation and as such an approximate scheme must be used that limits the number of pool simulations. A first step is to simulate only the next n actions, resulting in an approximate value function $\hat{V}^\pi$ as follows:

$$\hat{V}^\pi(S) = \mathbb{E}\left[\sum_{t=0}^{n} \gamma^t R\big(S_t, A_{t,\{r,h\}}\big)\right]\Bigg|_{S_0=S} \tag{19}$$

and approximate action-value function

$$\hat{Q}^\pi(S, A) = \mathbb{E}^\pi\left[\sum_{t=0}^{n} \gamma^t R(S_t, A_{t,\{r,h\}})\right]\Bigg|_{S_0=S, A_0=A}. \tag{20}$$

Yet this causes other problems as for the MAXPROB MDPs, a reward $R \neq 0$ is only assigned whenever a goal state is reached, i.e., a game is won. Thus in theory every branch of the tree must be expanded until reaching a goal state or a known dead end to evaluate its outcome, which, however, may take more than n actions. This is overcome by introducing an reward R_η as follows:

$$R_\eta = \begin{cases} +1 & \text{for every pocketed ball of the robot} \\ -1 & \text{for every pocketed ball of the human} \end{cases} \tag{21}$$

which helps to estimate the outcome of an action whenever the first ball gets pocketed. This serves as a bias of the planner toward strokes for which most balls of the own color and least balls of the opponent's color are pocketed. In addition, the original condition of a reward $R = 1$ when winning a game is relaxed to

$$R_\upsilon = \begin{cases} +w & \text{if robot wins} \\ -w & \text{if human wins} \end{cases} \tag{22}$$

with constant w. The combined reward R is defined as follows:

$$R = R_\eta + R_\upsilon. \tag{23}$$

The constant w weights the importance of pocketing a ball with respect to a win/loss of the game. A high w-value results in an aggressive game style if the planner sees a chance to win the game and a defensive game style if there is a chance for the opponent to win the game. The chosen reward R does have a physical meaning as for $w = 0$ and $\gamma = 1$ the value function $\hat{V}^\pi(S)$ represents the expected number of balls that the robot will pocket more than the human over the next n strokes following the policy π. Similarly, the action-value function $\hat{Q}^\pi(S, A)$ denotes the difference of pocketed balls for a given policy when executing the stroke A. During tree creation, pruning is used to reduce computational complexity further: only the most promising strokes are simulated, i.e., the ones that aim at pocketing a ball of the own color with a direct shot. The combination of all three methods leads to a considerable reduction of computational complexity as it is sufficient to simulate only a subset of all possible strokes over the next n turns.

Once a search tree is created, an optimized policy must be derived for computing the next stroke. Backward induction is used to iteratively evaluate the value function $\hat{V}^\pi(S_t)$ backwards in time to find $\hat{V}^\pi(S_{t-1})$ and the optimized stroke parameters for each stroke. It is started by calculating the action-value function for every leaf node of the search tree of depth n. This is the reward R weighted with its pocket probability as follows:

$$\hat{Q}^{\pi}(S_n, A_n) = \begin{cases} P_{s,i}R_n & \text{if robot's turn} \\ P_{h,i}R_n & \text{if human's turn.} \end{cases} \tag{24}$$

In (24), the variable $P_{s,i}$ stands for the pocket probability of the ith ball–pocket combination according to (11) using the AAD stroke difficulty and $P_{h,i}$ for the human pocket probability according to (16). Based on $\hat{Q}^{\pi}(S_n, A_n)$, the value function $\hat{V}^{\pi}(S_n)$ is calculated as follows:

$$\hat{V}^{\pi}(S_n) = \begin{cases} \max\limits_{i} P_{s,i}R_n & \text{if } r\text{'s turn} \\ \min\limits_{i} P_{h,i}R_n & \text{if } h\text{'s turn.} \end{cases} \tag{25}$$

The value function is split up into two cases, accounting for the competitive nature of the game. As such the robot tries to maximize the value function, whereas the human tires to minimize it. During backward induction, the action-value function $\hat{Q}^{\pi}(S_{t-1}, A_{t-1})$ is calculated in an intermediate step to derive $\hat{V}^{\pi}(S_{t-1})$. Because the outcome of each stroke is nondeterministic due to the limited motor skill of each player, it has to be weighted with the probability density function $f_{\{r,h\}}$ [see (10)]. This results in

$$\begin{aligned} &\hat{Q}^{\pi}(S_{t-1}, A_{t-1}) \\ &\quad = \mathbb{E}^{\pi}\Big[R_{t-1} + \gamma \hat{V}^{\pi}(S_t)\Big] \\ &\quad = \begin{cases} \int\int\Big(R_{t-1} + \gamma_r \hat{V}^{\pi}(S_t)\Big) f_r dp\, d\theta & \text{if } r\text{'s turn} \\ \int\int\Big(R_{t-1} + \gamma_h \hat{V}^{\pi}(S_t)\Big) f_h dp\, d\theta & \text{if } h\text{'s turn} \end{cases} \end{aligned} \tag{26}$$

with γ_r as the discount factor of the robot and γ_h as the human discount factor according to (18). Note that the integrals in (26) can only be approximated due to a discrete search tree. The value function $\hat{V}^{\pi}(S_{t-1})$ is then calculated as follows:

$$\begin{aligned} \hat{V}^{\pi}(S_{t-1}) &= \underset{A_{t-1}}{\text{optimize}}\, \hat{Q}^{\pi}(S_{t-1}, A_{t-1}) \\ &= \begin{cases} \max\limits_{\mu_p, \mu_\theta} \hat{Q}^{\pi}(S_t, A_t) & \text{if } r\text{'s turn} \\ \min\limits_{\mu_p, \mu_\theta} \hat{Q}^{\pi}(S_t, A_t) & \text{if } h\text{'s turn.} \end{cases} \end{aligned} \tag{27}$$

Having obtained $\hat{V}^{\pi}(S_0)$, the optimized stroke parameters μ_p, μ_θ for the actual stroke can be directly derived as they are the ones optimizing $\hat{V}^{\pi}(S_0)$. The framework can be seen as an expectiminimax tree [48] with an additional optimization step for each node to calculate the optimal reward.

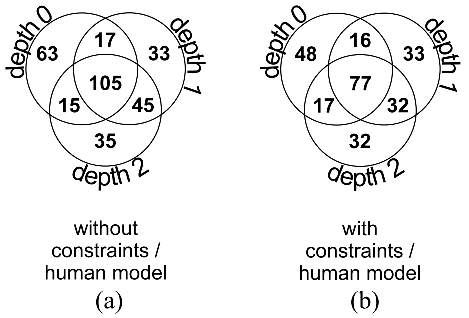
VI. Evaluation

The evaluation of the planning algorithm is both based upon simulations and by using an anthropomorphic robot facing a human opponent on a real-pool table. Simulations allow for a fast evaluation of a large number of strokes to measure the individual influence of various aspects. In contrast, the real-world scenario demonstrates the validity of the entire framework. For both experiments, the pool simulator described in [44] is used.

The individual parameters of robot and human in Table V are determined through hundreds of analyzed strokes recorded by a ceiling camera, indicating that the human opponent outperforms the robot with respect to motor skill (smaller σ_p, σ_θ) but is inferior to the robot concerning planning capabilities (smaller γ).

TABLE V
Parameters for the Experimental Evaluation

value	human	robot
$\sigma_p\,[Ns]$	0.01	0.03
$\sigma_\theta\,[rad]$	0.007	0.012
$\gamma\,[-]$	0.5	1



without constraints / human model
(a)

with constraints / human model
(b)

Fig. 6. Effect of a varied planning depth based upon 200 randomly selected table states using Venn diagrams. (a) Ignoring and (b) considering robot kinematic constraints and the human model.

A. Simulations

Simulations make it possible to estimate the individual influence of a variety of factors without the need to perform time-consuming experiments. For all simulations, a pool simulator programmed in C++\Qt is used [44], whereas the planning framework according to Section V is implemented in MATLAB R2013b.

A first simulation aims at displaying the influence of a varied search depth n [see (24)]. As discussed in [27], the effectiveness of a search depth of 2 is still an unsolved question. Differing from deterministic board games such as chess or backgammon where the search depth is among the most influencing factors, the probabilistic nature of pool makes it more difficult to answer the question. Archibald *et al.* [27] found out that both a search depth of 2 and a more densely sampled action space of depth 1 have a similar influence on the percentage to win. In our simulation, a total of 200 strokes for random pool states is processed with a search depth of 0–2. The first simulation scenario is similar to the experiment in [27] as it neither includes robot kinematic constraints nor the human model [we use (11) instead of (16) to model the human decision process]. In contrast, both human model and robot kinematic constraint are incorporated in the second scenario. Simulation results are shown in Venn diagrams in Fig. 6. As such for 17 strokes, the planning results for a depth of 0 and 1 are similar. Compared to [27], most of the strokes—105 for the first scenario and 77 for the second scenario—are similar for different search depths and a varied search depth influences the decision in just about 20% of all cases. This indicates that a search depth of 2 has only a

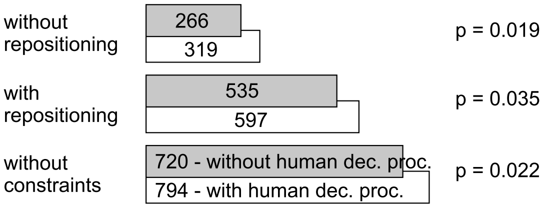


Fig. 7. Number of robot wins for different scenarios after 2400 games played in simulation. Right side: significance level of every comparison using a two-sample *t*-test.

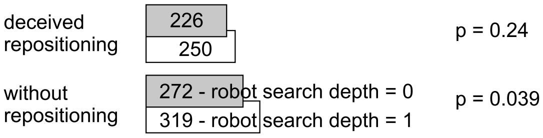


Fig. 8. Number of robot wins for different planning depths after 2400 games played in simulation. Right side: significance level of every comparison using a two-sample *t*-test.

minor influence on the decision process and can be probably overcome by a more densely sampled action space of depth 1.

The next simulation illustrates the effect of incorporating the human decision process in the planning framework. Three different runs are performed, depending on the way the robot kinematic constraints of Section III-A are treated. For the “without repositioning” run, whenever the robot is unable to reach a ball, it is considered as a foul. In the “with repositioning” run, the white ball is slightly moved such that the robot is able to execute a legal stroke whenever the robot is unable to reach a ball. Finally, in the “without constraints” run, there are no kinematic constraints, i.e., the robot is able to reach any position on the table. Every run consists of two scenarios evaluating the effect of the human decision process when planning ahead. For every scenario, 1200 games are evaluated twice, both with robot and human as starting player. When considering the human decision process, it is modeled according to Section IV. Otherwise, (11) is used instead of (16) to model the human pocket probability. The number of robot wins of every run is checked for significance using a two-sample *t*-test. Results are shown in Fig. 7. It is visible that for all runs the inclusion of the human decision process significantly improves the win rate.

The last simulation displays the necessity to include a precise model of the robot’s kinematic constraints as shown in Section III in the planner and the effect of a varied search depth of the robot. Similar to the previous simulation, 1200 games are evaluated twice. Two different runs are considered. For both runs, it is considered as a foul whenever the robot is unable to reach a ball during stroke execution. The difference is that no kinematic constraints are considered when planning ahead for the “deceived repositioning” run, whereas kinematic constraints are considered when planning ahead for the without repositioning run. On can see in Fig. 8 that a search depth of 1 significantly improves the robot’s win rate. Moreover, considering the robot’s kinematic constraints when planning ahead increases the win rate by around 20%–25%.

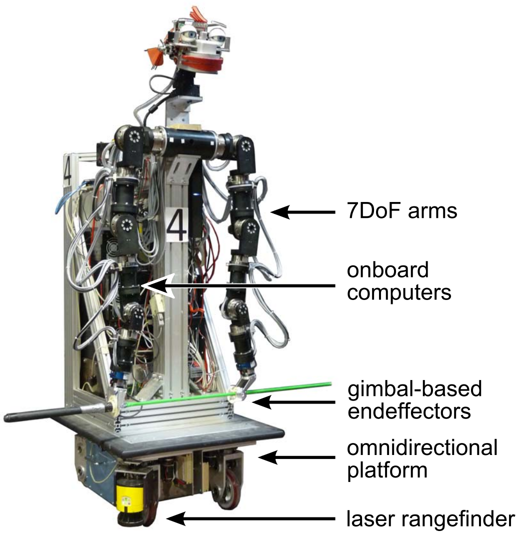


Fig. 9. Overview of the anthropomorphic robot.

B. Experimental Evaluation With Robotic Platform

The hardware setup for the experimental evaluation consists of a robot with a pair of 7-DoF anthropomorphic arms mounted on an omnidirectional platform (see Fig. 9). Special gimbal-based end effectors are designed to hold the cue properly. A Basler acA1300-30gm ceiling-mounted camera with a resolution of 1280 × 960 px and a frame rate of up to 40 Hz mounted approximately 2.50 m above the table provides information about the cue and ball positions, being able to distinguish between white, black, striped, and solid balls. In order to place the cue properly behind the cue ball, combined data of the camera, the robot’s Sick S300 laser range finders, arm pose data, and two JR3 end effector force/torque sensors is used. The robot is able to move autonomously around the table by detecting the legs of the pool table and execute a stroke independently. Two onboard computers with a 2.66 GHz Intel Pentium i7 920 CPU and 12 GB RAM control both arms and the platform. The combination of a proprietary real-time data base [49] for data exchange with a real-time architecture [50] allows an update rate of 1000 Hz while sufficing hard real-time constraints.

To create an search tree within reasonable time, pool simulations are parallelized over 40 computers equipped with a AMD Phenom II X6 1075T 3 GHz CPU and 8 GB RAM. Another computer with similar specifications is used to compute an optimal stroke as described in Section V based on the computed search tree. During tree creation, a variable branching factor in the range of 300 is used, resulting in a about 100 000 computed strokes for a search depth of 2. The planning capabilities of the robot are evaluated by comparing two scenarios with the same human opponent. In the first scenario, the search depth n is set to 0 [see (19)]. The stroke intensity p has been set to a constant value of 0.17 ns. In total, seven games have been played, resulting in 89 strokes of the robot. In the second scenario, the search depth n is set to 2 while also considering the human decision-making according to Section IV. Here six games have been played, resulting in 67 strokes of the robot.

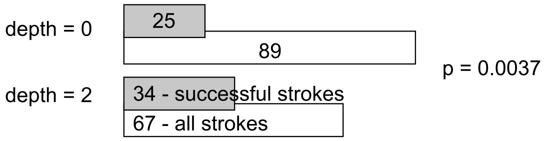


Fig. 10. Number of successful strokes for different search depths using an anthropomorphic robot. Right side: significance level of every comparison using a two-sample *t*-test.

There are not enough games played for any conclusion about the statistical significance of the number of wins/losses. It is, however, possible to draw conclusions by looking at the individual strokes. A stroke is considered as "successful" if the robot can pocket a ball and without committing a foul, thus corresponding to the with repositioning run in Fig. 7. All other strokes executed by the robot are considered to be "unsuccessful." Using this definition, the two results differ significantly from each other (based on a two sample *t*-test) as only 25 out of the 89 strokes are successful for the first scenario whereas 34 out of the 67 strokes are successful for the second scenario Fig. 10.

C. Discussion

The performed experiments show that both excessive planning and learning a model of the opponent significantly improve the outcome of a pool game. Still, there are some more general aspects regarding the human model to be discussed.

A quite unique problem at pool is the coupling between a player's skill in terms of success rate when pocketing a ball and his planning capabilities. A chess player's skill is primarily determined by his planning capabilities whereas the success of a golf player depends mainly on his skill when doing a golf swing. Yet with the exception of novice pool player, one cannot tell by just looking at the result of a game whether a victory comes from superior skill or planning. The used robotic platform during the experimental evaluation possesses superior planning capabilities but inferior motor skill compared to the human opponent. Thus hardware improvements resulting in lower standard deviations σ_p and σ_θ are expected to have a major impact on the outcome of a game.

Only strokes up to a search depth of 2 are simulated. Due to imperfect calibration, the deviation between simulation and experimental results increases as the search tree is extended deeper. To account for this problem, one may introduce an additional parameter which weights future strokes less. Still this parameter is different from the discount factor $\gamma_{\{r,h\}}$ that accounts for the planning capabilities of human and robot.

VII. Conclusion

This paper deals with the question how a robot facing a human in a competitive task can improve its performance through extensive planning and precise representation of the opponent's behavior. This allows the robot to adapt to human-specific decision-making and the opponent's individual motor skill while overcoming its own kinematic constraints. Simulations show that including the human model in the planning framework of the robot significantly improves the win ratio. Real-world experiments with an anthropomorphic robotic platform prove that extensive planning significantly helps for a better position play.

Future work will be focused on employing reinforcement learning for better online adaption to the human model, resulting in a better overall performance against each individual player.

References

[1] D. Kulić and E. A. Croft, "Safe planning for human–robot interaction," *J. Robot. Syst.*, vol. 22, no. 7, pp. 383–396, 2005.

[2] N. Sebanz, H. Bekkering, and G. Knoblich, "Joint action: Bodies and minds moving together," *Trends Cogn. Sci.*, vol. 10, no. 2, pp. 70–76, 2006.

[3] M. Huber *et al.*, "Spatiotemporal movement planning and rapid adaptation for manual interaction," *PLoS ONE*, vol. 8, no. 5, 2013, Art. ID e64982.

[4] D. Feil-Seifer and M. J. Mataric, "Defining socially assistive robotics," in *Proc. IEEE ICORR*, Chicago, IL, USA, 2005, pp. 465–468.

[5] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction," *Image Vis. Comput.*, vol. 25, no. 12, pp. 1875–1884, 2007.

[6] B. Kühnlenz *et al.*, "Increasing helpfulness towards a robot by emotional adaption to the user," *Int. J. Soc. Robot.*, vol. 5, no. 4, pp. 457–476, 2013.

[7] M. A. Goodrich and A. C. Schultz, "Human–robot interaction: A survey," *Found. Trends Human Comput. Interact.*, vol. 1, no. 3, pp. 203–275, 2007.

[8] T. Wösch and W. Feiten, "Reactive motion control for human–robot tactile interaction," in *Proc. IEEE ICRA*, Washington, DC, USA, 2002, pp. 3807–3812.

[9] A. De Santis, "Modelling and control for human–robot interaction," Ph.D. dissertation, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Napoli, Italy, 2007.

[10] J. R. Medina, M. Lawitzky, A. Mortl, D. Lee, and S. Hirche, "An experience-driven robotic assistant acquiring human knowledge to improve haptic cooperation," in *Proc. IEEE IROS*, San Francisco, CA, USA, 2011, pp. 2416–2422.

[11] Y. Demiris, "Prediction of intent in robotics and multi-agent systems," *Cogn. Process.*, vol. 8, no. 3, pp. 151–158, 2007.

[12] W. Erlhagen *et al.*, "Action understanding and imitation learning in a robot-human task," in *Proc. ICANN*, vol. 1. Warsaw, Poland, 2005, pp. 261–268.

[13] N. Sebanz and G. Knoblich, "Prediction in joint action: What, when, and where," *Topics Cogn. Sci.*, vol. 1, no. 2, pp. 353–367, 2009.

[14] S. S. Obhi and N. Sebanz, "Moving together: Towards understanding the mechanisms of joint action," *Exper. Brain Res.*, vol. 211, pp. 329–336, Jun. 2011.

[15] G. Bornstein, "Intergroup conflict: Individual, group and collective interests," Center Study Ration., Hebrew University, Jerusalem, Israel, Tech. Rep. dp297, 2002.

[16] Z. Wang *et al.*, "Probabilistic modeling of human movements for intention inference," in *Proc. RSS*, Sydney, NSW, Australia, 2012.

[17] P. Sastry, V. Phansalkar, and M. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 5, pp. 769–777, May 1994.

[18] M. Campbell, A. J. Hoane, Jr., and F.-H. Hsu, "Deep blue," *Artif. Intell.*, vol. 134, nos. 1–2, pp. 57–83, 2002.

[19] A. Al-Tamimi, M. Abu-Khalaf, and F. Lewis, "Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 240–247, Feb. 2007.

[20] M. M. Veloso and P. Stone, "Video: RoboCup robot soccer history 1997–2011," in *Proc. IROS*, Vilamoura, Portugal, 2012, pp. 5452–5453.

[21] W. Leckie and M. Greenspan, "An event-based pool physics simulator," in *Proc. 11th Int. Conf. Adv. Comput. Games*, Taipei, Taiwan, 2005, pp. 247–262.

[22] R. E. Wallace and M. C. Schroeder, "Analysis of billiard ball collisions in two dimensions," *Amer. J. Phys.*, vol. 56, no. 9, pp. 815–819, 1988.

[23] D. Gugan, "Inelastic collision and the hertz theory of impact," *Amer. J. Phys.*, vol. 68, no. 10, pp. 920–924, 1999.

[24] I. Han, "Dynamics in carom and three cushion billiards," *J. Mech. Sci. Technol.*, vol. 19, no. 4, pp. 976–984, 2005.

[25] R. Cross, "Cue and ball deflection in billiards," *Amer. J. Phys.*, vol. 76, no. 3, pp. 205–212, 2007.

[26] M. Smith, "Pickpocket: A computer billiards shark," *Artif. Intell.*, vol. 171, pp. 1069–1091, Nov. 2007.

[27] C. Archibald, A. Altman, and Y. Shoham, "Analysis of a winning computational billiards player," in *Proc. IJCAI*, Pasadena, CA, USA, 2009, pp. 1377–1382.

[28] J.-F. Landry and J.-P. Dussault, "AI optimization of a billiard player," *J. Intell. Robot. Syst.*, vol. 50, no. 4, pp. 399–417, 2007.

[29] J.-F. Landry, J.-P. Dussault, and P. Mahey, "Billiards: An optimization challenge," in *Proc. 4th Int. C* Conf. Comput. Sci. Softw. Eng.*, Montreal, QC, Canada, 2011, pp. 129–132.

[30] Z. M. Lin, J.-S. Yang, and C. Y. Yang, "Grey decision-making for a billiard robot," in *Proc. IEEE SMC*, The Hague, The Netherlands, 2004, pp. 5350–5355.

[31] S. C. Chua, E. K. Wong, and V. C. Koo, "Intelligent pool decision system using zero-order Sugeno fuzzy system," *J. Intell. Robot. Syst.*, vol. 44, no. 2, pp. 161–186, 2005.

[32] C. Archibald and Y. Shoham, "Modeling billiards games," in *Proc. AAMAS*, Budapest, Hungary, 2009, pp. 193–199.

[33] W. Cheung and S. Sang, "Automating skills using a robot snooker player," Ph.D. dissertation, Bristol Univ., U.K., 1994.

[34] M. Greenspan *et al.*, "Toward a competitive pool playing robot: Is computational intelligence needed to play robotic pool?" in *Proc. IEEE CIG*, Honolulu, HI, USA, 2007, pp. 380–388.

[35] (2010). *The pr2 Plays Pool. Willow Garage*. [Online]. Available: http://www.willowgarage.com/blog/2010/06/15/pr2-plays-pool

[36] T. Nierhoff, O. Kourakos, and S. Hirche, "Playing pool with a dual-armed robot," in *Proc. IEEE ICRA*, Shanghai, China, 2010, pp. 3445–3446.

[37] N. Jarrassé, T. Charalambous, and E. Burdet, "A framework to describe, analyze and generate interactive motor behaviors," *PLoS ONE*, vol. 7, no. 11, 2012, Art. ID e49945.

[38] A. Kolobov, Mausam, and D. S. Weld, "A theory of goal-oriented MDPs with dead ends," in *Proc. UAI*, Catalina Island, CA, USA, 2012, pp. 438–447.

[39] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[40] E. A. Hansen and S. Zilberstein, "Lao*: A heuristic search algorithm that finds solutions with loops," *Artif. Intell.*, vol. 129, nos. 1–2, pp. 35–62, 2001.

[41] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using realtime dynamic programming," *Artif. Intell.*, vol. 72, nos. 1–2, pp. 81–138, 1995.

[42] J. Hoffmann and B. Nebel, "The FF planning system: Fast plan generation through heuristic search," *J. Artif. Intell. Res.*, vol. 14, no. 1, pp. 253–302, 2001.

[43] A. Salazar and A. Sanchez-Lavega, "Motion of a ball on a rough horizontal surface after being struck by a tapering rod," *Eur. J. Phy.*, vol. 11, no. 4, pp. 228–232, 1990.

[44] T. Nierhoff, K. Heunisch, and S. Hirche, "Strategic play for a pool-playing robot," in *Proc. IEEE ARSO*, Munich, Germany, 2012, pp. 72–78.

[45] S. Chua, E. Wong, A. Tan, and V. Koo, "Decision algorithm for pool using fuzzy system," in *Proc. ICAIET*, Kota Kinabalu, Malaysia, 2002, pp. 370–375.

[46] J.-P. Dussault and J.-F. Landry, "Optimization of a billiard player—Tactical play," in *Computers and Games*. Berlin, Germany: Springer, 2006, pp. 256–270.

[47] K. Leibrandt, T. Lorenz, T. Nierhoff, and S. Hirche, "Modelling human gameplay at pool and countering it with an anthropomorphic robot," in *Proc. ICSR*, Bristol, U.K., 2013, pp. 30–39.

[48] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, NJ, USA: Pearson Educ., 2003.

[49] M. Goebl and G. Färber, "A real-time-capable hard-and software architecture for joint image and knowledge processing in cognitive automobiles," in *Proc. IEEE Intell. Veh. Symp.*, Istanbul, Turkey, 2007, pp. 734–740.

[50] D. Brscic *et al.*, "Multi joint action in CoTeSys—Setup and challenge," CoTeSys Cluster Excelence, Technische Universität München, Ludwig-Maximilians-Universität München, Munich, Germany, Tech. Rep. CoTeSys-TR-10-01, 2010.

Thomas Nierhoff received the B.Sc. and Dipl.-Ing. (M.Sc. equivalent) degrees in electrical engineering from Technische Universität München, Munich, Germany, in 2009 and 2010.

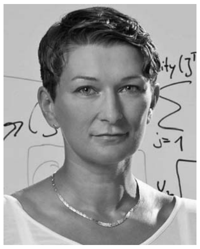
He was a Research Assistant with the Institute of Information-Oriented Control, Technische Universität München. From 2012 to 2014, he was a Visiting Researcher with Nakamura Laboratory, University of Tokyo, Tokyo, Japan. His current research interests include robotics, motion planning, and machine learning.

Konrad Leibrandt received the Dipl.-Ing. (M.Sc. equivalent) degree in electrical engineering from Technische Universität München, Munich, Germany, in 2012, and the M.Res. degree in medical robotics and image guided intervention from Imperial College London, London, U.K., in 2013, where he is currently pursuing the Ph.D. degree with the Hamlyn Centre for Robotic Surgery and the Department of Computing.

His current research interests include active constraints and virtual fixtures, ergonomic control for human–robot interaction in surgery, continuum robot control, and high-performance parallel computing for surgical applications.

Tamara Lorenz received the Diploma (M.Sc. equivalent) degree in mechanical engineering from Technische Universität München (TUM), Munich, Germany, specializing in human–machine interaction and medical engineering, and the Ph.D. degree from the Graduate School of Systemic Neurosciences, Ludwig–Maximilians Universität (LMU), Munich, in 2015.

She was a Research Assistant with the Institute of Information-Oriented Control, TUM. She was with the Department of Experimental Psychology, LMU from 2009 to 2013. Her current research interests include dynamics and rules that underlie human–human interaction and how human interaction can be modeled to improve the interaction between humans and robots.



Sandra Hirche received the Diploma degree in mechanical engineering and transport systems from the Technical University of Berlin, Berlin, Germany, in 2002, and the Doctor of Engineering degree in electrical engineering and information technology from Technische Universität München (TUM), Munich, Germany, in 2005.

She was an Associate Professor with TUM, where she has been a TUM Liesel Beckmann Distinguished Professor and the Chair of Information-Oriented Control with the Department of Electrical and Computer Engineering since 2013. She has published over 150 papers in international journals, books, and refereed conferences. Her current research interests include cooperative, distributed and networked control with applications in human–robot interaction, multirobot systems, and general robotics.

Dr. Hirche was a recipient of the Post-Doctoral Scholar Award from the Japanese Society for the Promotion of Science, Fujita Laboratory, Tokyo Institute of Technology, Tokyo, Japan, from 2005 to 2007, multiple awards such as the Rohde and Schwarz Award for her Ph.D. thesis in 2005, the IFAC World Congress Best Poster Award in 2005, and together with students the Best Paper Awards of the IEEE Worldhaptics and the IFAC Conference of Manoeuvring and Control of Marine Craft in 2009. From 2008 to 2012, she served on the Editorial Board of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY and the IEEE TRANSACTIONS ON HAPTICS.