Inverse Reinforcement Learning for Strategy Extraction

Katharina Muelling[1,3], Abdeslam Boularias[1], Betty Mohler[2], Bernhard Schölkopf[1], and Jan Peters[1,3]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
{muelling,boularias,bs,jrpeters}@tuebingen.mpg.de

[2]Max Planck Insitute for Biological Cybernetics, Tübingen, Germany
mohler@tuebingen.mpg.de

[3]Technische Universität Darstadt, FG IAS, Darmstadt, Germany
{muelling,peters}@ias.tu-darmstadt.de

Abstract. In competitive motor tasks such as table tennis, mastering the task is not merely a matter of perfect execution of a specific movement pattern. Here, a higher-level strategy is required in order to win the game. The data-driven identification of basic strategies in interactive tasks, such as table tennis is a largely unexplored problem. In order to automatically extract expert knowledge on effective strategic elements from table tennis data, we model the game as a Markov decision problem, where the reward function models the goal of the task as well as all strategic information. We collect data from players with different playing skills and styles using a motion capture system and infer the reward function using inverse reinforcement learning. We show that the resulting reward functions are able to distinguish the expert among players with different skill levels as well as different playing styles.

Keywords: Computational models of decision processes, Table tennis, Inverse reinforcement learning

1 Introduction

Understanding the complex interplay between learning, decision making and motion generation is crucial both for creating versatile, intelligent robot systems as well as for understanding human motor control. For example, in table tennis, a player usually cannot win the game by always returning the ball safely to the same position. Instead, players need a good strategy that defines where and how to return the ball to the opponent's court. An action should always be chosen to have a high probability of successfully returning the ball as well as to make the task of the opponent harder, i.e., it should improve the chance of winning the game. In this paper, we want to infer strategic information from a game of table tennis. Rather than identifying the frequencies and effectiveness of specific movement patterns [1–4], we want to model the decision process for choosing actions by players in a match of table tennis from a computational point of view. Thus, we are not only able to use the learned model for artificial systems, such as table tennis robots [5], but also yield a better insight into the reasons for choosing a

given action in a specific state. Therefore, we only consider basic features available to the player.

A common way to model decision processes in artificial systems is to use a Markov Decision Problem (MDP [6]). Here, an agent interacts with a dynamic environment. It chooses and executes an action that will change the state of the player and its environment. The agent can observe this state change and may receive a reward for its action. A strategy defines the general plan of choosing actions in specific states in order to achieve a goal. A strategy in the MDP framework is usually called a *policy*. The expert knowledge used to win the game can be captured in the reward function that defines the reward the agent will receive in a specific situation when executing an action.

The process of determining the reward function from an expert demonstration is referred to as *Inverse Reinforcement Learning* (IRL [7, 8]). IRL has been applied to many problems such as helicopter control [9], routing preferences of drivers [10] and, user simulation in spoken dialog management systems [11]. In most of these approaches, the underlying dynamics of the system is assumed to be known. However, the dynamics of human behavior is usually difficult to model. We avoid modeling these complex dynamics by learning the strategies directly from human demonstration.

In the remainder of this paper, we will proceed as follows. In Section 2, we present the theoretical background for modeling decision processes, including MDPs and IRL techniques. We present the experimental setup and evaluations in Section 3. In Section 4, we summarize our approach and the results.

2 Modeling Human Strategies

In this section, we will first introduce the notation and basic elements necessary for the table tennis model. Subsequently, we will discuss different model-free Inverse Reinforcement Learning (IRL) approaches and show how the states, actions and reward features in the table tennis task can be represented.

2.1 Preliminaries

To employ IRL, the problem at hand needs to be modeled as a Markov Decision Problem (MDP). Formally, a MDP is a tuple (S, A, T, R), where S is the state space, A is the action space, and $\mathcal{T}$ is a transition function $\mathcal{T}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = Pr(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, with states $\mathbf{s}_t, \mathbf{s}_{t+1} \in S$ and actions $\mathbf{a}_t \in A$. The function $R(\mathbf{s}, \mathbf{a})$ defines the reward for executing action $\mathbf{a}$ in state $\mathbf{s}$.

A deterministic policy π is a mapping: $S \mapsto A$ and defines which action is chosen in a state $\mathbf{s} \in S$. A stochastic policy is a probability distribution over actions in a given state $\mathbf{s}$ and is defined as $\pi(\mathbf{s}|\mathbf{a}) = Pr(\mathbf{a}|\mathbf{s})$. The performance of a policy is measured with the so-called *value function* $V^\pi(\mathbf{s})$. The value function of a policy π evaluated at state $\mathbf{s}$ for a finite horizon H is given by $V^\pi(\mathbf{s}) = \frac{1}{H}\mathbb{E}[\sum_{t=0}^{H-1} R(\mathbf{s}_t, \mathbf{a}_t)|\pi, \mathcal{T}, \mathbf{s}_0 = \mathbf{s}]$, and corresponds to the expected reward following policy πstarting from state $\mathbf{s}$. The optimal value function is defined by $V^*(\mathbf{s}) = \max_\pi V^\pi(\mathbf{s})\ \forall \mathbf{s} \in S$. The goal of an agent is to find the optimal policy π^*, i.e., a policy that maximizes the expected return for every $\mathbf{s} \in S$.

We assume that the reward function R is given by a linear combination of m feature functions f_i with weights w_i. The reward function is therefore defined by $R(\mathbf{s},\mathbf{a}) = \sum_{i=1}^{m} w_i f_i(\mathbf{s},\mathbf{a}) = \mathbf{w}^{\mathrm{T}}\mathbf{f}(\mathbf{s},\mathbf{a})$, where $\mathbf{w} \in \mathbb{R}^m$ and $\mathbf{f}(\mathbf{s},\mathbf{a}) \in \mathbb{R}^m$. The features f_i are fixed, known, bounded basis functions mapping from $S \times A$ into $\mathbb{R}$. Similarly to the value function, we can define the feature count f_i^π under policy π by $f_i^\pi(\mathbf{s}) = \frac{1}{H}\mathbb{E}[\sum_{t=0}^{H-1} f_i(s_t,a_t)|\pi,\mathcal{T},\mathbf{s}_0 = \mathbf{s}]$ as the expected features observed when following policy π. As a result, V^π can be written as $V_{\mathbf{w}}^\pi(\mathbf{s}) = \sum_{i=1}^{m} w_i f_i^\pi(\mathbf{s}) = \mathbf{w}^{\mathrm{T}}\mathbf{f}^\pi(\mathbf{s})$, where $\mathbf{f}^\pi \in \mathbb{R}^m$ is a vector containing the single feature counts $f_i^\pi(\mathbf{s})$ as entries.

2.2 Learning the Reward Function

The reward function is a crucial part of the MDP as it defines the goal of the task and shapes the policy optimization process. The problem of designing the right reward function led to the development of IRL methods. Given the actions of an agent that is assumed to behave in an optimal manner, the available sensory information about the environment and, if possible, a model of the environment, the goal of IRL is to determine a reward function that can (mostly) justify the demonstrated behavior. A recent review of IRL algorithms can be found in [12].

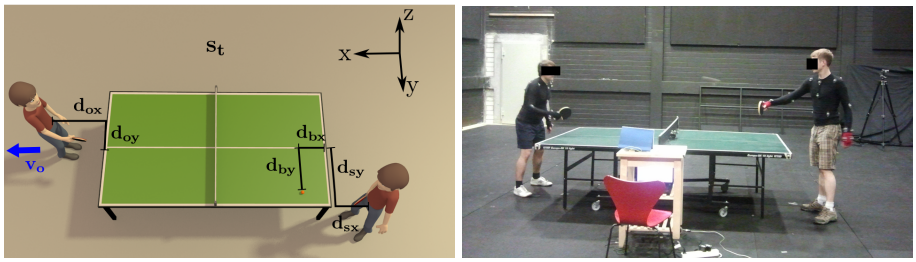
Most IRL approaches rely on a given model of the environment $\mathcal{T}$ or assume that it can be accurately learned from the demonstrations. In this paper, we want to estimate the underlying reward function for playing table tennis based on demonstrations *without having to model the correct dynamics model*. Only few model-free IRL methods have been suggested [13, 14].

Instead of collecting only demonstrations from an expert we use also demonstrations from less skilled players for finding the reward function. To compute the reward weights, we compared two different methods. The first evaluated method is based on the max-margin algorithm of Abbeel and Ng [15], while the second is the model-free relative entropy IRL algorithm [13]. In the following, we assume that we are given a set of expert demonstrations $D^E = \{\tau_p\}_{p=1}^{P}$, where $\tau_p = \mathbf{s}_1^p\mathbf{a}_1^p, ..., \mathbf{s}_{T_p}^p\mathbf{a}_{T_p}^p$ corresponds to one rally (i.e., state-action trajectory), as well as a set of non-optimal demonstrations $D^N = \{\tau_l\}_{l=1}^{L}$. Here, T_p is the number of volleys (i.e., state-action pairs) in the observed rally τ_p.

Model Free Maximum Margin. The max-margin method of Abbeel and Ng [15] aims at finding a policy π that has feature counts close to that of the expert, i.e., $\|\mathbf{f}^\pi - \mathbf{f}^{\pi_E}\|_2 \leq \epsilon$. Using the max-margin algorithm [15] in a model-free setup in a straight forward manner is not suited as it is unlikely that any player plans the strokes for more than only a few steps ahead. Therefore, we need to compare the values of the expert in every state in the recorded trajectories to the ones of the non-experts in the same state. We can find the weight vector $\mathbf{w}$ by solving the quadratic optimization problem

$$\max_{\mathbf{w}} \sum_{p=1}^{P}\sum_{t=0}^{T_p} \left(V_{\mathbf{w}}^{\pi_E}(\mathbf{s}_t^p) - \hat{V}_{\mathbf{w}}^{\pi_N}(\mathbf{s}_t^p)\right) - \lambda||\mathbf{w}||_2,$$

where $\hat{V}_{\mathbf{w}}^{\pi_N}(\mathbf{s}_t^p)$ is an estimated value of the non-expert players in the current state $\mathbf{s}_t^p$ of the expert. Estimating the value $\hat{V}^{\pi_N}$ in a given state $\mathbf{s}$ is a regression problem that we propose to solve by using the k-nearest neighbors method,



(a) State (b) Experimental Setup

Fig. 1. Fig. a illustrates the state of the system, defined by the relative position of the agent (d_{sx}, d_{sy}) and the the relative position (d_{ox}, d_{oy}) and velocity ($\mathbf{v}_o$) of the opponent towards the table, as well as the the position (d_{bx}, d_{by}) and velocity ($\mathbf{v}_b$) of the ball. Fig. b shows the experimental setup. A naive player (right side) plays against an skilled opponent (left side).

$\hat{V}^{\pi_N}_{\mathbf{w}}(\mathbf{s}) = \frac{1}{k}\sum_{\mathbf{s}'\in\mathcal{N}_k(\mathbf{s})} V^{\pi_N}_{\mathbf{w}}(\mathbf{s}')$, where $\mathcal{N}_k(\mathbf{s})$ is the set of k-nearest neighbors of $\mathbf{s}$ among all the states that have been observed in the non-optimal trajectories. We use a Gaussian kernel to define a similarity measure between states.

The value functions V^{π_E} and V^{π_N} of the expert's policy π_E and non-optimal policies π_N are computed as

$$V^{\pi}_{\mathbf{w}}(\mathbf{s}^p_t) = \frac{1}{H^p_t - t + 1}\sum_{i=t}^{H^p_t} \mathbf{w}^{\mathrm{T}}\mathbf{f}^{\pi}(\mathbf{s}^p_i, \mathbf{a}^p_i),$$

where $H^p_t = \min\{t + H - 1, T_p\}$ and H is the planning horizon, i.e., the number of steps we look into the future. In the following, we will refer to this algorithm as MM (Maximum Margin).

Relative Entropy Method. The relative entropy IRL method [13] finds a distribution $\mathcal{P}$ over trajectories that minimizes the KL-divergence to a reference distribution Q, while ensuring that the feature counts under $\mathcal{P}$ are similar to the feature counts in the expert trajectories. The solution to this problem takes the following form

$$\mathcal{P}(\tau|\mathbf{w}) = \frac{1}{Z(\mathbf{w})} Q(\tau)\exp\left(\mathbf{w}^T f^{\tau}_i\right),$$

where $Z(\mathbf{w}) = \sum_{\tau} Q(\tau)\exp\left(\mathbf{w}^T f^{\tau}_i\right)$. The reward weight vector $\mathbf{w}$ is found by solving the optimization problem $\max_{\mathbf{w}} \mathbf{w}^T\mathbf{f}^{\pi_E} - \ln Z(\mathbf{w}) - \lambda\|\mathbf{w}\|_1$. The gradient of this objective function is calculated by re-using the expert and non-optimal trajectories with importance sampling. For our experiments, we choose the reference distribution Q to be uniform. In the following, we will refer to this algorithm as RE (Relative Entropy).

2.3 Computational Model for Representing Strategies in Table Tennis

As a next step, we need to specify the states, actions and reward features of the table tennis task. The state of the system consist of all sensory information

experienced by the agent. However, learning in such high-dimensional continuous state domains is likely to be intractable. Therefore, we assume that the player has to decide where and how to hit the ball when the hitting movement is initiated. Furthermore, we assume that the decision depends on the following information: the planar Cartesian position of the agent $\mathbf{d}_s = [d_{sx}, d_{sy}]$, the opponent's position $\mathbf{d}_o = [d_{ox}, d_{oy}]$ and velocity $\mathbf{v}_o$, the state of the rally $\mathbf{g} \in \{\text{player serve, opponent serve, not served}\}$ as well as the ball position $\mathbf{d}_b = [d_{bx}, d_{by}]$, velocity $|\mathbf{v}_b|$ and direction given by the angles θ_{py} and θ_{pz} (see Fig. 1). The variables θ_{py} and θ_{pz} are defined as the horizontal and vertical bouncing angles of the ball at the moment of impact on the player's side of the table, respectively. θ_{pz} defines the bouncing angle in the xz-plane and corresponds to how flat the ball was played. θ_{py} defines the bouncing angle in the xy-plane. Additionally, we define a set of terminal states $s_T \in \{W, L\}$ for winning and loosing the rally respectively.

The action defines where and how to return the ball to the opponent's court. This decision includes the desired bouncing point $\mathbf{p}_b$ of the ball on the opponent's court, the corresponding bouncing angles θ_{oy} and θ_{oz}, the velocity $||\mathbf{v}_b||$ and the spin of the ball. Since the different kinds of spin are hard to capture without an expert classifying the sampled data, we discard the spin and use only basic strategic elements.

The reward features $f_i(\mathbf{s}, \mathbf{a})$ for each state-action pair are defined by: (i) the goal of the ball on the opponent's court, (ii) the proximity of the ball to the edge of the table $\boldsymbol{\delta}_{\text{t}}$, (iii) the distance of the bouncing point of the ball on the opponent's court and the right hand of the opponent $\boldsymbol{\delta}_{\text{o}}$, (iv) the proximity of the ball to the elbow δ_{elbow}, (v) the velocity of the ball $\|\mathbf{v}_b\|$, (vi) the velocity of the opponent v_o relative to the ball in y-direction, (vii) the bouncing angles θ_{oz} and θ_{oy} of the ball when bouncing on the opponent's side of the court, and (viii) whether the ball was a smash or not. All features are scaled to lie in an interval of [0 1], except for the direction sensitive features θ_{oy} and v_o, which lie in an interval of [-1 1].

3 Experiments and Evaluations

In this section we describe the experiments for the data collection and the results of the evaluation of the presented approach.

3.1 Experimental Setup and Data Collection

We recorded table tennis players with various skill levels. Therefore, we used eight right-handed subjects of all genders which could be grouped into naive and skilled players. The group of naive players consisted of five subjects and fulfilled the following requirements: (i) never played in a table tennis club, (ii) did not train any racket sports on a regular basis in the last five years, and (iii) did not participate in table tennis tournaments. The group of skilled players consisted of three subjects and fulfilled the following requirements: (i) played for at least eight years in a table tennis club, (ii) trained at least twice a week and (iii) participate regularly in table tennis competitions.

One of the skilled players were used as a permanent *opponent* and, therefore, was not considered part of the subject set. Each subject played a game of table tennis under the following three conditions. In Condition 1, the subjects played a cooperative game of table tennis for a ten minute period. In Condition 2, the subjects were told to perform a competitive game of table tennis, while the opponent was instructed to return the ball "nicely" (i.e., the opponent was instructed to play towards the subject in a cooperative way). In Condition 3, both the subject and the opponent were instructed to play a competitive game of table tennis.

In order to collect information about the position of the participants, the table and the ball during the game, we used a VICON motion capture system. With this setup a 3D kinematic model of the upper body of each individual was captured during the game. The experimental setup is also shown in Fig. 1b.

3.2 Results and Discussion

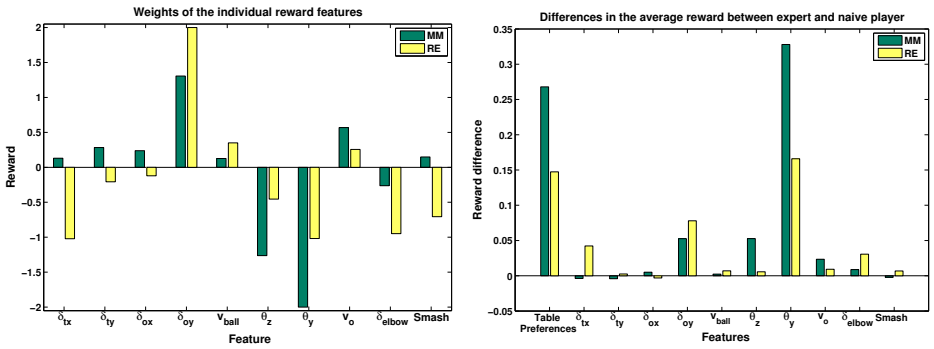
Only one of the skilled subjects was able to win against the opponent under Condition 3. All other games were won by the fixed opponent. The scoring results of the subjects that lost the game can be found in Table 1. Based on these results, the data was divided into two subsets: (1) a non-expert data set and (2) an expert data set. The non-expert data set included all games of the subjects who lost against the fixed opponent, i.e., all naive subjects and one of the skilled players, as well as all cooperative games. We will refer to the players that lost as Naive 1 to 5 and Skilled 1. The expert data set consisted of all rallies in the competitive game (Condition 3) of the skilled player that won against the opponent. We will refer to this player as Expert. When asked which player performed worst, the opponent stated that Naive 3 was the worst.

To evaluate the potential reward functions, we performed a leave-one-subject-out testing scheme. We computed the reward feature weights for each of the two methods as described in Section 2.2 seven times. Every time leaving out all rallies (i.e., state-action trajectories) of one of the subjects that lost or the rallies of the cooperative game of the Expert respectively. We also excluded 20 rallies of the Expert for the validations. For the MM algorithm we determined empirically an optimal planning horizon of three, which is used throughout the evaluations.

Classifying the skill levels of the players. We computed the differences in the average reward for a state-action pair of the spared expert and non-expert

Table 1. Summary of the results of the evaluations for the different methods. The differences in the average rewards with respect to the expert, define the differences between the reward of the expert and the spared test subject of the non-expert data set.

	Method	Naive 1	Naive 2	Naive 3	Naive 4	Naive 5	Skilled 1	Coop.
Average reward differences	MM	1.16	0.07	1.24	0.86	0.71	0.33	0.50
	RE	0.70	0.11	0.60	0.80	0.42	0.31	0.55
Scores in Condition 2		5:33	12:33	2:33	5:33	2:33	21:34	
Scores in Condition 3		13:33	17:33	10:33	5:33	17:33	20:33	



(a) Reward feature weights (b) Average reward differences

Fig. 2. Fig. (a) shows the weights of all other features for the MM algorithm and the RE algorithm, respectively. Fig. (b) shows the differences of the average reward of the expert and the naive player for each feature seperately.

data for the obtaind reward functions (see Table 1). All reward functions were able to distinguish between the non-expert games and the expert game, as well as between the different playing styles of the expert (competitive vs cooperative). In general the average reward for each player reflected the skill level of the players with the exception of Naive 2.

All reward functions obtained in the evaluation resulted in a very small difference in the average reward of the Expert and Naive 2, followed by Skilled 1 and Naive 5. Furthermore, both methods showed relatively large differences between the Expert and Naive 1, Naive 3 and Naive 4. However, they disagree in the ranking of these players. While the reward function obtained by the RE algorithm shows the highest difference for the Expert and Naive 4, the reward function obtained by the MM algorithm yields the highest difference between the Expert and Naive 3. Naive 4 being the worst player is in compliance with the scoring results for Condition 3, while Naive 3 being the worst player is in compliance with the statement of the opponent. Analyzing player Naive 2, we can conclude that the player chooses his actions based on the same principles as both skilled players, but lost against the opponent due to his inaccurate movement execution.

Individual reward features. Analyzing the reward weights individually, the different methods showed similar weights for the most important features (i.e., the features with the highest weights). The reward weights and differences for the individual features are displayed in Fig. 2a and b. The largest influence resulted from the bouncing angles θ_y and θ_z, the table preferences and the distance between the desired bouncing point and the racket of the opponent. We will discuss these features in the following. Other features as playing against the moving direction and the velocity of the ball were also positive correlated.

Goal preferences on the table. The resulting reward functions of the different algorithms showed a preference for the areas where the opponent would have to return the ball using the backhand, while the areas that are suited for returning

the ball with the forehand and the areas directly after the net are rather avoided.

Distance to the opponent. Maximizing the distance in y-direction (i.e., along the width of the table) between the bouncing point and the racket of the opponent resulted in a high reward in both reward functions. This feature also influenced the differences in the reward yield by the naive and expert table tennis player. The overall performance on average only increased slightly. The differences in the average reward for the features before a terminal state, increased dramatically and became a dominant factor in the reward function (see Fig. 2b). This observation suggests that the chance of winning a point increases with an increasing distance between the bouncing point and the racket between the player.

Bouncing Angles. The horizontal angle θ_z had a high negative reward value, i.e., playing the ball flat was preferred. The angle θ_y also had a high negative weight, i.e., playing the ball cross to the backhand area was preferred opposed to playing the ball cross towards the forehand area. These results are conform with the table preferences. This feature was one of the dominating factors in the reward function and in the evaluations of the excluded subjects. However, the average difference between expert and naive players for the state right before the terminal state was only decreased slightly. The average reward two states before the terminal state on the other side became one of the dominant factors.

This observation together with the results of the distance of the bouncing point and the racket, suggests the following strategy successfully applied by the Expert only. When playing the ball very cross to the outer backhand area of the opponent, the opponent was forced to move to his left. The expert used this opportunity to play the ball to the other side of the table in order to increase the distance between the ball and the opponent.

4 Conclusion

In this paper, we modeled table tennis as a MDP. We have shown that it is possible to automatically extract expert knowledge on effective elements of basic strategy in the form of a reward function using model-free IRL. To accomplish this step, we collected data from humans playing table tennis using a motion capture system. Participants with different skill levels played in both a competitive and a cooperative game during this study. We divided the data into an expert and a non-optimal data set and used them to infer and evaluate the reward functions.

We have tested two different model-free inverse reinforcement learning methods. One was derived from the model-based IRL method of Abeel and Ng [15]. The second algorithm was model-free relative entropy [13]. The resulting reward functions were evaluated successfully in a leave-one-subject-out testing scheme. All learned reward functions were able to distinguish strategic information of players with different playing skills and styles. The key elements revealed by the model were (i) playing cross to the backhand area of the opponent, (ii) maximizing the distance of the bouncing point of the ball and the opponent, and (iii) playing the ball in a flat manner. Other elements as playing against the moving direction and the velocity of the ball were also positively correlated.

References

1. P. Wang, R. Cai, and S. Yang, "A tennis video indexing approach through pattern discovery in interactive process," *Advances in Multimedia Information Processing*, vol. 3331, pp. 59 – 56, 2004.
2. J. Wang and N. Parameswaran, "Analyzing tennis tactics from broadcsting tennis video clips," in *Proceedings of the 11th International Multimedia Modelling Conference*, pp. 102 – 106, 2005.
3. J. Vis, W. Kosters, and A. Terroba, "Tennis patterns: Player, match and beyond," in *22nd Benelux Conference on Artificial Intelligence*, 2010.
4. A. Hohmann, H. Zhang, and A. Koth, "Performance diagnosis through mathematical simultion in table tennis," in *Science and Racket Sports III* (A. Lees, J.-F. Kahn, and I. Maynard, eds.), pp. 220 – 226, London: Routledge, 2004.
5. K. Muelling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 263 – 279, 2013.
6. M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1st ed., 1994.
7. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15 of *Studies in Applied Mathematics*. Philadelphia, PA: SIAM, June 1994.
8. A. Ng and X. Russel, "Algorithms for inverse reinforcement learning," in *Proceedings of the 17th International Conference of Machine Learning*, pp. 663 – 670, 2000.
9. P. Abbeel, A. Coates, and A. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, pp. 1608 – 1679, 2010.
10. B. Ziebart, A. Maas, A. Bagnell, and A. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the 23th National Conference of Articiial Intelligence (AAAI)*, pp. 1433 – 1438, 2008.
11. S. Chandramohan, M. Geist, F. Lefevre, and O. Pietquin, "User simulation in dialogue systems using inverse reinforcement learning," in *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, 2011.
12. S. Zhifei and E. Joo, "A survey of inverse reinforcement learning techniques," *International Journal of Intelligent Computing and Cybernetics*, vol. 5, no. 3, pp. 293 – 311, 2012.
13. A. Boularias, J. Kober, and J. Peters, "Relative entropy inverse reinforcement learning," in *Proceedings of the Artificial Intelligences and Statistics (AISTATS)*, pp. 20 – 27, 2011.
14. T. Mori, M. Howard, and S. Vijayakumar, "Model-free apprenticeship learning for transfer of human impedance behaviour," in *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, pp. 239 – 246, 2011.
15. P. Abbeel and A. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the 21st International Conference of Machine Learning (ICML)*, 2004.