Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Distilling Contact Planning for Fast Trajectory Optimization in Robot Air Hockey

Julius Jankowski*[†‡], Ante Marić*[†‡], Puze Liu[§¶], Davide Tateo[§], Jan Peters[§¶‖], Sylvain Calinon[†‡]
[†]Idiap Research Institute, Martigny, Switzerland
[‡]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[§]Intelligent Autonomous Systems, TU Darmstadt, Germany
[¶]German Research Center for AI (DFKI)
[‖]Centre for Cognitive Science, Hessian.AI

***Abstract*—Robot control through contact is challenging as it requires reasoning over long horizons and discontinuous system dynamics. Highly dynamic tasks such as Air Hockey additionally require agile behavior, making the corresponding optimal control problems intractable for planning in realtime. Learning-based approaches address this issue by shifting computationally expensive reasoning through contacts to an offline learning phase. However, learning low-level motor policies subject to kinematic and dynamic constraints can be challenging if operating in proximity to such constraints is desired. This paper explores the combination of distilling a stochastic optimal control policy for high-level contact planning and online model-predictive control for low-level constrained motion planning. Our system learns to balance shooting accuracy and resulting puck speed by leveraging bank shots and the robot's kinematic structure. We show that the proposed framework outperforms purely control-based and purely learning-based techniques in both simulated and real-world games of Robot Air Hockey.**

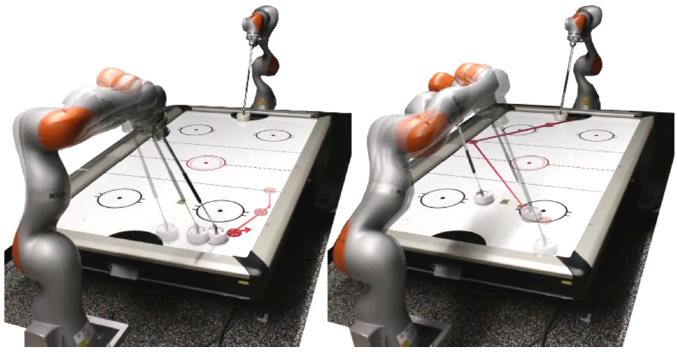

Fig. 1. The proposed control framework enables our robot to autonomously play matches of air hockey. The dynamic game requires the robot to predict puck trajectories, plan the best contact, and coordinate its joints to generate high velocities without hitting a wall or lifting the mallet from the table.

I. Introduction

Planning and control through non-prehensile contacts is an essential skill for robots to interact with their environment. Model-based approaches enable robots to anticipate the outcome of contact interactions given a candidate action, allowing them to find an action with the desired outcome. Although model-based planning approaches have been shown to be successful in generating contact-rich plans for slow tasks [24, 13], highly dynamic tasks require the agent to regenerate contact plans at a sufficiently high rate to react to inherent perturbations. These tasks have historically been used as a testbed for hardware and algorithms in robotics, with different types of games and sports, such as ball-in-a-cup [14, 17], juggling [26, 25], and diabolo [31]. Dynamic tasks that involve contact, such as soccer [9], tennis [32], table tennis [22, 5], and air hockey [19, 20], are typically approached with reinforcement learning methods to off-load computationally expensive reasoning through contacts to an offline exploration phase. Yet, these tasks have in common that contact with the ball or puck is instantaneous, resulting in a jump in the object state. Reasoning about the contact between the robot and the object of interest can therefore be divided into three segments of the planning horizon: *i)* Moving the robot into contact, *ii)* the contact itself at a single time instance, and *iii)* the passive trajectory of the object after contact.

This paper exploits such separability in the highly dynamic game of *air hockey* (Fig. 1) by combining a learning-based approach for contact planning with model-based control for moving the robot into contact. We show that distilling a stochastic optimal control policy enables us to effectively reduce the planning horizon required to generate desired behaviors, allowing our agent to operate in real-time. Furthermore, we highlight the interpretability of our approach by producing various shooting behaviors through different formulations of the optimal control cost, while exploiting the kinematic structure of the robot.

Fig. 2 illustrates the online control framework that consists of state estimation, a learning-based contact planner (shooting policy), and a subsequent model-based robot controller (MPC). For state estimation and prediction, we learn a stochastic model of contact between the robot and the puck from data as a mixture of linear-Gaussian modes. Based on the learned model, we generate a dataset of example contacts that are optimal w.r.t. a stochastic optimal control objective, and distill the resulting policy through behavioral cloning. During the online phase, we retrieve optimal contact plans from the

*Equal contribution
Corresponding author: ante.maric@idiap.ch

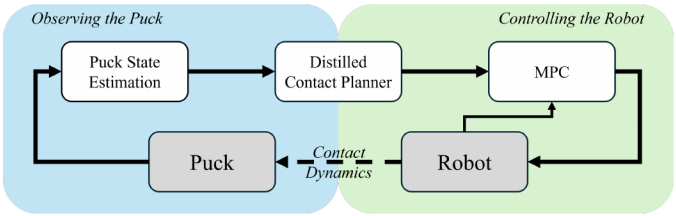

Fig. 2. Overview of the interplay between puck state estimation • and robot control • for closed-loop agile robot air hockey. The contact planner uses the estimated puck state to predict the puck trajectory based on the learned model. It subsequently plans a shooting angle that is used to construct an optimal control objective solved within a model-predictive controller. Robot trajectories are computed at a control rate of 50 Hz.

distilled policy using derivative-free inference in realtime. Finally, for the robot controller, we use a sampling-based model-predictive controller [12] that enforces the execution of the contact plan while respecting safety constraints such as collision avoidance with the walls. We summarize our key contributions as follows:

- We present an approach for learning the parameters of a stochastic model for discontinuous contact dynamics in robot air hockey as a mixture of linear-Gaussian modes.
- We formulate contact planning for robot air hockey as a chance-constrained stochastic optimal control problem.
- We propose an approach for distilling an optimal contact policy by training an implicit model to allow for planning in real-time.

After presenting our technical contributions, we provide experimental comparisons to control-based and reinforcement learning baselines. Our approach has furthermore outperformed all other approaches tested in a competitive setting [21].

II. Related Work

The air hockey task has been part of the robotics literature for a long time [4]. One of the first works using the air hockey task as a benchmark focused on skill learning of a humanoid robot [2, 3]. In more recent years, this benchmark has been used in combination with planar robots due to high-speed motion requirements [23, 27, 11, 28], and the possibility of adapting the playing style against the opponent [10]. This benchmark has been recently extended to the cobot setting, where a 7-DoF robotic arm controls the mallet and maintains the table surface while striking [1, 18, 7].

Another use of the robot air hockey setting is as a testbed for learning algorithms. In [29], deep reinforcement learning techniques are used to learn on planar robots, while in [19], both the planar 3-DoF and the 7-DoF cobot air hockey tasks are used to learn control policies in simulation. More recent techniques directly use the real 7-DoF air hockey setting as a testbed for learning algorithms: in [15], the authors use learning-to-plan techniques to generate air hockey hitting trajectories in a real-world setting, while in [20], this task is used to perform real-world reinforcement learning.

In general, existing solutions to the robot air hockey problem can be categorized in two main directions: learning-based approaches [3, 29, 20], and control-based approaches [28, 1, 18]. Generally speaking, pure control-based approaches lead to better and faster solutions than learning-based methods but require considerable efforts in engineering and model identification, and are particularly challenging to implement and run at realtime control rates. Instead, pure learning-based approaches obtain a lower-quality solution but make it possible to obtain more robust behaviors by relying on domain randomization and fine-tuning on the real platform. We aim to combine the advantages of learning-based and control-based approaches. We exploit both the optimality of control-based approaches for controlling the robot without considering the puck and the robustness and flexibility of learning-based approaches to efficiently generate plans for the contact between the robot and the puck to maximize the scoring probability.

III. Learning a Stochastic Contact Model

Planning and controlling the contacts of the robot with the puck requires the anticipation of puck trajectories before and after contact. To enable this, we learn a simplified stochastic model of the puck dynamics for *i)* estimating the current state of the puck online, *ii)* predicting the trajectory of the puck online, and *iii)* solving a stochastic optimal control problem to plan the next best contact between the robot and the puck.

A. Mixture of linear-Gaussian Contact Dynamics

Suppose that $\boldsymbol{x}_k^p \in \mathbb{R}^2$ is the position of the puck w.r.t. the surface of the air hockey table at time step k. The robot interacts with the puck by making contact with its mallet, i.e. the circular part of the robot's end-effector. The position of the mallet is denoted with $\boldsymbol{x}_k^m \in \mathbb{R}^2$ w.r.t. the surface of the air hockey table. We assume that the robot arm is controlled such that the mallet maintains contact with the table at all times. In order to efficiently perform rollouts of the puck dynamics, we impose a piecewise-linear structure on the model. Fig. 3 illustrates the three modes that we present in the following: *1) Floating*, *2) Puck-Wall Collision*, and *3) Puck-Mallet Collision*. To account for modeling errors introduced through the piecewise-linear structure, we model each mode as a conditional Gaussian distribution, resulting in a mixture of linear-Gaussian contact dynamics. In the following, we present the individual modes and their respective parameters that are learned subsequently.

1) Floating: The first mode captures the dynamics of the puck when it is freely floating on the table and is not in collision with the wall or mallet. The prediction of the puck velocity is modeled stochastically with

$$\Pr_1\left(\dot{\boldsymbol{x}}_{k+1}^p | \dot{\boldsymbol{x}}_k^p\right) = \mathcal{N}\left(\boldsymbol{\Theta}_1 \dot{\boldsymbol{x}}_k^p + \boldsymbol{\theta}_1, \boldsymbol{\Sigma}_1\right), \tag{1}$$

where $\boldsymbol{\Theta}_1, \boldsymbol{\theta}_1, \boldsymbol{\Sigma}_1$ are parameters of the conditional Gaussian distribution.

2) Puck-Wall Collision: The second mode models the dynamics of the puck reflecting against the wall. The prediction of the velocity is modeled in a coordinate system $\mathcal{C}$ that is

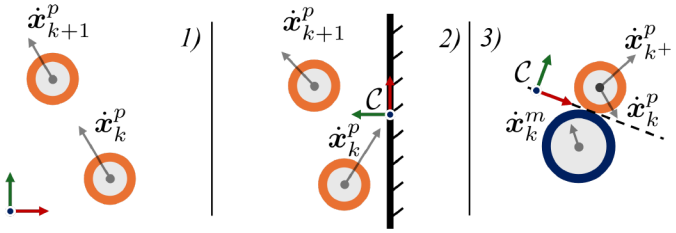

Fig. 3. Illustration of three modes of the puck dynamics that are parameterized as linear-Gaussian modes. Mode *1)* captures the dynamics of the puck ● when floating on the surface of the table. Mode *2)* captures collisions between puck and walls. Mode *3)* models collisions between puck and the mallet ● in a contact-aligned frame $\mathcal{C}$. The parameters for the nominal dynamics and the corresponding uncertainty are learned from data.

aligned with the contact surface of the corresponding wall. The one-step prediction of the puck velocity is modeled with

$$\Pr_2\left({}^c\dot{\boldsymbol{x}}^p_{k+1}|{}^c\dot{\boldsymbol{x}}^p_k\right) = \mathcal{N}\left(\boldsymbol{\Theta}_2\,{}^c\dot{\boldsymbol{x}}^p_k + \boldsymbol{\theta}_2, \boldsymbol{\Sigma}_2\right), \tag{2}$$

where ${}^c\dot{\boldsymbol{x}}^p$ is the puck velocity in the contact-aligned coordinate system. $\boldsymbol{\Theta}_2, \boldsymbol{\theta}_2, \boldsymbol{\Sigma}_2$ are parameters of this mode.

3) Puck-Mallet Collision: As a third mode, we model the interaction between the puck and the mallet as a collision in which the velocity of the puck changes instantaneously at the time of contact. We also model this mode using a conditional Gaussian distribution

$$\Pr_3\left({}^c\dot{\boldsymbol{x}}^p_{k^+}|{}^c\dot{\boldsymbol{x}}^p_{k^-}, {}^c\dot{\boldsymbol{x}}^m_k\right) = \mathcal{N}\left(\boldsymbol{\Theta}^{pc}_3\,{}^c\dot{\boldsymbol{x}}^p_{k^-} + \boldsymbol{\Theta}^{mc}_3\,{}^c\dot{\boldsymbol{x}}^m_k + \boldsymbol{\theta}_3, \boldsymbol{\Sigma}_3\right). \tag{3}$$

The velocities of the puck ${}^c\dot{\boldsymbol{x}}^p$ and of the mallet ${}^c\dot{\boldsymbol{x}}^m$, respectively, are expressed in the contact-aligned coordinate system $\mathcal{C}$. The index k^+ corresponds to time step k after applying the collision model, while k^- describes the instant right before the collision. The model parameters for the third mode are $\boldsymbol{\Theta}^p_3, \boldsymbol{\Theta}^m_3, \boldsymbol{\theta}_3, \boldsymbol{\Sigma}_3$.

B. Learning Model Parameters from Data

Given recorded trajectories of the puck and the mallet, the data is fragmented into consecutive puck velocity pairs together with the mallet velocity, i.e. $\dot{\boldsymbol{x}}^p_k, \dot{\boldsymbol{x}}^p_{k+1}, \dot{\boldsymbol{x}}^m_k$, and the corresponding mode is assigned to each data sample. As a result, we assume to obtain a dataset $\{\boldsymbol{y}_{i,n}, \boldsymbol{\xi}_{i,n}\}_{n=0}^{N_i}$ for each mode i, where $\boldsymbol{y}_{i,n}$ is the n-th velocity prediction sample for mode i, e.g. $\boldsymbol{y}_1 = \dot{\boldsymbol{x}}^p_{k+1}$, and $\boldsymbol{\xi}_{i,n}$ is the n-th prediction condition sample for mode i, e.g. $\boldsymbol{\xi}_1 = \dot{\boldsymbol{x}}^p_k$. To learn the parameters of the model, we fit a Gaussian distribution to the dataset for each mode modeling the joint probability distribution of prediction and condition with

$$\Pr_i(\boldsymbol{y}_i, \boldsymbol{\xi}_i) = \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_{\boldsymbol{y}_i}\\ \boldsymbol{\mu}_{\boldsymbol{\xi}_i}\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_{\boldsymbol{y}_i} & \boldsymbol{\Sigma}_{\boldsymbol{y}_i\boldsymbol{\xi}_i}\\ \boldsymbol{\Sigma}^\top_{\boldsymbol{y}_i\boldsymbol{\xi}_i} & \boldsymbol{\Sigma}_{\boldsymbol{\xi}_i}\end{pmatrix}\right). \tag{4}$$

Given the parameters of the joint probability distribution, the parameters of the linear-Gaussian models can be computed by conditioning the probability distribution on the input $\boldsymbol{\xi}$. Thus, the parameters are given by

$$\begin{aligned}\boldsymbol{\Theta}_i &= \boldsymbol{\Sigma}_{\boldsymbol{y}_i\boldsymbol{\xi}_i}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_i},\\ \boldsymbol{\theta}_i &= \boldsymbol{\mu}_{\boldsymbol{y}_i} - \boldsymbol{\Sigma}_{\boldsymbol{y}_i\boldsymbol{\xi}_i}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_i}\boldsymbol{\mu}_{\boldsymbol{\xi}_i},\\ \boldsymbol{\Sigma}_i &= \boldsymbol{\Sigma}_{\boldsymbol{y}_i} - \boldsymbol{\Sigma}_{\boldsymbol{y}_i\boldsymbol{\xi}_i}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_i}\boldsymbol{\Sigma}^\top_{\boldsymbol{y}_i\boldsymbol{\xi}_i}.\end{aligned} \tag{5}$$

C. Piecewise-linear Kalman Filtering

The learned linear-Gaussian models allow us to update the estimated state of the puck using the Kalman filter. As a result, an estimate of the puck state at time step k, i.e. $\hat{\boldsymbol{s}}_k = \left(\hat{\boldsymbol{x}}^{p\top}_k, \hat{\dot{\boldsymbol{x}}}^{p\top}_k\right)^\top$, is obtained based on a noisy measurement of the puck position $\tilde{\boldsymbol{x}}^p_k$. For this, the mode of the dynamics is detected at each time step such that the corresponding parameters are used within the Kalman filter update. The parameters are translated into linear-Gaussian state-space dynamics, i.e.

$$\Pr_i\left(\boldsymbol{s}_{k+1}|\boldsymbol{s}_k\right) = \mathcal{N}\left(\boldsymbol{A}_i\boldsymbol{s}_k + \boldsymbol{b}_i, \boldsymbol{Q}_i\right), \tag{6}$$

with system parameters $\boldsymbol{A}_i, \boldsymbol{b}_i$ and process noise covariance matrix $\boldsymbol{Q}_i$ computed with

$$\boldsymbol{A}_i = \begin{pmatrix}\boldsymbol{A}_{i,\boldsymbol{x}\boldsymbol{x}} & \boldsymbol{A}_{i,\boldsymbol{x}\dot{\boldsymbol{x}}}\\ \boldsymbol{0} & \boldsymbol{\Theta}_i\end{pmatrix};\ \boldsymbol{b}_i = \begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{\theta}_i\end{pmatrix};\ \boldsymbol{Q}_i = \begin{pmatrix}\boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Sigma}_i\end{pmatrix}. \tag{7}$$

Here, the model parameters $\boldsymbol{A}_{i,\boldsymbol{x}\boldsymbol{x}}, \boldsymbol{A}_{i,\boldsymbol{x}\dot{\boldsymbol{x}}}$ determine the prediction of the puck position at the next time step given the current puck position and velocity. These parameters are derived using numerical integration and are constant.

D. Probability of Hitting the Goal

For the robot to anticipate whether a candidate shot may lead to scoring a goal, we predict the probability of hitting the goal based on the learned linear-Gaussian puck dynamics. Note that the probability of hitting the goal does not account for a defending opponent. Without loss of generality, suppose that the collision between mallet and puck happens at $k = 0$. Given the puck state at the time of collision $\hat{\boldsymbol{s}}_{0^-}$ and the corresponding mallet state $\boldsymbol{x}^m_0, \dot{\boldsymbol{x}}^m_0$, the expected puck velocity after the collision is computed as defined in (3), resulting in the expected puck state $\hat{\boldsymbol{s}}_{0^+}$. By rolling out the discretized stochastic model with

$$\begin{aligned}\hat{\boldsymbol{s}}_{k+1} &= \boldsymbol{A}_{i_k}\hat{\boldsymbol{s}}_k + \boldsymbol{b}_{i_k},\\ \boldsymbol{P}_{k+1} &= \boldsymbol{A}_{i_k}\boldsymbol{P}_k\boldsymbol{A}^\top_{i_k} + \boldsymbol{Q}_{i_k},\end{aligned} \tag{8}$$

a Gaussian distribution of puck states, i.e. $\boldsymbol{s}_k \sim \mathcal{N}(\hat{\boldsymbol{s}}_k, \boldsymbol{P}_k)$ is obtained for each time step $k > 0$. The rollout is initialized with $\hat{\boldsymbol{s}}_0 = \hat{\boldsymbol{s}}_{0^+}$ and $\boldsymbol{P}_0 = \boldsymbol{Q}_3$, exploiting the separated stochastic model of collisions between mallet and puck.

To evaluate the probability of scoring a goal, we perform the stochastic rollout as defined in (8) until the expected puck position $\hat{\boldsymbol{x}}^p_k$ crosses the goal line. We denote this time step with k_{goal}. In the following, we denote the probability of scoring a goal, i.e. $\mathrm{G} = 1$, given a puck position as a Bernoulli distribution with

$$\Pr\left(\mathrm{G} = 1|\boldsymbol{x}^p_k\right) = \begin{cases}1, & \text{if } \boldsymbol{x}^p_k \in \mathcal{X}_{\text{goal}}\\ 0, & \text{else.}\end{cases} \tag{9}$$

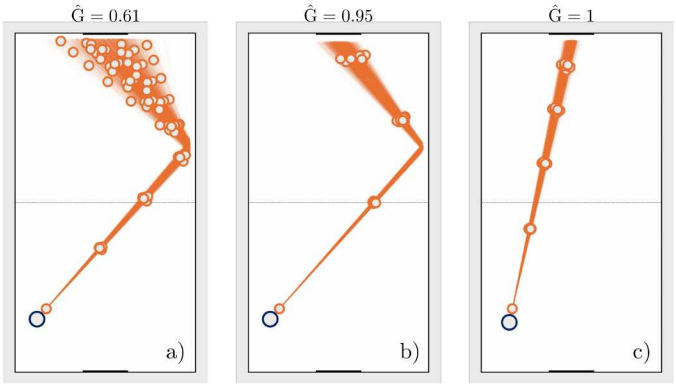

Fig. 4. A qualitative comparison of the probability of hitting the goal $\hat{\mathrm{G}}$ for different shooting angles and shooting speeds. The shooting angles are indicated by the mallet position • w.r.t. the puck position • at the time of contact. The shooting speed, i.e. the speed of the mallet at the time of contact, is 1.2 $\frac{\mathrm{m}}{\mathrm{s}}$ for a) and c), while the shooting speed is 2 $\frac{\mathrm{m}}{\mathrm{s}}$ for b).

The subset in puck position space $\mathcal{X}_{\text{goal}}$ represents the goal region. Consequently, we can compute the probability of scoring a goal given the initial conditions of a shot by marginalizing over the puck position at time step k_{goal} with

$$\Pr\left(\mathrm{G}=1|\hat{\boldsymbol{s}}_{0-}, \boldsymbol{x}_0^m, \dot{\boldsymbol{x}}_0^m\right) = \int_{\mathcal{X}_{\text{goal}}} \Pr\left(\boldsymbol{x}_{k_{\text{goal}}}^p\right) d\boldsymbol{x}_{k_{\text{goal}}}^p. \quad (10)$$

We compute the probability in (10) using Monte-Carlo approximation by sampling N_{G} puck positions from the Gaussian distribution at prediction time step k_{goal} and counting the number of samples that would hit the goal

$$\Pr\left(\mathrm{G}=1|\hat{\boldsymbol{s}}_{0-}, \boldsymbol{x}_0^m, \dot{\boldsymbol{x}}_0^m\right) \approx \frac{1}{N_{\mathrm{G}}} \sum_{n=1}^{N_{\mathrm{G}}} \Pr\left(\mathrm{G}=1|\boldsymbol{x}_{k_{\text{goal}},n}^p\right), \quad (11)$$

with $\boldsymbol{x}_{k_{\text{goal}},n}^p \sim \Pr(\boldsymbol{x}_{k_{\text{goal}}}^p)$. In the following, we denote the approximated probability of hitting the goal, corresponding to the right-hand side of (11), with $\hat{\mathrm{G}}$.

Fig. 4 illustrates stochastic rollouts for various initial conditions of a shot. Evaluating $\hat{\mathrm{G}}$ as defined in (11), we observe that those initial conditions have a significant effect even if the expected puck trajectory hits the center of the goal for all conditions. Fast shots (Fig. 4-b) accumulate less uncertainty compared to slow shots (Fig. 4-a) since the modeled process noise is constant over time. Direct shots accumulate less uncertainty during rollout than bank shots, as collisions with a wall add significant process noise (Fig. 4-c).

IV. Implicit Contact Planning under Uncertainty

The learned dynamics model enables the prediction of uncertain puck trajectories for contact planning. In particular, we search for contact states of the mallet that result in desired puck trajectories after contact. The proposed contact planning module is based on stochastic optimal control, optimizing the mallet's state at contact. We combine this optimization with a model-based robot controller driving the robot to the desired contact state at the desired time (cf. Fig. 2).

A. Stochastic Optimal Control for Shooting

Given the desired time of contact and the corresponding estimate of the puck state $\boldsymbol{s}_{0-}$ at that time, we pose contact planning for shooting as a stochastic optimal control problem searching for the mallet state $\boldsymbol{x}_0^m, \dot{\boldsymbol{x}}_0^m$ at the time of contact. For this, we aim to maximize a tradeoff between the probability of hitting the goal $\hat{\mathrm{G}}$ and the expected puck speed v_{puck} at the goal line. The expected puck speed is computed as the norm of the mean puck velocity at k_{goal} according to Sec. III-D. While the probability of hitting the goal $\hat{\mathrm{G}}$ does not account for a defending opponent, we use the speed of the puck as a measure of the difficulty of defending against the shot. The stochastic optimal control problem is given as

$$\begin{aligned} \max_{\boldsymbol{x}_0^m, \dot{\boldsymbol{x}}_0^m} \quad & \lambda_1 \hat{\mathrm{G}} + \lambda_2 v_{\text{puck}} \\ \text{s.t.} \quad & \hat{\mathrm{G}} > \beta, \end{aligned} \quad (12)$$

where we deploy an additional chance constraint to enforce the probability of hitting the goal to be higher than a threshold β based on the learned stochastic model. The weights λ_1 and λ_2 are used for tuning for the desired behavior. Based on the qualitative comparison illustrated in Fig. 4, we expect that solely optimizing for the probability of hitting the goal results only in direct shots, as bank shots induce uncertainty. Yet, due to the kinematics of the robot, the puck speed may be increased with bank shots. Thus, depending on the puck state, the tuned objective can produce both straight shots and bank shots, increasing the chances of scoring.

B. Shooting Angle as Reduced Action Space

The goal of the shooting policy is to find the optimal mallet state at the time of contact, i.e. $\boldsymbol{x}_0^m$ and $\dot{\boldsymbol{x}}_0^m$, respectively. Due to the underlying contact geometry and constraints, for a given puck position $\hat{\boldsymbol{x}}_0^p$ we parameterize the mallet position as a shooting angle $u \in \mathcal{U}$ between the mallet and puck. Note that a shooting angle of $u = 0$ corresponds to a straight shot that is parallel to the side walls of the table. We reduce the dimensionality of the action space further by imposing two heuristic constraints on the mallet velocity $\dot{\boldsymbol{x}}_0^m$: *i)* The mallet velocity at the time of contact aligns with the shooting angle, such that $\dot{\boldsymbol{x}}_0^m = v(\cos u, \sin u)^\top$ with scalar velocity $v > 0$ encoding the norm of the mallet velocity. While this constraint excludes shooting angles that are not aligned with the mallet velocity, it enforces maximum transmission of kinetic energy from the robot to the puck. *ii)* We impose that the norm of the mallet velocity is maximal given a shooting configuration $\boldsymbol{q}_0$ of the robot and velocity limits $\dot{\mathcal{Q}}$ of the joints of the robot, such that

$$\begin{aligned} v^* = & \max v \\ \text{s.t.} \quad & v\boldsymbol{e}_u = \boldsymbol{J}(\boldsymbol{q}_0)\dot{\boldsymbol{q}}_0, \\ & \dot{\boldsymbol{q}}_0 \in \dot{\mathcal{Q}}. \end{aligned} \quad (13)$$

Note that the unit vector $\boldsymbol{e}_u \in \mathbb{R}^3$ encodes the shooting direction including zero contribution in the z-direction. Accordingly, $\boldsymbol{J}(\boldsymbol{q}_0) \in \mathbb{R}^{3 \times n_{\text{dof}}}$ corresponds to the Jacobian w.r.t. the Cartesian position of the mallet.

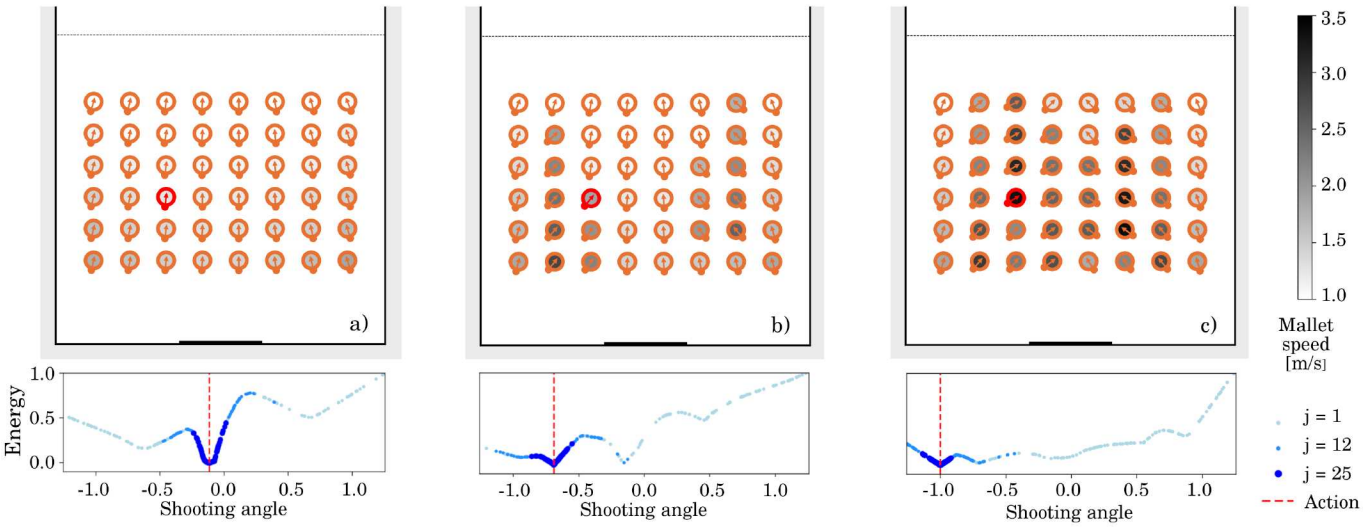

Fig. 5. Examples of differently tuned shooting plans and corresponding energy landscapes. Shooting direction and mallet speed are displayed for varying initial puck positions. Instance a) evaluates only scoring probability ($\lambda_1 = 1,\ \lambda_2 = 0,\ \beta = 0.5$); b) adds additional weight on expected puck speed at the goal line ($\lambda_1 = 1,\ \lambda_2 = 0.2,\ \beta = 0.5$); c) evaluates only the expected puck speed at the goal line ($\lambda_1 = 0,\ \lambda_2 = 1,\ \beta = 0.5$). The energy landscape and sampling process at timesteps $j \in \{1, 12, 25\}$ are visualized for an example shot denoted in red. All pucks are static at $j = 0$.

As a result, the shooting angle u is the action that we optimize for. With the imposed constraints, a shooting angle uniquely maps to a mallet state at the time of contact. Thus, in the following, we denote the probability of scoring a goal as a function of the shooting angle with $\Pr\left(\mathrm{G} = 1 | \hat{\boldsymbol{s}}_{0^-}, u\right)$.

C. Distilling an Optimal Contact Policy

The long-horizon predictions required to plan shooting actions make it difficult to operate at a rate that is sufficient for agile behavior. We address this challenge by training an implicit model to generate solutions to the stochastic optimal shooting problem (12) in realtime. Since the stochastic policy we aim to capture is multimodal, we opt for an implicit policy representation due to its ability to learn multimodal action distributions [8, 6]. In the case of air hockey, example modes correspond to the number of reflections against the bank of the table, ranging from straight shots without reflections to bank shots with multiple reflections. To learn the implicit policy, we use an energy-based model (EBM) [8]. Access to the learned energy[1] landscape allows us to accommodate various sampling strategies without retraining, enabling a balance between optimality and computational efficiency.

We train the EBM by solving the computationally expensive shooting angle optimization offline and using the results as training data. Namely, we first generate a dataset of shooting angles for N different puck states at time of contact $\{\boldsymbol{s}_i\}_{i=1}^N$. Due to the one-dimensional parametrization of the action space introduced in Sec. IV-B, we efficiently explore the space of shooting angles for each initial puck state by sampling M candidate shooting angles $\{u_i^j\}_{j=1}^M$. Subsequently, we compute the stochastic rollout of the puck trajectory as presented in Sec. III-D and evaluate the objective and chance constraint from (12). The best-performing sample $\hat{u}_i$ is then used as a positive example for training the implicit behavioral cloning model, with the remaining $M-1$ samples as negative counter-examples. This results in a dataset of $M \times N$ state-action pairs $\{\boldsymbol{s}_i, \hat{u}_i, \{u_i^j\}_{j=1}^{M-1}\}_{i=1}^N$, which we use to train the EBM $E_\theta(\boldsymbol{s}, u)$ using an InfoNCE-style [30] loss

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^{N} -\log\left(\tilde{p}_\theta\left(\hat{u}_i | \boldsymbol{s}_i, \{u_i^j\}_{j=1}^{M-1}\right)\right). \tag{14}$$

In the above, counter-examples $\{u_i^j\}_{j=1}^{M-1}$ are used to compute the likelihood $\tilde{p}_\theta$ as follows:

$$\tilde{p}_\theta\left(\hat{u}_i | \boldsymbol{s}_i, \{u_j^i\}_{j=1}^{M-1}\right) = \frac{e^{-E_\theta(\boldsymbol{s}_i, \hat{u}_i)}}{e^{-E_\theta(\boldsymbol{s}_i, \hat{u}_i)} + \sum_{j=1}^{M-1} e^{-E_\theta(\boldsymbol{s}_i, u_j^i)}}. \tag{15}$$

The described loss function reduces energy $E_\theta(\boldsymbol{s}, u)$ for shooting angles that solve the optimization problem in (12), while increasing the energy of non-optimal shooting angles. Once the model is trained, this allows us to infer optimal shooting angles using sampling-based optimization.

D. Online Inference with Warm-Starting

To solve (12) given the estimated puck state $\hat{\boldsymbol{s}}_{0^-}$, we search for a state-action pair that minimizes the learned energy, i.e.

$$\hat{u} = \arg\min_{u \in \mathcal{U}} E_\theta(\hat{\boldsymbol{s}}_{0^-}, u). \tag{16}$$

For realtime optimization, we leverage direct access to the learned energy landscape of the EBM by executing sampling iterations concurrently with other components of the control loop. This allows us to refine contact plans as the robot executes trajectories, while considering only the shrinking

[1]In this paper, the term *energy* refers to the negative logarithm of the unnormalized probability density function that we want to model.

Algorithm 1: Shooting policy (EBM inference)

Input: Puck state $\hat{s}_{0^-}$, variance σ, samples $\{\tilde{p}_i, \tilde{u}_i\}_{i=1}^N$
Output: Shooting angle $\hat{u}$, new samples $\{\tilde{p}_i, \tilde{u}_i\}_{i=1}^N$

$\{\tilde{u}_i\}_{i=1}^N \leftarrow\sim$ Multinomial$(N, \{\tilde{p}_i\}_{i=1}^N, \{\tilde{u}_i\}_{i=1}^N)$
$\{\tilde{u}_i\}_{i=1}^N \leftarrow \{\tilde{u}_i\}_{i=1}^N + \sim \mathcal{N}(0, \sigma)$
$\{\tilde{u}_i\}_{i=1}^N \leftarrow$ clip $\{\tilde{u}_i\}_{i=1}^N$ to $\mathcal{U}$
$\{E_i\}_{i=1}^N \leftarrow \{E_\theta(\hat{s}_{0^-}, \tilde{u}_i)\}_{i=1}^N$
$\{\tilde{p}_i\}_{i=1}^N \leftarrow$ softmax$(-\{E_i\}_{i=1}^N)$
$\hat{u} \leftarrow$ argmax$(\{\tilde{p}_i\}, \{\tilde{u}_i\})$

horizon from the current timestep to the time of contact. We base the online retrieval of optimal shooting angles on derivative-free optimization procedures from [8]. As in the offline scenario, shooting actions are generated online by uniformly sampling N candidate actions, inferring their energy values, and resampling with replacement to warm-start optimization in each subsequent timestep. To converge towards a solution with minimum implicit energy, reductions to the sampling variance are applied at each timestep, while keeping the optimal contact angle $\hat{u}$ as a reference for the mid-level trajectory planner. A full iteration of EBM inference is outlined in Alg. 1. We observe that the learned energy models and utilized optimization procedure efficiently retrieve multimodal contact plans to produce desired behaviors, as shown in Fig. 5.

V. Experimental Evaluation

This section details the simulated and real-world experiments used to validate our approach in an online contact planning setting. We evaluate the shooting performance of a robot arm controlled by our framework and compare it against state-of-the-art approaches for robot air hockey.

A. Implementation Details

1) Data Collection for Puck Dynamics: We use data collected in a physics-based simulator to learn model parameters of puck dynamics as presented in Sec. III. One set of data is collected by randomly moving the robot's end-effector into contact with the puck and the other set of data is collected without moving the robot and by initializing the puck with a high random velocity. In total, the training set consists of 100 episodes with 50 time steps each, which corresponds to a total of 100 seconds of observations of the puck dynamics.

2) EBM Architecture and Training: The energy-based shooting model consists of a multilayer perceptron with 2 hidden, fully connected layers of 128 neurons each. The model is trained on $N = 3000$ initial puck states, with $M = 100$ action samples. For each action sample, the corresponding state action pair is evaluated using the stochastic dynamics model learned from simulated data (Sec. III), and the optimal control cost (Sec. IV). Training required 500 epochs to converge for satisfactory performance using the Adam [16] optimizer with a decaying learning rate.

Fig. 6. The automated experimental setup consists of a robot placing the puck at a pre-defined grid of positions (left image). After releasing the puck, the KUKA robot executes the shooting policy (right image). We measure the speed of the puck at the goal line and whether the puck hit the goal as quality metrics.

B. Experimental Setup

The experiment is conducted with a KUKA iiwa14 LBR manipulator equipped with a mallet end-effector that is attached to a passive joint for seamless contact with the table surface. Experiments are carried out in a simulated *MuJoCo* environment and on a real-world setup (cf. Fig. 6) for evaluation of *sim2real* transfer. We evaluate three instances of the proposed approach by using different parameters for the chance-constrained optimization problem in (12). **Ours #1**: a *conservative* policy that prioritizes accuracy ($\lambda_1 = 1, \lambda_2 = 0, \beta = 0.5$); **Ours #2**: a *balanced* policy that compromises between accuracy and puck speed ($\lambda_1 = 1, \lambda_2 = 0.2, \beta = 0.5$); and **Ours #3**: an *aggressive* policy that prioritizes puck speed ($\lambda_1 = 0, \lambda_2 = 1, \beta = 0.5$). Simulated results are also illustrated in Fig. 5 for initial puck velocities of zero. We compare the three instances of our contact planner with: **CB**, a baseline that utilizes conventional planning and control methods [18]; and **ATACOM**, a safe reinforcement learning approach for learning a robot policy [19]. Note that the ATACOM policy is trained in simulation and deployed in the physical experiment without additional tuning or retraining.

We perform 100 shots with each policy and report the accuracy score, puck speed at the goal line, and the number of bank reflections for successful shots. Each shot is initialized by placing the puck within a grid in front of the robot. Due to imperfect air flow on the air hockey table, the puck moves after release, requiring the robot to adapt for a good shot.

C. Results

Recorded metrics are reported in Table III for the simulated environment and in Table II for the real-world environment. Compared to **CB** and **ATACOM**, we observe that our framework is capable of achieving higher scoring accuracy and significantly higher puck speeds in both environments. It can be seen that different instances of our policy obtain either a high score or high puck speeds according to the corresponding parameters of the stochastic optimal control problem. For example, when compared to **Ours #1**, it can be seen that **Ours #3** compromises scoring accuracy for faster puck speeds and a high number of bank reflections, potentially making the shots more difficult to defend against. The higher number of bank

TABLE I
RESULTS OF THE SIMULATED EXPERIMENTS.

	Score	Puck Speed $\left[\frac{\text{m}}{\text{s}}\right]$ (mean $\pm$ std.)	Num. Banks (mean)
CB	0.51	0.52 ± 0.24	0.00
Atacom	0.90	0.55 ± 0.05	0.00
Ours #1	**0.93**	1.00 ± 0.20	0.00
Ours #2	0.80	1.44 ± 0.63	0.53
Ours #3	0.61	$\mathbf{1.97 \pm 0.49}$	**1.13**

TABLE II
RESULTS OF THE REAL-WORLD EXPERIMENTS.

	Score	Puck Speed $\left[\frac{\text{m}}{\text{s}}\right]$ (mean $\pm$ std.)	Num. Banks (mean)
CB	0.49	1.09 ± 0.24	0.00
Atacom	0.13	0.66 ± 0.15	0.31
Ours #1	**0.78**	1.72 ± 0.20	0.00
Ours #2	0.60	2.02 ± 0.35	0.37
Ours #3	0.31	$\mathbf{2.37 \pm 0.50}$	**0.90**

reflections produced by **Ours #3** indicates that the robot kinematics allow for higher shooting speeds when hitting laterally, at the risk of missing the goal due to uncertainty gained with every bank reflection. Note that the score of **Ours #3** is lower than the score chance threshold $\beta = 0.5$ for this instance. This indicates that the learned model either has an error in the nominal dynamics or expects too little uncertainty gain due to bank reflections. We further note a decrease in performance for all agents due to the *sim2real* gap, with **ATACOM** showing the highest sensitivity to transfer as it requires fine-tuning on the real environment. **CB** shows the least decrease in performance, as it is parameterized for the real system. However, note that the shooting trajectory optimization loop of **CB** is slower than required to run at 50 Hz, making it prone to errors due to the puck moving unpredictably during the shooting motion. Since we combine closed-loop model-based control with distilled contact planning, our agents display robustness to *sim2real* transfer. Additionally, we note higher puck speeds in the real setting for all agents as a result of differences in real and simulated contact dynamics. Example physical shots of all approaches can be found in the supplementary video.

D. Ablation Studies

Ablation study on reduced action space. We conduct an ablation study on the impact of learning reduced actions, i.e. learning shooting angles (1 DoF) as described in Sec. IV-B, compared to learning full actions, i.e. the mallet state at impact (4 DoF). Table III reports numerical results on the shooting performance using the same number of demonstrations for both modes. It shows that our online inference algorithm with warm-starting also works with higher-dimensional action spaces. However, a higher-dimensional action space requires more data to achieve similar performance.

Ablation study on chance constraint. We furthermore conduct a study on the impact of the threshold β, which constrains the likelihood of hitting the goal as described in Sec. IV-A. Fig. 7 illustrates how the shooting behavior changes for various β when optimizing for puck speed (λ_1=0, λ_2=1). The score increases with β, while the main sources of uncertainty, puck speed and number of banks, are reduced with an increasing β.

VI. LIMITATIONS

Several contributing factors enable the highly performative behavior displayed by our robot air hockey agent. Firstly, the dimension of the task space is low, as the puck is constrained to a plane. While this makes our results relevant for many pushing tasks, most real-world manipulation tasks involve higher-dimensional task spaces. Along the same line, we were able to leverage existing physical models as a strong prior on contact dynamics. Such models are not available for tasks that involve more complex contact dynamics. While our approach enables learning on real-world data, we exploited a physics engine to collect data for learning a compact contact model.

VII. CONCLUSION

This paper investigated the combination of learning-based contact planning with model-predictive robot control to produce agile behavior in Robot Air Hockey. We show that distilling an optimal contact planning policy through behavior cloning effectively reduces the horizon required for lower-level trajectory optimization, enabling real-time operation. We show that the proposed approach is capable of accommodating different desired behaviors and sampling-based optimization schemes. Our results show that the proposed framework outperforms a purely control-based approach and a purely learning-based approach in simulated and real-world games of robot air hockey. Future work will seek to further leverage the sample efficiency of structured dynamics models to capture the underlying contact dynamics of physical systems from real data. Additionally, integrating physically-informed priors into the implicit model is an interesting direction for increasing the data efficiency of our approach. Lastly, we are interested in investigating the applicability of our approach to higher-dimensional task spaces with more complex contact interactions.

ACKNOWLEDGMENTS

This work was supported by the Swiss National Science Foundation (SNSF) through the CODIMAN project, by the State Secretariat for Education, Research and Innovation in Switzerland for participation in the European Commission's Horizon Europe Program through the INTELLIMAN project (https://intelliman-project.eu/, HORIZON-CL4-Digital-Emerging Grant 101070136) and the SESTOSENSO project (http://sestosenso.eu/, HORIZON-CL4-Digital-Emerging Grant 101070310), by the China Scholarship Council (Grant 201908080039) and by the German Federal Ministry of Education and Research (BMBF) through the KIARA project (Grant 13N16274).

REFERENCES

[1] Ahmad AlAttar, Louis Rouillard, and Petar Kormushev. Autonomous air-hockey playing cobot using op-

TABLE III
REDUCED ACTION SPACE (R. ACT.) V. FULL ACTION SPACE (F. ACT.).

	Score	Puck Speed $\left[\frac{\text{m}}{\text{s}}\right]$ (mean $\pm$ std.)	Num. Banks (mean)
R. Act.	**0.80**	**1.44 $\pm$ 0.63**	**0.53**
F. Act.	0.51	0.96 $\pm$ 0.13	0.00

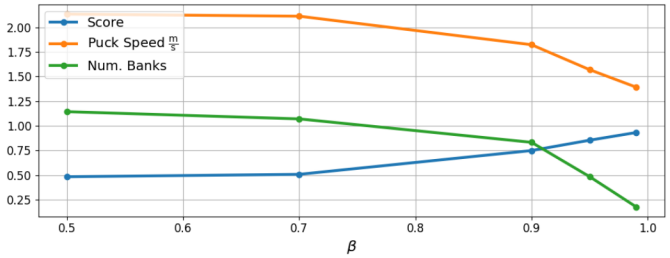

Fig. 7. Impact of the chance constraint threshold β on shooting performance.

timal control and vision-based bayesian tracking. In *International Conference Towards Autonomous Robotic Systems (TAROS)*, 2019. ISBN 9783030253318. doi: 10.1007/978-3-030-25332-5_31.

[2] Darrin C Bentivegna, Christopher G Atkeson, and Gordon Cheng. Learning tasks from observation and practice. *Robotics and Autonomous Systems*, 47(2-3):163–169, 2004.

[3] Darrin C Bentivegna, Christopher G Atkeson, Aleš Ude, and Gordon Cheng. Learning to act from observation and practice. *International Journal of Humanoid Robotics*, 1 (04):585–611, 2004.

[4] Bradley E Bishop and Mark W Spong. Vision based control of an air hockey playing robot. *IEEE Control Systems Magazine*, 19(3), 1999.

[5] Dieter Büchler, Simon Guist, Roberto Calandra, Vincent Berenz, Bernhard Schölkopf, and Jan Peters. Learning to play table tennis from scratch using muscular robots. *IEEE Transactions on Robotics*, 2022.

[6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. URL https://doi.org/10.15607/RSS.2023.XIX.026.

[7] Caleb Chuck, Carl Qi, Michael J Munje, Shuozhe Li, Max Rudolph, Chang Shi, Siddhant Agarwal, Harshit Sikchi, Abhinav Peri, Sarthak Dayal, et al. Robot air hockey: A manipulation testbed for robot learning with reinforcement learning. *arXiv preprint arXiv:2405.03113*, 2024.

[8] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 158–168, 2022.

[9] Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H Huang, Dhruva Tirumala, Jan Humplik, Markus Wulfmeier, Saran Tunyasuvunakool, Noah Y Siegel, Roland Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 9(89):eadi8022, 2024.

[10] Kazuki Igeta and Akio Namiki. A decision-making algorithm for an air-hockey robot that decides actions depending on its opponent player's motions. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1840–1845. IEEE, 2015.

[11] Kazuki Igeta and Akio Namiki. Algorithm for optimizing attack motions for air-hockey robot by two-step look ahead prediction. In *IEEE/SICE International Symposium on System Integration*, pages 465–470, 2017. ISBN 9781509033294. doi: 10.1109/SII.2016.7844042.

[12] Julius Jankowski, Lara Brudermüller, Nick Hawes, and Sylvain Calinon. Vp-sto: Via-point-based stochastic trajectory optimization for reactive robot behavior. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10125–10131, 2023. doi: 10.1109/ICRA48891.2023.10160214.

[13] Julius Jankowski, Lara Brudermüller, Nick Hawes, and Sylvain Calinon. Robust pushing: Exploiting quasi-static belief dynamics and contact-informed optimization. *arXiv preprint arXiv:2404.02795*, 2024.

[14] Mitsuo Kawato, Francesca Gandolfo, Hiroaki Gomi, and Yasuhiro Wada. Teaching by showing in kendama based on optimization principle. In *International Conference on Artificial Neural Networks*, pages 601–606. Springer, 1994.

[15] Piotr Kicki, Puze Liu, Davide Tateo, Haitham Bou-Ammar, Krzysztof Walas, Piotr Skrzypczyński, and Jan Peters. Fast kinodynamic planning on the constraint manifold with deep neural networks. *IEEE Transactions on Robotics*, 2023.

[16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[17] Jens Kober and Jan Peters. Policy search for motor primitives in robotics. *Advances in neural information processing systems*, 21, 2008.

[18] Puze Liu, Davide Tateo, Haitham Bou-Ammar, and Jan Peters. Efficient and reactive planning for high speed robot air hockey. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 586–593. IEEE, 2021.

[19] Puze Liu, Davide Tateo, Haitham Bou Ammar, and Jan Peters. Robot reinforcement learning on the constraint manifold. In *Conference on Robot Learning*, pages 1357–1366. PMLR, 2022.

[20] Puze Liu, Haitham Bou-Ammar, Jan Peters, and Davide Tateo. Safe reinforcement learning on the constraint manifold: Theory and applications. *arXiv preprint arXiv:2404.09080*, 2024.

[21] Puze Liu, Jonas Günster, Niklas Funk, Simon Gröger,

Dong Chen, Haitham Bou Ammar, Julius Jankowski, Ante Marić, Sylvain Calinon, Andrej Orsula, Miguel Olivares-Mendez, Hongyi Zhou, Rudolf Lioutikov, Gerhard Neumann, Amarildo Likmeta, Amirhossein Zhalehmehrabi, Thomas Bonenfant, Marcello Restelli, Davide Tateo, Ziyuan Liu, and Jan Peters. A retrospective on the robot air hockey challenge: Benchmarking robust, reliable, and safe learning techniques for real-world robotics. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=gPLE4siNjO.

[22] Katharina Mülling, Jens Kober, and Jan Peters. A biomimetic approach to robot table tennis. *Adaptive Behavior*, 19(5):359–376, 2011.

[23] Akio Namiki, Sakyo Matsushita, Takahiro Ozeki, and Kenzo Nonami. Hierarchical processing architecture for an air-hockey robot system. In *2013 IEEE International Conference on Robotics and Automation*, pages 1187–1192. IEEE, 2013.

[24] Tao Pang, H. J. Terry Suh, Lujie Yang, and Russ Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on Robotics*, 39(6):4691–4711, 2023. doi: 10.1109/TRO.2023.3300230.

[25] Kai Ploeger and Jan Peters. Controlling the cascade: Kinematic planning for n-ball toss juggling. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1139–1144. IEEE, 2022.

[26] Kai Ploeger, Michael Lutter, and Jan Peters. High acceleration reinforcement learning for real-world juggling with binary rewards. In *Conference on Robot Learning*, pages 642–653. PMLR, 2021.

[27] Hideaki Shimada, Yusuke Kutsuna, Shunsuke Kudoh, and Takashi Suehiro. A two-layer tactical system for an air-hockey-playing robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[28] Koichiro Tadokoro, Shotaro Fukuda, and Akio Namiki. Development of air hockey robot with high-speed vision and high-speed wrist. *Journal of Robotics and Mechatronics*, 34(5):956–964, 2022.

[29] A. Taitler and N. Shimkin. Learning control for air hockey striking using deep reinforcement learning. In *International Conference on Control, Artificial Intelligence, Robotics Optimization*, 2017. doi: 10.1109/ICCAIRO.2017.14.

[30] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018. URL https://api.semanticscholar.org/CorpusID:49670925.

[31] Felix von Drigalski, Devwrat Joshi, Takayuki Murooka, Kazutoshi Tanaka, Masashi Hamaya, and Yoshihisa Ijiri. An analytical diabolo model for robotic learning and control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4055–4061. IEEE, 2021.

[32] Zulfiqar Zaidi, Daniel Martin, Nathaniel Belles, Viacheslav Zakharov, Arjun Krishna, Kin Man Lee, Peter Wagstaff, Sumedh Naik, Matthew Sklar, Sugju Choi, et al. Athletic mobile manipulator system for robotic wheelchair tennis. *IEEE Robotics and Automation Letters*, 8(4):2245–2252, 2023.