AiRLIHockey: Highly Reactive Contact Control and Stochastic Optimal Shooting

Julius Jankowski[*]
Idiap Research Institute
Martigny, Switzerland
jjankowski@idiap.ch

Ante Marić[*]
Idiap Research Institute
Martigny, Switzerland
amaric@idiap.ch

Sylvain Calinon
Idiap Research Institute
Martigny, Switzerland
scalinon@idiap.ch

Abstract

Air hockey is a highly reactive game which requires the player to quickly reason over stochastic puck and contact dynamics. We implement a hierarchical framework which combines stochastic optimal control for planning shooting angles and sampling-based model-predictive control for continuously generating constrained mallet trajectories. Our agent was deployed and evaluated in simulation and on a physical setup as part of the *Robot Air-Hockey challenge* competition at NeurIPS 2023.

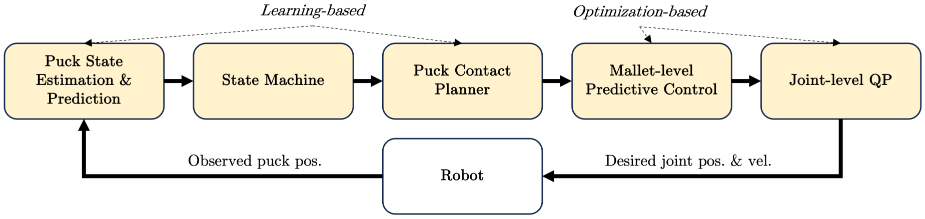

Figure 1: Diagram overview of the *AiRLIHockey* agent. Starting from noisy observations, we estimate the puck state subject to estimated model parameters. Depending on the state, a mode (e.g. shooting or defending) is triggered by a heuristic state machine. The contact state planner and the subsequent mallet controller generate control actions for the mallet. For joint-level control, we use constrained quadratic programming to find the next joint state that tracks the mallet trajectory planned by the higher layers, while staying close to a reference configuration (e.g. a high-manipulability configuration for shooting), with constraints on the joint position, joint velocity, and the z-coordinate of the mallet in order to stay in contact with the table at all times.

1 Introduction

The fast and stochastic nature of the game *air hockey* requires autonomous systems to reason over contact events in the future at a rate that is sufficient to react to inherent perturbations. The fact that a whole match is to be played without any breaks puts a particular focus on the robustness of the planning and control framework. In this paper, we present our planning and control framework that enables a KUKA iiwa robotic manipulator to play air hockey. While the objective of the game - scoring more goals than the opponent - is obvious, the horizon of a whole match is too long to optimize directly for that objective. Instead, we study primitive skills such as *shooting*, *defending* and *preparing* of the puck. Sequences of these skills can be interpreted as robot policies that aim at maximizing the chance of winning the match. In this work, we focus on the shooting skill and

[*] equal contribution

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

formulate the search for the best shooting angle given a puck state in the future as a stochastic optimal control problem. Since solving such optimization problems is time-consuming and we aim at 50 Hz replanning rates, we propose to solve many of these problems offline and train an energy-based model that represents an implicit shooting angle policy. Another key to the performance of our approach is the subsequent optimization of end-effector (i.e. mallet) trajectories also at a rate of 50 Hz by using zero-order optimization on top of a low-dimensional trajectory representation as in [1]. These trajectories are constrained to keep the mallet on the table without colliding with the walls and to connect the current mallet state with the shooting mallet position that is given by the optimal shooting angle. The objective of this second optimization is to maximize the mallet velocity in shooting direction without violating joint velocity limits.

Fig. 1 illustrates the sub-modules that are part of the framework. Section 2 presents our approach of learning locally linear stochastic puck dynamics. Using this model, we implement a stochastic optimal contact planner detailed in Sections 3 and 4. The contact planner operates across three different behavior modes - shooting, defending, and preparing for a shot. We use a heuristics-guided state machine to switch between different behavior modes. Section 5 gives an overview of competition results.

2 Puck State Estimation & Prediction

We model the puck dynamics as piecewise (locally) linear with three different modes: *a)* The puck is not in contact with a wall or a mallet. *b)* The puck is in contact with a wall. *c)* The puck is in contact with a mallet. We use data collected in simulation to learn linear parameters $\boldsymbol{A}_i, \boldsymbol{B}_i$, and an individual covariance matrix $\boldsymbol{\Sigma}_i$ representing the process noise for each mode. This gives us a probability distribution over puck trajectories

$$\Pr_i(\boldsymbol{s}^p_{k+1} | \boldsymbol{s}^p_k, \boldsymbol{s}^m_k) = \mathcal{N}\Big(\boldsymbol{A}_i \boldsymbol{s}^p_k + \boldsymbol{B}_i \boldsymbol{s}^m_k, \boldsymbol{\Sigma}_i(\boldsymbol{s}^p_k, \boldsymbol{s}^m_k)\Big), \tag{1}$$

with puck state $\boldsymbol{s}^p_k = (\boldsymbol{x}^p_k, \dot{\boldsymbol{x}}^p_k)$ and mallet state $\boldsymbol{s}^m_k = (\boldsymbol{x}^m_k, \dot{\boldsymbol{x}}^m_k)$. While each of the modes is a Gaussian distribution, marginalizing over the mallet state in order to propagate the puck state is not possible due to the discontinuity stemming from contacts. Therefore, we utilize an extended Kalman filter to estimate puck states across different timesteps for any given mallet state.

3 Stochastic Optimal Shooting Planner

Depending on the state machine status, the robot is required to generate a *shooting*, *defending* or *preparing* plan including the robot motion from its current state up until the mallet makes contact with the puck. In all cases, we simplify the planning by separating the problem into two phases: *1)* Generating a stochastic optimal contact state for the mallet and puck, and *2)* Generating an optimal mallet trajectory that connects the current mallet state with the contact mallet state planned in *1)*.

3.1 Shooting

Due to process and observation noise, we pose the planning of the shooting contact state as a stochastic optimal control problem and simplify it by making the following assumptions:

1. The contact between the mallet and the puck always happens exactly at $k = 0$. We are only interested in a single mallet state, and we can ignore all other mallet states after contact.
2. The planning horizon K, i.e. the time after the contact, is chosen such that the puck state of interest is exactly occurring at $k = K$. Thus, we can reduce the computation of the quality metric (the probability of scoring a goal) to the final puck state.
3. The time of contact is preset.
4. The mallet velocity at the time of the contact is always set to be the maximum that the robot can generate for a given mallet position while respecting the dynamic constraints.

Keeping these assumptions in mind, we can formulate a stochastic optimal control problem

$$\min_{\boldsymbol{s}^m_0} J\left(\Pr(\boldsymbol{s}^p_K)\right), \quad \text{s.t.} \quad ||\boldsymbol{x}^m_0 - \boldsymbol{x}^p_0||_2 = r^m + r^p, \tag{2}$$

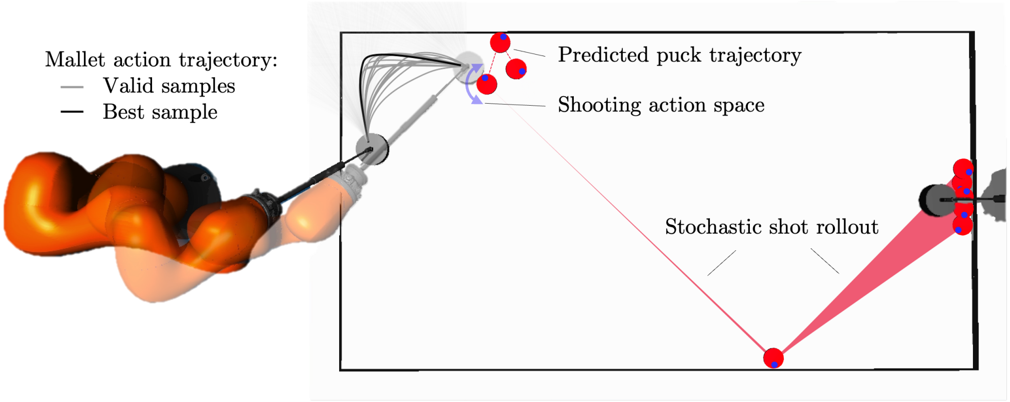

Figure 2: Overview of the interplay between the puck and the mallet for the subtask of scoring a goal. Given the mallet position and the estimated puck position, our framework generates a motion plan for the mallet such that the score probability is maximized.

in which we aim at minimizing a function of the probability distribution of the puck state. In order to compute the goal scoring cost given a candidate contact state, we approximate the probability distribution over puck trajectories as described in Section 2. We define the cost as a weighted sum of the probability of scoring a goal and the expected puck velocity at the goal line with an additional penalty term on all trajectories with a low probability of entering the goal. Different trade-offs between puck velocity and probability of scoring a goal can be achieved by tuning the cost coefficients. Lastly, we leverage the puck-mallet contact constraints to reduce our input space to a single dimension. In order to solve the posed optimization problem at the same control rate of 50 Hz as the middle-level controller, we transfer the heavy computational burden to an offline phase in which we collect optimal plans for a variety of scenarios of interest. We then use the collected data to train a behavior cloning model to rapidly recover an optimal plan for the online scenario at hand. We use an implicit energy-based model [2] to clone the behavior of our offline controller by mapping state-action pairs to an energy function. We use the trained energy-based model to find the optimal mallet angle $\hat{a}$ relative to the puck for a given initial puck state $\boldsymbol{s}_0^p$:

$$\hat{a} = \underset{a \in \mathcal{A}}{\operatorname{argmin}}\, E_\theta(\boldsymbol{s}_0^p, a). \tag{3}$$

During runtime we optimize by iteratively sampling a number of candidate actions at each timestep with recentering and reductions on the sampling variance. We select an action leading to the lowest energy when paired with the puck state, as shown in Figure 3.

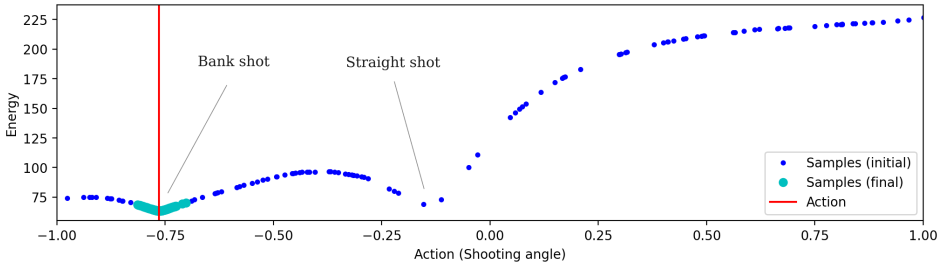

Figure 3: Sampling the action space for an example scenario shown in Figure 2. Dark blue points show samples at the initial timestep. Iterative recentering and variance reductions lead to the final samples shown in cyan. The red vertical line denotes the selected action.

3.2 Defense & preparation

To generate an optimal mallet state at the time of contact for deflecting the puck away from our goal, we apply a sample-based optimization technique online. The objective of the optimization is to achieve a desired vertical puck velocity after contact (e.g. close to zero). We apply the same sample-based optimization technique to prepare the puck for a shot. The goal here is to move the puck towards the horizontal center by bouncing the puck against the wall first. Since this task does not require high precision, we define again a target puck state after contact as the objective for the optimization problem. The target puck state, i.e. direction and speed, is computed heuristically based on the puck position at the time of the contact.

4 Mallet-level Model Predictive Control

The mid-level control layer is responsible for making the mallet hit the puck as planned by the contact planner without colliding with the walls of the table. Thus, the constraint for the mallet controller is to reach a certain mallet state at a given point in time. We reduce the dimensionality of the decision variable for lower computational burden by using a trajectory parameterization that ensures most of our constraints [1]. Our parameterization also minimizes an acceleration functional, i.e. the sum over the squared acceleration is minimal. By computing the basis functions of the trajectory parameterization offline, we again transfer a significant part of the computational burden from the online control loops. In order to exert control over the trajectory, we use the final mallet velocity as a decision variable and use the error w.r.t. to the desired velocity as the new objective. We find the optimal trajectory by sampling a number of candidate mallet velocities and evaluating the corresponding rolled-out trajectories (see Fig. 2). The first mallet action leading to the lowest-cost trajectory is then applied as a control reference.

5 Competition results

The tournament stage of the competition evaluated agents on a simulated KUKA iiwa14 LBR manipulator in a double-round robin tournament scheme. Our agent was able to win all matches within the tournament. We ranked first among 7 participants, with scoring based on the accumulated number of wins, draws, and losses. For detailed results and scoring information, we refer the reader to the competition website: `https://air-hockey-challenge.robot-learning.net/`.

6 Conclusion

This paper investigates the viability of combining a learned planner with sampling-based model-predictive control in order to achieve reactive behavior. We tested and validated our approach in simulation during the *Air Hockey challenge*, where it achieved best performance among the competitors. Future work will investigate the challenges of transferring our agent to a physical system.

Acknowledgments and Disclosure of Funding

We thank the organizers for initiating and providing support during the challenge. This work was supported in part by the Swiss National Science Foundation (SNSF) through the CODIMAN project, and 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).

References

[1] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, "Vp-sto: Via-point-based stochastic trajectory optimization for reactive robot behavior," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10125–10131, 2023.

[2] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Proceedings of the 5th Conference on Robot Learning* (A. Faust, D. Hsu, and G. Neumann, eds.), vol. 164 of *Proceedings of Machine Learning Research*, pp. 158–168, PMLR, 08–11 Nov 2022.