Learning to Play Cup-and-Ball with Noisy Camera Observations

Monimoy Bujarbaruah*,1, Tony Zheng*,1, Akhil Shetty*,1, Martin Sehr, Francesco Borrelli[1]

***Abstract*— Playing the cup-and-ball game is an intriguing task for robotics research since it abstracts important problem characteristics including system nonlinearity, contact forces and precise positioning as terminal goal. In this paper, we present a learning model based control strategy for the cup-and-ball game, where a Universal Robots UR5e manipulator arm learns to catch a ball in one of the cups on a Kendama. Our control problem is divided into two sub-tasks, namely (i) swinging the ball up in a constrained motion, and (ii) catching the free-falling ball. The swing-up trajectory is computed *offline*, and applied in open-loop to the arm. Subsequently, a convex optimization problem is solved *online* during the ball's free-fall to control the manipulator and catch the ball. The controller utilizes noisy position feedback of the ball from an Intel RealSense D435 depth camera. We propose a novel iterative framework, where data is used to learn the support of the camera noise distribution iteratively in order to update the control policy. The probability of a catch with a fixed policy is computed empirically with a user specified number of roll-outs. Our design guarantees that probability of the catch increases in the limit, as the learned support nears the true support of the camera noise distribution. High-fidelity Mujoco simulations and preliminary experimental results support our theoretical analysis (video link – GitHub link).**

I. Introduction

Kendama is the Japanese version of the classic cup-and-ball game, which consists of a handle, a pair of cups, and a ball, which are all connected by a string. Playing the cup-and-ball game is a task commonly considered in robotics research [1]–[8], where approaches ranging from classical PD control to reinforcement learning have been utilized to solve the task. The model-based approaches among the above typically decompose the task into two sub-tasks, namely (i) performing a swing-up of the ball when the string is taut, and (ii) catching the ball during its free-fall. The models of the joint system considered for both sub-tasks are different, thus resulting in hybrid control design for the robotic manipulator. The key drawbacks in such existing approaches are namely the need for expert demonstrations, and the lack of guarantees of operating constraint satisfaction and obtaining catches under modeling uncertainty and sensing errors.

In this paper, we propose a fully physics driven model-based hybrid approach for control design. The controller guarantees a *constrained* motion, while accounting for our best *estimates* of uncertainty in the system model and sensing errors. We use a mixed open-loop and closed-loop control design, motivated by works such as [9]–[11]. First, the swing-up phase is designed offline and then an *open-loop* policy is applied to the robotic manipulator. We use a *cart with inverted pendulum model* of the cup-and-ball joint system for swing-up policy design. For this phase, as we solve a constrained finite horizon non-convex optimization problem, we only consider a *nominal* disturbance-free model of the system. The swing-up trajectory is thus designed to ensure that the predicted difference in positions of the ball and the cup vanishes at a future time once the nominal terminal swing-up state is reached and the cup is held fixed.

After a swing-up, we switch to online closed-loop control synthesis once the ball starts its free-fall. We consider presence of only a camera that takes noisy measurements of the ball's position at every time step. We design the feedback controller in the manipulator's end-effector [12] space. This results in a Linear Time Invariant (LTI) model for the evolution of the difference between the cup and the ball's positions, thus allowing us to solve convex optimization problems online for control synthesis. In order to guarantee a catch by minimizing the position difference, it is also crucial to ensure that during the free-fall of the ball, the control actions to the manipulator do not yield a configuration where the string is taut, despite uncertainty in the model and noise in camera position measurements. Uncertainty in the LTI model primarily arises from low level controller mismatches in the manipulator hardware, and an upper bound of this uncertainty is assumed *known*. Bounds on the measurement noise induced by the camera are assumed unknown. This paper presents a method to increase the *probability* of a catch, as the estimate of the support of camera measurement noise distribution is updated. Our contributions are summarized as:

- *Offline*, before the feedback control of the manipulator, we design a swing-up trajectory for the nominal cup-and-ball system that plans the motion of the ball to a state from which a catch control is initiated.
- Using the notion of *Confidence Support* from [13] which is guaranteed to contain the true support of the camera measurement noise with a specified probability, we use *online* robust feedback control for enforcing bounds on the probability of failed catches.
- With high-fidelity Mujoco simulations and preliminary physical experiments we demonstrate that the manipulator gets better at catching the ball as the support of the camera measurement noise is learned and as the Confidence Support and closed-loop policy are updated.

II. Generating A Swing-up Trajectory

The swing-up phase begins with the arm in the home position such that the ball is hanging down at an angle of 0

[1]The authors are with UC Berkeley, USA, and author Martin Sehr is with Siemens Corporate Technology, USA; E-mails: {monimoyb, tony_zheng, shetty.akhil, fborrelli}@berkeley.edu, martin.sehr@siemens.com.

* These authors contributed equally to this work.

radians from the vertical plumb line, as seen in Fig. 1.

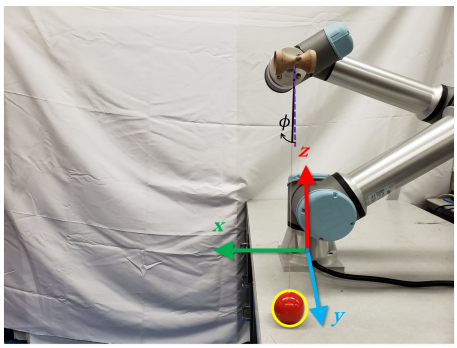

Fig. 1. Manipulator with Kendama along with coordinate frame.

A. System Modeling

We model the system such that the cup is a planar cart with point-mass m_c and the ball acts as a rigid pendulum (mass m_b and radius r) attached to the cup. Assuming planar xz-motion of the ball, we derive the Lagrange equations of motion [12] with three generalized coordinates $\mathbf{q}(t) = (x^{\text{cup}}(t), z^{\text{cup}}(t), \phi(t))$, which denote the x position of the cup, z position of the cup, and swing angle of the ball with respect to the plumb line of the cup respectively at any time $t \geq 0$. We reduce the equations to the general *nominal* form

$$M(\mathbf{q}(t))\ddot{\mathbf{q}}(t)+C(\mathbf{q}(t),\dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t)+G(\mathbf{q}(t)) = F(t),\ \forall t \geq 0, \tag{1}$$

where $M(\mathbf{q}(t))$ is the inertia matrix, $C(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ is the Coriolis matrix, $G(\mathbf{q}(t))$ is the gravity matrix, and $F(t)$ is the external input force at time t. Here $\dot{\mathbf{q}}(t)$ denotes the velocity of the cup and the angular velocity of the ball, and $\ddot{\mathbf{q}}(t)$ denotes the acceleration of the cup and the angular acceleration of the ball at any time $t \geq 0$. System (1) in state-space form is

$$\dot{\bar{x}}(t) = f(\bar{x}(t), F(t)), \tag{2}$$

where nominal state $\bar{x}(t) = [\mathbf{q}^\top(t), \dot{\mathbf{q}}^\top(t)]^\top \in \mathbb{R}^6$ for all time $t \geq 0$.

B. Optimization Problem

We discretize system (2) with one step Euler discretization and a sampling time of $T_s = 100$Hz. The discrete time system can then be written as

$$\bar{x}_{i+1} = \bar{x}_i + T_s f(\bar{x}_i, F_i) = f_d(\bar{x}_i, F_i),\ \forall i \in \{0, 1, \dots\},$$

where a_i denotes the sampled time version of continuous variable $a(t)$. To generate a force input sequence for the swing-up, we solve a constrained optimal control problem over a finite planning horizon of length N, given by:

$$\begin{aligned} \min_{F_0,\dots,F_{N-1}} \quad & \sum_{i=0}^{N-1} \bar{x}_i^\top Q_s \bar{x}_i + F_i^\top R_s F_i \\ \text{s.t.,} \quad & \bar{x}_{i+1} = f_d(\bar{x}_i, F_i), \\ & \bar{x}_i \in \mathcal{X}, F_i \in \mathcal{F}, \\ & \bar{x}_0 = x_{\text{init}}, \\ & \bar{x}_N = x_{\text{f}},\ i = 0, 1, \dots, (N-1), \end{aligned} \tag{3}$$

where weight matrices $Q_s, R_s \succ 0$, and constraint set $\mathcal{X}$ is chosen such that the ball remains within the reach of the UR5e manipulator. Initial state x_{init} is known in the configuration as shown in Fig. 1. Due to the nonlinear dynamics $f_d(\cdot,\cdot)$, the optimization problem (3) is non-convex. Moreover, typically a long horizon length N is required. Hence, we solve (3) *offline* and apply the computed input sequence $\mathbf{F}^\star = [F_0^\star, F_1^\star, \dots, F_{N-1}^\star]$ in open-loop to the manipulator.

C. Terminal Conditions of the Swing-Up

Predicted Behaviour: The nominal terminal state x_{f} in (3) is selected such that the ball is swinging to $\phi = 2.44$ rad with an angular velocity of $\dot{\phi} = 4.18$ rad/s. At these values, the string is calculated to lose tension and the ball begins free-fall. The chosen value of x_{f} ensures that the predicted difference in positions of the ball and the cup (both modeled as point masses) vanishes at a future time, if the cup were held fixed and the ball's motion is predicted under free-fall.

Actual Behaviour: When considering the nominal system (1), we have ignored the presence of uncertainties. Such uncertainties may arise due to our simplifying assumptions such as: (i) the string is mass-less so the swing angle is only affected by the ball and cup masses, (ii) there are no frictional and aerodynamic drag forces to hinder the conservation of kinetic and potential energy of the system, (iii) the cup mass is decoupled from the mass of the manipulator, and (iv) there is no mismatch of control commands from the low level controller of the manipulator and F. Due to such uncertainties, realized states x_i for $i \in \{0, 1, \dots, N\}$ do not *exactly* match their nominal counterparts.

A set of 100 measured roll-out trajectories of the ball after the swing-up are shown in Fig. 2 for a fixed open-loop input sequence $\mathbf{F}^\star$. We see from Fig. 2 that after N time steps

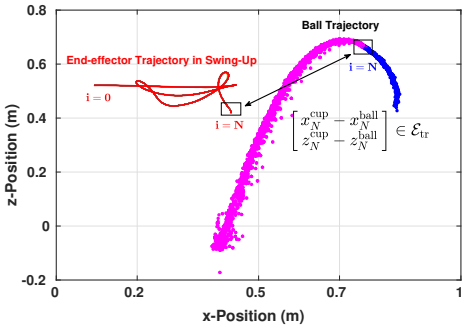

Fig. 2. Start of catch phase (i.e., $i = N$) for 100 trajectories. Red line indicates the trajectory of the cup/end-effector during swing-up. Blue dots indicate ball positions during swing-up and pink dots indicate a position after catch phase is started. Closed-loop control begins when the relative position is in $\mathcal{E}_{\text{tr}}$.

of swing-up, the ball and the cup arrive at positions where their relative position is in a set $\mathcal{E}_{\text{tr}}$. A key assumption of well posedness will be imposed on this set in Section III-D in order for our subsequent feedback control policy to deliver a catch in experiments.

III. Designing Feedback Policy In Catch Phase

For the catch phase we start the time index $t = 0$ where the swing up ends, i.e., $i = N$. There are two main challenges during the design of the feedback controller, namely (i) position measurements of the ball from a noisy camera, and (ii) presence of mismatch between desired control actions and corresponding low level controller commands.

Assumption 1: We assume that the UR5e end-effector gives an accurate estimate of its own position. The assumption is based on precision ranges provided in [14].

A. Problem Formulation

During free-fall of the ball we design our feedback controller for the manipulator position *only* in end-effector space, with desired velocity of the end-effector as our control input. The joint ball and end-effector system in one trial can be modeled as a single integrator as:

$$e_{t+1} = Ae_t + Bu_t + w_t(e_t, u_t), \tag{4a}$$
$$y_t = e_t + v_t, \tag{4b}$$

with error states and inputs (i.e., relative position and velocity)

$$e_t = \begin{bmatrix} x_t^{\text{cup}} - x_t^{\text{ball}} \\ z_t^{\text{cup}} - z_t^{\text{ball}} \end{bmatrix},\ u_t = \begin{bmatrix} v_{x,t}^{\text{cup}} - v_{x,t}^{\text{ball}} \\ v_{z,t}^{\text{cup}} - v_{z,t}^{\text{ball}} \end{bmatrix},$$

where $w_t(e_t, u_t) \in \mathbb{W}_m \subset \mathbb{R}^2$, is a bounded uncertainty which arises due to the discrepancy between (i) the predicted and the actual velocity of the ball at any given time step[1], and (ii) the commanded and the realized velocities of the end-effector, primarily due to the low level controller delays and limitations. System dynamics matrices $A = I_2$ and $B = dt \cdot I_2$ are known, where I_d denotes the identity matrix of size d, and sampling time $dt = 0.01$ second. We assume an outer approximation $\mathbb{W}$ to the set $\mathbb{W}_m$, i.e., $\mathbb{W}_m \subseteq \mathbb{W}$ is known, and is a polytope. We consider noisy measurements of states due to the noise in camera position measurements, corrupted by $v_t \overset{\text{i.i.d.}}{\sim} \mathcal{P}$, with $\text{Supp}(\mathcal{P}) = \mathbb{V}$, where $\text{Supp}(\cdot)$ denotes the support of a distribution. We assume $\mathbb{V}$ is *not* exactly known.

Using the set $\mathcal{E}_{\text{tr}}$ (see Fig. 2), a set $\mathcal{E}$ containing the origin where the string is not taut and (4) is valid can then be chosen. We choose:

$$\gamma^{(i)} = \|\text{vert}^{(i)}(\mathcal{E}_{\text{tr}})\|_\infty,\ i \in \{1, 2\},$$
$$\mathcal{E} = \{x : -\gamma \le x \le \gamma\},\ \gamma = [\gamma^{(1)}, \gamma^{(2)}]^\top, \tag{5}$$

where $\text{vert}^{(i)}(\mathcal{A})$ denotes i^{th} row of all the vertices of the polytope $\mathcal{A}$, and $\|\cdot\|$ denotes the vector norm. This ensures

$$e_0 \in \mathcal{E}_{\text{tr}} \implies e_0 \in \mathcal{E}. \tag{6}$$

As (6) holds true, we impose state and input constraints for all time steps $t \ge 0$ as given by:

$$e_t \in \mathcal{E},\ u_t \in \mathcal{U}, \tag{7}$$

[1]we use the camera position information for ball's velocity estimation

where set $\mathcal{U}$ is a polytope. We formulate the following finite horizon robust optimal control problem for feedback control design:

$$\begin{aligned} \min_{u_0, u_1(\cdot), \ldots} \quad & \sum_{t=0}^{T-1} \ell\left(\bar{e}_t, u_t\left(\bar{e}_t\right)\right) + Q(\bar{e}_T) \\ \text{s.t.,} \quad & e_{t+1} = Ae_t + Bu_t(e_t) + w_t(e_t, u_t), \\ & \bar{e}_{t+1} = A\bar{e}_t + Bu_t(\bar{e}_t), \\ & y_t = e_t + v_t, \\ & e_t \in \mathcal{E}, u_t(e_t) \in \mathcal{U}, \\ & \forall w_t(e_t, u_t) \in \mathbb{W},\ \forall v_t \in \mathbb{V}, \\ & e_0 \in \mathcal{E},\ t = 0, 1, \ldots, (T-1), \end{aligned} \tag{8}$$

where e_t, u_t and $w_t(e_t, u_t)$ denote the realized system state, control input and model uncertainty at time step t respectively, and $(\bar{e}_t, u_t(\bar{e}_t))$ denote the nominal state and corresponding nominal input. Notice that (8) minimizes the nominal cost over a task duration of length T decided by the user, having considered the safety restrictions during an experiment. The cost comprises of the positive definite stage cost $\ell(\cdot, \cdot)$, and the terminal cost $Q(\cdot)$. We point out that, as system (4) is uncertain, the optimal control problem (8) consists of finding $[u_0, u_1(\cdot), u_2(\cdot), \ldots]$, where $u_t : \mathbb{R}^2 \ni x_t \mapsto u_t = u_t(e_t) \in \mathbb{R}^2$ are state feedback policies.

The main challenge in solving problem (8) is that it is difficult to obtain the camera measurement noise distribution support $\mathbb{V}$. Resorting to worst-case a-priori set estimates of $\mathbb{V}$ as in [15], [16] might result in loss of feasibility of (8). To avoid this, we use a data-driven estimate of $\mathbb{V}$ denoted by $\hat{\mathbb{V}}(n)$, where n is the number of samples of noise v_t used to construct the set.

B. Control Formulation

As we have noisy output feedback in (12), we follow [17] for a tractable constrained finite time optimal controller design strategy. We repeatedly solve (8) at times $0 \le t \le (T-1)$ in a shrinking horizon fashion [18, Chapter 9]. We make the following assumption for this purpose:

Assumption 2: The sets $\mathbb{W}_m, \mathbb{W}, \mathbb{V}$, and $\mathcal{U}$ contain the origin in their interior.

1) Observer Design and Control Policy Parametrization: We design a Luenberger observer for the state as

$$\hat{e}_{t+1} = A\hat{e}_t + Bu_t + L(y_t - \hat{e}_t),$$

where the observer gain L is chosen such that $(A - L)$ is Schur stable. The control policy parametrization for solving (8) is chosen as:

$$u_t = \bar{u}_t + K(\hat{e}_t - \bar{e}_t),$$

where state feedback policy gain matrix K is chosen such that $(A + BK)$ is Schur stable.

2) Optimal Control Problem: Consider the tightened constraint sets,

$$\bar{\mathcal{E}}(n) = \mathcal{E} \ominus (\mathcal{R}^{\text{est}}(n) \oplus \mathcal{R}^{\text{con}}(n)), \tag{9a}$$
$$\bar{\mathcal{U}}(n) = \mathcal{U} \ominus K\mathcal{R}^{\text{con}}(n), \tag{9b}$$

where following [17, Proposition 1-2], the set $\mathcal{R}^{\text{est}}(n)$ is our best estimate of the minimal Robust Positive Invariant set $\mathcal{R}^{\text{est}}$ for the *estimation* error $\delta e_t^{\text{est}} = e_t - \hat{e}_t$ dynamics defined as

$$\delta e_{t+1}^{\text{est}} = (A - L)\delta e_t^{\text{est}} + w_t(e_t, u_t) - Lv_t, \tag{10}$$

and the set $\mathcal{R}^{\text{con}}(n)$ is our best estimate of the minimal Robust Positive Invariant set $\mathcal{R}^{\text{con}}$ for the *control* error $\delta e_t^{\text{con}} = \hat{e}_t - \bar{e}_t$ dynamics defined as

$$\delta e_{t+1}^{\text{con}} = (A + BK)\delta e_t^{\text{con}} + L\delta e_t^{\text{est}} + Lv_t, \tag{11}$$

with $v_t \in \hat{\mathbb{V}}(n)$ and $w_t(e_t, u_t) \in \mathbb{W}$. We use the phrase *best estimate* for the above sets, since $\hat{\mathbb{V}}(n)$ is an estimate of true and unknown set $\mathbb{V}$.

Using these sets we then solve the following tractable finite horizon constrained optimal control problem at any time step $t \geq 0$ as an approximation to (8):

$$\begin{aligned} V^{\star}_{t\to T}(\bar{\mathcal{E}}(n), \bar{\mathcal{U}}(n), \mathcal{R}^{\text{con}}(n), \hat{e}_t) := \\ \min_{\bar{e}_t, \bar{u}_t, \ldots, \bar{u}_{T-1}} \quad & \sum_{k=t}^{T-1} \ell(\bar{e}_k, \bar{u}_k) + Q(\bar{e}_T) \\ \text{s.t.,} \quad & \bar{e}_{k+1} = A\bar{e}_k + B\bar{u}_k, \\ & u_k = \bar{u}_k + K(\hat{e}_k - \bar{e}_k), \\ & \bar{e}_k \in \bar{\mathcal{E}}(n), \bar{u}_k \in \bar{\mathcal{U}}(n), \\ & \hat{e}_t - \bar{e}_t \in \mathcal{R}^{\text{con}}(n), \\ & \bar{e}_T = 0, \\ & \forall k \in \{t, t+1, \ldots, (T-1)\}, \end{aligned} \tag{12}$$

where $\hat{e}_t$ is the observed state at time step t, and $\{\bar{e}_k, \bar{u}_k\}$ denote the nominal state and corresponding input respectively predicted at time step $k \geq t$. After solving (12), in closed-loop we apply

$$u_t^{\star}(e_t) : u_t^{\star} = \bar{u}_t^{\star} + K(\hat{e}_t - \bar{e}_t^{\star}) \tag{13}$$

to system (4). We then resolve the problem (12) again at the next $(t+1)$-th time step, yielding a shrinking horizon strategy. The choice of initial observer state is made as follows:

$$\hat{e}_0 \in -(\mathcal{R}^{\text{est}}(n) \ominus \mathcal{E}). \tag{14}$$

Assumption 3 (Manipulator Speed): If any feasible solution is found to (12) satisfying velocity error constraints $\bar{\mathcal{U}}(n)$, the manipulator has enough velocity authority to satisfy these constraints, where the predicted ball velocity is obtained using forward Euler integration at free-fall.

Recall the set $\mathcal{E}_{\text{tr}}$ containing the set of all possible errors e_0 at the start of the catch phase, shown in Fig. 2. We now make the following assumption.

Assumption 4 (Well Posedness): We assume that given state $e_0 \in \mathcal{E}_{\text{tr}}$, optimization problem (12) is feasible at all time steps $0 \leq t \leq (T-1)$ with model uncertainty support $\mathbb{W}$, and true measurement noise support $\hat{\mathbb{V}}(n) = \mathbb{V}$ used in (10)-(11) and (14), when (13) is applied to (4) in closed-loop. This implies that $e_t \in \mathcal{E}$ for all $0 \leq t \leq T$, where $\mathcal{E}$ is obtained from $\mathcal{E}_{\text{tr}}$ following (5).

Definition 1 (Trial Failure): A Trial Failure at time step t is the event

$$[\text{TF}]_t : \ e_t \notin \mathcal{E}, \ 0 \leq t \leq T.$$

That is, a Trial Failure implies the violation of imposed constraints (7) by system (4) in closed-loop with feedback controller (13).

Note that a Trial Failure is a possible scenario only because $\mathbb{V}$ is unknown and is estimated with $\hat{\mathbb{V}}(n)$ in (12). Intuitively, a Trial Failure implies one of the following:

(P1) Problem (12) losing feasibility during $0 < t < T$. This happens if $\hat{\mathbb{V}}(n) \not\supset \mathbb{V}$.

(P2) Problem (12) losing feasibility initially at $t = 0$, and/or sets $\bar{\mathcal{E}}(n), \bar{\mathcal{U}}(n)$ becoming empty. This can happen if $\hat{\mathbb{V}}(n) \supset \mathbb{V}$.

C. Constructing Set $\hat{\mathbb{V}}(n)$

As described in Section III-A the set $\hat{\mathbb{V}}(n)$ is an estimate of the measurement noise support $\mathbb{V}$, derived from n samples of noise v_t. The set $\hat{\mathbb{V}}(n)$ is then used to compute $\mathcal{R}^{\text{est}}(n)$ and $\mathcal{R}^{\text{con}}(n)$ in (10)-(11), used in (12) and (14). We consider the following two design specifications while constructing set $\hat{\mathbb{V}}(n)$, given a fixed sample size n.

(D1) Probability of the event $\hat{\mathbb{V}}(n) \not\supset \mathbb{V}$ is bounded with a user specified upper bound ϵ.

(D2) Estimate $\hat{\mathbb{V}}(n)$ ensures event (P2) in Trial Failure occurs with a vanishing probability, while satisfying specification (D1).

Satisfying (D1) using Distribution Information: Fig. 1 shows the configuration of the system when n noise samples are collected to construct $\hat{\mathbb{V}}(n)$. Let Assumption 1 hold true and the ball is held still, vertically below the end-effector at a position, whose z-coordinate $z^{\text{cup}} = \bar{z}$ is fixed and *known* from previous UR5e end-effector measurements, and x-coordinate is fixed at $x^{\text{ball}} = 0$. We then collect n camera position measurements of the ball at this configuration. The discrepancy between the known position and the measurements yield values of noise samples $\mathbf{v}_n = [v_0, v_1, \ldots, v_n]$. For a fixed *environment*,[2] the distribution of collected samples is shown in Fig. 3, which is approximately a truncated normal distribution. We thereby consider this distribution

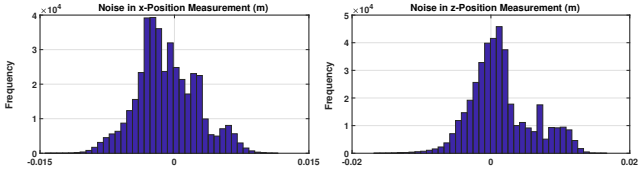

Fig. 3. Camera measurement noise distribution histogram for a fixed camera environment using $n = 400,000$ samples.

family in Fig. 3 conditioned on any environment as

$$\mathcal{P}^q_{\theta_q} | \text{env} = \mathcal{N}_{\text{trunc}}(\mu_q, \sigma_q^2, 3), \ \text{with} \ q \in \{1, 2\}, \tag{15}$$

[2]camera environment is parametrized by say lighting conditions, camera field of view, etc.

where $\mathcal{P}_\theta$ denotes that the distribution $\mathcal{P}$ belongs to a parametric family (truncated normal) parametrized by $\theta = (\mu, \sigma)$, q denotes the q^{th} dimension (x and z directions), and parameters (μ_q, σ_q) are unknown. For a parametric distribution such as (15), for any chosen $\epsilon \in (0,1)$, set $\hat{\mathbb{V}}(n)$ is then constructed as the $(1-\epsilon)$-Confidence Support of $\mathcal{P}_\theta|\text{env}$ using the method in [13], which ensures

$$\mathbb{P}(\hat{\mathbb{V}}(n) \not\supset \mathbb{V}) \leq \epsilon. \tag{16}$$

Note that (16) is a sufficient condition to guarantee that if (D2) holds, solving (12) and applying (13) to (4) gives

$$\mathbb{P}(e_t \notin \mathcal{E}) \leq \epsilon, \ 0 \leq t \leq T, \tag{17}$$

if $\hat{\mathbb{V}}(n)$ is used to construct sets $\mathcal{R}^{\text{est}}(n)$ and $\mathcal{R}^{\text{con}}(n)$.

Satisfying (D2) using Assumption 4: Since Assumption 4 holds, there exists a number of noise samples n_ϵ for any $\epsilon \in (0,1)$, such that $\hat{\mathbb{V}}(n_\epsilon)$ satisfies (D2). Thus, only the sample size n has to be chosen[3] for $\hat{\mathbb{V}}(n)$ appropriately to satisfy (D2), having ensured (17). This guarantees that constructing sets $\mathcal{R}^{\text{con}}(n)$ and $\mathcal{R}^{\text{est}}(n)$ using $\hat{\mathbb{V}}(n)$ and then designing a feedback control by solving (12) results in problem (12) being feasible throughout the task with probability at least $\beta = (1-\epsilon)^{T-1}$. Value of ϵ can be chosen small enough for any user-specified level β can be attained.

D. Obtaining Catches

Constructing $\hat{\mathbb{V}}(n)$ as per Section III-C to ensure (17) is still *not* a sufficient condition to obtain a catch in an experiment with specified probability β, as our model (4) does not account for additional factors such as object dimensions, presence of contact forces, etc.

To that regard, we introduce the notion of a *successful catch*, which is defined as the ball successfully ending up inside the cup at the end of a roll-out. Thus, a successful catch accounts for the dimensions of the ball and the cup, and the presence of contact forces.

Assumption 5 (Existence of a Successful Catch): We assume that given an initial state $e_0 \in \mathcal{E}_{\text{tr}}$, an input policy obtained by solving (12) can yield a successful catch, if true measurement noise support $\mathbb{V}$ were known *exactly*.

Remark 1: From [13] we know that as long as confidence intervals for parameters (μ, σ) in (15) converge, $\hat{\mathbb{V}}(n) \to \mathbb{V}$ as $n \to \infty$. So, if sample size n is increased iteratively approaching $n \to \infty$, obtaining a successful catch guaranteed owing to Assumption 5. However if a precise positioning system like Vicon is used to collect the noise samples, due to limited access to such environments, collecting more samples and increasing n could be expensive. We therefore stick to our method of constructing $\hat{\mathbb{V}}(n)$ for a *fixed* n as per Section III-C, and we attempt successful catches with multiple roll-outs by solving (12). For improving the empirical probability of successful catches in these roll-outs, one may then increase n and thus update the control policy. We demonstrate this in Section IV-B.

[3]for n fixed, ϵ can be increased while constructing $\hat{\mathbb{V}}(n)$ to satisfy (D2).

IV. Experimental Results

We present our preliminary experimental findings in this section. For our experiments, the original Kendama handle was modified to be attached to a 3D printed mount on the UR5e end-effector, as shown in Fig. 1. A single Intel RealSense D435 depth camera running at 60 FPS was used to estimate the position and velocity of the ball.

A. Control Design in the Catch Phase

Once the swing-up controller is designed as per Section II-B and an open-loop swing-up control sequence is applied to the manipulator, we design the feedback controller by finding approximate solutions to the following problem:

$$\begin{aligned}
&\min_{u_0, u_1(\cdot), \ldots} \sum_{t=0}^{T-1} 500 \left\|\bar{e}_t\right\|_2^2 + 0.4 \left\|u_t(\bar{e}_t)\right\|_2^2 \\
&\text{s.t.,} \\
&\qquad e_{t+1} = Ae_t + Bu_t(e_t), \\
&\qquad \bar{e}_{t+1} = A\bar{e}_t + Bu_t(\bar{e}_t), \\
&\qquad y_t = e_t + v_t, \\
&\qquad e_t \in \mathcal{E}, \ \begin{bmatrix} -8\text{m/s} \\ -8\text{m/s} \end{bmatrix} \leq u_t(e_t) \leq \begin{bmatrix} 8\text{m/s} \\ 8\text{m/s} \end{bmatrix}, \\
&\qquad \forall v_t \in \mathbb{V}, \\
&\qquad t = 0, 1, \ldots, (T-1),
\end{aligned} \tag{18}$$

where set $\mathcal{E}_{\text{tr}} = [-0.316\text{m}, 0.349\text{m}] \times [-0.2095\text{m}, 0.2457\text{m}]$, shown in Fig. 2. Note that for this specific scenario the presence of model uncertainty can be ignored. Set $\mathbb{V}$ is unknown, and we consider Assumption 4 holds. System matrices A, B are from Section III-A. We find solutions to (18) for $T = 50$ steps, i.e., 0.5 seconds.

B. Learning to Catch

We conduct 50 *roll-outs* of the catching task by solving (12), having formed $\hat{\mathbb{V}}(n)$ as per Section III-C, with $n = 100$ and then iteratively increasing to $n = 2000$. Sets $\hat{\mathbb{V}}(n)$ are formed using [13]. Fig. 4 shows the percentage of roll-outs conducted for each iteration (i.e., for each value of n), that resulted in the ball successfully striking the center of the cup. The percentage increases from 41.46% to 61.62%.

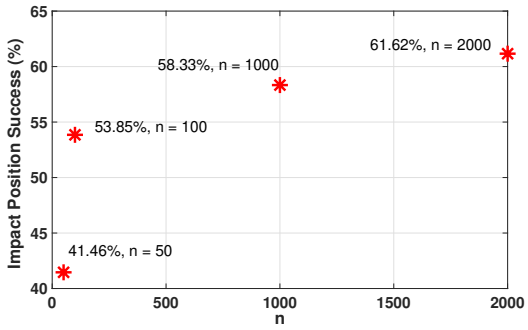

Fig. 4. Percentage of times the ball hitting the cup center among all roll-outs vs sample size n.

Furthermore, another crucial quantity at the time of impact is the commanded relative velocity (13) in z-direction, a

lower value of which indicates an increased likelihood of the ball not bouncing out. The average value and the standard deviation of of $(u^\star_{T_{\rm im}-1})^{*\tilde{m}}_z$ for $\tilde{m} \in \{1, 2, \ldots, 50\}$ is shown in Fig. 5, where $(\cdot)^{*\tilde{m}}$ denotes the $\tilde{m}^{\rm th}$ roll-out and $T_{\rm im} \leq T$ denotes the time of impact. As seen in Fig. 5, the mean

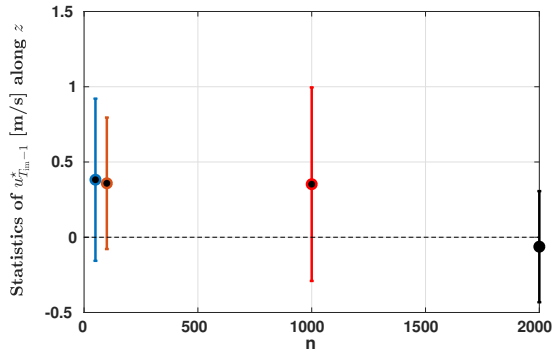

Fig. 5. One standard deviation interval around the mean (circle) of z-relative velocity at impact, i.e., $[u^\star_{T_{\rm im}-1}]_z$ vs sample size n.

of the relative velocity at impact lowers from 0.38 m/s to -0.06 m/s. This together with Fig. 4 indicates a possibility of increasing successful catch counts as n is increased.

C. Increasing Successful Catches

In order to prove that the trend shown in Fig. 4 and Fig. 5 results in an increasing number of successful catches, we resort to exhaustive Mujoco [19], [20] simulations[4]. The task duration in this case is $T = 25$ steps. The trend in

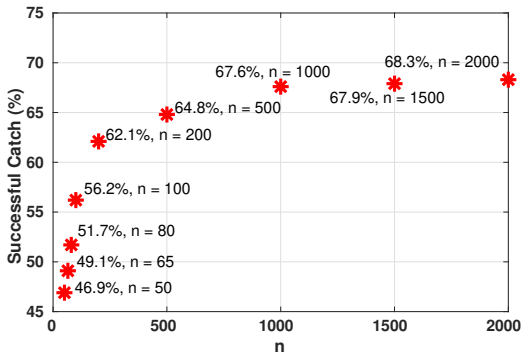

Fig. 6. Percentage of successful catches vs sample size n.

the percentage of successful catches with 1000 roll-outs corresponding to each n, varying from $n = 50$ to $n = 2000$, is shown in Fig. 6. For $n = 50$, 46.9% of the roll-outs result in a successful catch. The number increases to 68.3% for $n = 2000$. This verifies that the preliminary experimental results from Fig. 4 and Fig. 5 would very likely result in a similar trend as in Fig. 6. Thus we prove that our proposed approach enables successful learning of the kendama ball catching task.

[4]due to unavailability of laboratory access

V. Conclusions

We proposed a model based control strategy for the classic cup-and-ball game. The controller utilized noisy position measurements of the ball from a camera, and the support of this noise distribution was iteratively learned from data. Thus, the closed-loop control policy iteratively updates. We proved that the probability of a catch increases in the limit, as the learned support nears the true support of the camera noise distribution. Preliminary experimental results and high-fidelity simulations support our analysis.

Acknowledgement

We thank Yuri Glauthier, Charlott Vallon, and Sangli Teng for their contributions on the hardware experiments, as well as Vijay Govindarajan, Siddharth Nair and Edward Zhu for extremely useful reviews and discussions. The research was funded by grants ONR-N00014-18-1-2833, NSF-1931853, and Siemens.

References

[1] B. Nemec and A. Ude, "Reinforcement learning of ball-in-a-cup playing robot," in *2011 IEEE International Conference on Robotics and Biomimetics*, Dec 2011, pp. 2682–2987.

[2] J. Kober and J. Peters, "Learning motor primitives for robotics," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 2112–2118.

[3] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato, "A kendama learning robot based on bi-directional theory," *Neural Networks*, vol. 9, no. 8, pp. 1281 – 1302, 1996.

[4] T. Sakaguchi and F. Miyazaki, "Dynamic manipulation of ball-in-cup game," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 2941–2948 vol.4.

[5] A. Namiki and N. Itoi, "Ball catching in kendama game by estimating grasp conditions based on a high-speed vision system and tactile sensors," in *2014 IEEE-RAS International Conference on Humanoid Robots*, Nov 2014, pp. 634–639.

[6] D. Schwab, T. Springenberg, M. F. Martins, T. Lampe, M. Neunert, A. Abdolmaleki, T. Herkweck, R. Hafner, F. Nori, and M. Riedmiller, "Simultaneously learning vision and feature-based control policies for real-world ball-in-a-cup," *arXiv preprint arXiv:1902.04706*, 2019.

[7] T. Senoo, A. Namiki, and M. Ishikawa, "Ball control in high-speed batting motion using hybrid trajectory generator," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1762–1767.

[8] S. Li, "Robot playing kendama with model-based and model-free reinforcement learning," *arXiv preprint arXiv:2003.06751*, 2020.

[9] E. A. Hansen, A. G. Barto, and S. Zilberstein, "Reinforcement learning for mixed open-loop and closed-loop control," in *NIPS*, 1996.

[10] C. G. Atkeson and S. Schaal, "Learning tasks from a single demonstration," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, 1997, pp. 1706–1712.

[11] J. Z. Kolter, C. Plagemann, D. T. Jackson, A. Y. Ng, and S. Thrun, "A probabilistic approach to mixed open-loop and closed-loop control, with application to extreme autonomous driving," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 839–845.

[12] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[13] M. Bujarbaruah, A. Shetty, K. Poolla, and F. Borrelli, "Learning robustness with bounded failure: An iterative MPC approach," *arXiv preprint arXiv:1911.09910*, 2019.

[14] U. Robots, "e-Series from universal robots," *https://www.universal-robots.com/media/1802432/e-series-brochure.pdf*, 2014.

[15] M. Tanaskovic, L. Fagiano, R. Smith, and M. Morari, "Adaptive receding horizon control for constrained mimo systems," *Automatica*, vol. 50, no. 12, pp. 3019–3029, 2014.

[16] X. Lu and M. Cannon, "Robust adaptive tube model predictive control," in *2019 IEEE American Control Conference (ACC)*. IEEE, Jul. 2019, pp. 3695–3701.

[17] D. Q. Mayne, S. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, no. 7, pp. 1217–1222, 2006.

[18] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[19] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[20] Y. Tassa, S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, and N. Heess, "dm_control: Software and tasks for continuous control," *arXiv preprint arXiv:2006.12983*, 2020.