Learning the Inverse Hitting Problem

Harshit Khurana, *Graduate Student Member, IEEE*, James Hermus, Maxime Gautier, and Aude Billard, *Fellow, IEEE*

***Abstract*—This letter presents a data collection framework and a learning model to understand the motion of an object after being subject to an impulse. The data collection framework consists of an automated dual arm setup hitting an object to each other, like a collaborative air-hockey game. An impact aware extended Kalman filter is proposed for automation of the air-hockey setup which approximates the discontinuous impulse motion equations through a hitting force model by balancing the energies during collision. To capture the variance in the motion that stochasticity of friction introduces, the errors in the controls for the hitting flux, we model the stochastic relationship between hitting flux and object's resulting displacement, using full density modeling. Further we show the application of the learnt motion model for planning sequential hits with two or more robots, in a Golf-like principle, to enable an object to reach a location far beyond the reach of a single robot.**

***Index Terms*—Impact aware manipulation, robot collaboration, impact aware extended Kalman filter, dual arm system, GMM, GMR, golf.**

I. Introduction

IMPACTFUL contacts between a robot and its environment are rarely exploited in robotic manipulation. In moving objects from a given position to a desired position, methods such as pick and place (prehensile), or pushing (non-prehensile) are more widely used than hitting an object. In such manipulation strategies, the object's final desired position is typically within the workspace of the fixed robot. In pushing, this is due to establishing contact between the robot and the object at relatively low speeds. This is called the quasi-static assumption and the inertial effects of the bodies in contact are negligible. While *hitting* an object [1] has the benefit of increasing the space in which an object can be placed, it is important to consider the dynamics of the robot and the inertial properties of the object. Hence, the robot-object contact cannot be assumed to be quasi-static. Predicting object motion when pushing is challenging due to the multi-modality of the factors affecting the object's displacement, such as contact position, and variable friction. [2], predicting the final displacement of an object after an impactful hit requires a model of the impulse and the restitution coefficient that measures the amount of energy transmitted. This is particularly difficult when the impact is inelastic. Accurate modeling of the expected displacement after impact is crucial to invert the model, so as to plan the impact to achieve a desired motion. *Pushing* objects to move them from one place to another has benefited from the improvements in data driven methods such as reinforcement learning and modeling through heteroscedastic Gaussian Processes as they implicitly model the contact behavior [3], [4], [5]. Previously in [1], we proposed a method for impact aware motion planning and inertia-based control to generate repeatable impacts on objects of different masses with constraints on their shape and in [6], we proposed a method to model the process of hitting an object so as to place it at a desired position outside of the workspace of the robot through a regression model linking hitting velocity to the distance traveled by the object. In this letter we use hitting flux as the hitting control parameter and learn the relationship between the hitting flux and the object motion and show its applicability. We write the contributions of this letter as follows:

1) A semi-autonomous dual arm system for robot - object impulsive interaction data collection
2) An Impact Aware Extended Kalman Filter (IA-EKF) to predict object's final position subject to an impulse
3) A learning based inverse map to predict the hitting parameters to achieve the desired object's displacement, through modeling of the stochastic relationship between object displacement and hitting flux.
4) Experiments validating the learnt model on different robots and objects
5) A motion planning framework inspired by the principles of golf for placing objects outside the robot's reach in an unstructured environment.

II. Related Work

A. Manipulation Through Pushing

1) Analytical + Data Driven Models: In pushing, the motion of the object can be represented through an analytical model taking into account the contact forces [7], [8] or through data driven models such as [2], [3], [5]. In the above mentioned works, the quasi-static assumption implies that the inertia forces are negligible. Hence, the object moves only when the robot is in contact with the object and applying force on it. The motion of the object, when modeled analytically does not take into account the variability of the robot controller and the stochastic nature of friction at contact points, and in the sliding surface. To deal with the uncertainty of these parameters (friction, contact area etc), [8] estimates the parameters with data to make the prediction for the object motion closer to reality. Even if one knows such parameters, data-driven models exhibit stochasticity in the friction model; that is, given the same pushing strategy, the object's motion differs slightly each time.

Received 18 July 2024; accepted 9 February 2025. Date of publication 6 March 2025; date of current version 21 March 2025. This article was recommended for publication by Associate Editor K. Pertsch and Editor A. Faust upon evaluation of the reviewers' comments. This work was supported by Research Project I.AM. through the European Union H2020 program under GA 871899. *(Corresponding author: Harshit Khurana.)*

The authors are with the Learning Algorithms and Systems Laboratory, EPFL, 1015 Lausanne, Switzerland (e-mail: harshit.khurana@epfl.ch; james.hermus@epfl.ch; Maxime.Gautier@epfl.ch; Aude.Billard@epfl.ch).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3548496, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3548496

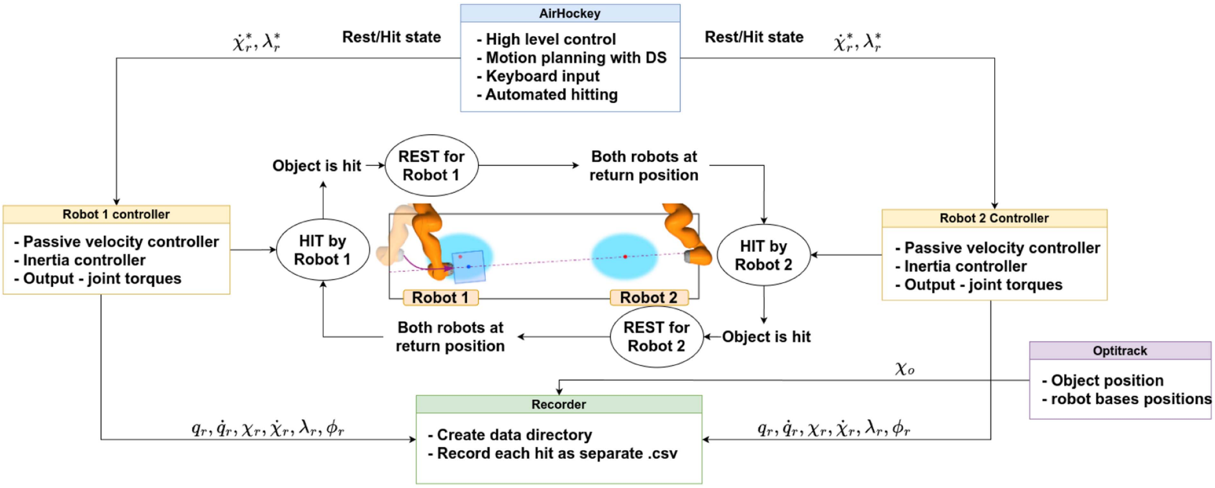


Fig. 1. From the center going outward: 1) Dual robot placement schematic. 2) High level motion planning for the robots, i.e., REST state and HIT state. 3) Detailed diagram of the control inputs, outputs and state variables.

TABLE I
OBJECTS / BOXES USED IN THE EXPERIMENTS

Object	1	2	3
Mass (kg)	0.36	1.96	0.49
size (cm^3)	18x19x19	26x26x27	h = 22 cm, r = 10 cm
shape	cuboid	cuboid	cylindrical

B. Manipulation Through Hitting

Robots interacting with the environment with impacts can be seen in examples such as RoboCup [9], robot table tennis [10], golf [11], baseball [12], and in quadrotors juggling using a small ball [13]. But little work has been done that incorporates the object and robot inertial properties. In our previous work we have established preliminary methodology of learning the object motion through data [6] and creating a motion for the robot for repeatable hits while controlling for hitting flux, which incorporates the hitting speed and the directional inertia of the robot [1].

C. Object Motion Estimation

Kalman and Particle filters have been extensively used for predicting motion of an object subject to external forces with incomplete measurement and noisy sensors. Yet, the literature lacks online estimation strategies for predicting motion of an object subject to impulse. [14] uses a Kalman filter to predict puck's motion after being hit by the robot and mitigates the discontinuity in the collision between the puck and the table using a particle filter. However the model of the motion of the puck does not incorporate the contact between the robot and the puck, the estimation depends on the measurements of the puck's motion, and in this case the assumptions such as an infinite mass for the stick and purely elastic collisions with a coefficient of restitution of were reasonable. In our experimental setup, both the inertial dynamics of the box and the robot are significant, the impact can not be modeled as perfectly elastic, and the friction between the box and the table is substantial.

III. METHODS

We describe the data collection framework, the experimental setup, and learning model for the object motion [1]

A. Hitting Flux and Motion Generation

Hitting flux (ϕ_h) is defined as

$$\phi_h = \frac{\lambda_h}{\lambda_h + m_o} \dot{\chi}_{rh}^{-} \tag{1}$$

, where λ_h is the directional inertia of the robot and $\dot{\chi}_{rh}^{-}$ is the hitting speed. We use the same method for robot motion generation as in [1] which works for planar sliding and for objects where the established contact normal is parallel to the sliding plane. This is assumed for objects used in this manuscript as well.

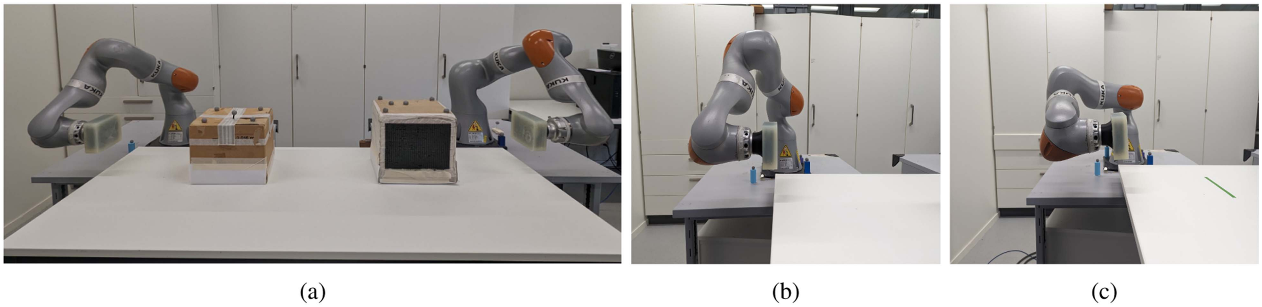
B. Physical Setup

Fig. 2 shows the dual arm framework for data collection. It consists of the robots arms KUKA lbr iiwa 7 and 14, two different sized box objects with uniform mass distribution (Refer Table I). A third cylindrical test object (Object 3) is used for testing the model as well. The setup is surrounded by 12 Optitrack 17 W infrared cameras which can accurately track objects in 3D space through markers.

C. Data Collection Through Dual-Arm Hitting Game

To learn the motion of an object subject to an impulse and environment variability, we need data. Most hits would make an object move out of the reachable workspace of the robot,

[1] The code for all the above is publicly available at: https://github.com/epfl-lasa/air_hockey.



(a) (b) (c)

Fig. 2. (a): The physical setup (KUKA lbr iiwa 7 and 14) robot arms and the two different uniform boxes used. A table top acts as the sliding surface. (b) and (c): The two different configurations of the robot used in hitting the boxes.

TABLE II
DATA COLLECTED FOR EACH ROBOT

Robot	$\{1, 2\}$
Hitting time	$t_h \in \mathbb{R}$
Hitting joint config. (rad)	$q_h \in \mathbb{R}^7$
Hitting joint vel. (rad/s)	$\dot{q}_h \in \mathbb{R}^7$
Hitting torque (Nm)	$\tau_a \in \mathbb{R}^7$
Commanded hitting torque (Nm)	$\tau_c \in \mathbb{R}^7$
EE hitting position (m)	$\chi_r \in \mathbb{R}^3$
EE hitting vel. (m/s)	$\dot{\chi}_r \in \mathbb{R}^3$
EE orientation	$\psi \in \mathbb{SO}3$
EE dir. inertia (kg)	$\lambda_r \in \mathbb{R}$
Achieved hitting flux (m/s)	$\phi_h \in \mathbb{R}$
Object trajectory (s, m)	$\{t, \chi_o\} \in \{\mathbb{R}, \mathbb{R}^3\}$
Object distance moved (m)	$d \in \mathbb{R}$

making it impossible for the robot itself to reset the box to its initial position. Having a human reset the environment is time consuming, while limiting the environment variability that data can inherit. Thus, we create a dual arm system inspired by the game of AirHockey, such as in [15], [16], [17] removing the need to reset the object while doubling the data collection speed. *High level task decision policy + Motion plan:* Fig. 1 shows diagrammatically the robots and the policy that is realized at high level control.

1) *Hitting:* The robot moves to generate the desired hitting flux (19) to hit an object.
2) *Rest:* The robot returns to its initial position after hit.

The variables recorded are as shown in Table II.

D. Impact Aware EKF

In the air hockey setup, where robots hit the puck back and forth, we propose an Impact-Aware Extended Kalman Filter to predict the puck's final position online, significantly reducing data collection time. This enables the robot to begin moving toward the expected box rest position before it arrived. The proposed method has two key aspects. Firstly, the approach incorporates not only information about the box, but also the robot, which is represented as a flux. Secondly, it employs a smoothed approximate model of the hitting force that acts on the box. This smooth Gaussian contact force function, eliminates the discontinuity, enabling the utilization of an EKF. While this approach results in a knowingly incorrect estimate of the absolute box position at impact, it provides the benefit of a velocity estimate which converges faster – the key aspect for estimating the final box position. In this section, both, the impact aware state transition model and box only state transition model are described, results of which are compared in Section III-E. The impact-aware model incorporates knowledge of the desired flux and a smooth hitting force model into the prediction process. Modeling the contact dynamics for the impact-aware model necessitates addressing non-linearity. In this work, we utilize a standard implementation of the EKF, which is presented in Appendix C. To ensure consistency, we employ the same standard EKF implementation presented in the Appendix C for both models.

Box Only Model: The motion model of the box alone, neglecting the robot, simplifies to the case of a linear Kalman Filter, with no need for the local linear approximation of an EKF. The state of the system is denoted by $\boldsymbol{x}^b = [\chi^o, \dot{\chi}^o, \mu]^T$ where,

$$\chi_o^{k+1} = \chi_o^k + (dt)\dot{\chi}_o^k,$$

$$\dot{\chi}_o^{k+1} = \dot{\chi}_o^k, \quad \mu^{k+1} = \mu^k$$

Here, χ_o represents the box position, $\dot{\chi}_o$ denotes the box velocity, μ represents the friction coefficient, ϵ denotes the coefficient of restitution, dt represents the time step, and m denotes the mass of the object. This model did not include any input $\boldsymbol{u}$, and the measurement was solely comprised of the box position, denoted by $\boldsymbol{y} = [\chi_o]$.

Impact Aware Model: This model incorporates an estimate of the robot end-effector position denoted by χ_r, velocity by $\dot{\chi}_r$, and the energy imparted to the box denoted by E. The state vector here is $\boldsymbol{x}^i = [\chi_o, \dot{\chi}_o, \chi_r, \dot{\chi}_r, E, \mu, \epsilon]^T$. The state transition function is defined as,

$$\chi_o^{k+1} = \chi_o^k + (dt)\dot{\chi}_o^k + \frac{dt^2}{2m}(f_i^k + f_f^k), \tag{2}$$

$$\dot{\chi}_o^{k+1} = \dot{\chi}_o^k + \frac{dt}{m}(f_i^k + f_f^k), \tag{3}$$

$$\chi_r^{k+1} = \chi_r^k + (dt)\dot{\chi}_r^k, \tag{4}$$

$$\dot{\chi}_r^{k+1} = \dot{\chi}_r^k, \tag{5}$$

$$E^{k+1} = \begin{cases} E^k + dt\left|\dot{\chi}_o^k f_i^k\right| & \text{if } E^k < E_{pred} \\ E_{pred} & \text{else} \end{cases} \tag{6}$$

$$\mu^{k+1} = \mu^k, \quad \epsilon^{k+1} = \epsilon^k \tag{7}$$

Where the impact force is defined as,

$$f_i = \begin{cases} \sigma_1 \exp(\frac{-\Delta\chi^2}{2\sigma_2^2}) & \text{if } E < E_{pred} \\ 0 & \text{else ,} \end{cases} \tag{8}$$

where the distance between the end-effector and the box is defined as $\Delta\chi$. The parameter σ_2 determines the width of the force Gaussian, while σ_1 is chosen such that the work done by the hitting force equals the work expected based on the desired flux. The selection of σ_2 is discussed in Appendix B. The predicted energy, E_{pred} is the integral of the impact force over the distance and is defined using the predicted flux equaling $\sigma_1\sigma_2\sqrt{2\pi}$. Additionally, the friction force is defined as,

$$f_f = -sgn(\dot{\chi}_o)\ \mu m g \tag{9}$$

where g is the gravitational constant. The only input was the reference flux, denoted as ϕ. The measurement includes both the box position and the end-effector position, represented as $\boldsymbol{y} = [\chi_o, \chi_r]^T$.

To estimate the final position denoted $\hat{\chi}_f$, knowledge of the predicted final position based on flux denoted $\hat{\chi}_f^{\phi}$ is combined with the state estimate from the EKF denoted $\hat{\chi}_f^{kalman}$. This assumes that the relation between flux, the coefficient of restitution, and the predicted post object velocity is $\dot{\hat{\chi}}^{+}_{o,\phi} = (1+\epsilon)\phi$. At the start of the task the best estimate of the final box position is based on the flux. This prediction assumes friction was the only force acting on the box after contact. The final position prediction based on hitting flux is,

$$\hat{\chi}_f^{\phi} = \chi_o^{init} + \frac{(\dot{\hat{\chi}}^{+}_{o,\phi})^2}{2\mu g}, \tag{10}$$

where χ_o^{init} is the initial box position. The predicted final box position based on the EKF velocity estimate is,

$$\hat{\chi}_f^{kalman} = \chi_o^k + \frac{(\dot{\chi}_o)^2}{2\mu g}. \tag{11}$$

A convex combination of the flux prediction and the EKF prediction is constructed based on the energy imparted on the box. In this case alpha is defined as $\alpha = E^k/E_{pred}$. This combination is used to initialize the final online prediction $\hat{\chi}_f^{\phi}$.

$$\hat{\chi}_f = (1-\alpha)\hat{\chi}_f^{\phi} + \alpha\hat{\chi}_f^{kalman} \tag{12}$$

In the experiments, the data collection was run at 200 Hz and the Kalman filter at 1000 Hz. The proposed Kalman filter is run over trials by initializing the friction and restitution coefficients (μ, ϵ) with the previous trial's converged values.

E. EKF Behavior

The results for Impact aware EKF and Box only EKF for Object 1 are shown in Fig. 3. We can immediately notice that the Box only EKF (purple dashed line) takes overshoots and converges once the box has reached its final position. In yellow, one can see the prediction with Impact Aware EKF, which has the following properties: One, it converges faster than the Box only EKF Model. Second, as the number of hits increase and the coefficients of friction and restitution converge, the final position prediction becomes quite accurate in both, the simulation and the real robot experiments. The convergence time improves by ~ 0.5s or $\sim 44\%$. As observed, after around 15 hits (10) can be used to predict the final position of the box after being hit online as the values of μ and ϵ have converged.

F. Ablation Studies

The data collection framework's requirements are to collect data accurately and fast. The design decisions are based on the time efficiency of data collection. Table III reports the time taken per data point for the dual arm system, and for the single arm setup (refer [6]). The time taken per data point improves by 94.3% with the design of the framework.

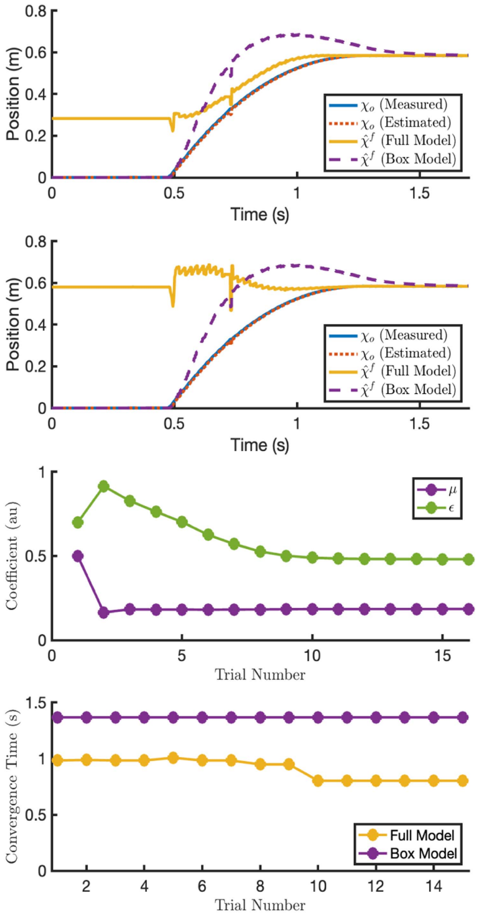


Fig. 3. From top to bottom: 1. Evolution of the EKF estimate of the estimated object's displacement after first hit, when the estimate of the friction μ and restitution ϵ coefficients has not yet converged; 2. Evolution of the EKF estimate for the 15th hit with known μ and ϵ; 3. Convergence of the estimates of μ and ϵ; 4. Convergence time for EKF estimate of final object's position across trials.

TABLE III
ABLATION STUDY ON TIME TAKEN PER DATA POINT ON DIFFERENT FRAMEWORKS

	Data Points	Total time (s)	time / data point (s)
Single Arm	60	$\sim$ 7200	120
Dual Arm	2000	$\sim$ 14400	7.20
Dual Arm (EKF)	3275	$\sim$ 22500	6.87

IV. DATA MODELING

A. Learning Model of Data Distribution

To encapsulate the stochastic nature of the relationship between the input, hitting flux (ϕ), and realized outcome, distance traveled by the object (d_ϕ), we learn a full density model of the joint probability distribution $P(\phi, d_\phi)$ using Gaussian Mixture Model (GMM). The model is trained through Maximum Likelihood, with k-means clustering for parameters' initialization. Bayesian Information Criterion is used to select the optimal number of Gaussians in the GMM, (in this case, 2). The inverse model to predict the desired hitting flux (ϕ^*) given the initial

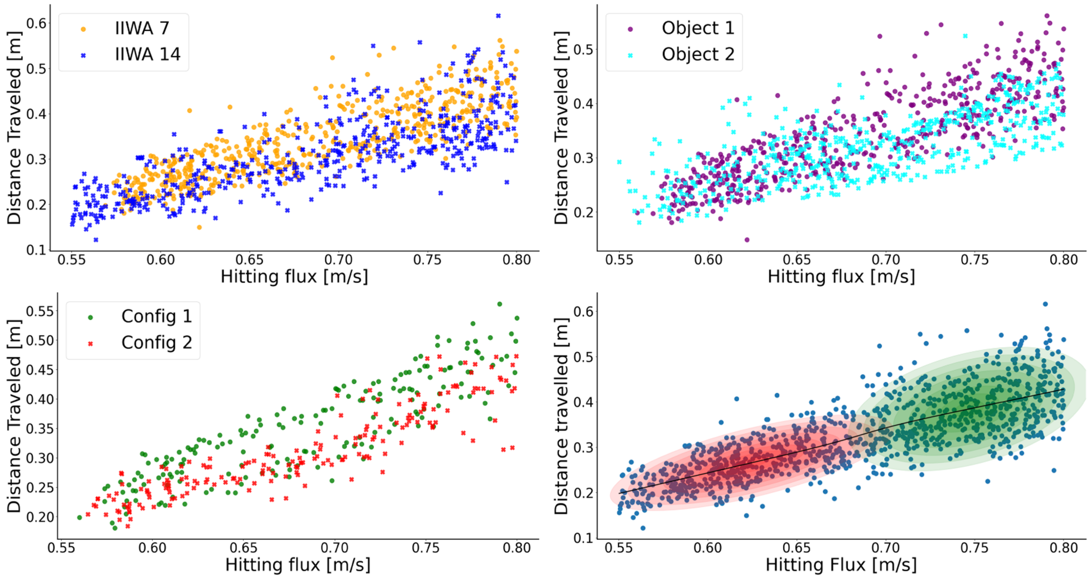


Fig. 4. All graphs show the distance of object traveled in relation to the hitting flux. Top left: In different colors are the points showcasing hits by different robots. Top right: In different colors are the points showcasing different objects being hit. Bottom left: In different colors are the points showcasing hits by a robot in two different configurations. Bottom right: Joint Probability distribution of the data modeled using a GMM with 2 Gaussians as determined using BIC criteria.

TABLE IV
DISTRIBUTION OF HITTING DATA

Robot	Object	No. of data points	total data points
iiwa 7	1	1285	2794
	2	1005	
iiwa 14	1	979	2481
	2	1002	

Configuration of the robot	No. of data points
1	188
2	186

TABLE V
KL DIVERGENCE AND RMSE FOR GMR PREDICTIONS USING SWAPPED MODELS

Distributions	KL-Divergence	RMSE (m) 1|2	RMSE (m) 2|1
Robot (1, 2)	0.2950	0.079	0.082
Objects (1, 2)	0.4497	0.060	0.081
Configurations (1, 2)	0.9835	0.053	0.054

and the final desired position of the object, we compute the expectation on the conditional probability $\phi^* = \mathrm{E}(P(\phi|d_\phi)$. Refer Appendix D for the details on GMM and GMR.

B. Data Properties and Learnt Model Assessment

The number of data points collected for each robot type,configuration, and each object is detailed in Table IV, and visually represented in Fig. 4.

Accuracy of desired flux prediction: We assess the accuracy of the GMR prediction of the hitting flux for the desired object distance through 10-fold cross-validation with a 70 $\%$ training / testing ratio. Using GMR, for a desired distance traveled, hitting flux is calculated and the mean RMSE for the desired and actual distance traveled by the object for the calculated hitting flux. The RSME (root mean squared error) in the desired and achieved distance is $0.054m$. We hypothesize that the learnt motion model of the object will have some degree of generalizability to the robot, its hitting configuration and the object. To calculate the degree of agnosticism we calculate KL divergence of the distributions of the object motion data when hit by different robots and in different robot configurations. Kullback-Leiber (KL) divergence measures the difference between two different probability distributions, and hence can be described as a measure of how much information is lost when one probability distribution is used to approximate another.

$$D_{KL}(p(z)||q(z)) = \sum_{z \in Data} p(z) ln \frac{p(z)}{q(z)}$$

In addition to the KL divergence values, generalization of the models is tested by using a specific robot / object / configuration model for prediction for hitting with the other robot / object / configuration respectively and the RMSE errors for the achieved and predicted distances are listed in Table V. The low values of distributions of KL divergence values for the distributions with different robots and different objects shows that there is a high similarity in such distributions allowing us to use one object to understand motion of the other and one robot to understand how to hit with another robot. The distributions of hitting with two different robot configurations and different robots have relatively higher KL divergence value. This can be attributed to small errors in inertia calculation using URDFs and its investigation is part of the future work.

C. Transfer Learning

The presence of a degree of generalizability gives rise to the idea that it should be possible to shift an object's known motion distribution to another object's motion distribution using a few data points, i.e., a robot having an idea of how Object 1 moves could understand the motion of Object 2 through fewer data points. We show the preliminary results of learning Object 2 motion model given the model for Object 1 by using incremental

TABLE VI
KL DIVERGENCE VALUES OF LEARNING A MODEL THROUGH 1. SCRATCH AND 2. INITIALIZING THE MODEL WITH A KNOWN OBJECT'S MODEL

Data points	10	20	50	99	143	228	304
Scratch learning	0.60	0.43	0.12	0.07	0.04	0.02	0.01
Transfer model	0.47	0.22	0.09	0.08	0.04	0.02	0.01

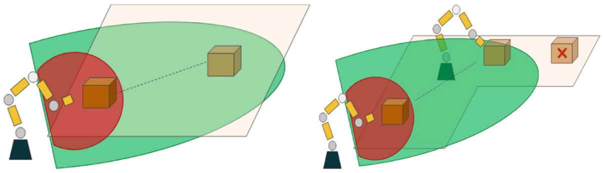


Fig. 5. Left: In red is the robot reachable workspace and in green is the object reachable space. Right: When the object cannot reach a desired position, we utilize multiple robots to achieve the desired result.

Fig. 6. In disc golf [18], the position '**T**' marks the place from where the player throws the first shot. The out of bound area (where the disc is not allowed to go) is marked by a white boundary. The shaded white area in the middle shows the area that a good disc throw will lead the disc to lie inside. The scoring average shows how many shots in average a player needed to reach the final goal position.

E-M algorithm. The GMM model for object 2 is initialized with the parameters of the object 1 model. The hypothesis is that learning a model through transferring a known model should require fewer data points. Model for object 2 is learnt with different number of data points sampled uniformly from the data and KL divergence is calculated with incrementally learned models and the model learnt from the entire dataset, cross-validated 5 times. Table VI shows the average KL divergence values. We see that the transferred model is already require fewer data points to be close to the actual model. Future research tackles understanding how to learn better with fewer data points.

V. Application: Object Motion Planning

The motion model of an object allows us to also model "*object reachable space*", i.e., the area that an object can reach after being hit, denoted by $\mathcal{O}$. In Fig. 5, in green we show the object reachable space and in red, the robot workspace. A robot can hit an object anywhere in the green space, and using multiple robots, we can increase the object reachable space, allowing placement of objects outside the object reachable space of just one robot. To position an object in an unstructured environment with multiple robots (or multiple hits) resembles the sport of golf, refer Fig. 6 where the first hit can be large inducing higher variability while the last hit is the most accurate hit.

A. Modeling the Object Reachable Space, $\mathcal{O}$

To generalize the object reachable space in any direction, the following assumptions are made:

1) The friction between the box and the sliding surface is isotropic, implying that the object reachable space of each robot remains orientation invariant.
2) The distance between the initial robot end effector and object initial position remains fixed, and so is the orientation of the object to that of the robot, represented as θ.

While the object motion can be modeled as a GMM, it also allows us to model the object reachable space. The object reachable space of each robot $\mathcal{O}_i$ is modeled using the GMM $P(\chi_f|\chi_i, \theta)$.

B. Optimization Problem

Problem statement: Determine the optimal orientation of the robot 1, $\mathcal{R}_1$, i.e., θ_1 around the initial position of the object $\chi_{o,i}$, the optimal mid-trajectory position of the object $\chi_{o,m}$, and the optimal orientation of the robot 2, $\mathcal{R}_2$, i.e., θ_2 around $\chi_{o,m}$, with the aim of having the object reach the final desired position, $\chi_{o,f}$, given the boundary function for the available space $\mathcal{B}$, the initial position of the object $\chi_{o,i}$, the target position of the object $\chi_{o,f}$ and the object reachable space for $\mathcal{O}_1$, $\mathcal{O}_2$.

1) Constraints: The object remains on the table throughout its motion $\chi_o \in \mathcal{B} \ominus \varepsilon$, where ε is a margin introduced to avoid the object falling off the table.

2) Objective Function: We implement and compare two different objective functions, namely, total optimization and golf optimization functions, and show how the golf optimization solution works on the real robots.

Total Optimization Objective Function: This approach aims to maximize the total probability of reaching the desired position. To achieve this, the objective function is defined as follows:

$$F_{\text{total}}(\theta_1, \chi_m, \theta_2) = P_1(\chi_m|\chi_i, \theta_1)P_2(\chi_f|\chi_m, \theta_2) \tag{13}$$

Golf Optimization Objective Function : Golf optimization aims to maximize the probability of the last hit, specifically maximizing the probability of reaching χ_f on the last hit. Additionally, a soft constraint (indicating an intersection between the mid-trajectory position χ_m and $\mathcal{O}_1$) is added to ensure that the probability of reaching χ_m is close to a certain threshold, (enabling robot 1 to hit the object to χ_m) and ensuring a feasible solution.

The objective function for golf optimization is defined as follows:

$$F_{\text{golf}}(\theta_1, \chi_m, \theta_2) = P_1(\chi_f|\chi_m, \theta_2) + cP_2(\chi_m|\chi_i, \theta_1) \tag{14}$$

Here, c refers to the weight the optimization function puts on the first hit.

Optimization Problem: Considering the assumptions, constraints and objective functions defined above, the final optimization problem is:

$$\begin{aligned} \theta_1^*, \chi_m^*, \theta_2^* &= \underset{\theta_1, \chi_m, \theta_2}{argmax} \quad F(\theta_1, \chi_m, \theta_2) \\ \text{s.t.} \quad & \chi_o \in \mathcal{B} \ominus \varepsilon \end{aligned} \tag{15}$$

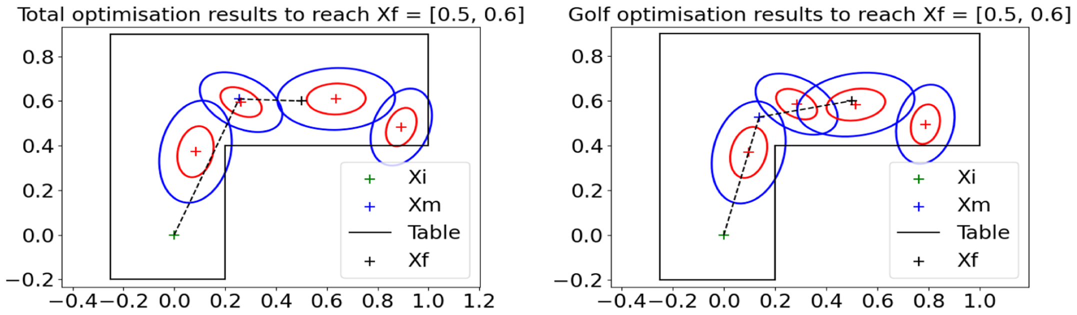


Fig. 7. The green cross represents the start position of the box and the black cross represents the final desired position of the box. The total optimization places the central object position closer to the mean of a Gaussian where as the golf optimization places the final desired position closer to the mean of the Gaussian representing the difference in likelihood of the final stroke.

Since the optimization function is either a product or a weighted summation of probability density represented by GMMs, this is a non convex problem. We use COBYLA solver that uses linear approximation for constrained optimization, which is efficient with a good initial guess but becomes less effective as the number of variables increases [19]. For n robots, the number of the variable to be optimized is $2n - 1$. COBYLA is generally preferred for only up to 9 variables [19], which limits the number of robots to 5. In terms of worst time complexity, linear programs are $\mathrm{O}(2^n)$, although given the iterative solver being used, the complexity is hard to define. Extensions of these ideas to dynamic programming for solving sequential problem is an area of future work.

C. *Robot Experiments*

Fig. 7 shows the optimization results and the difference between the total and golf optimization methods. The golf optimization places the Object reachable space GMM for the second hit such that the final position is close to the means of the Gaussians. The accompanying video contains two robots hitting an object to its desired location, where it cannot go through a single hit due to large distance and existence of an obstacle in its direct path.

VI. Discussion and Future Work

In this letter we learn the motion model of an object subject to an impulse by creating a probability distribution of hitting flux and object distance traveled. The data is collected through a semi-automated dual arm framework. The different kinds of objects, robots and robot configuration used in hitting allow us to understand better the difference in the distributions of the data when such entities in the system producing the hit are changed. An impact aware Extended Kalman Filter is introduced which circumnavigates the discontinuity in the motion of the object through a Gaussian hitting force model. The energy transfer equations is used for the force parameter selection. The learnt model is used in motion planning for movement of an object in an environment where one robot cannot hit the object to its desired location. The object motion model although incorporating the way an object has been hit does not yet incorporate a model of friction that is more complex than simple isotropic model allowing one to assume motion of the object will be the same regardless of any hitting direction. Other than incorporating transferability of the object motion to unseen objects, another future aspect is to control the pose of the object, thus including its orientation, which also is affected by the friction between the object and sliding surface.

Appendix A
Collision Mechanics

Through conservation of momentum, we have:[2]

$$\lambda_h(\dot{\chi}_{rh}^{-} - \dot{\chi}_{rh}^{+}) = m_o\dot{\chi}_{oh}^{+} \tag{16}$$

where $\lambda_h = (\hat{h}^T\Lambda^{-1}\hat{h})^{-1}$ [20] is the *directional effective inertia*[3] at the point of contact in the hitting direction $\hat{h}$, Λ is the effective inertia which generally can be calculated as: $\Lambda(q) = (J(q)M^{-1}(q)J^T(q))^{-1}$ [20] From the definition of restitution (ϵ) along the impact normal ($\hat{h}$),

$$\epsilon\ (\dot{\chi}_{rh}^{-}) = \dot{\chi}_{oh}^{+} - \dot{\chi}_{rh}^{+} \tag{17}$$

$$\dot{\chi}_{oh}^{+} = (1+\epsilon)\left(\frac{\lambda_h}{\lambda_h + m_o}\right)\dot{\chi}_{rh}^{-} \tag{18}$$

We write the directional hitting flux as a scaled post-impact object speed, independent of ϵ.

$$\phi_h = \frac{\lambda_h}{\lambda_h + m_o}\dot{\chi}_{rh}^{-} \tag{19}$$

The robot is controlled for a desired value of ϕ_h.

Appendix B
Parameter Selection for Force Function

The hitting force is modeled as a Gaussian, defined by a standard deviation parameter σ_2 and amplitude σ_1. Where δx is the distance between the hitting point on the object and the hitting point of the end-effector. It is assumed that the work of the hitting force over distance is equal to the kinetic energy of the box after impact such that,

$$\int_{-\infty}^{\infty} f_i(\Delta\chi)d\Delta\chi = \frac{1}{2}m\dot{\chi}_o^{+2}. \tag{20}$$

Substitution and rearranging yields,

$$\sigma_1 = \frac{m(1+\epsilon)^2\phi^2}{2\sigma_2\sqrt{2\pi}}. \tag{21}$$

[2] superscript $-$ represents pre-impact and $+$ represents post impact. Underscript h represents the hitting direction.

[3] $\lambda_h = \lambda_h(q), \Lambda_h = \Lambda_h(q)$ are functions of the joint configuration, q. At the impact, q is assumed to be constant.

Appendix C
EKF Prediction and Update Steps

The prediction step of the EKF consists of predicting the state and covariance of the system at the next time step. Given the current state estimate $\hat{\boldsymbol{x}}^{k|k-1}$, covariance estimate $\boldsymbol{P}^{k|k-1}$, and the system dynamics model:

$$\hat{\boldsymbol{x}}^{k|k-1} = f(\hat{\boldsymbol{x}}^{k-1|k-1}, u^k) \tag{22}$$

$$\boldsymbol{P}^{k|k-1} = \boldsymbol{F}^k \boldsymbol{P}^{k-1|k-1} \boldsymbol{F}^{kT} + \boldsymbol{Q}^k \tag{23}$$

where $f(\cdot)$ is the state transition function, $\boldsymbol{F}^k$ is the Jacobian matrix of f with respect to the state, and $\boldsymbol{Q}^k$ is the process noise covariance matrix. The update step of the EKF incorporates measurements to refine the state estimate. Given the predicted state $\hat{\boldsymbol{x}}^{k|k-1}$ and covariance $\boldsymbol{P}^{k|k-1}$, and a measurement $\boldsymbol{z}^k$, the update equations are:

$$\tilde{\boldsymbol{y}}^k = \boldsymbol{z}^k - h(\hat{\boldsymbol{x}}^{k|k-1}) \tag{24}$$

$$\boldsymbol{S}^k = \boldsymbol{H}^k \boldsymbol{P}^{k|k-1} \boldsymbol{H}^{kT} + \boldsymbol{R}^k \tag{25}$$

$$\boldsymbol{K}^k = \boldsymbol{P}^{k|k-1} \boldsymbol{H}^{kT} \boldsymbol{S}^{k-1} \tag{26}$$

$$\hat{\boldsymbol{x}}^{k|k} = \hat{\boldsymbol{x}}^{k|k-1} + \boldsymbol{K}^k \tilde{\boldsymbol{y}}^k \tag{27}$$

$$\boldsymbol{P}^{k|k} = (\boldsymbol{I} - \boldsymbol{K}^k \boldsymbol{H}^k) \boldsymbol{P}^{k|k-1} \tag{28}$$

where $\tilde{\boldsymbol{y}}^k$ is the measurement residual, $h(\cdot)$ is the measurement function, $\boldsymbol{H}^k$ is the Jacobian matrix of h with respect to the state, $\boldsymbol{R}^k$ is the measurement noise covariance matrix, $\boldsymbol{K}^k$ is the Kalman gain, and $\boldsymbol{I}$ is the identity matrix.

Appendix D
Gaussian Mixture Regression

Assume we have the joint probability distribution of the data which consists of both input and output. The probability that a data-point $\zeta = [O; I]$ (O being the output and I, input) belongs to a GMM is as follows:

$$P(\zeta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\zeta; \mu_k, \Sigma_k)$$

where π_k are the prior probabilities of the Gaussians of the GMM and $\mathcal{N}(\zeta; \mu_k, \Sigma_k)$ are the Gaussian distributions composing the GMM. μ_k and Σ_k are the means and covariance matrices of the k^{th} Gaussian and can be written as:

$$\mu_k = \begin{bmatrix} \mu_{Ik} \\ \mu_{Ok} \end{bmatrix}, \Sigma_k = \begin{bmatrix} \Sigma_{Ik} & \Sigma_{IOk} \\ \Sigma_{OIk} & \Sigma_{Ok} \end{bmatrix}$$

Once, we have the GMM, we compute the distribution of the output variable O, given the input variable I and Gaussian k

$$P(O|I,k) \sim \mathcal{N}(\hat{\mu}_k, \hat{\Sigma}_k)$$

where,

$$\hat{\mu}_k = \mu_{Ok} + \Sigma_{OIk} \Sigma_{Ik}^{-1} (I - \mu_{Ik})$$

$$\hat{\Sigma}_k = \Sigma_{Ok} - \Sigma_{OIk} \Sigma_{Ik}^{-1} \Sigma_{IOk}$$

Using the above equations, we can sum over all the Gaussians to generate conditional expectation of O, given I.

$$\hat{\mu} = \sum_{k=1}^{K} h_k \hat{\mu}_k, \qquad \hat{\Sigma} = \sum_{k=1}^{K} h_k^2 \Sigma_k$$

where,

$$h_k = \pi_k \frac{\mathcal{N}(I; \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \mathcal{N}(I; \mu_k, \Sigma_k)}$$

References

[1] H. Khurana and A. Billard, “Motion planning and inertia-based control for impact aware manipulation,” *IEEE Trans. Robot.*, vol. 40, pp. 2201–2216, 2024.

[2] M. Bauza and A. Rodriguez, “A probabilistic data-driven model for planar pushing,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3008–3015.

[3] F. H. M. Bauza and A. Rodriguez, “A data-efficient approach to precise and controlled pushing,” *CoRL*, 2018.

[4] K.-T. Yu, M. Bauza, N. Fazeli, and A. Rodriguez, “More than a million ways to be pushed: A high-fidelity experimental dataset of planar pushing,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, IEEE, 2016, pp. 30–37.

[5] C. Song and A. Boularias, “A probabilistic model for planar sliding of objects with unknown material properties: Identification and robust planning,” 2020.

[6] H. Khurana, M. Bombile, and A. Billard, “Learning to hit: A statistical dynamical system based approach,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 9415–9421.

[7] K. M. Lynch and M. T. Mason, “Stable pushing: Mechanics, controllability, and planning,” *Int. J. Robot. Res.*, vol. 15, no. 6, pp. 533–556, 1996.

[8] “Estimating the friction parameters of pushed objects,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1993, vol. 1, pp. 186–193.

[9] H. Kitano, M. Asada, I. Noda, and H. Matsubara, “Robocup: Robot world cup,” *IEEE Robot. Automat. Mag.*, vol. 5, no. 3, pp. 30–36, Mar. 1998.

[10] J. Tebbe, Y. Gao, M. Sastre-Rienietz, and A. Zell, “A table tennis robot system using an industrial kuka robot arm,” in *Pattern Recognition*, T. Brox, A. Bruhn, and M. Fritz, Eds. Cham, Switzerland: Springer International Publishing, 2019, pp. 33–45.

[11] S. M. Khansari-Zadeh, K. Kronander, and A. Billard, “Learning to play minigolf: A dynamical system-based approach,” *Adv. Robot.*, vol. 26, no. 17, pp. 1967–1993, 2012. [Online]. Available: http://infoscience.epfl.ch/record/181052

[12] Y.-B. Jia, M. Gardner, and X. Mu, “Batting an in-flight object to the target,” *Int. J. Robot. Res.*, vol. 38, no. 4, pp. 451–485, 2019, doi: 10.1177/0278364918817116.

[13] M. Müller, S. Lupashin, and R. D’Andrea, “Quadrocopter ball juggling,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 5113–5120.

[14] A. AlAttar, L. Rouillard, and P. Kormushev, “Autonomous air-hockey playing cobot using optimal control and vision-based Bayesian tracking,” in *Towards AutonomousRobotic Systems*, K. Althoefer, J. Konstantinova, and K. Zhang, Eds. Cham, Switzerland: Springer, 2019, pp. 358–369.

[15] P. Liu, D. Tateo, H. Bou-Ammar, and J. Peters, “Efficient and reactive planning for high speed robot air hockey,” *CoRR*, vol. abs/2107.06140, 2021.

[16] J. Jankowski, A. Marić, and S. Calinon, “Airlihockey: Highly reactive contact control and stochastic optimal shooting,” 2024, *arXiv:2401.14964*.

[17] E. Aljalbout, “Dual-arm adversarial robot learning,” in *Proc. 5th Annu. Conf. Robot. Learn., Blue Sky Submission Track*, 2021. [Online]. Available: https://openreview.net/forum?id=FI2HrMozlo_

[18] “Jomez productions,” https://www.jomezpro.com/

[19] M. J. D. Powell, “A direct search optimization method that models the objective and constraint functions by linear interpolation,” in *Proc. Adv. Optim. Numer. Anal.*, 1994, pp. 51–67. [Online]. Available: https://app.dimensions.ai/details/publication/pub.1046127469

[20] O. Khatib, “Inertial properties in robotic manipulation: An object-level framework,” *Int. J. Robot. Res.*, vol. 14, no. 1, pp. 19–36, 1995.