Hitting with Different Joints of a Robotic Manipulator

Harshit Khurana and Aude Billard

***Abstract*—This paper builds up the skill of impact aware non prehensile manipulation through a hitting motion by allowing the robot arm to come in contact with the environment with parts other than its end effector. Hitting with other joints allows us to manipulate heavier objects since the robot effective inertia is higher at joints other than its end effector. Preliminary work to align the robot to hit an object from a certain joint while having better understanding of desired directional inertia values is presented.**

***Index Terms*—dynamic manipulation, inertia, hitting, optimisation, impact aware manipulation**

I. Introduction

We have established the skill for object manipulation through the robot coming into contact at non zero relative speeds. We have shown that we can generate repeatable impacts on an object to move it from its position to another which lies outside the workspace of a fixed manipulator in our previous work [1], [2]. We call this manipulation through hitting. Humans are able to manipulate objects with more than just the ends of their fingers or the hand (analogous to the end effector of a robot arm). We can use elbows, the back of our palms, and even random points at our arms to move objects. In this paper, we extend the idea of hitting to use motion at joints other than the end effector of the robot to manipulate the object. While there are many potential ways to endow robots with the skills to use contacts throughout their body such as [3], there is a substantial lack of research in this area. This work aims to propose preliminary approaches for managing hitting throughout the robot linkage system. This preliminary work will be developed further in the future.

II. Preliminaries

In [2], we generate hitting motion using a dynamical system (refer Eq. 5), briefly described below, and control for *directional hitting flux*, ϕ_h (derived using collision mechanics, refer Eq. 4) through controlling for velocity and directional inertia of the robot. For a thorough description, refer [2]. The robot end effector position and velocity are represented as $(\chi_r, \dot{\chi_r})$ while the object position and velocity are represented as $(\chi_o, \dot{\chi_o})$.

A. *Collision Mechanics:*

Through conservation of momentum, we have[1]:

$$\lambda_h(\dot{\chi}_{rh}^{-} - \dot{\chi}_{rh}^{+}) = m_o \dot{\chi}_{oh}^{+} \tag{1}$$

where $\lambda_h = (\hat{h}^T \Lambda^{-1} \hat{h})^{-1}$ [4] is the ***directional effective inertia***[2] at the point of contact in the hitting direction $\hat{h}$, Λ is the effective inertia which generally can be calculated as: $\Lambda(q) = (J(q)M^{-1}(q)J^T(q))^{-1}$ [4] From the definition of restitution (ϵ) along the impact normal ($\hat{h}$),

$$\epsilon\,(\dot{\chi}_{rh}^{-}) = \dot{\chi}_{oh}^{+} - \dot{\chi}_{rh}^{+} \tag{2}$$

$$\dot{\chi}_{oh}^{+} = (1+\epsilon)(\frac{\lambda_h}{\lambda_h + m_o})\dot{\chi}_{rh}^{-} \tag{3}$$

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

$$\phi_h = \frac{\lambda_h}{\lambda_h + m_o}\dot{\chi}_{rh}^{-} \tag{4}$$

The robot is controlled for a desired value of ϕ_h.

B. *Pre-impact Dynamical System:*

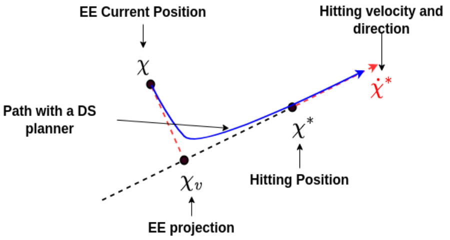

Fig. 1: The end effector (EE) moves toward its projection on the desired direction and a constant flow is added, thus forming the desired motion.

The current position of the robot end effector is given by χ_r and it needs to pass through χ_r^* with velocity $\dot{\chi}_r^*$ (refer Fig. 1). The DS is the reference velocity, $\dot{\chi}_r$ of the end effector of the robot. Consider the following DS:

$$\begin{aligned} \dot{\chi}_r = f(\chi_r) = \alpha(\chi_r)\dot{\chi}_r^* + (1 - \alpha(\chi_r))[A(\chi_r - \chi_{rv})] \\ f(\chi_r), \chi_r, \dot{\chi}_r, \chi_r^*, \dot{\chi}_r^* \in \mathbb{R}^3, \alpha(\chi_r) \in \mathbb{R}, A \in \mathbb{S}^3_{--} \end{aligned} \tag{5}$$

where,

$$\begin{aligned} \alpha(\chi_r) = e^{\frac{-\|\chi_r - \chi_{rv}\|}{\sigma^2}}, \quad \sigma \in \mathbb{R}_+ \\ \chi_{rv} = \chi_r^* + \frac{< \chi_r - \chi_r^*, \dot{\chi}_r^* >}{\|\dot{\chi}_r^*\|^2}\dot{\chi}_r^* \end{aligned} \tag{6}$$

$< .,. >$ is the inner dot product, $\chi_r, \dot{\chi}_r, \chi_r^*, \dot{\chi}_r^* \in \mathbb{R}^3$. χ_{rv} denotes a virtualized end effector. It is the projection of the current end effector position along the hitting direction. The tuning parameters depending on the robot and its workspace are σ. For its convergence and tuning, refer to [2]. Given a desired value of hitting flux, ϕ^*, we get $\dot{\chi}_r^* = \frac{\lambda_h + m_o}{\lambda_h}\phi^*$.

All authors are with the Learning Algorithms and Systems Laboratory, EPFL, Lausanne, Switzerland, e-mail: {firstname.lastname}@epfl.ch.

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

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

C. *Specific Directional Inertia in the Null Space:*

To achieve a desired directional inertia λ^* while tracking the dynamical system, with priority to the latter, we used the following control system.

$$\begin{aligned}\dot{q} = J^{\dagger}(q)f(\chi_r) + N[\beta_1(q_m - q)+ \\ \beta_2(-\nabla_q \lambda_h(q)(\lambda_h(q) - \lambda^*))], \quad \beta_1, \beta_2 \in \mathbb{R}_+ \quad (7)\end{aligned}$$

Here, we used IK[3] to find a joint configuration at the desired position in space and used the directional inertia at that configuration as a reference (λ^*), q_m is a maximum manipulability [5] configuration at the desired hitting position, $J^{\dagger}(q)$ is the Jacobian pseudoinverse, $N = I - J(q)^{\dagger}J(q)$is a null space projector, and β_1, β_2 are hyperparameters.

III. Problem Formulation

Let's take a look mathematically at what does hitting with other joints allow:[4]

A. *Maximum achievable hitting flux:*

Consider joints, $k \in \{5, 6, 7\}$. Let's assume that we are capable of hitting an object by the k^{th} joint. Here, we would like to calculate the maximum hitting flux obtained, hitting an object of mass $m_o = 2kg$ at the location, χ_o when hit with different joints. For each of the the specific cases, we write joints from $q_h = \{1, ..., k-1\}$ and $q_{nh} = \{k, ..., d\}$, where d is the DoF of the robot arm. q_h represents the joint configuration of the joints before the hitting joint, k, and q_{nh} represent the joint configuration of the rest of the joints. Given object position, χ_o, we use IK for placing different joints of the robot at that cartesian position for the joint, k.

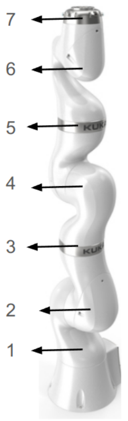

Fig. 3: Kuka lbr iiwa joints

$$FK(q_h) = \chi_o \ \ \forall k \in \{5, 6, 7\} \quad (8)$$

For proof of concept we use the same null space joint configuration while implementing IK at different joints, so that the final joint configurations are close to each other in joint space. For each k, then we move the k^{th} joint thus changing q_{nh}. This changes the directional hitting inertia, while keeping the maximum directional hitting velocity constant, and the hitting joint at the object position. Maximum hitting velocity at a specific joint configuration can be formulated as a linear program:

$$\begin{aligned}\dot{\chi}_{rh_{max}} &= \max_{\dot{q}} \hat{h}^T J(q)\dot{q} \\ s.t. \quad &\dot{q}_{min} \leq \dot{q} \leq \dot{q}_{max}\end{aligned} \quad (9)$$

Thus, we can write the maximum hitting flux with different joints hitting a box at a given position as follows:

$$\phi_{max} = (1 + \frac{m_o}{\lambda})\dot{\chi}_{rh_{max}} \quad \forall q_h \in (q_{h_{min}}, q_{h_{max}}) \quad (10)$$

The plots for the directional inertia, maximum hitting speed and the maximum flux are shown in Fig. 2. Here, when the mass of the object is of similar magnitude as the robot's directional inertia, the maximum hitting flux reduces. Joint 6 is able to produce stronger hits as opposed to hitting using joint 7. But we observe that joint 5 is also capable of producing relatively strong hits.

Point of interest: Kinematically, the cartesian velocity of a joint k does not depend on the configuration of the joints ahead of it in the kinematic tree, i.e. $\{k+1, ..., d\}$. Thus, if we would like to hit an object with the joint k, we can utilise the joints ahead of it in the kinematic tree to influence the effective inertia of the robot at the hitting joint. This fact will be exploited in Section IV-B.

B. *Robot's hitting and rebound velocity difference:*

With the Eqns 1 and 3, we can write

$$\dot{\chi}^+_{rh} - \dot{\chi}^-_{rh} = -(1+\epsilon)(\frac{m_o}{\lambda_h + m_o})\dot{\chi}^-_{rh} \quad (11)$$

This further extends the motivation to achieve a desired flux with higher robot's directional inertia to lower the difference between hitting and rebound velocity of the robot.

Thus, we have the following goals:

1) To determine the desired inertia values during the hitting motion (Eq. 5) so that the robot is not pushed towards an infeasible desired inertia value.
2) To determine the configuration of the robot to be able to hit an object with joints other than its end effector.

IV. Methods + Preliminary Results

A. *Desired Inertia Calculation (λ^*):*

While following the motion outlined by Eq. 5, and controlled through Eq. 7, in parallel, we run an optimisation as follows, which shows potentially what could be the next achievable joint velocities of the robot, following the motion 5 with flux and joint limit constraints.

For each control loop (running in time dt), we have

$$\begin{aligned}\dot{q}_t = \ &\underset{\dot{q}}{\operatorname{argmin}} \ \|J(q)\dot{q} - f(\chi)\|^2_W \\ \text{s.t.} \quad &\phi_h = \phi^* \\ &q_l \leq q_t + \dot{q}_t dt \leq q_u \\ &\dot{q}_l \leq \dot{q}_t \leq \dot{q}_u\end{aligned} \quad (12)$$

The joint velocities are then integrated to find the next potential joint configuration $q_{t+dt} = q_t + \dot{q}_t dt$, through which we can find λ^* for the next control loop.

$$\lambda^*_{t+dt} = (\hat{h}^T \Lambda^{-1}(q_{t+dt})\hat{h})^{-1} \quad (13)$$

which is fed into the Eq. 7 every control loop. W can be the identity matrix $\mathbb{I}_7$, or can be chosen in a way that is related to the sensitivity of the inertia of the robot to the changes in the joint angles. For e.g. $W = diag(\nabla_q(\lambda(q))$.

[3] IK = inverse kinematics, FK = forward kinematics function

[4] All simulations are setup in pybullet [6], with osqp [7] as solver for QP problems

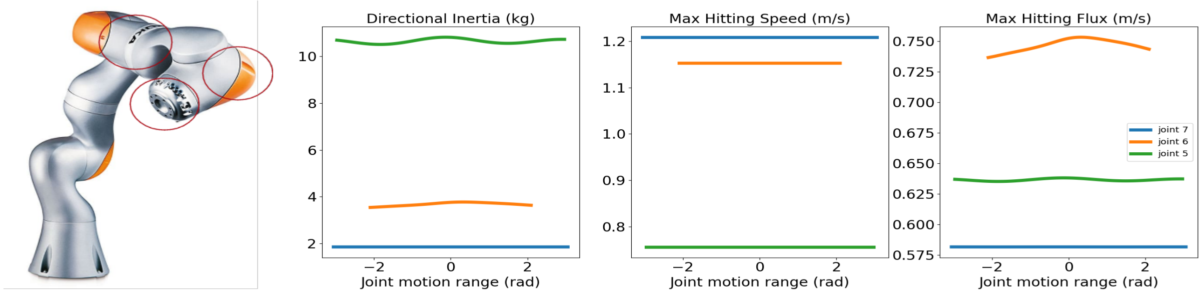

Fig. 2: Encircled are the joints of the KUKA lbr iiwa are that we are focussing on. The graphs depict the the directional inertia for each joint, maximum cartesian hitting speed at each joint and the maximum hitting flux achieved at the encircled joints.

1) Simulation Results: We compare the desired and achieved inertia values during the hitting motion for 1) a constant reference for λ^* and 2) reference from Eq. 13 while following the motion given by Eq. 5 controlled through Eq. 7. This is shown in the Fig. 4.

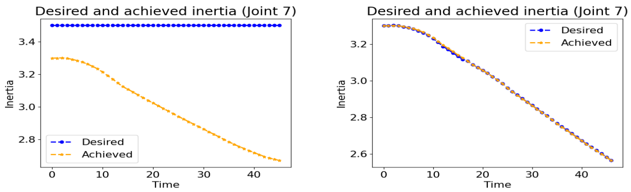

Fig. 4: Even though the robot's inertia evolution is almost the same, we are able to influence it through understanding the desired inertia values taking into account the motion and desired flux value.

B. *Hitting motion planning with different joints:*

As described previously in Section III-A, the joint configuration of the robot is written as $q = \{q_h, q_{nh}\}$, separating the joint configurations at the hitting joint. We would like to hit the object at a given position by joint $k \in \{5, 6, 7\}$. We keep the planning for the motion of the k^{th} joint governed by the dynamical system through Eq. 5. The configuration of the joints from k to d, i.e. q_{nh} can be optimised using different objective functions and it would not affect the cartesian speed of that joint. Here, we use a simple objective function to align the joints to maximise the directional inertia of the robot in the hitting direction and away from the object (through the dot product of hitting direction $\hat{h}$ and vector aligning the cartesian position of joints k (χ_k) and $k+1$ (χ_{k+1})).

$$\begin{aligned} q_{nh} = \ & \underset{q_{nh}}{\text{argmax}} \ \lambda_h(q) \\ \text{s.t.} \quad & q_{nh_l} \le q_{nh} \le q_{nh_u} \\ & \hat{h} \cdot (\chi_{k+1} - \chi_k) < 0 \end{aligned} \tag{14}$$

The results of the optimisation are quite sensitive to the initial configuration of the robot.

1) Simulation Results: Given an initial cartesian position for the robot to start its motion, we look at the configuration $q = q_h, q_{nh}$ given by Eq. 14. The initial configurations are shown in Fig. 5. Once the robots are allowed to follow the hitting motion, producting the same flux, we compare the directional inertia at the joint at the time of hitting. The directional inertia when hit by joint 6 (3.9 kg) is almost 56 % higher than that for joint 7 (2.5 kg).

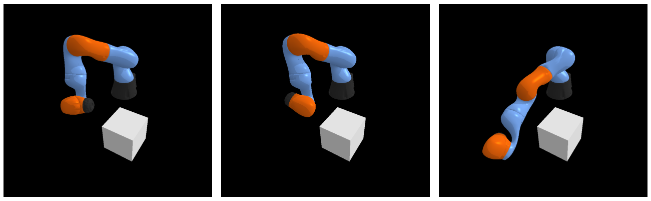

Fig. 5: The robot's initial configuration to hit the object with its joint 7, 6 and 5 respectively. All the joints start from the same initial cartesian position.

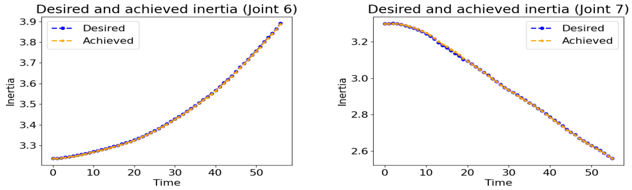

Fig. 6: The plots of the directional inertia when the box is hit by joint 7 and joint 6 are shown. The directional inertia when hit by joint 6 (3.9 kg) is almost 56 % higher than that for joint 7 (2.5 kg)

V. Discussion and Future Work

Current ongoing work expands this framework to include a learning strategy for understanding the feasibility of producing a desired hitting flux with different joints. A distribution over the robot configurations should help jump start the optimisation for hitting configuration that generates the desired hitting flux at the desired cartesian position of the object. A previous work [8] focuses on efficient data collection that helps us create the above desired probabilistic models.
Another point to be considered is the feasible motion at different joints of the robot. For e.g., joint 4 of a KUKA lbr iiwa arm is restricted to move on a sphere centered at joint 2. This restricts the direction of hit that is feasible by joint 5. It can only hit in the direction tangential to he radial distance between the object and joint 2 of the robot.

References

[1] H. Khurana, M. Bombile, and A. Billard, "Learning to hit: A statistical dynamical system based approach," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 9415–9421.

[2] H. Khurana and A. Billard, “Motion planning and inertia-based control for impact aware manipulation,” *IEEE Transactions on Robotics*, pp. 1–16, 2023.

[3] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake, “Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models,” *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4691–4711, 2023.

[4] O. Khatib, “Inertial properties in robotic manipulation: An object-level framework,” *The International Journal of Robotics Research*, vol. 14, no. 1, pp. 19–36, 1995.

[5] T. Yoshikawa, “Manipulability of robotic mechanisms,” *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985. [Online]. Available: https://doi.org/10.1177/027836498500400201

[6] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning,” 2016.

[7] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, “OSQP: an operator splitting solver for quadratic programs,” *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. [Online]. Available: https://doi.org/10.1007/s12532-020-00179-2

[8] F. Khadivar, S. Gupta, W. Amanhoud, and A. Billard, “Efficient configuration exploration in inverse dynamics acquisition of robotic manipulators,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1934–1941.