Onboard Sensing and Pushing of Unknown Payload for CoM Estimation with a Holonomic Mobile Robot

Steven M. Hyland, Jing Xiao, and Cagdas D. Onal

***Abstract*—This paper presents a novel approach for estimating the center of mass (CoM) direction of unknown payloads using only onboard sensing - namely, a force sensor and an RGB camera - on a mobile robot. Unlike methods requiring external infrastructure, extensive datasets, or machine learning, our technique employs an active perception framework to guide adaptive pushing of the payload and employs a robust search algorithm to find the CoM direction. By eliminating the need for prior inertial knowledge or global pose information, the proposed method converges on a translational line of action (LoA), indicated by zero rotational motion about the CoM. The method is validated on payloads of varying shape, size, and mass distribution, demonstrating consistent CoM estimation accuracy. Overall, this approach offers a reliable and adaptive solution for mobile robotic manipulation in environments where external global sensing is unavailable or impractical.**

I. Introduction

Handling payloads is a fundamental task in robotics, particularly within warehouse automation. The primary challenge lies in moving a payload to a target pose or state, often under constrained motion, such as navigating narrow aisles or congested work areas. While start and end states are often well-defined, designing a continuous trajectory that adheres to operational constraints, such as avoiding collisions and maintaining stability, is notably complex without some key information about the payload. In this context, the payload's center of mass (CoM) is of particular importance. With a known CoM, a robot can make robust decisions to remain within specified boundaries, plan more efficient paths, avoid collisions, and better anticipate the payload's behavior during manipulation, leading to increased throughput and safety in dynamic warehouse environments.

However, accurately estimating a payload's CoM remains a significant challenge. This difficulty is amplified when relying on limited local or onboard sensor data, which is inherently prone to noise. In warehouse automation, mobile robots are responsible for transporting payloads across the facility. Consequently, equipping mobile robots with onboard CoM estimation capabilities can provide more efficient, controlled, and safe approaches for payload transportation.

In this paper, we introduce a robot system that does not require global sensed information of the payload's pose, such as through an externally mounted camera system, but uses only the robot's onboard force sensor and camera feedback to estimate the payload's CoM direction (e.g. line of action).

Simpler optimization techniques that rely solely on model-based predictions would struggle with the inherent uncertainties of unknown payload properties and the presence of signal noise. Our approach, therefore, combines exploratory pushes with a modified binary search to create an active perception and optimization strategy for CoM direction estimation.

Pushing is more common for manipulating objects in contrast to pulling because it avoids the need for specialized end-effectors and does not require a prehensile attachment to the payload. Early foundational research by Mason et al. [1], Akella et al. [2], and Lynch et al. [3] explored open-loop control schemes and multi-point contacts using stable fences for pushing.

Subsequent efforts [4], [5], [6] introduced tactile feedback and active sensing for robotic manipulators, enabling inertial parameter estimation and more advanced planning. Building on this, researchers have also incorporated visual feedback [7] and combined it with tactile sensing [8] to enhance dynamic modeling of the pushed objects. In parallel, force/torque sensors on robot arms [9], [10], [11] have enabled direct measurements of inertial parameters, such as mass, friction, and CoM.

Despite this progress, much of the literature assumes either a fixed manipulator or external sensing infrastructure. Fixed manipulators are inherently more accurate than their mobile counterparts due to a fixed reference frame, which also reduces drift and provides more accurate motions. However, fixed manipulators have a limited workspace and are incompatible with transportation over long distances. Gao, et al. [12], [13] introduced an approach that uses visual feedback and a robot arm to infer payload CoM. In their most recent work [14], the authors switch to a local robot-mounted camera, reducing the reliance on global sensing. Using a manipulator arm leads to more precise robot motion, but their method is constrained by the manipulator's workspace and cannot transport a payload to a distant destination. Heins et al. [15] demonstrated single-point pushing with a mobile robot but circumvent CoM estimation by exploiting known payload properties.

More recently, large-scale learning-based methods have been applied to pushing [16]–[21] but they typically require extensive datasets for training, and have not yet been tailored to the specific challenges presented by mobile robot pushing. A comprehensive survey by Stüber et al. [22] underscores the breadth of robotic pushing research, noting that while pushing broadens a robot's manipulation capabilities, inertial

This material is based upon work partially supported by the National Science Foundation (NSF) under Grant No. DGE-1922761. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

The authors are affiliated with the Robotics Engineering Department, Worcester Polytechnic Institute, MA 01609, USA. All correspondence should be addressed to Cagdas Onal cdonal@wpi.edu.

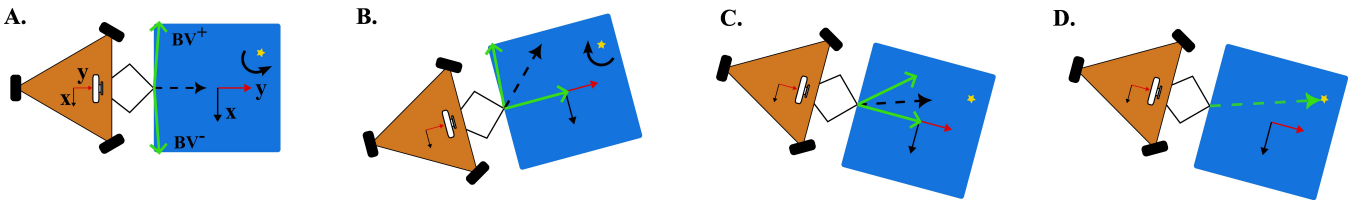

Fig. 1. Line of action estimation procedure. The robot iteratively pushes the payload and converges to the LoA passing through the CoM. Once minimal rotation is observed, the algorithm terminates. The robot and payload frames are defined as in (A), with the CoM positioned at the yellow star. The green arrows represent the bounds of the search, and converge to the estimated LoA (D). These bound vectors are updated based on the sensed *relative* rotation between the robot and payload, given by the combined force and camera information. The black dashed arrows in (A, B, C) presents the robot's executed push vector given from bisecting the two bound vectors.

parameter estimation and CoM estimation remain comparatively less explored areas.

Earlier work by McGovern et al. [23] applies force/torque sensing with a manipulator to predict inertial parameters for elongated objects, while more recent efforts [24] estimates a payload's CoM via a holonomic mobile robot and external motion capture. While motion capture provides full state feedback of the environment, it is far from a realistic representation of robotic manipulation in practice. The motion capture requires extensive, costly infrastructure which is prone to disturbances and requires frequent recalibration.

In contrast, the work presented in this paper eliminates reliance on external global sensing by focusing on CoM estimation with onboard sensing. This shift is motivated by warehouse automation scenarios, where mobile platforms are already tasked with moving payloads over larger distances in constrained corridors. By enabling onboard CoM estimation, we aim to provide robust, infrastructure-free manipulation capabilities without requiring a large workspace footprint, thereby generalizing push manipulation to novel objects under more realistic operational conditions.

II. Approach

This section describes our approach for conducting CoM line of action estimation. The robot in this paper is not privy to the payload's shape, material composition, nor inertial properties. The robot is equipped with just two sensing modalities: an onboard force sensor that provides vectorized 2-D force feedback and an onboard RGB camera. The payload is attached to the robot at a point along its edge by means of a pin joint located at the sensor's tip; thus allowing for free rotation of the object with respect to the robot.

For the purposes of validating and examining the efficacy of the estimation procedure, some foundational assumptions apply throughout this paper:

Non-slip condition exists between robot and payload. While the pin joint introduces a degree of prehension, it serves as a practical trade-off to ensure stable interaction during the pushing phase. By constraining lateral motion, the pin joint (1) guarantees persistent contact for reliable force transmission, (2) simplifies the dynamics model by eliminating slip/stick transitions between the robot and the payload, and (3) allows the robot to focus on resolving longitudinal uncertainties (e.g., CoM location) rather than lateral stabilization.

All robot pushes and resulting motions are quasi-static. All robot pushes and resulting payload motions are assumed to be quasi-static, yielding negligible inertial effects versus the applied and frictional forces. While it is not explicitly prohibited for this work, dynamic pushes provides far inferior stability as compared to quasistatic pushing.

Isotropic Coulomb friction is exhibited by the payload. For accurate determination of an object's Center of Mass (CoM), the presence of uniform and isotropic friction is paramount. This condition ensures that the center of friction (CoF) precisely coincides with the CoM, a critical factor for the algorithm's convergence to the true CoM rather than the CoF. While achieving perfectly smooth and uniform contact surfaces between the object and the table is challenging in practice, the robustness of the employed algorithm mitigates the effects of minor irregularities or localized friction variations.

The center of mass falls within the initial search bounds. The initial search bounds are established tangent to the payload's surface at the point of contact during pushing. This definition inherently excludes the half of the workspace proximal to the robot, based on the assumption that the CoM is initially situated within the half-plane directly ahead of the robot. Consequently, this approach does not accommodate non-convex objects, as their CoM may reside outside these predefined initial search boundaries.

The line of action is defined by the line that passes through both the CoM and the robot's point of attachment to the payload. Upon the robot estimating this LoA, the payload undergoes pure translation.

Key contributions include: (1) an onboard sensing strategy that replaces infrastructure-bound, tethered systems with local vision and force sensing, and (2) an adaptive pushing policy that ensures pure translational motion of the payload in the presence of motion and sensing uncertainty.

A. Force-based LoA Estimation

The proposed LoA estimation is inspired by Mason's Voting Theorem (VT) [1], wherein the quasistatic motion assumption indicates that frictional forces, rather than inertia, predominantly determine the motion of the payload. Under this assumption, the center of friction (CoF) coincides with

the CoM. According to VT, the CoM can be determined by directly observing the payload's rotation under some robotic manipulation. The work in this paper, conversely, leverages information embedded in *relative* forces and behaviors without relying on explicit payload rotation from an external frame of reference.

1) Search Bounds: Our LoA search algorithm, a form of optimization, is a modified binary search algorithm described in Algorithm 1. We consider the bounds for the binary search as *Bound Vectors* (BVs), which are rays starting from the payload's push point and extend to infinity. These are represented as $BV^{\pm}$, and are illustrated in Figure 1A. by the green arrows. These bound vectors are initialized approximately $\pm 90°$ from the initial push direction, which is in the robot's local y-direction. More specifically, they are initialized to $BV^{+} = [1, \epsilon]^T$ and $BV^{-} = [-1, \epsilon]^T$ where $\epsilon << 1$. A computationally efficient bound update (bisection) method requires the introduction of ϵ towards simple BV summation, $new\ push = BV^{+} + BV^{-}$.

2) Relative Sensing Modality: The proposed relative payload rotation criterion leverages local force-torque measurements to determine which half of the search space to explore and which to discard. We rely on changes in the force angle to determine the LoA, and when it is found, pushing along the LoA results in pure translation and a static force reading. The signed relative force angle is calculated using the four-quadrant inverse tangent method, *atan2*.

The stop criterion for this LoA search is based on the local angular rate of change $|\Delta\theta_p| > \Delta\theta_{thresh}$. The while loop in algorithm 1 continues until a minimal threshold absolute rotation between iterations is recorded. Once the stop condition is reached, the final LoA is recorded. Figure 1 illustrates the LoA search in action.

v_{des} is the desired speed of the robot in mm/s and its value has negligible effect on the system, as long as the speed is low enough to yield quasi-static motion. In testing, our robot's top speed and acceleration is low enough to always yield a quasi-static state. All vectors besides $BV^{\pm}$ and all sensor data is taken with respect to the robot center frame.

B. Attitude Control

The robot is operated under velocity control within its three degrees of freedom. Linear motion commands are issued as a planar velocity vector, $\vec{u}(t) \in \mathbb{R}^2$. The magnitude of this velocity is kept constant, adhering to predefined search boundaries.

While $\vec{u}(t)$ is two dimensional, the remaining control authority governs the robot's yaw. This yaw control, implemented using a null-space PID controller based on our prior work [24], prevents robot-payload collisions.

Payload relative orientation is measured using an onboard camera. As shown in Figure 1A, the robot is initialized to match the payload's frame. The attitude controller then minimizes deviations from this initial setpoint. Attitude error, $e_\theta = \theta_0 - \theta_p$ is then input to a PID attitude controller to generate a commanded angular velocity, ω_r. PID gains (1.25, 0.05, 0.20) were empirically tuned to robustly maintain

Algorithm 1 LoA Adaptive Search Algorithm

procedure LOASEARCH($\Delta\theta_{thresh}$, v_{des}, Δt)
  Initialize $\epsilon << 1$
  Initialize $\overrightarrow{BV}^{+} \Leftarrow [1, \epsilon]^T$
  Initialize $\overrightarrow{BV}^{-} \Leftarrow [-1, \epsilon]^T$
  Initialize $\theta_{prev}, \theta_{rel} \leftarrow 0$
  PUSHSTRAIGHT(Δt) */* Induce payload motion: push straight ahead for a small period */*
  while $|\Delta\theta_p| > \Delta\theta_{thresh}$ **do**
    $\vec{f_s} \leftarrow force\ vector$
    $\vec{f_r} \leftarrow prior\ push$
    $\Delta\theta_p \leftarrow \theta_{rel} - \theta_{prev}$
    $\theta_{prev} \leftarrow \theta_{rel}$
    $\theta_{rel} \leftarrow$ ATAN2($|\vec{f_r} \times \vec{f_s}|$, $\vec{f_r} \cdot \vec{f_s}$) */* Get relative angle between detected force and prior push direction */*
    if $\theta_{rel} < 0$ **then**
      $\overrightarrow{BV}^{+} \leftarrow \widehat{f_r}$
    else
      $\overrightarrow{BV}^{-} \leftarrow \widehat{f_r}$
    end if
    $\vec{u_r} \leftarrow v_{des} \cdot (\overrightarrow{BV}^{+} + \overrightarrow{BV}^{-})$
    MOVEROBOT($\vec{u_r}$)
  end while
  $LoA \Leftarrow \vec{u_r}$ */* Record final push direction /**
end procedure

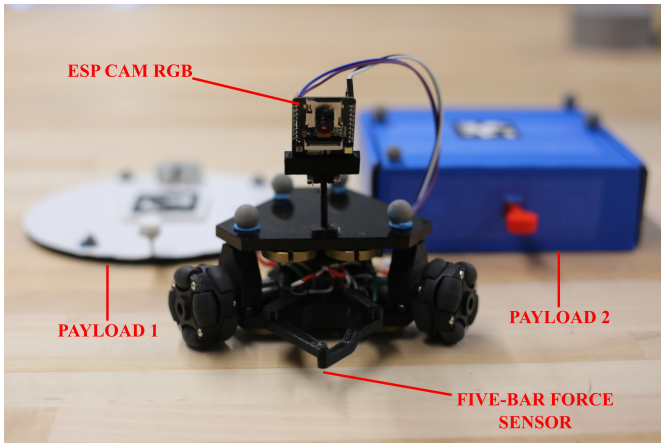

Fig. 2. The Delta-Rho holonomic mobile robot [25] with onboard force sensor and RGB camera, with the two tested payloads behind.

the setpoint for this specific robot hardware across varying payload geometries and dimensions.

A set velocity of $v_{des} = 90$ mm/s was selected to maintain quasi-static motion. An initial push period of $\Delta t = 0.5s$ was chosen to induce payload motion without offending the starting conditions. The threshold rotation rate, $\Delta\theta_{thresh} = 3°$, was selected as the minimum viable threshold for robust estimates.

C. Experiment Setup

The holonomic robot platform used in this study is adapted from a previous collective transport swarm platform [25]. We build upon this *delta-rho* platform, but now equipped with an

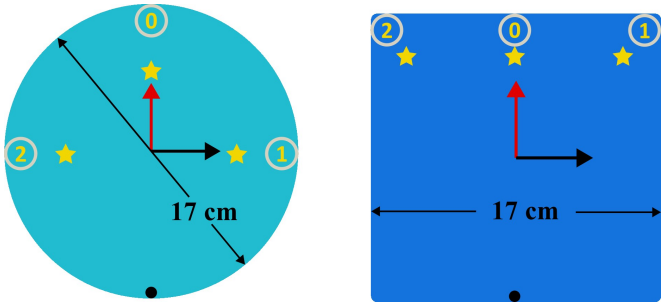

Fig. 3. CoM positions for payload 1 (left) and payload 2 (right). Each numbered circle represents placement of the weight for that CoM position. A $50\,g$ weight for payload 1 and $145\,g$ weight for payload 2 are used to keep net payload weight equivalent. Each star is the corresponding actual CoM position, when taking the weight of the payload itself into account. The black dot at the bottom of each is the robot attachment point.

in-house developed five-bar force sensor and an RGB camera. The force sensor measures both the magnitude and direction of forces applied at the end effector. For visual reference, see Figure 2.

The onboard camera is an ESP32 CAM module which streams real-time footage over WiFi and is used primarily for attitude control and for monitoring the stop condition, $|\Delta\theta_p| > \Delta\theta_{thresh}$, for the algorithm. Each frame is captured by the module and streamed to a local PC, where a lightweight Python script processes the images. This script detects the AprilTag on the payload [26] returning its orientation, or yaw, relative to the robot's frame. The orientation data is transmitted back to the camera, which sends it to the robot via an I2C interface. This process is optimized for a refresh rate exceeding 20 Hz.

For the experiment, two payloads were tested: a $175\,g$ circular disc with a surface area of $227\,cm^2$ and an $80\,g$ cardboard box with a surface area of $289\,cm^2$. These payloads are numbered 1 for the circular disc and 2 for the box for the remainder of this study. A $50\,g$ and $145\,g$ weight were placed at three distinct positions on payload 1 and 2, respectively to create different CoM locations. Figure 3 illustrates the two payloads and the selected CoM positions. This configuration allows us to evaluate the accuracy of the LoA estimate against varying mass distributions. For each CoM position, 10 trials were conducted to evaluate consistency across experiments. It is important to note that the actual CoM location shifts towards the centroid of the payload, away from the weight placement, due to the mass of the payload itself. Position zero acts as a control in which the CoM is aligned with the initial push vector. Although the robot does not use onboard localization, an Optitrack motion capture system was employed to validate results.

III. Experimental Results

A. Analysis of LoA Estimate Footprints

During each of the 10 LoA estimation trials per CoM position per payload, we recorded the convex region encompassing the entire pushing procedure. Because floor space is often limited in manufacturing environments, minimizing the footprint is critical. Due to the lack of studies using mobile robots for CoM or LoA estimation, we present our findings of footprint as a percentage of the payload size. Table I summarizes the total footprint required for each payload. Note that the footprint does not include the robot. In Figure 4, the cyan, magenta, and green shaded areas correspond to CoM positions 0, 1, and 2, respectively, and also illustrate the robot's and payload's paths throughout the manipulation.

TABLE I
Algorithm Footprint of Each Payload

Payload	Surface Area	CoM	Footprint (cm^2)	% of payload
1	227 cm^2	0	44	17.6%
		1	386	171.8%
		2	339	149.7%
2	289 cm^2	0	261	89.9%
		1	701	242.2%
		2	697	242.2%

Only the control configuration - CoM 0 - produced a footprint smaller than both payloads' top-down surface area. By contrast, all other CoM positions yielded footprints substantially larger than the payload's surface area. Additionally, a trend is observed of increasing footprint as the CoM moves farther from the attachment point. This finding suggests that whenever possible, selecting an attachment point closer to the suspected CoM can help maintain a more compact algorithmic footprint.

At the start of each trial, the payload was manually reset to the global origin, introducing slight variations in initial position and orientation. The distribution of paths for each configuration, as shown in Figure 4, reflects the impact of these small deviations. Nevertheless, the algorithm is designed to robustly estimate the LoA and eliminate any rotational motion.

As expected, the smallest footprint occurs at CoM position 0, indicating that when the CoM aligns closely with the robot's initial push vector, the system requires fewer corrective maneuvers. In contrast, the larger footprints at CoM positions 1 and 2 suggest increased variability, owing to a greater offset between the CoM and the point of pushing. This effect is especially pronounced for payload 2, which has a larger surface area and where the CoM was placed farther from the attachment point, increasing the moment of inertia and thereby resulting in larger, faster rotational effects.

B. LoA Estimation Performance

For the following analysis, we focus on the latter part of each trajectory as the LoA estimate converges. Since most of the initial rotational motion is overcome by the midpoint of the path (roughly at y=20cm), our inspection focuses on the data from this point onward.

To assess the linearity of the trajectory data, we fit a straight-line approximation for each run of each CoM position on each payload. From this we calculated the coefficient of determination (R^2) and standard deviation (STD).

The R^2 value measures how well the data fits the linear approximation - in other words, how effectively the algorithm eliminates payload rotation. It is calculated as 1 minus the sum of squared residuals over the sum of squares. The results

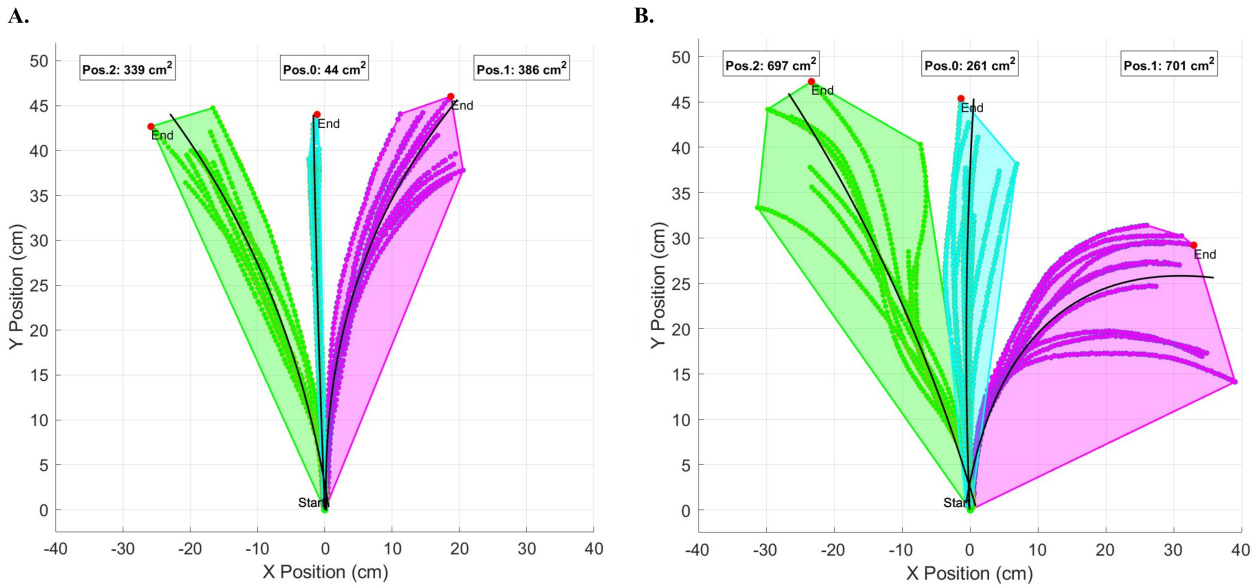

Fig. 4. Payload trajectory and footprints of each CoM position for payload 1 (A) and 2 (B). The black line is a 2nd order polynomial regression across each trajectory to visualize the spread of trajectories. The convex hull is calculated across all ten trials of each configuration; cyan, magenta, and green for positions 0, 1, and 2, respectively.

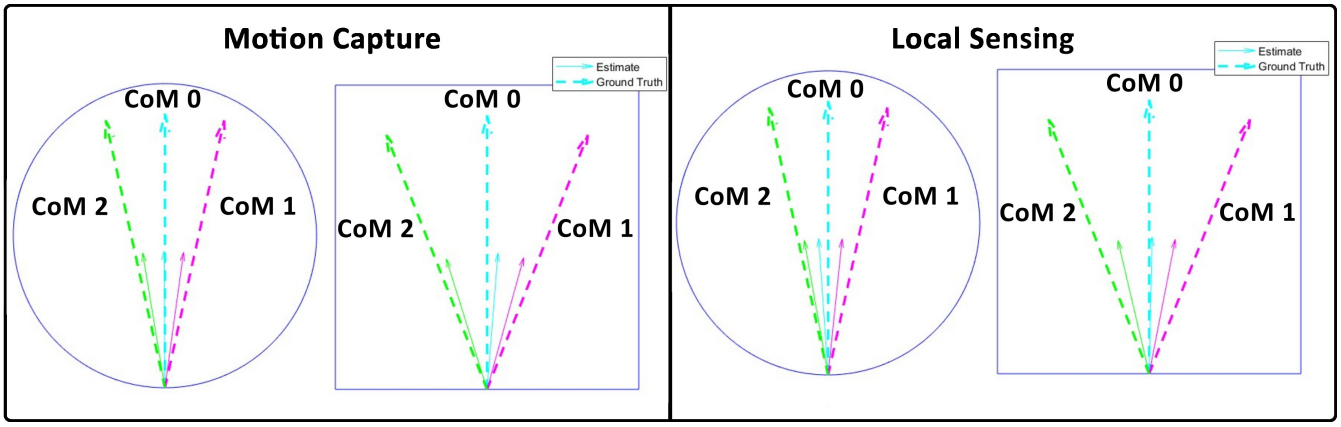

Fig. 5. Comparison of the baseline motion capture study (left) versus the local sensing (right) of this work. Cyan, magenta, and green lines are for positions 0, 1, and 2, respectively. The large dotted line represents the ground truth vector pointing at the CoM for that specific payload. The smaller solid lines are the corresponding estimates for that CoM position.

TABLE II
LINE OF ACTION ESTIMATION PERFORMANCE

Payload	CoM Position	Trials	R^2	STD (°)
1	0	10	0.93	1.71
	1	10	0.99	9.60
	2	10	0.96	2.92
2	0	10	0.99	3.85
	1	10	0.79	15.7
	2	10	0.91	11.3

TABLE III
THIS WORK VS BASELINE

Algorithm	Payload	CoM	Trials	Mean Error (°)	STD (°)
Baseline (motion capture)	1	0	10	-0.36	1.06
		1	10	-4.67	1.43
		2	10	3.14	1.12
	2	0	10	4.62	5.34
		1	10	-6.08	4.08
		2	10	4.27	5.29
This Work (local sensing)	1	0	10	-3.97	2.80
		1	10	-6.60	2.06
		2	10	2.47	1.98
	2	0	10	1.24	3.60
		1	10	-10.64	2.50
		2	10	8.44	6.78

in Table II indicate a very close approximation to a straight line by the estimation algorithm, with almost all values above 0.9. An $R^2 = 1$ would be a perfectly straight line. The notably lower R^2 value for payload 2, position 1, points to specific instances of significant trajectory deviation that we categorize as 'outliers'. These are primarily attributed to two factors: an amplified manual initialization bias and the increased moment-arm effect from the CoM's placement

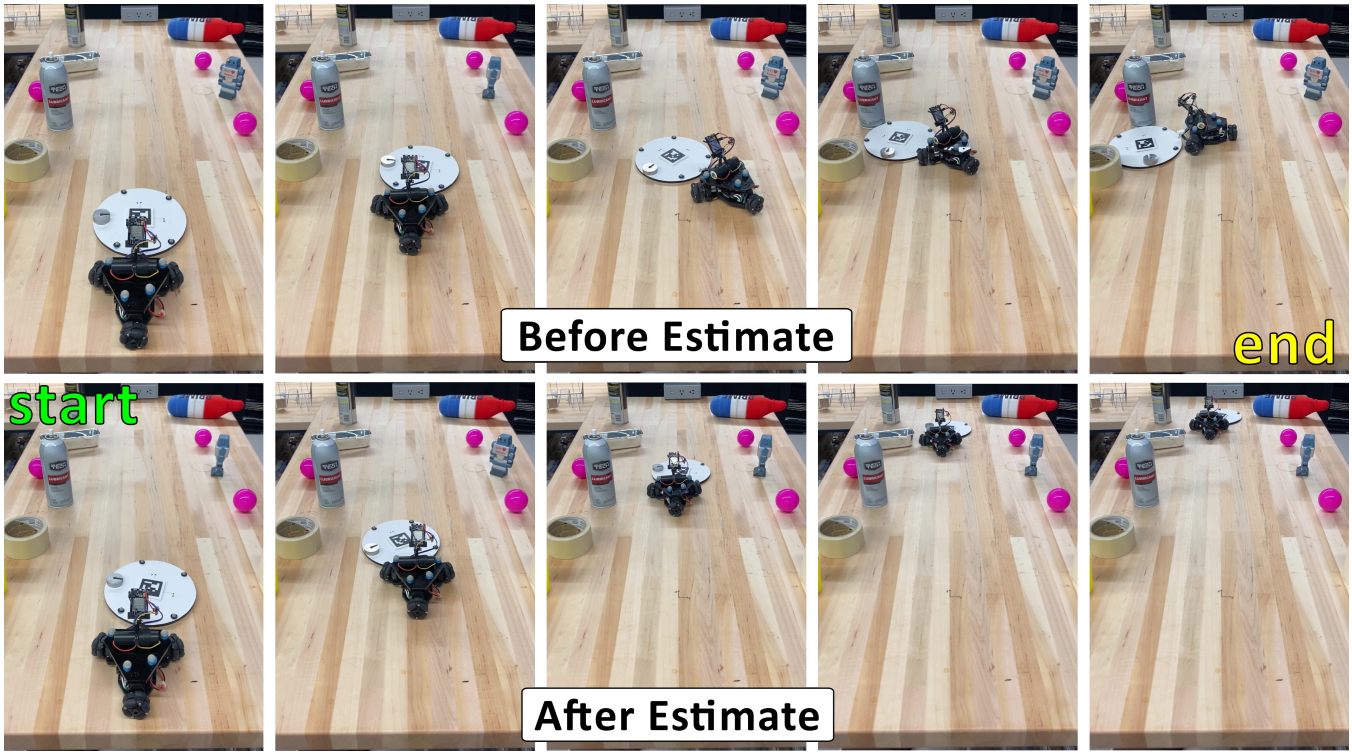

Fig. 6. Comparison of manipulation task before (top) and after (bottom) the estimation algorithm.

further from the attachment point.

For payload 2, which has a larger surface area and thus generates a greater moment, an initial misalignment or larger moment arm can more easily induce substantial rotational motion. This can, in turn, steer the binary search algorithm towards the local minimum (which is a circular motion) rather than the true LoA direction.

Overall, payload 2 exhibited considerably more variability, attributed to its larger surface area and generated moment. This is evident in Figure 4 where some trajectories for payload 2 deviate noticeably from the others.

To validate the effectiveness and accuracy of our approach, we compare against [24] which uses the same hardware, but is instead equipped with motion capture feedback. Table III presents a comparison of the mean angular error and its standard deviation for both methods. The mean error is defined by the angle between the final estimate vector and the ground truth vector, both originating from the robot's attachment point (see the dotted green line in Figure 1D).

Overall, the mean error across 10 trials using our local sensing approach is comparable to the motion capture method. However, the local sensing method exhibits a higher standard deviation across both payloads, largely due to occasional outliers from significant initialization bias and the inherent uncertainty of operating without global localization. Thus, by untethering the system from any external sensing infrastructure, our system gains flexibility at the cost of reduced precision. Strategies to mitigate this trade-off are discussed later in the paper.

Figure 5 shows the LoA estimates illustrated on each payload for both the baseline motion capture study and for this work. We notice CoM position 0 across all shapes as the most accurate. The local sensing performs similarly on average to the motion capture, without the overhead of fixed infrastructure.

The algorithm's ability to implicitly generalize across different payloads and CoM configurations is evident from these results. As the distance between the CoM and the push point increases, the footprint and rotational effects scale. This indicates that the algorithm effectively adapts to the varying payload geometries and associated dynamics.

Importantly, the algorithm shows robustness against most manual initialization bias, however egregious initial errors causes significant trajectory variation and final estimates. The consistence of performance across multiple trials, without the need for parameter tuning, underscores the flexibility of our approach. This robustness, coupled with the general flexibility validates this system's potential for broader application in payload manipulation, especially where global sensing is limited or unavailable.

C. Manipulation Task

To assess our estimate in a realistic setting, we designed a simple manipulation task where the robot transports the payload to the far end of a table within a tight corridor of obstacles. Both payloads and all CoM positions were tested, although only CoM position 2 on payload 1 is shown in Figure 6. In the top image, the robot attempts the task without the estimate, and blindly pushes until it soon collides with the left wall. However, once the estimate is provided to the robot

in the bottom image, it is able to push the payload across the entire table surface. This demonstration highlights the real-world viability of our method by showing how accurate LoA information enables stable, collision-free pushing in constrained environments.

IV. Conclusions

We introduced a novel, shape-agnostic method for locating the line of action (LoA) through a payload's center of mass using only onboard force and vision sensing on a holonomic mobile robot. Experimental results with payloads of varying shape, mass, and CoM positions confirm the iterative procedure converges to the true LoA within a few degrees.

To further improve consistency, we propose a brief calibration routine before each trial to refine initial alignment. Onboard robot localization would be a major improvement, as it would reduce uncertainty and also help overcome the local minima as seen by the residual circular motion.

Future experiments will focus on minimizing temporal discrepancies to reduce error margins. Furthermore, implementing machine learning for predictive analysis could be explored as a method to correct for nonlinearity and hysteresis in real-time. This step is vital to enhance the precision and reliability of force and visual readings in varying load conditions, fostering a more robust and adaptable algorithm and overall system.

References

[1] Matthew T. Mason. Mechanics and Planning of Manipulator Pushing Operations. *The International Journal of Robotics Research*, 5(3):53–71, September 1986.

[2] S. Akella and M.T. Mason. Posing polygonal objects in the plane by pushing. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, volume 3, pages 2255–2262, Nice, France, 1992. IEEE Comput. Soc. Press.

[3] Kevin M. Lynch and Matthew T. Mason. Stable Pushing: Mechanics, Controllability, and Planning. *The International Journal of Robotics Research*, 15(6):533–556, December 1996.

[4] K.M. Lynch, H. Maekawa, and K. Tanie. Manipulation And Active Sensing By Pushing Using Tactile Feedback. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 416–421, Raleigh, NC, 1992. IEEE.

[5] Kunpeng Yao, Mohsen Kaboli, and Gordon Cheng. Tactile-based object center of mass exploration and discrimination. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 876–881, Birmingham, November 2017. IEEE.

[6] Chao Wang, Xizhe Zang, Xuehe Zhang, Yubin Liu, and Jie Zhao. Parameter estimation and object gripping based on fingertip force/torque sensors. *Measurement*, 179:109479, July 2021.

[7] Alina Kloss, Maria Bauza, Jiajun Wu, Joshua B. Tenenbaum, Alberto Rodriguez, and Jeannette Bohg. Accurate Vision-based Manipulation through Contact Reasoning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6738–6744, Paris, France, May 2020. IEEE.

[8] Anirvan Dutta, Etienne Burdet, and Mohsen Kaboli. Push to Know! - Visuo-Tactile Based Active Object Parameter Inference with Dual Differentiable Filtering. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3137–3144, Detroit, MI, USA, October 2023. IEEE.

[9] Yong Yu, T. Arima, and S. Tsujio. Inertia parameter estimation of planar object in pushing operation. In *2005 IEEE International Conference on Information Acquisition*, pages 356–361, Hong Kong and Macau, China, 2005. IEEE.

[10] Changkyu Song and Abdeslam Boularias. A Probabilistic Model for Planar Sliding of Objects with Unknown Material Properties: Identification and Robust Planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5311–5318, Las Vegas, NV, USA, October 2020. IEEE.

[11] Jiacheng Yuan, Changhyun Choi, Ellad B. Tadmor, and Volkan Isler. Active Mass Distribution Estimation from Tactile Feedback. arXiv, March 2023. arXiv:2303.01010 [cs].

[12] Gao Ziyan, Armagan Elibol, and Nak Young Chong. Planar Pushing of Unknown Objects Using a Large-Scale Simulation Dataset and Few-Shot Learning. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 341–347, Lyon, France, August 2021. IEEE.

[13] Ziyan Gao, Armagan Elibol, and Nak Young Chong. Estimating the Center of Mass of an Unknown Object Via Dynamic Pushing. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, page 4, August 2022.

[14] Ziyan Gao, Armagan Elibol, and Nak Young Chong. On the Generality and Application of Mason's Voting Theorem to Center of Mass Estimation for Pure Translational Motion. *IEEE Transactions on Robotics*, 40:2656–2671, 2024.

[15] Adam Heins and Angela P. Schoellig. Force Push: Robust Single-Point Pushing With Force Feedback. *IEEE Robotics and Automation Letters*, 9(8):6856–6863, August 2024.

[16] Kuan-Ting Yu, Maria Bauza, Nima Fazeli, and Alberto Rodriguez. More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing, August 2016. arXiv:1604.04038 [cs].

[17] Zhongqiu Zhao, Xueyong Li, Changhou Lu, and Yonghui Wang. Center of Mass and Friction Coefficient Exploration of Unknown Object for a Robotic Grasping Manipulation. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 2352–2357, Changchun, August 2018. IEEE.

[18] Juekun Li, Wee Sun Lee, and David Hsu. Push-Net: Deep Planar Pushing for Objects with Unknown Physical Properties. In *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, June 2018.

[19] Nikos Mavrakis, Amir M. Ghalamzan E., and Rustam Stolkin. Estimating An Object's Inertial Parameters By Robotic Pushing: A Data-Driven Approach. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9537–9544, Las Vegas, NV, USA, October 2020. IEEE.

[20] Ziyan Gao, Armagan Elibol, and Nak Young Chong. Zero Moment Two Edge Pushing of Novel Objects With Center of Mass Estimation. *IEEE Transactions on Automation Science and Engineering*, 20(3):1487–1499, July 2023.

[21] Marek Kopicki, Jeremy Wyatt, and Rustam Stolkin. Prediction learning in robotic pushing manipulation. *Internation Conference on Advanced Robotics*, pages 1–6, July 2009.

[22] Jochen Stüber, Claudio Zito, and Rustam Stolkin. Let's Push Things Forward: A Survey on Robot Pushing. *Frontiers in Robotics and AI*, 7:8, February 2020.

[23] Sean McGovern and Jing Xiao. Learning and Predicting Center of Mass through Manipulation and Torque Sensing. In *2022 8th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pages 60–66, Munich, Germany, February 2022. IEEE.

[24] Steven M. Hyland, Jing Xiao, and Cagdas D. Onal. Predicting Center of Mass by Iterative Pushing for Object Transportation and Manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1615–1620, Detroit, MI, USA, October 2023. IEEE.

[25] Shadi Tasdighi Kalat, Siamak G. Faal, and Cagdas D. Onal. A Decentralized, Communication-Free Force Distribution Method With Application to Collective Object Manipulation. *Journal of Dynamic Systems, Measurement, and Control*, 140(9):091012, September 2018.

[26] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, Shanghai, China, May 2011. IEEE.