2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Performing Efficient and Safe Deformable Package Transport Operations Using Suction Cups

Rishabh Shukla[1*], Zeren Yu[1*], Samrudh Moode[1], Omey M. Manyar[1], Fan Wang[2], Siddharth Mayya[2], and Satyandra K. Gupta[1]

***Abstract*— Suction cups are popular for picking and transporting packages in warehouse applications. To maximize throughput, high transport speeds are desired. Many packages are deformable and may detach from the suction cups due to inertial loading if trajectories use excessive velocities. This paper introduces a novel methodology that analyzes package deformation through its curvature at the package-suction cup contact interface to generate a Factor-of-Safety (FOS) score for each waypoint in a given trajectory. By maintaining the FOS above a predetermined threshold, the trajectory planner is able to generate transport trajectories that are both safe and time-optimized. Experimental results show the method's efficacy, demonstrating a 21.92% reduction in transport times compared to a conservative trajectory generation. Our FOS predictor identified trajectories that ensured safe package transport with 100% accuracy across all 627 real-world experiments.**

I. Introduction

In warehouse environments, pick-and-place of various packages is an essential activity. Automating these tasks is necessary to keep these operations cost-effective. A popular option to perform these tasks is using manipulators equipped with suction cups [1]. These tasks encompass three fundamental steps: item pickup, transport, and placement. The imperative in these operations is optimizing transport speed to reduce the overall cycle time and operational costs [2]. However, transporting these goods at high speeds can lead to dropped items due to the physical limitations of the suction system. Therefore, any realistic warehouse pick-and-place operation must balance speed and safety. Finding this balance is further complicated because items are often packaged in deformable packages with diverse masses and sizes.

While investigating prior studies on optimizing transport operations, it's clear that much focus has been on handling rigid objects with suction cups [2], [3]. These studies highlight that high accelerations or decelerations can cause objects to detach from the suction cup due to inertial loading, limiting the operational speed at a given permissible pressure. This issue has been explored through both analytical and finite element models (FEM) [4], providing reasonably good insights, which will be discussed in detail in section II. However, in our exploratory studies, we observed that deformable packages detach at much lower velocities and accelerations compared to their rigid counterparts, indicating a need to predict detachment possibility during transport.

*Equal Contribution. [1]Realization of Robotic Systems Lab, University of Southern California, Los Angeles, CA, USA. [2]Amazon Robotics, North Reading, MA, USA. Emails: {fanwanf, sidmayya}@amazon.com
Address all correspondence to guptask@usc.edu

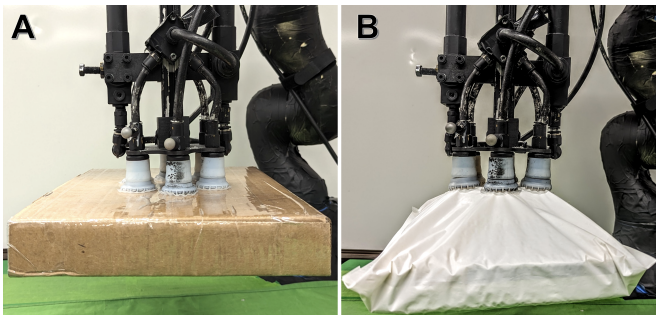

Fig. 1: Handling rigid box (A) versus deformable package (B) of fixed mass during a transport operation along predefined robot trajectory using suction cups. Deformation in (B) makes it more susceptible to detachment.

This is primarily because the interaction between suction cups and deformable packages is markedly affected by mutual deformation. This introduces new risks as the contact area and the shape of the deformable package change. Fig 1 illustrates the differences in package and suction cup interactions in rigid vs deformable packages.

Performing real-time FEM on deformable packages to determine the failure conditions is impractical due to limited information available upon package arrival. Therefore, we aim to develop a computationally fast method that can work with limited information. We designed an experimental testbed to track package characteristics, such as its curvature at the suction cup attachment interface, mass, and dimensions (see Section III-B). This setup enables data collection on package deformations and their effects on the likelihood of detachment. Using this data, we designed a physics-informed model that predicts package behavior during manipulation based on its characteristics after pickup and consequently predicts a Factor of Safety (FOS), a metric for assessing the safety margins of transport trajectories against grip failure (see Section IV). Based on the FOS, we generate time-optimized and safe trajectories that prevent package detachment from suction cups (see Section V).

Thus, the core contributions of our paper are as follows:

1) A new testbed for data collection to track the curvature of deformable packages across different trajectories.
2) A multi-level, physics-informed model to (A) predict the behavior of packages during manipulation and (B) quantitatively predict the Factor of Safety (FOS) of transport trajectories against potential detachment.
3) A trajectory planner for generating safe and time-optimized trajectories based on the FOS.

II. Related Work

Recent advancements have begun to explore the complexities of manipulating deformable objects. For instance, [5] introduces SoftGym, a simulation benchmark for deformable object manipulation that emphasizes the need for data-driven methods. Furthermore, studies like [6] and [7] delve into learning particle dynamics and intuitive physics for handling various objects. Similarly, [8] and [9] focus on sim-to-real reinforcement learning and other learning-based methods for simulating and interacting with deformable objects. These studies, while not requiring explicit modeling, underscore the importance of realistic simulation environments and direct learning from real-world interactions in tackling the complex dynamics involved in manipulating deformable packages.

Work in [10] demonstrates the use of Goal-Conditioned Transporter Networks for manipulating deformable objects like cables, fabrics, and bags, extending the capabilities of robotic manipulation to more complex and real-world applicable tasks [11]–[13]. The manipulation of deformable packages, characterized by their variable curvatures and susceptibility to deformation, introduces challenges not adequately addressed by current models, highlighting a considerable gap in research aimed at ensuring the successful manipulation of deformable packages in transport operations [14].

In the context of trajectory planning, traditional algorithms such as optimization-based planners like TrajOpt [15] and CHOMP [16], and TOPP-RA [17] offer solutions for time-optimal path planning. However, they fall short of addressing the constraints associated with the suction cup-object interface crucial for maintaining grasp stability. This is particularly critical during high-speed maneuvers or abrupt directional changes. Addressing the dynamic complexities of object manipulation, recent advancements such as the Grasp-Optimized Motion Planning (GOMP) framework [18] and its subsequent iterations (GOMP-ST [3] and GOMP-FIT [19]), alongside Pham et al.'s work [2], have introduced constraints specifically designed for handling objects with suction cups. However, these studies have primarily focused on rigid objects. Therefore, the unique challenges posed by the manipulation of deformable packages—namely, the increased risks of peeling, slipping, and twisting—are still not sufficiently mitigated by existing frameworks [14], highlighting the need for a safe package transport framework.

III. Problem Formulation

A. Background and Terminology

Understanding the dynamic behavior of the package as it is manipulated is fundamental to achieving our goal. We conducted exploratory experiments to understand if this dynamic behavior of the package could be understood based on its inertial characteristics and its motion during manipulation. Our experiments utilized packages with a uniformly distributed rectangular mass within a deformable package, as shown in Fig. 1(B). In our study, we assume that the contents within the packages are tightly packed to minimize motion of the object inside the package. We make this assumption to simplify how inertial loading triggers grip failure.

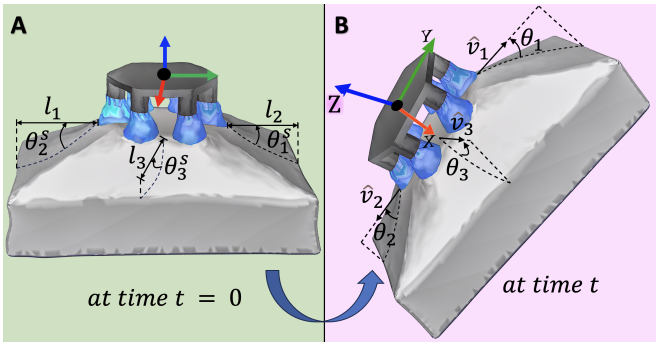

Fig. 2: Illustration of a deformable package during a suction-based transport operation, highlighting key terminologies for understanding package behavior. (A) demonstrates package curvatures when the robot is stationary (a.k.a. static curvatures). At time $t = 0$, when the TCP is parallel to the horizontal plane, these are also referred to as initial static curvatures. (B) shows the package at a trajectory waypoint at time t, showcasing dynamic curvatures during manipulation.

The Factor of Safety (FOS), denoted as $\mathcal{F}$, serves as a quantitative indicator of the manipulation's safety margin, influenced by curvatures (see Section IV) at the suction cup-package interface and the package's inertial loading as shown in Fig. 2. Preliminary experiments demonstrate that larger curvatures and higher loading inversely affect the likelihood of success. It is also observed that curvatures along planes perpendicular to each edge of the package, extending through the nearest suction cup (see fig. 2A), are particularly important because they usually represent the largest curvatures observed during manipulation. Similarly, the package's mass, which affects the inertial loading due to the Tool Center Point (TCP) acceleration, and the amplitude of package oscillation are critical factors to consider. For example, a heavier mass with large swings results in a lower FOS. Therefore, the $\mathcal{F}_j$ for $j = 1$ to 4, for each package-edge j, is a function of package mass (m), oscillation amplitude (A), interface curvature θ_j measured through the closest suction cup from the package edge, and TCP acceleration (a_{TCP}), expressed as $\mathcal{F}_j = f(\theta_j, a_{TCP}, m, A)$.

The package's curvature varies depending on its state of motion. When attached to a stationary robot, the curvatures observed are termed as static curvatures $\theta^s = \{\theta_j^s\}$ for $j = 1$ to 4 as shown in fig. 2A. The static state provides a baseline from which the effects of manipulation can be measured. Specifically, the static curvatures when the TCP is parallel to the horizontal plane are referred to as initial static curvatures $(\theta^s(t = 0))$, determined by the package's packing tightness, stiffness, and the pickup point location. The static curvature at any waypoint at time t can be determined based on the initial static curvature, TCP orientation (O), and package characteristics, such as mass and dimensions (d), expressed as $\theta_j^s(t) = f(m, d, \theta_j^s(t = 0), O)$.

The dynamic curvatures (θ), which are essentially the curvatures that change during the manipulation, emerge as the package is moved. This change is influenced by the package swing caused by the TCP motion. For any waypoint

at time t in the trajectory, the dynamic curvatures along the j^{th} edge $\theta_j(t)$ can be modeled based on the static curvatures at the waypoint $\theta_j^s(t)$, the TCP's acceleration ($a_{TCP}(t)$), and the package's characteristics. Package characteristics such as mass m and the distance between the edge of the package and the closest suction cup used for curvature measurement (l_j) are crucial in predicting the dynamic curvatures. For instance, a heavier mass with a larger suction cup-edge distance will have different amplitudes of oscillations and, consequently, different dynamic curvatures, expressed as $\theta_j(t) = f(\theta_j^s(t), a_{TCP}, m, l_j)$.

B. Robotic Cell Design

To investigate this problem, our experimental setup employs a KUKA IIWA 7 manipulator with a suction system featuring six suction cups operating at 25 psi, as depicted in Fig. 3. This setup operates through two distinct phases as follows:

Training Phase: We use a high-fidelity OptiTrack motion capture (MoCap) system to monitor the dynamic curvatures of packages during the training phase. By attaching motion capture markers directly to the packages, as illustrated in Fig. 3, we achieve accurate curvature tracking at 240 Hz. This data is used during the training phase. MoCap system is not used during the execution phase.

Execution Phase: We used a depth camera-based system during the execution phase to provide an estimation of initial static curvatures. Our model then predicts dynamic curvatures during manipulation informed by the initial static curvature estimations. To estimate the mass of the package, we utilize KUKA's joint torque sensors.

Through this two-phased approach, we balance the need for accurate and precise data collection in the model training stage with the practicalities of real-world application.

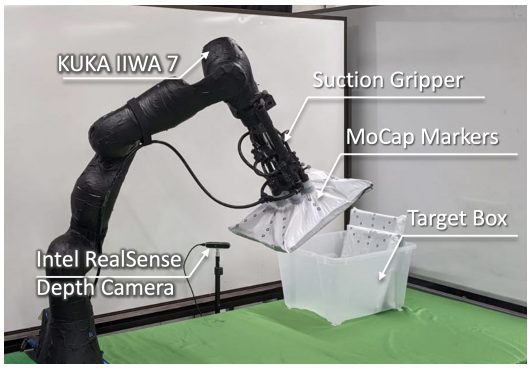

Fig. 3: Experimental cell showing manipulator with a suction gripper, used for package transport experiments during training and execution phases. MoCap cameras are not shown.

C. Problem Statement

FOS Predictor: Given a package characterized by known mass, dimensions, and initial static curvature after pickup, alongside the robot's proposed trajectory, our objective is to predict the FOS along each package edge at each waypoint in the trajectory, i.e., $\mathcal{F}_j(t)$. This predictor should be based on experimental data from various transport operations featuring diverse packages and trajectories. The dataset includes both successful and failed manipulations for model training and evaluation. Section IV further details the approach.

Trajectory Planner: The Trajectory Planner receives start and goal positions and package characteristics as inputs. It aims to plan a time-optimized trajectory from start to goal while ensuring that FOS along all package edges remains above a specific safety threshold. The planner constructs a trajectory as a sequence of timestamped waypoints from robot-specific constraints. This trajectory then acts as an initial trajectory for the optimization to meet package-specific constraints, thus upholding the FOS threshold at every waypoint. The approach of the Trajectory Planner is presented in Section V.

IV. Predicting FOS for Suction Failure

Fig. 4 illustrates the pipeline for predicting FOS for a proposed trajectory during a suction-based package transport operation. This pipeline comprises three components: (1) Model 1 forecasts static package curvatures at any given TCP orientation using initial static curvatures recorded after pickup, (2) Model 2 estimates package dynamics during transport and, based on these dynamics, predicts dynamic curvatures, and (3) Model 3 predicts the FOS at each waypoint of a trajectory.

A. Model 1: Static Curvature Estimation

The approach involves two steps: (1) using a depth camera to estimate the package's initial static curvatures $\theta_j^s(t=0)$; and (2) employing Model 1, represented as G, which takes $\theta_j^s(t=0)$ and TCP orientations across a given trajectory q_t as inputs to predict the static curvatures $\theta_j^s(t)$ at all times t along the trajectory. These steps are detailed as follows:

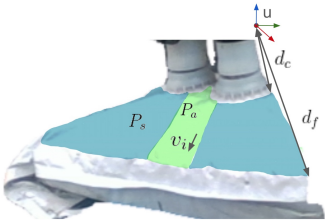

Fig. 5: Extract curvature along v_i from depth image

Estimate Initial Static Package Curvature: From a 3D point-cloud $P \in \mathbb{R}^3$ representing the package surface P_s, we extract the initial static curvatures $\theta_{1-4}^s(t=0)$. The vacuum gripper's grasp center is denoted as $u \in \mathbb{R}^3$. The distance from the farthest edge of the suction cup to this grasping center is d_c, and the distance from the package's edge to the grasping center is d_f, as shown in Fig. reffig:pcd. To specify the direction of curvature, we use a horizontal unit vector $v_j \in \mathbb{R}^3$ for each edge j of the package.

We isolate a subset of the package surface's point-cloud P_a along the direction of v_j. The initial static curvature $\theta_j^s(t=0)$ along v_j is then estimated by performing a linear regression on the horizontal component $w_i = \sqrt{{p_x}^2 + {p_y}^2}$

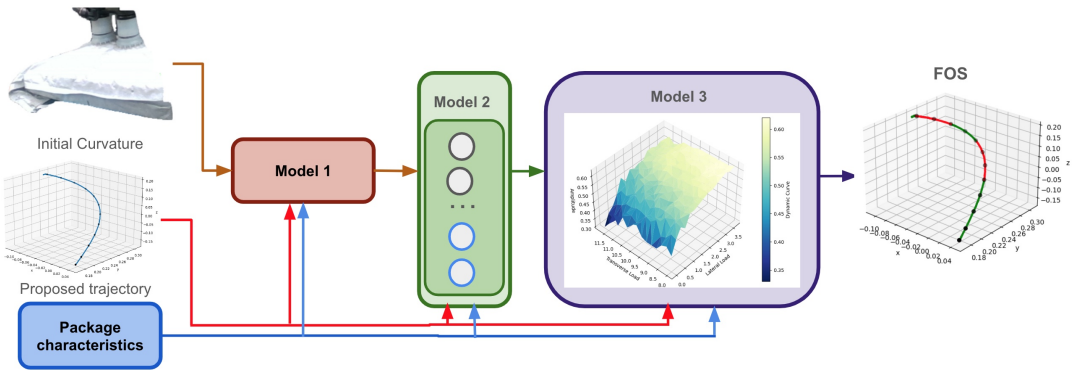

Fig. 4: The predictive pipeline for Factor of Safety (FOS) estimation consists of three interconnected models. Model 1 processes the initial depth image captured at the package-suction cup interface, extracting static curvature parameters. It then forecasts the static curvatures throughout the trajectory by incorporating the TCP orientation at each waypoint. Model 2 builds on the static curvature data from Model 1, integrating trajectory characteristics (e.g., speed, acceleration) and package-specific parameters (e.g., mass, dimensions) to predict dynamic curvatures caused by oscillatory motions during transport. Model 3 combines these predicted dynamic curvatures with the oscillation amplitude estimated by Model 2, along with lateral and axial load measurements, to compute the Factor of Safety (FOS) score at each waypoint.

and the vertical component $h_i = p_z$ for each point $p \in P_a$ (see Fig. reffig:pcd). The relationship is expressed as:

$$H = k \cdot W + B,\ \theta_j^s = \arctan(k) \tag{1}$$

where H is the vertical component, W is the horizontal component, k is the slope of the regression line, and B is the y-intercept. The static curvature θ_j^s is calculated as k, providing a measure of how much the package surface curves at that edge.

Estimate Static Curvature: To understand the variation of static curvatures with TCP orientation, we use an analytical model. This model considers the bending forces that act on the suction cup due to gripper rotation taking into account both the package deformation and the suction cup's elastic response. We model the deformable suction cup as an Euler-Bernoulli beam $M = -EI\frac{d^2w}{dx^2}$, where M is the bending moment, EI is the flexural rigidity of the suction cup, w is beam deflection, and dw/dx is the slope of the beam. A suction gripper has fixed EI as the material property and x represents a fixed distance to the point of application of load at the end of the suction cup. Hence, under small-angle approximation, where $\frac{\sin\alpha}{\alpha} \approx 1$, lateral load L_l is proportional to α, the angle between the gripper and vertical line. Thus, G can be further simplified to a proportional function between α and the curvature change $\Delta\theta_j$ of the package (eq.2). θ_j^s at each waypoint on the trajectory is calculated as (eq.3).

$$G : \Delta\theta_j = \kappa \cdot q \tag{2}$$

$$\theta_j^s(t) = \theta_j^s(t=0) + G(\alpha_i) \tag{3}$$

B. Model 2: Dynamic Curvature Estimation

Deformable packages are usually bottom-heavy and behave similarly to a driven pendulum during robot motion. Model 2 (D) takes trajectory as control input (U) and package characteristics (X) as package state to predict the amplitude (A) of the dynamic curvature θ_j change due to package oscillation.

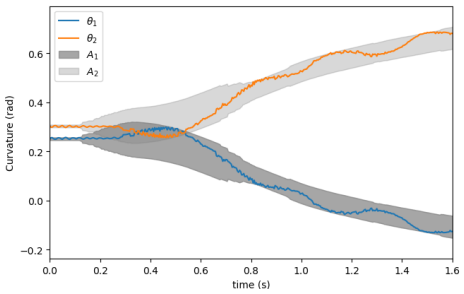

Fig. 6: Illustration of dynamic curvature

Accurately predicting dynamic curvatures at any given time t can be challenging due to the chaotic oscillations observed during package manipulation. Thus, Model 2 focuses on predicting a geometric envelope over package oscillation amplitude over time (see Fig. 6). This envelope represents the maximum deviation from the static curvature θ_j^s without being affected by the oscillation phase. In this context, $\hat{\theta}_j$ indicates the predicted dynamic curvature, where the $\hat{}$ symbol on top of θ_j specifies that it is a predicted value based on the model's calculations. The Hilbert transformation [20], a mathematical tool used for signal processing, assists in estimating A by analyzing the curvature data captured through the MoCap system. This approach allows us to integrate the estimated amplitude A with the static curvature θ_j^s to model the predicted dynamic curvature $\hat{\theta}_j$ as $\hat{\theta}_j = A + \theta_j^s$.

To accurately predict curvature in our system, we aim to capture its underlying dynamics amidst external forces and constraints. While a comprehensive dynamics model of the package could achieve this, it may necessitate extensive training data and be susceptible to noise in input data. Our experimental findings indicate that dynamic curvature correlates well with package oscillation amplitude and initial

curvature, obviating the need for a full-fledged dynamics model of the package. Thus, we opt for a simpler and more efficient approach that can effectively capture the influence of amplitude on curvature. We leverage the *Sparse Identification of Nonlinear Dynamics* (SINDy) method, ideally suited for nonlinear systems with noisy observations, to estimate the amplitude of package oscillation reliably.

SINDy is a data-driven sparsity-promoting method to discover nonlinear dynamics of a system X under control input U [21]. A sparse regression is performed on a library of candidate nonlinear term $\Theta(X, U)$ to identify significant active terms in the dynamic system. SINDy is employed to discover the non-linear amplitude dynamics A, resulting from robot motion. We model the control term U from a segment of gripper motion of length n, the state term X from the initially obtained package characteristics. U is a set of n trajectory waypoints represented by $(x, y, z, \alpha, \beta, \gamma)$ in $SE(3)$, its 1^{st} and 2^{nd} order derivatives, and the time gap between adjacent waypoints. X includes package characteristics, like mass m, distance to the edge l_j, and $\theta_j(t = 0)$(eq.4).

$$U_{[1\times 19\cdot n]} = [x\ y\ z\ ...\ \ddot{\alpha}\ \ddot{\beta}\ \ddot{\gamma}\ T],\ X_{[1\times 4]} = [m\ \theta_j(t=0)\ l_j] \tag{4}$$

To accommodate the system's dynamic behavior, which resembles that of a driven pendulum, sinusoidal terms and quadratic terms are included in the library to address rotation terms and combinations between X and U. The library $\Theta(X, U)$ is constructed as (eq.5), $X \bigotimes U$ defines a vector of all product combinations between X and U.

$$\begin{aligned}\Theta(X,U) =&[1^T\ X^T\ U^T\ sin(X)^T\ sin(U)^T \\ &(X \bigotimes X)^T\ ...\ (X \bigotimes sin(X))^T \\ &...\ (sin(X) \bigotimes sin(U))^T]\end{aligned} \tag{5}$$

A sparse regression is employed to identify significant terms in the prediction of envelope $\hat{A}$(eq.6). Ξ is the sparse coefficient for the dynamic system, ξ_k is k^{th} row of Ξ, λ is the regulation parameter.

$$\begin{aligned}&\hat{A} = \Xi\Theta^T(X,U) \\ &\xi_k = argmin_{\xi_k}\frac{1}{2}||\hat{A} - \hat{\xi}\theta^T(X,U)||_2^2 + \lambda||\hat{\xi}_k||\end{aligned} \tag{6}$$

C. Model 3: Factor of Safety Prediction

Model 3, represented by function F, takes lateral load L_l, axial load L_a, A, and θ_j as input to predict the FOS at a given waypoint. FOS is calculated as $\mathcal{F} = \frac{S_y}{S_w}$, where S_y is the system's yield stress and S_w is the working stress [22]. Yield stress is the point where failure occurs, and working stress is the stress the system withstands. From failure trials, in which packages fall off the gripper, critical points C_f chosen within time period T before the actual drop of the package is used as s_w of the system(eq. 7).

$$C_f = max(L_l, L_a, A_t, \theta_j(t))\ \ |t - t_f| < T \tag{7}$$

Create Synthetic Data and Proportional Function: Synthetic data is generated from critical points on the verge of failure, after which we generalize the FOS prediction to non-critical regions using a proportional function. The corresponding FOS is derived from a linear proportional function $P(C) = E \cdot C_f$ with coefficient E. Pseudo $\mathcal{F}' = P(C_f)/P(C')$. The coefficient E, essentially describes the significance of each input term in C, is searched with Bayesian optimizer [23], which minimizes the loss of an expansive black-box function J. Function J is designed as a GPR trainer trains on critical points combined with synthetic data and evaluated on a test set with the loss function(eq. 8). We use a sigmoid function σ to normalize the prediction of FOS and a penalty term for false prediction.

$$\begin{aligned}Loss = &\sum_{i\in TN}^{i}\sigma(\mathcal{F}_i - 1) - \sum_{i\in TP}^{i}\sigma(\mathcal{F}_i - 1) \\ &+ \lambda(|FP| + |FN|)\end{aligned} \tag{8}$$

Compute Lateral Load and Axial Load: Pseudo load (φ) is calculated from the tangential acceleration of gripper TCP and weight of the package. Transverse load along (v_j) and axial load along the z-axis (i.e., the axial direction) of vacuum gripper (v_a) are derived from φ(eq. 9).

$$\begin{aligned}&L_t = \varphi \cdot v_j \\ &L_a = \varphi \cdot v_a\ \ \varphi \in \mathbb{R}^3\end{aligned} \tag{9}$$

Construct a Gaussian Process Regression Based Model Bayesian regressions based on Gaussian process are described in [24]. For input/output pair x_i, y_i from distribution S where $x_i \in \mathbb{R}^D$, $y_i \in \mathbb{R}$, y_i can be considered as a result drawn from a zero-mean multi-variable Gaussian distribution in GP. The Covariant matrix is a function that draws y_i from the distribution on the training set (eq. 10). In model3, GPR computes FOS from x: $L_t, L_a, A_t, \theta_j(t)$ using the rational quadratic kernel discussed in [25] to extend FOS to non-critical points (eq. 11).

$$(x_1\ x_2\ ...|y_1\ y_2\ ...) \sim \mathcal{N}(0,\ \Lambda(S,S)) \tag{10}$$

$$k(x_i, x_j) = (1 + \frac{d(x_i, x_j)^2}{2\alpha l^2})^{-\alpha} \tag{11}$$

V. Trajectory Optimization

The method for FOS-constraint-based trajectory optimization is shown in Fig. 7. The optimization process has three steps:

1) **Initial Trajectory Generation:** We start by generating an initial trajectory that adheres to the robot's inherent constraints, such as joint position, velocity, and acceleration limits. This trajectory serves as a preliminary time-optimal trajectory from start to goal pose.
2) **Safety Assessment:** Each waypoint of the trajectory is assessed using the predicted Factor of Safety for every waypoint, indicating areas where the grip stability may be compromised.
3) **FOS-Based Trajectory Refinement:** In case any waypoint violates FOS constraint, this trajectory serves as initialization for a new trajectory optimization using FOS constraint.

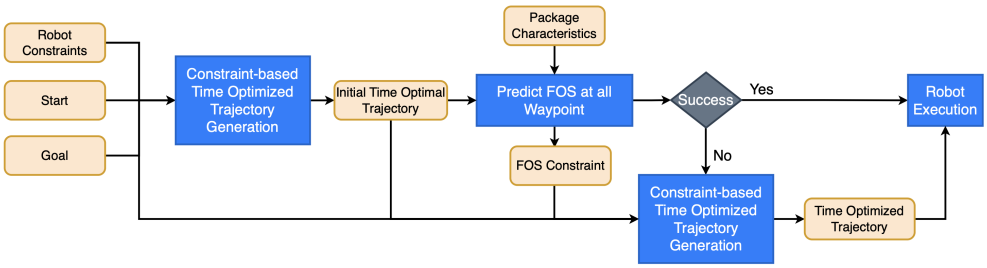

Fig. 7: Constraint-Based Trajectory Optimization

This three-step process is followed to reduce computation costs when an optimized trajectory search isn't necessary, such as for a lightweight package or a robot moving at a conservative speed.

We use a direct MPC-based approach as discussed in [3] for trajectory optimization. However, to integrate the nonlinear FOS constraint into the optimization process, we employ a gradient-based method for linearization, providing necessary first-order approximations. Subsequently, we use a Quadratic Programming (QP) formulation to solve for the trajectories. We utilize CVXPY [26], [27] as the QP solver.

VI. Results

A. Experimental Setup

For our experiments, we created a diverse set of deformable packages for model training and evaluation. Table I shows the packages' characteristics and assigns them an ID for referencing in later text.

	Mass	Suction-cup package edge distance →		
		6.5 cm	9 cm	10.25 cm
Training Phase	1150 g	P1	P2	P3
	1277 g	P4	P5	P6
	1500 g	P7	P8	P9
	1800 g	P10	-	-
Evaluation Phase	1200 g	P11	P12	P13
	1400 g	P14	P15	P16
	1600 g	P17	P18	P19

TABLE I: Characteristics of Packages Used in the Training and Evaluation Phases of the Model: The table lists package identifiers corresponding to each mass and suction cup-edge distance. Fields left empty indicate that no experiment was conducted for that specific parameter.

B. Individual Model Performance

We evaluated each model separately for predicting static and dynamic curvatures and FOS.

Model 1 Evaluation: To test the impact of the change in gripper orientation on the static curvatures of a package, we rotated the gripper slowly to eliminate package oscillations for collecting ground truth data. Specifically, we performed rotations at increments of 15, 30, and 45 degrees to assess how changes in gripper orientation affect the package's static curvature measurements. Data for this analysis was collected across 105 trials involving nine different packages, yielding a total of 429,951 data points. We allocated an 82/23 split of this data for training and testing purposes, respectively. Analysis of the collected data resulted in a proportional coefficient k of 0.7465, demonstrating the model's effective representation of the relationship between gripper orientation and static curvature. This was further substantiated by an RMSE of 0.01455 on the radius in the test set, underscoring the model's predictive reliability (See Fig.8).

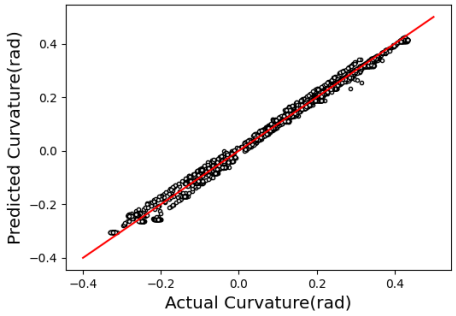

Fig. 8: Model 1 is able to predict static curvature with uniform accuracy across various curvature

Model 2 Evaluation: To assess Model 2's ability to predict dynamic curvatures influenced by gripper motion, we conducted 127 experiments. These experiments utilized 58 unique pick-and-place trajectories, varying across four orientation angles at the goal position and seven operational speeds in both linear and circular motions, applied to nine different packages.

From each experiment, we compiled all the frames leading up to the moment of package detachment from the suction cup, resulting in a total of 299,502 frames for analysis. For training the SINDy component of Model 2, we set the model parameter $n = 5$, thus including five sequential waypoints with a 0.02-second interval between each for the control term U. To ensure sequence continuity, the starting condition T_0 was duplicated for the first waypoints in each sequence.

Model 2 was optimized using the AdamW optimizer, with an initial learning rate of 1e-4. We applied a structured learning rate decay strategy, reducing it by 10% every 400 epochs, and a batch size of 50 was chosen. The model's performance was evaluated based on Mean Squared Error (MSE) and Mean Absolute Error (MAE) across training, testing, and validation datasets, as detailed in Table II. SINDy effectively identifies the package's oscillation patterns based on the gripper's motion and the package's characteristics.

Model 3 Evaluation: In examining package transportation trajectories, we evaluated 627 experiments to pinpoint critical

	Train	Test	Validation
MSE	0.0012	0.0020	0.0015
MAE	0.0248	0.0331	0.0274

TABLE II: Model 2 evaluation results on all three datasets perform consistently attributable to SINDy's utilization of sparse regression, which extracts underlying physical connections between control variables and observed behaviors.

failure points. Specifically, we identified 89 trajectories near the threshold between success and failure — termed 'critical trajectories' — where minor adjustments in speed could alter the outcome. From these, we carefully selected 41 critical points directly before a package drop for model training. In contrast, data from 48 trajectories that narrowly avoided failure were aggregated to refine our model through the optimization of proportional coefficients for synthetic data, achieving calibration values of $[0.28, 1, 0.73, 1.04]$ for parameters $[\theta, A, L_l, L_a]$. This enabled Model 3 to distinguish between failed and successful executions by predicting a low FOS for failures — with a mean FOS of 1.032 compared to 1.087 for successes. Implementing a conservative prediction approach, with a safety threshold set at 1.06, our model accurately identified all instances leading to failure, achieving an F1 score of 0.81 as detailed in Table III. A lower threshold of 1.05 is tested to have 3 false positive cases, and a higher threshold of 1.07 is evaluated for a lower F1 score of 0.73.

	Predicted Success	Predicted Fail
Actual Success	53.9%	24.7%
Actual Fail	0%	21.3%

TABLE III: FOS threshold set to 1.06 achieving highest F1 of 0.81 and 0 false positive to ensure the safety.

C. Evaluation of Factor of Safety Prediction

To evaluate the FOS predictions by our concatenated model, we conducted 270 new transport experiments with three packages of varying mass and dimension to ensure diverse testing conditions. For each package, we executed three distinct trajectories and categorized them based on their predicted FOS: 'safe' (FOS > 1.06), 'risky' (1.06 > FOS > 1), and 'failing' (FOS < 1). To neutralize the impact of the start and goal location on the results, each set of trajectories was executed along three different paths (τ). Furthermore, to assess safety, each trajectory execution was repeated 10 times to evaluate the probability of failure under each FOS category. The outcomes of these experiments are compiled in Table IV, demonstrating the model's predictive accuracy.

D. Evaluation of Trajectory Optimization for Time Efficiency

In industrial robotic systems, conservative transport speeds are established by uniformly restricting robot joint velocity and acceleration to the safest levels identified through extensive testing across various object and suction cup combinations. While this approach ensures no failure across tested conditions, it results in sub-optimal performance, as discussed in [2]. For our evaluation, we utilized path-specific conservative trajectories. A path-specific conservative trajectory is determined experimentally such that it has

Package ID/Path ID	FOS < 1	1<FOS< 1.06	FOS > 1.06)
P7 / τ_3	0/10	4/10	10/10
P7 / τ_4	1/10	5/10	10/10
P7 / τ_6	0/10	1/10	10/10
P8 / τ_3	0/10	3/10	10/10
P8 / τ_4	2/10	9/10	10/10
P8 / τ_6	0/10	3/10	10/10
P19 / τ_3	1/10	6/10	10/10
P19 / τ_4	1/10	6/10	10/10
P19 / τ_6	1/10	5/10	10/10

TABLE IV: All trajectories deemed safe by the model ('FOS > 1.06') executed successfully, while 93.4% of trajectories identified as failures ('FOS < 1') resulted in package detachment. The model reliably predicts both safe and unsafe outcomes. Risky trajectories ('1 < FOS < 1.06') exhibited unpredictable results, indicating a chaotic region where success is inconsistent, even with identical trajectories and packages. Thus, our framework aims to steer planned trajectories into the safe region with a high likelihood of success.

uniform joint velocities and accelerations and results in the successful transport of all packages (i.e. P11-19) across a given path. Optimized trajectories are those where the FOS prediction for each waypoint exceeds the safety threshold of 1.06. We conducted experiments with packages P11-19 across paths τ_{1-6} and compared time optimization for fast, conservative, and optimized trajectories. Additionally, three optimized trajectories were tested for each path to ensure that successful operations were not by chance.

Path ID	Fast(s)	Conservative (s)	Optimized (s)
τ_1	1.28	1.64	1.36
τ_2	0.89	0.93	0.91
τ_3	0.95	1.23	1.11
τ_4	0.85	2.85	1.73
τ_5	1.05	12.28	6.42
τ_6	0.91	1.16	0.98

TABLE V: Transport time comparison for fast, optimized, and conservative trajectories. Gains in paths with orientation changes (τ_{3-5}) show our model's effective optimization.

The optimized trajectory resulted in an average time improvement of 21.92% with a standard deviation of 6.75% against path-specific conservative trajectories. This improvement was achieved by maintaining the safety threshold which ensured that no packages were dropped during transport.

E. Ablation Study on Curvature

We conducted an ablation study on rigid objects and deformable packages with various curvatures to investigate the influence of curvature on package transport. For each of the three chosen packages, 20 experiments were conducted at critical speed, and 5 additional experiments were conducted on the same mass and rigid object with the same dimensions for comparison. Packages were adjusted for packing tightness between each run to create variance in curvature. Initial static package curvature and the paired operation results were captured for evaluation.

All experiments on rigid objects succeeded, whereas 48% of experiments with deformable packages of the same mass

failed. Table VI presents detailed results of curvature differences for all the deformable packages. These results indicate that deformable packages and rigid objects require different handling approaches, underscoring the significant impact of package curvature on the outcome of each experiment.

PackageID/TrajID	Success (rad)	Fail (rad)	P-value
P8 / τ4	0.29±0.049	0.40±0.007	6.45e-6
P7 / τ3	0.30±0.027	0.41±0.07	3.81e-4
P19 / τ6	0.34±0.018	0.42±0.013	1.63e-8

TABLE VI: Minimum curvature across v_{1-4} is selected. A statistically significant P-value showed a large difference in initial curvature between successful and failing execution across all three experiment groups.

VII. Conclusions

In this study, we developed an approach to optimize robotic transport operations in automated warehouses, focusing on the dynamic interaction between suction-equipped manipulators and deformable packages. By integrating a predictive Factor-of-Safety (FOS) model with a trajectory planner, we analyzed the package's deformation through its curvature at the suction cup-package interface. This enabled the generation of trajectories that ensure both safety and time efficiency by maintaining the FOS above a critical threshold. Experimentally, our approach improved operational time by 21.92% for handling deformable packages compared to conservatively generated transport trajectories. Future work will broaden our approach's applicability by extending the framework to handle packages with sliding contents and non-uniform mass distribution. This study assumed a fixed suction system configuration (size, suction pressure, number of suction cups) to isolate the effects of package curvature and trajectory on the FOS. Future work will explore integrating varying suction cup properties into the model for greater generalizability. We also plan to enhance trajectory planning by incorporating learning-based methods using historical data for faster, more time-optimized trajectory generation.

Acknowledgment

This work is supported in part by Amazon Robotics. The opinions expressed are those of the authors and do not necessarily reflect the opinions of the sponsors.

References

[1] S. Li, A. Keipour, K. Jamieson, N. Hudson, S. Szhao, C. Swan, and K. Bekris, "Pick planning strategies for large-scale package manipulation," in *IROS 2023 Workshop on Learning Meets Model-based Methods for Manipulation and Grasping*, 2023.

[2] H. Pham and Q.-C. Pham, "Critically fast pick-and-place with suction cups," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 3045–3051, iSSN: 2577-087X.

[3] Y. Avigal, J. Ichnowski, M. Y. Cao, and K. Goldberg, "GOMP-ST: Grasp Optimized Motion Planning for Suction Transport," in *Algorithmic Foundations of Robotics XV*, ser. Springer Proceedings in Advanced Robotics. Cham: Springer International Publishing, 2023.

[4] J. W. Eischen, S. Deng, and T. G. Clapp, "Finite-element modeling and control of flexible fabric parts," *IEEE Computer Graphics and Applications*, vol. 16, no. 5, pp. 71–80, 1996.

[5] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 432–448.

[6] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," in *ICLR*, 2019.

[7] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," *Advances in neural information processing systems*, vol. 29, 2016.

[8] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 734–743.

[9] R. Lee, D. Ward, V. Dasagi, A. Cosgun, J. Leitner, and P. Corke, "Learning arbitrary-goal fabric folding with one hour of real robot experience," in *Conference on Robot Learning*. PMLR, 2021.

[10] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[11] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2146–2153.

[12] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658.

[13] J. E. Hopcroft, J. K. Kearney, and D. B. Krafft, "A case study of flexible object manipulation," *The International Journal of Robotics Research*, vol. 10, no. 1, pp. 41–50, 1991.

[14] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, 2022.

[15] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, Aug. 2014.

[16] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 489–494.

[17] H. Pham and Q.-C. Pham, "A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 645–659, Jun. 2018, conference Name: IEEE Transactions on Robotics.

[18] J. Ichnowski, M. Danielczuk, J. Xu, V. Satish, and K. Goldberg, "Gomp: Grasp-optimized motion planning for bin picking," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5270–5277.

[19] J. Ichnowski, Y. Avigal, Y. Liu, and K. Goldberg, "GOMP-FIT: Grasp-Optimized Motion Planning for Fast Inertial Transport," in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 5255–5261.

[20] R. W. Schafer and A. V. Oppenheim, *Discrete-time signal processing*. Prentice Hall Englewood Cliffs, NJ, 1989, vol. 5.

[21] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit," *Proceedings of the Royal Society A*, vol. 474, no. 2219, 2018.

[22] F. Beer, E. Johnston, and J. DeWolf, "Mechanics of materials, 5th si edition," *Stress*, vol. 1, no. 10, pp. 1–12, 1999.

[23] P. I. Frazier, "Bayesian optimization," in *Recent advances in optimization and modeling of contemporary problems*. Informs, 2018, pp. 255–278.

[24] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.

[25] C. Cortes, P. Haffner, and M. Mohri, "Rational kernels: Theory and algorithms," *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 1035–1062, 2004.

[26] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[27] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.