

DBPF: A Framework for Efficient and Robust Dynamic Bin-Picking

Yichuan Li¹, *Student member, IEEE*, Junkai Zhao², Yixiao Li²,
Zheng Wu³, Rui Cao¹, Masayoshi Tomizuka³, *Fellow, IEEE*, Yun-Hui Liu^{1,2}, *Fellow, IEEE*

Abstract—Efficiency and reliability are critical in robotic bin-picking as they directly impact the productivity of automated industrial processes. However, traditional approaches, demanding static objects and fixed collisions, lead to deployment limitations, operational inefficiencies, and process unreliability. This paper introduces a Dynamic Bin-Picking Framework (DBPF) that challenges traditional static assumptions. The DBPF endows the robot with the reactivity to pick multiple moving arbitrary objects while avoiding dynamic obstacles, such as the moving bin. Combined with scene-level pose generation, the proposed pose selection metric leverages the Tendency-Aware Manipulability Network optimizing suction pose determination. Heuristic task-specific designs like velocity-matching, dynamic obstacle avoidance, and the resight policy, enhance the picking success rate and reliability. Empirical experiments demonstrate the importance of these components. Our method achieves an average 84% success rate, surpassing the 60% of the most comparable baseline, crucially, with zero collisions. Further evaluations under diverse dynamic scenarios showcase DBPF’s robust performance in dynamic bin-picking. Results suggest that our framework offers a promising solution for efficient and reliable robotic bin-picking under dynamics.

Index Terms—Reactive and Sensor-Based Planning, Perception for Grasping and Manipulation, Task and Motion Planning, Collision Avoidance, and Bin Picking.

I. INTRODUCTION

BIN-PICKING, a fundamental problem in robotics, involves a robot manipulator retrieving items from a bin. The existing bin-picking methods [1]–[3] widely rely on open-loop workflow, i.e., Sense-Plan-Act (SPA). This approach depends on a limited number of perceptions prior to robot execution and has demonstrated practicality in static scenarios where both objects and the environment remain stationary. However, many real-world applications in manufacturing, logistics, and retail industries often entail dynamic scenarios. A typical situation might require a robot manipulator tasked with picking target objects from a bin in transit, moved by a belt conveyor or a mobile robot. Under



Fig. 1. Dynamic bin-picking: The robot picks up a single object among a cluster of arbitrary objects stacked in a moving bin on a belt conveyor. Our novel framework enables the robot to achieve dynamic bin-picking, ensuring collision-free and adaptive motion. Our method outperforms traditional bin-picking methods in terms of time efficiency, showcasing exceptional effectiveness and reliability in dynamic scenarios.

such circumstances, the static assumptions of traditional bin-picking operations become inflexible and overly restrictive, leading to operational inefficiencies and unreliability.

Current bin-picking research primarily focuses on static environments, based on perception techniques such as 3D object localization [4], 6D pose estimation [5] [6], and grasping planning [7] [8] to process sensor data and generate effective picking poses for stationary items. However, these approaches are susceptible to unexpected external disturbances on initially stationary scenes once the robot commences a picking motion [9] [10]. Furthermore, if these methods were applied to pick objects from a moving bin, the effort for frequent halting and restarting of the bin could significantly degrade picking efficiency and lead to unnecessary energy consumption. In contrast, picking objects directly from a moving bin, i.e., dynamic bin-picking (Fig. 1), could reduce time overheads, enhance productivity, and save energy. Therefore, it is worth exploring dynamic bin-picking, which challenges the static assumptions of traditional bin-picking scenarios. The real-time generation of a feasible picking pose is essential to realize dynamic bin-picking. Some previous studies have addressed moving object picking [11]–[13] by integrating 6D Pose estimation and grasping planning. However, these strategies are primarily aimed at a single-moving object of limited categories and fall short when dealing with multiple-moving arbitrary objects inherent in dynamic bin-picking. As the quantity and diversity of objects increase, the preparatory work for the 6D pose estimation and instance segmentation models becomes more burdensome.

*This work is supported by the InnoHK of the Government of the Hong Kong Special Administrative Region via the Hong Kong Centre for Logistics Robotics. The work is also supported in part by RGC via grant 14207119. Corresponding to Yun-Hui Liu (e-mail: yhliu@mae.cuhk.edu.hk)

¹Yichuan Li, Rui Cao, Yun-Hui Liu are with the T Stone Robotics Institute, the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong (e-mail: yichuan.li@link.cuhk.edu.hk; rcao@mae.cuhk.edu.hk;)

²Junkai Zhao, Yixiao Li and Yun-Hui Liu are with the Hong Kong Centre for Logistics Robotics (HKCLR) (e-mail: zhaojunkai1515@gmail.com; yixili@student.ethz.ch)

³Masayoshi Tomizuka and Zheng Wu are with the Department of Mechanical Engineering, University of California at Berkeley (e-mail: tomizuka@berkeley.edu; zheng.wu@berkeley.edu)

Besides, the moving bin itself poses a significant dynamic obstacle, and the consideration of bin collisions, which are often overlooked [14] [15], is crucial. Dynamic bin-picking involving moving objects and dynamic obstacles necessitates a motion planning framework capable of real-time adaptation to changing objectives and constraints.

This work advances dynamic bin-picking by relaxing static assumptions common in traditional approaches. Our contributions are: (1) A novel framework utilizing horizon-based discrete trajectory optimization for rarely explored dynamic bin-picking, enabling efficient collision-free robot motion to pick moving objects from moving or disturbed bins. (2) The introduction of the pose selection metric, including the Tendency-Aware Manipulability Network (TAMN), with a combination of the scene-level suction pose generation, facilitating optimal pose determination for multiple moving arbitrary objects. (3) Heuristic, task-specific designs incorporated into optimization objectives, constraints, and task planning model, enhancing picking success rate and reliability in practice. (4) Empirical evidence from real-world experiments showcasing the effectiveness and reliability of our framework for dynamic bin-picking, outperforming competitive baseline methods.

II. RELATED WORK

Static Bin-Picking: The conventional static bin-picking method typically adheres to a Sense-Plan-Act framework. The Sense module analyzes sensor data to determine the picking poses of the objects, the Plan module devises an offline motion trajectory, and the Act module executes this predetermined trajectory to retrieve the target. The primary task in static bin-picking lies in determining a reliable picking pose for object [16] [17]. Numerous studies have delved into deep learning-based object picking pose estimation [2], [6], [18], [19]. Zeng et al. [20] proposed grasping primitives applicable to diverse objects, utilizing fully convolutional networks to infer grasping affordances. Similarly, we propose leveraging learning-based methods for picking pose generation, but we further extend beyond by incorporating Tendency-Aware Manipulability into consideration for moving objects. Ichnowski et al. [14] proposed a rapid bin-picking method using sequential quadratic programming (SQP) to optimize the robot’s motion. However, this approach might not be applicable in dynamic environments where bin movements introduce additional complexity and unpredictability that are not considered in a static SQP framework.

Moving object picking: In the process of moving object picking, the robot needs to continuously track the object and reason about picking pose, meanwhile adjusting the arm motion accordingly. Akinola et al. [12] proposed a dynamic object grasping framework where the appropriate grasp is filtered from reachability and motion-awareness and the arm motion is generated by PRM in a seeding way. However, their framework only specializes in grasping a single object moving on a belt conveyor and excludes environmental collision. In contrast, our work deals with more complex scenarios where the robot picks an object from a clutter

of arbitrary objects in a moving bin. Burgess-Limerick et al. [21] use the image-based visual servoing control to grasp objects with unpredictable movements. However, the top-down grasp pose lacks generality across various object types. Further, [11] proposes a dynamic grasp re-ranking metric to select the best 6D grasp pose for the moving object. Position-based visual servoing is adopted but often overlooks collision risks when reaching objects. Unlike most previous studies using two-fingered grippers, we employ suction cups to better suit industrial settings. We present a novel framework to pick multiple moving objects while ensuring the avoidance of moving obstacles, i.e., the moving bin.

III. DYNAMIC BIN-PICKING FRAMEWORK

In this section, we illustrate the proposed framework (As shown in Fig. 2) in detail, which is implemented in a fully closed-loop manner with several modules and a task-level planning model that allows the robot to actively adapt to changes during the dynamic bin-picking tasks.

A. Suction pose generation and selection

Suction pose generation: Pose candidates S with strong suction potential are continuously generated from the post-processed point cloud, which keeps objects only. Given the inevitable occlusions for a third-view camera due to the presence of bin walls or the robot arm, we utilize the in-hand configuration. Preliminary poses are randomly sampled as the normal on objects’ surfaces and ranked using the SuctionNet (SN) considering factors of cup seal formation and wrench resistance [22].

Suction pose selection: Considering manipulability at the goal pose can help avoid potential singularity and increase the execution success rate. The ability to continue moving in a specific direction from a pose greatly influences the robot’s continuous transition from the current pose along the predicted direction of motion. Therefore, we define the Tendency-Aware Manipulability (TAM) score for a given pose and its moving tendency. We employ the inverse condition number C as a more robust and representative metric for manipulability than $|J^T J|$ (as per Togai [23]), given end-effector pose p :

$$C(p) = \begin{cases} \frac{1}{\text{Cond}(J(q))}, & \text{if } \exists q \in IK(p) \\ -1, & \text{otherwise} \end{cases} \quad (1)$$

Where the q is joint positions from Inverse Kinematics (IK) and the J is the kinematic Jacobian of q . Furthermore, we propose the use of a quality index $I(q, n_p)$ (as in [24]), which is the magnitude of the Jacobian’s gradient with respect to a moving direction vector, n_p :

$$I(q, n_p) = |J(q)^T n_p| \quad (2)$$

By combining the manipulability metric C with this quality index, we derive the TAM score:

$$T(p, n_p) = I(IK(p), n_p) C(p) \quad (3)$$

However, performing online IK and TAM computations for a set of candidate suction poses can be time-consuming,

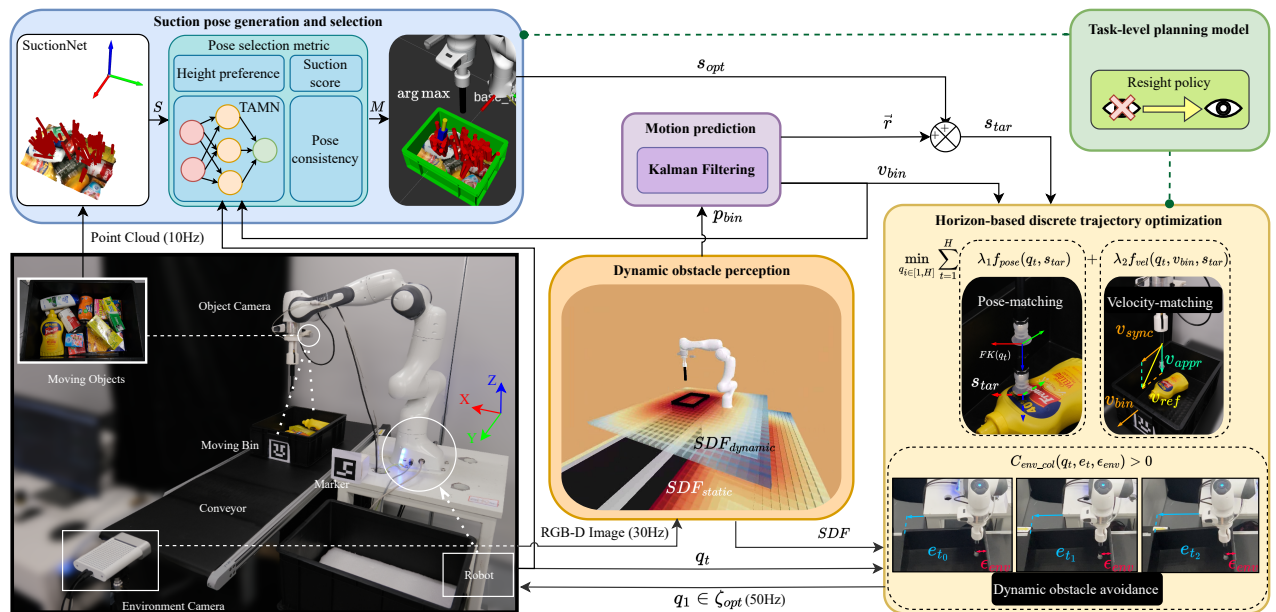


Fig. 2. Framework overview: The DBPF is implemented in a fully closed-loop manner to facilitate the robot’s reactivity in dynamic bin-picking. An eye-in-hand object camera captures the point cloud of moving objects within the bin at 10Hz. With the suction pose generation and selection, a set of candidate suction poses S are generated, and an optimal pose s_{opt} (yellow cylinder) is selected considering factors like the Tendency-Aware Manipulability (through TAMN), pose consistency, height preference, and suction score. Motion prediction forecasts bin’s displacement \vec{r} to advance the s_{opt} as s_{tar} and offers bin velocity v_{bin} . An eye-to-hand environment camera obtains RGB-D images at 30Hz, and the dynamic obstacle perception module detects the pose of the moving bin p_{bin} . Two Signed Distance Functions SDF_{static} and $SDF_{dynamic}$ are maintained actively corresponding to environmental collisions. Horizon-based discrete trajectory optimization solves the optimal trajectory ζ_{opt} with objectives like pose-matching f_{goal} and velocity-matching f_{vel} . Dynamic obstacle avoidance is realized via collision constraints. The first point q_1 of each optimal path ζ_{opt} is executed at 50Hz, and the robot’s current joint position q_t is constantly updated. Lastly, a task-level planning model integrates these modules tightly to effectively achieve dynamic bin-picking while preventing “Poor Observations” through the resight policy.

which could compromise reactivity [12]. To resolve this, we trained a Tendency-Aware Manipulability Network (TAMN) to infer TAM scores given 6D poses and their moving tendencies. TAMN was trained using a Multi-Layer Perceptron (MLP) network and optimized by minimizing the Mean Squared Error (MSE) loss, with a learning rate of 0.0001 for 200 epochs. The training and validation loss are lower than 0.004. For training dataset, we randomly sampled 100,000 picking pose p within the workspace, and calculated 128 TAM scores via Eq. 1-3 for each p by partitioning the space into 128 possible directions n_p uniformly. The re-ranking score of a suction pose s from a set of candidate suction poses S can be computed as the pose selection metric M :

$$M(s) = w_1 T(p_t, \frac{r(s) - r(p_t)}{|r(s) - r(p_t)|}) + w_2 T(s, \frac{v_{ref}(s)}{|v_{ref}(s)|}) + w_3 \frac{1}{pd(s, s_{t-1})} + w_4 \frac{Z(s)}{\bar{Z}(S)} + w_5 U(s) \quad (4)$$

Where the $r(s)$ is the translation vector of s , the p_t is the current end-effector pose derived from Forward Kinematics $FK(q_t)$ of the current joint position q_t , the $v_{ref}(s)$ from Eq.14 is the potential moving velocity upon reaching s , the $pd(s, s_{t-1})$ refer to Eq. 12 is the pose difference between s and the previously determined pose s_{t-1} , the $z(s)$ is the height of s , the $\bar{Z}(S)$ is the average height of all candidates S , the $U(s)$ is the suction score from SN. The M balances terms with weights w_i for (1) goal pose accessibility, (2) goal pose TAM, (3) pose consistency, (4) height preference, and (5) suction score. Thus the optimal suction pose s_{opt} with

the highest $M(s)$ is determined from a pool of candidate suction poses S as $s_{opt} = \arg \max M$.

B. Dynamic Obstacle Perception

Static collisions (e.g., table, conveyor) are configured before the task, while real-time obstacle perception is needed for dynamic collisions like the moving bin. The pose of the moving bin p_{bin} , represented by five cuboids, is detected using fiducial markers. The robot’s links are shown as spheres. Both static and dynamic collisions are modeled as cuboids and spheres, allowing analytical computation of Signed Distance Functions (SDF) in the defined workspace. For computational efficiency, two SDF are maintained: SDF_{static} is computed once at the task’s start, and $SDF_{dynamic}$ is updated with each new observation of dynamic collisions. These SDF are used to calculate minimum distance vectors between robot link spheres and collisions in real-time.

C. Motion prediction

Due to the inevitable time cost for perception, the inferred suction pose will always be antiquated. Consequently, prediction of the future state of the moving objects is necessary. As we have the online 6D pose of the moving bin p_{bin} refer to Section III-B, and assume the objects and bin are relatively stationary, we employ Kalman Filtering (KF) [25] to predict the bin’s displacement vector \vec{r} and velocity v_{bin} using a linear motion model. Thus, the s_{opt} is advanced to yield the target suction pose s_{tar} :

$$s_{tar} = s_{opt} + \vec{r} \quad (5)$$

D. Horizon-based discrete trajectory optimization

The primary objective of dynamic bin-picking is to facilitate robotic motion toward objects in a moving bin with constraints. To encapsulate all kinds of requirements comprehensively and solve appropriate trajectories online, we formulate the problem through a horizon-based discrete trajectory optimization structure:

$$\min_{q_i \in [1, H]} \sum_{t=1}^H \lambda_1 f_{pose}(q_t, s_{tar}) + \lambda_2 f_{vel}(q_t, v_{bin}, s_{tar}) \quad (6)$$

$$\text{subject to: } l_i \leq q_i \leq u_i, \quad (7)$$

$$\dot{l}_i \leq \dot{q}_i \leq \dot{u}_i, \quad \ddot{l}_i \leq \ddot{q}_i \leq \ddot{u}_i, \quad \forall i \quad (8)$$

$$q_{t+1} - q_t = \dot{q}_{t+1} dt \quad (8)$$

$$\dot{q}_{t+1} - \dot{q}_t = \ddot{q}_{t+1} dt \quad (9)$$

$$C_{self_col}(q_t, \epsilon_{self}) > 0 \quad (10)$$

$$C_{env_col}(q_t, e_t, \epsilon_{env}) > 0 \quad (11)$$

Where the Eq. 6 are the objective functions, the λ_1 and λ_2 are objective weights; Eq. 7 set the joint limitations; Eq. 8 and Eq. 9 establish the correlation among joint position, velocity, and acceleration; Eq. 10 is the self-collision constraint; Eq. 11 is the env-collision constraint; H , the horizons, reflect the future steps considered. Optimal H balances computational efficiency with planning quality, ascertained empirically.

Pose-matching objective: The pose difference between two Cartesian poses p_a and p_b are considered as in [26]:

$$pd(p_a, p_b) = \|\alpha_1(I - {}^w R_b^T {}^w R_a)\|_2 + \|\alpha_2({}^w R_b^T {}^w d_a - {}^w R_b^T {}^w d_b)\|_2 \quad (12)$$

Where the R is the rotation and the d is the translation of the pose. The α_1 and α_2 are weights to bias importance. Consequently, the Pose-matching objective f_{pose} can be derived as:

$$f_{pose}(q_t, s_{tar}) = pd(FK(q_t), s_{tar}) \quad (13)$$

This formulation enables the robot to approach the target pose by minimizing the pose difference through optimization.

Velocity-matching objective: To improve the effectiveness of picking during the final approach, we aim to maintain a relative still between the end-effector and the moving objects in the direction of bin motion. We refer to this as the Velocity-matching objective, inspired by the V-bar maneuver detailed in [27], a fundamental strategy employed in space station docking. Given the z-axis of a picking pose s as $N_z(s)$, a speed variable α_v determines the desired approaching velocity component of the end-effector through $v_{appr}(s) = \alpha_v N_z(s)$. The desired synchronizing velocity component of the end-effector follows $v_{sync} = v_{bin}$. Therefore, the reference velocity of end-effector v_{ref} and velocity matching objective f_{vel} are derived as:

$$v_{ref}(s) = v_{sync} + v_{appr}(s) = v_{bin} + \alpha_v N_z(s) \quad (14)$$

$$f_{vel}(q_t, v_{bin}, s_{tar}) = \|\dot{p}_t - v_{ref}(s_{tar})\|_1 \quad (15)$$

Where the \dot{p}_t is the current end-effector velocity calculated as $\dot{p}_t = J(q_t)\dot{q}_t$.

Collision avoidance constraint: As mentioned in Section

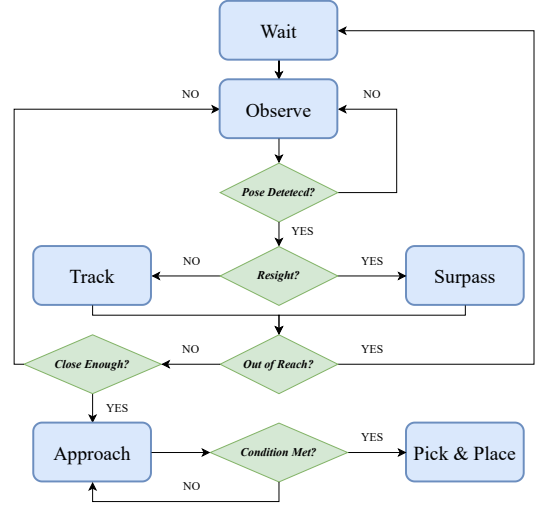


Fig. 3. Task-level planning model consists of six actions for dynamic bin-picking. The actions are the Wait action in a standby state, the Observe action to perceive the suction pose and bin state, the Track action to follow the target pose, the Surpass action for rushing forward and regaining a decent view for observation, the Approach action to travel the final distance to contact with object surface, and the Pick & Place action to attach the object from the bin and lift, and drop at placing location.

III-B, we can get the SDF for environmental collisions e_t in real-time. The minimum distance vectors between robot and collisions d_{static} and $d_{dynamic}$ are obtained given q_t . Thus, the env-collision constraint C_{env_col} is formulated as:

$$C_{env_col}(q_t, e_t, \epsilon_{env}) = \phi_1 d_{static} + \phi_2 d_{dynamic} - \epsilon_{env} \quad (16)$$

Where the ϵ_{env} is the env-collision distance threshold, the ϕ_1 and ϕ_2 are weights. The self-collision is also considered by training a self-collision distance checking network as in [26]. The trained network gets input as q_t and predicts the minimal distance between robot links, denoted as d_{links} . The self-collision constraint C_{self_col} is derived as:

$$C_{self_col}(q_t, \epsilon_{self}) = d_{links} - \epsilon_{self} \quad (17)$$

Where the ϵ_{self} is the self-collision distance threshold.

We solve the above optimization formulations by Model Predictive Path Integral (MPPI) Control refer to [28], and take the first point q_1 of the solved optimal trajectory ζ_{opt} for execution. The online time-optimal trajectory interpolation is employed inspired by [29] to further interpolate position commands for real-time position control of the robot.

E. Task-level Planning Model

The task-level planning model is shown in Fig. 3. During the picking process, the robot initially waits for the bin to enter the perception region before determining the suction pose and bin state. However, challenges arise from “Poor Observations” when the in-hand camera gets close to objects, including (a) occlusion causing insufficient point acquisition, (b) limited Field of View (FOV) missing points on the tracked object, and (c) perspective uncertainty from oblique views perceiving near-horizontal poses that are hard to execute. Thus, we developed a Resight policy (Alg.1) that triggers a Surpass action to exceed the target and regain a better observation perspective. When the end-effector is

within a distance d_{thr} of the target pose, it starts approaching the object surface, following the velocity-matching objective. Upon detecting apparent contact force on the z-axis of the end-effector via the robot’s built-in torque sensor, the vacuum generation is triggered. The contacted object is then picked, lifted, and dropped at a designated location.

Algorithm 1 Resight Policy

Input: Perception results from Observe action
Output: Track or Surpass action to be executed

- 1: PCL Points threshold $N_{min} = 1024$, Shift threshold $\epsilon_{shift} = 0.05$, TAM threshold $M_{min} = 0.3$, Scale-up factor $\sigma_{scale} = 1.3$
- 2: $M \leftarrow w_1 T(p_t, \frac{r(s_{opt})-r(p_t)}{|r(s_{opt})-r(p_t)|}) + w_2 T(s_{opt}, \frac{v_{ref}(s_{opt})}{|v_{ref}(s_{opt})|})$
- 3: **if** $N_{pcl} > N_{min}$ **then** \triangleright No Occlusion
- 4: **if** $\|r(s_{opt}) - r(s_{t-1})\|_2 < \vec{r} + \epsilon_{shift}$ **then** \triangleright No FOV Limitation
- 5: **if** $M > M_{min}$ **then** \triangleright No Oblique View
- 6: $s_{tar} \leftarrow s_{opt} + \vec{r}$
- 7: **return** $q_{track} \leftarrow \text{Optimization}(q_t, s_{tar})$
- 8: **end if**
- 9: **end if**
- 10: **end if**
- 11: $s_{tar} \leftarrow (s_{t-1} + \sigma_{scale}\vec{r}) + \vec{r}$
- 12: **return** $q_{surpass} \leftarrow \text{Optimization}(q_t, s_{tar})$

IV. EXPERIMENTAL VALIDATIONS

This section quantitatively evaluates the proposed DBPF for dynamic bin-picking tasks through several real-world experiments. First, we introduce the experiment setup and evaluation metric. Then, the ablation study is conducted. Followed by the comparisons of the proposed method with baseline methods are analyzed. Finally, The performance of our methods with varying characteristic variables of the fully dynamic scene is explored.

A. Experiments Setup and Evaluation Metrics

TABLE I
 ABLATION EXPERIMENTS OF THE PROPOSED FRAMEWORK ON
 DYNAMIC BIN-PICKING

| Methods | 11 objects | | |
|-------------|------------|----------------|-----|
| | SR | TT (s) | CR |
| Ours-no-VM | 34% | 4.7±1.0 | 0.0 |
| Ours-no-RS | 67% | 6.3±3.4 | 0.0 |
| Ours-no-DO | 79% | 3.9±1.3 | 19% |
| Ours-no-TAM | 61% | 4.7±1.0 | 0.0 |
| Ours-Full | 87% | 6.6±2.1 | 0.0 |

Experiments Setup. A standard logistic container bin containing several arbitrary objects is moved on a belt conveyor or disturbed by human hands. A 7-DOF Franka Panda manipulator equipped with a suction gripper system is employed to generate vacuum force on objects’ surfaces for picking. A Realsense D435 camera is mounted on the Franka wrist for object perception. A third-view fixed camera Azure Kinect is placed externally, observing the environment’s dynamic collision, i.e., the moving bin. For the sake of

precise and timely bin collision perception, we use ArUco fiducial markers [30] placed on one wall of the bin and a location with known transformation to the robot base. ROS is employed to enable seamless communication between all the modules.

Evaluation Metrics. We take the *Success Rate* (SR) - The percentage of successful trials concerning total trials, in which the success trial is defined as successfully picking the object from the moving bin; *Total Time* (TT) - Time in seconds elapsed to finish picking motion; *Collision Ratio* (CR) - The ratio of collisions over the total trials during the whole picking procedure. We implement 100 trials on each scenario setting for every method.

B. Ablation Study

We conducted ablation studies to evaluate the performance of novel designs in our framework. We implement the ablations: (1) *Ours-no-VM*: the velocity-matching is excluded, replacing by open-loop approaching based on predicted pose; (2) *Ours-no-RS*: the resight policy is discarded, turning to execute Track action always (3) *Ours-no-DO*: the dynamic obstacle detection and avoidance are disabled; (4) *Ours-no-TAM*: the TAM score is not considered during pose selection; (5) *Ours-Full*: the complete version of our DBPF. 11 objects are placed in a moving bin with a speed of 8cm/s.

The results are shown in Table I, *Ours-no-VM* achieves the lowest success rate of 34% compared to *Ours-Full* of 87%. The disparity reveals the significance of velocity matching. It is more effective to pick up the object by maintaining relative still, as this facilitates the suction cup seal formation on the object’s surface. In addition, the velocity matching during the approach action is implemented in a velocity-feedback closed-loop fashion, which should be more robust than the open-loop approach that only relies on initial prediction. *Ours-no-DO* achieves the shortest total picking time but collides with the bin wall, as it excludes the computationally intensive collision checking. *Ours-no-RS* has almost a 67% success rate showing a noticeable drop to *Ours-Full*. The robot often struggles with “Poor Observations” accompanied by the in-hand camera configuration during object tracking. The result indicates the effect of the resight policy, which helps the robot regain a better perspective to perceive more feasible picking poses. *Ours-no-TAM* exhibits a considerable decrease in success rate, with a degradation of 29.5% compared to *Ours-Full*. The absence of manipulability consideration in *Ours-no-TAM* leads to singularities in the robot’s motion and are disadvantageous for continuous moving along with moving target, thereby causing picking failures.

C. Comparison to Baseline methods

We evaluate our framework against three potentially competitive baselines. For fairness, we slightly modify these baselines in object perception by using our suction pose generation module, and make sure the same maximum joint velocity of the robot during motion:

Sense-Plan-Act (SPA): The picking pose is generated once at the beginning, and offline motion planning is conducted based on the classic sampling planner RRT [31] considering

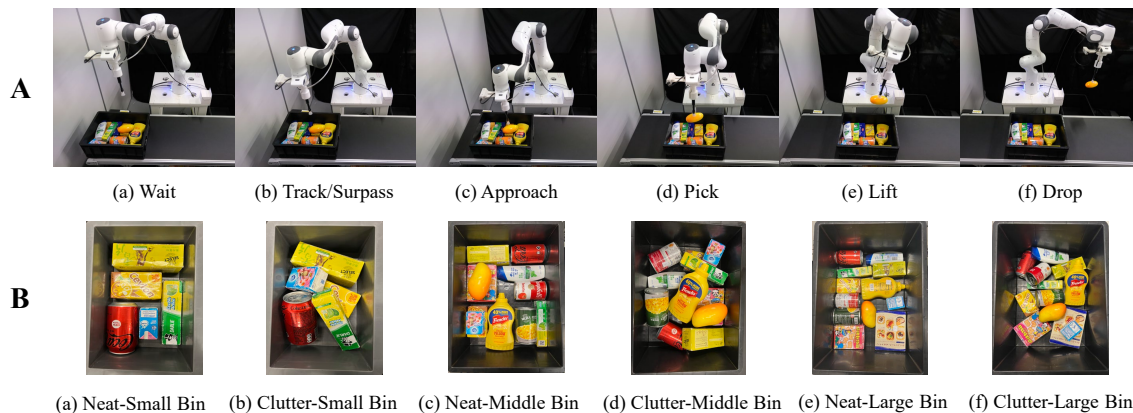


Fig. 4. Snapshots of dynamic bin-picking. A: (a)-(f) demonstrate the process of picking an object from a clutter of objects from a moving bin on a belt conveyor. B: (a)-(f) depict Neat and Clutter arrangements of objects in varying bins ranging from Small to Large in size.

TABLE II
BASELINE COMPARISON EXPERIMENTS WITH PROPOSED FRAMEWORK IN DYNAMIC BIN-PICKING TASKS

| Scene | Methods | 1 object | | | 5 objects | | | 11 objects | | |
|-------------------|---------|------------|----------------|------------|------------|----------------|------------|------------|----------------|------------|
| | | SR | TT (s) | CR | SR | TT (s) | CR | SR | TT (s) | CR |
| Dynamic-to-static | SPA | 96% | 12.3±0.7 | 0.0 | 95% | 13.9±1.2 | 0.0 | 96% | 13.0±0.8 | 0.0 |
| Fully dynamic | SPA | 5% | 5.6±0.4 | 69% | 19% | 6.0±0.5 | 71% | 14% | 5.5±0.4 | 68% |
| | PBVS | 14% | 4.7±1.3 | 33% | 30% | 5.5±0.9 | 44% | 28% | 5.1±1.0 | 27% |
| | LSPPA | 57% | 7.2±3.9 | 21% | 63% | 5.8±0.6 | 14% | 59% | 5.8±1.6 | 17% |
| | Ours | 89% | 5.0±1.0 | 0.0 | 81% | 5.5±2.3 | 0.0 | 80% | 5.2±1.3 | 0.0 |
| Disturbed static | SPA | 37% | 7.4±0.7 | 39% | 71% | 8.1±0.7 | 43% | 62% | 8.7±1.7 | 42% |
| | PBVS | 36% | 5.4±0.4 | 30% | 59% | 5.3±0.4 | 29% | 64% | 5.2±0.3 | 29% |
| | LSPPA | 75% | 5.6±0.7 | 18% | 72% | 6.1±1.0 | 12% | 64% | 6.3±0.9 | 14% |
| | Ours | 83% | 7.5±3.0 | 0.0 | 85% | 8.0±3.7 | 0.0 | 87% | 6.9±3.1 | 0.0 |

pre-defined static collision, and then the robot is triggered to execute the planned trajectory towards the target object. This method is widely applied in existing static robotic bin-picking scenes.

Position-based Visual Servoing (PBVS): The robot continuously tracks the perceived picking pose through position-based visual servoing [11]. When the tracking is close to the target pose, an open-loop approaching is conducted for the final distance.

Loop-Sense-Predict-Plan-Act (LSPPA): We adapted the method from Akinola et al. [12] as a baseline, designed for grasping a single moving object on a belt conveyor. Given the newly generated picking pose, the motion trajectory is planned using the PRM [32] planner with online trajectory seeding and static collision avoidance. The system implements trajectory blending for each new trajectory during execution and approaches a predicted pose in an open-loop manner when close enough. We adapted our TAMN to handle multiple objects, replacing their consideration of reachability.

In the following, we implement several experiments to evaluate the performance of our DBPF (**Ours**) against the above-mentioned baselines. The experiments are conducted in three scenes: (i) *Dynamic-to-Static*: This represents the classic industrial scene, in which the bin is transported in front of the robot and stopped. The bin-picking starts in a static situation. (ii) *Fully Dynamic*: The robot picks objects while the bin moves continuously. The bin moves linearly

with a constant speed of 6cm/s in all settings. (iii) *Disturbed Static*: The bin is stationary in the picking area of the robot. After the robot starts picking motion, the bin is disturbed manually by linear displacements in the xy plane. In addition, we vary the number of arbitrary objects packed in the bin to 1, 5, and 11 objects, respectively.

Fig.4A shows experiment snapshots of picking an object from the moving bin using our DBPF. The obtained results are summarised in Table II. Traditional SPA takes nearly 13 seconds to pick up an object from the bin in the *Dynamic-to-Static* scene, counting the time spent before the bin is stopped. In comparison, our DBPF significantly reduces the picking time to approximately 6 seconds while maintaining a slightly lower success rate, showcasing the significant improvement in picking efficiency enabled by our approach.

In the *Fully Dynamic* scene, our method displays a superior overall success rate of approximately 84%, exceeding the highest performing baseline (LSPPA) which achieves a success rate of 60%. The PBVS has the quickest picking time due to the approximate linear motion on the end-effector for the visual-servoing method, but it also has the second-lowest success rate. The SPA method exhibits the lowest success rate and highest collision ratio, attributable to its open-loop workflow. Remarkably, our method achieves zero collisions during picking, contrasting sharply with the SPA, which has the highest collision ratio at 70%. This can be attributed to the following reasons: 1) Our pose selection with the

TABLE III
PERFORMANCE OF PROPOSED FRAMEWORK WITH VARYING FULLY DYNAMIC CHARACTERISTIC VARIABLES

| Arrangement | Conveyor speed | Small Bin | | | Middle Bin | | | Large Bin | | |
|-------------|----------------|-----------|---------|-----|------------|---------|-----|-----------|---------|-----|
| | | SR | TT (s) | CR | SR | TT (s) | CR | SR | TT (s) | CR |
| Neat | Slow | 87% | 7.3±4.0 | 0.0 | 80% | 5.6±4.1 | 0.0 | 78% | 7.3±3.4 | 0.0 |
| | Middle | 77% | 7.8±6.9 | 0.0 | 90% | 7.4±1.0 | 0.0 | 81% | 6.7±1.9 | 0.0 |
| | Fast | 81% | 7.7±3.7 | 0.0 | 74% | 7.0±6.2 | 0.0 | 79% | 7.0±3.2 | 0.0 |
| | Varying | 73% | 7.6±4.6 | 0.0 | 72% | 6.7±2.6 | 0.0 | 70% | 7.3±4.4 | 0.0 |
| Cluttered | Slow | 74% | 6.4±3.9 | 0.0 | 83% | 5.3±1.7 | 0.0 | 73% | 7.2±3.5 | 0.0 |
| | Middle | 72% | 8.3±4.4 | 0.0 | 88% | 6.6±2.1 | 0.0 | 73% | 7.1±3.9 | 0.0 |
| | Fast | 80% | 7.0±7.0 | 0.0 | 71% | 6.9±1.7 | 0.0 | 81% | 7.3±3.7 | 0.0 |
| | Varying | 81% | 7.6±6.2 | 0.0 | 73% | 7.6±8.4 | 0.0 | 74% | 9.4±7.0 | 0.0 |

proposed TAMN filters out hard-to-execute or inconsistent picking poses. Instead, PBVS and LSPPA often fail due to jittery motion and oscillation because of frequently changing picking poses; 2) Our use of velocity matching ensures relative stillness and closed-loop approaching, enhancing the success rate over open-loop methods used in the baselines. This aids in cup seal formation and adapts to dynamics once the approaching begins; 3) Our resight policy handles unexpected “Poor Observations” situations frequently encountered in dynamic bin-picking, which causes the loss of tracking for baseline methods leading to picking failures; 4) The consideration of dynamic obstacles, ensures the safety and reliable operation throughout the process, which is proven to be collision-avoidance-effective in experiments.

As for the *Disturbed Static* scene, our method continues to show the highest success rate over the baselines when faced with external disturbances, as illustrated in Fig. 5 (Left). This property is crucial since many current static bin-picking applications typically assume completely controlled environments and disregard potential disturbances during picking. However, our method does exhibit a relatively high time variance under *Disturbed Static* scene compared to other baselines in Fig. 5 (Right). This is primarily due to the random disturbances that can easily lead to “Poor Observations.” Our resight policy helps to trigger the Surpass action, adjusting the camera’s perspective and converging the tracking to a more desirable state before approaching, albeit with a longer duration. Besides, changing collisions can disrupt the original tracking motion, requiring more time to avoid collisions. Despite these challenges, our experimental results demonstrate that our framework surpasses baseline methods in dynamic bin-picking scenes, providing expected efficiency and reliability.

D. Evaluation on varying Fully Dynamic scenarios

This section explores our framework’s potential for the *Fully Dynamic* scenes with varying characteristic variables. We define two object arrangements: *Neat* and *Clutter*. In the *Neat* arrangement, arbitrary objects in the bin are organized with minimal rotations and negligible overlap between objects. Conversely, the *Clutter* arrangement sees objects randomly placed in the bin, in contrast to the *Neat* arrangement. For each arrangement (Fig. 4B), we use bins with different sizes, i.e., *Small* of $30 \times 20 \times 12$ (cm), *Middle* of $41 \times 30 \times 15$

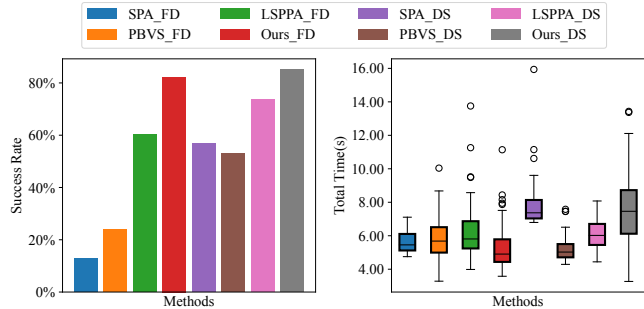


Fig. 5. Success rate (left) and total time (right) of dynamic bin-picking. *_FD: Methods under the Fully Dynamic scene. *_DS: Methods under the Disturbed Static scene.

(cm), and *Large* of $48 \times 35 \times 22$ (cm). Additionally, the bin moves at four different conveyor speeds: *Slow* at 6cm/s, *Middle* at 8cm/s, *Fast* at 10cm/s, and *Varying*, where the speed changes between 6cm/s to 10cm/s.

Referring to the experimental results (Table III), our framework displayed comparable success rates in both *Neat* and *Clutter* arrangements, emphasizing DBPF’s robustness across different levels of object arrangement complexity. For varying bin sizes, the success rate was found to be lower in the *Large* bin compared to the *Small* or *Middle* bins. This is primarily due to the increased impact of the larger bin walls on the robot’s motion, as collisions can disrupt normal picking actions. As for conveyor speed variation, it appears to have a minimal impact on our method. However, we noted a slight decrease in success rates under *Varying* speed conditions compared to constant speed conditions. The primary reason for this is that changing speeds increase the dynamism of the environment, consequently escalating perception demands and necessitating a higher refresh rate. This could lead to delays in pose determination and bin state prediction, yielding a higher rate of picking failures. Lastly, the robot exhibits a shorter picking time when the bin moves at a *Slow* speed. This is expected because minor environmental changes make tracking actions easier to converge, thereby facilitating a quicker approach to the object.

E. Failure case and Limitations

Our system has certain limitations. Deformable objects hinder suction grasping due to poor sealing. Exclusively considering the linear motion, pick failures occur when bins or objects get rotated. Requiring prior knowledge of the environment, like bin model and fiducial markers, reduces

system flexibility. Reflective and transparent objects cause low-quality point cloud, adversely affecting grasp pose decisions. With the need for obstacle avoidance, picking near bin walls encumbers the swift picking motion. The unknown relative pose between the gripper and picked object impedes further manipulation, due to scene-level pose generation.

V. CONCLUSIONS AND FUTURE WORK

We introduced a novel and efficient framework for dynamic bin-picking, efficiently identifying optimal suction poses for moving objects and leveraging horizon-based discrete trajectory optimization for reactive motion control and dynamic obstacle avoidance. Future work includes replacing the suction gripper by a two-fingered gripper with collision modeling and grasp pose generation network. Also, we will address current limitations such as considering the rotational motion of bins and objects, directly handling unknown environments via learning-based methods, and improving the post-picking process by adding an in-hand pose estimation module for subsequent precise placement or assembly tasks.

REFERENCES

- [1] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.
- [2] Z. Dong, S. Liu, T. Zhou, H. Cheng, L. Zeng, X. Yu, and H. Liu, "Ppr-net: point-wise pose regression network for instance segmentation and 6d pose estimation in bin-picking scenarios," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1773–1780.
- [3] J. Yang, Y. Gao, D. Li, and S. L. Waslander, "Robi: A multi-view dataset for reflective objects in robotic bin-picking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9788–9795.
- [4] H. Cao, J. Zhou, J. Huang, Y. Li, N. C. Meng, R. Cao, Q. Dou, and Y. Liu, "Two-stage grasping: A new bin picking framework for small objects," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2584–2590.
- [5] J. Yang, D. Li, and S. L. Waslander, "Probabilistic multi-view fusion of active stereo depth maps for robotic bin-picking," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4472–4479, 2021.
- [6] X. Li, R. Cao, Y. Feng, K. Chen, B. Yang, C.-W. Fu, Y. Li, Q. Dou, Y.-H. Liu, and P.-A. Heng, "A sim-to-real object recognition and localization framework for industrial robotic bin picking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3961–3968, 2022.
- [7] X. Zhang, Y. Domae, W. Wan, and K. Harada, "Learning efficient policies for picking entangled wire harnesses: An approach to industrial bin picking," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 73–80, 2022.
- [8] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1997–2004.
- [9] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2013.
- [10] M. Roy, R. A. Boby, S. Chaudhary, S. Chaudhury, S. D. Roy, and S. K. Saha, "Pose estimation of texture-less cylindrical objects in bin picking using sensor fusion," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2279–2284.
- [11] C. De Farias, M. Adjigle, B. Tamadazte, R. Stolkin, and N. Marturi, "Dual quaternion-based visual servoing for grasping moving objects," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 151–158.
- [12] I. Akinola, J. Xu, S. Song, and P. K. Allen, "Dynamic grasping with reachability and motion awareness," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9422–9429.
- [13] F. Islam, O. Salzman, A. Agarwal, and M. Likhachev, "Provably constant-time planning and replanning for real-time grasping objects off a conveyor belt," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1370–1384, 2021.
- [14] 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)*. IEEE, 2020, pp. 5270–5277.
- [15] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, 2023.
- [16] T. H. Bui, Y. G. Son, S. J. Moon, Q. H. Nguyen, I. Rhee, J. Y. Hong, and H. R. Choi, "Deep learning based 6-dof antipodal grasp planning from point cloud in random bin-picking task using single-view," *IEEE Robotics and Automation Letters*, 2023.
- [17] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.
- [18] D. Buchholz, M. Futterlieb, S. Winkelbach, and F. M. Wahl, "Efficient bin-picking and grasp planning based on depth data," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3245–3250.
- [19] K. Kleeberger, C. Landgraf, and M. F. Huber, "Large-scale 6d object pose estimation dataset for industrial bin-picking," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2573–2578.
- [20] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *The International Journal of Robotics Research*, vol. 41, no. 7, pp. 690–705, 2022.
- [21] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke, "DGBench: An open-source, reproducible benchmark for dynamic grasping," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3218–3224.
- [22] H. Cao, H.-S. Fang, W. Liu, and C. Lu, "Suctionnet-1billion: A large-scale benchmark for suction grasping," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8718–8725, 2021.
- [23] M. Togai, "An application of the singular value decomposition to manipulability and sensitivity of industrial robots," *Siam journal on algebraic discrete methods*, vol. 7, no. 2, pp. 315–320, 1986.
- [24] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 568–573.
- [25] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [26] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. D. Ratliff, D. Fox, F. Ramos, and B. Boots, "Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 750–759.
- [27] F. L. Markley and J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*. Springer, 2014, vol. 1286.
- [28] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.
- [29] L. Berscheid and T. Kröger, "Jerk-limited real-time trajectory generation with arbitrary target states," *arXiv preprint arXiv:2105.04830*, 2021.
- [30] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [31] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.
- [32] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.