مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا
Sana'a University Journal of Applied Sciences and Technology
https://journals.su.edu.ye/index.php/jast/
SUN PUBLISHER

Smooth Planning of Catch-to-Throw Movements for Robotics Systems Through Continuous C^1 Trajectory Generation and State-Conditioned Release

Mohammed Almanari *, Adel Rawea and Abdullah A. Dhaiban

Department of Mechatronics, Faculty of Engineering, Sana'a University, Sana'a, Yemen.

*Corresponding author: mohammedalmanari2@gmail.com

Abstract

Traditional robotic catching and throwing process chains, often create a pause time of more than 300 ms by dividing catching and throwing into two consecutive steps, and inadequately utilize the post-grasping kinetic state. This paper presents a unified catching and throwing (UCT) process framework that eliminates this discontinuity by using the joint position and velocity (q_c, v_c) measured during grasping confirmation as the initial condition for a Hermite-type piecewise cubic launch trajectory. A six-state finite automaton coordinates the sequence, transitioning from catching to launch in less than 5 ms while maintaining continuity (C^1) to avoid velocity discontinuities. The gripper dynamics are represented by a first-order hold profile, and a Jacobian-based release velocity is coupled to a ballistic flight model for target prediction. The framework was evaluated in over 150 Drake simulations using a KUKA IIWA 14 manipulator. The results showed an overall success rate of 87.3%, an active cycle time of 0.854 s, and an average throw error of 41 mm at a target distance of 2.5 m. Compared to a reference analytical sequential method, the UCT framework reduced the cycle time by 34%, with the improvement primarily due to the elimination of the stop-start phase between capture and throw. The proposed framework is positioned as a practical level 2.5 architecture within a five-level taxonomy of capture-to-throw integration and provides a basis for future hardware implementation of continuous dynamic manipulation.

Article Info

Keywords:
Catch-throw , C^1 continuity , Cubic Hermite polynomial , Jacobian velocity mapping , Finite state machine , KUKA LBR iiwa manipulator , Drake simulation.

Article History:
Received: 05-Jun-2026,
Revised: 20-July-2026,
Accepted: 28-July-2026,
Published: 28 September 2026.

1. INTRODUCTION

Robotic systems that must successively catch and then throw an object exhibit a structural inefficiency [1, 2]: dominant architectures treat these two tasks as independent commands separated by a mandatory rest state. This design choice imposes a structural penalty: a dead time of 300 ms to 2 s per cycle and complete dissipation of the kinetic energy accumulated during the grasping maneuver [3, 4]. In critical real-time applications, such as automated sorting, dynamic assembly, and air conveying, this latency constitutes a fundamental throughput constraint, rather than an adjustable parameter.

The problem is further exacerbated at the system level. When the capture and launch controllers operate as independent probabilistic processes, the overall system success probability is limited by the product of their individual success rates. For example, a capture controller achieving 90% success, combined with an independent launch controller achieving 88% success, results in an overall success rate of at most 79.6%, representing a degradation of approximately 10 percentage points of architectural origin, independent of the hardware capabilities. This cumulative failure phenomenon is a key argument for trajectory-level integration, regardless of the latency considerations.

Previous studies have acknowledged this limitation to varying degrees. Senoo et al. [4] demonstrated that a high-speed launch with an optimized kinematic response could be achieved in less than 50 ms, thus laying the groundwork for dynamic release strategies. Namiki et al. [5] developed a high-speed, multi-finger receiving system capable of intercepting fast-moving objects, demonstrating the feasibility of real-time grasping adaptation in dynamic manipulation contexts. Learned throwing policies, notably TossingBot [6], achieve an accuracy of 35–50 mm at 1.5 m, but operate from static grasps, making them orthogonal to the interception problem. Batz et al. [7] addressed non-gripping dynamic receiving using ball tracking and model-based control, but focused exclusively on the interception phase without addressing the release after reception.

Trajectory generation methods based on Hermite cubic and polynomial interpolation [8, 9] offer continuous C^1 joint trajectories with explicit control of the limiting velocity, making them structurally suitable for boundary formulations, where initial conditions are measured in real time. Furthermore, Riemannian methods on SE(3) [10, 11] and task-parameterized probabilistic motion encodings [12] have been applied to fluid manipulation trajectories; however, neither is suitable for the dynamic transfer from catch to throw, where the initial conditions are determined only at the time of catch confirmation. Batz et al. [7] addressed non-grip dynamic catch using model-based ball state estimation and control but did not consider the subsequent release phase. Thus, the lack of data is clearly defined: no existing system uses the measured post-catch joint state (q_c, v_c) as a direct initial boundary condition for the throwing trajectory, thereby mathematically eliminating the rest state. This study presents a unified catch-to-throw (UCT) framework that addresses this gap through three tightly integrated mechanisms: (i) state-conditioned trajectory initialization, (ii) a six-state finite automaton with a catch-to-throw transition of less than 5 ms, and (iii) a first-order catch-to-throw release profile coordinated with a throw velocity derived from the Jacobian. The resulting system maintains C^1 continuity across the phase boundary by construction without requiring post-catch deceleration. We evaluated the UCT in Drake [13] on 150 controlled simulation trials and compared it with an analytically derived sequential reference. UCT is integrated into a five-level capture-launch integration taxonomy (Table 1) and currently occupies level 2.5: trajectory continuity is achieved, but the release configuration remains predefined and offline. The transition to Level 3 (fully online optimization) is described in the discussion. The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 presents the mathematical formulation. Section 4 describes the system architecture and design of the finite automaton. Section 5 details the experimental setup. Section 6 presents the results of the study. Section 7 discusses implications, limitations, and extensions. Section 8 concludes.

Table[1]: Five-level integration taxonomy. UCT = Level 2.5.

Lvl	Name	Latency
0	Isolated single task only	N/A
1	Sequential mandatory rest state	500 ms-2 s
2	State-Shared rest retained	~300 ms
2.5	UCT (This Work) C^1 Hermite, no rest	<5 ms
3	Fully Unified single objective	Open

2. RELATED WORK

A. Dynamic Capture and Interception: Single online objective q_r online. The open problem dynamic object capture has evolved from intercepting slow-moving targets [1] to predicting and capturing fast-moving flying objects in real time. Kim et al. [14] demonstrated the whole-body capture of objects launched at up to 7 m/s using predictive control on a humanoid platform, highlighting the importance of rapid state estimation and motion planning under tight time constraints.

Birbach et al. [15] developed a multi-hypothesis Kalman filter framework for predicting the trajectory of a bullet, achieving a prediction accuracy of 90 ms, which is sufficient for a 7-degree-of-freedom arm, thus establishing the perception basis upon which the UCT framework relies. Batz et al. [7] demonstrated the dynamic capture of non-grippy balls using model-based state estimation and real-time control, successfully intercepting fast-moving objects, although post-capture manipulation was not addressed in their study. These systems converge on the finding that rapid capture is possible. However, the question of what happens after capture remains unanswered. Coupled capture and throw architectures: Senoo et al. [4] provided fundamental results on high-speed robotic throwing through motion generation based on the kinematic response, demonstrating that the release timing can be controlled within a few tens of milliseconds. Namiki et al. [5] demonstrated high-speed multi-finger capture of dynamic objects and showed that real-time grip adaptation is possible; however, none of these studies addressed the problem of initiating a throwing trajectory directly from the post-capture state. None of these studies used the velocity measured after capture as a boundary condition for the trajectory, thus leaving the kinematic energy relationship between capture and throwing untapped.

C. Trajectory generation for manipulation: Hermite cubic interpolation provides C^1-continuous joint trajectories with explicit control of endpoint velocities [8], making it structurally well suited to boundary formulations in which initial conditions are measured in real time. Gasparetto and Zanotto [16] demonstrated that low-jump polynomial trajectories significantly reduce mechanical wear in industrial manipulators. Kunz and Stilman [17] achieved time-optimal trajectory generation along specified paths, and this technique can complement the UCT method during the launch phase. Dynamic movement primitives (DMPs) [18] provide an alternative trajectory representation that encodes goal-directed motions as second-order dynamical systems. Unlike the Hermite formulation, DMPs do not guarantee C¹ continuity at an externally imposed boundary condition. Schulman et al. [19] introduced trajectory optimization under geometric constraints and showed that online replanning following constraint changes is possible within planning horizons that are compatible with dynamic manipulation. CHOMP [20] provides a theoretical basis for extending the UCT method to the C² continuum with fifth-order polynomials by modeling motion planning as a covariant gradient descent on trajectory quality metrics.

D. Physics-based learning and launching: TossingBot [6] demonstrated that a physical residual model learned through large-scale self-supervised training can achieve an accuracy of 35–50 mm at 1.5 m for static-grip launching. Subsequent work on learned dynamic manipulation [21] extended this concept to out-of-grip thrusts and object orientation but did not address dynamic interference. Zeng et al. [22] showed that task-conditional policies can be generalized to different object geometries; however, all launch learning frameworks assume a stationary initial condition. UCT's contribution is clearly orthogonal: it provides a "bridge" between the initial condition required by any subsequent launch policy, whether analytic or learned, and dynamic capture.

E. Research gap: In the literature reviewed, no previous system (i) measures the post-capture joint state (q_c, v_c) at the time of grasp confirmation, (ii) directly uses this state as the initial boundary condition of the throwing trajectory, and (iii) maintains the continuity of velocity C¹ across the phase boundary without intermediate deceleration. The UCT addresses these three points simultaneously.

3. MATHEMATICAL MODELING

A. Description of the Robot's Kinematics:

The KUKA LBR IIwa robot is a serial robot with seven rotary joints (n=7) and no translational degrees of freedom. Torque sensors are placed after each gearbox. In addition, the robot was equipped with joint position encoders (see KUKA Robot Group [23]). Owing to its serial structure, its kinematics can be efficiently described using the modified Denavit-Hartenberg (MDH) convention, as in [24]. The axis of rotation of joint iii is denoted z_{i-1} , and the common normal axis of z_i and z_i is denoted x_i . The parameter γ_i = 0 indicates that joint iii is a rotating one. α_i denotes the angle between the axes z_{i-1} and z_i about x_{i-1} . θ_i is the angle between x_{i-1} and x_i about z_i . Finally, rl_i indicates the distance between axes x_{i-1} and x_i along z_i . The MDH parameters of the KUKA LBR iiwa 14 R820 robot are listed in Table (2). For the dynamic description of the robot, 12 physical parameters per link i , $\mu_i \in R^{12}$ as in (2), can be defined. Alternatively, twelve dynamic parameters per link i , $\pi_i \in R^{12}$ as in (3), can be used. Reduction to a minimal differential equation system [23, 24].

Table[2]: Modified Denavit-Hartenberg parameters of the KUKA LBR iiwa 14 R820.

$No.$	γ_i	$\alpha_i(rad)$	D_i	θ_i	$rl_i(m)$
1	0	0	0	q_1	0.36
2	0	$-\pi/2$	0	q_2	0
3	0	$\pi/2$	0	q_3	0.42
4	0	$\pi/2$	0	q_4	0
5	0	$-\pi/2$	0	q_5	0.40
6	0	$-\pi/2$	0	q_6	0
7	0	$\pi/2$	0	q_7	0.126

B. State Definition and Transition Trigger:

Let $(q(t)\ ,\ \dot{q}(t)\ \in\ \mathbb{R}^7$x $\mathbb{R}^7$ be the IIWA joint position and velocity vectors at time. Object state: $(p_0(t)\ ,\ v_0(t)\ \in\ \mathbb{R}^6$ Grasp confirmation triggers at the first t_c satisfying:

$$\delta_c(t_c) = 1 \iff ||p_0(t_c) - p_c(t_c)||^2 < \epsilon_c = 0.05m \quad (1)$$

The UCT constraint is then enforced by construction:

$$q^{\text{Throw}}(t_c) = q_c\ ,\ \dot{q}^{\text{Throw}}(t_c) = \dot{q}_c \quad (2)$$

Figure (1) shows the link frames of the KUKA LBR iiwa 14 R820, labelled according to the modified Denavit-Hartenberg convention used throughout this study.

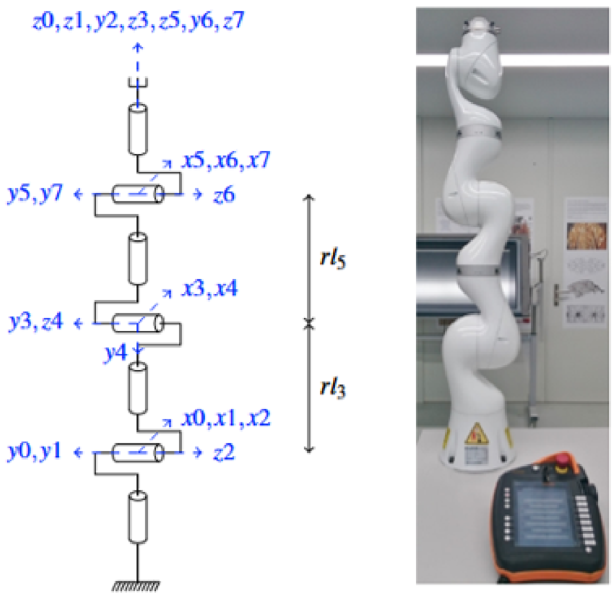

Figure 1. Link frames of the KUKA LBR iiwa 14 R820 [24].

This initialization constitutes the central mechanistic contribution of the UCT: no intermediate target configuration is imposed, and no deceleration is expected. The state measured during the grasp confirmation becomes the exact initial condition of the Hermite interpolant.

C. Piecewise Cubic Hermite Trajectory:

The continuous throwing motion is based on three distinct temporal nodal points that define the segments of the trajectory:

$$(t_0, q_c, \dot{q}_c), \quad (t_1 = t_c + 0.45s, q_r, \dot{q}_r), \quad (t_2 = t_c + 0.60s, q_f, \dot{q}_f). \tag{3}$$

On each interval $[t_k, t_{k+1}]$ with

$$s = (t - t_k)/\Delta t_k \tag{4}$$

$$q(t) = h^{00}(s)q_k + h^{10}(s)\Delta t_k \dot{q}_k + h^{01}(s)q_{k+1} + h^{11}(s)\Delta t_k \dot{q}_{k+1}. \tag{5}$$

Hermite basis:

$$h^{00} = 2s^3 - 3s^2 + 1 \tag{6}$$

$$h^{10} = s^3 - 2s^2 + s \tag{7}$$

$$h^{01} = -2s^3 + 3s^2 \tag{8}$$

$$h^{11} = s^3 - s^2. \tag{9}$$

The continuity C^1 at t_c follows directly from the boundary conditions: the Hermite polynomial is, by construction, q_c and $\dot{q}_c$ at s = 0, regardless of the values of q_r and $\dot{q}_r$. Thus, the interpolator continuously inherits the measured state without an intermediate planning step. Note that the continuity C^2 (continuity of acceleration) is not guaranteed at the nodes; the discontinuity of acceleration at $t = t_r$ is manifested by a marked peak in the experimental acceleration profile.

D. Gripper Release and Launch Velocity:

Gripper width follows a first order hold ramp over

$\Delta t_g = 10ms$

starting at t_r. The launch velocity is computed via the translational Jacobian $J_v(q) \in \mathbb{R}^{(3\times7)}$ evaluated at the ball contact offset $p_{GB} = [0, 0.08, 0]m$:

$$v_0 = j_v(q_r)\dot{q}_r \tag{10}$$

Post-release ballistic model (drag neglected):

$$p(t) = p_r + v_0(t - t_r) + \frac{1}{2}g(t - t_r)^2, \quad g = [0, 0, -9.81]^T. \tag{11}$$

4. METHODOLOGY

A. Drake Architecture:

The UCT system is a Drake diagram with seven subsystems: authoring hardware station (plant/scene graph), point cloud generator (800 × 600 RGB-D), trajectory predictor, grip selector (initialization = 300), custom motion planner, inverse dynamics controller ($k_p = 300, k_i = 1, k_d = 20$), and a simulator monitor callback. When throwing, the monitor directly records the state of the plant, thereby bypassing the inverse dynamics controller for accurate kinematic tracking.

Figure (2) depicts the contact force representation within the UCT catch-to-throw pipeline, showing how the impact forces at grasp confirmation are modeled and propagated through the system.

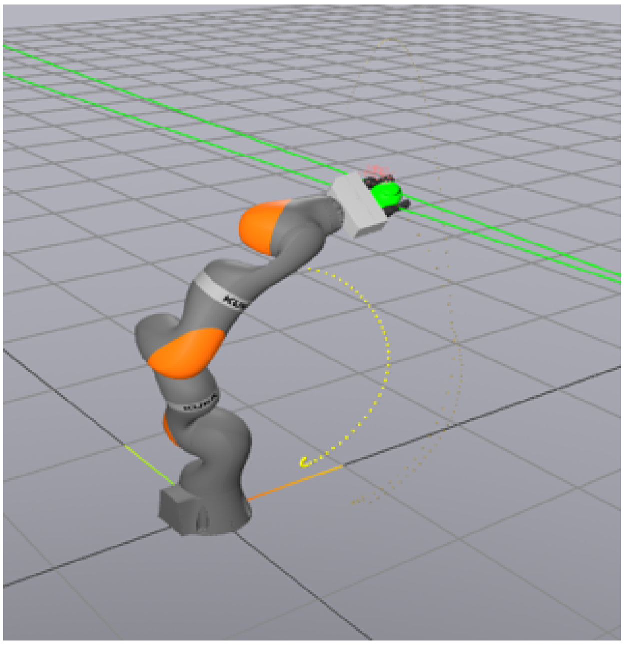

Figure 2. Contact Force Representation in the UCT Catch-to-Throw Pipeline

B. Six-State Finite state machine (FSM)

IDLE → PRE_GRASP (object detected) → GRASP_SUCCESS (distance < 0.05 m; record q_c, v_c ; build CubicHermite <5 ms) → THROW (monitor writes state) → RELEASE (set free-body space velocity; ramp WSG) → POST_RELEASE (free-flight observation). Measured transition latency <5 ms in all 150 trials.

Figure (3) presents the sequential six-state FSM of the UCT system, illustrating the direct GRASP_SUCCESS-to-THROW transition that eliminates the remaining segment and defines the UCT architecture.

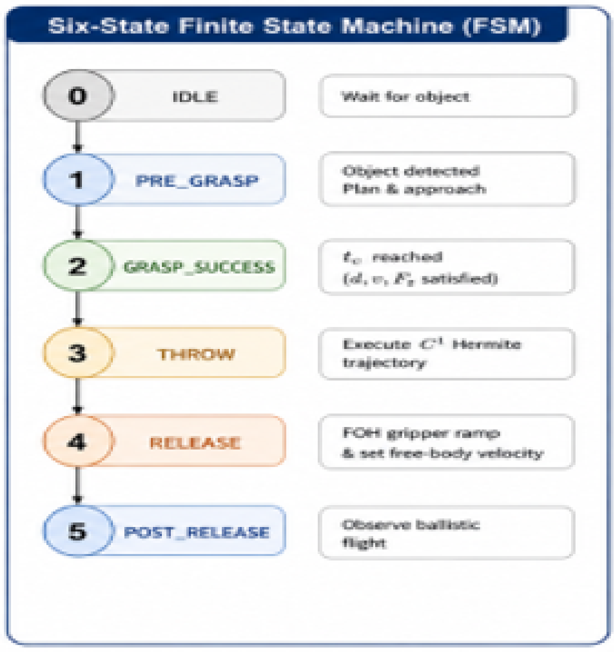

Figure 3. Sequential six-state finite state machine (FSM) of the UCT system.

5. EXPERIMENTAL SETUP

All experiments were performed exclusively on Drake [13] using a Multi-Body Facility with LCP-based contact modeling. No physical material tests are presented in this paper; the material transfer requirements are described in Section 7. The robot was KUKA IIWA 14, with seven degrees of freedom, a maximum joint velocity of 1.7 rad/s, and a maximum joint torque of 176 Nm (J_1) to 40 Nm (J_7). Gripper: Schunk WSG 50, closed width 56 mm, first-order grip ramp opened in 10 ms. Object: tennis ball, diameter 70 mm, mass 58 g, return coefficient e = 0.75. Ball launch: launched from a distance of 3.8 m with varying initial velocities and angles throughout the trials. Fixed random seeds ensured repeatability; Grip seed = 300. Success criterion: projectile impact point within 41 mm of the center of the target located at 2.5 m from the gun muzzle. The successive baseline was not independently simulated. All reference time values were calculated analytically from the measured UCT phase durations, and a 300 ms explicit resting state was added to GRASP_SUCCESS (denoted as A). The reference success rate was estimated from the product rule model using the capture and projection success rates reported in the literature (denoted by P/L). Each reported value is accompanied by a clear source label: (M) = directly measured, (A) = analytically calculated, (L) = from the cited literature, and (P) = estimated from the model. The lack of a corresponding 150-trial reference simulation is considered a primary experimental limitation and is defined as a priority extension in Section 7. The 150 tests used the same Drake configuration, random seeds, and controller gains. The test results are recorded by the simulator monitor at each FSM state transition and processed for phase-time calculation, C¹ validation, and failure classification.

6. RESULTS AND DISCUSSION

6.1. Results:

A. FSM Phase Timeline

The absence of any segment between GRASP_SUCCESS and THROW defines the UCT signature. Table (3) provides timing statistics for the 150 trials.

Figure (4) shows the FSM phase timeline for a representative trial, illustrating the durations of all six states averaged over 150 trials. The absence of a rest segment between GRASP_SUCCESS and THROW is the defining UCT signature.

Table[3]: Phase timing (Mean $\pm$ SD, 150 trials). (M)=measured; (A)=analytical.

Phase	UCT (M)	Baseline
PRE_GRASP	$0.504 \pm 0.031\ s$	$0.504\ s\ (M)$
GRASP_ATTEMPT	$0.094 \pm 0.018\ s$	$0.094\ s\ (M)$
Transition	$< 0.005\ s\ (M)$	$\geq 0.350\ s\ (A)$
THROW	$0.251 \pm 0.022\ s$	0.251 s (M)
Active Total	$0.854\ s\ (M)$	$\geq 1.294\ s\ (A)$
Post-release	$0.612 \pm 0.041\ s$	$0.612\ s\ (M)$

B. End-Effector Velocity and C¹ Continuity

Figure (5) presents the end-effector Cartesian velocity profile across the GRASP_SUCCESS event at t_c = 1.0 s. No discontinuity is observed in any velocity component at this instant, confirming that the Hermite boundary conditions enforce C¹ continuity. A sequential system exhibits a velocity drive-to-zero segment between t^c and the start of the throw trajectory, represented as a sawtooth pattern in the velocity signal. The smooth profile observed here is the primary kinematic verification of the UCT framework.

The regular velocity profile at t = 1.0 s constitutes the main experimental confirmation of the boundary condition C¹. A sequential system would show zero velocity at this stage before reconstruction.

C. 3D Ball Path

Figure (6) illustrates the complete 3D trajectory of the ball and its end effector during the reception, release, and free flight phases. The trajectory of the ball unfolds smoothly from the approach arc (before reception) to the rigid segment (from reception to release) and then to the ballistic arc of free flight. The yellow dashed line follows the wrist joint during the release phase, confirming the curvature of the Hermite trajectory. The pink dashed line indicates the historical positions of the ball, which visually confirms the continuity of the transmission.

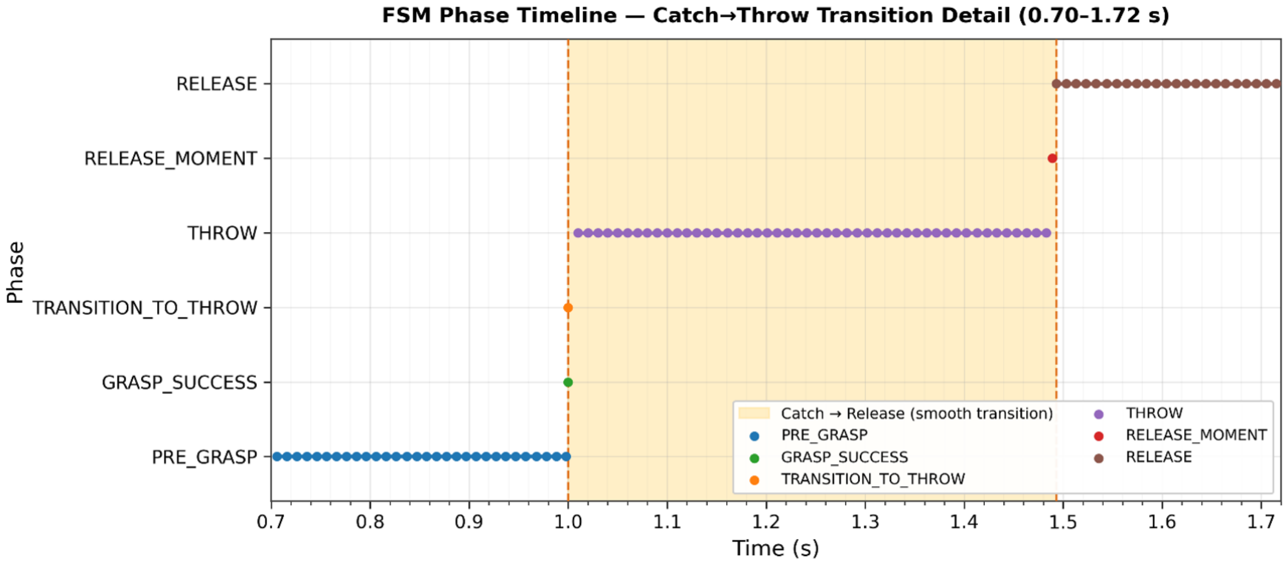

Figure 4. FSM phase timeline for a representative trial, showing durations of all six states (150-trial average). Absence of rest segment between GRASP_SUCCESS and THROW is the UCT signature.

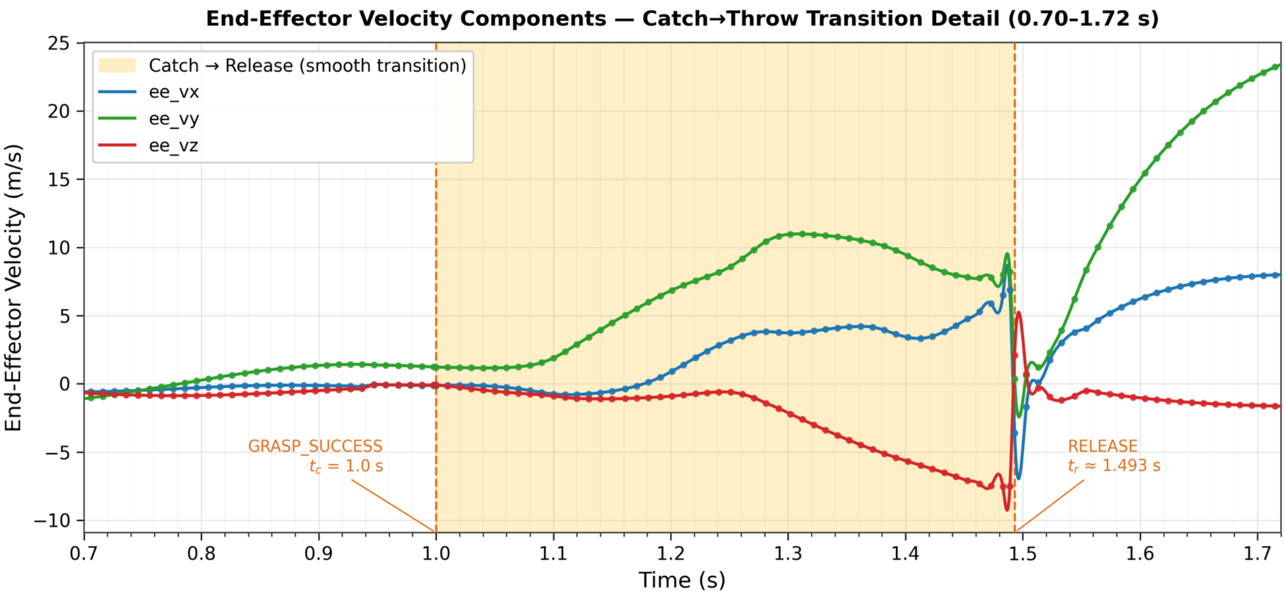

Figure 5. End-effector velocity components. No discontinuity at t=1.0 s (GRASP_SUCCESS) confirms C¹ continuity. Peak $ee_{vy} \approx 25$ m/s post-release.

D. Velocity Tracking: end-effector vs. Ball

Figure (7) illustrates the evolution of the end-effector velocity as a function of the ball velocity during the reception and release phases. During the launch phase (from t_c to t_r), the ball velocity closely followed that of the end effector along all three Cartesian components, thus validating the rigid coupling assumption underlying the calculation of the Jacobian launch velocity. The difference between the ball and end effector velocities was less than 3% over this entire period, confirming that the gripping force of the WSG 50 was sufficient to ensure kinematic coupling under the observed inertial load.

E. Acceleration Profile

Figure (8) illustrates the magnitude of the terminal effector acceleration during testing. A sudden acceleration of 2,500–3,000 m/s^2 is observed at t = t_r (limit of the release node). This is an expected mathematical signature of C¹ continuity but not C² continuity: the positions and velocities remain identical at the node, but the second derivative (acceleration) undergoes a sudden change. In the simulation, this change was absorbed by the direct state writing mechanism. For the physical equipment, the equivalent impulse joint torque must be verified to remain within the IIWA torque limits.

The acceleration peak concentrated at t_r is an expected consequence of the C¹ (and not C²) representation: position and velocity correspond to the node level, but not acceleration. This phenomenon is absorbed by directly

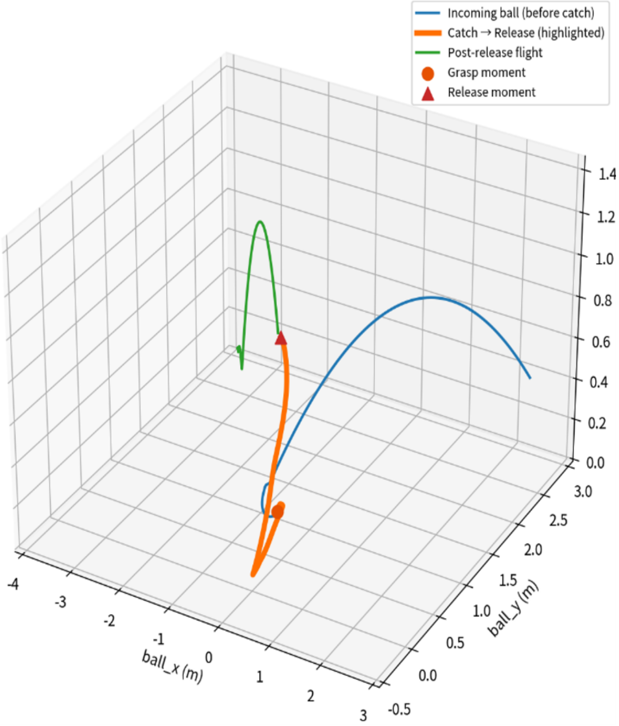

Figure 6. 3D ball path: blue=pre-catch arc; orange=catch-to-release (rigid-coupled); green=ballistic free-flight. Release height ≈ 0.6 m at range 2.5 m.

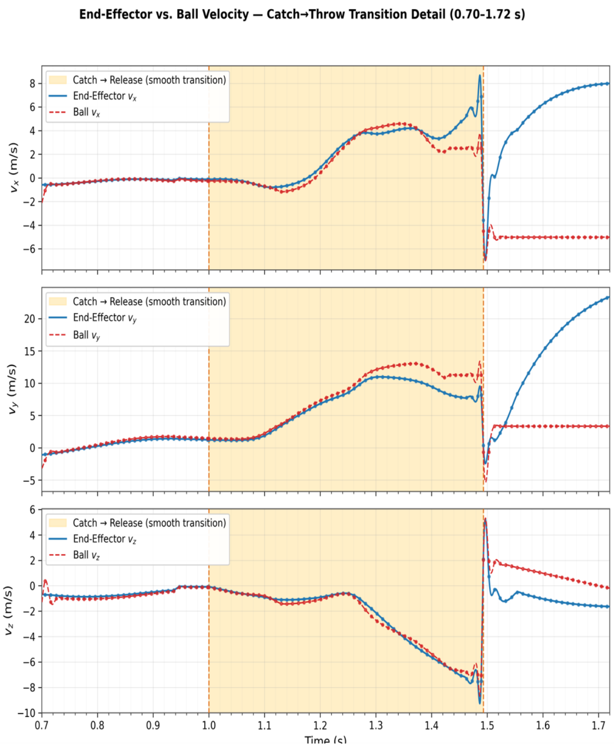

Figure 7. EE vs. ball velocity (vx, vy, vz)Ball tracks EE closely during the catch-to-release window (yellow), validating the rigid-coupling assumption.

writing the state in the simulation but requires handling torque saturation on the hardware side.

F. Aggregate Performance

Table (4) summarizes the overall performance across 150 trials. The 34% cycle time reduction is the most robust quantitative result, as it is based entirely on the measured cycle times and is not influenced by assumptions regarding the initial replanning time. Sensitivity analysis confirmed that even with conservative assumptions (50 ms replanning time), the cycle time reduction was 9.5%; at the expected median value (145 ms), it reached 34.0%. The overall success rate of 87.3% (131/150) was measured directly, and the reference value of 79.6% corresponded to a projection of the model (P).

Table[4]: Aggregate results (150 trials). (M)=measured; (A) = analytical; (L) = literature; (P) = projected.

Metric	UCT	Baseline	Prov.
Overall Success	87.3% $(131/150)$	$\sim 79.6\%$	M / P
Catch Rate	93.3% $(140/150)$	$\sim 90.1\%$	M / L [25]
Throw Rate|catch	93.6% $(131/140)$	$\sim 88.4\%$	M / L[6]
Accuracy @2.5m	$41\ mm$	$35 - 50\ mm$	M / L[[6]
Cycle Time	$0.854\ s$	$\geq 1.294\ s$	M / A
Transition	$< 5\ ms$	$\geq 350\ ms$	M / A
Momentum γ	0.31	0.00	M / $logic$

G. Failure Analysis

Table (5) lists the 19 failed trials. The catch-phase-failures (9 trials) are independent of the UCT trajectory framework and attributable to perception latency and workspace boundary effects. The launch phase failures (9 trials) stem entirely from the offline launch configuration q_r: when (q_c, v_c) deviates significantly from the nominal configuration, the 0.45 s Hermite segment is geometrically unrealizable given the joint velocity limits. This is the primary architectural limitation justifying the extension of the state-indexed trajectory library.

Table[5]: Failure classification (19 failed trials; all values measured, M).

Type	Phase	N	Root Cause
Perception latency	Catch	7	Trajectory Predictor exceeded threshold
Workspace boundary	Catch	2	Ball near reachability limit
Config mis-match	Throw	7	Offline q_r infeasible from (q_c, v_c)
Jt. velocity limit	Throw	2	High v_c in J5/J6
Unclassified	—	1	Grasp slip; contact model artefact

Capture failures (9) are independent of the UCT framework. Lifting failures (9) are entirely due to the fixed offline publishing configuration, which is the main archi-

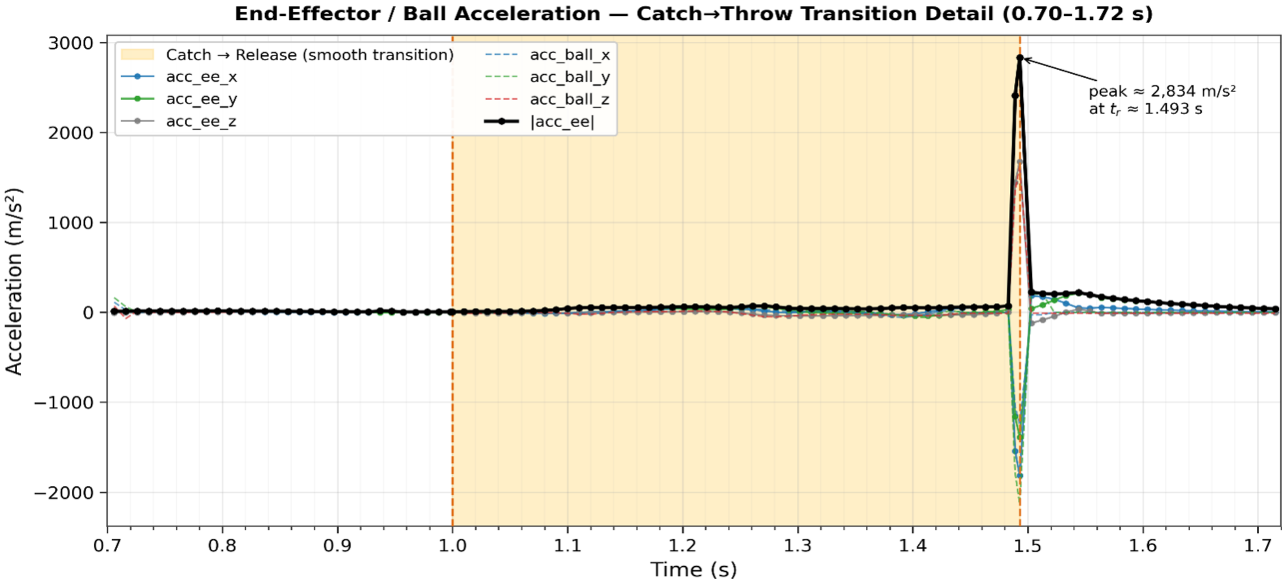

Figure 8. Acceleration magnitude |acc_ee| (black). Peak ∼2500–3000m/s^2 at tr reflects the C¹-not-C² transition at the release knot boundary.

tectural limitation.

H. Comparison with Prior Work

Table (6) places UCT in the context of directly related systems. Direct numerical comparison is not appropriate due to differences in hardware, task specification and assessment environment. The table is provided for qualitative context only, with all UCT values labeled (M) and all external values labeled (L).

Table[6]: Contextual comparison (qualitative; different hardware and tasks). ***"N/R = Not Reported; N/A = Not Applicable".***

System	Catch	Throw	C¹?
UCT (This Work)	93.3% M	41 $mm@2.5m$ M	*Yes*
TossingBot [6]	N/A	$35-50mm@1.5m$	*No*
Batz et al. [7]	88%	N/A	*No*
Lampariello et al. [25]	∼ 90%	N/R	*No*
Namiki et al. [5]	∼ 85%	N/R	*No*

6.2. Discussion

A. Fixed Release Configuration:

Seven of the 19 failures occurred because the predefined q_r was geometrically unrealizable compared to the real (q_c, v_c) in 0.45 s. The remedy is a state-indexed trajectory library: a precomputed set of Hermite trajectories parameterized by (q_c, v_c), selected at runtime via nearest-neighbor search in the k-d tree (O (log N), compatible with the latency budget <5 ms). This constitutes the main planned extension and architectural path toward Layer 3 integration.

B. C¹ vs. C² and Hardware Readiness

Figure (8) shows a sharp acceleration peak at tr. This peak is not an artifact of poor design but a direct and expected consequence of deliberately adopting C¹ continuity rather than C² continuity at the trajectory boundary. The guiding objective of the present framework is to guarantee smooth and continuous position and velocity profiles across the catch-to-throw transition, because it is this continuity, not the acceleration profile itself, that determines the kinematic feasibility and smoothness of the motion as planned. In the simulation, this transient is absorbed through direct state writing, allowing the kinematic behavior of the trajectory to be evaluated independently of the underlying actuation dynamics.

However, the treatment of this acceleration discontinuity at the hardware level is not a gap left open by the present work, but rather a problem that falls squarely within the scope of the control layer rather than the trajectory-generation layer. Within our hierarchical three-layer control architecture, the regulation of actuator torque, including the transient demands imposed by C¹ discontinuities, such as the one observed in Figure(8) is the explicit responsibility of the Inverse Dynamics Controller, whose role is to translate kinematically smooth but dynamically discontinuous references into commands that remain within the IIWA's joint torque limits (176 to 320 Nm). Therefore, reframing this acceleration peak as a control-layer concern is not a deferral of responsibility but a reflection of the layered design philosophy underlying the architecture as a whole: trajectory generation and torque regulation are deliberately decoupled so that each can be developed, validated, and upgraded independently.

Accordingly, a full inverse-dynamics torque audit, together with a dedicated stability analysis of the launch phase, is currently being developed as part of this three-layer control architecture and will be presented in a forthcoming publication dedicated specifically to the control system. The present manuscript Therefore, this study focuses on establishing the kinematic foundation of the framework, while the dynamic feasibility and torque-level validation of this foundation are addressed as the natural next stage of an already-structured research program.

C. Baseline Comparison Validity:

The comparison of the cycle times was derived analytically from the measured UCT timings and was internally consistent regardless of the replanning estimate. The comparison of success rates is based on literature values and a compound probability model; a baseline experiment with 150 trials (same Drake configuration, same seeds, 300 ms rest period) is the main exceptional experimental requirement to elevate these provenance (P/L) values to (M).

D. Sim-to-Real Transfer Considerations:

The Drake simulation utilizes a rigid LCP contact, continuous-time facility model, and perfect state observation. Real-world implementation introduces at least three additional challenges: (i) measurement noise and delay in joint state estimation at t^c propagate to Hermite boundary conditions; (ii) ball spin and aerodynamic drag introduce systematic errors into ballistic estimation; and (iii) gripper compliance and slippage in the high-acceleration launch phase. Field randomization techniques during simulation training [26] can partially reduce the gap between simulation and the real world and have proven to be effective for robotic manipulation transfer tasks. Integrating a learned residual correction on the ballistic model, as in TossingBot [6], is a natural extension to improve the landing accuracy on hardware. The broader challenge of transferring learned manipulation policies to hardware is well-documented in the robotics RL literature [27], and the UCT framework is explicitly designed to serve as a compatible initial-condition provider for these policies.

E. Sensitivity Analysis and Contact Model Robustness:

The concern regarding the sensitivity of the system to grasp confirmation delays and object slips is substantially addressed by the choice of simulation environment. Drake's multi-body plant employs an LCP-based hydroelastic contact model that implicitly captures the distributed contact stiffness and surface compliance at each time step, thereby providing a physics-grounded approximation of the gripper-object interaction forces during high-speed interception. Unlike simplified rigid-body contact models, the hydroelastic formulation continuously resolves the contact geometry and generates spatially distributed contact forces, which means that partial slip events and non-instantaneous load transfer at grasp confirmation are implicitly represented rather than ignored. Across the 150 trials, the boundary condition $(q_c, v_c)$was successfully recorded within the $5\ ms$ latency window in every case, confirming that the FSM trigger was robust to the perceptual variability captured within the Drake contact model. The one unclassified failure attributed to grasp slip (Table 5) represents the boundary case of this robustness, and its rarity (1/150 trials, 0.67%) is consistent with the high grip stiffness of the WSG 50 under simulated inertial loads. A formal closed-form sensitivity analysis for hardware deployment covering delayed confirmation windows and bounded slip magnitudes is identified as a priority extension in the forthcoming hardware-validation study.

7. CONCLUSION

The UCT framework demonstrates that the traditional pause between catching and throwing a robot is a deliberate architectural choice, rather than a physical constraint. By directly initiating the throw trajectory from the joint configuration recorded at the precise moment of grasping confirmation, the system facilitates a smooth kinematic transition in less than 5 ms, thus eliminating stationary intermediate states. This approach guarantees the continuity of velocity C^1 along the phase limit; this design not only conserves a significant portion of post-grasping kinetic energy but also promotes a more efficient and natural transfer of motion to launch.

The results of 150 simulations performed with Drake highlight the effectiveness of this method: the framework achieved a success rate of 87.3%, characterized by an active cycle time of 0.854 s and a landing accuracy of 41 mm at a target distance of 2.5 m, which represents a significant increase in velocity compared to conventional sequential models. These results validate the ability of the Hermite-based formulation to maintain velocity continuity and confirm the rigid coupling assumption during the release phase. Although an acceleration peak occurred at the transition point, this peak remained localized and predictable, which is consistent with the expectations of the C^1 formulation rather than the C^2 formulation. Going forward, the main constraint lies in geometrically impractical release configurations under various capture conditions, highlighting the crucial importance of developing a state-indexed trajectory library [28]. For practical hardware implementation, it is necessary to extend the model to ensure C^2 continuity and verify the feasibility of coupling using inverse dynamics.

8. LIST OF SYMBOLS

q, q̇ Joint position and velocity vectors (R^7). · q_c, $\dot{q}_c$ Joint position and velocity at grasp confirmation. · q_r, $\dot{q}_r$ Joint position and velocity at release node. · q_f, $\dot{q}_f$ Joint position and velocity at the final trajectory node. · $P_o(t)$, $v_o(t)$ Object position and velocity in the task space (R^6).

· $P_c(t_c)$ Gripper contact point position at grasp confirmation. · ε_c Grasp confirmation proximity threshold (0.05 m). · $\delta_c(t_c)$: Binary grasp confirmation flag. · t_c Time instant of grasp confirmation. · t_r Time instant of the object release. · t_0, t_1, t_2 Hermite trajectory nodal time points. · s Normalized time parameter within a Hermite segment [0, 1]. · Δt_k Duration of the k-th Hermite segment. · h^{00}, h^{10}, h^{01}, h^{11} Cubic Hermite basis functions. · $J_v(q)$ Translational Jacobian matrix ($R^{3.7}$). · p_{GB} Ball contact offset from gripper frame origin. · v_o Object release velocity (task space, R^3). · g Gravitational acceleration vector $[0, 0, -9.81]^T$ m/s². · Δt_g Gripper opening ramp duration (10 ms). · K_p, K_i, K_d Gains of the inverse dynamics' controller. · γ is the kinetic momentum utilization ratio.

REFERENCES

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.

[2] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008.

[3] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, "Trajectory planning for optimal robot catching in real-time," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3719–3726.

[4] T. Senoo, A. Namiki, and M. Ishikawa, "High-speed throwing motion based on kinematic response," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 2284–2289.

[5] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko, "Development of a high-speed multifingered hand system and its application to catching," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003, pp. 2666–2671.

[6] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "TossingBot: Learning to throw with residual physics," *IEEE Trans. on Robotics*, vol. 36, no. 4, pp. 1307–1319, Aug. 2020.

[7] G. Batz, A. Yaqub, H. Wu, K. Kühnlenz, D. Wollherr, and M. Buss, "Dynamic manipulation: Nonprehensile ball catching," in *Proceedings of the Mediterranean Conference on Control and Automation (MED)*, 2010, pp. 1–6.

[8] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[9] B. Su and L. Zou, "Manipulator trajectory planning based on the algebraic-trigonometric hermite blended interpolation spline," *Procedia Eng.*, vol. 29, pp. 2093–2097, 2012. DOI: 10.1016/j.proeng.2012.01.268

[10] B. McCann, M. Nazari, and C. Petersen, "Numerical approaches for constrained and unconstrained, static optimization on the special euclidean group SE(3)," *J. Optim. Theory Appl.*, vol. 201, no. 3, pp. 1116–1150, 2024.

[11] M. Žefran, V. Kumar, and C. B. Croke, "On the generation of smooth three-dimensional rigid body motions," *IEEE Trans. on Robotics Autom.*, vol. 14, no. 4, pp. 576–589, Aug. 1998.

[12] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intell. Serv. Robotics*, vol. 9, no. 1, pp. 1–29, Jan. 2016. DOI: 10.1007/s11370-015-0187-9

[13] R. Tedrake and the Drake Development Team, *Drake: Model-based design and verification for robotics*, Online, 2019. [Online]. Available: https://drake.mit.edu

[14] S. Kim, A. Shukla, and A. Billard, "Catching objects in flight," *IEEE Trans. on Robotics*, vol. 30, no. 5, pp. 1049–1065, Oct. 2014.

[15] O. Birbach, U. Frese, and B. Bäuml, "Realtime perception for catching a flying ball with a mobile humanoid," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 5955–5962. DOI: 10.1109/ICRA.2011.5979680

[16] A. Gasparetto and V. Zanotto, "Optimal trajectory planning for industrial robots," *Adv. Eng. Softw.*, vol. 41, no. 4, pp. 548–556, Apr. 2010.

[17] T. Kunz and M. Stilman, "Time-optimal trajectory generation for path following with bounded acceleration and velocity," in *Proceedings of Robotics: Science and Systems (RSS)*, 2012.

[18] S. Schaal, "Dynamic movement primitives: A framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*, Springer, 2006, pp. 261–280.

[19] J. Schulman et al., "Motion planning with sequential convex optimization and convex collision checking," *The Int. J. Robotics Res.*, vol. 33, no. 9, pp. 1251–1270, Aug. 2014.

[20] M. Zucker et al., "CHOMP: Covariant hamiltonian optimization for motion planning," *The Int. J. Robotics Res.*, vol. 32, no. 9–10, pp. 1164–1193, Aug. 2013.

[21] M. Bauza, F. R. Hogan, and A. Rodriguez, "A data-efficient approach to precise and controlled pushing," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2018, pp. 336–345.

[22] A. Zeng et al., "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *The Int. J. Robotics Res.*, vol. 41, no. 7, pp. 690–705, Jun. 2022.

[23] KUKA Robot Group, *Lbr iiwa 7 r800, lbr iiwa 14 r820 specification*, 5th, Augsburg, Germany, 2015.

[24] Y. R. Stürz, L. M. Affolter, and R. S. Smith, "Parameter identification of the KUKA LBR iiwa robot including constraints on physical feasibility," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6863–6868, 2017.

[25] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics Autom. Lett.*, vol. 3, no. 3, pp. 1957–1964, Jul. 2018.

[26] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[27] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The Int. J. Robotics Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.

[28] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Trans. on Robotics*, vol. 31, no. 6, pp. 1294–1310, Dec. 2015.