Articutool: Proactive Verification and Decoupled Control for Robust Robot-Assisted Feeding

Jose Jaime Martinez

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2025

Reading Committee:

Siddhartha S. Srinivasa, Chair

Blake Hannaford

Sam Burden

Program Authorized to Offer Degree:
Department of Electrical and Computer Engineering

University of Washington

Abstract

Articutool: Proactive Verification and Decoupled Control for Robust Robot-Assisted Feeding

Jose Jaime Martinez

Chair of the Supervisory Committee:
Siddhartha S. Srinivasa
Department of Computer Science & Engineering

For individuals with motor impairments, general-purpose assistive robots can offer increased independence. However, the practical utility of such systems can be undermined if they are unable to reliably handle common foods, leading to spillage that negatively impacts the user's dining experience. We propose that decoupling gross arm transport from fine-grained tool manipulation can enhance reliability. To this end, this paper introduces the Articutool: a modular, untethered, and locally intelligent 2-DOF wrist that a 6-DOF arm can temporarily equip to form a decoupled 8-DOF system. This decoupled approach separates the concerns of gross arm transport from fine-grained tool manipulation, empowering the arm's planner to find robust paths while the tool's onboard controller maintains utensil orientation. Our "plan-then-verify" control methodology proactively checks the arm's plans against the Articutool's kinematic and dynamic limitations to reduce the likelihood of spills before they happen. Our large-scale simulation benchmark, which isolates the challenging constrained-transport phase of feeding, demonstrates that this decoupled approach achieves a 96.0% transport planning success rate with a median planning time of 4.0 seconds. While monolithic baselines can achieve comparable success rates given sufficient computation time, they are orders of magnitude slower (median 75.7s for

8-DOF), rendering them impractical for real-time interaction. Physical experiments confirm these findings, showing that the system can successfully acquire challenging foods such as noodles and liquids, and achieves a 70.0% meaningful success rate (delivering a spill-free bite that meets an empirically-defined mass threshold) on the end-to-end feeding task, a task on which the baseline's meaningful success rate was only 10.0%. This work serves as a critical step toward an ecosystem of intelligent, task-specific tools for more capable, general-purpose assistive robots.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Restoring the ability to self-feed is a critical challenge in assistive robotics, promising to enhance the independence and quality of life for the millions of people who require feeding assistance. To address this, a significant portion of the research community has converged on using 6-DOF assistive arms, like the Kinova Jaco, as a capable, human-safe platform [2], [3], [4]. Researchers have developed and validated state-of-the-art systems on this hardware, such as the Assistive Dexterous Arm (ADA) [2], through multi-day, out-of-lab deployments that demonstrate the platform's viability for complex, real-world tasks.

Despite the success of these systems for many manipulation tasks, real-world studies have consistently highlighted a key limitation: the unreliability of spill-free bite acquisition and transport [1], [2], [5], [6], [7], [8], [9], [10]. In assistive feeding, spillage is not a minor inconvenience; it can negatively impact the user's dining experience and undermine confidence in the system's reliability [11]. As a participant in a user study powerfully articulated, "*If it can't get it on the first try, it's still on the plate, [the food's] not on me. If it drops it on the way that would be worse...*" [11], underscoring spillage as a primary design concern. This issue stems from a technical dilemma: to prevent spills, a 6-DOF arm must follow a path with strict, continuous orientation constraints, which severely restricts the planner's solution space and can lead to planning failures. While an 8-DOF manipulator might seem like a straightforward solution, this approach introduces significant practical and computational challenges. Beyond the added cost and bulk, the higher-dimensional configuration space imposes a severe computational penalty. As our results demonstrate, planning

constrained paths in this high-dimensional space is computationally expensive, often requiring planning times that far exceed the patience of a human user.

Our central thesis is that equipping a general-purpose 6-DOF arm with a modular, intelligent tool profoundly enhances its abilities. This modularity means that the tool can be used for a meal and then set aside afterward, freeing the arm for other tasks. This paper introduces the Articutool: a modular, untethered, and locally intelligent 2-DOF wrist that a 6-DOF arm can temporarily equip. This approach creates a crucial separation of concerns: the arm's planner focuses on finding a robust transport path using only loose orientation constraints, while the tool's onboard controller is dedicated to the fine-grained task of maintaining utensil orientation against gravity.

This decoupled control strategy trades the monolithic complexity of a fully synchronous 8-DOF system for the architectural simplicity and practical reliability of a decomposed problem. Rather than developing over-specialized, monolithic arms, we treat dexterous feeding as one of many daily tasks that benefits from a temporary, task-specific augmentation. This philosophy necessitates a truly modular, untethered appliance with self-contained computation, power, and sensing, echoing the long-standing vision of systems like the Handy 1 [12] to create an ecosystem of intelligent tools for general-purpose robots.

The contributions of this work are:

- The Articutool: a modular, untethered intelligent wrist that enhances a 6-DOF arm's reliability and dexterity when acquiring and transporting challenging foods.

- A "plan-then-verify" control architecture that proactively reduces spillage risk.

- A large-scale simulation benchmark and physical experiments that validate our decoupled approach, demonstrating superior transport planning, successful acquisition of difficult foods, and high end-to-end task success.

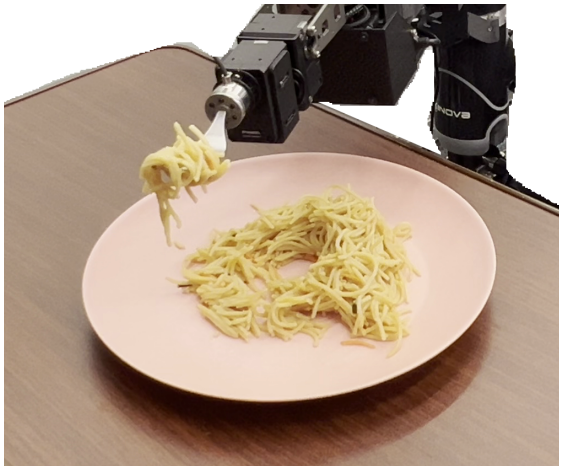
This leads to our core hypotheses:

- **(H1)** The Articutool's localized wrist actuation achieves more reliable and kinematically efficient acquisition of challenging foods (e.g., noodles, granular, liquid) than a 6-DOF arm equipped with a rigid tool.

- **(H2)** The decoupled control architecture significantly increases the success rate of spill-free transport by resolving the inherent conflict between path planning feasibility and strict orientation constraints faced by a 6-DOF baseline.

- **(H3)** The Articutool's decoupled architecture, by effectively decomposing the feeding task into robust solutions for acquisition (H1) and transport (H2), achieves a high overall end-to-end task success rate, measured by its ability to reliably deliver a meaningful, spill-free bite of food.

((a)) Dexterous Acquisition



((b)) Localized Leveling

((c)) Decoupled Transport

((d)) Successful Presentation

Figure 1.1: The Articutool performing a complete, successful feeding task with noodles, a challenging food item [1]. **(a) Dexterous Acquisition:** The 2-DOF wrist executes a localized twirl to secure the noodles. **(b) Localized Leveling:** The tool is leveled via a simple wrist motion before transport begins. **(c) Decoupled Transport:** The 6-DOF arm follows a smooth, efficient path while the Articutool's onboard controller actively maintains the tool's orientation to prevent spills. **(d) Presentation:** The bite is successfully delivered in a stable and accessible manner.

Chapter 2

RELATED WORK

Our work builds upon advancements in three core areas of robotics research: assistive feeding systems, intelligent end-effector design, and sensor-based control for manipulation.

2.1 Robotic Assistive Feeding Systems

The goal of restoring feeding independence has been a long-standing pursuit in assistive robotics, with landmark systems like the Handy 1 [12], the Assistive Dining Device [13], the Neater Eater [14], and Obi [15]. More recent research has focused on increasing autonomy, culminating in the state-of-the-art Assistive Dexterous Arm (ADA), the platform this work builds upon [2]. While the ADA system demonstrates sophisticated acquisition primitives, multi-day in-home deployments reveal a key limitation in the variety of foods it can successfully serve. A user notes that some of his favorite foods, like ramen, are "*difficult for it*", a sentiment echoed by his caregiver who states that the robot has "*too many limits with his diet and what he likes*" [2]. This challenge is rooted in a dual problem: the difficulty of acquiring entangle-able items like noodles, and the unreliability of transporting liquid components, like broth, using a standard rigid utensil.

While a 6-DOF arm can attempt a scooping motion for the solid components of such a meal, this strategy is itself often ineffective for entangle-able items like noodles, and its rigid utensil makes retaining liquids highly unreliable during transport. While theoretically possible, planning a path to keep a rigid-tool level imposes strict, continuous orientation constraints. These constraints restrict the valid configurations to

a lower-dimensional manifold in the C-space, a well-known challenge where standard sampling-based planners are known to fail [16].

While it is theoretically possible to plan a path that keeps a rigid tool perfectly level, doing so imposes such strict constraints that it dramatically increases the likelihood of planning failure in any non-trivial environment. This highlights the critical need for decoupled tool stabilization to handle soups, stews, and other liquid-based foods.

Even with ADA's sophisticated action primitives for single-utensil food acquisition [17], which include dexterous in-food twists, a key challenge remains: the reliability of the core robotic motions. Failures such as food falling off the tool or spillage during transport still occur [2]. Our work directly addresses this transport-phase spillage problem, not only for solid foods but also to improve the reliability of serving liquids and noodles.

2.2 Advanced End-Effector Design

The limitations of standard manipulators have led many researchers to develop specialized end-of-arm tooling. In feeding, this has included utensils with integrated sensors [18], [19] or cameras [1]. A few projects have explored actuated utensils. A notable existing technology is the handheld Liftware spoon [20], which uses active stabilization to assist individuals with hand tremors.

This highlights the value of active compensation, but its design addresses a different problem domain. Liftware's design cancels high-frequency, low-amplitude disturbance (tremors) through reactive stabilization. In contrast, we designed the Articutool to compensate for the low-frequency, high-amplitude trajectory of its robotic host. Furthermore, as a purely reactive stabilizer, Liftware lacks the communication interface to be commanded to specific non-level orientations required for scooping [20].

More recently, the FLAIR system [1] and the Kiri-Spoon [5], [6] have explored

actuated utensils controlled by the robot's central computer. The Articutool distinguishes itself from these approaches by being a fully untethered, self-powered, and locally intelligent module. This architectural choice is what enables the proactive verification and local control central to our methodology.

2.3 Decoupled and Sensor-Based Control

The concept of decoupling degrees of freedom to simplify control is a well-established paradigm in robotics [21]. Our work applies this philosophy to the manipulator itself. Our approach is a form of sensor-based compensation using an onboard IMU to maintain orientation relative to gravity.

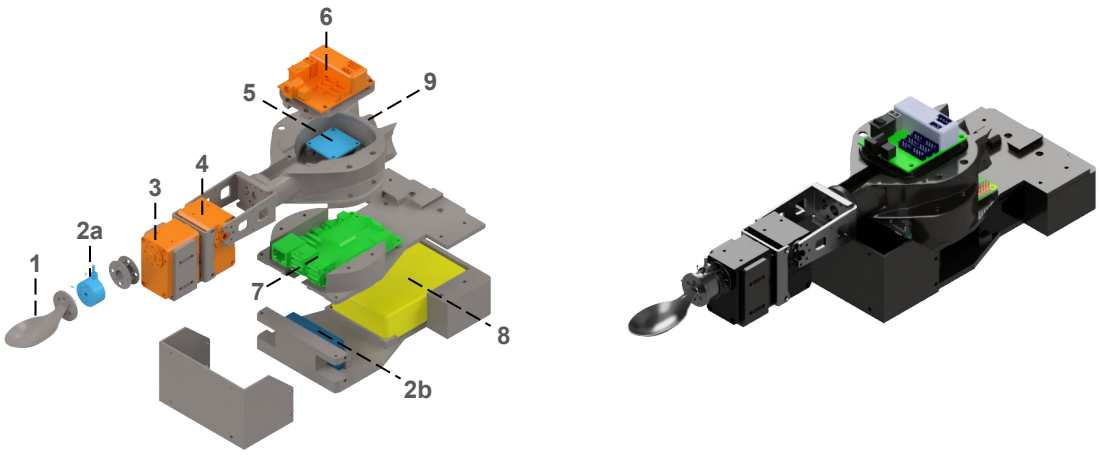
A key benefit is the localization of the orientation control loop. Unlike monolithic controllers, our architecture separates the 6-DOF arm's control from the Articutool's onboard orientation control. The system's role as a modular tool for a general-purpose robot motivates this architectural choice. It simplifies the planning problem for the central computer, which can treat the Articutool as an intelligent, task-space appliance that manages its own stability. This is particularly advantageous in real-world assistive settings. The untethered, modular design allows the primary robot to release the Articutool for storage or recharging, freeing the arm to perform other activities of daily living. This ability to treat the feeding tool as a temporary attachment is a key advantage over a permanently integrated 8-DOF system.

Chapter 3

SYSTEM DESIGN

Our platform is a distributed system founded on the principle of decoupling a primary manipulator from an intelligent, locally-controlled end-effector. Central to this approach is the philosophy of treating the Articutool not as a permanent fixture, but as a temporary, task-specific augmentation for a general-purpose robot. Assistive robotic arms enhance user independence and quality of life by assisting with a wide range of Activities of Daily Living (ADLs), from feeding a user to opening a door or retrieving a glass of water [22]. This modular approach allows the manipulator to equip the Articutool for a meal and then store it afterward, freeing the arm to perform other duties. Such a capability, however, requires the end-effector to be completely self-contained, as the robot must be able to equip or store it without the encumbrance of data tethers. This untethered design is a direct prerequisite for future work in autonomous tool changing.

To guide the Articutool's development, we established key performance requirements derived from the challenges of assistive feeding. Functionally, the tool requires 2-DOF actuation (Pitch/Roll) to perform both dexterous acquisition maneuvers and active leveling against gravity during transport. Architecturally, it needs to be a modular, untethered, and self-contained appliance with onboard computation, power, and sensing to enable its local, decoupled control loop. Its physical form factor has to be compact and graspable by a standard robotic gripper, while also incorporating features for user safety. Finally, we chose its performance targets to meet the requirements of human feeding motions. These targets were empirically derived by analyzing the maximum force and angular velocity specifications of the state-of-the-art ADA



((a)) Exploded View ((b)) Assembled View

Figure 3.1: The Articutool's modular hardware assembly. **Left: An exploded isometric view.** Key functional systems are color-coded: **Actuation (Orange)**, **Computation (Green)**, **Power (Yellow)**, and **Sensing (Blue)**. The numbered components are: **(1)** Utensil, **(2a)** Force/Torque Sensor, **(2b)** Force/Torque Electronics, **(3)** Roll Motor, **(4)** Pitch Motor, **(5)** IMU, **(6)** U2D2 Motor Controller, **(7)** Onboard Computation (Raspberry Pi 5), **(8)** Onboard Battery, and **(9)** Mounting Interface. **Right: An assembled isometric view.** The **(9) Mounting Interface** at the base allows the Articutool to attach to a robotic arm's end-effector.

action library. This library's primitives (e.g., in-food twists, high-force skewers) were developed directly from human-feeding datasets, which used instrumented forks to record the forces and motions of human subjects [18] and then clustered those motions into a discrete executable action scheme [17]. Our analysis of this library's most dynamic actions demanded a peak tool-tip force of 35N and angular velocities up to 2.3 rad/s. These requirements directly inform the selection of components and the overall hardware architecture described below.

3.1 Hardware Architecture

The physical system comprises two distinct modules: a 6-DOF Kinova Jaco 2 base manipulator and our custom Articutool wrist. We designed the Articutool as a self-

contained, 2-DOF (Pitch/Roll) wrist for complete operational independence. We designed its form factor so the Jaco arm's standard two-fingered gripper can securely hold it, allowing the arm to equip or release it with minimal effort. Its key innovation is the tight integration of all necessary components to eliminate the external power or data tethers that would make such modularity impossible. Its key modules include:

- **Onboard Computation:** We chose a Raspberry Pi 5 for its ability to run a full ROS 2 stack with native support for USB peripherals for simplified integration with the system's various sensors, all within a compact form factor.

- **Actuation:** We chose two Dynamixel XC430 motors to meet performance targets empirically derived from an analysis of velocity and force profiles observed in prior human and robot-assisted feeding studies [17], [18].

- **Integrated Sensing:** An OpenLog Artemis data logger with an integrated ICM-20948 IMU provides real-time orientation data for the leveling controller. For interaction forces, the system uses a distal Resense HEX21 6-Axis Force/-Torque sensor. We selected both components for their compact form factor and standard USB interfaces, which simplified integration with the onboard computer.

- **Power:** A dedicated onboard battery is a foundational design choice that enables the Articutool's core philosophy of modularity. Making the wrist self-powered eliminates the need for a physical tether, ensuring it is a truly decoupled appliance that can be equipped and used by any compatible arm, even a human's. To achieve this, a compact, commercial-off-the-shelf (COTS) 10,000 mAh USB-C power bank powers the system. A USB-C trigger module negotiates a 12V output from the battery to directly supply the two coreless DC motors, while a separate step-down converter provides 5V for the Raspberry Pi 5 and its connected peripherals.

To characterize the system's efficiency, we measured its total power draw from a 12V supply under typical operating conditions. The entire system consumes:

- 7.0W at idle (ROS 2 active, motors holding position)
- 8.2W during active transport (self-balancing)
- A conservative peak of 12.0W during simulated high-force acquisition.

Averaged over a representative feeding cycle, the total system power consumption is approximately 8.0W. The 10,000 mAh battery provides roughly 37 Wh of energy. Factoring in the battery's internal discharge efficiency, this empirically-grounded power profile yields an operational runtime of over 4 hours, sufficient for a full day's meals on a single charge.

This modular design is also extensible. While we designed the Articutool's current mounting interface for the Jaco arm's standard two-fingered gripper, the interface is modular, allowing easy adaptation for other manipulators to grasp it. Users can attach new utensils via a standardized mount, as shown in Fig. 3.2, allowing the system to adapt to different user needs or food types.

The key design insight of the Articutool lies in its hybrid hardware philosophy, combining high-performance commercial-off-the-shelf components (COTS) with simple, highly adaptable 3D-printed structural elements. For core functionality such as actuation, sensing, and computation, we selected proven COTS components. However, for the system's structure and its mounting interface, we designed CAD models that can be rapidly 3D-printed. These choices allow redesigning and fabricating the mounting interface so a different robotic manipulator can grasp it, or even adapting it for human use, making the system highly extensible. This strategy, which prioritizes architectural modularity and ease of replication, enabled us to develop a robust platform for the core software and control contributions of this work.

((a)) Fork (Noodles) ((b)) Spoon (Granulars) ((c)) Soup Spoon (Liquids)

Figure 3.2: The Articutool's interchangeable utensil end-effectors, allowing it to adapt to different food types. (a) The fork, used for acquiring entangle-able foods like noodles. (b) The standard spoon, used for granular foods like pecans. (c) The deep-welled soup spoon, used for acquiring liquids.

3.2 Software Architecture

We distributed the system's intelligence across two asynchronously communicating ROS 2 stacks [23], while orchestrating the feeding task using a Behavior Tree [24] to manage the sequence of high-level actions. This software architecture directly mirrors the modular hardware, creating a separation of concerns between the main control PC which handles high-level task and motion planning using MoveIt2 [25], and the Articutool's onboard processor which dedicates itself to real-time, low-latency tool orientation control.

The main control PC sends high-level commands to the Articutool over ROS 2; the onboard controller executes these commands autonomously. We separated the commands into two types. First, we use a ROS 2 service to set the tool's continuous control mode, such as engaging yaw-invariant leveling (`MODE_LEVELING`) for transport. Second, we use a ROS 2 action to execute discrete, parameterized motion primitives (e.g., `TWIRL` for noodles).

3.2.1 High-Level Task Orchestration

The main control PC runs a Behavior Tree that orchestrates the entire feeding task. This tree relies on a vocabulary of predefined robot configurations to structure the workspace, including `Home`, `AbovePlate`, `Resting`, and `Staging` configurations. To command the Articutool, the Behavior Tree uses a well-defined ROS 2 API, which exposes three distinct control modalities.

3.2.2 Articutool Control API

The Articutool's onboard controller exposes its functionality through a ROS 2 API, allowing the main PC to command it without needing to manage its low-level state.

Modal Control via Services

For persistent, stateful behaviors, the API provides the `SetOrientationControl` service. The Behavior Tree uses this service to engage or disengage continuous controllers, most notably the yaw-invariant leveling (`MODE_LEVELING`) required for spill-free transport. The service definition is as follows:

Listing 3.1: SetOrientationControl.srv Definition

```
# Request
int8 control_mode   # e.g., MODE_DISABLED, MODE_LEVELING
---
# Response
bool success
string message
```

Discrete Primitives via Actions

For discrete, parameterized motions, the API provides the `ExecuteArticutoolPrimitive` action. This is used for self-contained, open-loop behaviors like the `TWIRL` or `VIBRATE`

primitives. Using a ROS 2 action is ideal for these tasks as it provides continuous feedback on progress (e.g., percent complete) and a final result, allowing the Behavior Tree to monitor the execution without being dependent on a low-latency link.

Listing 3.2: ExecuteArticutoolPrimitive.action Definition

```
# Goal
string primitive_name # e.g., "TWIRL_CW", "VIBRATE_ROLL"
float32[] parameters  # e.g., rotations, speed, frequency
---
# Result
bool success
string message
float32[] final_joint_values
---
# Feedback
string feedback_string
float32 percent_complete
float32[] current_joint_values
```

Direct Joint Control via Trajectories

The third modality provides the main PC with direct control over the Articutool's joints. The main PC can command the Articutool's onboard controller manager to switch from its internal velocity controllers to a standard ROS 2 `joint_trajectory_controller`. This allows the main PC's motion planner to treat the Articutool as a simple 2-DOF appendage and send it a standard `JointTrajectory` message. This modality provides maximum flexibility and is used for motions that are planned centrally but only involve the Articutool's joints, such as the post-acquisition leveling motion.

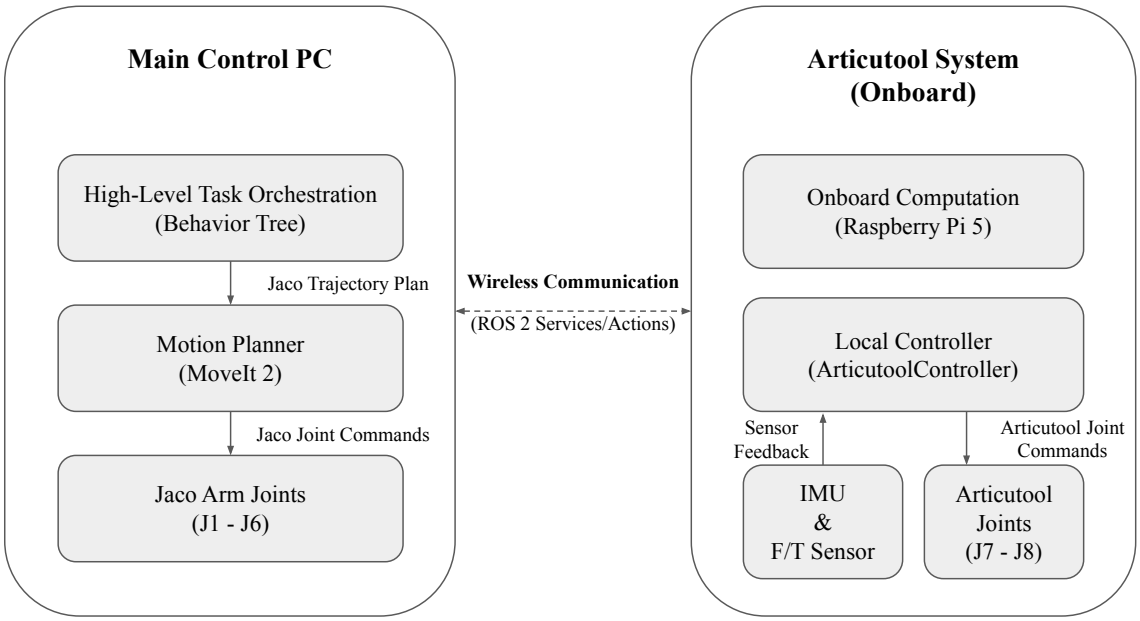


Figure 3.3: System Architecture Overview. Left: Main Control PC (Jaco Arm J1-J6) running MoveIt2 planning and a Behavior Tree. Right: Articutool System (Pitch J7, Roll J8) with an onboard Raspberry Pi 5, IMU, and battery, running the local `ArticutoolController` node. Wireless ROS 2 communication links them.

Chapter 4
CONTROL STRATEGIES AND KINEMATIC MODELING

The Articutool's ability to reliably transport food without spillage is rooted in a multi-layered control strategy. This chapter details the architecture of this strategy, which intelligently separates high-level planning from low-level, real-time execution.

Central to this entire architecture are the analytical kinematic models. We will first derive these models, as they are computationally essential for both the low-level, real-time execution and the high-level planning.

We will then examine their first application in the foundation of the system: the Articutool's onboard, IMU-based controller, which uses these models to achieve local stabilization. Finally, we will show how these same models are leveraged for our core contribution: a proactive, "plan-then-verify" methodology. This hierarchical approach allows the system to preemptively guarantee that a planned motion is both kinematically and dynamically feasible, forming the basis of its robustness.

4.1 Analytical Models for High-Level Verification

For the high-frequency demands of our verification methodology, we derived a dedicated analytical model of the Articutool's kinematics. These computationally efficient, closed-form solutions are essential for the high-level supervisor to perform the rapid feasibility checks required by our "plan-then-verify" approach.

This model is derived from a 2-DOF pitch-then-roll kinematic chain. We define the pitch joint (θ_p) as a rotation about the base frame's negative Y-axis, and the roll joint (θ_r) as a subsequent rotation about the new (pitched) frame's Z-axis. The "up" vector we control is defined as the Y-axis of the final tooltip frame ($\mathbf{y}_{\text{tool\_tip}} = [0, 1, 0]^T$).

The total rotation matrix $R_{0\leftarrow 2}$ from the tooltip to the base is $R_y(-\theta_p)R_z(\theta_r)$. The resulting "up" vector in the base frame, $\mathbf{y}_{F_0}$, is the second column of this matrix, $R_{0\leftarrow 2} \cdot \mathbf{y}_{\text{tool\_tip}}$.

The Forward Kinematics (FK) define the mapping from the tool's pitch and roll joint angles (θ_p, θ_r) to the orientation of its "up" vector, $\mathbf{y}_{F_0}$ (corresponding to the tool-tip's Y-axis), expressed in the tool's base frame F_0. Based on the kinematic model used in the controller, this vector is:

$$\mathbf{y}_{F_0} = \begin{bmatrix} \cos\theta_p \cos\theta r \\ \sin\theta_r \\ \sin\theta_p \cos\theta_r \end{bmatrix} \tag{4.1}$$

The Inverse Kinematics (IK) problem is then to find the joint angles required to align this vector with a desired world-frame direction (e.g., anti-gravity). Given a desired "up" vector $\mathbf{v} = (v_x, v_y, v_z)^T$ transformed into the tool's base frame, the IK solution is:

$$\theta_r = \arcsin(v_y) \tag{4.2}$$

$$\theta_p = \text{atan2}(v_z, v_x) \tag{4.3}$$

For real-time velocity control, we relate the Articutool's joint velocities, $\dot{\mathbf{q}}_{\text{atool}} = [\dot{\theta_p}, \dot{\theta_r}]^T$, to the linear velocity of the tool's leveling vector, $\dot{\mathbf{y}}_{F_0}$. The analytical Jacobian $\mathbf{J}_{\text{atool}}$ defines this relationship such that $\dot{\mathbf{y}}_{F_0} = \mathbf{J}_{\text{atool}}\dot{\mathbf{q}}_{\text{atool}}$. The columns of the Jacobian are the partial derivatives of $\mathbf{y}_{F_0}$ with respect to each joint angle:

$$\mathbf{J}_{atool} = \begin{bmatrix} \frac{\partial \mathbf{y}_{F_0}}{\partial \theta_p} & \frac{\partial \mathbf{y}_{F_0}}{\partial \theta_r} \end{bmatrix} = \begin{bmatrix} -\sin\theta_p \cos\theta_r & -\cos\theta_p \sin\theta_r \\ 0 & \cos\theta_r \\ \cos\theta_p \cos\theta_r & -\sin\theta_p \sin\theta_r \end{bmatrix} \tag{4.4}$$

The onboard `ArticutoolController` uses this model to nullify orientation errors detected by the IMU. The controller calculates a desired corrective angular velocity (ω_{corr}) from the orientation error, converts it to a desired linear velocity ($v_{corr} = \omega_{corr} \times \mathbf{y}_{F_0}$), and then uses the pseudo-inverse of this Jacobian ($\mathbf{J}^+_{atool}$) to solve for the joint velocities required to generate that correction: $\dot{q}_{atool} = \mathbf{J}^+_{atool} v_{corr}$ The high-level verifier (Section 4.3.2) uses this same model to ensure all planned trajectories are dynamically feasible for the controller.

4.2 Low-Level Regulation: IMU-Based Orientation Control

The `ArticutoolController` node, running on the onboard Raspberry Pi 5, enables the Articutool's ability to operate as an intelligent, decoupled appliance. This controller handles the tool's low-level regulation, translating high-level commands from the main PC into precise motor actions. Its primary function for this work is yaw-invariant leveling (`MODE_LEVELING`), the control mode essential for spill-free transport.

A fast, local PID control loop using the onboard IMU as the feedback sensor achieves this. We further enhance the controller's robustness by proactively damping commands near kinematic singularities and using a predictive model to smoothly enforce joint limits.

To estimate the tool's orientation, we use a complementary filter based on the approach described by Valenti et. al [26], which fuses raw accelerometer and gyroscope data. We deliberately configure the filter to ignore the magnetometer, a key design choice that makes the resulting orientation estimate purely relative to gravity. This ensures the leveling control is invariant to the arm's yaw rotation and immune to local magnetic disturbances, reinforcing the system's modularity. We also configure the

filter with adaptive gains and online gyroscope bias estimation to improve robustness during dynamic motions. The core `MODE_LEVELING` controller uses this gravity-aligned orientation directly, without needing any external calibration to the robot's frame.

4.2.1 Rationale for IMU Orientation Filter Selection

To achieve the stable, gravity-relative orientation required for the Articutool's leveling control, the system must fuse data from its onboard Inertial Measurement Unit (IMU). The primary challenge in orientation estimation is correcting the drift from the gyroscope, which integrates angular velocity to estimate orientation but accumulates error over time. This is typically corrected using accelerometer data, which provides a reliable, albeit noisy, measurement of the direction of gravity. The method chosen to fuse these sensor readings involves significant trade-offs between computational complexity, tuning effort, and estimation accuracy. Some of the more common approaches for this task include the Complementary Filter, the Madgwick Filter, and the Kalman Filter.

- **Kalman Filter:** The Kalman Filter is an optimal estimator in a statistical sense. It maintains a belief about the system's state (in this case, orientation) and uses a model of the system's dynamics to predict the next state. It then corrects this prediction based on incoming sensor measurements. While capable of providing highly accurate measurements, the Kalman Filter has two major drawbacks for this application. First, it is computationally expensive, typically requiring the propagation of a state covariance matrix and matrix inversion at each time step to compute the Kalman gain. Second, it requires careful and often difficult tuning of its noise covariance matrices which model the process and measurement noise. Incorrect tuning can lead to poor performance or filter divergence. Given the Articutool's reliance on a resource-constrained Raspberry Pi 5, and the absence of a complex dynamic model, the computational overhead

and tuning complexity of a Kalman filter was a less practical choice.

- **Madgwick Filter:** The Madgwick Filter [27] is a highly efficient, quaternion-based algorithm that also fuses gyroscope, accelerometer, and optionally magnetometer data. It uses a gradient descent method to compute the optimal orientation by minimizing the error between the measured and estimated gravity and magnetic field vectors. It can achieve excellent accuracy with low computational cost, and requires minimal tuning, typically involving a single parameter that determines how quickly the filter converges on the accelerometer/magnetometer solution.

- **Complementary Filter:** The Complementary Filter [26] is the simplest and most computationally efficient of the three. Its design is based on the insight that the gyroscope provides reliable orientation information in the short-term (high-frequency), while the accelerometer provides reliable information in the long-term (low-frequency). The filter combines a high-pass filtered gyroscope signal with a low-pass filtered accelerometer signal. This approach effectively "complements" the two data sources, using the accelerometer to correct the gyroscope's low-frequency drift. Its primary advantages are its extremely low computational cost and its intuitive tuning, which typically involves a single filter gain that balances the trust between the two sensors.

Given these trade-offs, the selection of the complementary filter from the ROS 2 package is a well-justified engineering decision for the Articutool. It provides a robust and sufficiently accurate orientation estimate without imposing a significant computational burden on the onboard processor, leaving ample resources for the primary control loop and communication tasks.

4.2.2 Complementary Filter Implementation

The implementation provided by the ROS 2 package is based on the work of Valenti et al. [26]. This method avoids the singularities of Euler angles (i.e. gimbal lock) and operates on a predict-correct cycle.

The filter first predicts a new orientation by integrating the angular rate from the gyroscope. This new orientation is then corrected using the accelerometer data. The algorithm calculates an error quaternion that represents the rotation required to align the gravity vector predicted by the filter with the gravity vector measured by the accelerometer. This error is scaled by a gain factor and applied to the predicted orientation, effectively nudging it toward the long-term, stable solution provided by the accelerometer. This predict-correct structure is computationally efficient and provides a robust orientation estimate suitable for real-time applications on embedded systems.

4.3 Hierarchical Task-Phase Control

To manage the complete feeding task, our system employs a hierarchical control strategy that separates non-critical preparatory motions from the critical, spill-sensitive transport phase.

4.3.1 Preparatory & Auxiliary Motions

For phases of the task where spillage is not a concern, the system uses a variety of simple, decoupled motions executed sequentially. These actions are straightforward and prioritize task setup over constrained movement. Examples include:

- **Arm-Only Motion:** Positioning the Jaco arm's wrist-mounted camera over the plate (`AbovePlate` configuration) for food detection.

- **Sequential Arm-Tool Motion:** First, executing a 6-DOF path for the arm,

then executing a 2-DOF local motion with the Articutool to achieve the desired scooping or skewering pose before entering the food.

- **Tool-Only Motion:** Performing a local post-acquisition leveling maneuver, where only the Articutool's pitch motor actuates to bring a newly acquired bite to a stable, horizontal orientation.

These maneuvers are the necessary building blocks for the feeding task, but the core challenge of spillage is addressed by our primary control strategy for transport.

4.3.2 Constrained Transport: The "Plan-Then-Verify" Approach

The core of our contribution is a proactive verification methodology designed for spill-free transport. It resolves the conflict between path planning feasibility and strict orientation constraints by decomposing the problem: we break a single, difficult 8-DOF constrained planning problem into a simple 6-DOF gross transport problem for the arm and a 2D local stabilization problem for the tool. The arm's planner uses only a loose orientation constraint (heuristically set to $\pm 45°$ for roll and $\pm 90°$ for pitch) to find a robust path, while the Articutool's local `MODE_LEVELING` controller performs the fine-grained stabilization.

Crucially, this "plan-then-verify" method checks that a planned trajectory is kinematically and dynamically feasible for the Articutool before execution begins. This contrasts with purely reactive systems, which can only attempt to correct for errors as they occur. Our system checks every waypoint in a proposed time-parameterized trajectory to ensure that the configuration required to achieve an anti-gravity orientation of the tool-tip is within the Articutool's joint limits, and that the motors are fast enough to produce the required corrective velocity. These feasibility checks serve to mitigate the risk of a spill before it can happen.

First, we calculate the disturbance angular velocity, $\boldsymbol{\omega}_{\text{disturbance}}$, created by the Jaco arm's motion ($\dot{\mathbf{q}}_{\text{jaco}}$) in the world frame. This requires the Jaco's Jacobian,

$\mathbf{J}_{\text{jaco}}$, which we compute at each waypoint using the Pinocchio robotics library [28]. This numerical approach is necessary for a complex 6-DOF manipulator and stands in contrast to the lightweight analytical Jacobian ($\mathbf{J}_{\text{atool}}$) we derived in Section 4.1 for the Articutool's 2-DOF kinematics. We must transform this disturbance into the Articutool's local base frame using the rotation matrix $\mathbf{R}_{\text{world}}^{\text{atool\_base}}$. The required corrective angular velocity in the Articutool's local frame is:

$$\boldsymbol{\omega}_{\text{correction}}^{\text{local}} = -(\mathbf{R}_{\text{world}}^{\text{atool\_base}})^T \boldsymbol{\omega}_{\text{disturbance}} \tag{4.5}$$

Next, we must convert this required angular correction into the linear velocity of the tool's "up" vector ($\mathbf{y}_{F_0}$) that our analytical Jacobian models. We compute this target linear velocity, $\boldsymbol{v}_{\text{correction}}^{\text{local}}$, using the cross product, where $\mathbf{y}_{F_0}$ is the tool's "up" vector expressed in the tool's base frame:

$$\boldsymbol{v}_{\text{correction}}^{\text{local}} = \boldsymbol{\omega}_{\text{correction}}^{\text{local}} \times \mathbf{y}_{F_0} \tag{4.6}$$

We can then use the analytical Jacobian's pseudo-inverse, $\mathbf{J}_{\text{atool}}^{+}$, to solve for the Articutool joint velocities, $\dot{\mathbf{q}}_{\text{atool}}$, needed to generate this correction:

$$\dot{\mathbf{q}}_{\text{atool}}^{\text{required}} = \mathbf{J}_{\text{atool}}^{+} \boldsymbol{v}_{\text{correction}}^{\text{local}} \tag{4.7}$$

The waypoint is dynamically feasible if and only if the required joint velocities are within the motors' maximum speed, $\dot{q}_{\max}$: $||\dot{\mathbf{q}}_{\text{atool}}^{\text{required}}||_\infty \leq \dot{q}_{\max}$.

4.4 Localized Primitives for Proactive Spill Prevention

Beyond the continuous leveling used during transport, our system leverages the Articutool's localized 2-DOF actuation to execute a vocabulary of food-specific motion primitives. We designed these primitives not as novel motions in themselves, but as targeted, proactive strategies to mitigate spillage before the main transport phase

begins. This contrasts directly with a simple scooping motion, which acquires an unstable bite and creates an unmitigated spillage risk before transport even begins.

These primitives are implemented as a library of parameterized, stateful actions that run on the Articutool's onboard controller. The behavior tree can command these primitives by name and pass a list of parameters to define their behavior. The core primitives used for acquisition and proactive food shedding are:

- **The Twirl Primitive:** This primitive (`TWIRL`) is parameterized by `[target_rotations, speed_rad_per_sec]`. It executes a continuous velocity command on the roll joint to achieve a specified number of full rotations at a set speed.

- **The Vibrate Primitive:** This primitive (`VIBRATE`) is parameterized by `[frequency_hz, amplitude_rad, duration_sec]`. It applies a high-frequency sinusoidal velocity command to the roll joint, causing the tool to oscillate at a given frequency and amplitude for a set duration.

- **The Tilt Primitive:** This primitive (`TILT`) is parameterized by `[target_roll_angle, velocity_rad_per_sec, hold_duration_sec]`. It smoothly rotates the tool to the target roll angle at a set velocity, holds that angle for a specified duration to allow the liquid to pour out, and then returns to a level orientation.

- **The Noodle-Shed Primitive:** This primitive (`NOODLE_SHED`) is parameterized by `[pitch_up_angle, pitch_speed, delay, vibrate_frequency, vibrate_amplitude, vibrate_duration]`. This complex primitive is designed to first congregate noodles by pitching the tool up at a set speed. It then holds this position for a specified delay, allowing the noodles to settle. After the delay, it applies a high-frequency sinusoidal velocity command to the roll joint to shed loose, unstable noodles. Finally, it returns the pitch joint to its original level orientation.

The execution of these primitives is orchestrated by the high-level behavior tree, which defines three distinct stages for primitive execution:

1. **Pre-Move-Into:** Executed after the arm arrives at the "above-food" pose but before it begins the insertion motion into the food.

2. **Post-Move-Into:** Executed after the tool has completed its servo-based grasp motion but before it retracts from the food.

3. **Post-Acquisition:** Executed after the local leveling maneuver is complete and the tool has been retracted to center itself over the plate or bowl.

This framework allows the system to be highly extensible. A new, complex acquisition strategy can be defined simply by populating these slots in an external action library, with no changes to the underlying control logic. The strategies for our target foods demonstrate this flexibility:

- **Strategy for Granulars (Tilted Scoop):** This strategy uses the `VIBRATE` primitive in the Post-Acquisition stage. After scooping, leveling and retracting, this primitive applies a short, high-frequency roll oscillation. This action settles the contents of the spoon and dislodges any unstable pieces, which fall harmlessly back into the bowl. This effectively sheds the food that would have otherwise spilled during transport, ensuring only a stable load is carried to the user.

- **Strategy for Liquids (Tilted Dip):** This strategy uses the `TILT` primitive in the Post-Acquisition stage. After dipping the soup spoon into the liquid, leveling and retracting, this primitive applies a gradual tilt to pour out excess liquid and achieve a consistent remaining mass of liquid, then tilts back to a level configuration. This effectively sheds the food that would have otherwise spilled during transport, ensuring only a stable load is carried to the user.

- **Strategy for Noodles (Skewer + Twirl):** This strategy uses a sequence of primitives across the three execution stages to create a stable bite.
  - The sequence begins with a clockwise twirl before insertion.
  - Once the tool is inside the noodles, the sequence continues with a counter-clockwise twirl to entangle and secure the noodles.
  - After leveling the tool, the system executes the `NOODLE_SHED` primitive. This complex primitive first pitches the fork up, pauses to let the noodles settle, and then vibrates the roll joint to dislodge any highly unstable noodles, which fall harmlessly back onto the plate. This action encourages only a tight, stable core of noodles to remain before transport.

4.5 Onboard Communication and Real-Time Considerations

A central challenge for the Articutool's decoupled architecture is managing two vastly different communication profiles. The system must handle:

1. **High-Level, Asynchronous Commands:** Non-time-critical instructions from the main PC over a high-latency wireless link (e.g. `SetOrientationControl`, `ExecuteArticutoolPrimitive`).

2. **Real-Time, Synchronous Feedback:** A high-frequency (50 Hz) stream of sensor data (IMU, joint states) needed for the local leveling controller to remain stable.

Attempting to run the 50 Hz control loop (Algorithm 2) by sending sensor data over the wireless link to the main PC for processing and back would be impossible; unpredictable network latency would lead to catastrophic instability.

The Articutool's architecture solves this by design. As shown in Fig. 3.3, all components relevant to the real-time control loop, including the motor drivers, the

IMU filter, and the `ArticutoolController` node itself, run locally on the onboard Raspberry PI 5. By design, ROS 2 automatically optimizes communication between nodes running on the same host machine, avoiding the physical Wi-Fi network entirely.

This local communication is achieved through two primary mechanisms:

- **Shared Memory Transport:** For nodes on the same machine, the default ROS 2 middleware (DDS) is configured to use a shared memory transport. This is a "zero-copy" inter-process communication method where a publisher writes data directly to a segment of memory that a subscriber can read from. This bypasses the computer's network stack entirely, providing extremely high-bandwidth and low-latency communication ideal for real-time control.

- **Loopback Interface:** In cases where shared memory is not used, the operating system's network stack still prevents local traffic from going over the physical network. When a node publishes data destined for another node on the same host, the traffic is routed internally through the loopback interface (`localhost`). While this involves the overhead of the network stack, it never engages the physical Wi-Fi or Ethernet hardware.

This architectural choice is a physical manifestation of the system's decoupled philosophy. It isolates the high-frequency, time-critical control loop from the high-latency, asynchronous task-level commands, ensuring a minimal and predictable latency that is critical for the stability and performance of the 50 Hz controller.

Algorithm 1 Articutool Leveling Control Loop

function ComputeLevelingCommand(dt)

▷ — *1. Get Current State & Target* —

$\mathcal{R}_{world \to imu} \leftarrow$ GetImuOrientation()

$q_{curr} \leftarrow$ GetCurrentJoints()

$y_{up\_target} \leftarrow [0, 0, 1]^T$ ▷ World Z-up vector (anti-gravity)

$y_{up\_tool} \leftarrow$ GetToolUpVector(q_{curr}) ▷ Effective "up" vector in tool frame

▷ — *2. Calculate Orientation Error in World Frame* —

$\mathcal{R}_{world \to tool\_base} \leftarrow$ GetCurrentToolBaseOrientation($\mathcal{R}_{world \to imu}$)

$y_{up\_world} \leftarrow \mathcal{R}_{world \to tool\_base} \cdot y_{up\_tool}$

$\omega_{error} \leftarrow$ CrossProduct($y_{up\_world}, y_{up\_target}$)

$\theta_{error} \leftarrow$ ArcCos(DotProduct($y_{up\_world}, y_{up\_target}$))

$\omega_{error\_vec} \leftarrow \omega_{error} \cdot \theta_{error}$

▷ — *3. PID Control to get Corrective Angular Velocity* —

$I_{err} \leftarrow I_{err} + \omega_{error\_vec} \cdot dt$

$I_{err} \leftarrow$ Clamp($I_{err}, -I_{max}, I_{max}$)

$D_{err} \leftarrow (\omega_{error\_vec} - \omega_{error\_last})/dt$

$\omega_{correction\_world} \leftarrow K_p \cdot \omega_{error\_vec} + K_i \cdot I_{err} + K_d \cdot D_{err}$

$\omega_{error\_last} \leftarrow \omega_{error\_vec}$

▷ — *4. Inverse Velocity Kinematics* —

$\omega_{correction\_local} \leftarrow \mathcal{R}^{-1}_{world \to tool\_base} \cdot \omega_{correction\_world}$

$J_{atool} \leftarrow$ ComputeArticutoolJacobian(q_{curr})

$v_{correction\_local} \leftarrow$ CrossProduct($\omega_{correction\_local}, y_{up\_tool}$) ▷ Convert angular to linear vel

$\dot{q}_{required} \leftarrow J^{+}_{atool} \cdot v_{correction\_local}$ ▷ J^{+} is pseudoinverse

▷ — *5. Damp Command Near Singularities* —

$damp\_factor \leftarrow$ ComputeSingularityDampFactor(q_{curr})

$\dot{q}_{required}$[pitch] $\leftarrow \dot{q}_{required}$[pitch] $\cdot\, damp\_factor$

return $\dot{q}_{required}$

end function

Algorithm 2 Articutool Dynamic Feasibility Verification

function VERIFYJACOTRAJECTORY($T_{jaco}, M_{jaco}, M_{atool}$)
  $q_{atool\_prev} \leftarrow$ null
  for each waypoint i in T_{jaco} **do**
    $q_{jaco}, v_{jaco} \leftarrow$ GetStateAtWaypoint(T_{jaco}, i)
    $T_{world \to atool\_base} \leftarrow$ ForwardKinematics(M_{jaco}, q_{jaco})
                                ▷ — *1. Kinematic Feasibility Check* —
    $y_{up\_target} \leftarrow T^{-1}_{world \to atool\_base} \cdot [0, 0, 1]^T$
    $Q_{solutions} \leftarrow$ SolveArticutoolIK(y_{up_target})
    $Q_{valid} \leftarrow$ FilterByJointLimits($Q_{solutions}, M_{atool}$)
    if Q_{valid} is empty **then**
      return false ▷ Kinematically infeasible
    end if
    $q_{atool} \leftarrow$ SelectClosestSolution($Q_{valid}, q_{atool\_prev}$)
    $q_{atool\_prev} \leftarrow q_{atool}$
                                ▷ — *2. Dynamic Feasibility Check* —
    $J_{jaco} \leftarrow$ ComputeJacoJacobian(M_{jaco}, q_{jaco})
    $\omega_{disturbance} \leftarrow (J_{jaco} \cdot v_{jaco})_{\text{angular}}$
    $\omega_{correction} \leftarrow -T^{-1}_{world \to atool\_base} \cdot \omega_{disturbance}$
    $J_{atool} \leftarrow$ ComputeArticutoolJacobian(M_{atool}, q_{atool})
    $v_{correction} \leftarrow$ CrossProduct($\omega_{correction}, y_{up\_target}$) ▷ Convert angular to linear vel
    $\dot{q}_{required} \leftarrow J^{+}_{atool} \cdot v_{correction}$ ▷ J^+ is pseudoinverse
    if $||\dot{q}_{required}||_\infty > M_{atool}$.max_velocity **then**
      return false ▷ Dynamically infeasible
    end if
  end for
  return true ▷ Trajectory is feasible
end function

Chapter 5

EXPERIMENTAL DESIGN

We designed a three-part evaluation to validate our system's core hypotheses regarding enhanced acquisition (H1), spillage reduction during transport (H2), and end-to-end success (H3). This structure used physical experiments to validate real-world performance and a large-scale simulation to provide robust, quantitative evidence for our core algorithmic contribution to transport planning. For the simulation benchmark (Exp. 2a), we used the `RRT-Connect` non-optimizing planner [29] to provide a clear measure of planning feasibility and time-to-first-solution.

To reflect the practical demands of a user-facing assistive system, we conducted all physical experiments using the `RRT*` optimizing planner [30] with a strict 15-second planning timeout. We empirically derived this timeout from our simulation results as a realistic upper bound on the Articutool's planning time (see Fig. **??**) and serves as a practical limit for maintaining a responsive interaction during a meal.

To establish a fair and robust baseline for physical trials, we first conducted a preliminary simulation study to characterize the 6-DOF system's performance. This study quantified the trade-off between planning reliability and constraint tightness. As shown in Fig. 5.1, a strict $\pm 10^\circ$ orientation constraint, while ideal for preventing spills, resulted in frequent planning timeouts, making it impractical for a responsive user-facing system. Conversely, looser constraints increased planning success but failed to represent a good-faith effort at preventing spills. We therefore selected a $\pm 20^\circ$ tolerance as a time-comparable baseline for all physical experiments, as it provided a high planning success rate while remaining reasonably constrained.

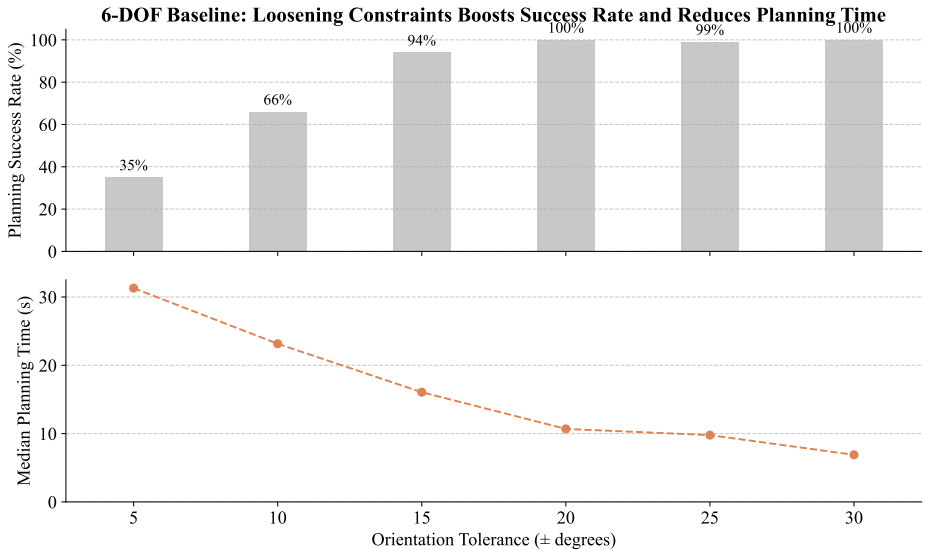


Figure 5.1: Justification for the 6-DOF baseline's orientation tolerance. This plot shows the trade-off between planning success rate and planning time as orientation constraints are loosened. A strict ±10° constraint results in a low success rate due to timeouts. A ±20° constraint achieves a high success rate with a significantly lower planning time, making it a fair and responsive baseline for physical experiments. We therefore use ±10° as a strict benchmark for the simulation (Exp. 2a) and ±20° for all physical trials (Exp. 2b, Exp. 3).

5.1 Experiment 1: Physical Acquisition Efficiency (H1)

This experiment tested the hypothesis that the Articutool's local dexterity enabled superior acquisition of challenging foods.

- **Task:** We ran 20 trials for each system and food type combination (120 total). Before each trial, we placed a constant, pre-measured mass of food (125 g of noodles, 250 g of water, or 125 g of pecans) in a source bowl. The robot started from a fixed `AbovePlate` configuration and executed its acquisition strategy, then executed a level-constrained motion to a fixed `Resting` configuration. After the maneuver, an experimenter measured the mass of food remaining on the utensil.

- **Comparison:** We compared the mass of food acquired by the Articutool's decoupled actions against the baseline's synchronous 6-DOF actions.

- **Metric:** A statistically significant higher mass acquired by the Articutool validated H1.

5.2 Experiment 2: Transport Reliability (H2)

To validate our core hypothesis regarding transport, we used a two-pronged approach combining large-scale simulation and physical tests.

5.2.1 Experiment 2a: Simulated Transport Reliability Benchmark

We designed this experiment to isolate the long-range, constrained transport problem and test the core hypothesis (H2) by requiring the planner to find a long path while adhering to strict orientation constraints that prioritize safety.

- **Task:** We first generated a large candidate set of 100 (start, goal) transport tasks with level orientation constraints. We then created a fair, common benchmark set by filtering for tasks that we confirmed (via an IK check) to be kinematically reachable by all three systems (6-DOF, 8-DOF, and Articutool). We then benchmarked each system on this common set of pre-validated tasks.

- **Comparison:** We compared the Articutool's decoupled approach (loose arm path constraints with local tool leveling) against monolithic 6-DOF and 8-DOF baselines, which both adhered to a strict, continuous orientation constraint of $\pm 10^{\circ}$ in pitch and roll throughout the entire trajectory.

- **Metric:** The primary metrics were the planning success rate and the distribution of planning times (time-to-solution). To rigorously distinguish between

kinematic infeasibility and computational complexity, we set an extended planning timeout of 1000 seconds. This allows us to characterize the planning distribution and determine if baselines fail because a solution does not exist, or simply because finding it is computationally expensive. A significantly lower median planning time for the Articutool system validated H2 in simulation.

5.2.2 Experiment 2b: Physical Transport Reliability

This experiment tested the hypothesis that the Articutool's decoupled, active leveling reduced spillage during transport when compared to the 6-DOF baseline.

- **Task:** We executed transport motions from a common start configuration to five diverse `Staging` goal configurations, shown in Fig. 5.2. For each system and food type, we ran 20 trials (80 total), distributing them evenly across the five goal configurations (4 trials each). For each trial, we placed a pre-defined mass of food on the utensil. We chose these payloads (2.5 g of pecans and 8.0 g of water) to represent generous but realistic acquisitions, exceeding the average masses observed in Exp. 1, thereby creating a challenging test case for evaluating transport reliability. We chose pecans and water as representative granular and liquid foods whose mass can be precisely controlled, allowing for a repeatable measurement of spillage. We defined these target `Staging` configurations as goal joint configurations for the Jaco arm, which the system then planned and executed a transport motion to.

- **Comparison:** We compared the spillage from the Articutool's active leveling strategy against the baseline's synchronous, constrained transport motion, with both systems targeting the same final arm configuration.

- **Metric:** A statistically significant lower average spillage compared to the 6-DOF baseline across all trials for the Articutool validated H2.

5.2.3 Experiment 2c: Proactive Spill-Mitigation Primitives

This experiment was designed to isolate and validate the effectiveness of the proactive shedding primitives described in Section 4.4, which are a key component of our transport reliability strategy. We designed this ablation study to test our two challenging, controllable food types: granular (pecans) and liquids (water). For pecans, this tests the `VIBRATE` primitive, and for liquids, this tests the `TILT` primitive.

We omitted noodles due to the high variability of the grasp, which makes it difficult to create a standardized starting condition for a controlled ablation study. The combined noodle strategy (acquisition + shedding) is therefore evaluated as a complete system in the end-to-end trials of Exp. 3.

- **Task:** To create a realistic "stress test," we first conducted a preliminary study to determine the "maximum capacity" load for each utensil. For this, an experimenter used the standard spoon and deep-welled soup spoon to manually perform multiple scooping trials with pecans and water, respectively. The goal of each trial was to acquire the maximum possible mass. The average mass from these trials was then used as the standardized "maximum capacity" load for this experiment. For each trial of Exp. 2c, the appropriate utensil was manually loaded with this exact, empirically-determined "stress test" mass (7 g for pecans, and 20 g for water). The arm then executed a single, standardized transport motion from a fixed `Resting` configuration (over the bowl) to a fixed `Staging` configuration. After the transport motion was complete, an experimenter measured the mass of food remaining on the utensil.

- **Comparison:** We compared two conditions for each food type (80 total trials):

  - **Control Groups (N=20 pecans, N=20 water):** The arm executed the standardized transport motion immediately after being loaded.

  - **Test Groups (N=20 pecans, N=20 water):** The arm first executed its respective shedding primitive while over the bowl (the `VIBRATE` primitive for pecans, the `TILT` primitive for water) and then executed the transport motion.

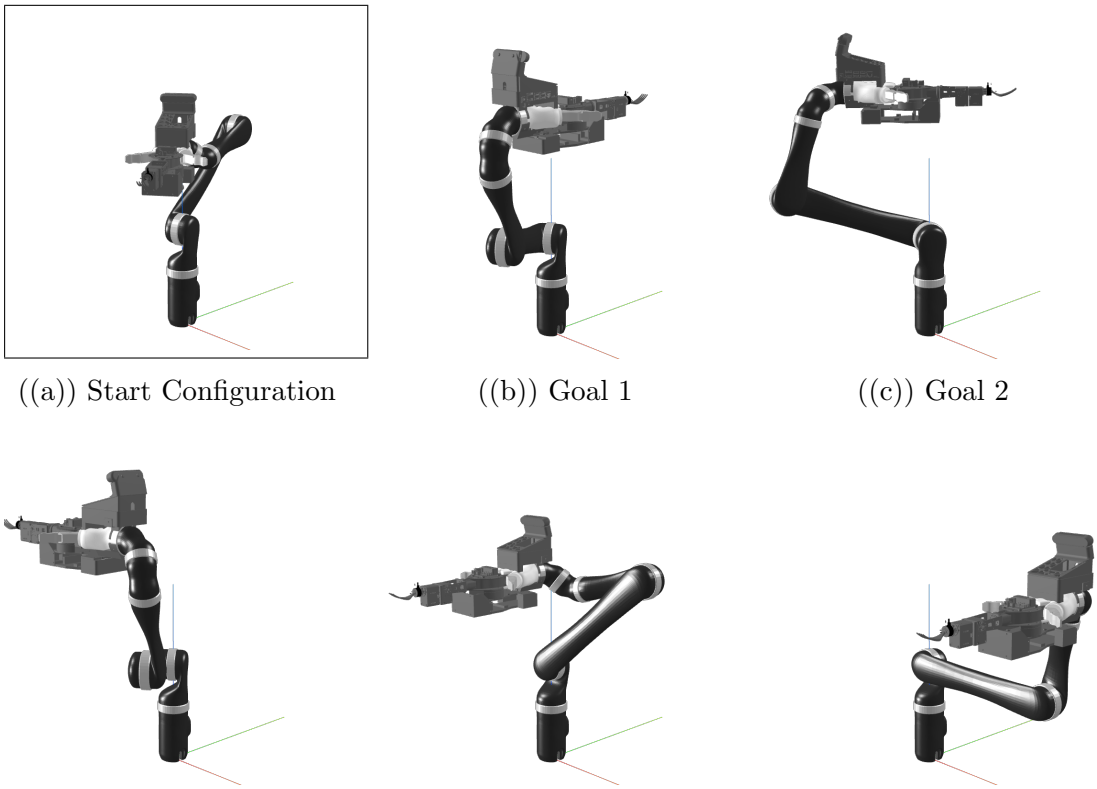
- **Metric:** To account for the failure mode of a primitive shedding too much food, the primary metric was a hierarchy of outcomes. A trial was categorized into one of the following mutually exclusive outcomes:

  - **Meaningful Success:** The trial was completed with zero transport spills, and the mass of food delivered was greater than the average acquisition mass from Exp. 1 (i.e., $> 1.83\ g$ for pecans and $> 5.02\ g$ for water).
  - **Trivial Success (Primitive Failure):** The trial was completed with zero transport spills, but the mass of food delivered was less than or equal to the average acquisition mass ($\leq 1.83\ g$ for pecans and $\leq 5.02\ g$ for pecans). This outcome explicitly captures the failure of a poorly-tuned or naive shedding primitive.
  - **Transport Failure (Spill):** Any amount of food was spilled during the transport phase, regardless of the final mass delivered.

  A statistically significant increase in the meaningful success rate for the test groups validated that our primitives effectively mitigate spill risk without compromising the primary goal of delivering a satisfying bite (H2).

5.3 Experiment 3: Physical End-to-End Task Success (H3)

To validate the overall architectural approach (H3), we evaluated the system's ability to perform the entire feeding task. We selected noodles for this end-to-end evaluation as they represent the most difficult food category, requiring the full, integrated



((a)) Start Configuration ((b)) Goal 1 ((c)) Goal 2

((d)) Goal 3 ((e)) Goal 4 ((f)) Goal 5

Figure 5.2: The common start configuration (a) and five diverse `Staging` goal configurations (b-f) used for the physical transport reliability experiment (Exp. 2b). The diversity of the goals tests the planner's ability to find reliable, spill-free paths across different parts of the workspace.

execution of the system's dexterous acquisition strategy and its decoupled, spill-free transport capabilities to validate H3.

- **Task:** The system attempted the full kinematic sequence for feeding: acquiring noodles (the most challenging food item), and transporting it to a `Staging` configuration. We ran 20 trials for each system (40 total). This task combines the challenges of acquisition (H1) and transport (H2).

- **Comparison:** We compared the complete Articutool system's performance

against the 6-DOF baseline attempting the same full, end-to-end task.

- **Metric:** To evaluate the overall system performance (H3), we defined a hierarchy of outcomes based on their real-world impact on the user. A trial was categorized into one of the following mutually exclusive outcomes:
  - **Meaningful Success:** The trial was completed with zero critical or non-critical spills, and a technically successful bite of food ($> 3.0\ g$) was delivered. The 3.0 g threshold for a technically successful bite was not set arbitrarily. It was empirically defined from the baseline's acquisition data collected during this end-to-end experiment (Exp. 3). That dataset showed a clear bimodal distribution, revealing two distinct technical outcomes: failed acquisitions and successful acquisitions. The 3.0 g threshold was chosen as the objective statistical boundary to separate these two outcomes, thus defining meaningful success as a trial that was both technically successful in acquisition and spill-free in transport.
  - **Trivial Success:** The trial was completed with zero critical or non-critical spills, but a trivial ($\leq 3.0\ g$) amount of food was delivered.
  - **Non-Critical Failure (Table Spill):** Any amount of food was spilled on the table, but not on the user.
  - **Critical Failure (User Spill):** Any amount of food was spilled on the user, their wheelchair, or the floor.

We consider meaningful success as the primary metric for validating H3. Spillage that was intentionally shed back into the bowl via localized primitives (Section 4.4) was not counted as a failure.

In addition to these quantitative metrics, all trials were video recorded to qualitatively assess the magnitude and nature of any spillage, to distinguish between minor spills and catastrophic failures.

Chapter 6

RESULTS

Our experiments validate the core hypotheses that a decoupled architecture improves acquisition dexterity (H1), transport reliability (H2), and overall end-to-end task success (H3).

6.1 *Localized Wrist Actuation Enables Efficient Acquisition of Challenging Foods (H1)*

The Articutool's localized dexterity demonstrated a clear advantage in acquiring a variety of foods (Exp. 1). For granular foods, the Articutool acquired 1.83 $g \pm 0.61$ g of pecans, a statistically significant improvement ($p = 0.0062$) over the baseline's 1.32 $g \pm 0.48$ g.

The benefits were most pronounced for more challenging foods. The Articutool successfully acquired an average of 24.80 $g \pm 9.56$ g of noodles and 5.02 $g \pm 1.62$ g of water. In contrast, the baseline's rigid scooping motion proved less effective for these food types, acquiring an average of 10.93 $g \pm 7.01$ g of noodles and 0.68 $g \pm 0.18$ g of water. For both noodles and water, the Articutool showed a statistically significant improvement of ($p = 0.0$) and ($p = 0.0$) respectively. These results, shown in Fig. 6.1, confirm that the Articutool's localized actuation provides a significant advantage, validating H1.

6.2 *Decoupled Control Achieves Superior Transport Reliability (H2)*

We present a two-part validation for H2, using a large-scale simulation (Exp. 2a) to demonstrate planning robustness and physical experiments (Exp. 2b, Exp. 2c) to

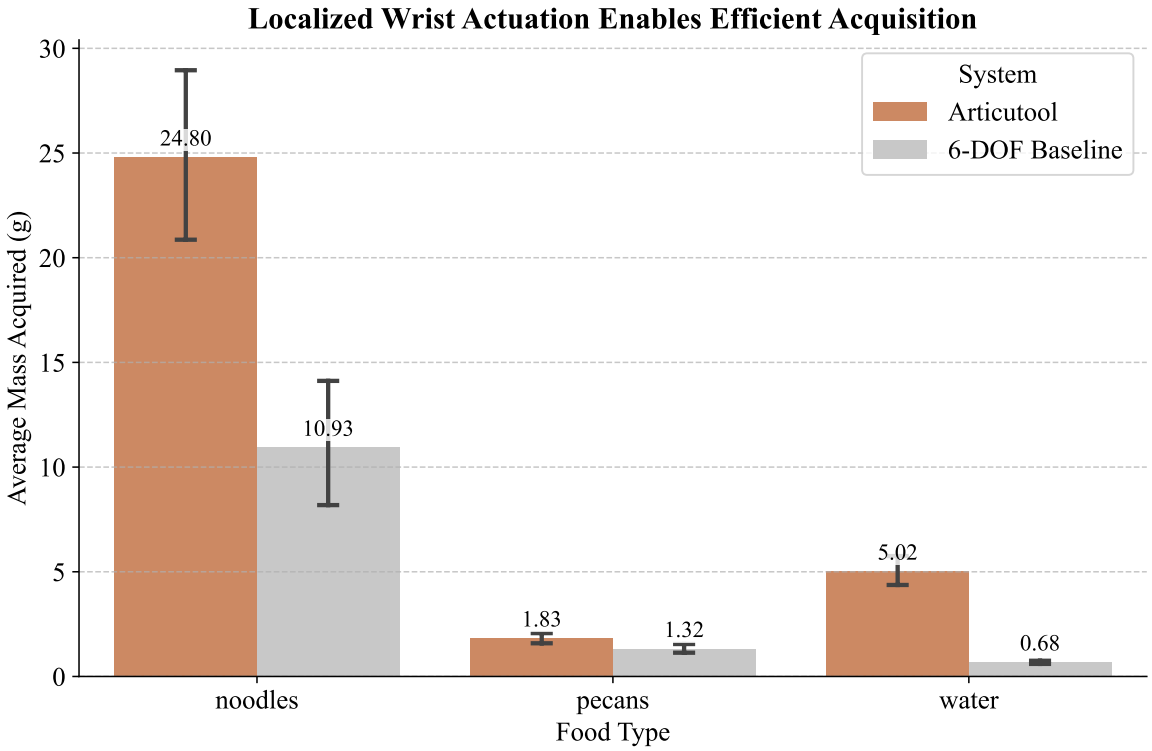


Figure 6.1: The Articutool acquires significantly more mass across all food types. The baseline proved ineffective for noodles and water. This demonstrates improved acquisition performance, validating H1. Error bars show 95% CI.

confirm real-world spillage reduction.

The simulation benchmark reveals that decoupled planning is not only robust but uniquely efficient. As shown in Fig. 6.2, the Articutool (Orange) solves 96.0% of transport tasks with a median planning time of just 4.02 seconds.

In contrast, the performance of the baselines highlights the high computational cost of monolithic planning. With the extended 1000-second timeout, the 8-DOF baseline (Dark Gray) achieved a 95.0% success rate, confirming that valid paths do exist in the higher-dimensional space. However, finding them is computationally expensive: the median planning time was 75.7 seconds, approximately 19 times slower than the Articutool.

The 6-DOF baseline (Light Gray), constrained by the "narrow passage" problem of strict orientation limits, required a median of 31.0 seconds and failed to find a solution in 14.0% of cases even with the extended timeout.

These results validate that while an 8-DOF system is kinematically capable of solving the task, the decoupled architecture is necessary to make that solution computationally efficient for real-time user interaction.

Physical experiments confirm reduced spillage. In physical trials (Exp. 2b), the Articutool's active leveling virtually eliminated spillage. When transporting 8.0 g of water, it spilled an average of 0.00 g, a statistically significant reduction ($p = 0.0030$) compared to the baseline's average of 1.31 g. This confirms that our decoupled approach is more reliable for spill-free transport in practice, as shown in Fig. 6.2.

Proactive primitives demonstrably reduce spill frequency without sacrificing bite quantity. The ablation study in Exp. 2c isolated the contribution of our proactive shedding primitives and confirmed they are a key component of transport reliability (H2). When transporting the unstable, "maximum capacity" loads, the control groups (without primitives) were highly prone to failure, spilling on 100.0% of pecan trials and 100.0% of water trials.

In stark contrast, the test groups, which executed a shedding primitive over the bowl before transport, showed a dramatic and statistically significant reduction in spillage. For the pecan test group, 90.0% of trials resulted in meaningful success, with only 5.0% resulting in a transport failure. Similarly, the water test group achieved a 100.0% meaningful success rate. Crucially, the test groups saw a trivial success rate of 5.0% for pecans and 0.0% for water, confirming our primitives are well-tuned and do not "cheat" by shedding an excessive amount of food.

This success stems from the primitives' consistency. By design, the `VIBRATE` and `TILT` primitives reliably shed the unstable portion of the load. The high rate of meaningful success in this experiment confirms that these primitives are well-tuned:

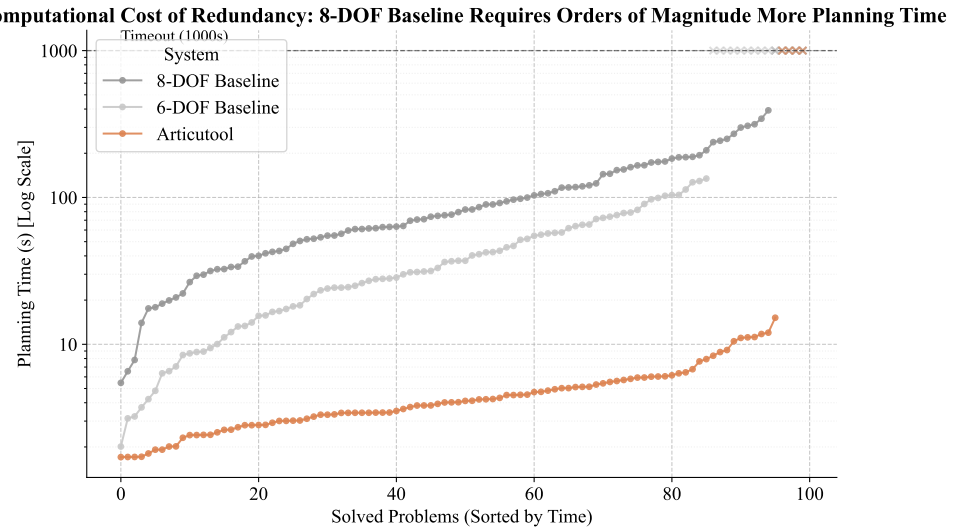


Figure 6.2: Planning time profiles (Cactus Plot) for constrained transport (Exp. 2a). The x-axis shows the number of solved problems sorted by time, and the y-axis (log scale) shows the time required. While the 8-DOF baseline (Gray) eventually solves 95.0% of problems with the extended 1000s timeout, it requires orders of magnitude more time (Median: 75.7s) than the Articutool (Orange, Median: 4.0s) due to the complexity of the high-dimensional search space. The Articutool achieves high success rates with planning times suitable for real-time interaction, validating H2.

they effectively mitigate spill risk without being overly aggressive. The primitives consistently left a stable mass of food that as greater than the average autonomous acquisition mass (1.83 g for pecans and 5.02 g for water) identified in Exp. 1. This result validates that these localized primitives are a critical, reliable component of our system's success, successfully reducing spill risk.

6.3 The Decoupled Architecture Achieves High End-to-End Task Success (H3)

To validate the overall architectural approach (H3), we evaluated the system on the complete, physical feeding task in Exp. 3. Based on the strict binary metric (zero spillage), the Articutool system achieved a 70.0% success rate (14/20 trials). In stark contrast, the 6-DOF baseline achieved only a 35.0% success rate (7/20 trials). This

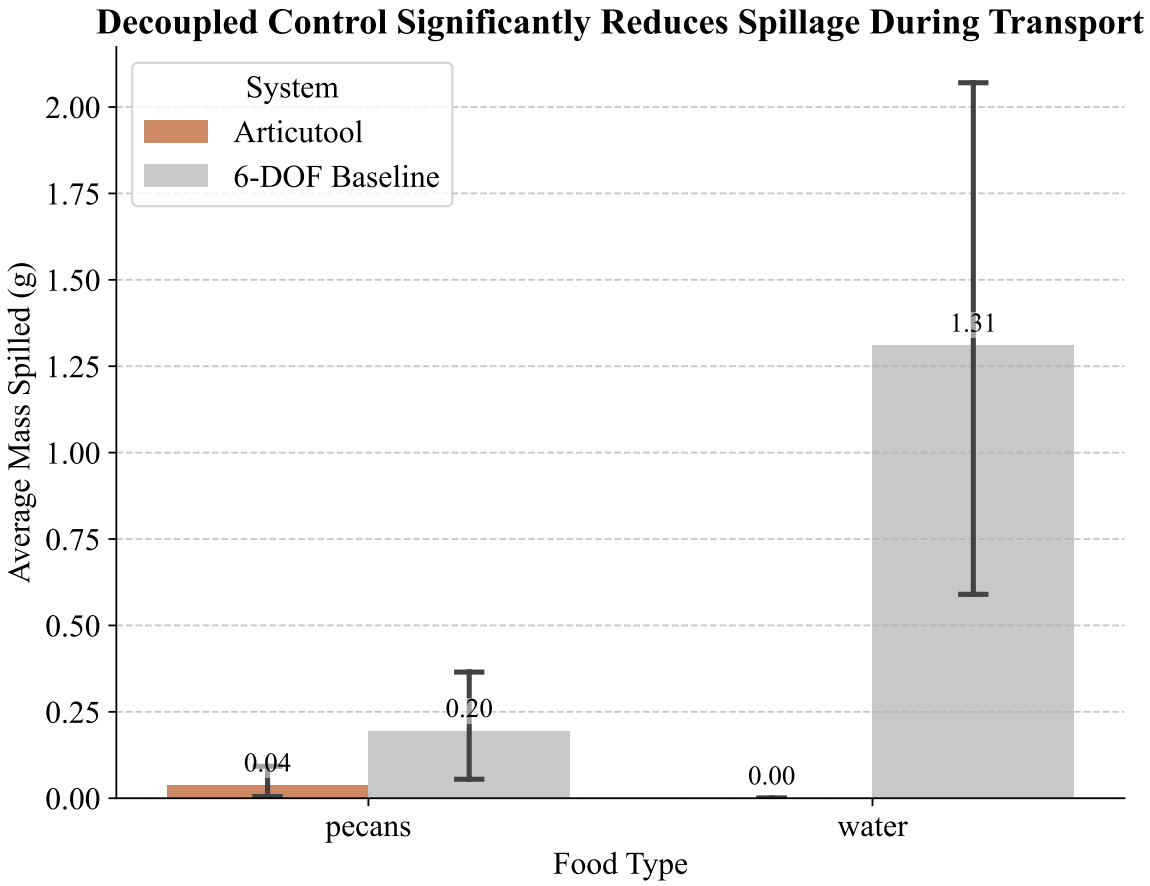


Figure 6.3: Physical experiment results (Exp. 2b) confirming the Articutool's decoupled control significantly reduces spillage during transport (H2). The system's active leveling virtually eliminated spillage for both pecans and water, a statistically significant improvement over the 6-DOF baseline.

difference is statistically significant ($p < 0.00012$) and provides initial validation for H3.

However, a deeper analysis of the baseline's 7 non-spill successes reveals that 5 out of 7 (71%) delivered a trivial mass of food ($\leq 2.1g$, from the set {0.9, 1.5, 1.5, 1.7, 2.1}). These "successes" are statistically indistinguishable from trials where this baseline simply failed to acquire a meaningful bite. In contrast, 100% of the Articutool's 14 successful trials delivered a technically successful bite (minimum 4.9

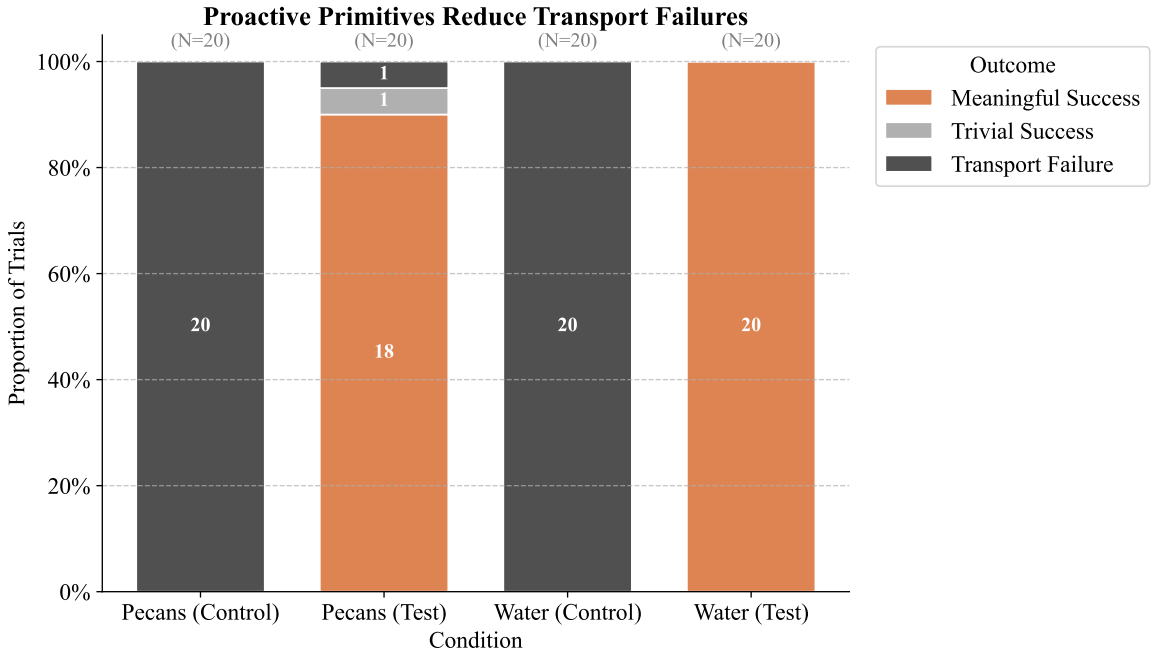


Figure 6.4: Proactive shedding primitives (Exp. 2c) significantly reduce transport failures. The `VIBRATE` primitive (for pecans) and `TILT` primitive (for water) convert the high transport failure rates seen in the control groups into meaningful success in the test groups. Crucially, the primitives do not often result in trivial success, confirming they are well-tuned to mitigate spill risk without sacrificing a satisfying bite, a key component of H2. (N=20 for each condition).

g).

This finding motivated the "meaningful success" metric defined in Exp. 3. When filtering for trials that both had zero spillage and delivered a meaningful bite (defined conservatively as > 3.0 g), the Articutool's success rate remains a robust 70.0% (14/20), while the baseline's success rate collapses to 10.0% (2/20).

Furthermore, the data reveals a significant difference in failure criticality. The Articutool caused a "critical spill" (spilling on or near the user) in only 10.0% of trials (2/20). The baseline was 2.5 times more likely to commit this error, spilling on the user in 25.0% of its trials (5/20).

Finally, the real-world data from Exp. 3 confirmed the simulation findings from

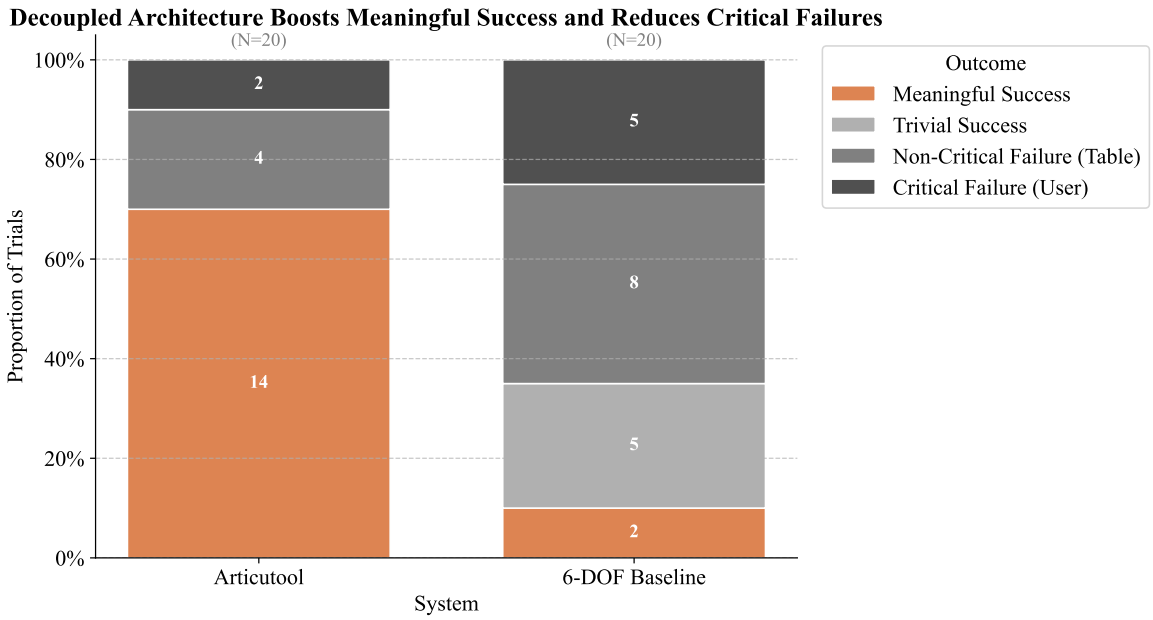


Figure 6.5: The decoupled architecture achieves a 70.0% (14/20) Meaningful Success rate. In contrast, 90% (18/20) of the baseline's trials result in a failure or a Trivial Success. The baseline is also 2.5 times more likely to cause a Critical Failure, spilling on the user. Annotations show raw trial counts for each outcome.

Exp. 2a. To find a valid path to the `Staging` configuration, the Articutool required an average of only 1.15 planning attempts, while the strictly-constrained baseline required 2.50 attempts, reinforcing that the decoupled approach is more computationally efficient and reliable. The combined findings provide comprehensive validation for H3.

The stacked bar chart does not capture the magnitude of these failures. The baseline's simple scooping motion acquires an unstable bite, creating a situation where loose food is destined to fall. As shown in Fig. 6.6, this unmitigated risk leads to a catastrophic worst-case failure (b). Conversely, the Articutool's architecture is designed to prevent this state. By using the twirling acquisition strategy, and using localized primitives to proactively shed unstable food back onto the plate before transport, its worst-case failure (a) is smaller, demonstrating a more robust and user-

((a)) Articutool Worst-Case Failure

((b)) 6-DOF Baseline Worst-Case Failure

Figure 6.6: Comparison of worst-case failure modes. (a) The Articutool's combined acquisition and mitigation strategy results in a more stable bite, so its worst-case failure is minor. (b) The baseline's simple scoop transports an unstable bite, leading to catastrophic failure. In both cases, "non-critical" spills are often just this inevitable spillage that, by chance, did not fall on the user.

safe design.

Chapter 7

CONCLUSION AND FUTURE WORK

This thesis presented the design, control, and evaluation of the Articutool, a modular and untethered 2-DOF intelligent wrist for assistive feeding. We demonstrated that by decoupling the concerns of gross arm transport from fine-grained tool manipulation, our system achieves significantly higher reliability on the challenging task of spill-free feeding. Our core contributions include the fully self-contained hardware, a hierarchical control strategy featuring proactive "plan-then-verify" feasibility checks, and a set of localized primitives for proactive spill mitigation.

Our results validate the core premise that decoupling the feeding task enhances robustness and reliability, directly addressing a key barrier to user acceptance identified in prior work [2], [11]. This work demonstrated that the Articutool's decoupled architecture achieves a 70.0% meaningful success rate on the end-to-end feeding task, in stark contrast to the baseline's 10.0%. This translates to a more dependable experience for the user, reducing the likelihood of frustrating or messy spills. Furthermore, the system's failures were 2.5 times less likely to be critical, user-impacting spills, providing quantitative validation of a more reliable and user-centric system.

This real-world success stems from robustly and efficiently solving the spill-free transport problem. Our large-scale simulation demonstrated that while monolithic 8-DOF systems are kinematically capable of performing these motions (95.0% success), the high dimensionality of their search space imposes a prohibitive computational cost (mean planning time $> 100s$). By decoupling the problem, the Articutool achieves comparable success rates (96.0% success) in a fraction of the time (mean planning time 4.7s). This confirms that problem decomposition is key to resolving the conflict

between strict constraints and real-time responsiveness. By solving this technical bottleneck, the Articutool architecture represents a crucial step towards enabling assistive robots to reliably serve a wider variety of the foods people actually want to eat, enhancing user independence and satisfaction.

Furthermore, the end-to-end trials demonstrated the value of the Articutool's architecture in enabling task-specific, intelligent acquisition strategies. By refining the acquisition primitive to prioritize bite suitability and proactive spill mitigation over sheer mass, we demonstrated a clear, user-centered design trade-off. This highlights that the system's success is not just from decoupled transport, but also from its ability to intelligently prepare a stable, practical bite for that transport.

It is also important to note the deliberate scoping of our experimental evaluation. The end-to-end evaluation in Exp. 3 focused exclusively on noodles, which was by design. Our previous experiments had already demonstrated that the Articutool could successfully acquire granulars and liquids (H1) and, crucially, that our decoupled control and proactive primitives could transport these food types with virtually zero spillage (H2). An end-to-end test with pecans or water would therefore have been redundant. Noodles, in contrast, represented the most challenging case, requiring the successful integration of our most complex acquisition strategy with the robust transport architecture. By demonstrating a 70.0% meaningful success rate on this most difficult task, we provide strong evidence that the system's performance on the simpler, previously-validated tasks of pecans and water would be at least as high, if not higher.

While a monolithic 8-DOF system is kinematically capable, the Articutool's primary advantages are architectural and philosophical. Instead of demanding complex, constrained planning from a single high-DOF system, our approach offers architectural simplicity through problem decomposition: a standard 6-DOF planner handles gross transport with loose constraints, while the Articutool's independent controller manages local stabilization. This separation greatly reduces planning complexity.

Furthermore, this highlights the system's modularity; the Articutool functions as an untethered, task-specific appliance, not a permanent fixture. This enables a versatile assistive system where a general-purpose arm can leverage an ecosystem of intelligent tools.

7.1 Limitations

While the Articutool demonstrates a significant advance in robust robot-assisted feeding, it has limitations which span hardware design, control methodology, and the scope of evaluation which provide clear avenues for future work.

7.1.1 Hardware and Physical Design

The current iteration of the Articutool is a fully functional prototype that successfully validates the core principles of decoupled control. However, as a research platform, its design has several limitations:

- **Lack of Autonomous Tool Changing:** The system currently relies on manual attachment to the Jaco arm's gripper. It does not possess the capability for autonomous docking, charging, or tool-tip swapping. This limits its practical autonomy, as a caregiver is still required for setup, and the system cannot yet function as a truly independent appliance within a larger ecosystem of tools.

- **Durability and Form Factor:** The prototype makes extensive use of 3D-printed structural components. While ideal for rapid prototyping and iteration, these materials may not be suitable for the long-term durability and cleaning requirements of a consumer-facing assistive device. Further refinement using injection-molded or machined parts would be necessary for a production-level system.

- **F/T Sensor Wire Management:** A key hardware limitation is the lack of active wire management for the distal 6-Axis Force/Torque sensor. While the multi-rotation `TWIRL` strategy was intentionally designed to "wind-up" and then "unwind" the cable to keep its net extension near zero, the physical "dangling" of the wire makes this process unreliable. Without a retractable module to keep the wire taut, the slack cable does not always wind neatly around its intended adapter. During physical experiments, this required manual intervention to prevent it from winding incorrectly around the base of the fork or, in a critical failure case, touching the food during the acquisition maneuver. This remains a significant point of failure for autonomous, real-world operation.

7.1.2 Control and Software Architecture

The "plan-then-verify" software architecture and localized primitives proved highly effective, but their scope is currently constrained:

- **Reliance on Pre-defined Primitives:** The acquisition strategies for challenging foods like noodles and pecans rely on a vocabulary of pre-defined, open-loop primitives (e.g., `TWIRL`, `VIBRATE`, `TILT`). While effective for the tested foods, this approach may not generalize to all food types or textures. A more advanced system might incorporate learning-based methods or closed-loop, sensor-driven primitives that can adapt their strategy in real-time based on force or visual feedback.

- **Focus on Trajectory-Level Compensation:** The leveling controller is designed to compensate for the low-frequency, high-amplitude orientation changes dictated by the arm's trajectory. It is not designed as a high-frequency active stabilizer to cancel out vibrations or sudden small-scale disturbances, nor is it designed to model and react to the dynamics of the food within the tool during transport. For example, while measurable spillage during transport was

eliminated, we observed that negligible micro-droplets could result from uncompensated high-frequency vibrations. This underscores the Articutool's focus on low-frequency trajectory compensation, distinct from high-frequency stabilizers like Liftware [20], and suggests a potential area for refinement.

7.1.3 Evaluation Scope

The experiments in this thesis provide a thorough technical validation, but the scope of this evaluation has three principal limitations that must be addressed in future work.

First, and most significantly, this work lacks formal user studies with the target population. All experiments were conducted in a controlled laboratory setting, and our "meaningful success" metric was based on a technical threshold rather than a subjective, user-perceived metric for a satisfying bite. Consequently, while we have demonstrated technical reliability, we have not yet measured the human-robot interaction metrics such as user trust, comfort, or perceived independence that are critical for real-world adoption.

Second, the generalizability of our acquisition strategies is not fully explored. While we targeted challenging food categories (granular, noodles, liquids), the system was only tested on a small number of specific food items. Its generalization capabilities across a wider and more diverse diet remain to be explored.

Finally, our experimental design included methodological factors that must be acknowledged. The 6-DOF baseline, by programmatically "locking" the Articutool's joints, was put at a significant kinematic disadvantage that likely contributed to its poor acquisition performance, particularly with liquids. We also observed inter-experiment variation in food properties (e.g., noodle moisture, noodle length), which prevents direct numerical comparison of mass between experiments, though the bimodal failure phenomenon that justified our success threshold remained consistent.

7.2 Future Work

The successful validation of the Articutool's architecture, combined with the limitations identified above, provides a clear roadmap for future research. This work can be extended by (1) realizing the full vision of modularity through autonomous tool management, and (2) translating the system's technical success into human-centric value through formal user studies.

7.2.1 Autonomous Tool Changing and Management

To address the hardware and modularity limitations (Section 7.1.1), the robot must be able to manage the Articutool without human intervention. This involves key capabilities:

- **Autonomous Docking and Charging:** We envision the development of a docking station where the Jaco arm can autonomously place the Articutool for storage and recharging. This would free the arm to perform other Activities of Daily Living (ADLs) with its standard gripper, such as opening doors or retrieving objects, before re-equipping the Articutool for a meal. The technical challenges include developing a robust visual servoing or force-based docking procedure to ensure precise alignment with charging contacts. Furthermore, the system's software architecture would need to manage the tool's state, automatically re-establishing communication when the tool is picked up.

- **Tool-Tip Quick-Swapping:** While the current work focuses on using a single utensil at a time, a diverse meal often requires multiple utensils. We propose a tool-tip quick-swapping procedure where the arm could dock the Articutool and autonomously exchange the fork for a spoon, or vice-versa. This would allow the user to seamlessly switch between utensils to best suit different food items during a single meal, further enhancing their dining experience.

- **Robust F/T Sensor Wire Management:** We envision the integration of an active wire management system incorporating a retractable cable module. This component would apply a constant retraction force to the F/T wire, keeping it taut as it automatically winds and releases. This would enable the roll motor to perform continuous rotations reliably, eliminating the dangling wire failure case, and making multi-rotation primitives, like the noodle-twirling strategy, fully autonomous.

7.2.2 Human-Robot Interaction and Formal User Studies

To address the critical gap in our evaluation (Section 7.1.3), the next crucial step is to conduct formal user studies with the target population to evaluate the system's real-world impact. These studies will move beyond technical success rates to measure the quality of the human-robot interaction. Key research to investigate include:

- **User Trust and Perceived Safety:** How does the system's enhanced reliability and proactive spill prevention impact a user's trust and sense of safety during the feeding process?

- **Cognitive Load:** Does the autonomous and reliable nature of the Articutool system reduce the cognitive load on the user, allowing them to focus on their meal and social interaction rather than supervising the robot?

- **Task Efficiency and Perceived Independence:** How does the system affect overall meal duration and the need for caregiver intervention?

Answering these questions is essential for translating the technical successes of the Articutool into a system that is not only functional but also improves the quality of life for its users.

Appendix A

BILL OF MATERIALS

The following table details the components used to construct the Articutool wrist. Prices are approximate as of late 2025 and are provided for estimation purposes.

Table A.1: Articutool Bill of Materials

Item	**Component**	**Qty**	**Part Number / Link**	**Approx. Price (USD)**	**Notes**
Computation	Raspberry Pi 5	1	Model B, 4GB RAM	120.00	Main onboard computer
Actuation	Dynamixel XC430-W250-T	2	902-0147-000	240.00	Pitch and Roll joints
	U2D2 Power Hub	1	902-0124-001	20.00	Motor controller interface
Sensing	OpenLog Artemis	1	SparkFun SEN-16832	54.95	IMU data logger
	ReSense HEX21	1	HEX-M-21-2.5	4,699.87	6-Axis Force/Torque sensor
Power	Anker 10,000mAh Power Bank	1	A1245	39.99	USB-C PD power source
	USB-C PD Trigger Module	1	ZY12PDN	7.99	Negotiates 12V from power bank
Mechanical	3D Printed Parts	1	N/A	N/A	Structural frame and mounts
	Utensil (Spoon/Fork)	1	N/A	600.00	Standard stainless steel

BIBLIOGRAPHY

[1] R. K. Jenamani, P. Sundaresan, M. Sakr, T. Bhattacharjee, and D. Sadigh, *FLAIR: Feeding via Long-horizon AcquIsition of Realistic dishes*, en, arXiv:2407.07561 [cs], Jul. 2024. DOI: 10.48550/arXiv.2407.07561. Accessed: Jun. 28, 2025. [Online]. Available: http://arxiv.org/abs/2407.07561.

[2] A. Nanavati, "Lessons Learned from Designing and Evaluating a Robot-assisted Feeding System for Out-of-lab Use," en,

[3] L. V. Herlant, "Algorithms, Implementation, and Studies on Eating with a Shared Control Robot Arm," en,

[4] S. Roy, J. Viju, and B. Bhattacharyya, "Adaptive Robotic Manipulator Simulation for Enhanced Feeding and Drinking Assistance," en, in *2024 IEEE 21st India Council International Conference (INDICON)*, Kharagpur, India: IEEE, Dec. 2024, pp. 1–6, ISBN: 979-8-3503-9128-2. DOI: 10.1109/INDICON63790.2024.10958301. Accessed: Sep. 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10958301/.

[5] M. N. Keely, H. Nemlekar, and D. P. Losey, "Kiri-Spoon: A Soft Shape-Changing Utensil for Robot-Assisted Feeding," en, in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Abu Dhabi, United Arab Emirates: IEEE, Oct. 2024, pp. 121–128, ISBN: 979-8-3503-7770-5. DOI: 10.1109/IROS58592.2024.10801346. Accessed: Jun. 28, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10801346/.

[6] M. Keely et al., *Kiri-Spoon: A Kirigami Utensil for Robot-Assisted Feeding*, en, arXiv:2501.01323 [cs], Jan. 2025. DOI: 10.48550/arXiv.2501.01323. Accessed: Sep. 12, 2025. [Online]. Available: http://arxiv.org/abs/2501.01323.

[7] A. Nanavati, "Towards In-Home Deployments of Physically Assistive Robots: Insights from Robot-Assisted Feeding for People with Motor Impairments," en,

[8] Won-Kyung Song, Won-Jin Song, Yale Kim, and Jongbae Kim, "Usability test of KNRC self-feeding robot," en, in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, Seattle, WA: IEEE, Jun. 2013, pp. 1–5, ISBN: 978-1-4673-6024-1 978-1-4673-6022-7. DOI: 10.1109/ICORR.2013.6650501. Accessed: Oct. 19, 2025. [Online]. Available: http://ieeexplore.ieee.org/document/6650501/.

[9] J. S. Kim, W.-K. Song, S. R. Han, J. E. Heo, and K. J. Lee, "A study on the scooping and serving Korean food using meal assistance robot," en, *Gerontechnology*, vol. 21, no. s, pp. 1–1, Oct. 2022, ISSN: 15691101, 1569111X. DOI: 10.4017/gt.2022.21.s.650.opp3. Accessed: Oct. 19, 2025. [Online]. Available: https://journal.gerontechnology.org/currentIssueContent.aspx?aid=3160.

[10] S. Ljungblad, "Applying "Designerly Framing" to Understand Assisted Feeding as Social Aesthetic Bodily Experiences," en, *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 2, pp. 1–23, Jun. 2023, ISSN: 2573-9522, 2573-9522. DOI: 10.1145/3583742. Accessed: Oct. 19, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3583742.

[11] A. Nanavati, P. Alves-Oliveira, T. Schrenk, E. K. Gordon, M. Cakmak, and S. S. Srinivasa, "Design Principles for Robot-Assisted Feeding in Social Contexts," en, in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, Stockholm Sweden: ACM, Mar. 2023, pp. 24–33, ISBN: 978-

1-4503-9964-7. DOI: 10.1145/3568162.3576988. Accessed: Jun. 28, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3568162.3576988.

[12] M. Topping, "An overview of the development of Handy 1, a rehabilitation robot to assist the severely disabled," en, *Artificial Life and Robotics*, vol. 4, no. 4, pp. 188–192, Dec. 2000, ISSN: 1433-5298, 1614-7456. DOI: 10.1007/BF02481173. Accessed: Jun. 28, 2025. [Online]. Available: http://link.springer.com/10.1007/BF02481173.

[13] Mealtime Partners, Inc., "Assistive Dining Technology 1970 through 2005," Mealtime Partners, Inc., Azle, TX, Tech. Rep., 2005.

[14] M. Hillman, "2 Rehabilitation Robotics from Past to Present – A Historical Perspective," en, in *Advances in Rehabilitation Robotics*, Z. Z. Bien and D. Stefanov, Eds., vol. 306, Series Title: Lecture Notes in Control and Information Science, Springer Berlin Heidelberg, 2004, pp. 25–44, ISBN: 978-3-540-21986-6. DOI: 10.1007/10946978_2. Accessed: Oct. 19, 2025. [Online]. Available: http://link.springer.com/10.1007/10946978_2.

[15] Obi Robotics, *Meet Obi: The Adaptive Eating Device*, 2024. [Online]. Available: https://meetobi.com.

[16] D. Berenson, S. Srinivasa, and J. Kuffner, "Task Space Regions: A framework for pose-constrained manipulation planning," en, *The International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, Oct. 2011, ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364910396389. Accessed: Oct. 18, 2025. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0278364910396389.

[17] E. K. Gordon, A. Nanavati, R. Challa, B. H. Zhu, T. A. K. Faulkner, and S. S. Srinivasa, "Towards General Single-Utensil Food Acquisition with Human-Informed Actions," en, 2023.

[18] T. Bhattacharjee, G. Lee, H. Song, and S. S. Srinivasa, "Towards Robotic Feeding: Role of Haptics in Fork-Based Food Manipulation," en, *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1485–1492, Apr. 2019, ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2019.2894592. Accessed: Jun. 28, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/8624330/.

[19] D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp, "Multimodal execution monitoring for anomaly detection during robot manipulation," en, in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden: IEEE, May 2016, pp. 407–414, ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487160. Accessed: Jun. 28, 2025. [Online]. Available: http://ieeexplore.ieee.org/document/7487160/.

[20] A. Pathak, J. A. Redmond, M. Allen, and K. L. Chou, "A noninvasive handheld assistive device to accommodate essential tremor: A pilot study," en, *Movement Disorders*, vol. 29, no. 6, pp. 838–842, May 2014, ISSN: 0885-3185, 1531-8257. DOI: 10.1002/mds.25796. Accessed: Sep. 12, 2025. [Online]. Available: https://movementdisorders.onlinelibrary.wiley.com/doi/10.1002/mds.25796.

[21] O. Khatib, "Mobile manipulation: The robotic assistant," en, *Robotics and Autonomous Systems*, vol. 26, no. 2-3, pp. 175–183, Feb. 1999, ISSN: 09218890. DOI: 10.1016/S0921-8890(98)00067-0. Accessed: Jun. 28, 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889098000670.

[22] M. T. Shahria and M. H. Rahman, "Activities of Daily Living Object Dataset: Advancing Assistive Robotic Manipulation with a Tailored Dataset," en, *Sensors*, vol. 24, no. 23, p. 7566, Nov. 2024, ISSN: 1424-8220. DOI: 10.3390/s24237566. Accessed: Oct. 18, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/24/23/7566.

[23] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, Architecture, and Uses In The Wild," en, *Science Robotics*, vol. 7, no. 66, eabm6074, May 2022, arXiv:2211.07752 [cs], ISSN: 2470-9476. DOI: 10.1126/scirobotics.abm6074. Accessed: Oct. 19, 2025. [Online]. Available: http://arxiv.org/abs/2211.07752.

[24] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction*, en. Jul. 2018, arXiv:1709.00084 [cs]. DOI: 10.1201/9780429489105. Accessed: Oct. 18, 2025. [Online]. Available: http://arxiv.org/abs/1709.00084.

[25] P. Stoop, T. Ratnayake, and G. Toffetti, "A Method for Multi-Robot Asynchronous Trajectory Execution in MoveIt2," en, in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan: IEEE, May 2024, pp. 17 694–17 700, ISBN: 979-8-3503-8457-4. DOI: 10.1109/ICRA57147.2024.10611498. Accessed: Nov. 11, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10611498/.

[26] R. Valenti, I. Dryanovski, and J. Xiao, "Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs," en, *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, Aug. 2015, ISSN: 1424-8220. DOI: 10.3390/s150819302. Accessed: Oct. 18, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/15/8/19302.

[27] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," en, in *2011 IEEE International Conference on Rehabilitation Robotics*, Zurich: IEEE, Jun. 2011, pp. 1–7, ISBN: 978-1-4244-9862-8 978-1-4244-9863-5 978-1-4244-9861-1. DOI: 10.1109/ICORR.2011.5975346. Accessed: Oct. 27, 2025. [Online]. Available: http://ieeexplore.ieee.org/document/5975346/.

[28] J. Carpentier et al., "The Pinocchio C++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical deriva-

tives," en, in *2019 IEEE/SICE International Symposium on System Integration (SII)*, Paris, France: IEEE, Jan. 2019, pp. 614–619, ISBN: 978-1-5386-3615-2. DOI: 10.1109/SII.2019.8700380. Accessed: Oct. 25, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/8700380/.

[29] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," en, in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, San Francisco, CA, USA: IEEE, 2000, pp. 995–1001, ISBN: 978-0-7803-5886-7. DOI: 10.1109/ROBOT.2000.844730. Accessed: Oct. 18, 2025. [Online]. Available: http://ieeexplore.ieee.org/document/844730/.

[30] S. Karaman and E. Frazzoli, *Sampling-based Algorithms for Optimal Motion Planning*, en, arXiv:1105.1186 [cs], May 2011. DOI: 10.48550/arXiv.1105.1186. Accessed: Oct. 18, 2025. [Online]. Available: http://arxiv.org/abs/1105.1186.