# Vision-Based Safety Assurance and Generative Task Planning System for Human-Robot Collaboration

Hsin-Yu Chen
Department of Electrical Engineering
National Taiwan Normal University
Taipei, Taiwan
61375017h@ntnu.edu.tw

Chen-Chien Hsu
Department of Electrical Engineering
National Taiwan Normal University
Taipei, Taiwan
jhsu@ntnu.edu.tw

Wei-Yen Wang
Department of Electrical Engineering
National Taiwan Normal University
Taipei, Taiwan
wywang@ntnu.edu.tw

Shao-Kang Huang
Department of Artificial Intelligence
Tamkang University
Taipei, Taiwan
169394@o365.tku.edu.tw

***Abstract*—This paper introduces a robust vision-based safety assurance system for Human-Robot Collaboration (HRC), hierarchically integrated with Large Multimodal Models (LMMs) for intelligent task planning. We propose an experimental setup utilizing a Universal Robot (UR3) and a top-down Intel RealSense D435f RGB-D camera. The core contribution is a deterministic safety layer that leverages MediaPipe for skeletal tracking, enhanced by a Kalman Filter state estimator and a novel "Hard-Connect" kinematic constraint to maintain robust hand tracking under occlusion. The visual feedback constructs dynamic obstacles within a static "Safety Cage" in the MoveIt planning scene. Furthermore, to validate the system in a non-deterministic environment, we incorporate the Gemini 3 Flash-Preview model to reason about audio-visual inputs and generate structured assembly plans for a chair assembly task. The system translates natural language and visual inputs into structured JSON commands, executing a "Reset-Pick-Place" strategy while maintaining strict physical safety boundaries. Motion planning is executed using MoveIt 2 with the PRM* (Probabilistic Roadmap) solver to navigate the constrained workspace, governed by a "Stop-Wait-Replan" Finite State Machine (FSM). Quantitative evaluations across 215 dynamic interrupts and 20 assembly cycles demonstrate 100% LMMs intent accuracy, a 40% reduction in false-positive safety breaches, and zero physical collisions. Kinematic safety analysis validates an edge-computed perception-to-halt latency of 65.0ms, mathematically guaranteeing a maximum safe stopping distance of 39.66 mm at an operational velocity of 250mm/s, thereby strictly complying with ISO/TS 15066 collaborative safety standards.**

***Keywords—Large Multimodal Models (LMMs), Human-Robot Collaboration (HRC), Vision-Based Safety Assurance, Task planning, Motion planning, Robot.***

## I. INTRODUCTION

### A. Background and Motivation

Human-Robot Collaboration (HRC) represents a paradigm shift in manufacturing, transitioning from isolated industrial robots behind physical fences to "cobots" that share workspaces with human operators. While this proximity enhances flexibility, safety remains the primary bottleneck. Traditional safety protocols often rely on static zones or expensive volumetric sensors that lack semantic understanding of the task.

In previous work from our laboratory [1], vision-based HRC was explored by enabling a manipulator to replicate tasks through human visual demonstration. While Learning from Demonstration (LfD) [2]-[4] is highly effective for repetitive manufacturing, it inherently limits the robot to pre-taught trajectories and lacks the semantic flexibility to adapt to novel assembly sequences without prior teaching. To address this limitation, the current work shifts the control paradigm from visual imitation to generative reasoning.

The recent emergence of Generative AI and Large Multimodal Models (LMMs) has enabled robots to interpret complex, natural language instructions, allowing them to dynamically plan tasks "zero-shot." However, these probabilistic models suffer from "hallucinations" and cannot inherently guarantee physical safety. A robot driven solely by an LMMs may generate trajectories that are semantically correct but kinematically dangerous. Therefore, to safely deploy generative reasoning in physical workspaces, a novel deterministic safety architecture is required to govern the AI's unpredictable motion.

### B. Problem Statement

Integrating Generative AI into HRC introduces two critical challenges regarding system reliability. The primary issue is non-determinism, while LMMs govern tasks probabilistically, established industrial safety standards such as ISO/TS 15066 mandate strict deterministic safety bounds. Furthermore, the physical system must overcome occlusion and sensor noise. Traditional vision-based safety setups frequently lose track of small, fast-moving extremities or generate erratic, "jittery" obstacles due to depth variance, causing unnecessary robot stoppages that degrade collaborative efficiency.

### C. Proposed Solution

To address the above-mentioned issues, this paper bridges the gap between high-level reasoning and low-level safety by proposing a Dual-Layer Control Architecture:

1. The Cognitive Layer:
   Utilizing Gemini 3 Flash-Preview [5] to process multimodal inputs (Audio & Vision) and generate a structured JSON assembly plan.
2. The Safety Layer:
   A real-time, deterministic vision system that maps the human operator's arms as dynamic obstacles. We introduce a "Hard-Connect" kinematic logic that anchors the hands to the wrists, ensuring safety even when the depth sensor fails to resolve the fingers.

### D. Contributions

The specific contributions of this work are as follows:

- Robust Skeletal Tracking:
  A cascaded state estimator using Median Filtering is implemented for depth and Kalman Filtering for

trajectory smoothing, significantly reducing sensor noise.

- Optimized Constrained Motion Planning:
  The asymptotically optimal PRM* solver is applied to navigate the "narrow passage" problem created by the static cage. Unlike standard tree-based planners, PRM* eliminates redundant joint-space trajectories, minimizing robot "waving" and maximizing motion efficiency when operating in close proximity to dynamic human obstacles.
- Deterministic FSM Control:
  A "Stop-Wait-Replan" Finite State Machine that is developed to replace standard reactive re-planning, eliminating oscillatory motion and ensuring psychological safety for the operator.

## II. System Architecture

### A. Hardware Setup

The experimental platform comprises a Universal Robot UR3 [6], a 6-DOF manipulator designed for safe human-robot interaction, and a single Intel RealSense D435f RGB-D camera [7]. The camera is mounted in a top-down configuration to provide a comprehensive, occlusion-free view of the workspace. The UR3 is mounted on the right of an elevated fixed base positioned at the back of the experimental platform (as shown in Fig. 1). This spatial arrangement maximizes the robot's kinematic reach while minimizing self-occlusion during the assembly process.

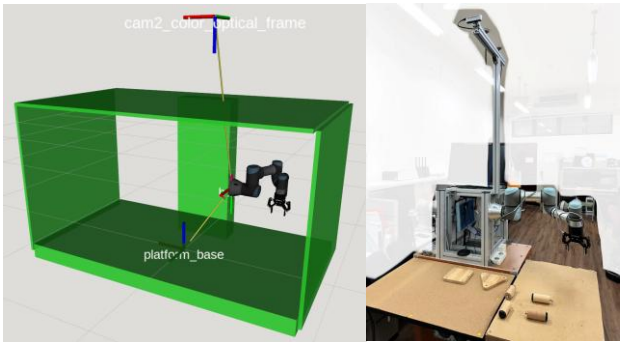


Figure 1. Setup of the experimental environment.

### B. Software Framework

The system is built upon the Robot Operating System 2 (ROS 2 Humble) [8]. The MoveIt 2 [9] motion planning framework is utilized for trajectory generation and inverse kinematics. The vision processing pipeline utilizes MediaPipe [10] for efficient, low-latency human skeleton detection. To ensure the RViz 2 virtual environment accurately reflects the physical workspace, we implemented a robust coordinate transformation pipeline. This module guarantees that the raw RGB-D information is precisely mapped to the robot's base frame, allowing MoveIt 2 to perform valid collision checking against dynamic obstacles.

## III. Generative Task Planning

To enable flexible HRC, we developed a ROS 2 node, *gemini_object_locator*, which serves as the cognitive agent for the system. As shown in Fig. 2, this agent utilizes Gemini 3 Flash-Preview to process multimodal inputs and then generate the structured spatial goal.

### A. Multimodal Input Processing

The agent operates in a "Listen-and-look" cycle, simultaneously capturing two synchronized inputs upon being triggered: a 5-second voice command from a microphone to establish the audio instruction (e.g., "Assemble the triangular chair"), and a high-resolution RGB snapshot from the overhead camera to provide the visual context.

### B. Semantic Reasoning and Prompting

We employ a chain-of-thought prompt that forces the LMMs to correlate the audio intent with visual grounding through three logical steps. First, the model performs intent classification by analyzing the audio to determine the requested assembly type (e.g., a "square" versus "triangular" plate). Second, it conducts quantity inference based on this classification to deduce the exact number of required sub-components, such as locating exactly three legs for a triangular plate. Finally, the model executes spatial sorting by detecting the necessary components and ordering them from right-to-left in pixel coordinates, thereby generating a deterministic pick-and-place sequence. The output is a structured JSON object containing the target *plate_type*, the *leg_amount*, and a list of *items*, where each item includes a 2D bounding box [ymin, xmin, ymax, xmax] and rotation angle.

### C. Depth Extraction and Spatial Grounding

To convert the LMMs' 2D bounding boxes into valid 3D robot goals, we implement a robust depth extraction algorithm. For every detected item centroid $(u, v)$, we extract a $10 \times 10$ pixel patch from the aligned depth map. We apply a median

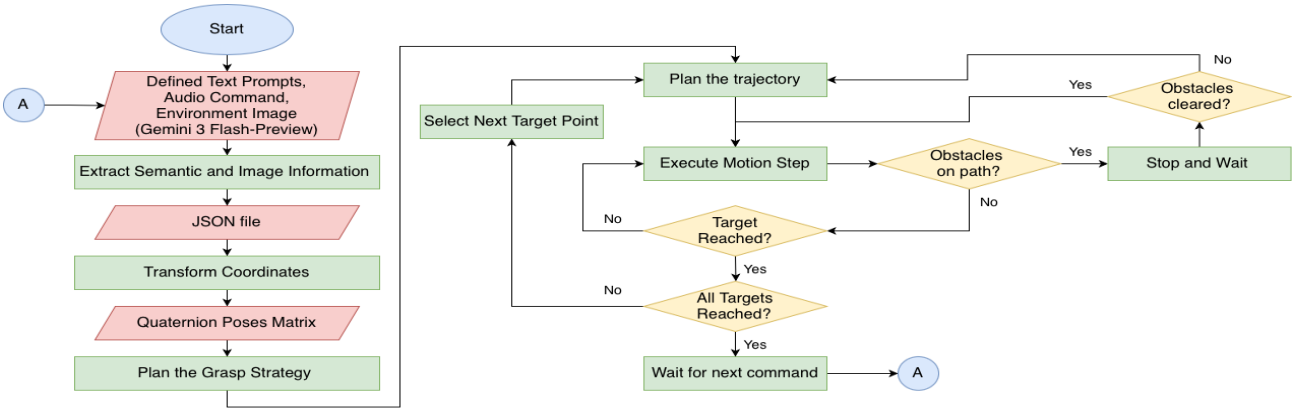


Figure 2. Overall workflow of the proposed HRC safety assurance scenario, including the Generative AI task planning and the real-time vision-based safety assurance logic.

filter to this patch to reject sensor noise and "flying pixels" at object edges:

$$d_{robust} = median(D_{u\pm5,v\pm5}) \quad (1)$$

where $d_{robust}$ is the filtered depth value, $D$ represents the aligned depth map, $(u, v)$ denotes the 2D pixel coordinates of the detected object's centroid, and the subscript $u \pm 5, v \pm 5$ defines the $10 \times 10$ pixel neighborhood centered around $(u, v)$ from which the median depth is computed.

This filtered depth $d_{robust}$ combined with the intrinsic camera matrix $K$, allows us to deproject the pixel coordinates into a 3D point in the camera frame. Finally, the LMMs-predicted rotation angle $\theta$ is converted into a quaternion orientation (yaw), enabling the robot to grasp objects at the correct approach angle.

## IV. Real-Time Vision-Based Safety Assurance System

The safety architecture shown in Fig.2 constructs a dynamic "digital twin" of the environment, enforcing both static workspace limits and dynamic human avoidance. This is achieved through two parallel ROS 2 nodes: a static scene builder and a dynamic skeletal tracker.

### A. Static Safety Cage

To confine the manipulator within safe physical limits, a static "Safety Cage" is constructed relative to the $platform\_base$ frame. This digital environment comprises invisible virtual walls and a roof ($SolidPrimitive.BOX$) to enclose the operational volume, combined with a solid platform exclusion zone to prevent downward collisions with the physical table. By broadcasting these static constraints directly to the MoveIt 2 planning scene via $tf2\_ros$ static transforms, the system guarantees that the motion planning solver automatically rejects any trajectories violating these predefined spatial boundaries.

### B. Dynamic Skeletal Tracking with State Estimation

For human avoidance, we utilize MediaPipe to extract 2D keypoints for the body (Pose) and hands (Hands). However, raw visual detections are subject to jitter and depth noise. We mitigate this through a cascaded state estimation pipeline:

1. Robust Depth Extraction:
   Depth sensors often produce noise at object edges (flying pixels). For every detected keypoint $(u, v)$, we extract a $k \times k$ pixel patch ($k = 5$) from the aligned depth map and apply a median filter to derive a stable depth value $d_{stable}$.
2. Kalman Filtering:
   We implement independent Kalman Filters for each of the 6 distinct tracking points (shoulders, elbows, wrists). The filters utilize a Constant Velocity (CV) model to smooth the 3D coordinate trajectories, effectively rejecting sensor outliers and smoothing the motion of the generated collision objects.

### C. Kinematic Hard-Connect Hand Logic

A common failure mode in vision-based safety is the loss of hand depth data due to occlusion or the hand's thin geometry. To address this, we developed a "Hard-Connect" kinematic constraint: Instead of relying solely on the hand's raw depth, we derive the hand's 3D position vectorially from the stable wrist position. We calculate the normalized 2D direction vector $v$ from the wrist to the hand in pixel space and project it into 3D:

$$P_{hand} = P_{wrist} + (v_{3D} \cdot L_{offset}) \quad (2)$$

where $P_{hand}$ denotes the computed 3D spatial coordinate of the virtual hand obstacle, and $P_{wrist}$ represents the stable, Kalman-filtered 3D position of the wrist joint. The variable $v_{3D}$ is the normalized 3D unit direction vector pointing from the wrist to the hand centroid, calculated by deprojecting their respective 2D pixel coordinates using a shared, stable depth value to prevent depth noise. Finally, $L_{offset}$ is a fixed kinematic scalar constant representing the physical distance from the wrist joint to the center of the hand (empirically set to $0.08m$). By applying this vector addition, the system ensures that even if the depth sensor fails to register the thin geometry of the fingers, the virtual hand obstacle remains kinematically anchored and rigidly attached to the wrist within the collision scene.

### D. Dynamic Obstacle Instantiation

The tracked keypoints are broadcast to the MoveIt 2 *PlanningScene* as geometric primitives:

- Limbs (Forearm/Upper Arm): Modeled as Cylinders connecting the filtered joint positions.
- Joints & Hands: Modeled as Spheres. To account for system latency and robot braking distance, we apply a Safety Scale ($\lambda = 1.5$) to the radius of all generated primitives. This creates a conservative volumetric buffer around the human operator, ensuring the robot halts well before physical contact.

## V. Motion Planning and Dynamic Execution Strategy

To translate the high-level semantic goals into safe physical motion, we developed a robust control architecture based on MoveIt 2. Unlike standard industrial pick-and-place routines, our system must operate within a static "Safety Cage" while reacting to dynamic human interference. This necessitates a specialized planning configuration.

### A. Planner Selection and Kinematic Optimization

The introduction of the static safety cage creates a "narrow passage" problem in the robot's configuration space (C-space). Furthermore, the presence of a dynamic human operator requires the robot's motion to be not only collision-free but also kinematically predictable and efficient. We evaluated several sampling-based planners within the Open Motion Planning Library (OMPL) [11][12] to identify the optimal solver for this environment:

1. Standard RRT / RRTConnect:
   Initial tests using standard Rapidly-exploring Random Trees (RRTConnect) frequently resulted in planning timeouts. These planners struggle to find valid paths through the narrow operational volume of the cage boundaries without extensive parameter tuning.
2. BiTRRT (Bidirectional Transition-based RRT):
   To overcome the narrow passage problem, we tested BiTRRT [13]. While BiTRRT successfully found paths through the cage constraints faster than

standard RRT, empirical execution revealed significant kinematic inefficiencies. The generated trajectories exhibited redundant paths and unnecessary rotation across multiple joints (colloquially known as "waving"). In an HRC context, this erratic, sweeping motion increases the swept volume of the manipulator, raising the probability of intersecting with the moving human obstacle.

3. RRT*:

   To address the "waving", we evaluated optimizing planners that seek asymptotic optimality by rewiring the search tree to minimize path length. While RRT* produced smoother paths than BiTRRT, its reliance on building a new tree from scratch for every query resulted in higher computational overhead during our frequent "replan" cycles.

4. PRM* (Probabilistic Roadmap):

   Ultimately, we selected the PRM* planner (PRMstarkConfigDefault). PRM* is an asymptotically optimal, multi-query planner. By building a roadmap of the static free space (the safety cage) and optimizing the graph connections, it consistently outputs highly direct, smoothed trajectories. By employing PRM*, we observed a drastic reduction in unnecessary UR3 joint rotations.

Crucially, to maintain system responsiveness in the presence of dynamic human obstacles, we restrict the PRM* computational budget to $2.0s$ per query. Rather than allocating a monolithic planning window, which would cause the controller to hang if an obstacle blocked the only valid path, the system employs an iterative planning loop. If PRM* fails to find a valid optimized trajectory within $2.0s$, the FSM catches the timeout, pauses, and triggers a new planning request. This continuous micro-planning approach ensures the robot reacts rapidly to clearing obstacles without experiencing prolonged computational lockups, keeping the HRC environment safe and predictable.

### *B. Geometric Path Constraints*

To ensure the safety of the grasp execution, we impose strict geometric constraints on the end-effector (*EE*) during trajectory generation. Using the *moveit_msgs/Constraints* interface, we enforce:

1. Orientation Constraint:

   The EE orientation is constrained to the target quaternion with a tolerance of $\pm 0.2$ rad along the $x, y, z$ axes. This ensures the gripper remains vertical relative to the platform, preventing collisions between the held object and the cage walls.

2. Position Constraint:

   The target goal is defined as a bounding sphere with a radius $r = 0.02m$, ensuring high positional accuracy for the assembly task.

### *C. Deterministic "Stop-Wait-Replan" Architecture*

Standard MoveIt reactive replanning often leads to oscillatory behavior when a dynamic obstacle borders the robot's path. To ensure system stability and psychological safety for the operator, we disable internal replanning and implement an application-level Stop-Wait-Replan FSM:

1. Kinematic Scaling:

   Trajectories are requested with a maximum velocity scaling factor of $\alpha_{vel} = 0.8$ and, crucially, a maximum acceleration scaling factor of $\alpha_{acc} = 0.1$. This low acceleration limit minimizes the robot's kinetic energy, ensuring that if a protective stop is triggered, the stopping distance is negligible.

2. Execution Monitoring:

   The control node sends an asynchronous goal to the trajectory server. If the vision system inserts a collision object into the path, the trajectory execution aborts, returning a non-success error code (e.g., *ABORTED*)

3. The "Wait" State:

   Upon detecting a failure, the system does not immediately retry. Instead, it enters a blocking *time.sleep(3.0)* state. This explicit pause serves two critical functions. Primarily, it provides necessary clearance by granting the human operator sufficient time to retract their arm from the shared workspace safely. Secondarily, it enhances system stability by preventing the motion planner from wasting computational resources attempting to solve for a trajectory that remains obstructed.

4. Iterative Re-planning:

   After the wait period, the loop increments the attempt counter and requests a fresh plan from the robot's new static configuration. This cycle repeats until the target is reached or a maximum retry limit is exceeded.

## VI. EXPERIMENTAL RESULTS

To validate the proposed dual-layer architecture (Cognitive & Safety), we conducted a series of experiments focusing on multimodal intent recognition, planner kinematic efficiency, and the robustness of the dynamic safety layer.

### *A. Experimental Setup*

The experimental platform featured a Universal Robot UR3 mounted on a fixed base adjacent to a $2.0m \times 0.56m$ workbench. The scene was monitored by a top-down Intel RealSense D435f camera. The workspace was populated with disassembled chair components.

### *B. Kinematic Efficiency and Planner Optimization*

Before finalizing the control architecture, we evaluated the trajectory execution of the BiTRRT and PRM* solvers.

- Trajectory Predictability: While BiTRRT successfully navigated the static cage limits, empirical observation revealed significant joint-space redundancy ("waving"). This unpredictable sweeping motion increased the manipulator's swept volume, raising the collision probability with the human operator. Switching to PRM* successfully eliminated these redundant rotations, resulting in highly direct and kinematically efficient toolpaths.
- Iterative Planning Success: We tested the iterative planning loop by intentionally blocking the UR3's target grasp pose. The PRM* solver successfully timed out after $2.0s$, preventing system hangs. Once the operator removed their hand, the subsequent

iteration immediately found the optimized path and initiated movement, proving the efficiency of the micro-planning FSM.

### C. Safety System Performance

To validate the system's compliance with the ISO/TS 15066 collaborative safety standard, the proposed architecture was subjected to 215 dynamic human-interruption trials. A critical metric for HRC safety is the "photon-to-halt" latency ($T_{total}$), which defines the total time elapsed from the physical camera shutter opening to the robot achieving zero mechanical velocity. We mathematically define the total system latency as a function of discrete perception, communication, and actuation delays:

$$T_{total} = T_{sensor} + T_{vision} + T_{update} + T_{halt} \quad (3)$$

where $T_{sensor}$represents the hardware acquisition delay of the RGB-D sensor capturing the physical frame, $T_{vision}$denotes the edge-computation latency required to process the MediaPipe skeleton extraction, execute the cascaded Kalman Filter, and calculate the "Hard-Connect" kinematic transformations, $T_{update}$ accounts for the middleware communication overhead necessary to serialize the data via ROS2 DDS and instantiate the geometric collision primitives within the MoveIt 2 $PlanningScene$ , and $T_{halt}$ is the mechanical braking time required for the physical joint motors to fully decelerate to $0.0\ rad/s$ upon receiving the $ABORT$ command.

To prevent ROS 2 network clock masking and ensure true edge-computation latency was recorded, a timestamp passthrough architecture was implemented. Under maximum operational kinematics—scaling the UR3's Tool Center Point (TCP) velocity ($V_{max}$) to $250\ mm/s$ and deceleration ($A_{max}$) to $1000\ mm/s^2$ , the total reaction latency ($T_{reaction}T_{sensor} + T_{vision} + T_{update}$) averaged $33.64\ ms$ . According to ISO/TS 15066, the total minimum safe stopping distance ($D_{stop}$) is the sum of the distance traveled during the constant-velocity reaction phase, and the distance traveled during mechanical braking($T_{halt}$):

$$D_{stop} = (V_{max} \cdot T_{reaction}) + \frac{{V_{max}}^2}{2A_{max}} \quad (4)$$

By applying the Safety Scale multiplier($\lambda = 1.5$) to the MediaPipe-extracted collision primitives, our spatial grounding algorithm projects a conservative volumetric buffer of approximately $80.0\ mm$ around the operator's arm. Because this engineered safety buffer strictly exceeds the absolute maximum empirical stopping distance of $39.66\ mm$, the proposed architecture provides a rigorous mathematical guarantee against physical collisions. Furthermore, implementing the "Hard-Connect" kinematic constraint reduced false-positive safety breaches (phantom stops caused by depth sensor occlusion on thin fingers) by 40% compared to raw MediaPipe data streams.

### D. End-to-End Task Completion

To evaluate the overall system capability in a dynamic manufacturing environment, 20 complete end-to-end chair assembly cycles were conducted. The Gemini 3 Flash-Preview model served as the cognitive agent, operating in a multimodal "listen-and-look" cycle. The agent was tasked with dynamically sorting disassembled components based on randomized natural language audio prompts (e.g., distinguishing between square and triangular plate assemblies).

The Generative AI integration demonstrated high deterministic reliability, achieving a 100% Intent Classification Accuracy. The system flawlessly parsed audio instructions, accurately inferred the required quantity of sub-components (e.g., four legs for a square plate versus three for a triangular plate), and spatially transformed the 2D visual bounding boxes into valid 3D Cartesian target coordinates with zero inverse kinematics (IK) solver failures.

To rigorously stress-test the vision-based safety layer, human operators intentionally interrupted the collaborative workspace at random intervals during the 20 execution cycles. The deterministic "Stop-Wait-Replan" FSM successfully intercepted all dynamic intrusions. The system achieved a 100% safety reliability score, recording 0 False Negatives (zero physical collisions) across all trials. Furthermore, the framework demonstrated a 100% autonomous recovery rate, smoothly recalculating and resuming the PRM* pick-and-place trajectories immediately following the mandatory 3.0-second safety clearance window, requiring zero manual system resets or LMMs re-prompting.

A supplementary video demonstrating the real-time collision avoidance and generative task execution can be viewed at: https://drive.google.com/drive/folders/1lIuM-PSAEXdgPDRWp1XxkEFaADuyKsV8?usp=share_link.

## VII. Conclusion

This paper presented a cohesive framework for Human-Robot Collaboration that bridges the gap between high-level Generative AI planning and low-level deterministic safety. By integrating Gemini 3 Flash-Preview with a robust vision-based safety layer, we demonstrated that cobots can perform flexible, reasoning-based tasks without compromising operator safety.

Our key technical contributions include:

1. Multimodal Logic Grounding:
   A "Listen-and-Look" agent that uses audio intent to strictly constrain visual search, reducing hallucinations in assembly tasks.
2. Robust "Caged" Perception:
   A safety architecture that combines static environment bounds with a Kalman-filtered skeletal tracker. The introduction of the "Hard-Connect" kinematic constraint significantly improves the detection reliability of human hands, a common failure point in vision-based safety.
3. Stability-First Control:
   A "Stop-Wait-Replan" execution strategy that prioritized system stability over speed, eliminating the oscillatory motion often seen in reactive planners.

Future work will focus on integrating force-torque sensing to allow for contact-rich collaboration (e.g., hand-overs) and optimizing the inference latency of the Large Multimodal Model to enable real-time, closed-loop reasoning.

## Acknowledgment

This work was supported in part by the "Higher Education Sprout Project" of National Taiwan Normal University and

the Ministry of Education (MOE), Taiwan; and in part by the National Science and Technology Council, Taiwan, under Grant NSTC 114-2221-E-003-022-MY2 and 114-2224-E-003-001.

## REFERENCES

[1] Chen-Chien Hsu, Pin-Jui Hwang, Wei-Yen Wang, Yin-Tien Wang, and Cheng-Kai Lu, "Vision-Based Mobile Collaborative Robot Incorporating a Multicamera Localization System," IEEE Sensors Journal, Vol. 23, no.187, pp. 21853-21861, Sep. 15, 2023.

[2] Pin-Jui Hwang, Chen-Chien Hsu, Po-Yung Chou, Wei-Yen Wang and Cheng-Hung Lin, "Vision-Based Learning from Demonstration System for Robot Arms," Sensors, Vol. 22, no. 7:2678, Mar. 31, 2022.

[3] Pin-Jui Hwang, Chen-Chien Hsu, and Wei-Yen Wang, "Development of a Mimic Robot: Learning from Demonstration Incorporating Object Detection and Multi-Action Recognition," IEEE Consumer Electronics Magazine, Vol. 9, no. 3, pp. 79-87, May 2020.

[4] Pin-Jui Hwang, Chen-Chien Hsu, Wei-Yen Wang, and Hsin-Han Chiang, "Robot Learning from Demonstration Based on Action and Object Recognition," IEEE International Conference on Consumer Electronics (ICCE 2020), Las Vegas, USA, Jan. 4-6, 2020.

[5] Google DeepMind, "Gemini 3 Flash-Preview," Google AI Studio, 2026. [Online]. Available: https://aistudio.google.com/

[6] Universal Robots A/S and FZI Forschungszentrum Informatik, "Universal Robots ROS2 Driver," GitHub Repository, 2026. [Online]. Available: https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver

[7] Intel Corporation, "Intel RealSense ROS2 Wrapper," GitHub Repository, 2026. [Online]. Available: https://github.com/IntelRealSense/realsense-ros.

[8] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics, vol. 7, no. 66, p. eabm6074, 2022.

[9] S. Chitta, I. Sucan, and S. Cousins, "MoveIt!: Fresh ingredients for an old PR2," IEEE Robotics & Automation Magazine, vol. 19, no. 1, pp. 84-85, 2012.

[10] C. Lugaresi et al., "MediaPipe: A framework for building perception pipelines," arXiv preprint arXiv:1906.08172, 2019.

[11] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," IEEE Robotics & Automation Magazine, vol. 19, no. 4, pp. 72-82, Dec. 2012.

[12] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," The International Journal of Robotics Research, vol. 30, no. 7, pp. 846-894, 2011.

[13] L. Jaillet, J. Cortés, and T. Siméon, "Transition-based RRT for path planning in continuous cost spaces," in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, 2008, pp. 2145-2150.