Collaborative Pick-and-Place Robot System

Summary

This project presents the design of a collaborative robot system for safe, fast, and maintainable industrial part handling. It covers architecture, vision-based perception, ROS 2-based control, human-robot interaction, and system validation — with a focus on simulation-driven development and long-term reliability. It serves as both a design document and a public demonstration of engineering capabilities in real-world robotics.

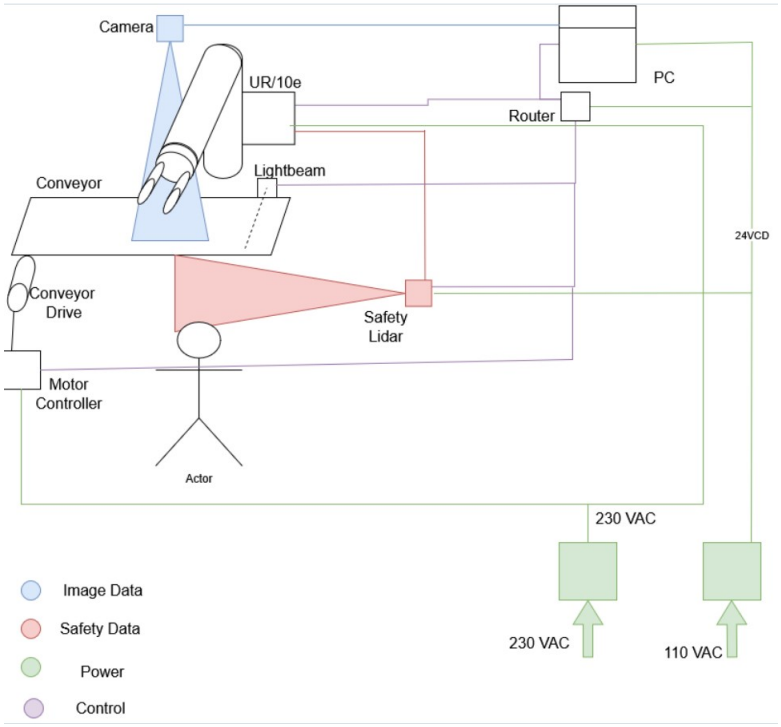


Figure 1. Functional block diagram showing perception, control, actuation, safety, and HMI layers.

About the Author

Rick Hudson is a full-stack robotics engineer focused on system design, perception, and real-time control. This project demonstrates design and analysis skills applicable to collaborative robotics, automation, and industrial HRI.

Email | Linkedin | GitHub

System Architecture

High-Level Flow

1. Vision identifies object
2. Grasp Planner finds grip pose
3. Arm moves to pick and then drop off point
4. Hand-off Module detects human
5. Gripper releases object
6. Loop, repeat, profit.

Explicit Assumptions

1. Maximum flexibility of parts is a priority, including fragile,articulated or flexible parts.
2. Parts have a proper grip point and handoff orientation.
3. Throughput is the major optimization goal.
4. Parts won't roll on the conveyor, are more or less flat and are spread out such that the robot can consistently grip them.

Functional Requirements

1. Object Detection & Recognition
    - The system shall detect and classify objects on a conveyor belt using vision sensors (e.g., RGB-D camera).
    - It shall identify the object's position and orientation with accuracy sufficient to lift the object by the designated grip point (to be validated) within ±5 mm and ±5°)
    - The conveyor shall stop when parts reach the end.

2. Pick-and-Place Operation
    - The robot shall grasp objects up to 1 kg without slip or pivot.
    - The robot shall have a working reach of at least 1 meter from its base.
    - The system shall position objects within the designated zone and at the desired orientation sufficient for human handoff (to be validated) within ±10 mm and ±10°.
    - The robot shall hand parts either by request or predetermined order and count.
    - The system shall request stop conveyor, advance or clear parts. (Assumed existing hardware.)
    - Parts shall not be dropped

3. Human-Robot Interaction
    - The robot shall detect the presence of a human coworker in the handoff zone using proximity or vision-based sensors.
    - The robot shall wait for an explicit human gesture (e.g., hand presence, tug on part, button press, or verbal cue) before releasing the object.

4. Collaborative Safety

    - The robot shall detect human presence within its workspace and adjust speed or stop motion as per ISO/TS 15066.
    - Emergency stop and pause/resume functionalities shall be accessible from within the robot and human workspaces.
5. Maintenance Modes
    - Adding new parts or part lists:
        i. Train new parts for visual recognition and verbal label (“Hand me a X”)
        ii. Train new path to part and grip orientation for robot
        iii. Train new handoff path and orientation for robot
    - Calibration
        i. Camera
        ii. Robot

Non-Functional Requirements

1. Performance
    - Cycle time per object (from detection to handoff) shall be under 5 seconds.
    - System uptime shall exceed 95% over a 24/7 operation schedule.
    - Training a new part should not take longer than 1 hour.
2. Scalability & Modularity
    - System architecture shall support integration of additional robots or sensor modules and additional up or down stream work cells.
    - Software should support easy updates and plug-and-play for new object types or conveyors.
3. Reliability & Maintenance
    - Mean time between failures (MTBF) should be >10,000 hours.
    - Diagnostics and error logging must be built-in for predictive maintenance.
4. Environmental Conditions
    - Operating temperature: 0°C to 45°C.
    - Must function reliably in dusty or noisy industrial environments.
    - IP54 or higher ingress protection for robot and sensors.
5. Compliance
    - Must conform to relevant safety standards (e.g., ISO 10218-1, ISO/TS 15066 for collaborative robotics).

Top-Level System Architecture

1. Perception Layer

- **RGB-D Camera (e.g., Basler blaze-101)**
    - Mounted above the conveyor for 3D object detection and localization.
- **Force/Torque Sensor**
    - Detects contact forces for part release.
- **Conveyor Stop Lightbeam**
    - Stops the conveyor if parts reach the end.

3. Actuation Layer

- **Collaborative Robotic Arm**
  - Payload ≥ 1 kg, reach ≥ 1 meter (UL 10e Collaborative Robot Arm, 10Kg)
- **Conveyor Motor & Motor Controller**
  - Speed control for conveyor belt.
- **Robot Gripper** (e.g., Robotiq 2F-85)
  - Robot end effector to grasp parts.

4. Human-Robot Interaction Layer

- **Touch Panel Interface**
  - For starting/stopping tasks, viewing system status, overrides.
- **Robot Pendant**
  - For calibration, new part training

5. Safety & Compliance Layer

- **Emergency Stop Buttons**
  - Dual accessible (robot + human side).
- **Safety Rated Lidar (e.g. SICK S300)**
  - For human detection and handoff zone monitoring.

7. Power, Wiring & Enclosure Layer

- **Power Supply**
  - 24V bus for sensors & PC
  - 480V for robot actuators
- **Industrial Enclosure**
  - Houses PC, controllers, relays, UPS, cooling fans (IP54 rated).
- **Wires and Cables**
  - All necessary wires, cables, plugs, connectors, crimps etc.
- **Cable Management**
  - Routed through the robot arm or external cable guides

8. Mechanical Layer

- **Conveyor Belt**
  - 24V bus for sensors & PC
  - 480V for robot actuators
- **Mounting Brackets, Bolts**
  - All necessary mounting components and hardware

Main Component Selection

1. Robotic Arm (Collaborative)

- **Model:** Universal Robots UR10e
- **Why:** 10 kg payload, 1300 mm reach, ISO/TS 15066 compliance, ROS-compatible.

2. Force/Torque Sensor

- **Model:** Robotiq FT 300-S
- **Why:** Seamlessly integrates with UR arms, precise force control for safe handoff.
- **Justification:** Enable delayless part hand off
- **Specs:** 1 lb

3. End-Effector / Gripper

- **Model:** Robotiq 2F-85 Adaptive Gripper
- **Why:**Programmable grip force and width, works well with varied object geometries (grippers close past 90°), plug-and-play with UR arms.
- **Specs:** 2 lbs, 50mm stroke, IP67

4. Vision System
- **Model**: blaze-101 Basler 3D camera
- **Why:** Color/Range camera, 640x480, 20 fps, Gigabit Ethernet, 67° x 51° FoV, 1.5m range, Global Shutter, <85ms Latency (per specs, see computation later in this doc), IP67

5. Proximity / Human Detection
- **Model:** SICK S300 Mini Safety Laser Scanner S30B-2011BA
- **Why:** Field-tested for collaborative applications, multiple zone-based slow-downs and stops.

6. Conveyor Motor Control
- **Model:** AB Powerflex 1/4 HP, 230 VAC Motor Controller
- **Why:** RS 485, 1.5 Amps

7. Conveyor Motor
- **Model:** Baldor 0.25 HP 230 VAC
- **Why:** AC induction motors last forever.

8. Processing Unit

- **Model:** Industrial PC with Intel i7 + discrete GPU
    - AIEdge-X80-NX8 NVIDIA Jetson Orin NX 6-Core, NVIDIA Ampere 1024 Core GPU

- **Why:** Handles vision, AI inference, and real-time control; ideal for ROS2 and neural network-based grasping.

9. Software Stack

- **Middleware:** ROS2 (Foxy/Humble)
    - For all control, perception, and planning nodes.
- **Motion Planning:** MoveIt 2
    - Seamless with UR5e and Robotiq grippers.
- **Vision Processing:** OpenCV + PCL
    - For segmentation, object pose estimation.
- **Grasp Planning (optional AI):** Dex-Net / GPD
    - For robust grasp pose predictions on novel objects.

10. Human Interface

- **HMI Panel:** Siemens KTP700 or UR teach pendant
- **Feedback:** RGB LED strips + buzzer module for state alerts

11. Conveyor Interface

- **Sensors:** Omron E3Z (for object arrival detection)
- **Interface:** EtherCAT or Modbus I/O (to sync with factory systems)

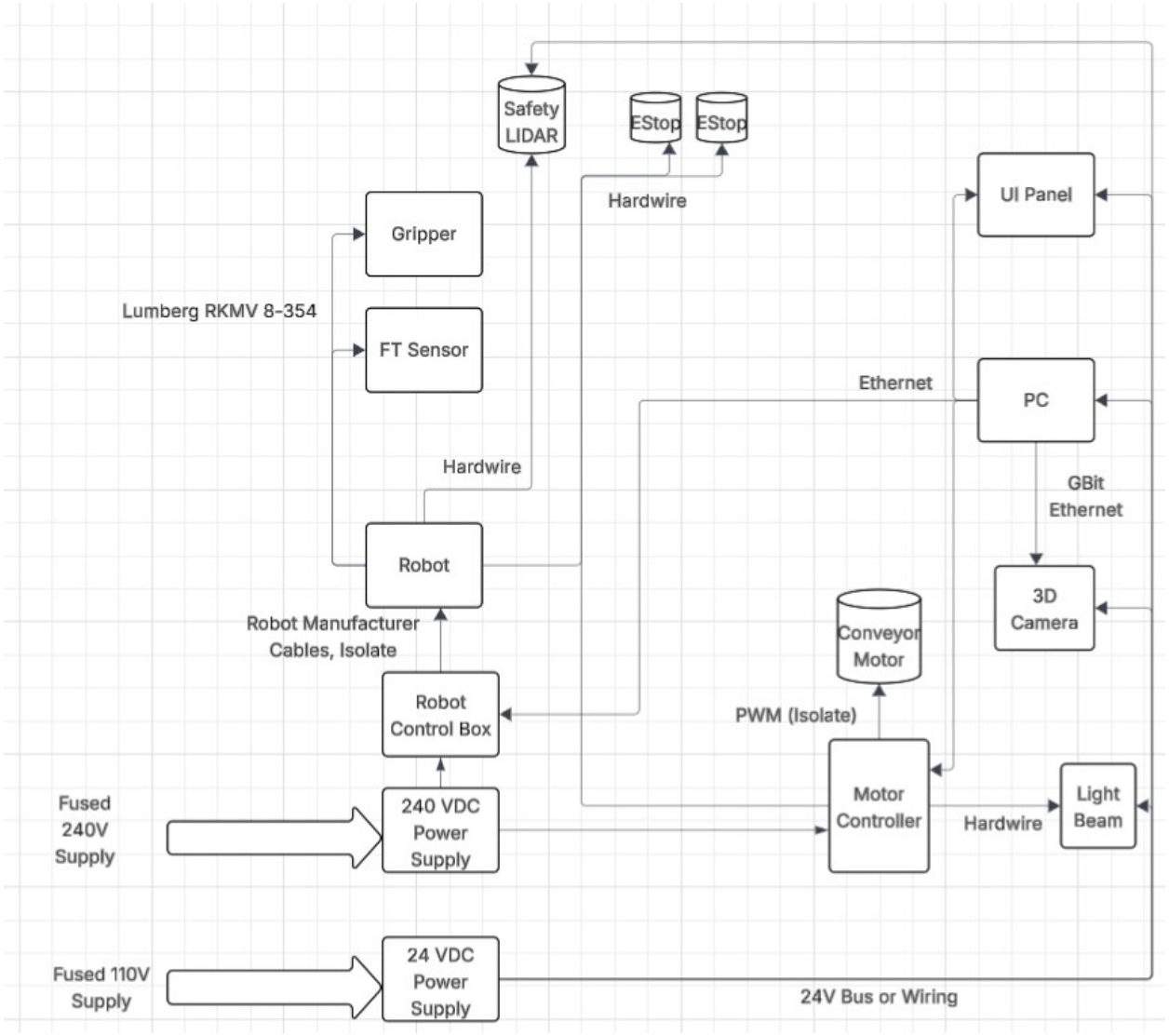
12. Safety Hardware

- **Emergency Stop:** Dayton 32W275 E-Stop (redundant placement)

13. Power Supply

- **Power Source:** 24V DC 500W PSU for sensors, PC; 230V AC line for robot arm.

Electrical Architecture



Calibration

Assumption: Tight accuracy tolerances justify rigorous calibration. The 3D camera (Basler blaze-101) is used during calibration, and final calibration is verified using physical gauges for binary PASS/NO PASS checks.

A. Intrinsic Camera Calibration (Not needed if using the blaze-101 3d camera)

Goal: Correct the camera's internal model (focal length, optical center, distortion).

Method:

- Capture checkerboard images (15–20 independently, identically distributed (IID) views).
- Compute intrinsic matrix using OpenCV camera_calibration.

Frequency: Once unless optics are moved or reassembled.

B. Camera-to-Robot (Extrinsic) Calibration

Goal: Align the 3D camera's coordinate system to the robot base frame.

Method:

- Mount a fiducial to the gripper or near the TCP (permanently, if possible).
- Move the robot to ~10 known IID poses.
- Record fiducial poses in camera frame.
- Use ROS2 handeye_calibration to solve:
  T_base_camera = T_base_ee * T_ee_target * T_target_camera
- **Tool**: ROS2 moveit_calibration or OpenCV scripts.

Verification:

- Place gauge on conveyor, measure with 3D camera
- Command robot to hover .3 mm above gauge.
  - Slip 0.2 mm gauge between gripper and object → should fit.
  - 0.4 mm gauge → should not fit (PASS/NO PASS criteria).
- **Frequency**: After moving the camera or robot.

C. Tool Center Point (TCP) Calibration

- **Goal:** Precisely locate the gripper's grasp center relative to the robot flange.

Method:

- Close the gripper. TCP point is center of the tip of the gripper (mark if not visually clear)
- Fix a 1mm point gauge (PG) to the conveyor; optionally reposition it to all 4 corners.
- Approach PG from varied IID orientations, stopping near contact.
- Use 0.2 mm and 0.4 mm feeler gauges to confirm uniform approach gap.

- Fit a least squares sphere to the contact points, then subtract 1.3 mm to account for gauge diameter and clearance.

Apply: Update robot TCP configuration.
Verification: Ensure that updated TCP consistently contacts the point gauge within PASS/NO PASS tolerance.

D. Conveyor-to-Robot Frame Calibration

Goal: Align conveyor object positions (seen in camera frame) to robot coordinates.

Method:

- Place fiducials on the conveyor and detect them with the 3D camera.
- Use TF transforms to align to the robot's base frame.
  Optional (for 2D cameras or high precision needs):
- Record predicted vs. actual X, Y, Z positions.
- Fit a polynomial surface to model and correct conveyor-to-robot discrepancies.

  Note: A similar technique allowed a ½ ton robot to write legibly with chalk on a blackboard.

E. Verification

All calibration stages are validated using 0.2 mm PASS, 0.4 mm NO PASS physical gauges.

F. Calibration Frequency

Calibration Type	When to Perform
Intrinsic Camera	Once if 2D camera used (unless camera moved/damaged)
Extrinsic	After moving camera or robot
TCP (Gripper)	On gripper change
Conveyor Alignment	As needed or after conveyor disturbance
Robot Joint Encoders	Rare (UR10e is self-calibrating)

Validation & Testing

Test Protocol: Reliable Picks

Objective

Validate that the proposed system requirements—specifically sensor pose estimation accuracy and robot motion precision—enable reliable detection and stable transport of representative customer parts. This addresses the highest-risk subsystem (grasp pipeline) and provides confidence that the system requirements support the intended function.

Assumptions:

- Target objects are non-trivial—potentially delicate, deformable, articulated, or lacking ideal grasp surfaces. Therefore, the grasping pipeline (vision → pose estimate → grasp execution) is the primary technical risk and must be validated early.
- Over-capable hardware (e.g., high-resolution 3D cameras, 7-DOF robot, precise calibration) is available to simulate system constraints in a general-purpose validation environment.
- Testing will involve injecting noise and locking unused joints to match the proposed system specification.
- If the production feeder is unavailable, manual part placement and conveyor advancement are acceptable.

Test Setup

- Conveyor, calibrated 3D camera, robot, and gripper mounted in production configuration.
- Proposed production end-effectors installed on the robot.
- 3D camera positioned over the drop-off zone to monitor grip integrity during motion.
- A representative set of customer parts (≤1 kg each) is available.

Test Procedure

- Place test objects in varied orientations on the conveyor. Move parts to the start position.
- Capture and log 3D camera data.
- Generate pose estimates
    - Ground truth pose annotations (may be manual or use precision automated pose system)
    - Pseudo system estimate using GT + noise envelope being validated
- For each detected object:

    - Execute grasp plan based on system estimate plus noise envelope
        - Record “original” grasp orientation (part vs gripper)
    - Move the object along a representative path simulating production dynamics to the drop-off zone (ensuring accelerations simulate realistic operation)
    - Capture final orientation, compare to original
- Log success: object picked, held, and transported without drop or slip
    - Target meets goal, see table.

Note: If grip slippage causes the part to be misaligned at drop-off, that is recorded. Determining what counts as "acceptable slip" may require follow-up user feedback or post-test handling, which is outside the scope of this protocol. Nonetheless, this test provides useful insight into slip behaviors and should readily expand to develop a slip tolerance for the system requirements.

Metrics to Log

Metric	Target Threshold
Grasp Success	True/False
Pose Accuracy	≤ ±3 mm / ±5°
Grasp Motion Accuracy	≤ ±3 mm / ±5°
Final Rate	**Grasp + Transport Success Rate** | **≥90%**

Notes

- Enables generation of “success vs. pose accuracy” curves to identify tolerances and optimize drop-off strategies.
- Separates robot and pose accuracy- while related they are not in the same space
- Directly supports early validation of the most failure-prone subsystem.
- Produces high-quality labeled data and a well-characterized noise envelope for future machine learning model development or benchmarking.
- If destructible parts are being moved, 10% droppage may be unacceptable.
- Asking the grippers be included in the Validation blurs system requirements with verification, the assumption is that it’s not a large ask.

Validation Apparatus

Critical Components

Hardware:

1. **Over-Capable Robot Arm (7-DOF or better)**
    - Supports joint constraints and application of known noise envelopes.
    - Must support precise, repeatable motion and configurable kinematic limits.
2. **High-Resolution 3D Camera**
    - Positioned above the pick zone to capture raw scene data.
    - Must support generating precise ground truth pose estimates.
3. **Gripper with Production End-Effector**
    - Must match the form and actuation method of the proposed design.
    - Important for evaluating real-world part handling and grasp-induced orientation change.
4. **Drop-Off Zone Camera**
    - Mounted to monitor part orientation after transport.
    - Used to detect slip or orientation drift by comparing final pose to initial grasp orientation logged during execution.
5. **Conveyor or Fixture Surface**
    - To place parts in varied, realistic poses.
    - May be manually operated if a feeder system is unavailable.

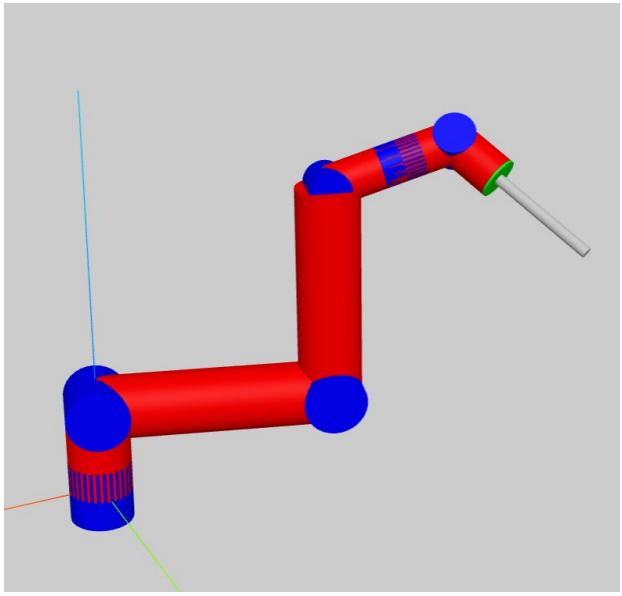
Software:

1. **Pose Estimation Pipeline**
    - Receives raw camera input and produces estimated part poses.
    - Includes noise injection and resolution degradation to simulate proposed system specs.
2. **Ground Truth Labeling Tool**
    - Allows manual or semi-automated annotation of object poses for GT comparison.
    - May use full-res camera data and known geometries.
3. **Motion Execution & Logging System**
    - Sends grasp and movement commands to the robot.
    - Logs pose data, gripper status, and motion trajectory.
4. **Analysis Framework**
    - Compares estimated vs ground truth pose (position and orientation error).
    - Evaluates grasp success, slip detection, and transport stability.
    - Generates pass/fail and “success vs. error” curves.

Summary:

The test apparatus combines flexible hardware with simulated constraints and a software stack capable of injecting sensor noise, recording GT data, executing grasps, and analyzing performance. This setup enables precise measurement of whether the proposed system requirements will support the intended operation on representative parts.

Robot Model



Visualization of URDF

Kinematic Structure

- **DOF (Degrees of Freedom):** 6 (all revolute)
- **Joint Types:** Revolute (R)
- **Kinematic Chain:** Base → Shoulder → Elbow → Wrist 1 → Wrist 2 → Wrist 3 → RTS→ End-effector

Links 0 (base) to 6(end effector)
Z Axis within rotation (since all are revolute)
X Axis along shortest distance from Z-1 to Z, with the origin at Z
Y Found using right hand rule

Denavit-Hartenberg (DH) Parameters
URDF Link
Approximate (mm):

Joint	a (|X touches Z-1|)	α (Z twist)	d (Z offset)	θ (Z angle)
J1	0	π/2	162.5	θ1
J2	-425	0	0	θ2

J3	-392.2	0	0	θ3
J4	0	π/2	133.3	θ4
J5	0	-π/2	99.7	θ5
J6	0	0	99.6	θ6

Force/Torque Sensor: Disc 41.5 mm thick, coaxial with last robot link (green)
Gripper (white)

Troubleshooting

Step 1: Define the symptoms of inaccuracy to guide troubleshooting. (IE- what do we know)

- Can we reproduce the error?
- Is it **missing the object** entirely?
- Is the **gripper misaligned** during grasp?
- Is the object being **dropped in the wrong handoff spot**?
- Is it **inconsistent** (random error) or **systematic** (always wrong in one direction)?

Step 2: Attempt to isolate problem area(s) by checking mid-pipeline signals

- General
    - Check if calibration passes
        - Verify joint (correct, commanded, sensed, actual) angles
    - Explicitly verify coordinate frames
        - (World, base, tool tip, camera, object)
    - For true “random occurrences”, set a halt condition if the problem is detected to capture it in real time.
        - Trigger a log of any normally unlogged data that may be of assistance (joints, commands, etc)
- Perception pipeline
    - Is the detected pose correct? Possibly use the GT images to check
        - Visualize the incoming image and range data
- Control pipeline
    - If the robot is given a destination pose, does it reach it to the goal precision?
        - Compare ‘correct’ joint angles (independently determined) to commanded/actual/sensed.
        - Move the commanded point about the workspace if there seems to be a location-specific issue
    - If it’s a dynamic issue, verify the trajectory
        - Visualize as a movie, if possible
        - Watch for impossible or garbage (inf) movement commands outside the robot operating envelope
    - I purposely didn’t define if the conveyor moves during pickup
        - If so, check if robot can pick up static objects but fails on moving objects
        - Compare resultant trajectory to part location
            - Watch for lag between sensed (camera), correct, commanded and sensed (encoder) robot positions during the dynamic pickup
            - Might use a high speed camera to verify
- Mechanical subsystem
    - If the robot move seems to be correct, but the end point is still off
        - Verify the end effector dimensions.
        - Verify the orientation/wiring of encoders and motor windings

Step 3: Fix it!

- Software fixes are easiest (hopefully)
    - Directly address if a simple bug
    - Out of range errors may require changing the inverse kinematics to use quaternions or address other precision limits
- Robotics Issues
    - Joint compliance tuning
    - Dynamic trajectory near feasible but impossible (physics model error or hard limit reached)
    - Recalibrate, reteach tool tip
- Located issue is misconstruction of existing design
    - Directly address if a simple wiring, alignment, etc problem
    - Replace faulty parts as needed
- Issue is design flaw
    - Is this a specification error or an implementation error?
    - Determine scope of minimal correction
- Consider if any of these issues may recur in the field and address
    - Logging and diagnostic improvements to detect earlier if likely to recurr

Step 4: Verify the fix

- Redo accuracy test, particularly ones that failed
    - If random, attempt to do enough “suspected” behaviors to verify issues is gone
    - Check for multiple bugs

Appendix

Major Sensor Bandwidth

blaze-101 Basler 3D camera

- **Resolution:** 640 × 480
- **Frame rate:** 20 fps
- **ColorIntensity:** 24 bits per pixel
- **Depth Precision:** 16 bits per pixel
- **Bandwidth (RGB):** 20*(24+16)*640*480= 245.8 Mbps

SICK S300 Mini Safety Laser Scanner S30B-2011BA

- **Resolution:** 270° arc sampled at 0.5° increments
- Zone Processing Done on Board
- **RS-422** has <= 500kBaud transmission rate, this is not significant.
- Onboard processing for safety zones, range data is not transmitted.

Sensing-to-Actuation Latency

The sensing to actuation is the mean ~43 ms for the camera to take an image and transmit it, the ~110 ms for the image processing and AI grasp planner, the ~25 ms for the trajectory generation, ~5 ms for command transmission, and then the arm response, which is maybe ~15 ms of lag but then the motion, which could be … 2 seconds?
Total perception-to-actuation latency is ~198 ms before motion begins. Motion duration is typically sub-second depending on travel distance.

Various System Latencies

1. Conveyor Moves Parts into Range'

- Moves at 250 mm / s, assume parts arrive 1s spacing

2. 3D Camera Exposure

- **Color Depth camera (**blaze-101 Basler 3D camera**):**
  - Mean Subsample Delay: ½ of interframe time 50 ms (20 fps)= 25ms
  - Exposure Time: 5 ms (adjust lighting to achieve this if important)
  - 1000 Mbit/s Gigabit Ethernet transferring 640x480 pixels * 5 bytes/pixel =12.3m Mb: 13 ms
- **Subtotal:** 25+5+13 = **43 ms**

3. Pose Estimation

- Processing the image/range data to detect and localize objects:
  - Depth + segmentation + centroid estimation: ~20–40 ms
- AI-based grasp planner (GPD): 50–100 ms
- **Subtotal: ~110ms**

4. Trajectory Generation

- Inverse kinematics + collision-free path (MoveIt 2): ~10–30 ms
- **Subtotal: ~25 ms**

5. Command Transmission

- ROS2 DDS message latency (Ethernet): ~5 ms
- **Subtotal: ~5 ms**

6. Robotic Arm Move to Grab

- UR5e command processing + joint motor actuation start:
  - Control loop is 500 Hz (2 ms loop time)

    - Mech latency for motors to start moving: ~10–20 ms
- Grasping motion estimated at 0.25m travel at 250 mm/s with linear accel/decel profile.
- **Subtotal: ~15 ms lag, 2000 motion**

Estimated Latency and Cycle Time

Component	Time (ms)
Conveyor	~1000
3D Camera	~43
Pose Estimate	~110
Robot Wait to Grasp	~15 lag, 500 transit
End Effector Close	~15 lag, 100 transit
Robot to Drop	~15 lag, 2000 transit
Human	~200 lag, 500 pull part
End Effector Open	~15 lag, 100 transit
Robot to Grasp Wait	~15, 2000 transit

Power Budget

The big users are the motors and the PC doing image processing.

Motors can consume more power for short bursts, and since the robot is backdrivable, it can be quite dependent on pose and loading even in static conditions. Typical line-power motor loading is cited at 4-7 times rated, but waveshaping controllers limit that to closer to 20% over rating. Regardless, using motor rated fusing/ breakers is required.

Power Budget Estimate

Component	Voltage	Current Draw	Nominal Power (W)	Peak Power (W)
UR10e Robot Arm	230V AC	2.5 A	200 W	600W
Force/Torque Sensor	Private Bus supplied by 230V	0.2 A	5 W	5W
Robotiq 2F-85 Gripper	Private Bus supplied by 230V	1 A	12 W	25W
Conveyor Motor/ Motor Controller	230V AC	1.5 A	100 W	276W
Total 230 VAC			**317 W**	**900W**
NVIDIA Jetson AGX Orin	24V DC	1 A	25W	75W
Safety LIDAR (SICK S300)	24V DC	0.3 A	8 W	40W
Basler Blaze 101 3D Camera	24 V DC	0.6 A	15 W	55W
UI Panel	24 V DC	1 A	25 W	25W
Router / Switch (Ethernet)	24V DC	0.5 A	12 W	12W
Misc Sensors / IO	24V DC	0.5 A	12 W	12W

Total 24 VDC	**97W**	**219W**
Total Power	**414W**	**1119W**