**IFARLAB — Intelligent Factory and Robotics Laboratory**

Eskişehir Osmangazi University

# Human–Robot Collaboration — Operator-Confirmed Screwdriving

Sequential collaboration, digital I/O handshake and planning-scene tool handling

*Table 1: Scenario at a glance.*

| Item | Value |
|---|---|
| Task | Four-screw assembly operation, each screw released by the operator |
| Robot | UR10e on a Festo linear axis (single-arm scenario) |
| End effector | OnRobot 2FG14 parallel gripper on the free slot of an OnRobot Dual Quick Changer |
| Tool | Electric screwdriver picked from a stand, driven through a UR digital output |
| Operator interface | Green push-button on the robot control box, read as a digital input |
| Collaboration model | Sequential — the robot is motionless while it waits for confirmation |
| Maturity | Runs end-to-end on the physical cell; screwdriving is open loop (no torque feedback) |

## 1. What this use case demonstrates

Most assembly steps are not candidates for full automation, and not candidates for manual work either. They are candidates for a division of labour in which the robot performs the repetitive, precise motion and a human keeps the decision. This use case implements exactly that division for a screwdriving operation.

In one tour the arm picks the electric screwdriver out of its stand, takes a screw, positions itself above the hole and stops. Nothing further happens until the operator presses the green button. On confirmation the screwdriver is switched on through a digital output, the arm descends at a tenth of its normal speed, the motor is switched off, and the arm retracts. After the fourth screw the tool is returned to its stand.

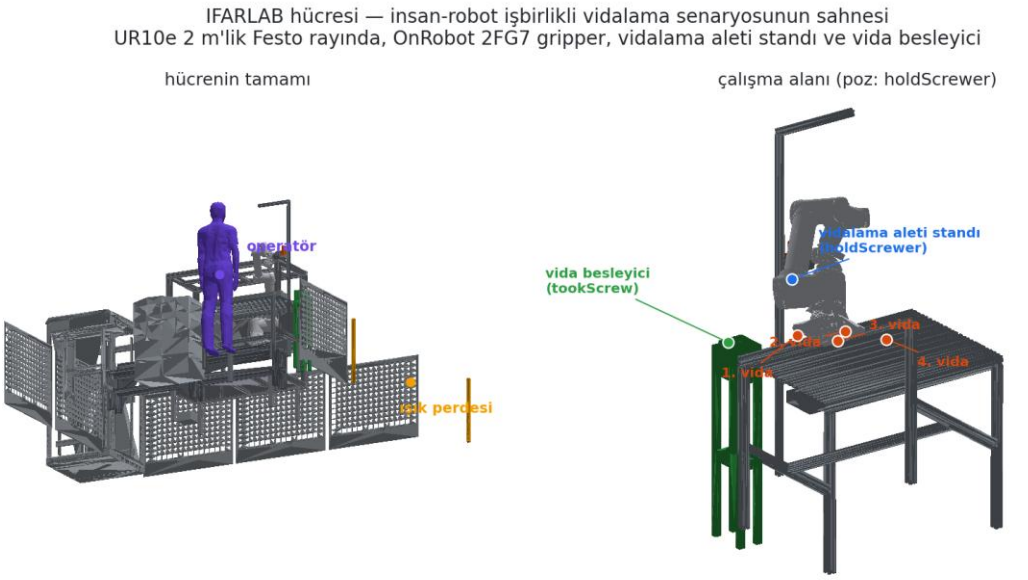



*Figure 1: The workstation: tool stand, screw feeder, assembly table, operator mannequin and light curtain are all separate links in the cell model, so the planner sees them as obstacles.*

## 2. Cell configuration

*Table 2: Hardware used in this scenario.*

| Component | Specification |
|---|---|
| Robot | UR10e; the linear axis is part of the planning chain but is held at a fixed position of 1.85 m throughout this scenario |
| Gripper | OnRobot 2FG14 parallel gripper on the free slot of the OnRobot Dual Quick Changer; commanded directly through its own trajectory controller |
| Tool | Electric screwdriver, 2.4 kg, standing in a fixed stand; grasped by the gripper rather than permanently mounted |
| Operator station | Green confirmation button wired to a standard digital input of the robot control box; the screwdriver motor is wired to a digital output |
| Safety equipment | Light curtain and physical cell barriers (see Section 6) |

## 3. The collaboration sequence

One tour consists of 48 steps: 27 motions and 21 events. The structure is a tool pick-up, four identical screw cycles, and a tool return. Each screw cycle is an approach–process pair: the arm travels at normal speed to a stand-off pose above the screw, and the descent to tightening depth is performed at a tenth of that speed.

*Table 3: One tour, stage by stage.*

| Stage | Motions | Events |
|---|---|---|
| Preparation | Gripper opened fully | Recovers from an interrupted previous tour |
| Tool pick-up | Front of stand → above stand → hold | Gripper closes, tool attached to the planning scene |
| Screw pick-up | Safe waypoints → feeder → out of feeder | — |
| Screw 1–4 | Stand-off → descent → retract, per screw | Wait for the green button, motor on, 1.5 s seating wait, motor off |
| Tool return | Back to the stand | Tool detached from the planning scene, then gripper opens |
| Shutdown | Return to the home posture | Both screwdriver outputs are dropped unconditionally |

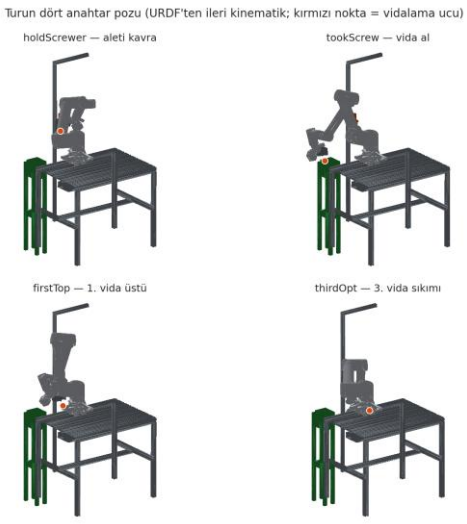



*Figure 2: Four key postures of the tour, drawn from the cell model; the marked point is the centre of the attached screwdriver tip.*

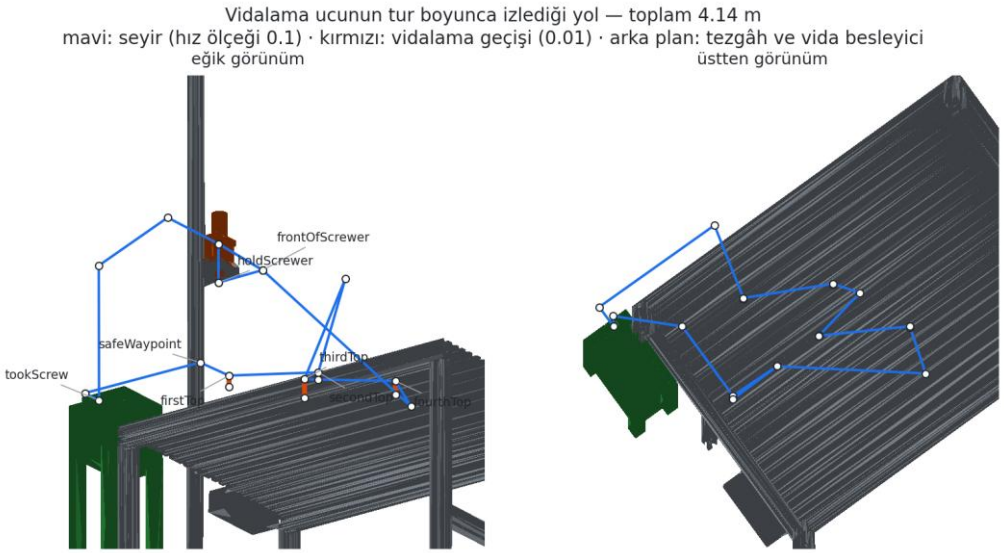



*Figure 3: Path traced by the screwdriver tip over one tour (total 4.14 m). The highlighted segments are the screwing descents, executed at one tenth of the travel speed.*

The measured descent depths differ per screw — 30.9, 19.1, 51.7 and 37.0 mm — because the stand-off poses were taught at the robot and the holes sit at different heights. At the reduced speed those descents take 1.2 to 2.9 s instead of 0.1 to 0.3 s; the slower motion is the point, not a side effect.

## 4. The digital handshake

The operator dialogue runs over the robot's own digital I/O rather than over an industrial bus or a separate PLC, which keeps the installation simple. Because pin labels configured on the teach pendant are not transmitted over the real-time interface, the pin map was measured once in the cell with a dedicated utility and is documented in the repository.

*Table 4: Measured input/output map.*

| Signal | Channel | Use |
|---|---|---|
| Screwdriver tighten | Digital output 0 | Held high for the whole descent |
| Screwdriver loosen | Digital output 1 | Wired and reserved, not used in the current tour |
| Green button | Digital input 7 | Operator confirmation, read as a rising edge |
| Red / white buttons | Digital inputs 6 and 5 | Wired and reserved |

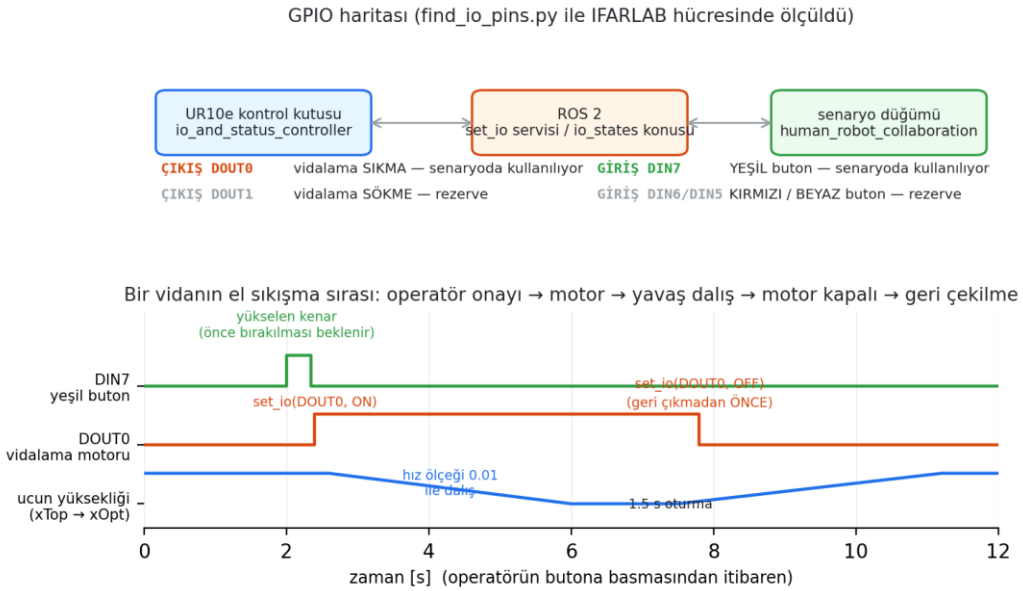



*Figure 4: Above: the pin map measured in the cell. Below: the handshake of one screw — operator confirmation, motor on, slow descent, motor off, retract.*

Reading the button as a rising edge matters: a button that is already held down when the arm arrives does not release the next screw, so one press can never trigger two operations. The wait has no timeout — the robot stands still indefinitely and never screws on its own initiative.

**Simulation behaves differently, by design.** Mock hardware always reports zero on the digital inputs, so a scenario that waited for a button would hang forever. The node detects mock hardware automatically and skips the screwing steps, optionally substituting a fixed delay, which lets the motion sequence be validated without the cell.

## 5. Handling the tool in the planning scene

When the gripper closes on the screwdriver, the robot's effective geometry changes: it now carries a 2.4 kg object that extends well beyond the gripper. The scenario therefore adds the tool to the planning scene as a collision body and attaches it to the gripper link with the matching mass, so every subsequent motion is planned around the tool rather than around the bare gripper.

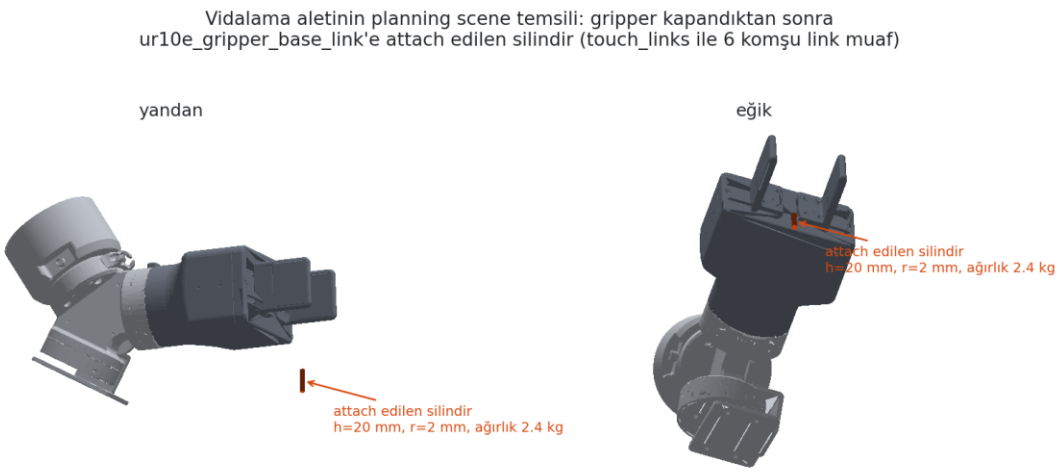



*Figure 5: The tool body attached to the gripper after the gripper closes; from this moment the planner carries it with the arm.*

The ordering is strict and is enforced in the code: attach after the gripper closes, detach before it opens. Reversed, the planning scene and the physical state disagree — either the tool is held but invisible to the planner, or it has been released and every later plan still avoids a body hanging in mid-air. Contact between the gripper and the attached tool is excluded from collision checking through an explicit touch-link list, otherwise the grasp itself would be reported as a collision.

Clearances in this workstation are tight, and they were computed from the cell geometry: the narrowest is 2.3 mm between the gripper base and the assembly table at the tool-holding pose, followed by 32.2 mm at the screw feeder and 32.8 mm at the third screw. This is why the 4 cm link padding used in the inspection scenarios cannot be reused here — it would make the tool-holding pose unplannable.

## 6. Safety framing

This point is stated explicitly because it determines how the cell may be operated. The green button is a process confirmation, not a safety function. It tells the robot that the operator is ready; it is not a protective stop and is not part of a safety-rated chain.

Operator protection in the current installation is physical: the light curtain and the cell barriers, together with the fact that the robot is motionless whenever it is waiting. The software contributes two guarantees: the screwdriver output is switched off before the arm retracts, and both outputs are dropped unconditionally on every exit path, including an interrupted or failed run.

A collaborative installation intended for production would add a safety-rated speed and separation function to this picture. The present system is a research and demonstration setup, and the distinction between a process handshake and a safety function is kept visible rather than blurred.

## 7. Measured and estimated timing

*Table 5: Cycle-time breakdown. Roughly one third of the cycle is operator wait plus deliberate slow motion.*

| Contribution | Duration | Basis |
|---|---|---|
| Travel motions at normal speed | ≈ 44 s | Computed from joint limits and the planned path |
| Screwing descents and retracts | ≈ 16 s | Eight motions at one tenth speed |
| Seating waits | 6 s | Four screws × 1.5 s, fixed in the sequence |
| Operator confirmations | ≈ 12 s | Assumption of 3 s per press — not measured |
| Fixed inter-step settling | 24 s | 0.5 s after each of the 48 steps |

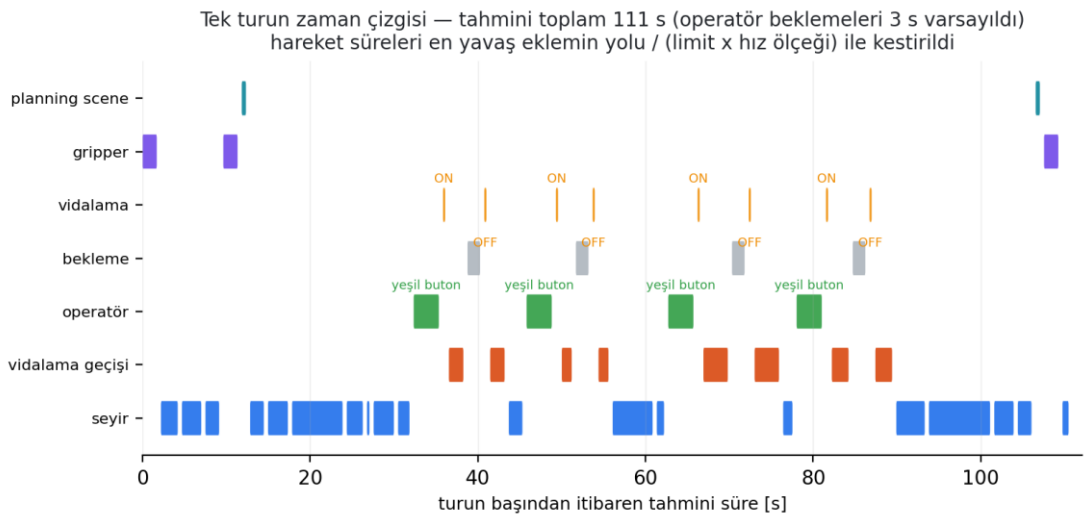



*Figure 6: Timeline of one tour; motion durations are computed from the joint limits and the operator waits are assumed at three seconds each.*

## 8. Current limitations

- Screwdriving is open loop: a fixed depth and a fixed seating time, with no torque or depth feedback. Reserved digital inputs exist for a tool-ready signal, which is the natural next step.
- The tool is represented in the planning scene by a simple collision body rather than its true volume; this is acceptable only because the tool-carrying motions follow taught waypoints with little free planning.
- The operator mannequin is modelled but excluded from collision checking against the moving arm links, so operator protection is physical rather than planner-enforced.
- The target poses are taught joint configurations hard-coded in the scenario; a part variant means re-teaching. Converting them to computed poses from a part model is the planned step.
- One launch executes exactly one tour and then shuts itself down; continuous operation would require the tour to be offered as a cancellable goal rather than a script.

## 9. Running the scenario

*The same two steps sit behind the "Human-Robot Collaboration" button of the cell dashboard.*

```
# 1) Cell, driver and motion planning, in the gripper configuration
ros2 launch my_robot_cell_control hil_test_whole_unified.launch.py use_gripper:=true

# 2) One collaboration tour
ros2 launch pymoveit2_real human_robot_collaboration_scenario.launch.py

# Dry run without the cell (digital I/O disables itself under mock hardware)
ros2 launch pymoveit2_real human_robot_collaboration_scenario.launch.py fake_button_delay:=2.0
```

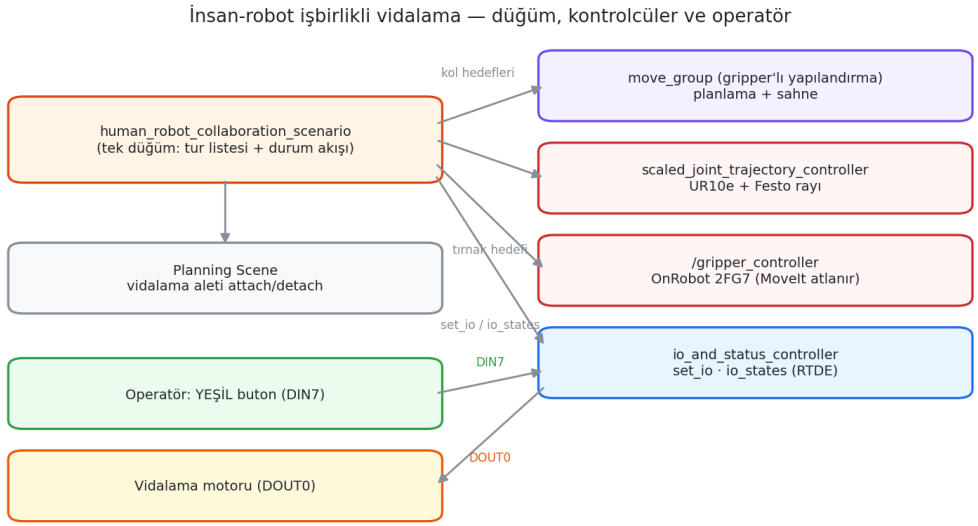



*Figure 7: System architecture: a single scenario node; arm goals through motion planning, the gripper straight to its own controller, and the operator dialogue over the control box digital I/O.*