Machine assembly & UX - what is so difficult?

Felix von Drigalski

Senior Researcher, OMRON SINIC X Corporation

2019/11/02 @ 1st MoveIt Workshop, Macau

OMRON

Agenda

1. World Robot Summit 2018 Assembly task

2. UX / Feedback from users

3. Conclusion

Self-introduction

Felix von Drigalski

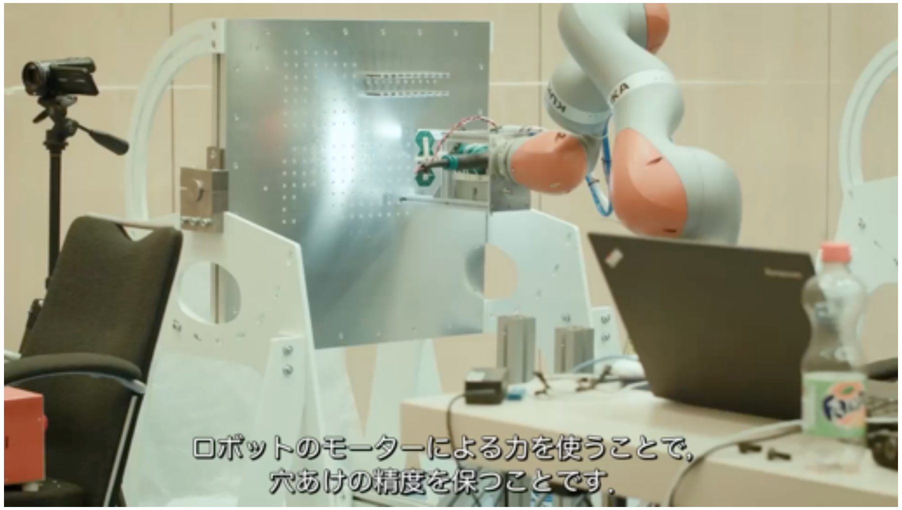
OMRON SINIC X Corporation
Robotics Group
Senior Researcher

2007-2013 Dual degree Mech. Eng. @ KIT (Germany) and INSA Lyon (France)
2015-2018 PhD Inf. Science @ NAIST (Japan): “Textile recognition and manipulation using tactile sensing based on active perception”
Since 2018 Robotics Group @ OMRON SINIC X Corporation

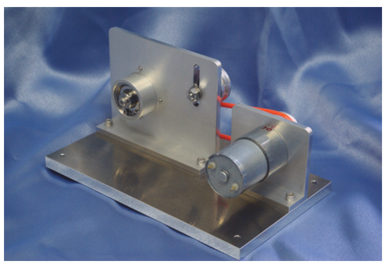
2016 ICRA, Airbus Shopfloor Challenge (1st place)
2017 Amazon Robotics Challenge (Finalist, 6th place)
2018 World Robot Summit Assembly Challenge (4th place, Special Award SICE)

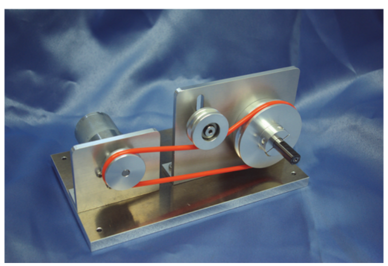
Self-introduction

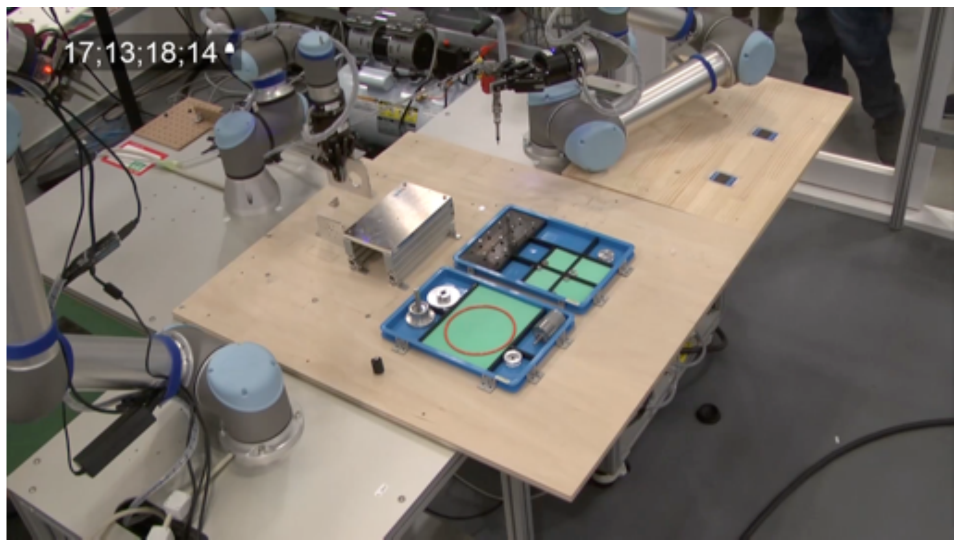
Airbus Shopfloor Challenge 2016 (1st place)



Amazon Robotics Challenge 2017 (6th, Finalist)



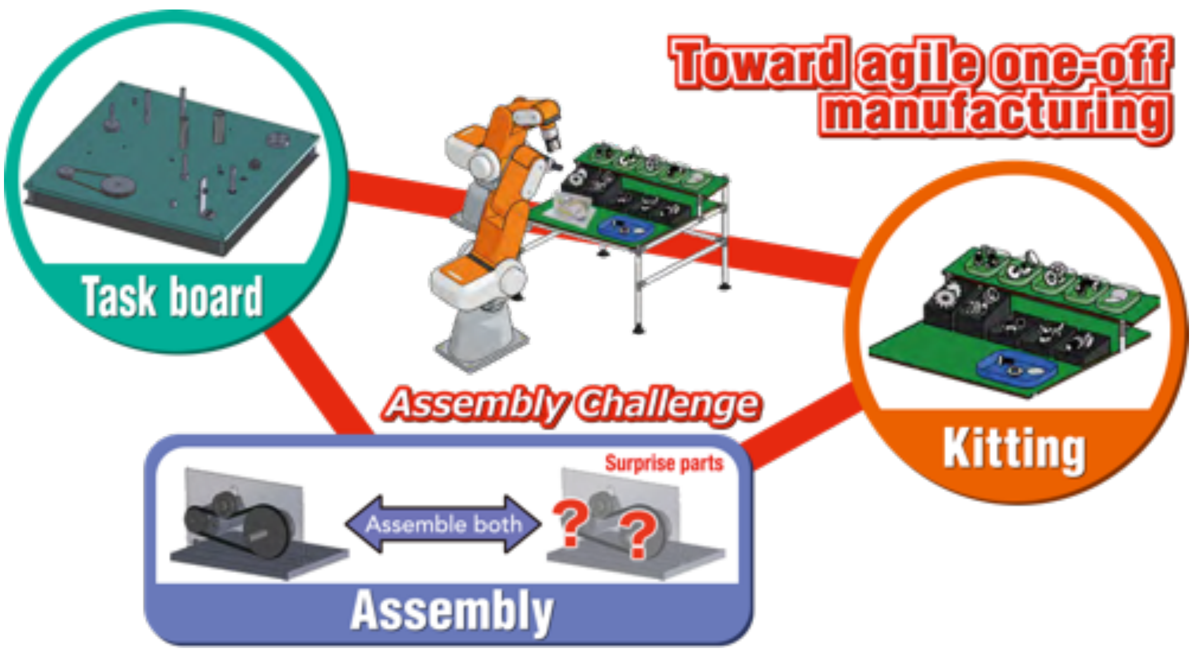


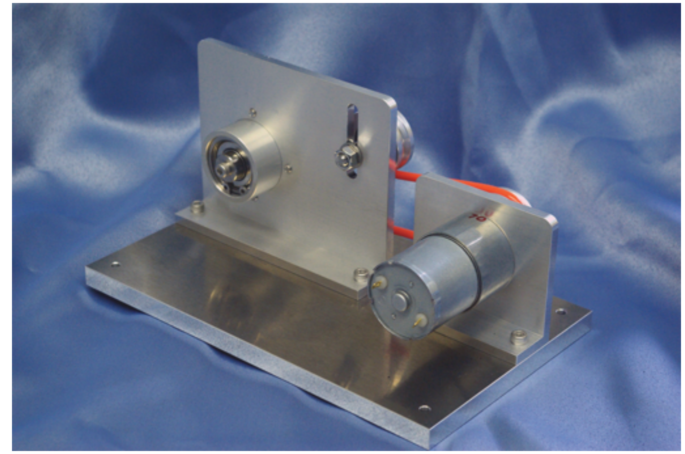


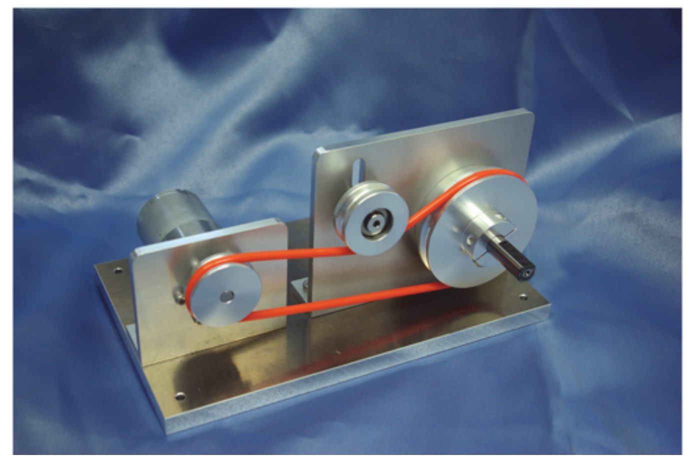
World Robot Summit Assembly Challenge 2018 (4th place, SICE Special Award)

World Robot Summit Assembly Challenge

World Robot Summit

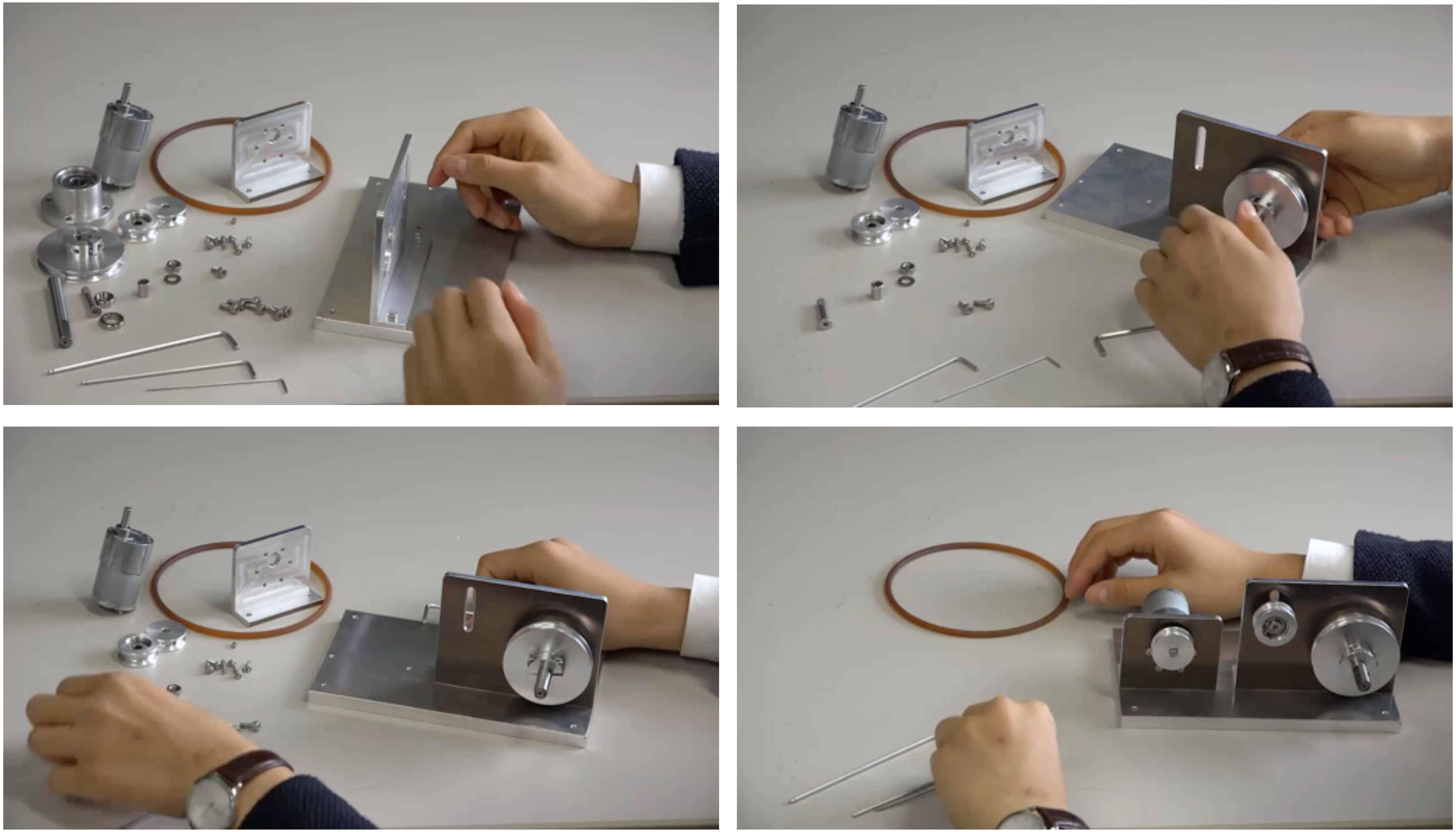






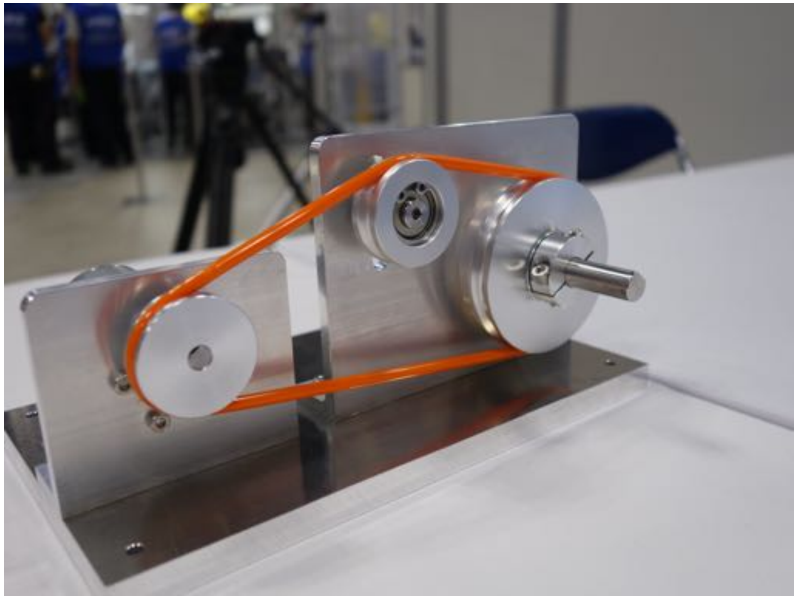
World Robot Summit Assembly Challenge

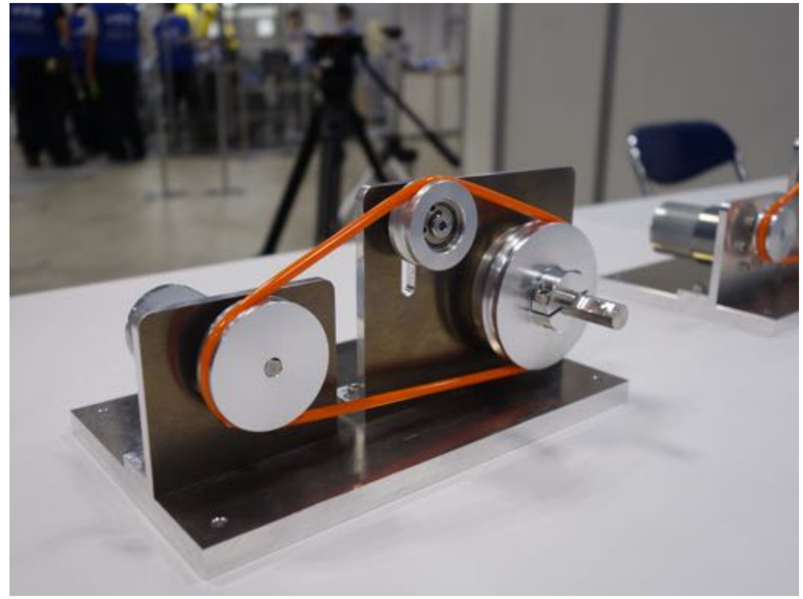
Human demonstration

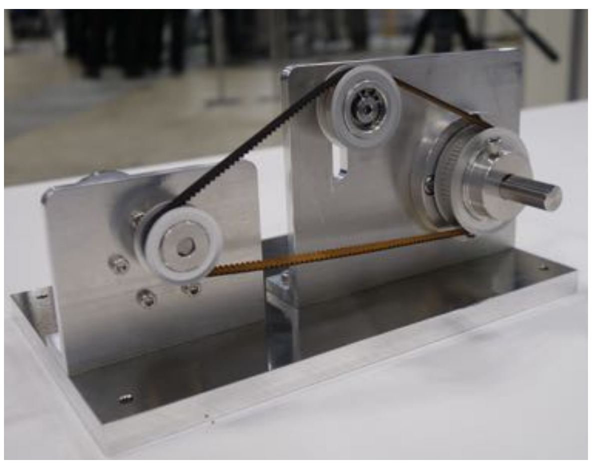


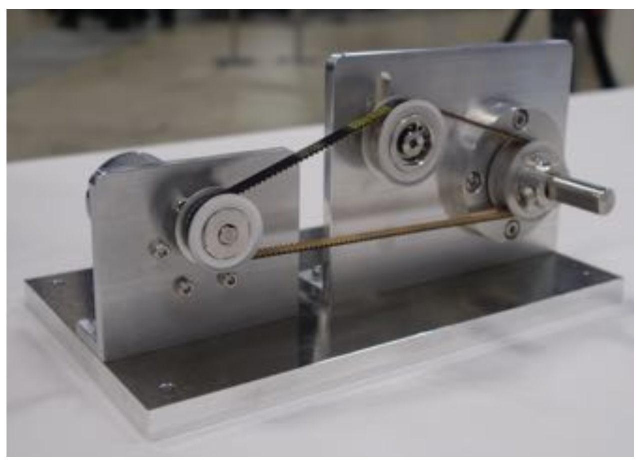
Surprise parts

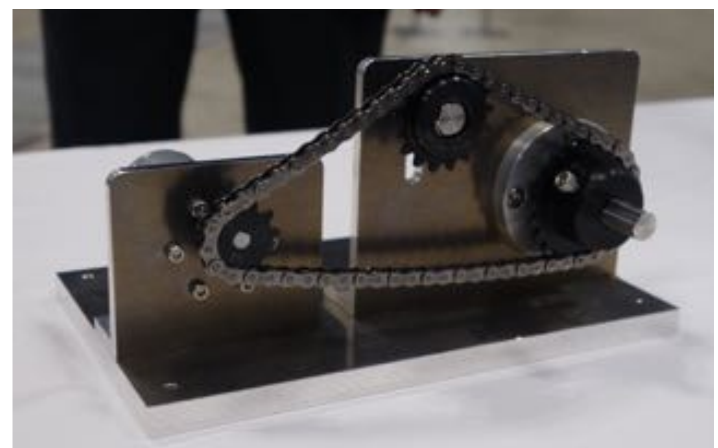
Announced 1 day in advance







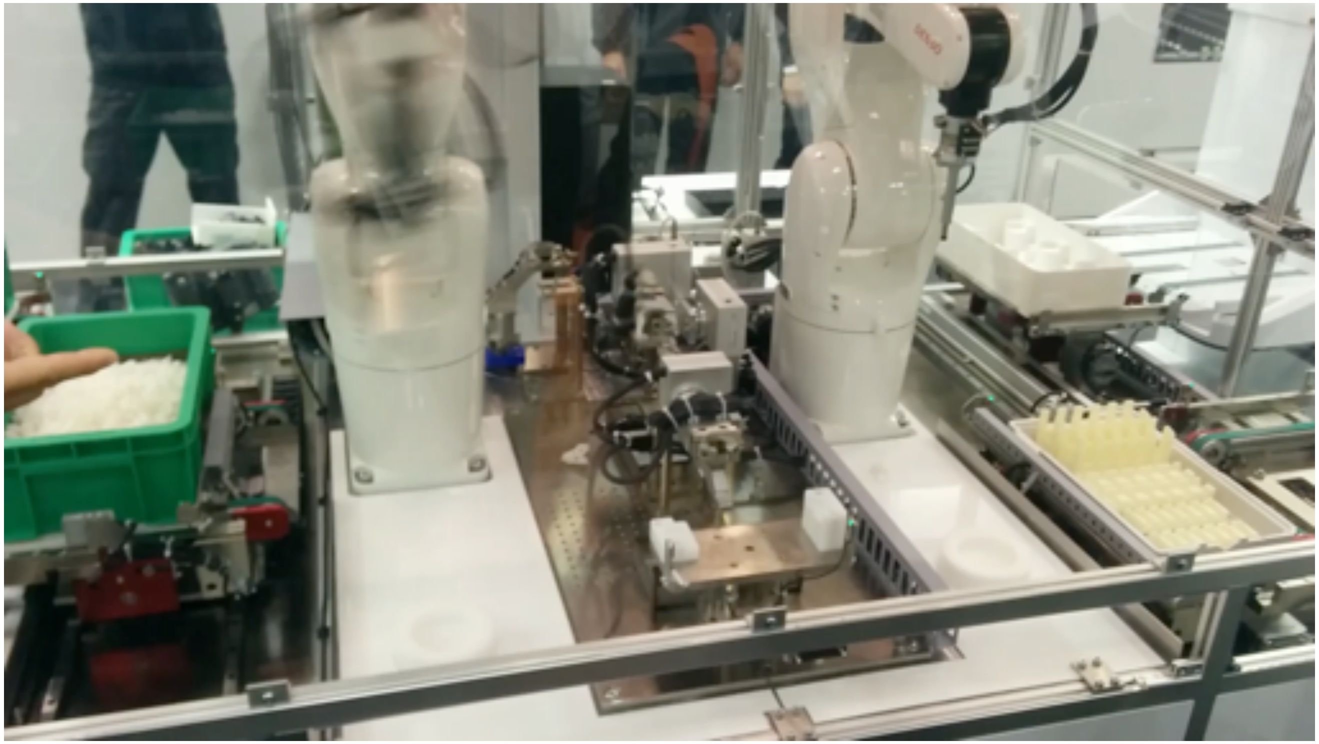




World Robot Summit Assembly Challenge

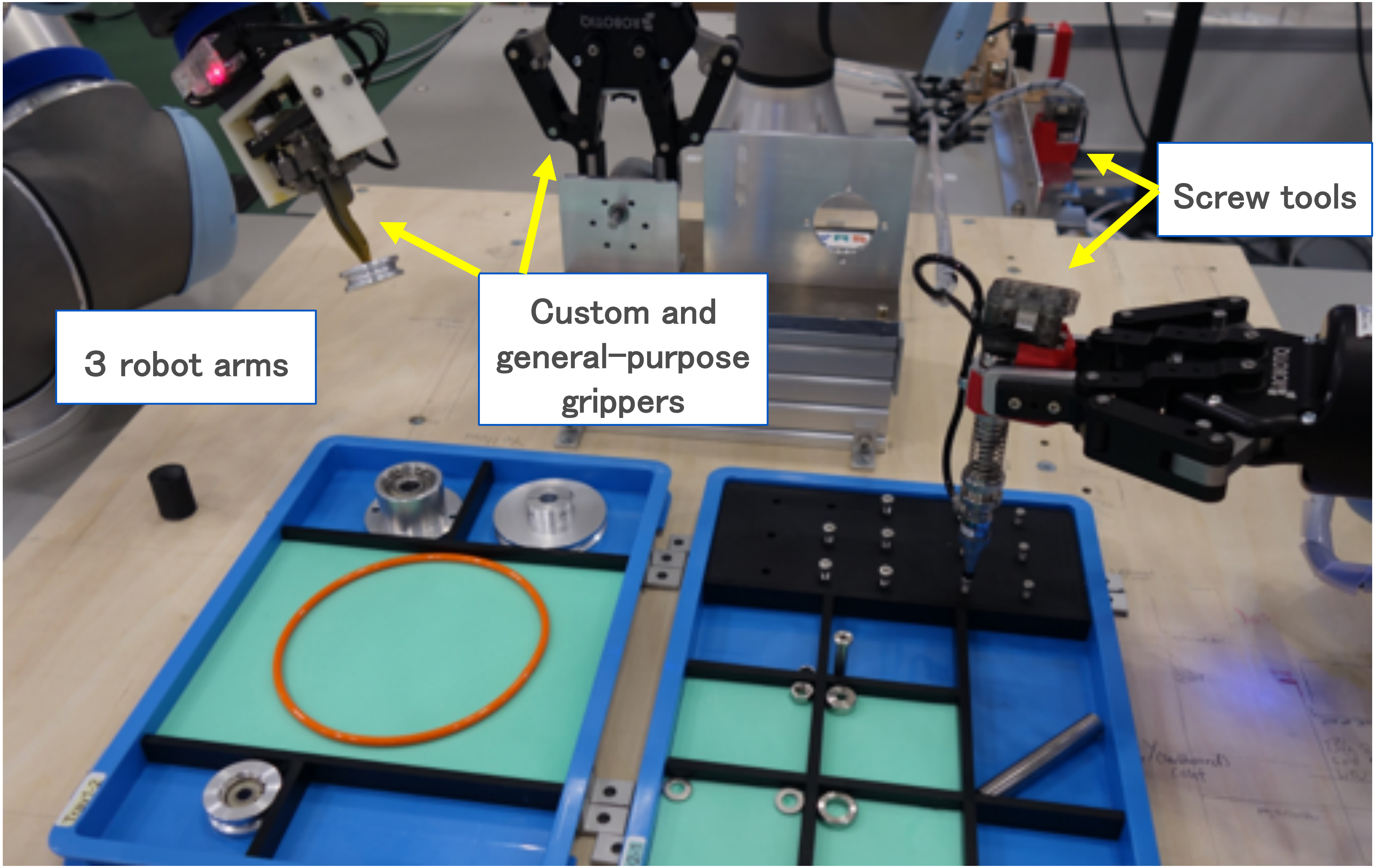
State of the art

- Task is separated into small, teachable sub-units
- Specialized jigs are used to position parts and ensure known state
- Inflexible, high engineering cost, long changeover times



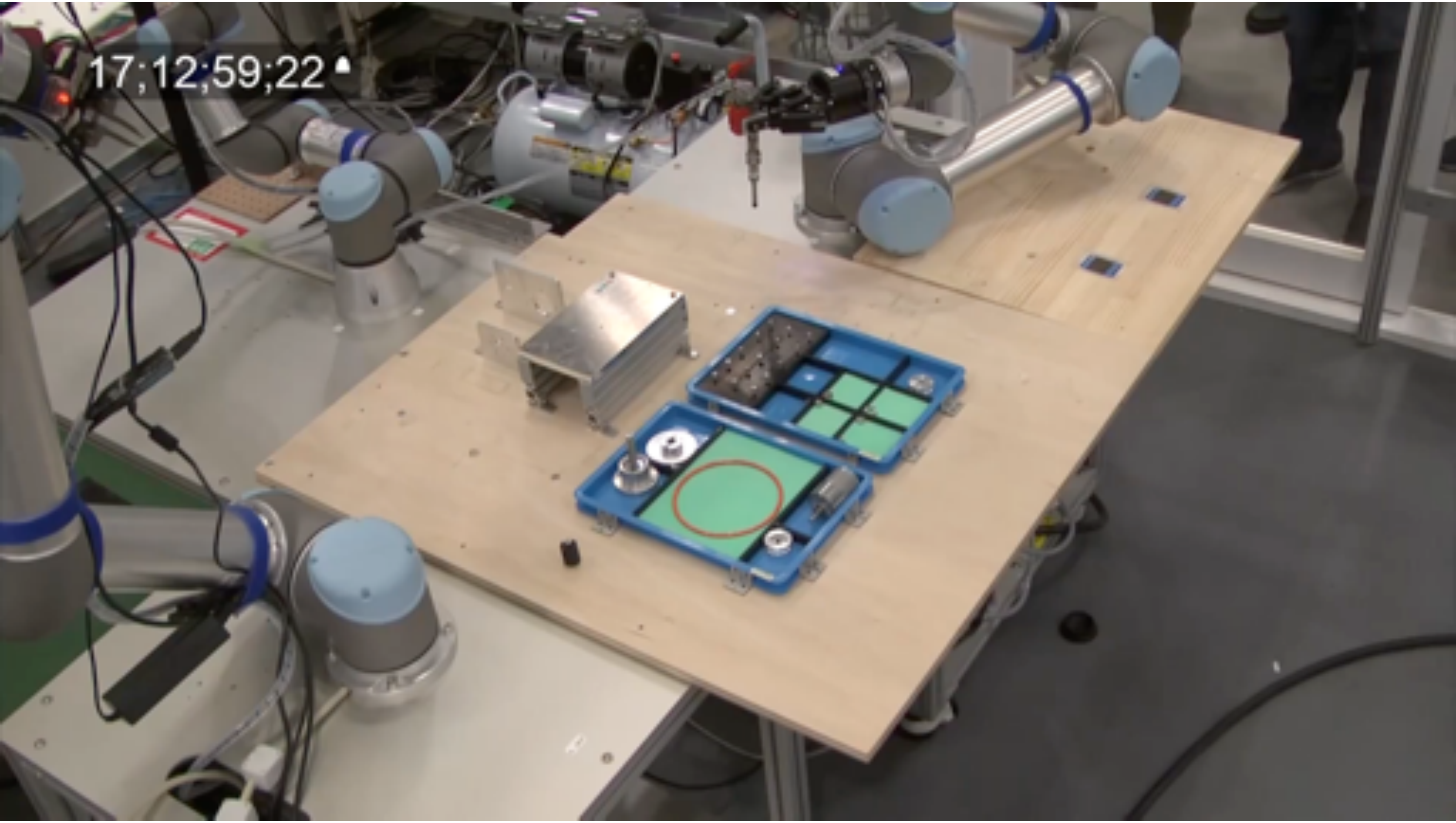
iREX 2017

System



3 robot arms
Custom and general-purpose grippers
Screw tools

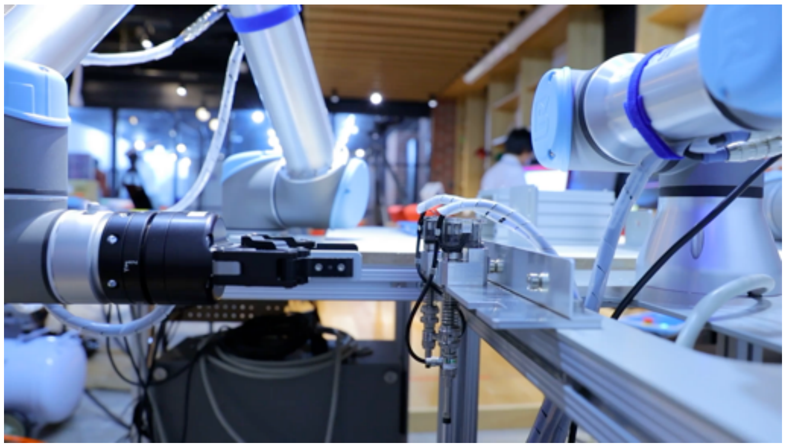
Result



System

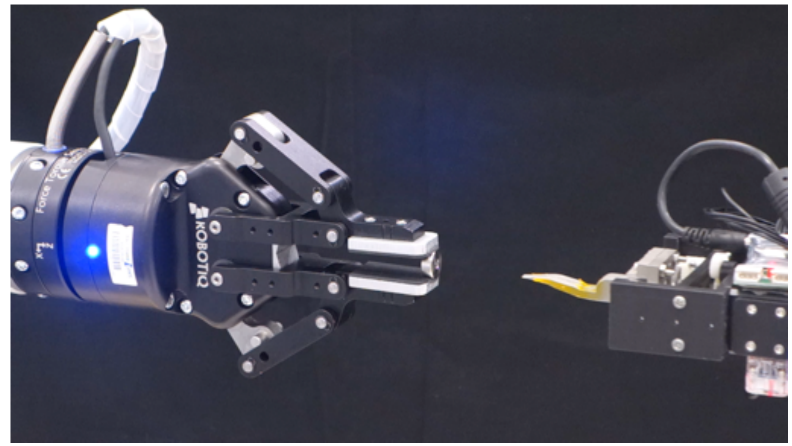
Flexible tool use without tool changers

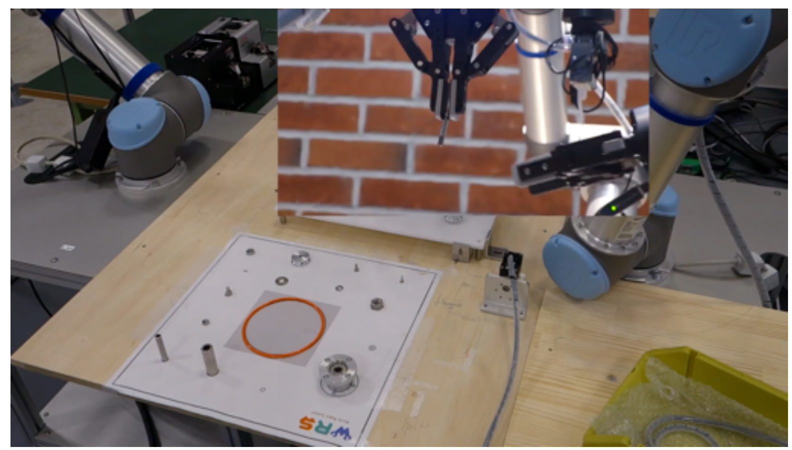
(SICE Award)



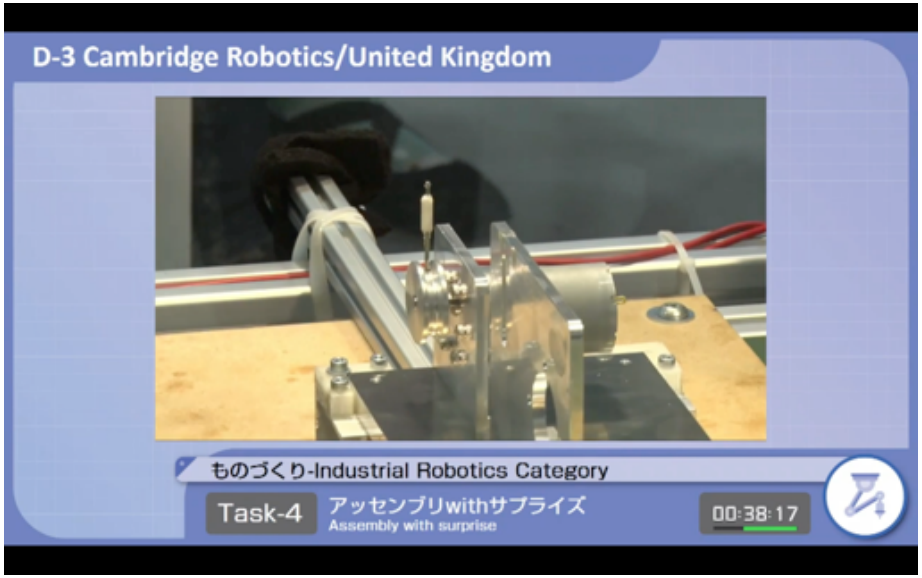
Compliance

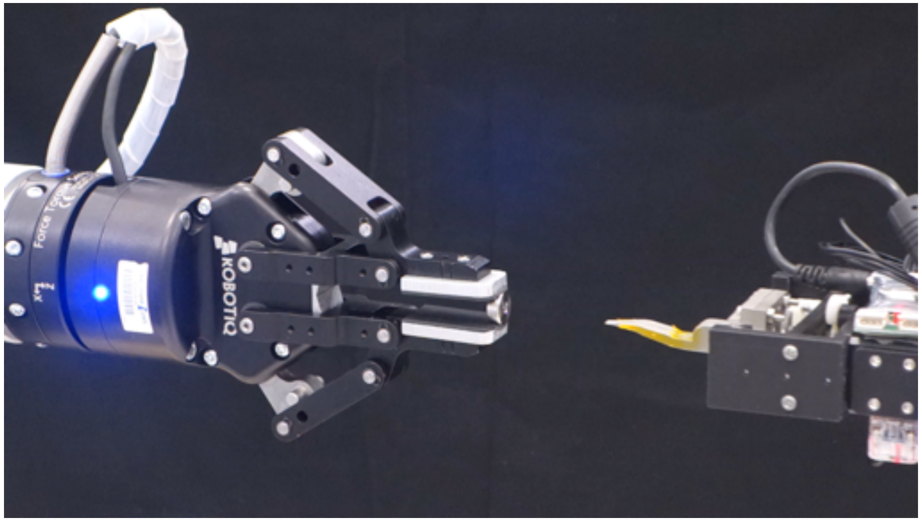
Jigless part centering



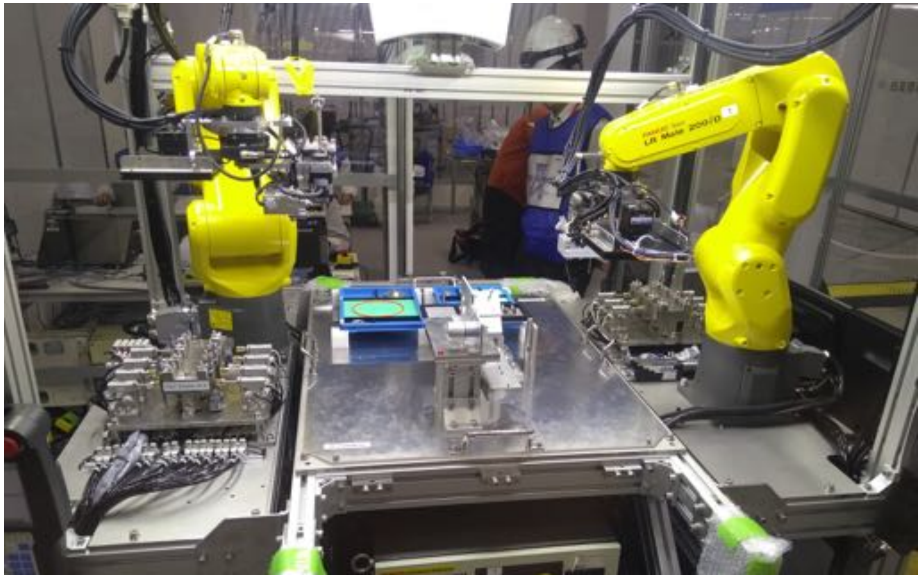


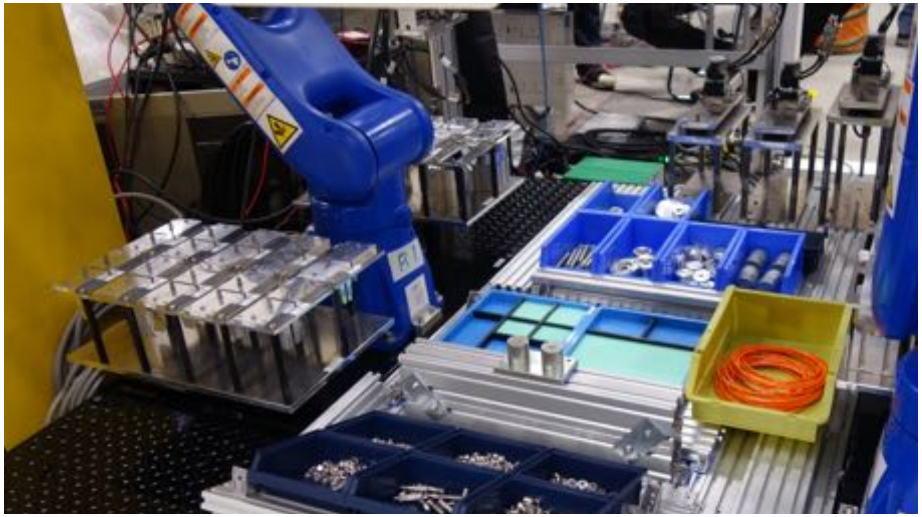
Strategy differences



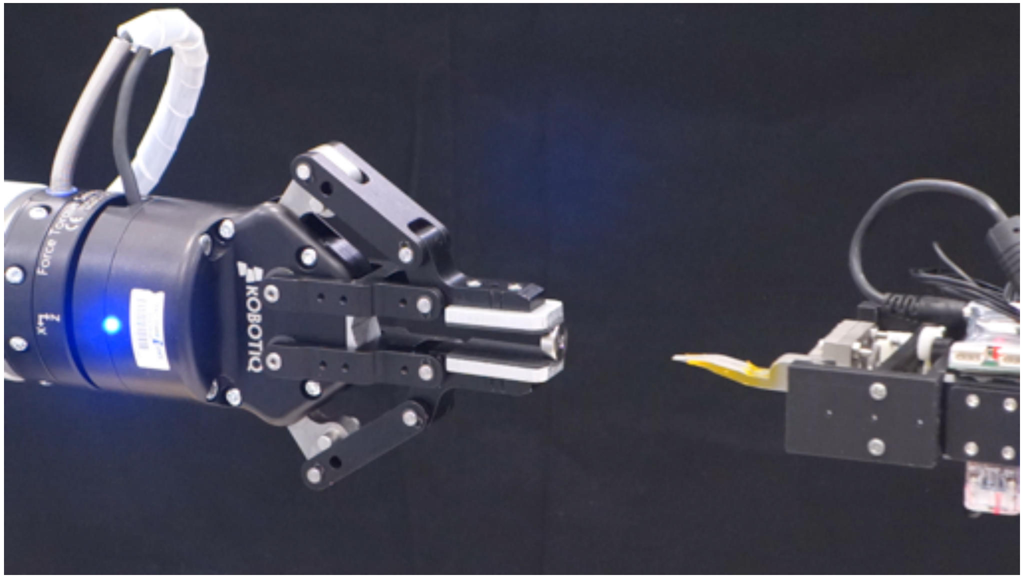


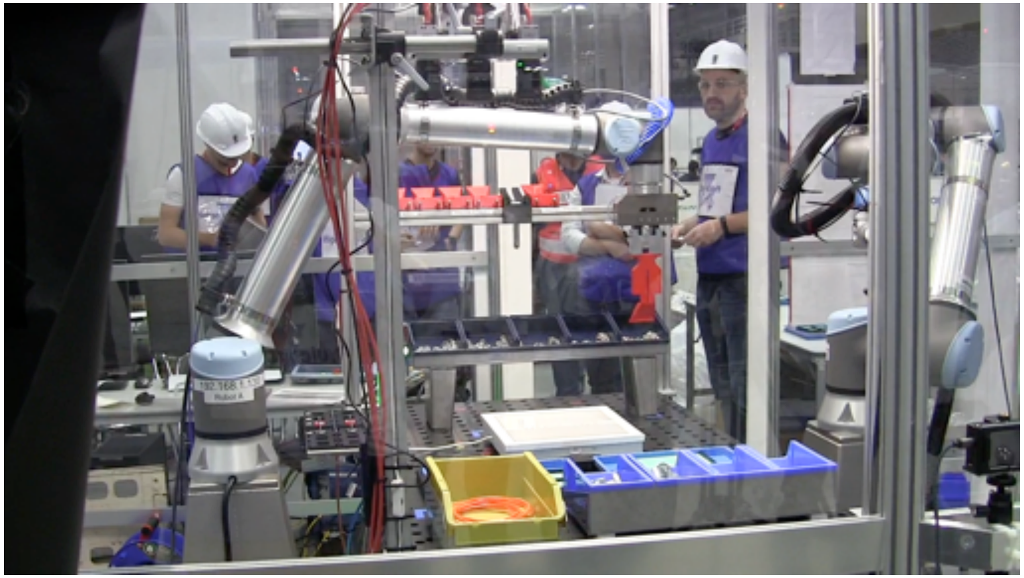
vs

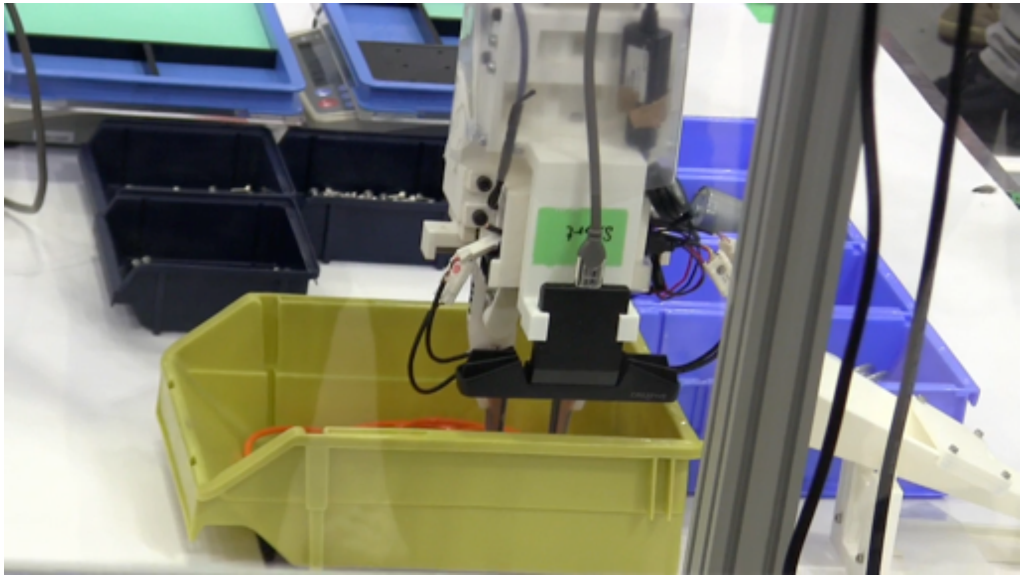




Flexibility vs Specialization

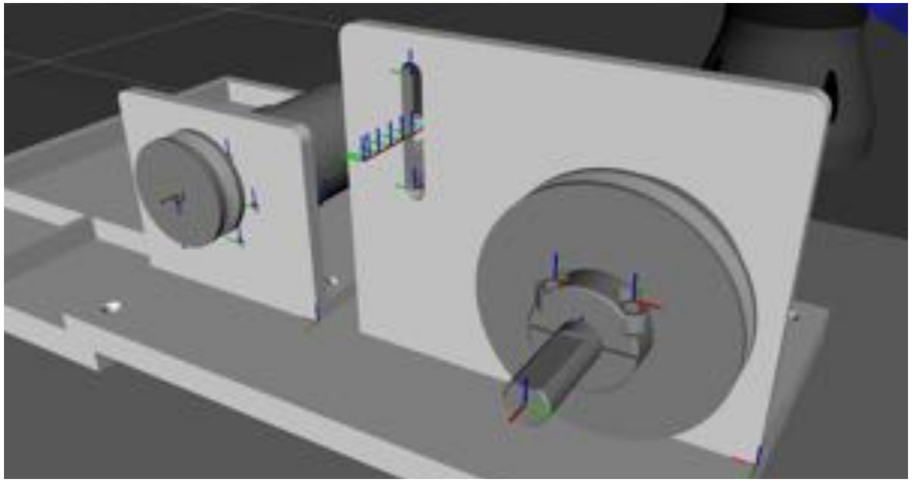


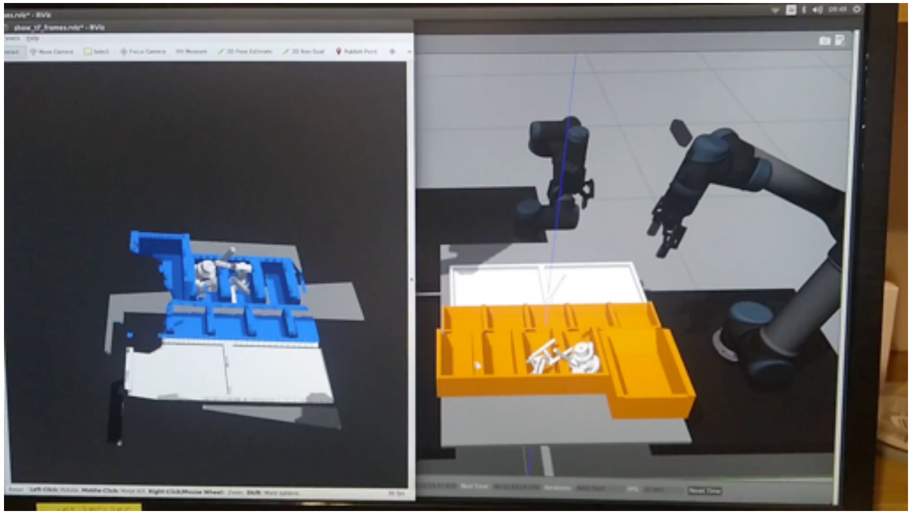




Example code

1. TF assembly generator
2. Multi-robot moveit config
3. Task instructions



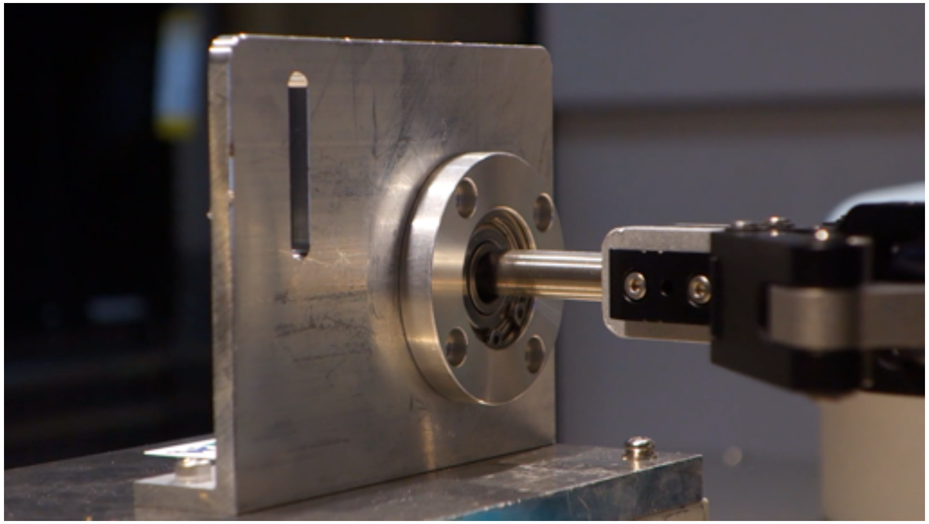


All code public:

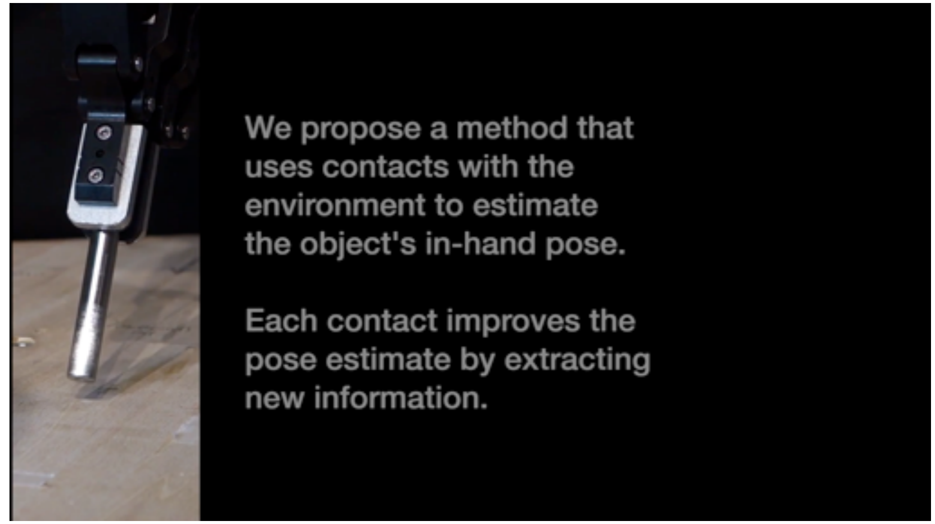
github.com/o2as/ur-o2as

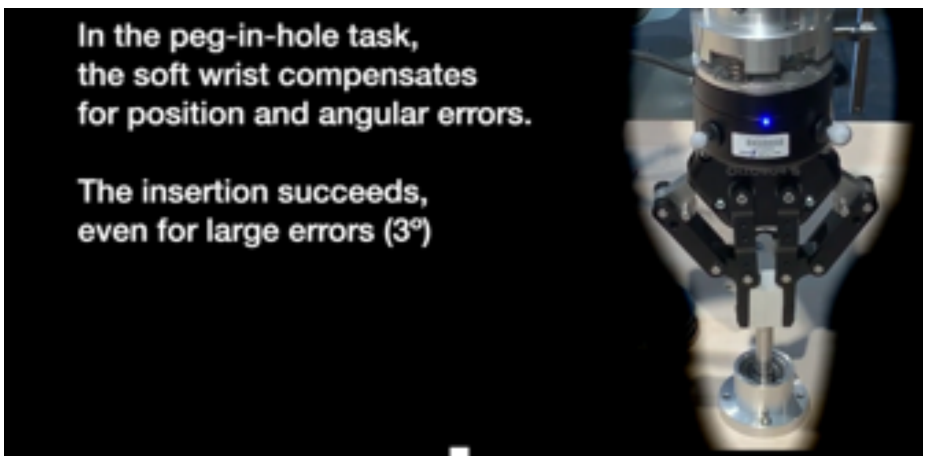
Where to go from here?

Flexible task description

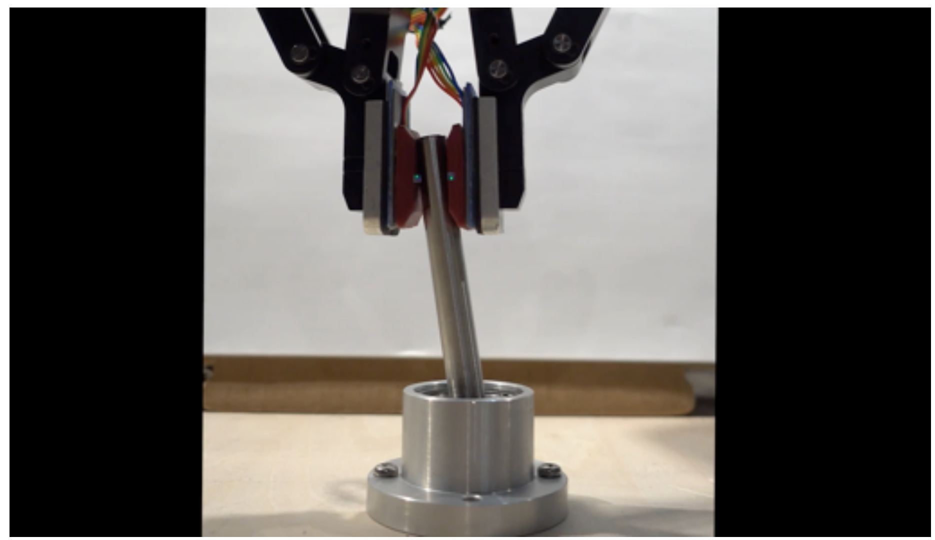


In-hand pose estimation





Compliant joints



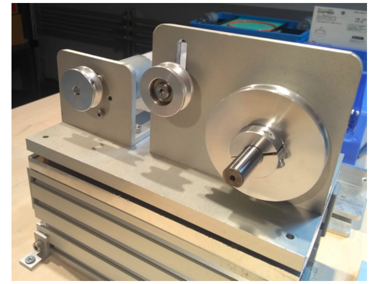
Tactile feedback

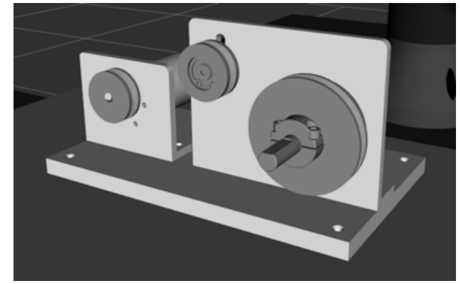
More challenges

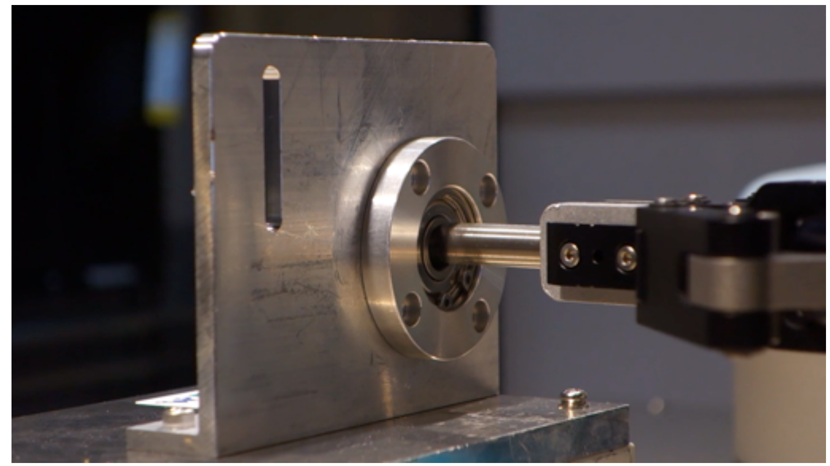
What we desired:

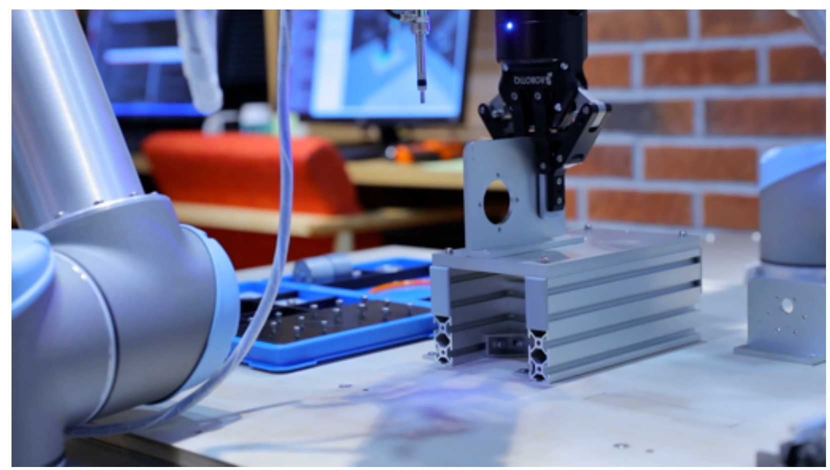
1. A visual success check
2. High certainty
   (if necessary from multiple views)
3. With easy setup (!)

- How to achieve this without explicit coding or demonstrations for every assembly step?
- Does it require semantic scene understanding?
- What is the easiest way to bring this to users or into automated routines?









Other notes

- No tutorial for multi-robot setup and best practices

→ Multiple move_groups

- Our student interns struggled to learn MoveIt within 1-2 months

→ Custom Python wrapper

  - pick (deterministic)
  - place (deterministic)
  - move_lin(robot_name, goal)

  …

3 robot arms

10 tools

5 move_groups

Custom Python wrapper

Safe design practices

Imagine designing this road section

How to make this road safe
and avoid accidents?

1. Direct: Remove cause accident unlikely (increase curve radius)
2. Indirect: Reduce impact (set up guard rails)
3. Indicative: Warn the driver (put up a sign saying "Please slow down")

"Accident": A user having a frustrating experience

Good UX practices

Decrease moments users feel lost, frustrated, unsupported
Increase moments users are learning, engaged, active

1. **Direct**
   - **Make things self-explanatory (and follow conventions)**
   - **Clean design**
2. **Indirect**
   - **Helpful error messages**
   - **Tooltips etc.**
3. **Indicative**
   - **Tutorials**
   - **Document idiosyncracies**

Audiences

1. Hobbyists & Students
2. Instructors
3. Researchers
4. Integrators
5. Robot manufacturers?

Expectations

1. Hobbyists & Students
   - Easy to learn, should work out of the box
2. Instructors
   - Easy to learn, with tutorials
3. Researchers
   - Easy to maintain, tweak, extend
4. Integrators
   - Easy to program
5. Robot manufacturers?
   - Easy to implement?

User voices

- Why does my robot spin around half the time?
- What do all these planner options mean?
- What are good defaults for all the values?
- Why does the planning fail?
  Collision? Joint limits? Attached objects?
- ”I have no idea how the end effector setting works or what it does”
- Hard to understand IK results
- Even with MoveGroupInterface, code gets lengthy.
  - “Classic” robot operators from industry: CIRC? LIN? PTP?

Advanced users:

- Hard to set up your own planner
- With ROS, many low-level challenges are solved (more or less)
  What to do about high-level? Manipulation, tasks, process planning…

Conclusion

- Setting up complex tasks is difficult
  → Higher-level planning? "Task" message?

- MoveIt can be a great tool for assembly planning – but which layer?

- Good defaults for out-of-the-box behavior – but for who?