This lecture will be recorded

COMPAS

064-0026-00L: COMPAS II

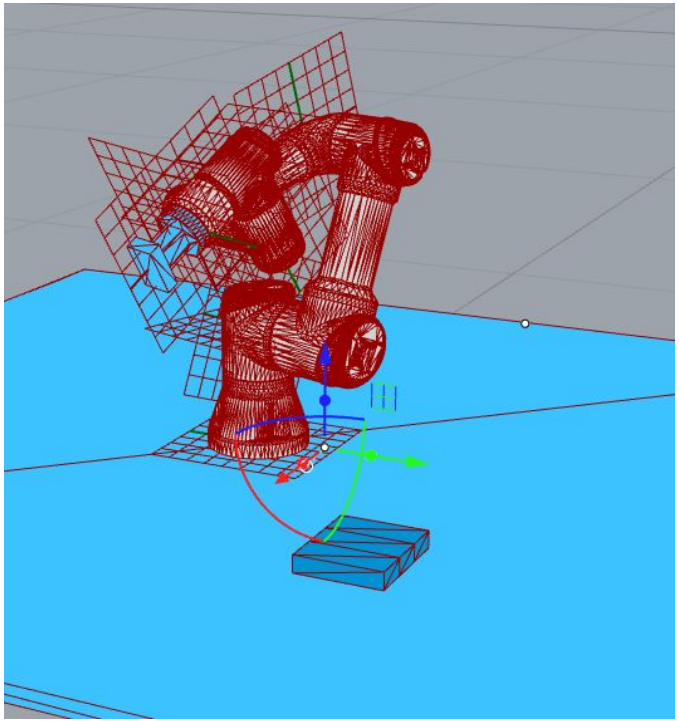
Introduction to Computational Methods for Digital Fabrication in Architecture

slides + code

https://`dfab.link/fs2022`

Review of last lecture assignment

- Define a parametric assembly based on examples 601-612.
- Goal 1: Ensure at least 20 parts are independently buildable (ie. there are trajectories for all).
- Goal 2: Ensure at least 20 parts are buildable taking into account previously built parts.



TODAY

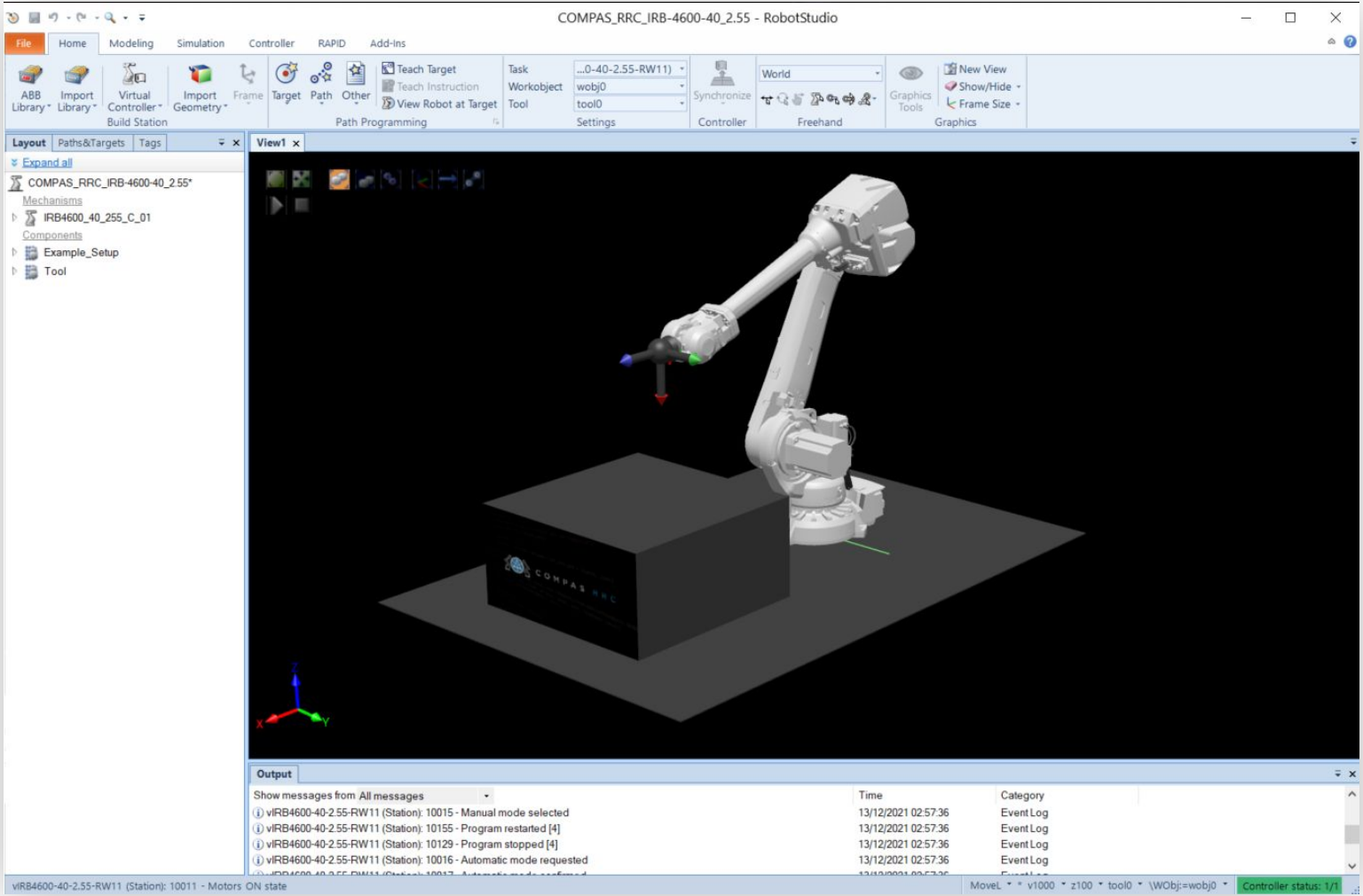
robot control

compas rrc

instructions overview

Today's goal

Apply **robot control using RRC** to solve a Pick and Place task.


COMPAS_RRC_IRB-4600-40_2.55 - RobotStudio
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Virtual Controller
Import Geometry
Frame
Target
Path
Other
Teach Target
Teach Instruction
View Robot at Target
Task
Workobject
Tool
...0-40-2.55-RW11)
wobj0
tool0
Synchronize
World
Graphics Tools
New View
Show/Hide
Frame Size
Build Station
Path Programming
Settings
Controller
Freehand
Graphics
Layout
Paths&Targets
Tags
Expand all
COMPAS_RRC_IRB-4600-40_2.55*
Mechanisms
IRB4600_40_255_C_01
Components
Example_Setup
Tool
View1
Output
Show messages from All messages
Time
Category
vIRB4600-40-2.55-RW11 (Station): 10015 - Manual mode selected
13/12/2021 02:57:36
Event Log
vIRB4600-40-2.55-RW11 (Station): 10155 - Program restarted [4]
13/12/2021 02:57:36
Event Log
vIRB4600-40-2.55-RW11 (Station): 10129 - Program stopped [4]
13/12/2021 02:57:36
Event Log
vIRB4600-40-2.55-RW11 (Station): 10016 - Automatic mode requested
13/12/2021 02:57:36
Event Log
vIRB4600-40-2.55-RW11 (Station): 10011 - Motors ON state
MoveL * v1000 * z100 * tool0 * \WObj:=wobj0
Controller status: 1/1

Search or jump to... / Pull requests Issues Marketplace Explore

compas-rrc / compas_rrc_start Private

Watch 2 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

fleischp pack and go updated with black background b12dcbd 15 minutes ago 40 commits

.vscode	pack and go updated with black background	15 minutes ago
Python	first brick example created (new pack&go, new viewer and new python c...	3 days ago
docker	first structure done	11 days ago
robotstudio-extras	names changed agording to the course	11 days ago
robotstudio-stations	pack and go updated with black background	15 minutes ago
robotstudio-viewers	first brick example created (new pack&go, new viewer and new python c...	3 days ago
README.md	Update README.md	6 days ago

README.md

Welcome to COMPAS RRC

> Online control for ABB robots over a simple-to-use Python interface.

Requirements

- Windows 10 Pro 64-bit
- CPU 2 GHz
- Memory 8 GB
- Disk 10 GB SSD
- Screen 1920x1080

Installation

Anaconda

- Anaconda 3: Anaconda is an open source scientific Python distribution. With this tool, we can easily create a Python environment. Install Anaconda using default options.

Docker

- Docker Desktop: Docker is a virtualization platform. We use it to run Linux containers for ROS on Windows machines.
- After installation, it is required to enable "Virtualization" on the BIOS of the computer. Usually this requires rebooting your computer and pressing a vendor-specific key (`F2`, `F4`, `Del`, are typical options) to enter the BIOS.

About

No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

Packages

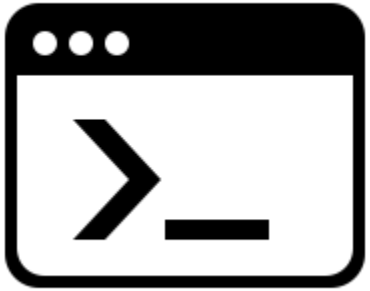
No packages published
Publish your first package

Contributors 2

fleischp Philippe Fleischmann
gonzalocasas Gonzalo Casas

Languages

Python 100.0%



`docker-compose up -d`

Right-click → Compose Up

docker/moveit-rrc-noetic

robot control

Traditional programming (offline control)

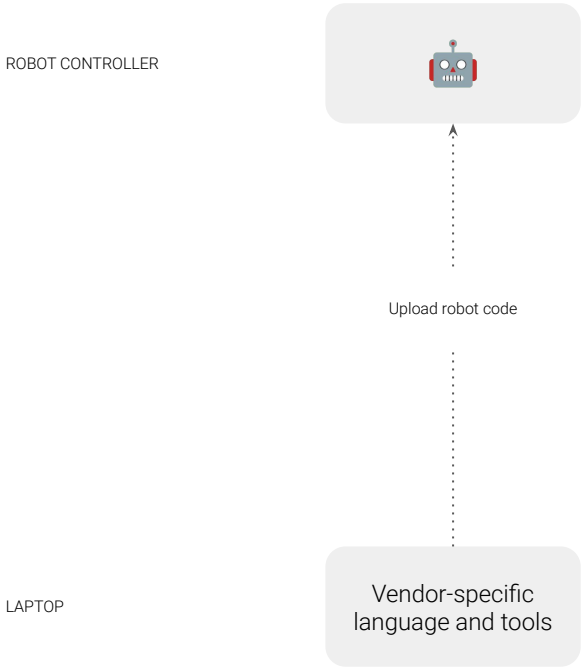


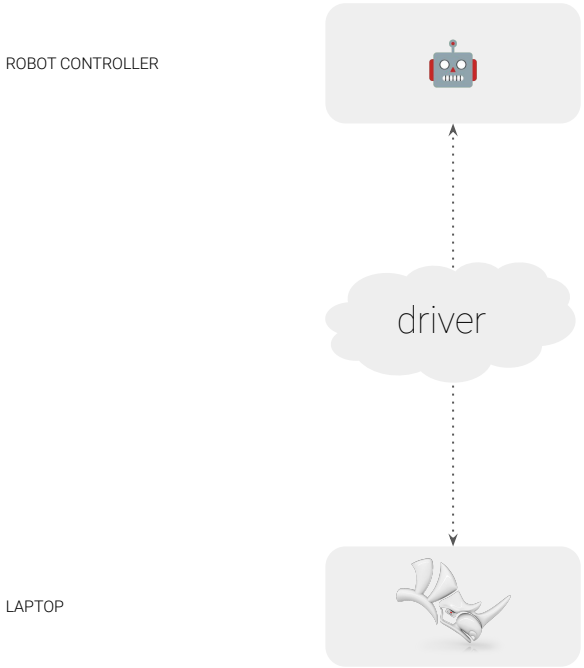
ABB: RAPID

Stäubli: VAL3

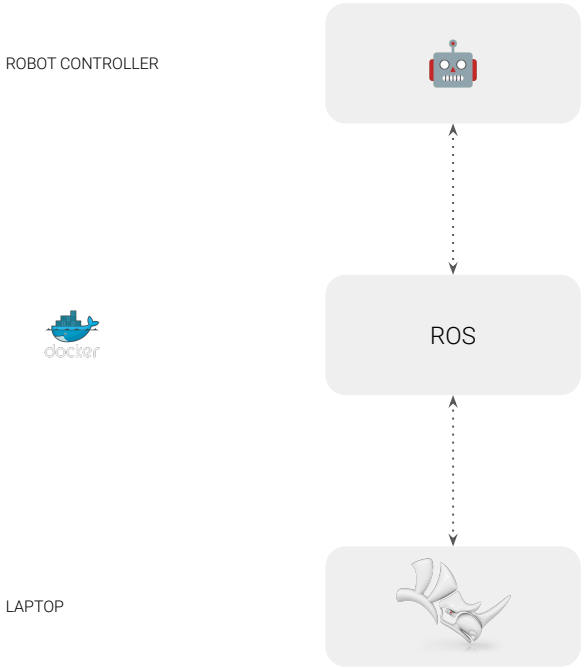
UR: URScript

Kuka: KRL

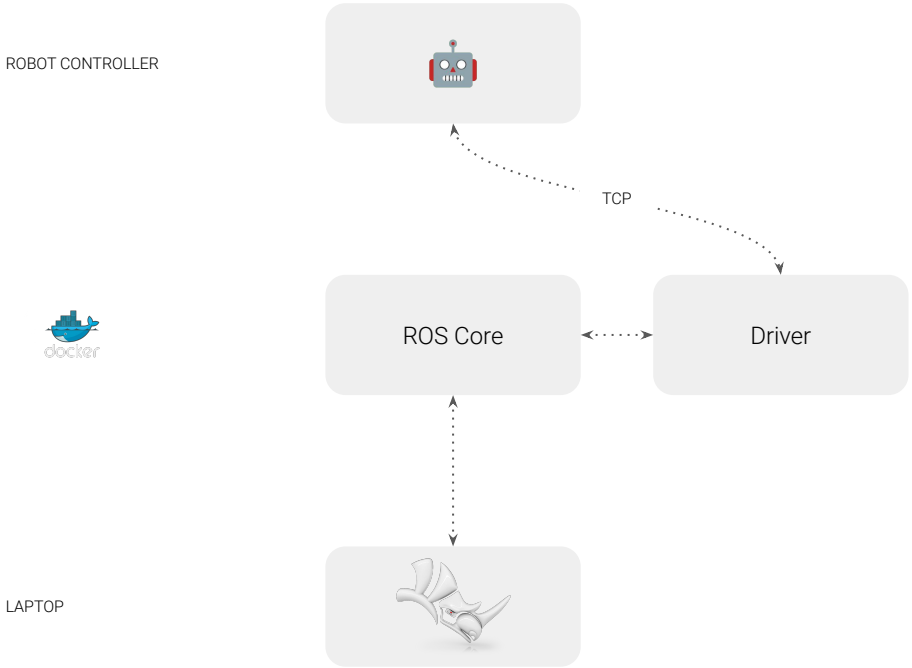
Our goal: online control



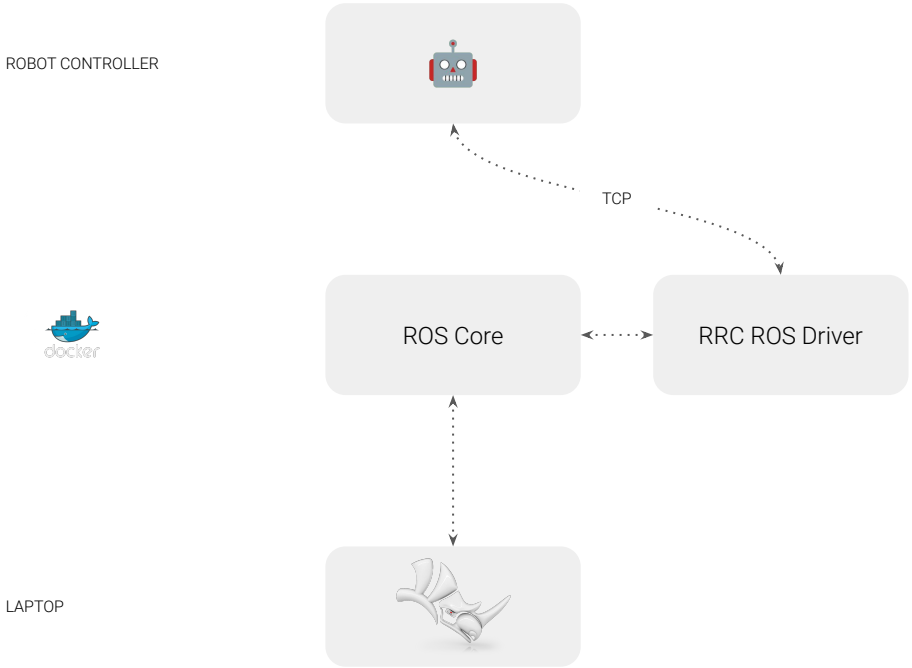
Overview: all the moving parts



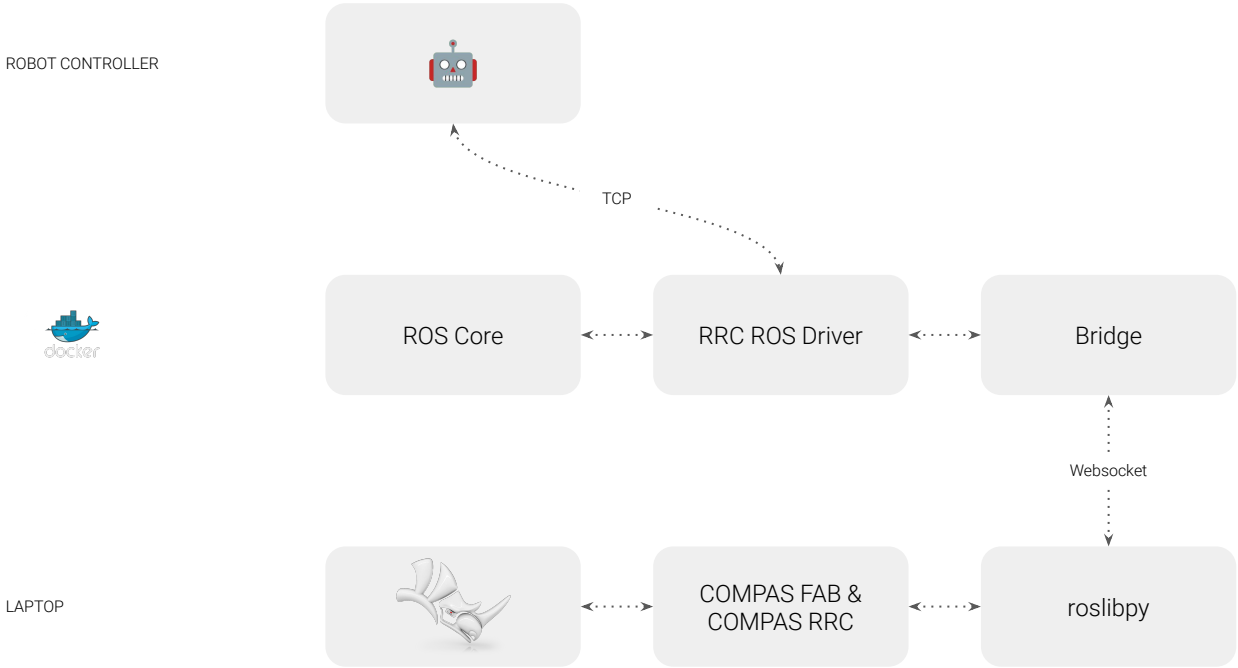
Overview: all the moving parts



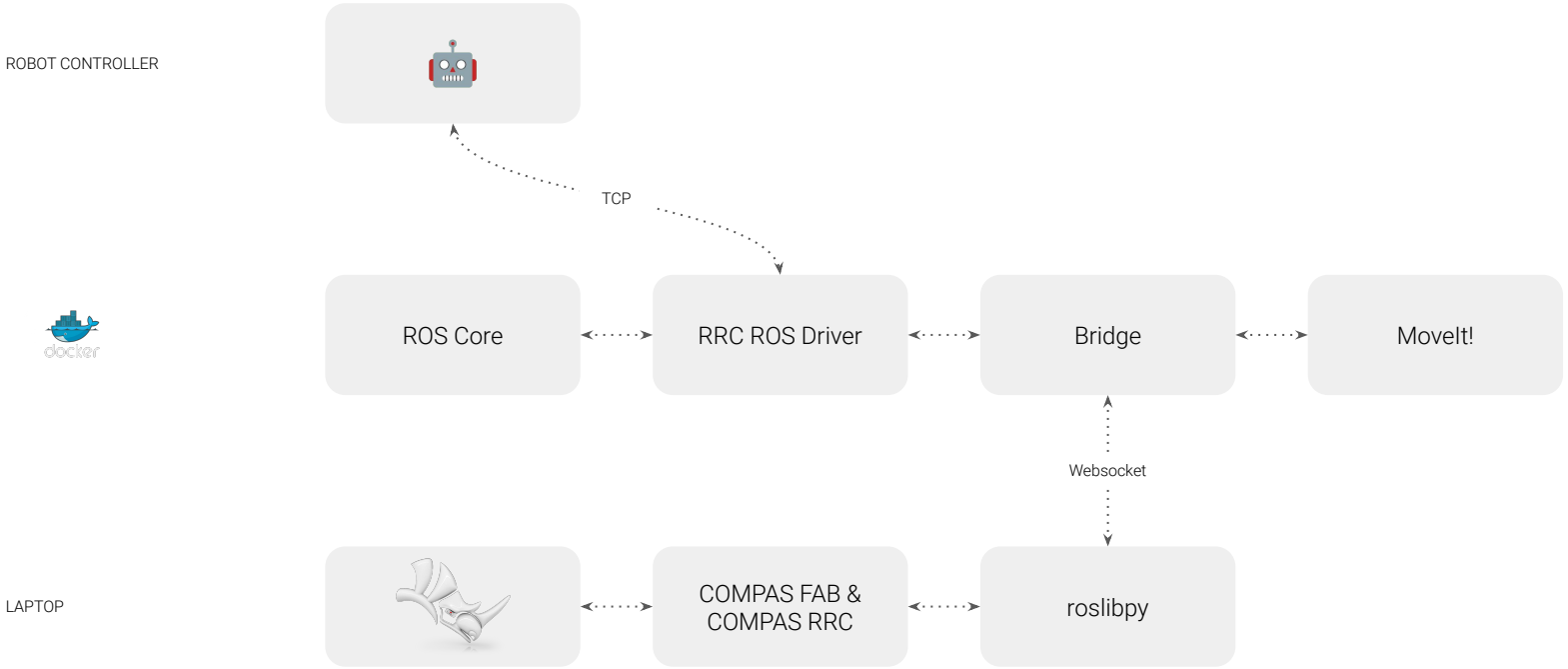
Overview: all the moving parts



Overview: all the moving parts



Overview: all the moving parts



Control

COMPAS RRC

Offline control

Online real-time control

Online non-real-time control

compas rrc

COMPAS RRC

COMPAS RRC

3D Concrete Printing

- Data streaming

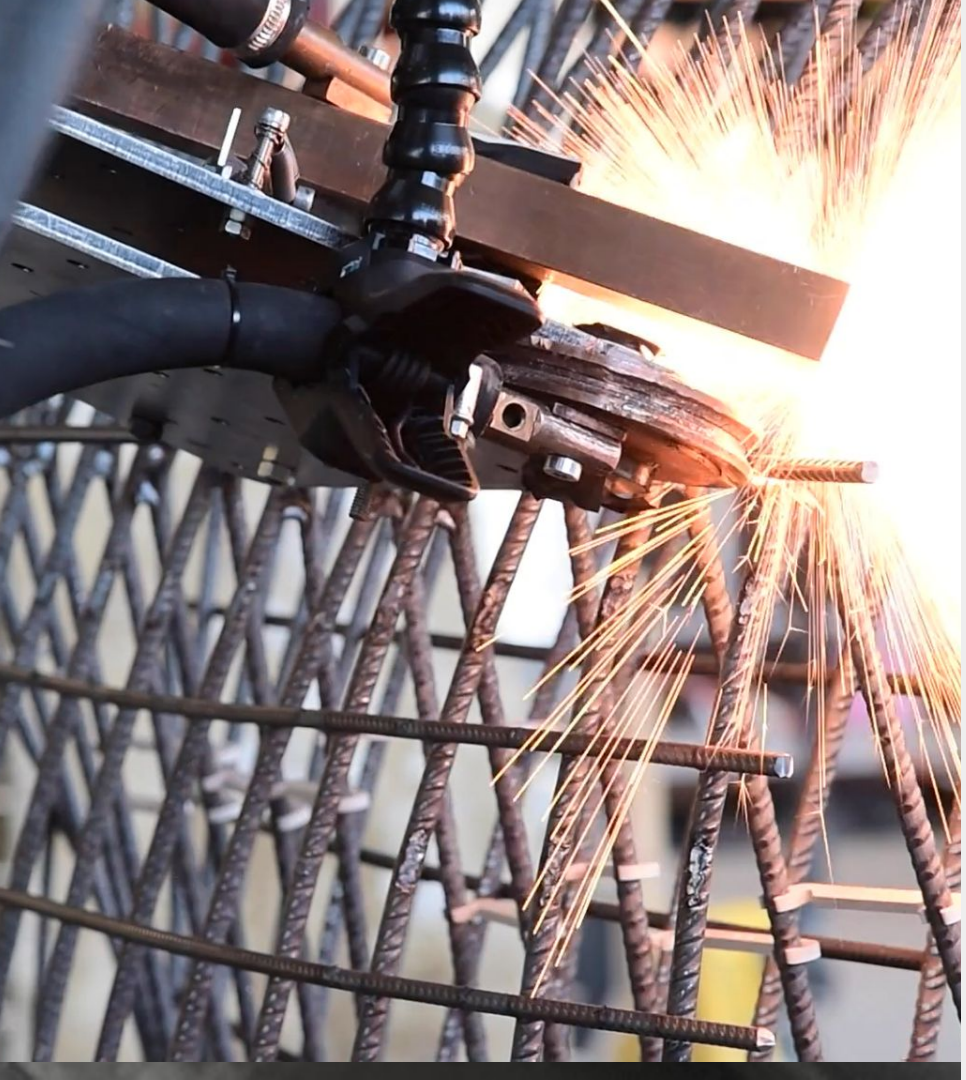
- More than 300’000 points
- More than 4 hours of printing

ETH zürich

WAAM

- Process Feedback

- Integrated Fronius welder
- Arc Weld PowerPac
- SmartTac



Mesh Mould

- Advanced Processing
- Slice positioning with
  - COMPAS RRC
- Rebar welding with
  - Externally Guided Motion

ETH zürich

COMPAS RRC

Features

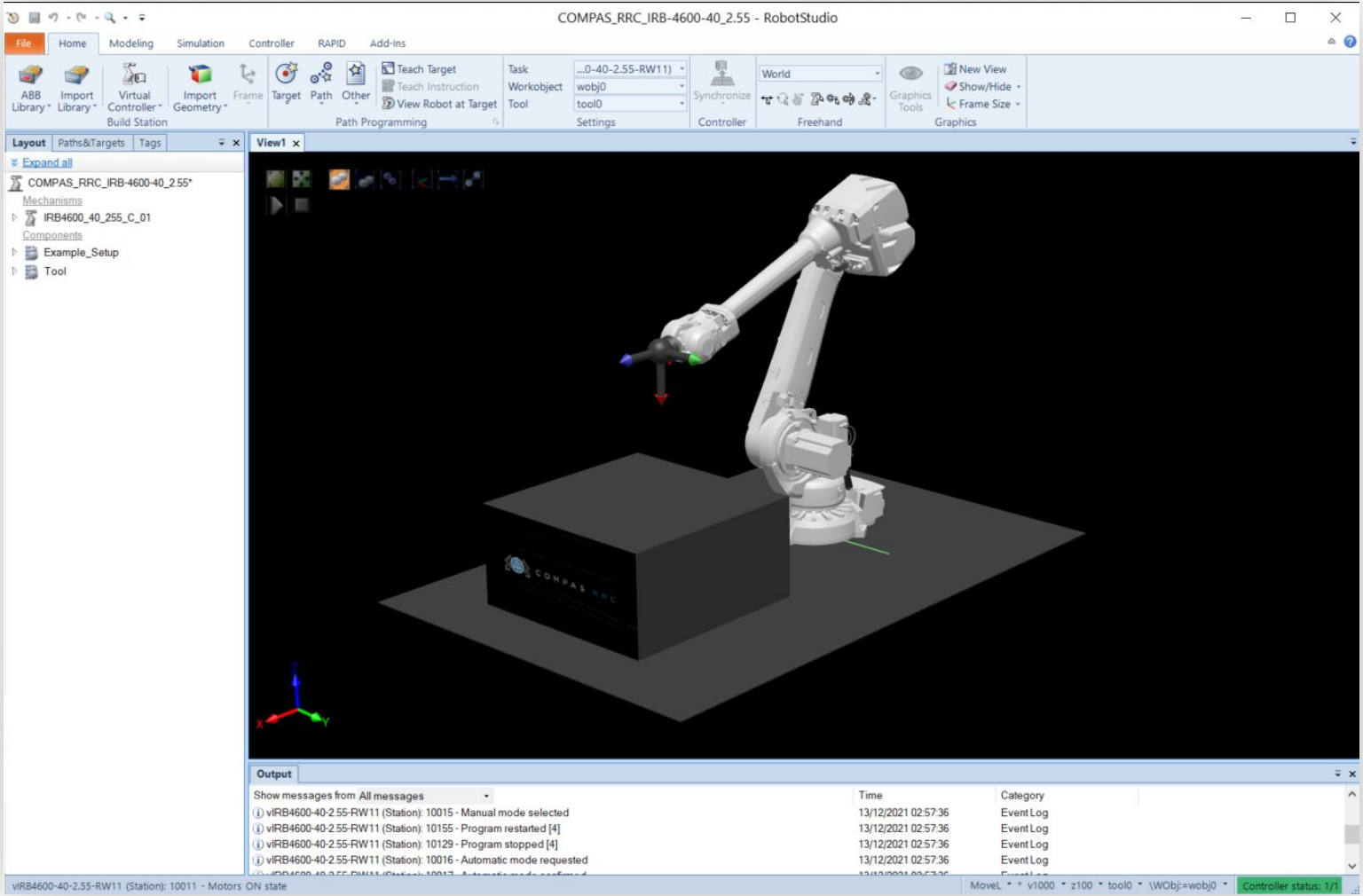
Live communication

Multi-Tasking

Multi-Move

Multi-Controller

Multi-Location


COMPAS_RRC_IRB-4600-40_2.55 - RobotStudio
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Virtual Controller
Import Geometry
Frame
Target
Path
Other
Teach Target
Teach Instruction
View Robot at Target
Task
Workobject
Tool
...0-40-2.55-RW11)
wobj0
tool0
Synchronize
World
Graphics Tools
New View
Show/Hide
Frame Size
Build Station
Path Programming
Settings
Controller
Freehand
Graphics
Layout
Paths&Targets
Tags
Expand all
COMPAS_RRC_IRB-4600-40_2.55*
Mechanisms
IRB4600_40_255_C_01
Components
Example_Setup
Tool
View1
Output
Show messages from All messages
Time
Category
vIRB4600-40-2.55-RW11 (Station): 10015 - Manual mode selected
13/12/2021 02:57:36
Event Log
vIRB4600-40-2.55-RW11 (Station): 10155 - Program restarted [4]
13/12/2021 02:57:36
Event Log
vIRB4600-40-2.55-RW11 (Station): 10129 - Program stopped [4]
13/12/2021 02:57:36
Event Log
vIRB4600-40-2.55-RW11 (Station): 10016 - Automatic mode requested
13/12/2021 02:57:36
Event Log
vIRB4600-40-2.55-RW11 (Station): 10011 - Motors ON state
MoveL * v1000 * z100 * tool0 * \WObj:=wobj0
Controller status: 1/1

Hello world

```
import compas_rrc as rrc

abb = rrc.AbbClient(ros, '/rob1')
done = abb.send_and_wait(rrc.PrintText('Hello World'))
```

COMPAS RRC

Communication

Send ~75ms

Send and Wait ~150ms

Send and Wait in the Future

Send and Subscribe ~75ms

Send instruction (non-blocking)

```
# Send an instruction without waiting for any kind of feedback
abb.send(rrc.PrintText('Hello.'))
```

Send instruction with feedback (blocking)

```
# Send and wait
done = abb.send_and_wait(rrc.PrintText('Sent with feedback.'))
```

Send instruction with feedback (non-blocking)

```
# Send and defer waiting
future = abb.send(rrc.PrintText('feedback',feedback_level=rrc.FeedbackLevel.DONE))

# Here you can do other stuff [..]

# Wait for feedback
done = future.result(timeout=3.0)
```

instructions overview

COMPAS RRC

Instructions

Motions

Signals

Basics

Utilities

Custom

ETH zürich

COMPAS RRC

Basics

Set Tool

Set WorkObject

Set Max Speed

Set Acceleration

COMPAS RRC

Motion

Get Frame
Move to Frame

Get Joints
Move to Joints

Get Robtarget
Move to Robtarget

ETH zürich

COMPAS RRC

Utilities

Wait Time (Delay)

Stop (Pause)

Stop Watch

COMPAS RRC

Utilities

No-op (Ping)

Print Text

COMPAS RRC

Utilities

Custom Instructions

COMPAS RRC

IO signals

Read analog
Set analog

Read digital
Set digital
Pulse digital

Read group signal
Set group signal

Work objects

```
# Define pick positions
frame_on_pick = Frame(Point(50, 50, 50), Vector(0, -1, 0), Vector(-1, 0, 0))
frame_on_place = Frame(Point(50, 50, 50), Vector(0, -1, 0), Vector(-1, 0, 0))

# Move to frame on pickup pallet (work object)
abb.send(rrc.SetWorkObject('ob_RRC_Brick_Pallet'))
abb.send_and_wait(rrc.MoveToFrame(frame_on_pick, speed, rrc.Zone.FINE))

# Move to frame on place (work object)
abb.send(rrc.SetWorkObject('ob_RRC_Build_Space'))
abb.send_and_wait(rrc.MoveToFrame(frame_on_place, speed, rrc.Zone.FINE))
```

Pick & place example

```
# Create a new brick
done = abb.send_and_wait(rrc.PulseDigital('doNewBrick',0.2))
# [..]

# Vacuum on
abb.send(rrc.SetDigital('doVacuumOn',1))

# Motion
abb.send(rrc.MoveToFrame(pre_place_position, speed, rrc.Zone.Z10))
abb.send(rrc.MoveToFrame(place_position, speed, rrc.Zone.FINE))

# Vacuum off
abb.send(rrc.SetDigital('doVacuumOn',0))
```

Next week

- No coding assignment
- RRC summary quiz, due next week Wed 4th May, 9AM.

  https://forms.gle/em7K2M5CWsHhFVr4A
- Ask for help if needed: Slack, Forum, Office Hours (Fridays, request via Slack)
- Next lecture:
  - 🤖🙌 **On-site at the RFL!**
  - Robot control exercise with a real robot

Thanks!