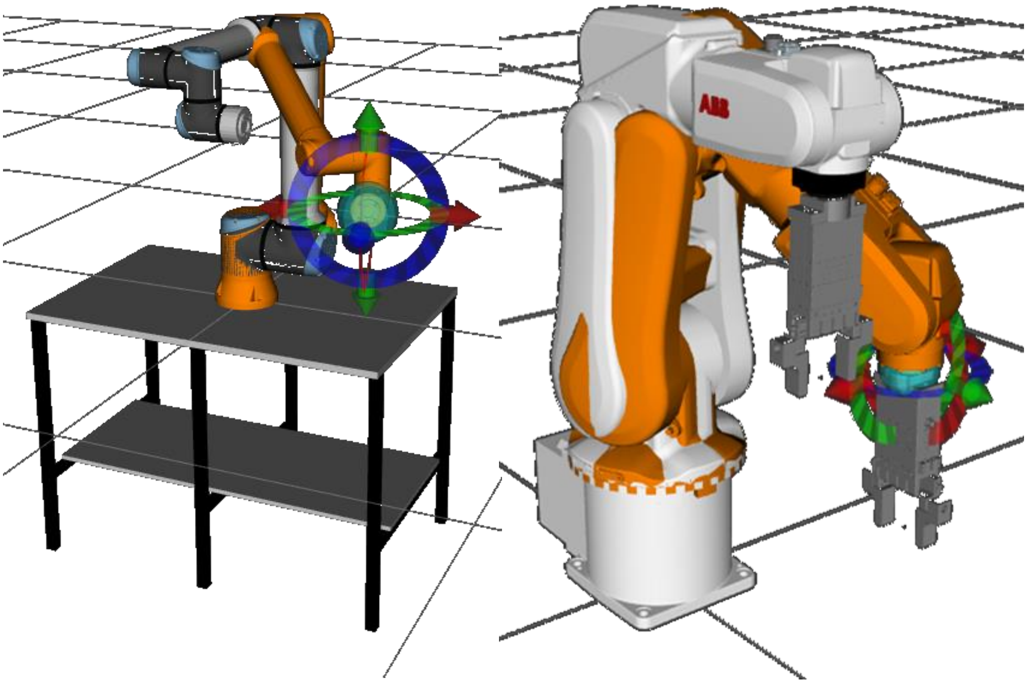
Open-Source Tool for the Simulation and Control of any Robot Arm in ROS 2

ROSCon UK 2025 – Workshop Preparation

This hands-on workshop introduces a modular ROS 2 framework developed by IFRA-Cranfield for simulating and controlling robot manipulators in both virtual and real environments. The framework combines Gazebo, MoveIt!2, and ROS 2-based tools to support streamlined design, deployment, and execution of robotic tasks. Participants will learn to configure custom robot workcells, execute robot movements, monitor system state, and integrate external software tools using ROS 2. Ideal for researchers, academics, and robotics engineers, the workshop offers a standardised and reusable platform for teaching, testing, research, and rapid prototyping of robotic applications.



Workshop Lead

Mr. Mikel Bueno Viso: RA in Intelligent Automation (Mikel.Bueno-Viso@cranfield.ac.uk)
Dr. Seemal Asif: Senior Lecturer in AI and Robotics (s.asif@cranfield.ac.uk)

IFRA-Cranfield: Intelligent Flexible Robotics & Assembly, Research Group
Centre for Robotics & Assembly
Faculty of Engineering & Applied Sciences
Cranfield University, UK

ROSCon UK 2025 Workshop: Robot Simulation and Control using ROS 2

We are excited to welcome you to our hands-on workshop at ROSCon UK 2025, where we will explore a modular, open-source framework for simulating and controlling robot manipulators in both virtual and real environments using ROS 2, MoveIt!2, and Gazebo.

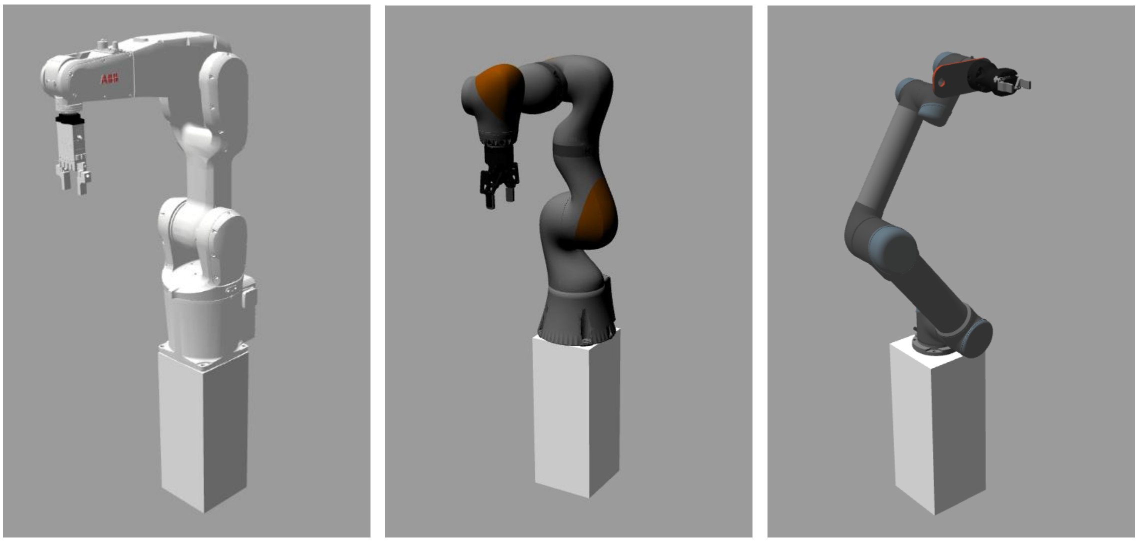
Developed by the IFRA-Cranfield Research Group at Cranfield University, this framework is designed to streamline the design, deployment, and operation of robotic workcells—whether you are working on industrial robots, collaborative robots, or research prototypes. Our aim is to provide you with a standardised, reusable, and flexible development workflow that supports teaching, research, prototyping, and real-world applications.

This workshop will be highly practical. You will follow each step alongside us to set up your own ROS 2 environment, run robot simulations, control a robot arm, and integrate external software tools such as intelligent vision systems and custom Human-Machine Interfaces (HMIs). All resources used in the workshop are fully open-source and will remain available after the session, allowing you to continue experimenting and adapting the framework to your own projects.

By the end of the workshop, you will:

- Understand the structure and functionality of our modular ROS 2 framework.
- Be able to deploy and operate robot workcells in both Gazebo simulation and on real hardware.
- Know how to integrate external perception and control tools into your ROS 2 workflows.
- Have a foundation to standardise your robotic development process using reusable packages and launch systems.

We encourage you to actively participate, ask questions, and share your own insights. The workshop is also a space for discussion on how we can collectively improve and standardise simulation and control practices in ROS 2.



Workshop Preparation: Requirements

To get the most out of this workshop and actively follow every step of the hands-on activities, it is essential that you arrive with a fully **prepared laptop**. This will allow you to replicate the demonstrations in real time, troubleshoot alongside us, and leave the session with a working setup you can continue using afterwards.

Please ensure before the workshop that your system meets the following requirements:

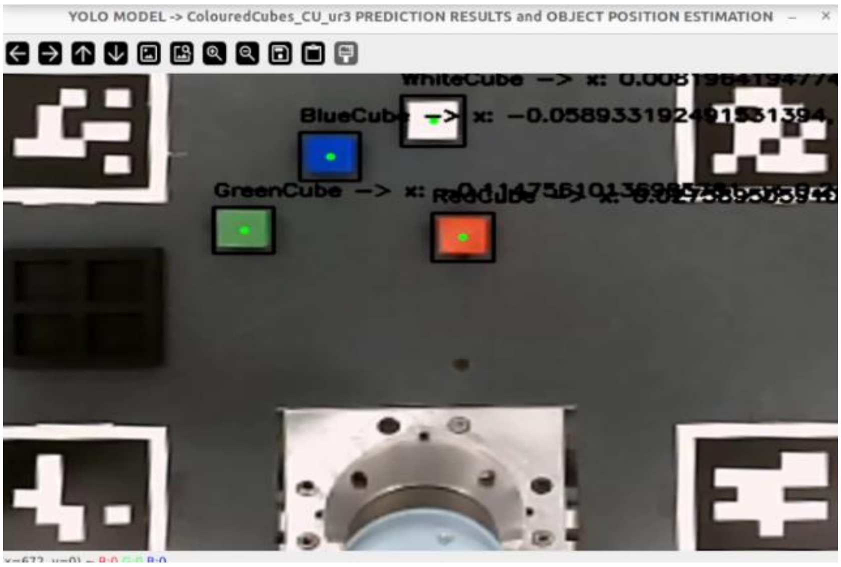
Required Software & Resources

1. **Ubuntu 22.04 LTS** installed (native installation is strongly recommended for performance and compatibility).
2. **ROS 2 Humble** fully installed and configured on your Ubuntu system.
3. **IFRA-Cranfield/ros2_SimRealRobotControl** repo cloned and installed from GitHub.
4. **IFRA-Cranfield/ur3_CranfieldRobotics** repository cloned and installed from GitHub.
5. Workshop's **IFRA-Cranfield/ROSCon_UK_2025** repository cloned from GitHub. This contains all the exercises and resources we will use during the session.

Why this matters: The workshop is designed for active, step-by-step replication. If your laptop is not set up in advance, you may miss critical parts of the practical work, as installation and configuration can take significant time. Completing the pre-work ensures you can focus on learning and experimenting, rather than troubleshooting basic setup issues during the session.

If you are unable to install Ubuntu and ROS 2 natively, we will provide a Virtual Machine image with all dependencies pre-installed. However, this is not recommended as VM performance may limit your ability to run real-time simulations and control processes effectively.

The next section will guide you through the detailed **installation steps** for each requirement so you can arrive fully ready to start building and controlling robot workcells from the first minute of the workshop.


YOLO MODEL -> ColouredCubes_CU_ur3 PREDICTION RESULTS and OBJECT POSITION ESTIMATION
WhiteCube --> x: 0.0081984194774
BlueCube --> x: -0.0589331924915313394
GreenCube --> x: 0.1147561013698
RedCube --> x: 0.0275895053

Workshop Preparation: Installation Steps (OPTION A – Recommended)

Follow the steps below before attending the workshop to ensure your system is fully prepared. Completing these installations in advance will allow you to focus entirely on the hands-on activities instead of setup and troubleshooting.

---

[1] **Install Ubuntu 22.04 and ROS 2 Humble**

We strongly recommend native installation of Ubuntu 22.04 LTS for best compatibility and performance.

(1.1) Install Ubuntu 22.04

- Download the ISO from: https://releases.ubuntu.com/22.04/
- Create a bootable USB and install: https://ubuntu.com/tutorials/install-ubuntu-desktop

(1.2) Install Git

```
# In your Terminal Shell:
sudo apt install git

# Git account configuration (recommended, not required):
git config --global user.name YourUsername
git config --global user.email YourEmail
git config --global color.ui true
git config --global core.editor code --wait # Link to Visual Studio Code.
git config --global credential.helper store
```

(1.3) Install ROS 2 Humble

- Follow the official ROS 2 Humble installation instructions: Debian packages, ros2-humble-desktop version: https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html
- Source the ROS 2 Humble Installation: Add the following statement at the bottom of the ***.bashrc*** file. This file is hidden in your /home folder in Ubuntu.

```
# At the following line at the bottom of your /.bashrc:
source /opt/ros/humble/setup.bash
```

---

[2] **Install Robot Arm Simulation and Control Packages**

We will use IFRA-Cranfield’s open-source robot control framework and the IFRA/UR3 robot repository for the hands-on exercises.

(2.1) Install MoveIt!2 Framework

```
# Command for the MoveIt!2 – Humble binary installation (recommended):
sudo apt install ros-humble-moveit
```

(2.2) Modify MoveIt!2's *move_group_interface.h* file

Once MoveIt!2 for ROS 2 Humble is installed, an additional step is required for our packages to work correctly. We need to replace the default MoveIt!2 header file move_group_interface.h with an improved version provided in our framework.

The improved file — move_group_interface_improved.h — must be copied into the following directory on your Ubuntu 22.04 system:

/opt/ros/humble/include/moveit/move_group_interface (requires admin rights to copy/paste)

Follow these steps:

```
# 1. Install Nautilus Admin (to access folders as an administrator):
sudo apt-get install nautilus-admin
# 2. Restart Nautilus so the admin options are enabled:
nautilus -q
# 3. Navigate to the MoveIt!2 /include repository:
# /opt/ros/humble/include/moveit/move_group_interface
# 4. Open the folder as an Administrator:
# Right-click inside the folder -> Select "Open as Administrator"
# 5. Copy and paste the improved file:
# Download it from the link above -> Paste it to the folder
```

(2.3) Create a *dev_ws* ROS 2 Workspace

- In your home folder, create a folder called **dev_ws**, with a folder called **src** inside.
- Source the ~/dev_ws workspace: Add the following statement at the bottom of the ***.bashrc*** file. This file is hidden in your /home folder in Ubuntu.

```
# At the following line at the bottom of your /.bashrc:
source ~/dev_ws/install/local_setup.bash
```

(2.4) Install required tools for ROS 2-based Robot Simulation and Control

- Install ROS 2 Development Tools:

```
sudo apt install ros-dev-tools
sudo apt install ros-humble-xacro # For urdf file processing.
```

- Install ROS 2 Control and ROS 2 Controllers for ROS 2 Humble:

```
sudo apt install ros-humble-ros2-control
sudo apt install ros-humble-ros2-controllers
sudo apt install ros-humble-gripper-controllers
```

- Install Gazebo Classic Simulator for ROS 2 Humble:

```
sudo apt install gazebo
sudo apt install ros-humble-gazebo-ros2-control
sudo apt install ros-humble-gazebo-ros-pkgs
```

Extra step required for Gazebo Simulator setup: Due to problems with URDF file processing for the newest version of ROS 2 Control-Gazebo plugin, Gazebo-ROS2-Control must be downgraded to the 0.4.6 version:

```
# Uninstall Gazebo ROS2 Control:
sudo apt remove ros-humble-gazebo-ros2-control

# Download and install the 0.4.6 version:
cd ~/dev_ws/src
git clone https://github.com/ros-controls/gazebo_ros2_control.git
cd gazebo_ros2_control
git reset --hard 9a3736c # Commit for the 0.4.6 version!
cd ~/dev_ws
colcon build
```

- Install CycloneDDS RMW for ROS 2 Humble:

```
sudo apt install ros-humble-rmw-cyclonedds-cpp
```

The installation of CycloneDDS must be sourced for ROS 2 to identify it. Therefore, the following statement must be added to the bottom of the /.bashrc file:

```
# At the following line at the bottom of your /.bashrc:
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
```

(2.5) Download and Install official ROS 2 Drivers for ABB and UR Robots

- ABB Driver for ROS 2: The installation of the abb_ros2 driver is required for the control of any real ABB robot using ROS 2. Installation steps:

```
# Execute the following in your terminal shell to install abb_ros2:
mkdir -p ~/dev_ws/src/ABBDriver
cd ~/dev_ws/src/ABBDriver
git clone https://github.com/PickNikRobotics/abb_ros2.git -b rolling
sudo rosdep init
rosdep update
vcs import < abb_ros2/abb.repos
rosdep install -r --from-paths . --ignore-src --rosdistro $ROS_DISTRO -y
cd ~/dev_ws
colcon build
```

- Universal Robots ROS 2 Driver: The installation of the ur-robot-driver is required for the control of any real UR robot using ROS 2. Binary install, for ROS2 Humble:

```
sudo apt-get install ros-humble-ur
```

(2.6) Installing supporting ROS 2 Packages required by ros2_SimRealRobotControl

In addition to the main framework, our simulation and control system relies on several auxiliary ROS 2 packages developed by IFRA-Cranfield.

These packages provide essential functionalities such as object linking, pose estimation, and gripper control, which are required for the correct operation of the ros2_SimRealRobotControl framework. The packages are:

- IFRA_LinkAttacher: Enables linking and unlinking of objects in simulation, useful for attaching tools or objects to the robot during a task.
- IFRA_ObjectPose: Publishes the pose (position and orientation) of simulated objects into ROS 2 topics for perception and planning.
- IFRA_LinkPose: Retrieves and publishes the pose of specific robot links in real-time.
- ros2_RobotiqGripper: Provides control interfaces for Robotiq grippers in ROS 2, including simulation and real-hardware modes.

Installation steps:

```
# 1. Download the repositories from IFRA-Cranfield’s GitHub:
cd ~/dev_ws/src
git clone https://github.com/IFRA-Cranfield/IFRA_LinkAttacher.git
git clone https://github.com/IFRA-Cranfield/IFRA_ObjectPose.git
git clone https://github.com/IFRA-Cranfield/IFRA_LinkPose.git
git clone https://github.com/IFRA-Cranfield/ros2_RobotiqGripper.git
# 2. Install the packages:
cd ~/dev_ws
colcon build
```

(2.7) Install the ros2_SimRealRobotControl repository

This repository developed by IFRA-Cranfield contains all the ROS 2 Packages, tools and resources required for the simulation and control of robot manipulators in ROS 2 Humble:

```
# 1. Download:
cd ~/dev_ws/src
git clone https://github.com/IFRA-Cranfield/ros2_SimRealRobotControl.git
# 2. Install:
cd ~/dev_ws
colcon build
```

(2.8) Install the ur3_CranfieldRobotics repository

In our workshop, all demonstrations will use a UR3 robot manipulator. This repository contains the UR3-specific configuration, description files, and resources needed to operate the UR3 Robot within the ros2srrc framework, both in simulation and on real hardware.

```
# 1. Download:
cd ~/dev_ws/src
git clone https://github.com/IFRA-Cranfield/ur3_CranfieldRobotics.git
# 2. Install:
cd ~/dev_ws
colcon build
```

Note: These installation steps are based on the official installation guide from the ros2_SimRealRobotControl GitHub repository.

----

[3] **Install OpenCV and YOLO for Object Pose Estimation Tasks**

As part of the workshop, we will perform an object pose estimation task in which the robot uses vision to detect and localize objects for manipulation.

To enable this, you will need to install OpenCV (for computer vision processing) and YOLO (You Only Look Once – a state-of-the-art object detection framework).

(3.1) Install OpenCV

We will use the opencv-contrib-python package, which includes both the main OpenCV modules and additional contributed modules (such as tracking algorithms) useful for advanced perception tasks.

```
pip install opencv-contrib-python
```

(3.2) Install YOLO

We will use the Ultralytics YOLO Python package, which provides a modern, user-friendly API for YOLOv8 and later versions. This will allow real-time object detection and pose estimation within our ROS 2 framework.

```
pip install ultralytics
```

Performance Tip: For this task, having a dedicated GPU is highly recommended to achieve real-time performance.

- *If your system has an NVIDIA GPU, ensure you have the correct NVIDIA drivers and CUDA toolkit installed before the workshop.*
- *You can follow NVIDIA's official guide here: CUDA Installation Guide for Linux.*

----

[4] **Install the workshop's GitHub Repository**

This repository contains all the exercises, resources, and example solutions that will be used during the workshop. Having it installed in advance will ensure you can follow each step, replicate demonstrations, and continue experimenting after the session.

```
# 1. Download:
cd ~/dev_ws/src
git clone https://github.com/IFRA-Cranfield/ROSCon_UK_2025.git
# 2. Install:
cd ~/dev_ws
colcon build
```

And… You're ready for the workshop!

Workshop Preparation: Installation Steps (OPTION B, backup – Virtual Machine)

For participants who do not have access to a laptop running Ubuntu 22.04 natively, we have prepared a pre-configured Virtual Machine (VM) that can be run using VMware.

This VM includes:

- Ubuntu 22.04 LTS.
- ROS 2 Humble fully installed and configured.
- All required IFRA-Cranfield packages (including ros2_SimRealRobotControl and ur3_CranfieldRobotics) already set up and ready to use.

Limitations:

- OpenCV and YOLO are not installed in the VM, as they require significant processing power and are not practical to run inside a virtualized environment.
- Performance may be slower compared to a native installation, especially for simulation and real-time robot control tasks.

Requirements to use the VM:

- VMware Workstation Player (Windows/Linux) or VMware Fusion (macOS) installed on your host machine.
- At least 4 GB of RAM (8GB recommended) and 2 CPU cores allocated to the VM.
- At least 30 GB of free disk space to store and run the VM image.

The VM is intended as a backup option. For the best workshop experience we strongly recommend using a native Ubuntu 22.04 installation.

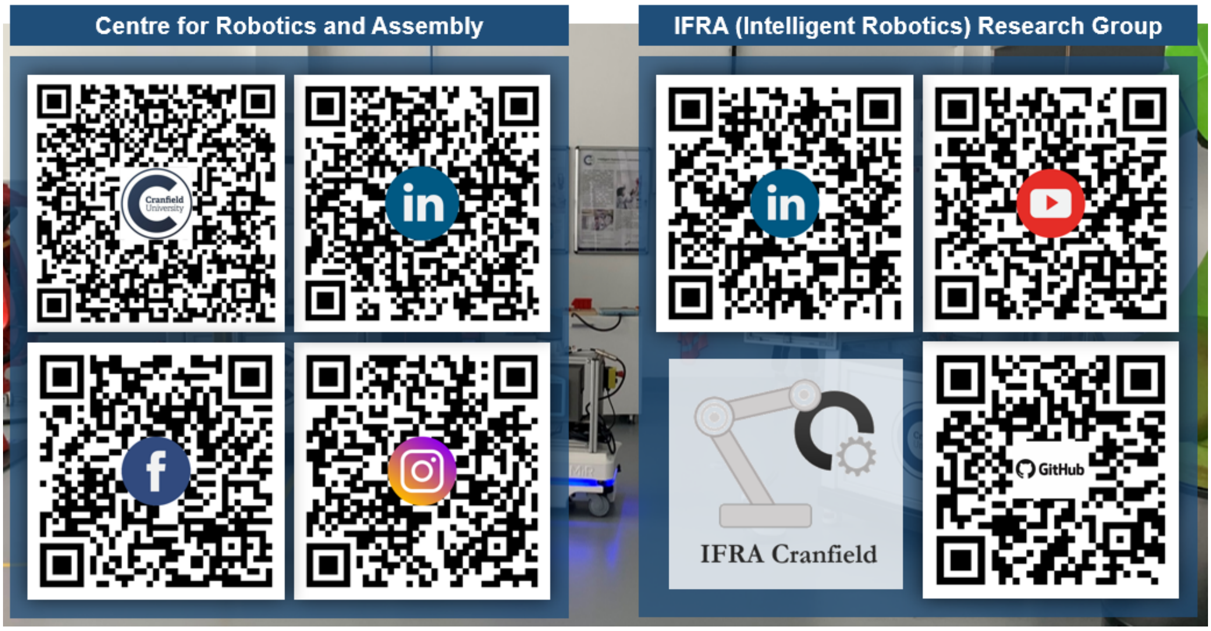
If this applies to you, please send an email to Mikel.Bueno-Viso@cranfield.ac.uk and we will be more than happy to provide you with a link to download the VM.

How do I know if all required packages have been installed successfully?

If you would like to test whether your installation has been completed successfully and everything is working as expected, please visit: UR3 CranfieldRobotics – Example Exercises, and follow the exercises provided there. These will help you verify that your ROS 2 environment, packages, and dependencies are correctly set up before the workshop.

If you have any questions or encounter any issues during the installation or setup process, please do not hesitate to contact us — we will be more than happy to assist you and ensure your environment is ready for the workshop.

Thank you for attending our workshop, looking forward to seeing you at Edinburgh!
Mikel & Seemal, IFRA-Cranfield


Centre for Robotics and Assembly
IFRA (Intelligent Robotics) Research Group
Cranfield University
in
f
in
IFRA Cranfield
GitHub