AURORA ROBOTICS CORE WORKSHOP 1.0

ROBOTIC ARM MODELLING

TODAY'S CLASS RULES

- All mics should be muted, unmute to interput me anytime cos things fit change very quickly😂
- I am always looking at the chat box so feel free drop questions anytime
- Feel free to use the "raise your hand" option anytime
- I recommend you have a note (could be digital), I tend to drop random knowledge casually

INCOMING COSPLAY

PLAY ALONG

“Aurora Robotics & Its Students in the Headlines!”

In a stunning development, the robotics community is buzzing again, Aurora Robotics Students have caught the attention of global industry leaders!

📢 “REPORTS CONFIRM THAT THE NEW WAVE OF TALENT RISING FROM AURORA ROBOTICS IS SETTING A NEW BENCHMARK IN ROBOTICS ENGINEERING.”

“Major Robotics Companies Reach Out to Aurora Students!”

After witnessing the impressive skills and teamwork demonstrated during recent workshops,
several leading robotics manufacturers have officially contacted Aurora Robotics Students.

BEFORE WE GO FAR,
COLLECT DISCLAIMER JOOR,
YOU THINK SAY I FORGET?

All slides, code and materials will be shared in the training repo. Feel free to use and share but do not modify

(I am very good with lawsuits)

FANUC M-710iC/50, Franka Emika, Kinova Robotics, KUKA Robotics, UFactory, Unitree Robotics, Universal Robots, one task!

We’ve built the mechanical design, we want you to handle the ROS2 Integration!

WHAT 'ROS INTEGRATION' ACTUALLY MEANS (FOR ROBOTIC ARMS)

When a company says “We want you to handle the ROS 2 integration,” they mean:

“Take our robot’s mechanical design (CAD or 3D model) and make it usable, controllable, and testable within the ROS 2 ecosystem.”

ROBOT DESCRIPTION LAYER (URDF/XACRO + VISUALIZATION)

This is always the first step of ROS2 Integration, it’s the foundation of everything else.

- Basically, we convert the CAD into meshes (.stl or .dae)
- Define links, joints, origins, and limits in a URDF or Xacro
- Add inertial, collision, and visual elements
- Visualize the robot in RViz 2
- This ensures the robot is properly defined and physically consistent in our ROS2 Ecosystem.

MOTION PLANNING INTEGRATION (MOVEIT 2)

Once your URDF is verified, the next layer is MoveIt 2, which handles:

- Kinematics (FK/IK): Understanding how joints move the end-effector
- Path & trajectory planning, calculating smooth, collision-free motion
- Motion execution: sending commands to controllers (real or simulated)
- Collision checking: ensuring safe paths between obstacles and links

So here, ROS integration involves connecting your URDF + SRDF + robot controllers to MoveIt 2 for motion planning.

CONTROL LAYER (ROS2_CONTROL + GAZEBO SIMULATION)

Controllllll!!! (I so much love this field)

- This adds realistic actuation and dynamics
- You can define joints, transmissions, and controllers in a ros2_control YAML file
- Load the gazebo_ros2_control plugin into the URDF
- Simulate movement in Gazebo
- This lets you test trajectories and control algorithms before touching real hardware.

HARDWARE INTEGRATION

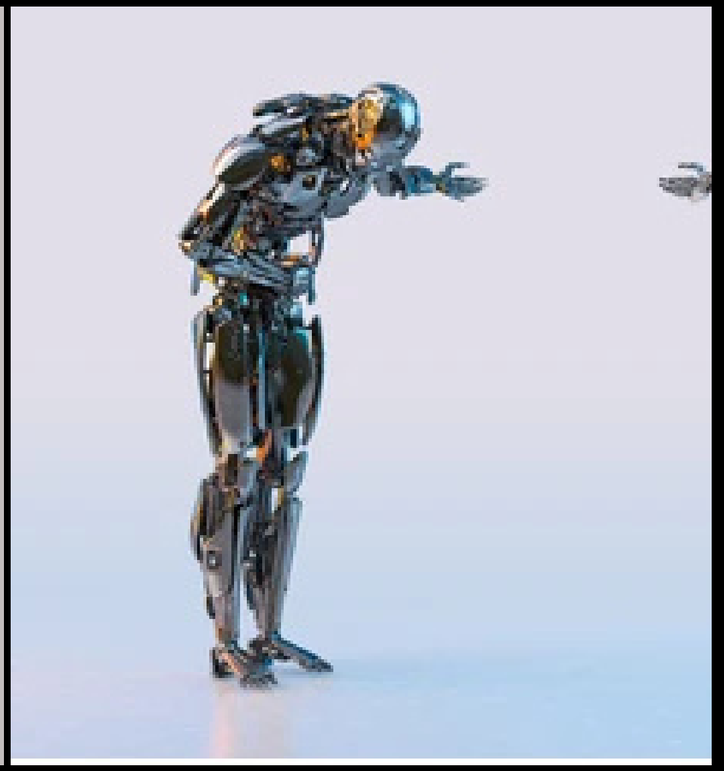
If it’s a real robot arm (like a UR, KUKA, or Kinova), you then connect ROS 2 nodes to the robot’s hardware API, driver, or firmware for live motion control.

'UNDERSTANDING URDF'

- URDF is an XML-based format used to describe a robot's physical structure.
- Defines links, joints, materials, geometry, and inertial properties.
- Every robot simulation, MoveIt setup, and control interface starts with a URDF.

TASK 4 REPO

THANK YOU FOR LISTENING



```
ros2 launch urdf_tutorial display.launch.py
model:=path/to/robot.urdf
```