Week 9: ROS2

ROS2 Introduction Hybrid 4 + Lab4

- Hybrid ROS2 introduction: ROS2 introductory tutorials.
- Lab – 4 ROS2 installation
- https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools.html
    - Nodes
    - Topics
    - Services
    - Actions

The big Picture

- Create Python packages **aisd_vision** and **aisd_motion**, use **MediaPipe Hands + OpenCV** to **detect hand positions and command motion**.
- Create **aisd_hearing** packages using **Whisper (STT) and gTTS (TTS)**; record audio, transcribe speech.
- Deploy these packages in real application (Create3 robot)

The big Picture

OpenCV (Open-Source Computer Vision Library)

A powerful Python/C++ library for working with images and videos.

Handles low-level tasks like capturing frames from a webcam, resizing, color conversion, drawing, etc.

OpenCV reads the camera stream and provides each frame to the next stage (MediaPipe).

MediaPipe Hands

A **pre-trained hand-tracking model** developed by Google.

Detects **21 key landmarks** (fingertips, joints, wrist) for each hand in real time.

Returns normalized (x, y, z) positions of each point in the image.

The big Picture

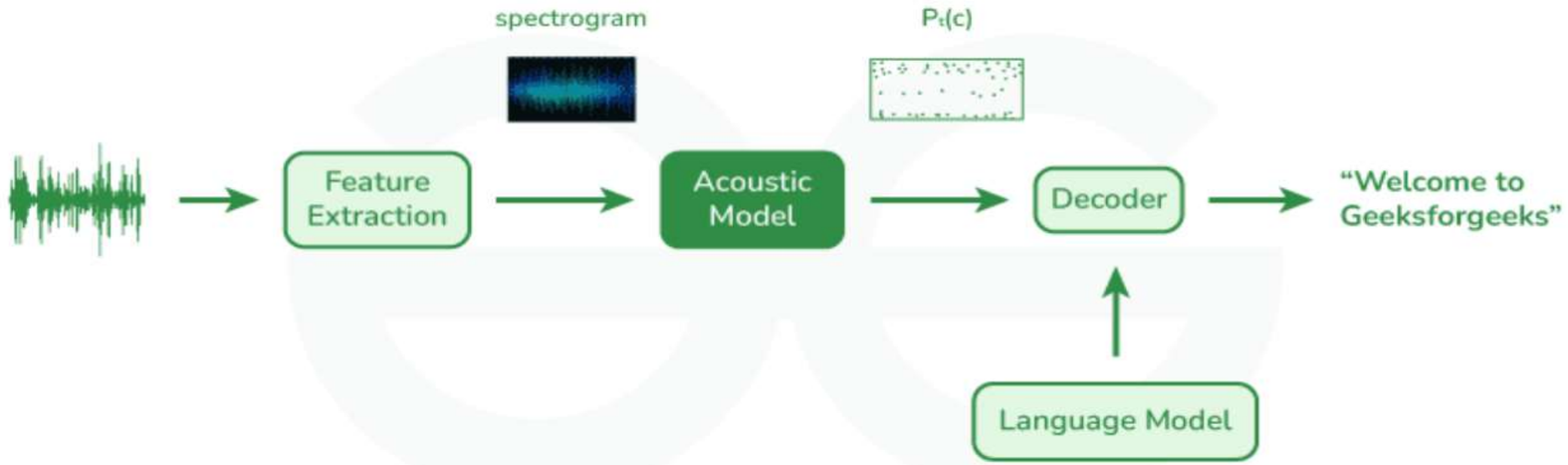
Whisper (Speech-to-Text, STT)

An **AI speech-recognition model** by **OpenAI**.

Converts spoken audio (from a microphone or WAV file) into written text.

Handles many languages and accents, robust to noise.

In your aisd_hearing package, it turns your voice into text strings.



The big Picture

- Train Physical Reinforcement learning agents (Next semester)

ROS2 Ecosystem

- ROS2 distributions (version): https://docs.ros.org/en/rolling/Releases.html
- We will use Humble

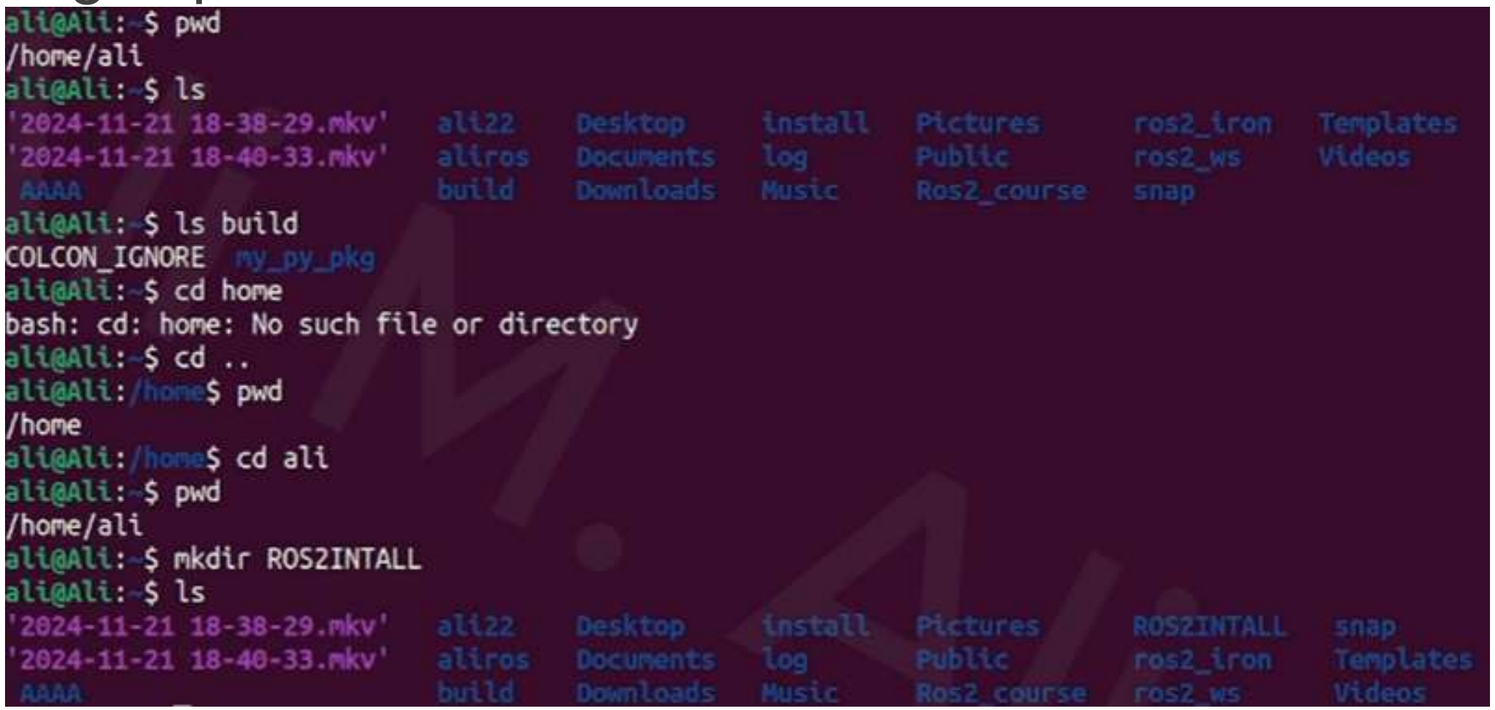
Humble Hawksbill	May 23, 2022		May 2027	Audrow Nash

ROS2 Ecosystem

Installations

- https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html

- If you never worked in Ubuntu, take a look at the Linux revision posted in Brightspace. You need to know basic commands like these

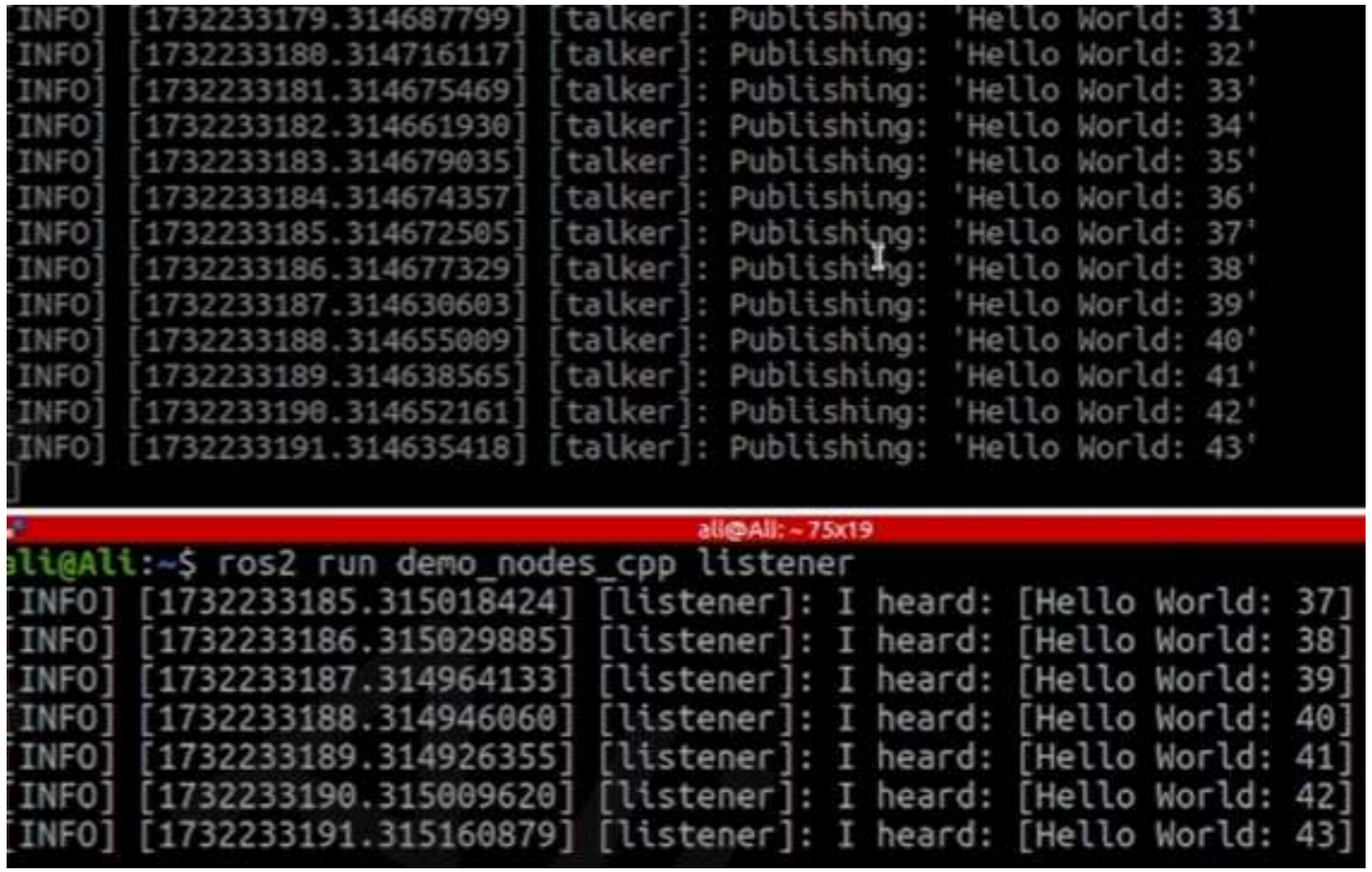


ROS2 Ecosystem

- After you install ROS2 humble distribution, check if it working or not. If you type ROS2 and then double tab you will see a list commands can be used.
- Try ros2 run demo_nodes_cpp talker ,,,This will run a C++ node called Talker inside a C++ Package called demo_nodes_cpp. The node will publish Hello World every second.

ROS2 Ecosystem

- Now in anther terminal you can type ros2 run demo_nodes_cpp listener. This will run a C++ node that will listen to the published message and Type “I heard:….”



Building Packages in ROS2

- We will use a Library called Colcon.
  - https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Colcon-Tutorial.html

- Install Colcon as follows

```
ali@Ali:~$ sudo apt install python3-colcon-common-extensions
[sudo] password for ali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-colcon-common-extensions is already the newest version (0.3.0-1).
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 364 not upgraded.
```

Building Workspace in ROS2

- Build now you first Workspace: using colcon build

```
ali@Ali:~$ pwd
/home/ali
ali@Ali:~$ ls
'2024-11-23 19-30-20.mkv'   install    ros2_iron
 build                      log        snap
 Desktop                    Music      Templates
 Documents                  Pictures   Videos
 Downloads                  Public
ali@Ali:~$ mkdir ros2_ws
ali@Ali:~$ ls
'2024-11-23 19-30-20.mkv'   install    ros2_iron
 build                      log        ros2_ws
 Desktop                    Music      snap
 Documents                  Pictures   Templates
 Downloads                  Public     Videos
ali@Ali:~$ cd ros2_ws
ali@Ali:~/ros2_ws$ colcon build

Summary: 0 packages finished [0.17s]
ali@Ali:~/ros2_ws$ ls
build  install  log
ali@Ali:~/ros2_ws$ ls install
COLCON_IGNORE     _local_setup_util_ps1.py  setup.ps1
local_setup.bash  _local_setup_util_sh.py   setup.sh
local_setup.ps1   local_setup.zsh           setup.zsh
local_setup.sh    setup.bash
```

Building Workspace in ROS2

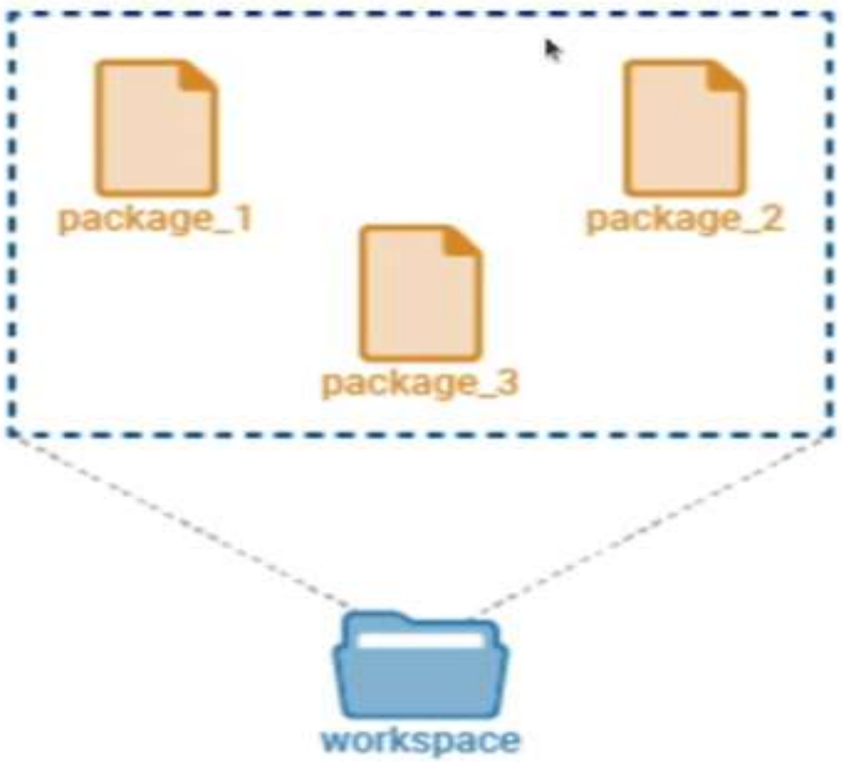
- Build now you first Packages: using colcon build

```
ali@Ali:~$ pwd
/home/ali
ali@Ali:~$ ls
'2024-11-23 19-30-20.mkv'   install    ros2_iron
 build                      log        snap
 Desktop                    Music      Templates
 Documents                  Pictures   Videos
 Downloads                  Public
ali@Ali:~$ mkdir ros2_ws
ali@Ali:~$ ls
'2024-11-23 19-30-20.mkv'   install    ros2_iron
 build                      log        ros2_ws
 Desktop                    Music      snap
 Documents                  Pictures   Templates
 Downloads                  Public     Videos
ali@Ali:~$ cd ros2_ws
ali@Ali:~/ros2_ws$ colcon build

Summary: 0 packages finished [0.17s]
ali@Ali:~/ros2_ws$ ls
build  install  log
ali@Ali:~/ros2_ws$ ls install
COLCON_IGNORE     _local_setup_util_ps1.py  setup.ps1
local_setup.bash  _local_setup_util_sh.py   setup.sh
local_setup.ps1   local_setup.zsh           setup.zsh
local_setup.sh    setup.bash
```

Building Packages in ROS2

- Every workspace can have serval packages. Each package contains modules (codes that send and receives messages).



Building Packages in ROS2

- Every workspace can have serval packages. Each package contains modules (codes that send and receives messages).
- You can build package a package called my_py_pkg using ament as follows:
- Note that you need rclpy to be able to write pyhton codes.

```
ali@Ali:~$ pwd
/home/ali
ali@Ali:~$ ls
'2024-11-24 19-05-27.mkv'   Desktop     Downloads   log      Pictures   ros2_iron   snap        Videos
 build                      Documents   install     Music    Public     ros2_ws     Templates
ali@Ali:~$ ls ros2_ws
build  install  log
ali@Ali:~$ cd ros2_ws
ali@Ali:~/ros2_ws$ mkdir src
ali@Ali:~/ros2_ws$ ls
build  install  log  src
ali@Ali:~/ros2_ws$ cd src
ali@Ali:~/ros2_ws/src$ ros2 pkg create my_py_pkg --build-type ament_python --dependencies rclpy
```

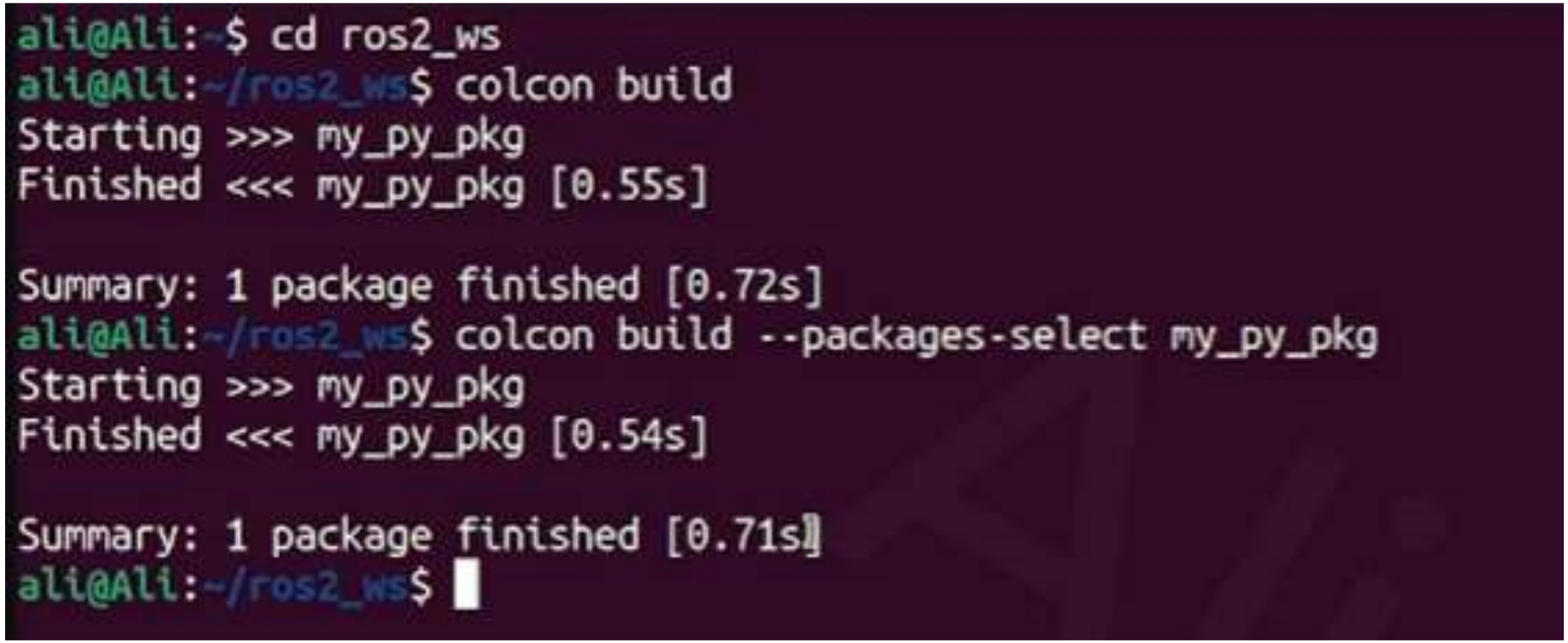
Building Packages in ROS2

- Every workspace can have serval packages. Each package contains modules (codes that send and receives messages).
- You can build package a package called my_py_pkg using ament as follows:
- Note that you need rclpy to be able to write pyhton codes.

```
ali@Ali:~$ pwd
/home/ali
ali@Ali:~$ ls
'2024-11-24 19-05-27.mkv'   Desktop     Downloads   log      Pictures   ros2_iron   snap        Videos
 build                      Documents   install     Music    Public     ros2_ws     Templates
ali@Ali:~$ ls ros2_ws
build  install  log
ali@Ali:~$ cd ros2_ws
ali@Ali:~/ros2_ws$ mkdir src
ali@Ali:~/ros2_ws$ ls
build  install  log  src
ali@Ali:~/ros2_ws$ cd src
ali@Ali:~/ros2_ws/src$ ros2 pkg create my_py_pkg --build-type ament_python --dependencies rclpy
```

Building Packages in ROS2

- Now check that your Python Package was build.
- You need to redo “colcon build” every time you install or update any of the packages inside the workspace.



Building Nodes in ROS2

```
ali@Ali:~$ cd ros2_ws
ali@Ali:~/ros2_ws$ ls
build  install  log  src
ali@Ali:~/ros2_ws$ cd src
ali@Ali:~/ros2_ws/src$ ls
my_cpp_pkg  my_py_pkg
ali@Ali:~/ros2_ws/src$ cd my_py_pkg
ali@Ali:~/ros2_ws/src/my_py_pkg$ ls
my_py_pkg  package.xml  resource  setup.cfg  setup.py  test
ali@Ali:~/ros2_ws/src/my_py_pkg$ cd my_py_pkg
ali@Ali:~/ros2_ws/src/my_py_pkg/my_py_pkg$ touch my_first_node.py
```

Building Nodes in ROS2