

# ROS 架構與流程

## 節點與功能：

節點名稱	訂閱主題	發布主題	功能描述
soldier	command	sensor	接收馬達指令執行動作，計算感測器數據，並發布感測器狀態
commander	sensor	command	訂閱感測器數據，計算並生成執行動作與平衡控制，並發布馬達指令

## 主題與數據：

主題名稱	資料來源	資料描述
sensor	soldier	包含 IMU、壓力感測數據，以及馬達當前狀態
command	commander	包含馬達目標位置、速度與控制模式指令

## 流程圖：

- [soldier 節點] →發布→ [sensor 主題] →訂閱→ [command 主題]
- [commander 節點] →發布→ [command 主題] →訂閱→ [sensor 主題]

## 節點詳細流程：

### listener.py（接收指令並執行動作）

主要功能：

- 訂閱“command”主題，接收來自“commander”的馬達指令。
- 根據收到的馬達目標指令與模式執行動作。
- 蒐集感測數據（IMU、壓力感測器），並發布到“sensor”主題。

核心流程：

- 初始化節點“soldier”。
- 訂閱“command”主題，更新當前馬達目標狀態。
- 執行動作（蒐集下一步的馬達狀態）。
- 周期性蒐集感測數據（IMU、壓力感測器）。
- 將馬達狀態與感測數據組合後發布到“sensor”主題。

### talker.py（控制與指令發布）

主要功能：

- 訂閱“**sensor**”主題，獲取感測器數據與馬達當前狀態。
- 執行動作控制，計算下一步馬達指令。
- 將計算結果發布到“**command**”主題。

核心流程：

- 初始化節點“**commander**”，建立訂閱與發布管道。

數據接收：

- 訂閱“**sensor**”主題，更新全域變數。
- **imu\_data**：IMU 資料，用於平衡控制。
- **l\_fsr** 和 **r\_fsr**：左右腳壓力感測數據。
- **pos**：馬達當前角度。

動作生成與控制：

- 載入動作序列（CSV 文件中的馬達位置動作）。
- 插值生成平滑動作曲線，並根據感測數據控制目標馬達位置。
- 將馬達指令（目標角度、速度、模式等）發布到“**command**”主題。

“**sensor**”主題（由“**soldier**”節點發布）

```
[[馬達 1 ID, 當前角度, 當前速度, 扭矩],
...
[馬達 13 ID, 當前角度, 當前速度, 扭矩],
[IMU_roll, IMU_pitch, IMU_yaw],
[左壓力傳感數據 1, ..., 左壓力傳感數據 4],
[右壓力傳感數據 1, ..., 右壓力傳感數據 4]]
```

“**command**”主題（由“**commander**”節點發布）

```
[[馬達 1 ID, 控制模式, 目標角度, 目標速度, 扭矩, kp, kd],
...
[馬達 13 ID, 控制模式, 目標角度, 目標速度, 扭矩, kp, kd]]
```