



CentraleSupélec

Soutenance

EI Drone

Equipe Pluto

8 novembre 2024

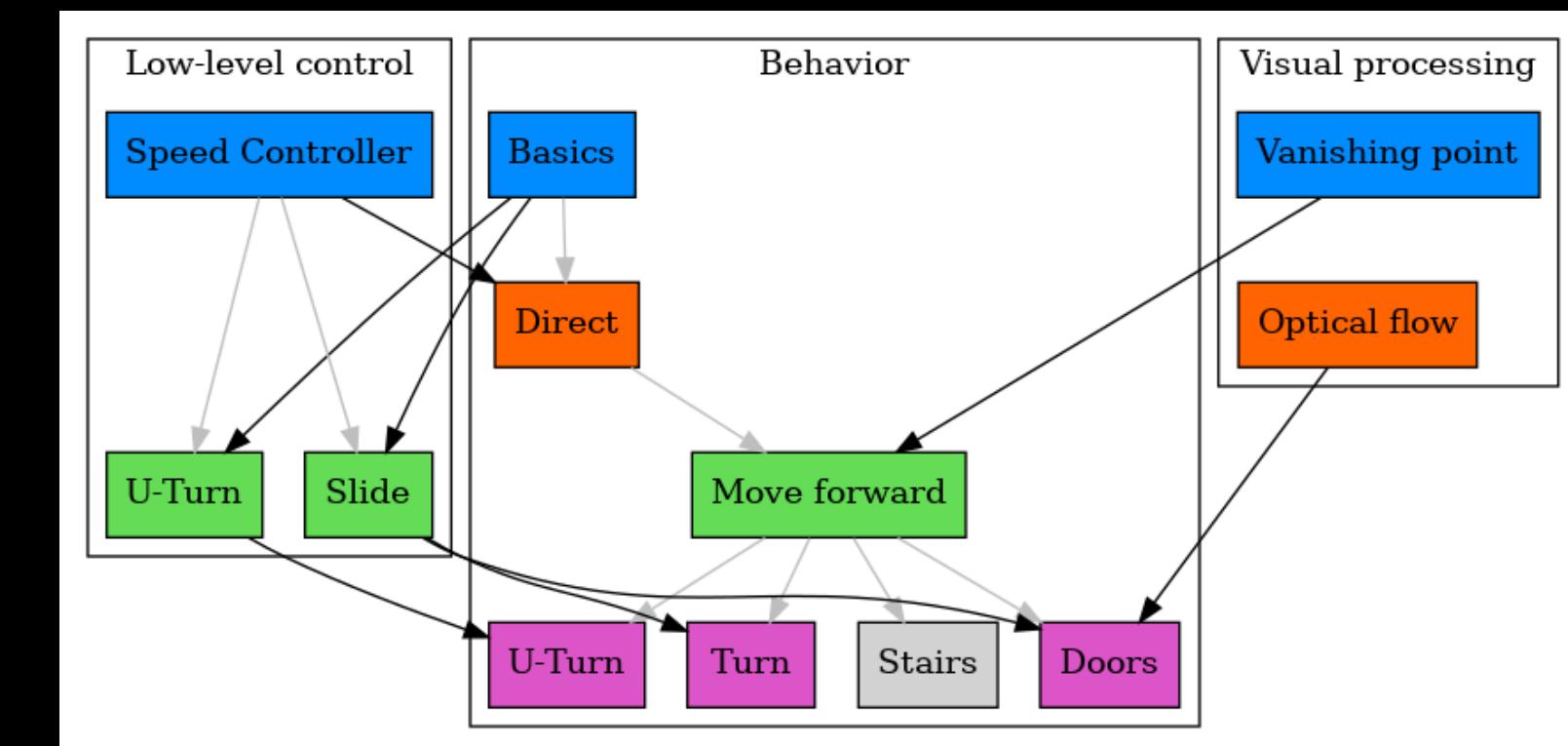
Sommaire

01 Fonctionnalités de contrôle

02 Fonctionnalités de comportement

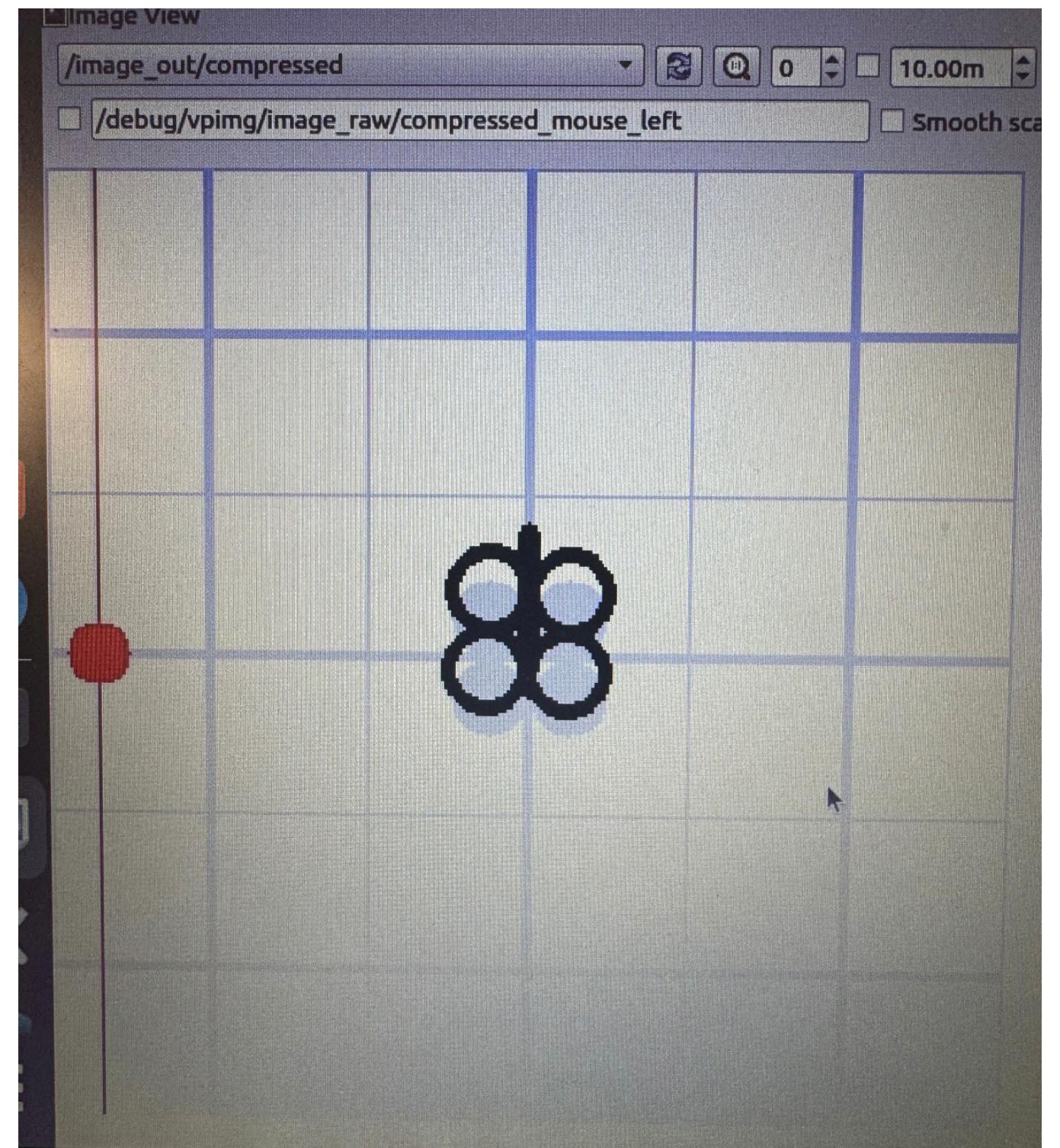
03 Traitement de flux vidéo

04 Résultats et test

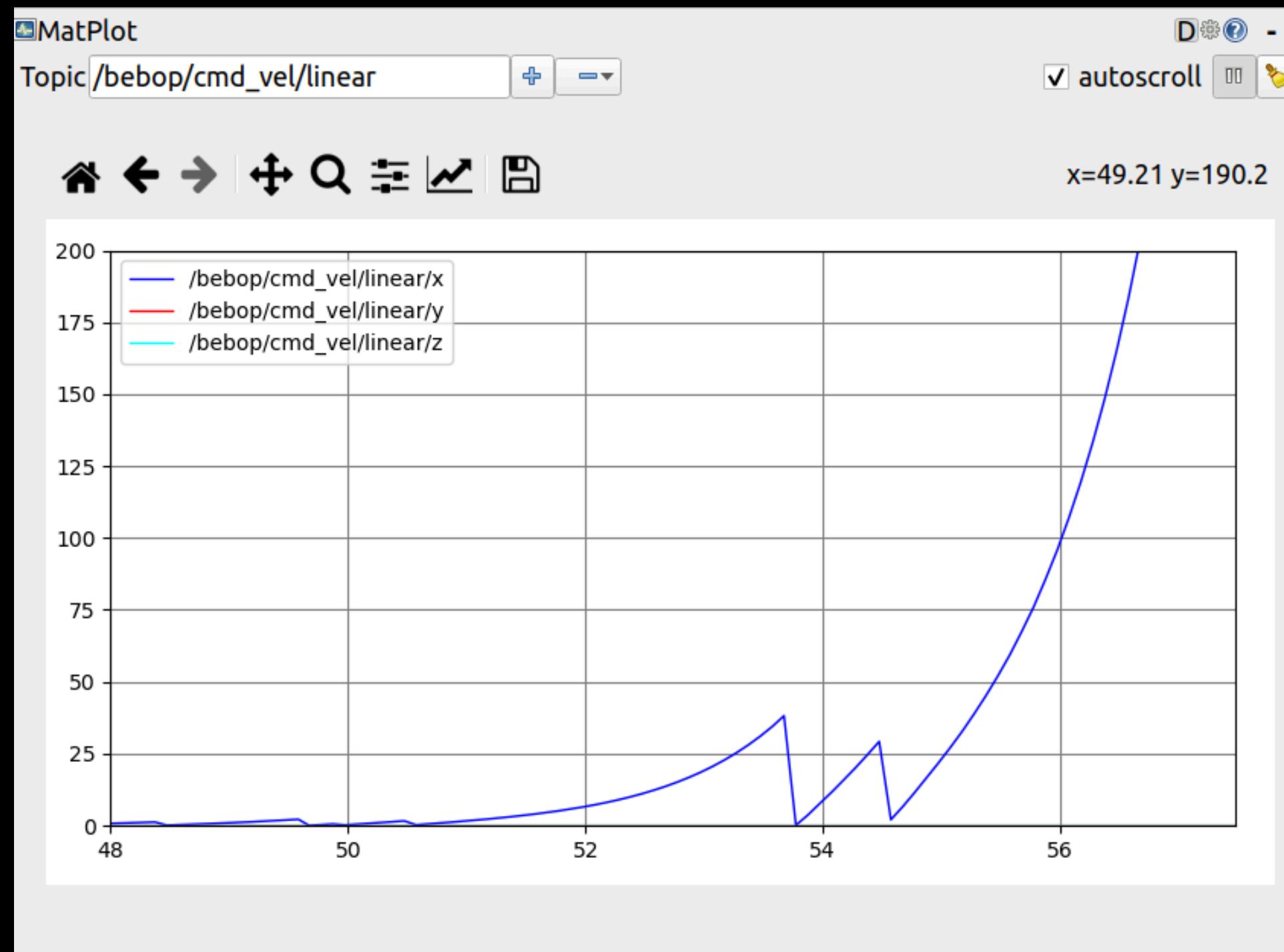


Premières fonctionnalités

- Twist2pic
- speed_controller
- Première version de joy_teleop



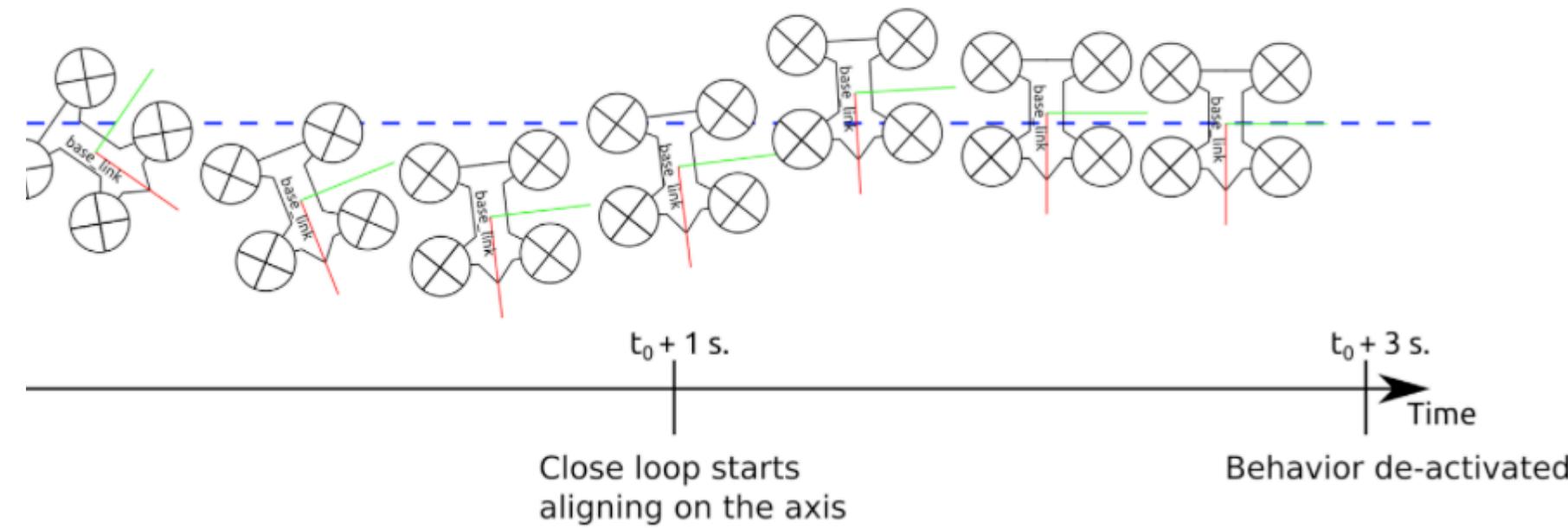
PID



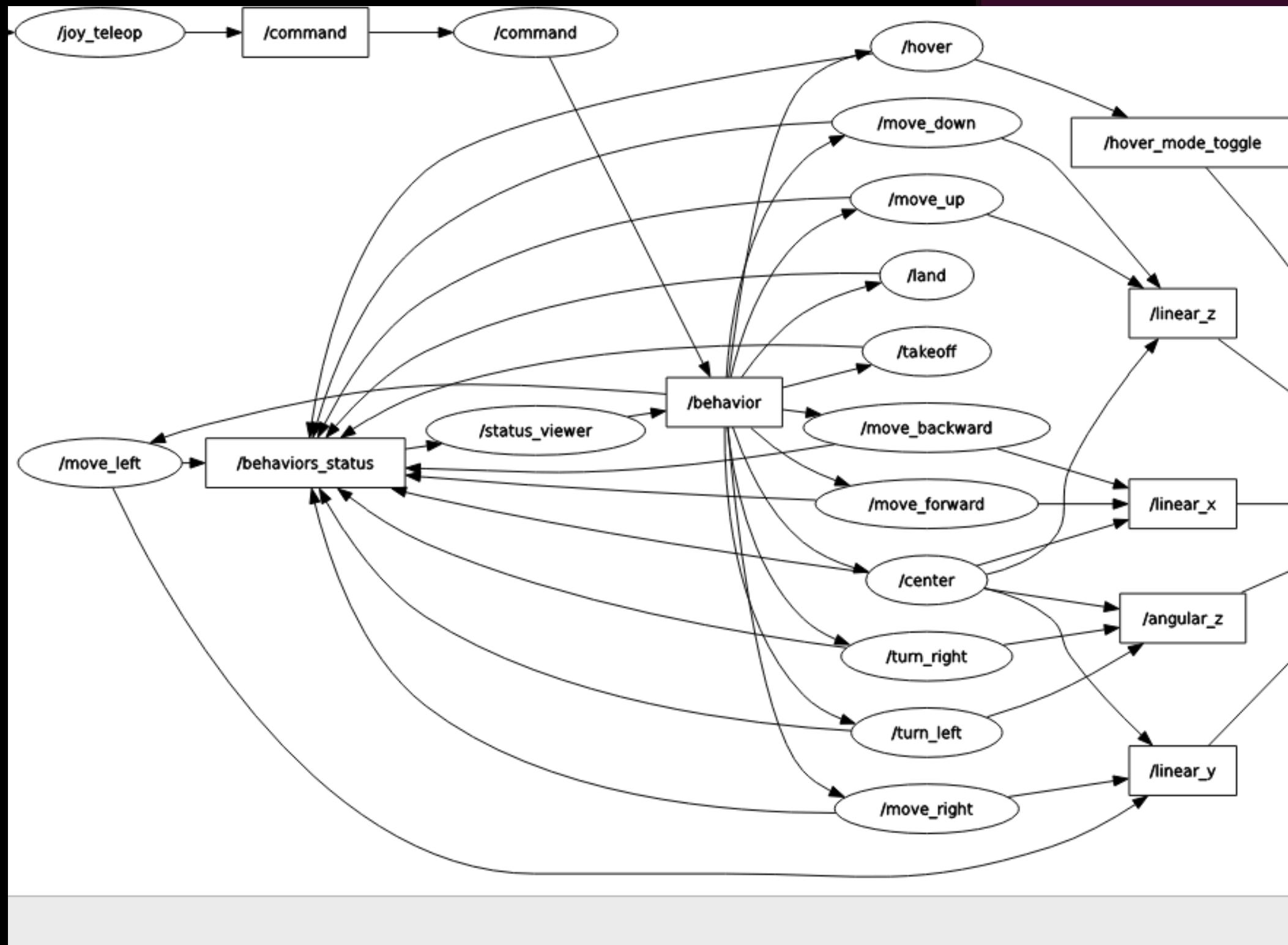
Slide et Uturn

- U_turn publish à angular z et lance pendant la bonne durée pour effectuer un Uturn

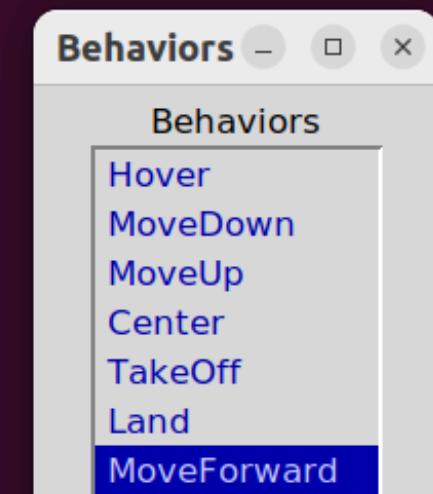
- Slide



Behaviors



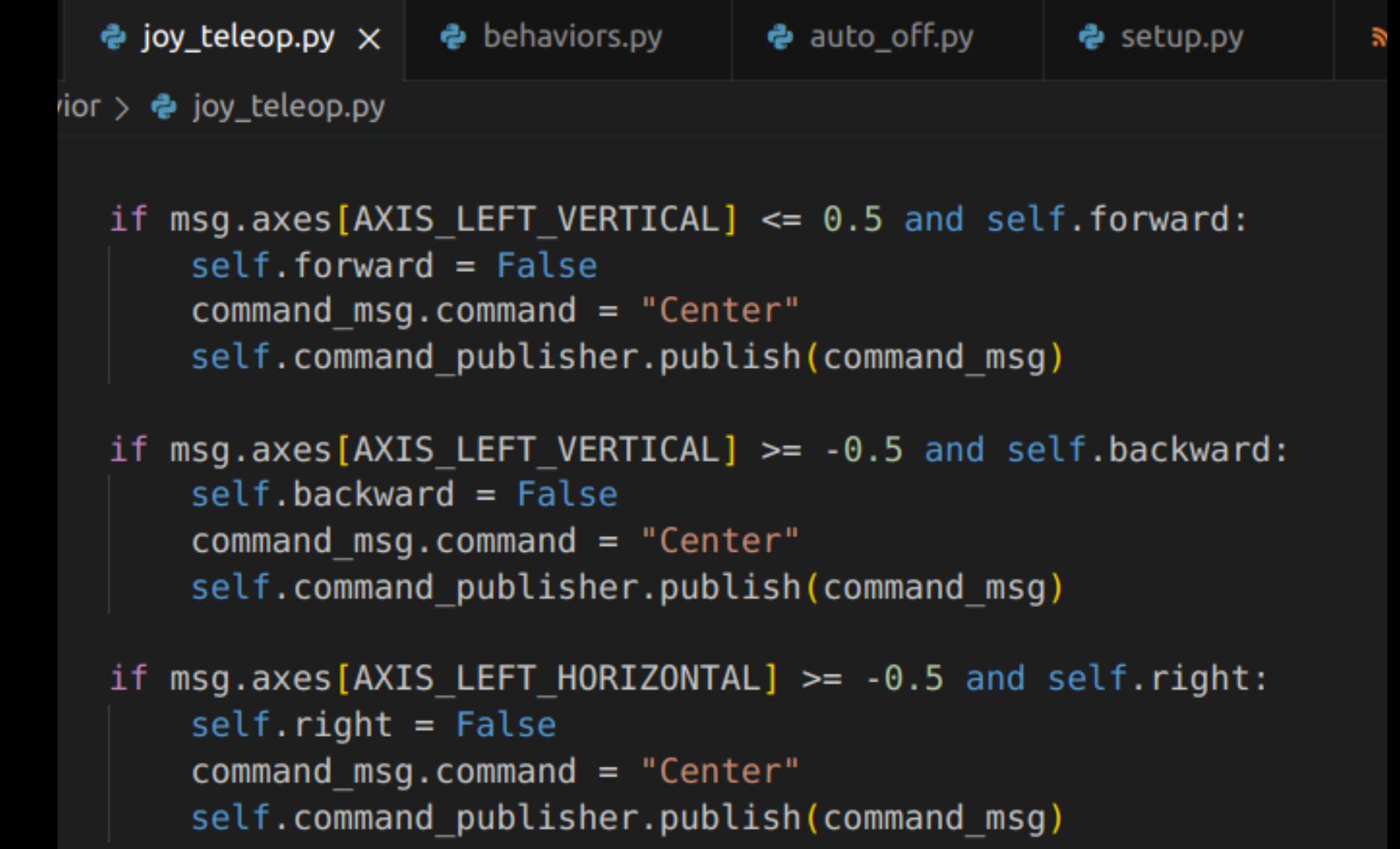
```
2023rocardra@mfopt13:~/ros2_ws$ ros2 topic echo linear_x  
data: 0.1  
---
```



Commandes

```
class Center(AutoOff):
    def __init__(self):
        super().__init__("Center")
        self.x_publisher = self.create_publisher(Float64,
        self.y_publisher = self.create_publisher(Float64,
        self.z_publisher = self.create_publisher(Float64,
        self.az_publisher = self.create_publisher(Float64,

    def on_status(self, status: bool):
        super().on_status(status)
        if status:
            self.timer.reset()
            vitesse = Float64()
            vitesse.data = 0.0
            self.x_publisher.publish(vitesse)
            self.y_publisher.publish(vitesse)
            self.z_publisher.publish(vitesse)
            self.az_publisher.publish(vitesse)
```



The image shows a code editor interface with a dark theme. In the top right corner, there is a tab bar with five tabs: 'joy_teleop.py X', 'behaviors.py', 'auto_off.py', 'setup.py', and another 'joy_teleop.py' tab which is currently active. The main workspace contains three blocks of Python code. The first block is identical to the one in the screenshot above. The second block starts with an if statement checking the left vertical axis. If the condition is met, it sets 'self.forward' to False, sets 'command_msg.command' to "Center", and publishes it via 'self.command_publisher'. The third block starts with another if statement checking the left vertical axis. If the condition is met, it sets 'self.backward' to False, sets 'command_msg.command' to "Center", and publishes it via 'self.command_publisher'. The fourth block starts with a third if statement checking the left horizontal axis. If the condition is met, it sets 'self.right' to False, sets 'command_msg.command' to "Center", and publishes it via 'self.command_publisher'.

```
if msg.axes[AXIS_VERTICAL] <= 0.5 and self.forward:
    self.forward = False
    command_msg.command = "Center"
    self.command_publisher.publish(command_msg)

if msg.axes[AXIS_VERTICAL] >= -0.5 and self.backward:
    self.backward = False
    command_msg.command = "Center"
    self.command_publisher.publish(command_msg)

if msg.axes[AXIS_HORIZONTAL] >= -0.5 and self.right:
    self.right = False
    command_msg.command = "Center"
    self.command_publisher.publish(command_msg)
```

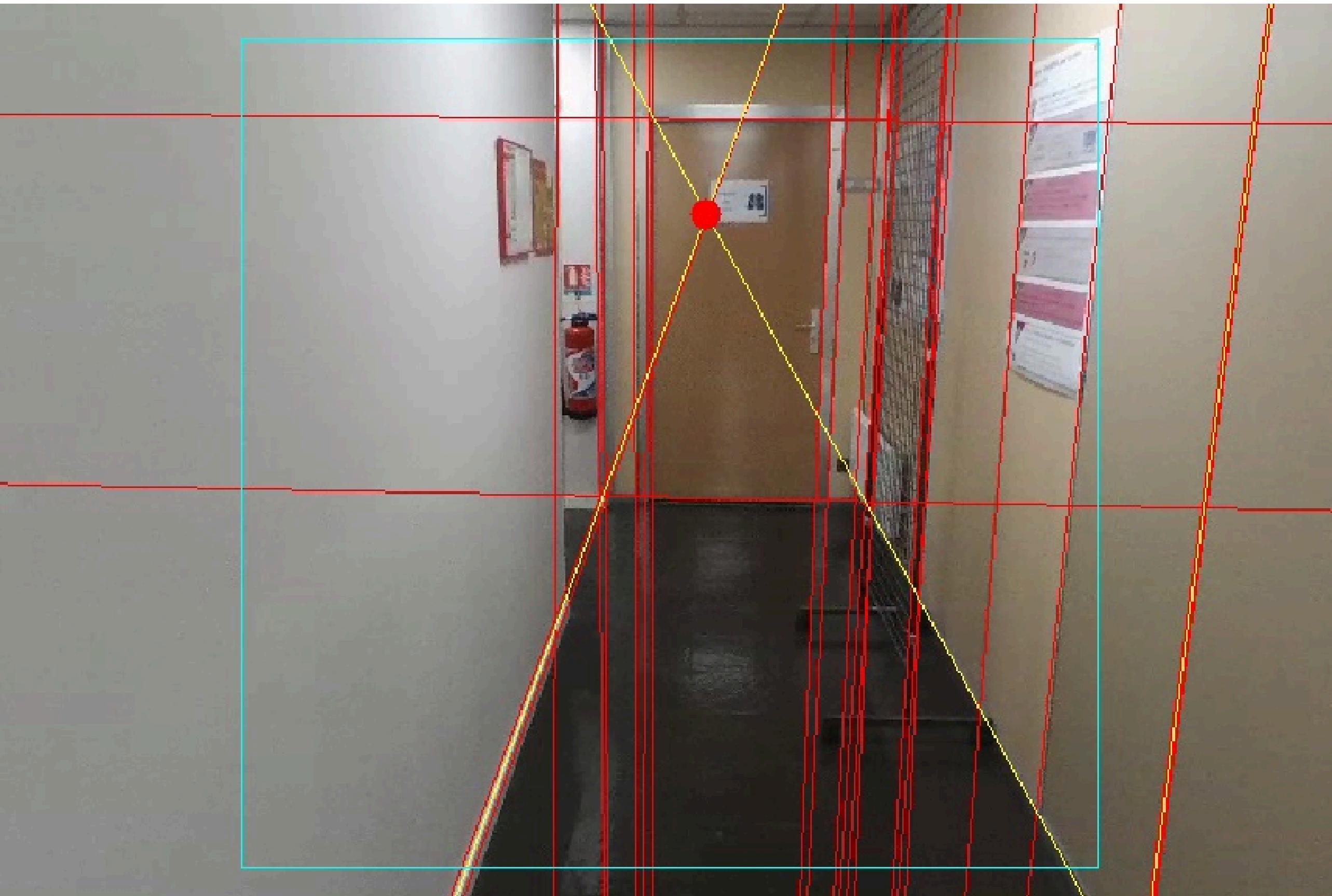
Commandes

```
2023rocardra@mfopt13:~/ros2_ws
command: TakeOff
---
command: Forward
---
command: Center
---
command: Left
---
command: Center
---
command: TurnRight
---
command: Center
---
command: Down
---
command: Center
---
command: EmergencyStop
---
command: Land
---
```

Vanishing point

Les filtres pour le VP :
ceiling, length, angle,
window, cluster

Valeurs adaptés du
turtlebot (ceiling plus
faible, fenêtre
différente)



Optical flow door detection



Partir de la mesure d'intensité sur deux images : obtenir le shift et du shift, obtenir les portes

Filtres appliqués à l'intensité, au shift

Zone de shift bas et de longueur suffisante = porte

Conclusion et test