

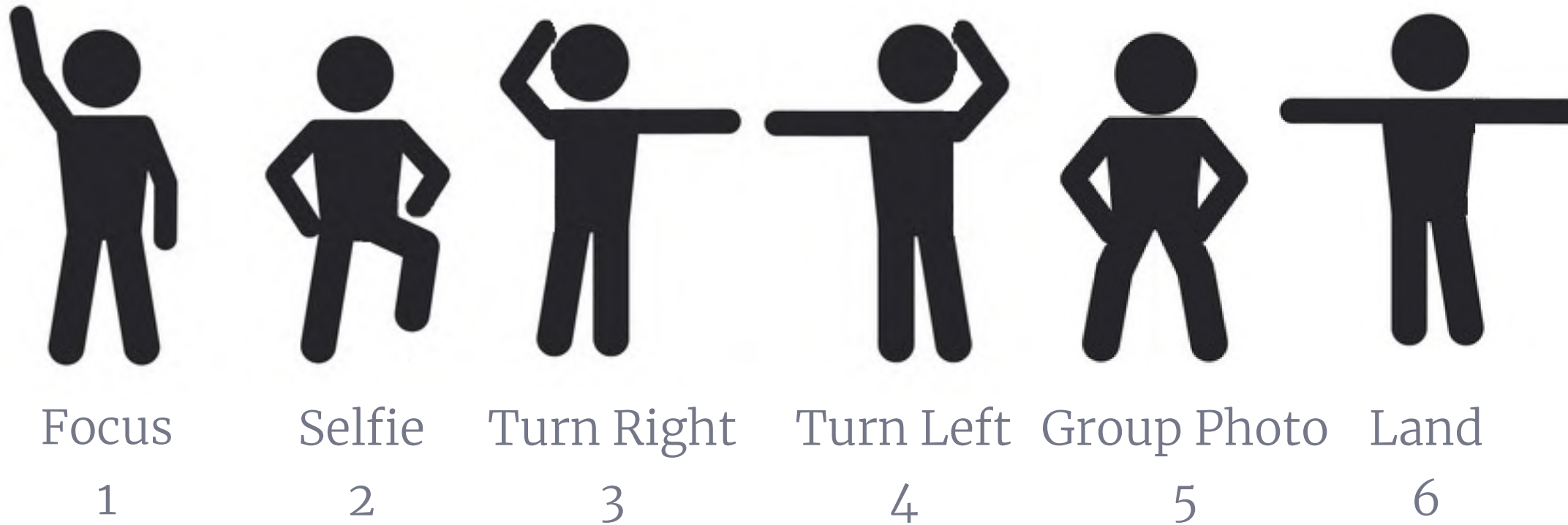
Team 3

# Selfie Drone

JIHYEOK KIM, RILEY TAYLOR, THEO MICHEL

**KAIST**

## GESTURES



Each gesture has a different purpose.

- Focus : Track and approach
- Selfie : Take a picture
- Turn Right : Rotate  $\pi/24$  right
- Turn Left : Rotate  $\pi/24$  left
- Group Photo: Move to the center of the group
- Landing : Go back to initial zone and land



# GESTURE DETECTION

## Code structure : Decompose in sub gestures

- Hand close to head

Measure distance between wrist and head



```
bool left_hand_close_to_face = (distance(left_wrist_pixel_x, left_wrist_pixel_y, nose_pixel_x,  
nose_pixel_y) < (0.25 * bbox_height))
```

- Hand on eye level

Measure if y axis is close to the one of the nose



```
bool right_hand_pointing = ((right_wrist_pixel_y < nose_pixel_y + (0.15 * bbox_height)) &&  
(right_wrist_pixel_y > nose_pixel_y - (0.10 * bbox_height)))
```

## Assemble the sub gestures



```
if (left_hand_close_to_face && right_hand_pointing)  
    gesture_id = 3; // left hand close to face, and right hand showing direction
```

---

# GESTURE DETECTION

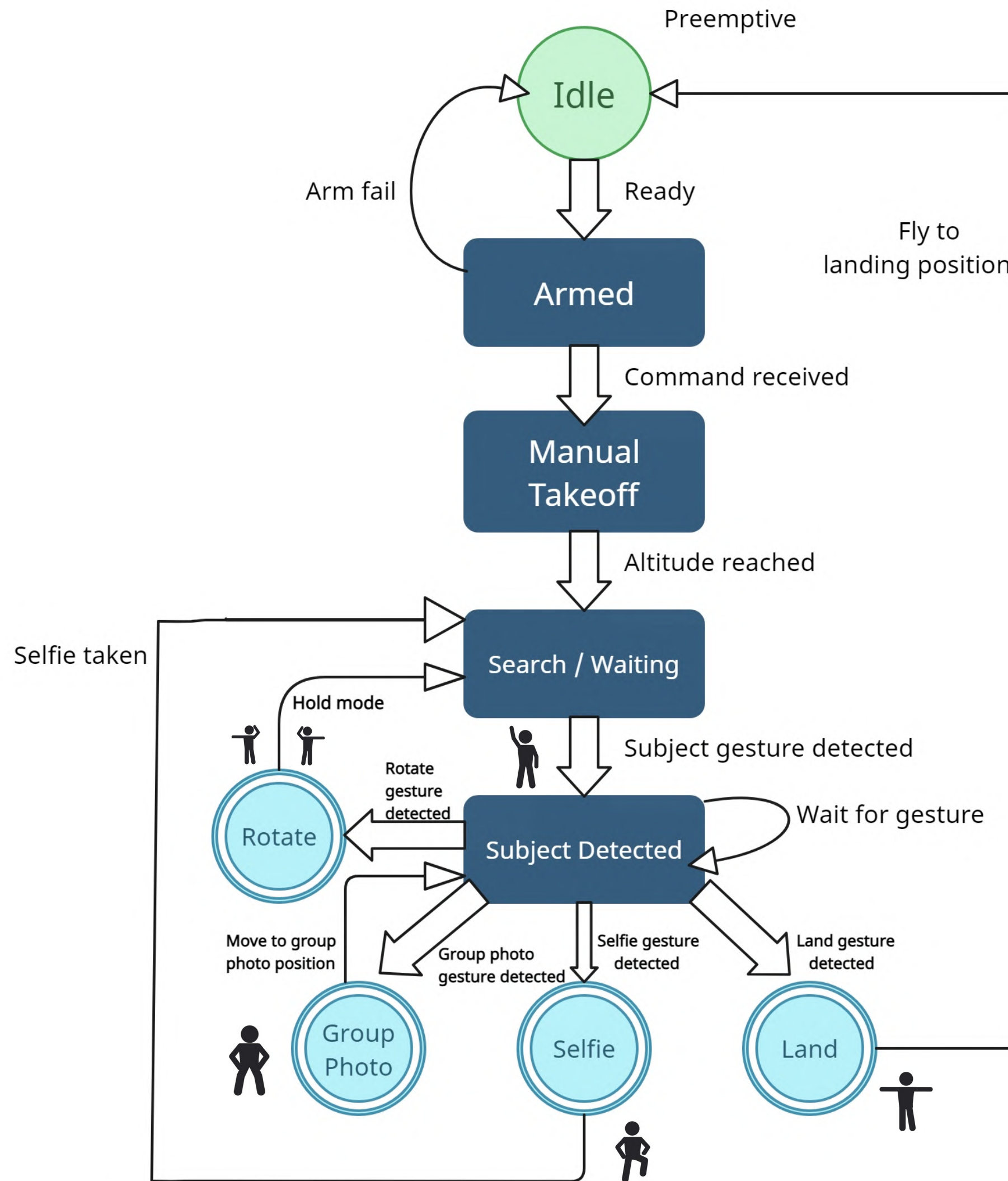
## Interesting learning points

1. Gesture has to be easily detectable in the context we are given, for example hands up gesture was easy to detect in lab, but really hard to detect in the 117 room
2. Gestures have to be non overlapping, you shouldn't pass through a gesture before going to another one, or it is dangerous for the control of the drone
3. You should always weight the gesture thresholds by a dimension of the subject (for example bbox\_height), in order to make the detection stable with distance

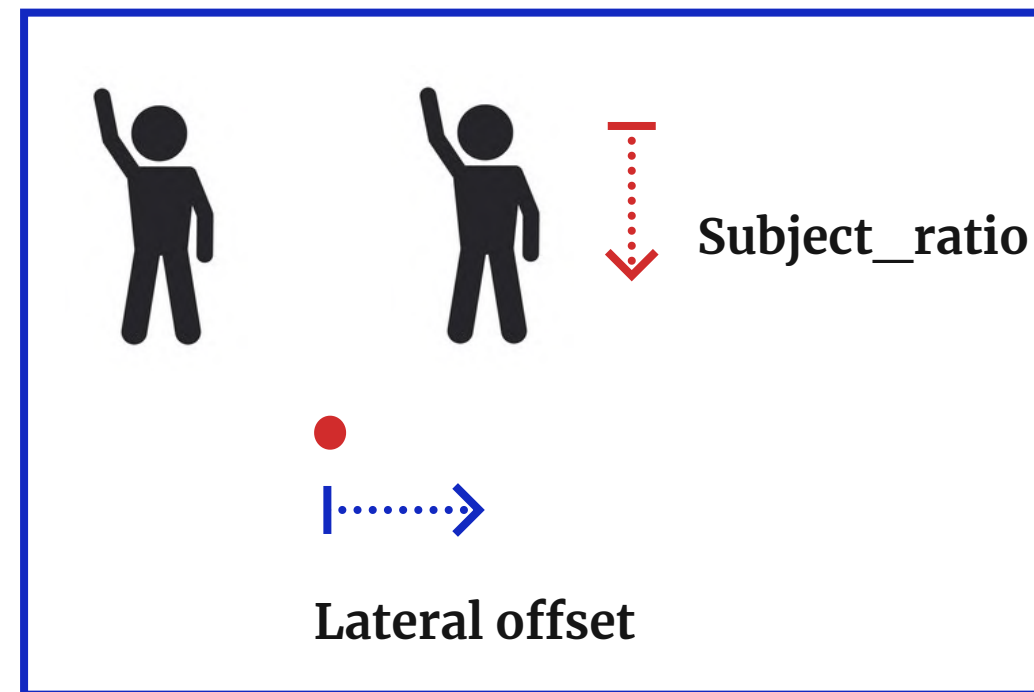
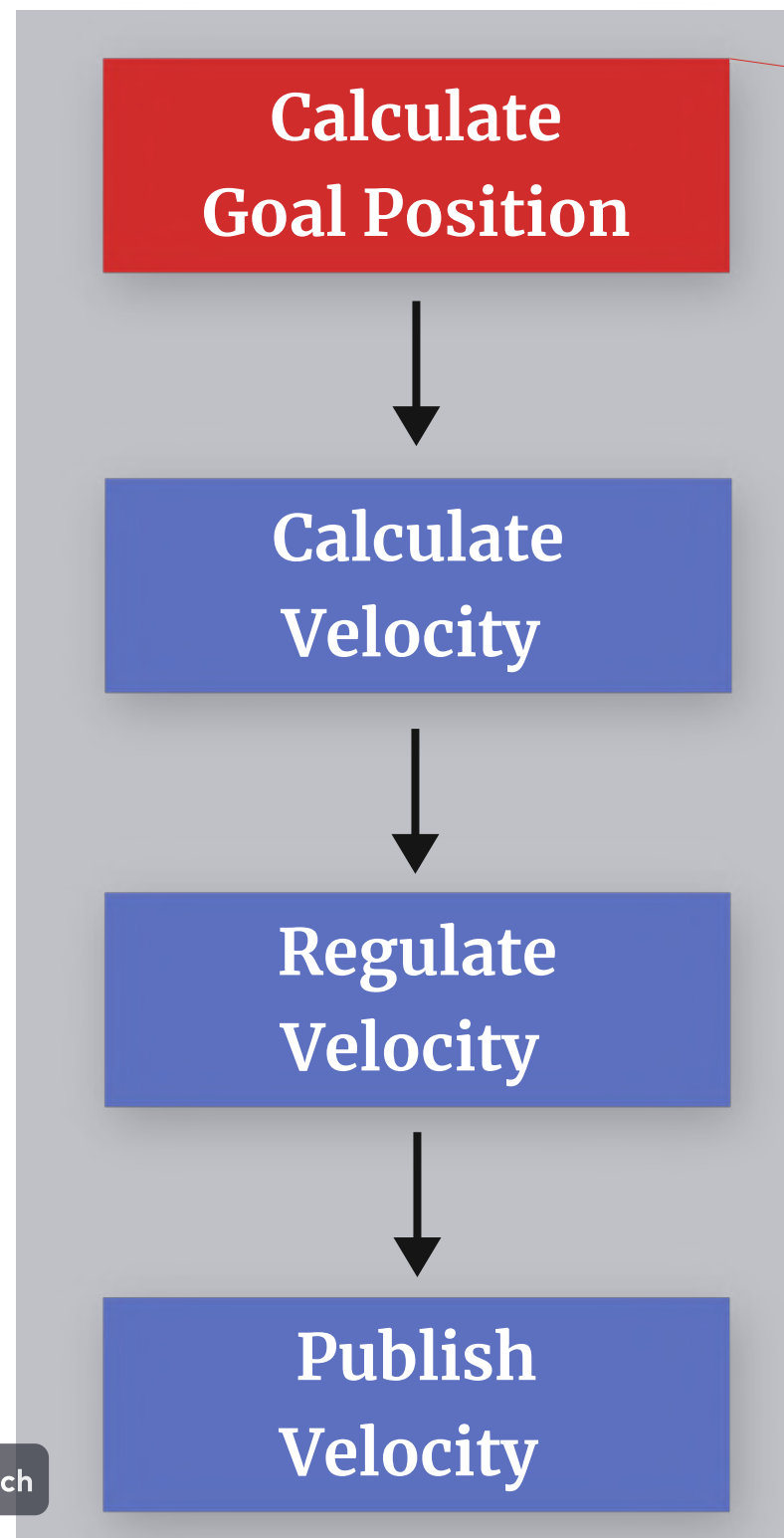


# FLIGHT PLANNER

State machine



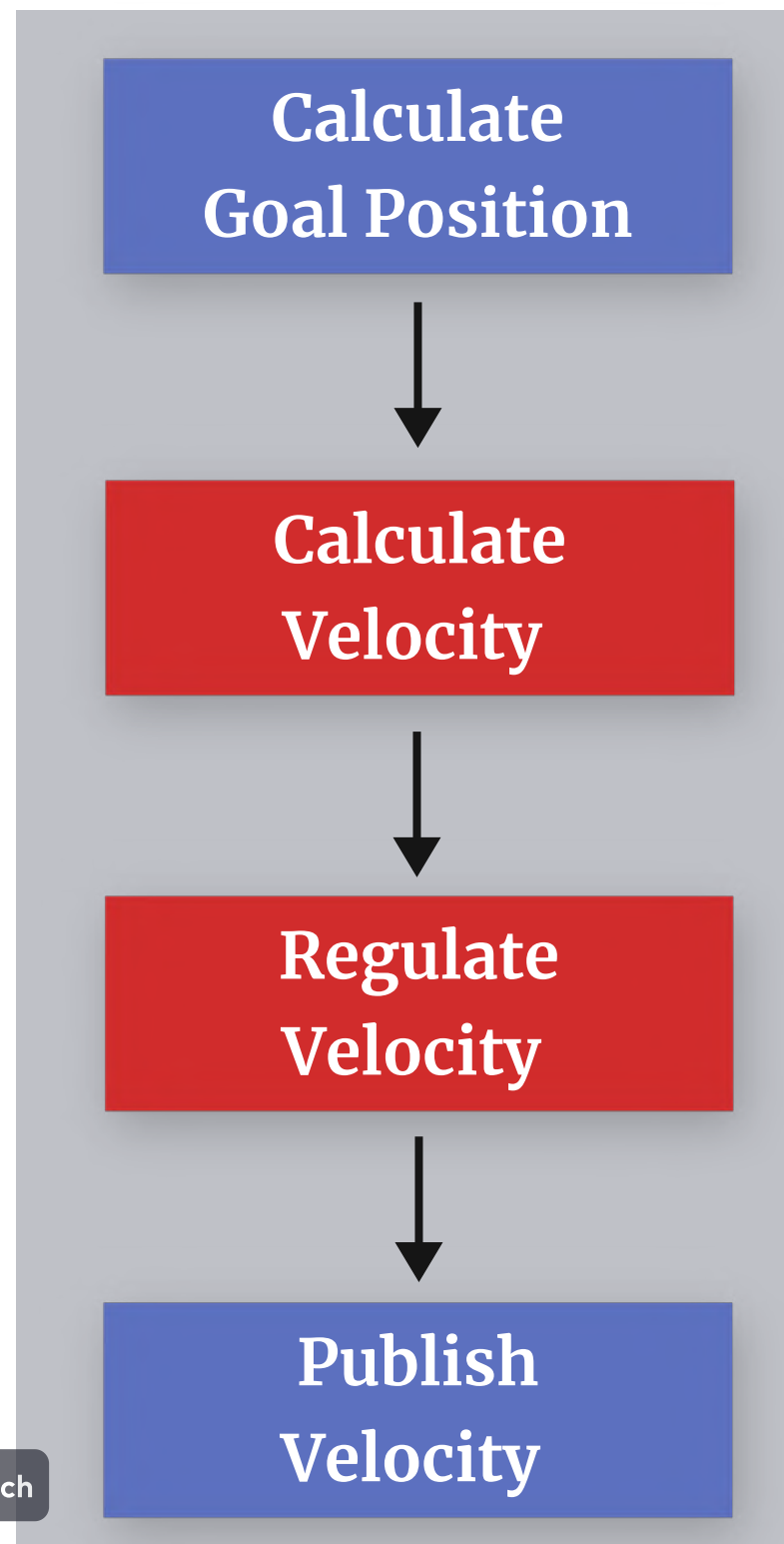
# PID CONTROLLER



- **x\_diff** =  
 $(\text{desired\_subject\_ratio} - \text{subject\_ratio}) * \text{ref\_x\_ratio}$
- **y\_diff** =  
 $\text{lateral\_offset} * \text{ref\_y\_ratio}$
- **goal\_position** =  
 $(\text{x\_diff} + \text{current\_position\_x}, \text{y\_diff} + \text{current\_position\_y})$



# PID CONTROLLER



Try Pitch

Tracking

Holding

Velocity =  
(goal position - current position) \* P gain

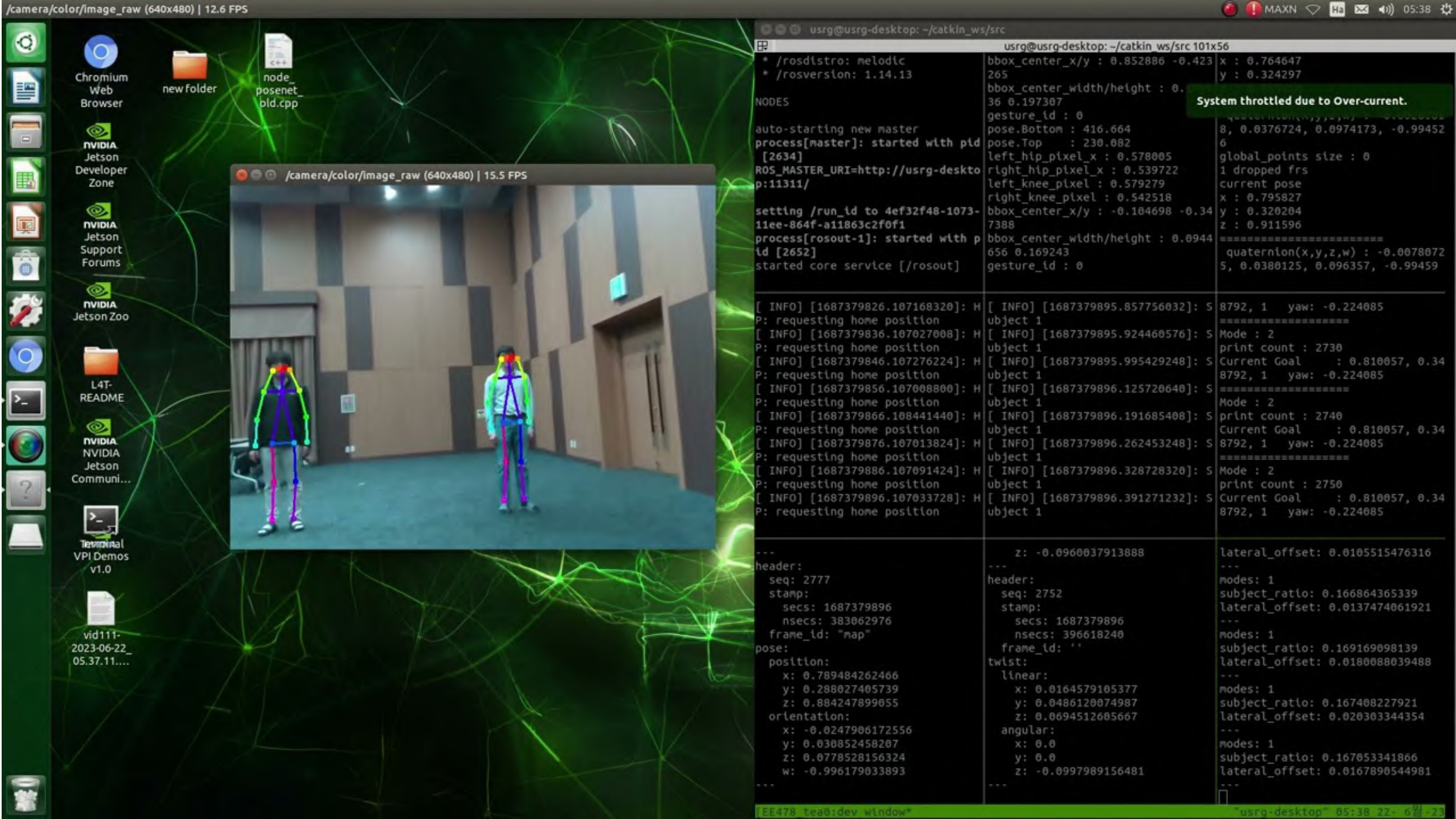
if (abs(velocity) > velocity limit):  
velocity = velocity / abs(velocity)

/mavros/setpoint\_velocity/cmd\_vel

**KAIST**



# MANUAL DEMONSTRATION



<https://youtu.be/NaegwWQG9.3Q>



# RESULT IMAGES

First selfie



Second selfie



Group selfie



---

## TESTFLIGHT ANALYSIS

# Flight #1 and #2

### Successful Items :

- Autonomous takeoff
- Hover
- Detection of person
- Gesture detection
- Yaw change when using gesture 3 or 4

### Unsuccessful Items :

- Ratio *desired\_subject\_ratio* was set to 0.2 the night before after a test flight, as before the drone was coming to close, but we might have been to conservative the drone wasn't moving enough



---

## TESTFLIGHT ANALYSIS

# Flight #3

### Successful Features:

- Autonomous **takeoff**
- **Hover**
- **Detection** of person
- **Yaw control** when using gesture 3 or 4 (**This is our special feature**)
- **Selfies** recorded
- Drone **approached and followed commands**, as we set a better desired\_\_subject\_\_ratio
- Entered mode 5 for group photo, and mode 6 for autonomous landing

### Unsuccessful Features:

- The drone approached **too close** as we have not yet found the right balance desired\_\_subject\_\_ratio
- We did the Autonomous landing gesture too late, and the TA took over before it could execute it.
- Flight planner has to be tuned to make a clearer flight path **more understandable by humans** (it is too slow, and doesn't show clear stopping points, can be unsettling if you don't know the inner workings).
- **Didn't detect** two subjects for the group photo, so it couldn't centre the frame.

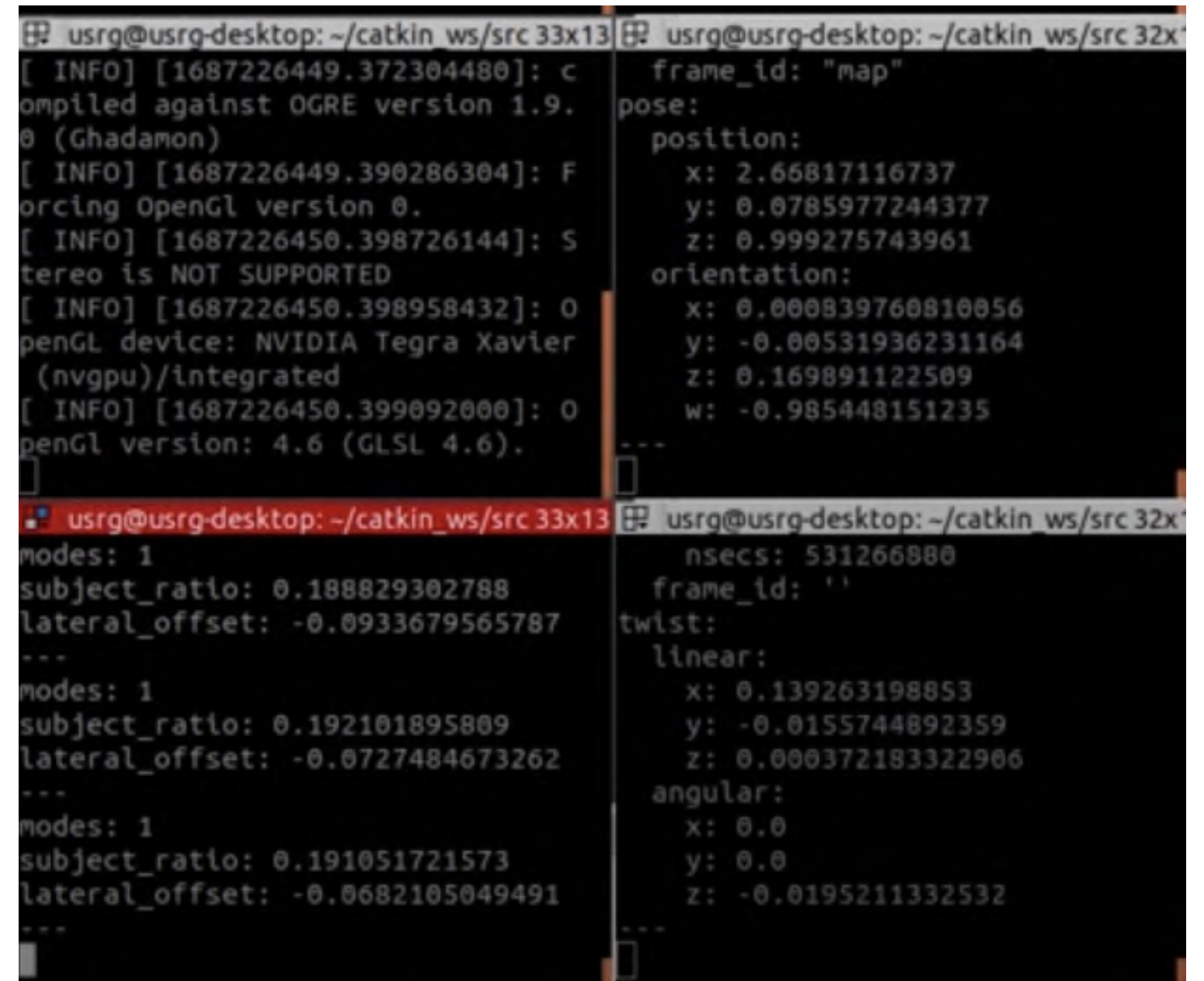
# TESTFLIGHT ANALYSIS

## Flight #3: Rosbag recording

### Key Panel description

- The local position : upper right
- The gesture (mode) the drone received : bottom left
- The linear & angular velocity command msg from PID controller : bottom right

<Panel Images from the video>



The image displays four terminal window screenshots arranged in a 2x2 grid, showing ROS log messages and drone data. The top-left window shows log messages for the 'map' frame, including position and orientation data. The top-right window shows log messages for the 'map' frame, including position and orientation data. The bottom-left window shows log messages for the 'map' frame, including position and orientation data. The bottom-right window shows log messages for the 'map' frame, including position and orientation data.

```
usrg@usrg-desktop: ~/catkin_ws/src 33x13 [ INFO] [1687226449.372304480]: c
ompile against OGRE version 1.9.
0 (Ghadamon)
[ INFO] [1687226449.390286304]: F
orcing OpenGL version 0.
[ INFO] [1687226450.398726144]: S
tereo is NOT SUPPORTED
[ INFO] [1687226450.398958432]: O
penGL device: NVIDIA Tegra Xavier
(nvgpu)/integrated
[ INFO] [1687226450.399092000]: O
penGL version: 4.6 (GLSL 4.6).
]

usrg@usrg-desktop: ~/catkin_ws/src 32x13 frame_id: "map"
pose:
position:
x: 2.66817116737
y: 0.0785977244377
z: 0.999275743961
orientation:
x: 0.000839760810056
y: -0.00531936231164
z: 0.169891122509
w: -0.985448151235
---

usrg@usrg-desktop: ~/catkin_ws/src 33x13 modes: 1
subject_ratio: 0.188829302788
lateral_offset: -0.0933679565787
---
modes: 1
subject_ratio: 0.192101895809
lateral_offset: -0.0727484673262
---
modes: 1
subject_ratio: 0.191051721573
lateral_offset: -0.0682105049491
---

usrg@usrg-desktop: ~/catkin_ws/src 32x13 nsecs: 531266880
frame_id: ''
twist:
linear:
x: 0.139263198853
y: -0.0155744892359
z: 0.000372183322906
angular:
x: 0.0
y: 0.0
z: -0.0195211332532
---
```



# Final Flight Video

The screenshot displays a ROS2 simulation environment. On the left, a sidebar contains various toolbars and panels. The 'Displays' panel shows 'Global Options' (Fixed Frame: map, Background Color: 48; 48; 48, Frame Rate: 30, Default Light: checked) and 'Global Status' (Fixed Frame: No tf data. Actual err...). The 'Grid' panel shows 'Status: Ok' and 'Topic: /mavros/local\_positio...'. The 'Image' panel shows 'Unreliable' and 'Prefer UDP topic transport'. The 'Time' panel shows 'ROS Time: 1687227034.74', 'ROS Elapsed: 582.75', 'Wall Time: 1687227034.77', and 'Wall Elapsed: 582.65'. The main 3D view shows a robot in a simulated environment with green markers. The 2D top-down view shows a grid with a red arrow. The terminal output on the right shows the following:

```
usrg@usrg-desktop: ~/catkin_ws/src
roscore http://usrg-desktop:11311/33x27, usrg@usrg-desktop: ~/catkin_ws/src/kais

started roslaunch server http://u
srg-desktop:42547/
ros_comm version 1.14.13

SUMMARY
=====
PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.13

NODES

auto-starting new master
process[master]: started with pid
[8656]
ROS_MASTER_URI=http://usrg-deskto
p:11311/

setting /run_id to 2df143fe-0f0e-
11ee-9051-918f9496527f
process[rosout-1]: started with p
id [8676]
started core service [/rosout]
[]

usrg@usrg-desktop: ~/catkin_ws/src 33x13 usrg@usrg-desktop: ~/catkin_ws/src 32x
[ INFO] [1687226449.372304480]: c
ompile against OGRE version 1.9.
0 (Ghadamon)
[ INFO] [1687226449.390286304]: F
orcing OpenGL version 0.
[ INFO] [1687226450.398726144]: S
tereo is NOT SUPPORTED
[ INFO] [1687226450.398958432]: O
penGL device: NVIDIA Tegra Xavier
(nvgpu)/integrated
[ INFO] [1687226450.399092000]: O
penGL version: 4.6 (GLSL 4.6).
[]

usrg@usrg-desktop: ~/catkin_ws/src 33x13 usrg@usrg-desktop: ~/catkin_ws/src 32x
modes: 1
subject_ratio: 0.180383056402
lateral_offset: 0.00569858541712
---
modes: 1
subject_ratio: 0.181401878595
lateral_offset: 0.00603179913014
---
modes: 1
subject_ratio: 0.184538722038
lateral_offset: -0.00386028294452
---
```

Try Pitch

<https://www.youtube.com/watch?v=gyPfKRZHUTk>



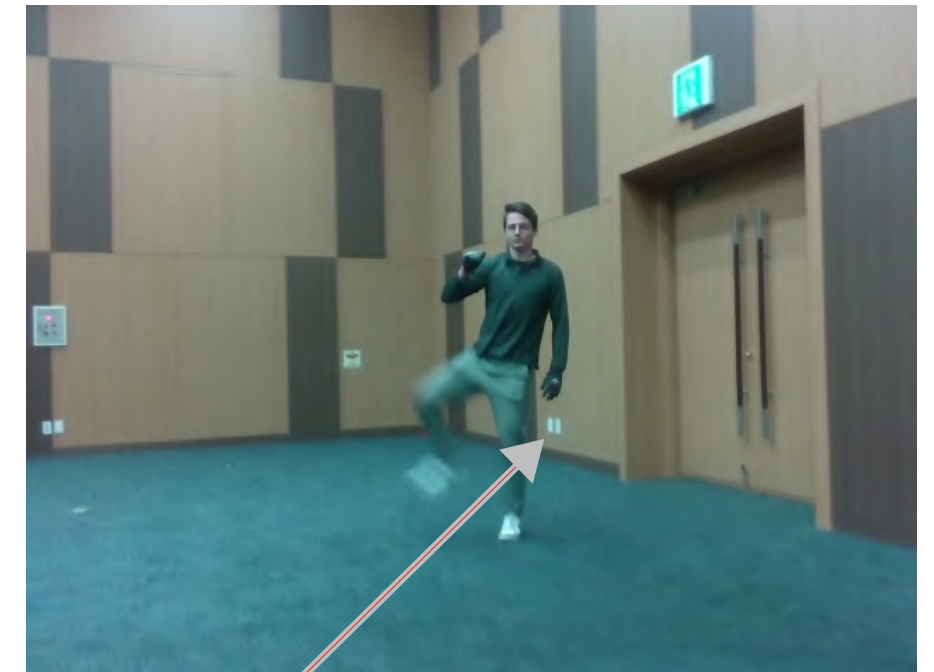
# TESTFLIGHT ANALYSIS

## Flight #3: Rosbag recording

### Flight description

1. The drone successfully **took off** autonomously.
2. When the drone detected the "raise hands gesture (1)", it received "mode 1", and **followed Theo** as you can see the arrow moves in the top view of the RVIZ.
3. The drone **took selfies** whenever the subject made a "selfie pose": "mode: 2".
4. When the drone detected "modes: 3": "turn left", the angular velocity of z became 0.15 and the **drone turned left**. It also worked in "turn right".
5. It detected the mode 5: "Group photo", the drone was supposed to move to the centre point of 2 people, however, **it only detected the right subject**, so it followed him and the other one was out of frame
6. At the end of the video, the drone detected "**land**" gestures, as you can see the "modes" is changed to 6 in the bottom left.

### Selfie image results

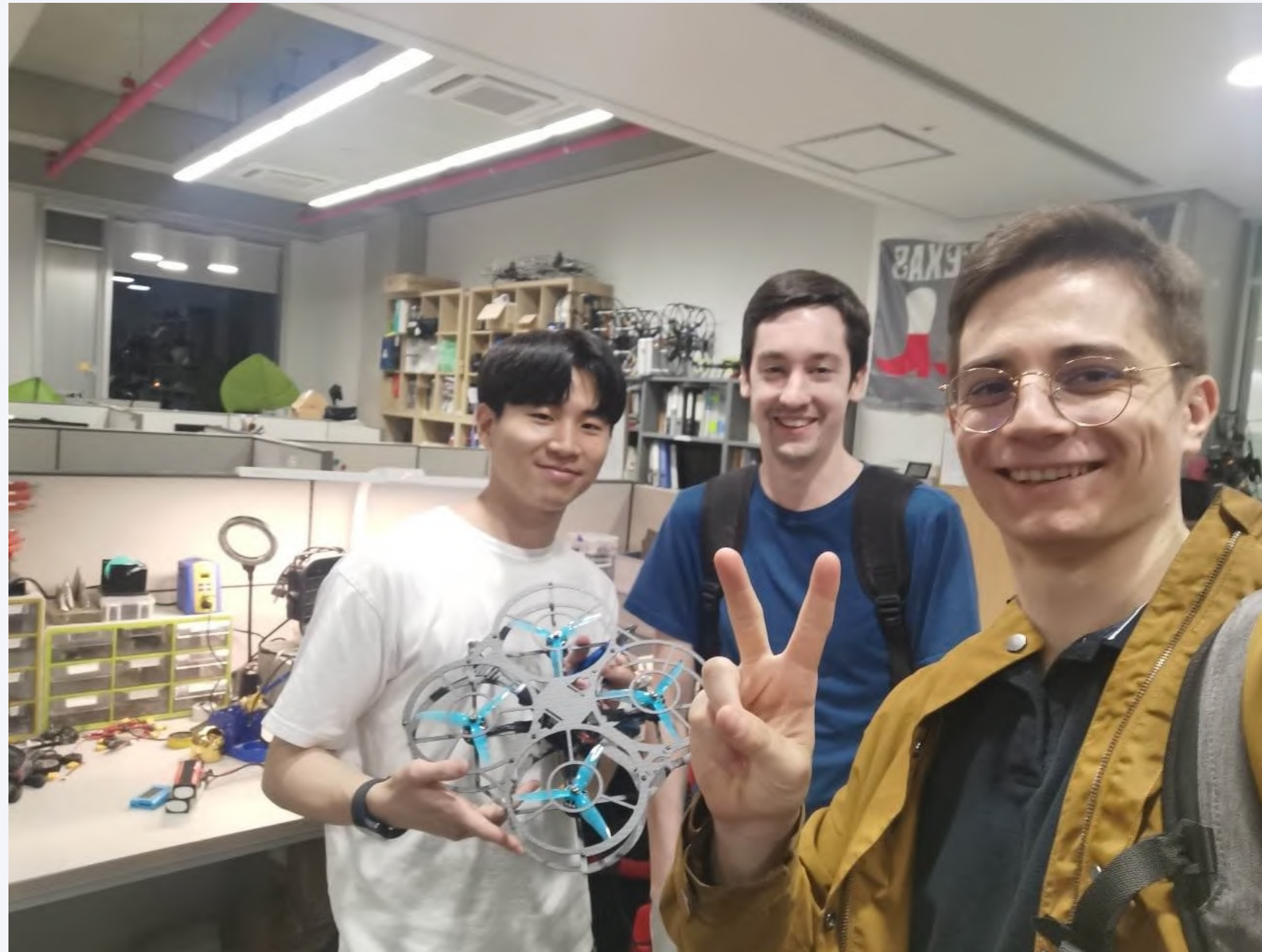


NOTICE THE GLOVES

**KAIST**



# Our team



Jihyeok Kim

Riley Taylor

Theo Michel

Thank you for the adventure

# Thank you for reading

