SIMON FRASER UNIVERSITY

Robustness validation of the trained NN controllers with Crazyflie 2.1

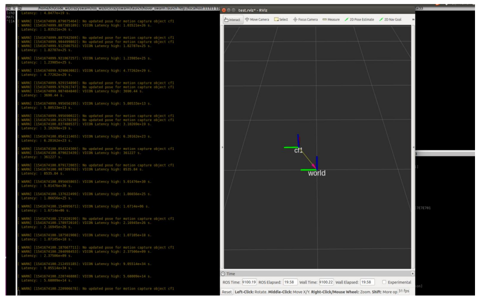
Wenxiang He, Hanjie Liu, Hanyang Hu.

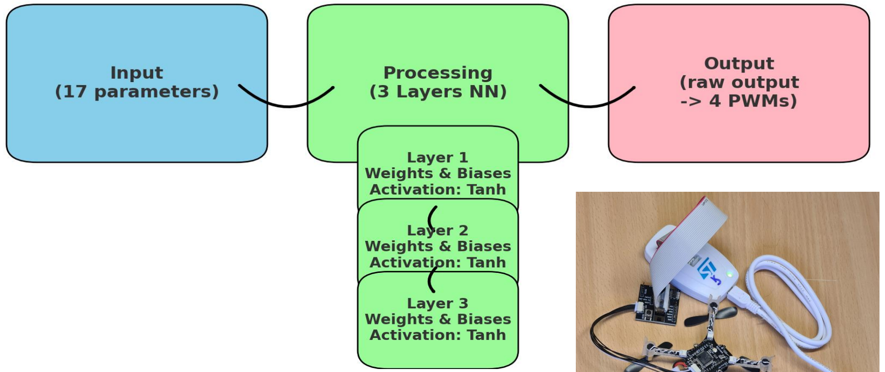
Simon Fraser University, MARS Lab

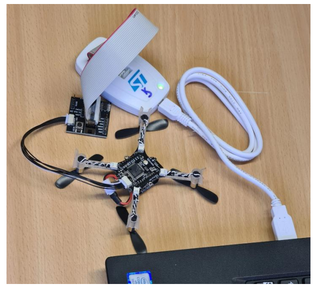
2024-07-31

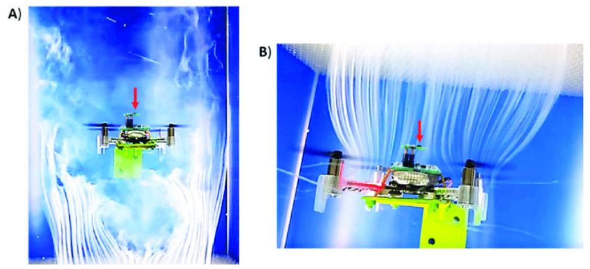
Motivation

- Objective: Verify robustness of the algorithm from Topic "Enhancing Robustness in Reinforcement Learning with Hamilton-Jacobi Reachability Analysis".
- Goal: Implement, integrate and test trained NN controllers on Crazyflie 2.1 to enhance its hover stability under various disturbances.









SIMON FRASER UNIVERSITY

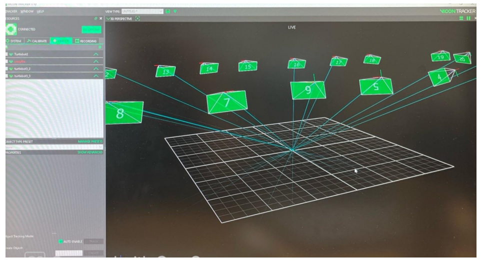
Background:

- Hardware Background: Crazyflie 2.1 Drone
  - a micro quadcopter known for its high programmability and modular design, making it ideal for research in flight control.
  - Its firmware containes a module called 'controller' responsible for flight control. We can integrate custom controllers into this module.

SFU SIMON FRASER UNIVERSITY

Background:

- Software Framework Background:
  - CrazySwarm
  - an open-source software framework built on top of the Crazyflie firmware.
    It allows us to integrate Crazyflie 2.1, the lab's VICON motion capture system,
    and using scripts for the control and testing of drones in real-world flight scenarios.
  - It also allow us to synchronously record the drone's spatial position information


Vicon Tracker 3.10

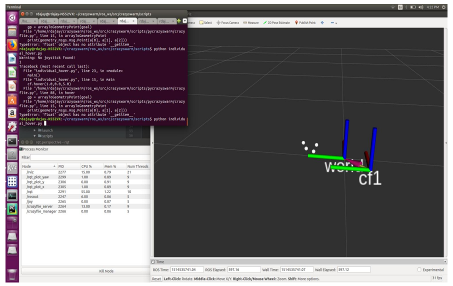
Capture and send drone object data →

Crazyswarm

Use scripts to control →

↓ Use spatial data capture from Vicon and visualize

Crazyflie 2.1

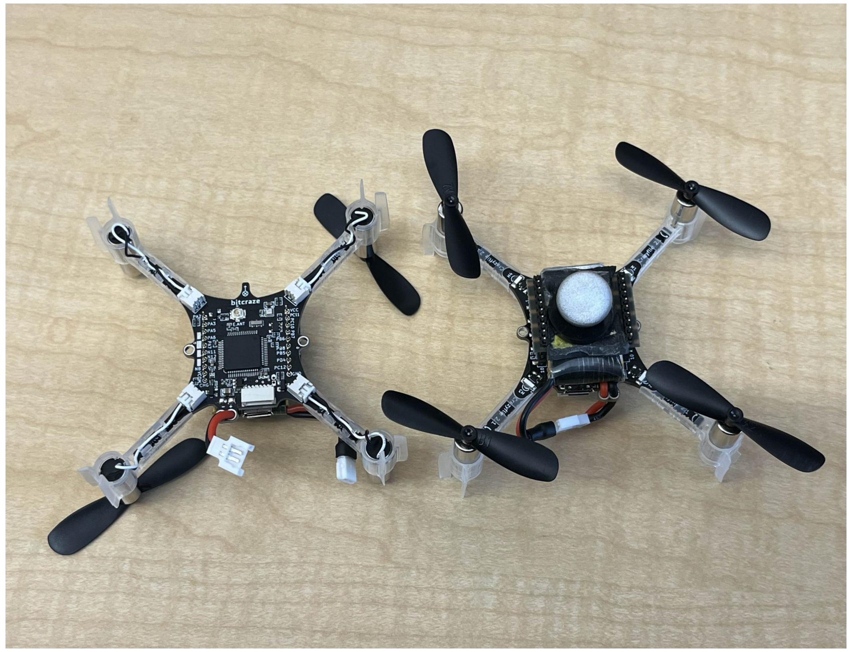


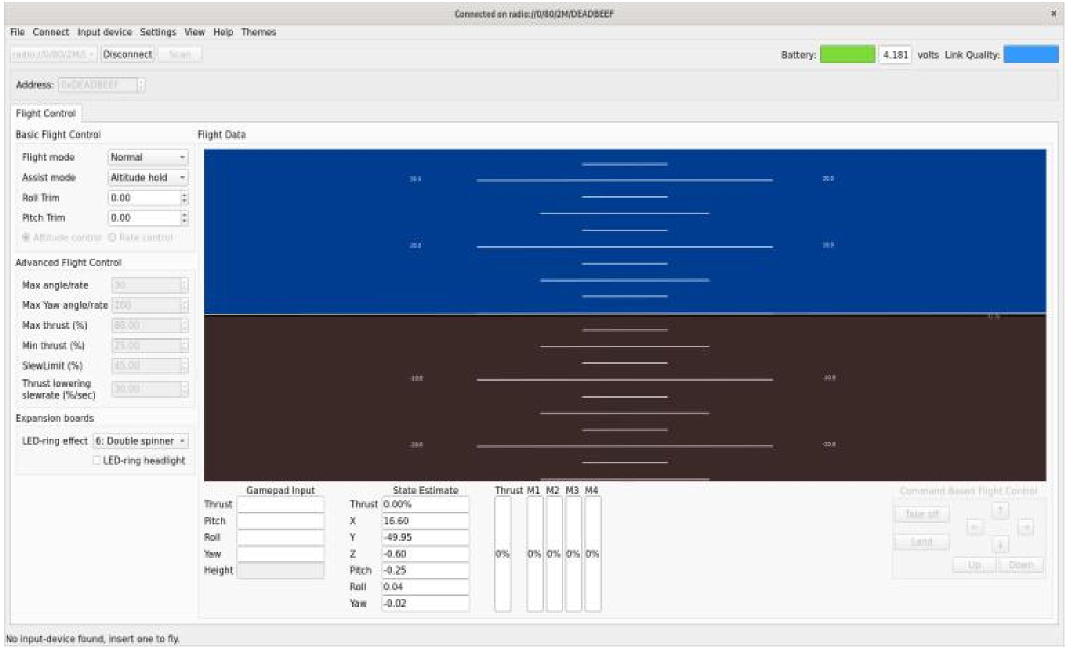
Rviz (3D visualization tool for ROS)

SIMON FRASER UNIVERSITY

Project Procedure

- Hardware Assembly and Software Configuration of Crazyflie 2.1:
  - Assemble the hardware components of the Crazyflie 2.1 drone.
  - Configure the drone using the appropriate software tools (cfclient).





Connected on radio://0/80/2M/DEADBEEF

Battery: 4.181 volts Link Quality:

Basic Flight Control

Flight mode	Normal
Assist mode	Altitude hold
Roll Trim	0.00
Pitch Trim	0.00

Advanced Flight Control

Max angle/rate · Max Yaw angle/rate · Max thrust (%) · Min thrust (%) · SlewLimit (%) · Thrust lowering slewrate (%/sec)

Expansion boards

LED-ring effect: 6: Double spinner

LED-ring headlight

Gamepad Input: Thrust, Pitch, Roll, Yaw, Height

State Estimate

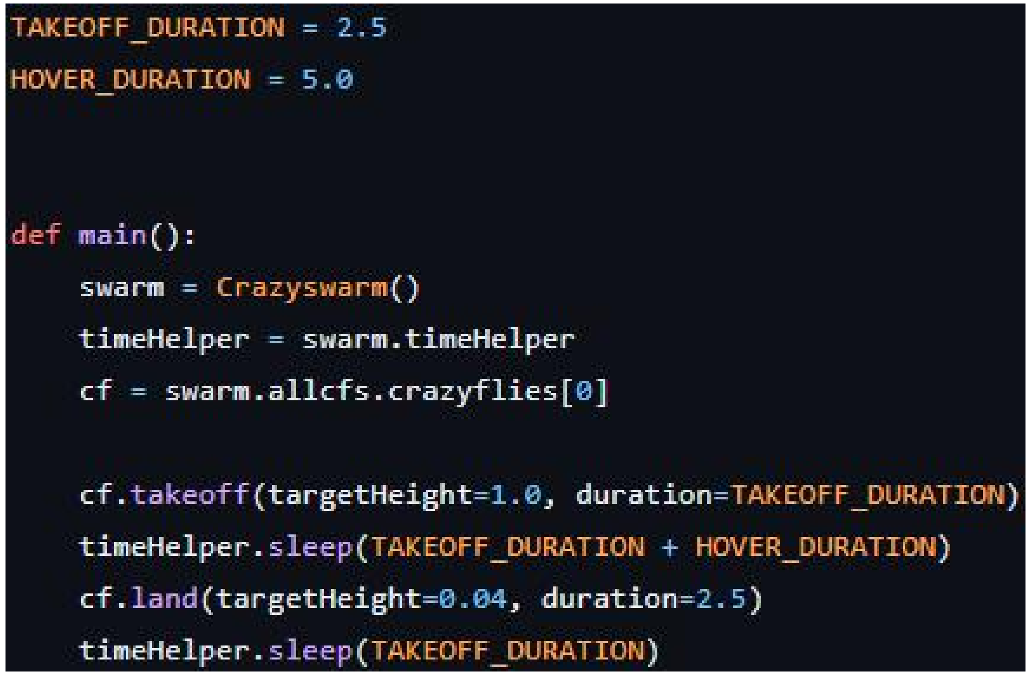
Thrust	0.00%
X	16.60
Y	-49.95
Z	-0.60
Pitch	-0.25
Roll	0.04
Yaw	-0.02

Thrust	M1	M2	M3	M4
0%	0%	0%	0%	0%

No input-device found, insert one to fly.

Project Procedure

- Crazyswarm scripting real-world test flight.
  - With Crazyflie 2.1 and CrazySwarm set up, we can use scripts to control the drone for simple operations such as takeoff, hover, and land.
  - Below is a script that implements a simple operation: taking off to a height of 1 meter within 2.5 seconds, hovering for 5 seconds, and then landing within 2.5 seconds.

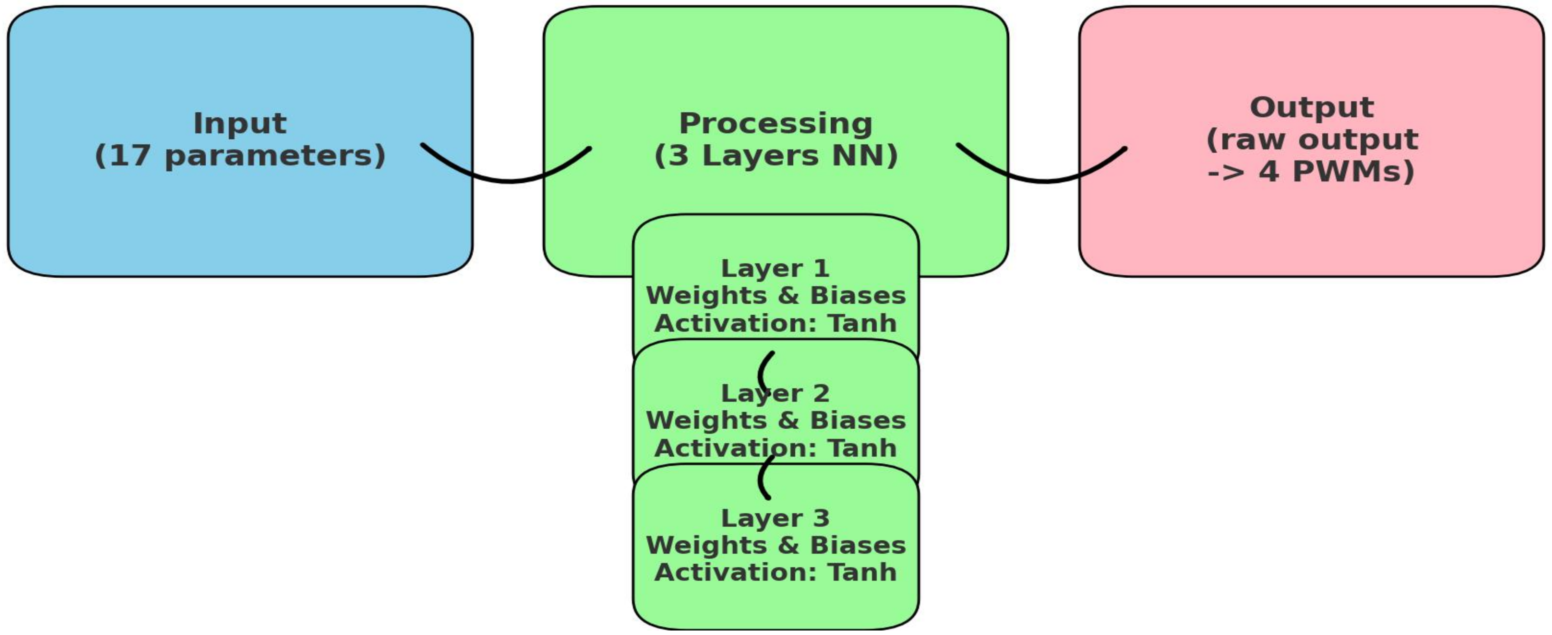


```
TAKEOFF_DURATION = 2.5
HOVER_DURATION = 5.0


def main():
    swarm = Crazyswarm()
    timeHelper = swarm.timeHelper
    cf = swarm.allcfs.crazyflies[0]

    cf.takeoff(targetHeight=1.0, duration=TAKEOFF_DURATION)
    timeHelper.sleep(TAKEOFF_DURATION + HOVER_DURATION)
    cf.land(targetHeight=0.04, duration=2.5)
    timeHelper.sleep(TAKEOFF_DURATION)
```

SIMON FRASER UNIVERSITY

Project Procedure

- Implemention and integration of trained NN controllers
  - We have several neural networks trained in a simulation environment.
  - They have generic model inputs/outputs and processing logic.



Project Procedure

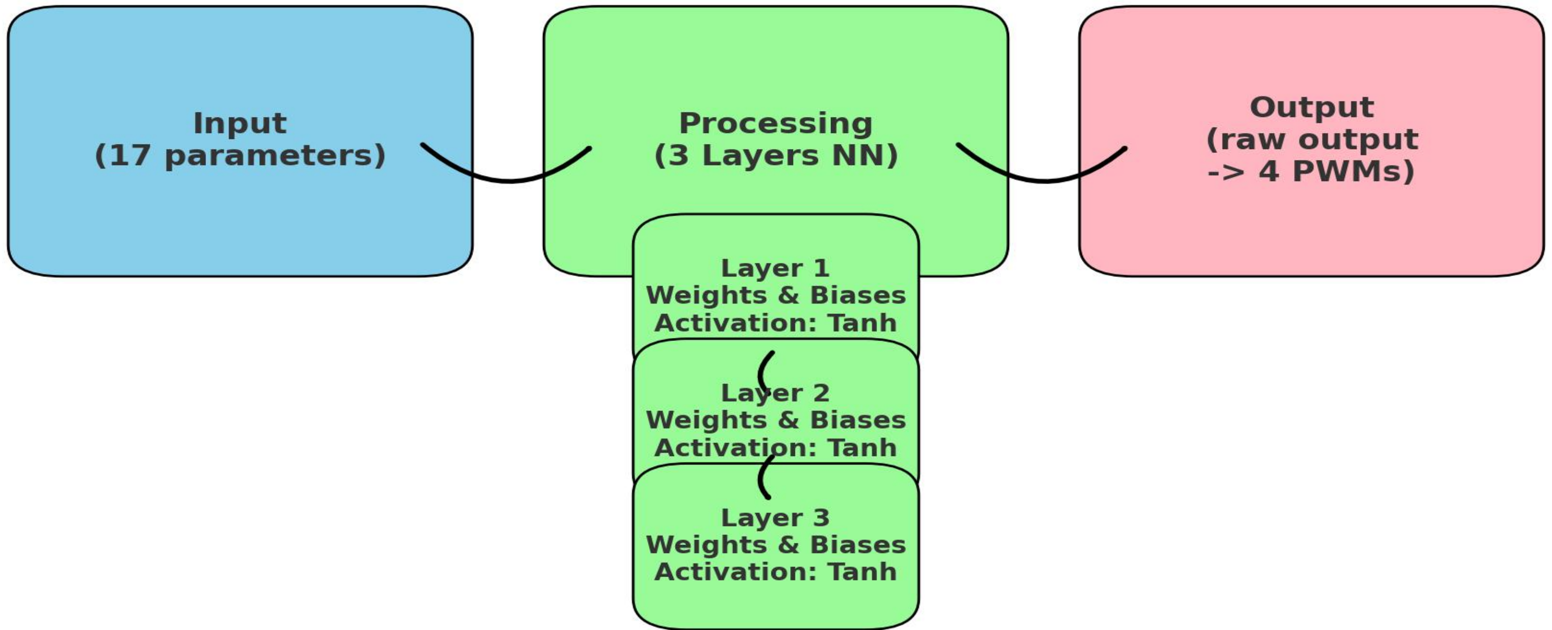
- Implemention and integration of trained NN controllers
  - We have several neural networks trained in a simulation environment.
  - The input values for these neural networks are:
    - Accept 17 input parameters:
      - Positions (3): xyz
      - Quaternions (4): quat
      - Velocities (3): vel
      - Angular velocities (3): ang_v
      - Last output of controller (4): last_action

Project Procedure

- Implemention and integration of trained NN controllers
  - We have several neural networks trained in a simulation environment.
  - The output values for these neural networks are:
    - Generate four output parameters:
      - Action (4): range 0~1
  - These actions are then converted into Pulse Width Modulation (PWM) signals using the following formula:
    - pwm=30000+np.clip(action, -1, +1)×30000
  - These PWM signals are used to control the four motors of the drone.

Project Procedure

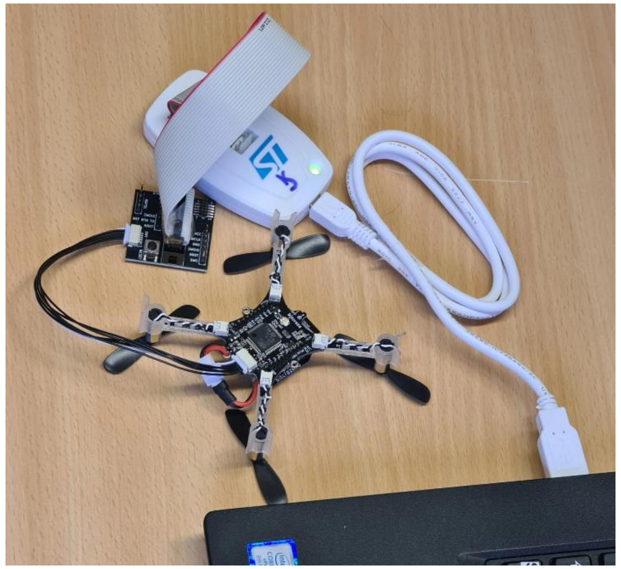
- Implemention and integration of trained NN controllers
  - The Crazyflie firmware is written in C language, we need to translate our **neural network processing logic** and **parameters** into C language.
  - This involves writing a C language NN controller to accept inputs, simulate the neural network processing, and generate outputs.

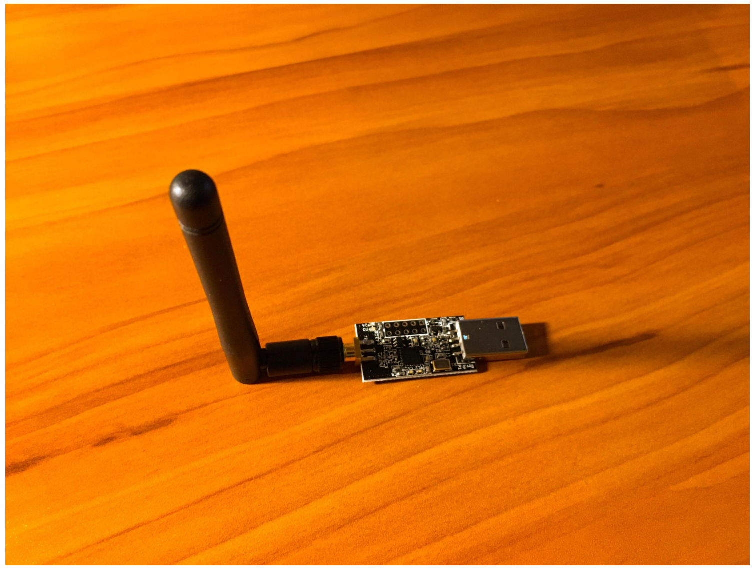


SIMON FRASER UNIVERSITY

Project Procedure

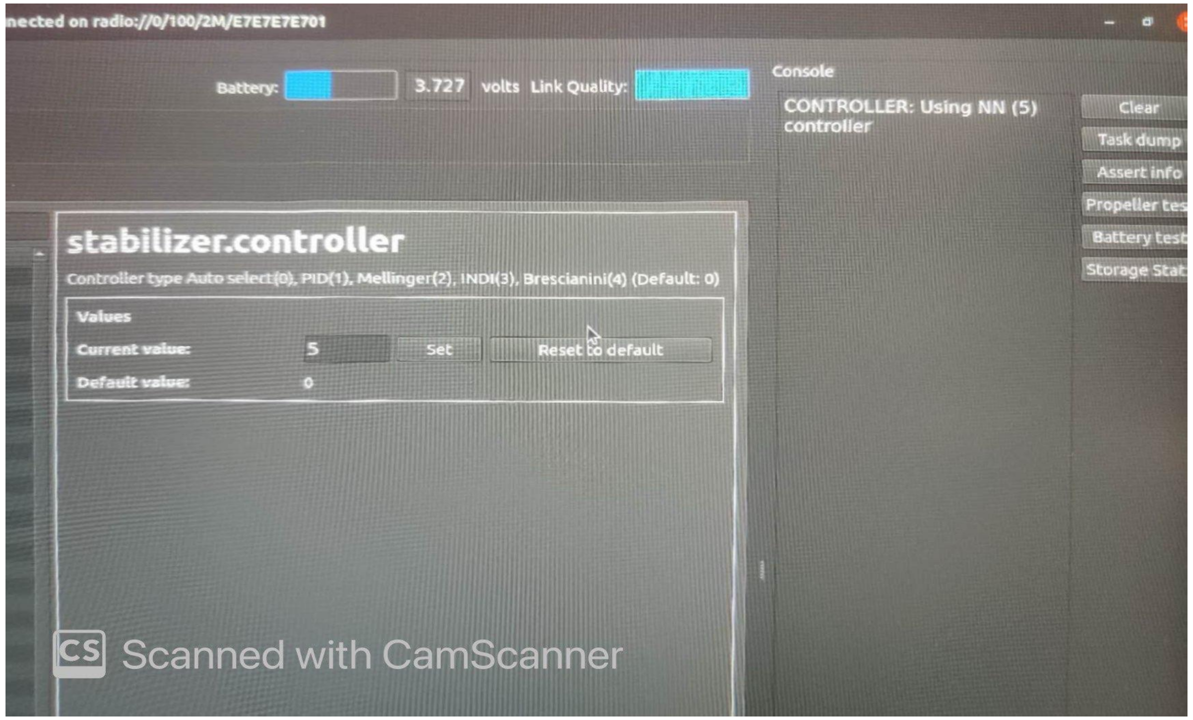
- Implemention and integration of trained NN controllers
  - Build and Flash Crazyflie Firmware with trained NN controllers.
  - After implementing a Trained NN controller, we build it into the crazyflie-firmware and flash the compiled firmware onto the Crazyflie 2.1 drone:
  - Here we use 2 tools for flashing: ST Link V2 & Crazyradio





Project Procedure

- Implemention and integration of trained NN controllers
  - Build and Flash Crazyflie Firmware with trained NN controllers.
  - Verify the trained NN controllers is integrated into Crazyflie 2.1.



nected on radio://0/100/2M/E7E7E7E701

Battery: 3.727 volts Link Quality:

Console

CONTROLLER: Using NN (5) controller

stabilizer.controller

Controller type Auto select(0), PID(1), Mellinger(2), INDI(3), Brescianini(4) (Default: 0)

Values

Current value: 5

Default value: 0

Project Procedure

- Verification and testing
  - Verification
    - We use debug tool to make sure that the Crazyflie-Firmware can compile successfully.
    - ST Link V2
  - Testing
    - Use CrazySwarm to switch controllers and perform flight tests through scripting to verify the stability.

- Successfully flashed a test trained NN controller into the firmware and conducted test flights to check its stability.
- We are currently working on completing our controllers and further validate their performance.

SFU
SIMON FRASER UNIVERSITY

Challenge Faced

- Initial Drone Assembly Issues
  - Drift due to balance problems; hardware-related.
  - Replaced propellers and motors to resolve the issue.
- VICON System Limitations
  - Tracker 3.10 requires three markers; space constraints.
  - Bypassed Tracker 3.10, configured in CrazySwarm.
- Library Version Discrepancies
  - Issues with Crazyflie and CrazySwarm libraries.
  - Debugged and corrected script syntax.

SIMON FRASER UNIVERSITY

What’s left

- Complete trained NN controllers Integration
- Returning Experimental Data
- Testing and Validation
- Documentation and Reporting

SFU
SIMON FRASER UNIVERSITY

MARS LAB

Thank you for listening !