Autonomous UAV for Solar-Induced Fluorescence Monitoring: T18 Octorotor Specification and Simulation Development

University of Toronto Institute for Aerospace Studies
Flight Systems and Control Lab

Joseph Halliday
Student Number: 1007316548

Supervisor: Professor Hugh H.T. Liu
August 25, 2025

Abstract

Solar-Induced Fluorescence (SIF) monitoring represents a critical advancement in agriculture and ecosystem assessment, providing direct insights into photosynthetic activity and plant stress responses. Current SIF measurement approaches face significant limitations: satellite-based methods provide broad coverage but lack spatial resolution, while ground-based methods offer high precision but limited scalability. This research addresses these challenges by developing specifications and a simulation environment for an autonomous Tarot T18 octorotor for SIF monitoring applications.

The research encompasses two primary components: (1) detailed specification development for the T18 octorotor, and (2) creation of a simulation environment using PX4, Gazebo, and ROS2 frameworks to validate autonomous flight operations in forest environments.

The T18 octorotor demonstrates superior performance characteristics with a total mass of 13.74 kg (excluding payload) and achieves maximum endurance of 19.8 minutes with a dual battery configuration ($2 \times 10{,}000$ mAh) or 39.6 minutes with a quadruple battery setup. The maximum range extends to 7.78 km and 15.57 km, respectively. Performance analysis utilized actuator disc theory and empirical battery modelling.

Simulation development successfully integrated PX4 v1.15.4 autopilot, Gazebo Harmonic physics engine, and ROS2 Humble framework to create a testing environment for the T18 octorotor. Key achievements include the development of a custom forest world with realistic pine trees distributed, integration of 3D CAD models with accurate dimensions and inertial properties derived from SolidWorks, creation of a custom PX4 airframe configuration file and Gazebo model files for octorotor control, and implementation of an autonomous waypoint following algorithm through both QGroundControl mission planning and ROS2 node architecture.

Flight testing in the simulated forest environment validated autonomous navigation and control capabilities, demonstrating successful takeoff, waypoint following, and landing sequences. However, the system exhibited undesired yaw rotation and stability issues during navigation between widely spaced waypoints, indicating the need for T18-specific control algorithm development. The integrated simulation framework provides a robust foundation for future SIF sensor integration, advanced autonomous monitoring algorithms, and hardware-in-the-loop testing, contributing to the broader SIF monitoring project objectives of transforming environmental monitoring through autonomous aerial systems.

Contents

List of Figures

List of Tables

1 Introduction

With increasing global attention to climate change and sustainability, forest data collection that measures ecosystem health, photosynthetic activity, and carbon uptake has become increasingly critical [1]. Solar-induced fluorescence (SIF), a faint glow emitted by plants, has emerged as an indicator that provides direct, real-time insights into photosynthetic activity, plant stress, and gross primary production (GPP) [1] [2]. However, current SIF measurement methodologies face a critical gap between satellite-based and ground-based measurements [1]. Satellite platforms such as NASA's Orbiting Carbon Observatory-2 (OCO-2) and the upcoming European Space Agency's FLuorescence EXplorer (FLEX) mission provide global coverage, but lack spatial resolution [1]. In contrast, ground-based SIF measurements offer high precision but are severely limited in spatial coverage, requiring complex equipment and expert operators, which restricts their practical applications in agriculture [1].

The integration of unmanned aerial vehicles (UAVs) with SIF sensors presents an optimal solution to bridge this gap by offering high spatial resolution and sensitivity while covering a wide range of forests [1]. Autonomous drones eliminate the need for skilled drone pilots, ensuring robust and repeatable measurements across diverse environments [1]. Meanwhile, the machine learning algorithm for the SIF sensor enables real-time measurement of plant stress levels [1]. The goal is to provide vital ground validation of satellite-based measurements, support forest health monitoring, and create a foundation for initiatives that link plant physiology and autonomous drones.

This research project addresses these limitations through the simulation development of an autonomous T18 octorotor for SIF monitoring applications. The work represents a critical initiation of the broader SIF monitoring project, a collaborative initiative between the Flight Systems and Control Lab and the Plant Physiology Lab, aimed at revolutionizing autonomous plant health monitoring.

2 Research Objectives

The primary objectives of this research are:

1. **T18 Octorotor Specification Development**: Create a specification sheet for the T18 octorotor, identifying components, dimensions, and performance capability.

2. **Simulation Environment Development**: Develop a simulation framework for the T18 octorotor using PX4, Gazebo, and ROS2 to validate autonomous flight operations and provide a testing platform for future algorithm development.

3 Background

3.1 Solar-Induced Fluorescence

Plants absorb solar radiation primarily to drive photosynthesis, with approximately 80% of the absorbed energy used for photosynthesis [3]. The remaining 20% is dissipated through radiation or other processes [3]. A small portion of this dissipated energy, approximately 0.5–2.0%, is in the form of infrared radiation known as solar-induced fluorescence (SIF) [3].

Gross Primary Production (GPP) is defined as the total amount of carbon captured by plants and is a key indicator of ecosystem health and climate change [3]. However, GPP cannot be measured directly using satellite-based techniques, since carbon uptake through photosynthesis occurs simultaneously with carbon release from plant respiration [3]. Ground-based methods such as eddy covariance can provide GPP measurements, but they lack broad spatial coverage and are impractical for large-scale monitoring [3].

These limitations highlight the value of SIF. Because SIF is emitted as a byproduct of photosynthesis, it can serve as a proxy for GPP while also functioning as an early indicator of plant stress [3]. By monitoring SIF, it can also help identify drought and heatwave risks before visible symptoms appear [3].

3.2 Current SIF Measurement Limitations

Current SIF measurement techniques face technological gaps that limit their practical application:

- **Satellite-Based Methods**: While missions such as NASA's OCO-2 and ESA's FLEX provide global coverage, they suffer from coarse spatial resolution that cannot resolve field-scale variations [1].

- **Ground-Based Methods**: Tower-mounted and handheld SIF sensors offer high precision and are used to calibrate and validate airborne, UAV, and spaceborne sensors since minimal atmospheric disturbances are captured by the ground-based sensor due to the short optical path from the target [4]. However, they are severely constrained by limited spatial coverage, requiring extensive human resources for deployment across large areas. Additionally, the ground-based methods require complex equipment and expert operators, limiting their scalability [1].

- **Current UAV Implementations**: Existing drone-based SIF systems require skilled human pilots and lack standardized measurement protocols, creating barriers to widespread adoption and repeatable data collection [1].

4 T18 Octorotor Specification Development

This section details the specification development for the Tarot T18 octorotor, including components, dimensions, and performance.

4.1 Overview of Tarot T18 Octorotor

The Tarot T18 octorotor was selected due to its exceptional payload capacity (up to 8kg), extended flight endurance capabilities, and compatibility with standard autopilot systems.

Figure 1: Tarot T18 Octorotor Customized by FSC

4.2 Components and Configuration

4.2.1 T18 Base Frame

The Tarot T18 octorotor frame is made from carbon fibre with aluminum parts, making it lightweight and durable [5]. The arms and the landing gear fold for easier transportation [5].

Table 1: T18 Base Frame Specifications

Parameter	Value
Axle-to-axle diameter	1,280 mm
Kit diameter	1,355 mm
Height	610 mm

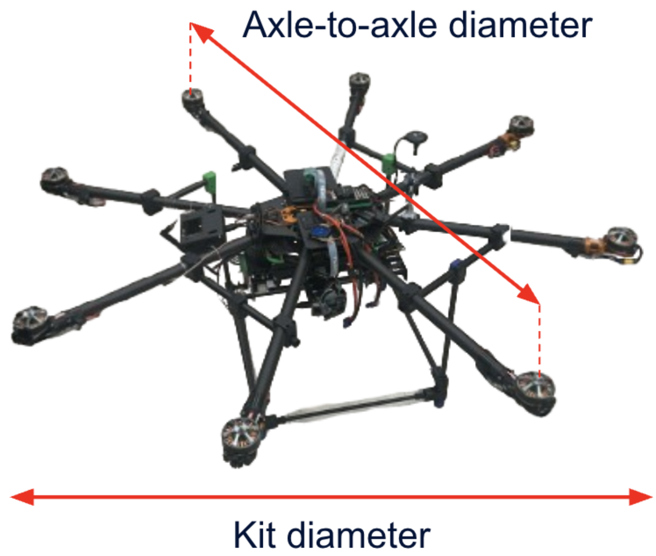

Figure 2: Tarot T18 Octorotor Dimensions

4.2.2 Flight Control and Navigation Systems

The flight controller used is the Pixhawk 6X autopilot, enabling any autonomous vehicle control with extensive sensor integration [6]. Fast and accurate positioning of the drone is accomplished by the Holybro M10 GPS module.

Table 2: Flight Control and Navigation System Specifications

Component	Model	Key Specifications
Flight Controller	Pixhawk 6X	
GPS	Holybro M10	2.0m CEP accuracy

4.2.3 Propulsion System

The T18 octorotor utilizes eight propellers and motors configured in the standard octorotor x configuration. Each motor is paired with T-Motor 18×6.1 carbon fibre propellers optimized for efficient thrust generation.

Table 3: Propulsion System Specifications

Component	Specification	Dimensions	Unit Weight	Total Weight
Motors	TMOTOR MN5212 KV340	ϕ59×33.5mm	249g	1,992g
Propellers	T-Motor 18×6.1 Carbon Fibre	ϕ457mm	30g	240g

4.2.4 Power System

The T18 octorotor can be equipped with either 2 or 4 batteries. The endurance and range were calculated for both configurations. The weight used in the simulation is for 2 batteries.

Table 4: Battery System Specifications

Parameter	Value
Battery	Tattu G-Tech LiPo Battery
Capacity	10,000 mAh
Discharge Rate	30C
Voltage	22.2V
Weight per battery	1,365 g

4.3 Mass Properties

The total mass excludes the payload and uses 2 batteries to establish the baseline mass properties. Payload is excluded in the simulation as well and can be updated once the payload is determined. The moment of inertia values were determined from the 3D CAD model built in SolidWorks.

Table 5: T18 Octorotor Mass

Component	Value
Base Frame	8,778g
Motors (8 × 249 g)	1,992g
Propellers (8 × 30 g)	240g
Batteries (2 × 1,365 g)	2,730g
Total (excluding payload)	**13,740g**
I_{xx}	1.045 $kg\,m^2$
I_{yy}	1.608 $kg\,m^2$
I_{zz}	0.965 $kg\,m^2$

4.4 Performance Analysis and Calculations

4.4.1 Known Values and Assumptions

Known Values:

- **Total Mass of UAV:** 13.74 kg (Assuming 0 kg payload)
- **Motor Efficiency:** 75 % (Assumed)
- **Propeller Diameter (D):** 0.457 m (18")

Assumptions:

- Actuator Disc (Momentum) Theory applies (axial flow)
- Uniform power distribution across eight motors
- Density of air is constant $\rho = 1.225 kg/m^3$
- All rotors are identical
- Used empirical factors to adjust power consumption
- Assumed effective battery capacity at optimal endurance to be $k_e = 0.7914$ and at maximum range to be $k_r = 0.6371$

4.4.2 Thrust Requirement and Disc Area Calculation

The fundamental thrust requirement equals the total system weight:

$$T_{total} = W_{total} = m_{total} \times g = 13.74 \text{ kg} \times 9.81 \text{ m/s}^2 = 134.79 \text{ N} \tag{1}$$

Individual motor thrust requirement:

$$T_{motor} = \frac{T_{total}}{8} = 16.85 \text{ N} \tag{2}$$

The rotor disc area for each propeller:

$$A_k = \pi \times \left(\frac{D}{2}\right)^2 = \pi \times \left(\frac{0.457}{2}\right)^2 = 0.164 \text{ m}^2 \tag{3}$$

4.4.3 Induced Velocity

Using momentum theory, the induced velocity during hover is:

$$v_h = \sqrt{\frac{W}{2\rho N A_k}} = \sqrt{\frac{134.79}{2 \times 1.225 \times 8 \times 0.164}} = 6.48 \text{ m/s} \tag{4}$$

4.4.4 Power Analysis

The induced power requirement during hover:

$$P_{ind} = W \times v_h = 134.79 \times 6.48 = 873.1 \text{ W} \tag{5}$$

Since there is no useful power when the UAV is hovering:

$$P_{tot_h} = P_{ind} = 873.1\text{W} \tag{6}$$

4.4.5 Flight Endurance Calculations

Power consumption was adjusted for different flight profiles using empirical factors:

Optimal Endurance: $P_{endurance} = P_{hover} \times 0.914 = 798$ W

Optimal Range: $P_{range} = P_{hover} \times 1.092 = 953$ W

Incorporating motor efficiency (assumed 75%):

Electric Power Demand at Optimal Endurance: $P_{mot,e} = \frac{P_{endurance}}{0.75} = 1064$ W

Electric Power Demand at Optimal Range: $P_{mot,r} = \frac{P_{range}}{0.75} = 1271$ W

4.4.6 Battery Performance Analysis

Known Values:

- **Number of Cells:** $6 \times 2 = 12$
- **Battery Capacity:** 10,000mAh = 10 Ah
- **Nominal Voltage:** 3.7(V/cell) (assumed)

Effective Capacity - Endurance: $C_{eff,e} = 0.7914 \times 10 \times 12 \times 3.7 = 351.4$ Wh
Effective Capacity - Range: $C_{eff,r} = 0.6371 \times 10 \times 12 \times 3.7 = 282.9$ Wh

4.4.7 Endurance and Range:

Endurance: $t_e = \frac{C_{eff,e} \times 3600(sec/h)}{P_{mot,e}} = 1189(s) = 19.8(min)$
Range: $t_r = \frac{C_{eff,r} \times 3600(sec/h)}{P_{mot,r}} = 801(s) = 13.4(min)$

Flight speed using empirical values:

$$v_e \approx 1.1 \times v_h = 7.12\, m/s$$

$$v_r \approx 1.5 \times v_h = 9.71\, m/s$$

Maximum range:

$$R = t_r \times v_r = 7783\,(m) = 7.78(km)$$

4.5 Performance Results

Following the same steps for 4 batteries, the performance analysis yielded the following operational capabilities:

Table 6: T18 Octorotor Performance Summary

Performance	Dual Battery (2×10Ah)	Quad Battery (4×10Ah)
Hover Power	873.1 W	873.1 W
Maximum Endurance	19.8 min	39.6 min
Max Range Flight Time	13.4 min	26.7 min
Maximum Range	7.78 km	15.57 km

The quadruple battery configuration provides substantial endurance improvements essential for extended SIF monitoring missions, achieving nearly 40 minutes of flight time.

5 Simulation Development

This section describes the simulation environment development process, creating a realistic testing platform for autonomous SIF monitoring missions.

5.1 Development Framework

The simulation development utilized multiple open-source frameworks.

- **Ubuntu 22.04**: Linux operating system used as the primary development environment
- **PX4 v1.15.4**: Open-source flight control stack for UAVs with advanced autonomy features
- **Gazebo Harmonic**: 3D physics simulator providing realistic environments for testing and validation
- **ROS2 Humble**: Framework for writing robot software, enabling distributed communication and modular system design
- **Micro XRCE-DDS Agent**: Lightweight DDS bridge enabling communication between ROS2 and PX4
- **QGroundControl**: Mission planning for simulation and actual drones.

This integration provides Software-in-the-Loop (SITL) simulation capabilities essential for validating autonomous flight algorithms without hardware dependencies.

5.2 Custom Gazebo World Development

The first step of simulation development was to create a custom Gazebo world that resembles a forest suitable for simulating SIF monitoring.

5.2.1 Initial Setup

The first step involved creating the base world file from the existing PX4 template and downloading the pine tree model:

1. Navigated to the PX4 Gazebo worlds directory:

```
cd ~/PX4-Autopilot/Tools/simulation/gz/worlds/
```

2. Copied the default world file and renamed it as customforest:

```
cp default.sdf customforest.sdf
```

3. Downloaded pine tree model from Gazebo Fuel Pine Tree and saved the folder in: `~/PX4-Autopilot/Tools/simulation/gz/models/`

5.2.2 World Configuration

The base world was modified to resemble the forest environment. This section covers the key changes made:

1. Opened `customforest.sdf`

2. Updated the world name in the sdf file to match the file name:

```
<world name="customforest">
```

3. Added sky and shadows to suit the outdoor environment:

```
<sky>
    <clouds>true</clouds>
</sky>
<shadows>true</shadows>
```

4. Modified ground surface colour to simulate a dirt environment:

```
<material>
    <ambient>0.2 0.1 0.05 1</ambient>
    <diffuse>0.4 0.3 0.2 1</diffuse>
    <specular>0.05 0.03 0.02 1</specular>
</material>
```

5. Added multiple pine trees at random coordinates throughout the forest environment:

```
<include>
  <uri>model://Pine Tree</uri>
  <name>pine_tree_1</name>
  <pose>2.50 0.90 0 0 0 0</pose>
</include>
```

Each tree instance requires a unique identifier to prevent naming conflicts within the simulation environment, hence the sequential naming convention `pine_tree_1`, pine_tree_2, etc. was used. The pose parameter specifies the tree's position and orientation in 6DOF format: `x y z roll pitch yaw`.

5.2.3 World File Integration

The custom SDF file created required integration into the PX4 build system:

1. Ensure the custom Gazebo world file (`customworld1.sdf`) is saved in the following directory:

```
~/PX4-Autopilot/Tools/simulation/gz/worlds/
```

2. Navigate to the following `CMakeLists.txt` and add `customworld1` under `set(gz_worlds)`:

```
cd ~/PX4-Autopilot/src/modules/simulation/gz_bridge
gedit CMakeLists.txt
```

```
set(gz_worlds
    default
    windy
    baylands
    lawn
    customforest
)
```

5.2.4 Custom Forest World in Gazebo

The following figure shows the final custom forest world in Gazebo:

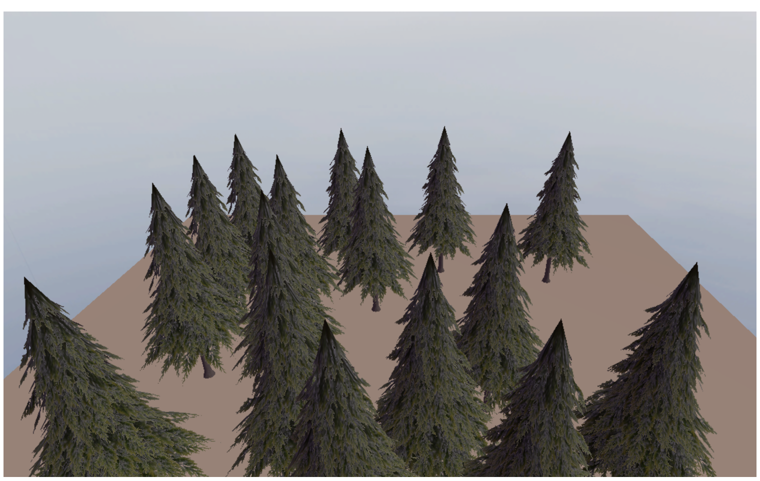

Figure 3: Custom Forest World in Gazebo

5.3 T18 CAD Model Development

A 3D model of the T18 octorotor is required to visualize the drone in simulation. The following CAD components must be created separately to ensure accurate simulation:

- Base frame
- Propellers (both clockwise and counterclockwise)
- Motors
- Motor mounts

For the motor and motor mount, the CAD files from x500, an existing PX4 quadrotor model, were used.

To create the 3D model, the following steps were performed:

1. **SolidWorks modelling**: Generated STL files for the base frame and propellers with accurate geometric representation.
2. **Blender processing**: Imported the STL files, scaled them down by a factor of 1000 (SolidWorks uses millimetres while Gazebo uses metres), applied textures, and exported the base frame as DAE and the propellers as STL.

Figure 4: CAD Model of the Base Frame with Carbon Fibre Texture Applied

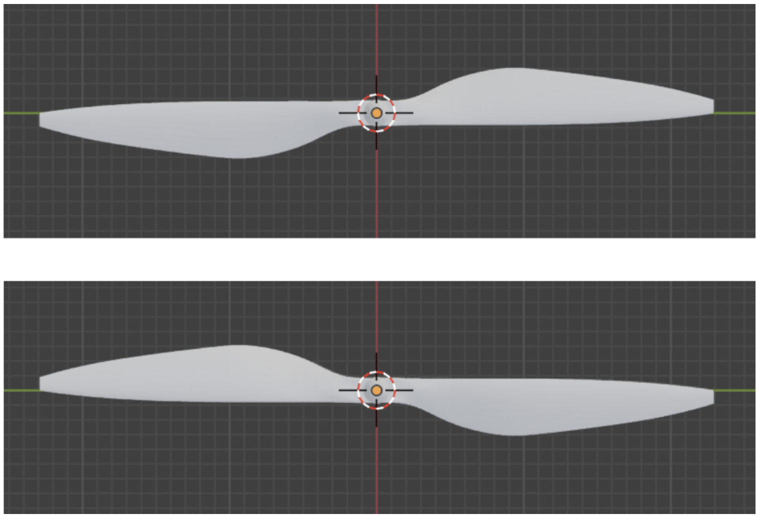

Figure 5: CAD Model of 18x6.1 Propeller (Top: CW and Bottom: CCW)

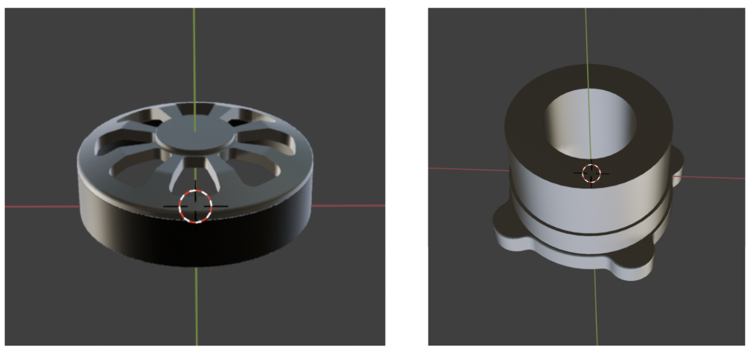

Figure 6: CAD Model of the Motor (Left) and the Motor Mount (Right)

5.4 PX4 Airframe Configuration File Development

5.4.1 Airframe Configuration

The PX4 airframe configuration file specifies the vehicle type and associated parameters, enabling PX4 to correctly interpret and control the UAV. Critical configuration elements include:

- Vehicle type specified as octorotor x

```
# @type Octorotor x
```

- Number of rotors set to 8 for octorotor

```
param set-default CA_ROTOR_COUNT 8
```

- The rotor positions following the configuration and coordinates of the PX4 octorotor x airframe reference. Note that in the figure, the rotor count starts from 1, but in the airframe configuration file, it starts from 0.

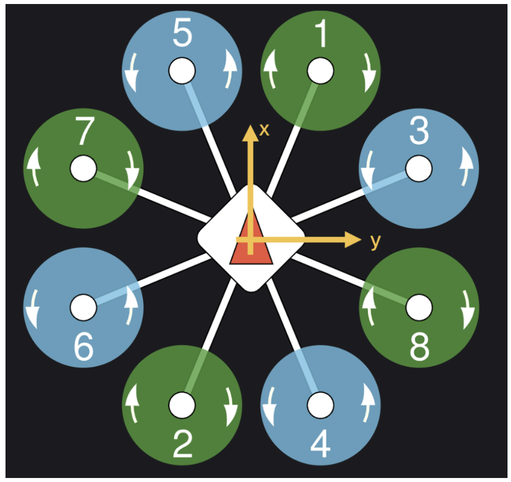

Figure 7: Rotor Definition of Octorotor x in PX4

```
param set-default CA_ROTOR0_PX 0.59
param set-default CA_ROTOR0_PY 0.245

param set-default CA_ROTOR1_PX -0.59
param set-default CA_ROTOR1_PY -0.245

param set-default CA_ROTOR2_PX 0.245
param set-default CA_ROTOR2_PY 0.59

param set-default CA_ROTOR3_PX -0.59
param set-default CA_ROTOR3_PY 0.245

param set-default CA_ROTOR4_PX 0.59
param set-default CA_ROTOR4_PY -0.245

param set-default CA_ROTOR5_PX -0.245
param set-default CA_ROTOR5_PY -0.59

param set-default CA_ROTOR6_PX 0.245
param set-default CA_ROTOR6_PY -0.59

param set-default CA_ROTOR7_PX -0.245
param set-default CA_ROTOR7_PY 0.59
```

- Torque coefficients were assigned according to the octorotor x configuration shown in Figure 7. Positive torque is defined as counterclockwise (CCW) rotation.

```
param set-default CA_ROTOR0_KM -0.05
param set-default CA_ROTOR1_KM -0.05
param set-default CA_ROTOR2_KM  0.05
param set-default CA_ROTOR3_KM  0.05
param set-default CA_ROTOR4_KM  0.05
param set-default CA_ROTOR5_KM  0.05
param set-default CA_ROTOR6_KM -0.05
param set-default CA_ROTOR7_KM -0.05
```

5.4.2 Airframe Configuration File Integration

The custom airframe configuration required integration into the PX4 build system:

1. Ensure the custom airframe configuration file is saved in the following directory:

```
~/PX4-Autopilot/ROMFS/px4fmu_common/init.d-posix/airframes
```

2. Navigate to the appropriate `CMakeLists.txt` and add the file name under `px4_add_romfs_files()`:

```
cd ~/PX4-Autopilot/ROMFS/px4fmu_common/init.d-posix/airframes/
gedit CMakeLists.txt
```

```
px4_add_romfs_files(
    4001_gz_x500
    4002_gz_x500_depth
    4003_gz_rc_cessna
    4004_gz_standard_vtol
    4005_gz_x500_vision
    4006_gz_px4vision
    4008_gz_advanced_plane
    4009_gz_r1_rover
    4010_gz_x500_mono_cam
    4011_gz_lawnmower
    4012_gz_t18_mono_cam
)
```

Note that each airframe configuration file must begin with a unique numeric identifier; in this case, the file was named `4012_gz_t18_mono_cam`.

5.5 Gazebo Model Development

5.5.1 Gazebo model.sdf and model.config Files

Following the structure of the x500 quadrotor model as a template, the Gazebo models were organized into hierarchical folders. Each model folder must contain a `model.config` file, which provides model identification and metadata about the model, and a `model.sdf` file, which defines the model's geometry, visuals, and physical properties. For the T18 octorotor, three folders were created: t18_base, t18, and t18_mono_cam.

- **t18_base**: Defines the geometry, mass, collision, and imports CAD models. Key parameters include:
  - Inertial properties

```
        <inertial>
            <mass>11.5</mass> <!-- excluding payload and rotor -->
            <inertia> <!-- found from SolidWorks -->
              <ixx>1.045</ixx>
              <ixy>0</ixy>
              <ixz>0</ixz>
              <iyy>1.608</iyy>
              <iyz>0</iyz>
              <izz>0.965</izz>
            </inertia>
        </inertial>
```

    Note that rotor mass is not included here as it is defined separately.
  - Collision geometry
  - Visual model from importing CAD models. The CAD models must be saved inside the folder called `meshes`.
  - Rotor definition. Note that Gazebo uses a different reference frame (Figure 8) than PX4 (Figure 7). However, the rotor locations and rotation direction in Gazebo must match those in PX4.

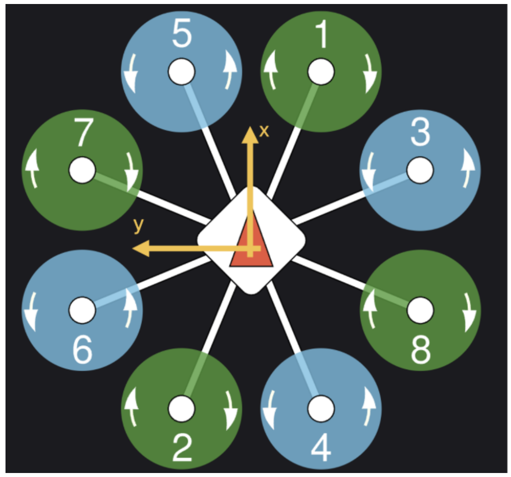

Figure 8: Rotor Definition of Octorotor x in Gazebo

The rotors were defined in Gazebo as:

```
<link name="rotor_0">
    <pose>0.59 -0.245 0.488 0 0 0</pose>
<link name="rotor_1">
    <pose>-0.59 0.245 0.488 0 0 0</pose>
<link name="rotor_2">
    <pose>0.245 -0.59 0.488 0 0 0</pose>
<link name="rotor_3">
    <pose>-0.59 -0.245 0.488 0 0 0</pose>
<link name="rotor_4">
    <pose>0.59 0.245 0.488 0 0 0</pose>
<link name="rotor_5">
    <pose>-0.245 0.59 0.488 0 0 0</pose>
<link name="rotor_6">
    <pose>0.245 0.59 0.488 0 0 0</pose>
<link name="rotor_7">
    <pose>-0.245 -0.59 0.488 0 0 0</pose>
```

- **t18**: Takes the `t18_base` model and adds motor simulation plugins. The parameters defined for each rotor were:
  - `<joint name>`: Name of the joint in the `t18_base` model
  - `<linkname>`: The link connected to the joint in the `t18_base` model
  - `<turningDirection>`: Motor turning direction (cw or ccw), matching the `t18_base`
  - `<timeConstantUp>`: Time constant simulating motor acceleration responsiveness (spin-up)

– `<timeConstantDown>`: Time constant for motor spin-down

– `<maxRotVelocity>`: Maximum rotor speed (rad/s) before clipping

– `<motorConstant>`: Maps motor input (PWM or command) to thrust (N/(rad/s)^2)

– `<momentConstant>`: Maps motor thrust to torque

– `<commandSubTopic>`: ROS2/Gazebo transport topic where PX4 sends motor commands

– `<motorNumber>`: Index of the motor

– `<rotorDragCoefficient>`: Drag coefficient for simulating rotor-induced drag force

– `<rollingMomentCoefficient>`: Rolling coefficient for rolling torques

– `<rotorVelocitySlowdownSim>`: Scales down rotor velocity to align with simulation (simulates gear ratio)

– `<motorType>`: Specifies the motor control type. `velocity` means PX4 sends angular velocity commands

Note that these parameters were tuned based on the simulation and may differ from the actual T18 parameters.

- **t18_mono_cam**: Takes the `t18` model and attaches a monocular camera.

5.5.2 Final Gazebo Model

The final T18 octorotor model in Gazebo is as follows:

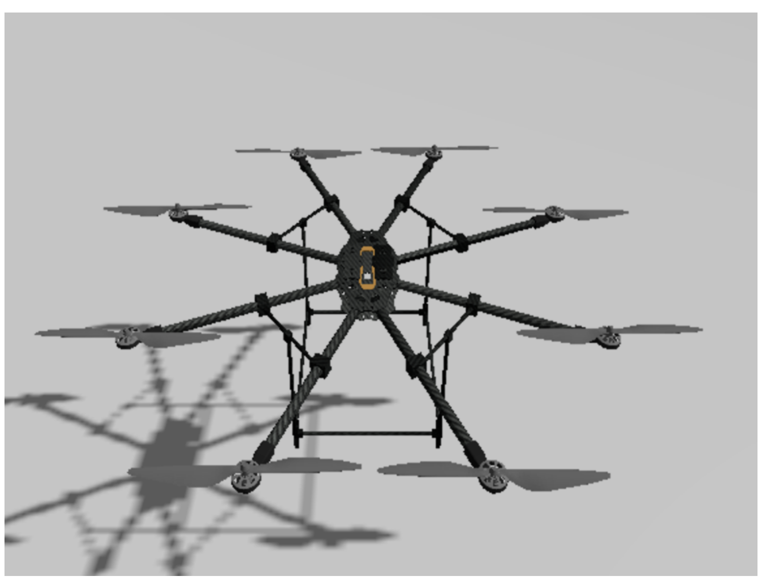

Figure 9: Gazebo Model of the T18 Octorotor

5.6 ROS2 Node Development

5.6.1 ROS2 Node Architecture

The autonomous flight control system was implemented using a ROS2 node interfacing with PX4 through the Micro XRCE-DDS communication bridge:

- **Publisher**:
  - Vehicle command messages for mode switching and system arming
  - Trajectory setpoint (waypoint) messages for position and velocity control
  - Offboard control mode activation and management
- **Subscriber**:
  - Real-time vehicle position feedback
  - Flight mode status monitoring and verification

5.6.2 Mission Execution Logic

The autonomous flight control implements a simple mission consisting of waypoint following:

1. **System Initialization**: Set up publisher and subscriber to establish communication with PX4
2. **Automated Arming**: Enable motors and perform controlled takeoff
3. **Waypoint Navigation**: Follow predetermined flight paths using PX4's closed-loop position control
4. **Mission Completion**: Execute controlled landing

5.7 System Integration and Execution

5.7.1 Simulating Onboard Control with QGroundControl

1. Open QGroundControl
2. Launch PX4 SITL and Gazebo:

```bash
cd ~/PX4-Autopilot/
PX4_SYS_AUTOSTART=4012 \
PX4_SIM_MODEL=gz_t18 \
PX4_GZ_MODEL_POSE="0,0,0,0,0,0" \
PX4_GZ_WORLD=customforest \
~/PX4-Autopilot/build/px4_sitl_default/bin/px4
```

Note that `4012` refers to the custom airframe configuration. Either `gz_t18` (T18 octorotor) or `gz_t18_mono_cam` (T18 with monocular camera) can be selected depending on the simulation purpose.

3. Set the desired path and upload the mission from QGroundControl
4. Slide to Confirm mission execution

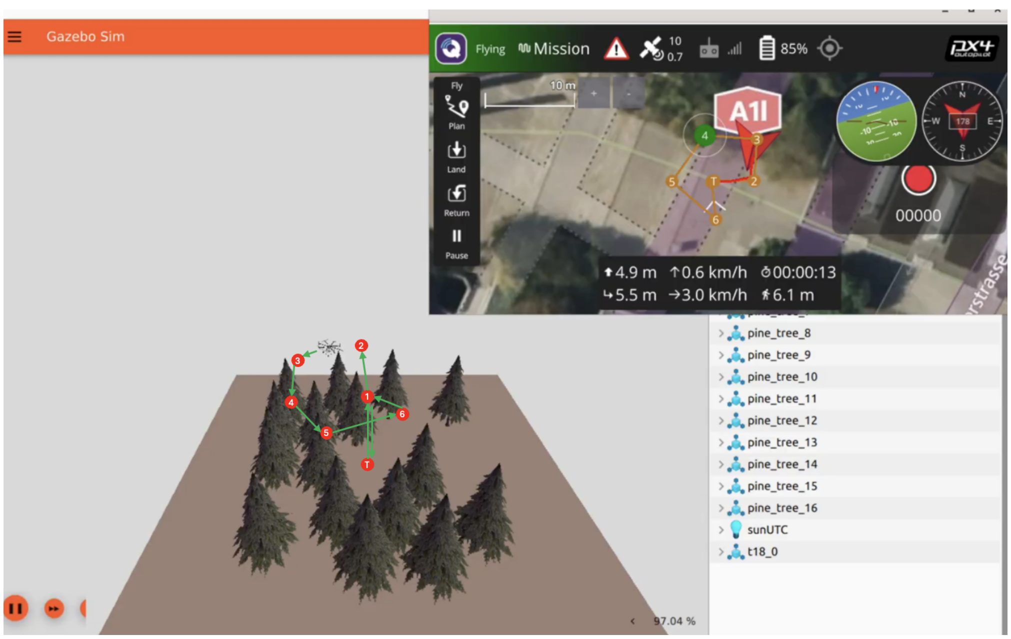

Figure 10: T18 Simulation with QGroundControl

5.7.2 Simulating Offboard Control with ROS2 Node

1. Initialize Micro XRCE Agent:

```bash
cd ~/px4_ros_uxrce_dds_ws/
source /opt/ros/humble/setup.bash
source install/local_setup.bash
MicroXRCEAgent udp4 -p 8888
```

2. Launch PX4 SITL and Gazebo:

```
cd ~/PX4-Autopilot/
PX4_SYS_AUTOSTART=4012 \
PX4_SIM_MODEL=gz_t18 \
PX4_GZ_MODEL_POSE="0,0,0,0,0,0" \
PX4_GZ_WORLD=customforest \
~/PX4-Autopilot/build/px4_sitl_default/bin/px4
```

3. Execute ROS2 Node:

```
cd ~/fsc_octorotor_simulation_ros2/ros2_node
colcon build
source install/local_setup.bash
ros2 run my_offboard_ctrl offboard_ctrl_example
```

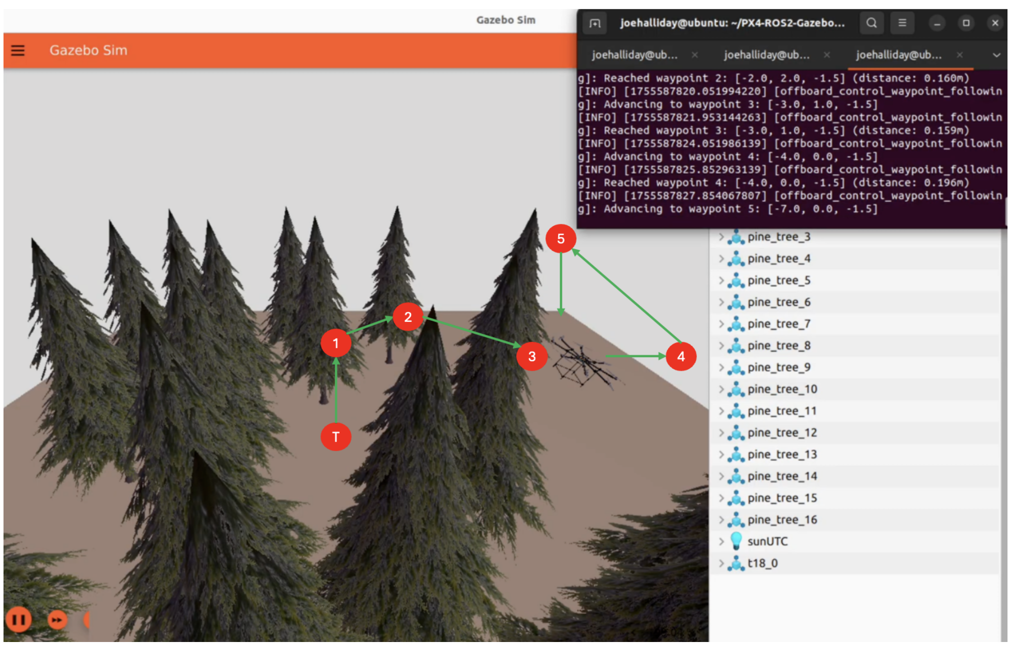

Figure 11: T18 Simulation with ROS2 Node

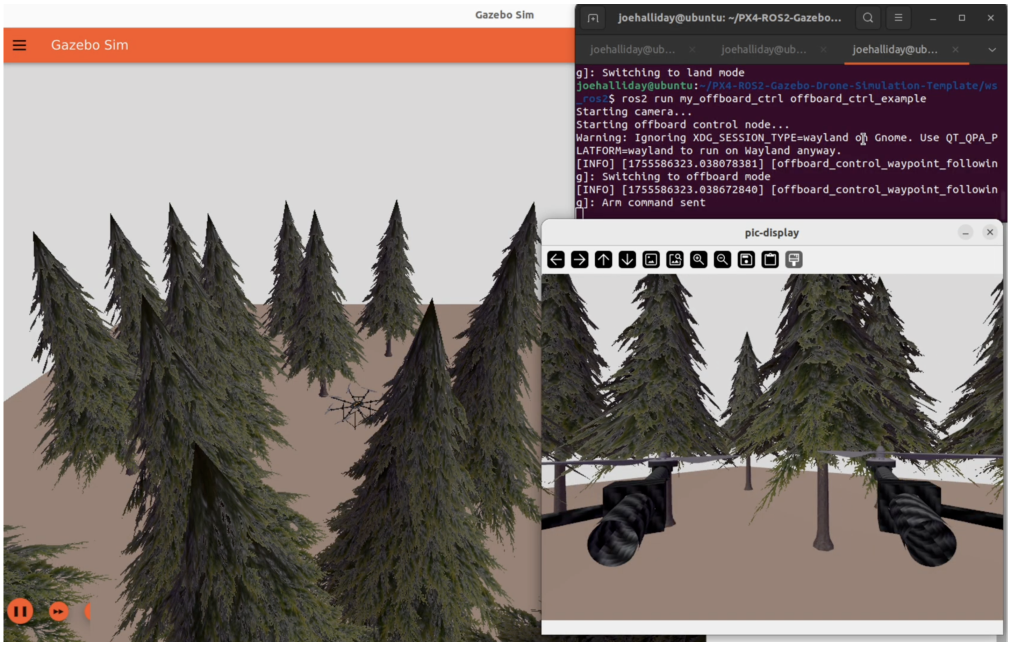

Figure 12: T18_mono_cam Simulation with ROS2 Node

6 Conclusions and Future Work

6.1 Research Summary and Achievements

This research successfully established the foundational framework for autonomous Solar-Induced Fluorescence (SIF) monitoring using the Tarot T18 octorotor. The work addressed critical gaps in current SIF measurement methodologies by developing a simulation environment that bridges the resolution limitations of satellite-based systems and the spatial coverage constraints of ground-based approaches. The primary achievements of this research include:

1. **T18 Octorotor Specification**: A detailed specification sheet for the Tarot T18 octorotor, including components, dimensions, and performance calculations. The payload capability of up to 8kg and endurance of up to 39.6 minutes make the T18 octorotor well-suited for SIF monitoring applications.

2. **Complete Simulation Framework Development**: A simulation environment was successfully developed using PX4, Gazebo Harmonic, and ROS2 Humble. This framework supports Software-in-the-Loop (SITL) simulations, allowing critical testing before performing real-world flights.

3. **Custom Forest Gazebo World Creation**: In Gazebo, a realistic forest simulation world was developed, incorporating multiple pine trees and environmental characteristics appropriate for SIF monitoring scenarios. This environment serves as a testing platform for algorithm development and mission validation, as well, additional worlds can be derived from it.

4. **Accurate 3D Model Integration**: CAD models of the T18 octorotor were created in SolidWorks and successfully integrated into the Gazebo simulation environment. The models include accurate dimensions, inertial characteristics, and visual representation derived from actual specifications.

5. **PX4 Airframe Configuration and Gazebo Models**: A custom airframe configuration file and Gazebo model files were developed and integrated into the PX4 build system, enabling proper flight control and stability for the T18 octorotor within the simulation environment.

6. **Autonomous Flight Capability**: Both onboard control through QGroundControl and offboard control via ROS2 nodes were successfully demonstrated, providing multiple pathways for autonomous mission execution and algorithm testing. The drone reliably reached designated waypoints with stable flight.

This work represents a critical initiation phase of the broader SIF monitoring project, establishing the technological foundation for autonomous plant health monitoring and showing the T18's capabilities for SIF monitoring. The developed simulation environment enables researchers to validate SIF sensor integration, test autonomous navigation and control algorithms, and optimize mission parameters before transitioning to physical hardware deployment.

6.2 Future Work

- **T18 Octorotor Specific Control Algorithm and Dynamic Model Development**: The drone demonstrated stable flight in simulation; however, it exhibited undesired yaw rotation and instability when navigating between widely spaced waypoints. A potential solution is the development of a control algorithm and dynamic model tailored specifically to the T18 octorotor.

- **Simulation vs. Real-World Performance and Parameters**: The performance metrics and parameters were derived through theoretical analysis and simulation-based tuning, but discrepancies are expected when compared to real-world flight performance.

- **SIF Sensor Integration**: The current simulation framework does not include actual SIF sensor models, data collection algorithms, or SIF sensor payload, limiting the ability to test complete end-to-end SIF monitoring workflows.

- **Payload Modelling**: The simulation currently excludes payload considerations, which will impact flight performance and may require recalibration of parameters.

- **Custom ROS2 Node Development**: FSC members are developing a custom ROS2 node and autopilot, with the next step being the integration of these codes into the simulation framework.

References

[1] (2025) Xseed 2025 description. University of Toronto Institute for Aerospace Studies.

[2] T. S. M. et al., “Mechanistic evidence for tracking the seasonality of photosynthesis with solar-induced fluorescence,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 24, pp. 11 640–11 645, Jun. 2019. [Online]. Available: https://doi.org/10.1073/pnas.1900278116

[3] F. Patel. (2025, Jul.) Solar induced fluorescence (sif) - observing the subtle glow of plants from space to better understand the carbon cycle. National Centre for Earth Observation. Accessed: Aug. 24, 2025. [Online]. Available: https://www.nceo.ac.uk/news-media/solar-induced-fluorescence-sif-observing-the-subtle-glow-of-plants-from-space-to-better-understand-the

[4] S. Bandopadhyay, A. Rastogi, and R. Juszczak, “Review of top-of-canopy sun-induced fluorescence (sif) studies from ground, uav, airborne to spaceborne observations,” *Sensors*, vol. 20, no. 4, p. 1144, Feb. 2020. [Online]. Available: https://doi.org/10.3390/s20041144

[5] UAV Systems International, *TAROT T18 User Manual*. [Online]. Available: https://www.scribd.com/document/466930342/tarot-t18-user-manual-1-1-1?doc_id=466930342&download=true&order=673285989

[6] Docs.px4.io. (2025) Holybro pixhawk 6x-rt — px4 guide (main). [Online]. Available: https://docs.px4.io/main/en/flight_controller/pixhawk6x-rt