Coppelia Scene: Omnidirectional MultiRotor UAV

By,
Akshay Dhonthi Ramesh Babu - 1887502
Fabian Humberto Fonseca Aponte - 1886565
David Esteban Imbajoa Ruiz - 1922212

Academic Year 2019 - 2020

Introduction

- Continuous search for improvement in terms of **autonomy** and **flexibility** for air vehicles. The trend continued in **unmanned vehicles**.

- The history of this line of research has branched out into many fields, from which terms like **Fully Actuated Drones - FAD** and **Over Actuated Drones - OAD** take an important part of this project.

- Each branch has its own difficulties in terms of *location*, *mapping* and *control theory* with respect to its **parameters** and characteristics.

- There is great interest in s**imulation environments** that test the performance of solutions without putting investment capital at risk.

Problem Statement

Build a virtual scene in CoppelliaSim for the design, implementation and evaluation of position and attitude control alternatives over an omnidirectional UAV

Objective

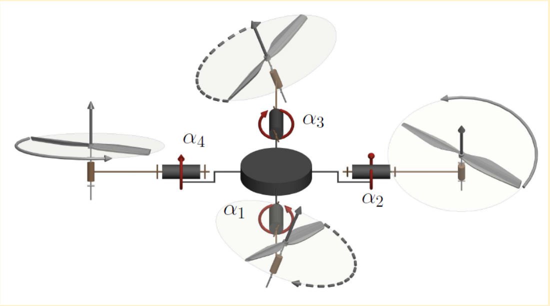
The presentation will cover the following section of the project:

- ***Problem Statement:*** The purpose of the project.
- ***Methodology:*** Methods and algorithms for the control proposed.
- ***Implementation:*** UAV Model design and integration to CoppeliaSim.
- ***Experiments and* results*:*** Proposed experiments and associated results.
- **Conclusion:** Milestones achieved at the end of the project.

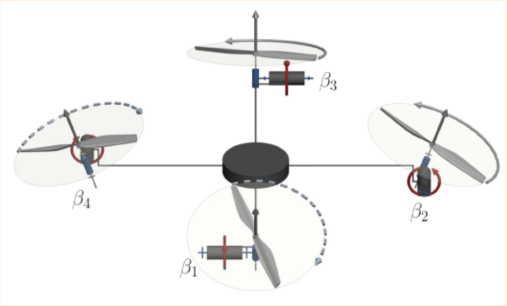
Basic Knowledge - Concepts

Atomic actuation units (AAU) characterization	Properties
Aerodynamic coefficients - Related to propellers. **(Uni/bi)-directional thrust** - The motors rotates in both directions or only one. **Fixed/actuated propulsion shafts** - Rigid body or actuated joints wrt propellers frame. **Propellers positions** - Spatial location.	**Hover Ability** - Keeps a constant configuration (position and orientation). **Trajectory tracking Ability** - Tracks position and orientation in the reference space. **Physical Interaction Ability** - Interacts with environment through changing forces.

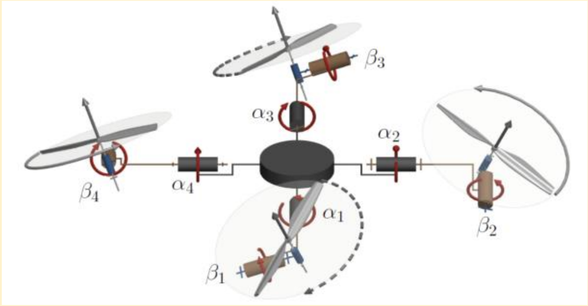
Basic Knowledge-Model



Radial Tilt Design



Tangential Tilt Design



Mixed Tilt Design

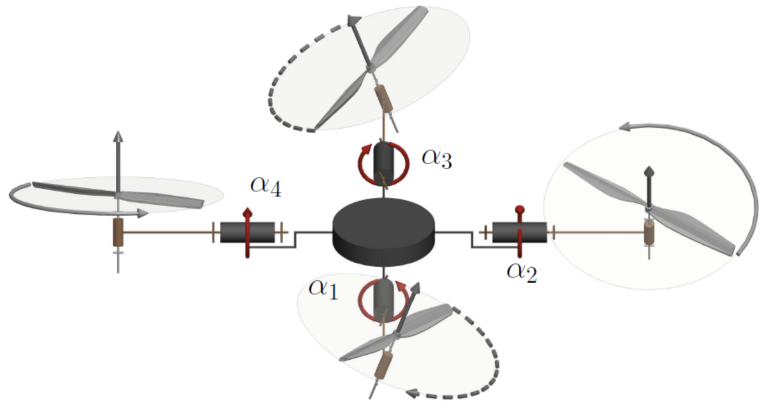
Control Model

CAD Model

All the designs are Over-actuated **OA**, but the control model can be modified to be Fully-actuated **FA**

Basic Knowledge - Assumptions and Properties

- Constant center of mass **CoM**.
- The **CoM** is not affected by physical components.
- Constant aerodynamic coefficients.
- Mirroring the orientation of opposing propellers.
- Unidirectional thrust.
- Constant propellers position.
- Actuated propellers, 2 DoF (CAD Design) and 1 DoF (Control Design).



Radial Tilt Design to control

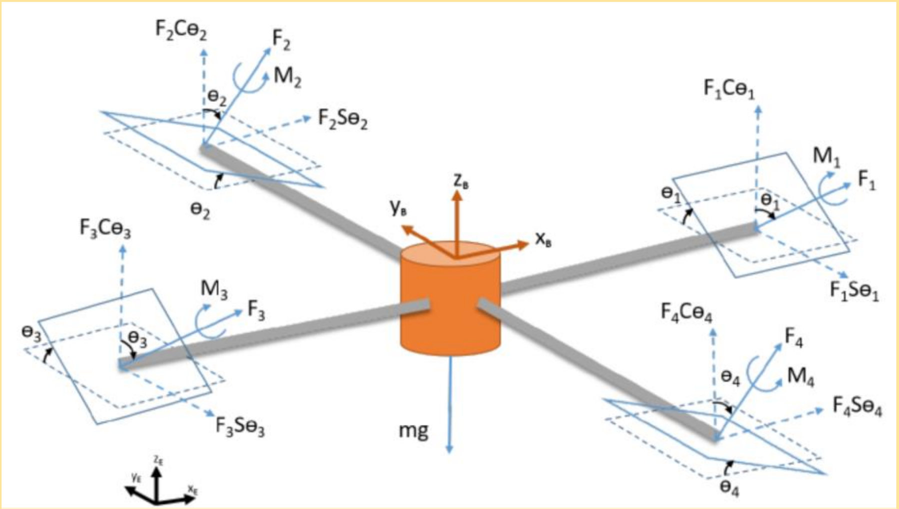
Basic Knowledge-Kinematic Model

- One coordinate axes for the base of the drone.
- One coordinate axes assigned for each propellant repeated in its counterparts.
- One coordinate axes for the environment.

Considerations:

➢ Control simplification by setting θ1 = θ3 and θ2 = θ4.

➢ Total of 6 control inputs:
4 angular velocities.
2 angular positions.



Kinematic model for an UAV with radial tilting

Basic Knowledge-Kinematic and Dynamic Model

$$R_{B/E} = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi \end{bmatrix}$$

Rotation matrix to modify the direction of the generated impulse for each propeller.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{R_{B/E}}{m} \begin{bmatrix} F_2\sin\theta_2 + F_4\sin\theta_4 \\ -F_1\sin\theta_1 - F_3\sin\theta_3 \\ F_1\cos\theta_1 + F_2\cos\theta_2 + F_3\cos\theta_3 + F_4\cos\theta_4 \end{bmatrix} - \begin{bmatrix} C_1\dot{x} \\ C_2\dot{y} \\ C_3\dot{z} + g \end{bmatrix}$$

Acceleration wrt global reference frame, where, ***C*** is the drag coefficient and ***F*** is the propellers force.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\frac{\sin\theta}{\cos\theta} & \cos\phi\frac{\sin\theta}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Global rate of change of orientation wrt body frame, where $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ represents roll, pitch and yaw respectively and ***p, q ,r*** are the rates of $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$.

Methodology - **PD** Position & attitude control

$$\ddot{r_{des}} = k_{pi}e_i + k_{di}\dot{e_i} = \begin{bmatrix} k_{px}e_x \\ k_{py}e_y \\ k_{pz}e_z \end{bmatrix} + \begin{bmatrix} k_{dx}\dot{e_x} \\ k_{dy}\dot{e_y} \\ k_{dz}\dot{e_z} \end{bmatrix} ; \forall i \in \{x, y, z\}$$

The desired acceleration is given by proportional (***Kp***) and derivative (***Kd***) gains, given the error calculations of the desired references.

$$\phi^{des} = \frac{\ddot{r_x}^{des} \sin\psi^{des} - \ddot{r_y}^{des} \cos\psi^{des}}{g}$$

$$\theta^{des} = \frac{\ddot{r_x}^{des} \cos\psi^{des} - \ddot{r_y}^{des} \sin\psi^{des}}{g}$$

The desired orientation, in Euler angles, for a given acceleration and a given $\boldsymbol{\psi}$ desired.

Methodology - **PD** Position & attitude control

$$w_h = \sqrt{\frac{mg}{2k_f(\cos\theta_1 + \cos\theta_2)}}$$

$$\Delta w_h = \frac{m\ddot{r}_z^{\,des}}{4k_f w_h(\cos\theta_1 + \cos\theta_2)}$$

The angular velocity and its rate of change required for hovering is given by ***ωh*** and ***Δωh***
where, ***kf*** is a constant from the motor forces.

$$F_i = k_f w_i^2$$

Associated forces produced by each propeller given a constant ***kf***.

Methodology - **PD** Position & attitude control

$$\Delta w_\phi = k_{p,\phi}(\phi^{des} - \phi) + k_{d,\phi}(p^{des} - p)$$
$$\Delta w_\theta = k_{p,\theta}(\theta^{des} - \theta) + k_{d,\theta}(q^{des} - q)$$
$$\Delta w_\psi = k_{p,\psi}(\psi^{des} - \psi) + k_{d,\psi}(r^{des} - r)$$
$$\Delta\theta_{T_\phi} = k_{p,\phi_T}(\phi^{des} - \phi) + k_{d,\phi_T}(p^{des} - p)$$
$$\Delta\theta_{T_\theta} = k_{p,\theta_T}(\theta^{des} - \theta) + k_{d,\theta_T}(q^{des} - q)$$
$$\Delta\theta_{T_x} = k_{p,x_T} e_x + k_{d,x_T} \dot{e_x}$$
$$\Delta\theta_{T_y} = k_{p,y_T} e_y + k_{d,y_T} \dot{e_y}$$

- $\boldsymbol{\Delta\omega}$ represents the change of angular velocity needed for the rotors to achieve the desired orientation of the UAV.
- $\boldsymbol{\Delta\theta}$ represents the change in pitch needed for the rotors to achieve the pose (position and orientation).

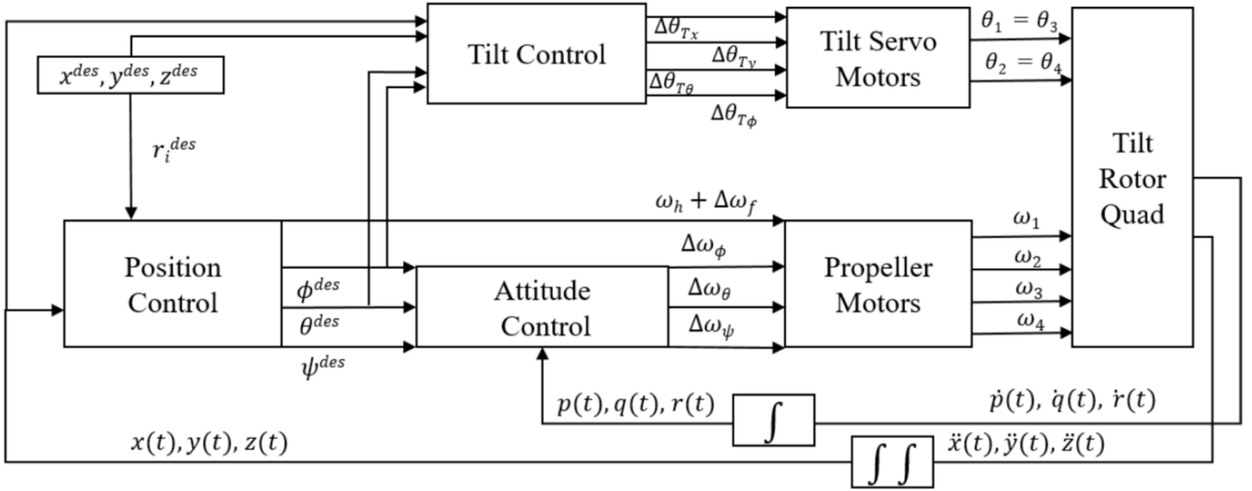
Methodology - **PD** Position & attitude control

$$\begin{bmatrix} w_1^{des} \\ w_2^{des} \\ w_3^{des} \\ w_4^{des} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_h + \Delta w_f \\ \Delta w_\phi \\ \Delta w_\theta \\ \Delta w_\psi \\ \Delta \theta_{T_y} \\ \Delta \theta_{T_x} \\ \Delta \theta_{T_\phi} \\ \Delta \theta_{T_\theta} \end{bmatrix}$$

Linearly dep.

Final representation for the control inputs as linear combination of angular velocities ($\boldsymbol{\omega}$) and tilt angles ($\boldsymbol{\theta}$).

Methodology - Position/attitude control



Control Architecture of Tilting-Rotor quadrotor

The desired **PD** controller synchronizes the actions taken on the angular velocities and the inclination of the rotors.

Implementation

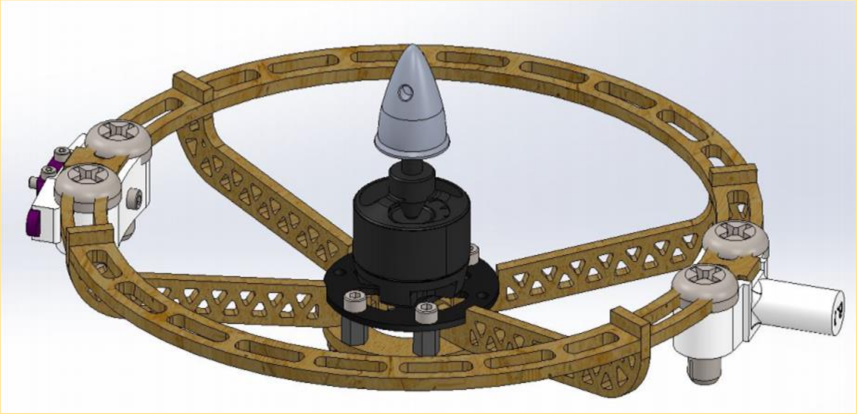
- **CAD Model Development**
  - Quadrotor design with 3 dofs in each propeller.
  - Radial and tangential movements for each propeller.
- **Integrating model into CoppeliaSim**
  - Grouping and materials.
  - Shapes development.
  - Joints and parent-child structure.
  - Lua Script development.
- **Tuning gains and Iterative Experimentation**
  - Experimentation route.
  - Fine tuning for **PD** gains.

Implementation - CAD Model Development

The preliminary model is obtained from GRABCAD Community

- The quadrotor with propellers tilting in $\mathbf{s}^2$ (radial and tangential).
- Over-actuated system.
- The tilt angles controlled by servo motors.
- Servo motors fast enough to ignore the control loop.

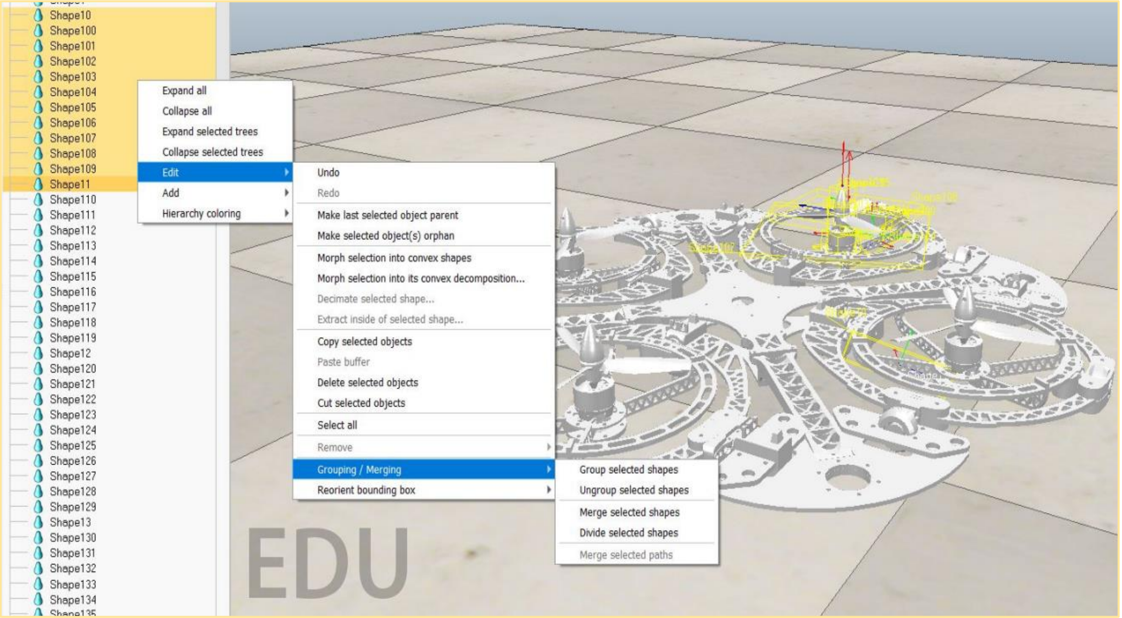
Implementation - CAD Model Development



Single propeller assembled in SolidWorks

- Designed parts in "SLDPRT" format and assembled.
- Propellers axis facing upward to obtain the proper coordinate frame.
- Body's Z-axis faces upward to maintain CoppeliaSim standards.

Implementation - Integration into CoppeliaSim

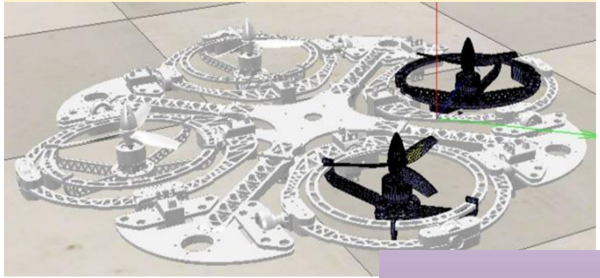


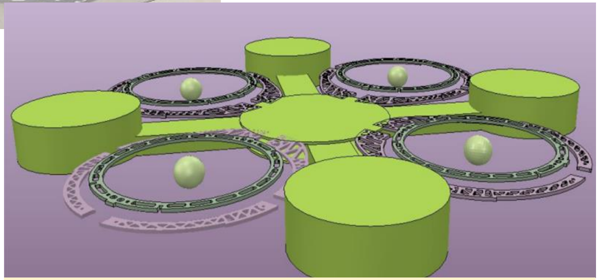
Grouping of multiple non-pure shapes on CoppeliaSim

- STL files generated by SolidWorks is exported to CoppeliaSim and grouped with their respective fixed parts.
- Finally obtain a total of 13 multishape non - pure objects (**1** body, **4** propellers, **4** radial rings and **4** tangential rings).

Implementation - Integration into CoppeliaSim

Triangle edit mode available in Coppelia Sim

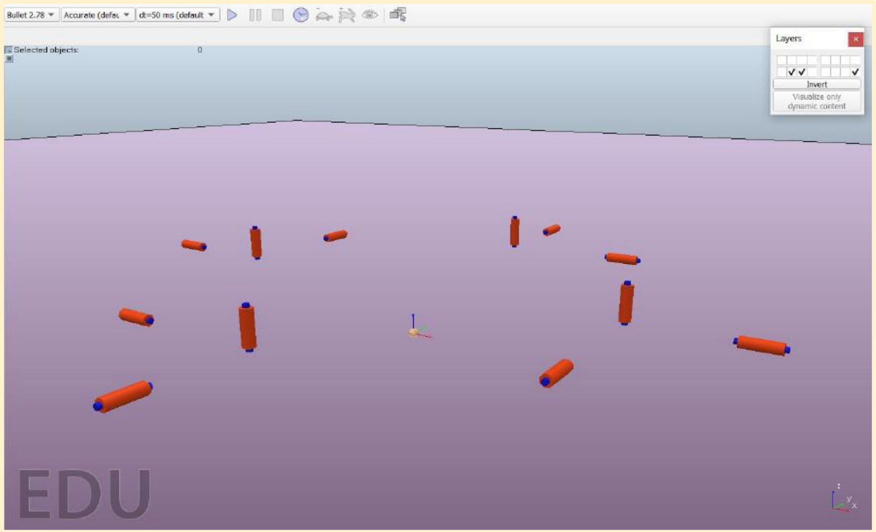




Final hidden layer with all pure shapes

- Model is converted to pure shapes.
- Done using triangle edit mode.
- Reduces total number of meshes.
- Results in smoother dynamic simulation.
- Guaranties immediate and smooth response as the shapes are stable.

Implementation - Integration into CoppeliaSim

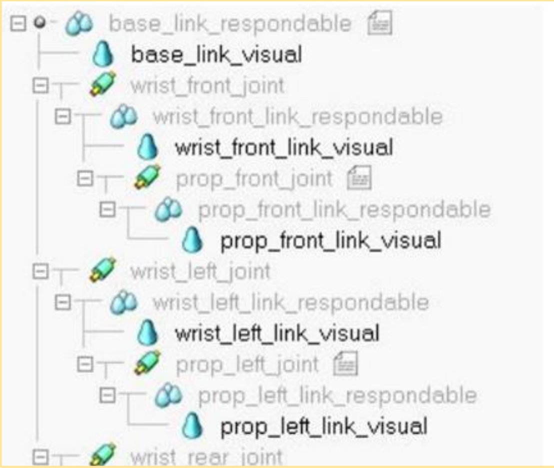


Joints in space for UAV model design

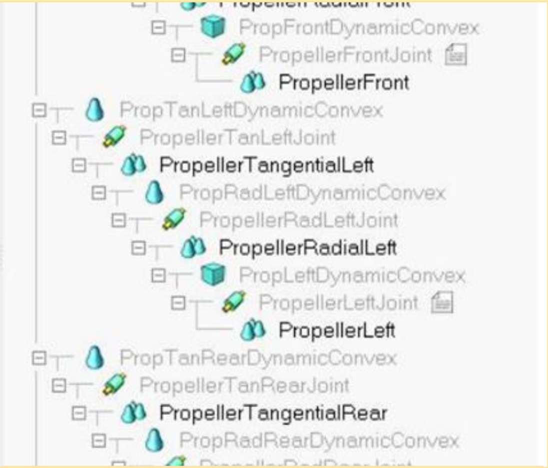
- Joint objects generated to assign the type of motion (revolute, prismatic or spherical).

- The exact position of each joint obtained by extracting pure triangles shapes from physical joint shafts.

Implementation - Integration into CoppeliaSim



Hierarchy for Radial tilt and non pure part



Hierarchy for Radial and tangential tilt with pure part

- The structure has pure shape then non pure shape and then the joint respectively for all DoFs.
- Revolute joints are added to their respective tangential tilt rings as shown in hierarchy structure.

Implementation - Integration into CoppeliaSim

Lua script development

- The force and torque are calculated and sent to the skeleton of the UAV.
- The propellers are rotated with fixed speed, just for visualisation.
- The correct tilt angles can be visualised.
- All the functions are programed on non-thread child script.

Tuning gains-Iterative experimentation

- There are around 20 gains which are tuned to perform smooth attitude control.
- Each control input gain is tuned individually by setting the others to zeros.
- Forces applied on Z-axis are set by moving the reference frame on the robot by transforming the ground and the robot frame.

Experimentation - Expected Results

Model DOF → Movement visualization and assembly on SolidWorks to understand the degrees of freedom

↓

Importing to CoppeliaSim → The propeller and tilt angles are tested on the CoppeliaSim scene to confirm proper placement of rotation joints.

↓

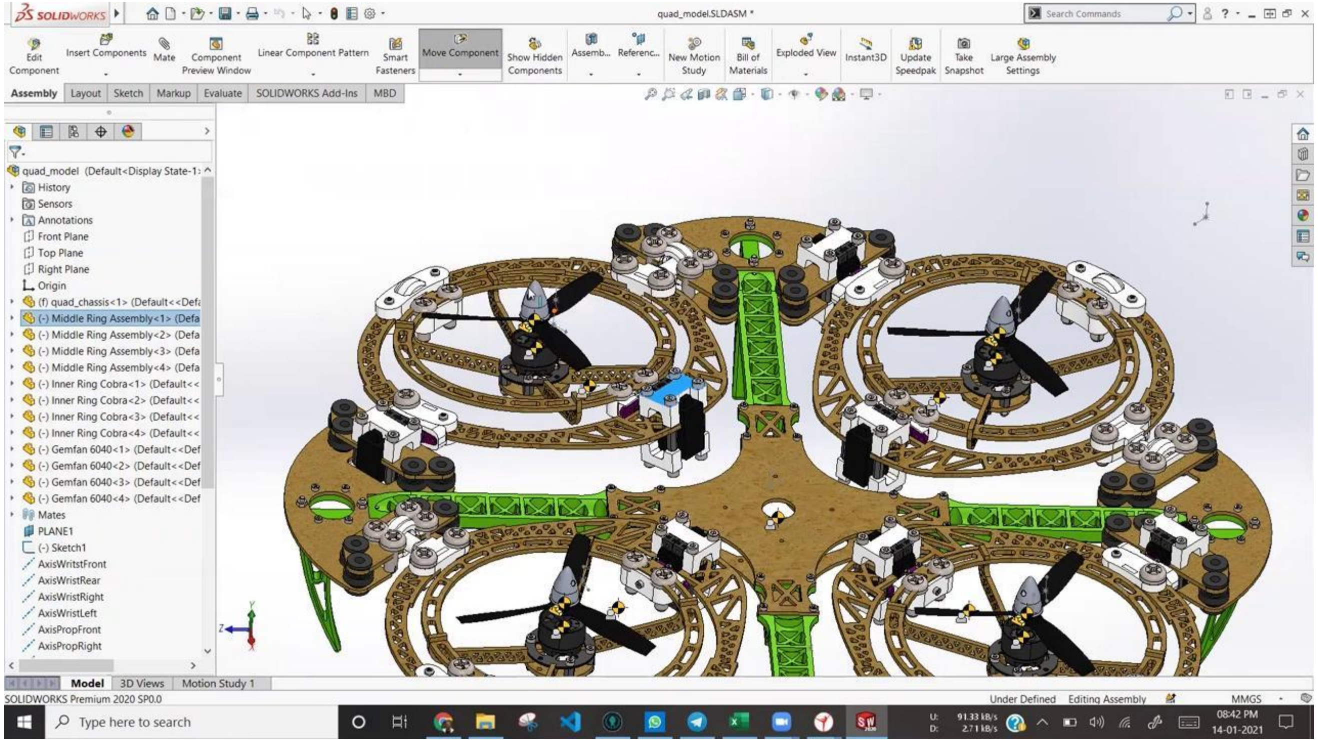
Simple Quad Attitude control → Initial simulation result with simple attitude control without tilt angles

↓

Attitude control with radial tilt angles → Final Attitude control with radial tilt angles to reach desired position and orientation.

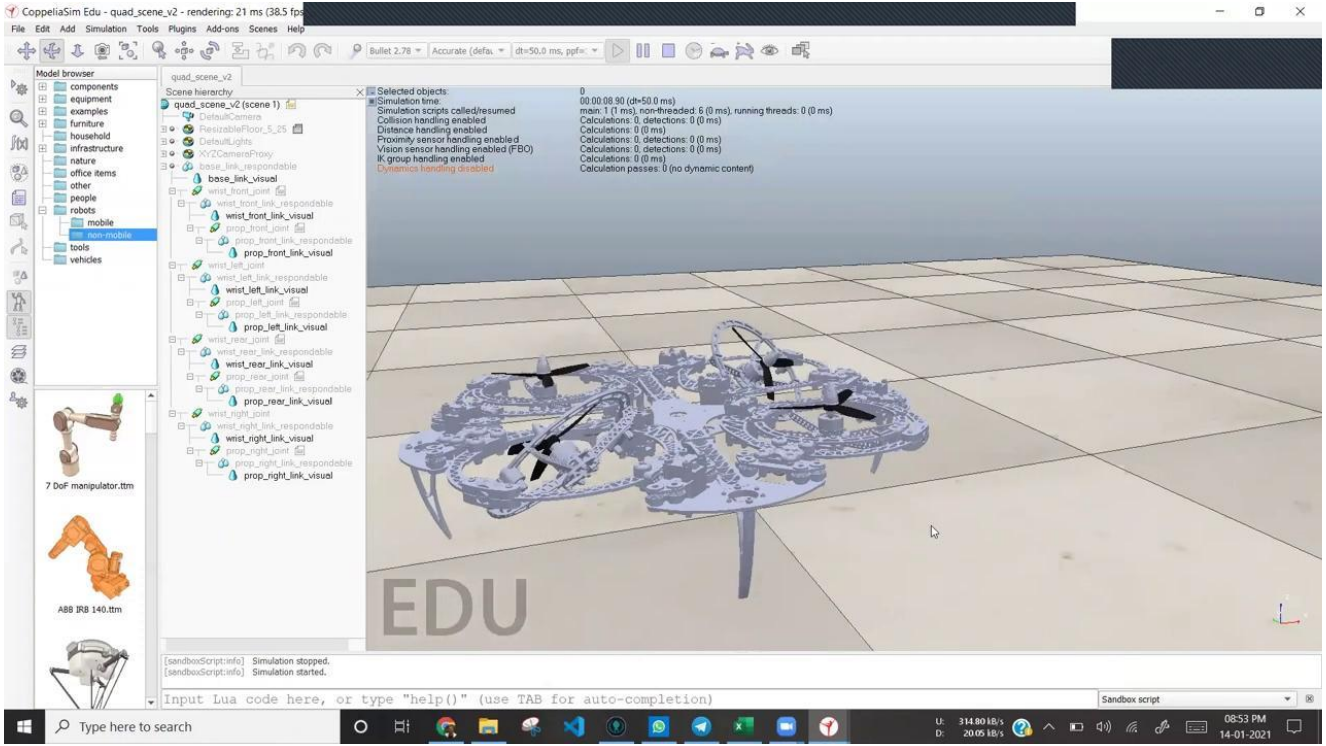
Experimentation

Model DOF



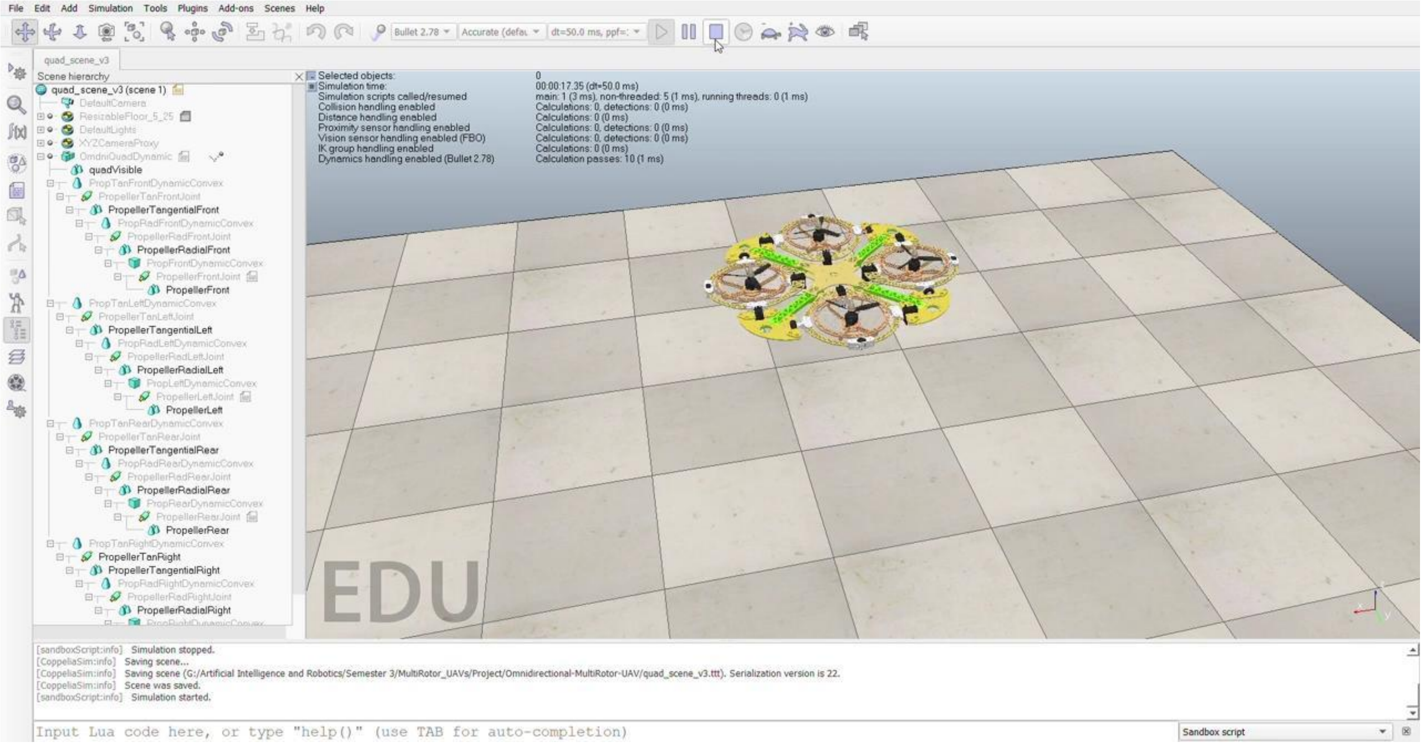
Experimentation

Importing to CoppeliaSim



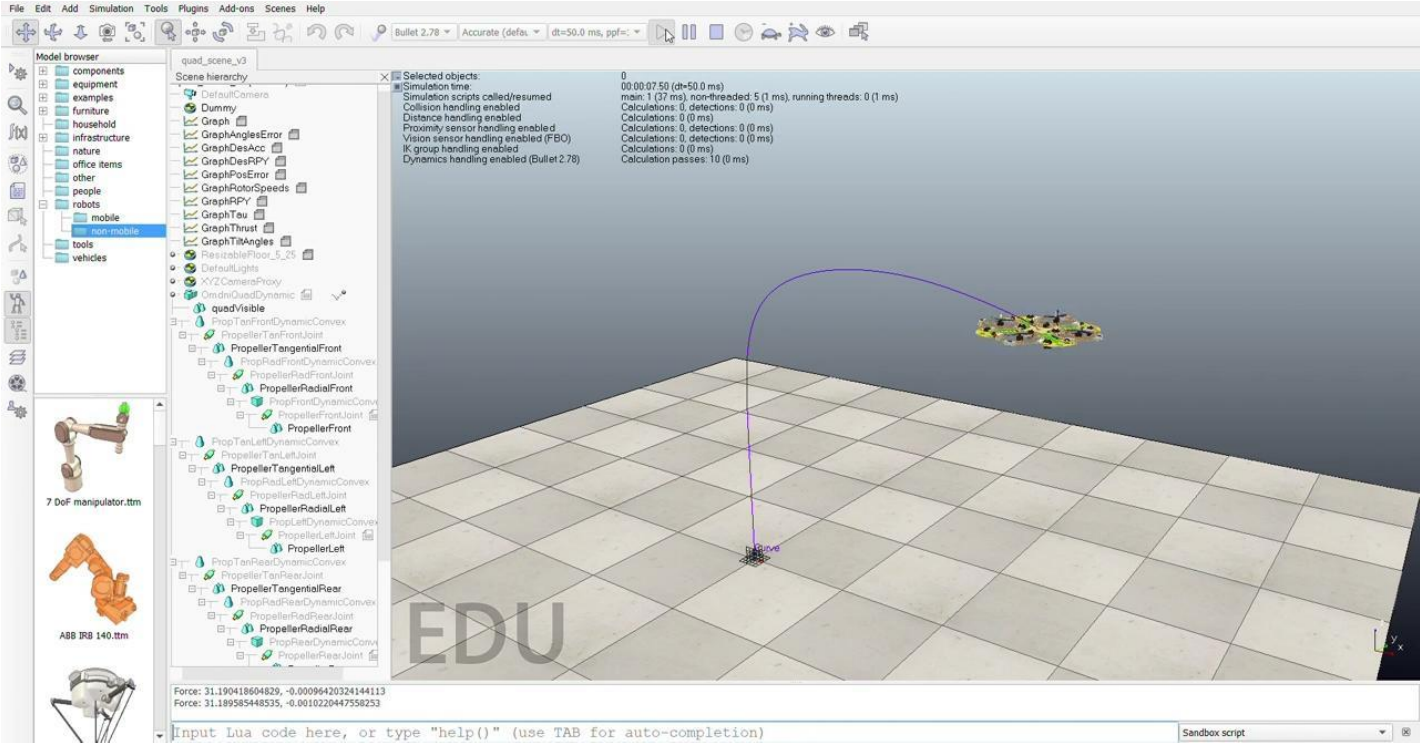
Experimentation

Simple Quad Attitude control



Experimentation

Attitude control with radial tilt angles



Conclusion

- A scene was successfully developed in CoppeliaSim with an **omnidirectional UAV** with 2 additional DoF in each of its propeller.

- The integration of a **LUA script** with the mechanics of the drone was achieved to control its movement in the simulated environment.

- A **fully actuated control** was implemented on the system by enabling only the radial tilt in the propellers and simplifying the model to a 6 inputs control scheme.

- The tuning of the gains was achieved presenting an acceptable behavior.

- The scene, the model and the code will serve as the basis for future projects and developments compatible with the taxonomy of this work.