# Fast-Planner Dev Tests Parameters

## **Intro**

Fast-Planner is a robust and computationally efficient planning system that enables quadrotor fast flight in complex unknown environments.

Fast Planner was originally developed by \*\*HKUST-Aerial-Robotics Group\*\*

Our goal in this project is to be able to navigate indoors around different obstacles.

In order to do so we configured the Kinodynamic path searching algorithm to work with Ardupilot and GAZEBO SITL.

This paper explains the meaning of the different parameters we explore in dev test

# Parameters

**Fast-Planner lambdas parameters:**

The fast-planner lambda parameters are different weights of different components of the path optimization problem (you can see full description in fast-planner paper <https://arxiv.org/abs/1907.01531>).

**lambda1 - Path smoothness weight:**

As this parameter increases the weight of path smoothness is higher means the path will be smoother. After manual tests we ran we discovered that for values lower than 1 the planned path has “sharp corners” that makes it difficult for the drone to follow and for values of 100 and bigger we didn't notice different behaviour . Hence in our tests we will use values from 1 to 100

**lambda2 - Distance from obstacles weight:**

As this parameter increases the weight of the safe distance which the planned path keeps from obstacles is higher. In addition if the calculated weight is bigger than a certain threshold the planned path will try to keep a distance of 1 meter from the closest obstacle. In addition for very small values of lambda2 2 the algo will plan a path tangent to the obstacles. After manual tests we ran we decided to use values of lambda2 from 0.1 to 100 in our tests.

**lambda3 - Path feasibility weight for optimization:**

As this parameter increases the weight of maximum velocity and maximum acceleration constraints increase. It appears that this parameter is not that significant since the maximum velocity and maximum acceleration constraints are being used again later in the code.

**lambda4 - Endpoint weight for optimization:**

As this parameter increases the weight of the two last b-spline control points increases. This parameter is not significant and we’ll use the default value of 0.8.

**lambda7 - Yaw b-spline waypoints weight for optimization:**

As this parameter increases the weight of the

**dt\_yaw - Delta time of yaw segment:**

This parameter determines how many segments the path will be split to. Segment = duration / dt\_yaw.

For higher values of dt\_yaw there are less segments, which means lower yaw rate**.**