auterion

Vision and Offboard Control Interfaces

Martina Rivizzigno - 06/21/2019

Agenda

> Current Obstacle Avoidance Interface

> The message definition

> Other Companion Interfaces

> Cool features

> Community Feedback

Why is an Obstacle Avoidance interface needed?

Offboard Mode

Possible inputs:

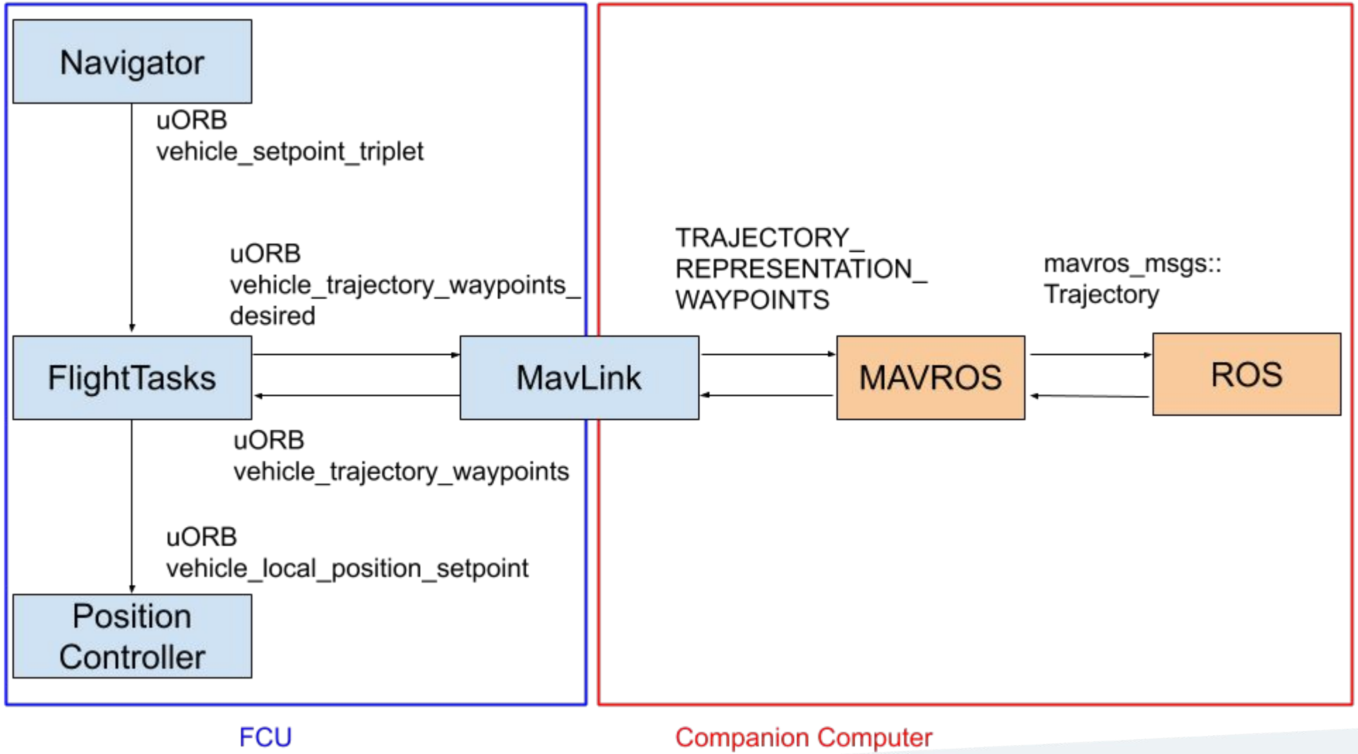
- Position setpoints
- Velocity setpoints
- Position setpoints + Velocity setpoints as feedforward
- Acceleration setpoints mapped to normalized thrust

Mavlink Message: SET_POSITION_TARGET_LOCAL_NED

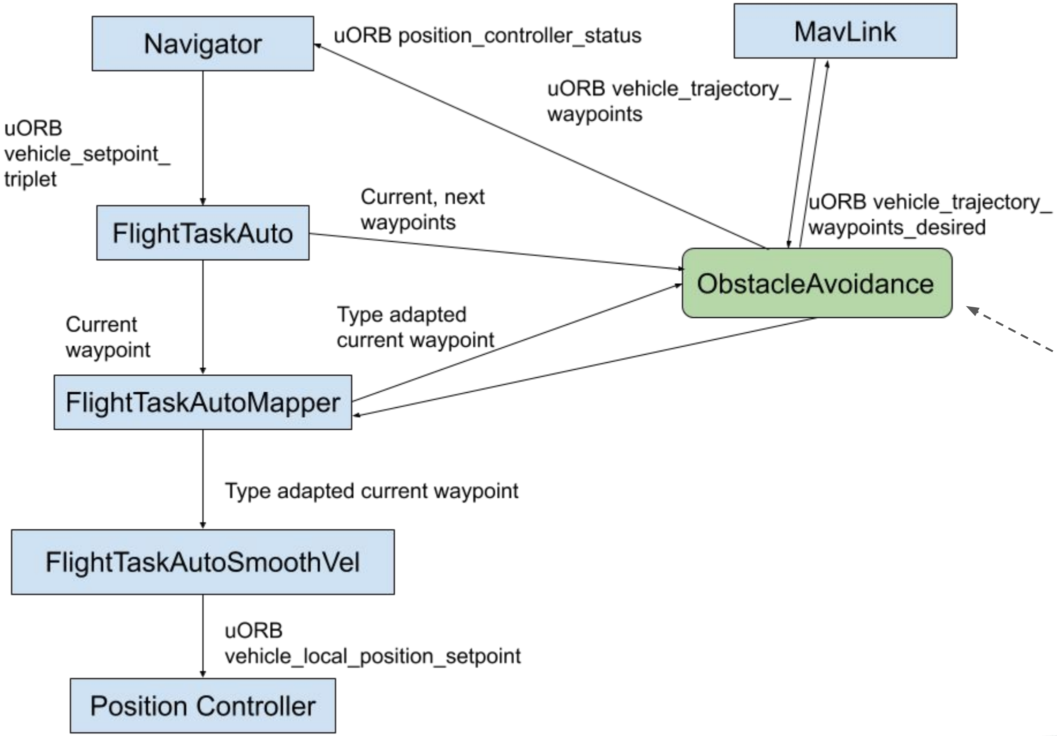
Obstacle Avoidance interface benefits

- Leverage all the features already present in the flight task when controlling the vehicle from offboard
- Achieve seamless integration of Offboard features in PX4
- Easier development cycle and testing
- Go towards a more product like integration of Obstacle Avoidance in PX4
- Not only for Obstacle Avoidance but for all computation intensive features that run on a companion computer, e.g. trajectory generation

Architecture Overview

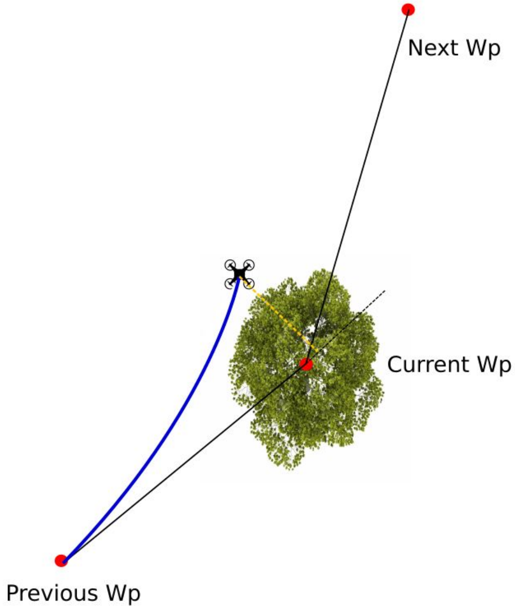


Integration into FlightTasks


Navigator
uORB position_controller_status
MavLink
uORB vehicle_trajectory_ waypoints
uORB vehicle_setpoint_ triplet
FlightTaskAuto
Current, next waypoints
uORB vehicle_trajectory_ waypoints_desired
ObstacleAvoidance
Current waypoint
Type adapted current waypoint
FlightTaskAutoMapper
Type adapted current waypoint
FlightTaskAutoSmoothVel
uORB vehicle_local_position_setpoint
Position Controller

Library running in FlightTasks

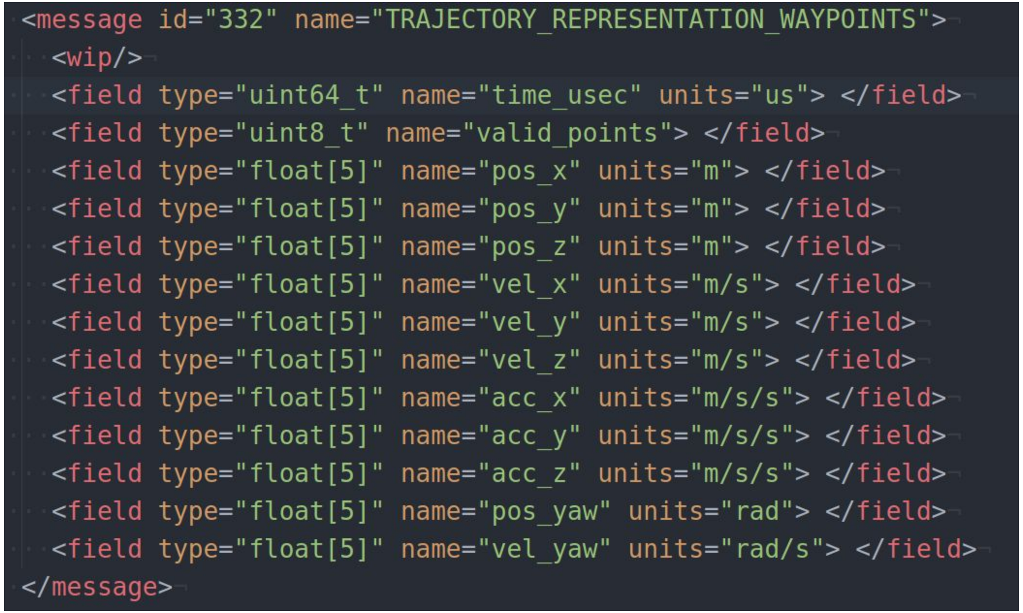
Mission Progression Check



The Message

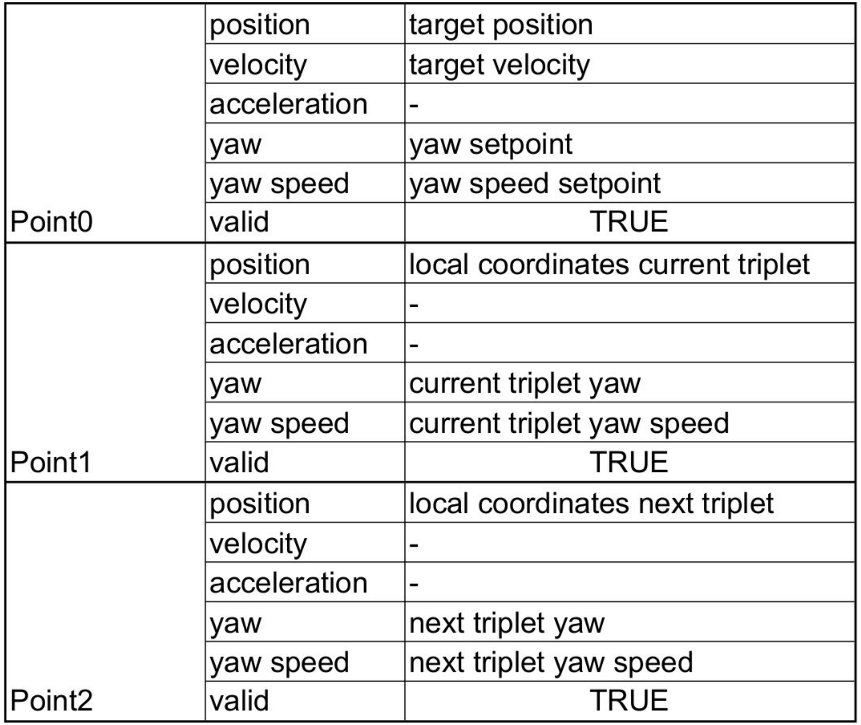
5 points, each can be described by:

- Position
- Velocity
- Acceleration
- Yaw
- Yaw Speed
- Valid flag



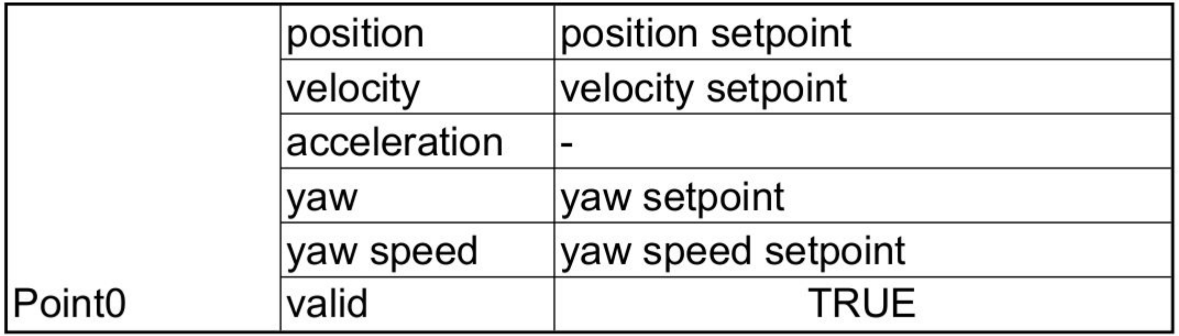
```
<message id="332" name="TRAJECTORY_REPRESENTATION_WAYPOINTS">
  <wip/>
  <field type="uint64_t" name="time_usec" units="us"> </field>
  <field type="uint8_t" name="valid_points"> </field>
  <field type="float[5]" name="pos_x" units="m"> </field>
  <field type="float[5]" name="pos_y" units="m"> </field>
  <field type="float[5]" name="pos_z" units="m"> </field>
  <field type="float[5]" name="vel_x" units="m/s"> </field>
  <field type="float[5]" name="vel_y" units="m/s"> </field>
  <field type="float[5]" name="vel_z" units="m/s"> </field>
  <field type="float[5]" name="acc_x" units="m/s/s"> </field>
  <field type="float[5]" name="acc_y" units="m/s/s"> </field>
  <field type="float[5]" name="acc_z" units="m/s/s"> </field>
  <field type="float[5]" name="pos_yaw" units="rad"> </field>
  <field type="float[5]" name="vel_yaw" units="rad/s"> </field>
</message>
```

From FCU to Companion Computer



Point0	position	target position
	velocity	target velocity
	acceleration	-
	yaw	yaw setpoint
	yaw speed	yaw speed setpoint
	valid	TRUE
Point1	position	local coordinates current triplet
	velocity	-
	acceleration	-
	yaw	current triplet yaw
	yaw speed	current triplet yaw speed
	valid	TRUE
Point2	position	local coordinates next triplet
	velocity	-
	acceleration	-
	yaw	next triplet yaw
	yaw speed	next triplet yaw speed
	valid	TRUE

From Companion Computer to FCU



Point0	position	position setpoint
	velocity	velocity setpoint
	acceleration	-
	yaw	yaw setpoint
	yaw speed	yaw speed setpoint
	valid	TRUE

What else?

- HEARTBEAT
  - Check healthiness of the Companion processes on the FCU side
  - Pre-flight checks to catch system failures as early as possible
- OBSTACLE DISTANCE
  - Send distance data from the Companion Computer to the FCU
  - Representation 360 degrees around the vehicle
  - Currently used for collision prevention

Work in Progress

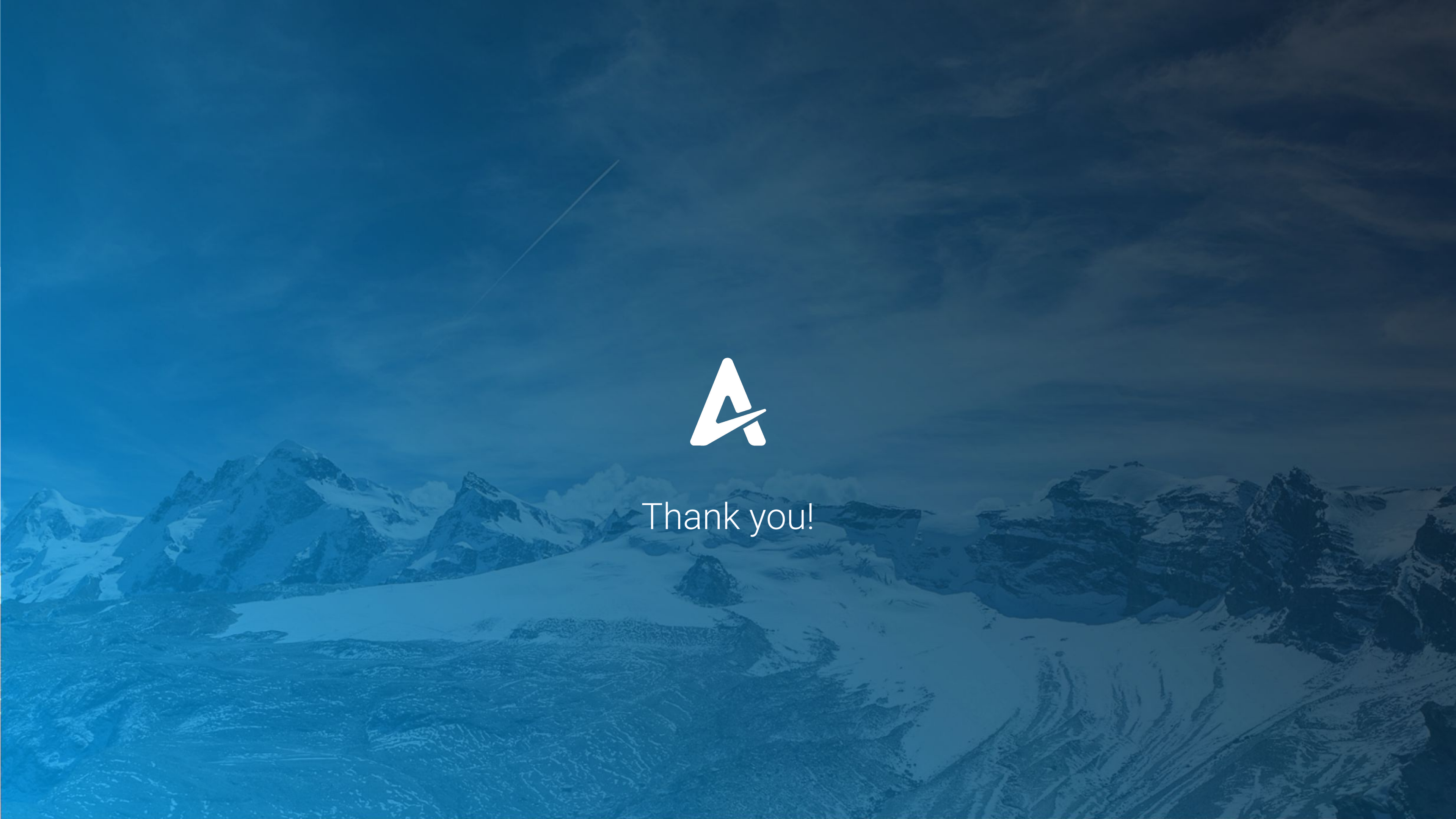
- Add type to each point to specify land, takeoff, loiter etc
- Use the OBSTACLE_DISTANCE message for Collision Prevention with distance sensors
- Seamless integration on all mission features, e.g. heading at waypoint, ROI
- Fast RTPS

Obstacle Avoidance - Local Planner

Safe Landing Planner

https://github.com/PX4/avoidance

Feedback



Thank you!