C Live View GUI

C.1 Compressed window

The live window of each drone automatically updates according to the set update rate (by default it is currently 2Hz). As this is only for the users benefit, it is not necessary to run these interfaces at the same operating frequency as other critical components. The GUI provides rounding of each numerical value to increase readability and allow the user to quickly scan the appropriate values.
Towards the bottom half of the interface other relevant readings and controls are included such as state, speed, link condition, battery level, emergency override, and land override. The state displayed is consistent with that accessible through the user API and can take one of five values. The speed slider provides multiplier speed adjustments to the specified drone, this value is only adjustable when the drone is in a 'LANDED' state. The user is able to adjust the multiplier to as low as 20% and and high as 2.0. The link condition variable communicates whether there are any communication breakdowns for the given drone. A break at any point in the communication flow will trigger this condition to change. The battery progress bar will monitor the battery level of the given drone if that value is available. The final two controls allow for override in a situation the user is able to notice something the platform is not and would like to trigger either an emergency request or a land request.

Using the live view application program, the user is available to launch the live

view GUI for each rigidbody on the drone server. These windows will spawn in a logical fashion and will space accordingly to maximise the number of windows that can be observed at any one time. In addition once overlap is inevitable, the windows are indented to allow the user to easily select lower nested windows, an example of this can be seen in figure C.1. Due to the lower operating frequency and the fact that only minor graphics are altered each iteration, many live view windows can remain open throughout platform operation.

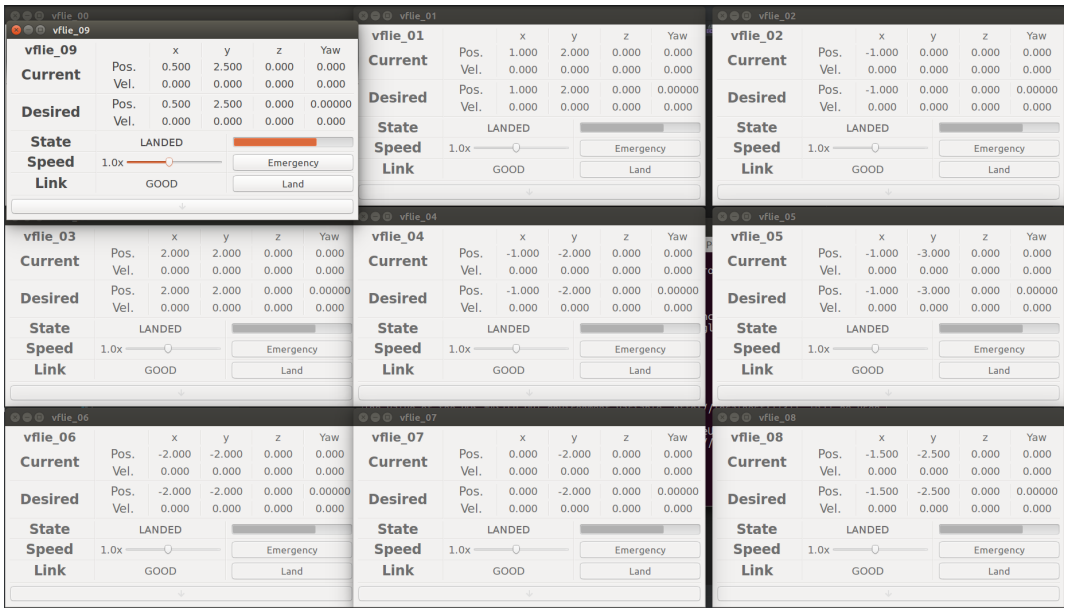

Figure C.1: *Screenshot of the overlapping Live View windows for each rigidbody on the drone server.*

C.2 Expanded window

The user is able to trigger an option for the live view window to launch in expanded mode, or the user is able to manually trigger it by selecting the button along the bottom of the basic live view window. The expanded mode can be seen below and adds two key features to the compressed window. These features

include drone specific logging output and safeguarding obstacle indicators.

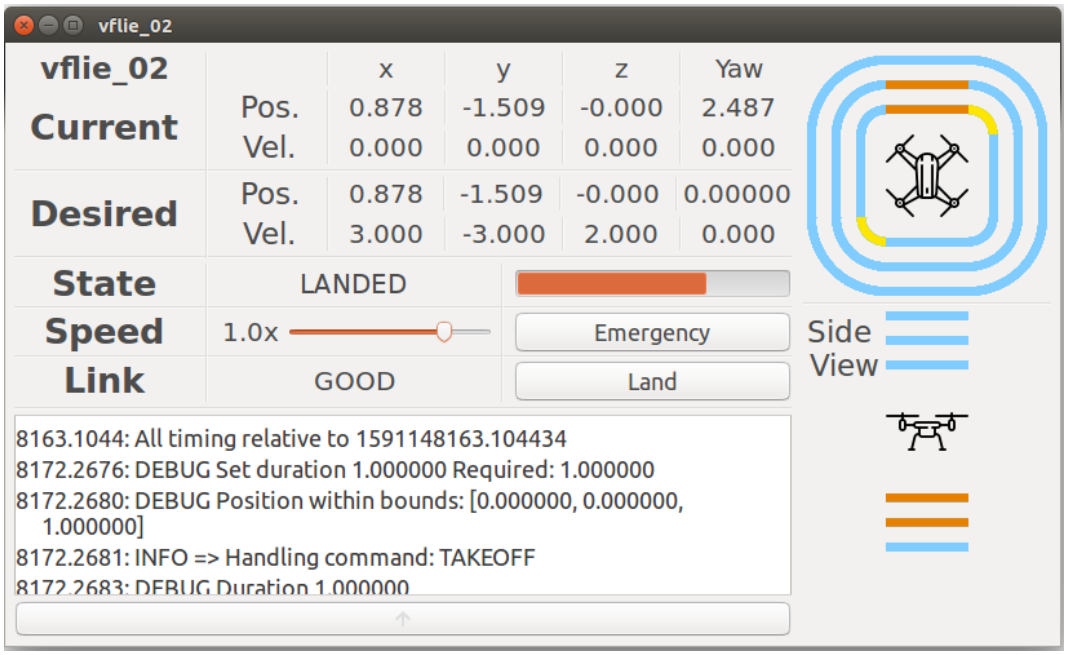

Figure C.2: *Screenshot of the Live View window expanded.*

The default color of a light blue indicates there is no obstacle in that vicinity. A yellow bar indicates an obstacle exists in that direction, and the obstacle is not within the influence distance of the collision avoidance algorithm. Orange indicates an obstacle is within influence distance prompting an increase in intensity (two bars). Red indicates an obstacle is in the drone's restricted distance and is indicated by three bars. These top and side view real-time diagrams provide an accurate visualisation of all obstacle influences on each drone. The dynamic diagrams use a number of drone dimension parameters provided through ROS parameters to decide which region an obstacle influences the given drone. The drone's orientation in the diagram is static and all obstacle influences are evaluated relative to the drone. The coordinate system adopted in the safeguarding visual feedback diagrams reflects the coordinate system of the physical Optitrack environment. In the top view diagram, the drone facing in the positive x axis and

the left side of the drone is in the negative y region. On the side view diagram, above the drone is positive z, whilst below the drone is negative z. A diagram has been provided in figure X below for clarity.

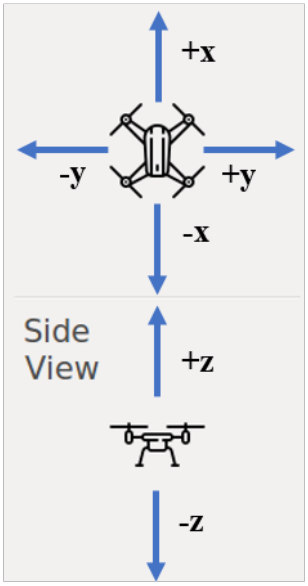

Figure C.3: *Relative coordinates for obstacle influences in live view window.*

If the expanded option is selected as default, each live view window will be spaced according to the expanded size of the window, as can be seen in figure C.4. Each live view window can be expanded and compressed as the user requires.

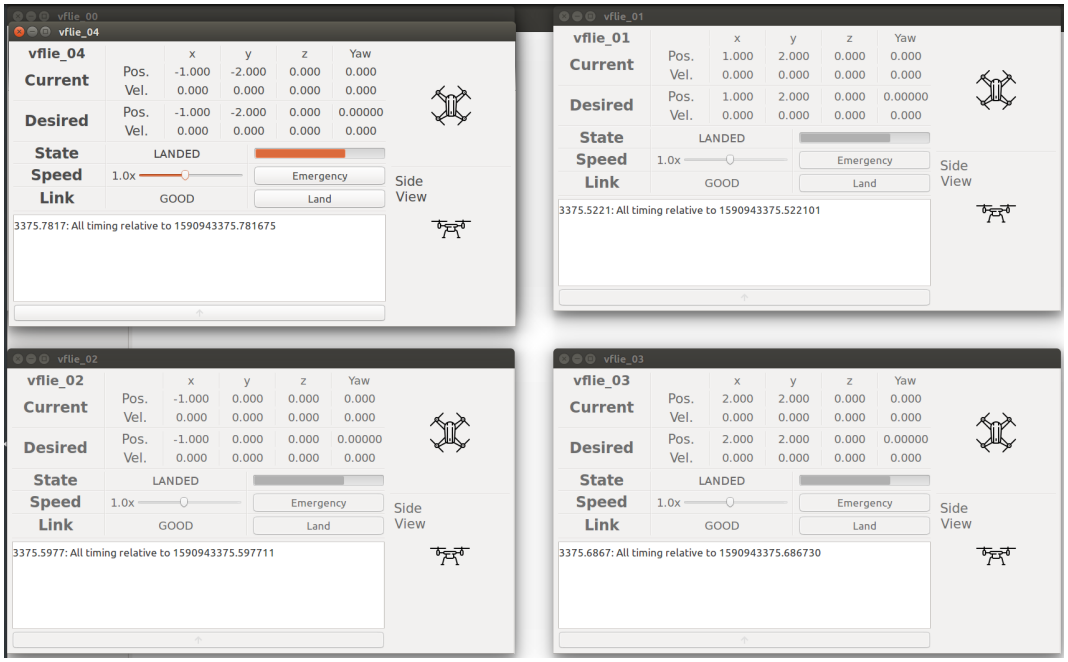

Figure C.4: *Screenshot of the Live View windows with expanded as default option.*