Flysight2 WScomp Mode

Introduction

WScomp Mode is used for Wingsuit flight navigation when competing in WS performance competitions. It is currently set up to follow FAI rules. 9 seconds after the exit was detected, a line between the current jumpers position and a ground reference point (GRP) is drawn. The jumper has to stay within a 600m corridor around this lane to avoid penalties (300m to each side). WScomp Mode is using chirpControl to navigate the jumper back towards the center line of the designated flight path (DFP).

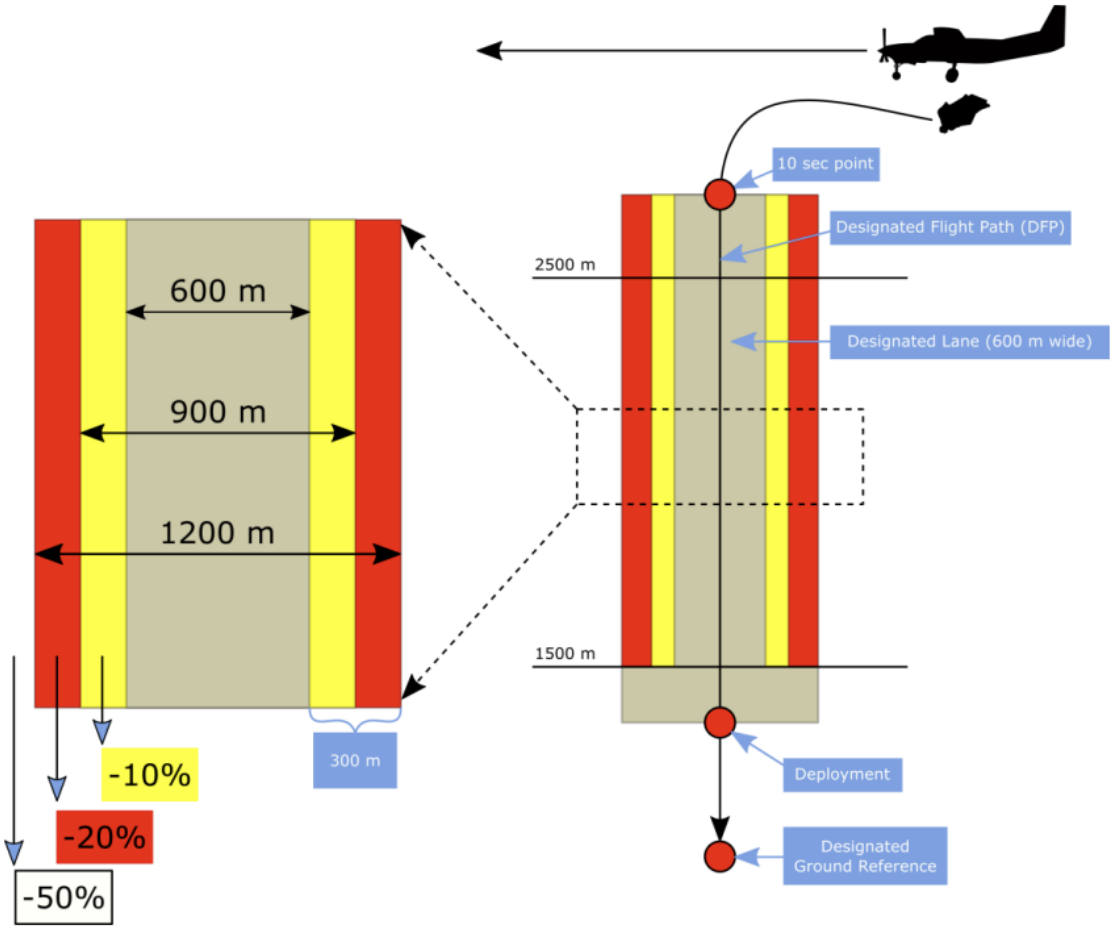


Figure 1 FAI Competition Rules Wingsuit Flying - ADDENDUM C – PERFORMANCE FLYING: DFP, DL, PENALTIES

How to program Flysight config file

There are multiple settings needed to make WScomp Mode work correctly:

WScomp_compwindow_top: [m] Top of the competition window (2500m for FAI)
WScomp_compwindow_bottom: [m] Bottom of the competition window (1500m for FAI)
WScomp_exit_max: [m] Maximum exit altitude for jump to count (3353m for FAI)
WScomp_exit_min: [m] Minimum exit altitude for jump to count (3200m for FAI)
WScomp_valwindow_speed [m/s] Vertical speed to be exceeded to start validation window (10m/s for FAI)

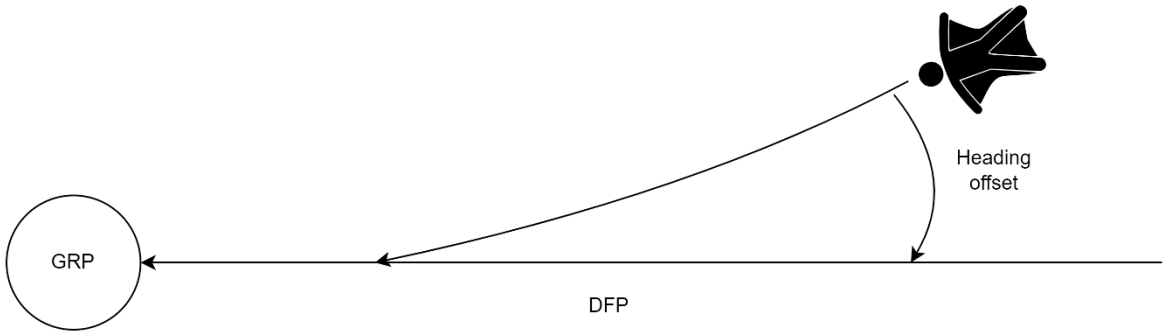
created by Max Krämer
If you have any suggestions on how to improve this code or how to improve this documentation, do not hesitate to send me a message to:
E-Mail: flysight@maze-engineering.com
Instagram: @maze_engineering

WScomp_valwindow_delay: [s] Delay between Exceeding vertical speed and actual start of validation window (9s for FAI).
WScomp_GRP_lat: [0.00001°] Ground Reference point Latitude. Used in combination with position at start of validation window to calculate designated flight path.
WScomp_GRP_lon: [0.00001°] Ground Reference point Longitude. Used in combination with position at start of validation window to calculate designated flight path.

How it works

In-Flight Experience

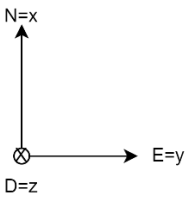
After takeoff, there will be no sounds (except for the config file start sound) until the minimum exit altitude is reached. After crossing this altitude, the sound “Approaching exit altitude” will be played. The code will wait for the vertical speed to exceed the value saved in WScomp_valwindow_speed. During this phase, the code will alert the jumper is the minimum or maximum exit altitude is exceeded by playing “too low” or “too high” repeatedly until the altitude is within the allowed range. Once the vertical speed is exceeded, the sound “exit” will play once. For the next 9 seconds, chirpControl will guide the jumper towards the GRP. After 9 seconds, the sound “alpha” will be played once, the current jumpers position is saved and the distance to the DFP is calculated. For every meter distance to the DFP, a heading offset of 0.1° towards the DPF is applied. For example, if the jumper is 100m away from the DFP, chirpControl will navigate the jumper towards the DFP with a heading difference of 10°. The maximum heading offset is limited to 20°. When the jumper crosses the upper boundary of the competition window, the sound “bravo” is played once. When the jumper crosses the bottom boundary of the competition window, the sound “charlie” is played once. After this, chirpControl stops to produce any sounds.



Code

Most of the code is documented inside the code using comments. Vector Operations will be explained in this section of the documentation.

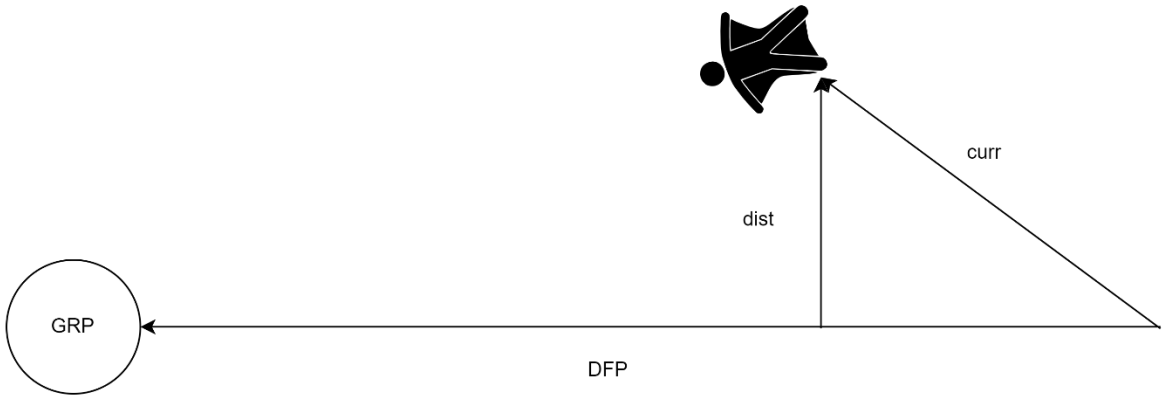
The NED (North-East-Down) coordination system is used for calculations:



created by Max Krämer
If you have any suggestions on how to improve this code or how to improve this documentation, do not hesitate to send me a message to:
E-Mail: flysight@maze-engineering.com
Instagram: @maze_engineering

All Headings are measured from the north axis and grow positive towards the east axis.

The following vectors are used to calculate nominal heading:



The vector “curr” is a vector from the start of the DFP to the wingsuiters current position.
The vector “DFP” is a vector from the start of the DFP in direction of the GRP with a length of 1m. (In the above image it is shown as the full DFP path. If we would draw above image according to DFP in the code, it would be very short.)
The vector “dist” is a vector pointing from a position on the DFP to the wingsuiters current position. It is always perpendicular to the DFP. The length of this vector is equal to the distance of the wingsuiter to the DFP.

To calculate the length of the vector “dist”, we can utilise the crossproduct of vector DFP and vector curr. Since the vector DFP has a length of 1, the result of the crossproduct will be the length of the vector dist. As we are using NED coordinates, we are using a left-hand coordination system. To get a positive result if the wingsuiter is to the right side of the DFP and a negative result if the wingsuiter is to the left side of the DFP, we have to use the following formula:

$$distance = \overrightarrow{curr}\, x\, \overrightarrow{DFP}$$

$$distance = DFP_x * curr_y - DFP_y * curr_x$$

Note: Currently the code is using a different approach by calculating the vector dist completely. This will be changed to a pure and more simpler crossproduct calculation soon.

created by Max Krämer
If you have any suggestions on how to improve this code or how to improve this documentation, do not hesitate to send me a message to:
E-Mail: flysight@maze-engineering.com
Instagram: @maze_engineering