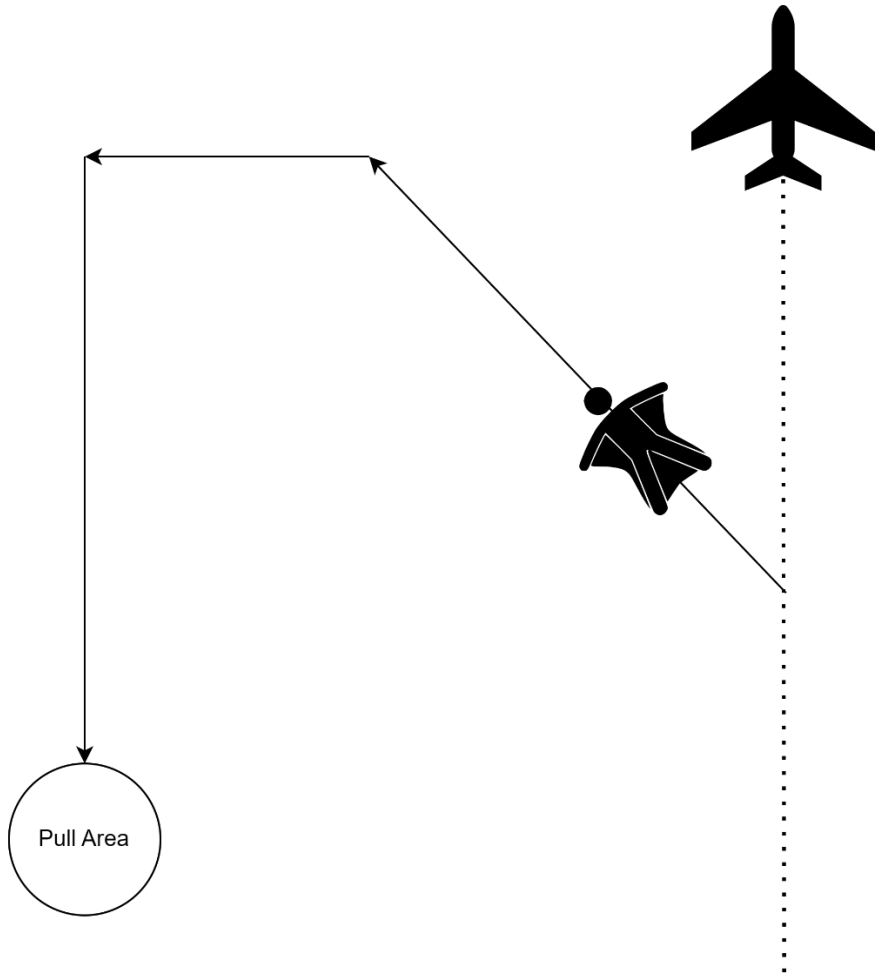
Flysight2 navMK Mode

Introduction

navMK Mode is used for Wingsuit flight navigation using 3 legs and a designated pull point. The following image shows 3 legs (315°, 270°, 180°) of a wingsuiters flying path when dropped on a jumprun exactly north.



How to program Flysight config file

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

navMK_Pull_Lat: Latitude of designated pull point (center of pull area)
navMK_Pull_Lon: Longitude of designated pull point (center of pull area)
navMK_Pull_Alt: Altitude of designated pull point
navMK_Leg1: Heading for first leg (315 in above image)
navMK_Leg2: Heading for second leg (270 in above image)
navMK_Leg3: Heading for third leg (180 in above image)
navMK_numLegs: Amount of legs to be used (if only 2 legs are used, leg1 is ignored. Fill out leg2 and leg3. If only one leg is used, use leg3.
navMK_Average_Glide_Ratio: Average glide ratio to be used for leg calculation

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

How it works

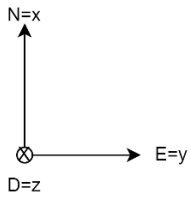
In-Flight Experience

After takeoff, there will be no sounds (except for the config file start sound) until an altitude of 3500m. After crossing this altitude, the sound “Approaching exit altitude” will be played. The code will start to calculate the three leg lengths to check if the pull point is reachable using the average glideratio. Once it is reachable, the sound “Pull Point reachable” will be played. If it is not reachable again, the sound “Pull Point out of reach” will be played. Once the exit is detected (downwards velocity higher than 75km/h), the sound “exit” will be played. After this, chirpControl will produce chirps to guide the jumper on the first legs heading. The code will constantly calculate all three leg lengths and notify the jumper with the “alpha” sound, once it is time to change to the next leg. ChirpControl will guide the jumper to the next leg after this. When it is time to change for the third leg, the sound “bravo” will be played. When the altitude is below the pull altitude, the sound “charlie” will be played. After this, chirpControl stops to produce any sounds.

Code

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

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



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

Three vectors which correspond to the different leg headings are used (L1, L2, L3). These vectors each have a length of 1m. The NED-coordinates from the jumpers current position to the designated pull point (P) can be calculated as followed:

$$\begin{matrix} P_x \\ P_y \\ P_z \end{matrix} = \begin{matrix} L1_x & L2_x & L3_x \\ L1_y & L2_y & L3_y \\ L1_z & L2_z & L3_z \end{matrix} * \begin{matrix} u \\ v \\ w \end{matrix}$$

u, v and w are the length of the corresponding legs L1, L2 and L3.

In our case, we know the value of P as well as the value of L1, L2 and L3. We need to find the values for u, v and w. To do this, we have to find the inverse of matrice L (which is then defined as matrice M in the code):

$$\widehat{L^{-1}} = \frac{adj(A)}{\det(A)} \widehat{=} \widehat{M}$$

With this, we can calculate the values for u,v and w:

$$\begin{matrix} u \\ v \\ w \end{matrix} = L^{-1} * \begin{matrix} P_x \\ P_y \\ P_z \end{matrix}$$

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

During the initialisation function (navMK_Init()), matrice L and matrice M are calculated.

The calculation of u, v and w is happening inside phase 1 and phase 2. If the headings are not dependend on each other (this happens if leg1 has the same heading as leg2), a solution of the equation should always be possible. u, v and w might take negative values though, so it's not a valid solution for a flight path. Only if all 3 vectors have positive distances, a valid flight path is found and the pull point is "reachable".

To give the jumper time to turn, a preturn alert of 10m per degree of turning is provided. So, for example, if the jumper has to turn 45°, the switching to the next leg will happen when there are 450m of flight path in leg1 left.

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