# Handbook creation of bird’s swarm trajectories for Digital Twin

[Setup the simulator on your computer 2](#_Toc175652891)

[Update of the command of the terminal and you python version 2](#_Toc175652892)

[Build the simualtor 3](#_Toc175652893)

[Parameters the Simulator 3](#_Toc175652894)

[Generate trajectories from raw data 3](#_Toc175652895)

[Split long trajectories into smaller one 4](#_Toc175652896)

[Convert the splited trajectories into .JSON 4](#_Toc175652897)

[Launch simulation using the simulator from ENAC 5](#_Toc175652898)

[Converting drones’ trajectories into digital twin files 5](#_Toc175652899)

[Creation of bird’s floc trajectories 7](#_Toc175652900)

[Multiplication of the birds 7](#_Toc175652901)

[Convert birds flock into digital twin tracks 7](#_Toc175652902)

[The Digital twin viewer 7](#_Toc175652903)

[General observations: 7](#_Toc175652904)

Notice : Every command / bash script need to be written in a Linux terminal or in a WSL terminal if your OS is Windows. Also if your computer is using Windows be sure to have [Visual C++ Build Tools](https://visualstudio.microsoft.com/fr/visual-cpp-build-tools/)

# Setup the simulator on your computer

## Update of the command of the terminal and you python version

Open a terminal and type the following command:

« sudo apt-get update »

« sudo apt-get install jq »

« sudo apt-get install bc  »

« sudo apt-get install python3-tk  »

“pip install pandas”

“pip install datetime”

“pip install basemap”

“pip install tkinter”

## Build the simualtor

Start by decompressing the archive file named “SimDrone6DOF-V1.0.0” be sure that the file is in the good repository (=Code\_clear/), be sure that when you click on the decompressed file you can see the 3 files (build, docs, src).

You can follow the tutorial in “Code\_clear/SimDrone6DOF-V1.0.0/src/drone\_sim\_6dof/READMDE”

Or move to this directory in the terminal “Code\_clear/SimDrone6DOF-V1.0.0/src/drone\_sim\_6dof” and then type

“python3 setup.py setup”

“python3 setup.py venv”

“source .venv/bin/activate”

“python3 setup.py build”

## Parameters the Simulator

After adding the file “flock\_personalized\_conversion.json” in the directory “file for the simulator” into the directory “Code\_clear/SimDrone6DOF-V1.0.0/src/drone\_sim\_6dof/examples/config”

Modify the file “Code\_clear/SimDrone6DOF-V1.0.0/src/drone\_sim\_6dof/examples/drone/conf.xml” to modify the speed of the drone, the boids force.

# Generate trajectories from raw data

The main goal of this part is to turn raw dataset collected on the internet and turn them into useful one. What I want to achieve after this step of processing is a standardized version of the data with only 4 rows (time, x, y, z). The x, y, z coordinates are in the absolute metric system, to turn them from the lat, long, height system into the classic one I used the “Basemap” library and the projection=”merc” and the time format is in this date format : "%Y-%m-%d %H:%M:%S.%f" . The output will be one file per trajectory in the raw trajectories’ files. The process that I used to is illustrated in the python file /Code\_clear/Script/ cutting\_raw\_data\_into\_trajectories.py”.

# Split long trajectories into smaller one

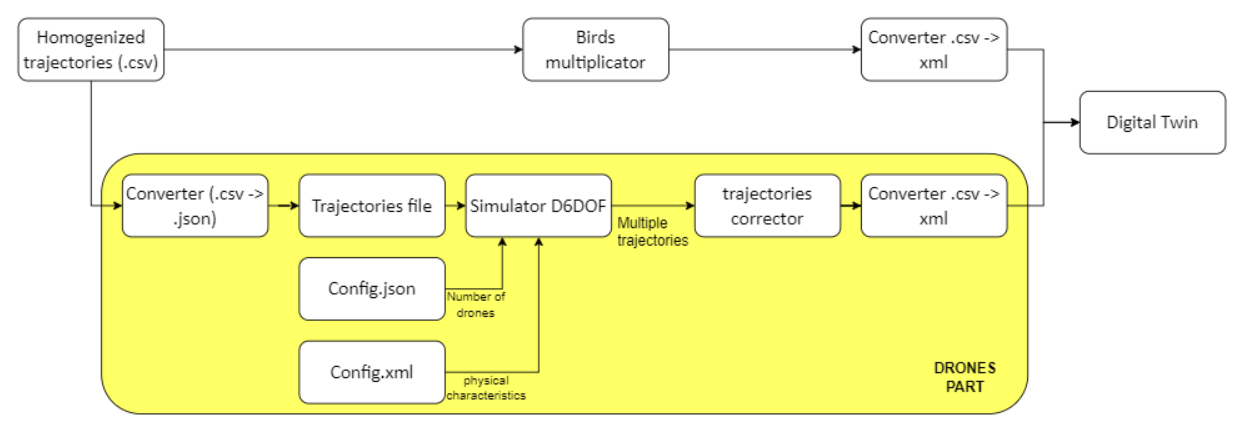
The goal of this part is to split the long trajectories previously generated into smaller trajectories of the same length to build a more homogeneous dataset. The file “Code\_clear/Script/ cutting\_trajectories\_into\_smaller.py” take as input the wanted length to modify it you must modify the value of “DISTANCE\_PER\_TRACK”.

The algorithm that is used to split the long trajectories is to compute the length of a trajectory, then if the trajectory is shorter than “DISTANCE\_PER\_TRACK”, the trajectory is skipped. If the trajectory is longer than “DISTANCE\_PER\_TRACK” the Euclidian division of the length of the track by “DISTANCE\_PER\_TRACK” is calculated, then I know the number of trajectory’s slice needed which will be these number + 1.

Also, if you are generating new tracks don’t forget to clear the previous generated tracks which a located in “/Dataset/Processed\_dataset/splitted\_trajectories”.

After generating these shorter trajectories, you can run “Script/ eliminate\_nonconforme\_traj.py” to delete every trajectory that are not in a specific wanted range, to modify this range you can change the parameters “MIN\_DISTANCE\_PER\_TRACK” and “MAX\_DISTANCE\_PER\_TRACK” those distances are in meter.

# Convert the splited trajectories into .JSON



The simulator from ENAC can’t take .csv file as input file for the trajectory, so you need to convert those files into .json files. The file “Code\_clear/Script/Converting\_csv\_into\_JSON.py” turn every .csv file into .csv the output trajectories are stored in “Code\_clear/Dataset/Processed\_dataset/drones/trajectories\_json\_format”

# Launch simulation using the simulator from ENAC

Before launching a simulation, be sure that the file config.xml and “flock\_personalized\_conversion.json” are the one you wanted to use. Also choose the .json file that you want to be processed by the simulator into the directory “Code\_clear/Dataset/Processed\_dataset/drones/trajectories\_json\_format/processing” if the directory “processing” does not already exist create it and then paste the trajectories you wanted. Open the bash script “Code\_clear/Script/ run\_drones\_simulation.sh” and modify the name of the parameter “SOURCE\_DIRECTORY” with the name of the origin of the files.

To run simulation place yourself in repository “Code\_clear/Script” and type in the terminal :

“./run\_drones\_simulation.sh”

Choose “yes” if you want .csv files as output and “no” if you want to see the animation of the drones flying.

Tips to run useful simulation

* Before running simulation and get .csv files, start by looking at the animation generation and see if your configuration of the different parameters does not end with problems such as drones to close from each other, a drones stay behind the group and turn around one point, drones flock bursting after a close turn
* Be sure that the drones follow the input path
* Be sure that the time length of the simulation is appropriate, not to short, be not too long otherwise when the drones reach the end of the waypoints, they will turn around the last one and the trajectory will become useless.

# Converting drones’ trajectories into digital twin files

The digital twin of the sea fire radar can only handle .xml file, so this script “Code\_clear/Script/ Convert\_drones\_trajectories\_into\_digital\_twin\_tracks.sh” aim to convert output trajectories from the simulator into .xml file.

To spare time of simulation, the script does not turn 1 initial trajectory into 1 digital twin trajectory. Multiple trajectories can be added on the same tracks. The script also increases the distance between each drones generated to increase the distance you can modify the parameters “dilatation-factor”. This factor will multiply the distance between each drone by this factor.

A diagram of a diagram

Description automatically generated

Parameters for the conversion of the output files from the ENAC simulator into digital twin trajectories

* list-path [list of string] : list of the path that you want to process (normally automatically handle by the script)
* speed [int] : speed that you want the drones to have after the modification (notice that you can also
* filter [int] : parameter that filter the number of point coming from the output file that will be happened to the end file
* helico [Boolean] : parameter that enable you to add the feature "helicoter" on the digital twin
* lenght-track [int] : parameter that enable to put more trajectories into the same digital twin simulation
* dilatation-factor [float] : parameter that enable to increase the between each drones by this factor
* output-path [String] : parameters that indicate where the generated trajectories should be placed
* tracks-number [int] : parameter that change the name of the output file (normally totally handle by the script)
* (- offset [list of int] : parameters to modify the starting point of your trajectories)

Tips :

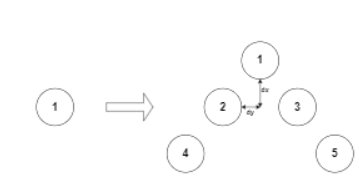
* weight of the output file, if it’s too big you can either increase the filter number or reduce the number of trajectories per output file. (don’t be afraid of using huge numbers for the filter parameters because the time step of file from the simulator is around 0.07s can be modified in the file “flock\_personalized\_conversion.json”
* Because you can’t directly fixe the distance between the drones with the dilatation-factor, at the beginning you need to try different value and see in the console if it’s result to the wanted interdrones distance.

# Creation of bird’s floc trajectories

## Multiplication of the birds

To turn single bird trajectory in a multiple use the bash script “Code\_clear/Script/ Multiplate\_birds\_script.sh”

This script assigns a random number within the specified lower and upper limits to each trajectory, with birds following a V-shaped formation along the path. The birds are added to the following the V-pattern with the distance of (dx, dy) provided.



## Convert birds flock into digital twin tracks

This script “Code\_clear/Script/Convert\_birds\_flock\_into\_digital\_twin\_tracks.sh” convert the generated tracks digital twin trajectory. This script works in the same way as the script to turn drones’ trajectories into digital twin trajectories.

# The Digital twin viewer

To be able to visualize the created trajectories without using the digital twin, I developed software to see the track. It is located in "Code\_clear/Tools/see\_graph". You just need to run the file "main\_plot.py", click on the button "Browse Files", search for the trajectories you want, click on the button "Parse the data", and wait a little (normally, you should see "Data analyzed" appear in the console). After that, click on the button "Plot graph".

# General observations:

* A huge problem that can appear is that the generated trajectories is too big for the digital twin and could not be loaded by the digital twin.
* The dataset Vulture which only contains 1 long trajectory led to a huge number of sliced trajectories than the other, the weight of this dataset should be consider for the future works
* Another thing related to the length of this trajectory is the distance between each point who compose the trajectory, some point has a huge distance between each other, that leads to sliced trajectory with around 20 points when the wanted distance of the track is around 5 km. Some tracks have only 3 points. Need to observe how it will influence the tracks on the digital twin.
* To solve this problem a solution could be to add a script that deletes every trajectory that is too short.
* The current trajectories in the directory “Dataset/Splited\_trajectories” are less than it should be because some files were corrupted, so it will be mandatory to recreate them. Although, this step is also mandatory if we want to be able to increase the dataset. The current code aims to turn some of the files that are in the raw\_dataset. The purpose of this step is to have in the end a .csv file with only 4 rows (x, y, z) in meters and in the absolute coordinate system and the date format: "%Y-%m-%d %H:%M:%S.%f".