

## TP1-1 — Analyse des messages ADS-B (MSG)

### Objectifs du TP

À l'issue de ce TP, l'étudiant sera capable de :

- Comprendre la structure des messages **ADS-B (type MSG)**.
- Charger, filtrer et nettoyer un dataset de messages MSG.
- Extraire des paramètres clés : position, altitude, vitesse, cap, horodatage.
- Reconstruire la trajectoire d'un aéronef.
- Visualiser la trajectoire sur une carte simple.
- Détecter des **comportements anormaux** : déviation de cap, descente brusque, hippodrome, entrée en zone interdite.
- Effectuer une première analyse comportementale d'un aéronef civil ou commercial.

### Données fournies

Vous disposez d'un fichier : « dataset\_msg\_adsb.csv »

Contenant uniquement des lignes de type **MSG** au format BaseStation SBS-1, par exemple :

```
MSG,3,,3C656C,,2025/10/13,08:29:50.342,2025/10/13,08:29:50.342,D-  
AIKL,5021,262.0,113.0,49.95295,8.05448,,,0,0,0,0
```

Champs typiques disponibles :

- ICAO (identifiant avion en hexadécimal)
- Immatriculation
- Altitude (ft)
- Vitesse (kt)
- Cap (°)
- Latitude
- Longitude
- Horodatage (date + heure)

Contient des zones circulaires ou polygonales représentant :

- aéroports
- zones militaires
- centrales
- espaces interdits

## Librairies Python recommandées

- pandas
- matplotlib
- numpy
- folium
- shapely
- datetime

## Partie 0 – Génération d'un dataset à partir du serveur ADS-B « [sbs.glidernet.org](https://sbs.glidernet.org) »

Dans cette partie, vous allez :

1. comprendre le fonctionnement du script Python fourni (`test.py`),
2. le tester en conditions réelles,
3. générer un fichier CSV local contenant les messages ADS-B (type MSG),
4. préparer un dataset exploitable pour la suite du TP.

## Partie 1 : Lecture & Pré-Analyse des messages MSG

### 1.1 — Charger le fichier

Lire le fichier CSV contenant les lignes MSG et afficher :

- `df.head(10)`
- `df.info()`
- `df.describe()`

### 1.2 — Extraire les colonnes utiles

À partir des lignes SBS-1, extraire :

Colonne	Description
<code>icao</code>	identifiant avion
<code>registration</code>	immatriculation
<code>timestamp</code>	date + heure du message
<code>lat</code>	latitude
<code>lon</code>	longitude
<code>altitude</code>	altitude en pieds
<code>velocity</code>	vitesse en nœuds
<code>heading</code>	cap

## 1.3 — Questions

- Combien d'avions distincts (`icao`) dans le dataset ?
- Quelle est la plage temporelle des données ?
- Quelle est l'altitude moyenne ? vitesse moyenne ?

## Partie 2 : Reconstruction d'une trajectoire

### 2.1 — Sélectionner un avion

Choisir un ICAO présent suffisamment longtemps (ex : plus de 50 points).

### 2.2 — Trier les points dans l'ordre temporel

- Convertir `timestamp` en `datetime`
- Trier les points par ordre chronologique

### 2.3 — Visualiser la trajectoire

Tracer :

- longitude en abscisse
- latitude en ordonnée
- couleur = altitude

Indications :

- marquer le **premier point** en *vert*
- marquer le **dernier point** en *rouge*

### 2.4 — Questions

- La trajectoire semble-elle linéaire, courbée, chaotique ?
- L'avion monte-t-il ou descend-il ?
- Le cap (`heading`) évolue-t-il de manière cohérente ?

## Partie 3 : Détection d'anomalies de trajectoire

Dans le cadre de la surveillance aérienne, plusieurs comportements peuvent être suspects :

- Déviation brutale de cap
- Descente rapide (possible détournement, panne)
- Hippodrome (boucles)
- Entrée dans une zone sensible
- Changement soudain de vitesse

Vous allez développer des métriques simples.

### 3.1 — Déviation de cap anormale

Définir un seuil :

déviaton > 30° en moins de 20 secondes → suspect

Questions :

- L'avion analysé présente-t-il de telles déviations ?
- À quel moment ?

### 3.2 — Descente ou montée brusque

Calculer la variation d'altitude entre deux points consécutifs :

$\text{rate} = (\text{altitude}(i) - \text{altitude}(i-1)) / \Delta t$

Définir :

- 2500 ft/min = montée rapide
- < -3000 ft/min = descente dangereuse

### 3.3 — Détection de motif “hippodrome”

Repérer si l'avion :

- tourne plusieurs fois en rond
- conserve une vitesse stable
- garde une altitude stable

Exemples d'usages :

- surveillance militaire
- attente ATC
- mission d'observation

### 3.4 — Zones sensibles

À partir de `zones_sensibles.json` :

- Charger les cercles ou polygones
- Vérifier si un avion passe à **moins de 20 km** d'une zone
- Lister les ICAO concernés

## Partie 4 : Synthèse



Rédiger un court rapport (2 pages max) :

- Statistiques générales
- Choix de l'avion étudié
- Visualisation de la trajectoire
- Détection d'anomalies
- Interprétation :
  - écart météo ?
  - panne ?
  - trajectoire normale IFR ?
  - comportement suspect ?

## Partie 5 :

- Créer une carte interactive Folium avec les points de trajectoire
- Coloration dynamique vitesse/altitude
- Heatmap des zones de forte activité
- Clustering des comportements anormaux

## Livrables

- Jupyter Notebook (.ipynb)
- Fichier CSV annoté
- Rapport PDF (synthèse & Figures de trajectoires)