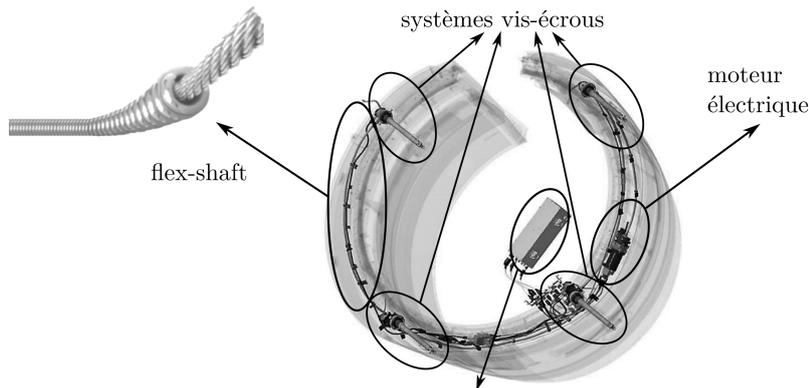


## Annexe 1 - Décomposition des éléments du système eTras<sup>®</sup>



TRCU (Thrust Reverser Control Unit)

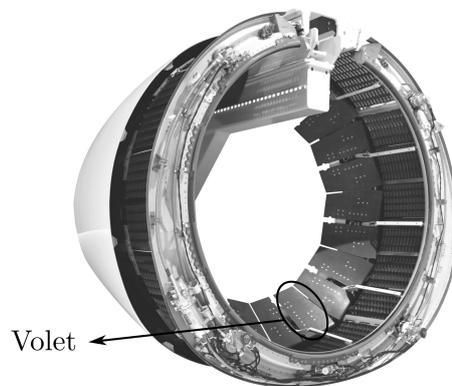
FIGURE 22 – Principaux constituants de la chaîne de puissance du eTras<sup>®</sup>.

FIGURE 23 – Vue des volets dans la nacelle.

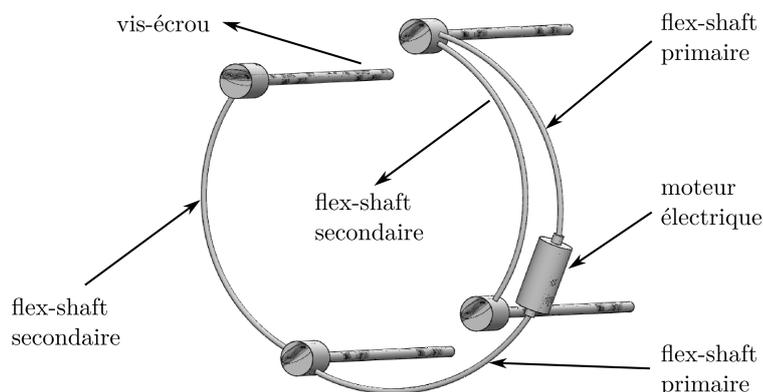


FIGURE 24 – Illustration de l'architecture des flex-shafts.

Comme le montre la Figure 24, le moteur électrique est raccordé aux flex-shafts primaires qui à leur tour sont raccordés en série via des renvois d'angle situés au niveau des premiers systèmes vis-écrou aux flex-shafts secondaires. Chaque système vis-écrou est mis en mouvement par l'intermédiaire d'un ou deux flex-shafts et d'un renvoi d'angle.

## Annexe 2 - Bibliothèque Scikit-Learn Python

Scikit-Learn est une bibliothèque Python spécifiquement conçue pour l'IA et le machine learning (ML). Ses principales fonctionnalités incluent la classification, le pré-traitement et la modélisation des données ainsi que la sélection de modèles. Elle fournit également une gamme d'algorithmes ML supervisés et non supervisés. Comme exemples de tâches ML utilisant Scikit-Learn, peuvent être citées la reconnaissance d'images, la prédiction ou la détection de spams. En outre, Scikit-Learn est interopérable avec la pile SciPy qui inclut NumPy, Matplotlib, IPython, SymPy et Pandas.

### Apprentissage supervisé

- la régression linéaire : **from sklearn.linear\_model import LinearRegression**

LinearRegression fits a linear model with coefficients  $w = (w_1, \dots, w_p)$  to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

```
1 import numpy
2 from sklearn.linear_model import LinearRegression
3
4 ## Exemple de fonction à linéariser :  $y = 1 * x_0 + 2 * x_1 + 3$ 
5 X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
6 y = np.dot(X, np.array([1, 2])) + 3
7
8 reg = LinearRegression().fit(X, y)
9 reg.score(X, y)
10 >>> 1.0
11
12 reg.coef_
13 >>> array([1., 2.])
14
15 reg.intercept_
16 >>> 3.0
17
18 reg.predict(np.array([[3, 5]]))
19 >>> array([16.])
```