

探査機やロケットなど、空間を自由に飛行する物体の位置と姿勢を表す数学として、ベクトル、行列を取り上げます。特に3次元の回転行列を応用することで

機体がどちらを向いているのかを表現することができます。

4-1 飛行体の位置表現 (基礎)

● 概要

ベクトルは大きさと向きを持った量で、数を1列に並べることで表現することができます。飛行体と地球、太陽など、物体と物体の相対的な位置関係を知りたいときにベクトル同士の計算を行うことで、これを求めることができます。

● 仕組み

▶ 空間内の相対的位置をベクトルを使って表す

3次元空間内で太陽を原点とし、地球と探査機が位置する場所を示す位置ベクトルをそれぞれ E 、 P とします。例えば、

$$E = (1, 0, 0), P = (1, 1, 0) \dots \dots \dots (1)$$

とすると、地球から探査機に向かうベクトル c は、

$$c = P - E = (0, 1, 0) \dots \dots \dots (2)$$

となり、地球から見て第2成分の方向に探査機がいることが分かります。

次に、探査機の運用上重要な、太陽-探査機-地球のなす角 θ [SPE角(Sun-Probe-Earth angle)と呼ぶ]を計算します(図1)。 P と c の内積は θ を使うと次式で表せます。

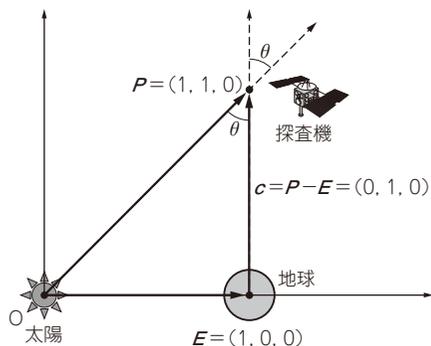


図1 太陽を原点とったときの太陽-探査機-地球のなす角 θ

$$P \cdot c = |P| |c| \cos \theta \dots \dots \dots (3)$$

$$\text{式 (3) を変形して, } \cos \theta = \frac{P \cdot c}{|P| |c|} = \frac{1}{\sqrt{2}}$$

従って、 $\theta = 45^\circ$ と求まります。

● コード

リスト1に先に示したベクトルの定義、差、内積の計算を示します。このように、ライブラリNumPyを使うとベクトル同士の基本的な計算ができます。

リスト1 ベクトルで飛行体の位置を表す

```
import numpy as np #numpyモジュールを名前npでimport

#ベクトルの定義
vec_E = np.array([1, 0, 0])
vec_P = np.array([1, 1, 0])
print("vec_E = ", vec_E)
print("vec_P = ", vec_P)

#ベクトルの差
vec_c = vec_P - vec_E
print("vec_c = ", vec_c)

#ベクトルのノルム
d = np.linalg.norm(vec_c)
print("d = ", d)

#ベクトルの内積
e1 = vec_P @ vec_c #演算子"@"を使った内積計算
e2 = np.dot(vec_P, vec_c) #numpy.dotを使った内積計算
e3 = np.inner(vec_P, vec_c) #numpy.innerを使った内積計算
print("vec_P \cdot vec_c = ", e1)
print("vec_P \cdot vec_c = ", e2)
print("vec_P \cdot vec_c = ", e3) #出力は全て同じ

#余弦cos(theta)
cos_theta = e1 / (np.linalg.norm(vec_P) * np.linalg.
                 norm(vec_c))

print("cos(theta) = ", cos_theta)
#逆余弦arccos(theta)
theta_in_rad = np.arccos(cos_theta) #角度はラジアンで求まる
theta_in_deg = np.degrees(theta_in_rad) #numpy.degreesで度に変換
print("SPE angle theta = ", theta_in_deg)
```