

SC651 : Assignment 3

Adityaya Dhande 210070005

TRIAD, QUEST and Wahba's algorithms

I rectified an oversight that I had made in Assignment 2, of not normalising the observed and reference vectors before fitting an orthogonal transformation between them. The solution obtained from Wahba's algorithm is

$$R_1 = \begin{bmatrix} 0.9962 & -0.0324 & -0.0812 \\ 0.0401 & 0.9947 & 0.0952 \\ 0.0777 & -0.0981 & 0.9921 \end{bmatrix}$$

and the corresponding mean squared error(between v and w) is **0.3272**

TRIAD

I implemented TRIAD using the first two rows of data. As TRIAD is not symmetric in its arguments, I implemented TRIAD for both the cases (1st row as argument 1 and 2nd row as argument 2 and vice-versa), and chose the one that gives smaller mean squared error. The estimated rotation matrix is

$$R_2 = \begin{bmatrix} 0.0057 & 0.9899 & 0.1415 \\ -0.3439 & -0.1310 & 0.9298 \\ 0.9390 & -0.0539 & 0.3397 \end{bmatrix}$$

and the corresponding mean squared error is **0.6677**. The following is the code for TRIAD.

```
1  r1 = v1 / np.linalg.norm(v1)
2  r2 = np.cross(v1, v2)
3  r2 = r2 / np.linalg.norm(r2)
4  r3 = np.cross(r1, r2)
5
6  s1 = w1 / np.linalg.norm(w1)
7  s2 = np.cross(w1, w2)
8  s2 = s2 / np.linalg.norm(s2)
9  s3 = np.cross(s1, s2)
10
11 s1 = s1.reshape((-1, 1))
12 s2 = s2.reshape((-1, 1))
13 s3 = s3.reshape((-1, 1))
14
15 r1 = r1.reshape((-1, 1))
16 r2 = r2.reshape((-1, 1))
17 r3 = r3.reshape((-1, 1))
18
19 Mobs = np.hstack((s1, s2, s3))
20 Mref = np.hstack((r1, r2, r3))
21 A = Mobs @ Mref.T
```

QUEST

I used `numpy.linalg.eig` to get the largest eigenvalue and the corresponding eigenvector of the K matrix. The estimated rotation matrix is

$$R_3 = \begin{bmatrix} 1.0000 & -0.0277 & -0.0709 \\ 0.0448 & 1.0000 & 0.1052 \\ 0.0880 & -0.0881 & 1.0000 \end{bmatrix}$$

and the corresponding mean squared error is **0.3252**. The following is the code for QUEST.

```
1  sigma = np.trace(B)
2  S = (B + B.T)
3  Z = np.zeros(3)
4  for i in range(len(DATA)) :
5      w = DATA[i][1]
6      v = DATA[i][0]
7      Z = Z + np.cross(w , v)
8  K = np.zeros((4,4))
9  K[:3, :3] = S - sigma * np.eye(3)
10 K[3,3] = sigma
11 K[3,:3] = Z
12 K[:3,3] = Z
13
14 q = np.linalg.eig(K)[1][:,3]
```

Comparison

The estimate from TRIAD is far worse as compared to QUEST and Wahba's algorithm. This is as expected as QUEST and Wahba's algorithm are optimum solutions and TRIAD is only a computationally simple algorithm which does not use the entire information available. More over only 2 out of 6 data rows were used by TRIAD. Another observation that is as expected is the similarity in the results of QUEST and Wahba's algorithm. Both these algorithms find optimum solutions to the same cost function. QUEST uses a quaternion to find the minimizer where as Wahba's algorithm directly gives the optimum rotation matrix. The slight difference between the matrices R_1 and R_3 is due to limited computer precision, which differs slightly on going from the quaternion to a rotation matrix.

The mean squared errors summarised :

1. Wahba's algorithm : 0.3272
2. QUEST : 0.3252
3. TRIAD : 0.6677