

MATLAB によるクォータニオン数値計算

宍戸 幹夫*
Mikio Shishido

クォータニオンは、航空機などでよく用いられるオイラー角のように角速度ベクトルとの関係が非線形で特異点をもつようなことはないため、飛行数値シミュレーションでは有用な回転表現である。一般に、オイラー角の微分方程式を解く数値シミュレーションにおいては、特異点に落ち込んでしまい、数値解が発散してしまう。

クォータニオンについては昨年度技報で「クォータニオン計算便利ノート」として鎌倉事業部矢田部より報告がなされている。本書では、この資料をもとにクォータニオン表現による剛体の運動、およびオイラー表現による剛体の運動をそれぞれ実際にMATLABで実装、数値計算を行い、両回転表現の比較について報告する。

The quaternion is a useful expression for rotation used in flight simulation. The relation between the quaternion and its angular rate is linear and the quaternion doesn't have a singular point like Euler angles. Euler angles are often used to describe aircraft dynamics.

Mr. Yatabe's paper "Convenient note for calculating Quaternion" appeared in last year's MSS Technical Report. In his paper, Mr. Yatabe summarized useful equations for attitude calculation using the quaternion. My paper compares quaternion and Euler angle expressions for rigid body dynamics. MATLAB is used to formulate and simulate rigid body dynamics as follow :

1. まえがき

力学では、位置や姿勢角を時間微分することにより速度、角速度を得られるとは限らない。これは、位置や姿勢角を定義する座標系の移動、回転を考慮して時間微分を行わなければならないためである。このようなことを避けるため、位置ベクトルについては、座標系の移動、回転を考慮しなくて良い慣性空間上で定義されることが多い。

しかし、姿勢角についてはこのようにはならず、航空機などで用いられるオイラー角では角速度ベクトルとの関係が非線形で特異点をもつ。そのため、オイラー角表現による飛行運動シミュレーションにおいては、上記特異点近傍で姿勢角が発散してしまう。

オイラー角に変わる回転表現としてクォータニオンがある。クォータニオンは、回転軸と回転角よりなる4成分で3次元空間の回転を表現する。クォータニオンでは、オイラー角による回転表現で生じるような特異点が存在しない。そのため、航空機等の姿勢角表現に有効な方法である。

そこで、本報告では、クォータニオン表現による回転表現を実際に数値計算が行えるようMATLABで実装および計算結果の確認を行い、そのプログラムソースコードを記載する。本報告による実装は、飛行機、ロケット、人工衛星等の運動シミュレーションや慣性装置、ジンバル機構を有するハードウェア等のモデリングとシミュレーションに適用できると考えられる。

本報告は、まず取り扱う座標系と方向余弦行列の定義を行い、次に、オイラー角、クォータニオンの定式化について述べる。この定式化をもとに数値解析ツールであるMATLABで実装を行った後に数値計算結果を示す。

2. 座標系と方向余弦行列の定義

本資料では、以下の2つの座標系を取り扱う。

- ・ 慣性座標系
- ・ 機体座標系

慣性座標系は、慣性空間に固定された座標系である。

機体座標系は、航空機等の機体の重心位置に固定され、慣性空間を移動、回転する座標系である。

両座標系とも右手系の直交座標系とする。機体座標系

では、x軸を機体前方にとり、z軸を下方にとることとする。機体座標系を図1に示す。航空力学では一般に機体座標系のx軸をロール軸、y軸をピッチ軸、z軸をヨー軸と呼ぶ。

方向余弦行列とは、任意の2つの座標系の関係を変換する行列である。本資料では、慣性座標系から機体座標系への変換行列を方向余弦行列 \mathbf{E} とする。ここで、慣性座標系の基底ベクトルを $[x_i \ y_i \ z_i]^T$ 、機体座標系の基底ベクトルを $[x_b \ y_b \ z_b]^T$ とすると、以下のような関係で表される。

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \mathbf{E} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (1)$$

3. オイラー角の定式化

オイラー角は各軸まわりの回転角の組として定義され、方向余弦行列と同様に座標系の関係を表現する要素である。

方向余弦行列が9つのパラメータからなるため剛体の回転を直感的に理解することは難しいが、オイラー角は3軸の回転角の組であるため回転を理解することは容易である。

3.1 オイラー角表現による方向余弦行列

慣性座標をz, y, x軸の順でそれぞれ ψ , θ , ϕ 回転させたとき機体座標となる角をオイラー角とする。航空力学ではx, y, z軸まわりの回転角をロール角 ϕ 、ピッチ角 θ 、ヨー角 ψ と呼ぶ。

ここで、方向余弦行列 \mathbf{E} はオイラー角を用い以下のように表わされる。

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (2)$$

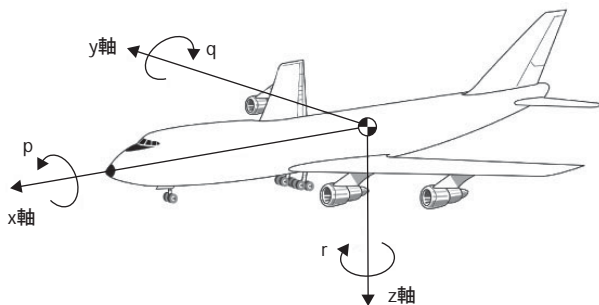


図1 機体座標系の定義

3.2 方向余弦行列からオイラー角表現

式(2)より以下のように方向余弦行列を用いてオイラー角を定義することができる。

$$\phi = \tan^{-1} \frac{E_{23}}{E_{33}} \quad (3a)$$

$$\theta = \tan^{-1} \frac{-E_{13}}{\sqrt{E_{23}^2 + E_{33}^2}} \quad (3b)$$

$$\psi = \tan^{-1} \frac{E_{12}}{E_{11}} \quad (3c)$$

3.3 オイラー角の微分方程式

機体座標系での各軸の角速度をそれぞれ p , q , r とすると、式(2)よりオイラー角の運動方程式は以下のように表わされる。

$$\frac{d}{dt} \phi = p + (q \sin \phi + r \cos \phi) \tan \theta \quad (4a)$$

$$\frac{d}{dt} \theta = q \cos \phi - r \sin \phi \quad (4b)$$

$$\frac{d}{dt} \psi = (q \sin \phi + r \cos \phi) \sec \theta \quad (4c)$$

上式のとおり、オイラー角の微分方程式はピッチ角 θ の±90degに特異点が存在する。これが、オイラー角表現による姿勢角計算の欠点である。オイラー角による姿勢角計算を行う際は、特異点をさけるため、回転順序を変えてオイラー角を定義する等の工夫が必要である。

4. クォータニオンの定式化

クォータニオンは、回転軸と回転角よりなる4成分で3次元空間の回転を表現する。クォータニオンでは、オイラー角による回転表現で生じるような特異点が存在しない。そのため、飛行機等の姿勢角表現に有効な方法である。

クォータニオンについて、「クォータニオン計算便利ノート」³⁾を参考に述べる。

4.1 クォータニオンの定義

クォータニオン \mathbf{q} は、回転方向の単位ベクトル \mathbf{n} と回転角 θ の4成分で定義される。クォータニオンを図2に示す。

$$\mathbf{q} = \cos \frac{\theta}{2} + \mathbf{n} \sin \frac{\theta}{2} \quad (5)$$

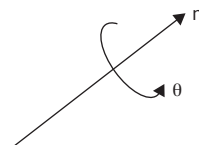


図2 クォータニオン

ここで、クォータニオンの定義から明らかなおりノルムは1である。

$$\sum_{i=1}^4 q_i = 1 \quad (6)$$

4.2 クォータニオン表現による方向余弦行列

クォータニオン表現による慣性座標系から機体座標系への方向余弦行列 \mathbf{E} は、以下のとおりである。

$$\mathbf{E} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (7)$$

4.3 方向余弦行列からクォータニオン表現

方向余弦行列からクォータニオンを求めるには、式(7)を用いる。まず、方向余弦行列の対角項からクォータニオンの1変数を求め、次に、求めた変数を用いてその他の変数を求める。対角項から q_4 を求めたときは、以下のとおりとなる。

$$q_4 = \pm \frac{1}{2} \sqrt{1 + E_{11} + E_{22} + E_{33}} \quad (8a)$$

$$q_1 = \frac{1}{4q_4} (E_{23} - E_{32}) \quad (8b)$$

$$q_2 = \frac{1}{4q_4} (E_{31} - E_{13}) \quad (8c)$$

$$q_3 = \frac{1}{4q_4} (E_{12} - E_{21}) \quad (8d)$$

ただし、方向余弦行列の対角項からクォータニオンの1変数を選ぶ際は、変数がゼロにならないようなものに行なければならない。

4.4 クォータニオンの微分方程式

クォータニオンの時間微分は、角速度ベクトル $\boldsymbol{\omega} = [p \ q \ r]^T$ を用いて以下のように表される。

$$\frac{d}{dt} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & r & -q & p \\ -r & 0 & p & q \\ q & -p & 0 & r \\ -p & -q & -r & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (9)$$

上式のとおり、クォータニオンの微分方程式には、オイラー角では式(4)のように存在する特異点を持たない。

5. MATLABによる実装

5.1 MATLABとは

MATLABは、データ解析、モデリング、シミュレーション、可視化などの数値解析、アルゴリズム開発に必

要な統合開発環境を提供するアプリケーションソフトウェアである。さらに、制御系設計、信号処理、画像処理など各種分野に対して利便性の高い数値演算ライブラリがToolboxと呼ばれるMATLABを拡張するパッケージの形で提供されている。また、プログラミング言語として、効率的な数値解析プログラム開発を行う事も可能である。

MATLABの特徴を以下に簡単に述べる。

- ・データアクセスの容易性

データの基本構造は行列であり、変数の型宣言や配列の大きさを宣言する必要がない。また行列を直接演算する事ができ、行列の1要素ごと演算する必要がない。

- ・対話型操作環境とプログラミング機能

統合開発環境により、コマンド実行、ファイル操作およびヘルプの参照などのオペレーションを対話形式で行える。また、MATLABはインタプリタ型の言語であるため、コンパイル、リンクといったプログラムの作成および変更から実行するまでの手間が少ない。

- ・豊富な数値計算関数とグラフィックス関数

MATLABのみで800以上の関数、コマンドが提供されている。拡張パッケージであるToolboxにより、さらに関数、コマンドが提供されている。また、豊富なグラフィックス関数を用いることにより、容易に計算結果を可視化することができる。

5.2 MATLABで作成した関数一覧

数値計算を行うにあたって、MATLABで実際に作成した関数の一覧を表1に示す。

また、クォータニオン、オイラー角、方向余弦行列の関係および作成した関数の関係を図3に示す。図3で示されるように、方向余弦行列を介して、クォータニオンとオイラー角の変換が可能である。

次節以降で作成した各関数についての概要を説明する。

5.3 方向余弦行列計算

オイラー角から方向余弦行列を計算する関数 `euler2dcm` のソースコードを図4に示す。この関数の

表1 関数一覧

| 関数 | 処理内容 | 図番号 |
|-----------------------------------|--------------------|-----|
| <code>dcm=euler2dcm(euler)</code> | オイラー角から方向余弦行列を計算 | 図4 |
| <code>dcm=quat2dcm(q)</code> | クォータニオンから方向余弦行列を計算 | 図5 |
| <code>euler=dcm2euler(dcm)</code> | 方向余弦行列からオイラー角を計算 | 図6 |
| <code>q=dcm2quat(dcm)</code> | 方向余弦行列からクォータニオンを計算 | 図7 |
| <code>dx=euler_eq(t, x)</code> | オイラー角の微分方程式を計算 | 図8 |
| <code>dx=quat_eq(t, x)</code> | クォータニオンの微分方程式を計算 | 図9 |
| <code>simdemo</code> | オイラー角とクォータニオンの比較計算 | 図10 |

引数はオイラー角 3 要素をもつベクトル **euler** であり、戻り値は方向余弦行列 **dcm** である。計算式は、式(2)である。

次に、クォータニオンから方向余弦行列を計算する関数 **quat2dcm** のソースコードを図 5 に示す。この関数の引数はクォータニオン 4 要素をもつベクトル **quat** であり、戻り値は方向余弦行列 **dcm** である。計算式は、式(7)である。

5.4 方向余弦行列からオイラー角、クォータニオン計算

方向余弦行列からオイラー角から計算する関数 **dcm2euler** のソースコードを図 6 に、クォータニオンから方向余弦行列を計算する関数 **dcm2quat** のソースコードを図 7 に示す。これらは、式(3)、式(8)を実装したものである。ただし、**dcm2quat** 関数については、対角項から

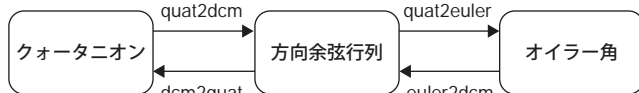


図 3 クォータニオン、オイラー角および方向余弦行列の関係

```

function dcm = euler2dcm(euler)
%function dcm = euler2dcm(euler)
%オイラー角から方向余弦行列を算出
%dcm: 方向余弦行列, euler: オイラー角 [rad]
sin_euler = sin(euler);
cos_euler = cos(euler);
dcm = [ cos_euler(2)*cos_euler(3), cos_euler(2)*sin_euler(3), -sin_euler(2);...
        -cos_euler(1)*sin_euler(3) + sin_euler(1)*sin_euler(2)*cos_euler(3),...
        cos_euler(1)*cos_euler(3) + sin_euler(1)*sin_euler(2)*sin_euler(3),...
        sin_euler(1)*cos_euler(2);...
        sin_euler(1)*sin_euler(3) + cos_euler(1)*sin_euler(2)*cos_euler(3),...
        -sin_euler(1)*cos_euler(3) + cos_euler(1)*sin_euler(2)*sin_euler(3),...
        cos_euler(1)*cos_euler(2)];
  
```

図 4 euler2dcm 関数

```

function dcm = quat2dcm(q)
%function dcm = quat2dcm(q)
%クォータニオンから方向余弦行列を計算
%dcm: 方向余弦行列, q: クォータニオン
q2 = q.^2;
dcm = [ q2(1) - q2(2) - q2(3) + q2(4), 2*(q(1)*q(2) + q(3)*q(4)),...
        2*(q(1)*q(3) - q(2)*q(4)),...
        2*(q(1)*q(2) - q(3)*q(4)), q2(2) - q2(1) - q2(3) + q2(4),...
        2*(q(2)*q(3) + q(1)*q(4)),...
        2*(q(1)*q(3) + q(2)*q(4)), 2*(q(2)*q(3) - q(1)*q(4)),...
        q2(3) - q2(1) - q2(2) + q2(4)];
  
```

図 5 quat2dcm 関数

各クォータニオン変数を算出し、その中で最大となる変数を選んで、その他の変数を再計算している。

5.5 微分方程式計算

オイラー角の微分方程式を計算する関数 **euler_eq** のソースコードを図 8 に示す。次に、クォータニオンの微分方程式を計算する関数 **quat_eq** のソースコードを図 9 に示す。また、この微分方程式を実際に解く関数が図 10 で示される **simdemo** である。この関数は、MATLAB 組み込み関数である常微分方程式のソルバ **ode23** を使用して両微分方程式を解いている。

```

function euler = dcm2euler(dcm)
%function euler = dcm2euler(dcm)
%方向余弦行列からオイラー角を算出
%euler: オイラー角 [rad], cm: 方向余弦行列
euler = [ atan2(dcm(2,3), dcm(3,3));...
          atan2(-dcm(1,3), sqrt(dcm(2,3)^2 + dcm(3,3)^2));...
          atan2(dcm(1,2), dcm(1,1))];
  
```

図 6 dcm2euler 関数

```

function q = dcm2quat(dcm)
%function q = dcm2quat(dcm)
%方向余弦行列からクォータニオンを計算
%q: クォータニオン, dcm: 方向余弦行列
q = [ sqrt(1 + dcm(1,1) - dcm(2,2) - dcm(3,3))/2;...
      sqrt(1 - dcm(1,1) + dcm(2,2) - dcm(3,3))/2;...
      sqrt(1 - dcm(1,1) - dcm(2,2) + dcm(3,3))/2;...
      sqrt(1 + dcm(1,1) + dcm(2,2) + dcm(3,3))/2];
[x, ix] = max(q);
switch ix
case 1
  q([2 3 4]) = 0.25/q(1) * [dcm(1,2)+dcm(2,1); dcm(1,3)+dcm(3,1); dcm(2,3)-dcm(3,2)];
case 2
  q([1 3 4]) = 0.25/q(2) * [dcm(1,2)+dcm(2,1); dcm(3,2)+dcm(2,3); dcm(3,1)-dcm(1,3)];
case 3
  q([1 2 4]) = 0.25/q(3) * [dcm(3,1)+dcm(1,3); dcm(3,2)+dcm(2,3); dcm(1,2)-dcm(2,1)];
case 4
  q([1 2 3]) = 0.25/q(4) * [dcm(2,3)-dcm(3,2); dcm(3,1)-dcm(1,3); dcm(1,2)-dcm(2,1)];
end
  
```

図 7 dcm2quat 関数

```

function dx = euler_eq(t,x)
%function dx = euler_eq(t,x)
%オイラー角の微分方程式
%dx: オイラー角の時間微分 [rad/s], t: 時間 [s], x: オイラー角 [rad]
global rate;
%オイラー角
phi = x(1);
the = x(2);
psi = x(3);
%角速度
p = rate(1);
q = rate(2);
r = rate(3);
sin_phi = sin(phi);
cos_phi = cos(phi);
%微分方程式
dx = [ p + (q*sin_phi + r*cos_phi)*tan(the);...
       q*cos_phi - r*sin_phi;...
       (q*sin_phi + r*cos_phi)*sec(the)];
  
```

図 8 euler_eq 関数

```

function dx = quat_eq(t,x)
%function dx = euler_eq(t,x)
%クォータニオンの微分方程式
%dx:クォータニオンの時間微分, t:時間[s], q:クォータニオン
global rate;
%角速度
p = rate(1);
q = rate(2);
r = rate(3);
dx = 0.5*[ 0 r -q p;...
           -r 0 p q;...
           q -p 0 r;...
           -p -q -r 0]*x;

```

図9 quat_eq 関数

```

%オイラー角とクォータニオンによる姿勢角計算比較
global rate;
%初期姿勢角
euler0 = [-30; -20; -10]*pi/180;
%euler0 = [0; 80; 0]*pi/180;
%角速度
rate = [5; 10; 15]*pi/180;
%rate = [0; 5; 0]*pi/180;
%初期姿勢角からクォータニオンを算出
q0 = dcm2quat(euler2dcm(euler0));
%計算時間、積分オプション設定
tspan = [0:10e-3:10];
ode_opt = odeset('InitialStep', 10e-3,...
                'MaxStep', 10e-3);
%オイラー表現による解法
[t1, y1] = ode23('euler_eq', tspan, euler0, ode_opt);
%クォータニオン表現による解法
[t2, y2] = ode23('quat_eq', tspan, q0, ode_opt);
%クォータニオン表現による解法結果をオイラー角に変換
y3 = zeros(length(t2), 3);
for ii = 1:length(t2)
    y3(ii, :) = dcm2euler(quat2dcm(y2(ii, :)))';
end

```

図10 simdemo 関数

6. 数値計算結果

数値計算は、表2に示す2条件で行った。条件1として、特異点をとらない場合、オイラー角、クォータニオンによる両手法が一致することを確認する条件を選んだ。次に、条件2として、オイラー角が特異点をとる条件を選んだ。

まず、条件1での数値計算結果を図11、12に示す。図11で示すクォータニオン表現による姿勢角は、クォータニオンを方向余弦行列に変換し、さらにオイラー角に変換したものである。条件1は、オイラー角が特異点による発散がない条件であるため、図11からわかるようにオイラー角、クォータニオンによる計算結果が一致する。オイラー角に変換する前のクォータニオンの各パラメータは図12を参照されたい。

条件2での数値計算結果を図13、14に示す。条件2は、時間2sでオイラー角の特異点であるピッチ姿勢角 θ

表2 シミュレーション条件

| | 条件1 | 条件2 |
|----------|---------|---------|
| 初期ロール姿勢角 | -30deg | 0 deg |
| 初期ピッチ姿勢角 | -20deg | 80deg |
| 初期ヨー姿勢角 | -10deg | 0 deg |
| ロール角速度 | 5 deg/s | 0 deg/s |
| ピッチ角速度 | 10deg/s | 5 deg/s |
| ヨー角速度 | 15deg/s | 0 deg/s |

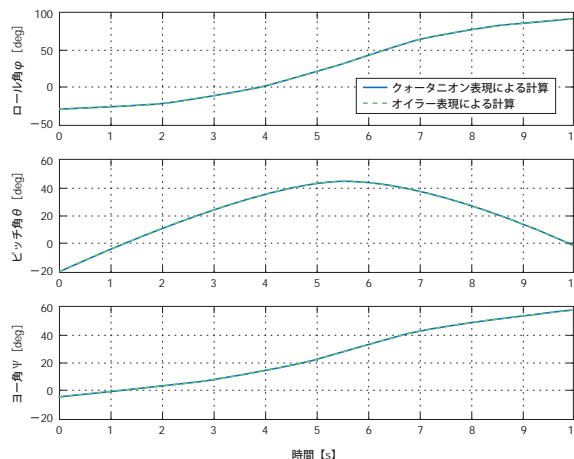


図11 条件1計算結果：オイラー角

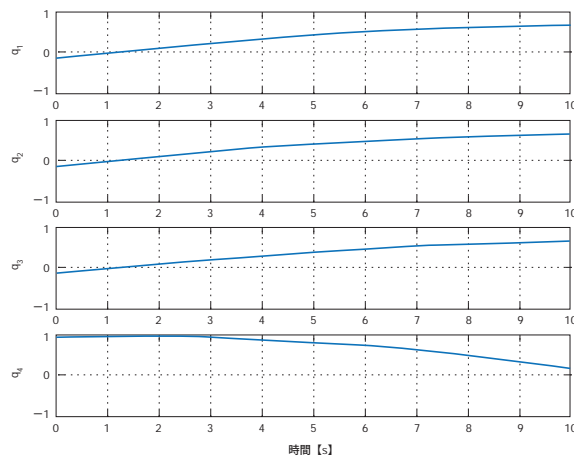


図12 条件1計算結果：クォータニオン

が90degになる条件である。そのため、図13からわかるようにオイラー角による計算では、2sにロール姿勢角とヨー姿勢角が発散している。これに対してクォータニオンによる計算では、図14で示されるように発散せず良好な結果である。クォータニオンから変換したピッチ姿勢角 θ が2s以降減少しているのは、方向余弦行列からオイラー角に変換する際に式(3)のとおりピッチ姿勢角の範囲が $-90\text{deg} \leq \theta \leq 90\text{deg}$ となるためである。ただし、クォータニオン自体に発散はないため問題ではない。

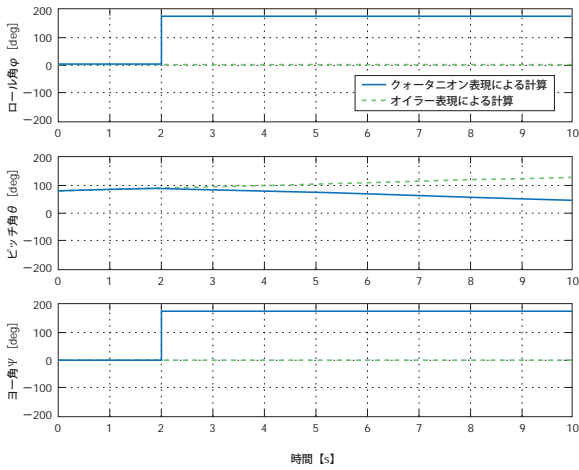


図13 条件2計算結果：オイラー角

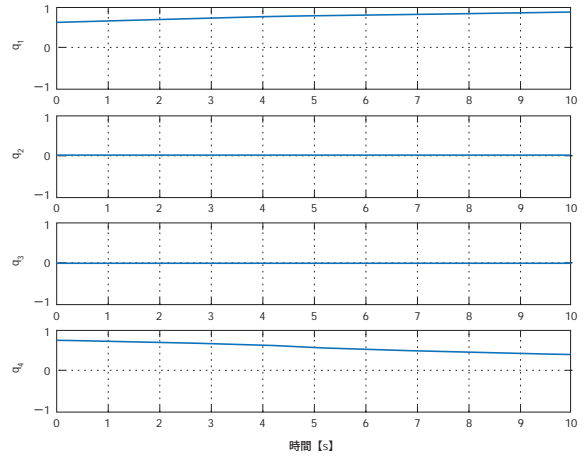


図14 条件2計算結果：クォータニオン

7. むすび

剛体の回転表現であるオイラー角とクォータニオンについて、定式化とMATLABによる実装および数値計算を行った。

数値計算結果から、特異点をとらない飛行条件ではオイラー角とクォータニオンの両手法の結果が一致することを確認した。また、特異点をとる飛行条件においては、クォータニオンの優位性を確認した。これにより、クォータニオンが飛行数値シミュレーションで有用な回転表現であることを示せた。

参考文献

- (1) James R. Wertz: *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, 1978.
- (2) 加藤寛一郎：航空機力学入門，東京大学出版会，1982.
- (3) 矢田部学：クォータニオン計算便利ノート，MSS技報vol.18，2007.