Attitude Estimation Examples from Chapter 3

Setup

In[214]:=
```
<< RigidBodyKinematics`
```

In[215]:=
```
tilde[x_] := {{0, -x[[3]], x[[2]]},
   {x[[3]], 0, -x[[1]]},
   {-x[[2]], x[[1]], 0}}
```

Example 3.14 (Triad)

Setup the true attitude states:

In[216]:=
```
θtrue = {30., 20., -10.} Degree;
BNtrue = Euler3212C[θtrue]
```

Out[217]=
```
{{0.813798, 0.469846, -0.34202},
 {-0.543838, 0.823173, -0.163176}, {0.204874, 0.318796, 0.925417}}
```

In[218]:=
```
v1N = {1, 0, 0};
v2N = {0, 0, 1};
```

In[220]:=
```
MatrixForm[v1Btrue = BNtrue.v1N ]
```

Out[220]//MatrixForm=

$$\begin{pmatrix} 0.813798 \\ -0.543838 \\ 0.204874 \end{pmatrix}$$

In[221]:=
```
MatrixForm[v2Btrue = BNtrue.v2N ]
```

Out[221]//MatrixForm=

$$\begin{pmatrix} -0.34202 \\ -0.163176 \\ 0.925417 \end{pmatrix}$$

Setup the measured attitude states:

In[222]:=
```
v1B = {0.8190, -0.5282, 0.2242};
v2B = {-0.3138, -0.1584, 0.9362};
v1B = v1B / Norm[v1B];
v2B = v2B / Norm[v2B];
```

Develop Triad Frame

From measured states

In[226]:=
```
MatrixForm[t1B = v1B]
t2B = Cross[v1B, v2B];
MatrixForm[t2B = t2B / Norm[t2B]]
MatrixForm[t3B = Cross[t1B, t2B]]
```

Out[226]//MatrixForm=

$$\begin{pmatrix} 0.818991 \\ -0.528194 \\ 0.224198 \end{pmatrix}$$

Out[228]//MatrixForm=

$$\begin{pmatrix} -0.459282 \\ -0.837639 \\ -0.295669 \end{pmatrix}$$

Out[229]//MatrixForm=

$$\begin{pmatrix} 0.343967 \\ 0.13918 \\ -0.928609 \end{pmatrix}$$

From inertial states of the measurements

In[230]:=
```
MatrixForm[t1N = v1N]
t2N = Cross[t1N, v2N ];
MatrixForm[t2N = t2N / Norm[t2N]]
MatrixForm[t3N = Cross[t1N, t2N]]
```

Out[230]//MatrixForm=

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Out[232]//MatrixForm=

$$\begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

Out[233]//MatrixForm=

$$\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Find Estimated Attitude

In[234]:=
```
MatrixForm[BbarT = Transpose[{t1B, t2B, t3B}]]
```

Out[234]//MatrixForm=

$$\begin{pmatrix} 0.818991 & -0.459282 & 0.343967 \\ -0.528194 & -0.837639 & 0.13918 \\ 0.224198 & -0.295669 & -0.928609 \end{pmatrix}$$

In[235]:=
```
NT = Transpose[{t1N, t2N, t3N}]
```

Out[235]=
```
{{1, 0, 0}, {0, -1, 0}, {0, 0, -1}}
```

In[236]:=
```
MatrixForm[BbarN = BbarT.Transpose[NT]]
```

Out[236]//MatrixForm=

$$\begin{pmatrix} 0.818991 & 0.459282 & -0.343967 \\ -0.528194 & 0.837639 & -0.13918 \\ 0.224198 & 0.295669 & 0.928609 \end{pmatrix}$$

Check accuracy of estimate

In[237]:=
```
MatrixForm[BbarB = BbarN.Transpose[BNtrue]]
```

Out[237]//MatrixForm=

$$\begin{pmatrix} 0.999929 & -0.0112026 & -0.00410552 \\ 0.0113209 & 0.999485 & 0.0300232 \\ 0.00376707 & -0.0300675 & 0.999541 \end{pmatrix}$$

In[238]:=
```
p = C2PRV[BbarB]
```

Out[238]=
```
{0.0300506, 0.00393698, -0.0112637}
```

In[239]:=
```
Norm[p] / Degree
```

Out[239]=
```
1.85253
```

Example 3.15 (Devenport’s q-Method)

Setup the true attitude states:

In[240]:=
```
θtrue = {30., 20., -10.} Degree;
BNtrue = Euler3212C[θtrue]
```

Out[241]=
```
{{0.813798, 0.469846, -0.34202},
 {-0.543838, 0.823173, -0.163176}, {0.204874, 0.318796, 0.925417}}
```

In[242]:=
```
v1N = {1, 0, 0};
v2N = {0, 0, 1};
```

In[244]:=
```
MatrixForm[v1Btrue = BNtrue.v1N ]
```
Out[244]//MatrixForm=

$$\begin{pmatrix} 0.813798 \\ -0.543838 \\ 0.204874 \end{pmatrix}$$

In[245]:=
```
MatrixForm[v2Btrue = BNtrue.v2N ]
```
Out[245]//MatrixForm=

$$\begin{pmatrix} -0.34202 \\ -0.163176 \\ 0.925417 \end{pmatrix}$$

Setup the measured attitude states:

In[246]:=
```
v1B = {0.8190, -0.5282, 0.2242};
v2B = {-0.3138, -0.1584, 0.9362};
v1B = v1B / Norm[v1B];
v2B = v2B / Norm[v2B];
```

Setup q-method parameters

In[250]:=
```
w1 = 1;
w2 = 1;
```

In[252]:=
```
B = w1 Outer[Times, v1B, v1N] + w2 Outer[Times, v2B, v2N]
```
Out[252]=
```
{{0.818991, 0., -0.313795}, {-0.528194, 0., -0.158398}, {0.224198, 0., 0.936185}}
```

In[253]:=
```
S = B + Transpose[B]
```
Out[253]=
```
{{1.63798, -0.528194, -0.0895975},
 {-0.528194, 0., -0.158398}, {-0.0895975, -0.158398, 1.87237}}
```

In[254]:=
```
σ = B[[1, 1]] + B[[2, 2]] + B[[3, 3]]
```
Out[254]=
```
1.75518
```

In[255]:=
```
Z = {B[[2, 3]] - B[[3, 2]],
   B[[3, 1]] - B[[1, 3]],
   B[[1, 2]] - B[[2, 1]]}
```
Out[255]=
```
{-0.158398, 0.537993, 0.528194}
```

In[256]:=
```
MatrixForm[
 K = {{σ, Z[[1]], Z[[2]], Z[[3]]},
   {Z[[1]], S[[1, 1]] - σ, S[[1, 2]], S[[1, 3]]},
   {Z[[2]], S[[2, 1]], S[[2, 2]] - σ, S[[2, 3]]},
   {Z[[3]], S[[3, 1]], S[[3, 2]], S[[3, 3]] - σ}}
]
```

Out[256]//MatrixForm=

$$\begin{pmatrix} 1.75518 & -0.158398 & 0.537993 & 0.528194 \\ -0.158398 & -0.117194 & -0.528194 & -0.0895975 \\ 0.537993 & -0.528194 & -1.75518 & -0.158398 \\ 0.528194 & -0.0895975 & -0.158398 & 0.117194 \end{pmatrix}$$

Solve for optimal attitude

In[257]:=
```
{λ, βvectors} = Eigensystem[K];
```

In[258]:=
```
λ
```

Out[258]= {1.99967, -1.99967, 0.0365659, -0.0365659}

In[259]:=
```
β = βvectors[[1]]
```

Out[259]= {0.948069, -0.117207, 0.141371, 0.259697}

In[260]:=
```
MatrixForm[BbarN = EP2C[β]]
```

Out[260]//MatrixForm=

$$\begin{pmatrix} 0.825143 & 0.459282 & -0.328936 \\ -0.525561 & 0.837639 & -0.148814 \\ 0.207182 & 0.295669 & 0.932553 \end{pmatrix}$$

Check accuracy of estimate

In[261]:=
```
MatrixForm[BbarB = BbarN.Transpose[BNtrue]]
```

Out[261]//MatrixForm=

$$\begin{pmatrix} 0.999794 & -0.0170009 & 0.0110648 \\ 0.0167585 & 0.999625 & 0.0216474 \\ -0.0114287 & -0.0214576 & 0.999704 \end{pmatrix}$$

In[262]:=
```
q = C2PRV[BbarB]
```

Out[262]= {0.0215556, -0.0112484, -0.0168821}

In[263]:=
```
ΦErrorQmethod = Norm[q] / Degree
```

Out[263]= 1.69597

Example 3.16 (QUEST)

Setup the true attitude states:

In[264]:=
```
θtrue = {30., 20., -10.} Degree;
BNtrue = Euler3212C[θtrue]
```

Out[265]=
```
{{0.813798, 0.469846, -0.34202},
 {-0.543838, 0.823173, -0.163176}, {0.204874, 0.318796, 0.925417}}
```

In[266]:=
```
v1N = {1, 0, 0};
v2N = {0, 0, 1};
```

In[268]:=
```
MatrixForm[v1Btrue = BNtrue.v1N]
```

Out[268]//MatrixForm=

$$\begin{pmatrix} 0.813798 \\ -0.543838 \\ 0.204874 \end{pmatrix}$$

In[269]:=
```
MatrixForm[v2Btrue = BNtrue.v2N]
```

Out[269]//MatrixForm=

$$\begin{pmatrix} -0.34202 \\ -0.163176 \\ 0.925417 \end{pmatrix}$$

Setup the measured attitude states:

In[270]:=
```
v1B = {0.8190, -0.5282, 0.2242};
v2B = {-0.3138, -0.1584, 0.9362};
v1B = v1B / Norm[v1B];
v2B = v2B / Norm[v2B];
```

Setup q-method parameters

In[274]:=
```
w1 = 1;
w2 = 1;
```

In[276]:=
```
B = w1 Outer[Times, v1B, v1N] + w2 Outer[Times, v2B, v2N]
```

Out[276]=
```
{{0.818991, 0., -0.313795}, {-0.528194, 0., -0.158398}, {0.224198, 0., 0.936185}}
```

In[277]:=
```
S = B + Transpose[B]
```

Out[277]=
```
{{1.63798, -0.528194, -0.0895975},
 {-0.528194, 0., -0.158398}, {-0.0895975, -0.158398, 1.87237}}
```

In[278]:=
```
σ = B[[1, 1]] + B[[2, 2]] + B[[3, 3]]
```

Out[278]=
```
1.75518
```

In[279]:=
```
Z = {B[[2, 3]] - B[[3, 2]],
  B[[3, 1]] - B[[1, 3]],
  B[[1, 2]] - B[[2, 1]]}
```

Out[279]= {-0.158398, 0.537993, 0.528194}

In[280]:=
```
MatrixForm[
 K = {{σ, Z[[1]], Z[[2]], Z[[3]]},
   {Z[[1]], S[[1, 1]] - σ, S[[1, 2]], S[[1, 3]]},
   {Z[[2]], S[[2, 1]], S[[2, 2]] - σ, S[[2, 3]]},
   {Z[[3]], S[[3, 1]], S[[3, 2]], S[[3, 3]] - σ}}
]
```

Out[280]//MatrixForm=

$$\begin{pmatrix} 1.75518 & -0.158398 & 0.537993 & 0.528194 \\ -0.158398 & -0.117194 & -0.528194 & -0.0895975 \\ 0.537993 & -0.528194 & -1.75518 & -0.158398 \\ 0.528194 & -0.0895975 & -0.158398 & 0.117194 \end{pmatrix}$$

Setup QUEST parameters

In[281]:=
```
λ = w1 + w2
```

Out[281]= 2

In[282]:=
```
q = Inverse[ (λ + σ) IdentityMatrix[3] - S].Z
```

Out[282]= {-0.123602, 0.1491, 0.273874}

In[283]:=
```
MatrixForm[BbarN = Gibbs2C[q]]
```

Out[283]//MatrixForm=

$$\begin{pmatrix} 0.825193 & 0.45922 & -0.328897 \\ -0.525482 & 0.837693 & -0.148793 \\ 0.207186 & 0.295613 & 0.93257 \end{pmatrix}$$

In[284]:=
```
MatrixForm[BbarB = BbarN.Transpose[BNtrue]]
```

Out[284]//MatrixForm=

$$\begin{pmatrix} 0.999793 & -0.0170853 & 0.0110912 \\ 0.0168417 & 0.999623 & 0.0216995 \\ -0.0114577 & -0.0215082 & 0.999703 \end{pmatrix}$$

In[285]:=
```
q = C2PRV[BbarB]
```

Out[285]= {0.021607, -0.0112761, -0.016966}

In[286]:=
```
Norm[q] / Degree
```

Out[286]= 1.70146

Iterate for Optimal Attitude

In[287]:=
```
Det[K - s * IdentityMatrix[4]]
```
Out[287]= $0.00534646 + 7.35523 \times 10^{-16}\, s - 4.\, s^2 - 2.22045 \times 10^{-16}\, s^3 + s^4$

In[288]:=
```
f[s_] := Det[K - s * IdentityMatrix[4]]
```

In[289]:=
```
λ0 = λ
```
Out[289]= 2

In[290]:=
```
f[λ0]
```
Out[290]= 0.00534646

In[291]:=
```
λ1 = λ0 - f[λ0] / f'[λ0]
```
Out[291]= 1.99967

In[292]:=
```
f[λ1]
```
Out[292]= 2.23288×10^{-6}

In[293]:=
```
λ2 = λ1 - f[λ1] / f'[λ1]
```
Out[293]= 1.99967

In[294]:=
```
f[λ2]
```
Out[294]= 3.90176×10^{-13}

In[295]:=
```
λ3 = λ2 - f[λ2] / f'[λ2]
```
Out[295]= 1.99967

In[296]:=
```
f[λ3]
```
Out[296]= -3.45856×10^{-16}

Iteration Accuracy Study

In[297]:=
```
Norm[C2PRV[ Gibbs2C[Inverse[ (λ0 + σ) IdentityMatrix[3] - S].Z].Transpose[BNtrue] ]] /
  Degree - ΦErrorQmethod
```
Out[297]= 0.00549088

In[298]:=
```
Norm[C2PRV[ Gibbs2C[Inverse[ (λ1 + σ) IdentityMatrix[3] - S].Z].Transpose[BNtrue] ]] /
  Degree - ΦErrorQmethod
```
Out[298]= 2.29401×10^{-6}

In[299]:=
```
Norm[C2PRV[ Gibbs2C[Inverse[ (λ2 + σ) IdentityMatrix[3] - S].Z].Transpose[BNtrue] ]] /
  Degree - ΦErrorQmethod
```
Out[299]= 3.76588×10^{-13}

```
In[300]:= Norm[C2PRV[ Gibbs2C[Inverse[ (λ3 + σ) IdentityMatrix[3] - S].Z].Transpose[BNtrue] ]] /
            Degree - ΦErrorQmethod
Out[300]= -2.90878 × 10^-14
```

Example 3.17 (OLAE)

Setup the true attitude states:

```
In[301]:= θtrue = {30., 20., -10.} Degree;
          BNtrue = Euler3212C[θtrue]
Out[302]= {{0.813798, 0.469846, -0.34202},
           {-0.543838, 0.823173, -0.163176}, {0.204874, 0.318796, 0.925417}}
```

```
In[303]:= v1N = {1, 0, 0};
          v2N = {0, 0, 1};
```

```
In[305]:= MatrixForm[v1Btrue = BNtrue.v1N ]
Out[305]//MatrixForm=
```

$$\begin{pmatrix} 0.813798 \\ -0.543838 \\ 0.204874 \end{pmatrix}$$

```
In[306]:= MatrixForm[v2Btrue = BNtrue.v2N ]
Out[306]//MatrixForm=
```

$$\begin{pmatrix} -0.34202 \\ -0.163176 \\ 0.925417 \end{pmatrix}$$

Setup the measured attitude states:

```
In[307]:= v1B = {0.8190, -0.5282, 0.2242};
          v2B = {-0.3138, -0.1584, 0.9362};
          v1B = v1B / Norm[v1B];
          v2B = v2B / Norm[v2B];
```

Evaluate OLAE states

```
In[311]:= W = IdentityMatrix[6];
```

```
In[312]:= d = Join[v1B - v1N, v2B - v2N]
Out[312]= {-0.181009, -0.528194, 0.224198, -0.313795, -0.158398, -0.0638147}
```

```
In[313]:= S = Join[tilde[v1B + v1N], tilde[v2B + v2N]]
Out[313]= {{0, -0.224198, -0.528194}, {0.224198, 0, -1.81899}, {0.528194, 1.81899, 0},
           {0, -1.93619, -0.158398}, {1.93619, 0, 0.313795}, {0.158398, -0.313795, 0}}
```

```
In[314]:= qBar = Inverse[Transpose[S].W.S].Transpose[S].W.d
Out[314]= {-0.12359, 0.148759, 0.274255}

In[315]:= BbarN = Gibbs2C[qBar]
Out[315]= {{0.825016, 0.459942, -0.328332},
           {-0.526039, 0.837338, -0.148823}, {0.206474, 0.295497, 0.932765}}
```

Check accuracy of estimate

```
In[316]:= BbarB = BbarN.Transpose[BNtrue]
Out[316]= {{0.999794, -0.0164875, 0.0118086},
           {0.0162316, 0.999638, 0.0214443}, {-0.0121579, -0.0212482, 0.9997}}

In[317]:= q = C2PRV[BbarB]
Out[317]= {0.0213494, -0.011985, -0.0163619}

In[318]:= ΦErrorQmethod = Norm[q] / Degree
Out[318]= 1.68721
```