Attitude Determination

- Using GPS
Table of Contents

- Definition of Attitude
- Attitude and GPS
- Attitude Representations
- Least Squares Filter
- Kalman Filter
- Other Filters
- The AAU Testbed
- Results
- Conclusion
What is Attitude?

Orientation of a coordinate system \((u,v,w)\) with respect to some reference system \((x,y,z)\)
When is Attitude information needed?

- Controlling an Aircraft, Boat or Automobile
- Onboard Satellites
- Pointing of Instruments
- Pointing of Weapons
- Entertainment industri (VR)
- Etc...
Attitude sensors

Currently used sensors include:

- Gyroscopes
- Rate gyros (+integration)
- Star trackers
- Sun sensors
- Magnetometers
- GPS
Advantages of GPS

- Adding new functionality to existing equipment
- No cost increase
- No weight increase
- No moving parts (solid-state)
- Measures the absolute attitude

Disadvantages

- Mediocre accuracy (0.1 - 1° RMS error)
- Low bandwidth (5-10 Hz maximum)
- Requires direct view of satellites
Interferometric Principle

Carrier Wave (from GPS satellite)

Master Antenna

Integer Component

k

Slave Antenna

Fractional Component, $\Delta \phi$

Baseline, $b(3 \times 1)$

e$(3 \times 1)$

Unit Vector to GPS Satellite


**Interferometric Principle**

Measurement equation:

\[
\Delta \varphi = \Delta r - k + \beta + v
\]

The full phase difference is the projection of the baseline vector onto the LOS vector:

\[
\Delta r = b \cdot e = |b| \cos \theta
\]
Attitude Matrix

9 parameters needed:

\[
A = \begin{bmatrix}
  u \cdot x & u \cdot y & u \cdot z \\
  v \cdot x & v \cdot y & v \cdot z \\
  w \cdot x & w \cdot y & w \cdot z
\end{bmatrix}
\]

When \((x,y,z)\) is a reference system:

\[
A = \begin{bmatrix}
  u_x & u_y & u_z \\
  v_x & v_y & v_z \\
  w_x & w_y & w_z
\end{bmatrix} = \begin{bmatrix}
  u^T \\
  v^T \\
  w^T
\end{bmatrix}
\]
Properties of “A”

“A” rotates a vector from the reference system to the body system

\[ a^B = A^B_R a^R \]

The transpose of “A” rotates in the opposite direction (back again)

\[ A^T A = I_{3\times3} \]
Properties of “A”

Rotation does not change the size of the vectors:

\[ \det A = 1 \]

Every rotation has a rotation-axis (and a rotation-angle)

\[ A e = e \]

The rotation-angle is the eigenvalue of “A”
Euler sequences

A sequence of rotations by the angles \((\phi, \theta, \psi)\) about the coordinate axes of the reference system

**Single axis:**

\[
A_1(\phi) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \phi & \sin \phi \\
0 & -\sin \phi & \cos \phi
\end{bmatrix}
\]

**Multiple axes:**

\[
A_{123}(\phi, \theta, \psi) = A_3(\psi)A_2(\theta)A_1(\phi)
\]
Quaternions

A quaternion consists of four composants

\[ q = q_4 + iq_1 + jq_2 + kq_3 \]

Where i, j and k are hyperimaginary numbers

\[
\begin{align*}
  i^2 &= j^2 &= k^2 &= -1 \\
  ij &= -ji &= k \\
  jk &= -kj &= i \\
  ki &= -ik &= j
\end{align*}
\]
Quaternions

A quaternion can be thought of as a 4 dimensional vector with unit length:

\[ q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} q \\ q_4 \end{bmatrix} \]

\[ q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \]
Quaternions

Quaternions represent attitude as a rotation-axis and a rotation-angle

\[ q_1 = e_1 \sin \frac{\Phi}{2} \]
\[ q_2 = e_2 \sin \frac{\Phi}{2} \]
\[ q_3 = e_3 \sin \frac{\Phi}{2} \]
\[ q_4 = \cos \frac{\Phi}{2} \]
Quaternions can be multiplied using the special operator $q'' = q' \otimes q$ defined as:

$$\Lambda(qq') = \Lambda(q') \Lambda(q)$$

$$
\begin{bmatrix}
q''_1 \\
q''_2 \\
q''_3 \\
q''_4
\end{bmatrix} = 
\begin{bmatrix}
q'_4 & q'_3 & -q'_2 & q'_1 \\
-q'_3 & q'_4 & q'_1 & q'_2 \\
q'_2 & -q'_1 & q'_4 & q'_3 \\
-q'_1 & -q'_2 & -q'_3 & q'_4
\end{bmatrix}
\begin{bmatrix}
q_1 \\
q_2 \\
q_3 \\
q_4
\end{bmatrix}
$$
Quaternions

The attitude matrix can be formed from the quaternion as:

\[ A(q) = \begin{bmatrix}
q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\
2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\
2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2
\end{bmatrix} \]

\[ = \begin{bmatrix}
(q_4^2 - |q|^2)I_{3\times3} + 2qq^T - 2q_4 Q^x
\end{bmatrix} \]

Where

\[ Q^x = \begin{bmatrix}
0 & -q_3 & q_2 \\
q_3 & 0 & -q_1 \\
-q_2 & q_1 & 0
\end{bmatrix} \]
Least Squares Solution

Including attitude information into the measurement equation

\[ \Delta r = b \cdot e \]
\[ = (b^B)^T A e^R \Rightarrow \]
\[ \Delta \varphi = (b^B)^T A e^R - k + \beta + \nu \]

Linearization of the attitude matrix

\[ A = \delta A \hat{A} = (I - 2Q^x) \hat{A} \]
Least Squares Solution

Forming the phase residual

\[
\Delta \varphi_{ij} = (b_j^B)^T (\hat{A}e_i^R) - (b_j^B)^T (2Q \times \hat{A}e_i^R) - k_{ij} + \beta_j + v_{ij}
\]

\[
\downarrow
\]

\[
\delta \varphi_{ij} = \Delta \varphi_{ij} - \Delta \hat{\varphi}_{ij}
\]

\[
= -(b_j^B)^T (2Q \times \hat{A}e_i^R)
\]

\[
= -2(\hat{A}e_i^R)^T B_j^x \delta q
\]
Least Squares Solution

\[
\begin{align*}
\mathbf{z} &= \begin{bmatrix} 
\vdots \\
\delta \varphi_{ij} \\
\vdots
\end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 
\vdots \\
-2(\hat{\mathbf{A}} \mathbf{e}_i^R)^T \mathbf{B}_j^x \\
\vdots
\end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix}
\delta q_1 \\
\delta q_2 \\
\delta q_3
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\mathbf{Hx} &= \mathbf{z} \\
\mathbf{H}^T \mathbf{Hx} &= \mathbf{H}^T \mathbf{z} \\
\mathbf{x} &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{z}
\end{align*}
\]
Least Squares Solution

Estimate update

\[ \delta \hat{q} = \text{norm} \left( \begin{bmatrix} \delta \hat{q}_1 \\ \delta \hat{q}_2 \\ \delta \hat{q}_3 \\ 1 \end{bmatrix} \right) \]

\[ \hat{q} = \delta \hat{q} \otimes \hat{q}' \]
Extended Kalman Filter

A Kalman filter consists of a model equation

\[ \dot{x}(t) = f[x(t), t] + w(t), \quad w(t) \sim N(0, Q(t)) \]

and a measurement equation

\[ z(t_k) = h[x(t_k), t_k] + v(t_k), \quad k = 1, 2, \ldots \quad v(t_k) \sim N(0, R(t_k)) \]
Extended Kalman Filter

And their linearized counterparts....

\[ F[x_n(t_k), t_k] = \left. \frac{\partial f[x, t_k]}{\partial x} \right|_{x=x_n(t_k)} \]

And

\[ H[x_n(t_k), t_k] = \left. \frac{\partial h[x, t_k]}{\partial x} \right|_{x=x_n(t_k)} \]
Extended Kalman Filter

Algorithm

\[ z(t_k) \]
\[ x(t_k^-), P(t_k^-) \]
Estimate
Update

\[ x(t_k), P(t_k) \]
\[ x(t_k^+), P(t_k^+) \]
\[ k = k + 1 \]

Time
Propagation

\[ x(t_{k+1}^-), P(t_{k+1}^-) \]

Estimate of

\[ K = PH^T \left( HPH^T + R \right)^{-1} \]
\[ \delta x = K \{ z - h \} \]
\[ P^+ = P^- - KHP^- \]
\[ \dot{x} = f \]
\[ \dot{P} = FP + PF^T + Q \]
Extended Kalman Filter

Tuning of the filter

\[ R = \begin{bmatrix}
\sigma^2_R & 0 & \ldots & 0 \\
0 & \sigma^2_R & & \vdots \\
\vdots & & \ddots & 0 \\
0 & \ldots & 0 & \sigma^2_R \\
\end{bmatrix} = \sigma^2_R I_{n \times n} \]

Noise variance determined experimentally

\[ \sigma_R = 0.028 \lambda \approx 0.5 \text{cm} \]
Extended Kalman Filter

Tuning of the filter

\[
Q = \begin{bmatrix}
\sigma_Q^2 & 0 & \ldots & 0 \\
0 & \sigma_Q^2 & \ddots & \\
\vdots & \ddots & \ddots & 0 \\
0 & \ldots & 0 & \sigma_Q^2
\end{bmatrix} = \sigma_Q^2 I_{n \times n}
\]

Noise variance determined by ‘trial-and-error’
Extended Kalman Filter

Determining the system model

\[ \dot{\omega} = I^{-1}( -\omega \times I\omega ) \]

\[ \dot{q} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} q \]
Other Filters

- Raw phase measurements
  - Integer resolution
- Extended phase measurements
  - Line bias calibration
- Corrected phase measurements
  - Vector Determination
    - Vector estimate
  - Attitude Determination
    - Attitude estimate

Non-iterative

Iterative

- Kalman filter
- ALLEGRO
- Cohen
- Least Squares
- New Algorithm
- Euler-q
- Geometric Descent
- Bar-Itzhack 2
- Bar-Itzhack
Testbed
Software

Start

Load simulated attitude

Start User Interface + Initialize Testbed

Proces 1
Store phases and LOS vectors from receiver in file

GPS receiver

phase.out

Proces 2
Store angle-values from encoders in file

angles.out

Proces 3
Send angle commands to LON nodes

LON nodes

los.out

 angles.txt

Stop
Motor Control

\[ A_{213} = \begin{bmatrix}
    \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \sin \psi \cos \theta & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi \\
    -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \cos \psi \cos \theta & \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\
    \cos \theta \sin \phi & -\sin \theta & \cos \theta \cos \phi 
\end{bmatrix} \]

\[ \downarrow \]

\[ \phi = \arctan(A_{31}/A_{33}) \]
\[ \theta = \arcsin(A_{32}) \]
\[ \psi = \arctan(A_{12}/A_{22}) \]
Motor Angles

![Graph of Motor Angles](image)

- Angle of Antenna Array
- Time [seconds]
- Angle [degrees]
Local Horizontal System

\[ A_{LH}^{ECEF} = A_{shift} \cdot A_2(-el) \cdot A_3(az) \]

\[ A_{shift} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \]
Results

Based on actual and simulated data, the following performance parameters were evaluated:

• Accuracy
• Computational efficiency
• Ability to converge
## Accuracy

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RSS error in degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 coplanar</td>
</tr>
<tr>
<td>Kalman filter</td>
<td>0.1756</td>
</tr>
<tr>
<td>Single-point</td>
<td>0.4933</td>
</tr>
<tr>
<td>ALLEGRO</td>
<td>0.4936</td>
</tr>
<tr>
<td>Geometric descent</td>
<td>(Does not converge)</td>
</tr>
<tr>
<td>New Algorithm + SVD</td>
<td>0.5366</td>
</tr>
<tr>
<td>Bar-Itzhack + SVD</td>
<td>0.5282</td>
</tr>
<tr>
<td>Cohen</td>
<td>0.5320</td>
</tr>
<tr>
<td>Euler-q</td>
<td>0.5412</td>
</tr>
</tbody>
</table>
Speed

Speed Analysis with three baselines

Number of Floating Point Operations

- Kalman filter w. bias est.
- Kalman filter
- Geometric Descent
- Bar Itzhack 2

Number of satellites

Number of Floating Point Operations

- ALLEGRO
- New Algorithm + Euler-q
- New Algorithm + SVD
- Single-point
- Cohen
- Bar Itzhack + SVD
Convergence

Kalman Filter
- Number of baselines = 3
- Convergence = 100%
- Average = 5.75

Single-point Algorithm
- Number of baselines = 3
- Convergence = 100%
- Average = 4.25

ALLEGRO Algorithm
- Number of baselines = 3
- Convergence = 100%
- Average = 4.56

Geometric Descent Algorithm
- Number of baselines = 3
- Convergence = 100%
- Average = 5.33
Convergence

Kalman Filter
Number of baselines = 2
Convergence = 96.61%
Average = 7.87

Single-point Algorithm
Number of baselines = 2
Convergence = 69.34%
Average = 4.64

ALLEGRO Algorithm
Number of baselines = 2
Convergence = 69.75%
Average = 4.76
Conclusion

- Kalman filter is by far most accurate, but also computationally very heavy

- Single-point (LSQ) offers good accuracy + high speed

- Vector matching algorithms has the lowest accuracy but does not suffer from convergence problems

- Performance depend on satellite constellation

- Results were affected by mechanical problems with levelling of the testbed