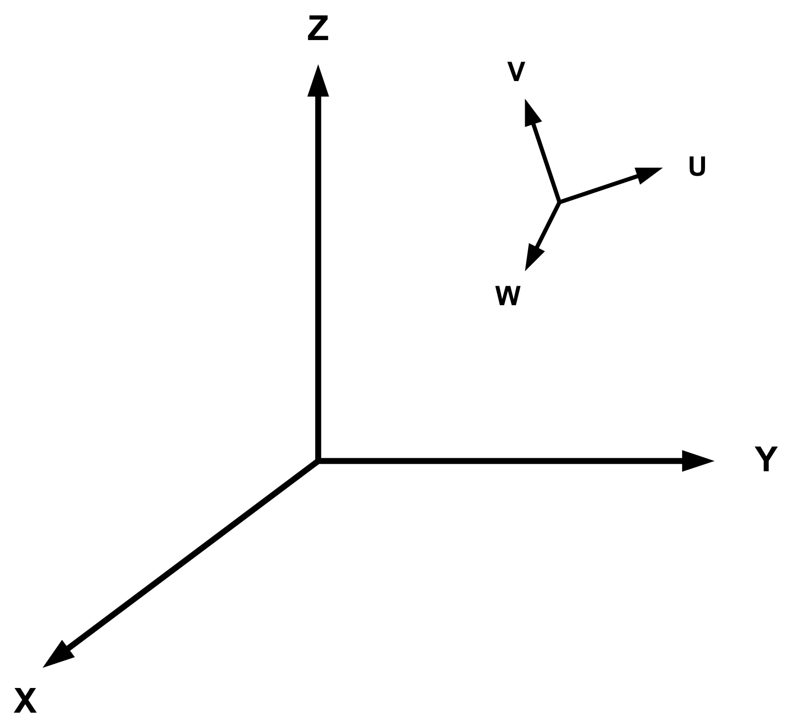
Attitude Determination

- Using GPS

Table of Contents

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**

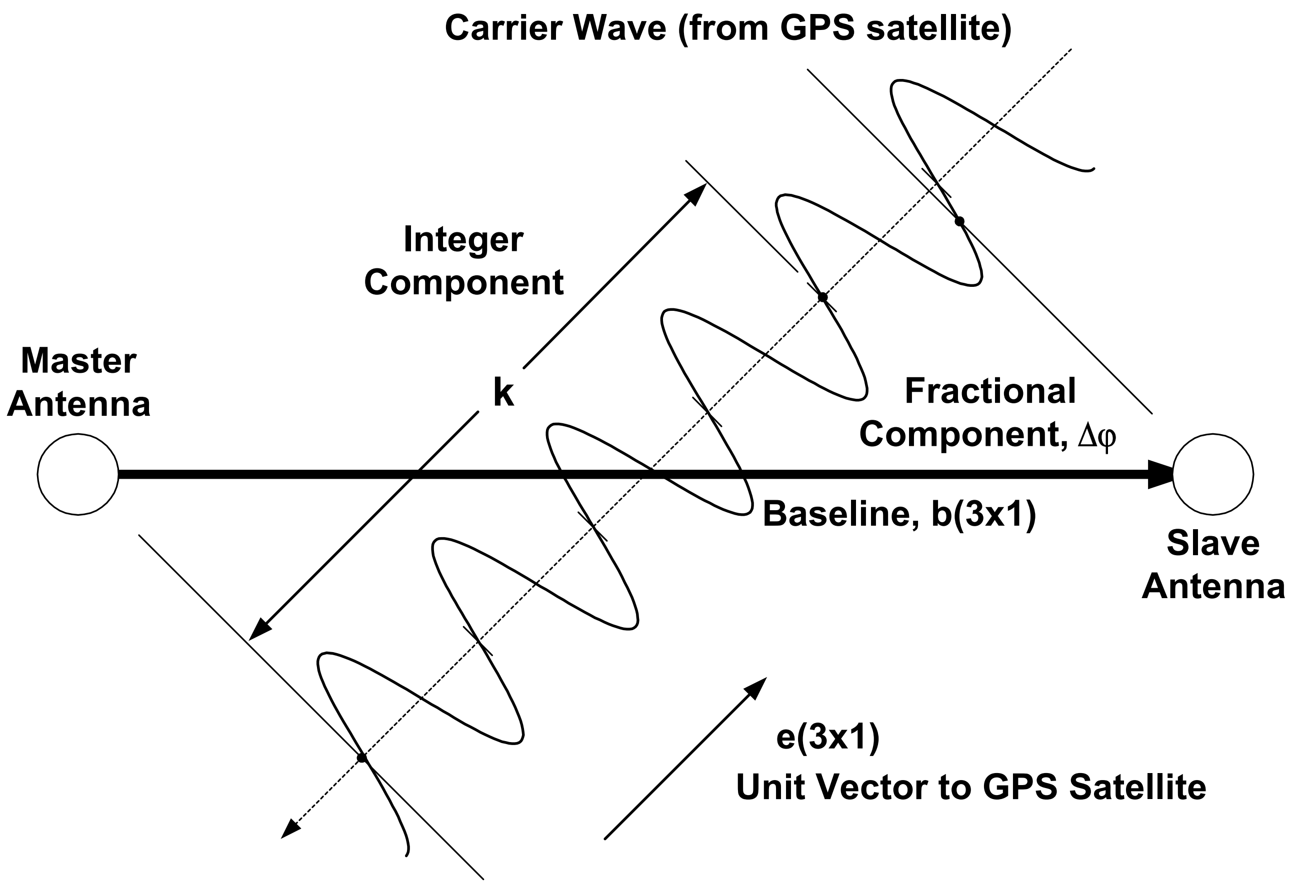
Advantages of GPS

- 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 - 1º RMS error)
- Low bandwidth (5-10 Hz maximum)
- Requires direct view of satellites

Interferometric Principle



Interferometric Principle

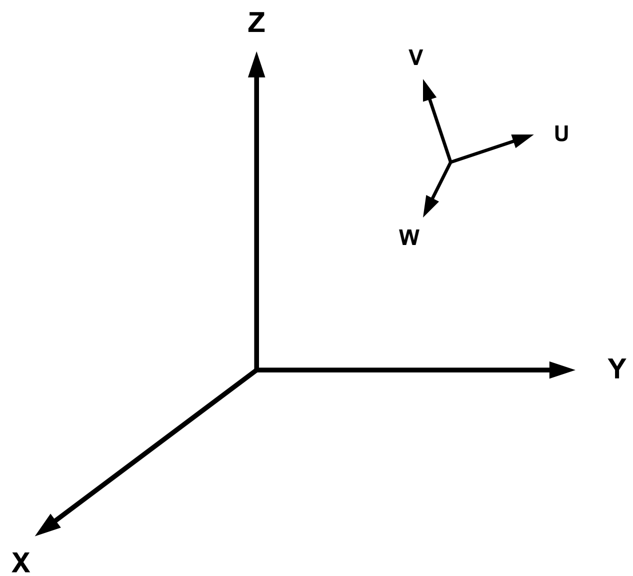
Measurement equation:

$$\Delta\varphi = \Delta r - k + \beta + v$$

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

$$\Delta r = \mathbf{b} \cdot \mathbf{e} = |\mathbf{b}| \cos\theta$$

Attitude Matrix



9 parameters needed:

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

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

$$\mathbf{A} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

Properties of "A"

"A" rotates a vector from the reference system to the body system

$$\mathbf{a}^B = \mathbf{A}_R^B \, \mathbf{a}^R$$

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

$$\mathbf{A}^T \mathbf{A} = \mathbf{I}_{3\times 3}$$

Properties of “A”

Rotation does not change the size of the vectors:

$$\det \mathbf{A} = 1$$

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

$$\mathbf{A}\mathbf{e} = \mathbf{e}$$

The rotation-angle is the eigenvalue of “A”

Euler sequences

A sequence of rotations by the angles (ϕ,θ,ψ) about the coordinate axes of the reference system

Single axis:

$$\mathbf{A}_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

Multiple axes:

$$\mathbf{A}_{123}(\phi,\theta,\psi) = \mathbf{A}_3(\psi)\mathbf{A}_2(\theta)\mathbf{A}_1(\phi)$$

Quaternions

A quaternion consists of four composants

$$\boldsymbol{q} = q_4 + iq_1 + jq_2 + kq_3$$

Where i,j and k are hyperimaginary numbers

$$\begin{aligned}
i^2 &= \quad j^2 &&= \quad k^2 = -1 \\
ij &= -ji &&= \quad k \\
jk &= -kj &&= \quad i \\
ki &= -ik &&= \quad j
\end{aligned}$$

Quaternions

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

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix}$$

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$$

Quaternions

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

$$
\begin{aligned}
q_1 &= e_1 \sin \frac{\Phi}{2} \\
q_2 &= e_2 \sin \frac{\Phi}{2} \\
q_3 &= e_3 \sin \frac{\Phi}{2} \\
q_4 &= \cos \frac{\Phi}{2}
\end{aligned}
$$

Quaternions

Quaternions can be multiplied using the special operator $\boldsymbol{q}'' = \boldsymbol{q}' \otimes \boldsymbol{q}$ **defined as:**

$$\mathbf{A}(\boldsymbol{q}\boldsymbol{q}') = \mathbf{A}(\boldsymbol{q}')\mathbf{A}(\boldsymbol{q})$$

$$\begin{bmatrix} q_1'' \\ q_2'' \\ q_3'' \\ q_4'' \end{bmatrix} = \begin{bmatrix} q_4' & q_3' & -q_2' & q_1' \\ -q_3' & q_4' & q_1' & q_2' \\ q_2' & -q_1' & q_4' & q_3 \\ -q_1' & -q_2' & -q_3' & q_4' \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

Quaternions

The attitude matrix can be formed from the quaternion as:

$$\begin{aligned}\mathbf{A}(\boldsymbol{q}) &= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \\ &= (q_4^2 - |\mathbf{q}|^2)\mathbf{I}_{3\times 3} + 2\mathbf{q}\mathbf{q}^T - 2q_4\mathbf{Q}^\times\end{aligned}$$

Where

$$\mathbf{Q}^\times = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q2 & q_1 & 0 \end{bmatrix}$$

Least Squares Solution

Including attitude information into the measurement equation

$$\begin{aligned}\Delta r &= \mathbf{b} \cdot \mathbf{e} \\ &= (\mathbf{b}^B)^T \mathbf{A} \mathbf{e}^R \Rightarrow \\ \Delta \varphi &= (\mathbf{b}^B)^T \mathbf{A} \mathbf{e}^R - k + \beta + v\end{aligned}$$

Linearization of the attitude matrix

$$\mathbf{A} = \delta \mathbf{A} \hat{\mathbf{A}} = (\mathbf{I} - 2\mathbf{Q}^{\times})\hat{\mathbf{A}}$$

Least Squares Solution

Forming the phase residual

$$
\begin{aligned}
\Delta\varphi_{ij} &= (\mathbf{b}_j^B)^T(\hat{\mathbf{A}}\mathbf{e}_i^R) - (\mathbf{b}_j^B)^T(2\mathbf{Q}^\times\hat{\mathbf{A}}\mathbf{e}_i^R) - k_{ij} + \beta_j + v_{ij} \\
& \Downarrow \\
\delta\varphi_{ij} &= \Delta\varphi_{ij} - \Delta\hat{\varphi}_{ij} \\
&= -(\mathbf{b}_j^B)^T(2\mathbf{Q}^\times\hat{\mathbf{A}}\mathbf{e}_i^R) \\
&= -2(\hat{\mathbf{A}}\mathbf{e}_i^R)^T\mathbf{B}_j^\times\delta\mathbf{q}
\end{aligned}
$$

Least Squares Solution

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

$$\begin{aligned} \mathbf{H}\mathbf{x} &= \mathbf{z} \\ \mathbf{H}^T\mathbf{H}\mathbf{x} &= \mathbf{H}^T\mathbf{z} \\ \mathbf{x} &= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z} \end{aligned}$$

Least Squares Solution

Estimate update

$$\delta\hat{\boldsymbol{q}} = norm\left(\begin{bmatrix}\delta\hat{q}_1 \\ \delta\hat{q}_2 \\ \delta\hat{q}_3 \\ 1\end{bmatrix}\right)$$

$$\hat{\boldsymbol{q}} = \delta\hat{\boldsymbol{q}} \otimes \hat{\boldsymbol{q}}'$$

Extended Kalman Filter

A Kalman filter consists of a model equation

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

and a measurent equation

$$\mathbf{z}(t_k) = \mathbf{h}[\mathbf{x}(t_k), t_k] + \mathbf{v}(t_k), \quad k = 1, 2, \ldots \quad \mathbf{v}(t_k) \sim N(0, \mathbf{R}(t_k))$$

Extended Kalman Filter

And their linearized counterparts….

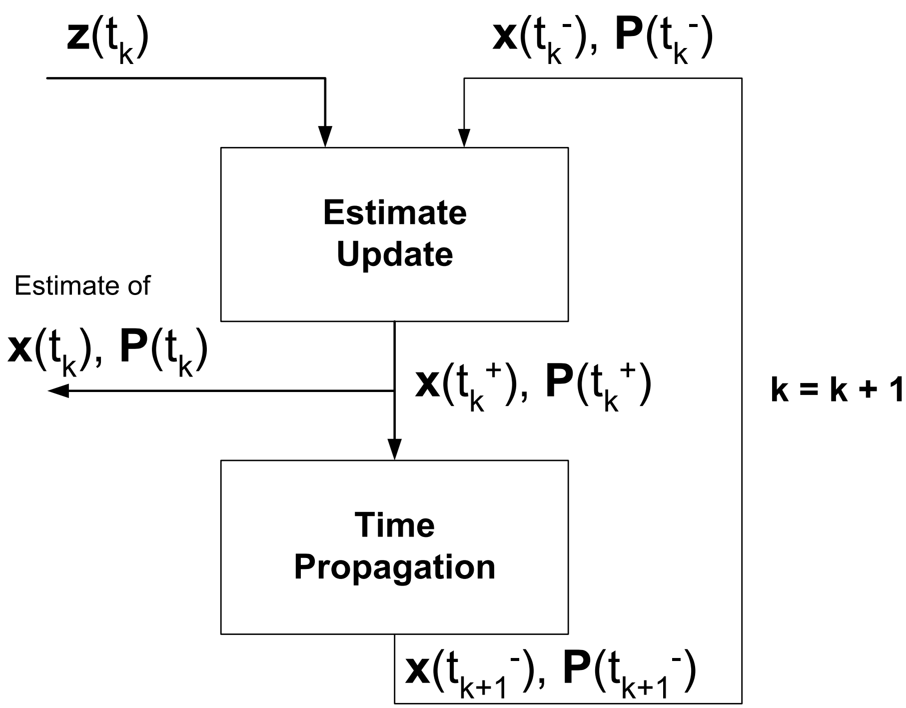
$$\mathbf{F}[\mathbf{x}_n(t_k), t_k] = \left.\frac{\partial \mathbf{f}[\mathbf{x}, t_k]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n(t_k)}$$

And

$$\mathbf{H}[\mathbf{x}_n(t_k), t_k] = \left.\frac{\partial \mathbf{h}[\mathbf{x}, t_k]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n(t_k)}$$

Extended Kalman Filter

Algorithm



$$
\begin{aligned}
\mathbf{K} &= \mathbf{P}\mathbf{H}^T\{\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}\}^{-1} \\
\delta\mathbf{x} &= \mathbf{K}\{\mathbf{z} - \mathbf{h}\} \\
\mathbf{P}^+ &= \mathbf{P}^- - \mathbf{K}\mathbf{H}\mathbf{P}^-
\end{aligned}
$$

$$
\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{f} \\
\dot{\mathbf{P}} &= \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{Q}
\end{aligned}
$$

Extended Kalman Filter

Tuning of the filter

$$\mathbf{R} = \begin{bmatrix} \sigma_R^2 & 0 & \dots & 0 \\ 0 & \sigma_R^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \sigma_R^2 \end{bmatrix} = \sigma_R^2 \mathbf{I}_{n \times n}$$

Noise variance determined experimentally

$$\sigma_R = 0.028\lambda \approx 0.5cm$$

Extended Kalman Filter

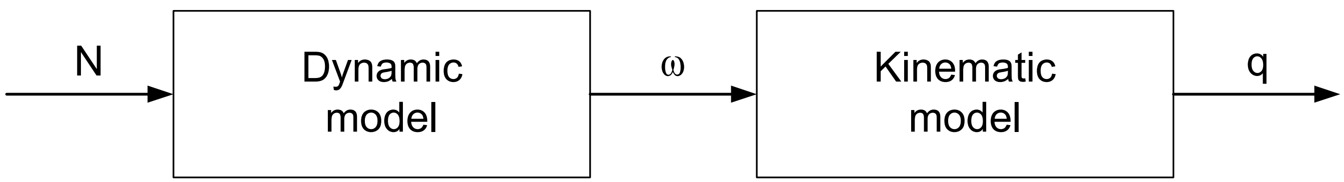
Tuning of the filter

$$\mathbf{Q} = \begin{bmatrix} \sigma_Q^2 & 0 & \dots & 0 \\ 0 & \sigma_Q^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \sigma_Q^2 \end{bmatrix} = \sigma_Q^2 \mathbf{I}_{n \times n}$$

Noise variance determined by ‘trial-and-error’

Extended Kalman Filter

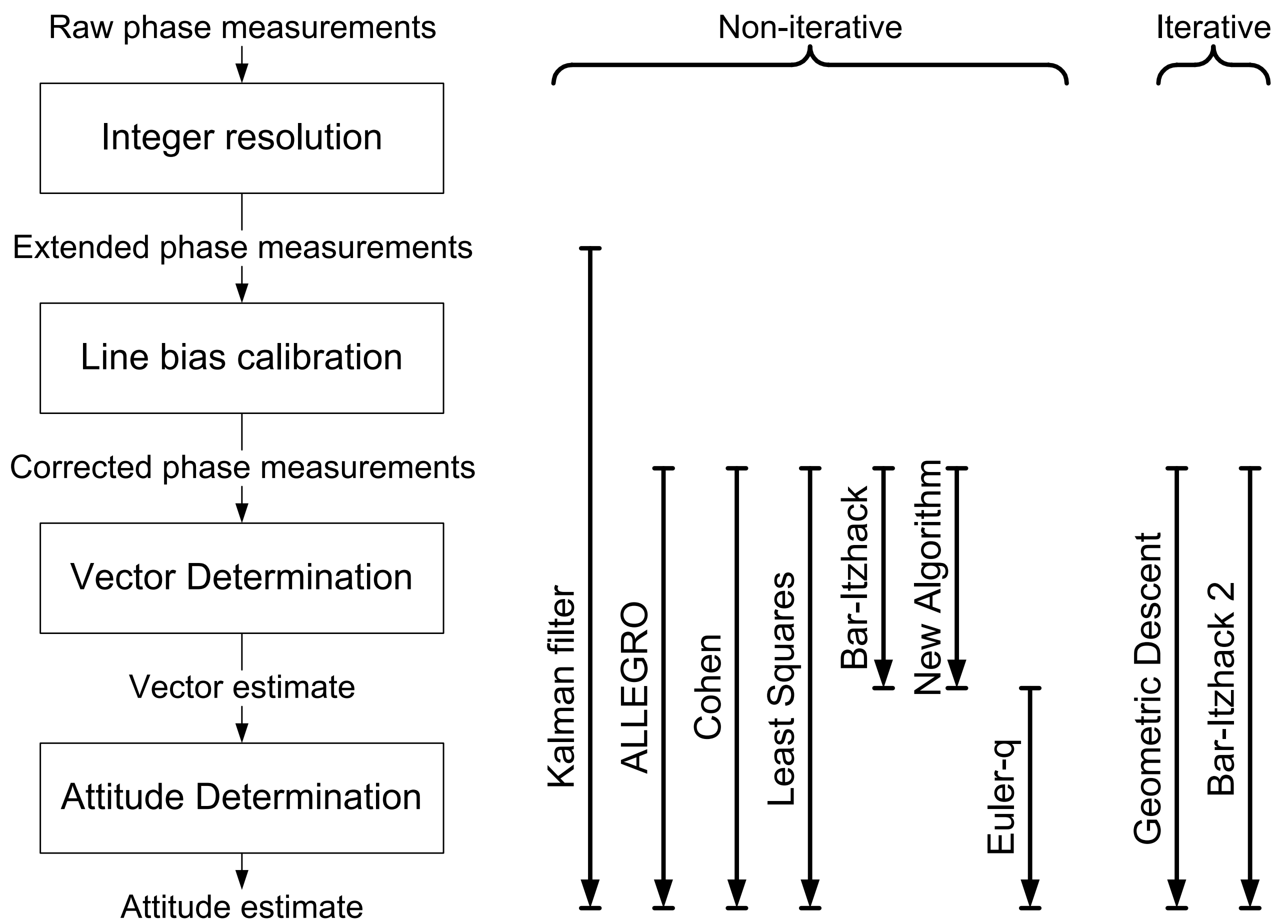
Determining the system model



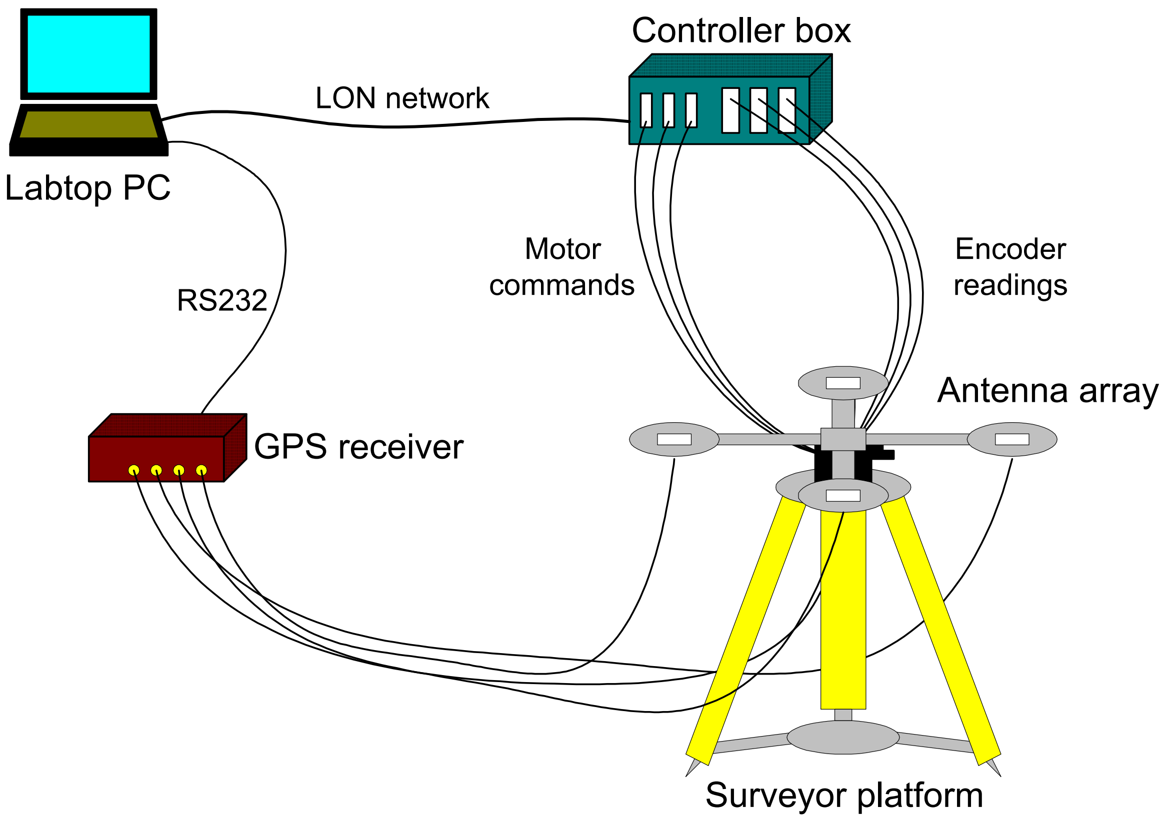
$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(-\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega})$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \boldsymbol{q}$$

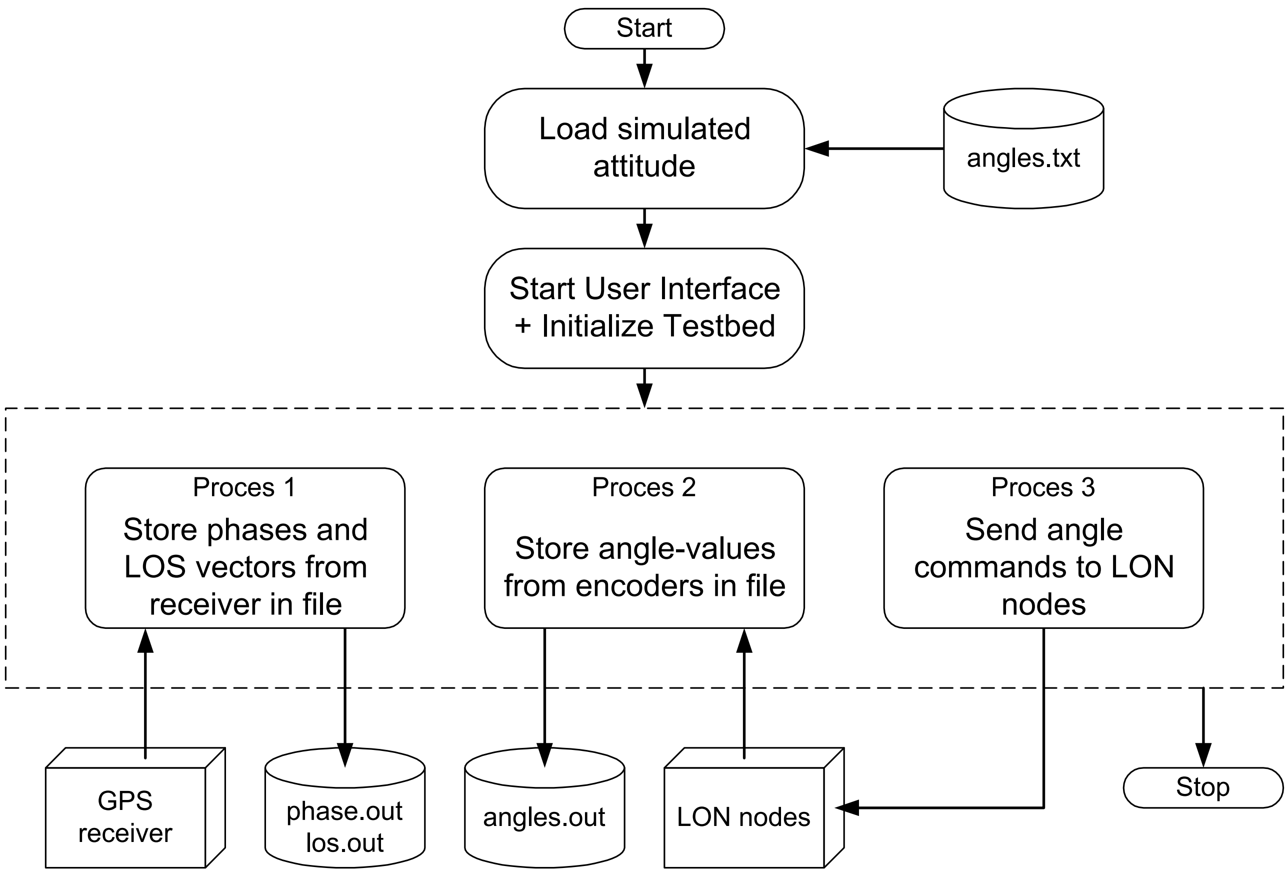
Other Filters



Testbed

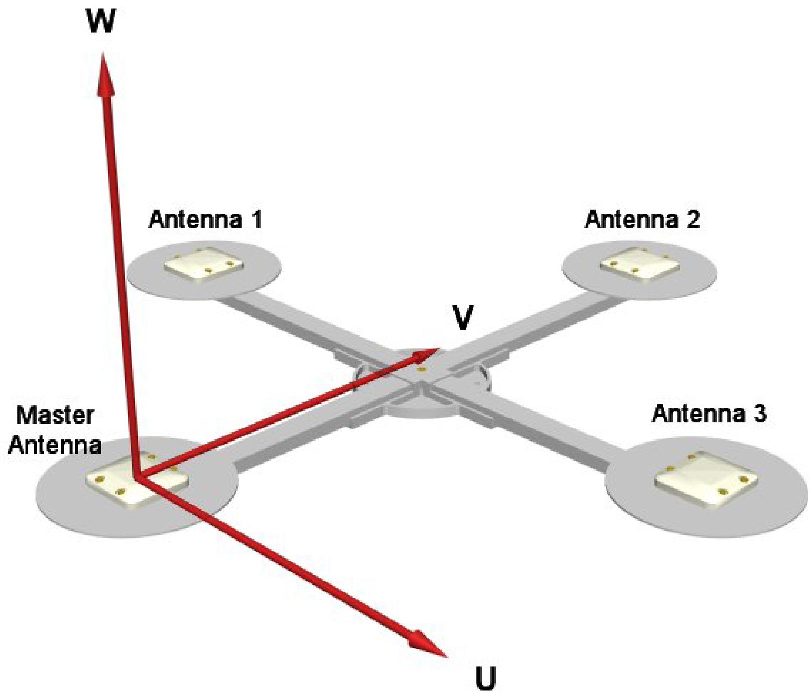

Controller box
LON network
Labtop PC
Motor
commands
Encoder
readings
RS232
Antenna array
GPS receiver
Surveyor platform

Software


Start
Load simulated attitude
angles.txt
Start User Interface + Initialize Testbed
Proces 1
Store phases and LOS vectors from receiver in file
Proces 2
Store angle-values from encoders in file
Proces 3
Send angle commands to LON nodes
GPS receiver
phase.out los.out
angles.out
LON nodes
Stop

Motor Control

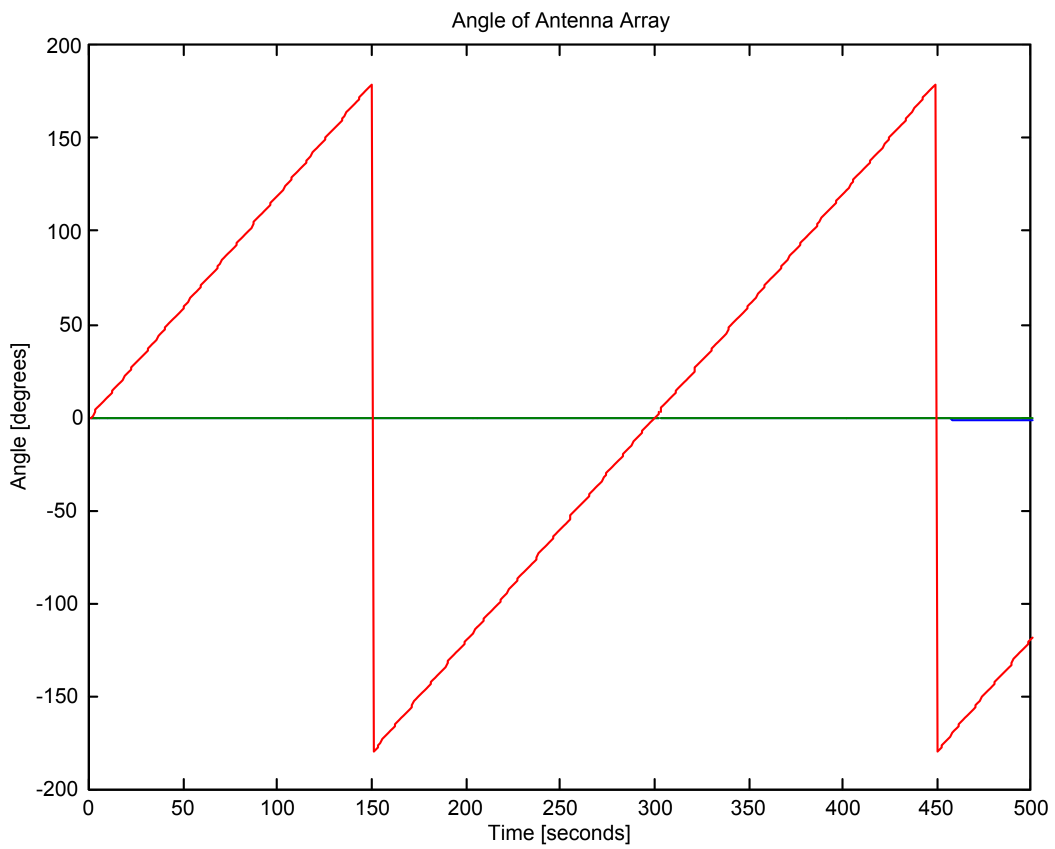
$$\mathbf{A}_{213} = \begin{bmatrix} \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \sin\psi\cos\theta & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\theta & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \cos\theta\sin\phi & -\sin\theta & \cos\theta\cos\phi \end{bmatrix}$$



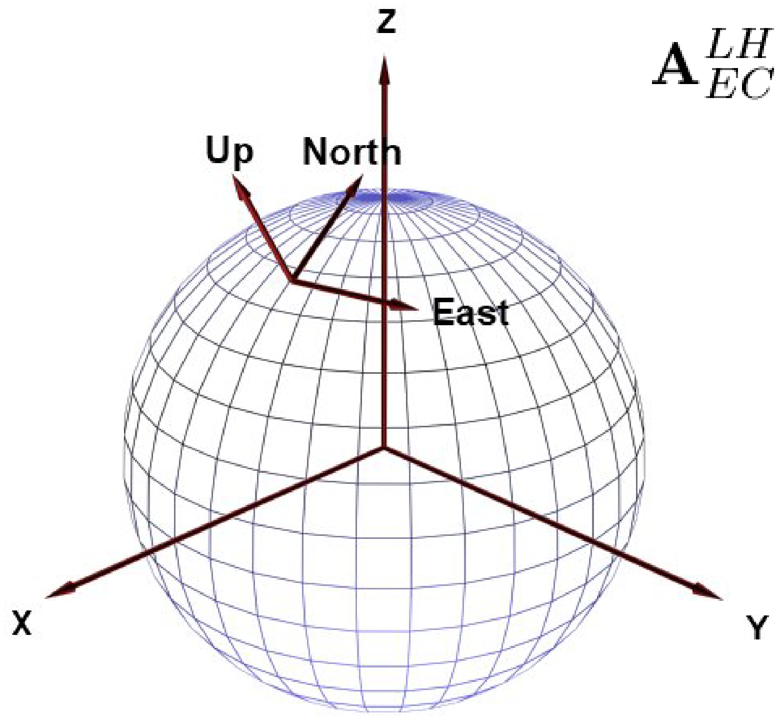
$$\Downarrow$$

$$\begin{aligned} \phi &= \arctan(A_{31}/A_{33}) \\ \theta &= \arcsin(A_{32}) \\ \psi &= \arctan(A_{12}/A_{22}) \end{aligned}$$

Motor Angles



Local Horizontal System



$$\mathbf{A}_{ECEF}^{LH} = \mathbf{A}_{shift} \cdot \mathbf{A}_2(-el) \cdot \mathbf{A}_3(az)$$

$$\mathbf{A}_{shift} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Results

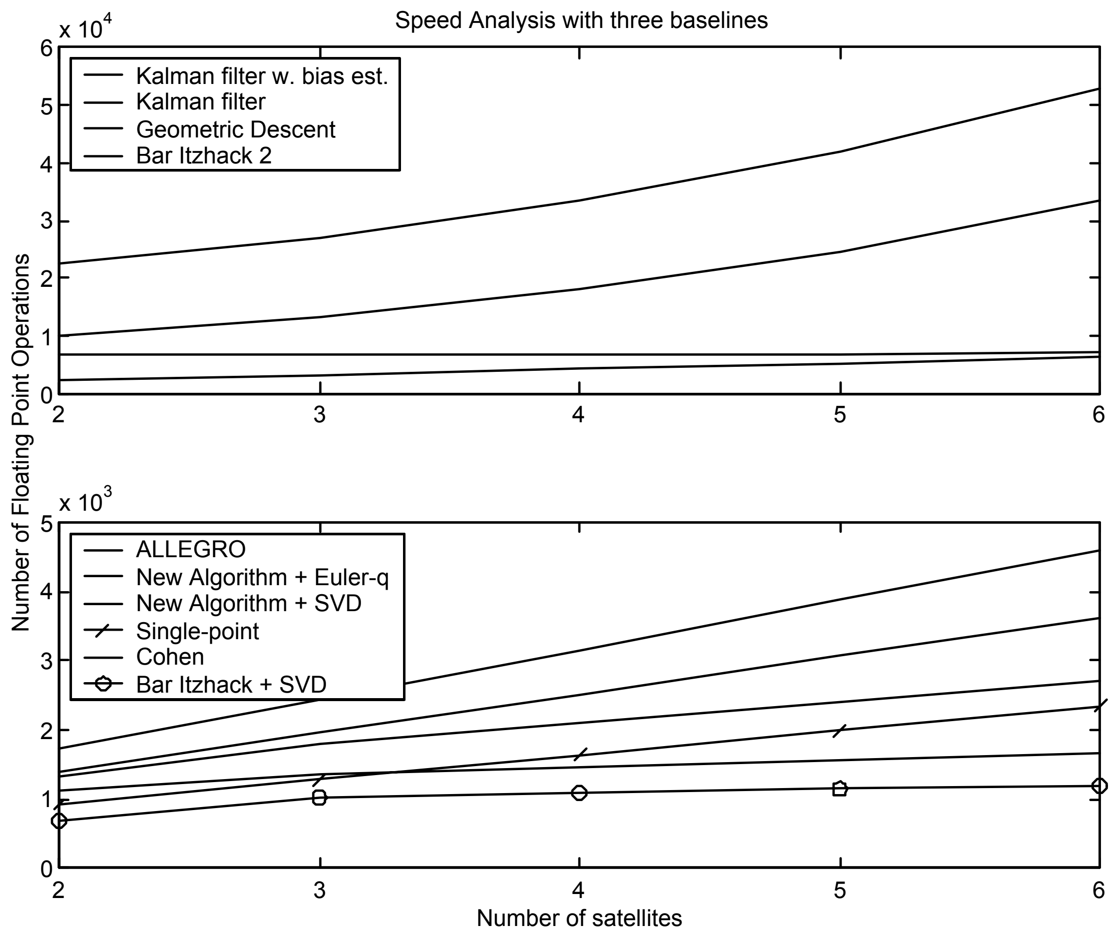
Based on actual and simulated data, the following performance parameters were evaluated

- **Accuracy**
- **Computational efficiency**
- **Ability to converge**

Accuracy

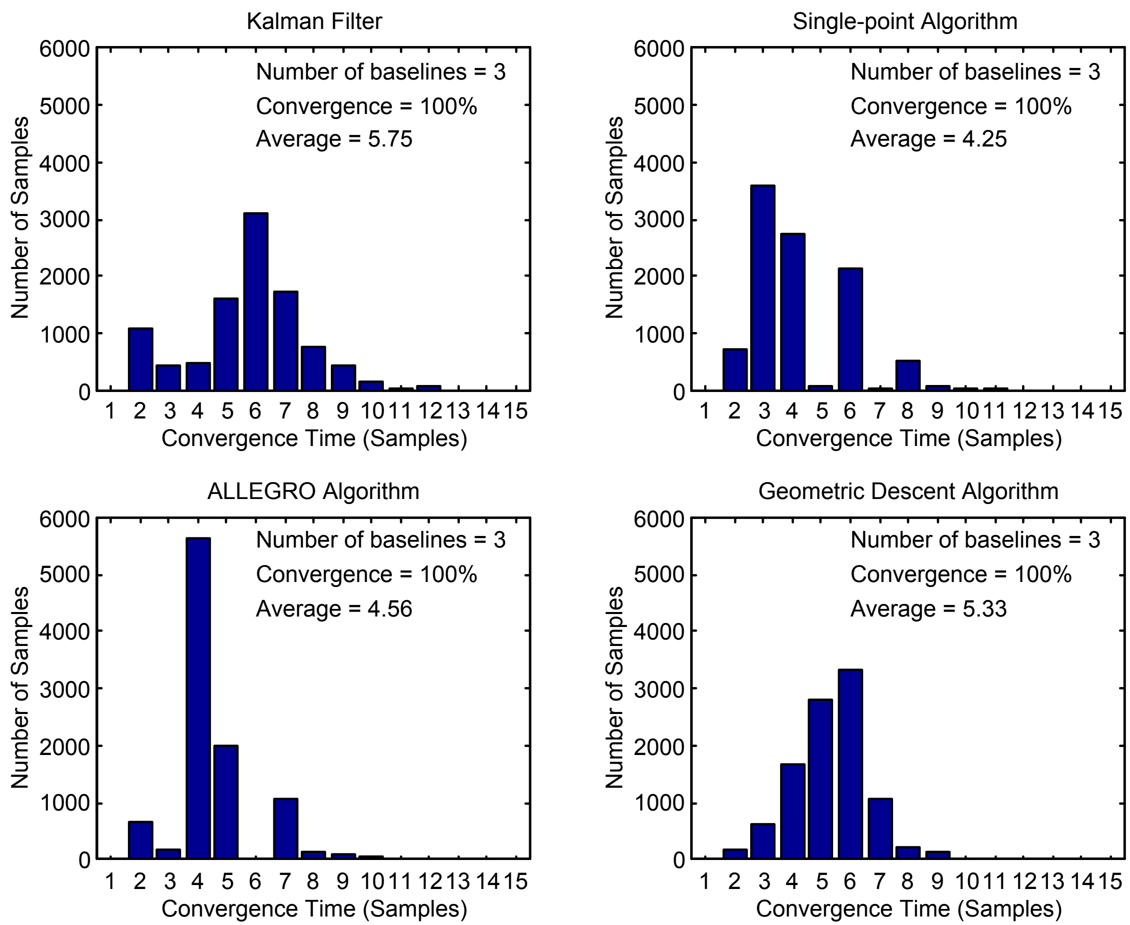
Algorithm	RSS error in degrees	
	2 coplanar	3 coplanar
Kalman filter	0.1756	0.1513
Single-point	0.4933	0.3779
ALLEGRO	0.4936	0.3682
Geometric descent	(Does not converge)	
New Algorithm + SVD	0.5366	0.4043
Bar-Itzhack + SVD	0.5282	(Undefined)
Cohen	0.5320	0.4102
Euler-q	0.5412	0.4084

Speed

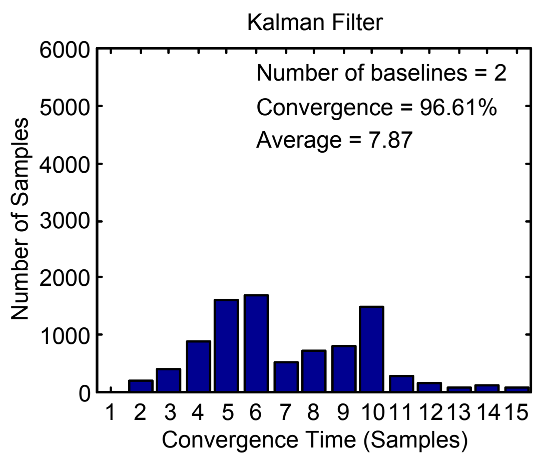


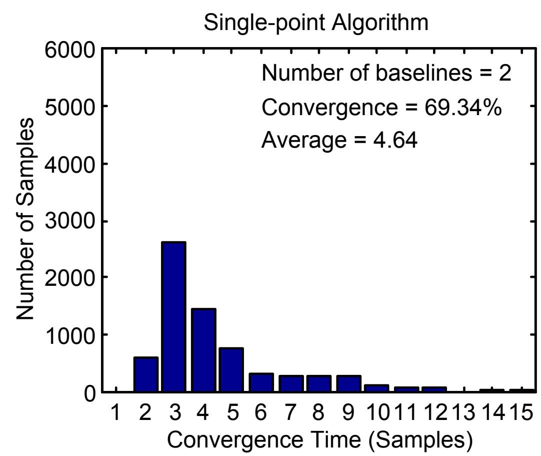
Speed Analysis with three baselines
x 10^4
6
5
4
3
2
1
0
Kalman filter w. bias est.
Kalman filter
Geometric Descent
Bar Itzhack 2
2
3
4
5
6
Number of Floating Point Operations
x 10^3
5
4
3
2
1
0
ALLEGRO
New Algorithm + Euler-q
New Algorithm + SVD
Single-point
Cohen
Bar Itzhack + SVD
2
3
4
5
6
Number of satellites

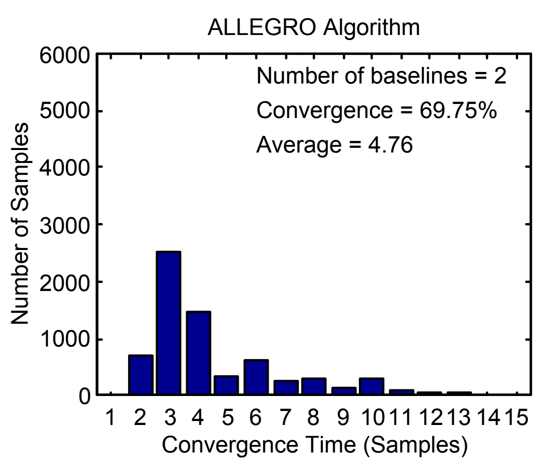
Convergence



Convergence







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