Integrated navigation examples

Lorenzo Pollini

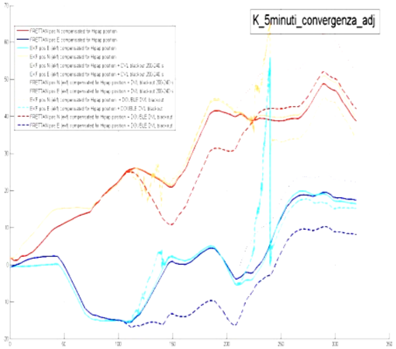
Dept. Electrical Systems and Automation

University of Pisa – Italy

LPOLLINI@DSEA.UNIPI.IT

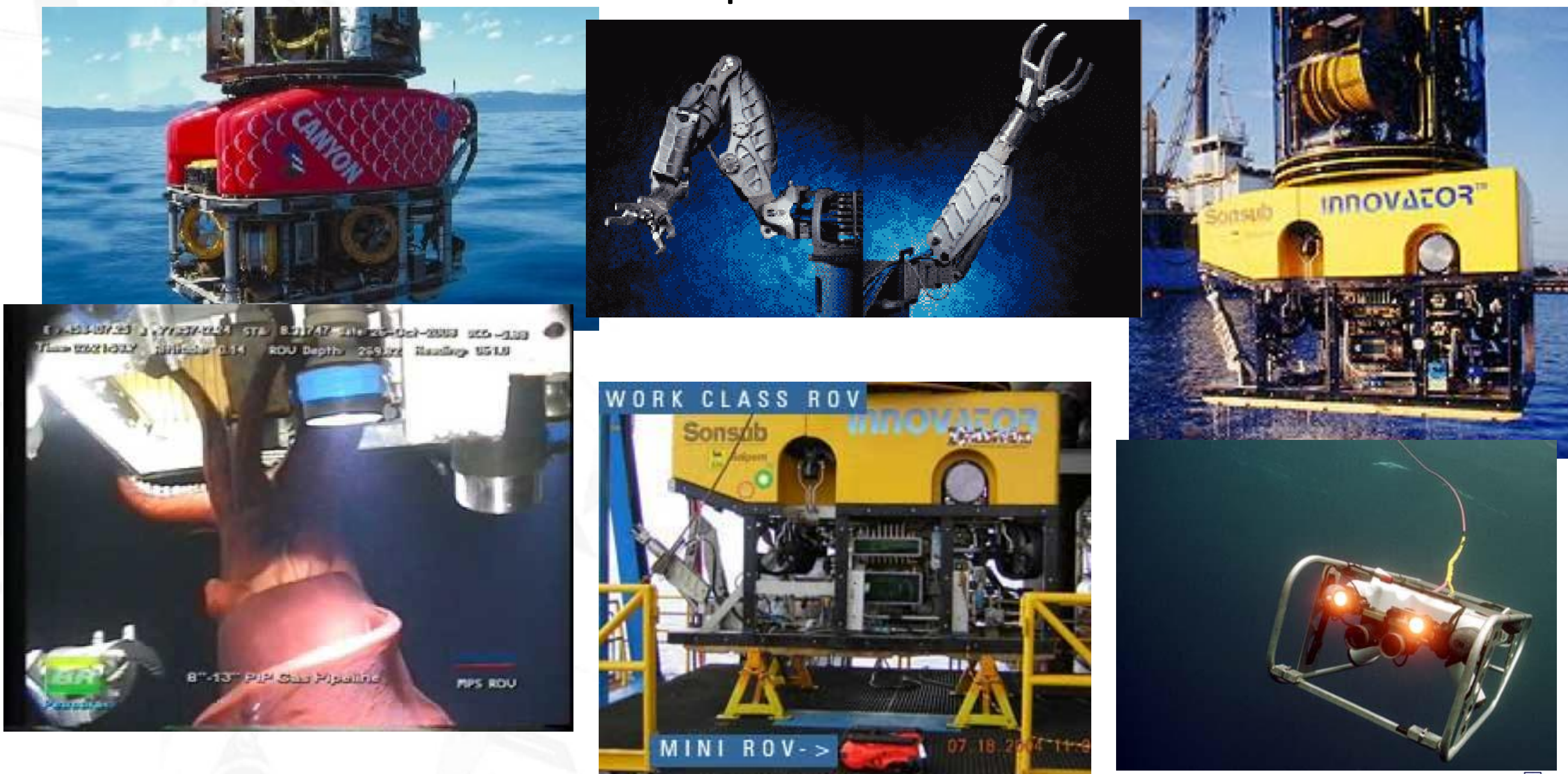
Case Study: Remotely Operated Vehicles

- What is a ROV
- The need for estimation
- Sensors for underwater navigation
- Indirect Extended Kalman Filter design
- Simulations
- Experimental data



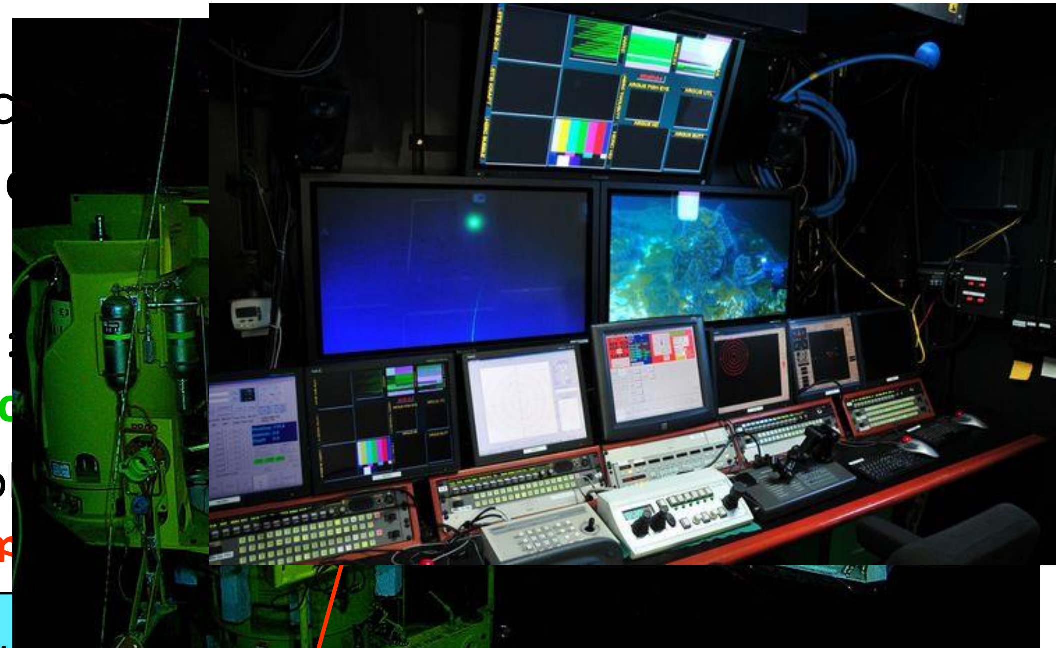
Case Study : Remotely Operated Vehicles

- Remotely Operated Vehicles (ROVs) are widely used in underwater intervention operational scenarios

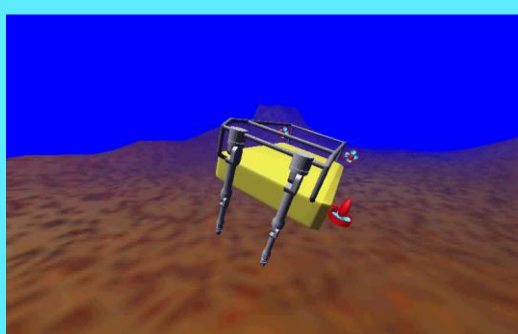


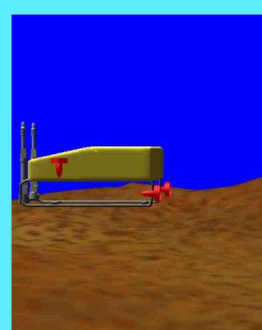
Case Study : Remotely Operated Vehicles

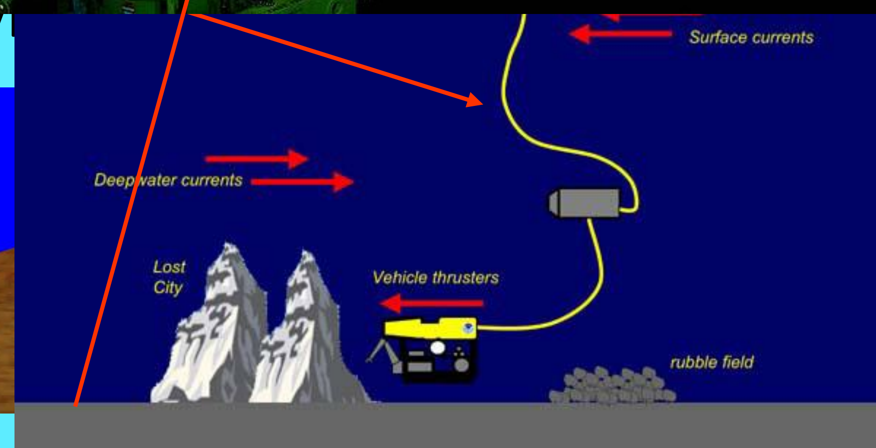
- Remotely Operated Vehic
  underwater intervention
  - Depth : 1000s of meters
  - Datalink with mother ship :
    the umbilical **=> communic**
  - Tools: task dependent + ro
    coordination between arm



Simulations of ROV/Arms/







Sensor models : Gyros

Gyroscopes

True-North AHRS (already used for attitude control)

- Fiber Optic Gyrocompass and AHRS
- Extremely sensitive to angular rates
- Provides true North
- Provides tilt angles (roll/pitch) using inclinometers

Usage:

- Measure roll/pitch/yaw rates from FOGs
- Measure Heading (yaw) angle (valid after 30-45’)

  $\Rightarrow$ Obtain roll/pitch angles via inertial mechanization (INS)

MfS: slowly varying gyro drift, white/colored noise, 32Hz, 16 bit quantization

MfSF: constant gyro bias, periodic sampling, white noise

Model for Sensor Fusion : for Navigation system development

Model for Simulation : for Validation

Sensor models: Accelerometers

Accelerometers

Inertial package: **custom built using COTS sensors**

- Sensor characteristics
  - Outputs accelerations at 4 Khz
  - Sampled, and anti-alias filtered in software

Usage:

- Measure X/Y/Z ROV accelerations in body frame

  $\Rightarrow$ Obtain ROV velocity and position in navigation frame via inertial mechanization (INS)

MfS: slowly varying drift, white/colored noise, 32Hz, 16 bit quantization

MfSF: constant bias, periodic sampling, white noise

Sensor models : Velocity Vector Measurement

Doppler Velocity Log

DVL: **Workhorse Navigator**
Measures:
- Bottom and Water Track Velocity
  - Sensor frame / inertial frame
- Heading (magnetic?) and Tilt (roll/pitch)
- Altitude (w.r.t. sea bottom), Pressure and depth

Usage:
- Measure bottom track velocity **in sensor frame**
  $\Rightarrow$ Transform from sensor to inertial frame via roll/pitch/heading angles computed by INS and use to correct inertial mechanization position drift

MfS: slowly varying drift, white/colored noise, 7Hz, 16 bit quantization
MfSF: constant bias, sporadic sampling, white noise (variable with speed)

Sensor models: depth

Pressure Depth Sensor

COTS sensor: **Paroscientific Digiquartz**

- Measures depth with extreme accuracy and resolution
- Update rate and mode of operation (to be defined)

Usage:

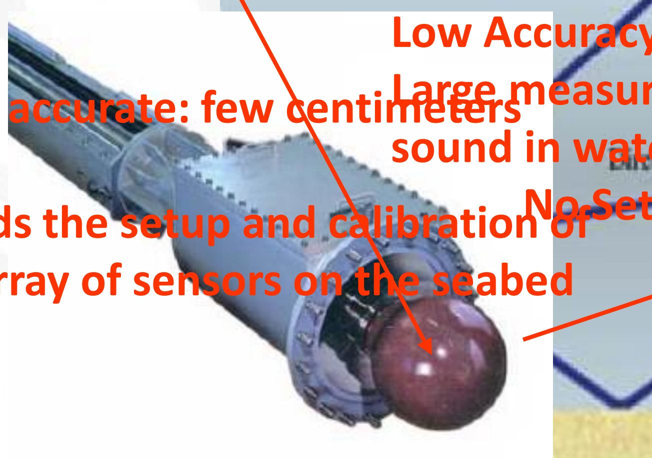
- Measure Z position in vessel/navigation frame
  $\Rightarrow$ Use to correct inertial mechanization vertical position drift

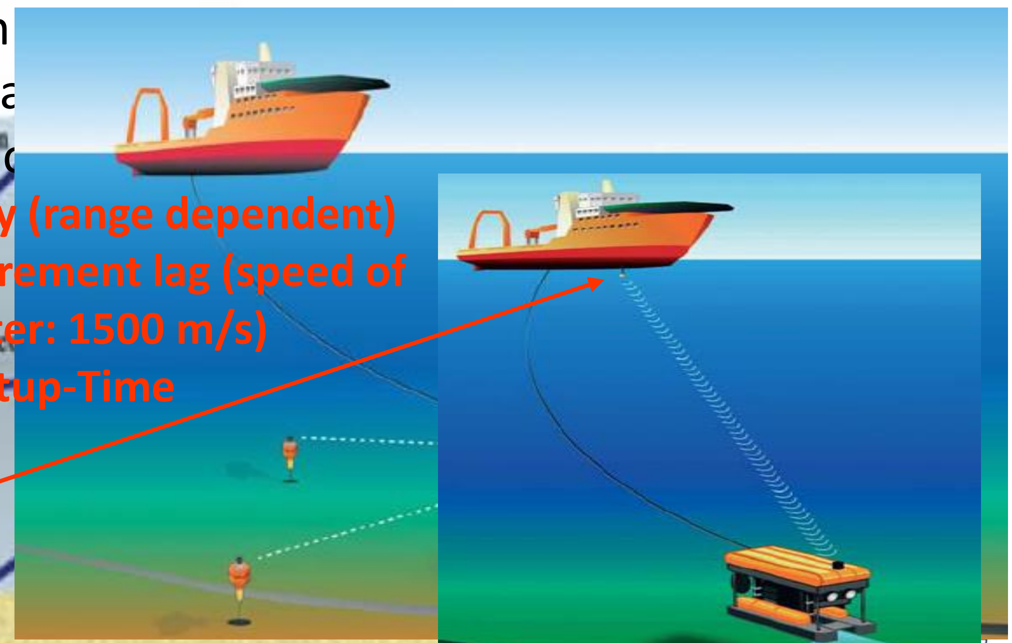
MfS: white/colored noise, Sampling 1Hz, quantization (1 mm)

MfSF: periodic sampling, (extremely low) white noise

Sensor models: Acoustic Positioning

- GPS does not work underwater!
- Ultrasound waves transmit well and fast in water => may be used for:
  - Underwater Communications
  - Underwater Localization/self-localization by triangulation of Time-of-Flight data (as in GPS) => Long Baseline Localization (LBL)
  - Underwater Localization
    range, azimuth and eleva
    Short Baseline Localizatio



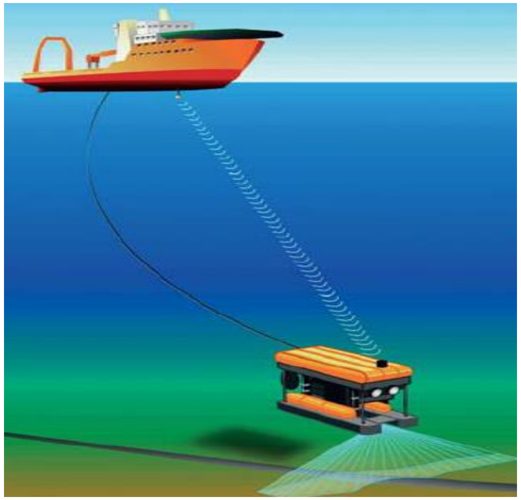


Low Accuracy (range dependent)
Large measurement lag (speed of sound in water: 1500 m/s)
No Setup-Time

Very accurate: few centimeters
Needs the setup and calibration of an array of sensors on the seabed

Sensor models: Absolute Positioning

Ultra Short Baseline Localization System



COTS sensor: **Kongsberg HiPAP**

- Measures 3D position relative to the vessel
- Accuracy decreases with depth
- Update rate and mode of operation (to be defined)

Usage:

- Measure X/Y position in vessel/navigation frame
  $\Rightarrow$ Use to correct inertial mechanization horizontal position drift

MfS: slowly varying DGPS-induced drift, white/colored noise (variable with depth), Sampling: 1Hz, No quantization

MfSF: periodic sampling, (large) white noise (variable with depth

Sensor Fusion Algorithm

INS with Strapdown Technology:

- Gyroscopes
  - High frequency measurement (KHz) is possible

on

Da

/ no

Sensor fusion

"Exploit the best from every available measurement to reduce navigation errors"

$\Downarrow$

Use noisy non-drifting velocity and position direct measurements from DVL, USBL and DPS to estimate biases in inertial sensors and improve INS accuracy

$\Downarrow$

Mathematical tool: the **Kalman Filter**

- Direct measurement : very low frequency / extremely noisy / no bias / **no drift**

The Indirect Kalman Filter

- Does not estimate navigation states but **their errors**
- Does not depend on vehicle dynamics but on **sensor models**
- State variables are **navigation errors** and inputs are **gyro and accelerometer measurement errors**

State and error variables

$$
\begin{aligned}
\hat{R}_b^n &= (I_{3\times 3} - \Gamma^n)\, R_b^n \\
\hat{V}^n &= V^n + \delta V^n \\
\hat{\phi} &= \phi + \delta\phi \\
\hat{\lambda} &= \lambda + \delta\lambda \\
\hat{h} &= h + \delta h \\
\tilde{\omega}_{ib}^b &= \omega_{ib}^b + \delta\omega_{ib}^b \\
\tilde{f}_{ib}^b &= f_{ib}^b + \delta f_{ib}^b
\end{aligned}
$$

Attitude error equation :

$$
\begin{aligned}
\dot{\gamma}^n &= -\omega_{in}^n \wedge \gamma^n + \delta\omega_{in}^n - R_b^n\, \delta\omega_{ib}^b \\
\gamma^n &= \begin{bmatrix} \delta\theta_r & \delta\theta_p & \delta\theta_y \end{bmatrix} \\
\Gamma^n &= \gamma^n \wedge
\end{aligned}
$$

Velocity error equation :

$$
\delta\dot{V}^n = f_{ib}^n \wedge \gamma^n + F_{vv}\, \delta V^n + F_{vr}\, \delta r^n + R_b^n\, \delta f_{ib}^b - \begin{bmatrix} 0 \\ 0 \\ \delta g \end{bmatrix}
$$

Position error equation :

$$
\delta\dot{r}^n = F_{rv}\, \delta V^n + F_{rr}\, \delta r^n
$$

9-states error dynamics in compact form

$$
\begin{bmatrix} \dot{\gamma}^n \\ \delta\dot{V}^n \\ \delta\dot{r}^n \end{bmatrix} = \begin{bmatrix} -\omega_{in}^n \wedge & F_{\gamma v} & F_{\gamma r} \\ f_{ib}^n \wedge & F_{vv} & F_{vr} \\ 0 & F_{rv} & F_{rr} \end{bmatrix} \begin{bmatrix} \gamma^n \\ \delta V^n \\ \delta r^n \end{bmatrix} + \begin{bmatrix} -R_b^n & 0 \\ 0 & R_b^n \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\omega_{ib}^b \\ \delta f_{ib}^b \end{bmatrix}
$$

The Indirect Kalman Filter

- Gyroscopes and accelerometer measurement errors: biases and white noise

$$\begin{aligned} \delta\boldsymbol{\omega}_{ib}^{b} &= \delta\boldsymbol{\omega}_{ib,b}^{b} + \boldsymbol{w}_{\omega} & \boldsymbol{w}_{\omega} &\sim N\left(0, \boldsymbol{Q}_{\omega}\right) \\ \delta\boldsymbol{f}_{ib}^{b} &= \delta\boldsymbol{f}_{ib,b}^{b} + \boldsymbol{w}_{f} & \boldsymbol{w}_{f} &\sim N\left(0, \boldsymbol{Q}_{f}\right) \end{aligned}$$

- Biases can be considered as slowly varying with respect to system dynamics

$$\begin{aligned} \delta\dot{\boldsymbol{\omega}}_{ib,b}^{b} &= 0 \\ \delta\dot{\boldsymbol{f}}_{ib,b}^{b} &= 0 \end{aligned}$$

- Biases dummy dynamics can be incorporated into filter dynamics

$$\begin{bmatrix} \dot{\boldsymbol{\gamma}}^{n} \\ \delta\dot{\boldsymbol{V}}^{n} \\ \delta\dot{\boldsymbol{r}}^{n} \\ \delta\dot{\boldsymbol{\omega}}_{ib,b}^{b} \\ \delta\dot{\boldsymbol{f}}_{ib,b}^{b} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\omega}_{in}^{n}\wedge & \boldsymbol{F}_{\gamma v} & \boldsymbol{F}_{\gamma r} & -\boldsymbol{R}_{b}^{n} & 0 \\ \boldsymbol{f}_{ib}^{n}\wedge & \boldsymbol{F}_{vv} & \boldsymbol{F}_{vr} & 0 & \boldsymbol{R}_{b}^{n} \\ 0 & \boldsymbol{F}_{rv} & \boldsymbol{F}_{rr} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}^{n} \\ \delta\boldsymbol{V}^{n} \\ \delta\boldsymbol{r}^{n} \\ \delta\boldsymbol{\omega}_{ib,b}^{b} \\ \delta\boldsymbol{f}_{ib,b}^{b} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{R}_{b}^{n} & 0 & 0 & 0 \\ 0 & \boldsymbol{R}_{b}^{n} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\boldsymbol{R}_{b}^{n} & 0 \\ 0 & 0 & 0 & \boldsymbol{R}_{b}^{n} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_{\omega} \\ \boldsymbol{w}_{f} \\ \boldsymbol{w}_{r\omega} \\ \boldsymbol{w}_{rf} \end{bmatrix}$$

$$\delta\dot{\boldsymbol{x}}\left(t\right) = \boldsymbol{F}\left(t\right)\,\delta\boldsymbol{x}\left(t\right) + \boldsymbol{G}\left(t\right)\,\boldsymbol{u}_{c}\left(t\right)$$

The Indirect Kalman Filter

- Aiding sensors form the measurement (output) equation
- Aiding sensors systemic errors are modeled as additional state variables: **PDS bias**, **DVL bias**, and **Heading bias**.
- Measurement noise: **PDS noise, DVL noise, Heading noise, USBL noise**.

Model of navigation error *Model of measurement error*

Depth measurement (PDS) →

$$
\begin{aligned}
\hat{d} - d_{meas} &= (d + \delta d) - (d + d_b + v_d) \\
&= \delta d - d_b - v_d
\end{aligned}
$$

Velocity measurement (DVL) →

$$
\begin{aligned}
\hat{\boldsymbol{V}}^n - \boldsymbol{V}^n_{meas} &= (\boldsymbol{V}^n + \delta \boldsymbol{V}^n) - \hat{\boldsymbol{R}}^n_b \left(\boldsymbol{V}^b_{DVL} + \boldsymbol{V}^b_{DVL,b} + \boldsymbol{v}_{DVL}\right) \\
&= \delta \boldsymbol{V}^n - \boldsymbol{V}^n \wedge \boldsymbol{\gamma} - \hat{\boldsymbol{R}}^n_b \boldsymbol{V}^b_{DVL,b} - \hat{\boldsymbol{R}}^n_b \boldsymbol{v}_{DVL}
\end{aligned}
$$

Heading measure. (AHRS) →

$$
\hat{\theta}_y - \theta_{y,meas} = -\delta\theta_y - \theta_{y,b} - v_{\theta_y}
$$

Position Measurement (HiPAP) →

$$
\begin{aligned}
\hat{X}_{north} - X_{north,meas} &= (X_{north} + \delta X_{north}) - (X_{north} + v_{X_{north}}) \\
&= \delta X_{North} - v_{X_{north}} \\
\hat{Y}_{east} - Y_{east,meas} &= (Y_{east} + \delta Y_{east}) - (Y_{east} + v_{Y_{east}}) \\
&= \delta Y_{east} - v_{Y_{east}}
\end{aligned}
$$

The Indirect Kalman Filter

20-states non-linear navigation error and sensor biases dynamics

Dynamics is non linear

$\Downarrow$

The Filter is an Indirect Extended Kalman Filter (EKF)

$\Downarrow$

In order to implement the filter in real-time, error dynamics must be time-discretized.

INS mechanization

$$\boldsymbol{x}_{k+1} = f\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) + \boldsymbol{u}_k^{'} \qquad \boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{\Theta}_k \\ \boldsymbol{V}_k^n \\ \boldsymbol{r}_k^n \end{bmatrix} \qquad \boldsymbol{u}_k = \begin{bmatrix} \boldsymbol{\omega}_{ib,k}^b \\ \boldsymbol{f}_{ib,k}^b \end{bmatrix}$$

Filter states

$$\delta\boldsymbol{x}_{k+1} = \boldsymbol{\Psi}_k\, \delta\boldsymbol{x}_k + \boldsymbol{u}_{d,k}$$

Measurements

$$\delta\boldsymbol{y}_k = \boldsymbol{H}_k\, \delta\boldsymbol{x}_k + \boldsymbol{J}_k\, \boldsymbol{v}_{d,k}$$

$$\delta\dot{\boldsymbol{x}}\left(t\right) = \boldsymbol{F}\left(t\right)\, \delta\boldsymbol{x}\left(t\right) + \boldsymbol{G}\left(t\right)\, \boldsymbol{u}_c\left(t\right)$$

The Indirect Kalman Filter

First compensate inertial sensor for biases and perform exact INS mechanization f

- Inertial Mechanization and measurement Covariance update happens at the fastest sampling time (that of Inertial sensors).

Navigation errors, first 9 states, can be estimated only if a aiding fix is available, and then reset

$$
\begin{aligned}
\hat{\boldsymbol{u}}_k^- &= \tilde{\boldsymbol{u}}_k + \left[\delta \hat{\boldsymbol{x}}_k^-\right]_{10:15} \\
\hat{\boldsymbol{x}}_{k+1}^- &= f\left(\hat{\boldsymbol{x}}_k^-, \hat{\boldsymbol{u}}_k^-\right) \\
\left[\delta \hat{\boldsymbol{x}}_{k+1}^-\right]_{1:9} &= 0 \\
\left[\delta \hat{\boldsymbol{x}}_{k+1}^-\right]_{10:20} &= \left[\Psi_k\right]_{(10:20,10:20)} \left[\delta \hat{\boldsymbol{x}}_k^-\right]_{10:20} \\
\boldsymbol{P}_{k+1}^- &= \boldsymbol{\Psi}_k \boldsymbol{P}_k^- \boldsymbol{\Psi}_k^T + \boldsymbol{Q}_{d,k}
\end{aligned}
$$

Sensor errors are kept constant

Knowledge of navigation and sensor errors worsen in time without fixes

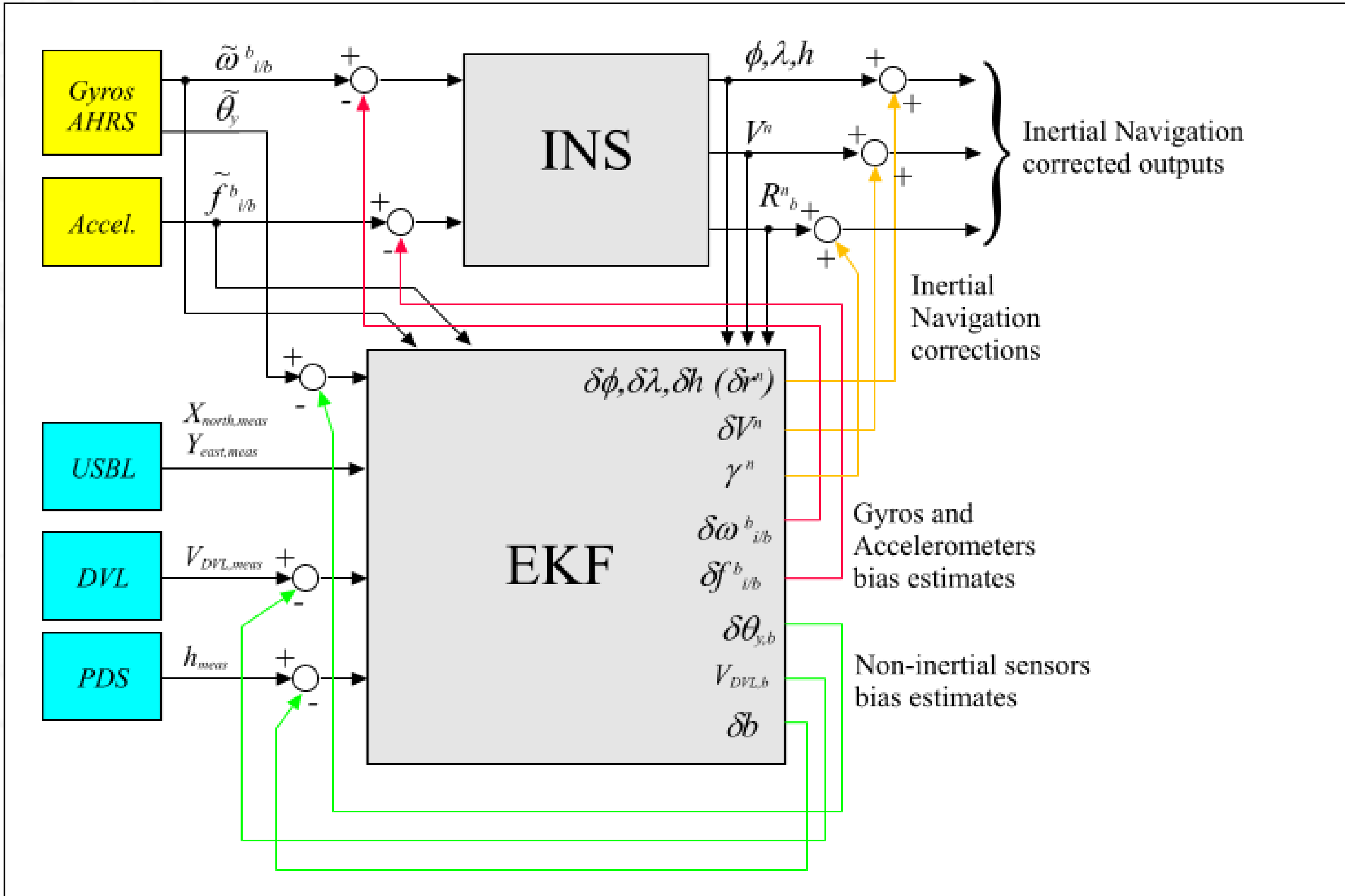
- When new measurements are available (from DVL, USBL and PDS) update estimates of biases and navigation errors, and correct navigation output

$$
\begin{aligned}
\boldsymbol{K}_k &= \boldsymbol{P}_k^- \boldsymbol{H}_k^T \left(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{J}_k \boldsymbol{R}_k \boldsymbol{J}_k^T\right)^{-1} \\
\boldsymbol{R}_k &= diag\left(R_d \quad \boldsymbol{R}_{DVL} \quad R_{\theta_y} \quad \boldsymbol{R}_{XY} \right) / \Delta T \\
\hat{\boldsymbol{y}}_k &= \tilde{\boldsymbol{y}}_k + \left[\delta \hat{\boldsymbol{x}}_k^-\right]_{16:20} \\
\delta \hat{\boldsymbol{x}}_k &= \delta \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \left(\hat{\boldsymbol{y}}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-\right) \\
\hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}_k^- + \delta \hat{\boldsymbol{x}}_k \\
\boldsymbol{P}_k &= \left(\boldsymbol{I}_{20\times 20} - \boldsymbol{K}_k \boldsymbol{H}_k\right) \boldsymbol{P}_k^-
\end{aligned}
$$

Measurement are corrected according to current sensor error estimates

in case of a aiding fix, compute navigation and sensor error and correct estimate

The Indirect Kalman Filter



Critical Design Review

- Early simulations provided very good results
- The Pressure Depth Sensor is extremely accurate
  - An eventual bias is difficult to remove
    - Hipap depth measures are more noisy
    - DVL vertical velocity is useless to detect a bias
  - Thus the bias was removed from EKF state vector
- A better tuning was performed (even though tuning is required on the field with real data)
- The innovation process was redesigned to have a variable structure
  - The Filter is run at the frequency of the fastest sensor
  - The output matrix $\boldsymbol{H}_k$ is varied accordingly to the measurements available at each time step

The Indirect Kalman Filter

- Aiding sensors form the measurement (output) equation
- Aiding sensors systemic errors are modeled as additional state variables: **DVL bias**, and **Heading bias**.
- Measurement noise: **PDS noise, DVL noise, Heading noise, USBL noise**.

Depth measurement (PDS) $\rightarrow$

$$\begin{aligned} \hat{d} - d_{meas} &= (d + \delta d) - (d + v_d) \\ &= \delta d - v_d \end{aligned}$$

Velocity measurement (DVL) $\rightarrow$

$$\begin{aligned} \hat{\boldsymbol{V}}^n - \boldsymbol{V}^n_{meas} &= (\boldsymbol{V}^n + \delta \boldsymbol{V}^n) - \hat{\boldsymbol{R}}^n_b \left(\boldsymbol{V}^b_{DVL} + \boldsymbol{V}^b_{DVL,b} + \boldsymbol{v}_{DVL}\right) \\ &= \delta \boldsymbol{V}^n - \boldsymbol{V}^n \wedge \boldsymbol{\gamma} - \hat{\boldsymbol{R}}^n_b \boldsymbol{V}^b_{DVL,b} - \hat{\boldsymbol{R}}^n_b \boldsymbol{v}_{DVL} \end{aligned}$$

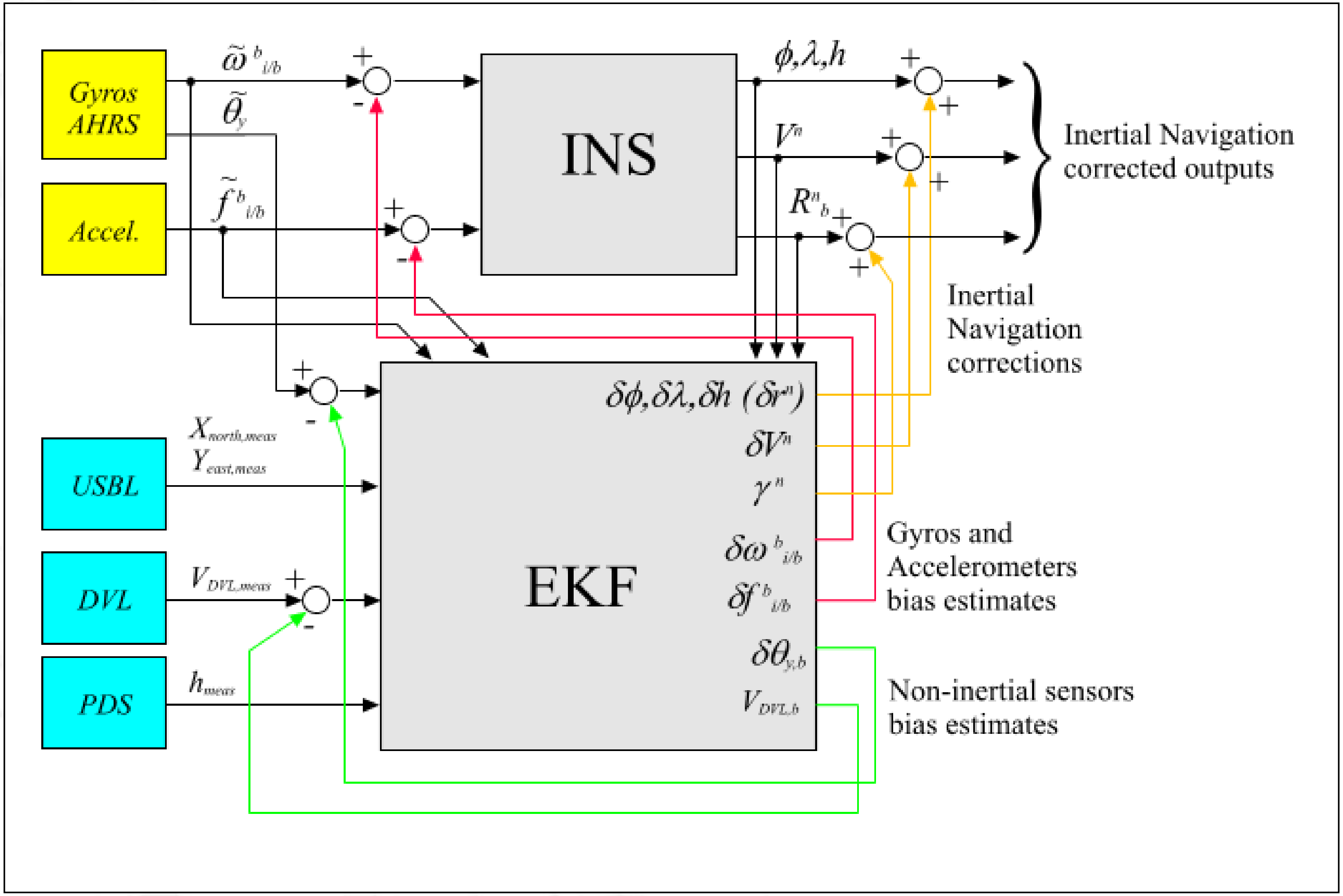
Heading measurement (Litef) $\rightarrow$

$$\hat{\theta}_y - \theta_{y,meas} = -\delta\theta_y - \theta_{y,b} - v_{\theta_y}$$

Position Measurement (HiPAP) $\rightarrow$

$$\begin{aligned} \hat{X}_{north} - X_{north,meas} &= (X_{north} + \delta X_{north}) - (X_{north} + v_{X_{north}}) \\ &= \delta X_{North} - v_{X_{north}} \\ \hat{Y}_{east} - Y_{east,meas} &= (Y_{east} + \delta Y_{east}) - (Y_{east} + v_{Y_{east}}) \\ &= \delta Y_{east} - v_{Y_{east}} \end{aligned}$$

The Indirect Kalman Filter

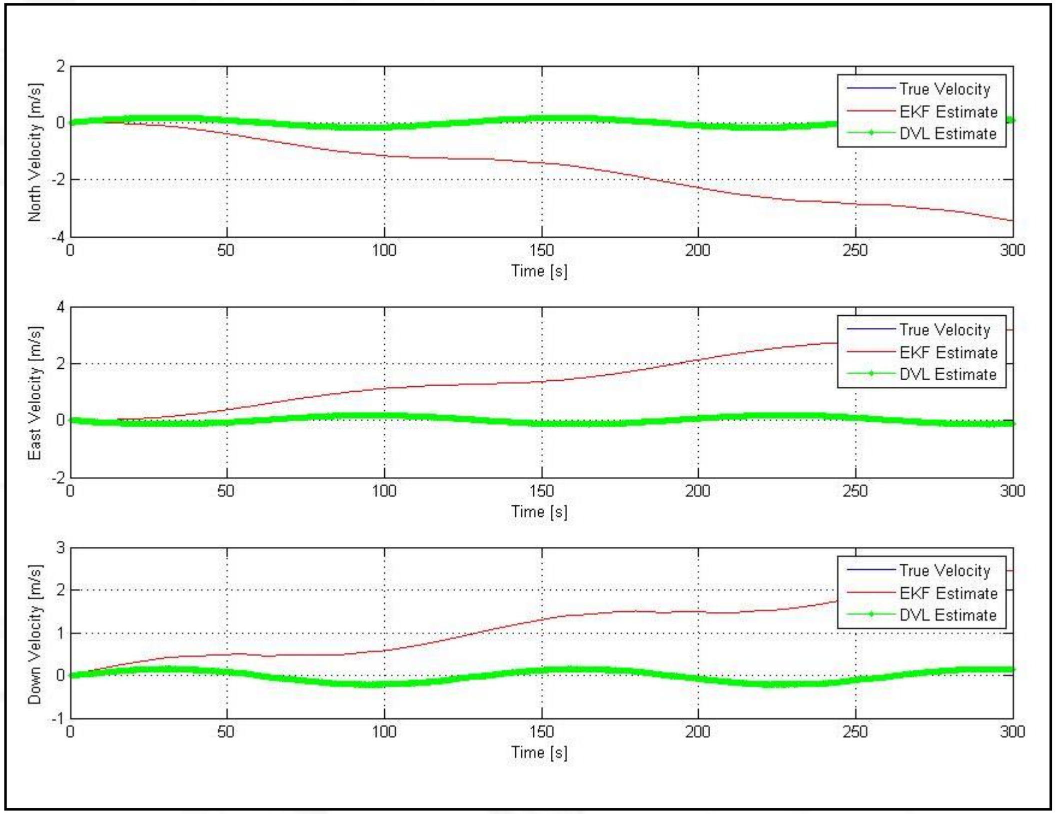


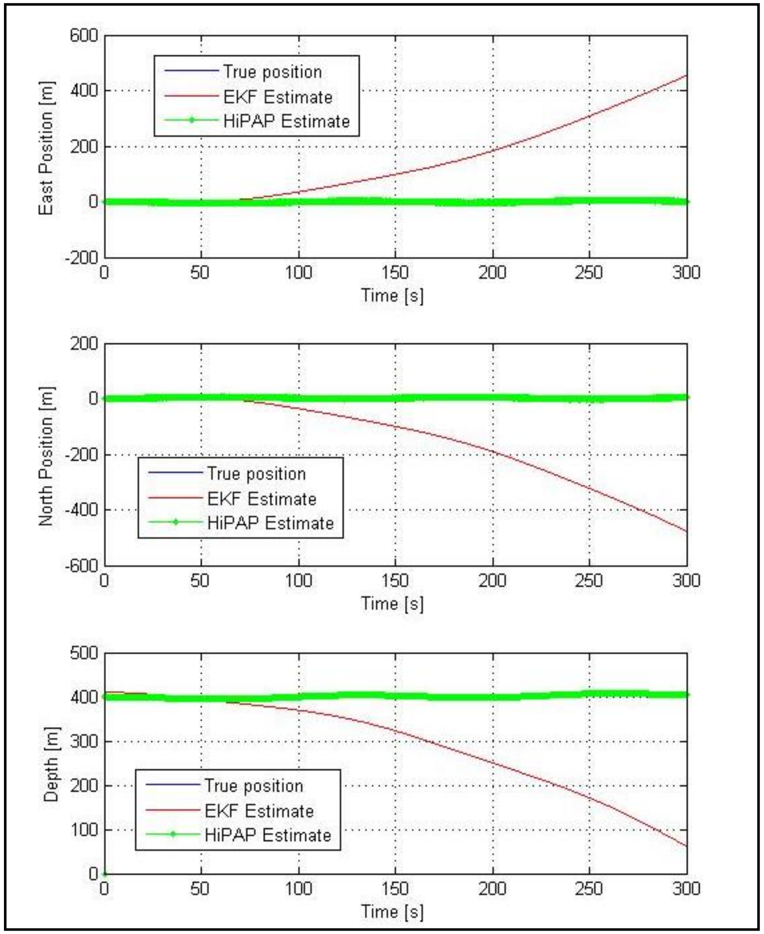
Simulations

- No ROV model or control
- Simple periodic acceleration and angular velocity profiles
- **No special tuning of the EKF gains**
- 3 cases shown:
  - Open loop: plain Inertial Navigation
  - Ideal case: high frequency periodic sampling, constant biases, no quantization
  - More realistic case: sampling frequency chosen from datasheets, slowly varying biases, quantization (on some sensors)

Simulations

Open Loop INS: no corrections, no bias estimates





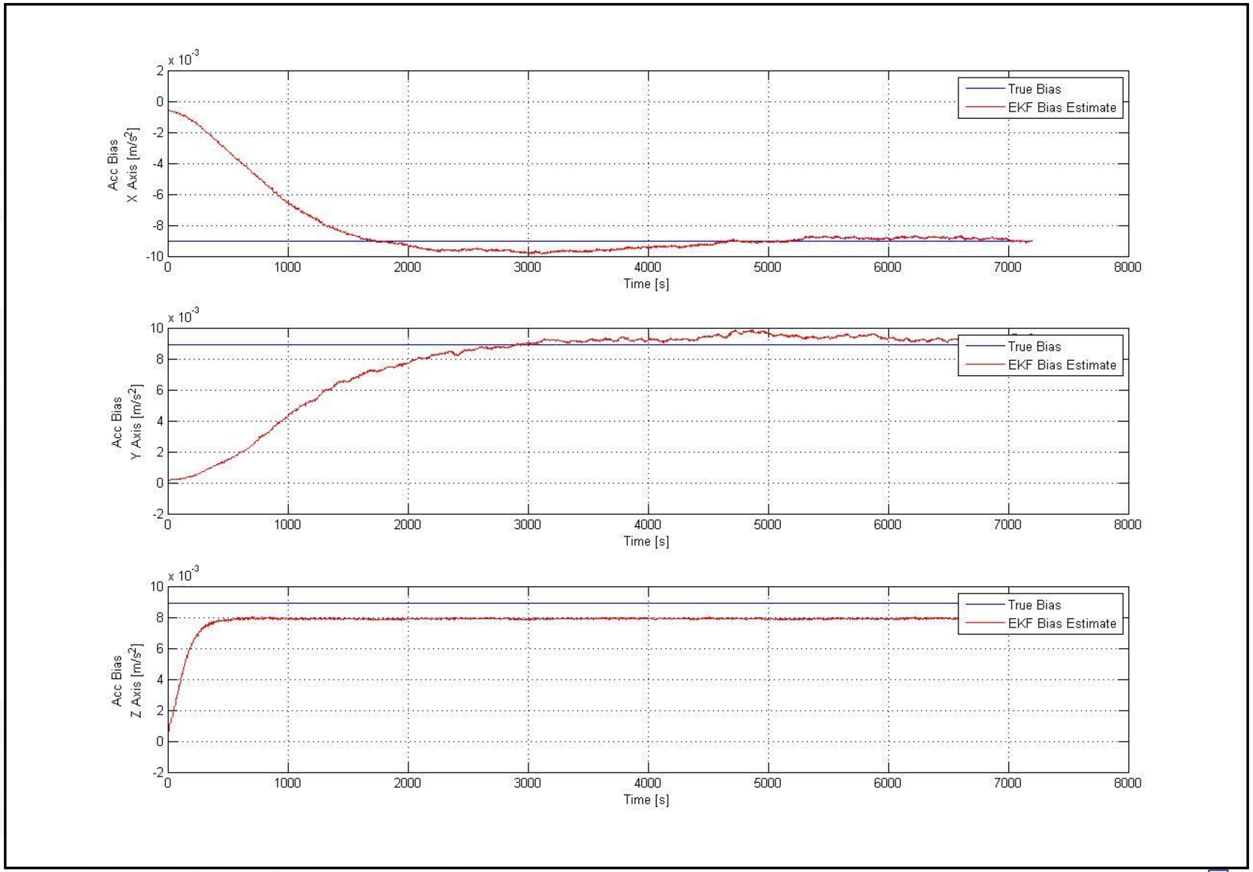
Simulations

With Indirect Extended Kalman Filter

Inertial sensors: 100Hz, Aiding sensors: 10 Hz

No quantization / No time-varying biases

Accelerometer biases



Almost perfect convergence to simulated bias values

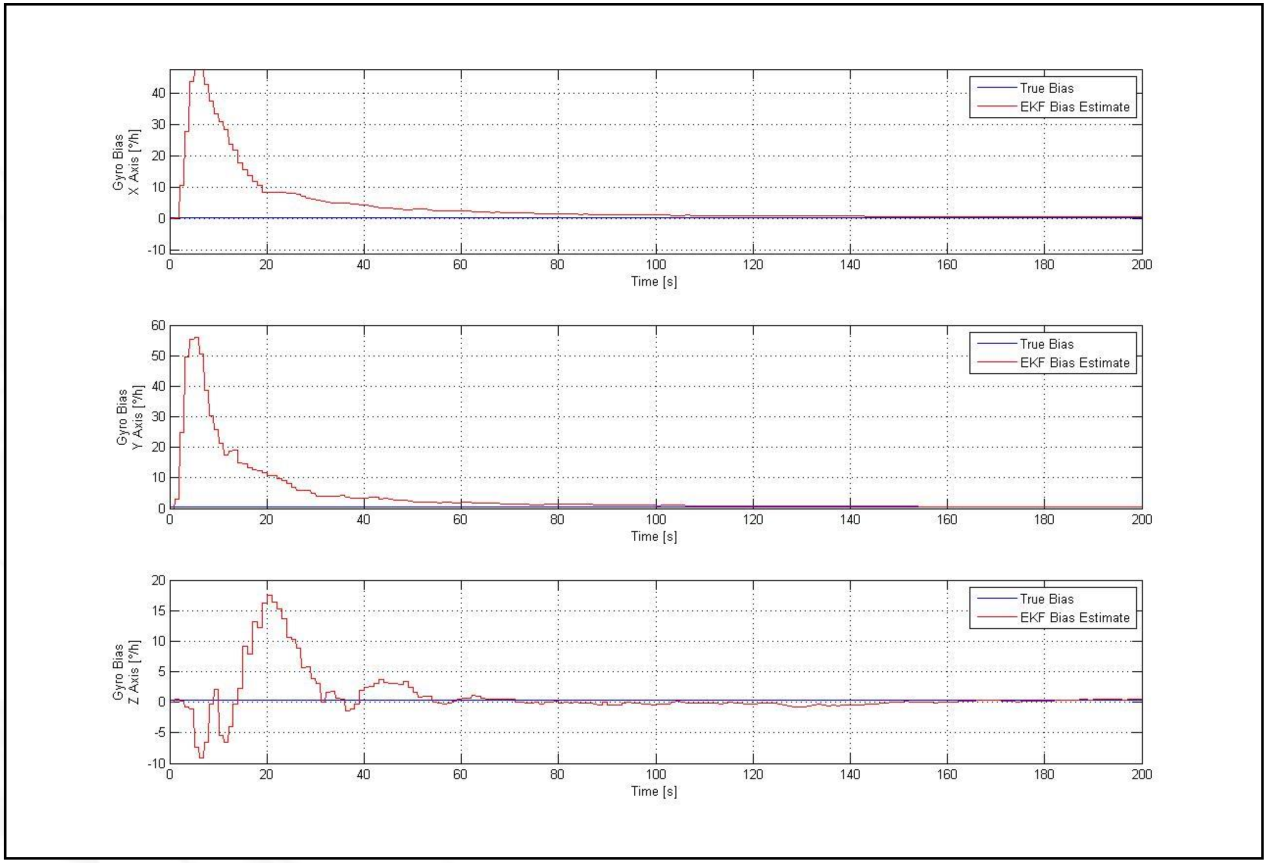
Simulations

With Indirect Extended Kalman Filter

Inertial sensors: 100Hz, Aiding sensors: 10 Hz

No quantization / No time-varying biases

Gyroscope biases



Almost perfect and **very fast** convergence to simulated bias values

Simulations

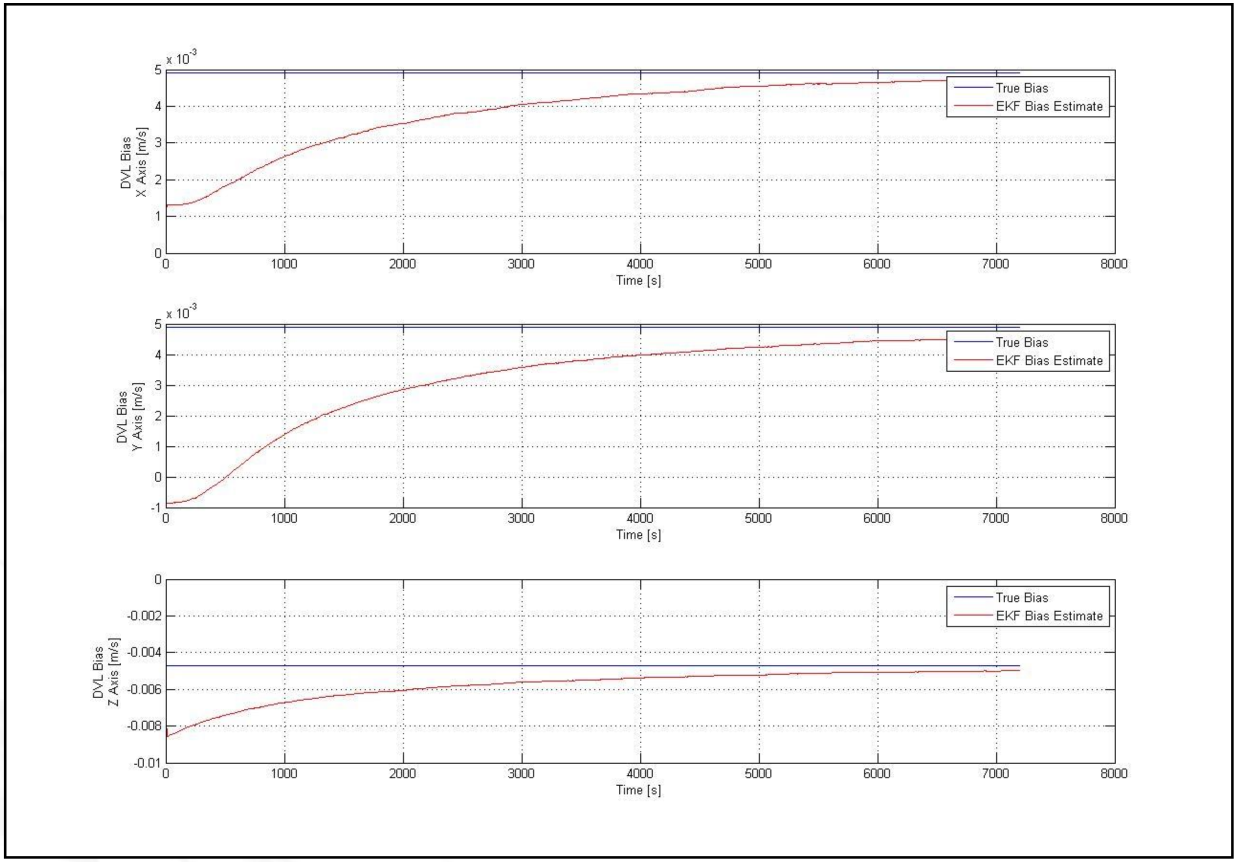
With Indirect Extended Kalman Filter

Inertial sensors: 100Hz, Aiding sensors: 10 Hz

No quantization / No time-varying biases

DVL

biases



Almost perfect convergence to simulated bias values

Simulations

With Indirect Extended Kalman Filter

Inertial sensors: 100Hz, Aiding sensors: 10 Hz

No quantization / No time-varying biases

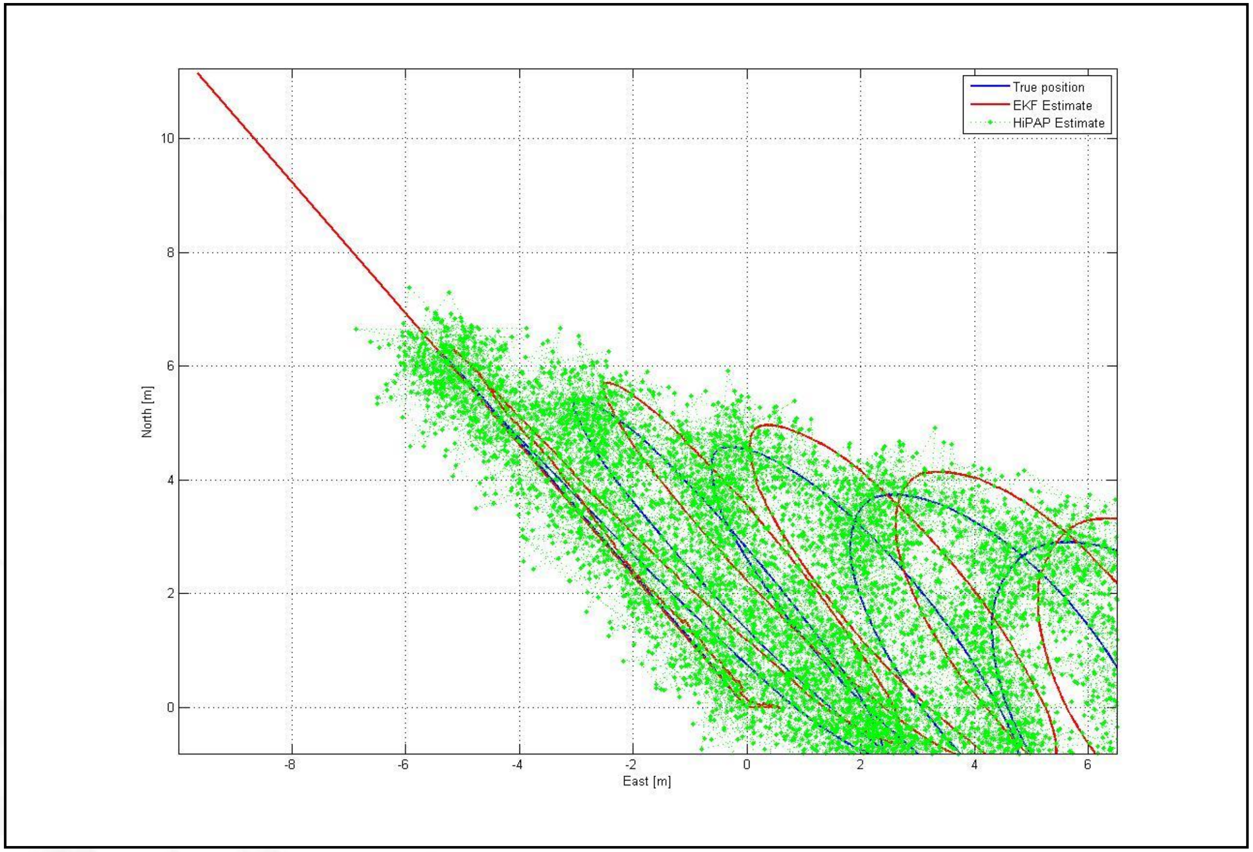
Estimated position

(soon after EKF initialization)

No more Drift

Noise in USBL Highly filtered

Biased estimates



Simulations

With Indirect Extended Kalman Filter

Inertial sensors: 100Hz, Aiding sensors: 10 Hz

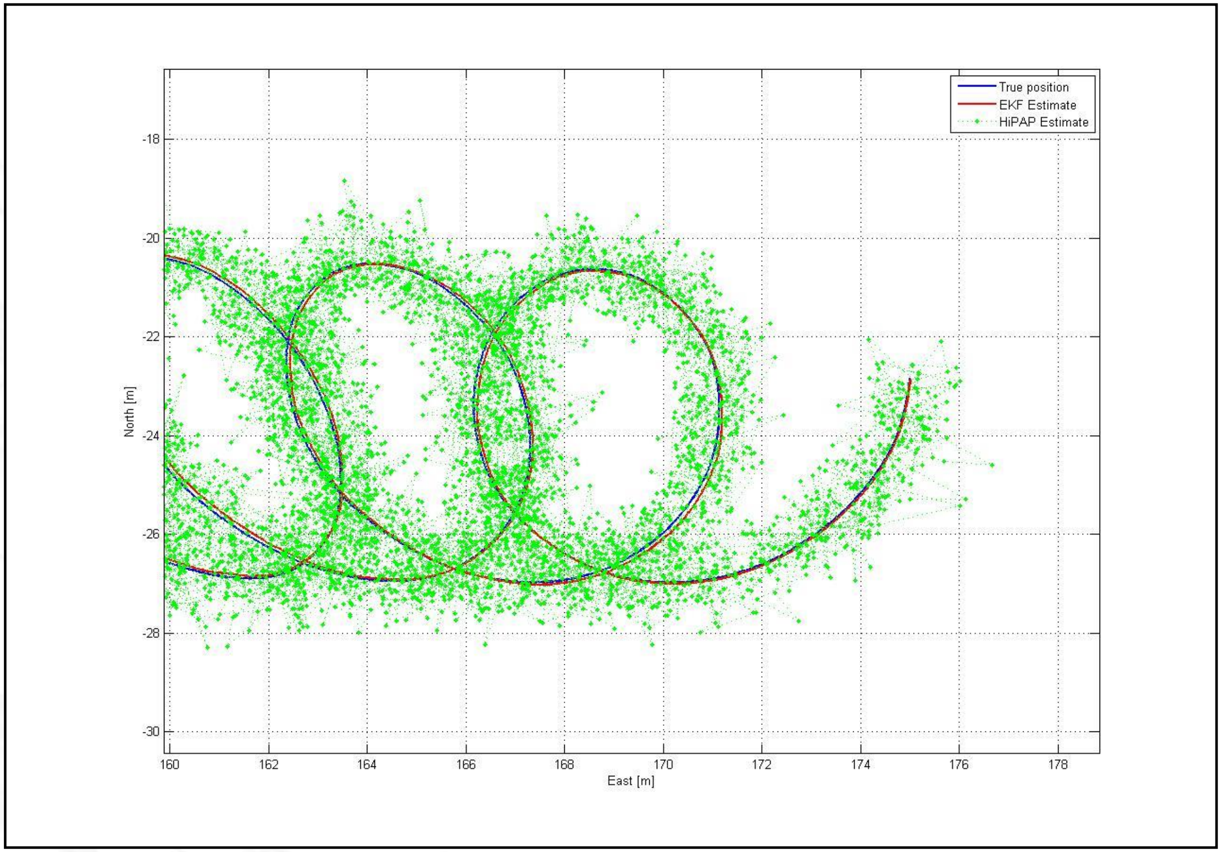
No quantization / No time-varying biases

Estimated position

(after convergence of all filter states)

No more Drift

Almost perfect convergence to real trajectory



Simulations

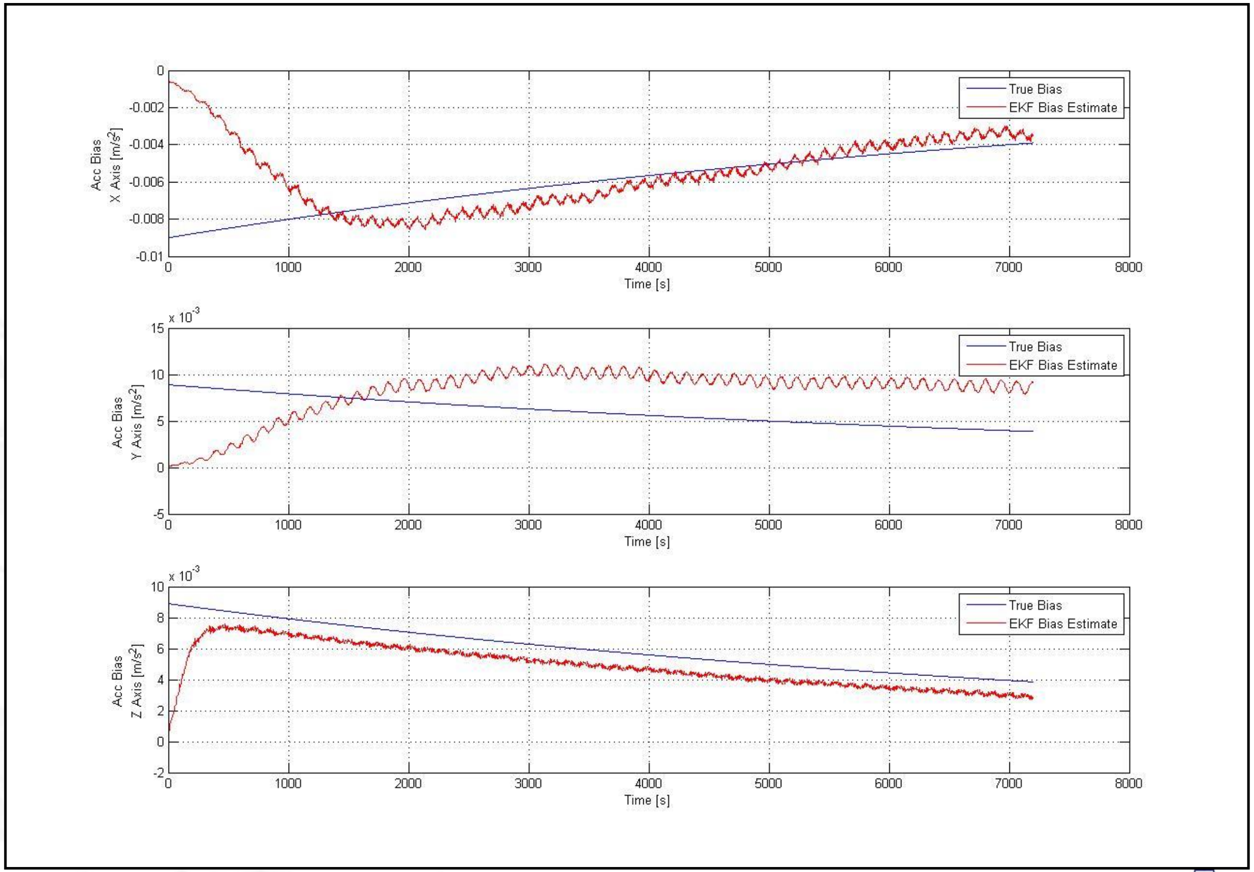
With Indirect Extended Kalman Filter

Inertial sensors: 32Hz, Aiding sensors: 1 Hz

16-bit quantization on Gyros, Acc., Heading / Time-varying biases

Accelerometer biases

Good tracking of changing bias values



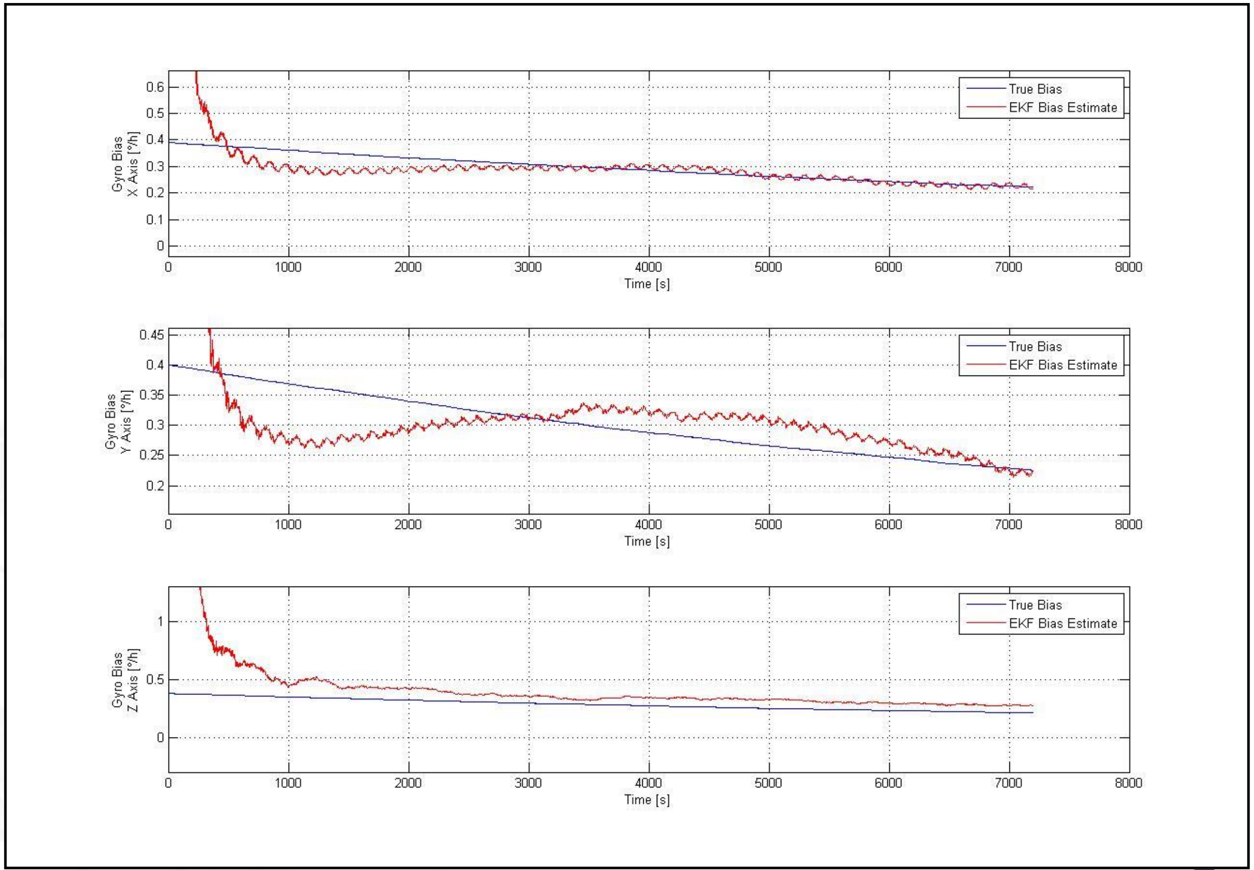
Simulations

With Indirect Extended Kalman Filter

Inertial sensors: 32Hz, Aiding sensors: 1 Hz

16-bit quantization on Gyros, Acc., Heading / Time-varying biases

Gyroscope biases



Good tracking of changing bias values

Simulations

With Indirect Extended Kalman Filter

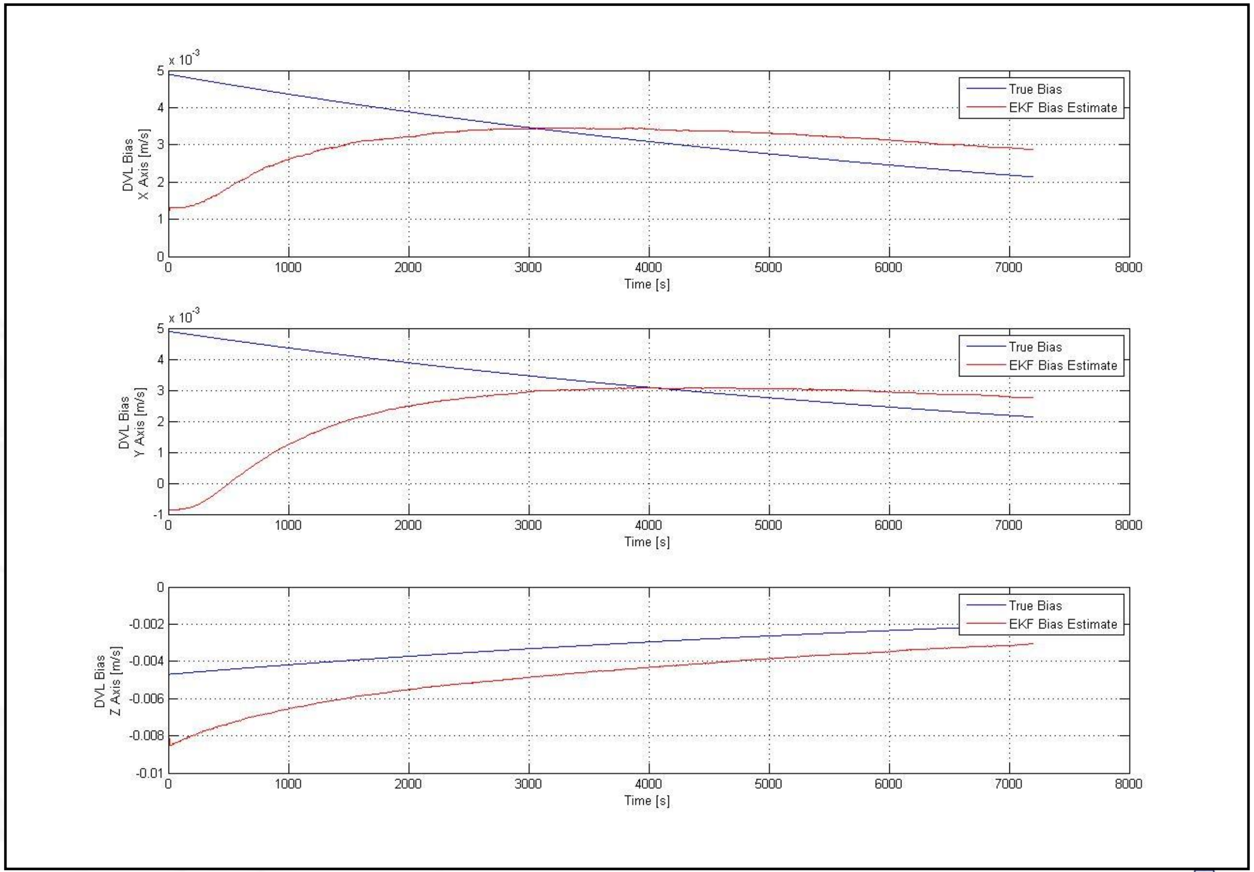
Inertial sensors: 32Hz, Aiding sensors: 1 Hz

16-bit quantization on Gyros, Acc., Heading / Time-varying biases

DVL

biases

Good tracking of changing bias values



Simulations

With Indirect Extended Kalman Filter

Inertial sensors: 32Hz, Aiding sensors: 1 Hz

16-bit quantization on Gyros, Acc., Heading / Time-varying biases

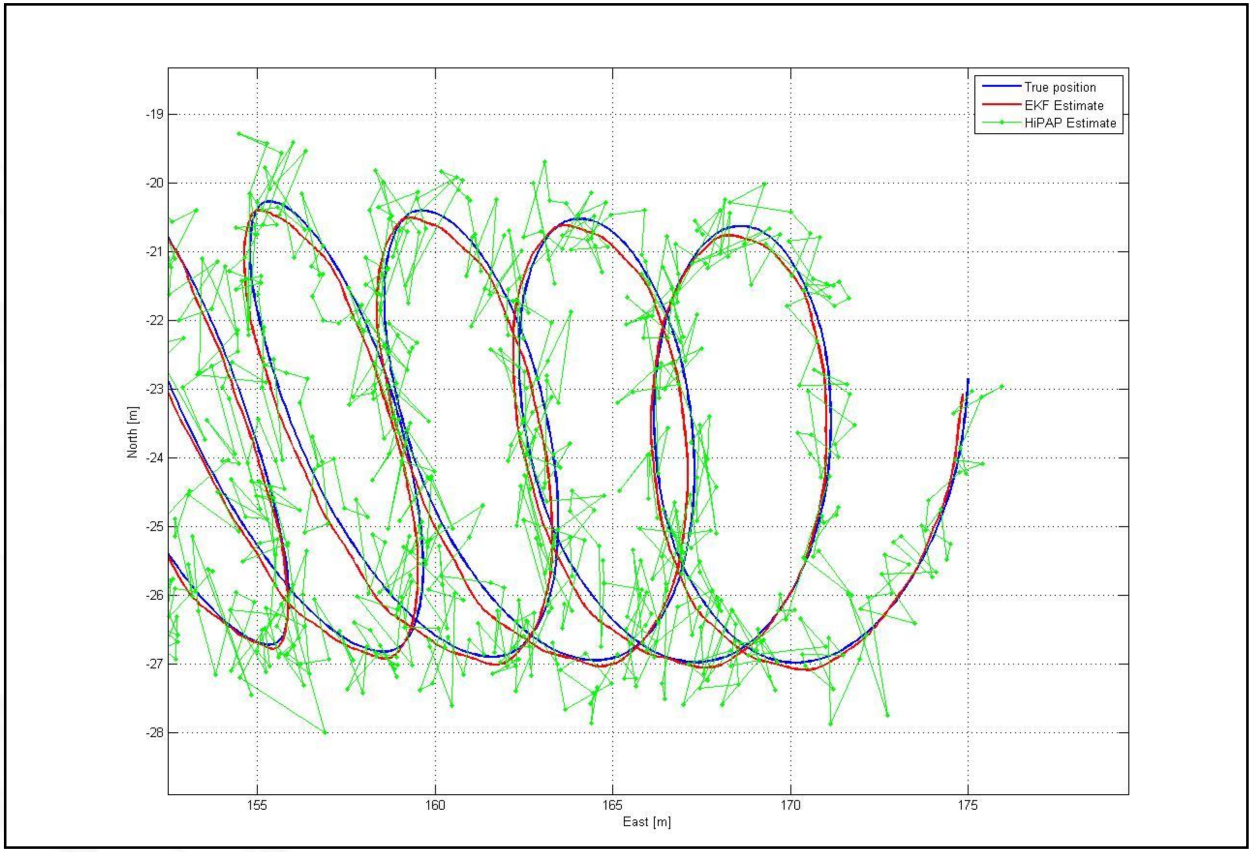
Estimated position

(after convergence of all filter states)

No Drift

Noise in USBL Highly filtered

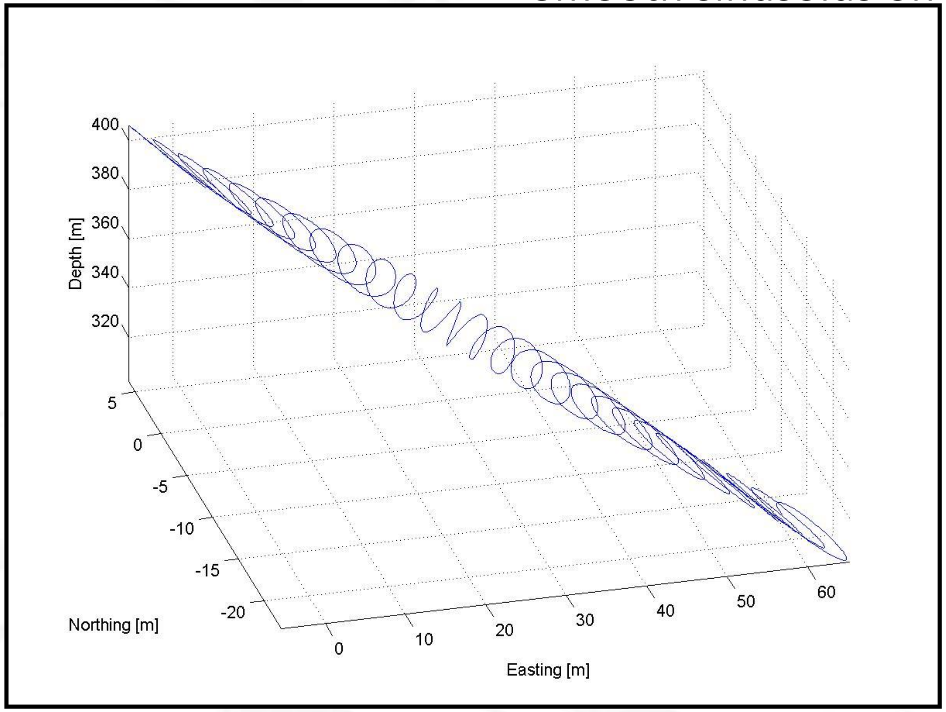
Good Tracking of real trajectory

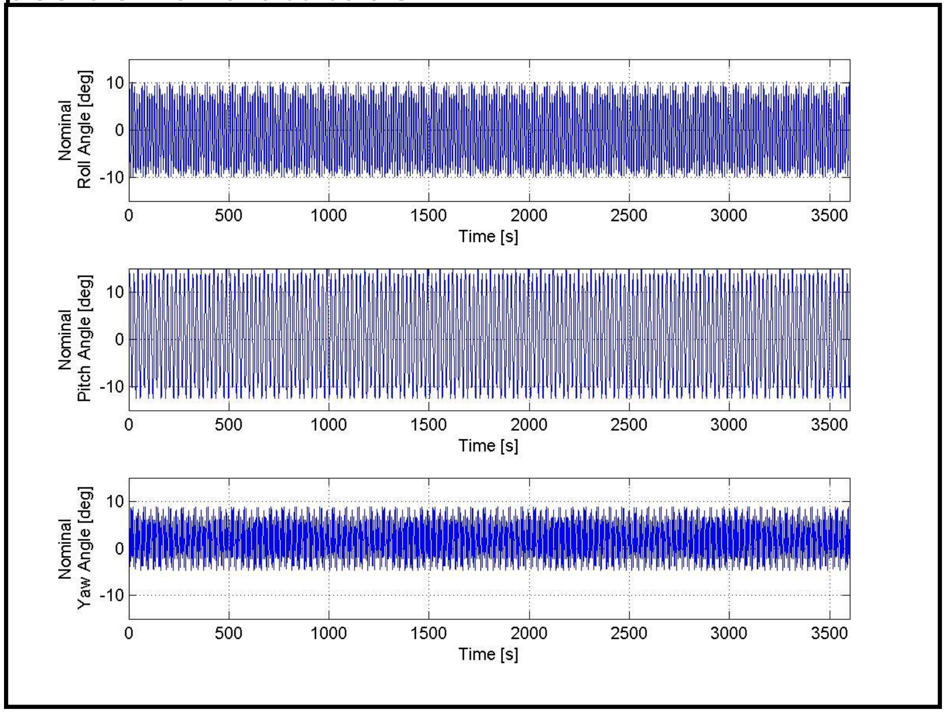


Simulations

Reference trajectory

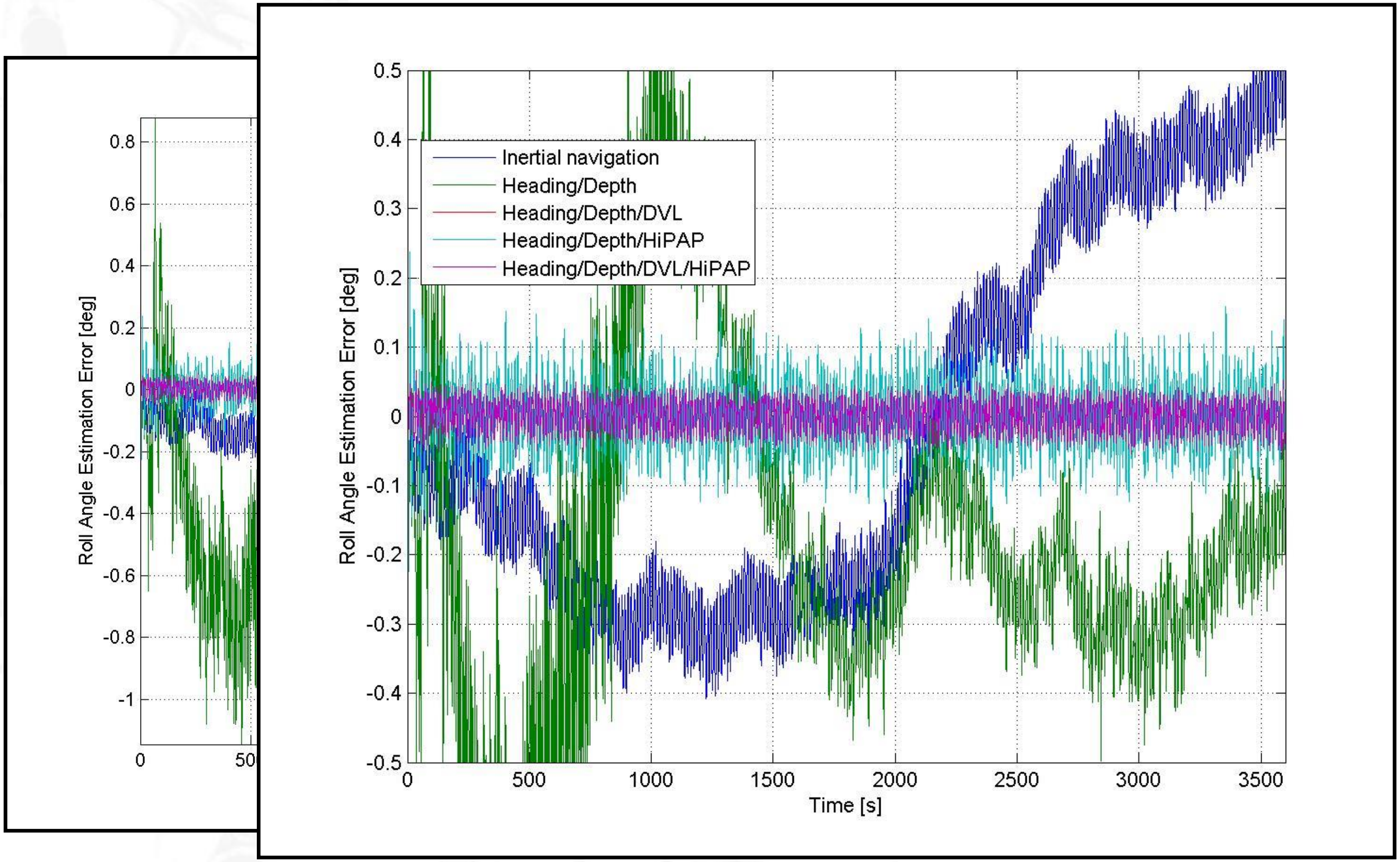
Smooth sinusoids on position and attitude





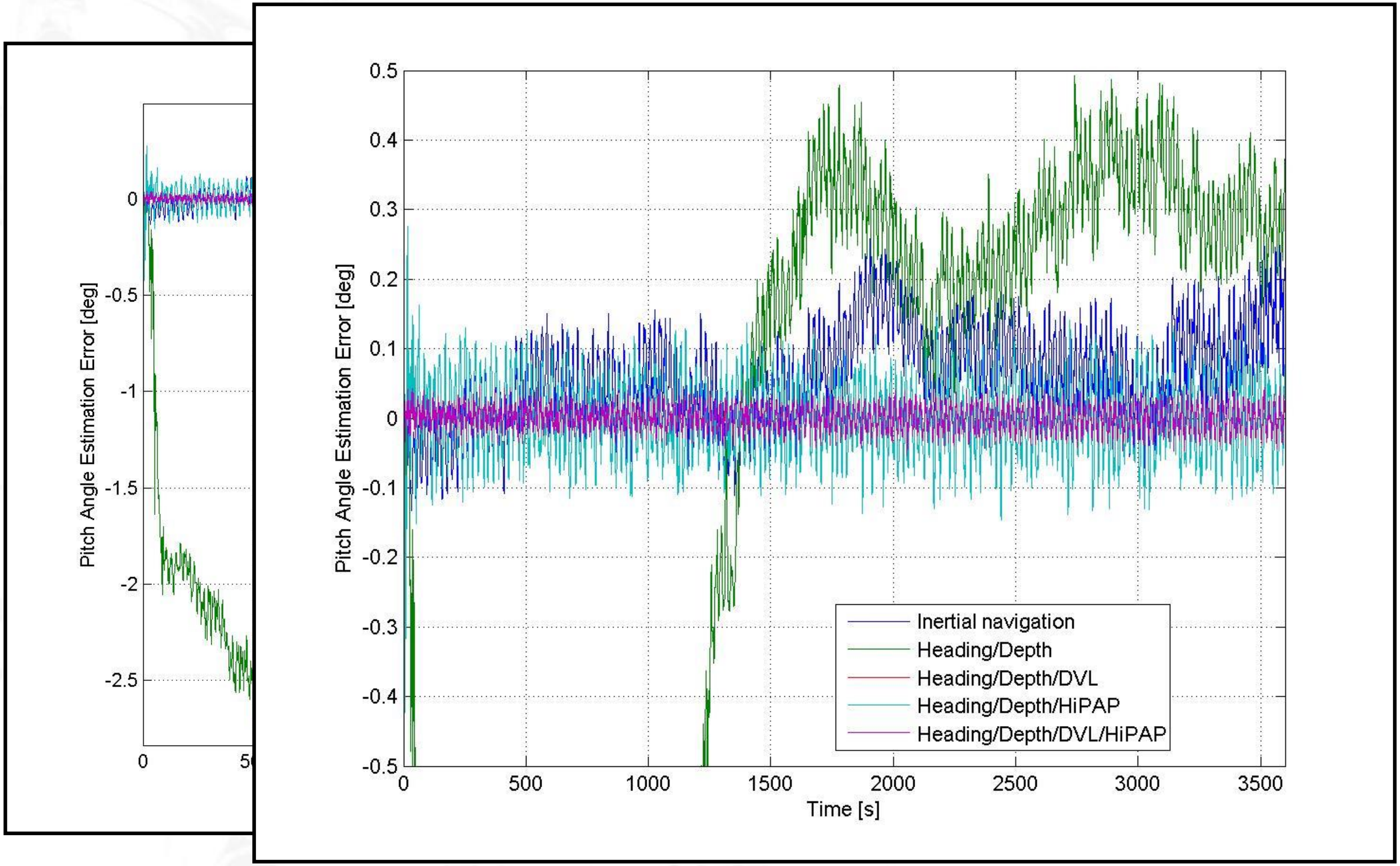
Simulations

Comparison of performances with different navigation aids



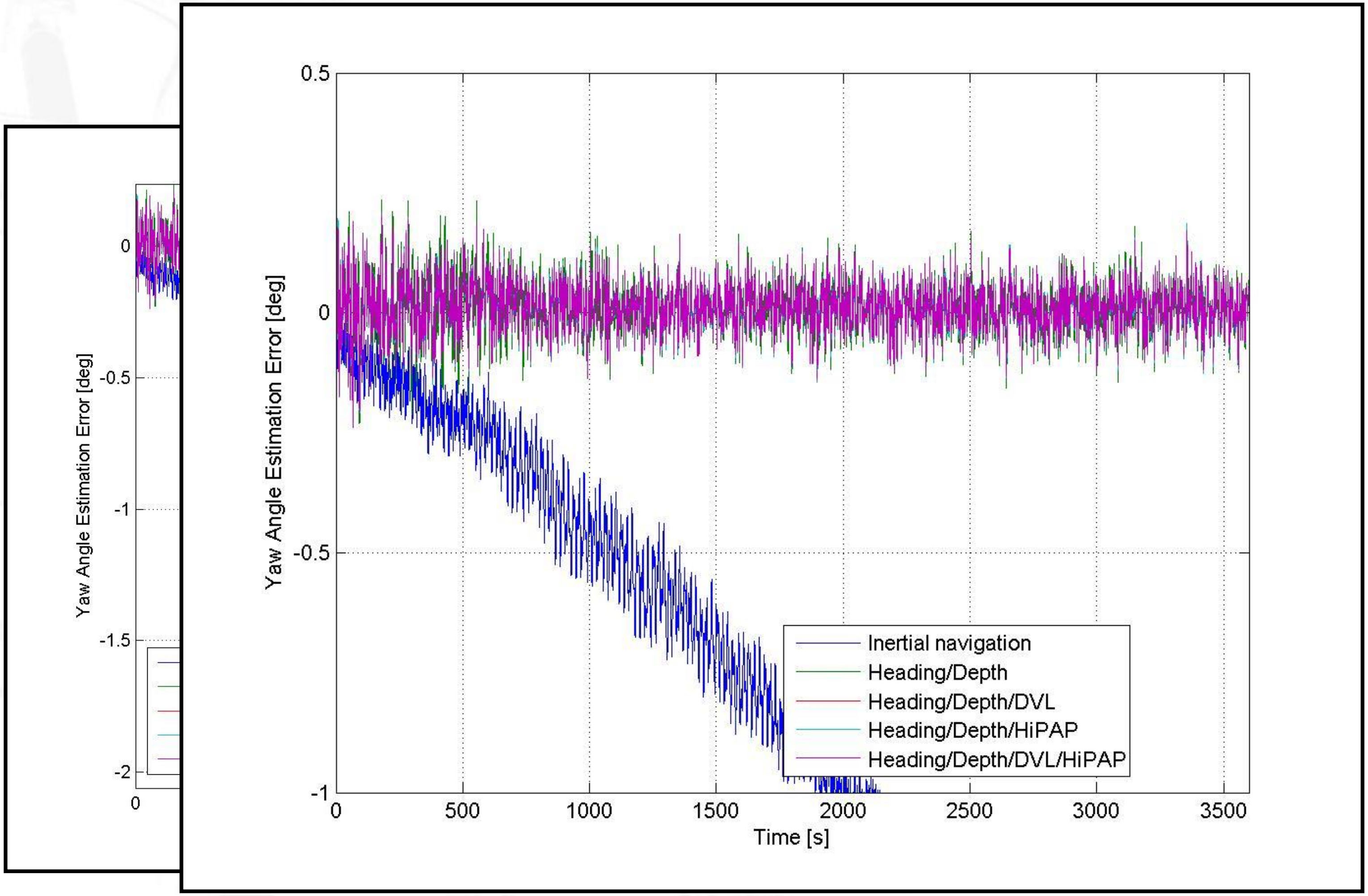
Simulations

Comparison of performances with different navigation aids



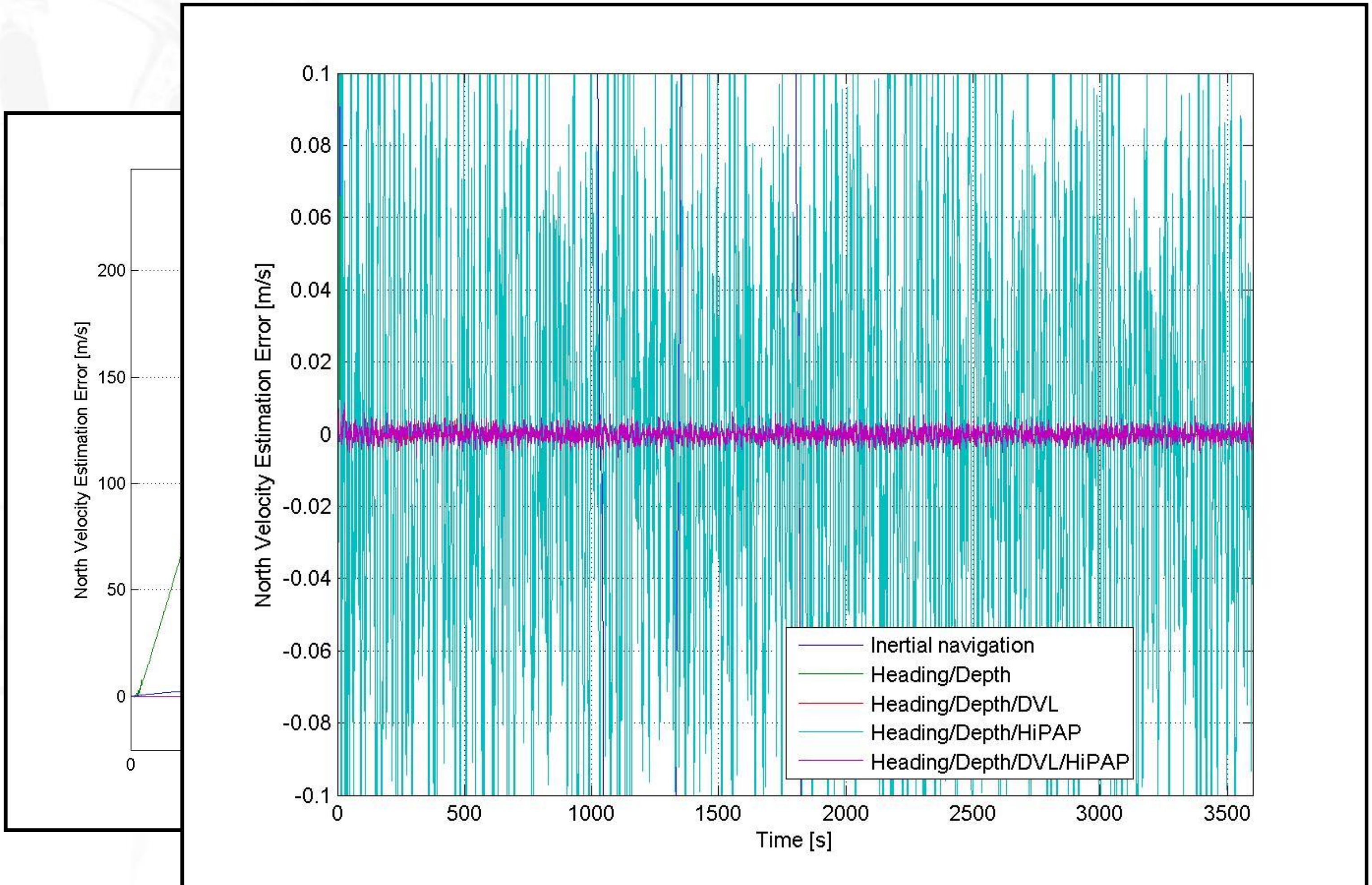
Simulations

Comparison of performances with different navigation aids



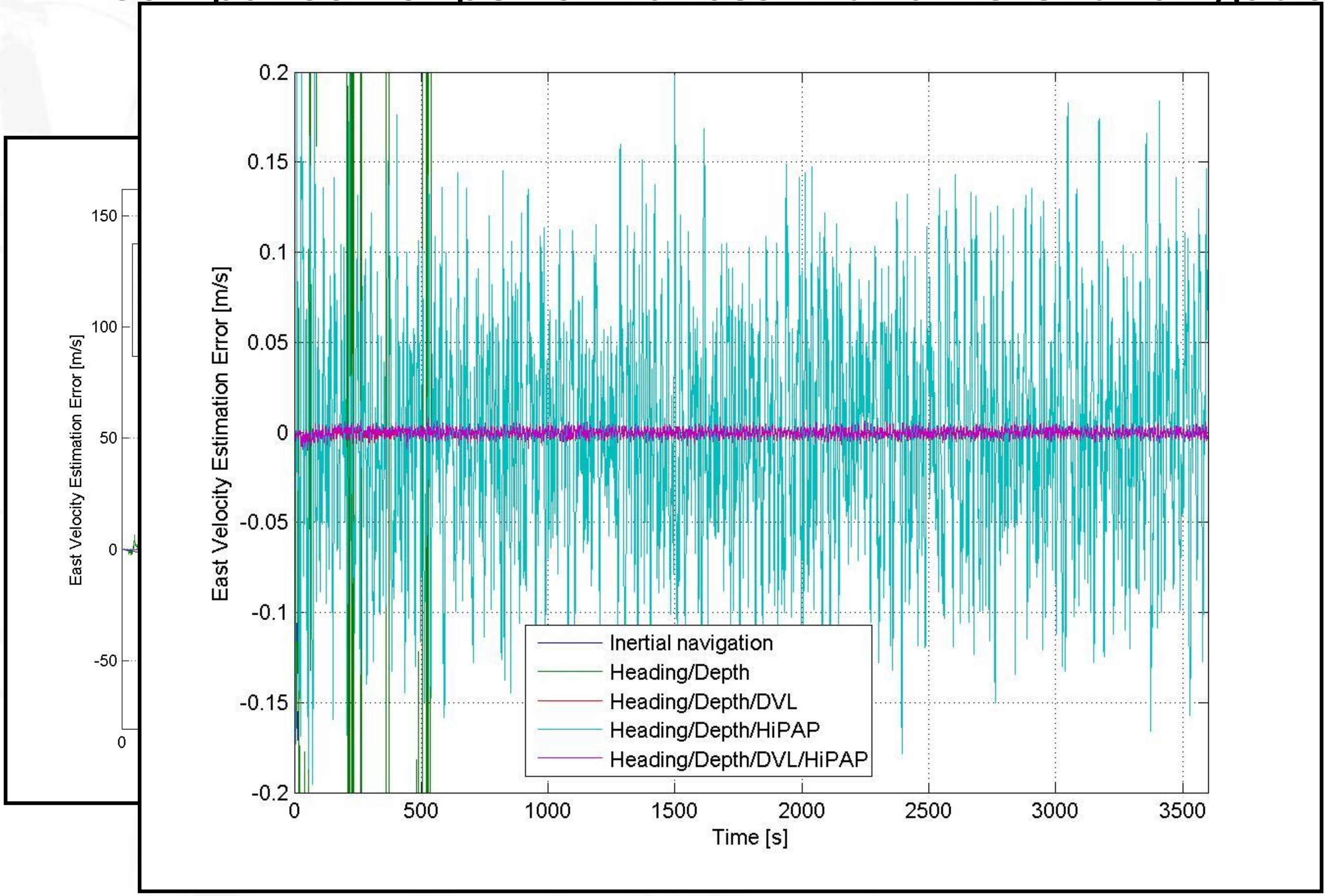
Simulations

Comparison of performances with different navigation aids



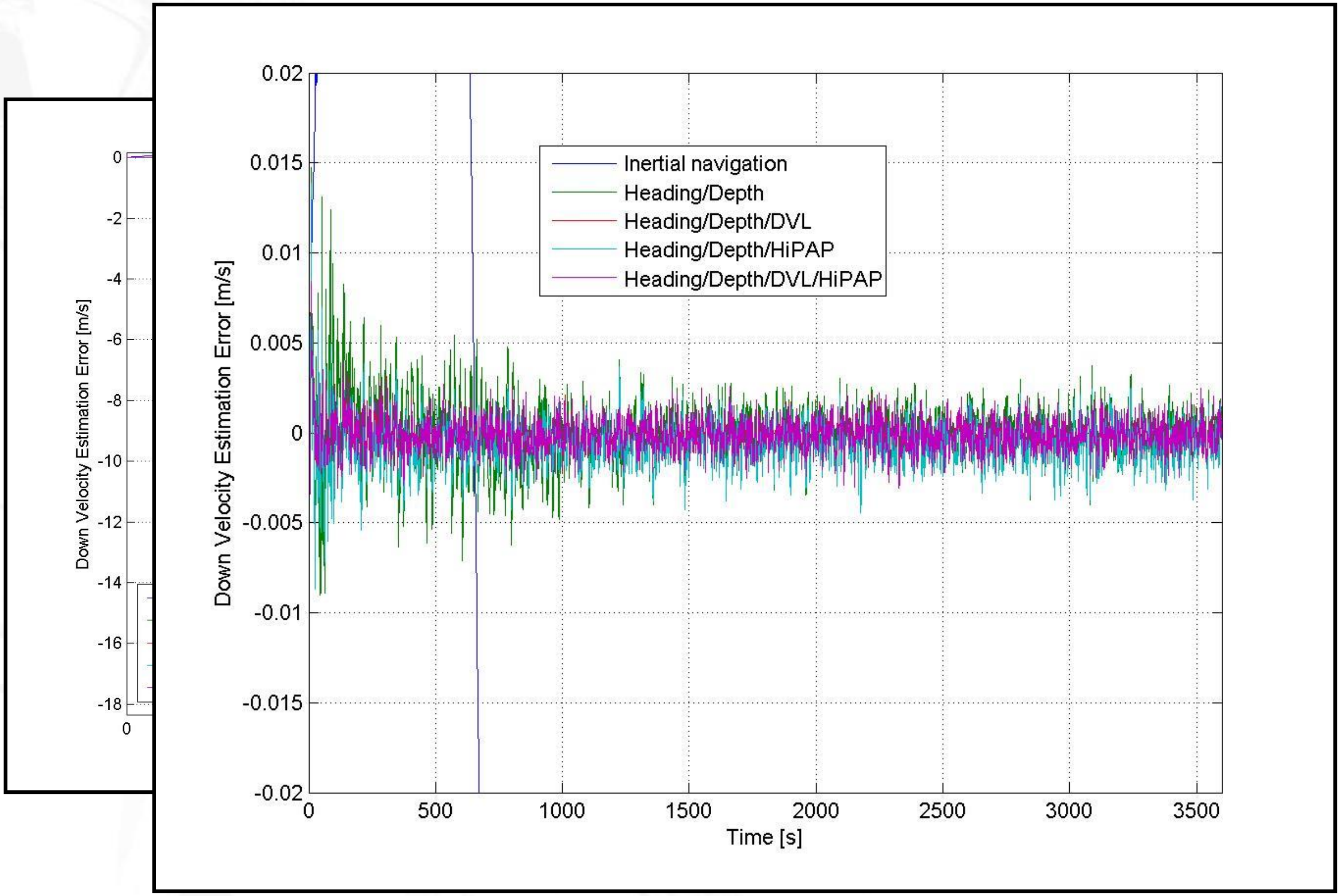
Simulations

Comparison of performances with different navigation aids



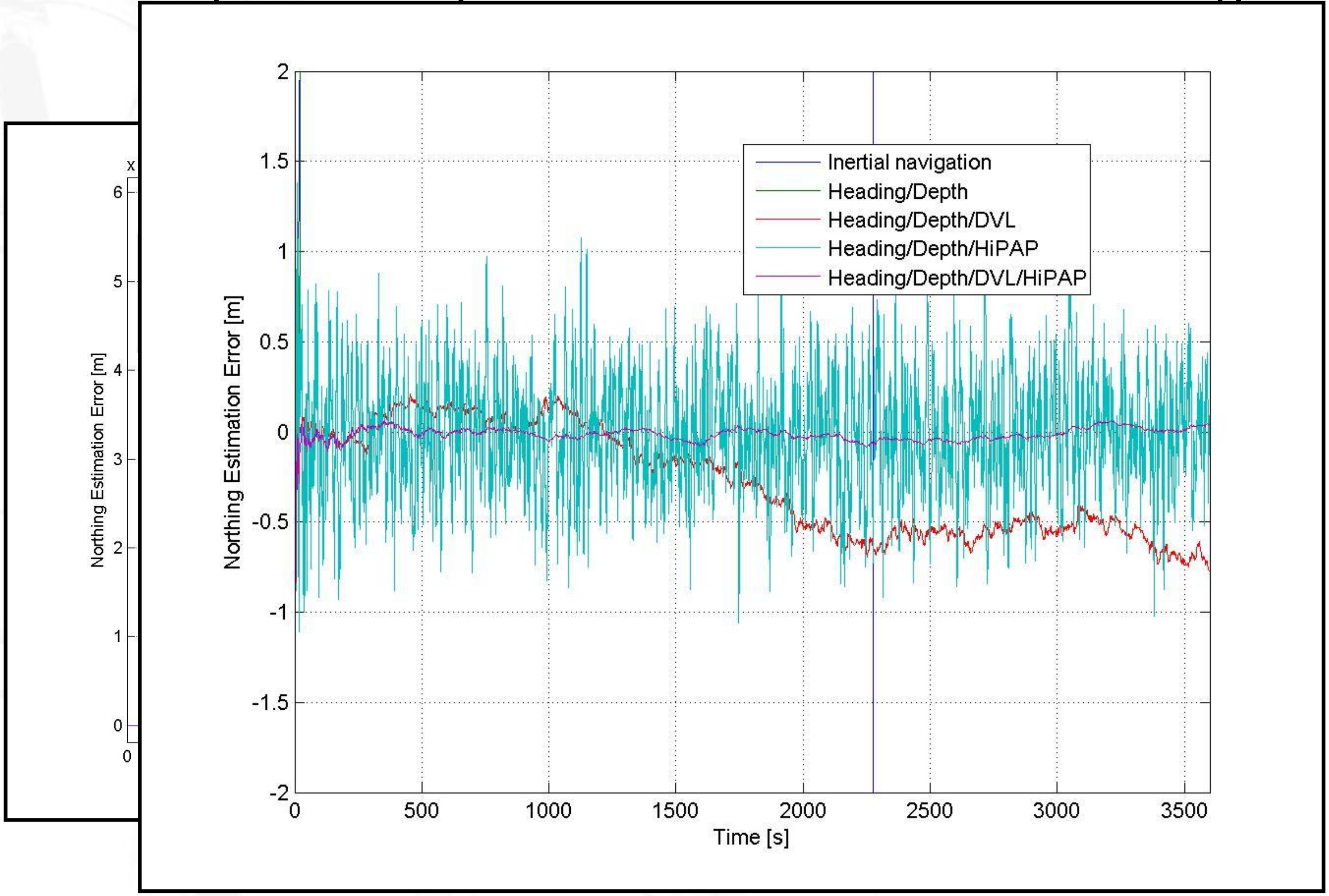
Simulations

Comparison of performances with different navigation aids



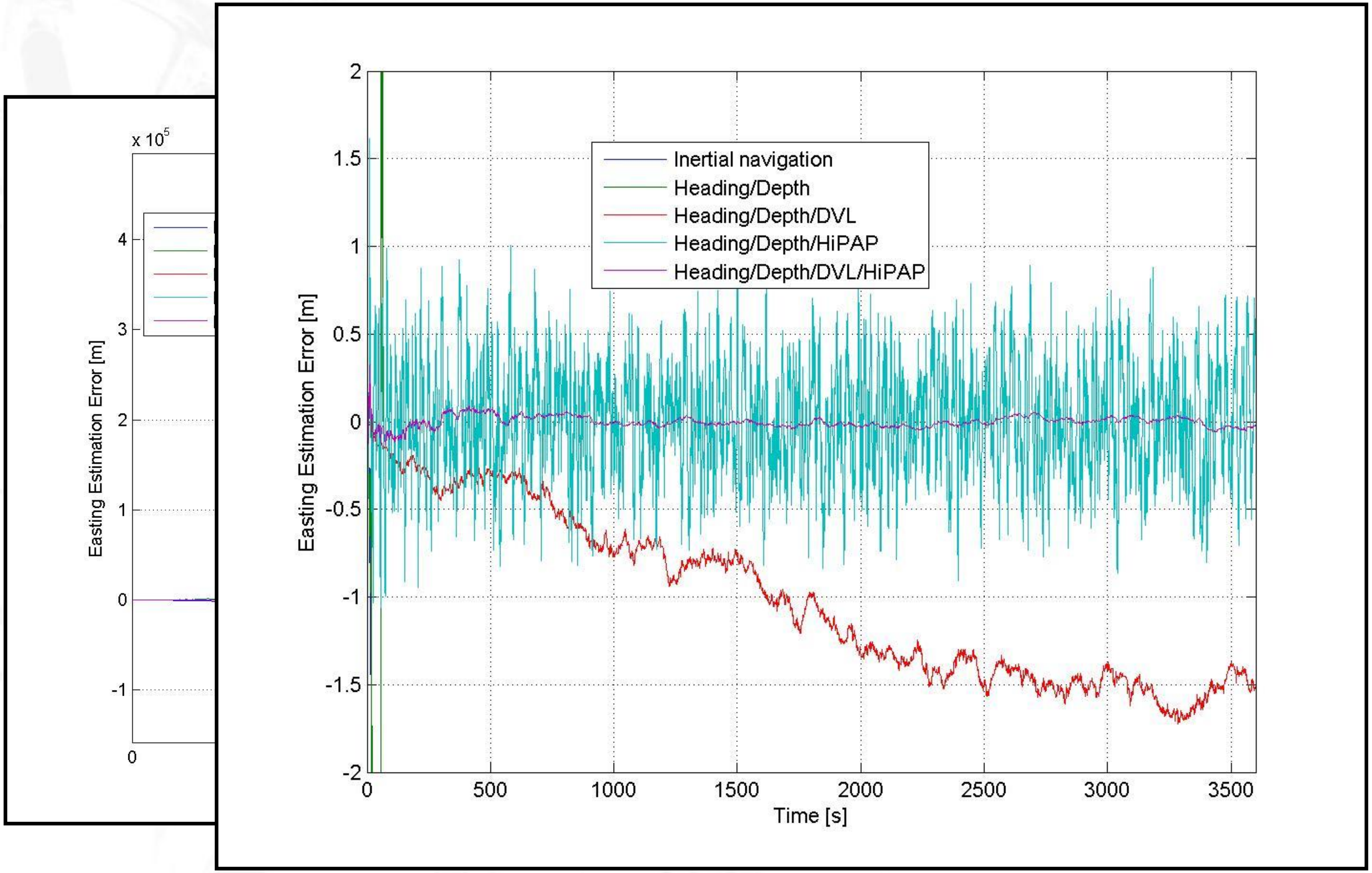
Simulations

Comparison of performances with different navigation aids



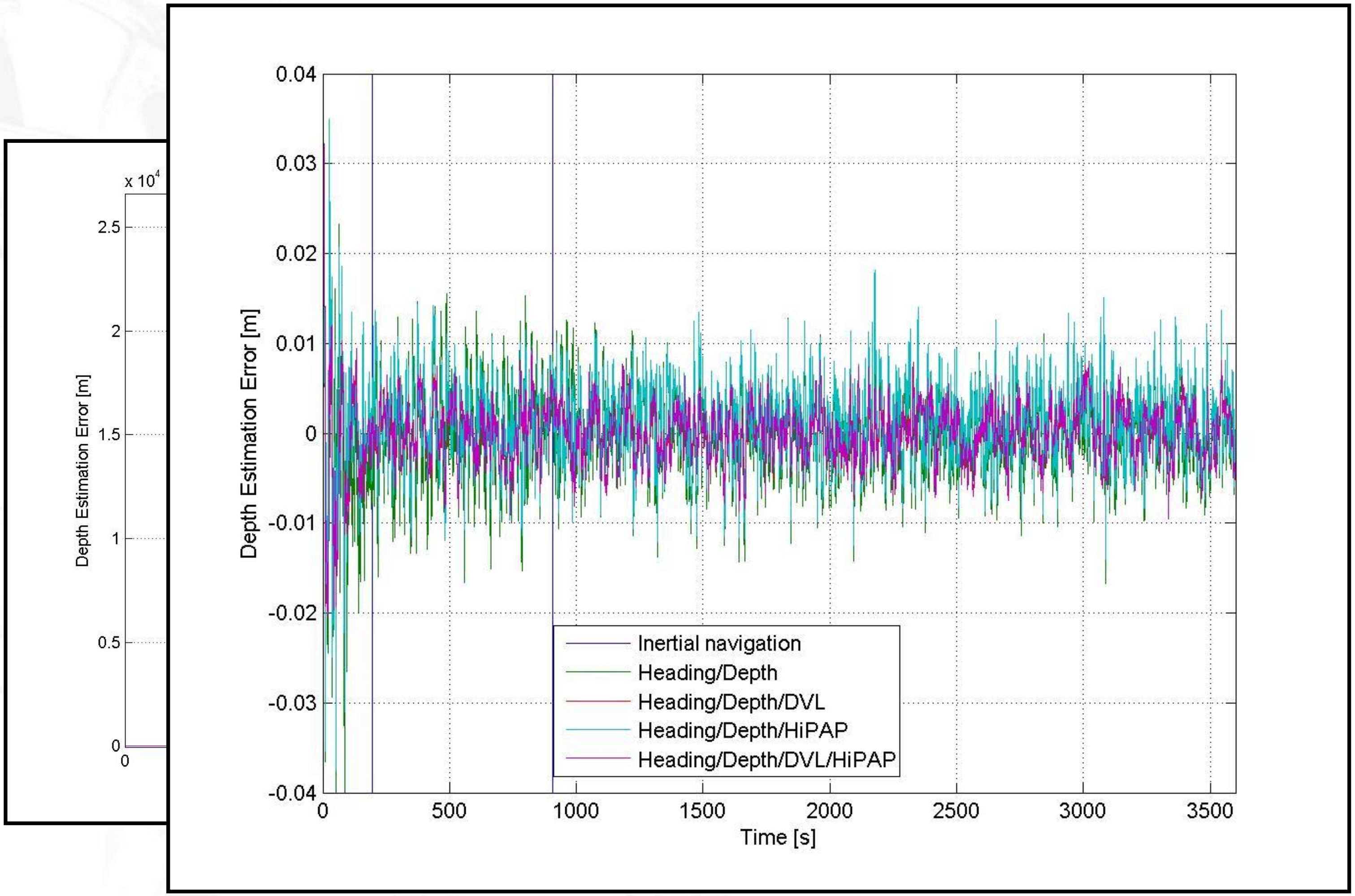
Simulations

Comparison of performances with different navigation aids



Simulations

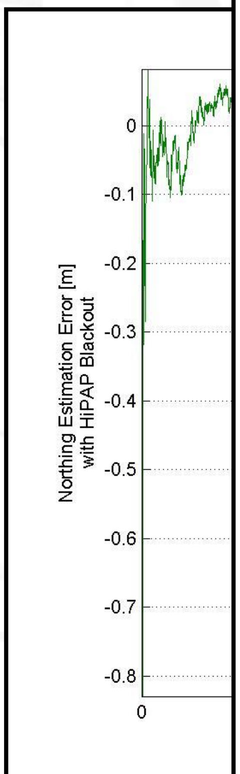
Comparison of performances with different navigation aids

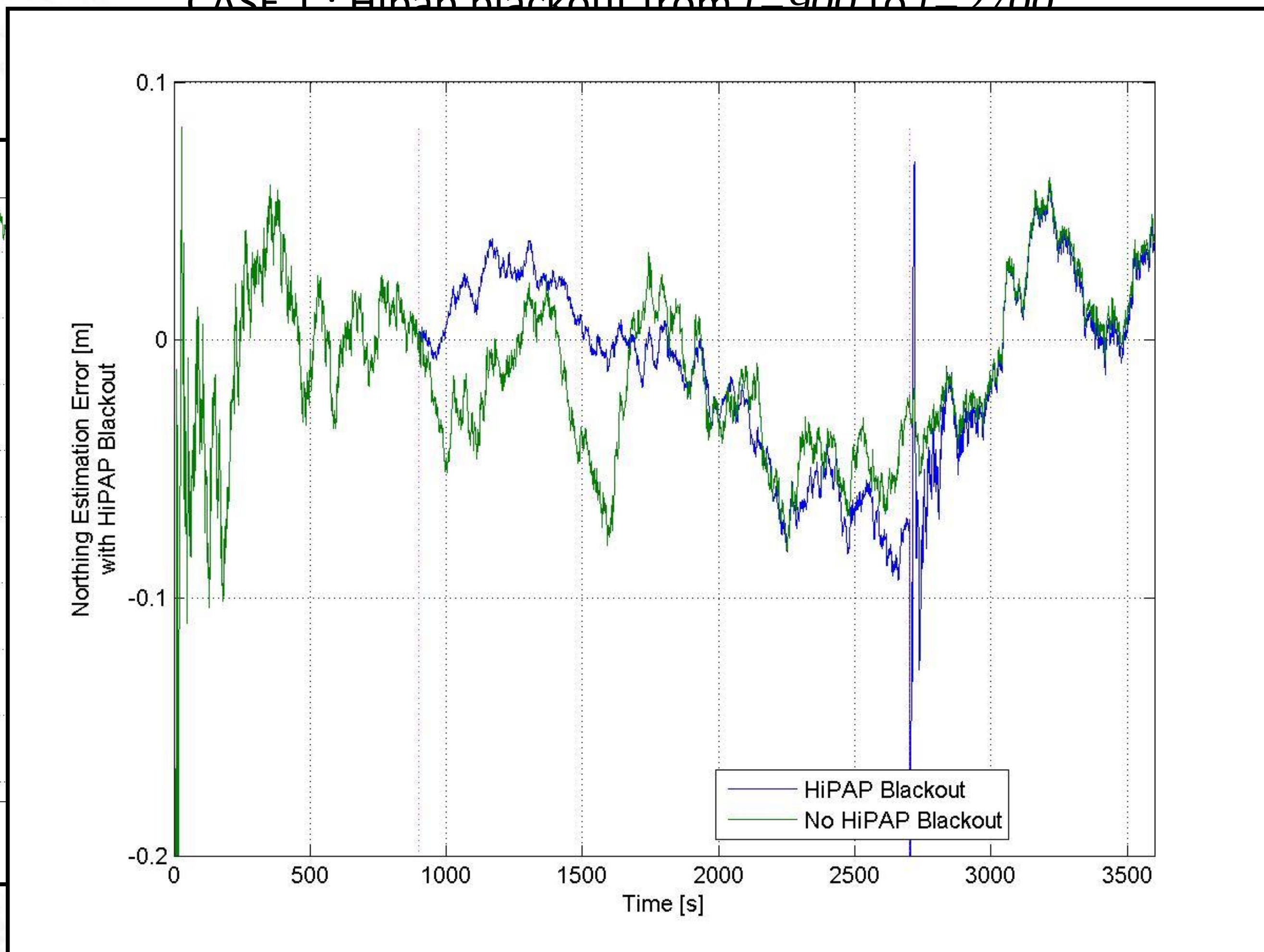


Simulations

Simulation of long term aiding sensor black-out

CASE 1 : Hipap blackout from $t=900$ to $t=2700$

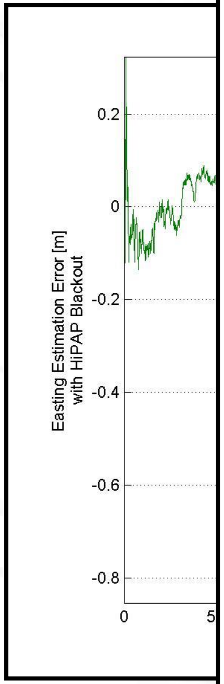


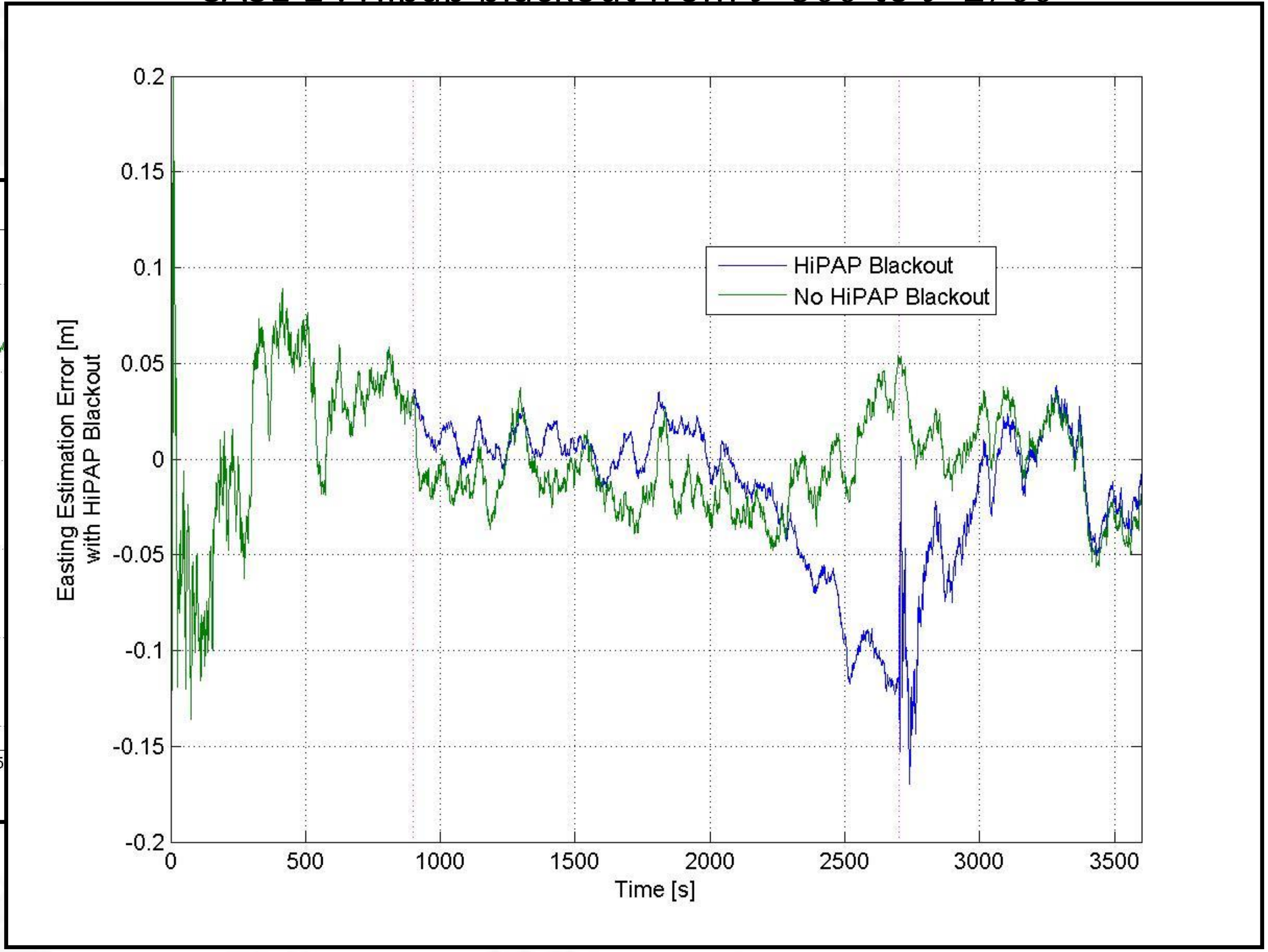


Simulations

Simulation of long term aiding sensor black-out

CASE 1 : Hipap blackout from *t=900* to *t=2700*

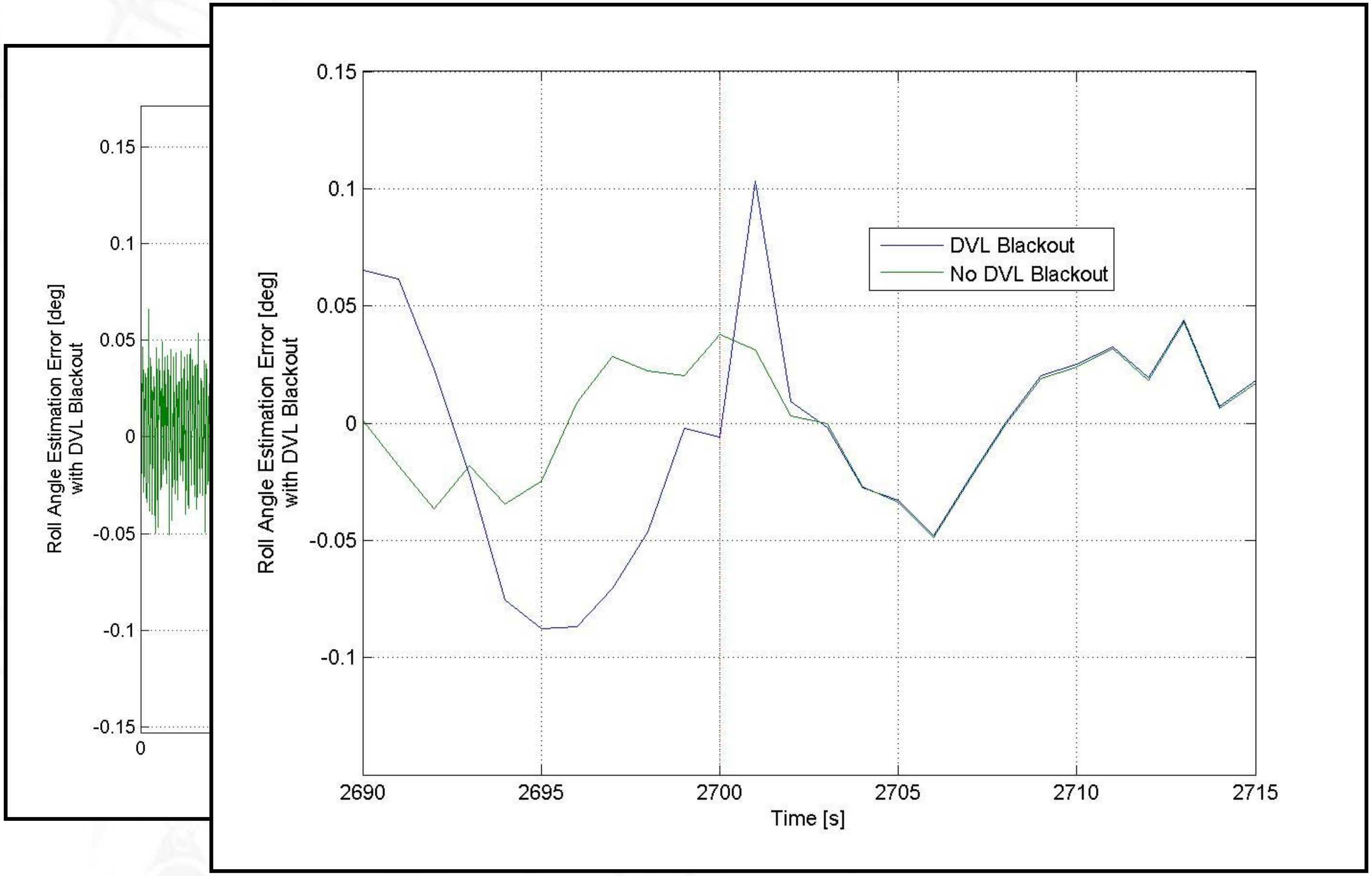




Simulations

Simulation of long term aiding sensor black-out

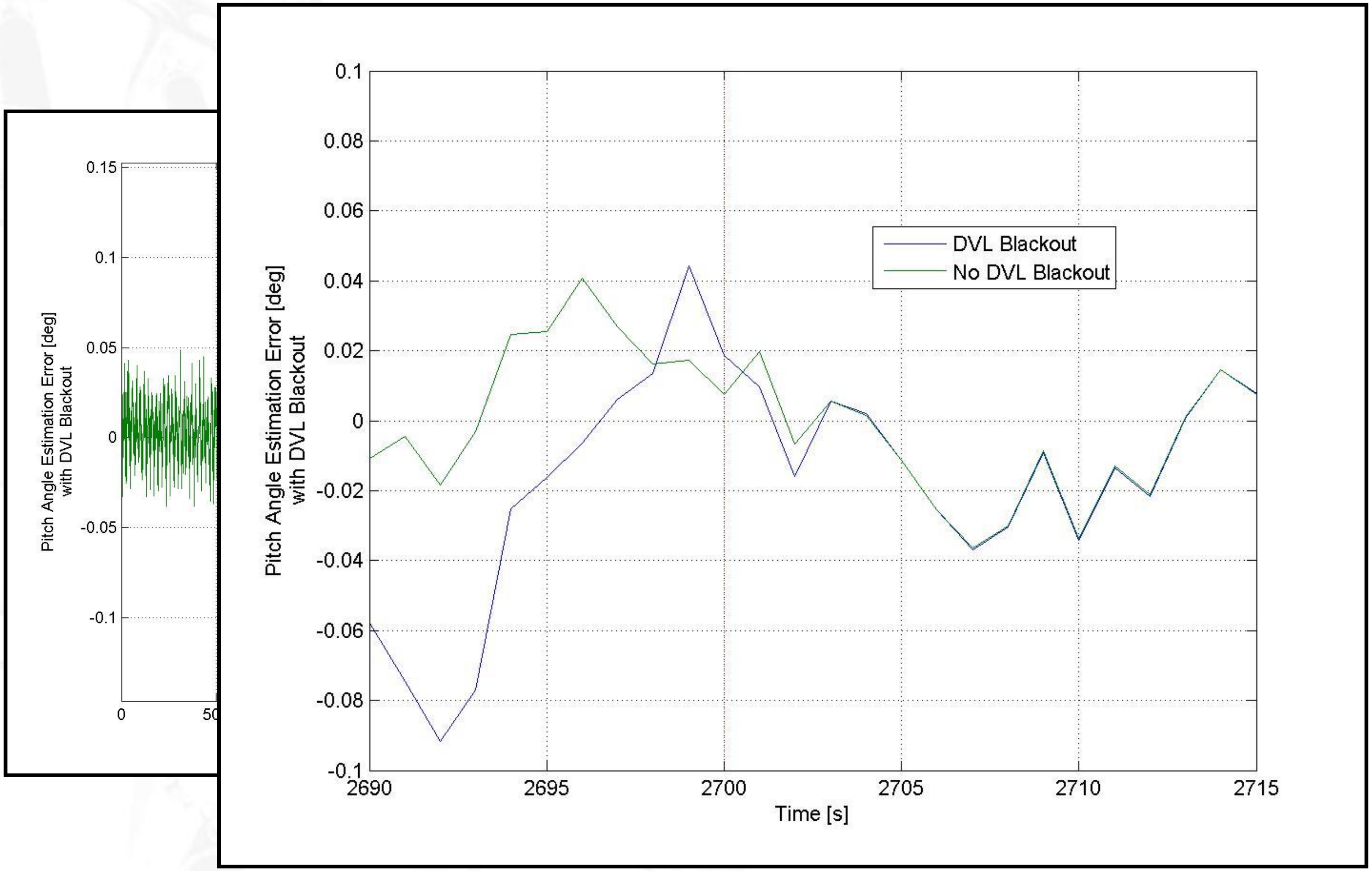
CASE 2 : DVL blackout from $t=900$ to $t=2700$



Simulations

Simulation of long term aiding sensor black-out

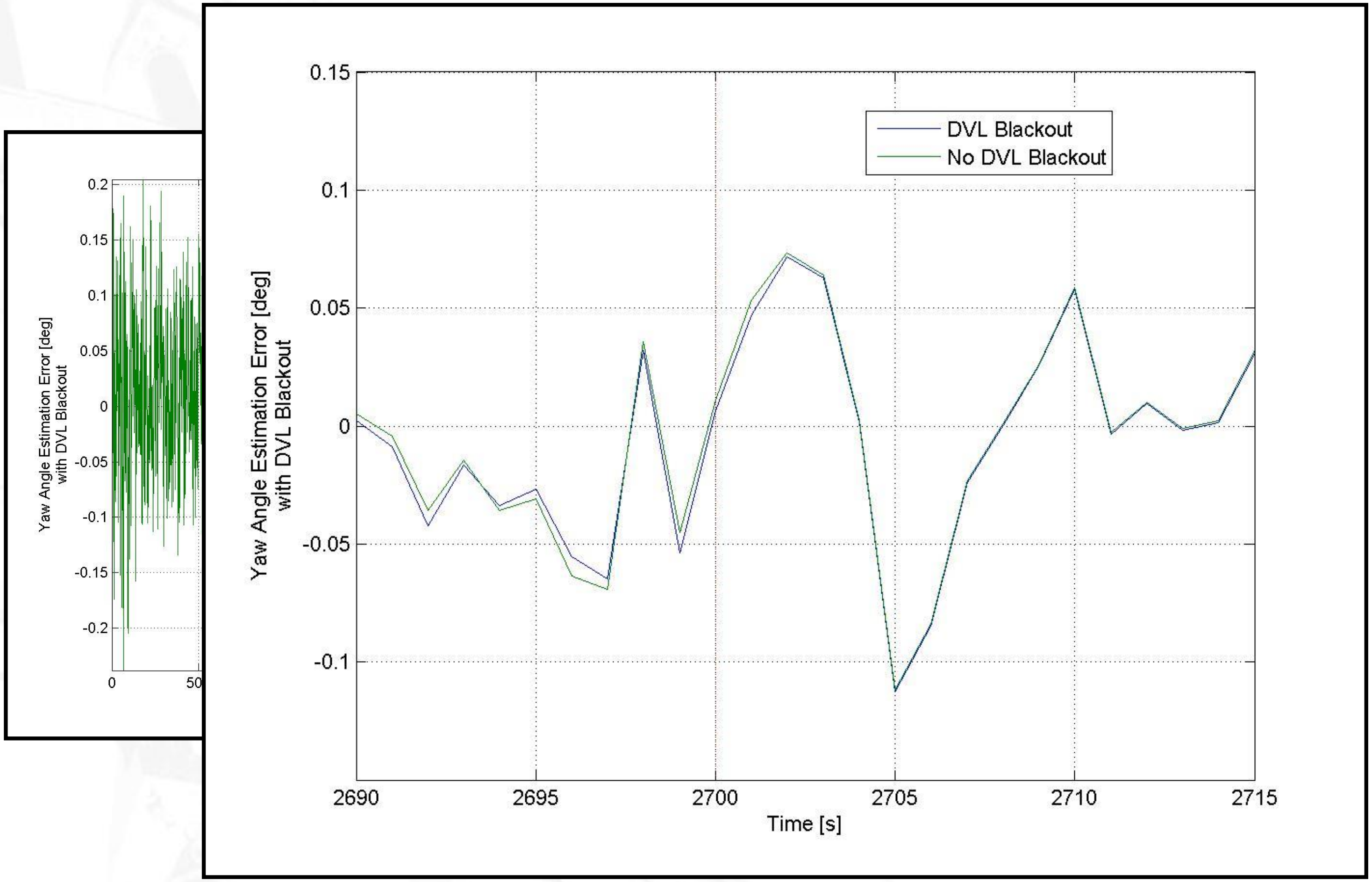
CASE 2 : DVL blackout from $t=900$ to $t=2700$



Simulations

Simulation of long term aiding sensor black-out

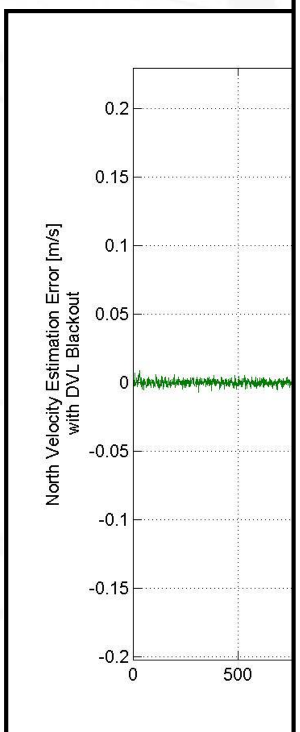
CASE 2 : DVL blackout from $t=900$ to $t=2700$

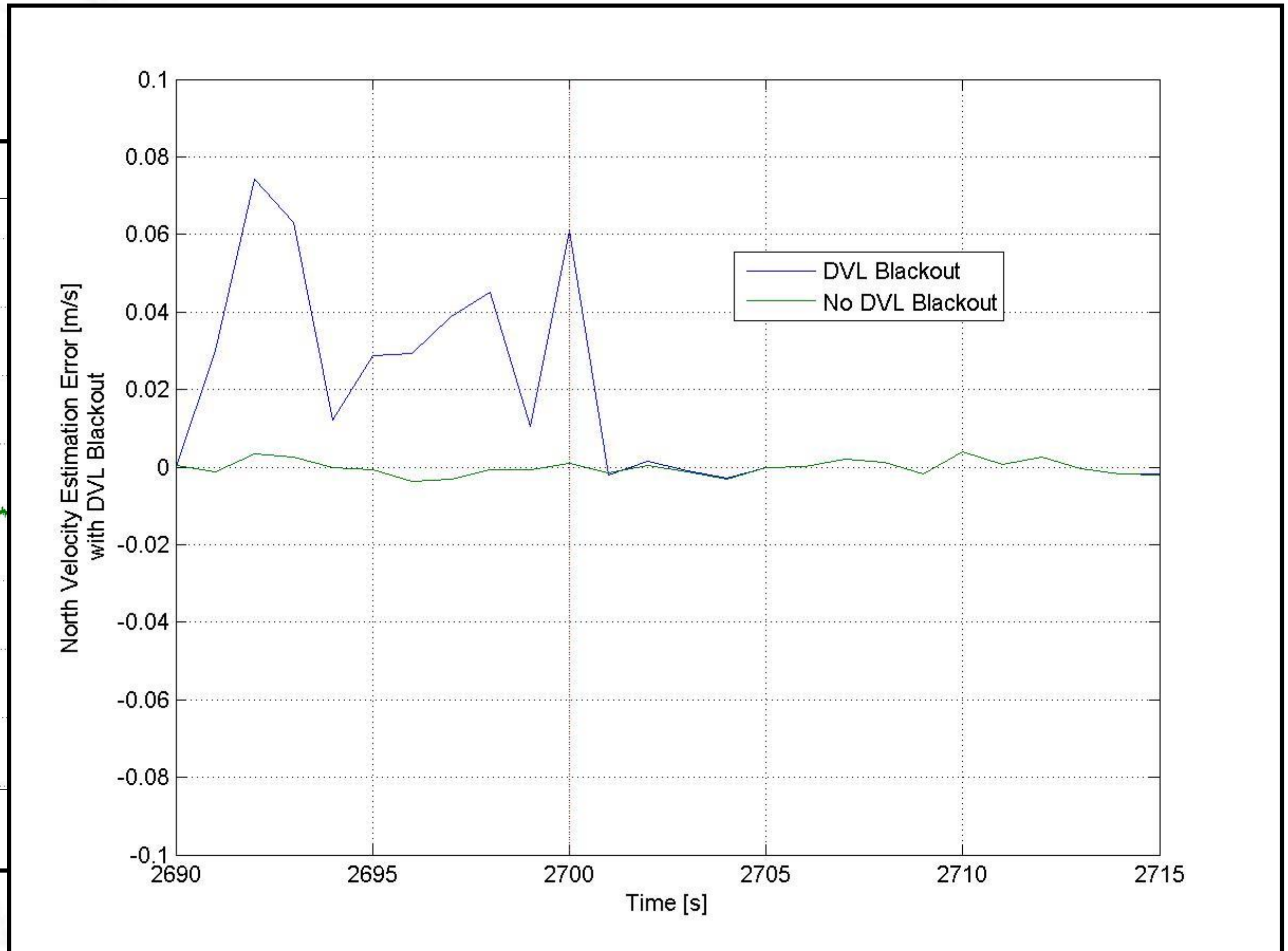


Simulations

Simulation of long term aiding sensor black-out

CASE 2 : DVL blackout from $t=900$ to $t=2700$

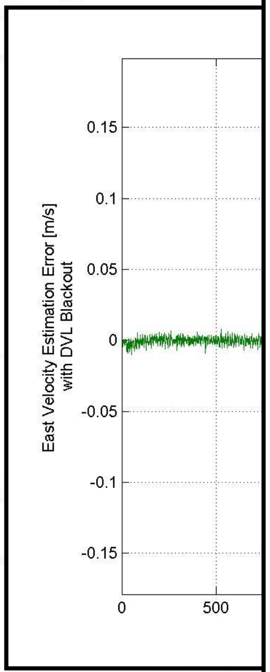


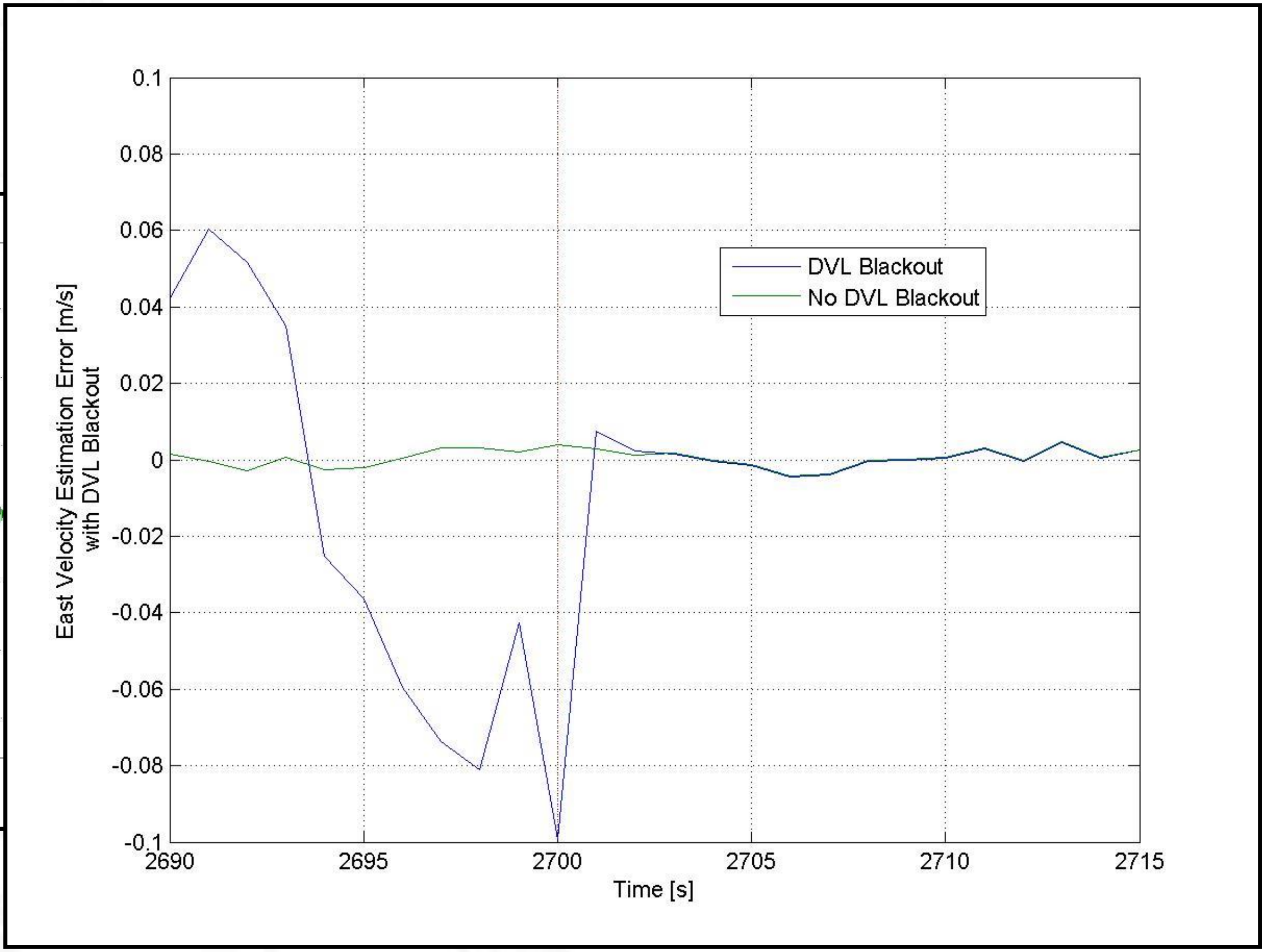


Simulations

Simulation of long term aiding sensor black-out

CASE 2 : DVL blackout from $t=900$ to $t=2700$

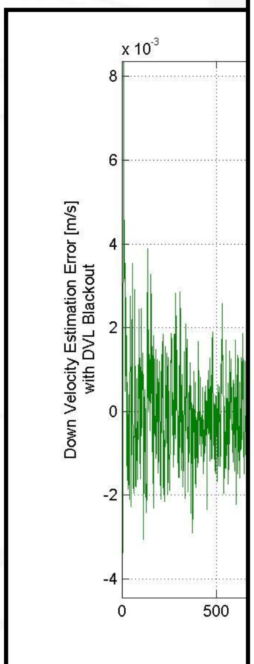


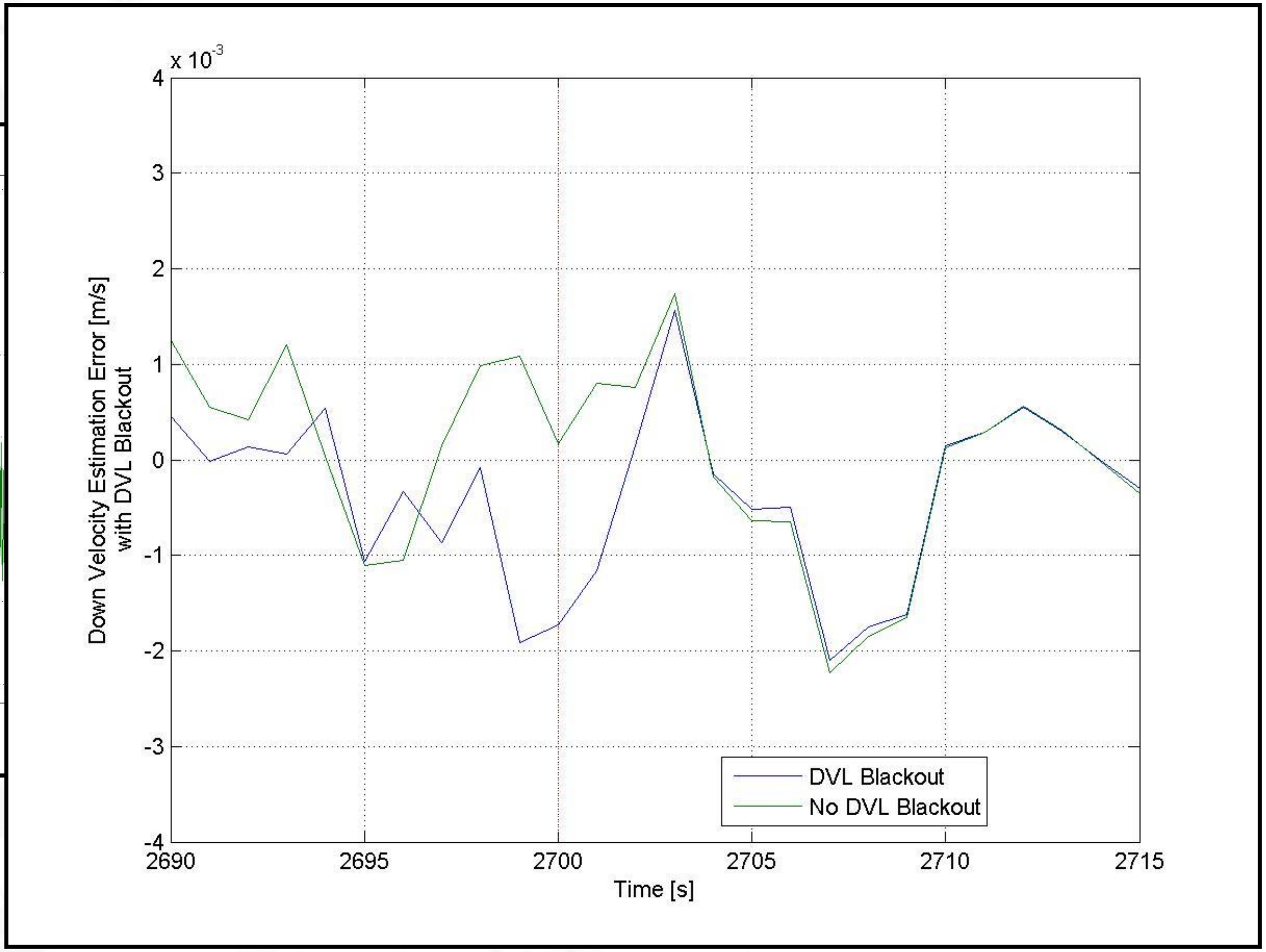


Simulations

Simulation of long term aiding sensor black-out

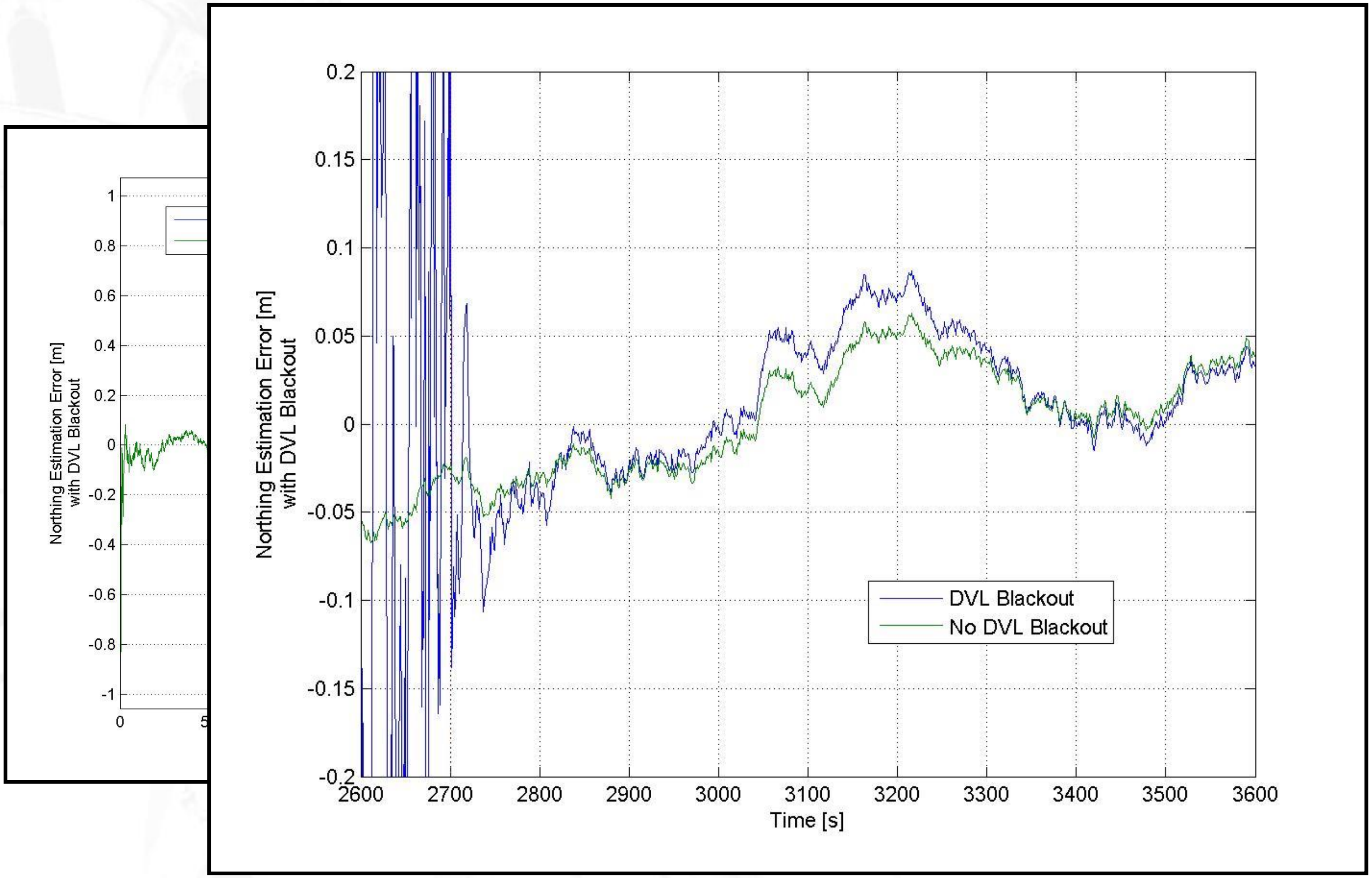
CASE 2 : DVL blackout from $t=900$ to $t=2700$





Simulation of long term aiding sensor black-out

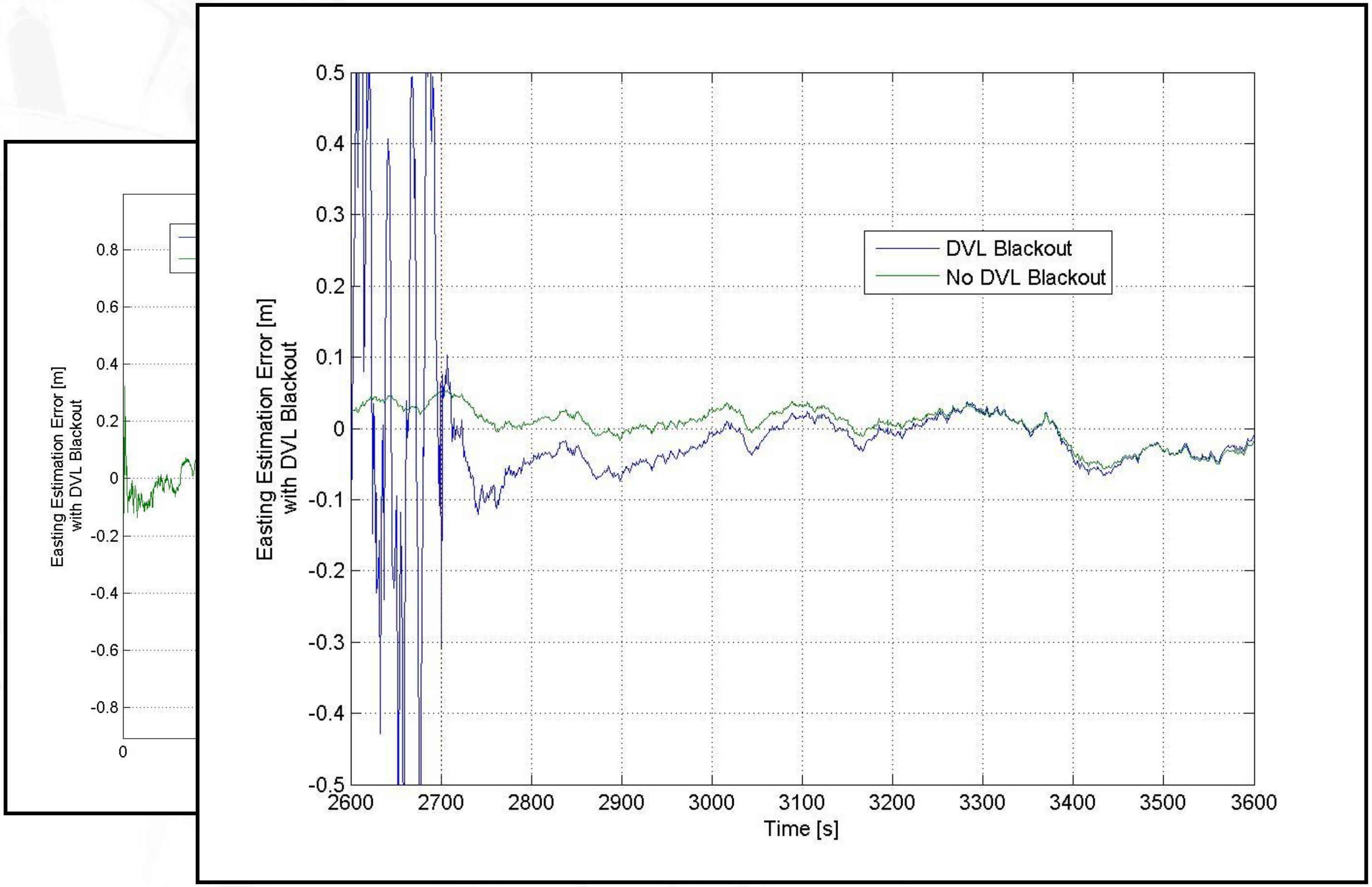
CASE 2 : DVL blackout from $t=900$ to $t=2700$



Simulations

Simulation of long term aiding sensor black-out

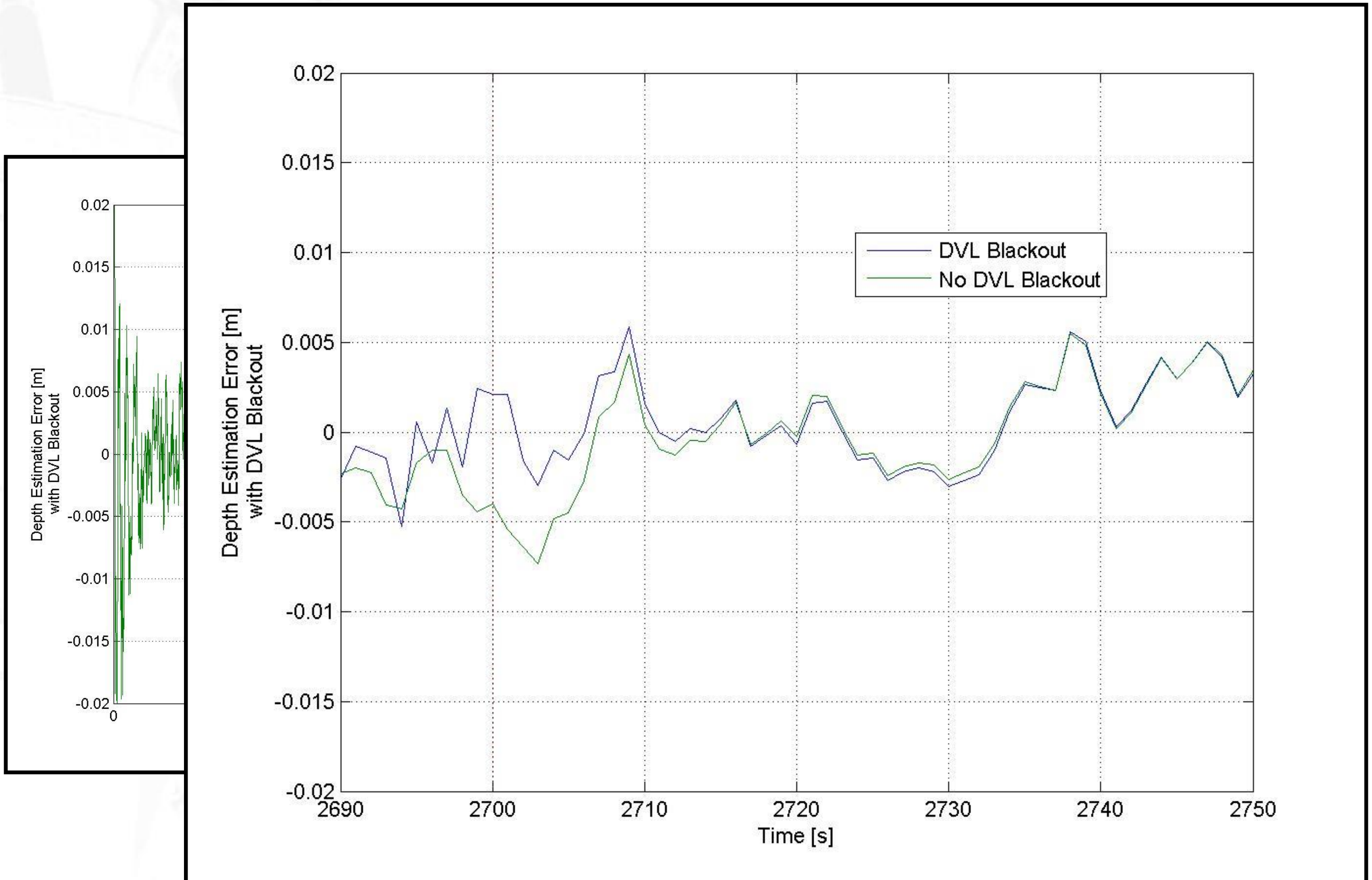
CASE 2 : DVL blackout from $t=900$ to $t=2700$



Simulations

Simulation of long term aiding sensor black-out

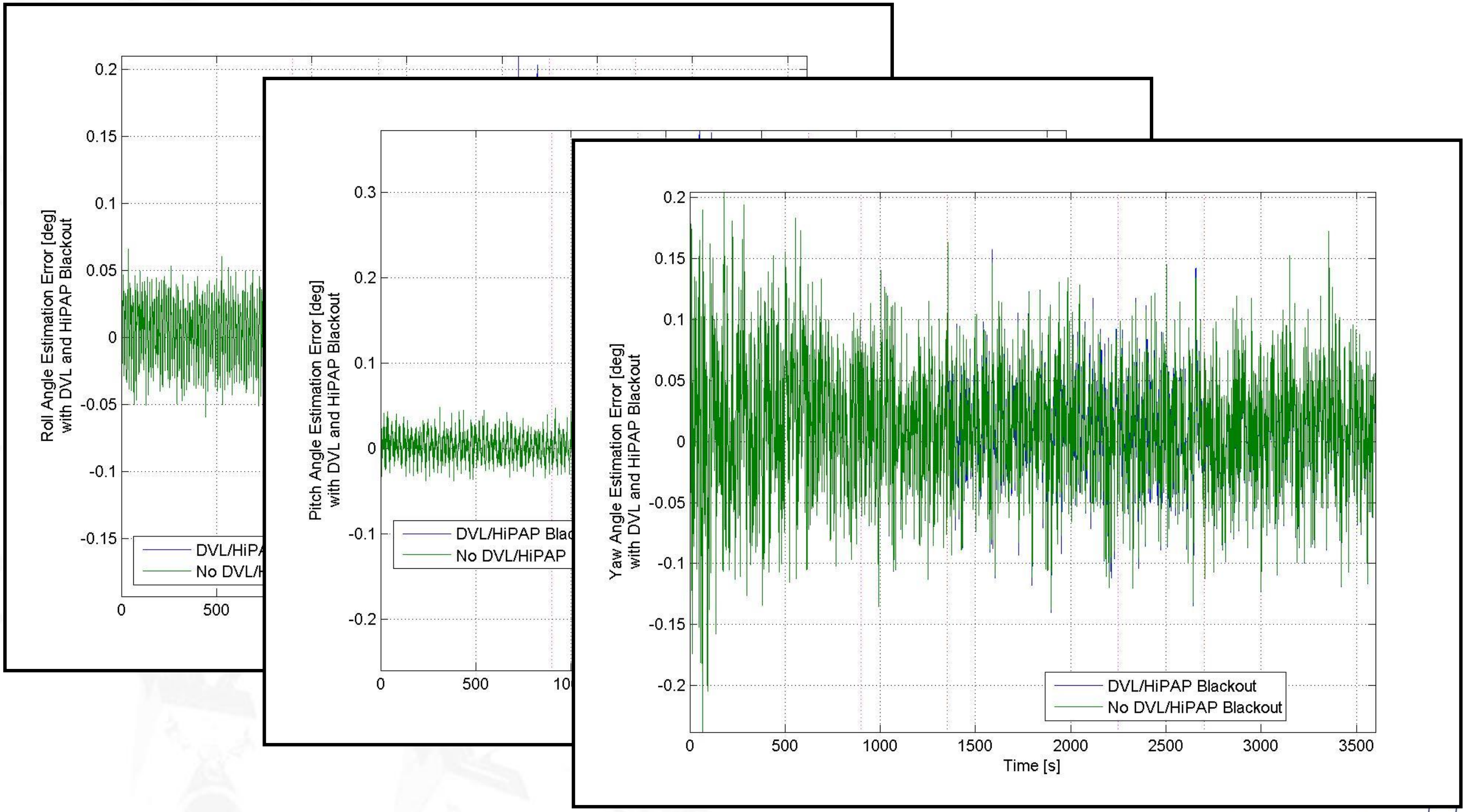
CASE 2 : DVL blackout from $t=900$ to $t=2700$



Simulations

Simulation of long term aiding sensor black-out

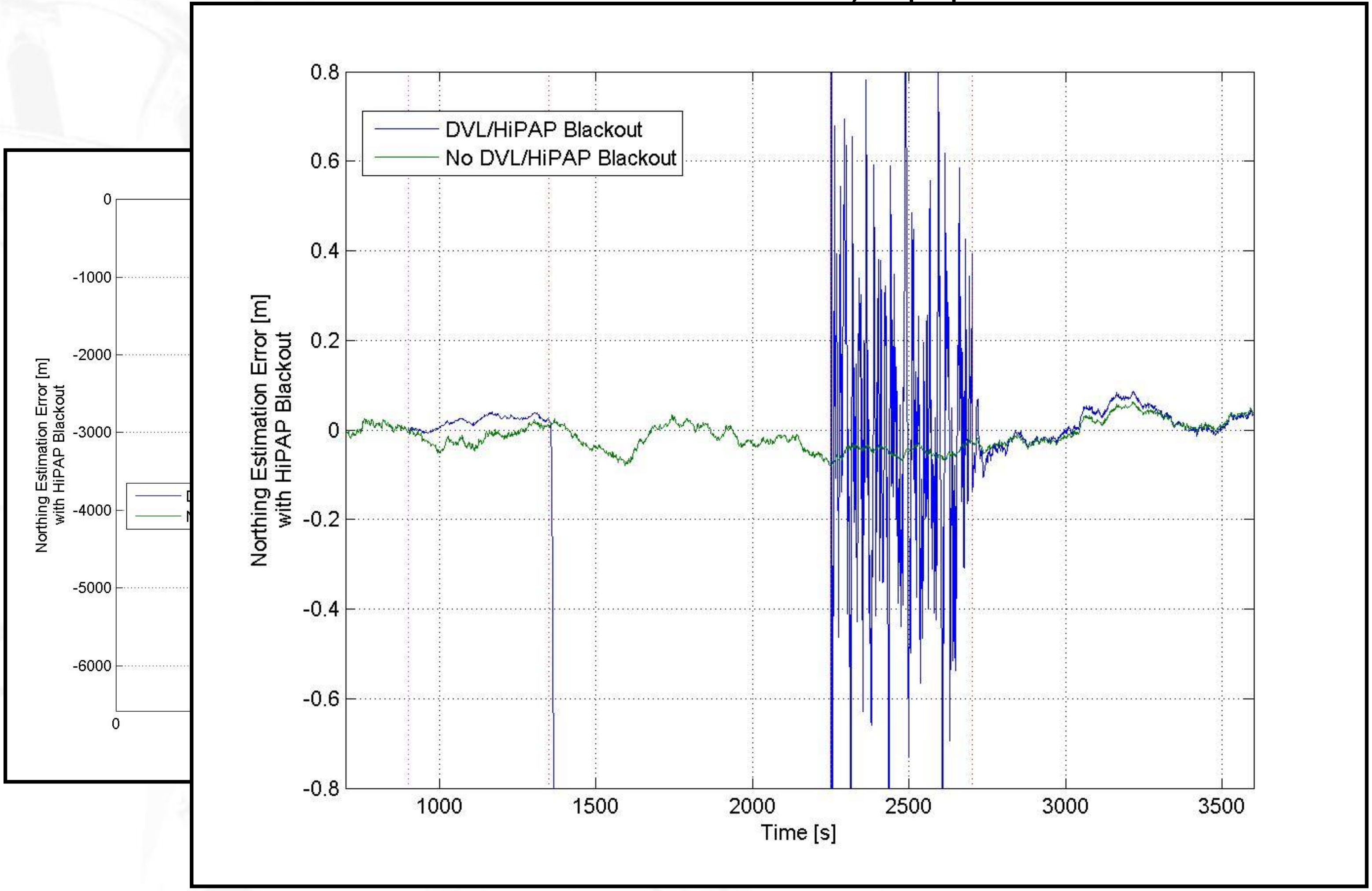
CASE 3 : DVL off from $t=1350$ to $t=2700$, Hipap off from $t=900$ to $t=2250$



Simulations

Simulation of long term aiding sensor black-out

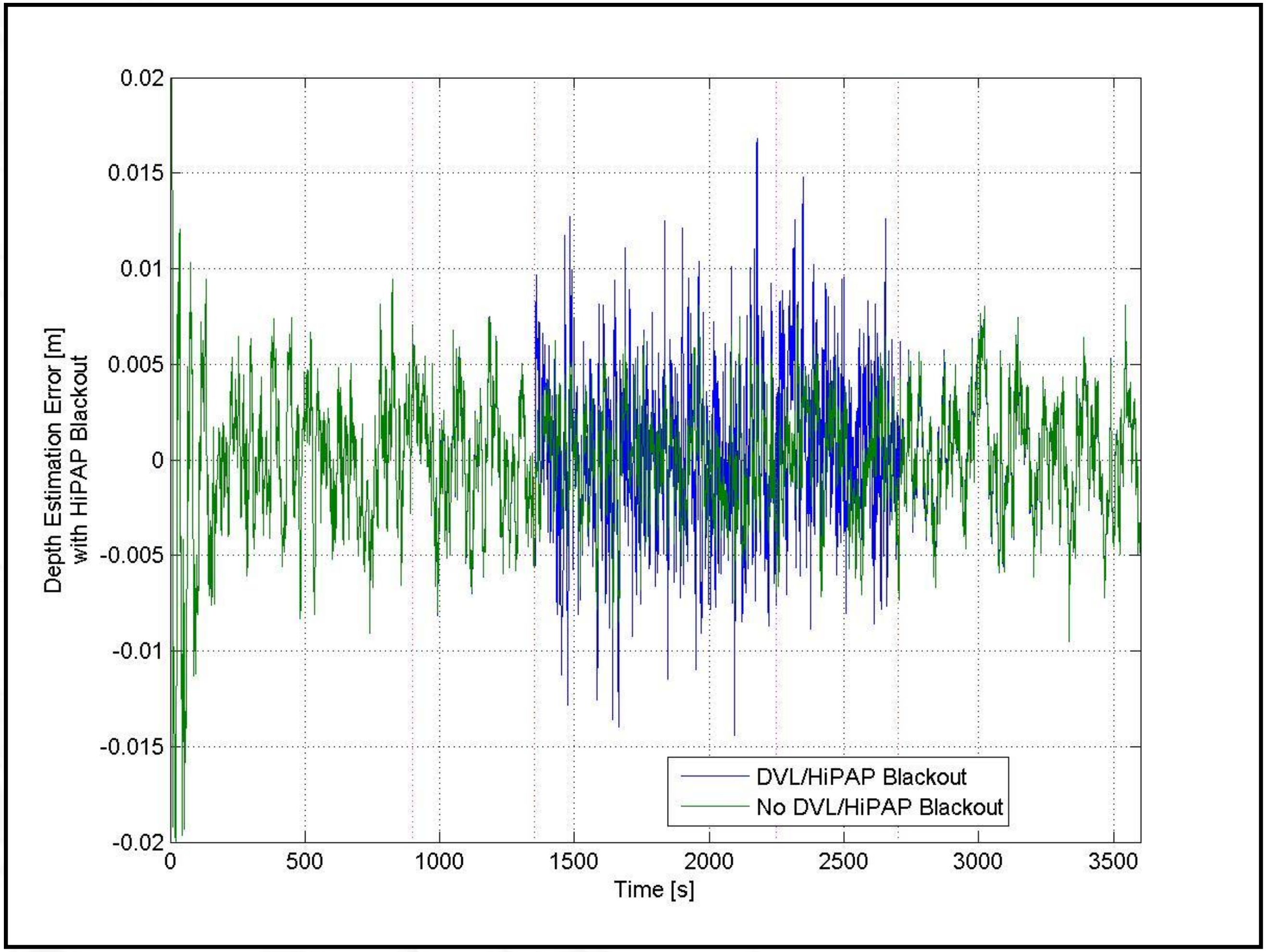
CASE 3 : DVL off from $t=1350$ to $t=2700$, Hipap off from $t=900$ to $t=2250$



Simulations

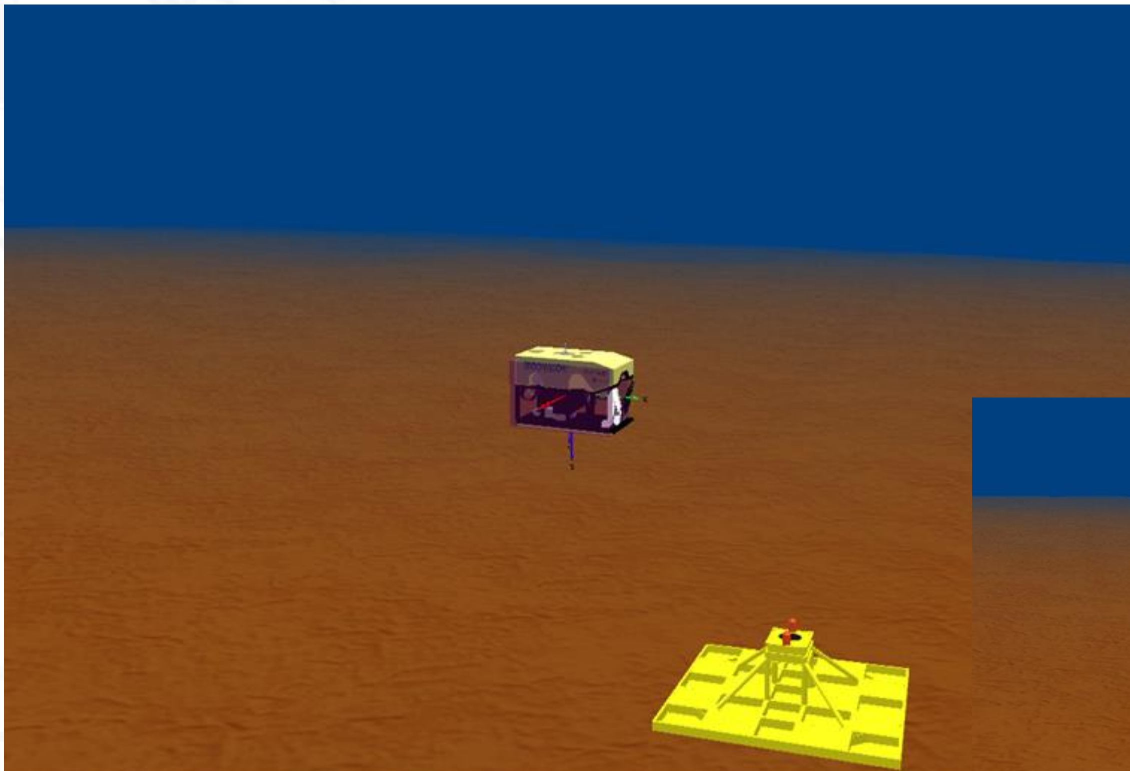
Simulation of long term aiding sensor black-out

CASE 3 : DVL off from $t=1350$ to $t=2700$, Hipap off from $t=900$ to $t=2250$



Closed Loop Simulations with ROV Model

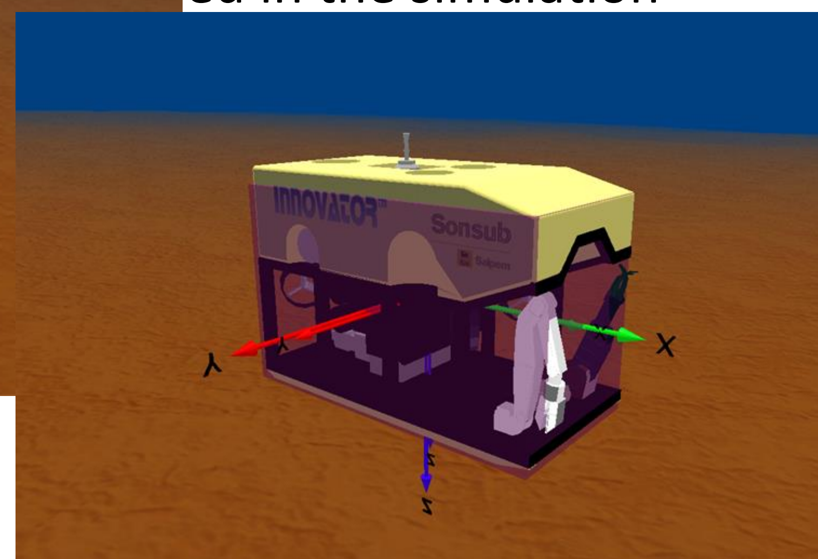
ROV simulation and Closed loop simulations



ts provided by ROV

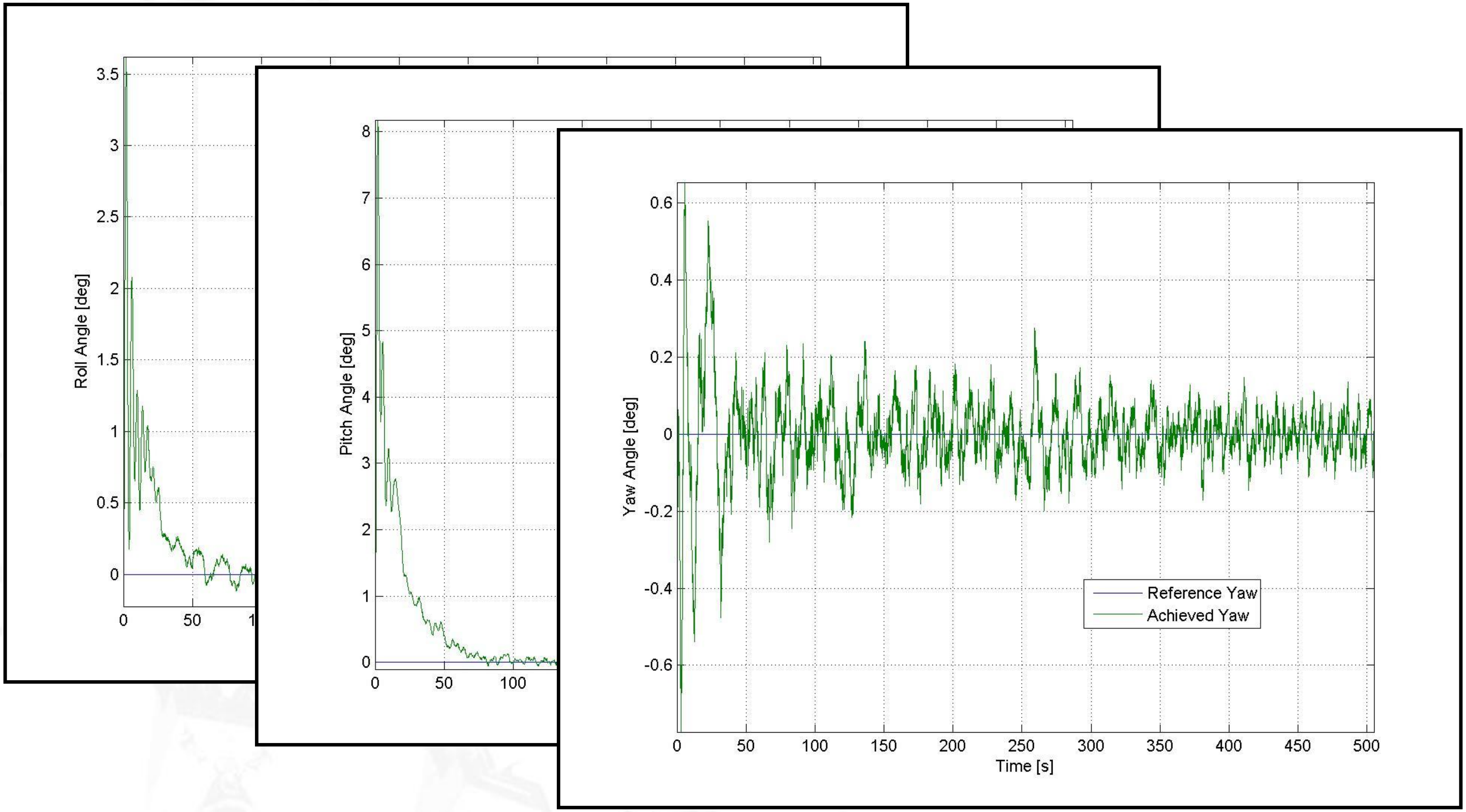
modeling)

ed in the simulation



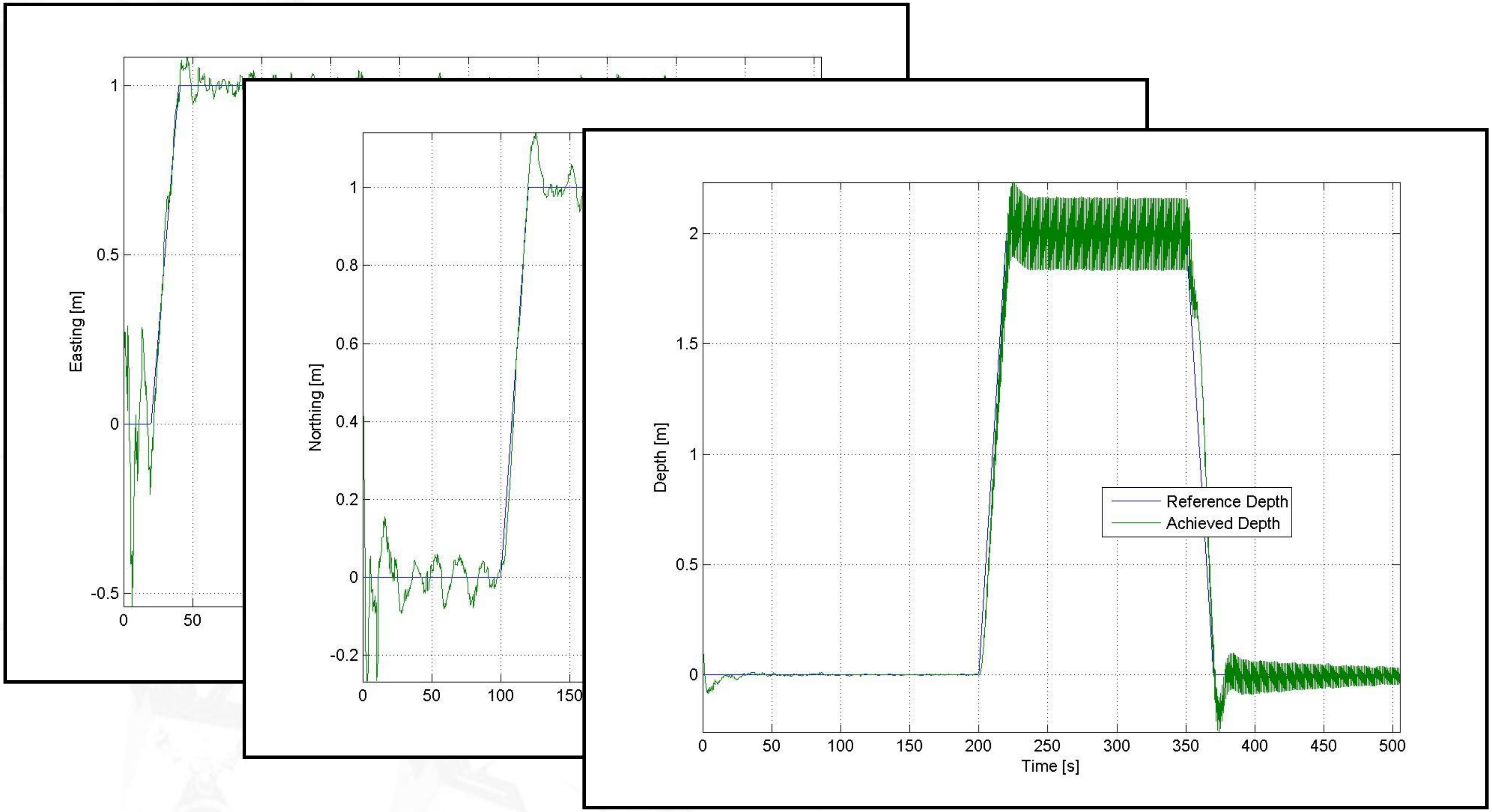
Simulations

ROV simulation and Closed loop simulations



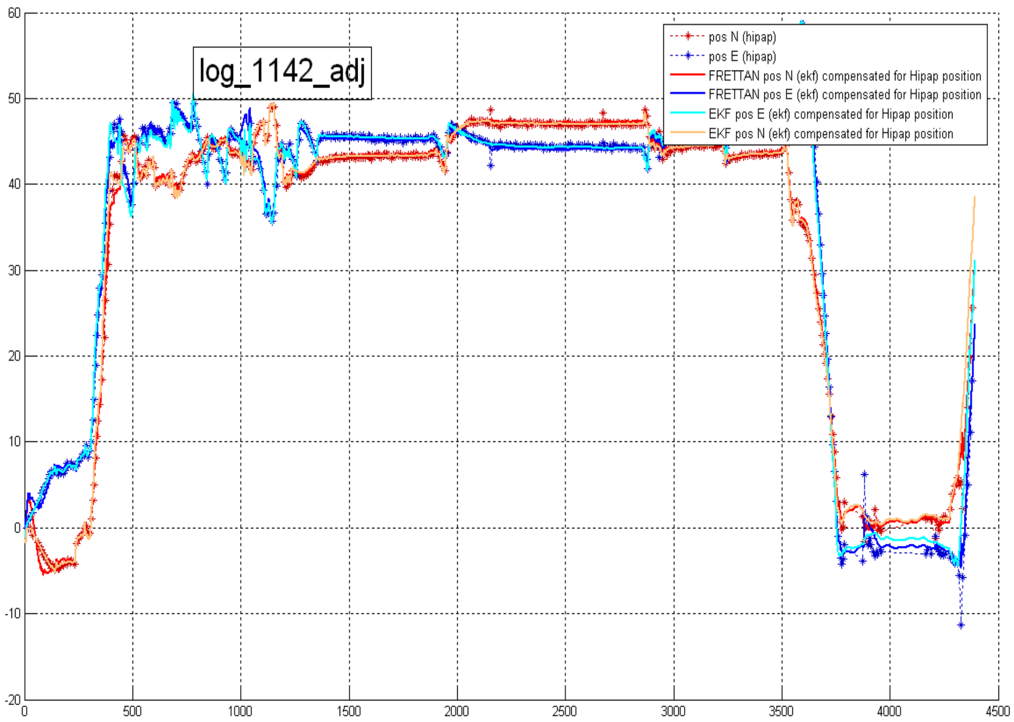
Simulations

ROV simulation and Closed loop simulations



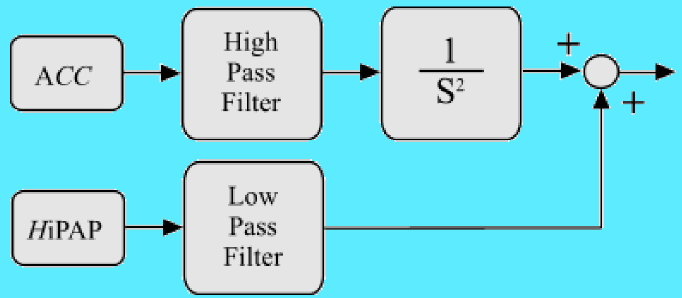
Experimental tests

- Very long recordings (more than 1 hour)
- DVL (1 Hz) and Hipap (0.1 Hz) fixes almost always available

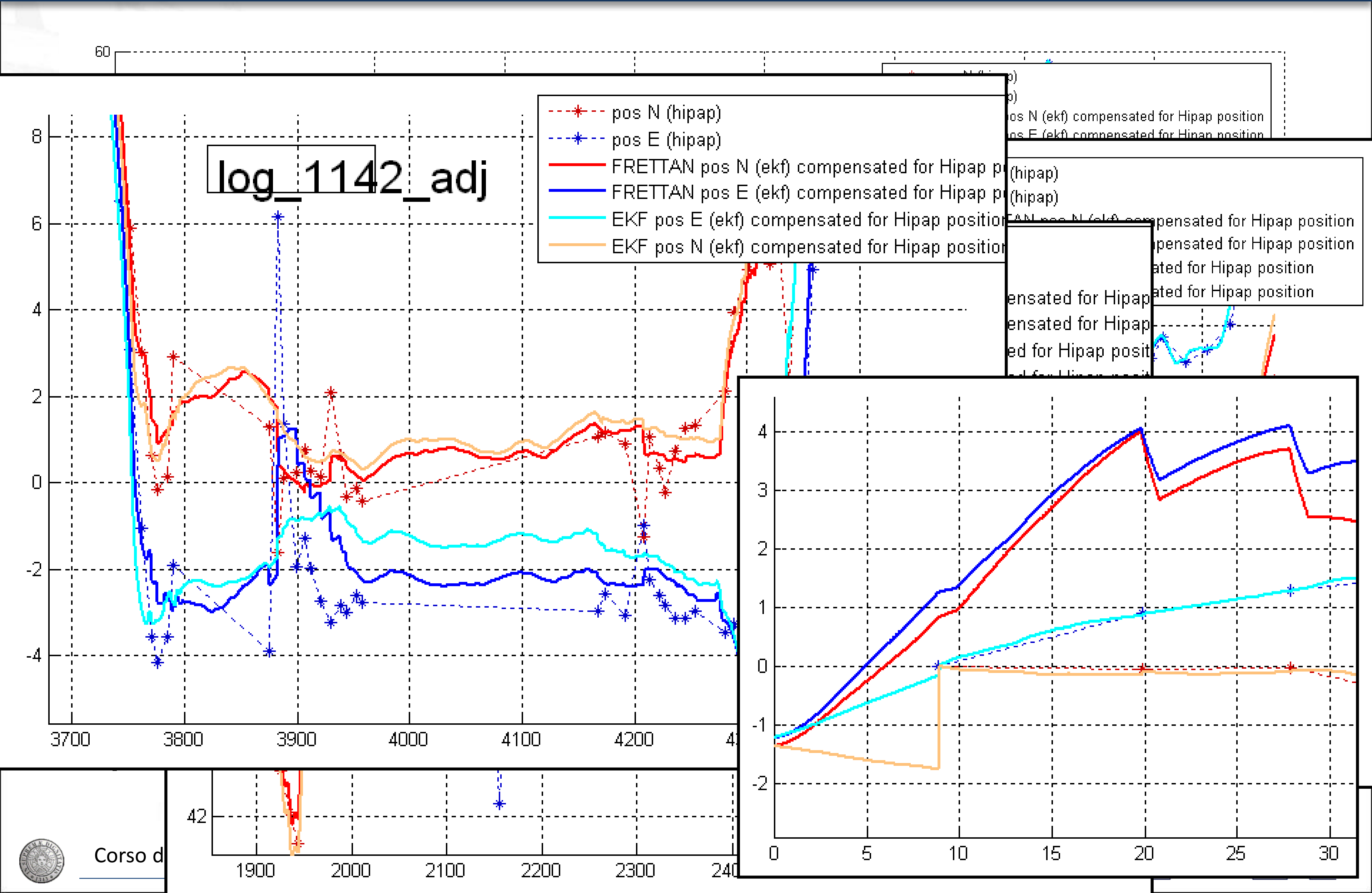


A **Complementary Filter** is used as comparison term

Complementary Filtering exploits the typical frequency separation of inertial errors (low frequency/biases/drifts) and aiding errors (high frequency) by linear filtering

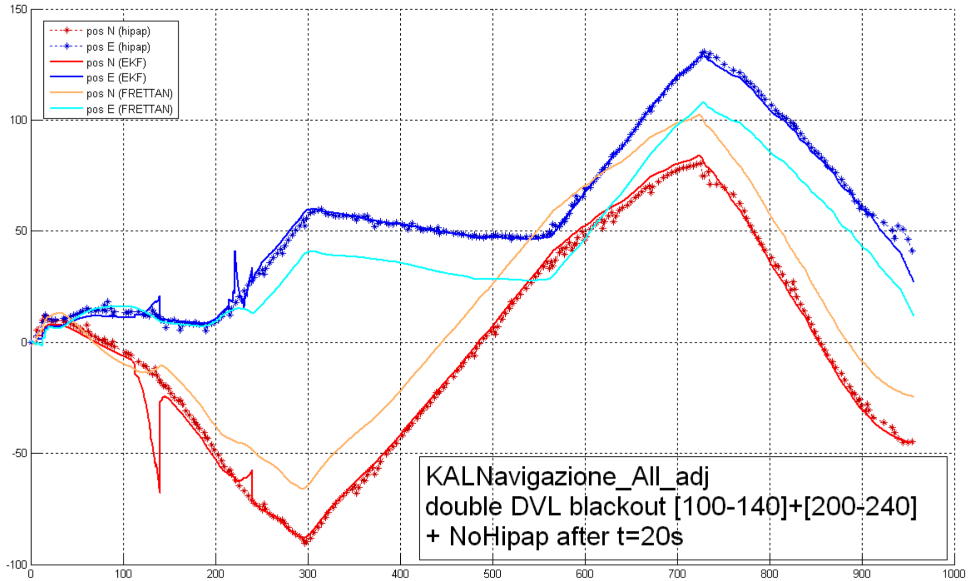


Experimental tests

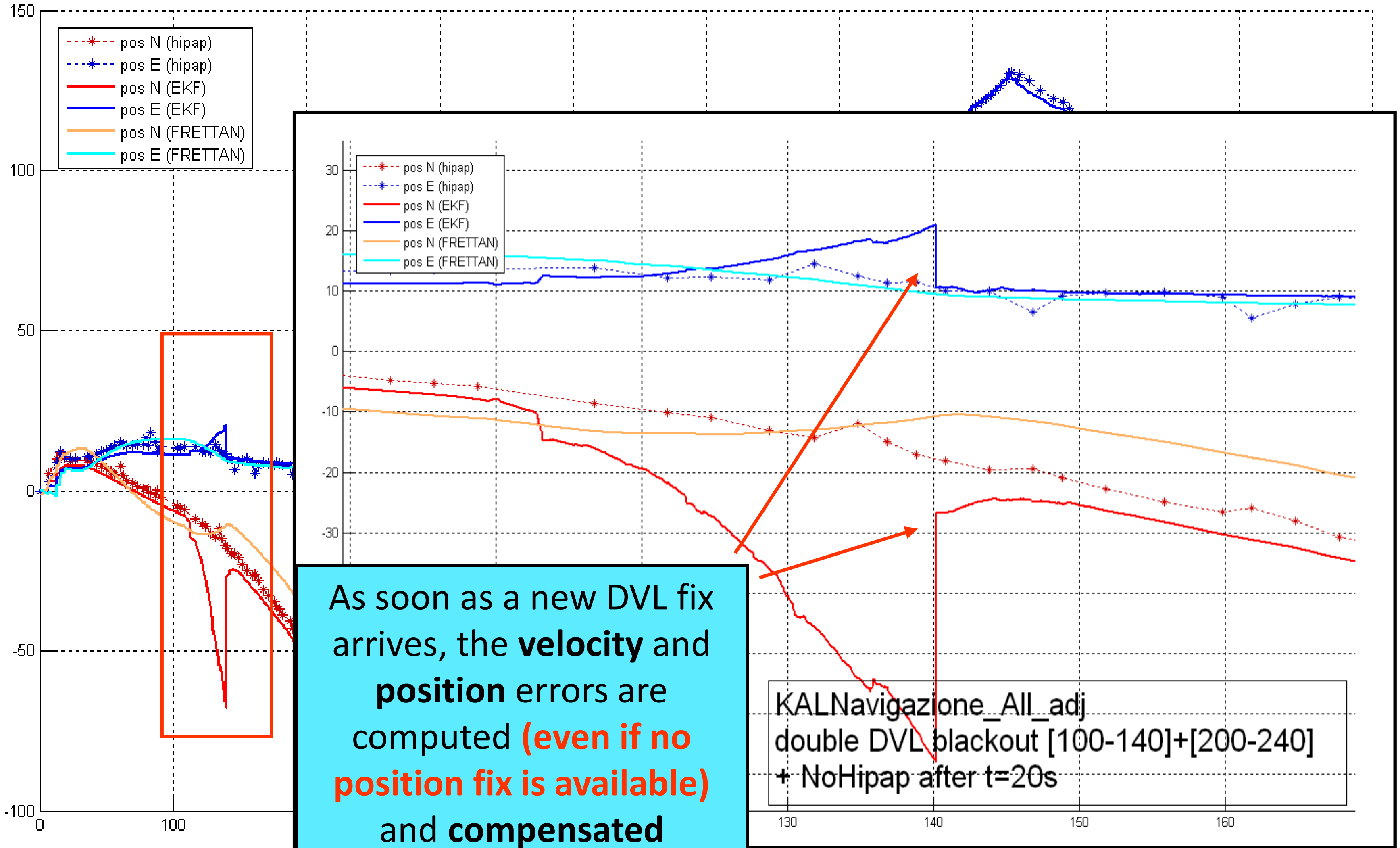


Experimental tests

- 15 minutes of data
- Hipap fixes (of good quality and with low noise) used for 20 seconds only – after that used as gound truth for validation
- 2 simulated DVL blackouts [100s to 140s] and [200s to 240s] that force the system in almost-pure inertial mode



Experimental tests



As soon as a new DVL fix arrives, the **velocity** and **position** errors are computed **(even if no position fix is available)** and **compensated**

Experimental tests