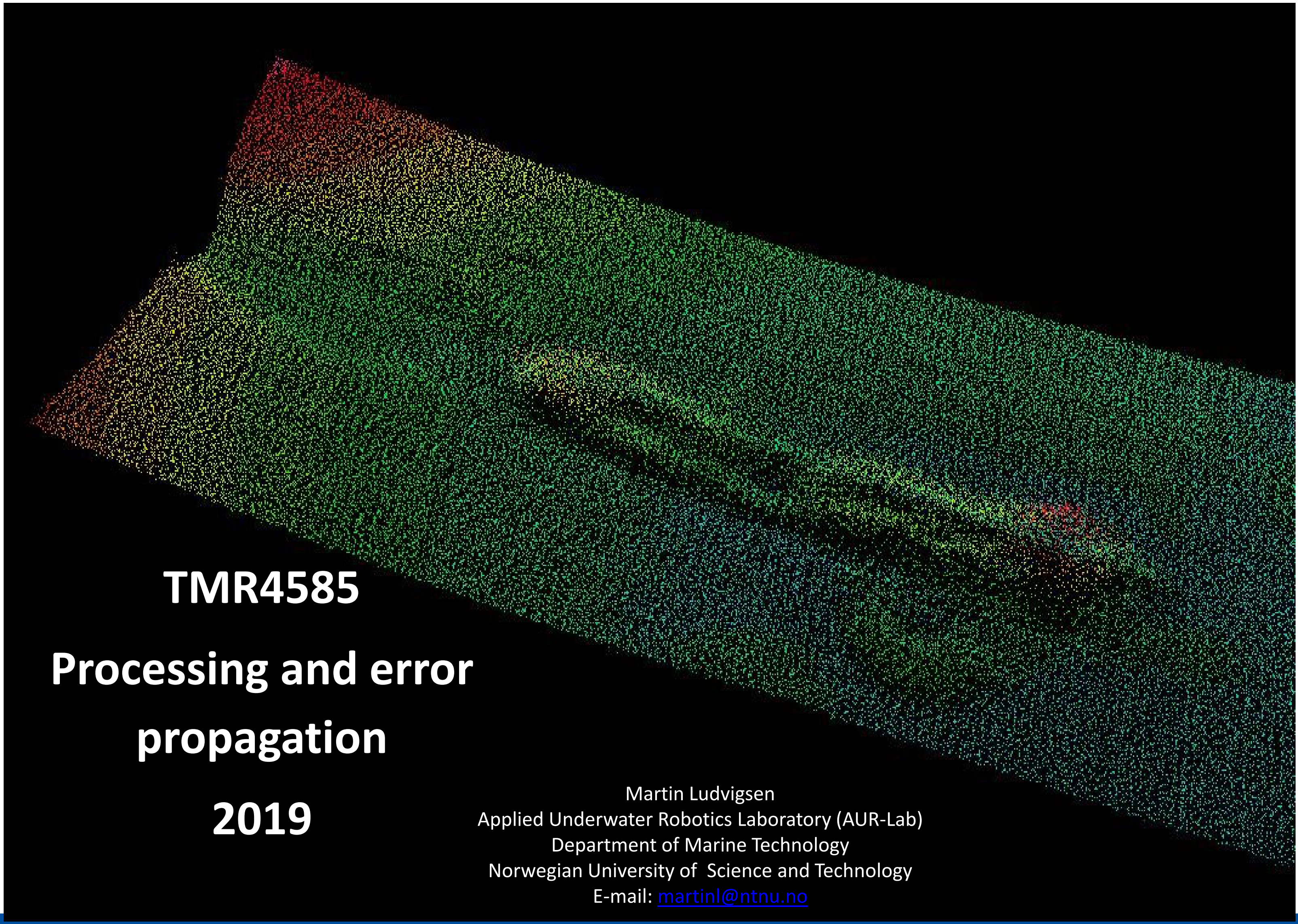


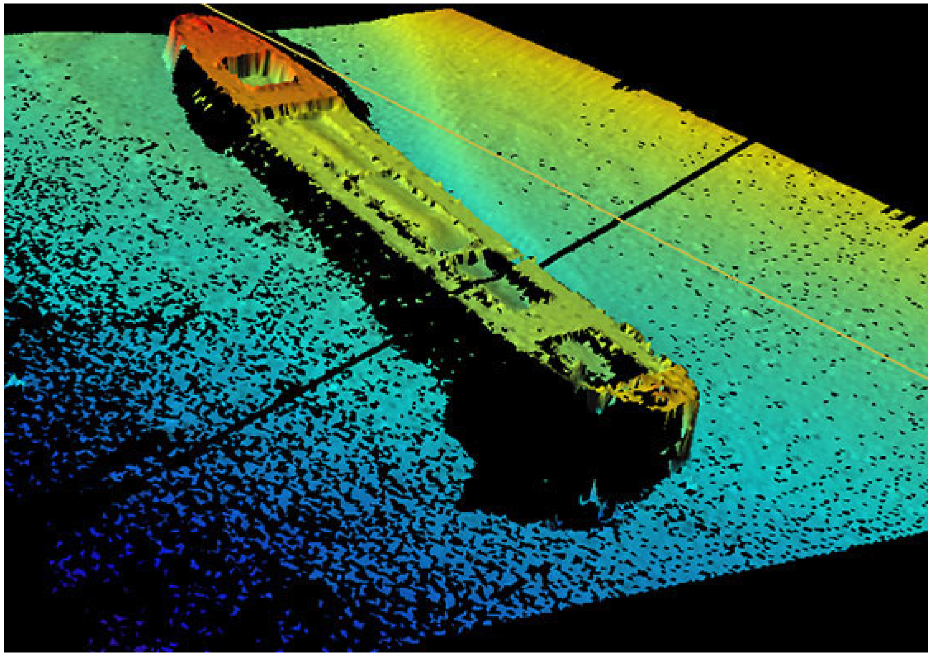
TMR4585

Processing and error propagation

2019

Martin Ludvigsen
Applied Underwater Robotics Laboratory (AUR-Lab)
Department of Marine Technology
Norwegian University of Science and Technology
E-mail: martinl@ntnu.no

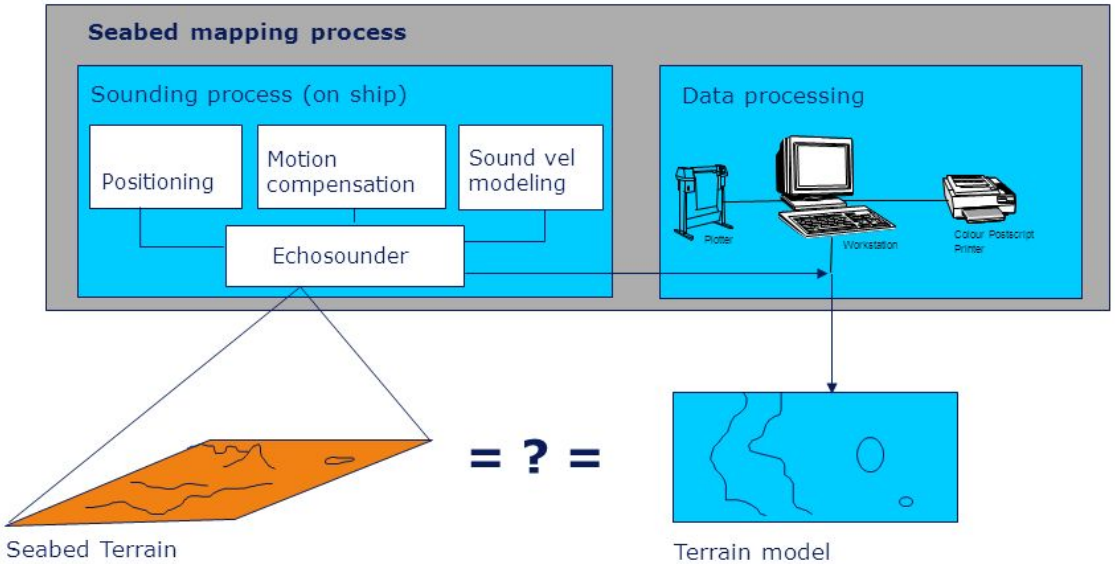
Kahoot



Learning objectives

- Data processing
- Error propagation
- Error budgets
- Timing and latency
- Calibrations

Seabed mapping from ping to map



How well does the terrain model represent the seabed terrain?

Sonar survey data processing

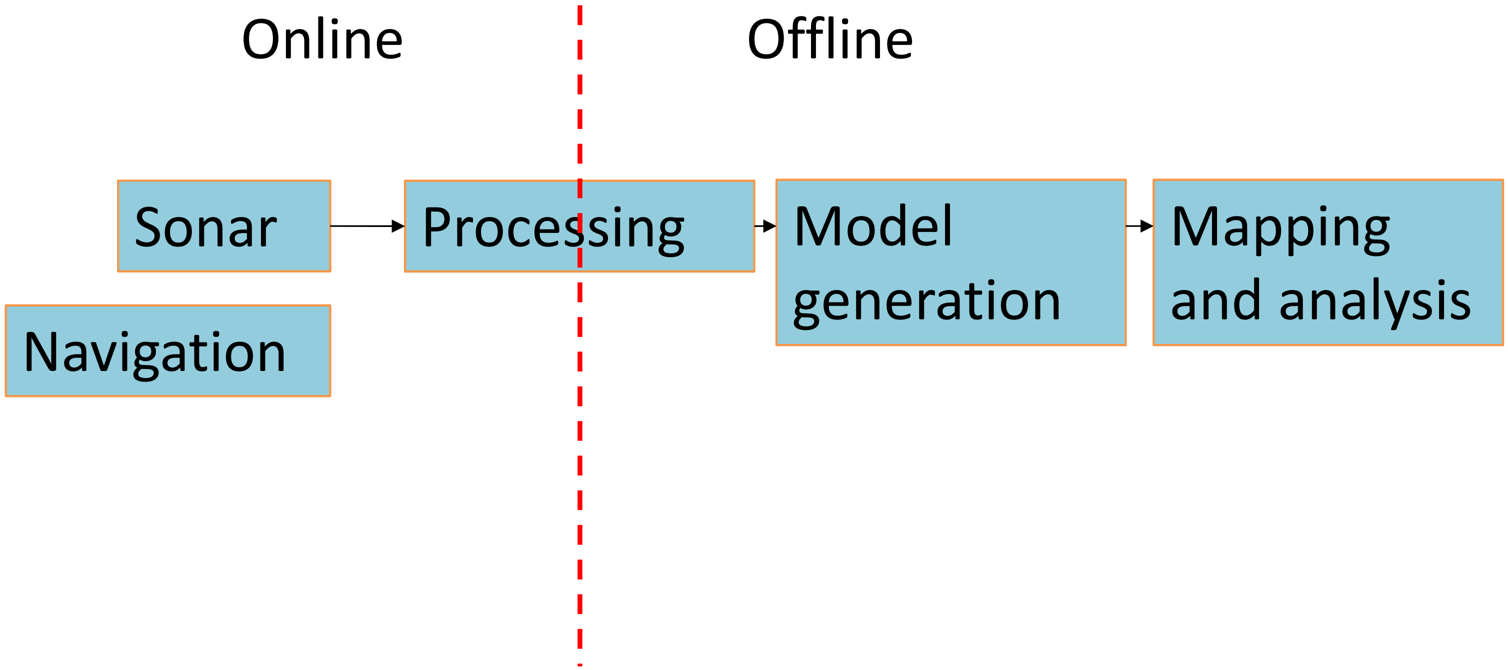
Online

- Logging
- Data storage
- Data integrity
  - Frozen signals
  - Standard deviation
  - Jumps

Offline

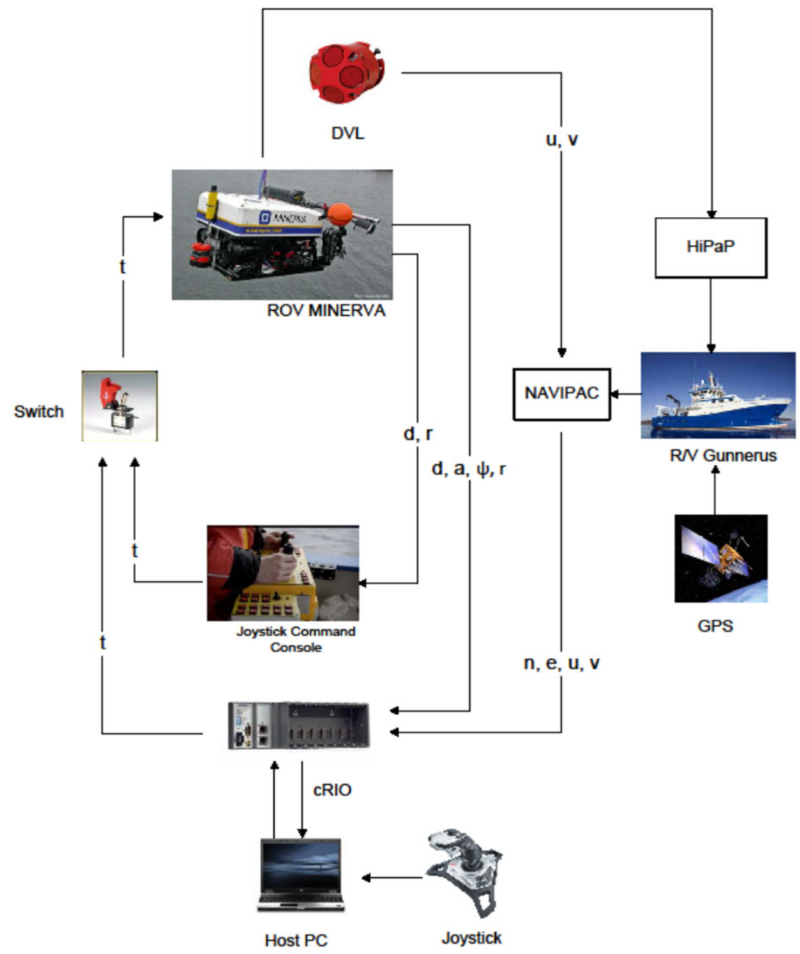
- Compensation
  - Motion
  - Speed of sound
- Data filtering
- Gridding
- Modelling
- Maps

Processing sonar mapping data

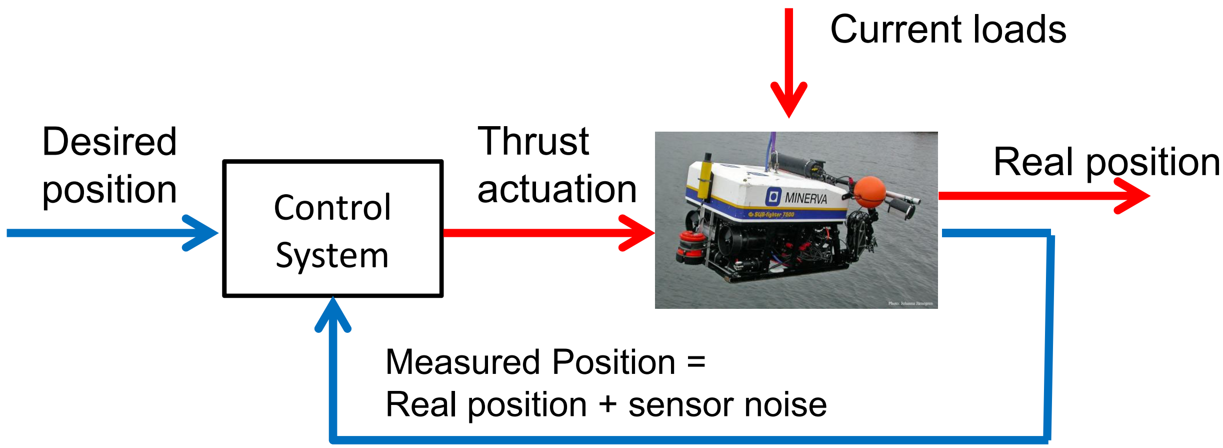


Control systems

- Minerva implementation



Feedback control



The control system will dominate the dynamics, and we have to consider a new dynamic system – "closed-loop system"

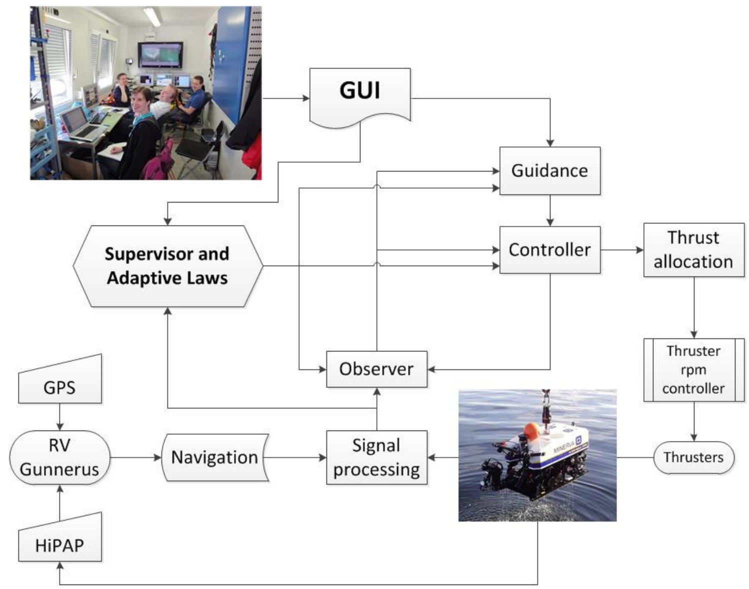
The control system is also a dynamic system with

- internal states – state feedback or output feedback using observers
- actuator dynamics, saturation, control allocation, power limitation
- sensor dynamics, integration and accuracy
- computers conducting real time control

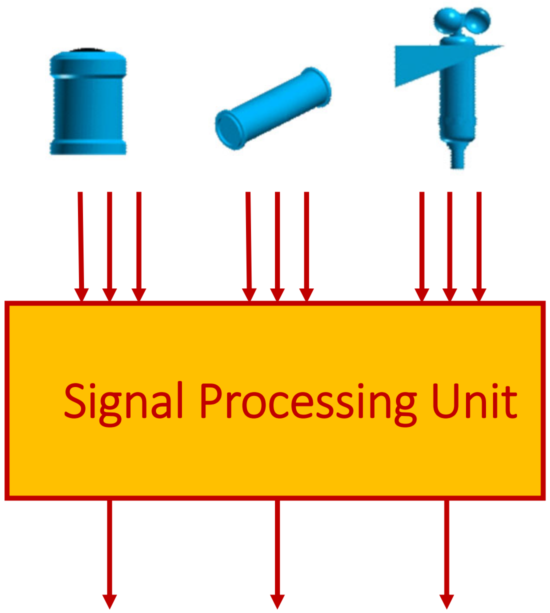
Control system

- Station keeping
- Track lines

- GUI – user
- Guidance
- Navigation
- Controller



Signal processing



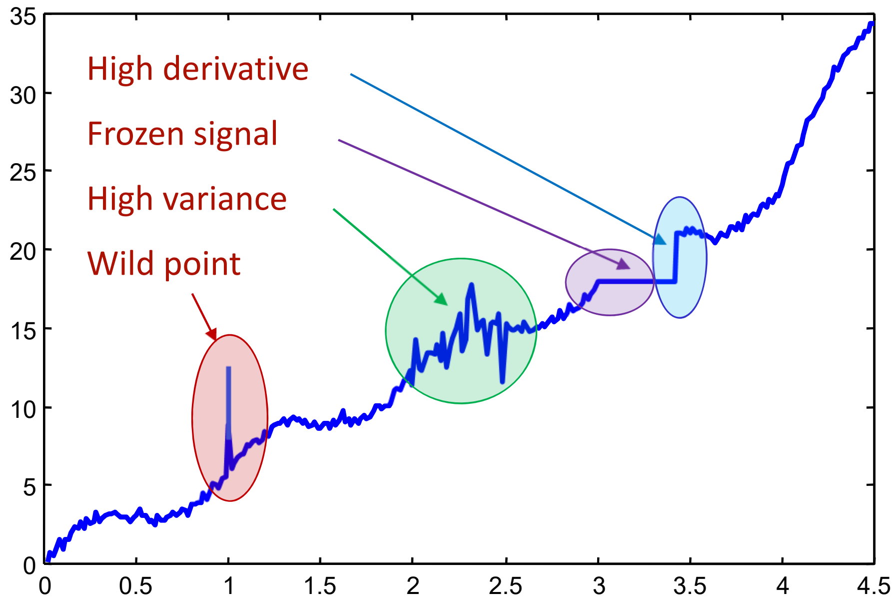
HW Signal Communication already checked

Features:

- Online Signal Quality Check
- Online weighting of sensor signals
- Multiple signal voting algorithms
- Filtering and smoothing of signals

Signal processing - example

Examples of four different signal failures the signal QA module is detecting.



Signal quality checking

Three level testing:

1. Tests on individual signals
   - Range check
   - Variance check
   - Wild point detection and removal
2. Sensor voting
   - Detection of sensor drift
3. Sensor weighting
   - Unbiased minimum variance measurements
   - Manual weighting

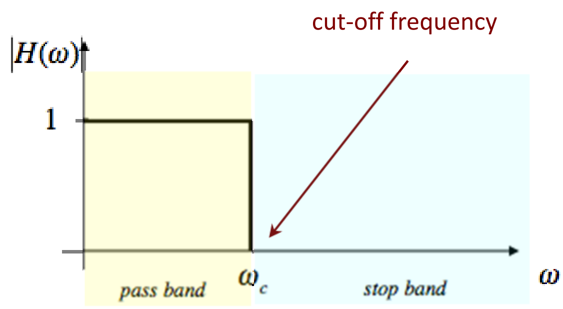
If observer (state estimator such as Kalman filter):

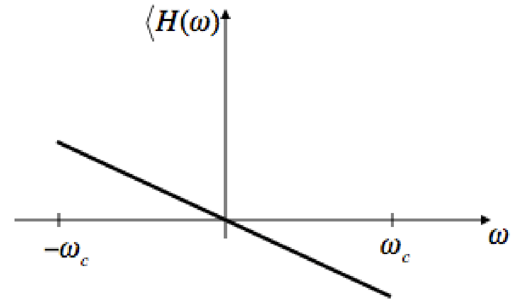
4. Observer test/Prediction error (Innovation/injection term)

Conventional filters used in marine control systems

- Low pass filter suppressing e.g. noise
- High pass filter
- Band stop filters, e.g. notch filter
- Band pass filter
- Cascaded low pass and notch filter for wave filtering

Ideal Filters: lowpass case





Linear Phase: all the harmonic components in the passband are delayed of the same amount of time.
In other words: delay t_0 independent by the angular velocity: $f(t-t_0) \rightarrow F(\omega)e^{-j\omega t_0}$

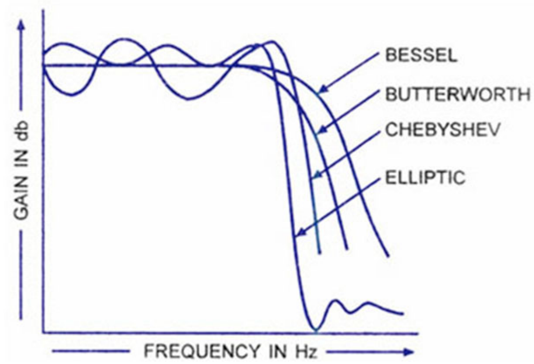
Can this filter be realized?

Response of a passband ideal filter (Fourier Transform) is described by $y(t) = \frac{\sin(\omega_c t)}{t}$ (sinc):

- not causal ⇨ cannot be obtained!

Low pass filters comparison

Causal filter type.	Gain approximation	Linearity of the phase
Butterworth	good	good
Chebyshev	good	bad
Elliptic	very good	very bad
Bessel	bad	good



A first order low-pass filter with time constant T_f can be designed according to:

$$h_{lp}(s) = \frac{1}{1+T_f s} \qquad \omega_b < \frac{1}{T_f} < \omega_e \qquad \left[\frac{\text{rad}}{s}\right]$$

This filter will suppress disturbances over the frequency $\frac{1}{T_f}$

remember the effect on the Bode diagram?

Butterworth Filter

- Good characteristics concerning the passband.
- Transition band characteristics are less good.
- It doesn't have linear phase in the passband but we can consider it as a good approximation.
- It can realize higher order low-pass filters:

$$h_{lp}(s)=\frac{1}{p(s)}$$

p(s) is found by solving the Butterworth polynomial:

$$p(s)p(-s)=1+\left(\frac{s}{j\omega_f}\right)^{2n}$$

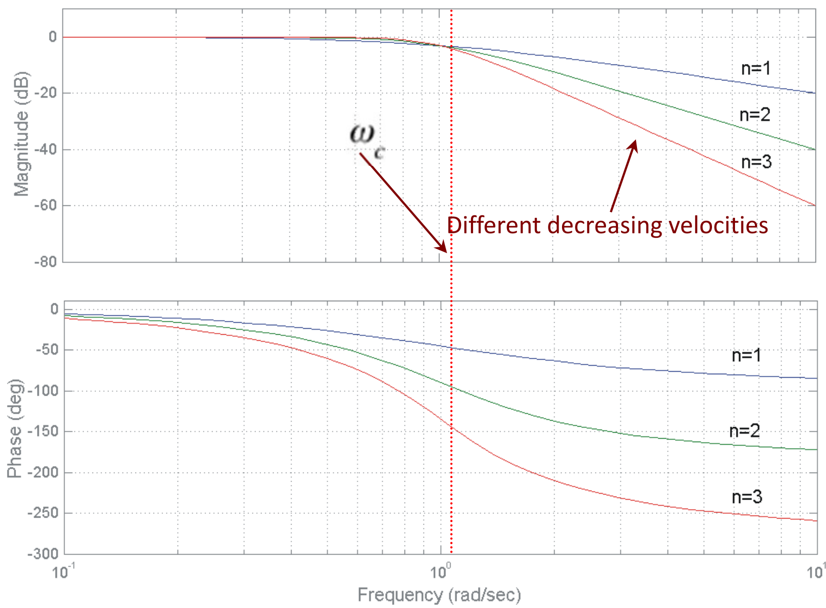
$$(n=1) \quad h_{lp}(s)=\frac{1}{1+\frac{s}{\omega_f}}$$

$$(n=2) \quad h_{lp}(s)=\frac{\omega_f^2}{s^2+2\zeta\omega_f s+\omega_f^2}; \qquad \zeta=\sin(45°)$$

$$(n=3) \quad h_{lp}(s)=\frac{\omega_f^2}{s^2+2\zeta\omega_f s+\omega_f^2}\cdot\frac{1}{1+\frac{s}{\omega_f}}; \qquad \zeta=\sin(30°)$$

$$(n=4) \quad h_{lp}(s)=\prod_{i=1}^{2}\frac{\omega_f^2}{s^2+2\zeta_i\omega_f s+\omega_f^2}; \qquad \zeta_1=\sin(22.5°), \quad \zeta_2=\sin(67.5°)$$

Butterworth Filter



$$n=1:\ b(s)=\frac{\omega_c}{s+\omega_c}$$

$$\dot{x}_f + \omega_c x_f = \omega_c x$$

$$n=2:\ b(s)=\frac{\omega_c^2}{s^2+\sqrt{2}\omega_c s+\omega_c^2}$$

$$\ddot{x}_f + \sqrt{2}\omega_c \dot{x}_f + \omega_c^2 x_f = \omega_c^2 x$$

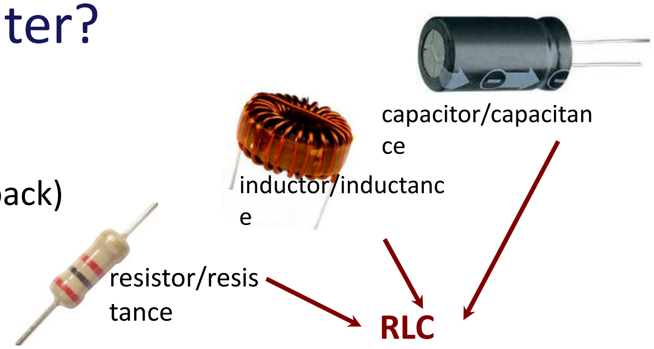
$$n=3:\ b(s)=\frac{\omega_c^3}{\left(s^2+\omega_c s+\omega_c^2\right)\left(s+\omega_c\right)}$$

$$\dddot{x}_f + 2\omega_c \ddot{x}_f + 2\omega_c^2 \dot{x}_f + \omega_c^3 = \omega_c^3 x$$

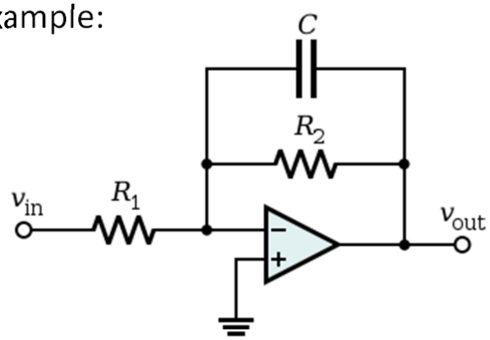
- cut frequency independent from the order N of the filter;
- attenuation magnitude in the stop band depends by the order N of the filter
- no oscillations in the passband or in the stop-band;
- no general rule about the phase.

How to practical realize an analog filter?

- RLC filters
- active RC filters (operational amplifier + passive feedback)
- ...

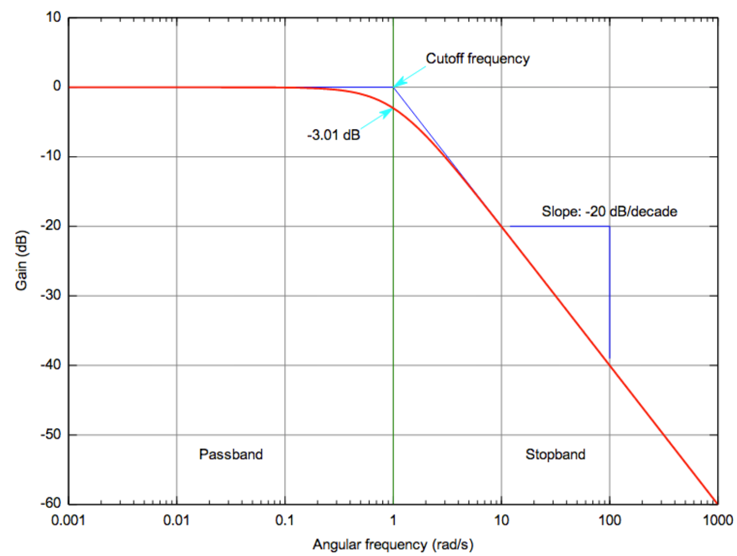


Example:



$$v_{out} = \frac{1}{1+s\cdot\tau} v_{in}$$

$$\tau = R_2 C \qquad f_c = \frac{1}{2\pi R_2 C}$$



18

High pass filters

High pass filters design can be seen as "complementary" as the low pass filter design.

They may be designed by substituting

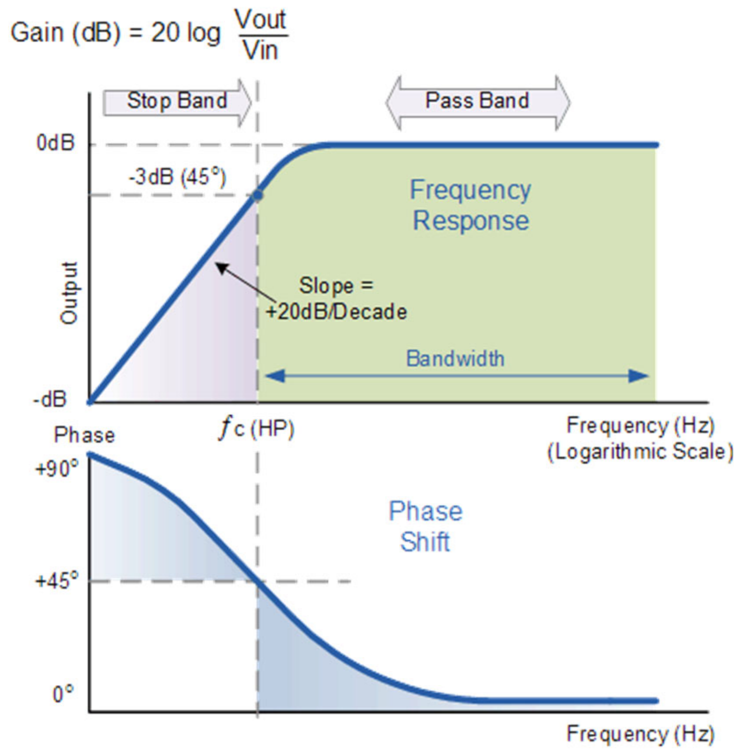
$$s \rightarrow \frac{1}{s}, \qquad \omega_c \rightarrow T_c$$

In the equation describing low pass filter, e.g.

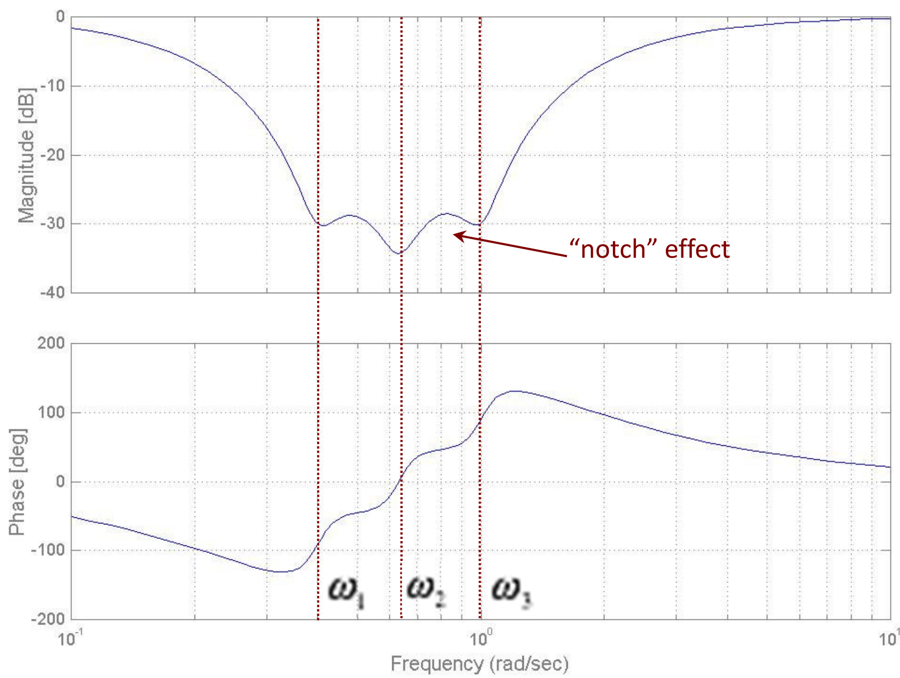
$$b(s) = \frac{\omega_c}{s+\omega_c} = \frac{1}{\frac{s}{\omega_c}+1}$$

The corresponding high pass filter becomes

$$b^{hp}(s) = \frac{T_c}{\frac{1}{s}+T_c} = \frac{sT_c}{1+sT_c}$$



Band-stop Filters



Example:
Third order filter

$$h_c = \prod_{i=1}^{r} \frac{s^2 + 2\varsigma_{ni}\omega_i s + \omega_i^2}{s^2 + 2\varsigma_{di}\omega_i s + \omega_i^2}$$

$\varsigma_{di} = 1$

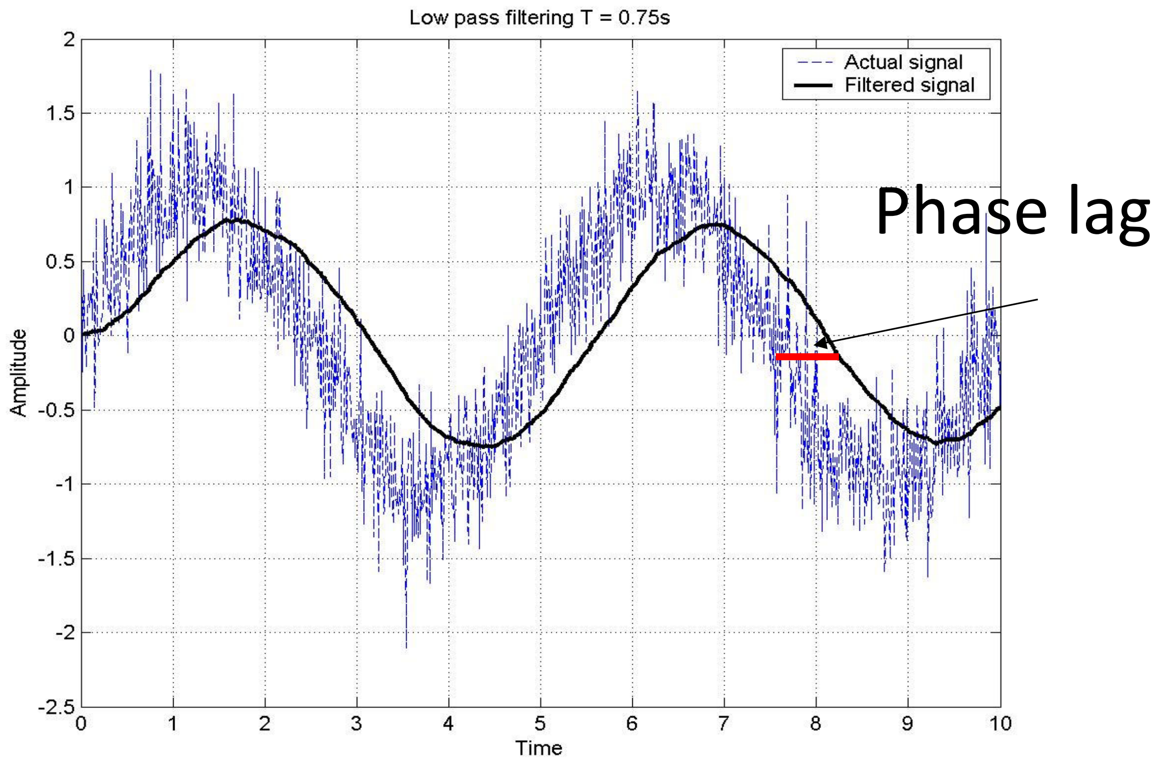
$\varsigma_{di} = 0.1$

$\omega_1 = 0.4$

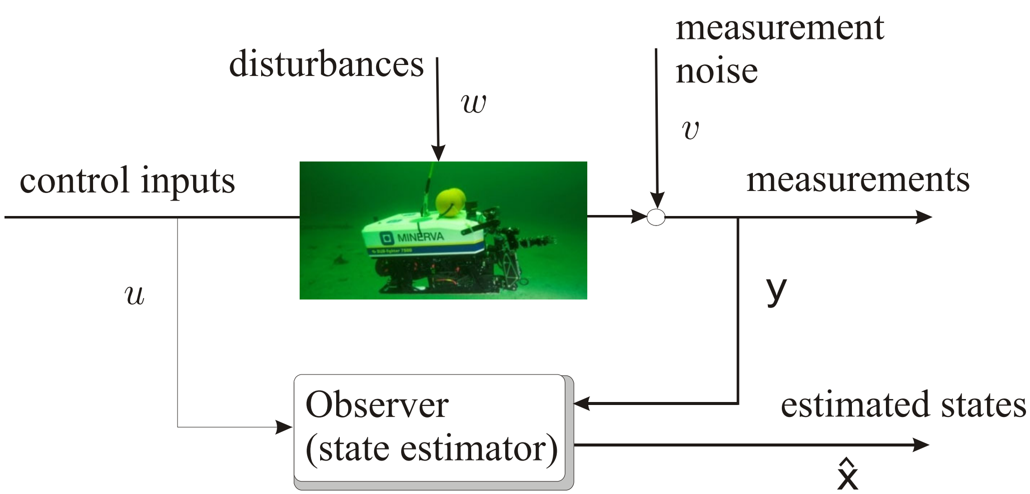
$\omega_2 = 0.63$

$\omega_3 = 1.0$

Low pass filter



Observer: filtering and state estimation

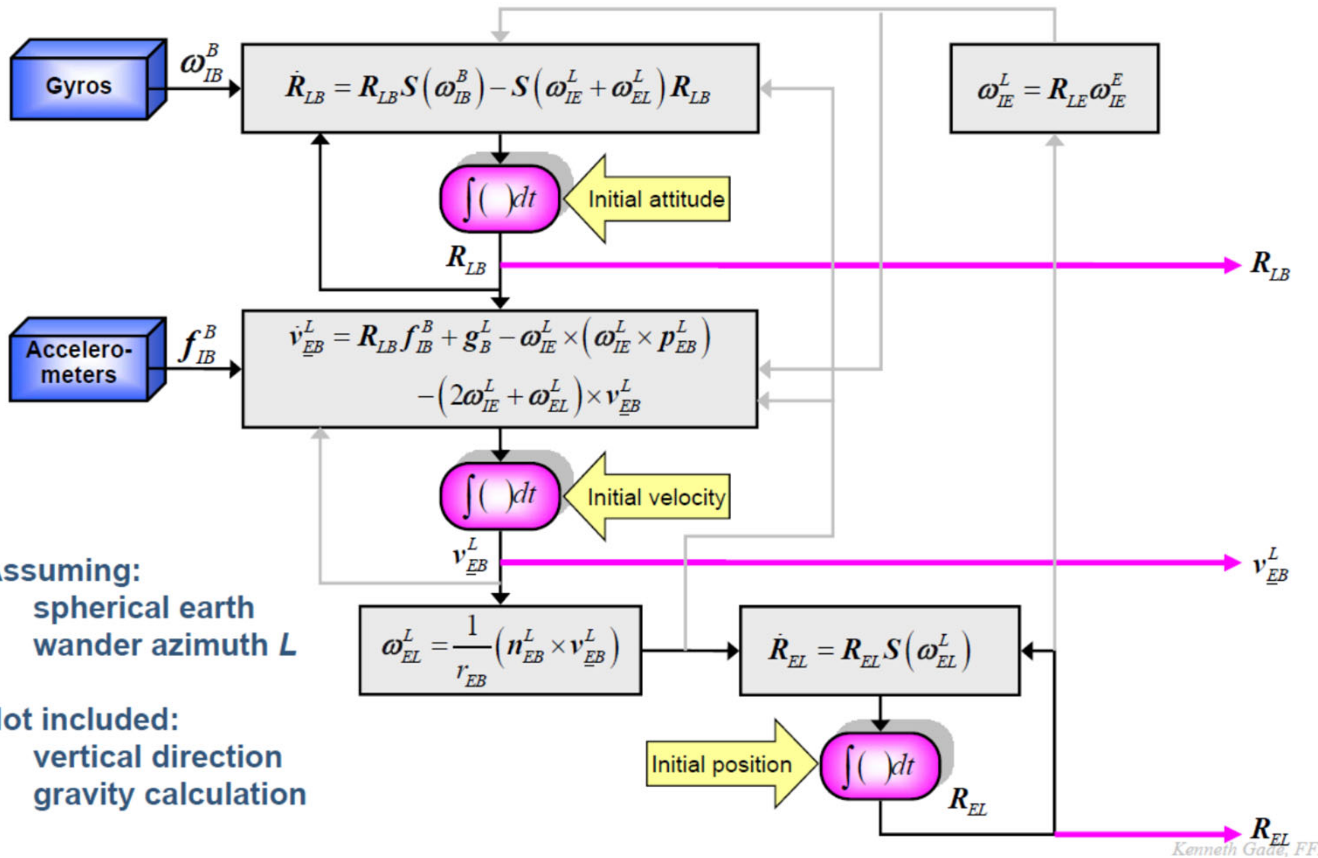


Definition

An observer or state estimator produces the state of a system from measurements of inputs and outputs

Observers reduce the phase lag and also can be used to estimate variables that are not measured.

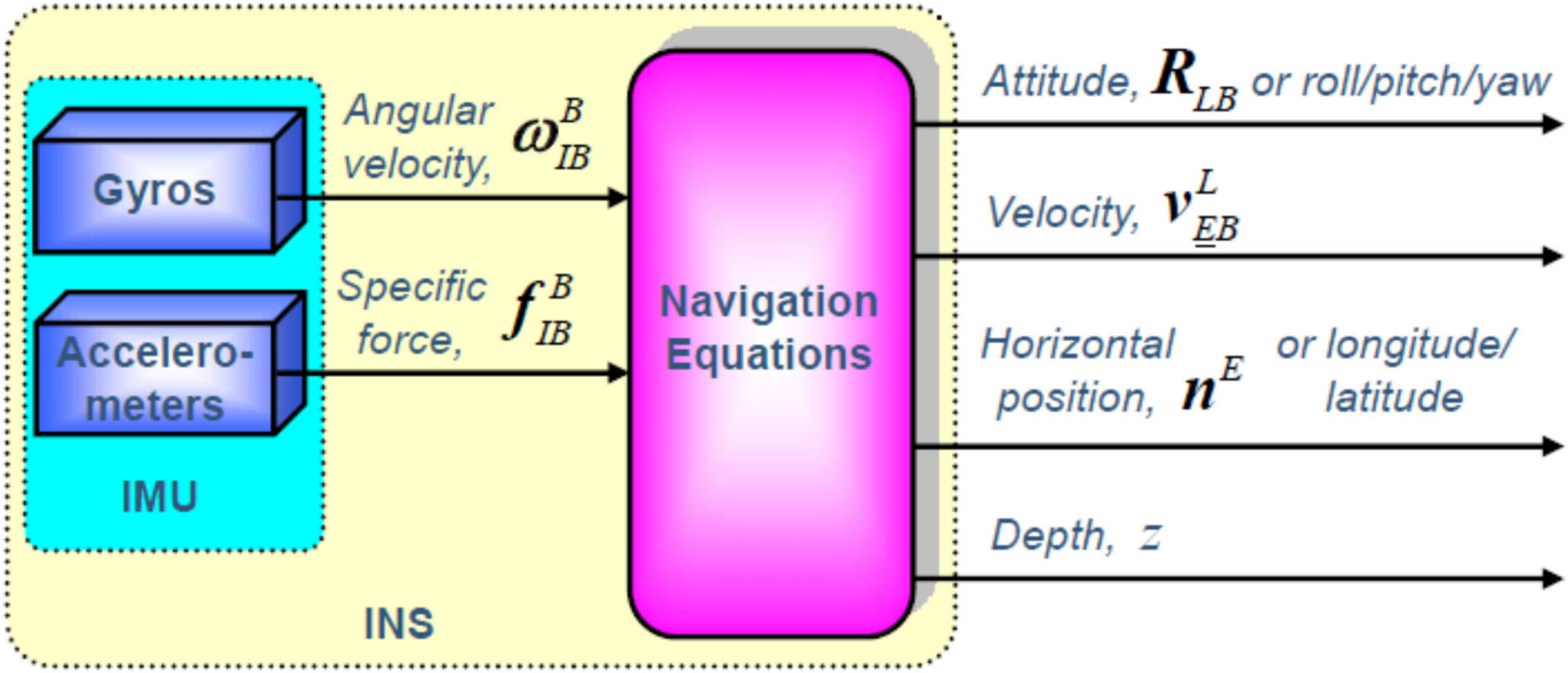
Inertial navigation



Assuming:
- spherical earth
- wander azimuth *L*

Not included:
- vertical direction
- gravity calculation

Inertial navigation



Kalman filter

- A Kalman filter is a recursive algorithm for estimating *states*in a system.
- Examples of states:
  - Position, velocity etc for a vehicle
  - pH-value, temperature etc for a chemical process

- Two sorts of information are utilized:
  - ***Measurements***from relevant sensors
  - ***Mathematical model*** of the system (describing how the different states depend on each other, and how the measurements depend on the states)

- In addition the *accuracy* of the measurements and the model must be specified.

Kalman filter algorithm

1. At t_0 the Kalman filter is provided with an *initial estimate*, including its uncertainty (covariance matrix).
2. Based on the mathematical modeland the initial estimate, a new estimate valid at t_1 is *predicted*. The uncertainty of the *predicted estimate*is calculated based on the initial uncertainty, and the accuracy of the model (*process noise*).
3. Measurementsvalid at t_1 give new information about the states. Based on the accuracy of the measurements (*measurement noise*) and the uncertainty in the predicted estimate, the two sources of information are weighed and a new *updated estimate* valid at t_1 is calculated. The uncertainty of this estimate is also calculated.
4. At t_2 a new estimate is predicted as in step 2, but now based on the updated estimate from t_1.

The prediction and the following update are repeated each time a new measurement arrives.

If the models/assumptions are correct, the Kalman filter will deliver optimal estimates.

Kalman filter equations

State space model:

$$\boldsymbol{x}_k = \boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{v}_{k-1}, \quad \boldsymbol{v}_k \sim N(\boldsymbol{0}, \boldsymbol{V}_k)$$
$$\boldsymbol{y}_k = \boldsymbol{D}_k\boldsymbol{x}_k + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim N(\boldsymbol{0}, \boldsymbol{W}_k)$$

Initial estimate (k = 0):

$$\hat{\boldsymbol{x}}_0 = E(\boldsymbol{x}_0), \quad \hat{\boldsymbol{P}}_0 = E\left((\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T\right)$$

State and covariance prediction:

$$\overline{\boldsymbol{x}}_k = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{x}}_{k-1}$$
$$\overline{\boldsymbol{P}}_k = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{V}_{k-1}$$

Measurement update (using y_k):

$$\hat{\boldsymbol{x}}_k = \overline{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{D}_k\overline{\boldsymbol{x}}_k)$$
$$\hat{\boldsymbol{P}}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{D}_k)\overline{\boldsymbol{P}}_k$$

Kalman gain matrix:

$$\boldsymbol{K}_k = \overline{\boldsymbol{P}}_k\boldsymbol{D}_k^T\left(\boldsymbol{D}_k\overline{\boldsymbol{P}}_k\boldsymbol{D}_k^T + \boldsymbol{W}_k\right)^{-1}$$

Kalman Filter Design for Navigation

- **Objective:** Find the vehicle position, attitude and velocity with the best accuracy possible

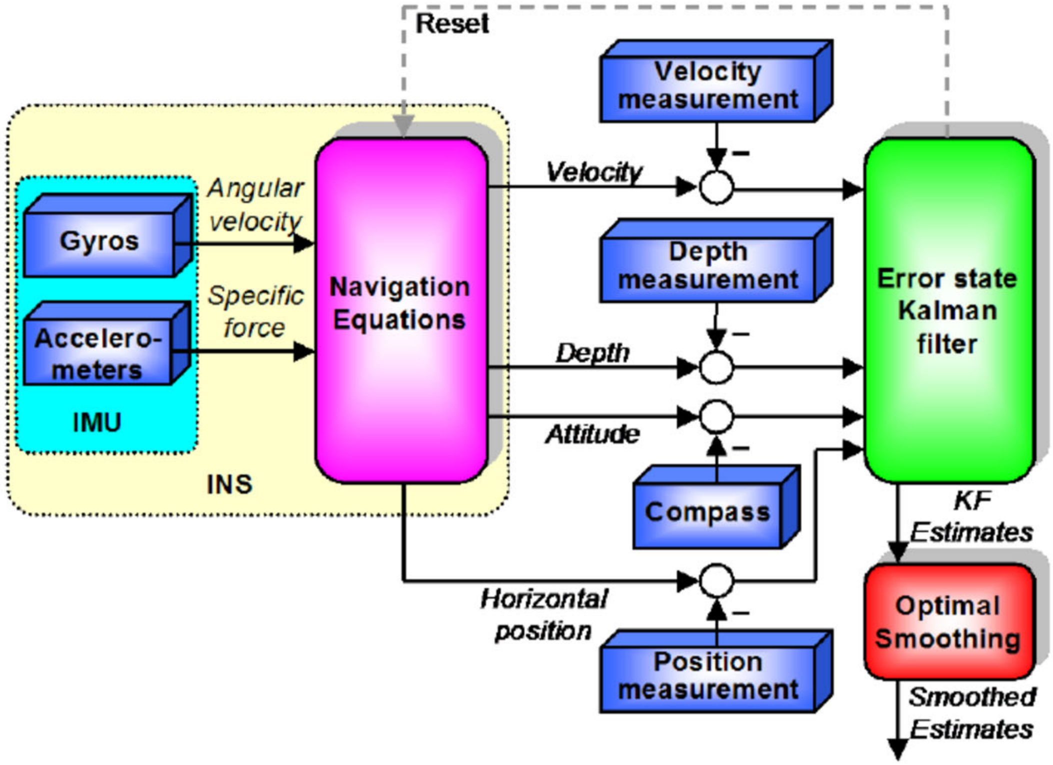
- **Possible basis:**
  - Sensor measurements (measurements)
  - System knowledge (mathematical model)
  - Control variables (measurements)

- We utilize sensor measurements and knowledge of their behavior (error models).
- This information is combined by means of an error-state Kalman filter.

Measurements

Sensor	Measurement	Symbol
IMU	Angular velocity, specific force	$\boldsymbol{\omega}_{IB}^{B}$, $\boldsymbol{f}_{IB}^{B}$
DGPS/USBL	Horizontal position measurement	$\boldsymbol{p}_{EB}^{E}$
Pressure sensor	Depth	
DVL	AUV velocity (relative the seabed) projected into the body (B) coordinate system	$\boldsymbol{v}_{EB}^{B}$
Compass	Heading (relative north)	ψ_{north}

Aided Inertial Navigation System



Kalman filter equations

State space model:

$$\boldsymbol{x}_k = \boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{v}_{k-1}\,, \quad \boldsymbol{v}_k \sim N(\boldsymbol{0}, \boldsymbol{V}_k)$$

$$\boldsymbol{y}_k = \boldsymbol{D}_k\boldsymbol{x}_k + \boldsymbol{w}_k\,, \quad \boldsymbol{w}_k \sim N(\boldsymbol{0}, \boldsymbol{W}_k)$$

Initial estimate (k = 0):

$$\hat{\boldsymbol{x}}_0 = E(\boldsymbol{x}_0)\,, \quad \hat{\boldsymbol{P}}_0 = E\left((\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T\right)$$

State and covariance prediction:

$$\overline{\boldsymbol{x}}_k = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{x}}_{k-1}$$

$$\overline{\boldsymbol{P}}_k = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{V}_{k-1}$$

Measurement update (using y_k):

$$\hat{\boldsymbol{x}}_k = \overline{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{D}_k\overline{\boldsymbol{x}}_k)$$

$$\hat{\boldsymbol{P}}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{D}_k)\overline{\boldsymbol{P}}_k$$

Kalman gain matrix:

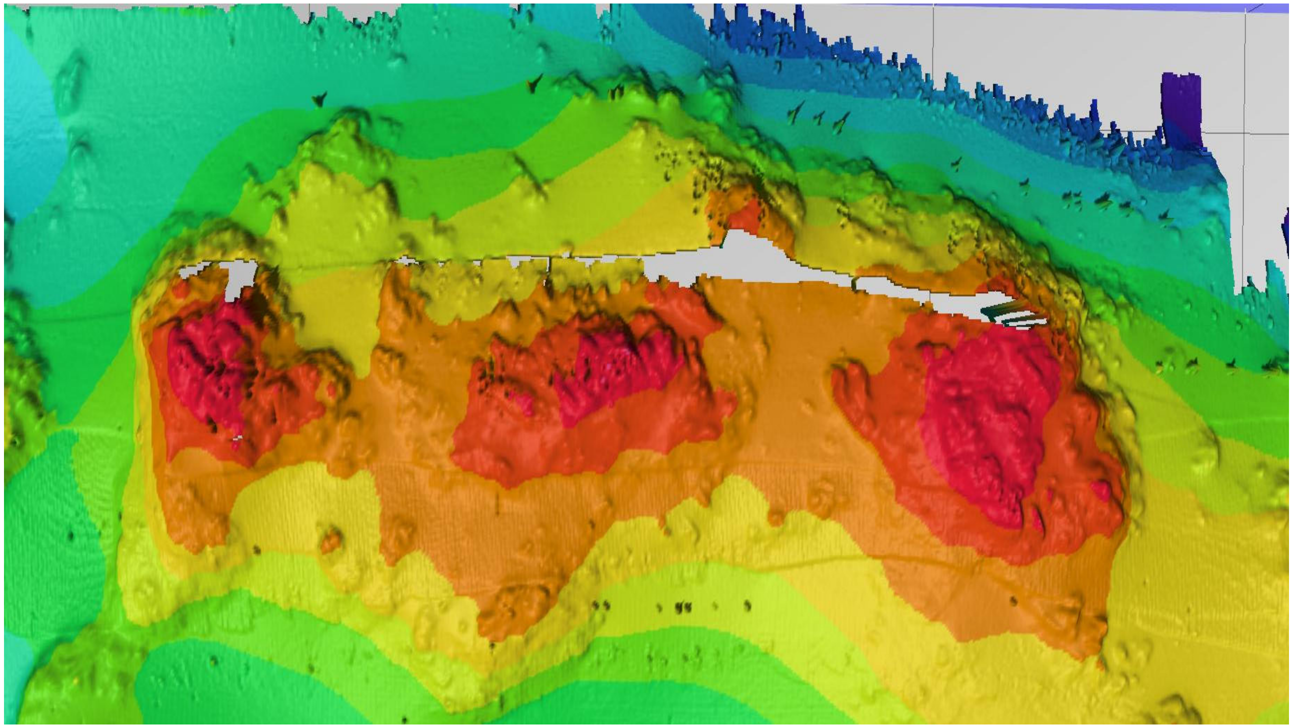
$$\boldsymbol{K}_k = \overline{\boldsymbol{P}}_k\boldsymbol{D}_k^T\left(\boldsymbol{D}_k\overline{\boldsymbol{P}}_k\boldsymbol{D}_k^T + \boldsymbol{W}_k\right)^{-1}$$

Accuracy in seabed mapping

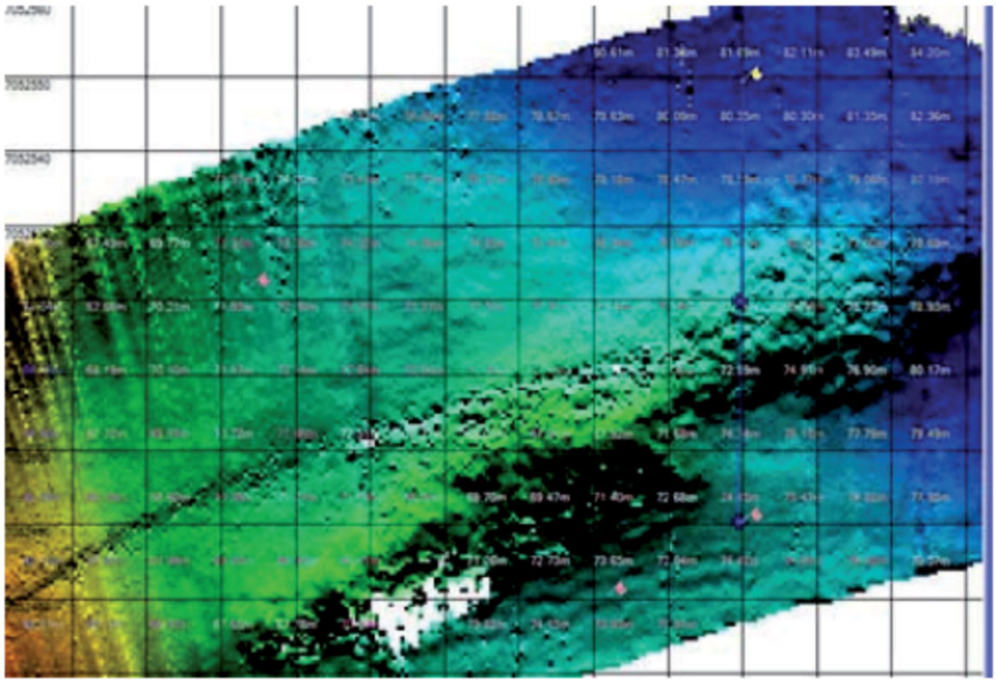
- Errors
  - Stochastics
  - Systematic errors
  - Blunders

- Instrument accuracy
- Navigation accuracy
- Set up integrity
  - Offset
  - Time
  - Angular
  - Level arms
- Environmental parameters

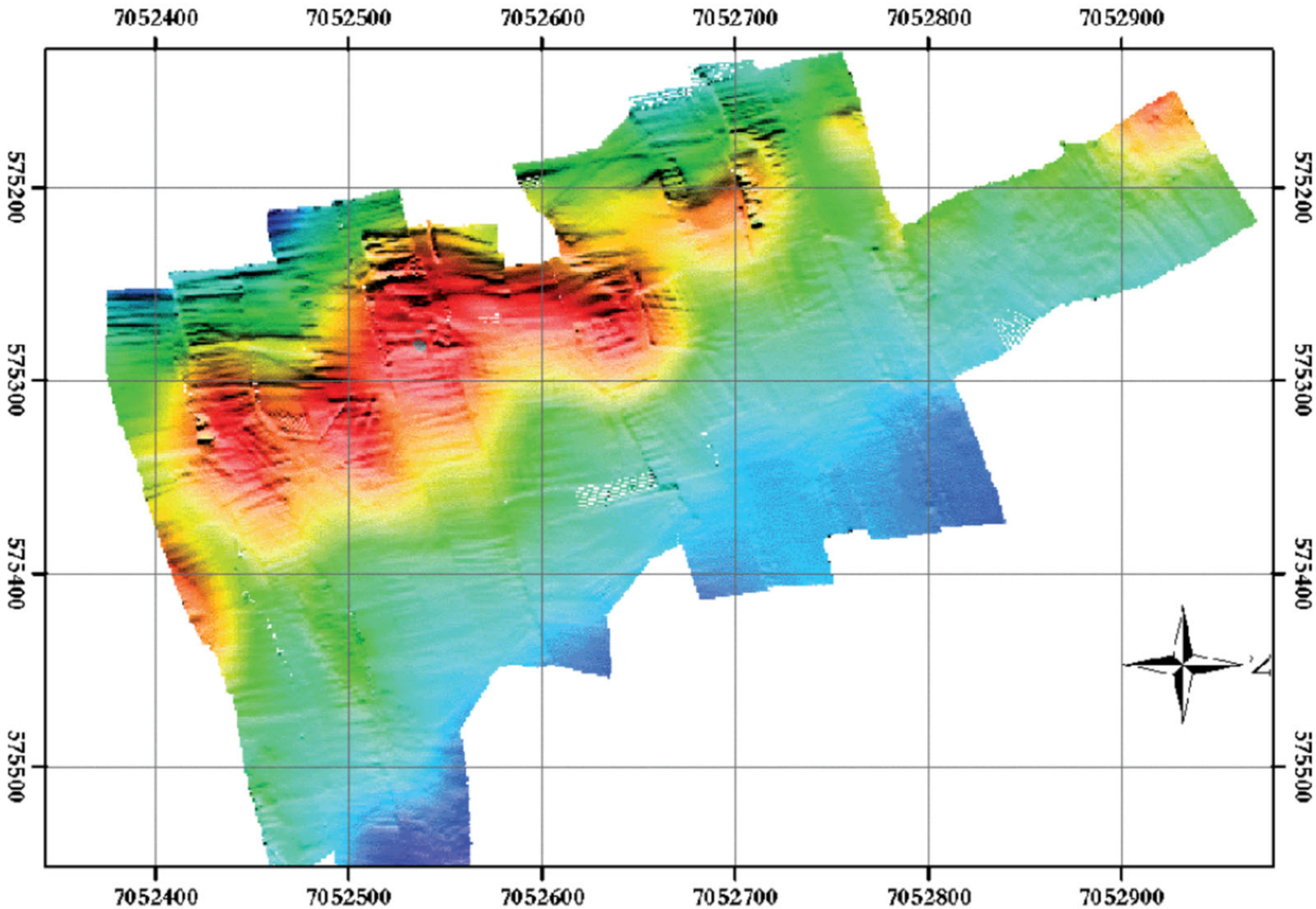
Results – EM 3002



MBE

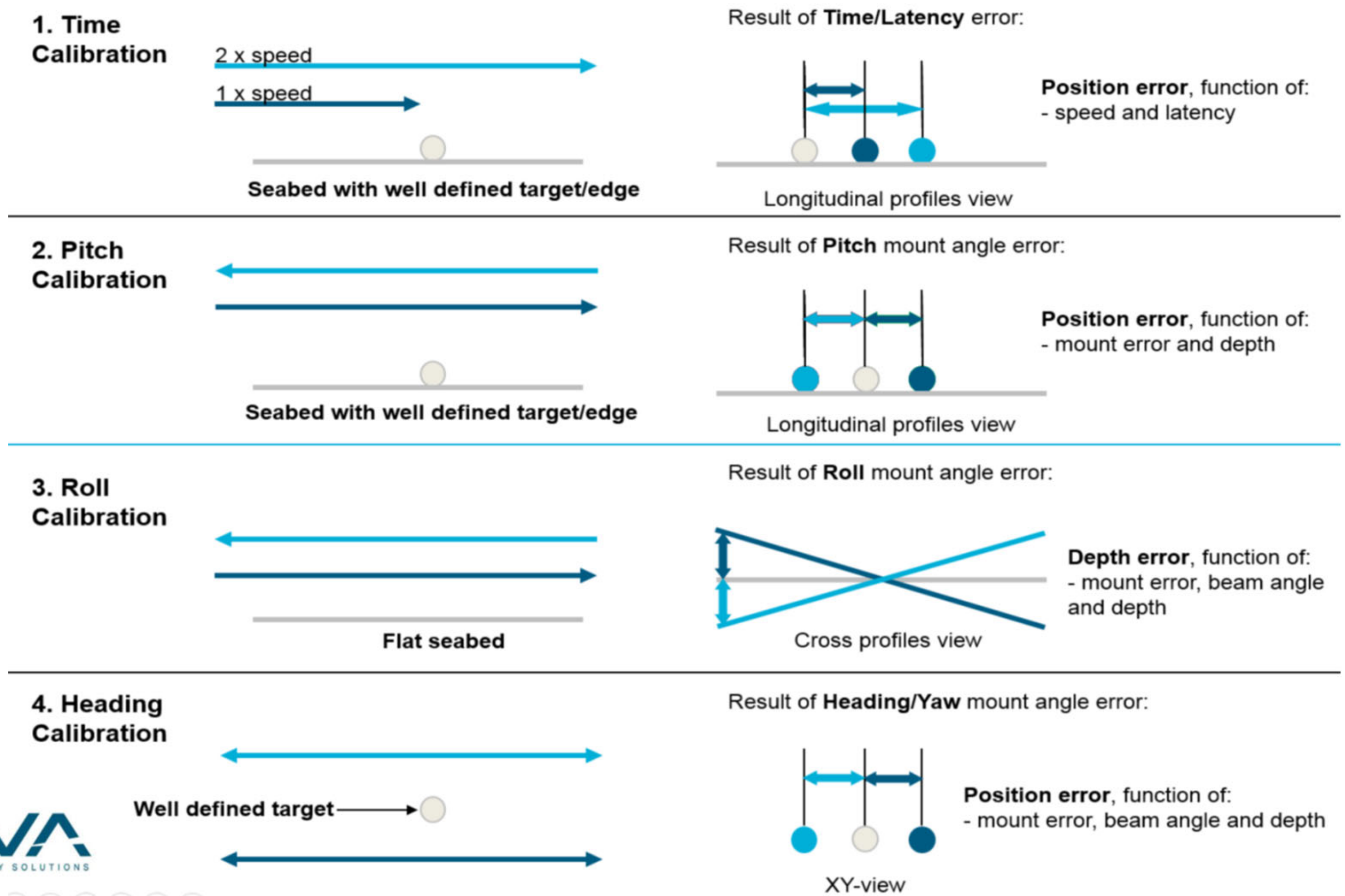


MBE

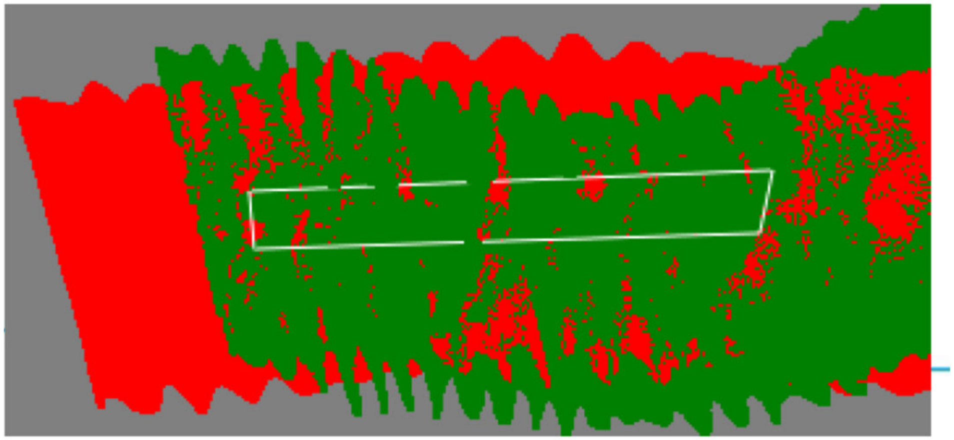


7052400
7052500
7052600
7052700
7052800
7052900
575200
575300
575400
575500
N

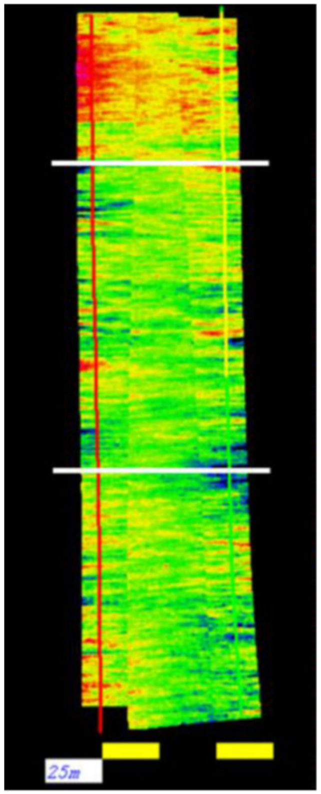
Patch test

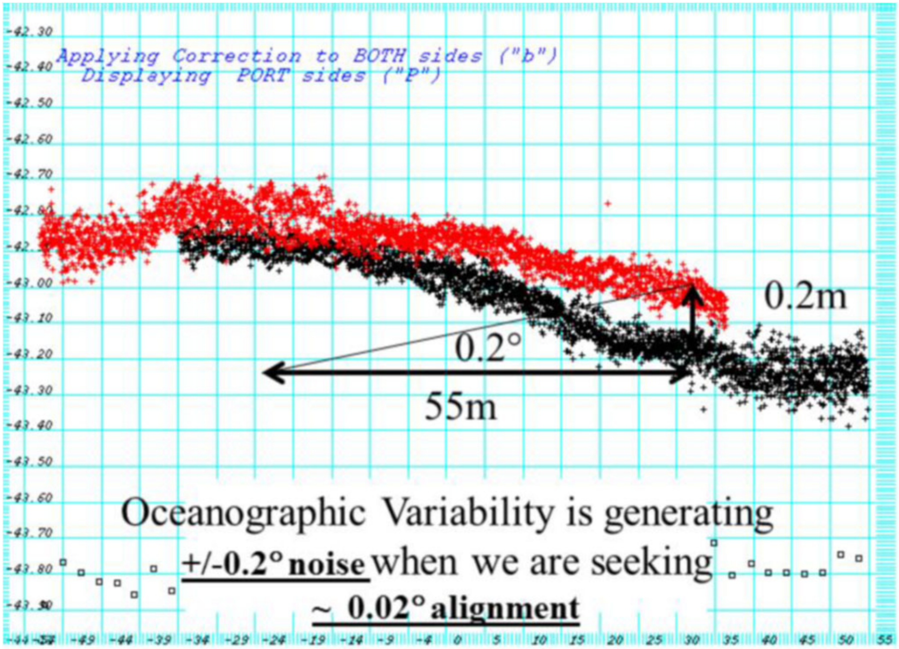

1. Time Calibration
2 x speed
1 x speed
Seabed with well defined target/edge
Result of Time/Latency error:
Position error, function of:
- speed and latency
Longitudinal profiles view
2. Pitch Calibration
Seabed with well defined target/edge
Result of Pitch mount angle error:
Position error, function of:
- mount error and depth
Longitudinal profiles view
3. Roll Calibration
Flat seabed
Result of Roll mount angle error:
Depth error, function of:
- mount error, beam angle and depth
Cross profiles view
4. Heading Calibration
Well defined target
Result of Heading/Yaw mount angle error:
Position error, function of:
- mount error, beam angle and depth
XY-view

Patch test



MBE





© J.E. Hughes Clarke

Error propagation

$$x = f(a, b, c)$$

$$dx_i = f(da_i, db_i, dc_i)$$

$$dx = \left(\frac{\delta x}{\delta a}\right)_{b,c} da,\ \left(\frac{\delta x}{\delta b}\right)_{a,c} db,\ \left(\frac{\delta x}{\delta c}\right)_{a,b} dc$$

$$\sigma_x^2 = \left(\frac{\delta x}{\delta a}\right)^2 \sigma_a^2 + \left(\frac{\delta x}{\delta b}\right)^2 \sigma_b^2 + \left(\frac{\delta x}{\delta c}\right)^2 \sigma_c^2$$

$$x = a + b - c \qquad \sigma_x = \sqrt{{\sigma_a}^2 + {\sigma_b}^2 + {\sigma_c}^2}$$

Assuming cross terms cancel out

LAW OF PROPAGATION OF VARIANCES

$$A = b + 2bc + 3d$$

A = the unknown quantity

b, c, and d = the known quantities

$$\sigma_A^2 = \left(\frac{\partial A}{\partial b}\right)^2 \sigma_b^2 + \left(\frac{\partial A}{\partial c}\right)^2 \sigma_c^2 + \left(\frac{\partial A}{\partial d}\right)^2 \sigma_d^2 + 2\left(\left(\frac{\partial A}{\partial b}\right)\left(\frac{\partial A}{\partial c}\right)\sigma_{bc} + \left(\frac{\partial A}{\partial b}\right)\left(\frac{\partial A}{\partial d}\right)\sigma_{bd} + \left(\frac{\partial A}{\partial c}\right)\left(\frac{\partial A}{\partial d}\right)\sigma_{cd}\right)$$

$$\left(\frac{\partial A}{\partial b}\right) = 1 + 2c \qquad \left(\frac{\partial A}{\partial c}\right) = 2b \qquad \left(\frac{\partial A}{\partial d}\right) = 3$$

$$\sigma_A^2 = (1+2c)^2 \sigma_b^2 + (2b)^2 \sigma_c^2 + (3)^2 \sigma_d^2$$

$$\sigma_A^2 = \left(1 + 4c + 4c^2\right)\sigma_b^2 + 4b^2\sigma_c^2 + 9\sigma_d^2$$

Error propagation for a USBL set up

USBL positioning

- P_s – ship position
- ω – bearing angle
- ν – depression angle

$$P_{USBL} = P_S + \begin{bmatrix} c_\omega & -s_\omega & 0 \\ s_\omega & c_\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\nu & 0 & s_\nu \\ 0 & 1 & 0 \\ -s_\nu & 0 & c_\nu \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} = P_S + \begin{bmatrix} c_\omega s_\nu \\ s_\omega s_\nu \\ c_\nu \end{bmatrix} r$$

$$P_{USBL} = P_S + R_{z,\omega} R_{y,\nu} \left[0, 0, r\right]^T$$

$$\sigma^2_{USBL,X} \approx \sigma^2_{S,X} + \left(\frac{\delta X}{\delta \omega}\right)^2 \sigma^2_\omega + \left(\frac{\delta X}{d\nu}\right)^2 \sigma^2_\nu + \left(\frac{\delta X}{\delta r}\right)^2 \sigma^2_r$$

$$\sigma^2_{USBL,X} \approx \sigma^2_{S,X} + \left(\frac{\delta c_\omega s_\nu r}{\delta \omega}\right)^2 \sigma^2_\omega + \left(\frac{\delta c_\omega s_\nu r}{\delta \nu}\right)^2 \sigma^2_\nu + \left(\frac{\delta c_\omega s_\nu r}{\delta r}\right)^2 \sigma^2_r$$

$$\sigma^2_{USBL,X} \approx \sigma^2_{S,X} + (-s_\omega s_\nu r)^2 \sigma^2_\omega + (c_\omega c_\nu r)^2 \sigma^2_\nu + (c_\omega s_\nu)^2 \sigma^2_r$$

Example

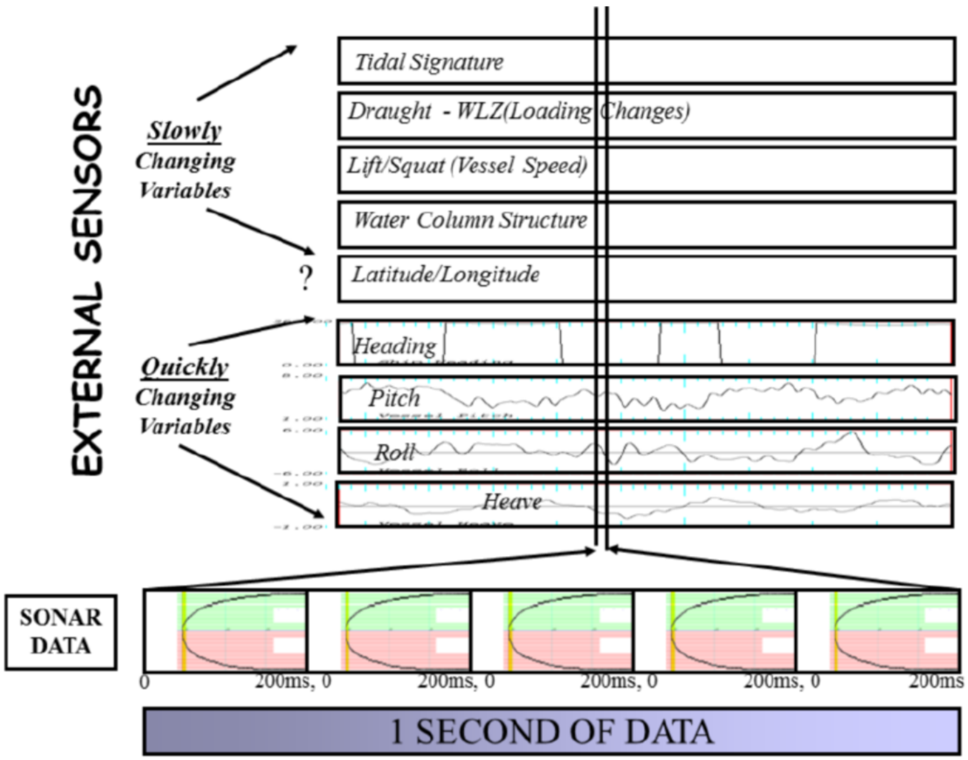
Measurement	Instrument	Example	Stddev
Range	USBL HiPaP	200	0.1 [m]
Depression angle	USBL HiPaP	70	0.12 [deg]
Bearing angle	USBL HiPaP	40	0.12 [deg]
Position	GPS - Seatex DPS 116	0	0.5 [m]

$$P_{USBL} = P_S + R_{z,\omega} R_{y,\nu} \left[0, 0, r\right]^T$$

$$\sigma^2_{USBL,X} \approx 0.5^2 + (-\sin(40)\cdot\sin(70)\cdot 200)^2 (0.12\cdot\frac{\pi}{180})^2 + (\cos(40)\cdot\cos(70)\cdot 200)^2 (0.12\cdot\frac{\pi}{180})^2 + (\cos(40)\cdot\sin(70))^2 (0.1)^2$$

$$\sigma_{USBL,X} \approx \sqrt{\sigma^2_{USBL,X}} = 0.5755\,\mathrm{m}$$

Timing and latency



NTNU

Learning objectives

- Data processing
- Error propagation
- Error budgets
- Timing and latency
- Calibrations

More reading

Processing

- Anonsen 2013
- Kebkal 2017

Error propagation

- Jalving 1999
- Gade 2005