

Chapter 13 – Model-Based Navigation Systems

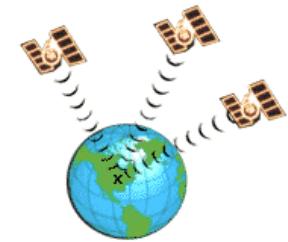
13.1 Sensors for Marine Craft

13.2 Wave Filtering

13.3 Fixed-Gain Observer Design

13.4 Kalman Filter Design

13.5 Passive Observer Design



Conventional ship and underwater vehicle control systems are implemented with a model-based state estimator for processing of the sensor and navigation data.

The quality of the raw measurements (GNSS, accelerometers, gyros, compass etc.) are usually monitored and handled by a signal processing unit or a program for quality check and wild-point removal. The processed measurements are transmitted to the sensor and navigation computer which uses a state estimator capable of noise filtering, prediction and reconstruction of unmeasured states.



Sensor STIM300 IMU
<https://www.sensonor.com>

The most advanced navigation system for marine applications is the *Inertial Navigation System (INS)*; see Chapter 14.

Chapter Goals

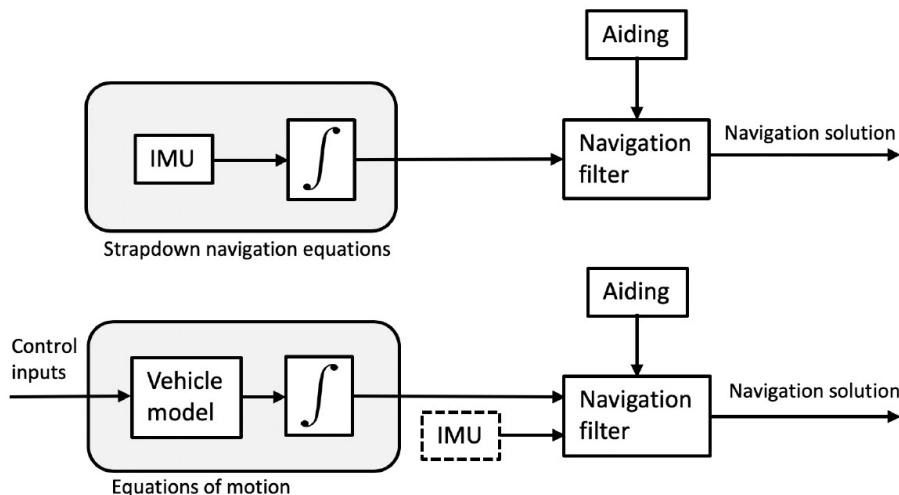
Sensors for marine craft:

- Understand the principles for GNSS position, GNSS heading, magnetic compass and gyrocompass
- Understand what we mean with **wave filtering** and when to apply a wave filter algorithm
- Be able to **estimate the wave encounter frequency** of a marine craft

Model-based state estimation:

- Understand the principles and design methods for fixed-gain Luenberger observers, Kalman filters and passive observers
- Be able to model marine craft under DP and heading control, and include **dynamic models of the sensor and navigation systems using realistic measurement noise**
- Be able to design **Kalman filters** for DP and heading autopilots with wave filtering capabilities
- Be able to design **passive observers** for DP and heading autopilots with wave filtering capabilities

Chapter 13 – Model-Based Navigation Systems



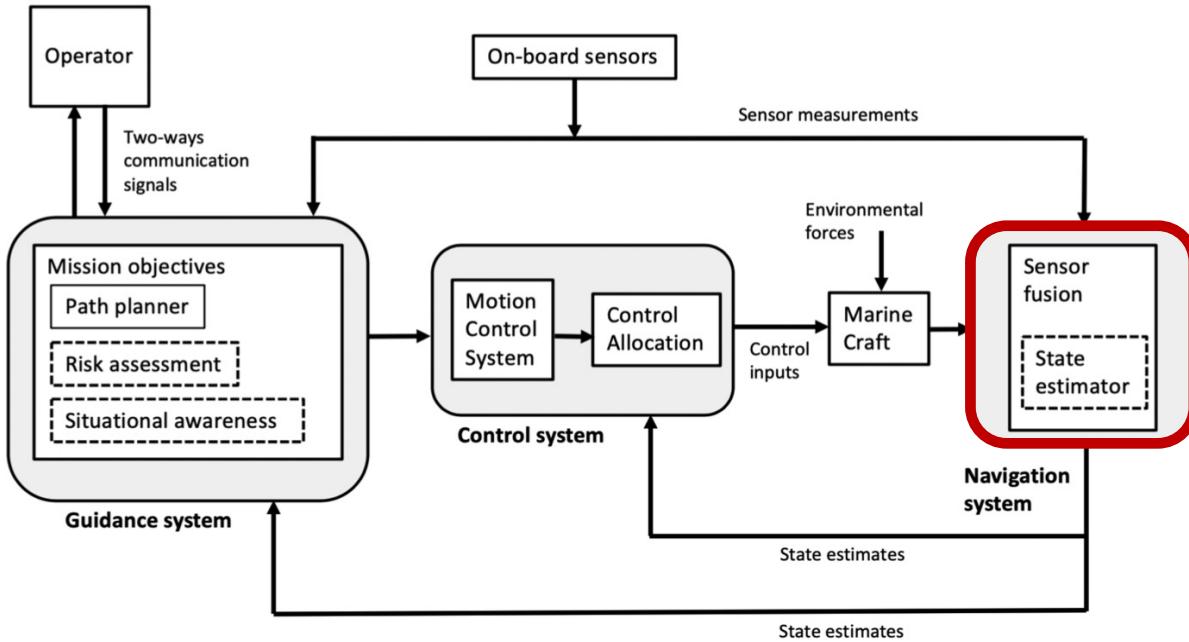
In a model-based KF, the craft position, velocity and attitude are states in the estimator, while linear acceleration and angular rates are generated using a mathematical model ([Chapter 13](#)).

Alternatively, the model can be avoided by using accelerometers and angular rate sensor (ARS) measurements as inputs and integrate the kinematic equations ([Chapter 14](#)) . This is an Inertial Navigation System (INS).

The **drawback** of the model-based approach to aided INS is **model uncertainty** when implemented in a KF. One obvious advantage is that additional sensors such as the inertial measurement unit (IMU) are avoided. Another benefit is that the mathematical model can be used for fault detection and isolation, as well as fault recovery.

The marine craft equations of motion when implemented in a KF is in fact a **predictor**, which can be used to predict future motions of the craft when sensors fails or have outages for shorter periods of time.

Chapter 13 – Model-Based Navigation Systems



The sensor and navigation system is usually implemented as an optimal state estimator (Kalman filter) using GNSS measurements combined with motion sensors such as accelerometers and attitude rate sensors (ARS).

Navigation is the science of directing a craft by determining its position/attitude, course and distance traveled. In some cases, velocity and acceleration are determined as well. Navigation is derived from the Latin *navis*, “ship”, and *agere*, “to drive”. It originally denoted the art of ship driving, including steering and setting the sails.

Guidance, Navigation and Control (GNC)

GNC is a branch of engineering dealing with the design of systems to control the movement of vehicles, especially, automobiles, drones, marine craft, aircraft, and spacecraft.

- **Guidance** focuses on the methods and systems used to follow a route or path to a destination once it has been identified. It's about steering, direction, and instruction on moving from one point to another.
 - **How do I get there?** is fundamentally a question of guidance.

Once you know your current location and your destination (navigation), guidance provides the step-by-step instructions or **actions required to reach the destination**.

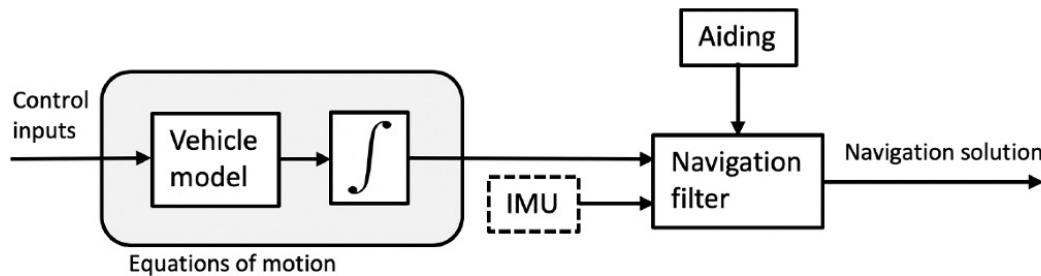
- **Navigation** refers to the process of **determining one's current position** and **planning a route** to a desired destination.
 - **Where am I?** identifies your current location. This is the primary navigation question.
 - **Where do I want to go?** decides your destination
 - **Which way should I go?** plans the route to get there, considering various factors such as distance, time, obstacles, and preferences
- **Control** refers to the manipulation of the forces, by way of steering controls, thrusters, etc., needed to execute guidance commands whilst maintaining vehicle stability.

13.1 Sensors for Marine Craft

The primary measurement systems for model-based navigation filters when used onboard a surface craft are global satellite navigation systems (GNSS) and heading angle sensors. More specific,

- GNSS position
- GNSS heading
- Magnetic compass
- Gyrocompass

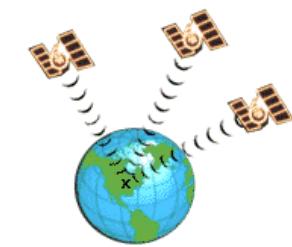
The position and heading angle measurements are used as **aiding** – to prevent drift when integrating the equations of motion. Sometimes an inertial measurement unit (IMU) is included in the state estimator as optional measurements.



13.1.1 GNSS Position

GNSS position is the primary sensor for terrestrial navigation. The four commercial systems are:

- **NAVSTAR Global Positioning System (GPS):** The United States NAVSTAR GPS was started by the U. S. Department of Defense in 1973, with the first prototype spacecraft launched in 1978 and the full constellation of 24 satellites operational in 1993.
- **GLONASS:** From Russian *GLObal'naya NAVigatsionnaya Sputnikovaya Sistema*. The development of the Russian GLONASS satellite navigation system began in the Soviet Union in 1976 and the constellation was completed in 1995. After a decline in capacity in the 90s, GLONASS was restored. Full orbital constellation of 24 satellites was achieved in 2011, enabling full global coverage.
- **Galileo:** The European Union's Galileo positioning system went live in 2016. It is an independent civilian positioning system designed by European nations so they do not have to rely on GPS, GLONASS or BeiDou, which could be disabled or degraded by their operators at any time.
- **BeiDou:** Chinese for the Big Dipper or the North Star. In 2015, China launched the third generation BeiDou (BeiDou-3) for global navigation. BeiDou-3 consists of 35 satellites and the system has provided global services since 2020.



Differential and augmented GNSS: The main idea of a differential GNSS system is that a fixed receiver located, for example, on shore with a known position is used to calculate the GNSS position errors. The position errors are then transmitted to the GNSS receiver on board the ship and used as corrections to the actual ship position. In a differential GNSS the horizontal positioning errors are squeezed down to less than 1 m (typical accuracy of a ship positioning system today).

13.1.2 GNSS Heading

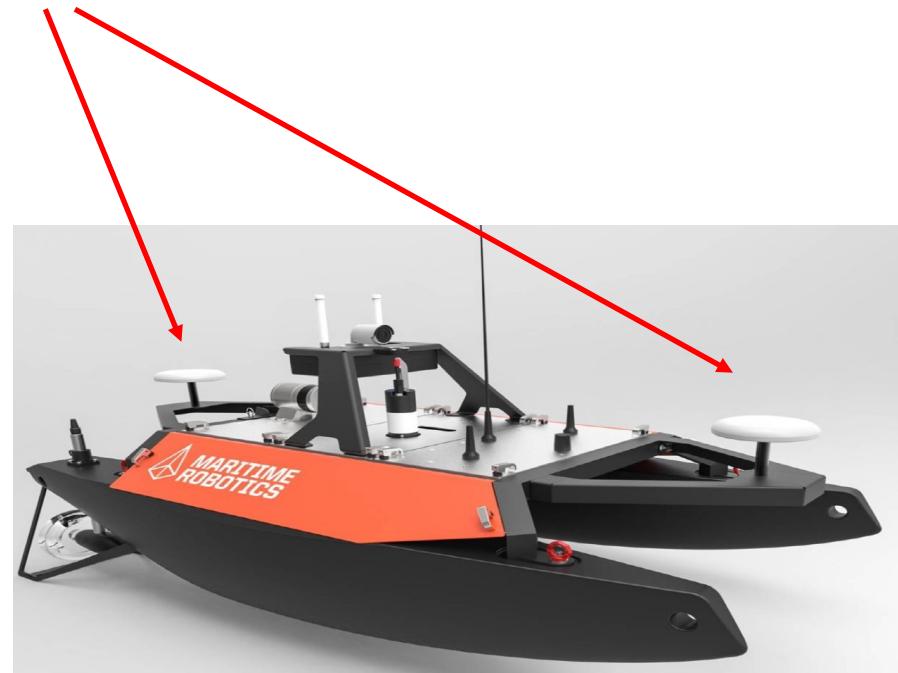
The GNSS system can be used to determine the heading angle, even though it was not designed for this purpose.

A “**GNSS compass**” uses a **pair of antennas separated by 50 cm or more** to detect the phase difference in the carrier signal from a particular GNSS satellite.

Given the positions of the satellite, the position of the antenna, and the phase difference, the orientation of the two antennas can be computed.

The accuracy can be further improved by using three antennas in a triangle to get three separate readings with respect to each satellite. It is also beneficial to increase the distance between the antennas.

The GNSS heading solution is not subject to magnetic declination, but it will be **sensitive to ionospheric disturbances and multipath effects**.



The Otter USV manufactured by www.maritimerobotics.com

13.1.3 Magnetic Compass

A **magnetic compass** is in fact an extremely simple device (as opposed to a gyroscopic compass). It consists of a small, lightweight magnet balanced on a nearly frictionless pivot point. The magnet is generally called a needle.

The magnetic field inside the Earth has its south end at the North Pole and opposite. Hence, the North end of the compass needle points towards the North Pole (opposite magnets attract). The magnetic field of the Earth is, however, not perfectly aligned along the Earth's rotational axis. It is skewed slightly off center. This skew or bias is called the declination and it must be compensated for.



Wikimedia Commons

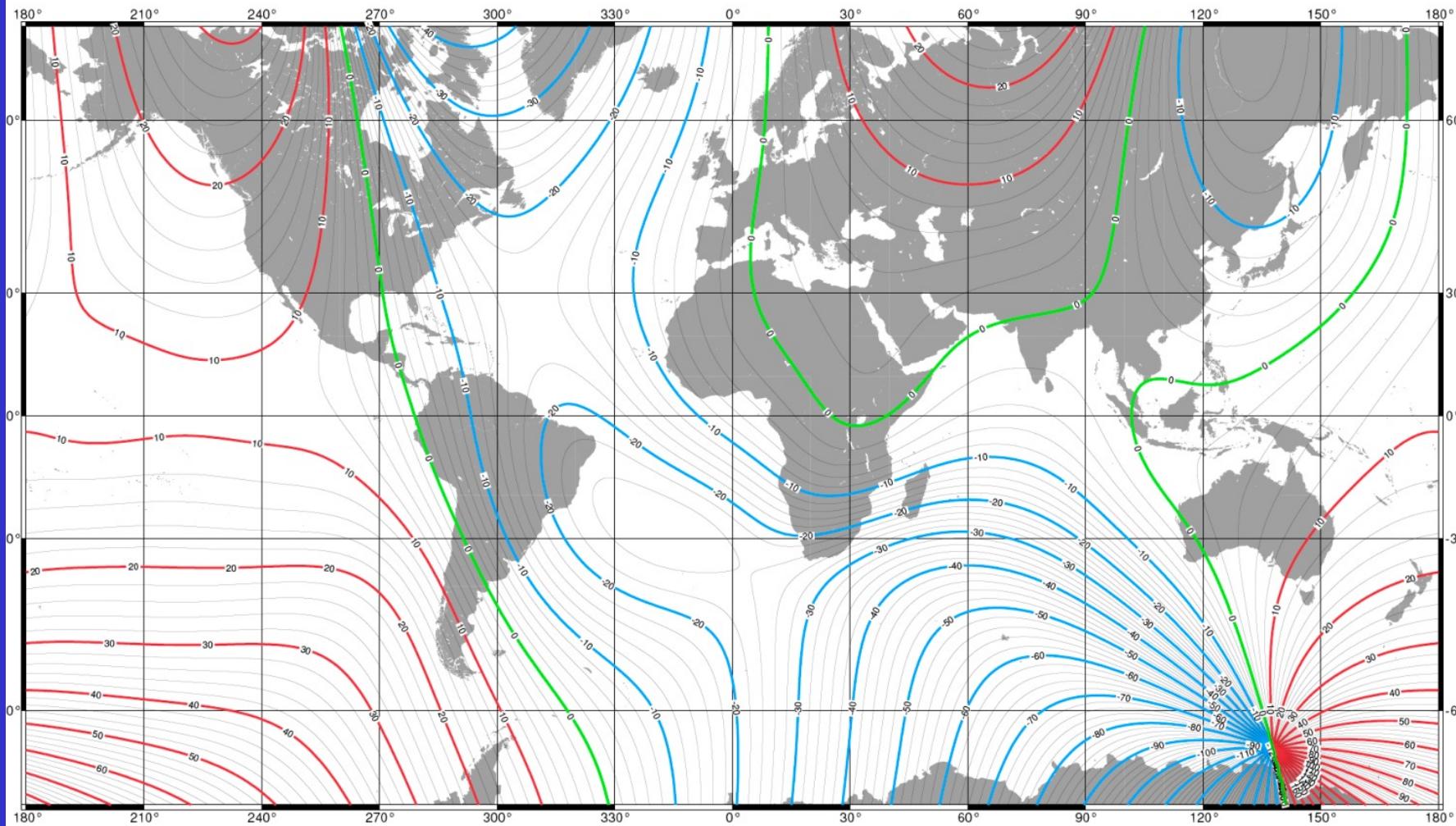
https://commons.wikimedia.org/wiki/File:Kompas_Sofia.jpg

The heading angle is the sum of the the **magnetic heading measurement** ψ_m and the **declination angle** δ given by

$$\psi = \psi_m + \delta$$

The declination angle for a given longitude λ and latitude μ can be calculated using the *World Magnetic Model* (WMM), which is a joint project by the United States' National Geospatial-Intelligence Agency (NGA) and the United Kingdom's Defence Geographic Centre (DGC); [see next slide](#). The WMM magnetic model comes with [C software and executables](#).

Sensitivity to magnetic variations and declination cause problems in ship navigation. These problems were overcome after the introduction of the gyroscopic compass.

Magnetic field declination δ according to the US/UK World Magnetic Model (WMM)

13.1.4 Gyrocompass

The large variations in the magnetic character of ships caused by electrical machinery and weapon systems make the construction of accurate declination or deviation tables very difficult.

In parallel works, Dr H. Anschütz-Kaempfe of Germany and Elmer Sperry of the USA worked on a practical application of Hopkins' gyroscope. In 1908 Anschütz patented the first North-seeking gyrocompass, while Elmer Sperry was granted a patent for his ballistic compass, which includes vertical damping, three years later.

Today gyroscopic compasses are widely used for navigation, because they have significant advantages over magnetic compasses. In particular they are **unaffected by ferromagnetic materials**, such as in a ship's steel hull, which distort the magnetic field. Another important aspect is that they are **not affected by electromagnetic fields**, which are generated by rotating machinery and engines moving electric charges.

Unfortunately, a gyrocompass is quite expensive, which limits their use to large ships and safety-critical vehicle systems. Smaller vehicles usually navigate by using magnetic compasses, course over ground or GNSS heading.



Elmer A. Sperry, Sr. (1860-1930)



Hermann Anschütz-Kaempfe (1872-1931)

Wikipedia commons:

https://commons.wikimedia.org/wiki/File:Elmer_Ambrose_Sperry.jpg

https://commons.wikimedia.org/wiki/File:Hermann_Anschütz-Kaempfe.jpg



Kongsberg gyrocompass and INS MGC R3

Sperry MK37 VT Digital Gyrocompass

13.2 Wave Filtering

- *Wave filtering* is one of the most important issues to consider when designing ship control systems.
- It is important that only the slowly-varying disturbances are counteracted by the steering and propulsion systems; the oscillatory **wave-frequency (WF) motion** due to the waves (1st-order wave-induced forces) should be prevented from entering the feedback loop (Balchen 1976).

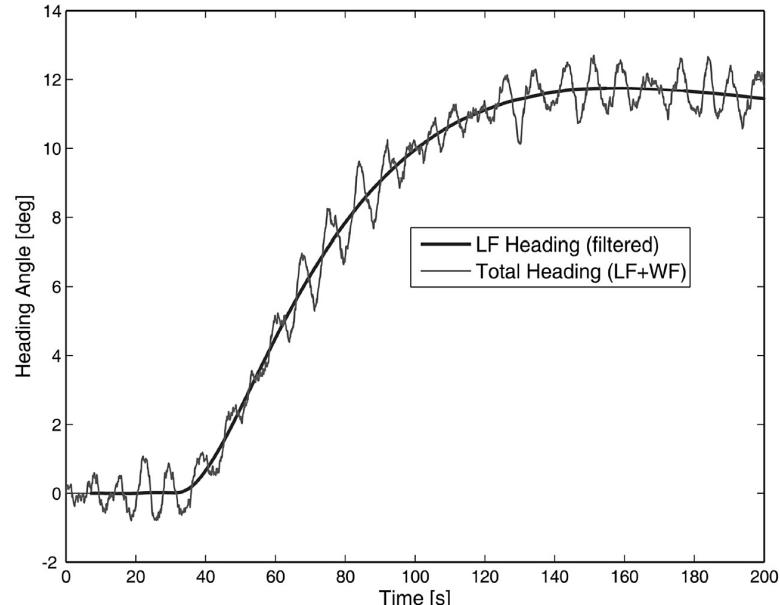


Definition: Wave Filtering

Wave filtering can be defined as the reconstruction of the **low-frequency (LF) motion** components from wave-induced noisy measurements of position, heading and in some cases velocity and acceleration by means of a state estimator or a filter.

Remark: If a state estimator such as the Kalman filter is applied, estimates of the WF motion components (first-order wave-induced forces) can also be computed.

$$\text{Total motion} = \text{LF motion} + \text{WF motion}$$



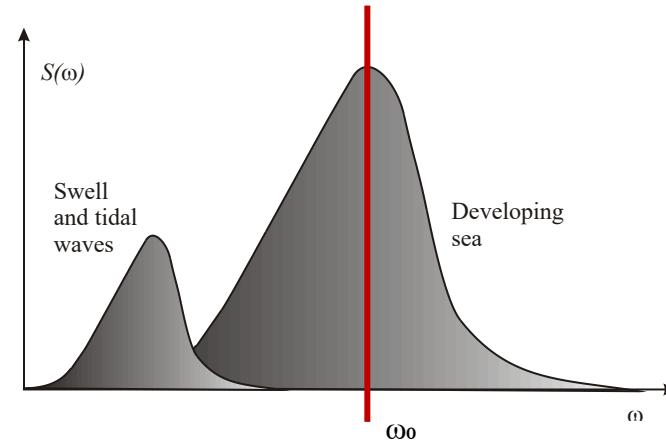
13.2.1 Low-Pass Filtering

For wave periods in the interval $5s < T_0 < 20s$, the dominating wave frequency (modal frequency) f_0 of a wave spectrum will be in the range

$$0.05 \text{ Hz} < f_0 < 0.2 \text{ Hz}$$

The circular frequency $\omega_0 = 2\pi f_0$ corresponding to periods $T_0 > 5s$ is

$$\omega_0 < 1.3 \text{ rad/s}$$



Waves can be accurately described by 1st- and 2nd-order linear wave theory:

- **1st-order wave-induced forces (WF forces)** produce large **oscillations** about a mean wave force.
WF forces are represented as a wave spectrum.
Compensated for by using **wave filtering in the state estimator**
- **2nd-order wave-induced forces** or mean wave (**drift**) forces are slowly varying forces.
Compensated for by using **integral action in the control law**

13.2.1 Low-Pass Filtering

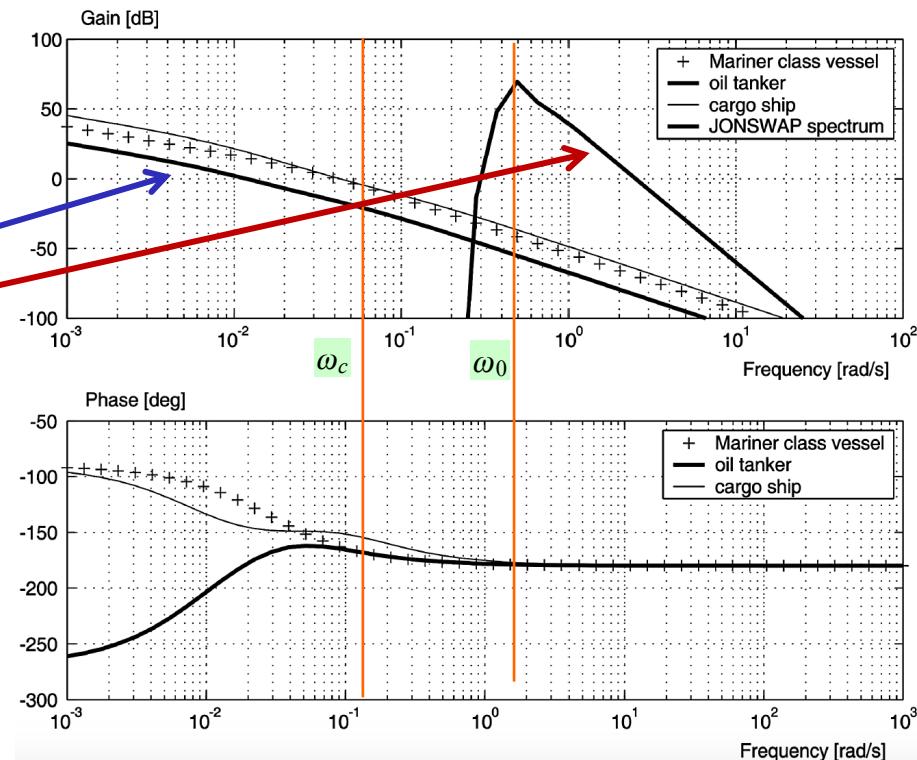
A feedback control system will typically move the bandwidth ω_b of the vessel up to 0.1 rad/s which still is below the wave spectrum.

The wave disturbances will typically be inside the bandwidth of the servos and actuators of the vessel. Hence, the wave disturbances must be filtered out before feedback is applied in order to avoid unnecessary control action.

LF vessel motion

WF motion

For a large oil tanker, the crossover frequency ω_c can be as low as a 0.01 rad/s, while smaller vessels like cargo ships and the *Mariner class vessel*, are close to 0.05 rad/s.

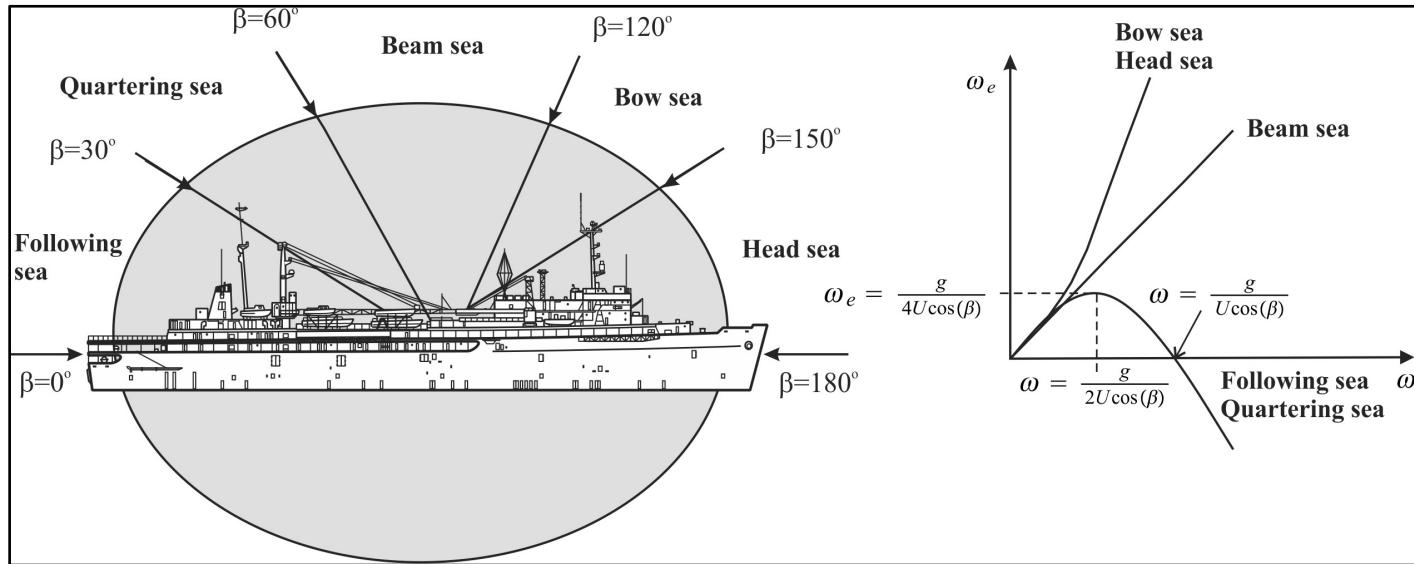


13.2.1 Low-Pass Filtering

For a ship moving at forward speed $U > 0$, there will be a shift in the wave spectrum peak frequency ω_0 .

The shifted frequency is referred to as the frequency of encounter ω_e and it depends on ship speed U , modal wave frequency ω_0 and wave direction β

$$\omega_e = \left| \omega_0 - \omega_0^2 \frac{U}{g} \cos(\beta) \right|$$



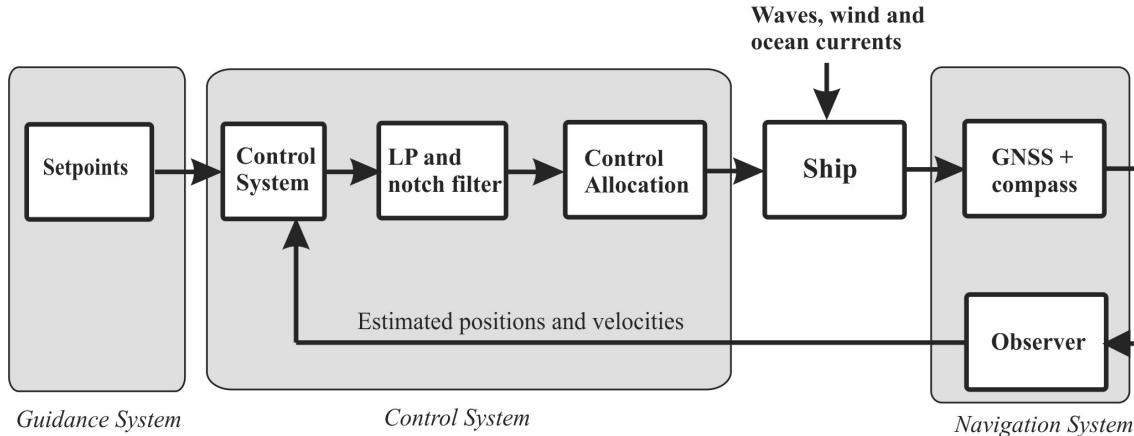
13.2.1 Low-Pass Filtering

For sea states where the encounter frequency ω_e is much higher than the bandwidth ω_b of the control system,

$$\omega_b \ll \omega_e$$

a LP-filter can be used to filter out the 1st-order wave-induced forces. This is typically the case for **large vessels such as oil tankers**.

For smaller vessels, a LP filter in cascade with a notch filter is quite common to use.



LP and notch filters in series with the control system

13.2.1 Low-Pass Filtering

Autopilot measurement equation

$$y(s) = \underbrace{h_{\text{ship}}(s)\delta(s)}_{\psi(s)} + \underbrace{h_{\text{wave}}(s)w(s)}_{\psi_w(s)}$$

where

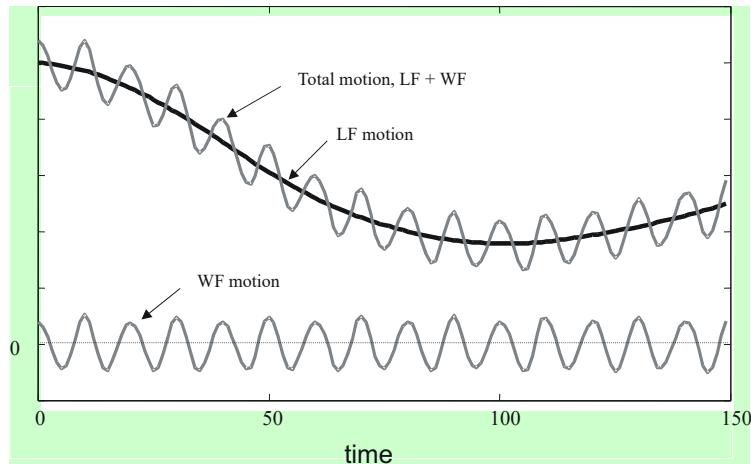
$y(s)$ is the compass measurement

$w(s)$ is zero-mean Gaussian white noise

$\delta(s)$ is the rudder input.

$\psi(s)$ is the LF motion

$\psi_w(s)$ is the WF motion



Linear theory:

Consequently, the feedback control law $\delta(s)$ should be a function of $\psi(s)$ and not $y(s)$ in order to avoid 1st-order wave-induced rudder motions.

$$h_{\text{wave}}(s) = \frac{K_w s}{s^2 + 2\lambda\omega_0 s + \omega_0^2}$$

$$h_{\text{ship}}(s) = \frac{K(T_3 s + 1)}{s(T_1 s + 1)(T_2 s + 1)}$$

13.2.1 Low-Pass Filtering

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

$$h_{lp}(s) = \frac{1}{1+T_f s} \quad \omega_b < \frac{1}{T_f} < \omega_e \quad (\text{rad/s})$$

This filter will suppress disturbances over the frequency $1/T_f$.

This criterion is hard to satisfy for smaller vessels.

Higher-order low-pass filters can be designed by using a *Butterworth filter*:

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

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

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

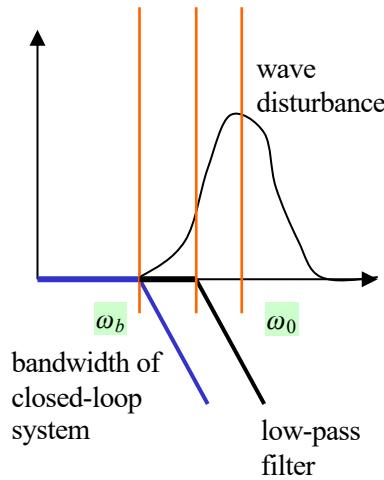
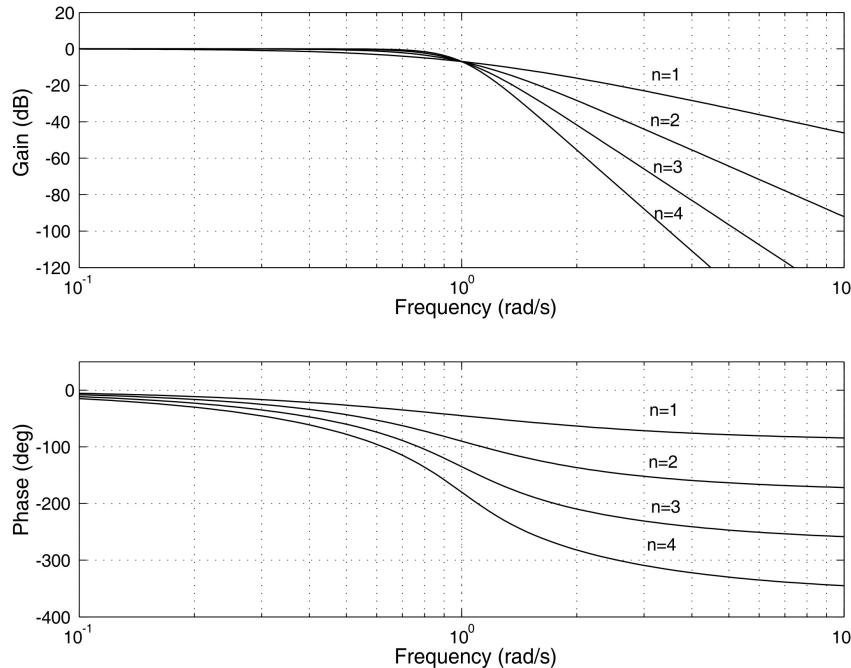
$$(n = 1)h_{lp}(s) = \frac{1}{1 + s/\omega_f}$$

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

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

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

13.2.1 Low-Pass Filtering



A higher-order low-pass filter implies better disturbance suppression to the price of additional phase lag

13.2.2 Cascaded Low-Pass and Notch Filtering

For **smaller craft** the bandwidth of the controller can be close to or within the range of the wave spectrum. This problem can be handled by using a *low-pass filter in cascade with a notch filter*:

$$\hat{\psi}(s) = h_{lp}(s)h_n(s)y(s)$$

where

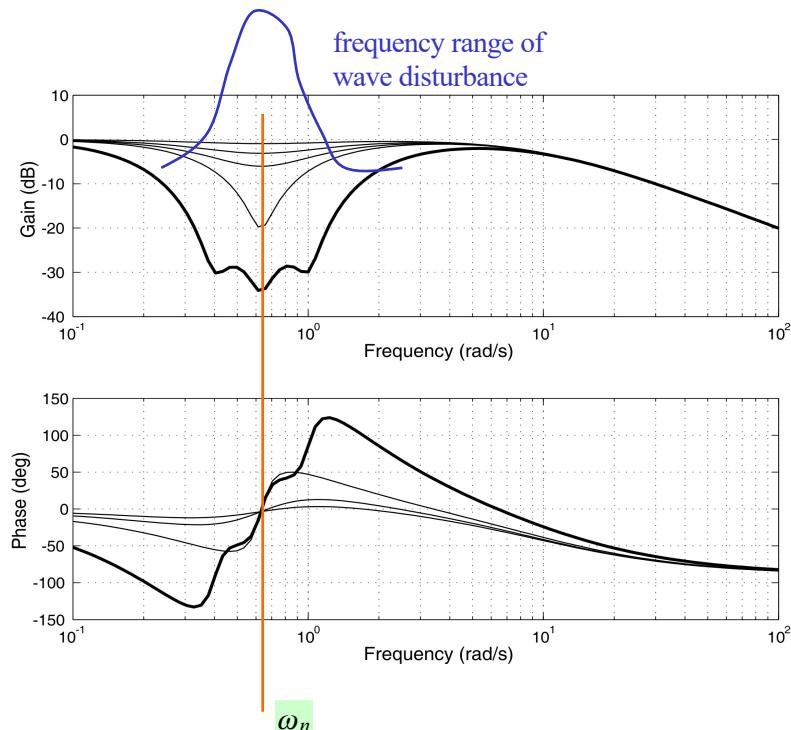
$$h_n(s) = \frac{s^2 + 2\zeta\omega_n s + \omega_n^2}{(s + \omega_n)^2}$$

For a vessel moving at forward speed ***U*** the optimal notch frequency will be:

$$\omega_n = \omega_e$$

but... notch filtering also introduces additional phase lag!

therefore... use Kalman filtering or a linear/nonlinear observer



13.2.3 Wave-Frequency Estimation

FFT for computation of the heave-response spectrum
 Unfortunately, creating a FFT frequency spectrum takes time and it results in back-dated information when estimating the time-varying wave encounter frequency. This is due to the moving window necessary for applying the FFT algorithm.

However, it is possible to estimate ω_e if speed and heading are constant for a period of time (typically 30 minutes). The best results are obtained by using the heave response, which can be logged by using an accelerometer. Good results are also obtained for pitch angle response data.

Matlab:

The Matlab example file `ExFFT.m` in the MSS toolbox shows how the wave encounter frequency can be estimated from response data using FFT. The heave acceleration time series are generated using the wave spectrum

$$a_z = \frac{10 \text{ s}}{s^2 + 2\lambda\omega_e + \omega_e^2} w \quad (13.16)$$

where w is Gaussian white noise. This is compared to the data of a regular wave $a_z = A \cos(\omega_e t)$. The unknown peak frequency is chosen as $\omega_e = 0.8 \text{ rad/s}$ and the peaks are easily observed in the FFT plots of Figure 13.7.

```

we = 0.8;                                % peak frequency [rad/s]

fs = 500;                                 % IMU sampling frequency [Hz]
h = 1/fs;                                 % sampling time [s]
N = 30 * 60 * fs;                         % 30 minutes data
t = (0:N-1) * h;                          % time vector [s]

% Wave spectrum data and sinusoidal (regular) waves
Kw = 10; lambda = 0.1;                     % wave spectrum
sys = tf([Kw 0],[1 2*lambda*we we*we]);
[mag,phase,wout] = bode(sys,logspace(-1,0.2,1000));
mag = reshape(mag(1,:),1,1000);
x1 = lsim(sys,randn(1,length(t)),t,0,'zoh');    % time responses
x2 = cos(we * t);
X = [x1; x2];

% Fast Fourier transform (FFT)
n = 2^nextpow2(N);                      % pad the input with trailing zeros
Y = fft(X,n,2);                         % compute the FFT
P2 = abs(Y/N);                           % double-sided spectrum of each signal
P1 = P2(:,1:n/2+1);                     % single-sided spectrum of each signal
P1(:,2:end-1) = 2*P1(:,2:end-1);

% Plots
f = 0:(fs/n):(fs/2-fs/n); w = 2*pi*f;  % frequency vectors
M = 600;                                  % no of samples to plot
subplot(2,1,1);
plot(w(1:M),P1(1,1:M)/max(P1(1,1:M)),wout,mag/max(mag));
title(['Normalized wave spectrum in the frequency domain']);
subplot(2,1,2);
plot(w(1:M),P1(2,1:M));
title(['Normalized sinusoidal in the frequency domain']);

```

13.2.3 Wave-Frequency Estimation

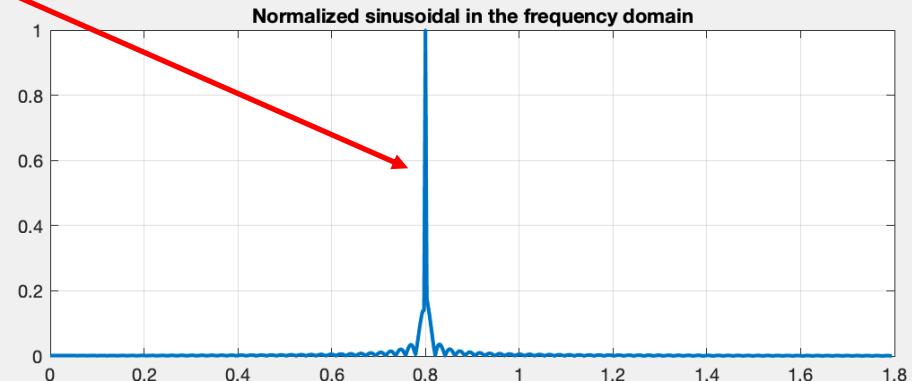
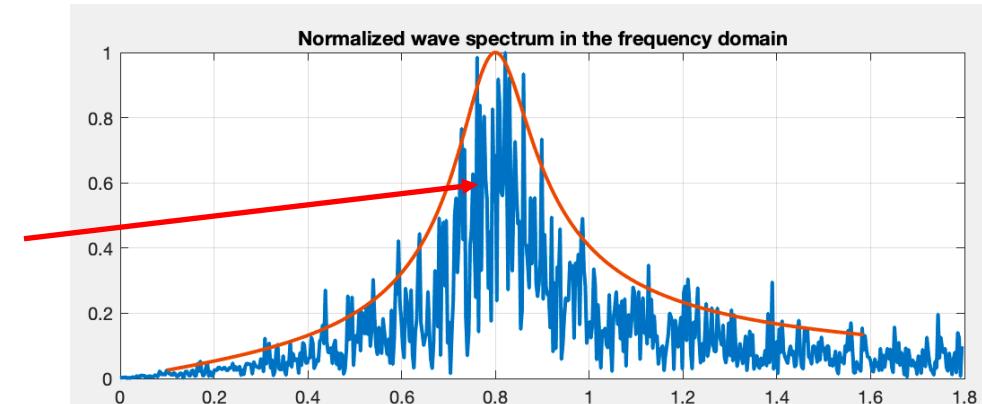
Matlab example file ExFFT.m in the MSS toolbox

The FFT applied to a moving window of data generated by using the following signals:

Wave spectrum $a_z = \frac{10 \text{ s}}{s^2 + 2\lambda\omega_e + \omega_e^2} w$

Sinusoidal wave $a_z = A \cos(\omega_e t)$

The wave encounter frequency $\omega_e = 0.8 \text{ rad/s}$ is observed as the peak frequency of both data sets.



13.2.3 Wave-Frequency Estimation

Nonlinear observer for online estimation of the wave encounter frequency

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \omega_f^2(y - x_1) - 2\omega_f x_2 \\ \dot{\hat{\theta}}_w &= k_f x_1 (\dot{x}_2 - \hat{\theta}_w x_1) \\ \hat{\theta}_w &:= -\hat{\omega}_e^2\end{aligned}$$

Adaptive gain-switching algorithm

$$k(\hat{A}) = \begin{cases} k_{\text{init}} & \text{if } t \leq t_{\text{init}} \\ k_{\text{min}} & \text{if } t > t_{\text{init}} \text{ and } \hat{A} > A_c \\ k_{\text{max}} & \text{if } t > t_{\text{init}} \text{ and } \hat{A} \leq A_c \end{cases}$$

$$T_f \dot{k}_f + k_f = k(\hat{A})$$

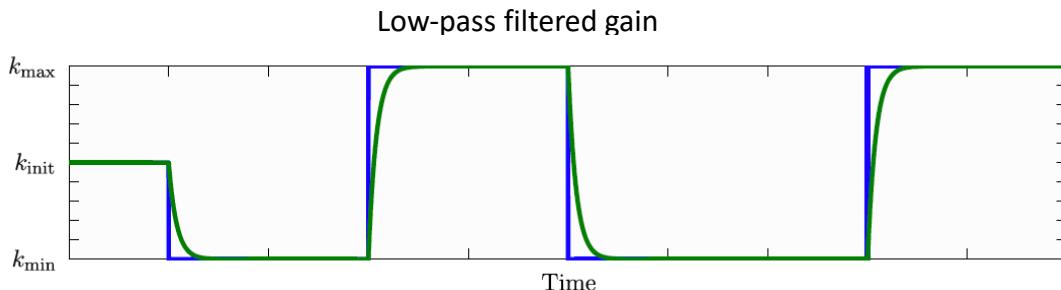
Regular wave with unknown frequency

$$y = A \sin(\omega_e t + \epsilon)$$

$$\omega_e < \omega_f,$$

Amplitude estimator

$$\begin{aligned}\chi &= \frac{1}{Ts + 1} y^2 \\ \hat{A} &= \sqrt{2\chi}\end{aligned}$$



D. J. Belleter, R. Galeazzi and T. I. Fossen. Experimental Verification of a Globally Exponentially Stable Nonlinear Wave Encounter Frequency Estimator. *Ocean Engineering*, Elsevier, Volume 97, No. 15 March 2015, pp. 48–56.

13.2.3 Wave-Frequency Estimation

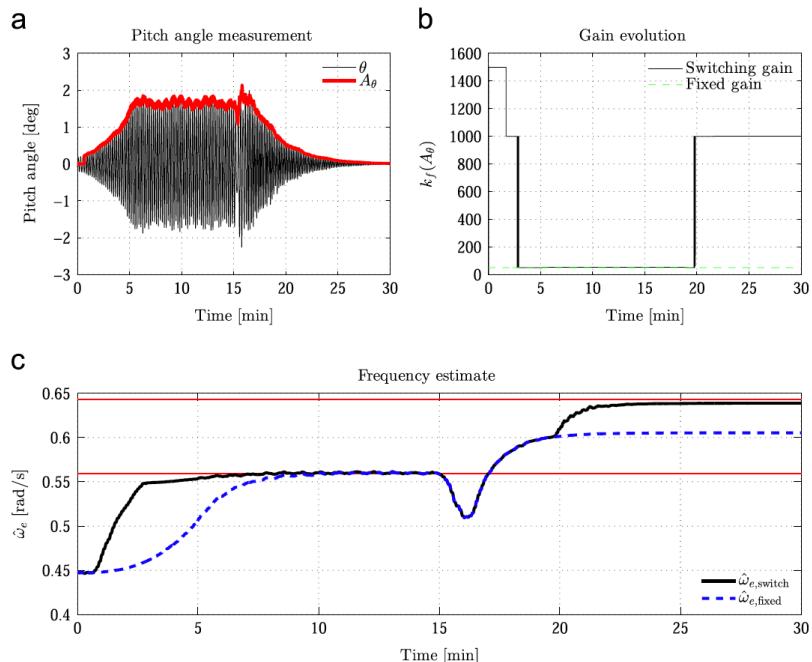
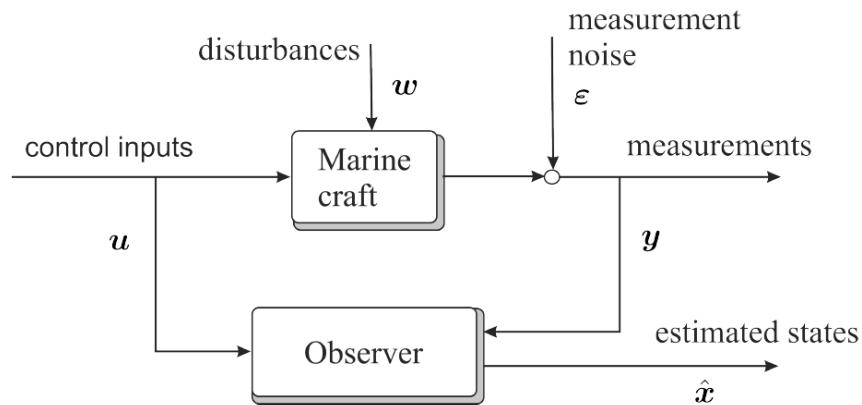


Fig. 4. Comparison of the frequency estimator with and without a gain switching mechanism.

D. J. Belleter, R. Galeazzi and T. I. Fossen. Experimental Verification of a Globally Exponentially Stable Nonlinear Wave Encounter Frequency Estimator. *Ocean Engineering*, Elsevier, Volume 97, No. 15 March 2015, pp. 48–56.

13.3 Fixed-Gain Observer Design

The simplest state estimator is designed as a fixed-gain observer where the goal of the observer is to reconstruct the unmeasured state vector \mathbf{x} from the measurements \mathbf{u} and \mathbf{y} of a dynamical system.



Observers are derived from **deterministic models**, which neglects process w and measurement ε noise. However, an observer will still work when adding Gaussian white noise to the system if the gains are tuned properly.

State estimators and observers can only be designed for systems that are observable!

13.3.1 Observability

Definition (Observability in LTI Systems)

Consider the linear time-invariant system

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

The pair (A, C) is observable if and only if the observability matrix

$$\mathcal{O} = [C^\top \mid A^\top C^\top \mid \dots \mid (A^\top)^{n-1} C^\top]$$

has full column rank or (at least) a left inverse exists.

Definition (Observability in LTV Systems)

Consider the linear time-varying system

$$\dot{x} = A(t)x + B(t)u$$

$$y = C(t)x + D(t)u$$

The pair $(A(t), C(t))$ is observable if and only if the is observable at time t_0 if and only if there exists a finite $t_1 > t_0$ such that the $n \times n$ the Observability Gramian

$$W(t_0, t_1) = \int_{t_0}^{\infty} \Phi^\top(t_1, \tau) C^\top(\tau) C(\tau) \Phi(t_1, \tau) d\tau$$

is nonsingular when $\Phi(t, \tau)$ is the state transition matrix of the system $\dot{x} = A(t)x$.

13.3.2 Luenberger Observer

Observer goal: reconstruct the unmeasured state vector \mathbf{x} from the measurements \mathbf{u} and \mathbf{y}

LTI system

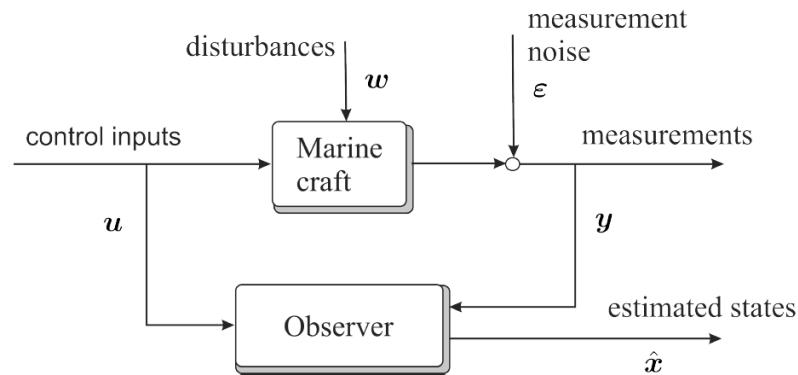
$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

An observer copying the dynamics is

$$\begin{aligned}\dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}(\mathbf{y} - \hat{\mathbf{y}}) \\ \hat{\mathbf{y}} &= \mathbf{C}\hat{\mathbf{x}} + \mathbf{D}\mathbf{u}\end{aligned}$$

Injection term



where \mathbf{K} is an observer gain matrix to be constructed such that $\hat{\mathbf{x}} \rightarrow \mathbf{x}$ as $t \rightarrow \infty$

Error dynamics

$$\dot{\tilde{\mathbf{x}}} = (\mathbf{A} - \mathbf{K}\mathbf{C})\tilde{\mathbf{x}}$$

Asymptotic convergence of $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ to zero can be obtained for a constant \mathbf{K} if the pair (\mathbf{A}, \mathbf{C}) is observable.

Matlab:

If the observability matrix \mathcal{O} is nonsingular, the poles of the error dynamics can be placed in the left half-plane. The rank of \mathcal{O} is checked by `rank(obsv(A, C))` while the observer gain matrix \mathbf{K} is computed using

```

p = [p1, ..., pn]' % vector of distinct observer error poles
K = place(A', C, p)', % observer gain matrix
  
```

Note that both \mathbf{K} and \mathbf{A} are transposed, since the dual problem of the regulator problem is solved.

13.3.3 Case Study: Luenberger Observer for Heading Autopilot

Nomoto Ship Model Exposed to Wind, Waves and Ocean Currents

Let a 1st-order [Nomoto model](#) describe the LF motion of a ship:

$$\begin{aligned}\dot{\psi} &= r \\ \dot{r} &= -\frac{1}{T}r + \frac{K}{T}(\delta - b) + w_2 \\ \dot{b} &= w_3\end{aligned}$$

where b is the rudder offset (counteracts slowly-varying moments on the ship due to wave drift forces, LF wind and ocean currents). A [linear wave model](#) can be used to model the wave response

$$\begin{aligned}\dot{\xi}_w &= \psi_w \\ \dot{\psi}_w &= -\omega_0^2 \xi_w - 2\lambda\omega_0 \psi_w + K_w w_1\end{aligned}$$

$$h(s) = \frac{K_w s}{s^2 + 2\lambda\omega_0 s + \omega_0^2}$$

The process noise terms, w_1, w_2 , and w_3 are modeled as white noise processes.

The [compass measurement equation](#) can be expressed by the sum

$$y = \psi + \psi_w + \varepsilon$$

where ε represents zero-mean Gaussian measurement noise.

Notice that the yaw rate r nor the wave states ξ_w and ψ_w are measured.



13.3.3 Case Study: Luenberger Observer for Heading Autopilot

Nomoto Ship Model Exposed to Wind, Waves and Ocean Currents

State-space model:

$$\dot{x} = Ax + bu + Ew$$

$$y = c^\top x + \varepsilon$$

$$x = [\xi_w, \dot{\psi}_w, \psi, r, b]^\top$$

$$w = [w_1, w_2, w_3]^\top$$

$$u = \delta,$$

$$A = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 & 0 & 0 \end{array} \right], \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \\ 0 \end{bmatrix}$$

$$E = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ K_w & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right], \quad c^\top = [0, 1, 1, 0, 0]$$



Is the ship model exposed to environmental forces observable? **YES**

This implies that *yaw rate*, *bias* and *waves states* can be estimated using a single compass measurement!

→ see example on next page

13.3.3 Case Study: Luenberger Observer for Heading Autopilot

Matlab:

The following example shows how the Luenberger observer gains of a ship autopilot system can be computed in Matlab.

Example 13.2 (Luenberger Observer Gains)

It is straightforward to see that the autopilot model with WF, wind and ocean current model (13.52)–(13.53) is observable from the input δ to the compass measurement y . Let $K = 1$, $T = 50$, $\lambda = 0.1$ and $\omega_0 = 1$, then

```
K = 1; T = 50; lambda = 0.1; w0 = 1;
A = [0 1 0 0 0
      -w0^2 -2*lambda*w0 0 0 0
      0 0 0 1 0
      0 0 0 -1/T -K/T
      0 0 0 0 0];
c = [0 1 1 0 0]';
n = rank(obsv(A, c'))
```

results in $n = 5$ corresponding to $\text{rank}(\mathcal{O}) = 5$. Hence, the system is observable according to Definition 13.2, implying that the states r , b , ψ_w and ξ_w can be reconstructed from a single measurement $y = \psi + \psi_w + \varepsilon$ using a Luenberger

observer

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + bu + k(y - \hat{y}) \\ \hat{y} &= c^\top \hat{x}\end{aligned}$$

The filter gains can be computed by pole placement, for instance

```
k = place(A', c, [p1, p2, p3, p4, p5])'
```

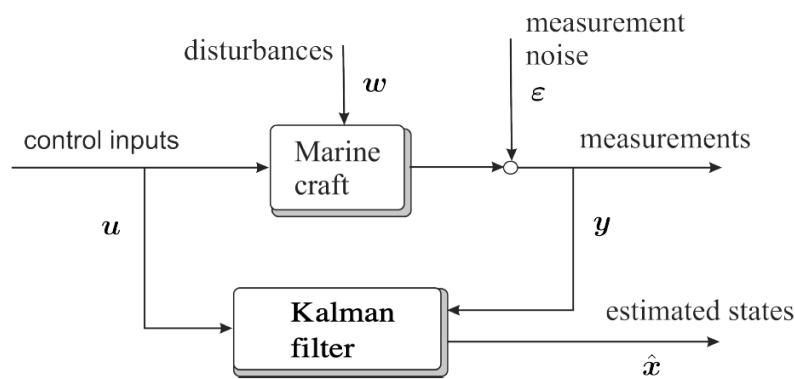
where $p1, p2, p3, p4$ and $p5$ are the desired closed-loop poles of the error dynamics (13.43).



13.4 Kalman Filter Design

Since 1960 the discrete-time [Kalman filter](#) (KF), and nonlinear extensions thereof, has been used to provide integrated navigation solutions based on different types of measurements.

- KF is an alternative solution to the pole-placement technique to apply the discrete-time [Kalman filter](#) (Kalman 1960) to compute the estimator gain matrix \mathbf{K} .
- Kalman filtering (or *optimal state estimation* in sense of minimum variance) allows the user to estimate the state \mathbf{x} of a dynamic system from a noise-contaminated input-output pair (\mathbf{u} , \mathbf{y}).



w is a zero-mean Gaussian white noise process with covariance matrix $\mathbf{Q} = \mathbf{Q}^T > 0$

ε is a zero-mean Gaussian white noise process with covariance matrix $\mathbf{R} = \mathbf{R}^T > 0$

If the system is [observable](#), the state vector \mathbf{x} can be reconstructed recursively through the measurement vector \mathbf{y} and the control input vector \mathbf{u}

The discrete-time KF is used in millions of applications, and it is the core algorithm of all modern navigation systems.

13.4.1 Discrete-Time Kalman Filter

The KF is implemented in discrete time since the measurements are transmitted at different frequencies. The discrete-time KF (Kalman 1960) is optimal for computer simulations and practical computations. Hence, we will not study the continuous-time KF.

Discrete-time state-space model

$$\begin{aligned}\mathbf{x}[k+1] &= \mathbf{A}_d[k]\mathbf{x}[k] + \mathbf{B}_d[k]\mathbf{u}[k] + \mathbf{E}_d[k]\mathbf{w}[k] \\ \mathbf{y}[k] &= \mathbf{C}_d[k]\mathbf{x}[k] + \mathbf{D}_d[k]\mathbf{u}[k] + \boldsymbol{\varepsilon}[k]\end{aligned}$$

LTI systems where \mathbf{A}^{-1} exists (Appendix B1.1)

$$\begin{aligned}\mathbf{A}_d[k] &= \Phi \\ \mathbf{B}_d[k] &= \mathbf{A}^{-1}(\Phi - \mathbf{I}_n)\mathbf{B} \\ \mathbf{C}_d[k] &= \mathbf{C} \\ \mathbf{D}_d[k] &= \mathbf{D} \\ \mathbf{E}_d[k] &= \mathbf{A}^{-1}(\Phi - \mathbf{I}_n)\mathbf{E}\end{aligned}$$

$$\Phi \approx \mathbf{I}_n + \mathbf{A}h + \frac{1}{2}(\mathbf{A}h)^2 + \cdots + \frac{1}{N!}(\mathbf{A}h)^N$$

General solution avoiding \mathbf{A}^{-1}

$$\begin{aligned}\mathbf{A}_d &= e^{\mathbf{A}h} \\ \mathbf{B}_d &= \int_0^h e^{\mathbf{A}\tau} \mathbf{B} d\tau\end{aligned}$$

Matlab:

The discretized system matrices for an LTI systems are computed in Matlab by

$$\begin{aligned}[\mathbf{Ad}, \mathbf{Bd}] &= \text{c2d}(\mathbf{A}, \mathbf{B}, h) \\ [\mathbf{Ad}, \mathbf{Ed}] &= \text{c2d}(\mathbf{A}, \mathbf{E}, h)\end{aligned}$$

13.4.1 Discrete-Time Kalman Filter

Wikimedia commons:

https://commons.wikimedia.org/wiki/File:Rudolf_Kalman.jpg

Table 13.1: Discrete-time Kalman filter.

Initial values	$\hat{\mathbf{x}}^-[0] = \mathbf{x}_0$ $\hat{\mathbf{P}}^-[0] = \mathbb{E}[(\mathbf{x}[0] - \hat{\mathbf{x}}^-[0])(\mathbf{x}[0] - \hat{\mathbf{x}}^-[0])^\top] = \mathbf{P}_0$
KF gain	$\mathbf{K}[k] = \hat{\mathbf{P}}^-[k] \mathbf{C}_d^\top[k] \left(\mathbf{C}_d[k] \hat{\mathbf{P}}^-[k] \mathbf{C}_d^\top[k] + \mathbf{R}_d[k] \right)^{-1}$
State corrector	$\hat{\mathbf{x}}[k] = \hat{\mathbf{x}}^-[k] + \mathbf{K}[k] (\mathbf{y}[k] - \mathbf{C}_d[k] \hat{\mathbf{x}}^-[k] - \mathbf{D}_d[k] \mathbf{u}[k])$
Covariance corrector	$\hat{\mathbf{P}}[k] = (\mathbf{I}_n - \mathbf{K}[k] \mathbf{C}_d[k]) \hat{\mathbf{P}}^-[k] (\mathbf{I}_n - \mathbf{K}[k] \mathbf{C}_d[k])^\top + \mathbf{K}[k] \mathbf{R}_d[k] \mathbf{K}^\top[k]$
State predictor	$\hat{\mathbf{x}}^-[k+1] = \mathbf{A}_d[k] \hat{\mathbf{x}}[k] + \mathbf{B}_d[k] \mathbf{u}[k]$
Covariance predictor	$\hat{\mathbf{P}}^-[k+1] = \mathbf{A}_d[k] \hat{\mathbf{P}}[k] \mathbf{A}_d^\top[k] + \mathbf{E}_d[k] \mathbf{Q}_d[k] \mathbf{E}_d^\top[k]$

where

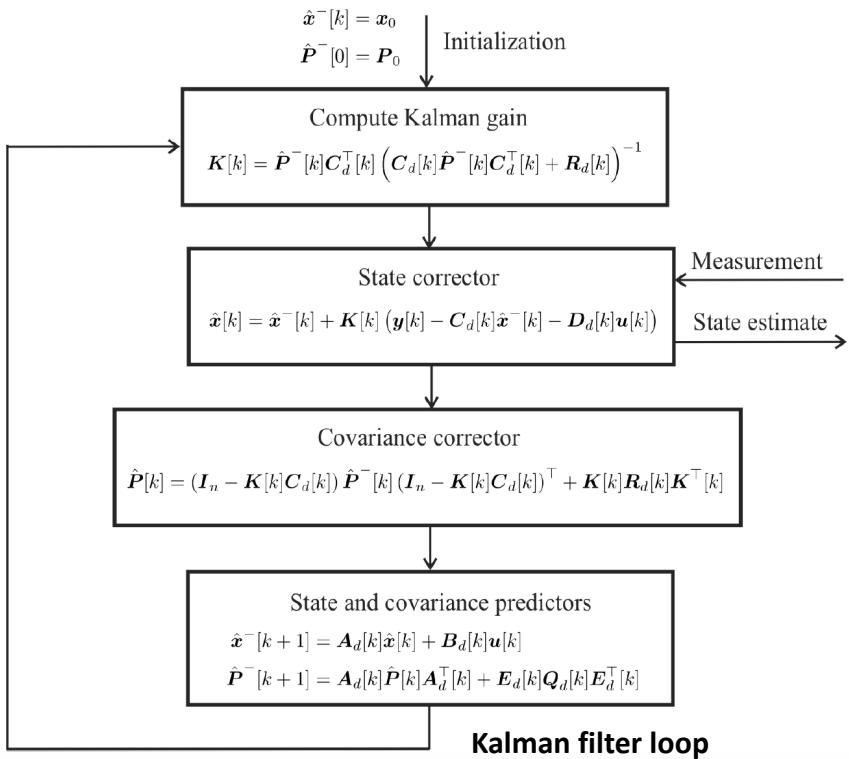
- $\mathbf{Q}_d, \mathbf{R}_d$ Covariance matrices for the process and measurement noises
- $\hat{\mathbf{x}}^-, \hat{\mathbf{P}}^-$ *A priori* state and covariance matrix estimates (before update)
- $\hat{\mathbf{x}}, \hat{\mathbf{P}}$ *A posteriori* state and covariance matrix estimates (after update)



Rudolf E. Kálmán (1930 – 2016)
Hungarian-American electrical
engineer, mathematician and inventor.



13.4.1 Discrete-Time Kalman Filter



Matlab:

The pseduocode for the KF loop in Figure 13.11 is:

```

% initialization
x_prd = x0;    Qd = constant;
P_prd = P0;    Rd = constant;

% MAIN LOOP
for i=1:N

    % KF gain: K[k]
    K = P_prd * Cd' * inv( Cd * P_prd * Cd' + Rd );
    IKC = eye(n) - K * Cd;

    % Measurement: y[k]

    y = ...;

    % Corrector: x_hat[k] and P_hat[k]
    x_hat = x_prd + K * ( y - Cd * x_prd - Dd * u );
    P_hat = IKC * P_prd * IKC' + K * Rd * K';

    % Predictor: x_prd[k+1] and P_prd[k+1]
    x_prd = Ad * x_hat + Bd * u;
    P_prd = Ad * P_hat * Ad' + Ed * Qd * Ed';

end

```

13.4.2 Discrete-Time Extended Kalman Filter

The Kalman filter can also be applied to nonlinear systems with non-additive input and process noise

$$\dot{x} = f(x, u, w)$$

$$y = h(x, u) + \varepsilon$$

If the system is observable, the state vector can be estimated using the **discrete-time EKF**, which is based on the discrete-time predictor is

$$x[k+1] = x[k] + h f(x[k], u[k], 0) \quad \text{Euler's method}$$

$$y[k] = h(x[k], u[k])$$

Disadvantages

- The linear KF is an optimal estimator and it is easy to establish stability and convergence properties thanks to linear system theory. Unfortunately, optimality is lost when applying the EKF since it relies on linearization.
- If the initial estimate of the state is wrong, or if the process is modeled incorrectly, the EKF may quickly diverge, owing to its linearization.
- Another problem with the EKF is that the estimated covariance matrix tends to underestimate the true covariance matrix and therefore risks becoming inconsistent in the statistical sense. Care should also be taken with respect to covariance blow-up and instability.

However, the EKF gives excellent performance in most navigation systems at it is the de facto standard in aided INS

13.4.2 Discrete-Time Extended Kalman Filter

Table 13.2: Discrete-time extended Kalman filter.

Initial values	$\hat{\mathbf{x}}^-[0] = \mathbf{x}_0$ $\hat{\mathbf{P}}^-[0] = \mathbb{E}[(\mathbf{x}[0] - \hat{\mathbf{x}}^-[0])(\mathbf{x}[0] - \hat{\mathbf{x}}^-[0])^\top] = \mathbf{P}_0$
KF gain	$\mathbf{K}[k] = \hat{\mathbf{P}}^-[k] \mathbf{C}_d^\top[k] \left(\mathbf{C}_d[k] \hat{\mathbf{P}}^-[k] \mathbf{C}_d^\top[k] + \mathbf{R}_d[k] \right)^{-1}$
State corrector	$\hat{\mathbf{x}}[k] = \hat{\mathbf{x}}^-[k] + \mathbf{K}[k] (\mathbf{y}[k] - \mathbf{h}[k](\hat{\mathbf{x}}^-[k], \mathbf{u}[k]))$
Covariance corrector	$\hat{\mathbf{P}}[k] = (\mathbf{I}_n - \mathbf{K}[k] \mathbf{C}_d[k]) \hat{\mathbf{P}}^-[k] (\mathbf{I}_n - \mathbf{K}[k] \mathbf{C}_d[k])^\top + \mathbf{K}[k] \mathbf{R}_d[k] \mathbf{K}^\top[k]$
State predictor	$\hat{\mathbf{x}}^-[k+1] = \mathbf{I}_n + h \mathbf{f}[k](\hat{\mathbf{x}}[k], \mathbf{u}[k], \mathbf{0})$
Covariance predictor	$\hat{\mathbf{P}}^-[k+1] = \mathbf{A}_d[k] \hat{\mathbf{P}}[k] \mathbf{A}_d^\top[k] + \mathbf{E}_d[k] \mathbf{Q}_d[k] \mathbf{E}_d^\top[k]$

The discrete-time system matrices are defined by the *Jacobians*

$$\mathbf{A}_d[k] = \mathbf{I}_n + h \left. \frac{\partial \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k], \mathbf{w}[k])}{\partial \mathbf{x}[k]} \right|_{\mathbf{x}[k]=\hat{\mathbf{x}}[k], \mathbf{w}[k]=\mathbf{0}}$$

$$\mathbf{E}_d[k] = h \left. \frac{\partial \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k], \mathbf{w}[k])}{\partial \mathbf{w}[k]} \right|_{\mathbf{x}[k]=\hat{\mathbf{x}}[k], \mathbf{w}[k]=\mathbf{0}}$$

$$\mathbf{C}_d[k] = \left. \frac{\partial \mathbf{h}(\mathbf{x}[k], \mathbf{u}[k])}{\partial \mathbf{x}[k]} \right|_{\mathbf{x}[k]=\hat{\mathbf{x}}^-[k]}$$

13.4.3 Modification for Euler Angles to Avoid Discontinuous Jumps

Care must be taken when implementing attitude controllers and state estimators using Euler angles since the roll, pitch and yaw angles are confined to the interval $[0, 2\pi)$ or $[-\pi, \pi)$, also known as the 1-sphere or the topological space S^1 corresponding to a circle in the plane.

When implementing a heading control system for a marine craft, it is crucial that the angle difference is mapped to the **smallest signed angle (SSA)** between the current heading and the reference .

To illustrate why, consider a craft with an actual heading of zero degrees and a heading setpoint of 355 degrees. A naive controller implementation would calculate a heading error of -355 degrees, thus commanding a near full rotation, going clockwise, although the setpoint is only 5 degrees away in the opposite direction.

Definition 13.5 (Smallest Signed Angle (SSA))

The operator $ssa : \mathbb{R} \rightarrow [-\pi, \pi)$ maps the unconstrained angle $\tilde{x} = x - x_0 \in \mathbb{R}$ representing the difference between the two angles x and x_0 to the smallest difference between the angles

$$\tilde{x}_s = ssa(\tilde{x})$$

where $\tilde{x}_s \in \mathbb{S}^1$.



13.4.3 Modification for Euler Angles to Avoid Discontinuous Jumps

Matlab:

MSS Matlab function `ssa.m` for the smallest signed angle.

```
function angle = ssa(angle,unit)
% SSA is the "smallest signed angle" or the smallest difference
% between two angles. Examples:
% >> angle = ssa(angle) maps an angle in rad to [-pi pi)
% >> angle = ssa(angle,'deg') maps an angle in deg to [-180 180)
%
% Note that in many languages (C, C++, C#, JavaScript), the
% operator mod(x,y) returns a value with the same sign as x.
% For these use a custom mod function: mod(x,y) = x-floor(x/y)*y
% For the Unity game engine use: Mathf.DeltaAngle

if (nargin == 1)
    angle = mod( angle + pi, 2 * pi ) - pi;
elseif strcmp(unit,'deg')
    angle = mod( angle + 180, 360 ) - 180;
end
```



13.4.3 Modification for Euler Angles to Avoid Discontinuous Jumps

PD control using the **Smallest Signed Angle** to avoid Discontinuous Jumps

Consider the yaw angle dynamics

$$\begin{aligned}\dot{\psi} &= r \\ T\dot{r} + r &= K\delta\end{aligned}$$

$$\delta = T\dot{r}_d + r_d - K_p \text{ssa}(\tilde{\psi}) - K_d \tilde{r}$$

ssa modified yaw angle error

Error dynamics

$$\ddot{\tilde{\psi}} + \frac{(1 + KK_d)}{T} \dot{\tilde{\psi}} + \frac{KK_p}{T} \tilde{\psi}_s = 0 \quad \tilde{\psi}_s = \text{ssa}(\tilde{\psi}) \quad \tilde{\psi} = \psi - \psi_d$$



The equilibrium point $(\tilde{\psi}_s, \dot{\tilde{\psi}}) = (0, 0)$ is exponentially stable and globally attractive for the the entire domain $\mathbb{S}^1 \times \mathbb{R}$

Proof: see Coates et al. (2021).

Remark:

as shown by Bhat and Bernstein (2000), systems with rotational degrees of motion cannot be globally stabilized by continuous feedback due to the topological obstruction imposed by $\text{SO}(3)$, that is $\tilde{\psi}$ is defined on the 1-sphere \mathbb{S}^1 and not on \mathbb{R} .

13.4.3 Modification for Euler Angles to Avoid Discontinuous Jumps

State Estimation using the **Smallest Signed Angle** to avoid Discontinuous Jumps

Consider the discrete-time Nomoto model for a marine craft

$$\psi[k+1] = \psi[k] + hr[k]$$

$$r[k+1] = r[k] + \frac{h}{T} (-r[k] + K\delta[k] + w[k])$$

$$y[k] = \psi[k] + \varepsilon[k]$$

State estimator

ssa modified injection terms

$$\hat{\psi}[k] = \hat{\psi}^-[k] + K_1 \text{ssa}(y[k] - \hat{y}^-[k])$$

$$\hat{r}[k] = \hat{r}^-[k] + K_2 \text{ssa}(y[k] - \hat{y}^-[k])$$

$$\hat{y}^-[k] = \hat{\psi}^-[k]$$

For the continuous system where the estimation error is denoted $\tilde{y} = y - \hat{y}$ it follows from Coates *et al.* (2021) that the equilibrium point of the estimation error $(\tilde{y}_s, \dot{\tilde{y}}) = (0, 0)$ is exponentially stable and globally attractive for the the entire domain $\mathbb{S}^1 \times \mathbb{R}$.



13.4.4 Modification for Asynchronous Measurement Data

Synchronous Measurements

When implementing a discrete-time KF it is practical to choose the sampling frequency

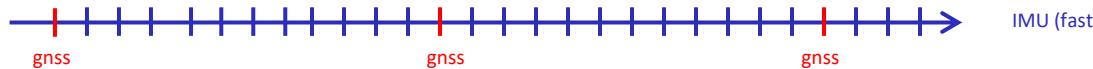
$$f_s = 1/h$$

of the system model equal to the measurement frequency such that the states can be propagated from time t_k to time $t_{k+1} = h t_k$ where h is the sampling time.

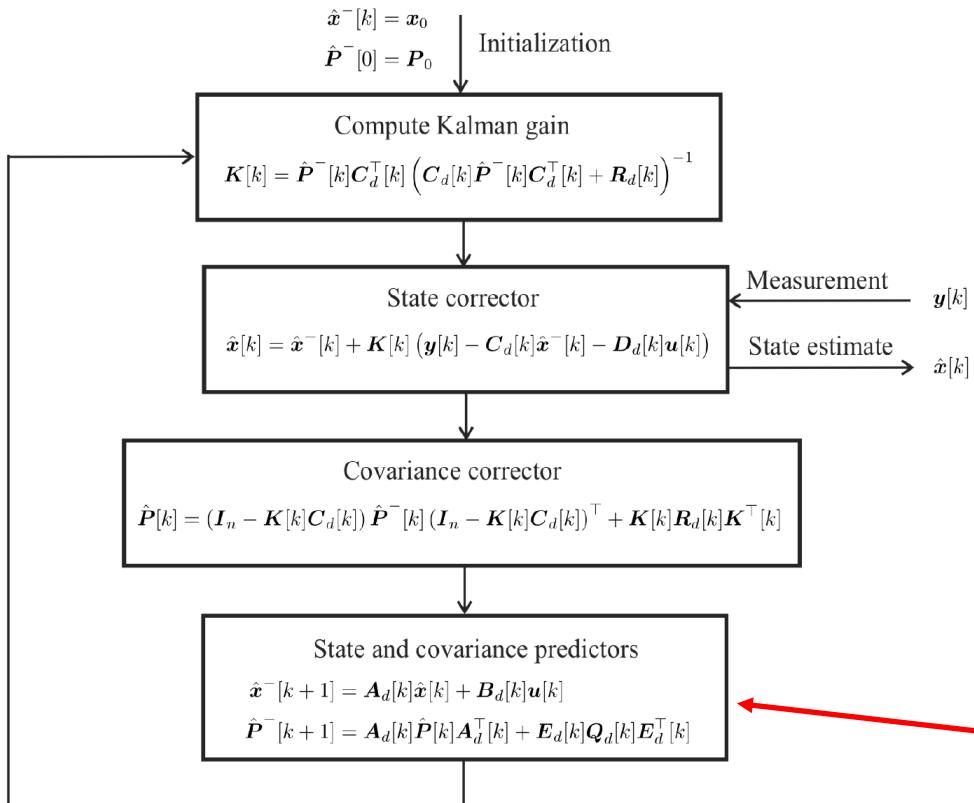
Asynchronous Measurements

In embedded computer systems the measurements can be received at different frequencies than the sampling frequency of the system.

- **GNSS** measurements are typically sampled at $f_{\text{gnss}} = 1 \text{ Hz}$
- **Inertial Sensors** (accelerometers and gyros) can be sampled at a user specified frequency $f_{\text{imu}} >> f_{\text{gnss}}$ (MEMS-based IMUs can operate at frequencies much larger than 1000 Hz)



13.4.4 Modification for Asynchronous Measurement Data



Assume that the sampling frequency is chosen equal to the frequency of the fastest measurement frequency

$$f_s = f_{\text{imu}}$$

Let the **integer Z** denote the ratio between the sampling frequency and the slower GNSS measurement frequency

$$Z = \frac{f_s}{f_{\text{gnss}}} > 1$$

It is practical to choose Z as an integer such that the slow measurement appear each Z time in the KF logical loop.

In addition, it is necessary to set the corrector equations equal to the predicted values when there are **no measurement** (see pseudocode on next page)

$$\hat{x}[k] = \hat{x}^-[k]$$

$$\hat{P}[k] = \hat{P}^-[k]$$

13.4.4 Modification for Asynchronous Measurement Data

Matlab:

Pseudocode for implementation of a discrete-time KF with asynchronous measurement data:

```

f_s      = 100;          % sampling frequency [Hz]
f_imu   = f_s;          % IMU (fast) measurement frequency [Hz]
f_gnss  = 1;            % GNSS (slow) measurement frequency [Hz]

h = 1/f_s;             % sampling time
h_gnss = 1/f_gnss;     % GNSS sampling time

% MAIN LOOP
for i=1:N

    % GNSS measurements are Z = f_s/f_gnss times slower than
    % the sampling frequency f_s
    if mod( t, h_gnss ) == 0
        y = ...;           % new measurement: y[k] ←

        % KF gain: K[k]
        K = P_prd * Cd' * inv( Cd * P_prd * Cd' + Rd );
        IKC = eye(n) - K * Cd;

        % Corrector: x_hat[k] and P_hat[k]
        x_hat = x_prd + K * ( y - Cd * x_prd - Dd * u );
        P_hat = IKC * P_prd * IKC' + K * Rd * K';

    else
        x_hat = x_prd;      % no measurement ←
        P_hat = P_prd;
    end

    % Predictor: x_prd[k+1] and P_prd[k+1]
    x_prd = Ad * x_hat + Bd * u;
    P_prd = Ad * P_hat * Ad' + Ed * Qd * Ed';

end

```

Kalman filter runs at a sampling frequency of **100 Hz**

IMU measurements are received at **100 Hz**

GNSS measurements are received at **1 Hz**

IF a slow GNSS position measurement (1 Hz) is received, compute the Kalman gain and run the corrector

ELSE, there are **no GNSS measurements** and the corrector equations are set equal to the predicted values

13.4.5 Case Study: KF Design for Heading Autopilots

The main sensor components for a heading-controlled ship are:

- Magnetic and/or gyroscopic compasses measuring the yaw angle
- Attitude rate sensor (ARS) measuring the yaw rate

In many commercial systems only the compass is used for feedback control since the yaw rate can be estimated quite well by a state estimator.

State-space model:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u + \mathbf{E}\mathbf{w}$$

$$y = \mathbf{c}^\top \mathbf{x} + \varepsilon$$

$$\mathbf{x} = [\xi_w, \dot{\psi}_w, \psi, \dot{r}, b]^\top$$

$$u = \delta,$$

$$\mathbf{w} = [w_1, w_2, w_3]^\top$$

$$\mathbf{A} = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 & 0 & 0 \end{array} \right], \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \\ 0 \end{bmatrix}$$

$$\mathbf{E} = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ K_w & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right], \quad \mathbf{c}^\top = [0, 1, 1, 0, 0]$$

$$\mathbf{A}_d = \mathbf{I}_5 + h\mathbf{A},$$

$$\mathbf{b}_d = h\mathbf{b},$$

$$\mathbf{c}_d = \mathbf{c}$$

$$\mathbf{E}_d = h\mathbf{E}$$

$$\dot{\xi}_w = \psi_w$$

$$\dot{\psi}_w = -\omega_0^2 \xi_w - 2\lambda\omega_0 \psi_w + K_w w_1$$

$$\dot{\psi} = r$$

$$\dot{r} = -\frac{1}{T}r + \frac{K}{T}(\delta - b) + w_2$$

$$\dot{b} = w_3$$



13.4.5 Case Study: KF Design for Heading Autopilots

Matlab:

Pseudocode for simulating a discrete-time KF for a ship exposed to first-order WF motions. The only measurement is a gyrocompass but the 5-state ship-wave model is observable so all states can be reconstructed from a single measurement (see Example 13.3.3).

```

x = x0; x_prd = x0;                                % initialization
P_prd = P0;

Qd = diag([q11 q22 q33]);                          % covariance matrices
Rd = r11;

A = [ 0 1 0 0 0
      -w0^2 -2*lambda*w0 0 0 0
      0 0 0 1 0
      0 0 0 -1/T -K/T
      0 0 0 0 0 ];                                     % continuous-time ship model
B = [0 0 0 K/T 0]';
C = [0 1 1 0 0];
E = [0 0 0; Kw 0 0; 0 0 0; 0 1 0; 0 0 1];

Ad = eye(5) + h * A; Bd = h * B;                  % discrete-time KF model
Cd = C; Ed = h * E

% MAIN LOOP
for i=1:N

    % KF gain: K[k]
    K = P_prd * Cd' * inv( Cd * P_prd * Cd' + Rd );

    % PREDICTOR
    x_prd = Ad * x_prd + Bd * u;
    P_prd = Ad * P_prd * Ad' + Ed * Qd * Ed';

    % CORRECTOR
    y = C * x_prd + v;
    x_hat = x_prd + K * (y - Cd * x_prd);
    P_hat = P_prd - K * Cd * P_prd * Cd' * K;

    % PREDICTOR
    x_prd = Ad * x_hat + Bd * u;
    P_prd = Ad * P_hat * Ad' + Ed * Qd * Ed';

    % SHIP-WAVE SIMULATOR
    x = x + h * (A * x + B * u + E * w);
end

```

The example illustrates how the KF gains can be computed in Matlab for a ship exposed to waves. The KF must be modified to handle yaw angle measurements $y = \psi + \psi_w + \varepsilon$ on the intervals $[0^\circ, 360^\circ]$ or $[-180^\circ, 180^\circ]$. The yaw angle injection term is the difference between the measured and the estimated yaw angles. It is necessary to map this signal to $[-\pi, \pi]$ to avoid discontinuous jumps in the estimates. The tool for this is the MSS function `ssa.m` (see Section 13.4.3), which is used to modify the corrector.

```

IKC = eye(5) - K * Cd;

% Control input and measurement: u[k] and y[k]
u = delta;                                     % control system
y = C * x + v;                                  % compass measurement with white noise v

% Corrector: x_hat[k] and P_hat[k]
x_hat = x_prd + K * ssa( y - Cd * x_prd ); % ssa modification
P_hat = IKC * P_prd * IKC' + K * Rd * K';

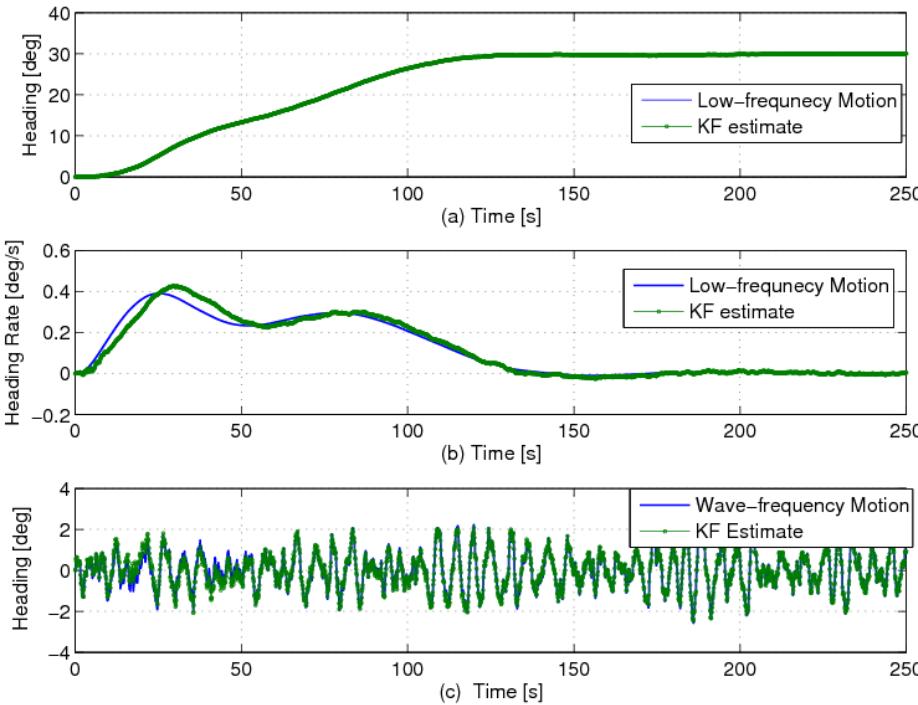
% Predictor: x_prd[k+1] and P_prd[k+1]
x_prd = Ad * x_hat + Bd * u;
P_prd = Ad * P_hat * Ad' + Ed * Qd * Ed';

% Ship-wave simulator: x[k+1]
x = x + h * ( A * x + B * u + E * w );

end

```

13.4.5 Case Study: KF Design for Heading Autopilots

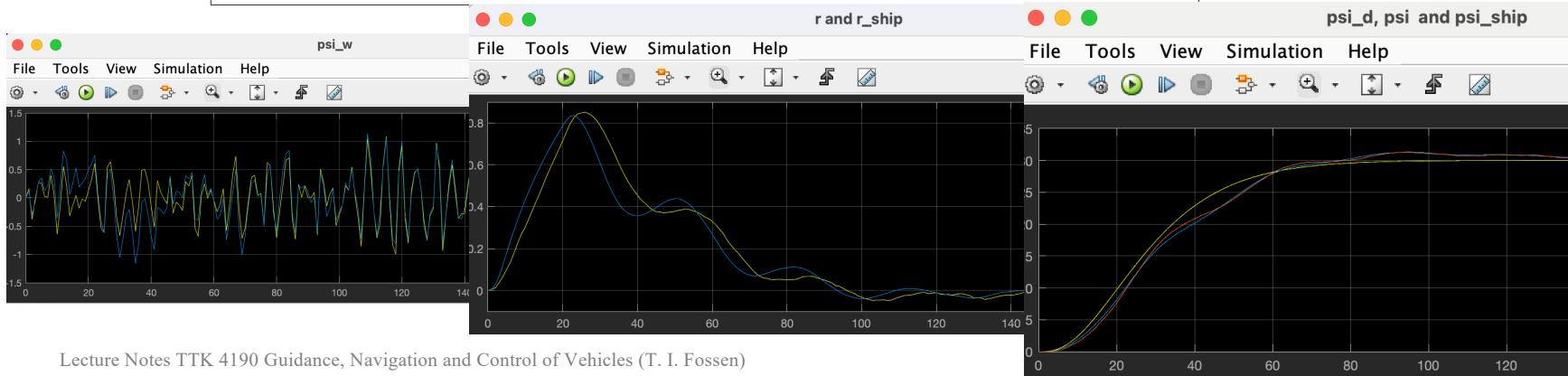
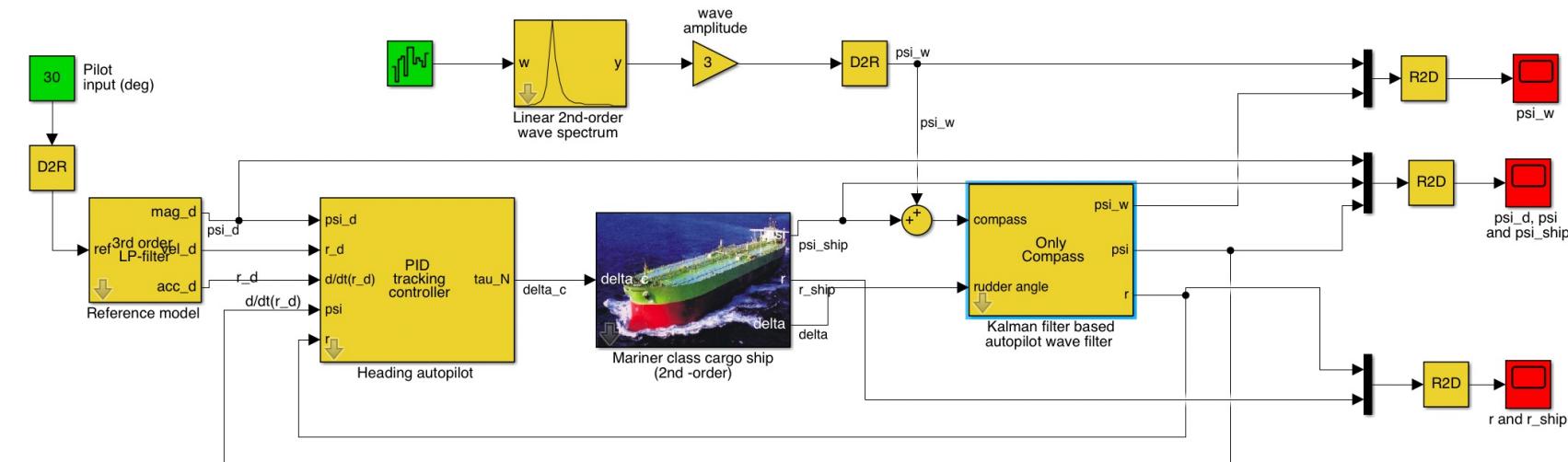


True LF heading Ψ and estimate.

True LF heading rate r and estimate.

True WF component of the heading Ψ_w and estimate.

MSS Simulink: demoKalmanWavefilterAutop.slx



13.4.6 Case Study: KF for Dynamic Positioning Systems

WF model

$$x_w^n(s) = \frac{K_{w_1}s}{s^2 + 2\lambda\omega_0s + \omega_0^2} w_1(s)$$

$$y_w^n(s) = \frac{K_{w_2}s}{s^2 + 2\lambda\omega_0s + \omega_0^2} w_2(s)$$

$$\psi_w(s) = \frac{K_{w_3}s}{s^2 + 2\lambda\omega_0s + \omega_0^2} w_3(s)$$

$$\dot{\xi} = A_w \xi + E_w w_1$$

$$\eta_w = C_w \xi$$

$$\eta = [x^n, y^n, \psi]^\top$$

$$\nu = [u, v, r]^\top$$

LTV state-space model (LF + WF)

$$\dot{\xi} = A_w \xi + E_w w_1$$

$$\dot{\eta} = R(t)\nu$$

$$\dot{b} = w_2$$

$$M\dot{\nu} = -D\nu + R^\top(t)b + \tau + \tau_{\text{wind}} + w_3$$

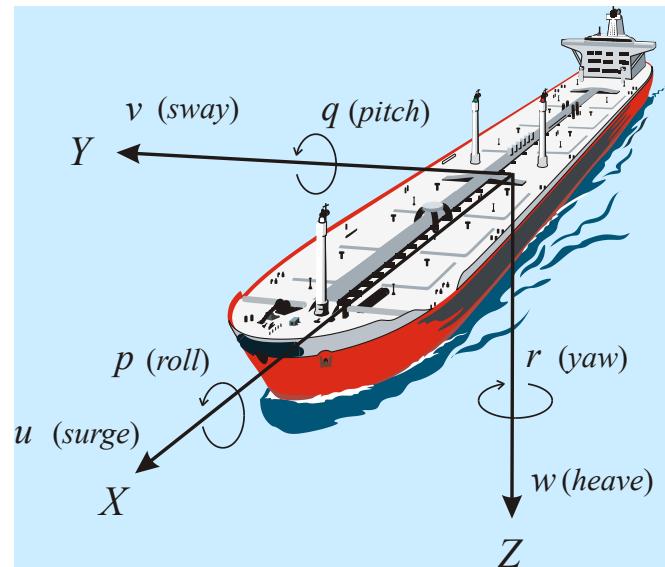
$$y = \eta + C_w \xi + \varepsilon$$

- Second-order wave drift forces
- Ocean currents
- Mean wind forces

Known R matrix

$$R(\psi(t)) := R(t)$$

$$\tau = -\hat{\tau}_{\text{wind}} + B_u u$$



13.4.6 Case Study: KF for Dynamic Positioning Systems

Model matrices

$$\mathbf{A}(t) = \left[\begin{array}{c|ccc} \mathbf{A}_w & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ \hline \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{R}(t) \\ \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} & \mathbf{M}^{-1} \mathbf{R}^\top(t) & -\mathbf{M}^{-1} \mathbf{D} \end{array} \right], \quad \mathbf{B} = \left[\begin{array}{c} \mathbf{0}_{6 \times p} \\ \hline \mathbf{0}_{3 \times p} \\ \mathbf{0}_{3 \times p} \\ \mathbf{M}^{-1} \mathbf{B}_u \end{array} \right]$$

$$\mathbf{E} = \left[\begin{array}{c|cc} \mathbf{E}_w & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ \hline \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{M}^{-1} \end{array} \right], \quad \mathbf{C} = [\mathbf{C}_w \mid \mathbf{I}_3 \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 3}]$$

Discrete-time state-space model (Euler's method)

$$\mathbf{x}[k+1] = \mathbf{A}_d[k] \mathbf{x}[k] + \mathbf{B}_d \mathbf{u}[k] + \mathbf{E}_d \mathbf{w}[k]$$

$$\mathbf{y}[k] = \mathbf{C}_d \mathbf{x}[k] + \boldsymbol{\varepsilon}[k]$$

$$\begin{aligned} \mathbf{A}_d[k] &\approx \mathbf{I}_{15} + h \mathbf{A}(t_k), & \mathbf{C}_d &= \mathbf{C} \\ \mathbf{B}_d &\approx h \mathbf{B}, & \mathbf{E}_d &\approx h \mathbf{E} \end{aligned}$$



13.4.6 Case Study: KF for Dynamic Positioning Systems

Matlab:

Pseudocode showing the computation of the Kalman gain and the corrector-predictor with ssa modification.

```
% KF gain: K[k]
K = P_prd * Cd' * inv( Cd * P_prd * Cd' + Rd );
IKC = eye(15) - K * Cd;

% Corrector: x_hat[k] and P_hat[k]
e = y - Cd * x_prd;
e(3) = ssa(e(3)); % estimation error
x_hat = x_prd + K * e; % ssa modification
P_hat = IKC * P_prd * IKC' + K * Rd * K';

% Predictor: x_prd[k+1] and P_prd[k+1]
x_prd = Ad * x_hat + Bd * u;
P_prd = Ad * P_hat * Ad' + Ed * Qd * Ed';
```

ssa modification for DP

Dead reckoning

Dead reckoning refers to the case where there are no updates, for instance GNSS position and/or compass signal losses for a period of time. During sensor failures, the best thing to do is to trust the model without any updates. Consequently, the corrector in the KF is bypassed by setting $K[k] = 0$ and prediction is based on the system model only. During dead reckoning (signal loss) the KF must be modified according to

Corrector:

$$\hat{x}[k] = \hat{x}^-[k]$$

$$\hat{P}[k] = \hat{P}^-[k]$$

Predictor:

$$\hat{x}^-[k+1] = A_d[k]\hat{x}[k] + B_d[k]u[k]$$

$$\hat{P}^-[k+1] = A_d[k]\hat{P}^-[k]A_d^\top[k] + E_d[k]Q_d[k]E_d^\top[k]$$

13.5 Passive Observer Design

The drawbacks of the Kalman filter for DP applications are:

- It is difficult and time-consuming to tune the KF (stochastic system with [15 states](#) and [120 covariance equations](#)).
The main reason for this is that the numerous covariance tuning parameters may be difficult to relate to physical quantities. This results in an ad hoc tuning procedure for the process covariance matrix **Q** while the measurement covariance matrix **R** usually is well defined in terms of sensor specifications. only local results
- Another drawback with KF-based design techniques is that a relatively large number of parameters must be determined through experimental testing of the craft

This motivated the research of a nonlinear passivity-based observer, since passivity arguments simplify the tuning procedure significantly (Fossen and Strand 1999). Hence, the time needed for sea trials and tuning can be drastically reduced. The nonlinear passive observer, guarantees convergence of all estimation errors (including the bias terms) to zero. Hence, only one set of observer gains is needed to cover the whole state space. In addition, the number of observer tuning parameters is significantly reduced and the wave filter parameters are directly coupled to the dominating wave frequency.

Reference:

Fossen, T. I. and J. P. Strand (1999). Passive Nonlinear Observer Design for Ships Using Lyapunov Methods: Experimental Results with a Supply Vessel, *Automatica*, Vol. 35, No. 1, pp. 3-16, January 1999.

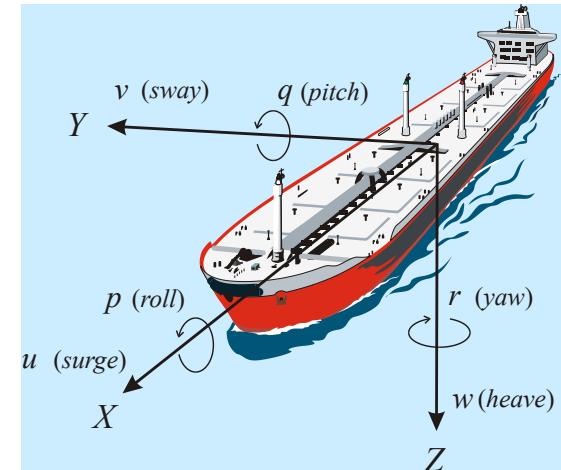
13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Assumption P1: $w = 0$ and $\varepsilon = 0$. The zero-mean Gaussian white noise terms are omitted in the deterministic stability analysis of the observer. If they are included in the Lyapunov function analysis the error dynamics will be uniformly ultimately bounded (UUB) instead of asymptotical/exponential stable.

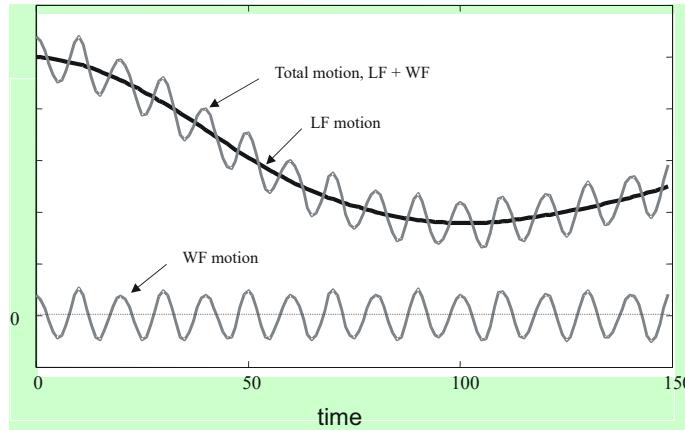
Assumption P2: $R(\psi(t)) := R(t)$ is known for all $t \geq 0$. This is a good assumption since the yaw angle is usually measured by a gyrocompass.

$$\begin{aligned}\dot{\xi} &= A_w \xi \\ \dot{\eta} &= R(t) \nu \\ \dot{b} &= -T^{-1} b \\ M \dot{\nu} &= -D \nu + R^\top(t) b + \tau + \tau_{\text{wind}} \\ y &= \eta + C_w \xi\end{aligned}$$

The matrix $T = \text{diag}\{T_1, T_2, T_3\} > 0$ of bias time constants is included as additional tuning parameters to obtain passivity



13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements



Measurement equation (GNSS + compass)

$$y = \eta + \eta_w + \varepsilon$$

← measurement noise

Low-frequency motion from the ship model

Wave frequency motion generated by a wave spectrum

PSD

frequency

Observer requirements:

- Observer must reconstruct η , η_w and v from y
- Only η and v are used for feedback

Position/heading (Earth)

$$\eta = [x, y, z]^T$$

Velocity (Body)

$$v = [u, v, r]^T$$

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

$$\begin{aligned}
 \dot{\hat{\xi}} &= A_w \hat{\xi} + \mathbf{K}_1(\omega_0) \tilde{y} \\
 \dot{\hat{\eta}} &= \mathbf{R}(t) \hat{\nu} + \mathbf{K}_2 \tilde{y} \\
 \dot{\hat{b}} &= -\mathbf{T}^{-1} \hat{b} + \mathbf{K}_3 \tilde{y} \\
 M \dot{\hat{\nu}} &= -\mathbf{D} \hat{\nu} + \mathbf{R}^\top(t) \hat{b} + \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \mathbf{R}^\top(t) \mathbf{K}_4 \tilde{y} \\
 \hat{y} &= \hat{\eta} + \mathbf{C}_w \hat{\xi}
 \end{aligned}$$

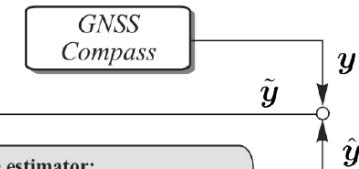
wave model

kinematics

bias

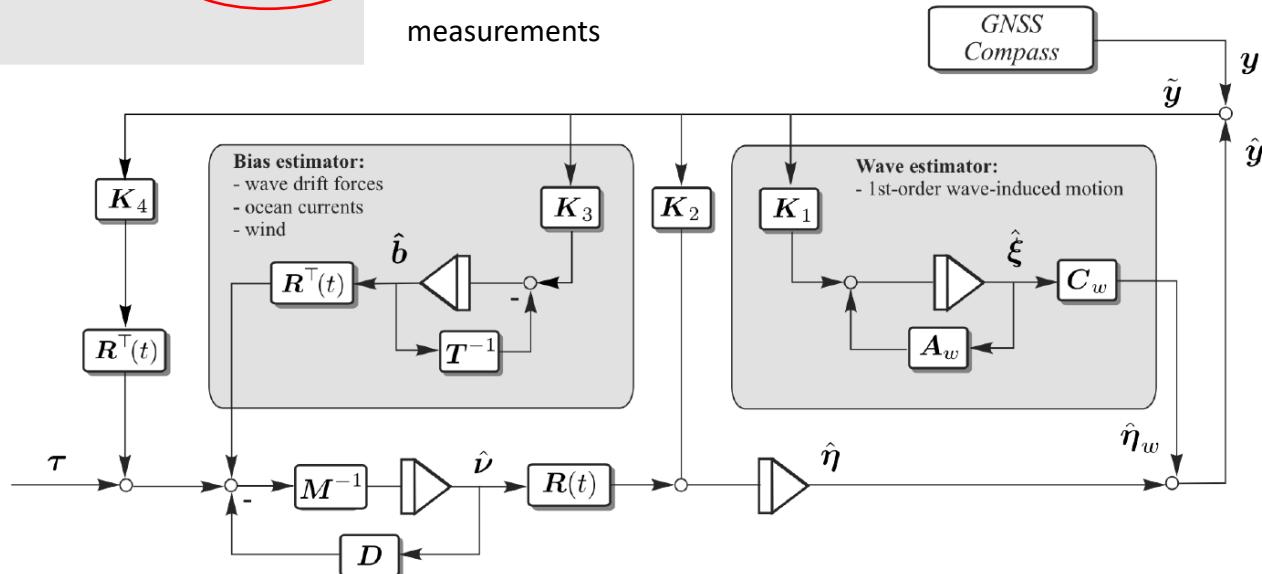
dynamics

measurements



Goal:

choose the *gains* \mathbf{K}_i
 such that the error
 dynamics is *passive* and
exponentially stable,



13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Observer error dynamics (position and WF models):

$$\begin{aligned}\dot{\hat{\eta}}_0 &= A_0 \hat{\eta}_0 + B_0 R(t) \hat{\nu} + K_0(\omega_0) \tilde{y} \\ \hat{y} &= C_0 \hat{\eta}_0\end{aligned}\quad K_0(\omega_0) = \begin{bmatrix} K_1(\omega_0) \\ K_2 \end{bmatrix} \quad \hat{\eta}_0 = [\hat{\xi}^\top, \hat{\eta}^\top]^\top$$

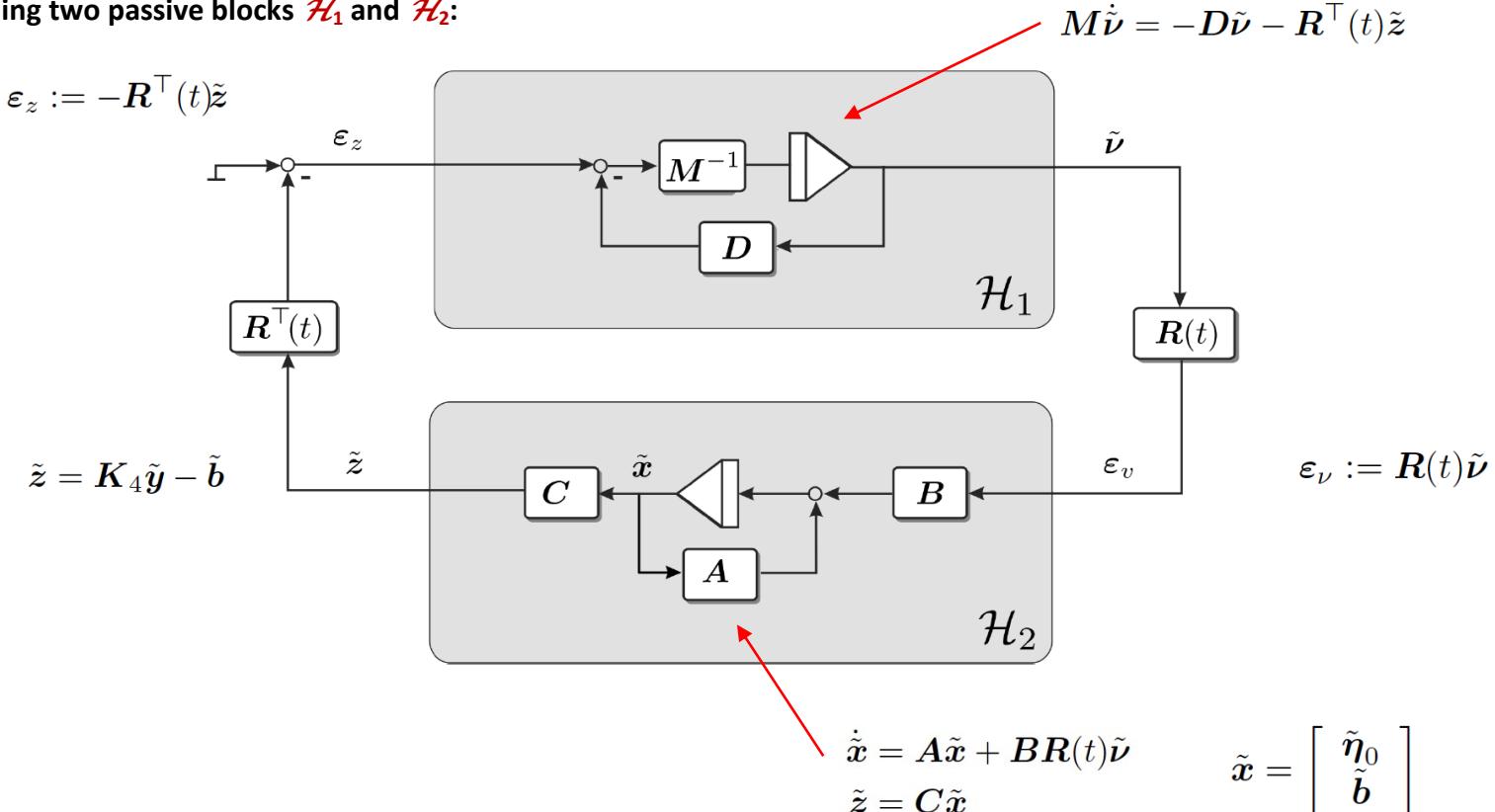
$$A_0 = \begin{bmatrix} A_w & \mathbf{0}_{6 \times 3} \\ \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad B_0 = \begin{bmatrix} \mathbf{0}_{6 \times 3} \\ I_3 \end{bmatrix}, \quad C_0 = [C_w \quad I_3]$$

Observer error dynamics including velocity/bias:

$$\begin{aligned}\dot{\tilde{\eta}}_0 &= (A_0 - K_0(\omega_0)C_0)\tilde{\eta}_0 + B_0 R(t) \tilde{\nu} \\ \dot{\tilde{b}} &= -T^{-1} \tilde{b} - K_3 \tilde{y} \\ M \dot{\tilde{\nu}} &= -D \tilde{\nu} + R^\top(t) \tilde{b} - R^\top(t) K_4 \tilde{y}\end{aligned}\quad \begin{aligned}\tilde{\nu} &= \nu - \hat{\nu} \\ \tilde{b} &= b - \hat{b} \\ \tilde{\eta}_0 &= \eta_0 - \hat{\eta}_0\end{aligned}$$

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Forming two passive blocks \mathcal{H}_1 and \mathcal{H}_2 :



13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Passivity is a property of engineering systems, most commonly used in electronic engineering and control systems.

A **passive component**, may be either a component that consumes (**but does not produce**) energy, or a component that is incapable of power gain. A component that is not passive is called an **active component**.

An electronic circuit consisting entirely of passive components is called a **passive circuit** (and has the same properties as a passive component).

A transfer functions $h(s)$ must **have phase greater than -90°** in order to be **passive**.

Passivity is related to stability and Lyapunov analysis can be used to prove passivity/stability in nonlinear systems while for linear systems the [Kalman-Yakubovich-Popov \(KYP\) Lemma](#) can be used to prove stability.

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Definition 6.3 (Khalil 2002) A nonlinear system is said to be **passive** if there exists a continuously differentiable positive definite function $V(x)$ (called storage function) such that:

$$u^T y \geq \dot{V}$$

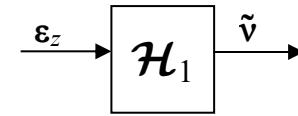
Moreover, it is said to be

- *Lossless if $u^T y = \dot{V}$*
- *Input-feedforward passive if $u^T y \geq \dot{V} + u^T \varphi(u)$ for some function $\varphi(u)$*
- *Input strictly passive if $u^T y \geq \dot{V} + u^T \varphi(u)$ and $u^T \varphi(u) > 0$, for all $u \neq 0$*
- *Output-feedback passive if $u^T y \geq \dot{V} + y^T \rho(y)$ for some function $\rho(y)$*
- *Output strictly passive if $u^T y \geq \dot{V} + y^T \rho(y)$ and $y^T \rho(y) > 0$, for all $y \neq 0$*
- *Strictly passive if $u^T y \geq \dot{V} + \psi(x)$ for some positive definite function $\psi(x)$*

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Proposition 13.1 (Strictly Passive Velocity Error Dynamics)

The mapping \mathcal{H}_1 is *strictly passive*.



Proof: Let, $S_1 = \frac{1}{2}\tilde{\nu}^\top M \tilde{\nu}$

be a positive definite storage function. Time differentiation of S_1 , yields:

$$\dot{S}_1 = -\frac{1}{2}\tilde{\nu}^\top (\mathbf{D} + \mathbf{D}^\top)\tilde{\nu} - \tilde{z}^\top \mathbf{R}(t)\tilde{\nu}$$

Using the fact that $\epsilon_z := -\mathbf{R}^\top(t)\tilde{z}$, yields

$$\epsilon_z^\top \tilde{\nu} = \dot{S}_1 + \frac{1}{2}\tilde{\nu}^\top (\mathbf{D} + \mathbf{D}^\top)\tilde{\nu}$$

This proves that \mathcal{H}_1 is *strictly passive*.

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Theorem 6.3 (Khalil 2002) *The feedback connection of two time-invariant dynamical systems is GAS if the origin of the nominal system ($u = 0$) is asymptotically stable and*

- *both feedback components are strictly passive*
- *both feedback components are output strictly passive and zero-state observable, or*
- *one component is strictly passive, and the other is output strictly passive and zero-state observable*

In addition, the storage function for each component must be radially unbounded

1. The mapping $\varepsilon_z \mapsto \tilde{v}$ is strictly passive (block \mathcal{H}_1)
 2. Post-multiplication with the bounded transformation matrix $R(t)$ and pre-multiplication by its transpose will not affect the passivity properties.
 3. Hence, it only remains to show that the mapping $\varepsilon_v \mapsto \tilde{z}$ (block \mathcal{H}_2) is strictly passive
-

For linear systems passivity can easily be checked by applying the Kalman-Yakubovich-Popov (KYP) Lemma.

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Lemma 13.1 (Kalman-Yakubovich-Popov)

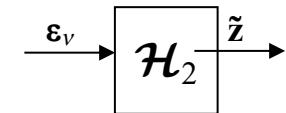
Let $H(s) = C(sI - A)^{-1}B$ be an $m \times m$ transfer function matrix, where A is Hurwitz, (A, B) is controllable, and (A, C) is observable. Then $Z(s)$ is strictly positive real (SPR) if and only if there exist positive definite matrices $P = P^T$ and $Q = Q^T$ such that

$$PA + A^T P = -Q$$

$$B^T P = C$$

Since \mathcal{H}_1 is strictly passive and \mathcal{H}_2 , given by three matrices (A, B, C) according to

$$\mathcal{H}_2 : \begin{cases} \dot{\tilde{x}} = A\tilde{x} + B\epsilon_\nu \\ \tilde{z} = C\tilde{x} \end{cases}$$



can be made SPR by choosing the gain matrices K_i ($i=1, \dots, 4$) according to the KYP Lemma. Hence, according to Lemma 6.4 (Khalil 2002), \mathcal{H}_2 is strictly passive since \mathcal{H}_2 is SPR



Interconnected system \mathcal{H}_1 and \mathcal{H}_2 is GAS

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Determination of the Observer Gains

The mapping \mathcal{H}_2 describes three decoupled systems in *surge*, *sway*, and *yaw*.

This suggests that the observer gain matrices should have a diagonal structure:

$$\mathbf{K}_1(\omega_0) = \begin{bmatrix} \text{diag}\{K_{11}(\omega_0), K_{12}(\omega_0), K_{13}(\omega_0)\} \\ \text{diag}\{K_{14}(\omega_0), K_{15}(\omega_0), K_{16}(\omega_0)\} \end{bmatrix}$$

$$\mathbf{K}_2 = \text{diag}\{K_{21}, K_{22}, K_{23}\}$$

$$\mathbf{K}_3 = \text{diag}\{K_{31}, K_{32}, K_{33}\}$$

$$\mathbf{K}_4 = \text{diag}\{K_{41}, K_{42}, K_{43}\}$$



function of the wave frequency.

Opens for

- gain scheduling
- adaptive observer design

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Three decoupled transfer functions

$$\begin{aligned}\tilde{z}(s) &= \mathbf{H}(s)\boldsymbol{\varepsilon}_\nu(s) \\ &:= \mathbf{H}_0(s)\mathbf{H}_B(s)\boldsymbol{\varepsilon}_\nu(s)\end{aligned}$$

$$\begin{aligned}\mathbf{H}_0(s) &= \mathbf{C}_0(s\mathbf{I}_3 + \mathbf{A}_0 - \mathbf{K}_0(\omega_0)\mathbf{C}_0)^{-1}\mathbf{B}_0 \\ \mathbf{H}_B(s) &= \mathbf{K}_4 + (s\mathbf{I}_3 + \mathbf{T}^{-1})^{-1}\mathbf{K}_3\end{aligned}$$

\mathbf{H}_0 is determined by using pole placement

$$h_{0i}(s) = \frac{s^2 + 2\lambda\omega_0s + \omega_0^2}{s^3 + (K_{1(i+3)} + K_{2i} + 2\lambda\omega_0)s^2 + (\omega_0^2 + 2\lambda\omega_0K_{2i} - K_{1i}\omega_0^2)s + K_{2i}\omega_0^2}$$

$$h_{di}(s) = \frac{s^2 + 2\lambda\omega_0s + \omega_0^2}{(s^2 + 2\zeta_{ni}\omega_0s + \omega_0^2)(s + \omega_{ci})}$$



$$\begin{aligned}K_{1i}(\omega_0) &= -2(\zeta_{ni} - \lambda)\frac{\omega_{ci}}{\omega_0} \\ K_{1(i+3)}(\omega_0) &= 2\omega_0(\zeta_{ni} - \lambda) \\ K_{2i} &= \omega_{ci}\end{aligned}$$

The desired transfer function is low-pass + notch

13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

The remaining gains K_3 and K_4 in \mathbf{H}_B is found by frequency shaping. The transfer functions $h_i(s)$ must all have phase greater than -90° in order to be **passive**.

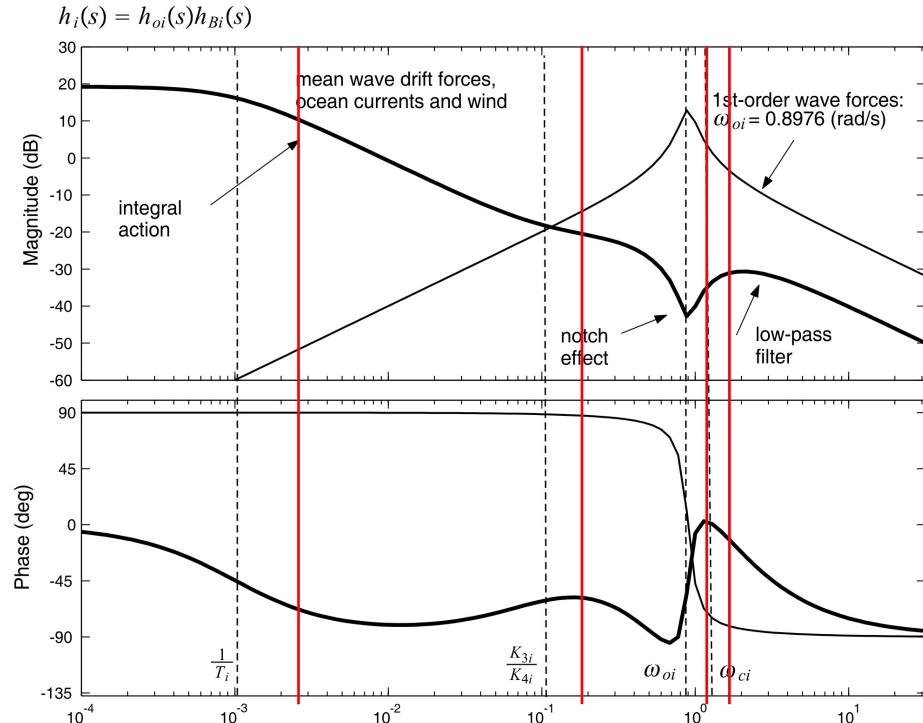
This is satisfied for

$$1/T_i \ll K_{3i}/K_{4i} < \omega_0 < \omega_{ci} \quad (i = 1, \dots, 3)$$

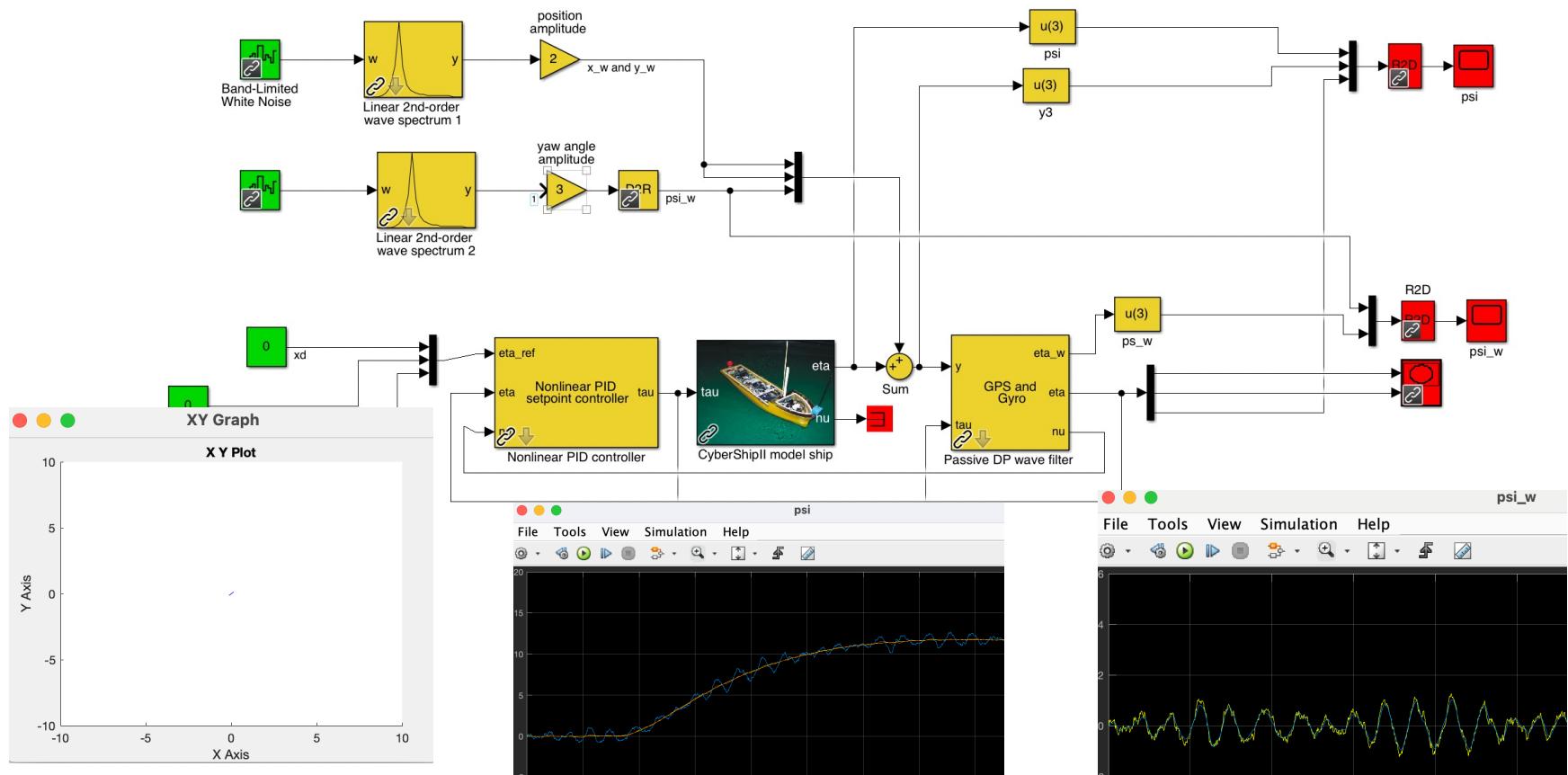
$$\mathbf{H}(s) = \mathbf{H}_0(s)\mathbf{H}_B(s)$$

$$h_{Bi}(s) = K_{4i} \frac{s + \left(\frac{1}{T_i} + \frac{K_{3i}}{K_{4i}}\right)}{s + \frac{1}{T_i}} \approx K_{4i} \frac{s + \frac{K_{3i}}{K_{4i}}}{s + \frac{1}{T_i}}$$

$$h_{di}(s) = \frac{s^2 + 2\lambda_i \omega_{oi} s + \omega_{oi}^2}{(s^2 + 2\zeta_{ni} \omega_{oi} s + \omega_{oi}^2)(s + \omega_{ci})}$$

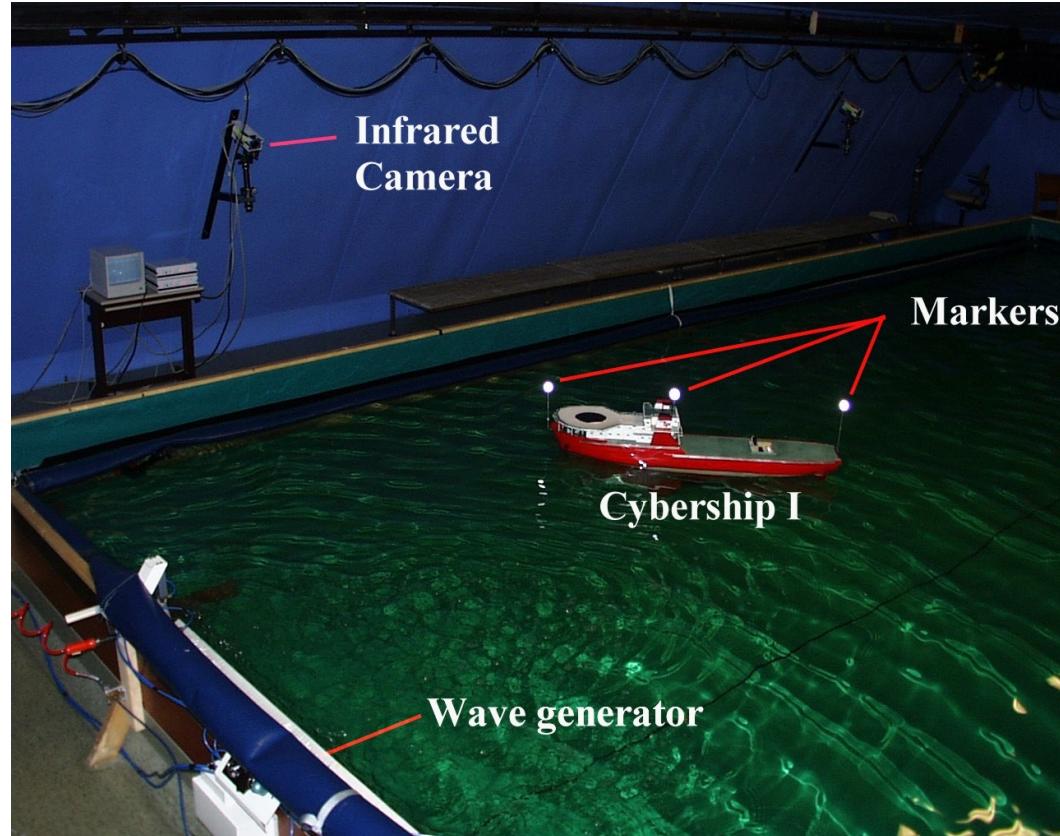


MSS Simulink: demoCS2passiveObserverDP.slx



13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

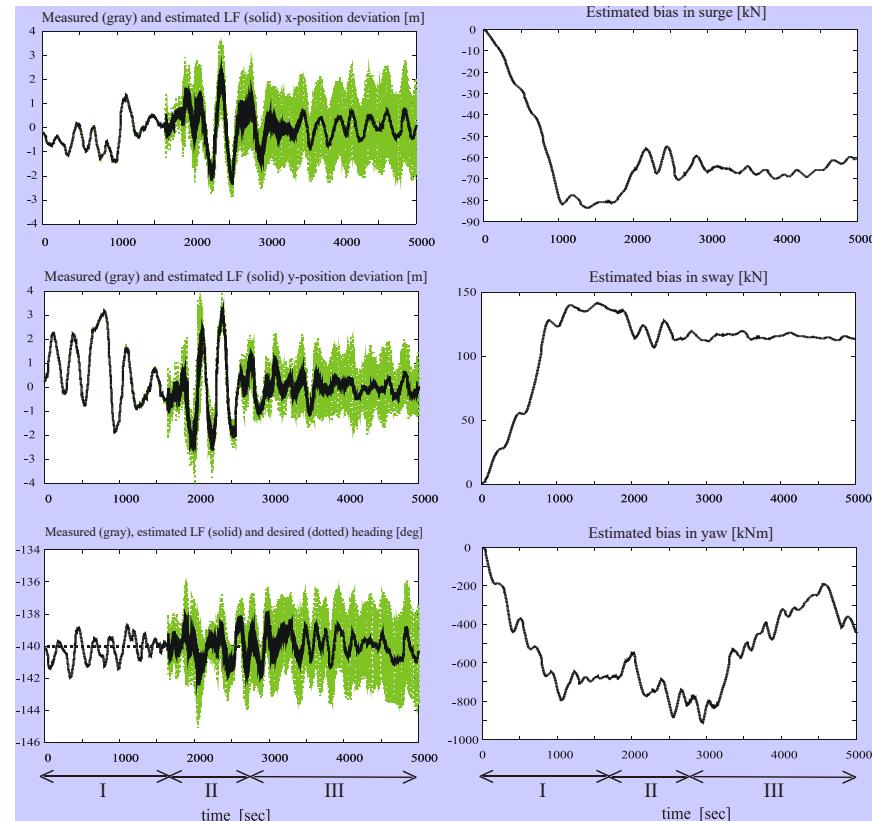
The first experiments were performed in the [GNC laboratory](#) at the Department of Engineering Cybernetics, NTNU using [CyberShip I](#) which is offshore supply vessel scale 1:70.



13.5.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

Experimental results: implemented and tested onboard several ships and rigs offshore.

Reduced commissioning time: easy to tune compared to the Extended Kalman Filter.



13.5.2 Case study: Passive Observer for Heading Autopilots Using only Compass Measurements

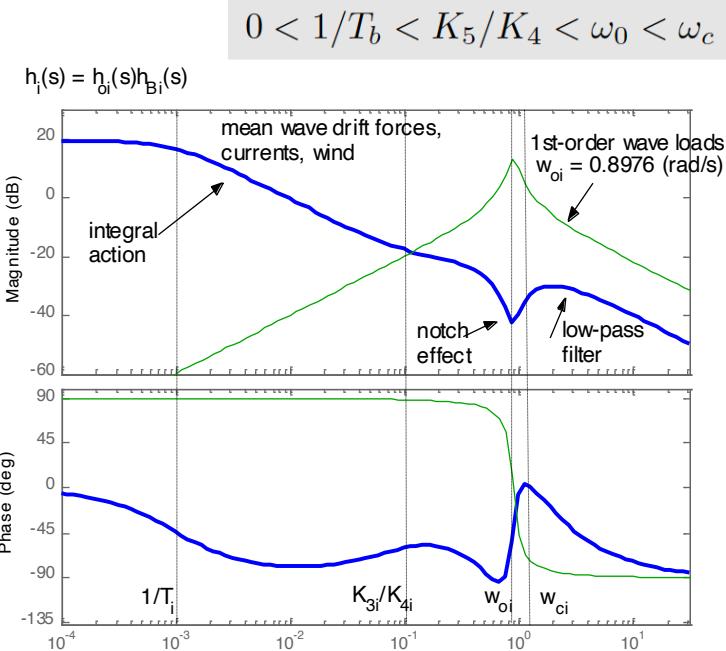
Passivity-Based Pole Placement

The observer error dynamics can be reformulated as two subsystems for yaw angle/rudder bias, and yaw rate. Fossen and Strand (1999) have shown that these systems forms a *passive interconnection* if the observer gains are chosen according to

$$\mathbf{k} = \begin{bmatrix} -2(1-\lambda)\frac{\omega_c}{\omega_0} \\ 2\omega_0(1-\lambda) \\ \omega_c \\ K_4 \\ K_5 \end{bmatrix}$$

where $\omega_c > \omega_0$ is the filter cut-off frequency

$$\begin{aligned} \dot{\hat{\xi}}_w &= \hat{\psi}_w + K_1 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w) \\ \dot{\hat{\psi}}_w &= -\omega_0^2 \hat{\xi}_w - 2\lambda\omega_0 \hat{\psi}_w + K_2 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w) \\ \dot{\hat{\psi}} &= \hat{r} + K_3 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w) \\ \dot{\hat{r}} &= -\frac{1}{T} \hat{r} + \frac{1}{m}(\tau_{\text{wind}} + \tau_N) + \hat{b} + K_4 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w) \\ \dot{\hat{b}} &= -\frac{1}{T_b} \hat{b} + K_5 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w) \end{aligned}$$



13.5.2 Case study: Passive Observer for Heading Autopilots Using only Compass Measurements

Example 13.6 (Passive Wave Filtering)

Consider the *Mariner class cargo ship* with $K = 0.185 \text{ s}^{-1}$ and $T = T_1 + T_2 - T_3 = 107.3 \text{ s}$ (Strøm-Tejsen 1965). The bias time constant is chosen to be rather large, that is $T_b = 100 \text{ s}$. The wave response model is modeled by a linear approximation to the JONSWAP spectrum with $\lambda = 0.1$ and $\omega_0 = 1.2 \text{ rad/s}$.

State-space model

wave-frequency
model

$$\mathbf{A} = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ -1.44 & -0.24 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -0.0093 & 1 \\ 0 & 0 & 0 & 0 & -0.01 \end{array} \right], \quad \mathbf{b} = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0.0017 \end{array} \right]$$

ship + bias models

$$\mathbf{E} = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ 0.24 \sigma & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right], \quad \mathbf{c}^\top = [0, 1, 1, 0, 0]$$

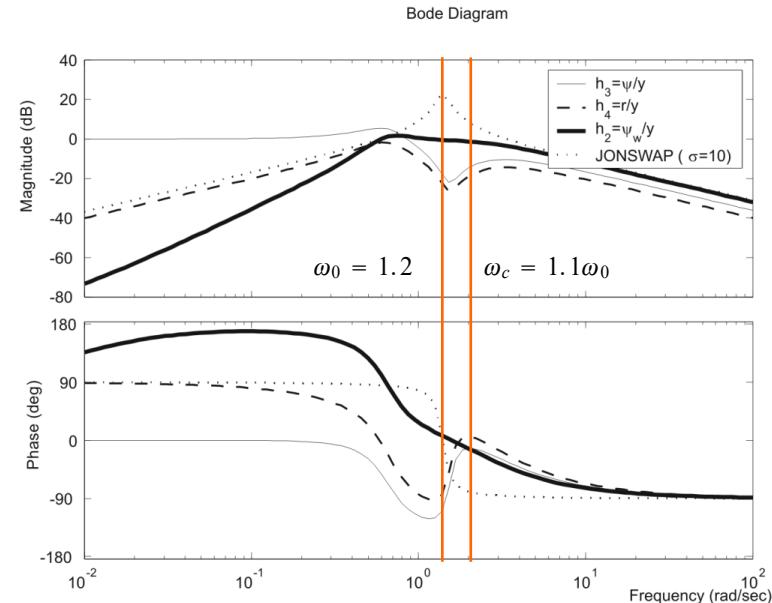
13.5.2 Case study: Passive Observer for Heading Autopilots Using only Compass Measurements

Example 13.6 (Passive Wave Filtering, cont.)

Using passivity as a tool for filter design with cut-off frequency $\omega_c = 1.1\omega_0$, yields:

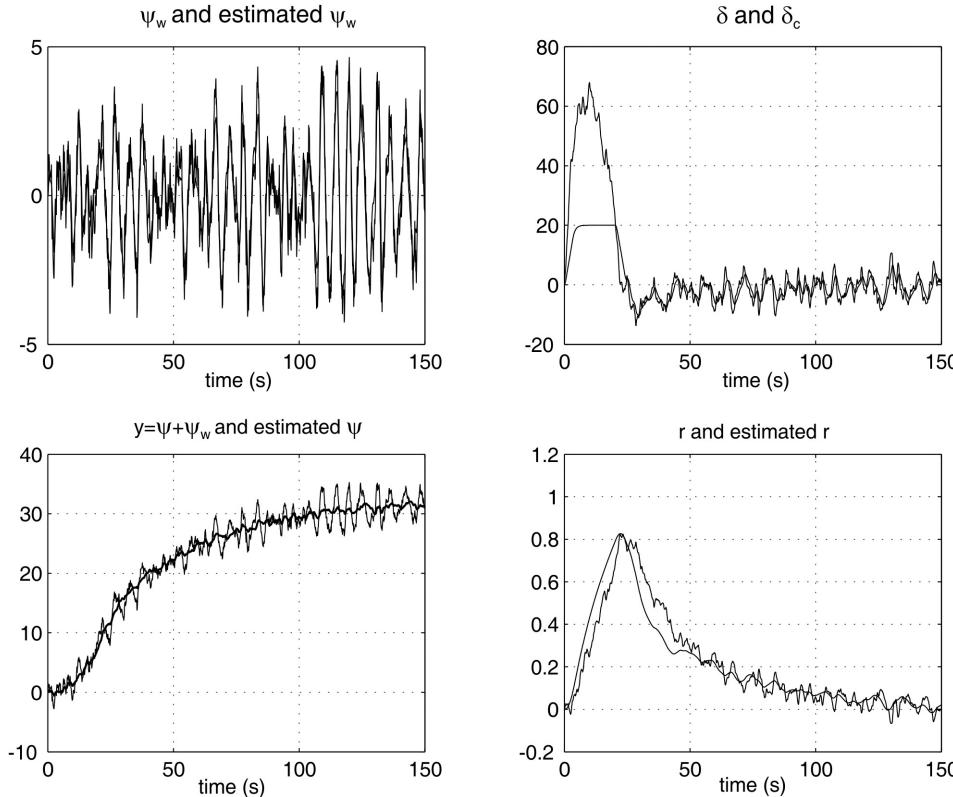
$$\mathbf{k} = \begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \\ K_5 \end{bmatrix} = \begin{bmatrix} -2(1-\lambda)\frac{\omega_c}{\omega_0} \\ 2\omega_0(1-\lambda) \\ \omega_c \\ K_4 \\ K_5 \end{bmatrix} = \begin{bmatrix} -1.98 \\ 1.80\omega_0 \\ 1.10\omega_0 \\ K_4 \\ K_5 \end{bmatrix}$$

Note that the notch effect at ω_0 is more than -20 dB for $h_3(s)$ and $h_4(s)$ representing the state estimates $\hat{\psi}$ and \hat{r} . We also see that high-frequency motion components above ω_c is low-pass filtered. Finally, the transfer function $h_2(s)$ representing reconstruction of the WF motion $\hat{\psi}_w$ filters out signals on the outside of the wave response spectrum.



Bode plot showing the wave filter transfer functions and the JONSWAP spectrum.

13.5.2 Case study: Passive Observer for Heading Autopilots Using only Compass Measurements



13.5.3 Case study: Passive Observer for Heading Autopilots Using both Compass and Angular Rate Sensor Measurements

It is advantageous to **integrate gyro and compass measurements** in the observer. This results in less variance and better accuracy of the state estimates. One simple way to do this is to treat the *gyro measurements* as an *input* to the system model

$$\dot{\psi} = u_{\text{ars}} + b_{\text{ars}}$$

where b_{ars} denotes the gyro bias and u_{ars} is the ARS measurement.

This model will give proper wave filtering of the state ψ . However, the estimate of r is not wave filtered, since this signal is taken directly from the gyro measurement u_{gyro} . This can be solved by filtering u_{gyro} with a notch filter $h_{\text{notch}}(s)$ and a low-pass filter to the cost of some phase lag

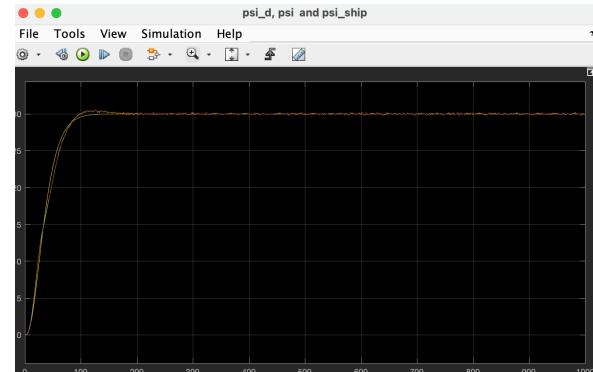
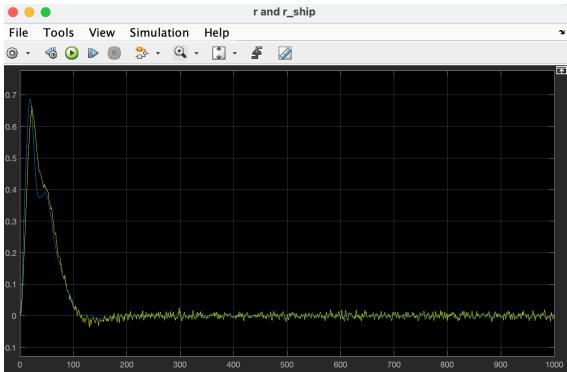
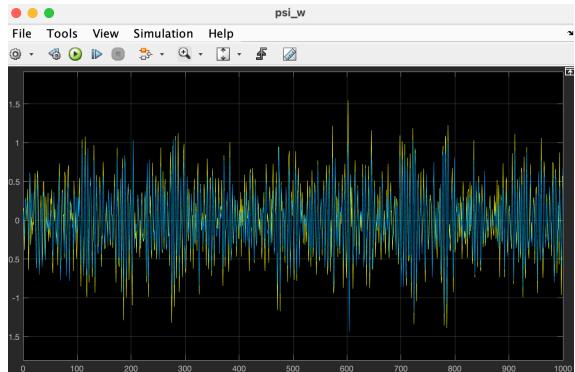
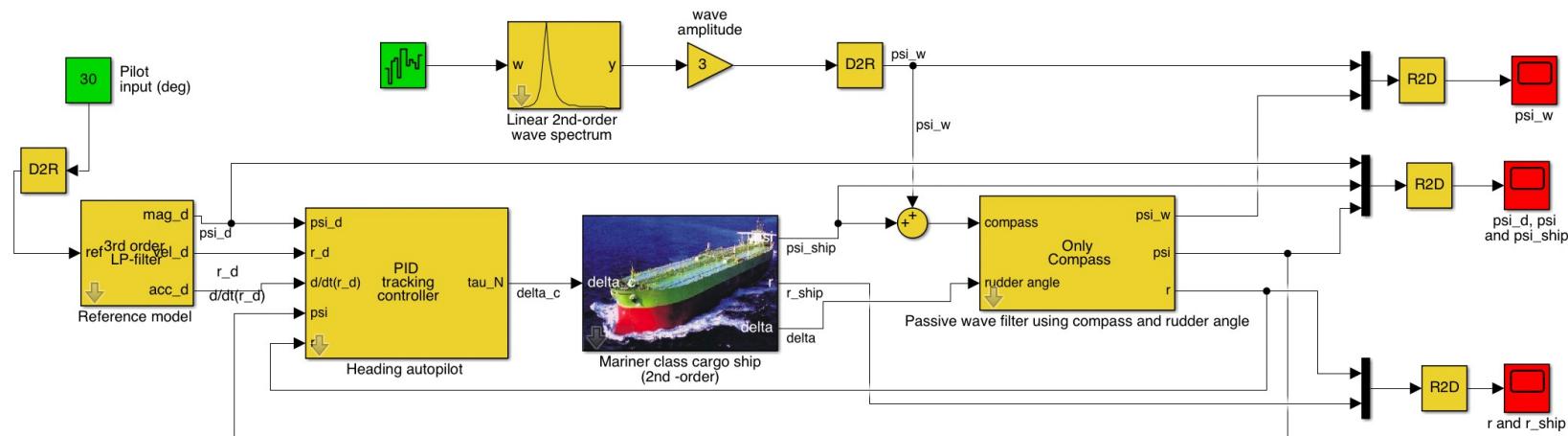
$$\begin{aligned}\dot{\hat{\xi}}_w &= \hat{\psi}_w + K_1 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w) \\ \dot{\hat{\psi}}_w &= -\omega_0^2 \hat{\xi}_w - 2\lambda\omega_0 \hat{\psi}_w + K_2 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w) \\ \dot{\hat{\psi}} &= u_f + \hat{b}_{\text{ars}} + K_3 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w) \\ \dot{\hat{b}}_{\text{ars}} &= -\frac{1}{T_b} \hat{b}_{\text{ars}} + K_4 \text{ssa}(y - \hat{\psi} - \hat{\psi}_w)\end{aligned}$$

$$T_b \gg 0$$

$$u_f = h_{\text{notch}}(s) h_{\text{lp}}(s) u_{\text{ars}}$$

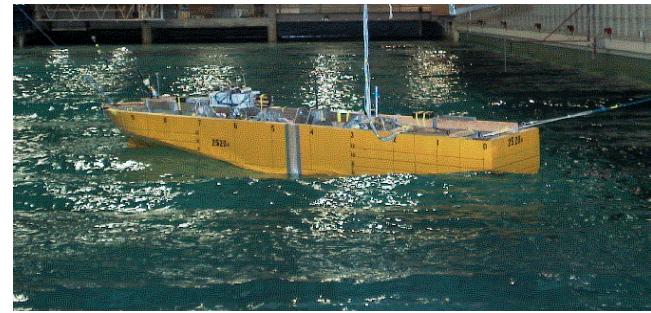
$$\mathbf{k} = \begin{bmatrix} -2(1-\lambda)\frac{\omega_c}{\omega_0} \\ 2\omega_0(1-\lambda) \\ \omega_c \\ K_4 \\ K_5 \end{bmatrix}$$

MSS Simulink: demoPassiveWavefilterAutopilot1.slx



13.5.3 Case study: Passive Observer for Heading Autopilots Using both Compass and Angular Rate Sensor Measurements

The wave filter has been tested on a scale model of *MV Autoprestige* of the United European Car Carriers (UECC)



The maneuvering test were performed in the Ocean Basin at MARINTEK in Trondheim April 2001.

13.5.3 Case study: Passive Observer for Heading Autopilots Using both Compass and Angular Rate Sensor Measurements

It is seen that the WF motion components are quite well removed from the estimate of resulting in good course-keeping capabilities.

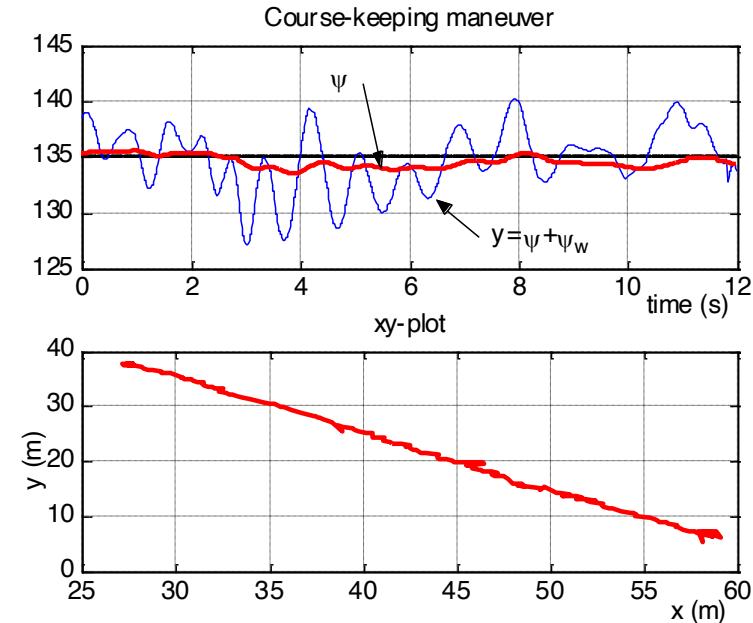
We also notice that the estimate of ψ_w is quite good, while r could be slightly improved by changing the observer gains.



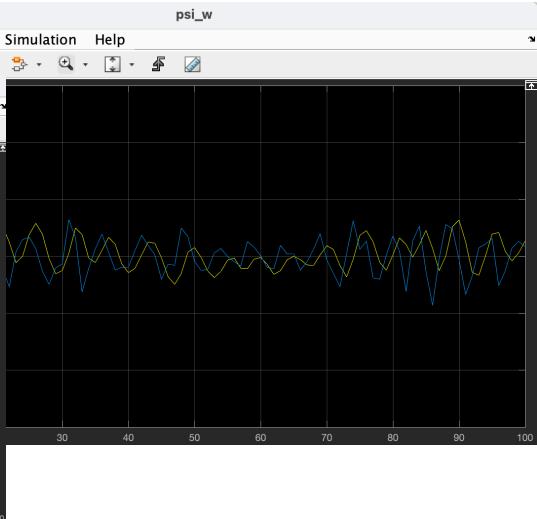
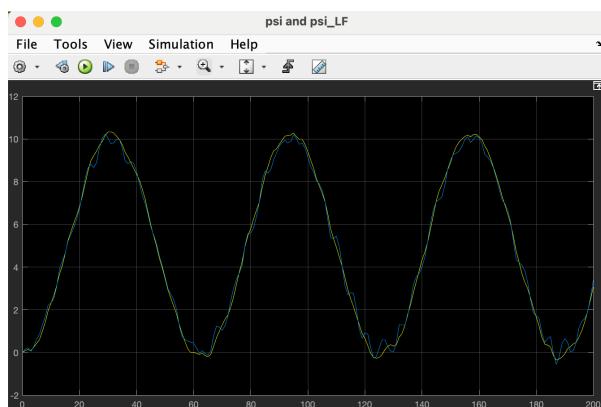
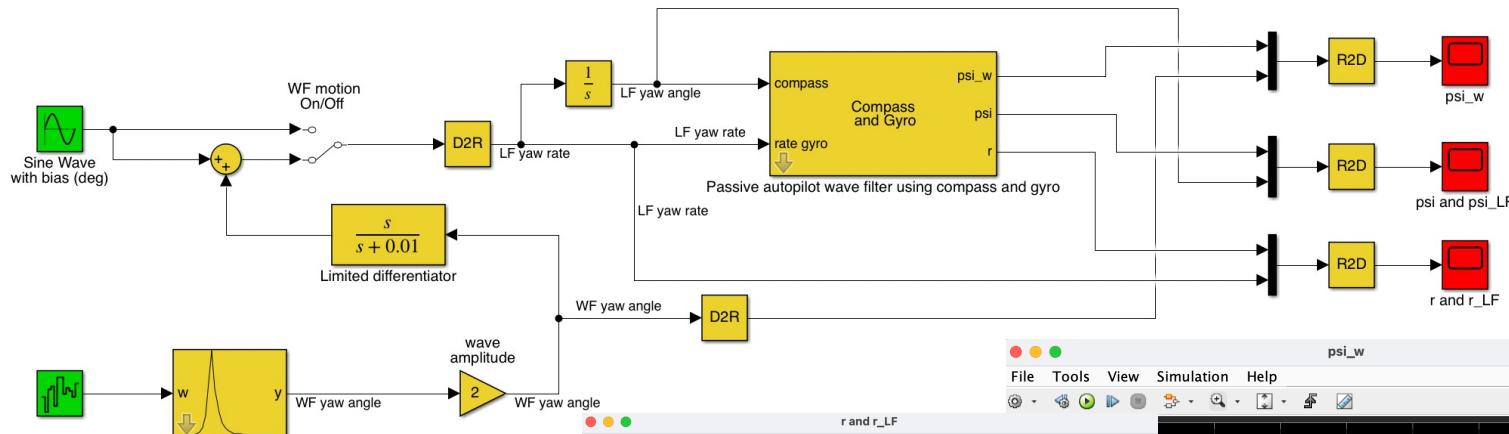
Significant wave height: $H_s = 1.3 \text{ m}$ (full scale)

Frequency of encounter: $\omega_e = 1.07 \text{ rad/s}$

Cruise speed: $U = 2.3 \text{ m/s}$ (model scale)



MSS Simulink: demoPassiveWavefilterAutopilot2.slx



Chapter Goals – Revisited

Sensors for marine craft:

- Understand the principles for GNSS position, GNSS heading, magnetic compass and gyrocompass
- Understand what we mean with **wave filtering** and when to apply a wave filter algorithm
- Be able to **estimate the wave encounter frequency** of a marine craft

Model-based state estimation:

- Understand the principles and design methods for fixed-gain Luenberger observers, Kalman filters and passive observers
- Be able to model marine craft under DP and heading control, and include **dynamic models of the sensor and navigation systems using realistic measurement noise**
- Be able to design **Kalman filters** for DP and heading autopilots with wave filtering capabilities
- Be able to design **passive observers** for DP and heading autopilots with wave filtering capabilities