Chapter 16 – Advance Motion Control Systems

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{w}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\theta(\boldsymbol{\eta})\boldsymbol{\nu}$$
$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{w}$$

State-of-the-art motion control systems are usually designed using PID control methods as described in Chapter 15. This chapter presents more advanced methods for optimal and nonlinear control of marine craft. The main motivation for this is design simplicity and performance.

Nonlinear control theory can often yield a more intuitive design than linear theory. Linearization destroys model properties, and the results can be a more complicated design process with limited physical insight.

Chapter 16 is written for the advanced user who want to exploit a more advanced model and use this model to improve the performance of the control system. Readers of this chapter need background in optimal and nonlinear control theory.

Chapter Goals

Linear-quadratic (LQ) optimal control

- Be able to modify linear-quadratic controllers to include integral action
- Be able to extend linear-quadratic optimal controllers from setpoint regulation to trajectory-tracking control
- Be able to design linear-quadratic optimal controllers for heading control, rudder-roll-damping, and DP
- Understand how to use the ISO standards for motion sickness when designing roll-damping control systems

Feedback linearization

- Be able to design to design velocity and position trajectory-tracking controllers for marine craft by feedback linearization and pole placement.

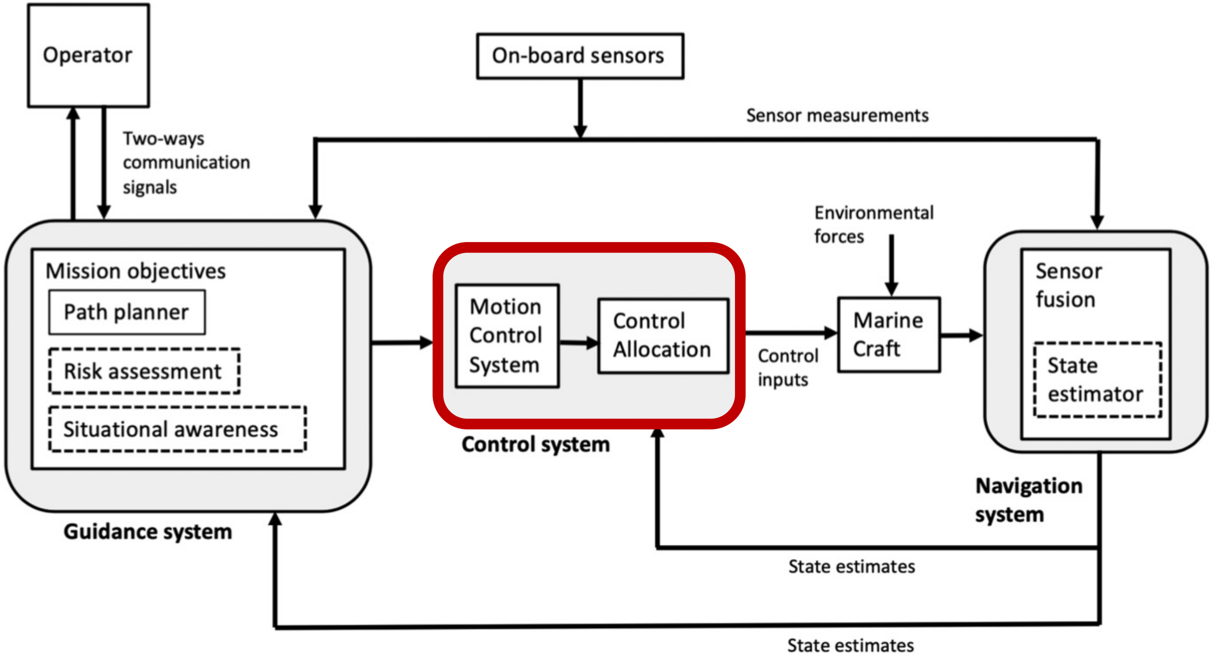
Nonlinear backstepping

- Understand how nonlinear backstepping relates to feedback linearization. Pros and cons.
- Be able to design nonlinear backstepping controllers for heading control, speed control, DP, and other motion control scenarios.

Sliding-mode control

- Understand the difference of conventional sliding-mode control (SMC), integral sliding-mode control (ISMC), and adaptive super-twisting SMC
- Be able to design motion controllers for marine craft using ISMC and adaptive super-twisting SMC

Chapter 16 – Advance Motion Control Systems



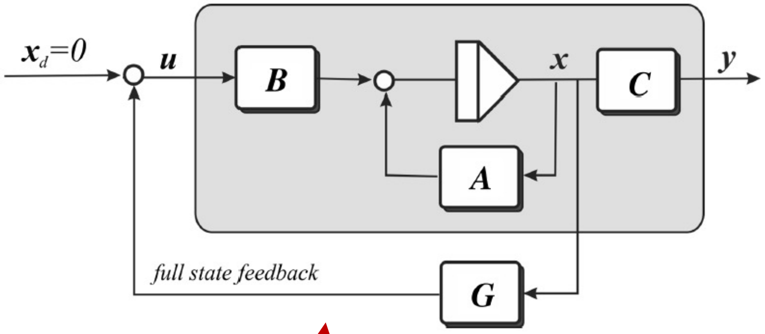
Control, or more specifically *motion control* and control allocation, is the action of determining the necessary control forces and moments to be provided by the craft in order to satisfy a certain control objective.

Control allocation: Distribution of generalized control forces to the actuators (see Section 11.2)

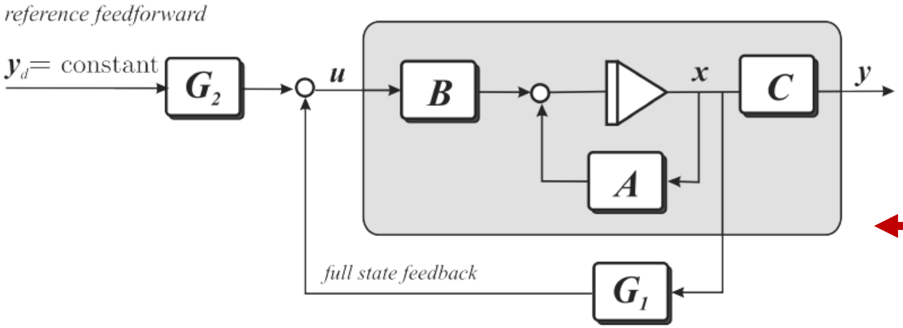
Chapter 16.1 Linear-Quadratic Optimal Control

Optimal motion control systems are designed by considering the linearized equations of motion in the following form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{w}$$



Optimal setpoint regulation



Optimal trajectory-tracking control

Chapter 16.1 Linear-Quadratic Optimal Control

A linear-quadratic regulator (LQR) can be designed for the state-space model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

$$\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{u} \in \mathbb{R}^r \text{ and } \boldsymbol{y} \in \mathbb{R}^m$$

In order to design a linear optimal control law the system (**A**, **B**, **C**) must be controllable.

Definition 16.1 (Controllability)

The state and input matrix (**A**, **B**) must satisfy the controllability condition to ensure that there exists a control **u**(t) which can drive any arbitrary state **x**(t_0) to another arbitrary state **x**(t_1) for $t_1 > t_0$. The controllability condition requires that the matrix

$$\mathcal{C} = [\boldsymbol{B} \mid \boldsymbol{A}\boldsymbol{B} \mid \cdots \mid (\boldsymbol{A})^{n-1}\boldsymbol{B}]$$

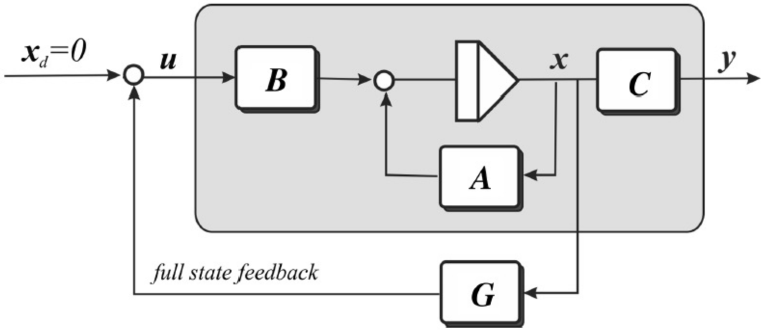
must be of full row rank such that a least a right inverse exists.

16.1.1 Linear-Quadratic Regulator

Design a linear optimal controller for setpoint regulation

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

regulate **y = Cx** to zero



Quadratic cost function

$$J = \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \int_0^T \left(\boldsymbol{y}^\top \boldsymbol{Q} \boldsymbol{y} + \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u} \right) \mathrm{d}t \right.$$
$$\left. = \frac{1}{2} \int_0^T \left(\boldsymbol{x}^\top \boldsymbol{C}^\top \boldsymbol{Q} \boldsymbol{C} \boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u} \right) \mathrm{d}t \right\}$$

Design weights

$\mathbf{Q} = \mathbf{Q}^T \geq 0$ (output weight)
$\mathbf{R} = \mathbf{R}^T > 0$ (input weight)

Optimal solution (Athans and Falb 1966)

$$\boldsymbol{u} = \underbrace{-\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}_\infty}_{\boldsymbol{G}} \boldsymbol{x}$$

Linear Quadratic Regulator

$$\boldsymbol{P}_\infty \boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{P}_\infty - \boldsymbol{P}_\infty \boldsymbol{B} \boldsymbol{R}^{-1} \boldsymbol{B}^\top \boldsymbol{P}_\infty + \boldsymbol{C}^\top \boldsymbol{Q} \boldsymbol{C} = \boldsymbol{0}$$

Algebraic Riccati Equation (ARE) - solving for $\mathbf{P} = \mathbf{P}^T > 0$

16.1.1 Linear-Quadratic Regulator

Matlab Example: MSS Toolbox script ExLQR.m

```
% Weights
Q = diag([1]);        % tracking error weights (dim m x m)
R = diag([1]);        % input weights (dim r x r)

% System matrices
A = [0 1; -1 -2];     % state matrix (dim n x n)
B = [0; 1];           % input matrix (dim n x r)
C = [1 0];            % output matrix (dim m x n)

% Compute the optimal feedback gain matrix G
[K,P,E] = lqr(A, B, C'*Q*C, R);
G = -K
```

The Matlab function `lqr.m` also returns the eigenvalues of the closed-loop system

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{G})\boldsymbol{x}$$

denoted by the symbol `E`.

16.1.2 LQR Design for Trajectory Tracking and Integral Action

Transformation of the LQ Tracker to a Set-Point Regulation Problem

The LQR can be redesigned to track a time-varying reference trajectory $\mathbf{x}_d$ for a large class of mechanical systems possessing certain structural properties

Two solutions will be presented:

- Simple solution
  Transformation of the trajectory tracking problem to a LQ setpoint regulation problem using reference feedforward. Hence, the solution can be found by solving the *Algebraic Riccati Equation* (ARE)

- General solution
  Compute the optimal reference and disturbance feedforward gains by solving a linear quadratic performance index. This solution requires that a set of *Differential Riccati Equations (DRE)* are solved.

16.1.2 LQR Design for Trajectory Tracking and Integral Action

Example 16.1 (Mass—Damper—Spring Trajectory-Tracking Problem)

x v $\quad$ FF LQ $\quad$ LQ feedback + reference feedforward

mv dv kx $\quad$ FF mv_d dv_d kx_d

Original model is transformed to an Error Dynamics Model

$m\ddot{e}$ de ke LQ $\quad$ e x x_d, $\dot{e}$ v v_d

The desired states are computed using a reference model with **r** as input.

The trajectory-tracking control problem has now been transformed to a LQ setpoint regulation problem

$$\dot{\boldsymbol{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{\boldsymbol{B}} u \qquad \boldsymbol{x} = [e,\ \dot{e}]^{\top}$$

$$e = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\boldsymbol{C}} \boldsymbol{x}$$

Solve this as a standard LQR problem where $u = \tau_{LQ}$

16.1.2 LQR Design for Trajectory Tracking and Integral Action

Integral Action

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \qquad \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

An integral state **z**

$$\dot{\boldsymbol{z}} = \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

is augmented to the state vector **x** such that

$$\dot{\boldsymbol{x}}_a = \boldsymbol{A}_a \boldsymbol{x}_a + \boldsymbol{B}_a \boldsymbol{u}$$

$$\boldsymbol{A}_a = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{A} \end{bmatrix}, \quad \boldsymbol{B}_a = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{B} \end{bmatrix}$$

$$\boldsymbol{x}_a = [\boldsymbol{z}^\top, \boldsymbol{x}^\top]^\top$$

Hence, the solution of the LQR setpoint regulation problem for the augmented state-space model is

$$J = \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \int_0^t \left(\boldsymbol{x}_a^\top \boldsymbol{Q}_a \boldsymbol{x}_a + \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u} \right) \mathrm{d}\tau \right\}$$

$$\boxed{\begin{aligned} \boldsymbol{u} &= -\boldsymbol{R}^{-1} \boldsymbol{B}_a^\top \boldsymbol{P}_\infty \boldsymbol{x}_a \\ &= -\boldsymbol{R}^{-1} [\boldsymbol{0} \quad \boldsymbol{B}^\top] \begin{bmatrix} \boldsymbol{P}_{11} & \boldsymbol{P}_{12} \\ \boldsymbol{P}_{21} & \boldsymbol{P}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{x} \end{bmatrix} \\ &= -\underbrace{\boldsymbol{R}^{-1} \boldsymbol{B}^\top \boldsymbol{P}_{21}}_{\boldsymbol{K}_i} \boldsymbol{z} - \underbrace{\boldsymbol{R}^{-1} \boldsymbol{B}^\top \boldsymbol{P}_{22}}_{\boldsymbol{K}_p} \boldsymbol{x} \end{aligned}}$$

$$\boxed{\boldsymbol{P}_\infty \boldsymbol{A}_a + \boldsymbol{A}_a^\top \boldsymbol{P}_\infty - \boldsymbol{P}_\infty \boldsymbol{B}_a \boldsymbol{R}^{-1} \boldsymbol{B}_a^\top \boldsymbol{P}_\infty + \boldsymbol{Q}_a = \boldsymbol{0}}$$

Optimal LQ controller with integral action (feedback from the integral state **z**)

16.1.3 General Solution of the LQ Trajectory-Tracking Problem

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

Model Assumption: the state **x** is measured

If **x** is estimated using a *Kalman filter*, stability can be proven by applying a separation principle (*LQG control*)

Control Objective

Design a LQ optimal tracking controller using a time-varying reference trajectory $\mathbf{y}_d$

Assume that the desired output $\mathbf{y}_d = \mathbf{C}\mathbf{x}_d$ is known for time $t \in [0, T]$ where T is the final time and $\mathbf{x}_d$ is the desired state vector. The goal is to design an optimal tracking controller that tracks the desired output

$$J = \min_{\boldsymbol{u}} \left\{ \frac{1}{2}\boldsymbol{e}^\top(T)\boldsymbol{Q}_f\boldsymbol{e}(T) + \frac{1}{2}\int_0^T (\boldsymbol{e}^\top\boldsymbol{Q}\boldsymbol{e} + \boldsymbol{u}^\top\boldsymbol{R}\boldsymbol{u})\,\mathrm{d}t \right\}$$
$$\text{subject to} \quad \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0$$

$$\begin{aligned} \boldsymbol{e} &:= \boldsymbol{y} - \boldsymbol{y}_d \\ &= \boldsymbol{C}(\boldsymbol{x} - \boldsymbol{x}_d) \end{aligned}$$

Design weights:

$\mathbf{Q}_f$ = final tracking errors

$\mathbf{Q}$ = tracking errors

$\mathbf{R}$ = control inputs

This is a finite time-horizon optimal control problem which can be solved by using the Differential Riccati Equation (DRE); Athans and Falb (1996), pp. 793-801.

16.1.3 General Solution of the LQ Trajectory-Tracking Problem

Cost function

$$J = \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \boldsymbol{e}^\top(T) \boldsymbol{Q}_f \boldsymbol{e}(T) + \frac{1}{2} \int_0^T (\boldsymbol{e}^\top \boldsymbol{Q} \boldsymbol{e} + \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u}) \, \mathrm{d}t \right\}$$

$$\text{subject to} \quad \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0$$

$$\begin{aligned} \boldsymbol{e} &:= \boldsymbol{y} - \boldsymbol{y}_d \\ &= \boldsymbol{C}(\boldsymbol{x} - \boldsymbol{x}_d) \end{aligned}$$

Solution for Linear Time-Varying (LTV) Systems

state feedback

reference feedforward

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^\top(\boldsymbol{P}\boldsymbol{x} + \boldsymbol{h})$$

Final conditions

$$\begin{aligned} \dot{\boldsymbol{P}} &= -\boldsymbol{P}\boldsymbol{A} - \boldsymbol{A}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P} - \boldsymbol{C}^\top\boldsymbol{Q}\boldsymbol{C} \\ \dot{\boldsymbol{h}} &= -(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P})^\top\boldsymbol{h} + \boldsymbol{C}^\top\boldsymbol{Q}\boldsymbol{C}\boldsymbol{x}_d \end{aligned}$$

$$\begin{aligned} \boldsymbol{P}(T) &= \boldsymbol{C}^\top\boldsymbol{Q}_f\boldsymbol{C} \\ \boldsymbol{h}(T) &= -\boldsymbol{C}^\top\boldsymbol{Q}_f\boldsymbol{C}\boldsymbol{x}_d(T) \end{aligned}$$

16.1.3 General Solution of the LQ Trajectory-Tracking Problem

Numerical Solution of Differential Equations

Note that the initial conditions for **P** and **h** are not known, but the final conditions at time *t = T* are known.

Consequently, these equations must be integrated backward in time (*a priori*) to find the initial conditions, and then be executed forward in time again with the closed-loop plant from [*0, T*].

A nonlinear system can be simulated backwards in time by the following change of the integration variable $t = T - \tau$ with $dt = -d\tau$

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{G}(\boldsymbol{x}, t)\boldsymbol{u}, \qquad t \in [T,\, 0]$$

integrate backward in time

$$-\frac{\mathrm{d}\boldsymbol{x}(T-\tau)}{\mathrm{d}\tau} = \boldsymbol{f}(\boldsymbol{x}(T-\tau), T-\tau) + \boldsymbol{G}(\boldsymbol{x}(T-\tau), T-\tau)\boldsymbol{u}(T-\tau)$$

Finally, let $\boldsymbol{z}(\tau) = \boldsymbol{x}(T-\tau)$

$$\boxed{\frac{\mathrm{d}\boldsymbol{z}(\tau)}{\mathrm{d}\tau} = -\boldsymbol{f}(\boldsymbol{z}(\tau), T-\tau) - \boldsymbol{G}(\boldsymbol{z}(\tau), T-\tau)\boldsymbol{u}(T-\tau)}$$

integrate forward in time

16.1.3 General Solution of the LQ Trajectory-Tracking Problem

Example 16.2 (Optimal Time-Varying LQ Trajectory-Tracking Problem)

Mass-damper-spring system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

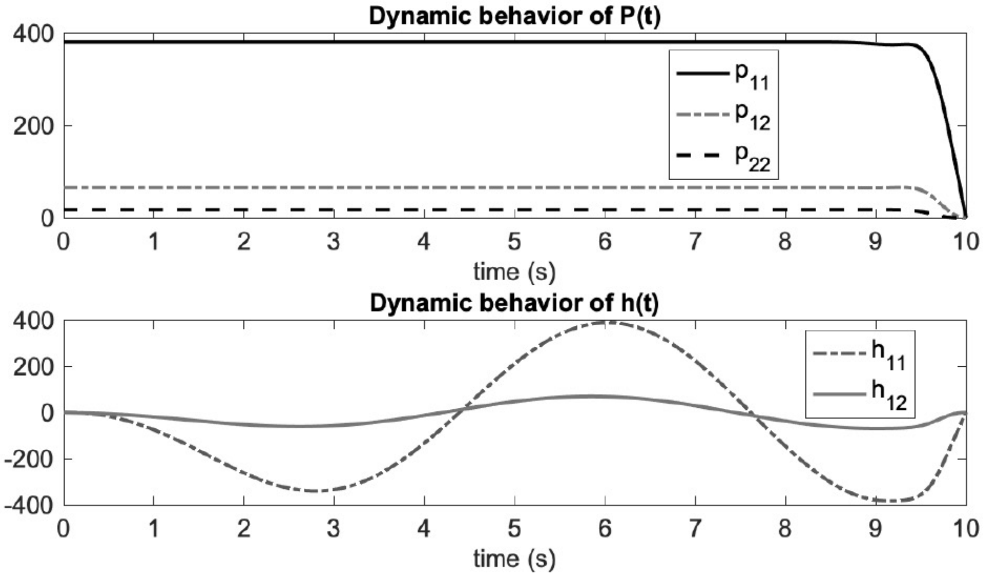
Reference model

$$\dot{\boldsymbol{x}}_d = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \boldsymbol{x}_d + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \qquad r = \sin(t)$$

$$y_d = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_d$$

See Matlab MSS toolbox script ExLQFinHor.m

16.1.3 General Solution of the LQ Trajectory-Tracking Problem



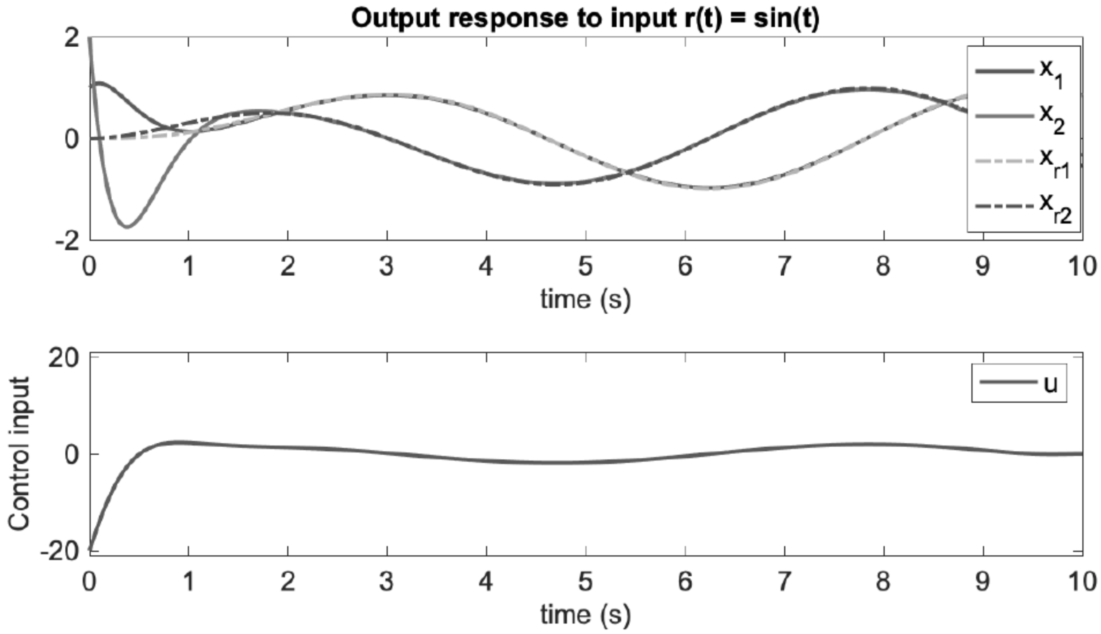
Optimal solutions found by backward integration.

Final conditions at $T = 10\ s$:

$\mathbf{P}(T) = \mathbf{0}$
$\mathbf{h}(\mathbf{T}) = \mathbf{0}$

see MSS toolbox ExLQFinHor.m

16.1.3 General Solution of the LQ Trajectory-Tracking Problem



Perfect tracking with reference feedforward

r(t) = sin(t) is the input to the reference model

16.1.3 General Solution of the LQ Trajectory-Tracking Problem

$$J = \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \int_0^T (\boldsymbol{e}^\top \boldsymbol{Q} \boldsymbol{e} + \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u})\, \mathrm{d}t \right\}$$

Approximate Solution for Linear Time-Invariant (LTI) Systems

For the case

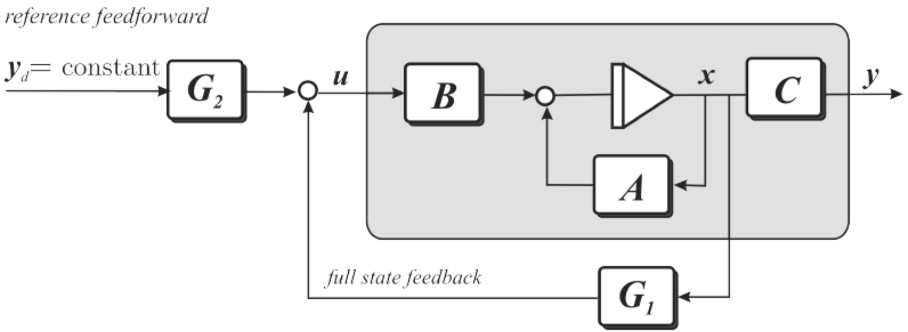
$\mathbf{x}_d$ constant, (can be relaxed to slowly varying)

the steady-state solutions of **P** and **h** can be used.

This gives

$$\boldsymbol{u} = \boldsymbol{G}_1\, \boldsymbol{x} + \boldsymbol{G}_2\, \boldsymbol{y}_d$$

$$\boldsymbol{G}_1 = -\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}_\infty$$
$$\boldsymbol{G}_2 = -\boldsymbol{R}^{-1}\boldsymbol{B}^\top (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{G}_1)^{-\top} \boldsymbol{C}^\top \boldsymbol{Q}$$



16.1.3 General Solution of the LQ Trajectory-Tracking Problem

Matlab:
The function `lqtracker.m` is implemented in the MSS toolbox for computation of the matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$.

```
function [G1,G2] = lqtracker(A,B,C,Q,R)
[K,P,E] = lqr(A,B,C'*Q*C,R);
G1 = -inv(R) * B' * P;
G2 = -inv(R) * B' * inv((A+B*G1)') * C' * Q;
```

For a mass–damper–spring system the optimal trajectory tracking controller can be designed is using the script `ExLQtrack.m`:

```
% Weights
Q = diag([1]);        % tracking error weights
R = diag([1]);        % input weights

% System matrices

A = [0 1; -1 -2];     % state matrix
B = [0; 1];           % input matrix
C = [1 0];            % output matrix

% Optimal gain matrices
[G1,G2] = lqtracker(A,B,C,Q,R)
```

16.1.4 Operability and Motion Sickness Incidence Criteria

- Operability criteria for manual and intellectual work as well as motion sickness are important design criteria for evaluation of autopilot and roll damping systems.
- Sea sickness is especially important in high-speed craft and ships with high vertical accelerations.

Standard Deviation (Root Mean Square) Criteria			
Vertical acceleration (w	Lateral acceleration (v	Roll angle (	Description of work
0.20 g	0.10 g	6.0 deg	Light manual work
0.15 g	0.07 g	4.0 deg	Heavy manual work
0.10 g	0.05 g	3.0 deg	Intellectual work
0.05 g	0.04 g	2.5 deg	Transit passengers
0.02 g	0.03 g	2.0 deg	Cruise liner


Courtesy to Scuttlebutt Sailing News

The table gives an indication on what type of work that can be expected to be carried out for different roll angles/sea states (Faltinsen 1990).

16.1.4 Operability and Motion Sickness Incidence Criteria

The ISO 2631-3:1997 Criterion for MSI

The International Standardization Organization (ISO) motion seasickness incidence criterion is reported in ISO 2631-3 (1997); see http://www.iso.ch.

The most important factors for seasickness are:

- vertical (heave) accelerations a_z (m/s²)
- exposure time t (hours)
- encounter frequency ω_e (rad/s)

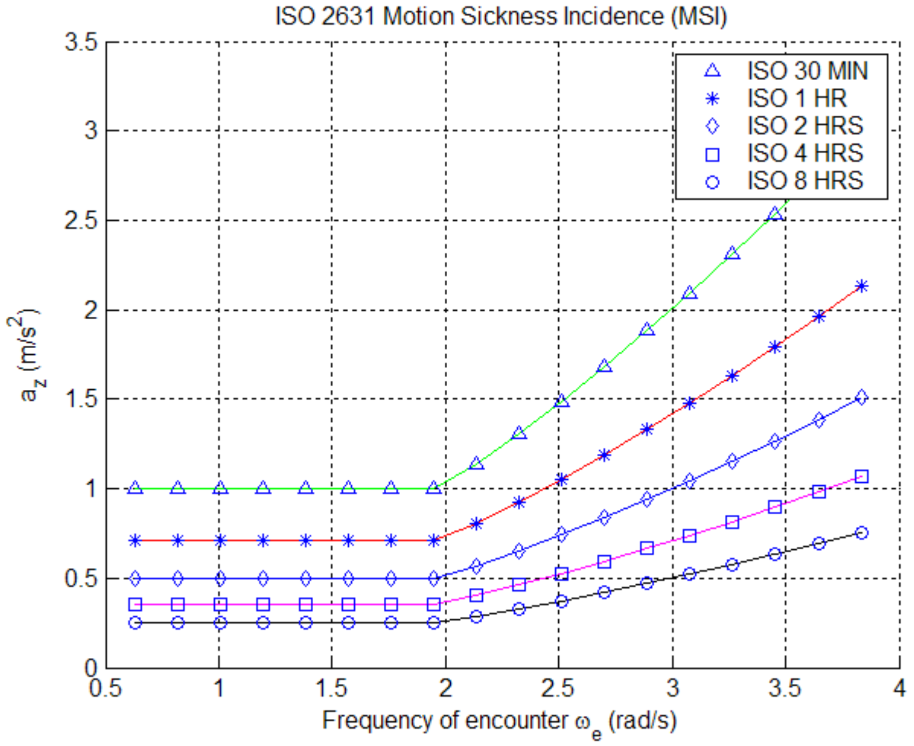
The ISO standard proposes an MSI of 10 % which means that 10 % of the passengers become seasick during t hours.

The MSI curves are given by:

$$a_z(t,\omega_e) = \begin{cases} 0.5\sqrt{2/t} & \text{for} \quad 0.1 \text{ Hz} < \frac{\omega_e}{2\pi} \leq 0.315 \text{ Hz} \\ 0.5\sqrt{2/t} \cdot 6.8837 \left(\frac{\omega_e}{2\pi}\right)^{1.67} & \text{for} \quad 0.315 \text{ Hz} \leq \frac{\omega_e}{2\pi} \leq 0.63 \text{ Hz} \end{cases}$$

ISO International Organization for Standardization

16.1.4 Operability and Motion Sickness Incidence Criteria



Matlab MSS toolbox: ExMSI.m

```
>> [a_z,w_e] = ISOmsi(t)
```

The ISO standard criterion for MSI proposes an MSI of 10 %. This means that 10 % of the passengers get seasick.

International Organization for Standardization

http://www.iso.ch

16.1.4 Operability and Motion Sickness Incidence Criteria

The O'Hanlon and McCauley (1974) Criterion for MSI

This method (the probability integral method) is frequently used since it produces a MSI criterion in percent for combinations of heave acceleration a_z (m/s²) and frequency of encounter ω_e (rad/s).

The criterion is given by

$$\boxed{\mathrm{MSI} = 100\left[0.5 \pm \operatorname{erf}\left(\frac{\pm\log_{10}(a_z/g) \mp \mu_{\mathrm{MSI}}}{0.4}\right)\right] \quad (\%)}$$

The MSI index is defined as the number of sea sick people in percentage for an exposure time of two hours

where

$$\mu_{\mathrm{MSI}} = -0.819 + 2.32\,(\log_{10}\omega_e)^2$$

$$\operatorname{erf}(x) = \operatorname{erf}(-x) = \frac{1}{\sqrt{2\pi}}\int_0^x \exp\left(-\frac{z^2}{2}\right)\mathrm{d}z$$

The major drawback of the O'Hanlon and McCauley method is that it only applies to a two-hour exposure time.

Another effect to consider is that the O'Hanlon and McCauley MSI criterion is derived from tests with young men seated separately in insulated cabins.

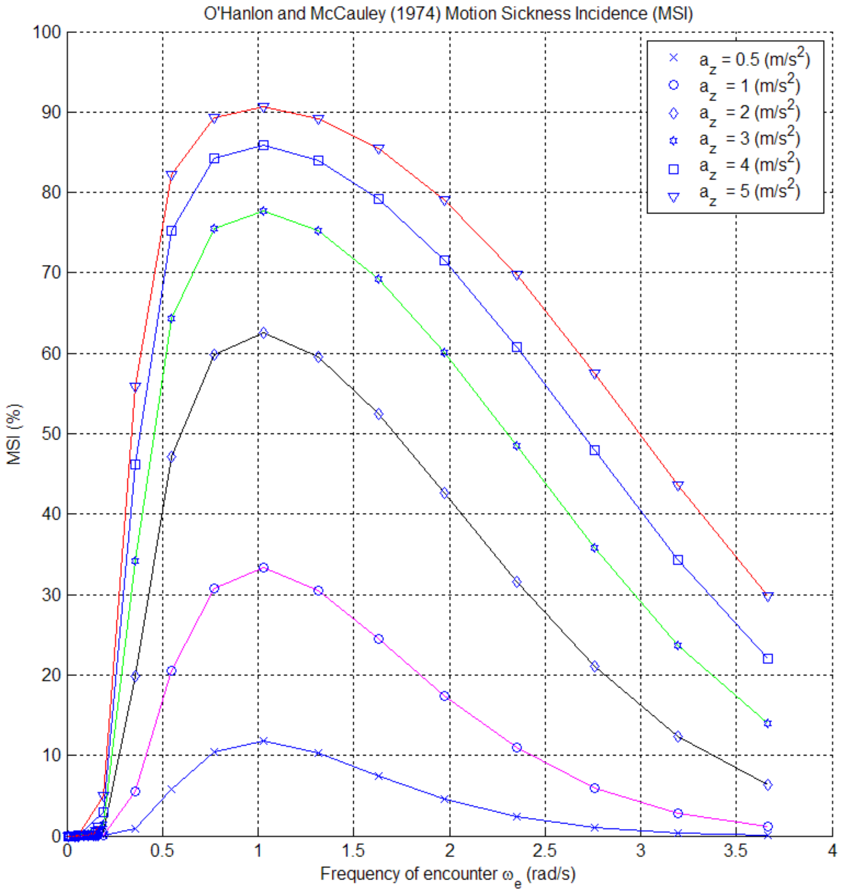
16.1.4 Operability and Motion Sickness Incidence Criteria

Matlab MSS toolbox: ExMSI.m

```
>> msi = HMmsi(a_z,w_e)
```

The MSI index is defined as the number of sea sick people in percentage for an exposure time of two hours

Courtesy to Scuttlebutt Sailing News



16.1.5 Case Study: Optimal Heading Autopilot for Marine Craft

The ship autopilot problem can be solved as a linear-quadratic (LQ) optimization problem

$$J = \min_{\delta} \left\{ \frac{\alpha}{T} \int_0^T \left(e^2 + \lambda_1 r^2 + \lambda_2 \delta^2 \right) \mathrm{d}\tau \right\}$$

α is a constant to be interpreted later

$e = \psi_d - \psi$ is the heading error

δ is the actual rudder angle

λ_1 and λ_2 are two factors weighting the cost of heading errors e and heading rate r against the control effort δ

We will discuss three steering criteria for control weighting

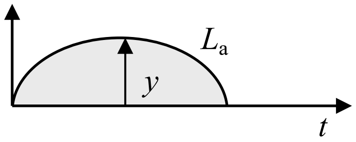
- **Koyama (1967)**
- **Norrbin (1971)**
- **Van Amerongen and Van Nauta Lemke (1978)**

16.1.5 Case Study: Optimal Heading Autopilot for Marine Craft

The Steering Criterion of Koyama (1967)

The first criterion was derived by Koyama who observed that the ship's swaying motion $e = \psi - \psi_d$ could be approximated by a sinusoid during autopilot control, that is:

$$y \quad \sin\ et \qquad\qquad y \quad e\cos\ et$$



The length of one arch L_a of the sinusoid is:

$$L_a \quad {}_0\sqrt{1 \quad y^2}\ d \qquad {}_0\sqrt{1 \quad e^2\cos^2 e \quad}\ d \qquad \left(1 \quad \frac{e^2}{4}\right)$$

Hence, the relative elongation due to a sinusoidal course error is:

$$\frac{L}{L} \qquad \frac{L_a \ \ L}{L} \qquad \frac{1 \ \ e^2/4}{} \qquad \frac{e^2}{4}$$

Koyama proposed minimizing the speed loss term $e^2/4$ against the increased resistance due to steering given by the term δ^2:

α

$$J = \min_{\delta}\left\{100\left(\frac{\pi}{180}\right)^2 \frac{1}{T}\int_0^T \left[\frac{e^2}{4} + \lambda_2\delta^2\right] \mathrm{d}\tau \approx \frac{0.0076}{T}\int_0^T [e^2 + \lambda_2\delta^2]\mathrm{d}\tau\right\}$$

where J is the loss of speed (%) Notice that $\lambda_1 = 0$ in this analysis

16.1.5 Case Study: Optimal Heading Autopilot for Marine Craft

The Steering Criterion of Norrbin (1971)

Consider the surge equation in the form

$$m \quad X_u \; u \qquad X_{|u|u}|u|u \qquad 1 \quad t \; T \quad T_{\text{loss}}$$

$$T_{\text{loss}} \qquad m \quad X_{vr} \; vr \quad X_{cc} \; c^2 \quad {}^2 \qquad X_{rr} \quad mx_g \; r^2 \quad X_{\text{ext}}$$

Norrbin minimizes the loss term T_{loss} to obtain maximum forward speed u. Consequently, the controller should minimize the centripetal term vr, the square rudder angle δ^2 and the square heading rate r^2 while the unknown disturbance term X_{ext} is neglected in the analysis. The assumptions are:

1. The sway velocity v is approximately proportional to r. Recall the Nomoto model gives

$$v \;\; s \qquad \frac{K_v \;\; 1 \;\; T_v s}{K \;\; 1 \;\; Ts} r \;\; s \qquad \frac{K_v}{K} r \;\; s$$ (assuming that $T_v \approx T$)

Hence, the centripetal term vr will be approximately proportional to the square of the heading rate, that is $vr \approx (K_v/K)r^2$.

2. The ship's yawing motion is periodic (sinusoid) under autopilot control such that

$$r_{\max} \qquad {}_r \; e_{\max}$$

where ω_r is the frequency of the sinusoidal yawing.

16.1.5 Case Study: Optimal Heading Autopilot for Marine Craft

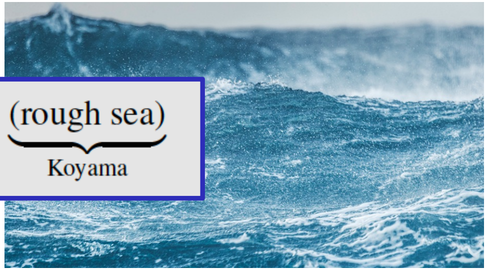
These two assumptions suggest that the loss term

$$T_{\text{loss}} = (m + X_{vr})vr + X_{cc\delta\delta}c^2\,\delta^2 + (X_{rr} + mx_g)r^2 + X_{\text{ext}}$$

can be minimized by minimizing e^2 and δ^2 which is the same result obtained in Koyama's analysis.

The only difference between the criteria of Norrbin and Koyama is that the λ_2 -values arising from Norrbin's approach will be different when computed for the same ship. The performance of the controller also depends on the sea state. This suggests that a trade-off between the λ_2 -values proposed by Koyama and Norrbin could be made according to:

$$\underbrace{\text{(calm sea)}}_{\text{Norrbin}} \qquad 0.1 \;\le\; \lambda_2 \;\le\; 10 \qquad \underbrace{\text{(rough sea)}}_{\text{Koyama}}$$



16.1.5 Case Study: Optimal Heading Autopilot for Marine Craft

The Steering Criterion of Van Amerongen and Van Nauta Lemke (1978)

Experiments with the steering criteria of Koyama and Norrbin showed that the performance could be further improved by considering the squared yaw rate r^2, in addition to e^2 and δ^2 . Consequently, the following criterion was proposed

$$J = \min_{\delta} \left\{ \frac{0.0076}{T} \int_0^T (e^2 + \lambda_1 r^2 + \lambda_2 \delta^2)\, \mathrm{d}\tau \right\}$$

For a tanker and a cargo ship, Van Amerongen and Van Nauta Lemke (1978, 1980) gave the following values for the weighting factors λ_1 *and* λ_2 corresponding to the data set of Norrbin (1972)

$$\begin{array}{llll} \text{Tanker:} & L_{pp} = 300\ \text{m}, & \lambda_1 = 15\,000, & \lambda_2 = 8.0 \\ \text{Cargo ship:} & L_{pp} = 200\ \text{m}, & \lambda_1 = 1600, & \lambda_2 = 6.0 \end{array}$$

16.1.5 Case Study: Optimal Heading Autopilot for Marine Craft

It can be shown that the PD controller (gain-scheduled with respect to speed *U*)

$$\delta = -K_p(\psi - \psi_d) - K_d r$$

$$K_p = \sqrt{\frac{1}{\lambda_2}}$$

$$K_d = \frac{L}{U}\,\frac{\sqrt{1 + 2K_pK'T' + K'^2(U/L)^2\,(\lambda_1/\lambda_2)} - 1}{K'}$$

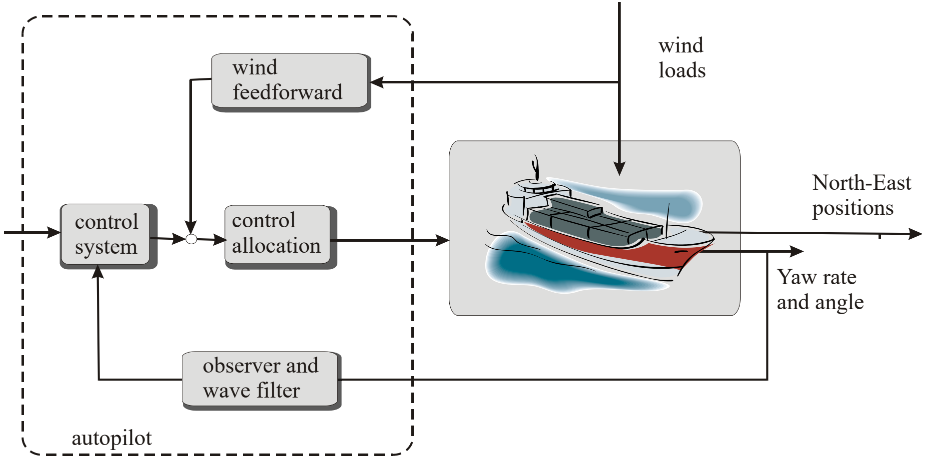
is the optimal solution solving

$$J = \min_{\delta}\left\{\frac{0.0076}{T}\int_0^T (e^2 + \lambda_1 r^2 + \lambda_2 \delta^2)\,\mathrm{d}\tau\right\}$$

for the prime-scaled Nomoto model

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

$$T'\dot{r} + (U/L)r = (U/L)^2 K'\,\delta$$

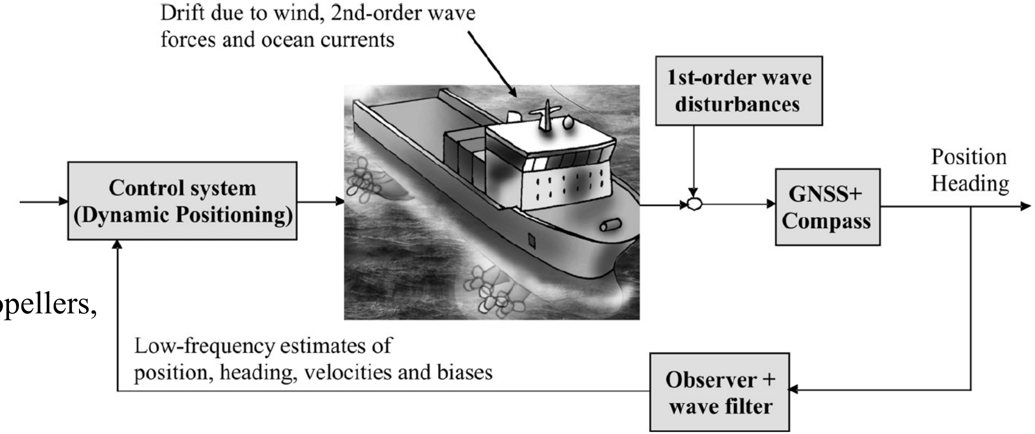


16.1.6 Case study: Optimal DP System for Surface Vessels

An alternative to the nonlinear PID controller is to formulate the problem as a linear optimal control problem. The LQ controller will be designed under the assumption that all states can be measured. This assumption can, however, be relaxed by combining the LQ controller with a linear Kalman filter for optimal state estimation (LQG control).

Recall from Section 6.7.3

$$
\begin{aligned}
\dot{\boldsymbol{\eta}} &= \boldsymbol{R}(t)\boldsymbol{\nu} \\
\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}\boldsymbol{\nu} &= \boldsymbol{R}^\top(t)\boldsymbol{b} + \boldsymbol{\tau} \\
\dot{\boldsymbol{b}} &= \boldsymbol{0}
\end{aligned}
$$



In order to incorporate the limitations of the propellers, the model is augmented by actuator dynamics

$$\dot{\boldsymbol{\tau}} = \boldsymbol{A}_{\text{thr}}(\boldsymbol{\tau} - \boldsymbol{\tau}_{\text{com}})$$

Augmented controller model

$$\boxed{\dot{\boldsymbol{x}}_c = \boldsymbol{A}_c(t)\boldsymbol{x}_c + \boldsymbol{B}_c\boldsymbol{\tau}_{\text{com}}}$$

$$\boldsymbol{x}_c := [\boldsymbol{\eta}^\top, \boldsymbol{\nu}^\top, \boldsymbol{\tau}^\top]^\top ;$$

$$\boldsymbol{A}_{\text{thr}} = -\text{diag}\{1/T_{\text{surge}},\ 1/T_{\text{sway}},\ 1/T_{\text{yaw}}\}$$

$$\boldsymbol{A}_c(t) = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{R}(t) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & -\boldsymbol{M}^{-1}\boldsymbol{D} & \boldsymbol{M}^{-1}\boldsymbol{R}^\top(t) \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{A}_{\text{thr}} \end{bmatrix}, \quad \boldsymbol{B}_c = \begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} \\ -\boldsymbol{A}_{\text{thr}} \end{bmatrix}$$

16.1.6 Case study: Optimal DP System for Surface Vessels

The **Kalman filter** can be designed using only position and heading angle measurements. For this purpose, the filter states are chosen as

$$\boldsymbol{x}_f := [\boldsymbol{\eta}^\top, \boldsymbol{b}^\top, \boldsymbol{\nu}^\top]^\top$$

The WF model is omitted for simplicity but in an industrial system six more states should be added following the approach in Section 13.4.6. This is necessary to obtain proper wave filtering.

The LTV Kalman filter model takes the following form

$$\boxed{\begin{aligned} \dot{\boldsymbol{x}}_f &= \boldsymbol{A}_f(t)\boldsymbol{x}_f + \boldsymbol{B}_f\boldsymbol{\tau} + \boldsymbol{E}_f\boldsymbol{w} \\ \boldsymbol{y}_f &= \boldsymbol{C}_f\boldsymbol{x}_f + \boldsymbol{\varepsilon} \end{aligned}}$$

$$\boldsymbol{A}_f(t) = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{R}(t) \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{M}^{-1}\boldsymbol{R}^\top(t) & -\boldsymbol{M}^{-1}\boldsymbol{D} \end{bmatrix}$$

$$\boldsymbol{B}_f = \begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} \\ \boldsymbol{M}^{-1} \end{bmatrix} \qquad \boldsymbol{C}_f = [\boldsymbol{I}_3, \boldsymbol{0}_{3\times6}]$$

Position and heading angle measurements

Matlab:
A DP model of a supply vessel length 76.2 m in surge, sway and yaw is included in the MSS toolbox as `supply.m`. The nondimensional system matrices are (Fossen *et al.* 1996)

$$\boldsymbol{M}'' = \begin{bmatrix} 1.1274 & 0 & 0 \\ 0 & 1.8902 & -0.0744 \\ 0 & -0.0744 & 0.1278 \end{bmatrix}, \quad \boldsymbol{D}'' = \begin{bmatrix} 0.0358 & 0 & 0 \\ 0 & 0.1183 & -0.0124 \\ 0 & -0.0041 & 0.0308 \end{bmatrix}$$

These values are defined in accordance to the bis system (see Appendix D.1) such that

$$\boldsymbol{M} = m\boldsymbol{T}^{-2}(\boldsymbol{T}\boldsymbol{M}''\boldsymbol{T}^{-1}), \quad \boldsymbol{D} = m\sqrt{g/L}\,\boldsymbol{T}^{-2}(\boldsymbol{T}\boldsymbol{D}''\boldsymbol{T}^{-1})$$

where $\boldsymbol{T} = \mathrm{diag}\{1,\ 1,\ L\}$.

16.1.6 Case study: Optimal DP System for Surface Vessels

Optimal control law with wind feedforward and integral action

$$\boldsymbol{\tau}_{\text{com}} = \boldsymbol{\tau}_{\text{LQ}} - \hat{\boldsymbol{\tau}}_{\text{wind}}$$

$\boldsymbol{\tau}_{LQ}$ is the optimal feedback and $\boldsymbol{\tau}_{\text{wind}}$ is an estimate of the wind forces.

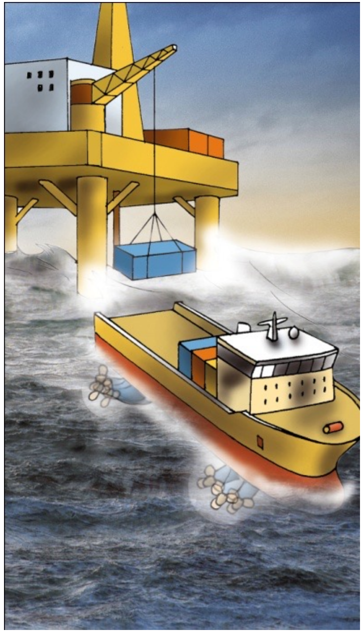
The LQ control objective is to obtain $\mathbf{x} = \mathbf{0}$ such that $\boldsymbol{\eta} = \boldsymbol{\nu} = \boldsymbol{\tau} = \mathbf{0}$.
Hence, we can compute τ_{LQ} by minimizing the performance index:

$$J = \min_{\boldsymbol{\tau}_{\text{LQ}}} \left\{ \frac{1}{2} \int_0^T (\boldsymbol{x}_c^\top \boldsymbol{Q} \boldsymbol{x}_c + \boldsymbol{\tau}_{\text{LQ}}^\top \boldsymbol{R} \boldsymbol{\tau}_{\text{LQ}})\, \mathrm{d}\tau \right\}$$

where $\mathbf{R} = \mathbf{R}^\top > 0$ and $\mathbf{Q} = \mathbf{Q}^\top \geq 0$ are two cost matrices to be specified by the user. The $\mathbf{Q}$ matrix is defined as $\mathbf{Q} = \text{diag}\{\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3\}$ where the weights $\mathbf{Q}_1$, $\mathbf{Q}_2$ and $\mathbf{Q}_3$ put penalty on position/heading $\boldsymbol{\eta}$, velocity $\boldsymbol{\nu}$ and actuator dynamics $\boldsymbol{\tau}$, respectively.

$$\boldsymbol{\tau}_{\text{LQ}} = \underbrace{-\boldsymbol{R}^{-1}\boldsymbol{B}_c^\top \boldsymbol{P}}_{\boldsymbol{G}} \boldsymbol{x}_c$$

$$\boldsymbol{P}\boldsymbol{A}_c + \boldsymbol{A}_c^\top \boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}_c\boldsymbol{R}^{-1}\boldsymbol{B}_c^\top \boldsymbol{P} + \boldsymbol{Q} = -\dot{\boldsymbol{P}}$$



16.1.6 Case study: Optimal DP System for Surface Vessels

Integral Action

In order to obtain zero steady-state errors in surge, sway and yaw, we must include integral action in the control law. Integral action can be included by using state augmentation.

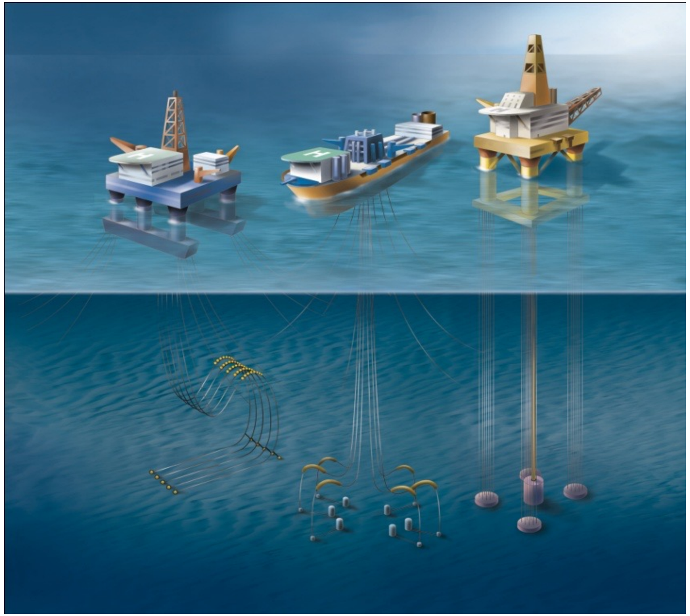
Since we want the three outputs (x^n, y^n, ψ) to be regulated to zero, we can augment no more than 3 integral states to the model.

$$\boldsymbol{z}_c := \int_0^t \boldsymbol{y}_c(\tau)\mathrm{d}\tau \quad \Longrightarrow \quad \dot{\boldsymbol{z}}_c = \boldsymbol{y}_c$$

$$\boldsymbol{y}_c = \boldsymbol{C}_c\boldsymbol{x}_c, \qquad \boldsymbol{C}_c = [\boldsymbol{I}_3,\ \boldsymbol{0}_{3\times 6}]$$

$$\dot{\boldsymbol{x}}_a = \boldsymbol{A}_a(t)\boldsymbol{x}_a + \boldsymbol{B}_a\boldsymbol{\tau}_{\mathrm{com}} \qquad \boldsymbol{x}_a := [\boldsymbol{z}_c^\top,\ \boldsymbol{x}_c^\top]^\top$$

$$\boldsymbol{A}_a(t) = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & \boldsymbol{C}_c \\ \boldsymbol{0}_{9\times 3} & \boldsymbol{A}_c(t) \end{bmatrix}, \quad \boldsymbol{B}_a = \begin{bmatrix} \boldsymbol{0}_{3\times 3} \\ \boldsymbol{B}_c \end{bmatrix}$$



16.1.6 Case study: Optimal DP System for Surface Vessels

The performance index for the integral controller becomes:

$$J = \min_{\boldsymbol{\tau}_{\mathrm{LQ}}} \left\{ \frac{1}{2} \int_0^T \left(\boldsymbol{x}_a^\top \boldsymbol{Q}_a \boldsymbol{x}_a + \boldsymbol{\tau}_{\mathrm{LQ}}^\top \boldsymbol{R} \boldsymbol{\tau}_{\mathrm{LQ}} \right) \mathrm{d}\tau \right\} \qquad \boldsymbol{Q}_a = \begin{bmatrix} \boldsymbol{Q}_I & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} \end{bmatrix} \geq 0$$

where $\mathbf{R} = \mathbf{R}^\top > 0$. The matrix $\mathbf{R}_I = \mathbf{R}_I > 0$ is used to specify the integral times in surge, sway and yaw.

The optimal PID controller is

$$\boldsymbol{\tau}_{\mathrm{LQ}} = \boldsymbol{G}_a \boldsymbol{x}_a = \boldsymbol{G} \boldsymbol{x}_c + \boldsymbol{G}_I \underbrace{\int_0^t \boldsymbol{y}_c(\tau)\, \mathrm{d}\tau}_{\boldsymbol{z}_c}$$

$$\boldsymbol{G}_a = -\boldsymbol{R}^{-1} \boldsymbol{B}_a^\top \boldsymbol{P}$$
$$\boldsymbol{P}\boldsymbol{A}_a + \boldsymbol{A}_a^\top \boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}_a \boldsymbol{R}^{-1} \boldsymbol{B}_a^\top \boldsymbol{P} + \boldsymbol{Q}_a = -\dot{\boldsymbol{P}}$$

Controllability of the augmented system $(\mathbf{A}_a, \mathbf{B}_a)$ is checked in Matlab by using the command

```
>> n_ctr = rank(ctrb(Aa,Ba)) = 12
```

Hence, a marine craft with additional states for integral action is controllable.

16.1.6 Case study: Optimal DP System for Surface Vessels

LQG Control - Linear Separation Principle

In practice only some of the states are measured. A minimum requirement is that the position and heading of the ship is measured such that velocities and bias terms can be estimated by an observer. This is done under the assumption that the states **x** can be replaced with the estimated states:

$$\boldsymbol{\tau}_{\mathrm{LQ}} = \boldsymbol{G}\hat{\boldsymbol{x}}_c + \boldsymbol{G}_I \boldsymbol{C} \int_0^t \hat{\boldsymbol{x}}_c(\tau)\, \mathrm{d}\tau$$

where the state estimate can be computed using a

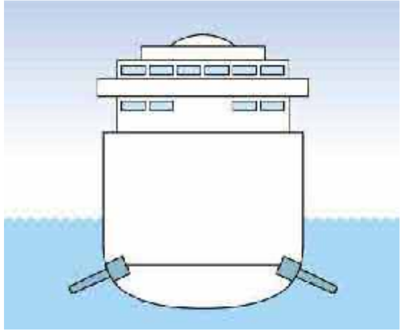
- **Kalman filter (Section 13.4.6)**
- **Nonlinear passive observer (Section 13.5.1)**

For the Kalman filter in cascade with the LQ controller there exists a linear separation principle guaranteeing that the estimation and regulation errors go to zero.

This is referred to as LQG control and it was first applied to design DP systems by Balchen et al. (1976, 1980a, 1980b), and Grimble et al. (1980a, 1980b).

16.1.7 Case Study: Optimal Rudder-Roll Damping Systems for Ships

Active Roll Damping



Fin Stabilizers:

- Considerable damping if the speed is not too low
- Drawback: increased drag and underwater noise
- Retractable fins (inside the hull when not in use)
- High costs associated with the installation
- Fin stabilizers were patented by Thornycroft in 1889

Rudder-Roll Damping (RRD)

- RRD by means of the rudder is relatively inexpensive compared to fin stabilizers, has approximately the same effectiveness, and causes no drag or underwater noise if the system is turned off
- Drawback: RRD requires a relatively fast rudder to be effective, typically rudder rates 5-20 deg/s are needed. RRD will not be effective at low ship speeds

For a history of ship stabilization, see Bennett (1991). A detailed evaluation of different ship roll stabilization systems can be found in Sellars and Martin (1992)

16.1.7 Case Study: Optimal Rudder-Roll Damping Systems for Ships

Transfer Functions for Simultaneously Steering and Rudder-Roll Damping

The transfer functions for the state-space model (6.150) is

$$\frac{\phi}{\delta}(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \approx \frac{K_{\text{roll}}\,\omega_{\text{roll}}^2\,(T_5 s + 1)}{(T_4 s + 1)(s^2 + 2\zeta\omega_{\text{roll}} s + \omega_{\text{roll}}^2)}$$

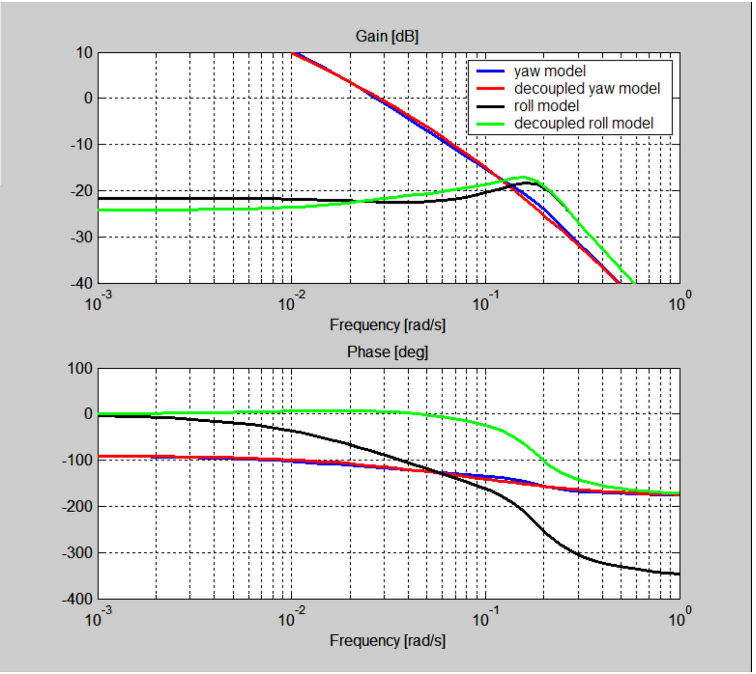
$$\frac{\psi}{\delta}(s) = \frac{c_3 s^3 + c_2 s^2 + c_1 s + c_0}{s(s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0)} \approx \frac{K_{\text{yaw}}\,(T_3 s + 1)}{s(T_1 s + 1)(T_2 s + 1)}$$

This is a rough approximation since all interactions are neglected.

In particular, the roll mode is inaccurate as seen from the plots.

Matlab MSS toolbox: ExRRD1.m
Container ship model: Lcontainer.m

Plot showing roll and yaw transfer functions



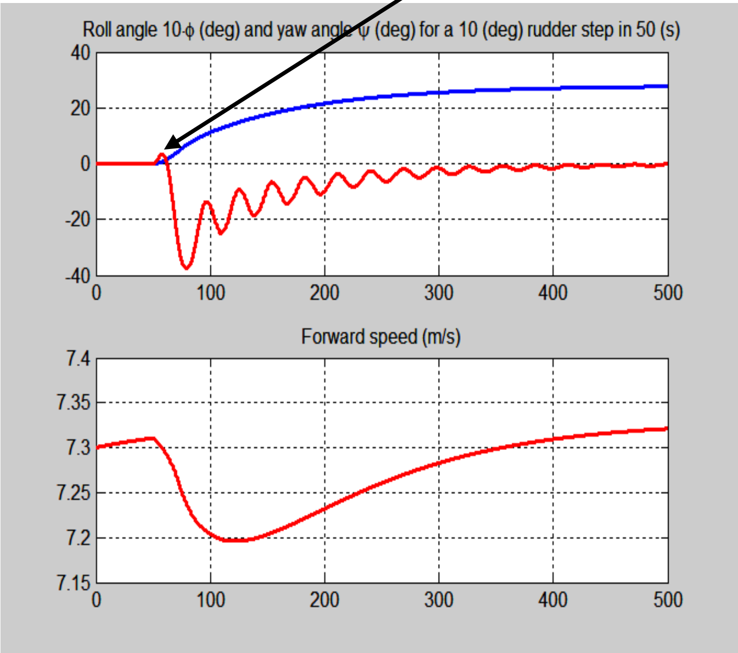
Container ship: Son and Nomoto (1981)

16.1.7 Case Study: Optimal Rudder-Roll Damping Systems for Ships

Length: L = 175 (m)
Displacement: 21,222 (m³)
Service speed u_0 = 7.0 (m/s)

Right-half-plane zero: $z \quad 0.036$ (rad/s)

Nonminimum phase property



$$— \; s \quad \frac{0.0032\;\; s \quad 0.036 \;\; s \quad 0.077}{s \quad 0.026 \;\; s \quad 0.116 \;\; s^2 \;\; 0.136s \quad 0.036} \quad \frac{0.083 \;\; 1 \quad 49.1s}{1 \quad 31.5s \;\; s^2 \;\; 0.134s \quad 0.033}$$

$$_{\text{roll}} \quad 0.189 \text{ (rad/s)}$$

$$0.36$$

$$— \; s \quad \frac{0.0024 \;\; s \quad 0.0436 \;\; s^2 \;\; 0.162s \quad 0.035}{s \;\; s \quad 0.0261 \;\; s \quad 0.116 \;\; s^2 \;\; 0.136s \quad 0.036} \quad \frac{0.032 \;\; 1 \quad 16.9s}{s \;\; 1 \quad 24.0s \;\; 1 \quad 9.2s}$$

Matlab MSS toolbox: ExRRD3.m

16.1.7 Case Study: Optimal Rudder-Roll Damping Systems for Ships

Consider the 4-DOF linear model (6.50)

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \qquad \boldsymbol{x} = [v,\ p,\ r,\ \phi,\ \psi]^\top$$

$$\phi = \boldsymbol{c}_{\text{roll}}^\top \boldsymbol{x}, \qquad \psi = \boldsymbol{c}_{\text{yaw}}^\top \boldsymbol{x}$$

The control objective is simultaneously

- **Heading control:** $\psi = \psi_d = \text{constant}$
- **RRD:** $\phi = \phi_d = 0$

There will be a trade-off between accurate heading control (minimizing d) and control action needed to increase:

- Natural frequency ω_{roll}
- Relative damping ratio ζ_{roll}

Notice that it is impossible to regulate φ to a nonzero value while simultaneously controlling the heading angle ψ to a nonzero value by means of one single rudder.

16.1.7 Case Study: Optimal Rudder-Roll Damping Systems for Ships

Optimization problem for course keeping, roll damping and minimum fuel consumption

$$J = \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \int_0^T \left(\tilde{\boldsymbol{y}}^\top \boldsymbol{Q} \tilde{\boldsymbol{y}} + \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u} \right) \mathrm{d}\tau \right\}$$

$$\boldsymbol{x} = [v,\ p,\ r,\ \phi,\ \psi]^\top$$

$$\boldsymbol{y} = [p,\ r,\ \phi,\ \psi]^\top, \qquad \boldsymbol{y}_d = [0,\ 0,\ 0,\ \psi_d]^\top$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} \qquad \boldsymbol{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

LQ Solution

$$\boldsymbol{u} = \boldsymbol{G}_1 \boldsymbol{x} + \boldsymbol{G}_2 \boldsymbol{y}_d$$

where

$$\boldsymbol{G}_1 = -\boldsymbol{R}^{-1} \boldsymbol{B}^\top \boldsymbol{P}_\infty$$

$$\boldsymbol{G}_2 = -\boldsymbol{R}^{-1} \boldsymbol{B}^\top (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{G}_1)^{-\top} \boldsymbol{C}^\top \boldsymbol{Q}$$

with $\boldsymbol{P}_\infty = \boldsymbol{P}_\infty^\top > 0$ given by

$$\boldsymbol{P}_\infty \boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{P}_\infty - \boldsymbol{P}_\infty \boldsymbol{B} \boldsymbol{R}^{-1} \boldsymbol{B}^\top \boldsymbol{P}_\infty + \boldsymbol{C}^\top \boldsymbol{Q} \boldsymbol{C} = \boldsymbol{0}$$

16.1.7 Case Study: Optimal Rudder-Roll Damping Systems for Ships

Frequency Separation and Bandwidth Limitations

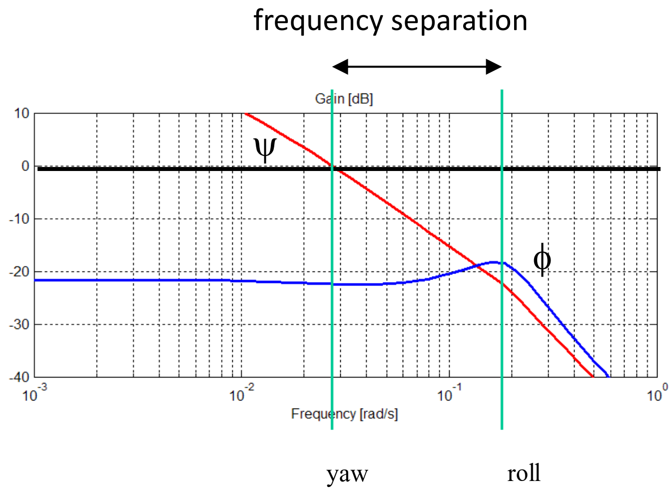
If one rudder is used to control both φ and ψ, frequency separation is necessary.

Assume that the steering dynamics is slower than the frequency $1/T_l$, and that the natural frequency in roll is higher than $1/T_h$.

Then the VRU and compass measurements can be low-pass and high-pass filtered:

$$\underline{}_{\text{vru}}\ s \qquad h_h\ s \qquad \frac{T_h s}{1 \quad T_h s}$$

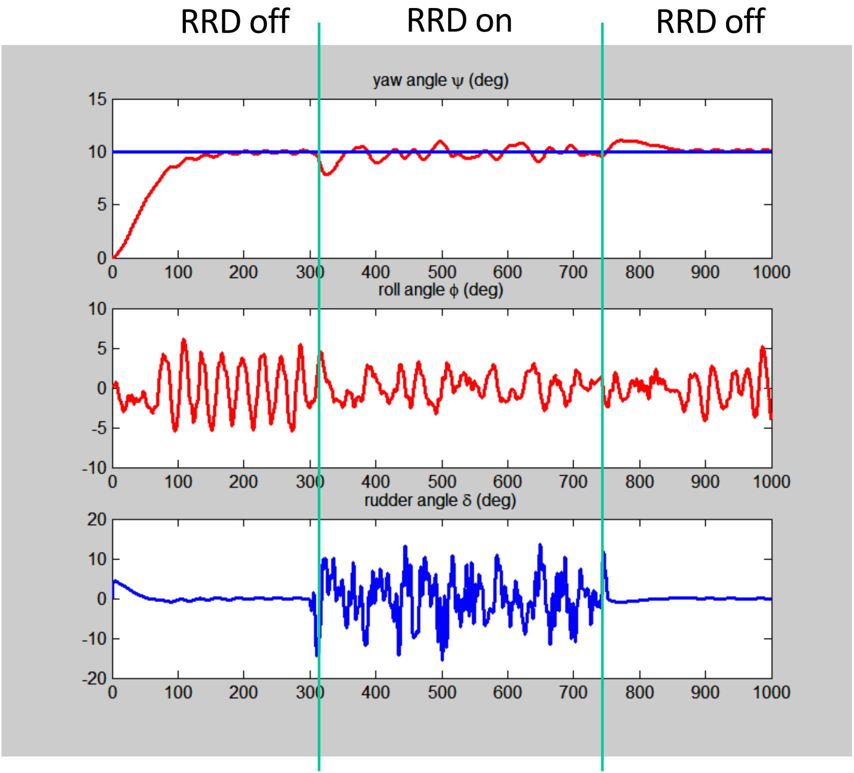
$$\underline{}_{\text{compass}}\ s \qquad h_l\ s \qquad \frac{1}{1 \quad T_l s}$$



Frequency separation criterion:

$$\underbrace{\omega_{\text{yaw}}}_{\substack{\text{cross-over}\\ \text{frequency}}} < \underbrace{\omega_b}_{\substack{\text{bandwidth}\\ \text{in yaw}}} < \underbrace{1/T_l}_{\substack{\text{low-pass filter}\\ \text{frequency}}} < \underbrace{1/T_h}_{\substack{\text{high-pass filter}\\ \text{frequency}}} < \underbrace{\omega_{\text{roll}}}_{\substack{\text{natural}\\ \text{frequency}}}$$

16.1.7 Case Study: Optimal Rudder-Roll Damping Systems for Ships



Matlab MSS toolbox: ExRRD2.m

16.1.7 Case Study: Optimal Rudder-Roll Damping Systems for Ships

The percentage roll reduction of RRD system can be computed by using the following criterion of Oda et el. (1992)

$$\text{Roll reduction} = \frac{\sigma_{\text{AP}} - \sigma_{\text{RRD}}}{\sigma_{\text{AP}}} \times 100\ \%$$

$$\begin{aligned} \sigma_{\text{AP}} &= \text{standard deviation of roll rate during course-keeping (RRD off)} \\ \sigma_{\text{RRD}} &= \text{standard deviation of roll rate during course-keeping (RRD on)} \end{aligned}$$

For the case study with Lcontainer.m, we obtained:

$\sigma_{\text{AP}} = 0.0105$

$\sigma_{\text{RRD}} = 0.0068.$

This resulted in a roll reduction of approximately 35 % during course keeping.

For small high-speed vessels, a roll reduction as high as 50-75 % can be obtained.

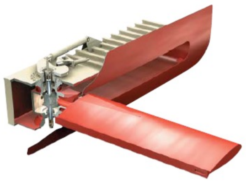
16.1.8 Case study: Optimal Fin and RRD Systems for Ships

This section discusses methods for autopilot roll stabilization using fins alone or in combination with rudders. The main motivation is:

- Prevent cargo damage and to increase the effectiveness of the crew by avoiding or reducing seasickness. This is also important from a safety point of view.
- For naval ships, critical marine operations like landing a helicopter, formation control, underway replenishment, or effectiveness of the crew during combat are critical operations.

Several passive and active (feedback control) systems have been proposed to accomplish roll reduction; Burger and Corbet (1967), Lewis (1989), and Bhattacharyya (1978).

Perez (2005). *Ship Motion Control: Course Keeping and Roll Stabilization using Rudder and Fins*, Springer.

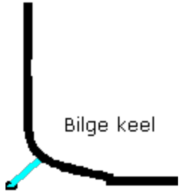


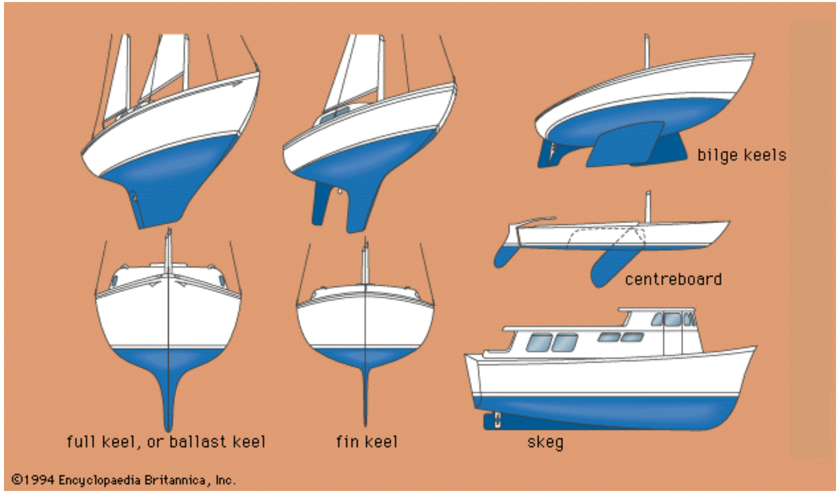
16.1.8 Case study: Optimal Fin and RRD Systems for Ships

Passive Roll Damping

Bilge keels are fins in planes approximately perpendicular to the hull or near the turn of the bilge. The longitudinal extent varies from about 25-50 % of the length of the ship. Bilge keels are widely used and inexpensive but increase the hull resistance. In addition to this, they are effective mainly around the natural roll frequency of the ship. This effect significantly decreases with the speed of the ship. Bilge keels were first demonstrated in about 1870.

Hull Modifications: The shape and size of the ship hull can be optimized for minimum rolling using hydrostatic and hydrodynamic criteria. This must, however, be done before the ship is built.





16.1.8 Case study: Optimal Fin and RRD Systems for Ships

Optimal Fin and Rudder-Roll Damping Systems

Note that a stand-alone fin stabilization system can be constructed by simply removing the rudder inputs from the input matrix. When designing an LQ optimal fin/RRD system the following model

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}\boldsymbol{\nu} = \boldsymbol{T}\boldsymbol{f}, \qquad \boldsymbol{f} = \boldsymbol{K}\boldsymbol{u}$$

$$\boldsymbol{\nu} = [u,\, v,\, p,\, r]^\top$$

$$\dot{\boldsymbol{\nu}} = -\underbrace{\boldsymbol{M}^{-1}\boldsymbol{D}}_{\boldsymbol{A}}\boldsymbol{\nu} + \underbrace{\boldsymbol{M}^{-1}\boldsymbol{T}}_{\boldsymbol{B}}\boldsymbol{f}$$

Energy-Optimal Criterion for Combined Fin and RRD Stabilization

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{y}_d$$

$$\boldsymbol{\tau} = \boldsymbol{T}\boldsymbol{f}.$$

$$J = \min_{\boldsymbol{f}} \left\{ \frac{1}{2}\int_0^T (\boldsymbol{e}^\top \boldsymbol{Q}\boldsymbol{e} + \boldsymbol{f}^\top \boldsymbol{R}_f \boldsymbol{f})\, \mathrm{d}t \right\}$$

Force f

$$= \min_{\boldsymbol{\tau}} \left\{ \frac{1}{2}\int_0^T (\boldsymbol{e}^\top \boldsymbol{Q}\boldsymbol{e} + \boldsymbol{\tau}^\top \underbrace{(\boldsymbol{T}_w^\dagger)^\top \boldsymbol{R}_f \boldsymbol{T}_w}_{\boldsymbol{R}_\tau} \boldsymbol{\tau})\, \mathrm{d}t \right\}$$

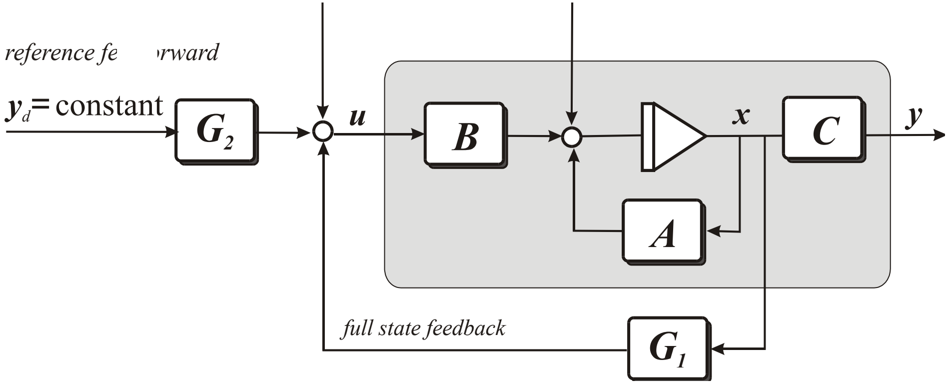
Generalized force τ

16.1.8 Case study: Optimal Fin and RRD Systems for Ships

The solution to the LQ problem is

$$\boldsymbol{\tau} = \boldsymbol{G}_1 \boldsymbol{x} + \boldsymbol{G}_2 \boldsymbol{y}_d$$
$$\boldsymbol{G}_1 = -\left((\boldsymbol{T}_w^{\dagger})^{\top} \boldsymbol{R}_f \boldsymbol{T}_w^{\dagger}\right)^{-1} \boldsymbol{B}^{\top} \boldsymbol{P}_{\infty}$$
$$\boldsymbol{G}_2 = -\left((\boldsymbol{T}_w^{\dagger})^{\top} \boldsymbol{R}_f \boldsymbol{T}_w^{\dagger}\right)^{-1} \boldsymbol{B}^{\top} (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{G}_1)^{-\top} \boldsymbol{C}^{\top} \boldsymbol{Q}$$

$$\boldsymbol{P}_{\infty} \boldsymbol{A} + \boldsymbol{A}^{\top} \boldsymbol{P}_{\infty} - \boldsymbol{P}_{\infty} \boldsymbol{B} [(\boldsymbol{T}_w^{\dagger})^{\top} \boldsymbol{R}_f \boldsymbol{T}_w^{\dagger}]^{-1} \boldsymbol{B}^{\top} \boldsymbol{P}_{\infty} + \boldsymbol{C}^{\top} \boldsymbol{Q} \boldsymbol{C} = \boldsymbol{0}$$



16.2 State Feedback Linearization

The basic idea with feedback linearization is to transform the nonlinear system dynamics into a linear system (Freund 1973) and use linear pole placement.

Conventional control techniques like pole placement and linear-quadratic optimal control theory can then be applied to the linear system. In robotics, this technique is commonly referred to as *computed torque control*.

Transformations that can be used both for BODY and NED applications will be presented:

- ✓ **Decoupling in the BODY frame:** velocity control
- ✓ **Decoupling in the NED frame:** position and attitude control

16.2.1 Decoupling in the BODY Frame (Velocity Control)

The control objective is to transform the marine craft dynamics

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{\tau}$$

$$\boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta})$$

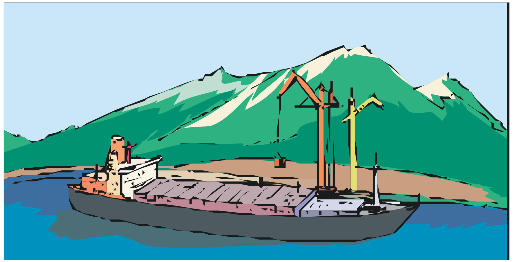
into a linear system expressed in {b}

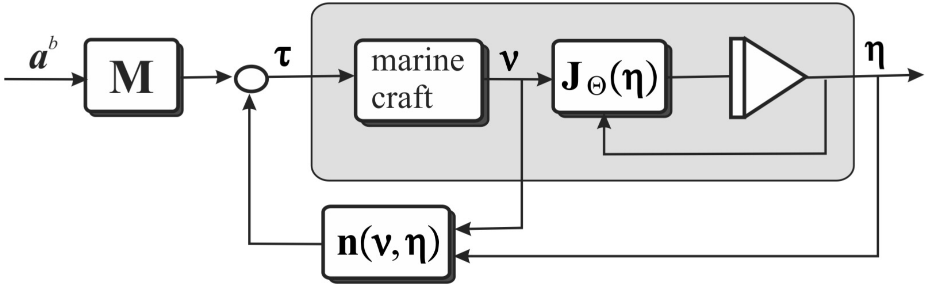
$$\boxed{\dot{\boldsymbol{\nu}} = \boldsymbol{a}^b}$$

Solution: choose the control law as:

$$\boldsymbol{\tau} = \boldsymbol{M}\boldsymbol{a}^b + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta})$$

where the commanded acceleration vector $\mathbf{a}^b$ can be chosen by e.g. pole placement or linear-quadratic optimal control theory.





16.2.1 Decoupling in the BODY Frame (Velocity Control)

Pole Placement

$$\boldsymbol{a}^b = \dot{\boldsymbol{\nu}}_d - \boldsymbol{K}_p\tilde{\boldsymbol{\nu}} - \boldsymbol{K}_i \int_0^t \tilde{\boldsymbol{\nu}}(\tau)\,\mathrm{d}\tau$$

$$\boldsymbol{K}_p = 2\boldsymbol{\Lambda}, \qquad \boldsymbol{K}_i = \boldsymbol{\Lambda}^2 \qquad\qquad \boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$$

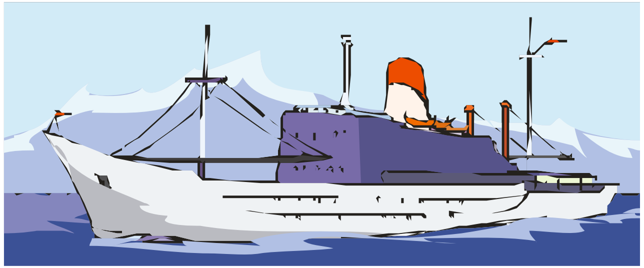
$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{\tau} \qquad\qquad \boldsymbol{\tau} = \boldsymbol{M}\boldsymbol{a}^b + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta})$$

Error dynamics

$$\boldsymbol{M}(\dot{\tilde{\boldsymbol{\nu}}} - \boldsymbol{a}^b) = \boldsymbol{M}\left(\dot{\tilde{\boldsymbol{\nu}}} + 2\boldsymbol{\Lambda}\tilde{\boldsymbol{\nu}} + \boldsymbol{\Lambda}^2 \int_0^t \tilde{\boldsymbol{\nu}}(\tau)\,\mathrm{d}\tau\right) = \boldsymbol{0}$$

$$(s + \lambda_i)^2 \int_0^t \tilde{\boldsymbol{\nu}}(\tau)\,\mathrm{d}\tau = 0 \quad (i = 1, \ldots, n)$$

The poles are located at $-\lambda_i$



16.2.2 Decoupling in the NED Frame (Position and Attitude Control)

The control objective is to transform the marine craft dynamics

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{n}(\boldsymbol{\nu},\boldsymbol{\eta}) = \boldsymbol{\tau}$$

$$\boldsymbol{n}(\boldsymbol{\nu},\boldsymbol{\eta}) = \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta})$$

into a linear system expressed in {n}

$$\boxed{\ddot{\boldsymbol{\eta}} = \boldsymbol{a}^n}$$

Solution: choose the control law as

$$\boldsymbol{\tau} = \boldsymbol{M}\boldsymbol{a}^b + \boldsymbol{n}(\boldsymbol{\nu},\boldsymbol{\eta})$$

$$\dot{\boldsymbol{\nu}} = \boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})[\ddot{\boldsymbol{\eta}} - \dot{\boldsymbol{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}]$$

Error dynamics

$$\boldsymbol{M}(\dot{\boldsymbol{\nu}} - \boldsymbol{a}^b) = \boldsymbol{M}\boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})[\ddot{\boldsymbol{\eta}} - \dot{\boldsymbol{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} - \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{a}^b] = \boldsymbol{0}$$

Choosing

$$\boldsymbol{a}^n = \dot{\boldsymbol{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} + \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{a}^b \qquad \boldsymbol{a}^b = \boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})[\boldsymbol{a}^n - \dot{\boldsymbol{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}]$$

gives

$$\boldsymbol{M}^*(\ddot{\boldsymbol{\eta}} - \boldsymbol{a}^n) = \boldsymbol{0} \qquad \boldsymbol{M}^* = \boldsymbol{J}_\Theta^{-\top}(\boldsymbol{\eta})\boldsymbol{M}\boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta}) > 0$$

16.2.2 Decoupling in the NED Frame (Position and Attitude Control)

$$\boldsymbol{\tau} = \boldsymbol{M}\boldsymbol{a}^b + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta})$$

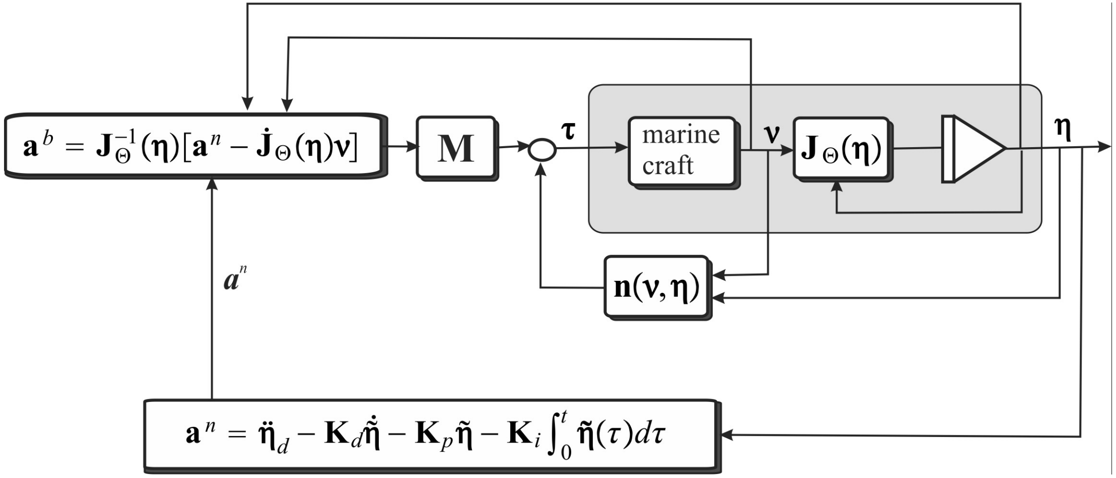
$$\boldsymbol{a}^b = \boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})[\boldsymbol{a}^n - \dot{\boldsymbol{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}]$$

$$\boldsymbol{a}^n = \ddot{\boldsymbol{\eta}}_d - \boldsymbol{K}_d\dot{\tilde{\boldsymbol{\eta}}} - \boldsymbol{K}_p\tilde{\boldsymbol{\eta}} - \boldsymbol{K}_i\int_0^t \tilde{\boldsymbol{\eta}}(\tau)\,\mathrm{d}\tau$$

Error dynamics (pole placement)

$$\ddot{\tilde{\boldsymbol{\eta}}} + \boldsymbol{K}_d\dot{\tilde{\boldsymbol{\eta}}} + \boldsymbol{K}_p\tilde{\boldsymbol{\eta}} + \boldsymbol{K}_i\int_0^t \tilde{\boldsymbol{\eta}}(\tau)\,\mathrm{d}\tau = \boldsymbol{0}$$

$$\begin{aligned}
\boldsymbol{K}_d &= 3\boldsymbol{\Lambda} = \mathrm{diag}\{3\lambda_1, 3\lambda_2, \ldots, 3\lambda_n\}\\
\boldsymbol{K}_p &= 3\boldsymbol{\Lambda}^2 = \mathrm{diag}\{3\lambda_1^2, 3\lambda_2^2, \ldots, 3\lambda_n^2\}\\
\boldsymbol{K}_i &= \boldsymbol{\Lambda}^3 = \mathrm{diag}\{\lambda_1^3, \lambda_2^3, \ldots, \lambda_n^3\}
\end{aligned}$$



16.2.3 Case Study: Speed Control Based on Feedback Linearization

Consider a simplified model of a ship in surge

$$mu \quad d_1 u \quad d_2|u|u$$

The commanded acceleration is calculated as

$$a^b \quad u_d \quad K_p \; u \quad u_d \quad K_i \; {}_0^t \; u \quad u_d \; d$$

PI controller with reference feedforward

This suggests that the speed controller should be computed as:

$$m \; u_d \quad K_p \; u \quad u_d \quad K_i \; {}_0^t \; u \quad u_d \; d \quad d_1 u \quad d_2|u|u$$

$$ma^b$$

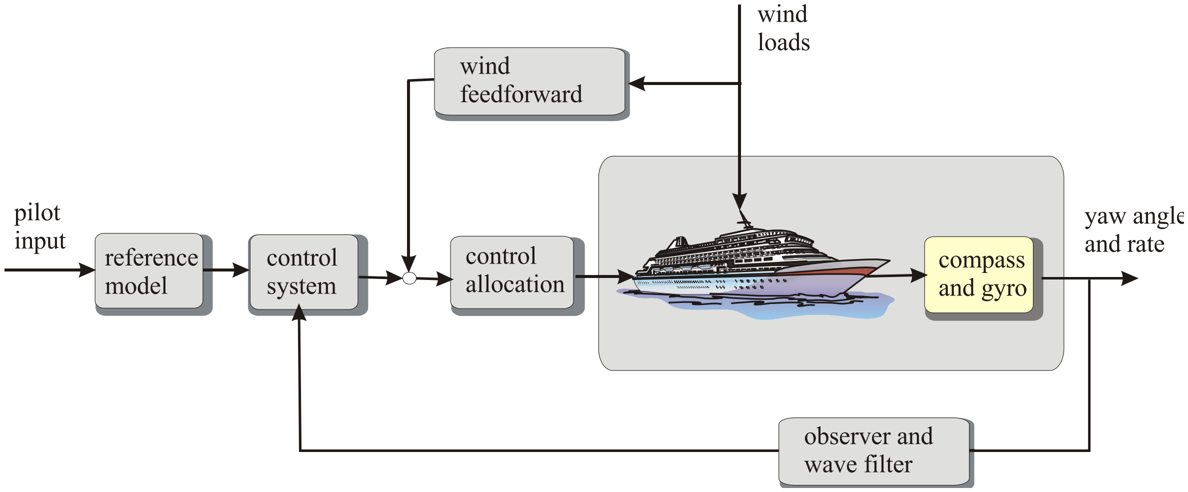
compensation of nonlinear elements

with reference model:

$$\ddot{u}_d + 2\zeta\omega_n\dot{u}_d + \omega_n^2 u_d = \omega_n^2 r^b$$

where $\mathbf{r}^b$ is the commanded input (desired surge speed).

16.2.4 Case Study: Autopilot Based on Feedback Linearization



Reference model

$$\psi_d^{(3)} + (2\zeta + 1)\omega_n \ddot{\psi}_d + (2\zeta + 1)\omega_n^2 \dot{\psi}_d + \omega_n^3 \psi_d = \omega_n^3 r^n$$

16.2.4 Case Study: Autopilot Based on Feedback Linearization

Consider the nonlinear autopilot model of Norrbin (1963)

$$r$$
$$mr \quad d_1 r \quad d_2|r|r$$

The commanded acceleration can be calculated as

$$a^n \quad r_d \quad K_d \; r \quad r_d \quad K_p \qquad {}_d \quad K_i \int_0^t \qquad {}_d \; d$$

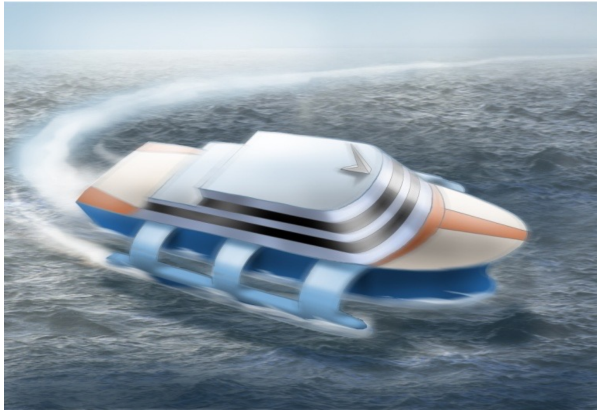
where r_d is the desired angular velocity and ψ_d is the desired heading angle. This implies that $a^n = a^b$
Choosing the decoupling control law as

$$\boxed{\tau = m\left[\dot{r}_d - K_d(r - r_d) - K_p(\psi - \psi_d) - K_i \int_0^t (\psi - \psi_d)\,\mathrm{d}\tau\right] + d_1 r + d_2|r|r}$$

the resulting error dynamics become

$$r$$
$$r \quad K_d r \quad K_p \qquad 0$$

Chapter 16 – Advanced Motion Control Systems



16.3 Integrator Backstepping

A brief History of Backstepping

The idea of integrator backstepping seems to have appeared simultaneously, often implicit, in the works of Koditschek (1987), Sonntag and Sussmann (1988), Tsinias (1988) and Byrnes and Isidori (1989).

Integrator backstepping appeared as a recursive design technique in Saberi et al. (1990) and it was further developed by Kanellakopoulos et al. (1992).

The relationship between backstepping and passivity has been established by Lozano et al. (1992). A tutorial overview of backstepping is given by Kokotovic (1991).

Adaptive and nonlinear backstepping designs are described in detail by **Krstic et al. (1995)**. This includes methods for parameter adaptation, tuning functions, and modular designs for both full state feedback and output feedback (observer backstepping).

The concept of vectorial backstepping was first introduced by Fossen and Berge (1997). Vectorial backstepping exploits the structural properties of nonlinear MIMO systems.

Techniques for integral action in nonlinear systems using backstepping designs were first addressed by Loria, Fossen and Teel (1999).

16.3.2 The Main Idea of Integrator Backstepping

Backstepping is a design methodology for construction of a feedback control law through a recursive construction of a control Lyapunov function (CLF).

A smooth positive definite and radially unbounded function $V:\mathbb{R}^n \rightarrow \mathbb{R}_+$ is called a Control Lyapunov Function (CLF) for

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

$$\inf_{\boldsymbol{u} \in \mathbb{R}^r} \left\{ \frac{\partial V}{\partial \boldsymbol{x}}(\boldsymbol{x}) \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \right\} < 0, \quad \forall \boldsymbol{x} \neq \boldsymbol{0}$$

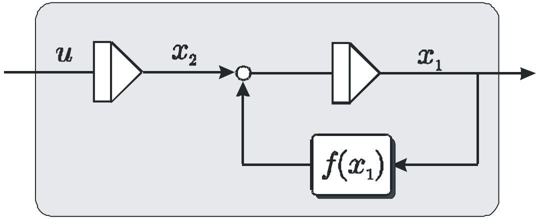
Nonlinear backstepping designs are strongly related to feedback linearization.
Feedback linearization methods cancel all nonlinearities in the system while backstepping gives more design flexibility.

In particular, the designer is given the possibility to exploit "good" nonlinearities while "bad" nonlinearities can be dominated e.g. by adding nonlinear damping. Hence, additional robustness is obtained, which is important in industrial control systems since cancellation of all nonlinearities require precise models that are difficult to obtain in practice.

16.3.2 The Main Idea of Integrator Backstepping

Considering the nonlinear system

$$\begin{array}{lll} x_1 & f\ x_1 & x_2 \\ x_2 & u & \\ y & x_1 & \end{array}$$



Let the design objective be regulation of $y(t) \to 0$ as $t \to \infty$.

The recursive design starts with the system x_1 and continues with x_2.
Two steps are needed, one for each integrator.

A change of coordinates will be introduced:

$$\mathbf{x} \quad x_1, x_2$$

$$\mathbf{z} \quad \mathbf{x} \qquad \mathbf{x} \quad {}^{1}\ \mathbf{z} \qquad\qquad \mathbf{z} \quad z_1, z_2$$

Global diffeomorphism - Global invertible function with the property that both the function and its inverse are smooth.

16.3.2 The Main Idea of Integrator Backstepping

Step 1

For the first system, the state x_2 is chosen as a virtual control input

The first backstepping variable is chosen as (wants to regulate y to zero):

$$z_1 \quad y \quad x_1$$

$$\begin{array}{lll} x_1 & f\,x_1 & x_2 \\ x_2 & u & \\ y & x_1 & \end{array}$$

The virtual control is defined as:

$$x_2 \; : \quad {}_1 \quad z_2$$

second backstepping variable

stabilizing function to be specified later

Hence, the z_1-system can be written:

$$\dot{z}_1 \quad f\,z_1 \qquad {}_1 \quad z_2$$

The new state variable z_2 will not be used in the first step, but its presence is important since z_2 is needed to couple the z_1-system to the next system, that is the z_2-system to be considered in the next step.

16.3.2 The Main Idea of Integrator Backstepping

$$\dot{z}_1 \quad f \; z_1 \qquad 1 \quad z_2$$

A CLF for this system is

$$V_1 \quad \frac{1}{2} z_1^2$$

$$V_1 \quad z_1 \dot{z}_1$$

$$z_1 \; f \; z_1 \qquad 1 \quad z_1 z_2$$

leave this term to **Step 2**
(stabilizing of the z_2-dynamics)

Choose the stabilizing function such that it stabilizes the z_1-system

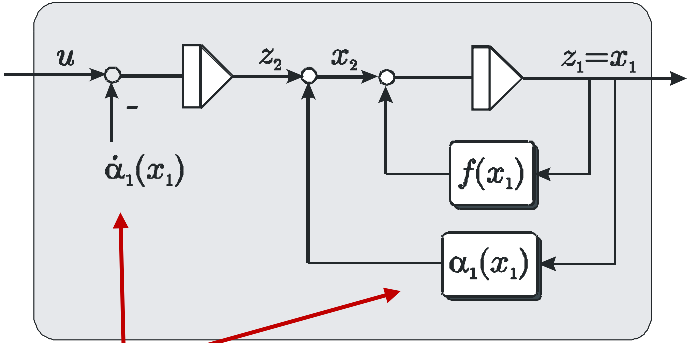
$$1 \qquad f \; z_1 \qquad k_1 z_1$$

gives

$$V_1 \qquad k_1 z_1^2 \quad z_1 z_2$$

If $z_2 = 0$, then the z_1-system is stable

$$\dot{z}_1 \qquad k_1 z_1 \quad z_2$$



these terms cancel

16.3.2 The Main Idea of Integrator Backstepping

Step 2

Time differentiation of $z_2 \quad x_2 \quad {}_1$ gives

$$\begin{aligned} x_1 &\quad f\,x_1 \quad x_2 \\ x_2 &\quad u \\ y &\quad x_1 \end{aligned}$$

$$\begin{aligned} \dot{z}_2 &\quad x_2 \quad {}_1 \\ &\quad u \quad {}_1 \end{aligned}$$

A CLF for the z_2-system is

$$\begin{aligned} V_2 &\quad V_1 \quad \frac{1}{2} z_2^2 \\ V_2 &\quad V_1 \quad \dot{z}_2 z_2 \\ &\quad k_1 z_1^2 \quad z_1 z_2 \quad \dot{z}_2 z_2 \\ &\quad k_1 z_1^2 \quad z_2 \; z_1 \quad \dot{z}_2 \\ &\quad k_1 z_1^2 \quad z_2 \; u \quad {}_1 \quad z_1 \end{aligned}$$

Since the system has relative degree 2, the control input u appears in the second step. Hence, choosing the control law as

$$u \quad {}_1 \quad z_1 \quad k_2 z_2 \qquad V_2 \quad k_1 z_1^2 \quad k_2 z_2^2 \quad 0, \quad z_1 \quad 0, z_2 \quad 0$$

16.3.2 The Main Idea of Integrator Backstepping

Implementation Aspects

When implementing the control law, it is important to avoid expressions involving the time derivatives of the states. For this simple system only 1 must be evaluated.

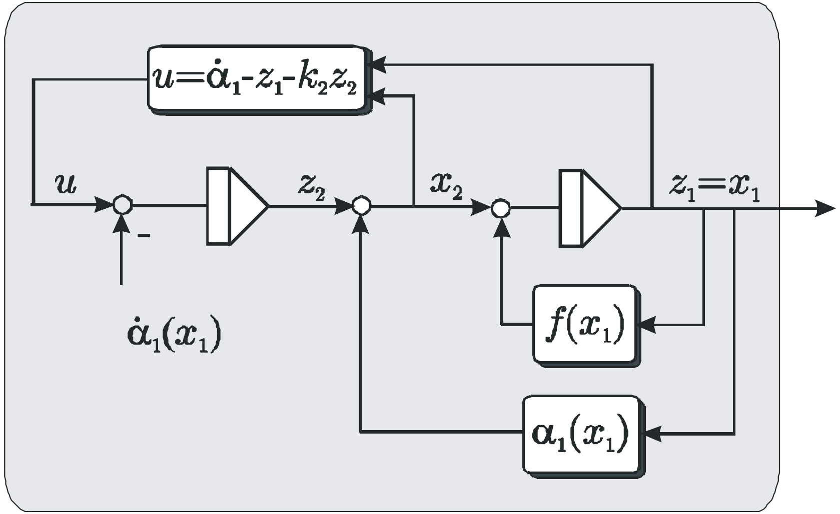
This can be done by time differentiation of $\alpha_1(x_1)$ along the trajectory of x_1

$$1 \qquad f\, z_1 \qquad k_1 z_1$$

$$z_1 \qquad y \qquad x_1$$

$$1 \qquad \frac{f\, x_1}{x_1} x_1 \qquad k_1 x_1$$

$$\left(\frac{f\, x_1}{x_1} \qquad k_1 \right) f\, x_1 \qquad x_2$$



16.3.2 The Main Idea of Integrator Backstepping

Backstepping Coordinate Transformation

$$\mathbf{z} \qquad \mathbf{x}$$

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \quad f\, x_1 \qquad k_1 x_1 \end{bmatrix}$$

$$\mathbf{x} \qquad {}^{1}\ \mathbf{z}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \quad f\, z_1 \qquad k_1 z_1 \end{bmatrix}$$

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = -\underbrace{\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}}_{\text{diagonal matrix}} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\text{skew-symmetrical matrix}} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$\dot{\boldsymbol{z}} = -\boldsymbol{K}\boldsymbol{z} + \boldsymbol{S}\boldsymbol{z}$$

$$V_2 \qquad \frac{1}{2}\mathbf{z}\ \mathbf{z}$$

$$V_2 \qquad \mathbf{z} \qquad \mathbf{Kz} \qquad \mathbf{Sz}$$

$$\mathbf{z}\ \mathbf{Kz}$$

The equilibrium point **z** = **0** is globally exponentially stable (GES) if **x** and thus **z** (global diffeomorphism) span R^n

16.3.2 The Main Idea of Integrator Backstepping

Theorem A.3 (Global Exponential Stability)

Let $\mathbf{x}_e$ be the equilibrium point and assume that $\mathbf{f}(\mathbf{x})$ is locally Lipschitz in $\mathbf{x}$.

Let $V\colon \mathbb{R}^n \to \mathbb{R}_+$ be a continuously differentiable and radially unbounded function satisfying

$$V \ \mathbf{x} \qquad \mathbf{x} \ \mathbf{P}\mathbf{x} \quad 0, \qquad \mathbf{x} \quad \mathbf{0}$$

$$V \ \mathbf{x} \qquad \mathbf{x} \ \mathbf{Q}\mathbf{x} \quad 0, \qquad \mathbf{x} \quad \mathbf{0}$$

constant matrices $\mathbf{P} = \mathbf{P}^\top > 0$ and $\mathbf{Q} = \mathbf{Q}^\top > 0$, then the equilibrium point $\mathbf{x}_e$ is GES and the state vector satisfies

$$\mathbf{x} \ t \quad {}_2 \qquad \sqrt{\frac{{}_{\max} \ \mathbf{P}}{{}_{\min} \ \mathbf{P}}} \ \exp \qquad t \quad \mathbf{x} \ 0 \quad {}_2$$

is a bound on the convergence rate

$$\frac{{}_{\min} \ \mathbf{Q}}{2 \ {}_{\max} \ \mathbf{P}} \qquad 0$$

Proof: see Appendix A.

16.3.2 The Main Idea of Integrator Backstepping

Relationship between Backstepping and Conventional PD Control

The resulting backstepping control law can be written:

$$u \quad \Big(\frac{f\, x_1}{x_1} \quad k_1 \Big) f\, x_1 \quad x_2 \quad x_1 \quad k_2\; x_2 \quad f\, x_1 \quad k_1 x_1$$

If $f(x_1) = -x_1$ (linear system), it is seen that

$$\begin{aligned} u \quad & 1 \quad k_1 \quad x_1 \quad x_2 \quad x_1 \quad k_2\; x_2 \quad x_1 \quad k_1 x_1 \\ & \underbrace{2 \quad k_1 k_2 \quad k_1 \quad k_2}_{K_p} \; x_1 \quad \underbrace{k_1 \quad k_2 \quad 1}_{K_d} \; x_2 \end{aligned}$$

This confirms that backstepping applied to a 2nd-order linear system gives PD control

16.3.2 The Main Idea of Integrator Backstepping

Backstepping versus Feedback Linearization

The backstepping control law is in fact equal to a feedback linearizing controller since the nonlinear function $f(x_1)$ is perfectly compensated for by choosing the stabilizing function as

$$ {}_1 \qquad f \; x_1 \qquad k_1 z_1 $$

One of the nice features of backstepping is that the stabilizing functions can be modified to exploit so-called "good" nonlinearities. For instance, assume that

$$ f \; x_1 \qquad a_0 x_1 \qquad a_1 x_1^2 \qquad a_2 |x_1| x_1 $$

good damping

where a_0, a_1 and a_2 are assumed to be unknown positive constants.

Good damping should be exploited in the control design and not cancelled out.

The destabilizing term $a_1 x_1^2$ must be perfectly compensated for or dominated by adding a nonlinear damping term proportional to x_1^3

16.3.2 The Main Idea of Integrator Backstepping

Nonlinear damping suggests the following candidate for the stabilizing function

$$_1 \quad k_1 z_1 \quad _1 z_1^3 \qquad k_1 \quad 0 \text{ and } _1 \quad 0$$

linear damping

nonlinear damping

$$\dot{z}_1 \quad f\ z_1 \quad _1 \quad z_2$$

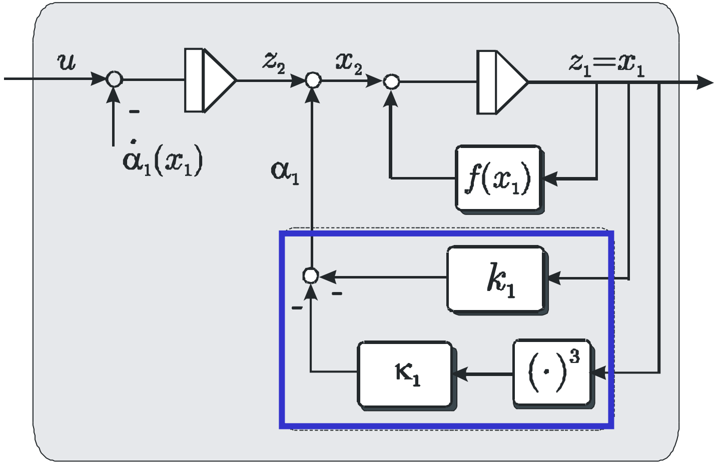
$$a_0 z_1 \quad a_1 z_1^2 \quad a_2 |z_1| z_1 \quad k_1 \quad _1 z_1^2\ z_1 \quad z_2$$

$$a_0 \quad a_2 |z_1| \quad k_1\ z_1 \quad a_1 z_1^2 \quad _1 z_1^3 \quad z_2$$

good damping

bad damping

A CLF will now be constructed to dominate bad damping and exploit good damping



16.3.2 The Main Idea of Integrator Backstepping

Consider the CLF

$$V_1 \quad \frac{1}{2} z_1^2$$

$$V_1 \qquad a_0 \quad a_2|z_1| \quad k_1 \; z_1^2 \quad a_1 z_1^3 \qquad {}_1 z_1^4 \quad z_1 z_2$$

Step 1

$$V_2 \quad V_1 \quad \frac{1}{2} z_2^2$$

$$V_2 \qquad a_0 \quad a_2|z_1| \quad k_1 \; z_1^2 \quad a_1 z_1^3 \qquad {}_1 z_1^4 \quad z_2 \; z_1 \quad u \qquad {}_1$$

Step 2

energy dissipation

energy dissipation or generation

Control law

$$u \qquad {}_1 \quad k_2 z_2 \quad z_1$$

$$V_2 \qquad a_0 \quad a_2|z_1| \quad k_1 \; z_1^2 \quad a_1 z_1^3 \qquad {}_1 z_1^4 \quad k_2 z_2^2$$

negative term

can be rewritten by *completing the squares*

16.3.2 The Main Idea of Integrator Backstepping

Completing the squares:

$$\left(\frac{1}{2\sqrt{\ \ {}_1}}x \quad \sqrt{\ \ {}_1}\,y\right)^2 \quad \frac{1}{4\ \ {}_1}x^2 \quad xy \quad {}_1y^2 \quad 0$$

$$xy \quad {}_1y^2 \quad \left(\frac{1}{2\sqrt{\ \ {}_1}}x \quad \sqrt{\ \ {}_1}\,y\right)^2 \quad \frac{1}{4\ \ {}_1}x^2$$

$x \quad a_1z_1$ and $y \quad z_1^2$, yields:

$$V_2 \quad a_0 \quad a_2|z_1| \quad k_1 \quad z_1^2 \quad a_1z_1^3 \quad {}_1z_1^4 \quad k_2z_2^2$$

$$V_2 \quad \left(\frac{a_1}{2\sqrt{\ \ {}_1}}z_1 \quad \sqrt{\ \ {}_1}\,z_1^2\right)^2 \quad \frac{a_1^2}{4\ \ {}_1}z_1^2 \quad a_0 \quad a_2|z_1| \quad k_1 \quad z_1^2 \quad k_2z_2^2$$

$$0 \qquad\qquad 0$$

$$V_2 \quad \left(a_0 \quad k_1 \quad \frac{a_1^2}{4\ \ {}_1}\right)z_1^2 \quad k_2z_2^2$$

$$V_2 \quad 0 \quad \text{if} \quad {}_1 \quad 0, \quad k_1 \quad \frac{a_1^2}{4\ \ {}_1} \quad a_0, \quad k_2 \quad 0$$

bad damping is dominated by nonlinear damping

$$u \quad {}_1 \quad k_2z_2 \quad z_1$$

$${}_1 \quad k_1z_1 \quad {}_1z_1^3$$

Controller is implemented without using the unknown parameters a_0, a_1, a_2

ROBUST CONTROL

16.3.3 Backstepping of a SISO Mass-Damper-Spring System

$$\dot{x} = v$$
$$m\dot{v} + d(v)v + k(x)x = \tau$$
$$y = x$$

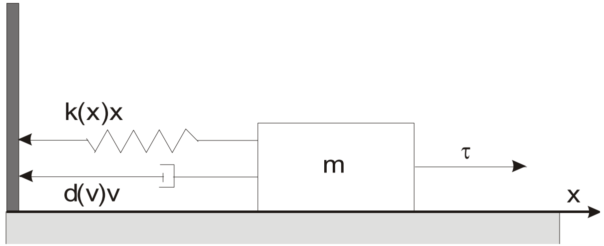
m	mass (positive)
$d\ v$	nonlinear damping force (non-negative)
$k\ x$	nonlinear spring force (non-negative)

Backstepping of the mass-damper-spring can be performed by choosing the output:

$$e \quad y \quad y_d$$

Change of coordinates

$$\dot{e} = v - \dot{y}_d$$
$$m\dot{v} = \tau - d(v)v - k(x)x$$



16.3.3 Backstepping of a SISO Mass-Damper-Spring System

Step 1:

Let $z_1 = e = y - y_d$, such that:

$\dot{z}_1 \quad v \quad y_d$ choosing the virtual control as $v \quad {}_1 \quad z_2$

where z_2 is a new state variable to be interpreted later. This yields:

$\dot{z}_1 \quad {}_1 \quad z_2 \quad y_d$ (leave z_2-term to Step 2)

Next, the stabilizing function is chosen as linear + nonlinear damping:

$${}_1 \quad y_d \quad k_1 \quad n_1 \; z_1 \quad z_1$$

Resulting dynamics Step 1:

$$\dot{z}_1 \quad k_1 \quad n_1 \; z_1 \quad z_1 \quad z_2$$

CLF:

$$V_1 \quad \frac{1}{2} z_1^2$$
$$V_1 \quad z_1 \dot{z}_1$$
$$k_1 \quad n_1 \; z_1 \quad z_1^2 \quad z_1 z_2$$

16.3.3 Backstepping of a SISO Mass-Damper-Spring System

Step 2:
The second step stabilizes the z_2 dynamics

$$mv \qquad d\ v\ v \quad k\ x\ x \quad \xrightarrow{v \qquad 1 \quad z_2}$$

$$m\dot{z}_2 \quad mv \quad m \;\; 1$$
$$d\ v\ v \quad k\ x\ x \quad m \;\; 1$$

CLF motivated by "pseudo kinetic energy"

$$V_2 \quad V_1 \quad \frac{1}{2}mz_2^2, \quad m \quad 0$$
$$V_2 \quad V_1 \quad mz_2\dot{z}_2$$
$$k_1 \quad n_1 \; z_1 \;\; z_1^2 \quad z_1z_2 \quad z_2 \qquad d\ v\ v \quad k\ x\ x \quad m \;\; 1$$

The control appears in the second step (system with 2 integrators). Hence

$$m \;\; 1 \quad d\ v\ v \quad k\ x\ x \quad z_1 \quad k_2z_2 \quad n_2 \; z_2 \; z_2$$

- includes nonlinear damping
- cancel all nonlinearities

This yields:

$$V_2 \qquad k_1 \quad n_1 \; z_1 \;\; z_1^2 \qquad k_2 \quad n_2 \; z_2 \;\; z_2^2$$

16.3.3 Backstepping of a SISO Mass-Damper-Spring System

When implementing the control law, $_1$ is computed by taking the time derivative of α_1 along the trajectories of y_d and z_1

$$_1 \quad y_d \quad k_1 \quad n_1 \; z_1 \quad z_1$$

$$_1 \quad \frac{1}{y_d}\ddot{y}_d \quad \frac{1}{z_1}\dot{z}_1 \quad \ddot{y}_d \quad \frac{1}{z_1} \; v \quad y_d$$

does not depend on the derivative of the system states (only derivatives of smooth reference signals)

Resulting Error Dynamics (use this to verify your design)

$$\begin{bmatrix} 1 & 0 \\ 0 & m \end{bmatrix}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} \quad \underbrace{\begin{bmatrix} k_1 \quad n_1 \; z_1 & 0 \\ 0 & k_2 \quad n_2 \; z_2 \end{bmatrix}}_{\text{diagonal}}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{\text{skew-symmetric}}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$\mathbf{Mz} \qquad \mathbf{K} \;\; \mathbf{z} \;\; \mathbf{z} \qquad \mathbf{Sz}$$

16.3.4 Integral Action by Constant Parameter Adaptation

The integral state can be treated as a constant parameter and estimated online using adaptive control techniques.

Integral action based on “direct matching” between the disturbance and the control input:

$$x \quad v$$

$$mv \quad d\ v\ v \quad k\ x\ x \qquad w$$

$$w \quad 0$$

$$1 \quad x_d \quad k_1 z_1$$

$$d\ v \quad 1 \quad k\ x\ x \quad \hat{w} \quad m x_d \quad m k_1\ v \quad x_d \quad z_1 \quad k_2 z_2 \qquad z_1 \quad x \quad x_d$$

$\dot{\hat{w}} \quad p z_2$ parameter update law $\qquad z_2 \quad v \quad 1$

$$V_1 \quad \tfrac{1}{2} z_1^2 \quad \tfrac{1}{2p} w^2, \quad p \quad 0 \qquad V_2 \quad V_1 \quad \tfrac{1}{2} m z_2^2$$

For details see:
Fossen, Loria and Teel (2001). A Theorem for UGAS and ULES of (Passive) Nonautonomous Systems: Robust Control of Mechanical Systems and Ships. *Int. Journal of Robust and Nonlinear Control,* **JRNC-11**:95-108

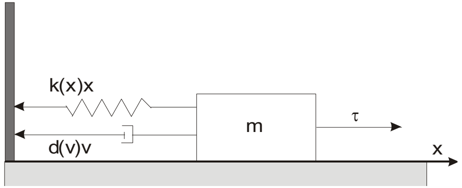
16.3.5 Integrator Augmentation Technique

Consider the 2nd-order mass-damper-spring system

$$x \quad v$$

$$mv \quad d\ v\ v \quad k\ x\ x \qquad w$$

$$y \quad x$$



where *w* is a constant unknown disturbance. Let *e* denote the tracking error

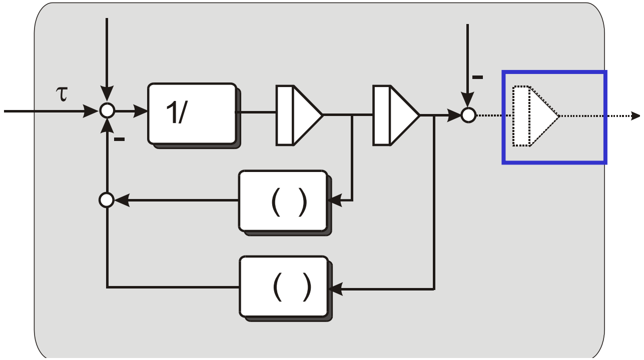
$$e \quad y \quad y_d$$

Hence, backstepping applied to the augmented model

integral state $\dot{e}_I \quad e$

$$\dot{e} \quad v \quad y_d$$

$$mv \quad d\ v\ v \quad k\ x\ x \qquad w$$



gives integral action.

16.3.6 Case Study: Backstepping Design for Mass–Damper–Spring

$$\dot{\boldsymbol{x}} = \boldsymbol{v}$$
$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{B}\boldsymbol{u}$$

Control objective: regulate the tracking errors $\mathbf{v}$ $\mathbf{v}$ $\mathbf{v}_d$ and $\mathbf{x}$ $\mathbf{x}$ $\mathbf{x}_d$ to zero.

Introduce the concept of "Vectorial Backstepping" to solve this MIMO control problem.

For the mass-damper-spring, two vectorial steps are required.

Reference:

Fossen and Berge (1997). Nonlinear Vectorial Backstepping Design for Global Exponential Tracking of Marine Vessels in the Presence of Actuator Dynamics. *Proc. IEEE Conf. on Decision and Control (CDC'97),* San Diego, CA. pp. 4237-4242.

16.3.6 Case Study: Backstepping Design for Mass–Damper–Spring

Step 1 (Vectorial Backstepping)

Consider

$$\dot{\boldsymbol{x}} = \boldsymbol{v}$$

virtual control $\boldsymbol{v} = \boldsymbol{\sigma} + \boldsymbol{\alpha}_1$

where

$\boldsymbol{\sigma} = \tilde{\boldsymbol{v}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{x}}$ New state vector used for tracking control

$\boldsymbol{\alpha}_1$ Stabilizing vector field to be defined later

Next, let us define

$$\begin{aligned} \boldsymbol{v}_r &= \boldsymbol{v} - \boldsymbol{\sigma} \\ &= \boldsymbol{v}_d - \boldsymbol{\Lambda}\tilde{\boldsymbol{v}} \end{aligned}$$

The stabilizing vector field is chosen such that:

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}} &= \boldsymbol{v} - \boldsymbol{v}_d \\ &= \boldsymbol{\sigma} + \boldsymbol{\alpha}_1 - \boldsymbol{v}_d \qquad (\boldsymbol{\alpha}_1 = \boldsymbol{v}_r = \boldsymbol{v} - \boldsymbol{\sigma}) \\ &= -\boldsymbol{\Lambda}\tilde{\boldsymbol{x}} + \boldsymbol{\sigma} \qquad \text{(leave } \boldsymbol{\sigma} \text{ to Step 2)} \end{aligned}$$

CLF

$$\boxed{\begin{aligned} \dot{V}_1 &= \tilde{\boldsymbol{x}}^\top \boldsymbol{K}_p \dot{\tilde{\boldsymbol{x}}} \\ &= \tilde{\boldsymbol{x}}^\top \boldsymbol{K}_p(-\boldsymbol{\Lambda}\tilde{\boldsymbol{x}} + \boldsymbol{\sigma}) \\ &= -\tilde{\boldsymbol{x}}^\top \boldsymbol{K}_p\boldsymbol{\Lambda}\tilde{\boldsymbol{x}} + \boldsymbol{\sigma}^\top \boldsymbol{K}_p\tilde{\boldsymbol{x}} \end{aligned}}$$

16.3.6 Case Study: Backstepping Design for Mass–Damper–Spring

The definition of the σ vector is motivated by Slotine and Li (1987) who introduced σ as a measure of tracking when designing their adaptive robot controller.

In Fossen and Berge (1997) the following state transformation is applied

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{B}\boldsymbol{u}$$

$$\xrightarrow[\boldsymbol{v}_r = \boldsymbol{v} - \boldsymbol{\sigma}]{\boldsymbol{\sigma} = \tilde{\boldsymbol{v}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{x}}}$$

$$\begin{aligned}\boldsymbol{M}\dot{\boldsymbol{\sigma}} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{\sigma} &= \boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{M}\dot{\boldsymbol{v}}_r - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}_r \\ &= \boldsymbol{B}\boldsymbol{u} - \boldsymbol{M}\dot{\boldsymbol{v}}_r - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}_r - \boldsymbol{K}(\boldsymbol{x})\boldsymbol{x}\end{aligned}$$

Step 2 (Vectorial Backstepping)

Consider a CLF based on "pseudo-kinetic energy"

$$V_2 = \frac{1}{2}\boldsymbol{\sigma}^\top \boldsymbol{M}\boldsymbol{\sigma} + V_1, \qquad \boldsymbol{M} = \boldsymbol{M}^\top > 0$$

$$\begin{aligned}\dot{V}_2 &= \boldsymbol{\sigma}^\top \boldsymbol{M}\dot{\boldsymbol{\sigma}} + \dot{V}_1 \\ &= \boldsymbol{\sigma}^\top(\boldsymbol{B}\boldsymbol{u} - \boldsymbol{M}\dot{\boldsymbol{v}}_r - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}_r - \boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{\sigma}) - \tilde{\boldsymbol{x}}^\top \boldsymbol{K}_p \boldsymbol{\Lambda}\tilde{\boldsymbol{x}} + \boldsymbol{\sigma}^\top \boldsymbol{K}_p\tilde{\boldsymbol{x}} \\ &= \boldsymbol{\sigma}^\top(\boldsymbol{B}\boldsymbol{u} - \boldsymbol{M}\dot{\boldsymbol{v}}_r - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}_r - \boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{\sigma} + \boldsymbol{K}_p\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}^\top \boldsymbol{K}_p \boldsymbol{\Lambda}\tilde{\boldsymbol{x}}\end{aligned}$$

16.3.6 Case Study: Backstepping Design for Mass–Damper–Spring

$$\dot{V}_2 = \boldsymbol{\sigma}^\top(\boldsymbol{B}\boldsymbol{u} - \boldsymbol{M}\dot{\boldsymbol{v}}_r - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}_r - \boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{\sigma} + \boldsymbol{K}_p\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}^\top\boldsymbol{K}_p\boldsymbol{\Lambda}\tilde{\boldsymbol{x}}$$

Control law

$$\boldsymbol{B}\boldsymbol{u} = \boldsymbol{M}\dot{\boldsymbol{v}}_r + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}_r + \boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} - \boldsymbol{K}_p\tilde{\boldsymbol{x}} - \boldsymbol{K}_d\boldsymbol{\sigma}, \quad \boldsymbol{K}_d > 0$$

$$\dot{V}_2 = -\boldsymbol{\sigma}^\top(\boldsymbol{D}(\boldsymbol{v}) + \boldsymbol{K}_d)\boldsymbol{\sigma} - \tilde{\boldsymbol{x}}^\top\boldsymbol{K}_p\boldsymbol{\Lambda}\tilde{\boldsymbol{x}}$$

Since V_2 is positive definite and $\dot{V}_2$ is negative definite it follows from Theorem A.3 that the equilibrium point

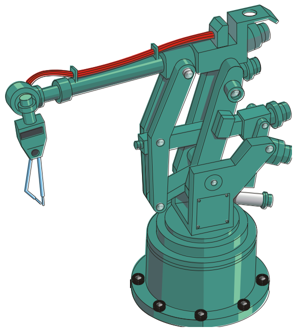
$$(\tilde{\boldsymbol{x}}, \boldsymbol{\sigma}) = (\boldsymbol{0}, \boldsymbol{0})$$

is GES.

16.3.7 Case Study: Backstepping Design for Robot Manipulators

Consider the nonlinear robot manipulator model

$$\begin{aligned} \dot{\boldsymbol{q}} &= \boldsymbol{v} \\ \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{q}) &= \boldsymbol{\tau} \end{aligned}$$



$$\dot{\boldsymbol{q}} = \boldsymbol{v} := \boldsymbol{\sigma} + \boldsymbol{\alpha}_1$$

$$\boldsymbol{\alpha}_1 = \boldsymbol{v}_r, \qquad \boldsymbol{v}_r = \boldsymbol{\nu}_d - \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}$$

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}}_r + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{v})\boldsymbol{v}_r + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_d\boldsymbol{\sigma} - \boldsymbol{K}_q\tilde{\boldsymbol{q}}$$

CLF

$$V = \frac{1}{2}\left(\boldsymbol{\sigma}^\top \boldsymbol{M}(\boldsymbol{q})\boldsymbol{\sigma} + \tilde{\boldsymbol{q}}^\top \boldsymbol{K}_q \tilde{\boldsymbol{q}}\right)$$

$$\dot{V} = -\boldsymbol{\sigma}^\top \boldsymbol{K}_d \boldsymbol{\sigma} - \tilde{\boldsymbol{q}}^\top \boldsymbol{K}_q \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}$$

16.3.8 Case Study: Backstepping Design for Surface Craft

$$\boldsymbol{\tau} = \boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu}_r + \boldsymbol{g}(\boldsymbol{\eta}) - \boldsymbol{J}_\Theta^\top(\boldsymbol{\eta})\boldsymbol{K}_p\tilde{\boldsymbol{\eta}} - \boldsymbol{J}_\Theta^\top(\boldsymbol{\eta})\boldsymbol{K}_d\boldsymbol{\sigma}$$
$$\boldsymbol{u} = \boldsymbol{B}^\dagger\boldsymbol{\tau}$$

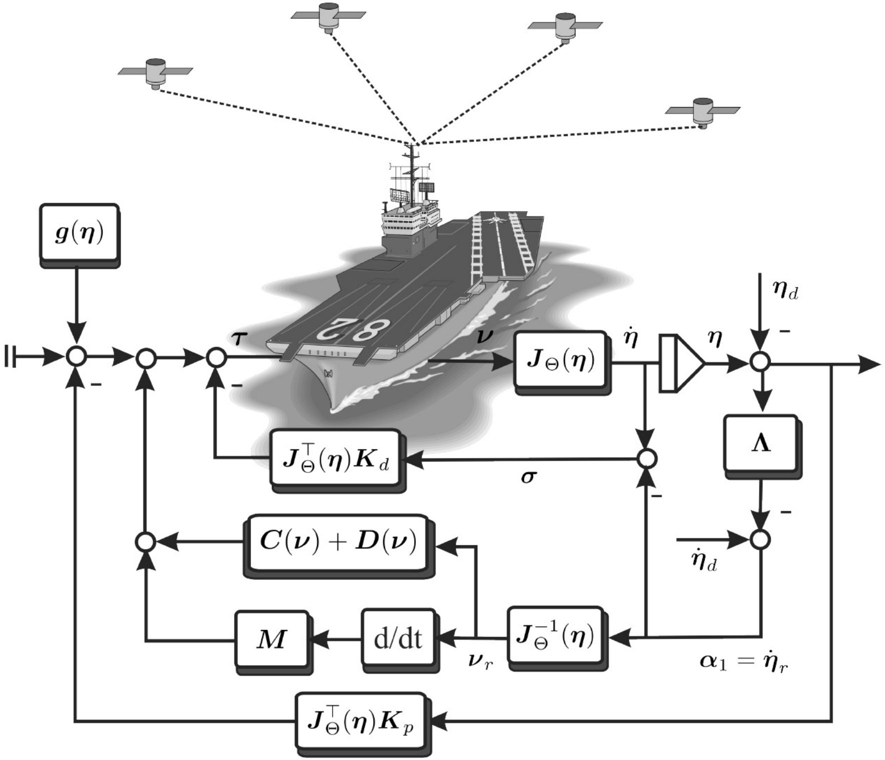
$$\boldsymbol{\sigma} = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_r = \dot{\tilde{\boldsymbol{\eta}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\eta}}$$

$$\dot{\boldsymbol{\eta}}_r := \dot{\boldsymbol{\eta}}_d - \boldsymbol{\Lambda}\tilde{\boldsymbol{\eta}}$$
$$\boldsymbol{\nu}_r := \boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}_r$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}$$
$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}$$
$$\boldsymbol{\tau} = \boldsymbol{B}\boldsymbol{u}$$

CLF

$$V_2 = \frac{1}{2}\boldsymbol{\sigma}^\top\boldsymbol{M}^*(\boldsymbol{\eta})\boldsymbol{\sigma} + V_1$$
$$\dot{V}_2 = -\boldsymbol{\sigma}^\top(\boldsymbol{D}^*(\boldsymbol{\nu},\boldsymbol{\eta}) + \boldsymbol{K}_d)\boldsymbol{\sigma} - \tilde{\boldsymbol{\eta}}^\top\boldsymbol{K}_p\boldsymbol{\Lambda}\tilde{\boldsymbol{\eta}}$$



16.3.9 Case Study: Autopilot Based on Backstepping

The linear Nomoto models can be extended to include nonlinear effects by adding static nonlinearities referred to as maneuvering characteristics.

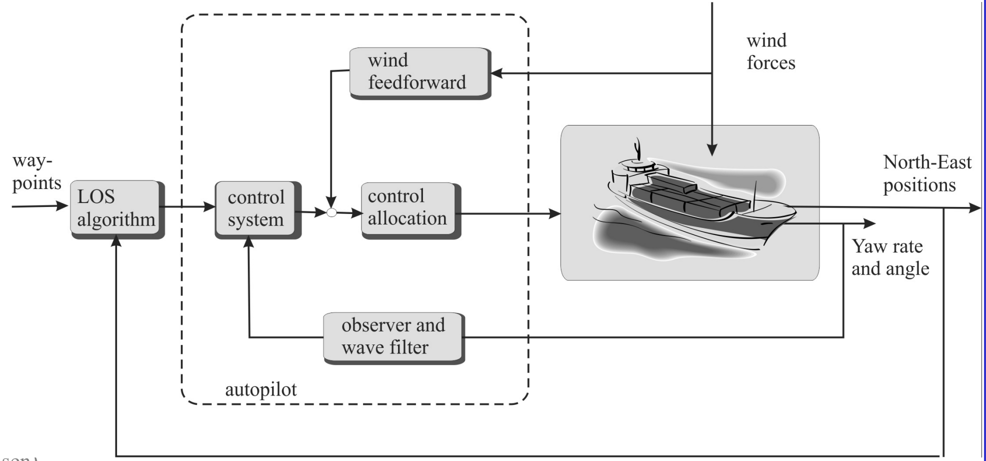
Nonlinear Extension of Nomoto's 1st-Order Model (Norrbin 1963)

$$T r \quad H_N\ r \quad\quad K$$

$$H_N\ r \quad\quad n_3 r^3 \quad n_2 r^2 \quad n_1 r \quad n_0$$

where $H_N(r)$ is a nonlinear function describing the maneuvering characteristics.
For $H_N(r) = r$, the linear model is obtained.

$$\frac{r}{\ } \ s \quad\quad \frac{K}{1 \quad Ts}$$



16.3.9 Case Study: Autopilot Based on Backstepping

A nonlinear backstepping controller can be designed by writing the autopilot model according to

$$r$$
$$mr \quad d\ r\ r$$

$$m \quad \frac{T}{K}$$
$$d\ r \quad \frac{1}{K}H_{\mathrm{N}}\ r$$

SISO mass-damper system that follows the standard procedure in Section 16.3.3.

$$z_1 \qquad d$$
$$z_2 \quad r \quad 1$$

$$m \ \ 1 \quad d\ r\ r \quad z_1 \quad k_2 z_2 \quad n_2\ z_2\ z_2$$
$$1 \quad r_d \quad k_1 \quad n_1\ z_1 \quad z_1$$

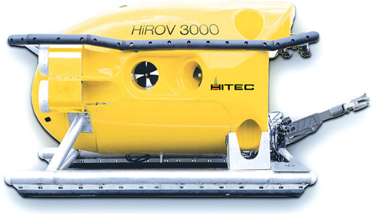
SISO Backstepping Controller (Section 16.3.3)

$$V_2 \quad V_1 \quad \frac{1}{2}mz_2^2, \quad m \quad 0$$
$$V_2 \quad V_1 \quad mz_2\dot{z}_2$$
$$k_1 \quad n_1\ z_1 \quad z_1^2 \quad z_1 z_2 \quad z_2 \quad d\ v\ v \quad k\ x\ x \quad m \ \ 1$$

$$m \ \ 1 \quad d\ v\ v \quad k\ x\ x \quad z_1 \quad k_2 z_2 \quad n_2\ z_2\ z_2$$
$$1 \quad y_d \quad k_1 \quad n_1\ z_1 \quad z_1$$

Chapter 16 – Advanced Motion Control Systems

16.1 Linear-Quadratic Optimal Control
16.2 State Feedback Linearization
16.3 Integrator Backstepping
16.4 Sliding Mode Control



Chapter 16.4 Sliding Mode Control

Sliding mode control (Utkin 1977) is a robust design method, which incorporates techniques to handle

- Structured (parametric) uncertainties
- Unstructured uncertainties (unmodeled dynamics)

Robustness is achieved by altering the dynamics of the nonlinear system by application of a discontinuous control signal. The state-feedback control law is not a continuous function of time. Instead, it can switch from one continuous structure to another based on the current position in the state space. Hence, sliding mode control is a variable structure control method.

This section considers the following sliding mode methods and their application to marine craft:

- Conventional integral sliding mode control (SMC)
- Adaptive-gain super-twisting algorithm (STA)

Sliding mode techniques are discussed by Slotine and Li (1991), Utkin (1992) and Shtessel et al. (2014) while applications to marine craft are found in Yoerger and Slotine (1985), Healey and Lienard (1993) and McGookin et al. (2000a, 2000b), for instance.

16.4.1 Conventional Integral SMC for 2nd-order Systems

Consider the 2nd-order nonlinear system

$$\ddot{x} = f(\dot{x}, x) + bu + d \qquad |d| \le d^{\max}$$

Unknown bounded disturbance with known bound d^{max}

The sliding surface S is defined as

$$S := \{x : \sigma(x) = 0\}$$

and *sliding mode* takes place if the states x evolve with time such that $\sigma(x(t_r)) = 0$ for some finite $t_r \in \mathbb{R}^+$ and $\sigma(x(t)) = 0$ for all $t > t_r$.

The feedback control law is modified (change of coordinates) to use a sliding variable σ instead of the states x. Consider

$$\boxed{\sigma := \dot{\tilde{x}} + 2\lambda\tilde{x} + \lambda^2 \int_0^t \tilde{x}(\tau)\,\mathrm{d}\tau}$$

$\lambda > 0$ is a design parameter reflecting the bandwidth of the controller

$\tilde{x} = x - x_d$

For $\sigma = 0$ this expression describes a sliding surface (manifold) with exponentially stable error dynamics. This ensures that the tracking error $\tilde{x} \rightarrow 0$ on the manifold $\sigma = 0$.

Consequently, the control objective is reduced to finding a nonlinear control law which ensures that

$$\lim_{t\to\infty} \sigma = 0$$

16.4.1 Conventional Integral SMC for 2nd-order Systems

Sliding variable (PID control)

$$\sigma := \dot{\tilde{x}} + 2\lambda\tilde{x} + \lambda^2 \int_0^t \tilde{x}(\tau)\,\mathrm{d}\tau \qquad \tilde{x} = x - x_d$$

Define

$$\sigma_0 := \tilde{x} + \lambda \int_0^t \tilde{x}(\tau)\,\mathrm{d}\tau$$

such that $\sigma = 0$ can be rewritten as

$$\sigma = \dot{\sigma}_0 + \lambda\sigma_0 = 0$$

and

$$\boxed{\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\sigma}_0 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ 0 & -\lambda \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \sigma_0 \end{bmatrix}}$$

Hence, both eigenvalues are at $-\lambda$

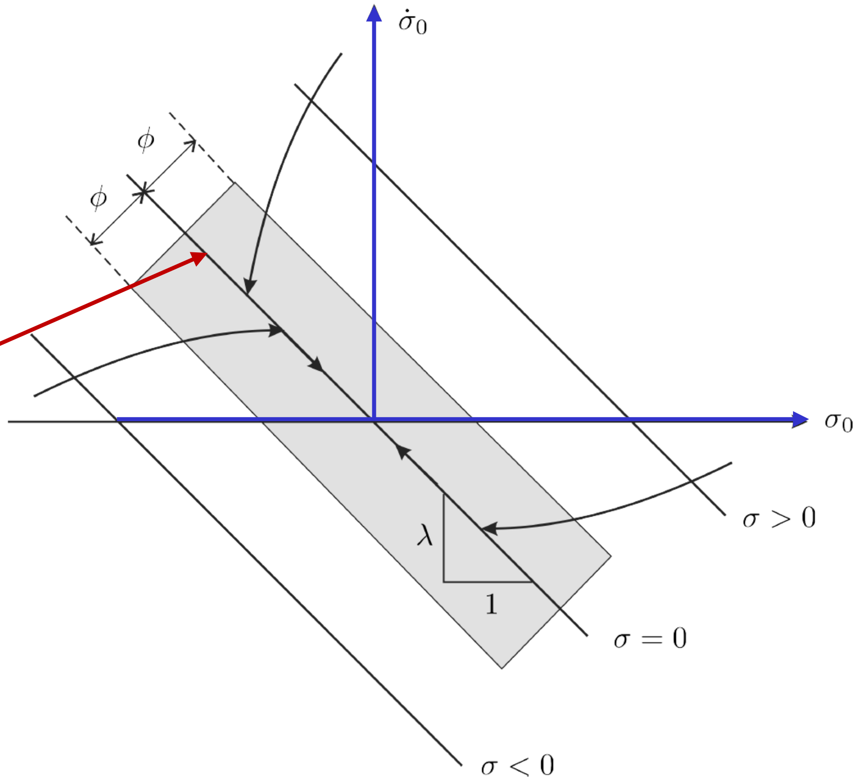


Figure 16.25: Graphical interpretation of the sliding surface $S = \{\tilde{x} : \sigma(\tilde{x}) = 0\}$, the sliding variable $\sigma = \dot{\sigma}_0 + \lambda\sigma_0$ and boundary layer $\phi > 0$.

16.4.1 Conventional Integral SMC for 2nd-order Systems

Consider the CLF

$$
\begin{aligned}
V &= \frac{1}{2}\sigma^2 \\
\dot{V} &= \sigma\dot{\sigma} \\
&= \sigma\left(f(\dot{x},x) + bu + d - \ddot{x}_d + 2\lambda\dot{\tilde{x}} + \lambda^2\tilde{x}\right)
\end{aligned}
$$

Choosing the integral SMC as

$$
\boxed{u = \frac{1}{b}\left(\ddot{x}_d - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x} - f(\dot{x},x) - K_\sigma \mathrm{sgn}(\sigma)\right)}
$$

gives

$$
\begin{aligned}
\dot{V} &= \sigma d - K_\sigma|\sigma| \qquad K_\sigma > d^{\max} \\
&< 0, \quad \forall \sigma \neq 0
\end{aligned}
$$

Hence, the equilibrium point $\sigma = 0$ is GAS and

$\tilde{x} = x - x_d$ converges to zero on the sliding surface.

Why is $K_\sigma \mathrm{sat}(\sigma)$ a saturated PID controller?

- For small σ: PID controller equal to $(K_\sigma/\phi)\ \sigma$
- For large σ: Saturating limit is K_σ

$$
\sigma := \dot{\tilde{x}} + 2\lambda\tilde{x} + \lambda^2\int_0^t \tilde{x}(\tau)\,\mathrm{d}\tau
$$

$$
\ddot{x} = f(\dot{x},x) + bu + d \qquad |d| \leq d^{\max}
$$

$$
\mathrm{sgn}(\sigma) := \begin{cases} 1 & \text{if } \sigma > 0 \\ 0 & \text{if } \sigma = 0 \\ -1 & \text{otherwise} \end{cases}
$$

chattering

It is well known that the switching term $K_\sigma \mathrm{sgn}(\sigma)$ can lead to chattering for large values of K_σ. Chattering in the controller can be eliminated by replacing the signum function with a saturating function. Slotine and Li (1991) suggest smoothing out the control discontinuity inside a boundary layer according to

$$
\mathrm{sat}(\sigma) = \begin{cases} \mathrm{sgn}(\sigma) & \text{if } |\sigma/\phi| > 1 \\ \sigma/\phi & \text{otherwise} \end{cases}
$$

Another possibility is to replace $K_\sigma \mathrm{sgn}(\sigma)$ with $K_\sigma \tanh(\sigma/\phi)$.

16.4.2 Conventional Integral SMC for 3rd-order Systems

It is straightforward to extend the results of the previous section to 3rd-order systems in the form

$$x^{(3)} = f(\ddot{x}, \dot{x}, x) + bu + d \qquad |d| \leq d^{\max}$$

Sliding variable

$$\sigma = \ddot{\tilde{x}} + 3\lambda\dot{\tilde{x}} + 3\lambda^2\tilde{x} + \lambda^3 \int_0^t \tilde{x}(\tau)\,\mathrm{d}\tau$$

CLF

$$\begin{aligned} V &= \frac{1}{2}\sigma^2 \\ \dot{V} &= \sigma\dot{\sigma} \\ &= \sigma\left(f(\ddot{x}, \dot{x}, x) + bu + d - x_d^{(3)} + 3\lambda\ddot{\tilde{x}} + 3\lambda^2\dot{\tilde{x}} + \lambda^3\tilde{x}\right) \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \dot{V} &= \sigma d - K_\sigma|\sigma| \\ &< 0, \quad \forall\sigma \neq 0 \end{aligned}$$

Control law

$$\boxed{u = \frac{1}{b}\left(x_d^{(3)} - 3\lambda\ddot{\tilde{x}}_d - 3\lambda^2\dot{\tilde{x}} - \lambda^3\tilde{x} - f(\ddot{x}, \dot{x}, x) - K_\sigma \mathrm{sgn}(\sigma)\right)}$$

$$K_\sigma > d^{\max}$$

Hence, the equilibrium point $\sigma = 0$ is GAS and $\tilde{x} = x - x_d$ converges to zero on the sliding surface.

16.4.3 Super-Twisting Adaptive Sliding Mode Control

One of the most powerful second-order continuous sliding mode control algorithms is the **super-twisting algorithm (STA)** that handles a relative degree equal to one (Levant 2003, Levant 2005, Shtessel et al. 2007, Shtessel et al. 2010).

The STA generates a continuous control signal that drives the sliding variable σ and its derivative $d\sigma/dt$ to zero in finite time in the presence of matched disturbances, which are assumed to be upper bounded.

The conventional SMC can robustly handle such a problem but you need to know the upper bound on the disturbance. On the contrary, the adaptive-gain STA

- does not need to know the upper bound on the disturbance, but it must exist
- chattering due to the sgn function is attenuated

Nonlinear system model

$$\boxed{x^{(n)} = f(\boldsymbol{x}, t) + b(\boldsymbol{x}, t)u + d}$$

$$\boldsymbol{x} = [x,\ \dot{x},\ ...,\ x^{(n-1)}]^{\top} \in \mathbb{R}^n$$

$$u \in \mathbb{R}$$

$d \in \mathbb{R}$ Bounded disturbance with unknown bound

Sliding variables for 2nd and 3rd order systems

$$(n=2) \qquad \sigma = \dot{\tilde{x}} + \lambda\tilde{x}$$

$$(n=3) \qquad \sigma = \ddot{\tilde{x}} + 2\lambda\dot{\tilde{x}} + \lambda^2\tilde{x}$$

$$\sigma := x^{(n-1)} - x_r^{(n-1)}$$

$$(n=2) \qquad \dot{x}_r = \dot{x}_d - \lambda\tilde{x}$$

$$(n=3) \qquad \ddot{x}_r = \ddot{x}_d - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x}$$

16.4.3 Super-Twisting Adaptive Sliding Mode Control

$$\sigma := x^{(n-1)} - x_r^{(n-1)}$$

$$\dot{\sigma} = \underbrace{f(\boldsymbol{x},t) + d - \dot{x}_r^{(n)}}_{\phi(\boldsymbol{x},t)} + b(\boldsymbol{x},t)u$$

$$\boxed{\begin{aligned}\dot{\sigma} &= \phi(\boldsymbol{x},t) + b(\boldsymbol{x},t)u \\ &= \phi(\boldsymbol{x},t) + \omega, \qquad u = b^{-1}(\boldsymbol{x},t)\omega\end{aligned}}$$

for some $\delta > 0$ and $b(\boldsymbol{x},t) \neq 0$ for all $\boldsymbol{x}$ and $t \in [0,\infty)$

Theorem 16.2 (Adaptive Super-Twisting Control Law (Shtessel* et al. *2010))
Consider the system (16.469) *and suppose that the nonlinear vector field* ϕ *satisfies* (16.468) *for some unknown constant* $\delta > 0$. *Then for any initial conditions* $\boldsymbol{x}(0)$ *the sliding surface* $\sigma = 0$ *will be reached in finite time for the adaptive-gain STA*

$$\omega = -\alpha|\sigma|^{1/2}\,\mathrm{sgn}(\sigma) + v \qquad (16.470)$$

$$\dot{v} = -\beta\,\mathrm{sgn}(\sigma) \qquad (16.471)$$

where the adaptive gains are

$$\dot{\alpha} = \begin{cases} \omega^* \sqrt{\frac{\gamma^*}{2}} & \textit{if } \sigma \neq 0 \\ 0 & \textit{otherwise} \end{cases} \qquad (16.472)$$

$$\beta = 2\varepsilon\alpha + \lambda^* + 4\varepsilon^2 \qquad (16.473)$$

and $\omega^* > 0$, $\gamma^* > 0$, $\lambda^* > 0$, *and* $\varepsilon > 0$ *are four constants. The stability proof guarantees that* $\sigma \to 0$ *and* $\dot{\sigma} \to 0$ *in finite time.*

Proof: *See Shtessel* et al. *(2010).*

Bounded disturbance d where the bound is assumed to be unknown!

However, you need to know that there exists a δ (no numerical value) such that

$$|\phi(\boldsymbol{x},t)| \leq \delta|\sigma|^{1/2}$$

You also need to know the nonzero function b(x,t) to compute the control input u from ω (control allocation).

However, uncertainties in b(x,t) will affect both performance and stability.

16.4.4 Case Study: Heading Autopilot Based on Conventional Integral SMC

Nonlinear yaw model (Nomoto 1963)

$$T\dot{r} + n_3 r^3 + n_1 r = K\delta + d_r + \tau_{\text{wind}} \qquad |d_r| < d_r^{\max}$$

Bounded yaw disturbance with <u>known</u> bound

Sliding variable

$$\sigma := \tilde{r} + 2\lambda\tilde{\psi} + \lambda^2 \int_0^t \tilde{\psi}(\tau)\,\mathrm{d}\tau$$

$$\sigma = r - r_r \qquad r_r := \dot{r}_d - 2\lambda\tilde{\psi} - \lambda^2 \int_0^t \tilde{\psi}(\tau)\,\mathrm{d}\tau$$

Change of coordinates

$$\begin{aligned} T\dot{\sigma} &= T\dot{r} - T\dot{r}_r \\ &= K\delta + d_r + \tau_{\text{wind}} - (n_3 r^2 + n_1) r - T\dot{r}_r \\ &= K\delta + d_r + \tau_{\text{wind}} - (n_3 r^2 + n_1)(\sigma + r_r) - T\dot{r}_r \end{aligned}$$

CLF

$$\begin{aligned} \dot{V} &= \sigma T\dot{\sigma} \\ &= \sigma[K\delta + d_r + \tau_{\text{wind}} - (n_3 r^2 + n_1)(\sigma + r_r) - T\dot{r}_r] \\ &= -(n_3 r^2 + n_1)\sigma^2 + \sigma[K\delta + d_r + \tau_{\text{wind}} - (n_3 r^2 + n_1) r_r - T\dot{r}_r] \end{aligned}$$

Control law

$$\delta = \frac{1}{\hat{K}}\left(\hat{T}\dot{r}_r + (\hat{n}_3 r^2 + n_1)\, r_r - \hat{\tau}_{\text{wind}} - K_d\sigma - K_\sigma \operatorname{sgn}(\sigma)\right)$$

The **hat** denotes the best estimate of the variable. The errors must be compensated for by increasing K_σ

Note that d_r is not in the control law. The signal is unknown, but you know it is bounded

16.4.4 Case Study: Heading Autopilot Based on Conventional Integral SMC

$$\delta = \frac{1}{\hat{K}}\left(\hat{T}\dot{r}_r + \left(\hat{n}_3 r^2 + n_1\right) r_r - \hat{\tau}_{\text{wind}} - K_d\sigma - K_\sigma \text{sgn}(\sigma)\right)$$

$$\begin{aligned}
\dot{V} &= \sigma T \dot{\sigma} \\
&= \sigma[K\delta + d_r + \tau_{\text{wind}} - (n_3 r^2 + n_1)(\sigma + r_r) - T\dot{r}_r] \\
&= -(n_3 r^2 + n_1)\sigma^2 + \sigma[K\delta + d_r + \tau_{\text{wind}} - (n_3 r^2 + n_1) r_r - T\dot{r}_r]
\end{aligned}$$

$$\begin{aligned}
\dot{V} = &-\left(n_3 r^2 + n_1 + \frac{K}{\hat{K}} K_d\right)\sigma^2 - \frac{K}{\hat{K}} K_\sigma |\sigma| \\
&+ \left[d_r + \left(\tau_{\text{wind}} - \frac{K}{\hat{K}}\hat{\tau}_{\text{wind}}\right) - \left(T - \frac{K}{\hat{K}}\hat{T}\right)\dot{r}_r \right.\\
&\left. - \left(n_3 - \frac{K}{\hat{K}}\hat{n}_3\right) r^2 r_r - \left(1 - \frac{K}{\hat{K}}\right) n_1 r_r\right]\sigma
\end{aligned}$$

For course unstable ships ($n_1 = -1$) we choose $K_d > \hat{K}/K$ while for course stable ships ($n_1 = 1$) no additional feedback is needed and thus $K_d = 0$. This guarantees

$$-\left(n_3 r^2 + n_1 + \frac{K}{\hat{K}} K_d\right)\sigma^2 < 0, \qquad \forall \sigma \neq 0$$

In order to guarantee that $\dot{V} < 0$, the switching gain K_σ must be chosen large enough

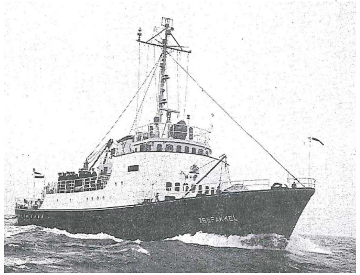
$$\begin{aligned}
K_\sigma \geq \frac{\hat{K}}{K} d_r^{\max} &+ \left|\left(\frac{\hat{K}}{K}\tau_{\text{wind}} - \hat{\tau}_{\text{wind}}\right)\right| + \left|\left(\frac{\hat{K}}{K} T - \hat{T}\right)\dot{r}_r\right| \\
&+ \left|\left(\frac{\hat{K}}{K} n_3 - \hat{n}_3\right) r^2 r_r\right| + \left|\left(\frac{\hat{K}}{K} - 1\right) n_1 r_r\right|
\end{aligned}$$

$$\boxed{K_\sigma \geq 0.2 d_r^{\max} + 0.2\,|\hat{\tau}_{\text{wind}}| + 0.2\left|\hat{T}\dot{r}_r\right| + 0.2\left|\hat{n}_3 r^2 r_r\right| + 0.2\,|n_1 r_r|}$$

20 % uncertainty in all terms

MSS Toolbox - ROVzefakkel

```matlab
function [psi_dot, r_dot,delta_dot] = ROVzefakkel(r,U,delta,delta_c,d_r)
% [rdot] = ROVzefakkel(r,U,delta,delta_c,d_r) returns the yaw acceleration,
% yaw rate and rudder angle of the Norrbin (1963) nonlinear autopilot model
%
%                    psi_dot = r                       - yaw kinematics
%  T r_dot + n3 r^3 + n1 r = K delta + d_r             - Norrbin (1963) model
%                  dot_delta = f(delta, delta_c)       - rudder dynamics
%
% for the ROV Zefakkel (Length 45 m). The ship is controlled by
% controllable pitch propeller and a rudder with rudder dynamics.
% The model parameters K, T and n3 are speed dependent, while n = 1
% (course stable ship). The parametersare interpolated for varying speeds U.
%
% Inputs:
% r      = yaw rate                 (rad/s)
% U      = speed                    (m/s)
% delta = rudder angle              (rad)
% delta_c = commanded rudder angle  (rad)
% d_r    = yaw moment disturbance   (optional)
%
% Reference : Van Amerongen, J. (1982). Adaptive Steering of Ships – A
% Model Reference Approach to Improved Maneuvering and Economical Course
% Keeping. PhD thesis. Delft University of Technology, Netherlands.
%
% Author:     Thor I. Fossen
% Date:       19 June 2020
% Revisions:
```



```matlab
% model parameters
n1 = 1; n3 = 0.4;
delta_max  = 30 * (pi/180);   % max rudder angle (rad)
Ddelta_max = 10 * (pi/180);   % max rudder rate (rad/s)

% ROV Zefakkel (Van Amerongen 1982)
data = [...                                  data = [ U K T ]
    2       0.15    33
    2.6     0.19    33
    3.6     0.29    33
    4       0.37    33
    5       0.50    31
    6.2     0.83    43 ];

% interpolate to find K and T as a function of U
K = interp1(data(:,1),data(:,2),U,'linear','extrap');
T = interp1(data(:,1),data(:,3),U,'linear','extrap');

% Rudder saturation and dynamics
if abs(delta_c) >= delta_max
   delta_c = sign(delta_c) * delta_max;
end

delta_dot = delta_c - delta;
if abs(delta_dot) >= Ddelta_max
   delta_dot = sign(delta_dot) * Ddelta_max;
end

% yaw dynamics
psi_dot = r;
r_dot = (1/T) * (K * delta + d_r - n3 * r^3 - n1 * r );
```

16.4.4 Case Study: Heading Autopilot Based on Conventional Integral SMC

Matlab:
The conventional integral SMC has been implemented in the MSS toolbox script `ExSMC.m` where a nonlinear model

$$\dot{\psi} = r \tag{16.486}$$

$$T\dot{r} + n_3 r^3 + n_1 r = K\delta + d_r \tag{16.487}$$

of the ROV Sefakkel has been used to generate the measurements. The ROV Sefakkel is 45 m long training ship (Van Amerongen 1982). The unknown disturbance term d_r is chosen as $d_r = K\delta_0$ where $\delta_0 = 1$ deg is a rudder offset caused by environmental disturbances. The numerical values for T and K for varying speeds $U \in [1, \dots 7]$ m/s are computed using interpolation and extrapolation of the ROV Sefakkel data set

```
data = [...                    % data = [ U K T ]
    2       0.15    33
    2.6     0.19    33
    3.6     0.29    33
    4       0.37    33
    5       0.50    31
    6.2     0.83    43 ];
```

For the ROV Sefakkel $n_3 = 0.4$ and $n = 1$ (course stable). The rudder dynamics is modeled using (see Section 9.5.2)

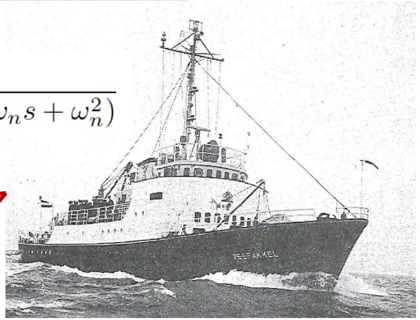
```
if abs(delta_c) >= delta_max              % saturated rudder angle
   delta_c = sign(delta_c) * delta_max;
end
delta_dot = delta_c - delta;              % saturated rudder rate
if abs(delta_dot) >= Ddelta_max
   delta_dot = sign(delta_dot) * Ddelta_max;
end
```

SSA modification

$$\sigma := \dot{\tilde{r}} + 2\lambda\, \mathrm{ssa}(\tilde{\psi}) + \lambda^2 \int_0^t \mathrm{ssa}(\tilde{\psi}(\tau))\, d\tau$$

Reference model

$$\frac{\psi_d}{\psi_{\mathrm{ref}}}(s) = \frac{\omega_n^3}{(s+\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$$

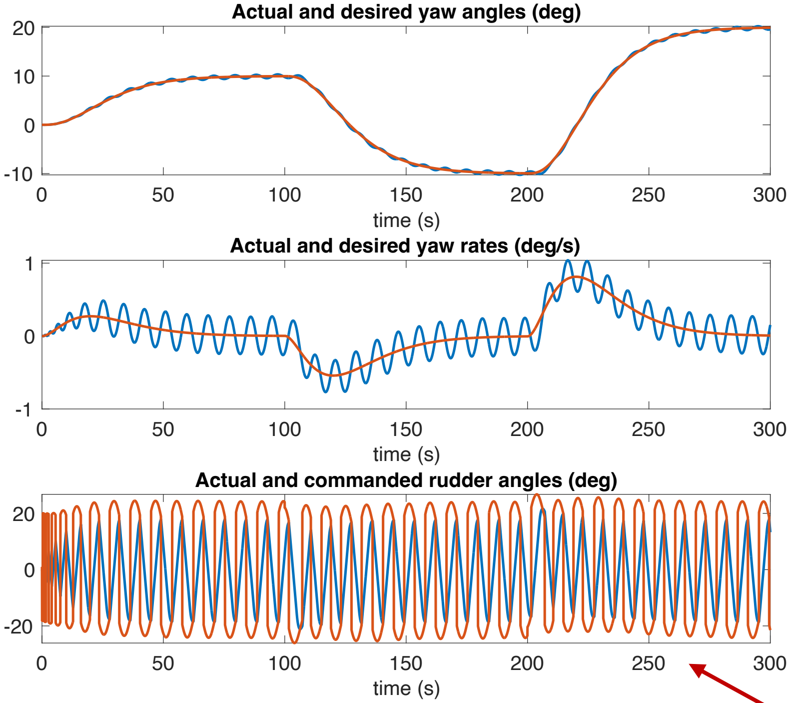


The ship model including rudder dynamics is implemented in the MSS toolbox function

```
[psi_dot,r_dot,delta_dot] = ROVzefakkel(r,U,delta,delta_c,d_r)
```

Figures 16.26–16.27 show the performance of the integral SMC for yaw angle commands ψ_{ref} which are filtered using a 3rd-order reference model for generation of ψ_d, r_d and $\dot{r}_d$ as described in Section 12.1.1. Figure 16.26 illustrates the chattering problem when using the $\mathrm{sgn}(\sigma)$ function in the feedback controller. This problem is avoided by using $\tanh(\sigma/\phi)$ instead as shown in Figure 16.27 where the control signal is smooth. A similar behavior is obtained for $\mathrm{sat}(\sigma)$. All these options can be tested by specifying the `flag` parameter equal to `1,2,3` in the script.

16.4.4 Case Study: Heading Autopilot Based on Conventional Integral SMC

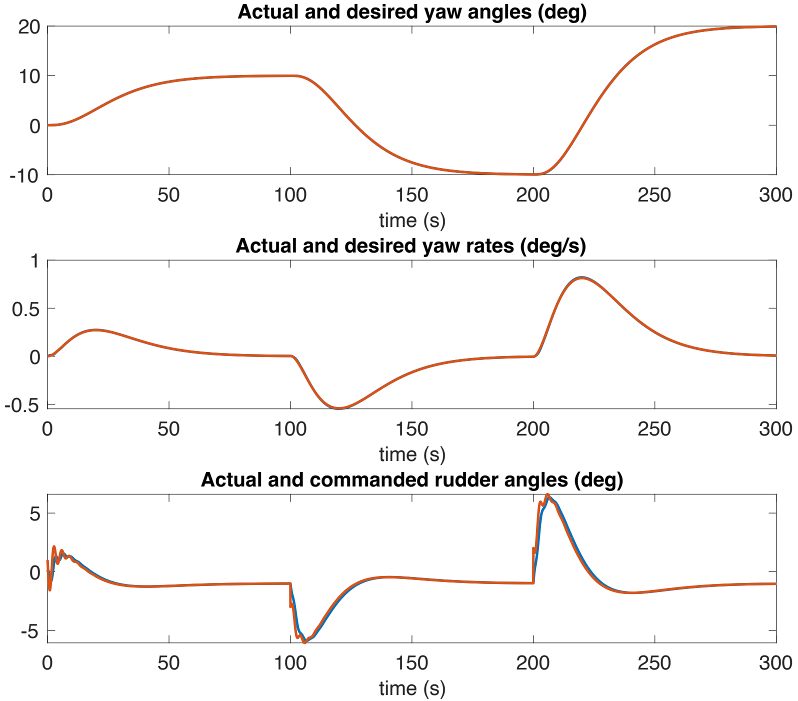


Flag = 1 demonstrates sgn(sigma)

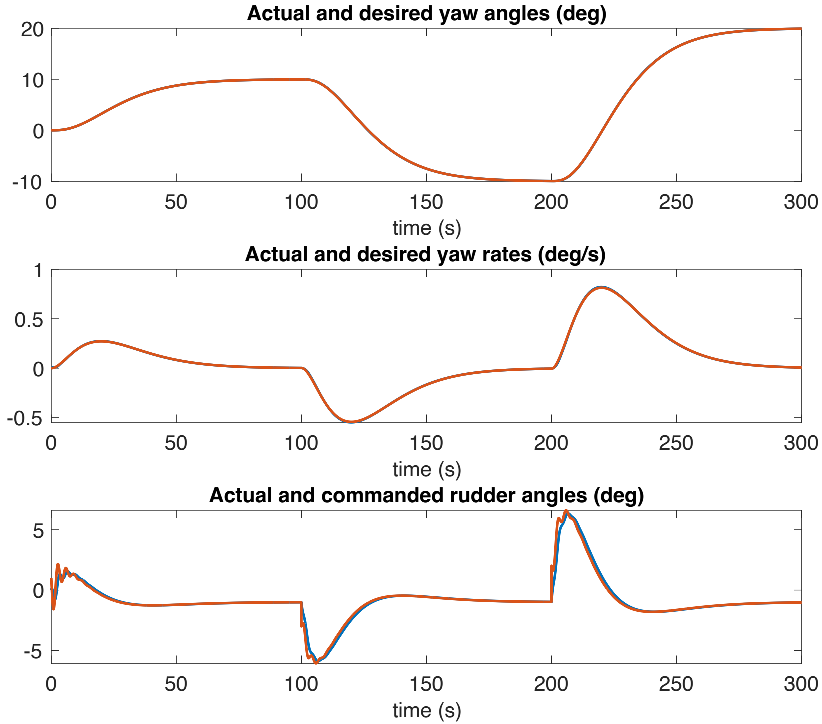
```
% ExSMC integral sliding mode control (SMC) design for heading autopilot.
% A conventional SMC is designed for the Norrbin (1963) nonlinear yaw model
%
%                        psi_dot = r
%       T r_dot + n3 r^3 + n1 r = K delta + d_r
%
% where d_r is an unknown bounded disturbance in yaw and
%
%    [psi_dot, r_dot, delta_dot] = ROVzefakkel(r,U,delta,delta_c,d_r)
%
% is the Norrbin model for the ROV Zefakkel (Length 45 m) inclduing
% actuator dynamics and saturation
%
% Author:     Thor I. Fossen
% Date:       19 June 20200
% Revisions:

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% USER INPUTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h    = 0.05;     % sampling time [s]
N  =   6000;    % no. of samples

psi_ref = 10 * pi/180;  % desired yaw angle

flag = 3;       % 1 = conventional SMC using sgn(sigma)
                % 2 = conventional SMC using tanh(sigma/phi)
                % 3 = conventional SMC using sat(sigma)
```

Chattering – wear and tear of the rudder servo.

16.4.4 Case Study: Heading Autopilot Based on Conventional Integral SMC



Flag = 2 demonstrates sgn(sigma/phi)



Flag = 3 demonstrates sat(sigma)

16.4.5 Case study: Depth Autopilot for Diving Based on Conventional Integral SMC

$$(I_y - M_{\dot q})\ddot\theta - M_q\dot\theta + \mathrm{BG}_z W\theta = M_{\delta_S}\delta_S + d_\theta$$

For "flying vehicles", the stern rudders are used to dive and control the depth.

Decoupled pitch equation

$$\ddot\theta + a_1\dot\theta + a_0\theta = b_0\delta_S + d_\theta$$

where $a_0 = \mathrm{BG}_z W/(I_y - M_{\dot q})$, $a_1 = -M_q/(I_y - M_{\dot q})$ and $b_0 = M_{\delta_S}/(I_y - M_{\dot q})$

Depth

$$\begin{aligned}\dot z^n &= -u\sin(\theta) + v\cos(\theta)\sin(\phi) + w\cos(\theta)\cos(\phi)\\ &\approx -U\theta + d_z\end{aligned}$$

θ, ϕ, v and w are small

$u \approx U = \text{constant}$

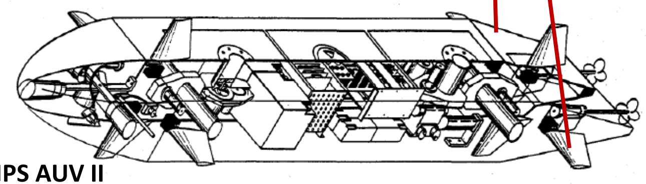
Double differentiation with respect to time

$$(z^n)^{(3)} = -U\ddot\theta + \ddot d_z$$

The resulting system (depth) is of third order

$$(z^n)^{(3)} = \underbrace{a_1U\dot\theta + a_0U\theta}_{f(\cdot)} - b_0U\delta_S + \underbrace{(\ddot d_z - Ud_\theta)}_{d}$$

We assume that the upper bound $|d| \leq d^{\max}$ is known. In practise this is a tuneable parameter

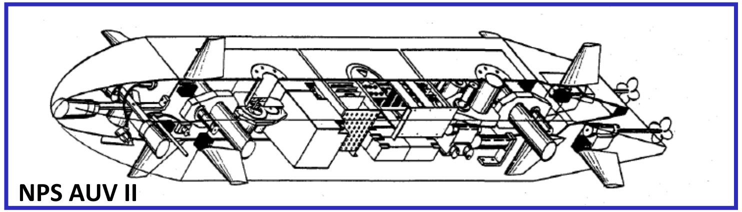


NPS AUV II

16.4.5 Case study: Depth Autopilot for Diving Based on Conventional Integral SMC

Third-order model for depth

$$(z^n)^{(3)} = \underbrace{a_1 U\dot{\theta} + a_0 U\theta}_{f(\cdot)} - b_0 U\delta_S + \underbrace{(\ddot{d}_z - U d_\theta)}_{d}$$



Sliding variable

$$\sigma = \ddot{\tilde{z}}^n + 3\lambda\dot{\tilde{z}}^n + 3\lambda^2\tilde{z}^n + \lambda^3\int_0^t \tilde{z}^n(\tau)\,\mathrm{d}\tau$$

$$\tilde{z}^n = z^n - z_d^n$$

Reference model for depth

$$\frac{z_d^n}{z_{\mathrm{ref}}^n}(s) = \frac{\omega_n^3}{(s+\omega_n)(s^2+2\zeta\omega_n s+\omega_n^2)}$$

Control law

$$\delta_S = -\frac{1}{b_0 U}\Big((z_d^n)^{(3)} - 3\lambda\ddot{\tilde{z}}_d^n - 3\lambda^2\dot{\tilde{z}}^n - \lambda^3\tilde{z}^n - a_1 U\dot{\theta} - a_0 U\theta - K_\sigma \mathrm{sgn}(\sigma)\Big)$$

$$\boxed{K_\sigma > d^{\max}} \qquad |d| \leq d^{\max}$$

Consequently, $\sigma \to 0$ and thus $\tilde{z}^n \to 0$. When implementing the control law, higher-order derivatives in z^n can be avoided by using the measured pitch angle θ and pitch rate q to compute

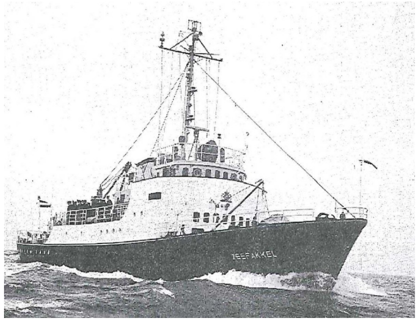
$$\dot{z}^n \approx -U\theta$$
$$\ddot{z}^n \approx -Uq$$

16.4.6 Case Study: Heading Autopilot Based on the Adaptive-Gain Super Twisting Algorithm

Nonlinear yaw model (Nomoto 1963)

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

$$T\dot{r} + n_3 r^3 + n_1 r = K\delta + d_r + \tau_{\text{wind}}$$



Sliding variable

$$\sigma := \tilde{r} + \lambda\tilde{\psi} \qquad \sigma = r - r_r \qquad r_r = r_d - \lambda\tilde{\psi}$$

$$\dot{\sigma} = \underbrace{\frac{1}{T}\left(d_r + \tau_{\text{wind}} - n_3 r^3 - n_1 r\right) - \dot{r}_r}_{\phi(\cdot)} + \underbrace{\frac{K}{T}}_{b}\,\delta$$

Control variable and its inverse

$$\omega = \frac{K}{T}\delta \quad \Longrightarrow \quad \delta = \frac{T}{K}\omega$$

Adaptive-Gain Super Twisting Algorithm

$$\boxed{\begin{aligned} \omega &= -\alpha|\sigma|^{1/2}\,\mathrm{sgn}(\sigma) + v \\ \dot{v} &= -\beta\,\mathrm{sgn}(\sigma) \end{aligned}}$$

$$\boxed{\begin{aligned} \dot{\alpha} &= \begin{cases} \alpha_0 & \text{if } \sigma \neq 0 \\ 0 & \text{otherwise} \end{cases} \\ \beta &= \beta_1\alpha + \beta_0 \end{aligned}}$$

where $\alpha_0 > 0$, $\beta_0 > 0$, and $\beta_1 > 0$ are three tunable parameters.

16.4.6 Case Study: Heading Autopilot Based on the Adaptive-Gain Super Twisting Algorithm

Matlab:
The MSS toolbox function

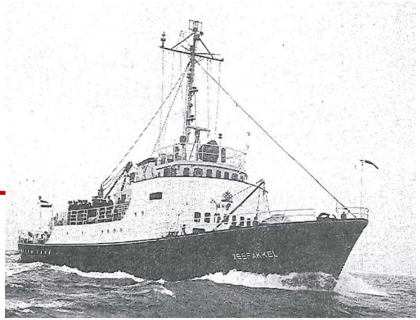
```
[psi_dot,r_dot,delta_dot] = ROVzefakkel(r,U,delta,delta_c,d_r)
```

is used to generate measurements for the adaptive-gain STA, which is implemented in the MSS toolbox example script `ExSTA.m`. The ROV Sefakkel is 45 m long training ship (Van Amerongen 1982). The unknown disturbance term d_r is chosen as $d_r = K\delta_0$ where $\delta_0 = 1$ deg is a rudder offset caused by environmental disturbances. The numerical values for T and K in the model depend on the cruise speed $U \in [1, \ldots 7]$ m/s while $n_3 = 0.4$ and $n = 1$ (course stable).

Figure 16.28 show the performance of the adaptive-gain STA for yaw angle commands ψ_{ref} which are filtered using a 3rd-order reference model for generation of ψ_d, r_d and $\dot{r}_d$ as described in Section 12.1.1. The STA is implemented using (16.507)–(16.510) with

$$\alpha_0 = 0.03, \quad \beta_0 = 0.0001, \quad \beta_1 = 0.0001$$

In addition, the sigmoid function $\text{sgn}(\sigma)$ in (16.508) has been replaced by $\tanh(\sigma/\phi)$ to further improve performance and avoid chattering. It is seen from Figure 16.28 that the controller adapts quite well by adjusting the states α and v. The control input δ is free from chattering. Also note that the control signal ω does not depend on the ship model parameters but when mapped to rudder angle commands δ_c accurate estimates of K and T are needed to implement (16.512) successfully.

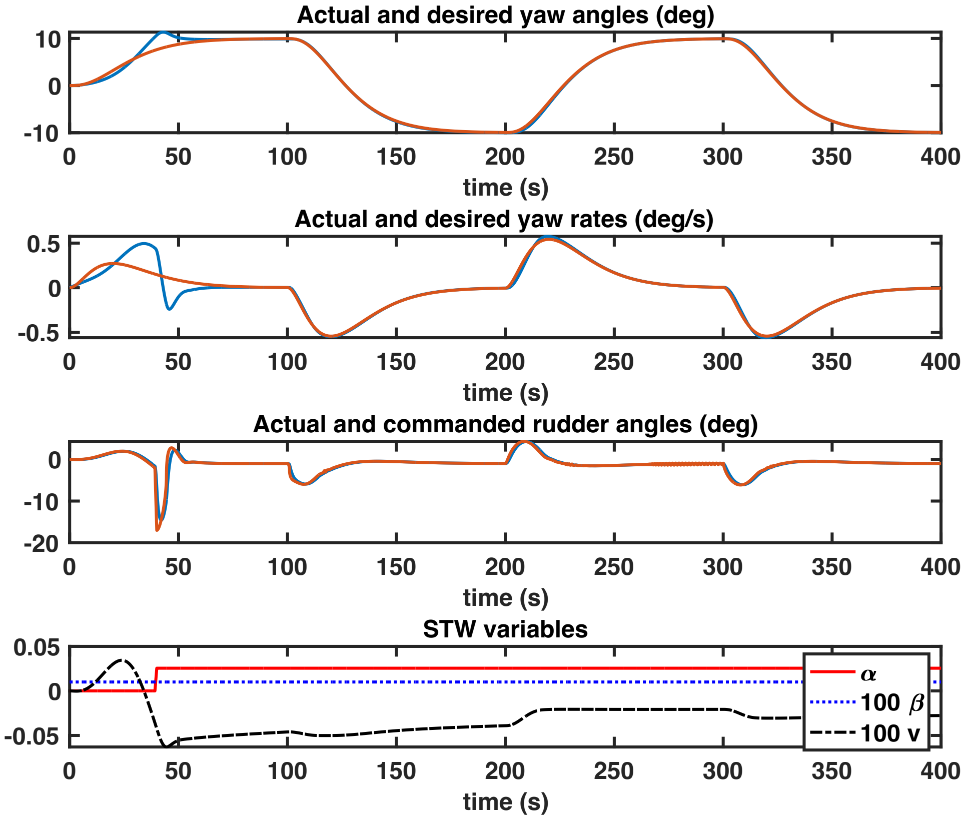


Adaptive-Gain Super Twisting Algorithm

$$\boxed{\begin{aligned} \omega &= -\alpha|\sigma|^{1/2}\,\text{sgn}(\sigma) + v \\ \dot{v} &= -\beta\,\text{sgn}(\sigma) \end{aligned}}$$

$$\boxed{\begin{aligned} \dot{\alpha} &= \begin{cases} \alpha_0 & \text{if } \sigma \neq 0 \\ 0 & \text{otherwise} \end{cases} \\ \beta &= \beta_1\alpha + \beta_0 \end{aligned}}$$

16.4.6 Case Study: Heading Autopilot Based on the Adaptive-Gain Super Twisting Algorithm



Super twisting adaptive sliding mode controller for heading control.

Lower plot shows the STW variables including the adaptive gains.

Chapter Goals – Revisited

Linear-quadratic (LQ) optimal control

- Be able to modify linear-quadratic controllers to include integral action
- Be able to extend linear-quadratic optimal controllers from setpoint regulation to trajectory-tracking control
- Be able to design linear-quadratic optimal controllers for heading control, rudder-roll-damping, and DP
- Understand how to use the ISO standards for motion sickness when designing roll-damping control systems

Feedback linearization

- Be able to design to design velocity and position trajectory-tracking controllers for marine craft by feedback linearization and pole placement.

Nonlinear backstepping

- Understand how nonlinear backstepping relates to feedback linearization. Pros and cons.
- Be able to design nonlinear backstepping controllers for heading control, speed control, DP, and other motion control scenarios.

Sliding-mode control

- Understand the difference of conventional sliding-mode control (SMC), integral sliding-mode control (ISMC), and adaptive super-twisting SMC
- Be able to design motion controllers for marine craft using ISMC and adaptive super-twisting SMC