Lecture Notes:
TTK 4190 Guidance, Navigation and Control of Vehicles

Professor Thor I. Fossen

Department of Engineering Cybernetics
NTNU Norwegian University of Science and Technology
Trondheim, Norway

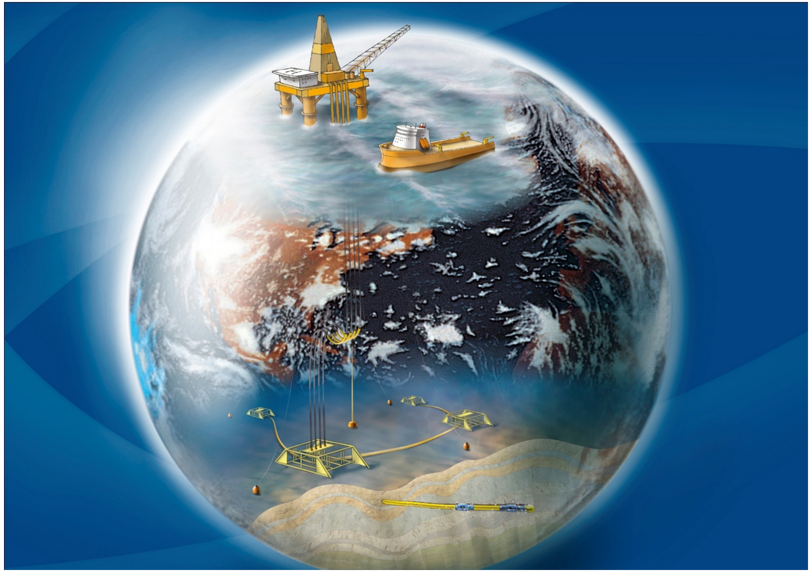
Home page: https://www.fossen.biz

What does the Marine Craft Hydrodynamics and Motion Control Lecture Notes Cover?

1. Mathematical modeling of marine craft.

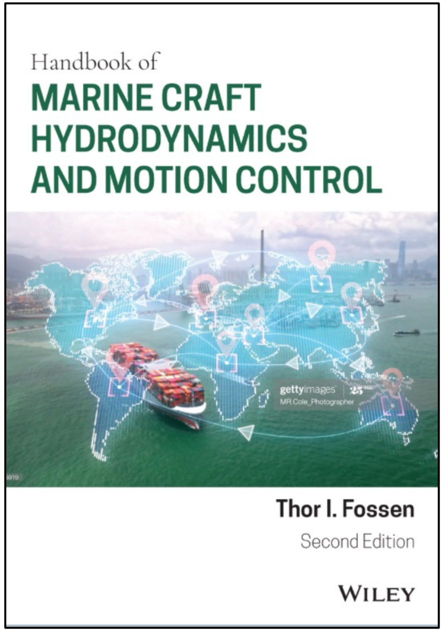
   This includes:

   - Kinematics
   - Kinetics
   - Equations of motion for marine craft
   - Wind, waves, and ocean current models
   - Hydrostatics (ships and underwater vehicles)
   - Hydrodynamics: maneuvering and seakeeping theory



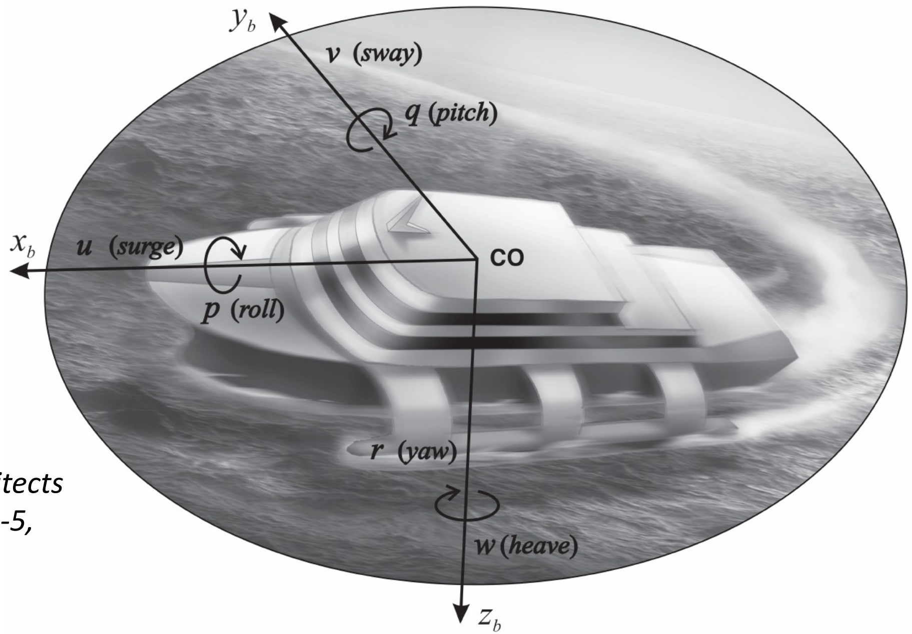
2. Design of guidance, navigation, and motion control systems for many applications including ships, semisubmersibles, and autonomous vehicles (AUVs and USVs).
3. Equations of motion for simulation of marine craft in the time-domain using advanced hydrostatic and hydrodynamic models.

Textbook and Notation



Fossen, T. I. (2021)
Handbook of Marine Craft Hydrodynamics and Motion Control (2nd ed.). Wiley.
Textbook supplements: https://wiley.fossen.biz

MSS Toolbox: https://github.com/cybergalactic/MSS
Python Vehicle Simulator: https://github.com/cybergalactic/PythonVehicleSimulator



SNAME (1950). Nomenclature for Treating the Motion of a Submerged Body Through a Fluid. *The Society of Naval Architects and Marine Engineers, Technical and Research Bulletin No. 1-5, April 1950, pp. 1-15.*

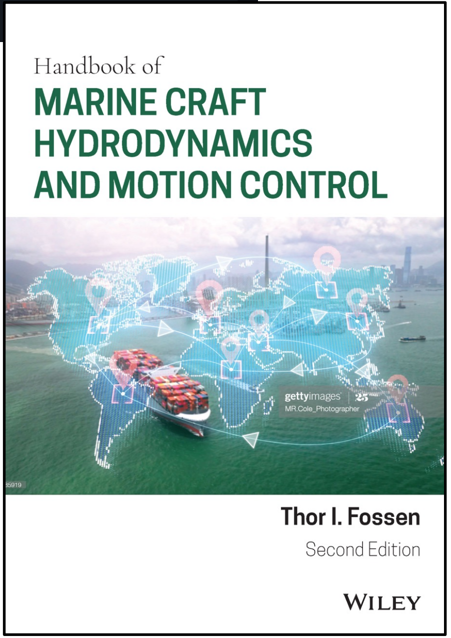
https://github.com/cybergalactic/MSS

MSS (Marine Systems Simulator)

The Marine Systems Simulator (MSS) is a Matlab and Simulink library for marine systems. It includes models for ships, underwater vehicles, unmanned surface vehicles and floating structures. The library also contains guidance, navigation, and control (GNC) blocks for real-time simulation. The algorithms are described in:

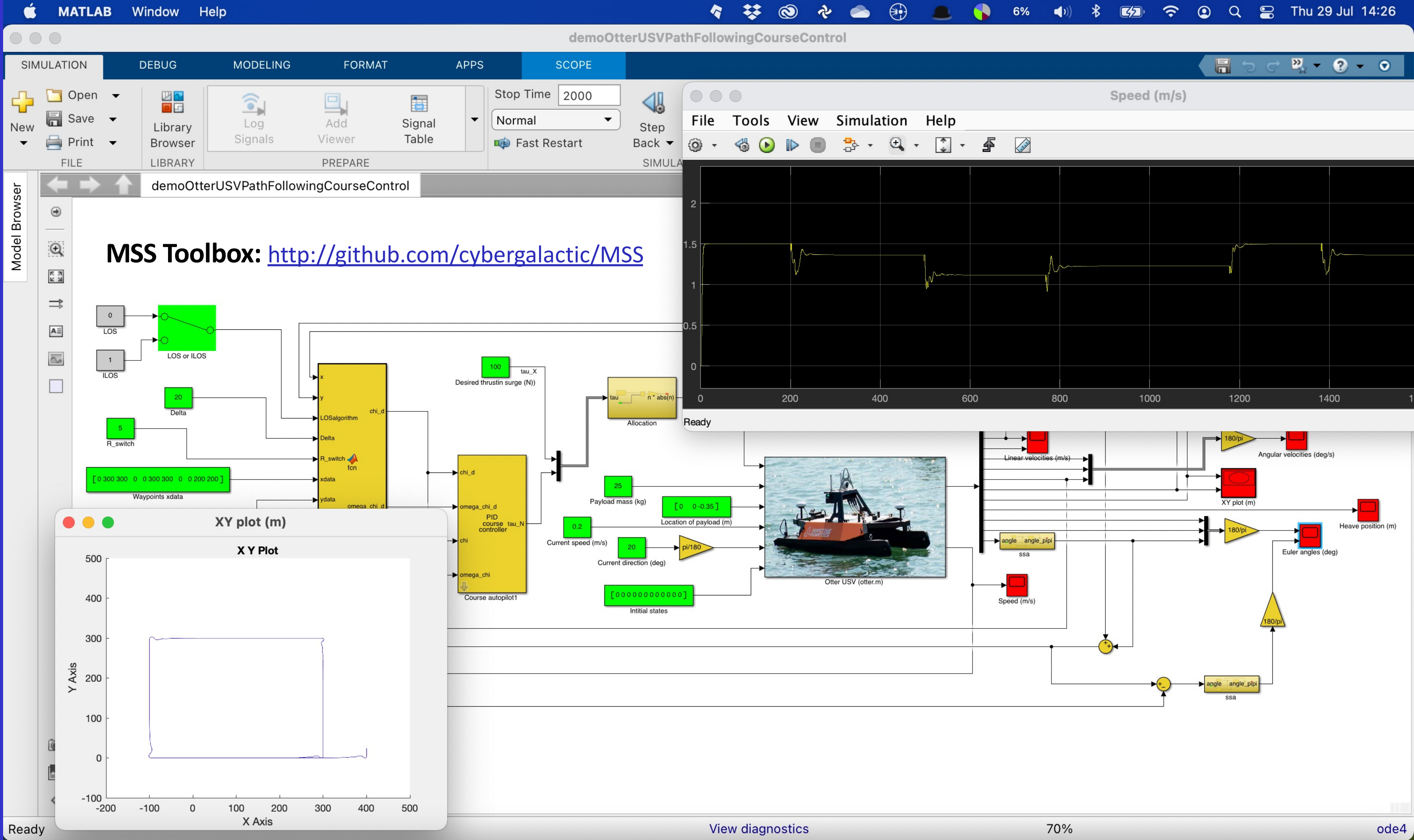
```
T. I. Fossen (2021). Handbook of Marine Craft Hydrodynamics and Motion Control. 2nd. Edition, Wiley. ISBN-13: 978-1119
Lecture notes: https://www.fossen.biz/wiley
```

```
function [xdot,U] = otter(x,n,mp,rp,V_c,beta_c)
% [xdot,U] = otter(x,n,mp,rp,V_c,beta_c) returns the speed U in m/s (optionally)
% and the time derivative of the state vector:
%    x = [ u v w p q r x y z phi theta psi ]'
% for the Maritime Robotics Otter USV, see www.maritimerobotics.com.
% The length of the USV is L = 2.0 m, while the state vector is defined as:
%
%  u:     surge velocity          (m/s)
%  v:     sway velocity           (m/s)
%  w:     heave velocity          (m/s)
%  p:     roll velocity           (rad/s)
%  q:     pitch velocity          (rad/s)
%  r:     yaw velocity            (rad/s)
%  x:     position in x direction (m)
%  y:     position in y direction (m)
%  z:     position in z direction (m)
%  phi:   roll angle              (rad)
%  theta: pitch angle             (rad)
%  psi:   yaw angle               (rad)
```



NTNU

MSS Toolbox: http://github.com/cybergalactic/MSS



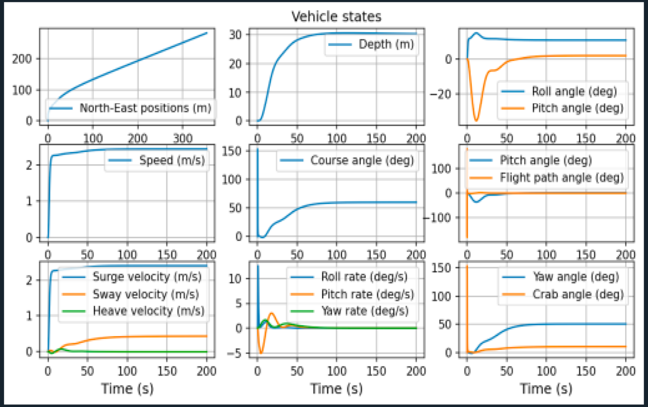
NTNU

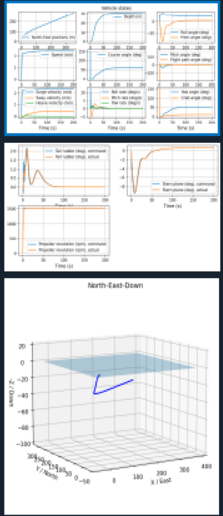
.rs/thor/Dropbox/Mac (3)/Documents/GitHub/PythonVehicleSimulator/src/python_vehicle_simulator/main.py

main.py | remus100.py | test_simulate.py

```python
numDataPoints = 50                  # number of 3D data points
FPS = 10                            # frames per second (animated GIF)
filename = '3D_animation.gif'       # data file for animated GIF
browser = 'safari'                  # browser for visualization of animated GIF

###############################################################################
# Vehicle constructors
###############################################################################
printSimInfo()

"""
DSRV('depthAutopilot',z_d)
frigate('headingAutopilot',U,psi_d)
otter('headingAutopilot',psi_d,V_c,beta_c,tau_X)
ROVzefakkel('headingAutopilot',U,psi_d)
semisub('DPcontrol',x_d,y_d,psi_d,V_c,beta_c)
shipClarke83('headingAutopilot',psi_d,L,B,T,Cb,V_c,beta_c,tau_X)
supply('DPcontrol',x_d,y_d,psi_d,V_c,beta_c)
tanker('headingAutopilot',psi_d,V_c,beta_c,depth)
remus100('depthHeadingAutopilot',z_d,psi_d,V_c,beta_c)

Call constructors without arguments to test step inputs, e.g. DSRV(), otter(), etc.
"""

no = input("Please enter a vehicle no.: ")

match no:                       #  the match statement requires Python >= 3.10
    case '1': vehicle = DSRV('depthAutopilot',60.0)
    case '2': vehicle = frigate('headingAutopilot',10.0,100.0)
    case '3': vehicle = otter('headingAutopilot',100.0,0.3,-30.0,200.0)
    case '4': vehicle = ROVzefakkel('headingAutopilot',3.0,100.0)
    case '5': vehicle = semisub('DPcontrol',10.0,10.0,40.0,0.5,190.0)
    case '6': vehicle = shipClarke83('headingAutopilot',-20.0,70,8,6,0.7,0.5,10.0,1e5)
    case '7': vehicle = supply('DPcontrol',4.0,4.0,100.0,0.5,20.0)
    case '8': vehicle = tanker('headingAutopilot',-20,0.5,150,20,80)
    case '9': vehicle = remus100('depthHeadingAutopilot',30,50,1525,0.5,170)
    case _: print('Error: Not a valid simulator option'), sys.exit()

printVehicleinfo(vehicle, sampleTime, N)

###############################################################################
# Main simulation loop
###############################################################################
def main():

    [simTime, simData] = simulate(N, sampleTime, vehicle)

    plotVehicleStates(simTime, simData, 1)
    plotControls(simTime, simData, vehicle, 2)
    plot3D(simData, numDataPoints, FPS, filename, 3)

    # webbrowser.get(browser).open_new_tab('file://' + os.path.abspath(filename))

    plt.show()
    plt.close()
```

42 %





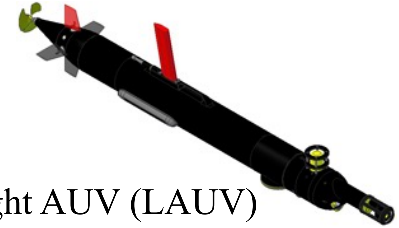
Help | Variable Explorer | Plots | Files

Console 1/A

```
5 - Semisubmersible: controlled by tunnel thrusters and main propellers, L = 84.5 m
6 - Ship: linear maneuvering model specified by L, B and T using the Clarke (1983) formulas
7 - Offshore supply vessel: controlled by tunnel thrusters and main propellers, L = 76.2 m
8 - Tanker: rudder-controlled ship model including shallow water effects, L = 304.8 m
9 - Remus 100: AUV controlled by stern planes, a tail rudder and a propeller, L = 1.6 m
---------------------------------------------------------------------------------------
Please enter a vehicle no.: 9
---------------------------------------------------------------------------------------
Remus 100 cylinder-shaped AUV (see 'remus100.py' for more details)
Length: 1.6 m
Depth and heading autopilots, z_d = 30, psi_d = 50 deg
Sampling frequency: 50 Hz
Simulation time: 200 seconds
```

IPython Console | History | Terminal

The NTNU Fleet


Light AUV (LAUV)

ROV Minerva

Remus 100 AUV


Eelume snake robot


AutoFerry


Hugin AUV

RV Gunnerus

Photo: Fredrik Skoglund


AutoNaut

The NTNU Uncrewed Aerial Vehicles (UAVs)

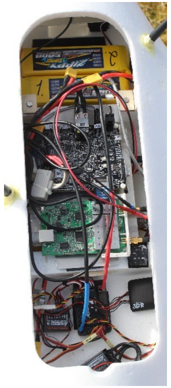
Procurement and operation license from Norwegian CAA (Civil Aviation Authority) for VLOS/BLOS operations since 2014

- Penguin B fixed-wing (VLOS/EVLOS/BLOS)
- 3D Robotics hexa-copters (VLOS)
- Microdrone quadro-copter   (VLOS)
- X8 fixed-wing (VLOS)

UAV Factory Penguin B with Piccolo SL

- 28 m/s cruise speed
- Gasoline, 8 hr endurance
- MTOW 21 kg
- 2-5 kg payload capacity
- Large payload bay
- 80W generator
- Avionics system integration made with Maritime Robotics based on Cloudcap technology
- Telemetry on 2.4 GHz radio, GPRS (and VHF)
- Catapult launch
- Custom payload system integration with avionics interface

Skywalker X8 with Ardupilot



- 18 m/s cruise speed
- Catapult launch
- Belly or net landing
- Electric, 1hr endurance
- Large payload bay
- >1 kg payload capacity
- Inexpensive
- Flexible avionics and payload system integration with ArduPilot open source autopilot and mission planning SW
- Currently telemetry on 433 MHz or 5.8 GHz radio for VLOS
- Can be set up for BLOS with GPRS and VHF radio links

Microdrone Quadcopter

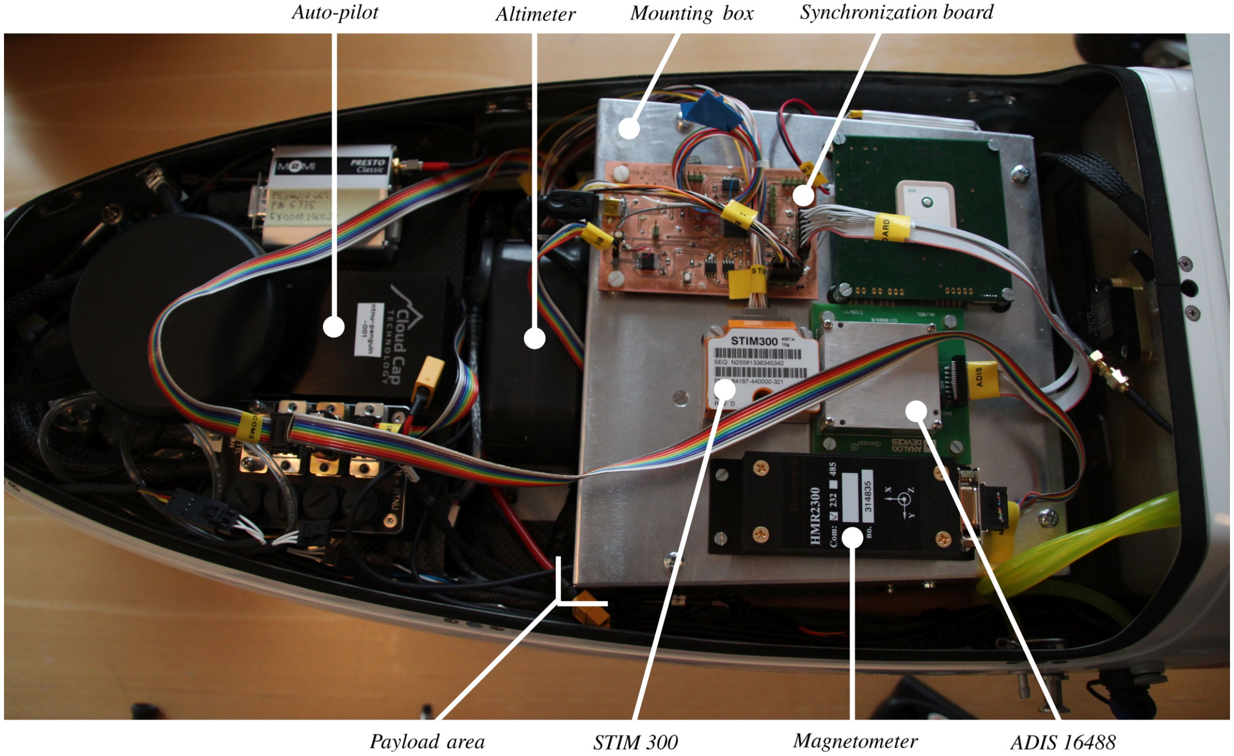
- Turn-key solution
- Various camera, video and radio systems
- Electric, 45 min endurance
- 2-3 kg payload capacity

NTNU Airfield at Agdenes

Located 94 km North-West of Trondheim

UAV as a Remote Sensing Platform

Penguin equipped with Camera, INS and GPS Sensor Suite for Data Logging



Penguin navigation payload: Two IMUs, optical camera, infrared camera, RTK GPS and embedded controller for data logging

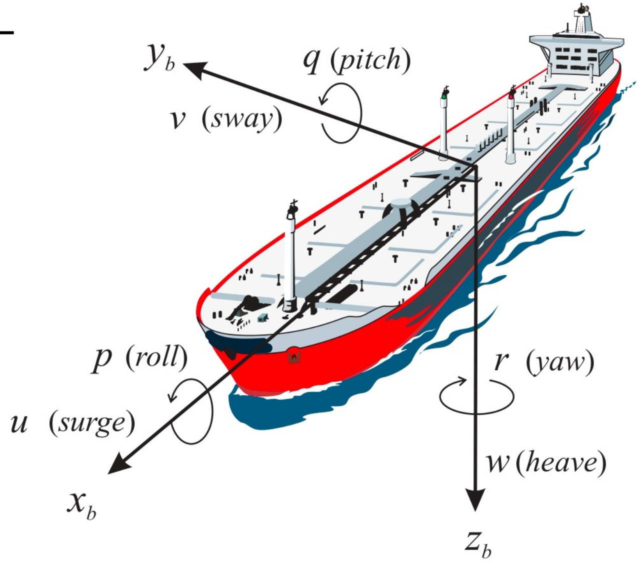
Degrees of Freedom (DOFs)

		BODY		NED
DOF		Forces and moments	Linear and angular velocities	Positions and Euler angles
1	Motions in the x_b-direction (surge)	X	u	x^n
2	Motions in the y_b-direction (sway)	Y	v	y^n
3	Motions in the z_b-direction (heave)	Z	w	z^n
4	Rotation about the x_b-axis (roll)	K	p	ϕ
5	Rotation about the y_b-axis (pitch)	M	q	θ
6	Rotation about the z_b-axis (yaw)	N	r	ψ

The notation is adopted from ***SNAME (1950).***

For a marine craft, DOF is the set of independent displacements and rotations that completely specify the displaced position and orientation of the craft. A craft that can move freely in the 3-D space has maximum 6 DOFs—three translational and three rotational components.

Consequently, a fully actuated marine craft operating in 6 DOFs must be equipped with actuators that can produce independent forces and moments in all directions.



Degrees of Freedom (DOFs)

When designing feedback control systems for marine craft, reduced-order models are often used since most vehicles do not have actuation in all DOFs. This is usually done by decoupling the motions of the vessel according to:

1-DOF models can be used to design forward speed controllers (surge), heading autopilots (yaw) and roll damping systems (roll).

3-DOF models are usually horizontal-plane models (surge, sway and yaw) for ships, semi-submersibles and underwater vehicles that are used in DP systems, trajectory-tracking control systems and path-following systems. For slender bodies such as torpedo-shaped AUVs and submarines, it is also common to assume that the motions can be decoupled into longitudinal and lateral motions.

- Longitudinal models (surge, heave and pitch) for forward speed, diving and pitch control.
- Lateral model (sway, roll and yaw) for turning and heading control.

4-DOF models (surge, sway, roll and yaw) are usually formed by adding the roll equation to the 3-DOF horizontal-plane model. These models are used in maneuvering situations where the purpose is to reduce roll by active control of fins, rudders or stabilizing liquid tanks.

6-DOF models (surge, sway, heave, roll, pitch and yaw) are fully coupled equations of motion used for simulation and prediction of coupled vessel motions. These models can also be used in advanced control systems for underwater vehicles, which are actuated in all DOFs.

Marine Craft

Marine craft: ships, high-speed craft, semi-submersibles, floating rigs, submarines, remotely operated and autonomous underwater vehicles, torpedoes and other propelled/powered structures for instance a floating airfield.

Vehicles that do not travel on land (ocean and flight vehicles) are usually called craft.

Vessel: "hollow structure made to float upon the water for purposes of transportation and navigation; especially, one that is larger than a rowboat ".

The words vessel, ship and boat are often used interchangeably. In Encyclopedia Britannica, a ship and a boat are distinguished by their size through the following definition:

Ship: "any large floating vessel capable of crossing open waters, as opposed to a boat, which is generally a smaller craft. The term formerly was applied to sailing vessels having three or more masts; in modern times it usually denotes a vessel of more than 500 tons of displacement.

Submarine: "any naval vessel that is capable of propelling itself beneath the water as well as on the water's surface.

Underwater Vehicle: "small vehicle that is capable of propelling itself beneath the water surface as well as on the water's surface. This includes unmanned underwater vehicles (UUV), remotely operated vehicles (ROV) and autonomous underwater vehicles (AUV).

Marine Craft

Marine vessels are also classified according to their maximum operating speed. For this purpose it is common to use the Froude number

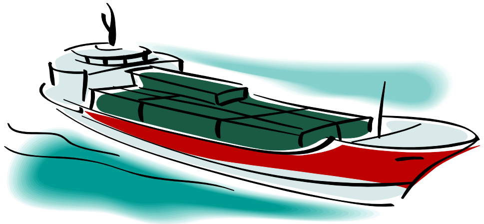
$$F_n := \frac{U}{\sqrt{gL}}$$

U: ship speed
L: overall (submerged length of the ship)
g: acceleration of gravity

The pressure carrying the vessel can be divided into hydrostatic and hydrodynamic pressure. The corresponding forces are:

- Buoyancy force due to the hydrostatic pressures (proportional to the displacement of the ship)

- Hydrodynamic force due to the hydrodynamic pressure (approximately proportional to the square of the speed)

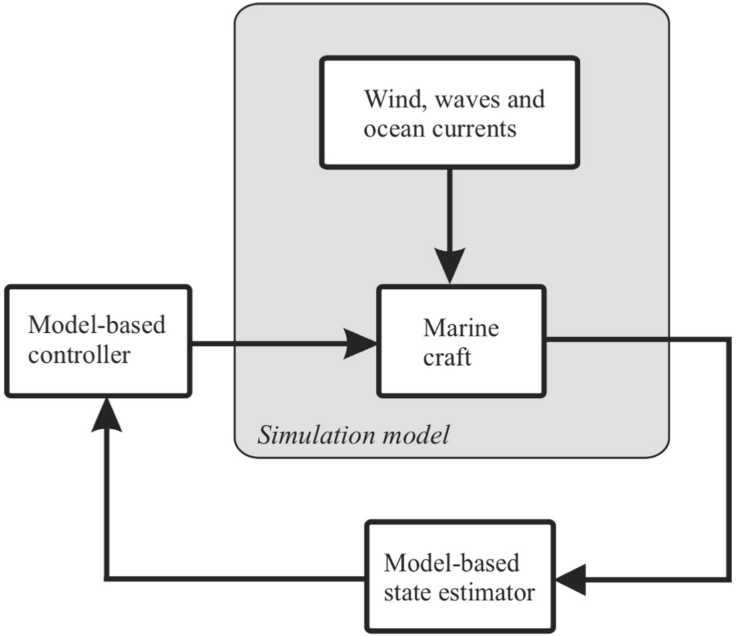


In this course, only displacement vessels are covered

Then we can classify the vessels according to (Faltinsen 2005):

- Displacement vessels (F_n < 0.4): The buoyancy force dominates.
- Semi-displacement vessel (0.4-0.5 < F_n < 1.0-1.2): The buoyancy force is not dominant at the maximum operating speed.
- Planning vessels (F_n > 1.0-1.2): The hydrodynamic force mainly carries the weight.

Classification of Models



Simulation Model
This model is the most accurate description of a system, for instance a 6-DOF high-fidelity model for simulation of coupled motions in the time domain.

Control Design Model
The controller model is a reduced-order or simplified simulation model that is used to design the motion control system. In its simplest form, this model is used to compute a set of constant gains for a PID controller.

State Estimator Design Model:
Stochastic state estimators (Kalman filters) and deterministic state observers are both designed using mathematical models which are different from the models used in the simulator and the controller since the purpose is to capture the additional dynamics associated with the sensors and navigation system as well as disturbances

The Classical Models in Naval Architecture

The motions of a marine craft exposed to wind, waves and ocean currents are usually modeled in 6 DOFs by applying Newton's 2nd law.

$$
\begin{aligned}
m\left[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})\right] &= X \\
m\left[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})\right] &= Y \\
m\left[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})\right] &= Z \\
I_x\dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} & \\
+ m\left[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)\right] &= K \\
I_y\dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} & \\
+ m\left[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)\right] &= M \\
I_z\dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} & \\
+ m\left[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)\right] &= N
\end{aligned}
$$

The Classical Models in Naval Architecture

The external forces and moments X,Y,Z,K,M and N acting on a marine craft are usually modeled by using:

Maneuvering Theory: The study of a ship moving at constant positive speed U in calm water within the framework of maneuvering theory assumes that the hydrodynamic coefficients are frequency independent (no wave excitation).

The zero-frequency assumption is only valid for surge, sway and yaw since the natural frequency of a PD controlled ship (closed loop) will be close to zero. The natural period will typically be in the range of 100 s to 150 s. Hence, the minimum natural frequency is obtained for T = 150 *s,* which gives

$$\begin{aligned} \omega_n &= \frac{2\pi}{T} \\ &\approx 0.04 \; rad/s \end{aligned}$$

Seakeeping Theory: The motions of ships at zero or constant speed in waves can be analyzed using seakeeping theory where the hydrodynamic coefficients and wave forces are computed as a function of the wave excitation frequency using the hull geometry.

Fossen's Equation

The dynamics of a robot manipulator can be expressed by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau$$

- **q** is a vector of joint angels
- $\boldsymbol{\tau}$ is a vector of torque
- **M** and **C** are the system inertia and Coriolis matrices

Inspired by this, Fossen (1991) elegantly formulated the 6-DOF equations of motion for a marine craft within a concise matrix-vector framework, now recognized as Fossen's equation

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_0 = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}}$$

- body-fixed velocities:
- position and Euler ang $\boldsymbol{\nu} = [u, v, w, p, q, r]^\top$
- **M, C** and **D** denote the sy $\boldsymbol{\eta} = [x^n, y^n, z^n, \phi, \theta, \psi]^\top$ damping matrices
- **g** is a vector of gravitational and buoyancy forces and moments

Fossen, T. I. (1991). Nonlinear Modelling and Control of Underwater Vehicles. Doctoral thesis, Department of Engineering Cybernetics, Norwegian Institute of Technology (NTH), June 1991.

Publication Database

If you are interested in my publications on aircraft and marine craft as well as guidance, navigation and control systems, you can search the following MySQL database for title keywords:

https://www.fossen.biz/publications/

Publications

Professor Thor I. Fossen

Database Keyword Search: #1 AUV **#2** **#3**

Submit

Home | Publications | Students | Teaching | Lecture notes | Software | GitHub | Recognitions | Patents | Appointments | Labs

Titles containing the keyword *AUV*

Journals

[1] **Fjellstad, O.-E. and T. I. Fossen (1994).** Position and Attitude Tracking of AUVs: A Quaternion Feedback Approach. IEEE Journal of Oceanic Engineering 19(4), 512-518.
doi.org/10.1109/48.338387

Conference Papers

[2] **Moe, S., K. Y. Pettersen, T. I. Fossen and J. T. Gravdahl (2016).** Line-of-Sight Curved Path Following for Underactuated USVs and AUVs in the Horizontal Plane under the influence of Ocean Currents. 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, IEEE Xplore, pp. 38-45.
Preprint doi.org/10.1109/MED.2016.7536018

[3] **Borup, K. T., T. I. Fossen, J. Braga and J. Borges de Sousa (2014).** Nonlinear Observer for Depth-Aided INS: Experimental Evaluation using an AUV. 22nd Mediterranean Conference on Control and Automation, Palermo, Italy, IEEE Xplore, pp. 1231-1236.
Preprint doi.org/10.1109/MED.2014.6961544

[4] **Lekkas, A. M. and T. I. Fossen (2013).** A Quaternion-Based LOS Guidance Scheme for Path Following of AUVs. 9th IFAC Conference on Control Applications in Marine Systems (CAMS), Osaka, Japan, IFAC Proceedings Volumes 46(33), pp. 245-250.
Open Access doi.org/10.3182/20130918-4-JP-3022.00070