Chapter 8 – Models for Underwater Vehicles

8.1 6-DOF Models for AUVs and ROVs
8.2 Longitudinal and Lateral Models for Submarines
8.3 Decoupled Models for “Flying Underwater Vehicles”
8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls
8.5 Spherical-Shaped Vehicles

The foundation for the models are the kinematic equations (Chapter 2), rigid-body kinetics (Chapter 3), hydrostatics (Chapter 4) and maneuvering models (Chapter 6).
Chapter Goals

• Be able to write down the 6-DOF equations of motion of an underwater vehicle using Euler angles and unit quaternions.
• Be able to apply symmetry conditions to 6-DOF models and identify which elements in the $M$, $C$ and $D$ matrices which are zero.
• Be able to write down the gravity and buoyance vector $g$ for different types of underwater vehicles.
• Understand what we mean with a neutrally buoyant vehicle and how the location of the CG and the CB affects the restoring forces of a submerged vehicle.
• Understand how different models for underwater vehicles are build up and be able to distinguish between:
  • Longitudinal and lateral models for submarines
  • Decoupled Models for “Flying Underwater Vehicles”
  • Cylinder-Shaped Vehicles and Myring-Type Hulls
  • Spherical-Shaped Vehicles
8.1 6-DOF Models for AUVs and ROVs

Underwater vehicles with actuation (thrusters, moving weights, spinning rotors and control surfaces) in all DOFs can control the position and attitude in 6 DOFs.

\[
\dot{\eta} = J_k(\eta) (\nu_r + \nu_c) \quad k \in \{\Theta, q\}
\]

\[
M \dot{\nu}_r + C(\nu_r) \nu_r + D(\nu_r) \nu_r + g(\eta) + g_o = \tau
\]

\[
M = M_{RB} + M_A \\
C(\nu_r) = C_{RB}(\nu_r) + C_A(\nu_r) \\
D(\nu_r) = D + D_n(\nu_r)
\]

Euler angles

\[
J_\Theta(\eta) = \begin{bmatrix} R(\Theta_{nb}) & 0_{3 \times 3} \\ 0_{3 \times 3} & T(\Theta_{nb}) \end{bmatrix}, \quad \eta = [x^n, y^n, z^n, \phi, \theta, \psi]^T
\]

Unit quaternions

\[
J_q(\eta) = \begin{bmatrix} R(q^n_{\theta}) & 0_{3 \times 3} \\ 0_{4 \times 3} & T(q^n_{\theta}) \end{bmatrix}, \quad \eta = [x^n, y^n, z^n, \eta, \epsilon_1, \epsilon_2, \epsilon_3]^T
\]

Starboard–port symmetrical underwater vehicles

\[
y_g = 0 \quad \text{and} \quad I_{xy} = I_{yz} = 0
\]

\[
M = \begin{bmatrix}
\dot{m} - X_w & 0 & -X_\dot{w} & 0 & m z_g - X_\dot{q} & 0 \\
0 & m - Y_\dot{w} & 0 & -m z_g - Y_\dot{p} & 0 & m x_g - Y_\dot{r} \\
-\dot{X}_w & 0 & \dot{m} - Z_\dot{w} & 0 & -m x_g - Z_\dot{q} & 0 \\
0 & -m z_g - Y_\dot{p} & \dot{m} - Z_\dot{w} & 0 & I_z - K_\dot{p} & 0 \\
m z_g - X_\dot{q} & 0 & -m x_g - Z_\dot{q} & 0 & I_y - M_\dot{q} & 0 \\
0 & m x_g - Y_\dot{r} & 0 & -I_{zz} - K_\dot{r} & 0 & I_z - N_\dot{r}
\end{bmatrix}
\]

\[
g(\eta) = \begin{bmatrix}
(W - B) \sin(\theta) \\
- (W - B) \cos(\theta) \sin(\phi) \\
- (W - B) \cos(\theta) \cos(\phi) \\
(y_g W - y_b B) \cos(\theta) \cos(\phi) + (z_g W - z_b B) \cos(\theta) \sin(\phi) \\
(z_g W - z_b B) \sin(\theta) + (x_g W - x_b B) \cos(\theta) \cos(\phi) \\
-x_g W - x_b B) \cos(\theta) \sin(\phi) - (y_g W - y_b B) \sin(\theta)
\end{bmatrix}
\]
8.1 6-DOF Models for AUVs and ROVs

Damping will be nonlinear and coupled for an underwater vehicle moving in 6 DOFs at high speed

\[
D_n(\nu_r)\nu_r = \begin{bmatrix}
\nu_r^T \begin{bmatrix}
D_{n1}\nu_r \\
D_{n2}\nu_r \\
D_{n3}\nu_r \\
D_{n4}\nu_r \\
D_{n5}\nu_r \\
D_{n6}\nu_r \\
\end{bmatrix}
\end{bmatrix}
\]

\[
\nu_r^T = [u_r, v_r, w_r, p, q, r]
\]

However, if the vehicle is performing a noncoupled motion, we can assume a diagonal structure

\[
D(\nu_r) = -\text{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\}
- \text{diag}\{X_{|u|u}|u_r|, Y_{|v|v}|v_r|, Z_{|w|w}|w_r|, K_{|p|p}|p|, M_{|q|q}|q|, N_{|r|r}|r|\}
\]

Alternatively, the current coefficient representation in Section 6.7.1 can be used

\[
d(V_{rc}, \gamma_{rc}) = -\frac{1}{2}\rho V_{rc}^2 \begin{bmatrix}
A_{Fc}C_X(\gamma_{rc}) \\
A_{Lc}C_Y(\gamma_{rc}) \\
A_{Fc}C_Z(\gamma_{rc}) \\
A_{Lc}H_{Lc}C_K(\gamma_{rc}) \\
A_{Fc}H_{Fc}C_M(\gamma_{rc}) \\
A_{Lc}L_{oa}C_N(\gamma_{rc}) \\
\end{bmatrix}
\]
8.1 6-DOF Models for AUVs and ROVs

Equations of motion expressed in NED – Euler angles

\[ M^*(\eta) \ddot{\eta} + C^*(\nu, \eta) \dot{\eta} + D^*(\nu, \eta) \dot{\eta} + g^*(\eta) + g_o^*(\eta) = \tau^* \]

\[ J_\Theta(\eta) = \begin{bmatrix} R(\Theta_{nb}) & 0_{3 \times 3} \\ 0_{3 \times 3} & T(\Theta_{nb}) \end{bmatrix}, \quad J_\Theta^{-1}(\eta) = \begin{bmatrix} R^T(\Theta_{nb}) & 0_{3 \times 3} \\ 0_{3 \times 3} & T^{-1}(\Theta_{nb}) \end{bmatrix} \]

where \( \eta := [x^n, y^n, z^n, \phi, \theta, \psi]^T \). The representation singularity at \( \theta \neq \pm \pi/2 \) in the expression for \( T_\Theta \) implies that the inverse matrix \( J_\Theta^{-1}(\eta) \) does not exist at this value. The transformation is as follows

\[ \dot{\eta} = J_\Theta(\eta)\nu \quad \iff \quad \nu = J_\Theta^{-1}(\eta)\dot{\eta} \]

\[ \ddot{\eta} = J_\Theta(\eta)\ddot{\nu} + J_\Theta(\eta)\dot{\nu} \quad \iff \quad \ddot{\nu} = J_\Theta^{-1}(\eta)[\ddot{\eta} - J_\Theta(\eta)J_\Theta^{-1}(\eta)\dot{\eta}] \]

\[ M^*(\eta) = J_\Theta^{-T}(\eta)MJ_\Theta^{-1}(\eta) \]

\[ C^*(\nu, \eta) = J_\Theta^{-T}(\eta)[C(\nu) - MJ_\Theta^{-1}(\eta)J_\Theta(\eta)]J_\Theta^{-1}(\eta) \]

\[ D^*(\nu, \eta) = J_\Theta^{-T}(\eta)D(\nu)J_\Theta^{-1}(\eta) \]

\[ g^*(\eta) + g_o^*(\eta) = J_\Theta^{-T}(\eta)[g(\eta) + g_o] \]

\[ \tau^* = J_\Theta^{-T}(\eta)\tau \]
8.1 6-DOF Models for AUVs and ROVs

Equations of motion expressed in NED – Unit quaternions

\[ M^*(\eta) \ddot{\eta} + C^*(\nu, \eta) \dot{\eta} + D^*(\nu, \eta) \dot{\eta} + g^*(\eta) + g_o^*(\eta) = \tau^* \]

\[ J_q(\eta) = \begin{bmatrix} R(q_b^n) & 0_{3\times3} \\ 0_{4\times3} & T(q_b^n) \end{bmatrix}, \quad J_q^\dagger(\eta) = \begin{bmatrix} R^T(q_b^n) & 0_{3\times4} \\ 0_{4\times3} & 4T^T(q_b^n) \end{bmatrix} \]

Notice that pseudo-inverse \( J_q^\dagger(\eta) \) is computed using the left Moore–Penrose pseudo-inverse and by exploiting the property \( T^T(q_b^n)T(q_b^n) = 1/4 I_3 \). Moreover, the left inverse of \( T(q_b^n) \) is

\[ T^l(q_b^n) = (T^T(q_b^n)T(q_b^n))^{-1} T^T(q_b^n) = 4T^T(q_b^n) \]

\[ M^*(\eta) = J_q^\dagger(\eta)^T M J_q^\dagger(\eta) \]

\[ C^*(\nu, \eta) = J_q^\dagger(\eta)^T [C(\nu) - M J_q^\dagger(\eta) \dot{J}_q(\eta)] J_q^\dagger(\eta) \]

\[ D^*(\nu, \eta) = J_q^\dagger(\eta)^T D(\nu) J_q^\dagger(\eta) \]

\[ g^*(\eta) + g_o^*(\eta) = J_q^\dagger(\eta)^T [g(\eta) + g_o] \]

\[ \tau^* = J_q^\dagger(\eta)^T \tau \]
8.1 Properties of the 6-DOF Model

Property 8.1 (System Inertia Matrix $M$)
For a rigid body the system inertia matrix is positive definite and constant, that is

$$M = M^T > 0, \quad \dot{M} = 0$$

Property 8.2 (Coriolis and Centripetal Matrix $C$)
For a rigid body moving through an ideal fluid the Coriolis and centripetal matrix $C(\nu)$ can always be parameterized such that it is skew-symmetric, that is

$$C(\nu) = -C^T(\nu), \quad \forall \nu \in \mathbb{R}^6$$

For the vector representation in $\{n\}$ it is straightforward to show that

1. $M^*(\eta) = M^*(\eta)^T > 0, \quad \forall \eta$
2. $x^T[M^*(\eta) - 2C^*(\nu, \eta)]x = 0, \quad \forall x \neq 0, \quad \forall \nu, \eta$
3. $D^*(\nu, \eta) > 0, \quad \forall \nu, \eta$
8.1 Lyapunov Stability Exploiting 6-DOF Model Properties

\[ V = \frac{1}{2} \nu^T M \nu + \frac{1}{2} \eta^T K_p \eta \]

\[ K_p = K_p^T > 0 \]
\[ M = M^T > 0 \text{ and } \dot{M} = 0 \]

\[ \ddot{V} = \nu^T M \ddot{\nu} + \eta^T K_p \dot{\eta} \]
\[ = \nu^T M \ddot{\nu} + \eta^T K_p J_k(\eta) \nu \]
\[ = \nu^T [M \ddot{\nu} + J_k^T(\eta) K_p \eta] \]

\[ \dot{\eta} = J_k(\eta) \nu \]
\[ M \ddot{\nu} + C(\nu) \nu + D(\nu) \nu + g(\eta) = \tau \]

\[ \dot{V} = \nu^T [\tau - C(\nu) \nu - D(\nu) \nu - g(\eta) + J_k^T(\eta) K_p \eta] \]
\[ \nu^T C(\nu) \nu \equiv 0 \]

\[ \tau = g(\eta) - K_d \nu - J_k^T(\eta) K_p \eta \]
\[ K_d > 0 \]
\[ \nu^T D(\nu) \nu > 0 \]

\[ \dot{V} = -\nu^T [K_d + D(\nu)] \nu \leq 0 \]
8.1 Symmetry Considerations of the System Inertia Matrix

These rules also apply to the linear damping matrix $D$

36 elements for which $M_{ij} = M_{ji}$

since $M$ is positive definite
8.1 Symmetry Considerations of the System Inertia Matrix

(ii) $xz$ plane of symmetry (port/starboard symmetry):

\[
M = \begin{bmatrix}
m_{11} & 0 & m_{13} & 0 & m_{15} & 0 \\
0 & m_{22} & 0 & m_{24} & 0 & m_{26} \\
m_{31} & 0 & m_{33} & 0 & m_{35} & 0 \\
0 & m_{42} & 0 & m_{44} & 0 & m_{46} \\
m_{51} & 0 & m_{53} & 0 & m_{55} & 0 \\
0 & m_{62} & 0 & m_{64} & 0 & m_{66}
\end{bmatrix}
\]

(iii) $yz$ plane of symmetry (fore/aft symmetry):

\[
M = \begin{bmatrix}
m_{11} & 0 & 0 & 0 & m_{15} & m_{16} \\
0 & m_{22} & m_{23} & m_{24} & 0 & 0 \\
0 & m_{32} & m_{33} & m_{34} & 0 & 0 \\
0 & m_{42} & m_{43} & m_{44} & 0 & 0 \\
m_{51} & 0 & 0 & 0 & m_{55} & m_{56} \\
0 & m_{61} & 0 & 0 & m_{65} & m_{66}
\end{bmatrix}
\]

(iv) $xz$ and $yz$ planes of symmetry (port/starboard and fore/aft symmetries):

\[
M = \begin{bmatrix}
m_{11} & 0 & 0 & 0 & m_{15} & 0 \\
0 & m_{22} & 0 & m_{24} & 0 & 0 \\
0 & 0 & m_{33} & 0 & 0 & 0 \\
0 & 0 & m_{44} & 0 & 0 & 0 \\
m_{51} & 0 & 0 & 0 & m_{55} & 0 \\
0 & 0 & 0 & 0 & 0 & m_{66}
\end{bmatrix}
\]

(v) $xz$, $yz$ and $xy$ planes of symmetry (port/starboard, fore/aft and bottom/top symmetries):

\[
M = \text{diag}\{m_{11}, m_{22}, m_{33}, m_{44}, m_{55}, m_{66}\}
\]

Ship, submarine, underwater "flying vehicle", Myring-type AUV

Semisubmersible, rig, barge, cylinder-shaped AUV, torpedo, ROV

Spherical-shaped underwater vehicle
8.2 Longitudinal and Lateral Models for Submarines

The 6-DOF equations of motion can in many cases be divided into two noninteracting (or lightly interacting) subsystems:

- Longitudinal subsystem: states $u_r$, $w_r$, $q$ and $\theta$
- Lateral subsystem: states $v_r$, $p$, $r$, $\phi$ and $\psi$

This decomposition is good for slender symmetrical bodies (large length/width ratio) or so-called “flying vehicles”, as shown in Figure 8.1; typical applications are aircraft, missiles and submarines (Gertler and Hagen 1967; Feldman 1979; Tinker 1982).

![Figure 8.1: Slender body submarine (large length/width ratio).](image)

\[
M_{\text{long}} = \begin{bmatrix}
m_{11} & m_{13} & m_{15} \\
m_{31} & m_{33} & m_{35} \\
m_{51} & m_{53} & m_{55}
\end{bmatrix}
\]

\[
M_{\text{lat}} = \begin{bmatrix}
m_{22} & m_{24} & m_{26} \\
m_{42} & m_{44} & m_{46} \\
m_{62} & m_{64} & m_{66}
\end{bmatrix}
\]
8.2 Longitudinal and Lateral Models for Submarines

Longitudinal equations (linear theory)

\[
\begin{bmatrix}
\dot{z} \\
\dot{\theta}
\end{bmatrix}
= \begin{bmatrix}
\cos(\theta) & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
w \\
q
\end{bmatrix}
+ \begin{bmatrix}
-\sin(\theta) \\
0
\end{bmatrix}
u
\approx \begin{bmatrix}
w_r - U\theta \\
q
\end{bmatrix}
+ \begin{bmatrix}
1 \\
0
\end{bmatrix}w_c
\]

\[
\begin{bmatrix}
m - X\dot{u} & -X\dot{w} & mz_g - X\dot{q} \\
-X\dot{w} & m - Z\dot{w} & -mx_g - Z\dot{q} \\
mz_g - X\dot{q} & -mx_g - Z\dot{q} & I_y - M\dot{q}
\end{bmatrix}
\begin{bmatrix}
\dot{u} \\
\dot{w} \\
\dot{q}
\end{bmatrix}
+ \begin{bmatrix}
-X - X - X \\
-\dot{Z} - \dot{Z} - \dot{Z} \\
-\dot{M} - \dot{M} - \dot{M}
\end{bmatrix}
\begin{bmatrix}
u_r \\
w_r \\
q
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & -(m - X\dot{u})U \\
0 & 0 & m\dot{x}U
\end{bmatrix}
\begin{bmatrix}
u_r \\
w_r \\
q
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
BG_zW\sin(\theta)
\end{bmatrix}
= \begin{bmatrix}
\tau_1 \\
\tau_3 \\
\tau_5
\end{bmatrix}
\]

Decoupled pitch model

\[(I_y - M\dot{q})\ddot{\theta} - M\dot{q}\dot{\theta} + WBG_z\theta = \tau_5\]

\[\omega_5 = \sqrt{\frac{WBG_z}{I_y - M\dot{q}}}, \quad T_5 = \frac{2\pi}{\omega_5}\]
8.2 Longitudinal and Lateral Models for Submarines

Lateral equations (linear theory)

\[ \dot{\phi} = p \]
\[ \dot{\psi} = r \]

\[
\begin{bmatrix}
    m - Y_v & -mz_g - Y_\dot{p} & mx_g - Y_{\dot{r}} \\
    -mz_g - Y_\dot{p} & I_x - K_{\dot{p}} & -I_{zz} - K_{\dot{r}} \\
    mx_g - Y_{\dot{r}} & -I_{zx} - K_{\dot{r}} & I_z - N_r
\end{bmatrix}
\begin{bmatrix}
    \dot{v}_r \\
    \dot{p} \\
    \dot{r}
\end{bmatrix}
+ \begin{bmatrix}
    0 & 0 & (m - X_{\dot{u}})U \\
    0 & 0 & 0 \\
    (X_{\dot{u}} - Y_\dot{v})U & 0 & mx_gU
\end{bmatrix}
\begin{bmatrix}
    v_r \\
    p \\
    r
\end{bmatrix}
+ \begin{bmatrix}
    0 \\
    W BG_z \sin(\phi) \\
    0
\end{bmatrix} = \begin{bmatrix}
    \tau_2 \\
    \tau_4 \\
    \tau_6
\end{bmatrix}
\]

Decoupled roll model

\[(I_x - K_{\dot{p}})\ddot{\phi} - K_p \dot{\phi} + W BG_z \phi = \tau_4\]

\[\omega_4 = \sqrt{\frac{W BG_z}{I_x - K_{\dot{p}}}}, \quad T_4 = \frac{2\pi}{\omega_4}\]

DOF 2, 4, 6 sway, roll, yaw
MSS m-file function: Deep Submergence Rescue Vehicle (DSRV.m)

A deep-submergence rescue vehicle (DSRV) is a type of deep-submergence vehicle used for rescue of downed submarines.

```matlab
function [xdot,U] = DSRV(x,u)
% [xdot, U] = DSRV(in), with in=[x,u] returns returns the speed U in m/s
% (optionally) and the time derivative of the state vector
% x = [ w q x z theta ]' for a deep submergence rescue vehicle (DSRV)
% L = 5.0 m, where
% w: heave velocity (m/s)
% q: pitch velocity (rad/s)
% x: x-position (m)
% z: z-position, positive downwards (m)
% theta: pitch angle (rad)
% Input:
% u: delta (rad), where delta is the stern plane
% Problem Sets, Naval Postgraduate School (NPS), Monterey, CA.
```

Deep submergence rescue vehicle (DSRV.m)

mssSImulink library block for numerical integration of the m-file function DSRV.m
MSS Simulink: demoDSRVdepthControl

Deep submergence rescue vehicle (DSRV) depth control system
Simulink interface to the MSS m-file vessel models
Author: Thor I. Fossen
Tue May 04 16:56:22 2021
8.3 Decoupled Models for “Flying Underwater Vehicles”

For slender symmetrical bodies (large length/width ratio) or so-called “flying underwater vehicles” it is common to decompose the 6-DOF equations of motion into three noninteracting (or lightly interacting) subsystems:

- Forward speed subsystem: state $u_r$
- Course subsystem: states $v_r, p, r, \phi$ and $\chi$
- Pitch-depth subsystem: states $w_r, q, z$ and $\theta$

These subsystems are used to design forward speed, heading/course and pitch–depth control systems for AUVs.

Naval Postgraduate School AUV
https://nps.edu/web/cavr/auv

Infant AUV developed at the University of Lisbon
8.3 Decoupled Models for “Flying Underwater Vehicles”

Forward speed subsystem

\[(m - X\ddot{u})\dot{u}_r - X_{|u|u}|u_r|u_r = T\]

Transfer function approximation

\[u_r = A_1 \cos(\omega t)\]

\[X_{|u|u}|u_r|u_r \approx \frac{8A_1}{3\pi} X_{|u|u} u_r\]

\[u(s) = \frac{K_u}{T_u s + 1} T(s) + d_u(s)\]

Equivalent linearization where \(A_1 = \) design amplitude for harmonic motion

Alternatively, the time constant \(T_u\) can be found by a step response

Transfer function for speed control system. Usually a PI controller using thrust \(T\) as input. Integral action is needed to compensate for the current
8.3 Decoupled Models for “Flying Underwater Vehicles”

Course subsystem

\[
\begin{bmatrix}
    m - Y \dot{\psi} & mx_g - Y \dot{r} \\
    mx_g - Y \dot{r} & I_z - N \dot{r}
\end{bmatrix}
\begin{bmatrix}
    \dot{v}_r \\
    \dot{r}
\end{bmatrix}
+ \begin{bmatrix}
    -Y_v & -N_r \\
    -N_v & -N_r
\end{bmatrix}
\begin{bmatrix}
    v_r \\
    r
\end{bmatrix}
+ \begin{bmatrix}
    0 \\
    (X_\psi - Y_\psi)U
\end{bmatrix}
\begin{bmatrix}
    m - X_\psi U \\
    mx_g U
\end{bmatrix}
\begin{bmatrix}
    v_r \\
    r
\end{bmatrix}
= \begin{bmatrix}
    \tau_2 \\
    \tau_6
\end{bmatrix}
\]

State-space model

\[
\dot{x} = Ax + bu
\]

\[
\begin{bmatrix}
    \dot{v}_r \\
    \dot{r} \\
    \dot{\psi}
\end{bmatrix}
= \begin{bmatrix}
    a_{11} & a_{12} & 0 \\
    a_{21} & a_{22} & 0 \\
    0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    v_r \\
    r \\
    \psi
\end{bmatrix}
+ \begin{bmatrix}
    b_1 \\
    b_2 \\
    0
\end{bmatrix}
\delta_r
\]

Transfer functions

\[
\psi(s) = \frac{K(T_3s + 1)}{s(T_1s + 1)(T_2s + 1)} \delta_r(s) + d_\psi(s)
\]

\[
\approx \frac{K}{s(Ts + 1)} \delta_r(s) + d_\psi(s)
\]

Linear optimal control can be used to control the yaw angle by means of the rudder

Alternatively, a PID controller can be designed using the transfer function for heading or course

\[
\chi = \psi + \beta_c \\
\dot{\chi} = d_\psi + \beta_c
\]

\[
\chi(s) = \frac{K}{s(Ts + 1)} \delta_r(s) + d_\chi(s)
\]
8.3 Decoupled Models for “Flying Underwater Vehicles”

Pitch—depth subsystem

\[
\begin{align*}
\dot{\theta} &= q \cos(\phi) - r \sin(\phi) \approx q \\
\dot{z}^n &= -u \sin(\theta) + v \cos(\theta) \sin(\psi) + w \cos(\theta) \cos(\psi) \approx (w_r + w_c) - U\theta \\
\end{align*}
\]

\[
\begin{bmatrix}
\dot{w}_r \\
\dot{q} \\
\dot{\theta} \\
\dot{z}^n
\end{bmatrix} =
\begin{bmatrix}
a_{11} & a_{12} & 0 & 0 \\
a_{21} & a_{22} & a_{23} & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & -U & 0
\end{bmatrix}
\begin{bmatrix}
w_r \\
q \\
\theta \\
z^n
\end{bmatrix} +
\begin{bmatrix}
b_1 \\
b_2 \\
0 \\
0
\end{bmatrix} \delta_e +
\begin{bmatrix}
0 \\
0 \\
0 \\
1
\end{bmatrix} w_c
\]

State-space model

Linear optimal control can be used to control the pitch angle and the depth by means of the elevator

Alternatively, a PID controller can be designed using the transfer function for depth and pitch angle

Transfer functions

\[
\begin{align*}
\dot{z}^n(s) &= -\frac{U}{s} \theta(s) + d_z(s) \\
\theta(s) &= \frac{a_3}{s^2 + a_1 s + a_2} \delta_e(s) + d_\theta(s)
\end{align*}
\]
function [xdot,U] = npsauv(x,ui)
% [xdot,U] = NPSAUV(x,ui) 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
% an Autonomous Underwater Vehicle (AUV) at the Naval Postgraduate School, Monterey.
% The length of the AUV is L = 5.3 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)
% xpos = position in x-direction (m)
% ypos = position in y-direction (m)
% zpos = position in z-direction (m)
% phi = roll angle (rad)
% theta = pitch angle (rad)
% psi = yaw angle (rad)
% The input vector is:
% ui = [ delta_r delta_s delta_b delta_bp delta_bs n ]' where
% delta_r = rudder angle (rad)
% delta_s = port and starboard stern plane (rad)
% delta_b = top and bottom bow plane (rad)
% delta_bp = port bow plane (rad)
% delta_bs = starboard bow plane (rad)
% n = propeller shaft speed (rpm)
% for Autonomous Diving and Steering of Unmanned Underwater Vehicles,

mssSmulink library block for numerical integration of the m-file function npsauv

https://nps.edu/web/cavr/auv
MSS Simulink: demoNPSAUV
8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

Cylinder-Shaped AUVs such as the

- REMUS AUV (Hydroid 2019; REMUS 2019)

can be represented by the Myring hull profile equations (Myring 1976)

The REMUS AUV:
https://en.wikipedia.org/wiki/REMUS_(AUV)
https://www.ntnu.edu/aur-lab/auv-remus-100

The Light AUV (LAUV):
https://www.oceanscan-mst.com
https://www.ntnu.edu/aur-lab/lauv-harald
8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

Myring hull profile equations (Myring 1976)

\[
r(x) = \begin{cases} 
\frac{1}{2}D \left( 1 - \left( \frac{x-L_n}{L_n} \right)^2 \right)^{1/n}, & 0 \leq x \leq L_n \\
\frac{D}{2}, & L_n < x < L_n + L_c \\
\frac{1}{2}D - \left( \frac{3D}{2L_t^2} - \frac{\tan(\alpha_t)}{L_t} \right) (x - (L_n + L_c))^2 + \left( \frac{D}{L_t^3} - \frac{\tan(\alpha_t)}{L_t^2} \right) (x - (L_n + L_c))^3, & L_n + L_c \leq x \leq L 
\end{cases}
\]

Spheroid

\[
\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1
\]

Myring-type hull (solid) for \(D = 0.2\) m, \(L = 1.3\) m, \(L_n = 0.2\) m, \(L_c = 0.7\) m, \(L_t = 0.4\) m, \(n = 1.8\), and \(\alpha_t = 30^\circ\) approximated by a spheroid (dotted line).
A prolate spheroid is obtained by letting $b = c$ and $a > b$

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

Rigid-body inertia matrix about the CO

$$M_{RB} = H^T(r_{bg}^b) \text{ diag } \left\{ m, m, \frac{2}{5}mb^2, \frac{1}{5}m(a^2 + b^2), \frac{1}{5}m(a^2 + b^2) \right\} H(r_{bg}^b)$$

Rigid-body Coriolis-centripetal matrix matrix about the CO

$$C_{RB}(\nu_{r}) = H^T(r_{bg}^b) \begin{bmatrix}
0 & -mr & mq & 0 & 0 & 0 \\
-0 & 0 & -mp & 0 & 0 & 0 \\
-mq & mp & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & I_z r & -I_y q & 0 \\
0 & 0 & 0 & -I_z r & 0 & I_x p \\
0 & 0 & 0 & I_y q & -I_x p & 0
\end{bmatrix} H(r_{bg}^b)$$

$$I_x = \frac{1}{5}m(b^2 + c^2) = \frac{2}{5}mb^2$$
$$I_y = \frac{1}{5}m(a^2 + c^2) = \frac{1}{5}m(a^2 + b^2)$$
$$I_z = \frac{1}{5}m(a^2 + b^2) = I_y$$
MSS Toolbox (Rigig-Body Mass Matrices of a Spheroid)

\[
M_{RB} = H^T(r_{bg}^b) \text{diag} \left\{ m, m, \frac{2}{5}mb^2, \frac{1}{5}m(a^2 + b^2), \frac{1}{5}m(a^2 + b^2) \right\} H(r_{bg}^b)
\]

\[
C_{RB}(\nu_r) = H^T(r_{bg}^b) \begin{bmatrix}
0 & -mr & mq & 0 & 0 & 0 \\
-mr & 0 & -mp & 0 & 0 & 0 \\
-mq & mp & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & I_x^r & -I_y^q & 0 \\
0 & 0 & 0 & -I_z^r & 0 & I_x^p \\
0 & 0 & 0 & I_y^q & -I_x^p & 0 \\
\end{bmatrix} H(r_{bg}^b)
\]

\[
H(r_{bg}^b)
\]

Function [MRB,CRB] = spheroid(a,b,nu2,r_bg)
% [MRB,CRB] = spheroid(L,D,nu2,r:bg) computes the 6x6 rigid-body mass
% and Coriolis-centripetal matrices of a prolate spheroid of length
% L=2*a and diameter D = 2*b. The spheroid can be used to approximate a
% cylinder-shaped autonomous underwater vehicle (AUV). In general
% nu = [u,v,w,p,q,r]', while linear and angular velocities are denoted by
% nu1 = [u, v, w]' and nu2 = [p, q, r]'. The CRB matrix is computed using
% the linear velocity-independent representation (Fossen 2021, Section 3.3.1)
% according to:
% % [MRB,CRB] = spheroid(a, b, [p, q, r]',[xg, yg, zg]')
% % This is particular useful for the relative equations of motion where
% % nu_r = nu - nu_c and nu_c = [u_c, v_c, w_c, 0, 0, 0]' is the irrotational
% % current velocity. This implies that the AUV equations of motion
% % expressed in the CO satisfies:
% % MRB * d/dt nu + CRB(nu) * nu = MRB * d/dt nu_r + CRB(nu_r) * nu_r = tau
%   \]
8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

Added mass (Lamb 1932)

\[ k_1 = \frac{\alpha_0}{2 - \alpha_0} \]
\[ k_2 = \frac{\beta_0}{2 - \beta_0} \]
\[ k' = \frac{e^4(\beta_0 - \alpha_0)}{(2 - \epsilon^2)[2e^2 - (2 - \epsilon^2)(\beta_0 - \alpha_0)]} \]

\[ \alpha_0 = \frac{2(1 - \epsilon^2)}{e^3} \left( \frac{1}{2} \ln \frac{1 + \epsilon}{1 - \epsilon} - \epsilon \right) \]
\[ \beta_0 = \frac{1}{e^2} - \frac{1 - \epsilon^2}{2e^3} \ln \frac{1 + \epsilon}{1 - \epsilon} \]
\[ e := 1 - (b/a)^2 \]

Added inertia matrix about the CO

\[ M_A = -\text{diag}\{X\ddot{u}, Y\ddot{v}, Z\ddot{w}, K\dot{p}, M\dot{q}, N\dot{r}\} \]
\[ = \text{diag}\{mk_1, mk_2, mk_2, 0, k'I_y, k'I_y\} \]

Added Coriolis-centripetal matrix matrix about the CO

\[ C_A(\nu_r) = \begin{bmatrix}
0 & 0 & 0 & 0 & -Z\ddot{w}_r & Y\dot{v}_r \\
0 & 0 & 0 & Z\ddot{w}_r & 0 & -X\dot{u}_r \\
0 & 0 & 0 & 0 & -Y\dot{v}_r & X\dot{u}_r \\
Z\ddot{w}_r & 0 & -X\dot{u}_r & 0 & -N\dot{r}r & M\dot{q}q \\
-Y\dot{v}_r & X\dot{u}_r & N\dot{r}r & 0 & -K\dot{p}p & 0 \\
-\dot{X}_\nu & -X\dot{u}_r & 0 & -M\dot{q}q & K\dot{p}p & 0
\end{bmatrix} \]
MSS Toolbox (Added Mass Matrices by Imlay 1961)

\[ M_A = -\text{diag}\{X_u, Y_v, Z_i, \dot{K}_p, M_q, N_r\} = \text{diag}\{m_k, m_k, m_k, 0, kI_y, k' I_y\} \]

\[
C_A(\nu_r) = \begin{bmatrix}
0 & 0 & 0 & 0 & -Z_{\dot{w}}w_r & Y_{\ddot{v}}v_r \\
0 & 0 & 0 & 0 & -Y_{\dot{v}}v_r & X_{\ddot{u}}u_r \\
0 & 0 & 0 & 0 & -X_{\dot{u}}u_r & Z_{\ddot{w}}w_r \\
0 & 0 & 0 & 0 & -N_{r}r & M_{dq}q \\
-Z_{\dot{w}}w_r & 0 & -X_{\ddot{u}}u_r & N_{r}r & 0 & -K_{pp}p \\
-Y_{\dot{v}}v_r & 0 & -X_{\ddot{u}}u_r & 0 & -M_{dq}q & 0
\end{bmatrix}
\]

function [MA, CA] = imlay61(a, b, nu, r44)

% [MA, CA] = imlay61(a, b, nu, r44) computes the 6x6 hydrodynamic added mass system matrix MA and the 6x6 added mass Coriolis and centripetal matrix CA for a prolate spheroid with semi-axes a > b using the Lamb's K-factors k1, k2 and k_prime (Fossen 2021, Section 8.4.2). The matrix MA is assumed to be diagonal when the CO is chosen on the centerline midships. The length of the AUV is L = 2*a while the diameter is D = 2*b.

% Inputs: a, b: spheroid semi-axes a > b
% nu = [u, v, w, p, q, r]': generalized velocity vector
% r44: hydrodynamic added moment MA(4,4) = r44 * Ix in roll.
% Typically for r44 are 0.2-0.4.

% Output: MA: 6x6 diagonal hydrodynamic added mass system matrix
% CA: 6x6 hydrodynamic added Coriolis and centripetal matrix

% Example: [MA, CA] = imlay61(a, b, [u,v,w,p,q,r]')
% [MA, CA] = imlay61(a, b, [u,v,w,p,q,r]', r44)

% David Taylor Model Basin. Washington D.C.
8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

\[
\begin{bmatrix}
X \\
Z
\end{bmatrix} = \begin{bmatrix}
\cos(\alpha) & -\sin(\alpha) \\
\sin(\alpha) & \cos(\alpha)
\end{bmatrix} \begin{bmatrix}
-F_{\text{drag}} \\
-F_{\text{lift}}
\end{bmatrix}
\]

\[Y = \frac{1}{2} \rho V_r^2 S C_Y(\beta)\]

Nonlinear damping about the CO (Beard and McLain 2012)

\[d(\nu_r) = \frac{1}{2} \rho V_r^2 S \begin{bmatrix}
C_D(\alpha) \cos(\alpha) - C_L(\alpha) \sin(\alpha) & \quad \\ C_Y(\beta) & \\
C_D(\alpha) \sin(\alpha) - C_L(\alpha) \cos(\alpha) & \\
0 & \\
0 & \\
0 & \\
0 &
\end{bmatrix} \]

Linear damping about the CO

\[D = -\text{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\}\]

\[F_{\text{drag}} = \frac{1}{2} \rho V_r^2 S C_D(\alpha)\]

\[F_{\text{lift}} = \frac{1}{2} \rho V_r^2 S C_L(\alpha)\]

\[C_D(\alpha) \approx C_{D_0} + C_{D_\alpha} \alpha + C_{D_{\delta_e}} \delta_e\]

\[C_L(\alpha) \approx C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\delta_e}} \delta_e\]

\[C_Y(\beta) \approx C_{Y_0} + C_{Y_\beta} \beta + C_{Y_{\delta_r}} \delta_r\]

Nonlinear damping is important for high-speed flying vehicles, while linear damping dominates at low speed.
The function `force_liftDrag.m` calls

```
function tau_liftdrag = forceLiftDrag(b,S,CD_0,alpha,U_r)
    % tau_liftdrag = forceLiftDrag(b,S,CD_0,alpha,U_r) computes the hydrodynamic
    % lift and drag forces of a submerged "wing profile" for varying angle of
    % attack (Beard and McLain 2012). Application:
    %
    % M d/dt nu_r + C(nu_r)*nu_r + D*nu_r + g(eta) = tau + tau_liftdrag
    %
    % Output:
    % tau_liftdrag: 6x1 generalized force vector
    %
    % Inputs:
    % b: wing span (m)
    % S: wing area (m^2)
    % CD_0: parasitic drag (alpha = 0), typically 0.1-0.2 for a streamlined body
    % alpha: angle of attack, scalar or vector (rad)
    % U_r: relative speed (m/s)
    %
    % Example:
    %
    % Cylinder-shaped AUV with length L = 1.8, diameter D = 0.2 and CD_0 = 0.1:
    % tau_liftdrag = forceLiftDrag(0.2, 1.8*0.2, 0.1, alpha, U_r)
    %
    % Author: Thor I. Fossen
    % Date: 25 April 2021
```
MSS Toolbox (Linear Damping Matrix)

\[ a = 1; \ b = 0.7; \ \text{nu} = [1 \ 0 \ 0 \ 0.1 \ 0.2 \ 0.3]; \ r_{bg} = [0 \ 0 \ 0.1]; \]
\[ \text{MRB} = \]
\[ \begin{bmatrix}
1.0e+03 & 0 & 0 & 0 & 0.2106 & 0 \\
2.1059 & 0 & 0 & -0.2106 & 0 & 0 \\
0 & 2.1059 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.1059 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.4338 & 0 & 0 \\
0.2106 & 0 & 0 & 0 & 0.6486 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.6275 \\
\end{bmatrix} \]
\[ \text{CRB} = \]
\[ \begin{bmatrix}
0 & -631.7617 & 421.1745 & 63.1762 & 0 & 0 \\
631.7617 & 0 & -210.5872 & 0 & 0 & 0 \\
-421.1745 & 210.5872 & 0 & -210.5872 & -42.1744 & 0 \\
63.1762 & 0 & 21.0587 & 0 & 0 & -125.5100 \\
0 & -63.1762 & 42.1174 & -181.9474 & 0 & 41.2751 \\
0 & 0 & 0 & 0 & 125.5100 & -41.2751 \\
\end{bmatrix} \]

\[ \text{Surge, sway and yaw} \]
(T1, T2, T6)

\[ d = \frac{m}{I} \]

\[ \text{Roll and Pitch} \]
(zeta4, zeta5)

\[ d = 2\zeta\sqrt{km}, \quad \zeta = \sqrt{1 - r^2} \]

\[ \text{Heave} \]
T3 = T2 since W = B

\[ D = -\text{diag}\{X_u, Y_u, Z_w, K_p, M_q, N_r\} \]

\begin{verbatim}
function D = Dmtrx(T_126,zeta_45,MRB,MA,hydrostatics)
% D = Dmtrx([T1, T2, T6],[zeta4, zeta5],MRB,MA,hydrostatics)
% computes the 6x6 linear damping matrix for marine craft (submerged and floating) by specifying the time constants [T1, T2, T6] in DOFs 1,2 and 6.
% The time constants can be found by open-loop step responses. For roll and pitch the relative damping ratios are specified using [zeta4, zeta5].
% For floating vessels it is assumed that zeta3 = 0.2 in heave, while submerged vehicles are assumed to be neutrally buoyant, W = B, with equal time constants in heave and sway, that is T3 = T2.
% Inputs: T_126 = [T1, T2, T6]: time constants for DOFs 1, 2 and 6
% zeta_45 = [zeta4, zeta5]: relative damping ratios in DOFs 4 and 5
% MRB: 6x6 rigid-body system matrix (see rbbody.m)
% MA: 6x6 hydrodynamic added mass system matrix
% hydrostatics = G for surface craft (see Gmtrx.m)
% hydrostatics = [W_r_bg' r_bb'] for neutrally buoyant submerged vehicles where W = m*g, r_bg = [xg,yg,zg]' and r_bb = [xb,yb,zb]' %
% Output: D: 6x6 diagonal linear damping matrix
\end{verbatim}
8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

Gravitational/buoyancy forces about the CO

\[
\begin{align*}
\text{CB} = \text{CO} & \quad r_{bb}^b = [0, 0, 0]^T \\
\text{CG} & \quad r_{bg}^b = [0, 0, z_g]^T
\end{align*}
\]

\[
g(\eta) = \begin{bmatrix} 0 \\ 0 \\ z_g W \cos(\theta) \sin(\phi) - y_g W \cos(\theta) \cos(\phi) \\ z_g W \sin(\theta) + x_g W \cos(\theta) \cos(\phi) \\ x_g W \cos(\theta) \sin(\phi) - y_g W \sin(\theta) \end{bmatrix}
\]

neutrally buoyant vehicle \( W = B \)

6-DOF Equations of Motion (Relative Velocity)

\[
\dot{\eta} = J_k(\eta)(\nu_r + \nu_c) \\
M \dot{\nu}_r + C(\nu_r) \nu_r + D \nu_r + d(\nu_r) + g(\eta) = \tau
\]
MSS Toolbox (Gravity and Buoyancy)

```matlab
function g = gvect(W,B,theta,phi,r_bg,r_bb)
% g = gvect(W,B,theta,phi,r_bg,r_bb) computes the 6x1 vector of restoring
% forces about an arbitrarily point CO for a submerged body. For floating
% vessels, use Gmtrx.m
%
% Inputs:  W, B: weight and buoyancy
%          phi,theta: roll and pitch angles
%          r_bg = [x_g y_g z_g]: location of the CG with respect to the CO
%          r_bb = [x_b y_b z_b]: location of the CB with respect to th CO
%
% Author:  Thor I. Fossen
% Date:    14th June 2001
% Revisions: 20 oct 2008 improved documentation
%            22 sep 2013 corrected sign error on last row (Mohammad Khani)
%            24 Apr 2021, minor updates

sth = sin(theta); cth = cos(theta);
sphi = sin(phi);  cphi = cos(phi);

0 = [...
(W-B) * sth
-(W-B) * cth * sphi
-(W-B) * cth * cphi
-(r_bg(2)*W-r_bb(2)*B) * cth * cphi + (r_bg(3)*W-r_bb(3)*B) * cth * sphi
(r_bg(3)*W-r_bb(3)*B) * sth + (r_bg(1)*W-r_bb(1)*B) * cth * cphi
-(r_bg(1)*W-r_bb(1)*B) * cth * sphi - (r_bg(2)*W-r_bb(2)*B) * sth]
```

\[
g(\eta) = \begin{bmatrix}
(W - B) \sin(\theta) \\
-(W - B) \cos(\theta) \sin(\phi) \\
-(W - B) \cos(\theta) \cos(\phi) \\
-(y_g W - y_b B) \cos(\theta) \cos(\phi) + (z_g W - z_b B) \cos(\theta) \sin(\phi) \\
(z_g W - z_b B) \sin(\theta) + (x_g W - x_b B) \cos(\theta) \cos(\phi) \\
-(x_g W - x_b B) \cos(\theta) \sin(\phi) - (y_g W - y_b B) \sin(\theta)
\end{bmatrix}
\]
function [xdot,U] = remus100(x,ui,v_current)
% [xdot,U] = remus100(x,ui,v_current) returns the time derivative of the
% state vector: x = [ u v w p q r x y z phi theta psi ]' and speed U in m/s
% (optionally) for the Remus 100 autonomous underwater vehicle (AUV). The
% length of the AUV is L = 1.7 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 rate (rad/s)
% q: pitch rate (rad/s)
% r: yaw rate (rad/s)
% x: North position (m)
% y: East position (m)
% z: downwards position (m)
% phi: roll angle (rad)
% theta: pitch angle (rad)
% psi: yaw angle (rad)
% The control inputs are:
% ui = [ delta_r delta_s n ]' where
% delta_r: rudder angle (rad)
% delta_s: aft stern plane (rad)
% n: propeller revolution (rpm)
% The last argument v_current is an optional argument for ocean current
% velocities v_current = [u_c v_c w_c]' expressed in the BODY frame.

mssSImulink library block for numerical integration of the m-file
function remus100.m
MSS Simulink: demoAUVdepthHeadingControl
8.5 Spherical-Shaped Vehicles

6-DOF Equations of Motion (Relative Velocity)

\[ \dot{\eta} = J_k(\eta)(\nu_r + \nu_c) \]
\[ M \dot{\nu}_r + C(\nu_r) \nu_r + D \dot{\nu}_r + g(\eta) = \tau \]

Rigid-body matrices about the CO

\[
M_{RB} = \begin{bmatrix}
m & 0 & 0 & 0 & mz_g & 0 \\
0 & m & 0 & -mz_g & 0 & 0 \\
0 & 0 & m & 0 & 0 & 0 \\
0 & -mz_g & 0 & I_x & 0 & 0 \\
mz_g & 0 & 0 & 0 & I_y & 0 \\
0 & 0 & 0 & 0 & 0 & I_z \\
\end{bmatrix}
\]

\[
C_{RB}(\nu_r) = \begin{bmatrix}
0 & -mr & mq & mz_g r & 0 & 0 \\
mr & 0 & -mp & 0 & mz_g r & 0 \\
-mq & mp & 0 & -mz_g p & -mz_g q & 0 \\
-mz_g r & 0 & mz_g p & 0 & I_zr & -I_y q \\
0 & -mz_g r & mz_g q & -I_z r & 0 & I_x p \\
0 & 0 & I_y q & I_x p & 0 & 0 \\
\end{bmatrix}
\]

The ODIN omni-directional underwater vehicle (Choi et al. 2003)

CB = CO  \[ r_{bb}^b = [0, 0, 0]^T \]
CG  \[ r_{bg}^b = [0, 0, z_g]^T \]

\[ I_x = I_y = I_z = \frac{2}{5} mR^2 \]
8.5 Spherical-Shaped Vehicles

Added mass about the CO

\[ M_A = -\text{diag}\{X_\dot{u}, Y_\dot{v}, Z_\dot{w}, 0, 0, 0\} \quad X_\dot{u} = Y_\dot{v} = Z_\dot{w} = -\frac{2}{3}\pi R^2 \]

\[
C_A(\nu_r) = \rho \frac{2}{3}\pi R^2 \\
\begin{bmatrix}
0 & 0 & 0 & 0 & w_r & -v_r \\
0 & 0 & 0 & -w_r & 0 & u_r \\
0 & 0 & 0 & v_r & -u_r & 0 \\
0 & w_r & -v_r & 0 & 0 & 0 \\
-w_r & 0 & u_r & 0 & 0 & 0 \\
v_r & -u_r & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Linear damping (low-speed application)

\[ D = -\text{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\} \]

Gravitational/buoyancy forces about the CO

\[ g(\eta) = \text{diag}\{0, 0, 0, z_g W \cos(\theta) \sin(\phi), z_g W \sin(\theta), 0\} \]

neutral buoyant vehicle \( W = B \)
Chapter Goals – Revisited

• Be able to write down the 6-DOF equations of motion of an underwater vehicle using Euler angles and unit quaternions.
• Be able to apply symmetry conditions to 6-DOF models and identify which elements in the $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{D}$ matrices which are zero.
• Be able to write down the gravity and buoyance vector $\mathbf{g}$ for different types of underwater vehicles.
• Understand what we mean with a neutrally buoyant vehicle and how the location of the CG and the CB affects the restoring forces of a submerged vehicle.
• Understand how different models for underwater vehicles are build up and be able to distinguish between:
  • Longitudinal and lateral models for submarines
  • Decoupled Models for “Flying Underwater Vehicles”
  • Cylinder-Shaped Vehicles and Myring-Type Hulls
  • Spherical-Shaped Vehicles