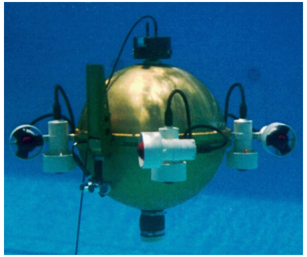
Chapter 8 – Models for Underwater Vehicles

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

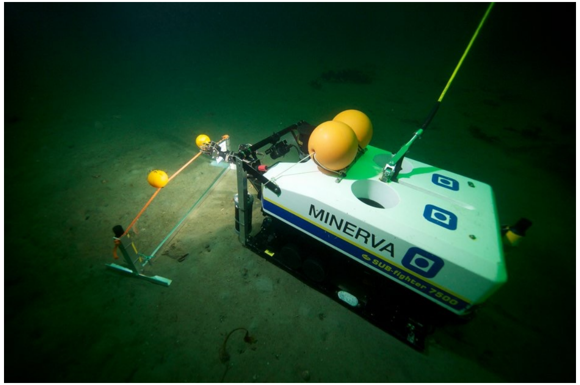


The ODIN omni-directional underwater vehicle
Choi, H. T., A. Hanai, S. K. Choi and J. Yuh (2003). Development of an Underwater Robot, ODIN-III. IEEURSJ Int. Conf. on Intelligent Robots and Systems Las Vegas, NV.

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

NTNU REMUS 100

NTNU LAUV



NTNU ROV Minerva

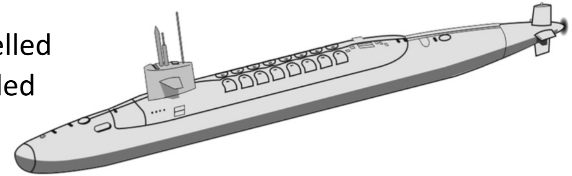
https://www.ntnu.edu/aur-lab/vehicles

Chapter Goals

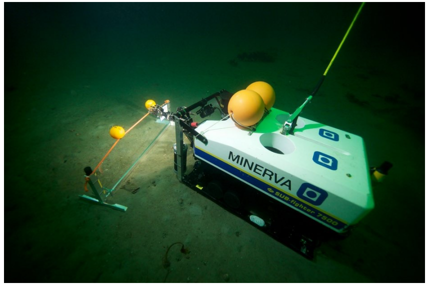
- Write down the 6-DOF equations of motion of an underwater vehicle using Euler angles and unit quaternions.
- Be able to apply symmetry conditions to 6-DOF models and identify which elements in the **M**, **C** and **D** matrices are zero.
- Write down the gravity and buoyancy vector **g** for different 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 built 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

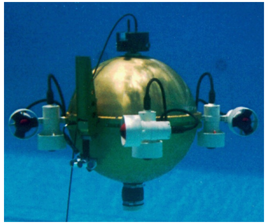
Definitions

Submarine: A submarine is a specific type of underwater vehicle that is designed to operate underwater and is capable of carrying a crew of people. They are versatile vessels used for purposes such as military defense, research, exploration, and transportation. Submarines can travel significant distances underwater, often propelled by both electric and diesel engines, and some are even nuclear-powered for extended endurance and range.



Underwater Vehicle: Underwater vehicles can vary in size, purpose, and complexity. They are often used for tasks such as scientific research, exploration, data collection, and inspection in aquatic environments. Typical classifications are unmanned underwater vehicles (UUV), remotely operated vehicles (ROV), and autonomous underwater vehicles (AUV).





8.1 6-DOF Models for AUVs and ROVs

Equations of motion expressed in BODY

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

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_k(\boldsymbol{\eta})(\boldsymbol{\nu}_r + \boldsymbol{\nu}_c) \qquad \boxed{k \in \{\Theta, q\}}$$
$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_o = \boldsymbol{\tau}$$

$$\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A$$
$$\boldsymbol{C}(\boldsymbol{\nu}_r) = \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r) + \boldsymbol{C}_A(\boldsymbol{\nu}_r)$$
$$\boldsymbol{D}(\boldsymbol{\nu}_r) = \boldsymbol{D} + \boldsymbol{D}_n(\boldsymbol{\nu}_r)$$

Euler angles $\quad \boldsymbol{J}_\Theta(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\Theta}_{nb}) & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{T}(\boldsymbol{\Theta}_{nb}) \end{bmatrix}, \qquad \boldsymbol{\eta} = [x^n, y^n, z^n, \phi, \theta, \psi]^\top$

Unit quaternions $\quad \boldsymbol{J}_q(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{q}_b^n) & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{4\times 3} & \boldsymbol{T}(\boldsymbol{q}_b^n) \end{bmatrix}, \qquad \boldsymbol{\eta} = [x^n, y^n, z^n, \eta, \varepsilon_1, \varepsilon_2, \varepsilon_3]^\top$

Starboard–port symmetrical underwater vehicles $y_g = 0$ and $I_{xy} = I_{yz} = 0$.

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & -X_{\dot{w}} & 0 & mz_g - X_{\dot{q}} & 0 \\ 0 & m - Y_{\dot{v}} & 0 & -mz_g - Y_{\dot{p}} & 0 & mx_g - Y_{\dot{r}} \\ -X_{\dot{w}} & 0 & m - Z_{\dot{w}} & 0 & -mx_g - Z_{\dot{q}} & 0 \\ 0 & -mz_g - Y_{\dot{p}} & 0 & I_x - K_{\dot{p}} & 0 & -I_{zx} - K_{\dot{r}} \\ mz_g - X_{\dot{q}} & 0 & -mx_g - Z_{\dot{q}} & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & mx_g - Y_{\dot{r}} & 0 & -I_{zx} - K_{\dot{r}} & 0 & I_z - N_{\dot{r}} \end{bmatrix}$$

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} (W-B)\sin(\theta) \\ -(W-B)\cos(\theta)\sin(\phi) \\ -(W-B)\cos(\theta)\cos(\phi) \\ -(y_gW - y_bB)\cos(\theta)\cos(\phi) + (z_gW - z_bB)\cos(\theta)\sin(\phi) \\ (z_gW - z_bB)\sin(\theta) + (x_gW - x_bB)\cos(\theta)\cos(\phi) \\ -(x_gW - x_bB)\cos(\theta)\sin(\phi) - (y_gW - y_bB)\sin(\theta) \end{bmatrix}$$

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

$$\boldsymbol{D}_n(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \begin{bmatrix} |\boldsymbol{\nu}_r|^\top \boldsymbol{D}_{n1}\boldsymbol{\nu}_r \\ |\boldsymbol{\nu}_r|^\top \boldsymbol{D}_{n2}\boldsymbol{\nu}_r \\ |\boldsymbol{\nu}_r|^\top \boldsymbol{D}_{n3}\boldsymbol{\nu}_r \\ |\boldsymbol{\nu}_r|^\top \boldsymbol{D}_{n4}\boldsymbol{\nu}_r \\ |\boldsymbol{\nu}_r|^\top \boldsymbol{D}_{n5}\boldsymbol{\nu}_r \\ |\boldsymbol{\nu}_r|^\top \boldsymbol{D}_{n6}\boldsymbol{\nu}_r \end{bmatrix} \qquad |\boldsymbol{\nu}_r|^\top = [\,|u_r|\,,\,|v_r|\,,\,|w_r|\,,\,|p|\,,\,|q|\,,\,|r|\,]$$

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

$$\begin{aligned}\boldsymbol{D}(\boldsymbol{\nu}_r) \;=\; & -\mathrm{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\} \\ & -\mathrm{diag}\{X_{|u|u}|u_r|, Y_{|vv|}|v_r|, Z_{|w|w}|w_r|, K_{|p|p}|p|, M_{|q|q}|q|, N_{|r|r}|r|\}\end{aligned}$$

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

$$\boldsymbol{d}(V_{rc},\gamma_{cr}) = -\frac{1}{2}\rho V_{rc}^2 \begin{bmatrix} A_{F_c}C_X(\gamma_{rc}) \\ A_{L_c}C_Y(\gamma_{rc}) \\ A_{F_c}C_Z(\gamma_{rc}) \\ A_{L_c}H_{L_c}C_K(\gamma_{rc}) \\ A_{F_c}H_{F_c}C_M(\gamma_{rc}) \\ A_{L_c}L_{oa}C_N(\gamma_{rc}) \end{bmatrix}$$

8.1 6-DOF Models for AUVs and ROVs

Equations of motion expressed in NED – Euler angles

$$\boldsymbol{M}^*(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \boldsymbol{C}^*(\boldsymbol{\nu},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \boldsymbol{D}^*(\boldsymbol{\nu},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \boldsymbol{g}^*(\boldsymbol{\eta}) + \boldsymbol{g}_o^*(\boldsymbol{\eta}) = \boldsymbol{\tau}^*$$

$$\boldsymbol{J}_\Theta(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\Theta}_{nb}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}(\boldsymbol{\Theta}_{nb}) \end{bmatrix}, \quad \boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}^{-1}(\boldsymbol{\Theta}_{nb}) \end{bmatrix}$$

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

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} & &\Longleftrightarrow & \boldsymbol{\nu} &= \boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}} \\ \ddot{\boldsymbol{\eta}} &= \boldsymbol{J}_\Theta(\boldsymbol{\eta})\dot{\boldsymbol{\nu}} + \dot{\boldsymbol{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} & &\Longleftrightarrow & \dot{\boldsymbol{\nu}} &= \boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})[\ddot{\boldsymbol{\eta}} - \dot{\boldsymbol{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}] \end{aligned}$$

$$\boxed{\begin{aligned} \boldsymbol{M}^*(\boldsymbol{\eta}) &= \boldsymbol{J}_\Theta^{-\top}(\boldsymbol{\eta})\boldsymbol{M}\boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta}) \\ \boldsymbol{C}^*(\boldsymbol{\nu},\boldsymbol{\eta}) &= \boldsymbol{J}_\Theta^{-\top}(\boldsymbol{\eta})[\boldsymbol{C}(\boldsymbol{\nu}) - \boldsymbol{M}\boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{J}}_\Theta(\boldsymbol{\eta})]\boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta}) \\ \boldsymbol{D}^*(\boldsymbol{\nu},\boldsymbol{\eta}) &= \boldsymbol{J}_\Theta^{-\top}(\boldsymbol{\eta})\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{J}_\Theta^{-1}(\boldsymbol{\eta}) \\ \boldsymbol{g}^*(\boldsymbol{\eta}) + \boldsymbol{g}_o^*(\boldsymbol{\eta}) &= \boldsymbol{J}_\Theta^{-\top}(\boldsymbol{\eta})[\boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_o] \\ \boldsymbol{\tau}^* &= \boldsymbol{J}_\Theta^{-\top}(\boldsymbol{\eta})\boldsymbol{\tau} \end{aligned}}$$

8.1 6-DOF Models for AUVs and ROVs

Equations of motion expressed in NED – Unit quaternions

$$\boldsymbol{M}^*(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \boldsymbol{C}^*(\boldsymbol{\nu},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \boldsymbol{D}^*(\boldsymbol{\nu},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \boldsymbol{g}^*(\boldsymbol{\eta}) + \boldsymbol{g}_o^*(\boldsymbol{\eta}) = \boldsymbol{\tau}^*$$

$$\boldsymbol{J}_q(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{q}_b^n) & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{4\times 3} & \boldsymbol{T}(\boldsymbol{q}_b^n) \end{bmatrix}, \qquad \boldsymbol{J}_q^\dagger(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}^\top(\boldsymbol{q}_b^n) & \boldsymbol{0}_{3\times 4} \\ \boldsymbol{0}_{4\times 3} & 4\,\boldsymbol{T}^\top(\boldsymbol{q}_b^n) \end{bmatrix}$$

Notice that pseudo-inverse $\boldsymbol{J}_q^\dagger(\boldsymbol{\eta})$ is computed using the left *Moore–Penrose pseudo-inverse* and by exploiting the property $\boldsymbol{T}^\top(\boldsymbol{q}_b^n)\boldsymbol{T}(\boldsymbol{q}_b^n) = 1/4\,\boldsymbol{I}_3$. Moreover, the left inverse of $\boldsymbol{T}(\boldsymbol{q}_b^n)$ is

$$\begin{aligned} \boldsymbol{T}^\dagger(\boldsymbol{q}_b^n) &= \left(\boldsymbol{T}^\top(\boldsymbol{q}_b^n)\boldsymbol{T}(\boldsymbol{q}_b^n)\right)^{-1}\boldsymbol{T}^\top(\boldsymbol{q}_b^n) \\ &= 4\,\boldsymbol{T}^\top(\boldsymbol{q}_b^n) \end{aligned}$$

$$\boxed{\begin{aligned} \boldsymbol{M}^*(\boldsymbol{\eta}) &= \boldsymbol{J}_q^\dagger(\boldsymbol{\eta})^\top \boldsymbol{M} \boldsymbol{J}_q^\dagger(\boldsymbol{\eta}) \\ \boldsymbol{C}^*(\boldsymbol{\nu},\boldsymbol{\eta}) &= \boldsymbol{J}_q^\dagger(\boldsymbol{\eta})^\top [\boldsymbol{C}(\boldsymbol{\nu}) - \boldsymbol{M}\boldsymbol{J}_q^\dagger(\boldsymbol{\eta})\dot{\boldsymbol{J}}_q(\boldsymbol{\eta})]\boldsymbol{J}_q^\dagger(\boldsymbol{\eta}) \\ \boldsymbol{D}^*(\boldsymbol{\nu},\boldsymbol{\eta}) &= \boldsymbol{J}_q^\dagger(\boldsymbol{\eta})^\top \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{J}_q^\dagger(\boldsymbol{\eta}) \\ \boldsymbol{g}^*(\boldsymbol{\eta}) + \boldsymbol{g}_o^*(\boldsymbol{\eta}) &= \boldsymbol{J}_q^\dagger(\boldsymbol{\eta})^\top [\boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_o] \\ \boldsymbol{\tau}^* &= \boldsymbol{J}_q^\dagger(\boldsymbol{\eta})^\top \boldsymbol{\tau} \end{aligned}}$$

8.1 Properties of the 6-DOF Model

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

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

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

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

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

1. $\boldsymbol{M}^*(\boldsymbol{\eta}) = \boldsymbol{M}^*(\boldsymbol{\eta})^\top > 0, \quad \forall \boldsymbol{\eta}$
2. $\boldsymbol{x}^\top [\dot{\boldsymbol{M}}^*(\boldsymbol{\eta}) - 2\boldsymbol{C}^*(\boldsymbol{\nu}, \boldsymbol{\eta})]\boldsymbol{x} = 0, \quad \forall \boldsymbol{x} \neq \boldsymbol{0}, \quad \forall \boldsymbol{\nu}, \boldsymbol{\eta}$
3. $\boldsymbol{D}^*(\boldsymbol{\nu}, \boldsymbol{\eta}) > 0, \quad \forall \boldsymbol{\nu}, \boldsymbol{\eta}$

8.1 Lyapunov Stability Exploiting 6-DOF Model Properties

Lyapunov function candidate based on kinetic and potential energies

$$\boxed{V = \frac{1}{2}\boldsymbol{\nu}^\top \boldsymbol{M}\boldsymbol{\nu} + \frac{1}{2}\boldsymbol{\eta}^\top \boldsymbol{K}_p \boldsymbol{\eta}}$$

$$\boldsymbol{K}_p = \boldsymbol{K}_p^\top > 0$$ Positive definite

$$\boldsymbol{M} = \boldsymbol{M}^\top > 0 \; and \; \dot{\boldsymbol{M}} = \boldsymbol{0},$$

$$\begin{aligned}\dot{V} &= \boldsymbol{\nu}^\top \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{\eta}^\top \boldsymbol{K}_p \dot{\boldsymbol{\eta}} \\ &= \boldsymbol{\nu}^\top \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{\eta}^\top \boldsymbol{K}_p \boldsymbol{J}_k(\boldsymbol{\eta})\boldsymbol{\nu} \\ &= \boldsymbol{\nu}^\top [\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{J}_k^\top(\boldsymbol{\eta})\boldsymbol{K}_p\boldsymbol{\eta}]\end{aligned}$$

$$\boxed{\begin{aligned}\dot{\boldsymbol{\eta}} &= \boldsymbol{J}_k(\boldsymbol{\eta})\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) &= \boldsymbol{\tau}\end{aligned}}$$

$$\dot{V} = \boldsymbol{\nu}^\top [\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{J}_k^\top(\boldsymbol{\eta})\boldsymbol{K}_p\boldsymbol{\eta}]$$

$$\boldsymbol{\nu}^\top \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} \equiv \boldsymbol{0}$$ Skew-symmetric property

Nonlinear PID controller with gravity/buoyancy compensation

$$\boxed{\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{\eta}) - \boldsymbol{K}_d\boldsymbol{\nu} - \boldsymbol{J}_k^\top(\boldsymbol{\eta})\boldsymbol{K}_p\boldsymbol{\eta}}$$

$$\boldsymbol{K}_d > 0$$ Positive definite

Time-derivative of the Lyapunov function candidate

$$\boxed{\dot{V} = -\boldsymbol{\nu}^\top[\boldsymbol{K}_d + \boldsymbol{D}(\boldsymbol{\nu})]\boldsymbol{\nu} \leq 0}$$

$$\boldsymbol{\nu}^\top \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} > 0,$$ Strictly positive

8.1 Symmetry Considerations of the System Inertia Matrix

$$
\boldsymbol{M} = \begin{bmatrix}
m - X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{w}} & -X_{\dot{p}} & mz_g - X_{\dot{q}} & -my_g - X_{\dot{r}} \\
-X_{\dot{v}} & m - Y_{\dot{v}} & -Y_{\dot{w}} & -mz_g - Y_{\dot{p}} & -Y_{\dot{q}} & mx_g - Y_{\dot{r}} \\
-X_{\dot{w}} & -Y_{\dot{w}} & m - Z_{\dot{w}} & my_g - Z_{\dot{p}} & -mx_g - Z_{\dot{q}} & -Z_{\dot{r}} \\
-X_{\dot{p}} & -mz_g - Y_{\dot{p}} & my_g - Z_{\dot{p}} & I_x - K_{\dot{p}} & -I_{xy} - K_{\dot{q}} & -I_{zx} - K_{\dot{r}} \\
mz_g - X_{\dot{q}} & -Y_{\dot{q}} & -mx_g - Z_{\dot{q}} & -I_{xy} - K_{\dot{q}} & I_y - M_{\dot{q}} & -I_{yz} - M_{\dot{r}} \\
-my_g - X_{\dot{r}} & x_g - Y_{\dot{r}} & -Z_{\dot{r}} & -I_{zx} - K_{\dot{r}} & -I_{yz} - M_{\dot{r}} & I_z - N_{\dot{r}}
\end{bmatrix}
$$

36 elements for which

$M_{ij} = M_{ji}$

since $\boldsymbol{M}$ is positive definite

(i) xy plane of symmetry (bottom/top symmetry):

$$
\boldsymbol{M} = \begin{bmatrix}
m_{11} & m_{12} & 0 & 0 & 0 & m_{16} \\
m_{21} & m_{22} & 0 & 0 & 0 & m_{26} \\
0 & 0 & m_{33} & m_{34} & m_{35} & 0 \\
0 & 0 & m_{43} & m_{44} & m_{45} & 0 \\
0 & 0 & m_{53} & m_{54} & m_{55} & 0 \\
m_{61} & m_{62} & 0 & 0 & 0 & m_{66}
\end{bmatrix}
$$

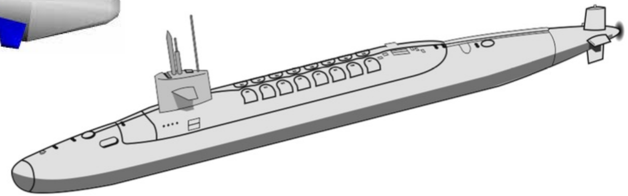
These rules also apply to the linear damping matrix $\boldsymbol{D}$

8.1 Symmetry Considerations of the System Inertia Matrix

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

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

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



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

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

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

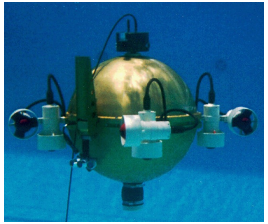
Semisubmersible, rig, barge, cylinder-shaped AUV, torpedo, and ROV

$$M = \begin{bmatrix} m_{11} & 0 & 0 & 0 & m_{15} & 0 \\ 0 & m_{22} & 0 & m_{24} & 0 & 0 \\ 0 & 0 & m_{33} & 0 & 0 & 0 \\ 0 & m_{42} & 0 & m_{44} & 0 & 0 \\ m_{51} & 0 & 0 & 0 & m_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{66} \end{bmatrix}$$

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

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

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 θ
- Lateral subsystem: states v_r, p, r, ϕ and ψ

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).

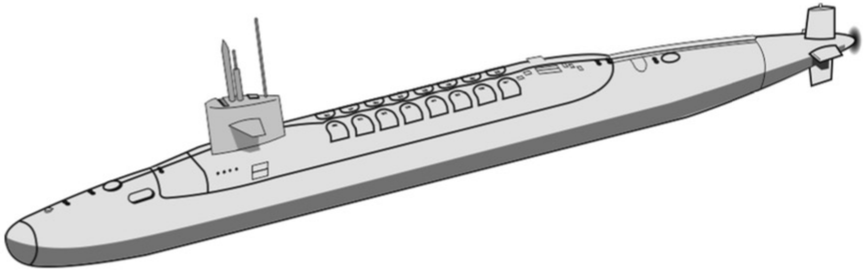


Figure 8.1: Slender body submarine (large length/width ratio).

$$\boldsymbol{M}_{\text{long}} = \begin{bmatrix} m_{11} & m_{13} & m_{15} \\ m_{31} & m_{33} & m_{35} \\ m_{51} & m_{53} & m_{55} \end{bmatrix}$$

$$\boldsymbol{M}_{\text{lat}} = \begin{bmatrix} m_{22} & m_{24} & m_{26} \\ m_{42} & m_{44} & m_{46} \\ m_{62} & m_{64} & m_{66} \end{bmatrix}$$

Feldman, J. (1979). DTMSRDC Revised Standard Submarine Equations of Motion. Technical Report DTNSRDC-SPD-0393-09. Naval Ship Research and Development Center. Washington D.C.

Gertler, M. and G. R. Hagen (1967). Standard Equations of Motion for Submarine Simulation. Technical Report DTMB-2510. Naval Ship Research and Development Center. Washington D.C.

Tinker, S. J. (1982). Identification of Submarine Dynamics from Free-Model Test. In: Proceedings of the DRG Seminar. Netherlands.

8.2 Longitudinal and Lateral Models for Submarines

Longitudinal equations (linear theory)

DOF 1, 3, 5
(surge, heave, pitch)

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

$$\begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{w}} & mz_g - X_{\dot{q}} \\ -X_{\dot{w}} & m - Z_{\dot{w}} & -mx_g - Z_{\dot{q}} \\ mz_g - X_{\dot{q}} & -mx_g - Z_{\dot{q}} & I_y - M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{u}_r \\ \dot{w}_r \\ \dot{q} \end{bmatrix}$$
$$+ \begin{bmatrix} -X_u & -X_w & -X_q \\ -Z_u & -Z_w & -Z_q \\ -M_u & -M_w & -M_q \end{bmatrix} \begin{bmatrix} u_r \\ w_r \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -(m - X_{\dot{u}})U \\ 0 & (Z_{\dot{w}} - X_{\dot{u}})U & mx_g U \end{bmatrix} \begin{bmatrix} u_r \\ w_r \\ q \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ 0 \\ \mathrm{BG}_z W \sin(\theta) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_3 \\ \tau_5 \end{bmatrix}$$

Decoupled pitch model

$$(I_y - M_{\dot{q}})\ddot{\theta} - M_q\dot{\theta} + W\,\mathrm{BG}_z\,\theta = \tau_5$$

$$\omega_5 = \sqrt{\frac{W\,\mathrm{BG}_z}{I_y - M_{\dot{q}}}}, \qquad T_5 = \frac{2\pi}{\omega_5}$$

8.2 Longitudinal and Lateral Models for Submarines

Lateral equations (linear theory)

DOF 2, 4, 6
(sway, roll, yaw)

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

$$\begin{bmatrix} m - Y_{\dot{v}} & -mz_g - Y_{\dot{p}} & mx_g - Y_{\dot{r}} \\ -mz_g - Y_{\dot{p}} & I_x - K_{\dot{p}} & -I_{zx} - K_{\dot{r}} \\ mx_g - Y_{\dot{r}} & -I_{zx} - K_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v}_r \\ \dot{p} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} -Y_v & -Y_p & -Y_r \\ -K_v & -K_p & -K_r \\ -N_v & -N_p & -N_r \end{bmatrix} \begin{bmatrix} v_r \\ p \\ r \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & 0 & (m - X_{\dot{u}})U \\ 0 & 0 & 0 \\ (X_{\dot{u}} - Y_{\dot{v}})U & 0 & mx_g U \end{bmatrix} \begin{bmatrix} v_r \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ W\,\mathrm{BG}_z \sin(\phi) \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_2 \\ \tau_4 \\ \tau_6 \end{bmatrix}$$

Decoupled roll model

$$(I_x - K_{\dot{p}})\ddot{\phi} - K_p\dot{\phi} + W\,\mathrm{BG}_z\,\phi = \tau_4$$

$$\omega_4 = \sqrt{\frac{W\,\mathrm{BG}_z}{I_x - K_{\dot{p}}}}, \qquad T_4 = \frac{2\pi}{\omega_4}$$

MSS m-file function: Deep Submergence Rescue Vehicle (DSRV.m)

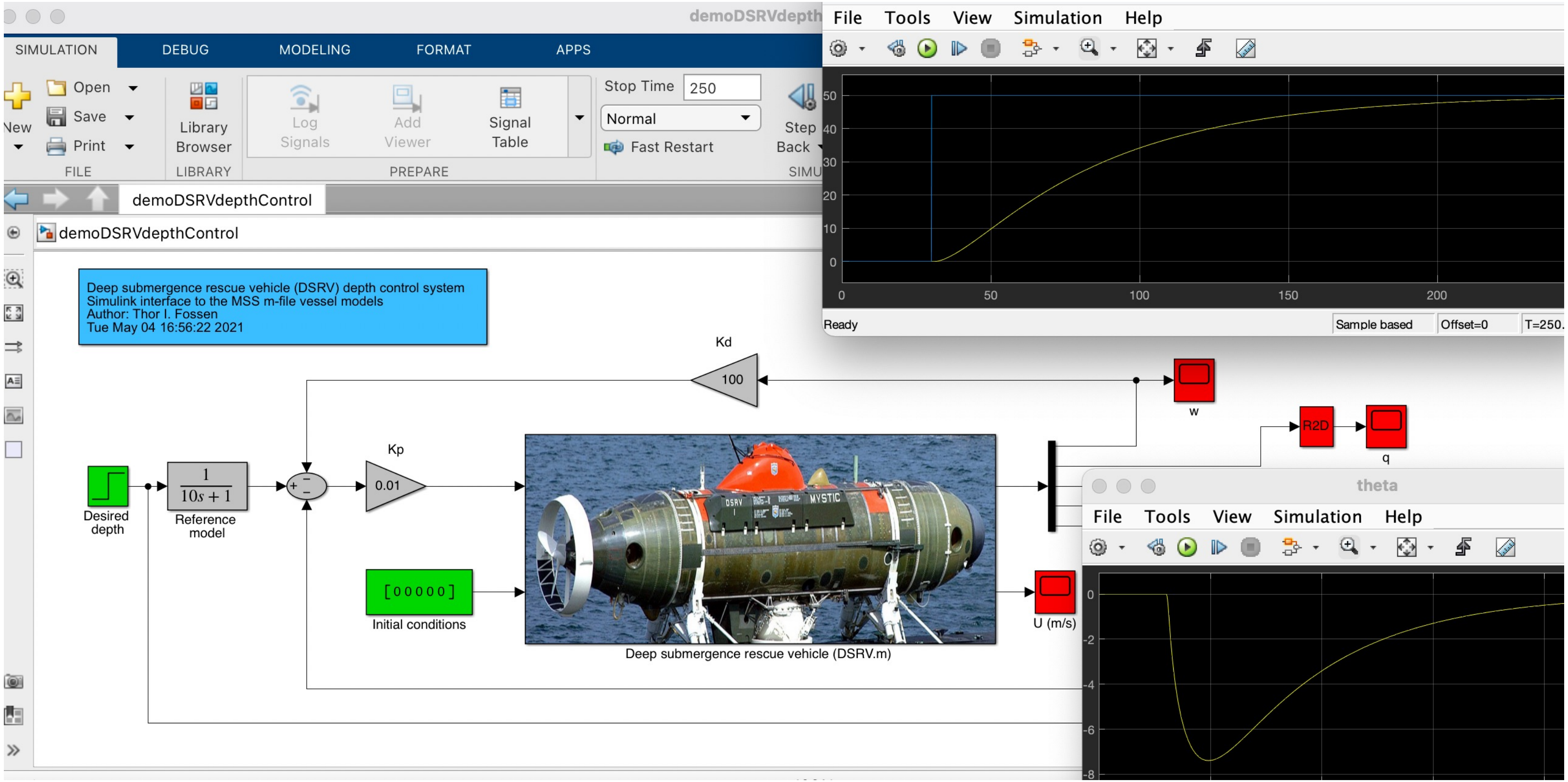
A deep-submergence rescue vehicle (DSRV) is a type of deep-submergence vehicle used for rescue of downed submarines.

```
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
%
% Reference : A.J. Healey (1992). Marine Vehicle Dynamics Lecture Notes and
%             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



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, ϕ and χ
- Pitch-depth subsystem: states w_r, q, z and θ

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

Infante AUV developed at the University of Lisbon

C. Silvestre, A. Pascoal (2005). Depth Control of the Infante AUV using Gain-Scheduled Reduced-Order Output Feedback, IFAC Proceedings Volumes 38(1), 91-96,

8.3 Decoupled Models for "Flying Underwater Vehicles"

Forward speed subsystem

$$(m - X_{\dot{u}})\dot{u}_r - X_{|u|u}|u_r|u_r = T$$

DOF 1
(surge)

Transfer function approximation

$$u_r = A_1 \cos(\omega t)$$

Equivalent linearization where A_1 = design amplitude for harmonic motion.

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

Alternatively, the time constant T_u can be found by a step response.

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

Transfer function for speed control system. Usually, a PI controller using thrust T as input. Integral action is needed to compensate for ocean currents.

$$K_u = \frac{1}{-\frac{8A_1}{3\pi} X_{|u|u}}, \quad T_u = \frac{m - X_{\dot{u}}}{-\frac{8A_1}{3\pi} X_{|u|u}} \qquad d_u(s) = \frac{1}{T_u s + 1} u_c(s)$$

8.3 Decoupled Models for "Flying Underwater Vehicles"

Course subsystem

DOF 2, 6
(sway, yaw)

$$\begin{bmatrix} m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v}_r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} -Y_v & -Y_r \\ -N_v & -N_r \end{bmatrix} \begin{bmatrix} v_r \\ r \end{bmatrix} + \begin{bmatrix} 0 & (m - X_{\dot{u}})U \\ (X_{\dot{u}} - Y_{\dot{v}})U & mx_g U \end{bmatrix} \begin{bmatrix} v_r \\ r \end{bmatrix} = \begin{bmatrix} \tau_2 \\ \tau_6 \end{bmatrix}$$

State-space model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u$$

$$\Updownarrow$$

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

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.

Transfer functions

$$\psi(s) = \frac{K(T_3 s + 1)}{s(T_1 s + 1)(T_2 s + 1)} \delta_r(s) + d_\psi(s) \approx \frac{K}{s(Ts + 1)} \delta_r(s) + d_\psi(s)$$

$$\chi = \psi + \beta_c \qquad d_\chi = d_\psi + \beta_c$$

$$\chi(s) = \frac{K}{s(Ts + 1)} \delta_r(s) + d_\chi(s)$$

8.3 Decoupled Models for "Flying Underwater Vehicles"

Pitch—depth subsystem

DOF 3, 5
(heave, pitch)

$$\dot{\theta} = q\cos(\phi) - r\sin(\phi) \approx q$$
$$\dot{z}^n = -u\sin(\theta) + v\cos(\theta)\sin(\psi) + w\cos(\theta)\cos(\psi) \approx (w_r + w_c) - U\theta$$

$$\begin{bmatrix} m - Z_{\dot{w}} & -mx_g - Z_{\dot{q}} \\ -mx_g - Z_{\dot{q}} & I_y - M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{w}_r \\ \dot{q} \end{bmatrix} + \begin{bmatrix} -Z_w & -Z_q \\ -M_w & -M_q \end{bmatrix} \begin{bmatrix} w_r \\ q \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & -(m - X_{\dot{u}})U \\ (Z_{\dot{w}} - X_{\dot{u}})U & mx_g U \end{bmatrix} \begin{bmatrix} w_r \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ W\,\mathrm{BG}_z\sin(\theta) \end{bmatrix} = \begin{bmatrix} \tau_3 \\ \tau_5 \end{bmatrix}$$

State-space model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u + \boldsymbol{\varphi}(t)$$
$$\Updownarrow$$
$$\begin{bmatrix} \dot{w}_r \\ \dot{q} \\ \dot{\theta} \\ \dot{z}^n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -U & 0 \end{bmatrix} \begin{bmatrix} w_r \\ q \\ \theta \\ z^n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ 0 \end{bmatrix} \delta_e + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} w_c$$

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

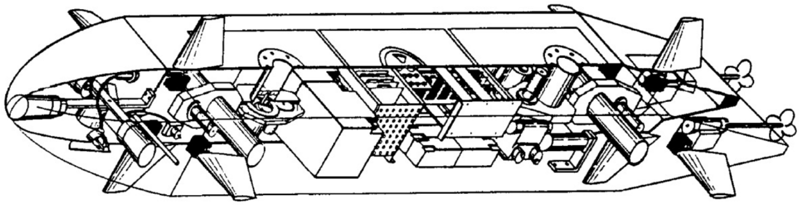
$$z^n(s) = -\frac{U}{s}\theta(s) + d_z(s)$$
$$\theta(s) = \frac{a_3}{s^2 + a_1 s + a_2}\delta_e(s) + d_\theta(s)$$

MSS m-file function and Simulink Interface: NPS AUV

```
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 Autnomous Underwater Vehicle (AUV) at the Naval Postgraduate School, Monterrey.
% 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)
%
% Reference : Healey, A.J. and Lienard, D. (1993). Multivariable Sliding Mode Control
%             for Autonomous Diving and Steering of Unmanned Underwater Vehicles,
%             IEEE Journal of Ocean Engineering, JOE-18(3):327-339
```

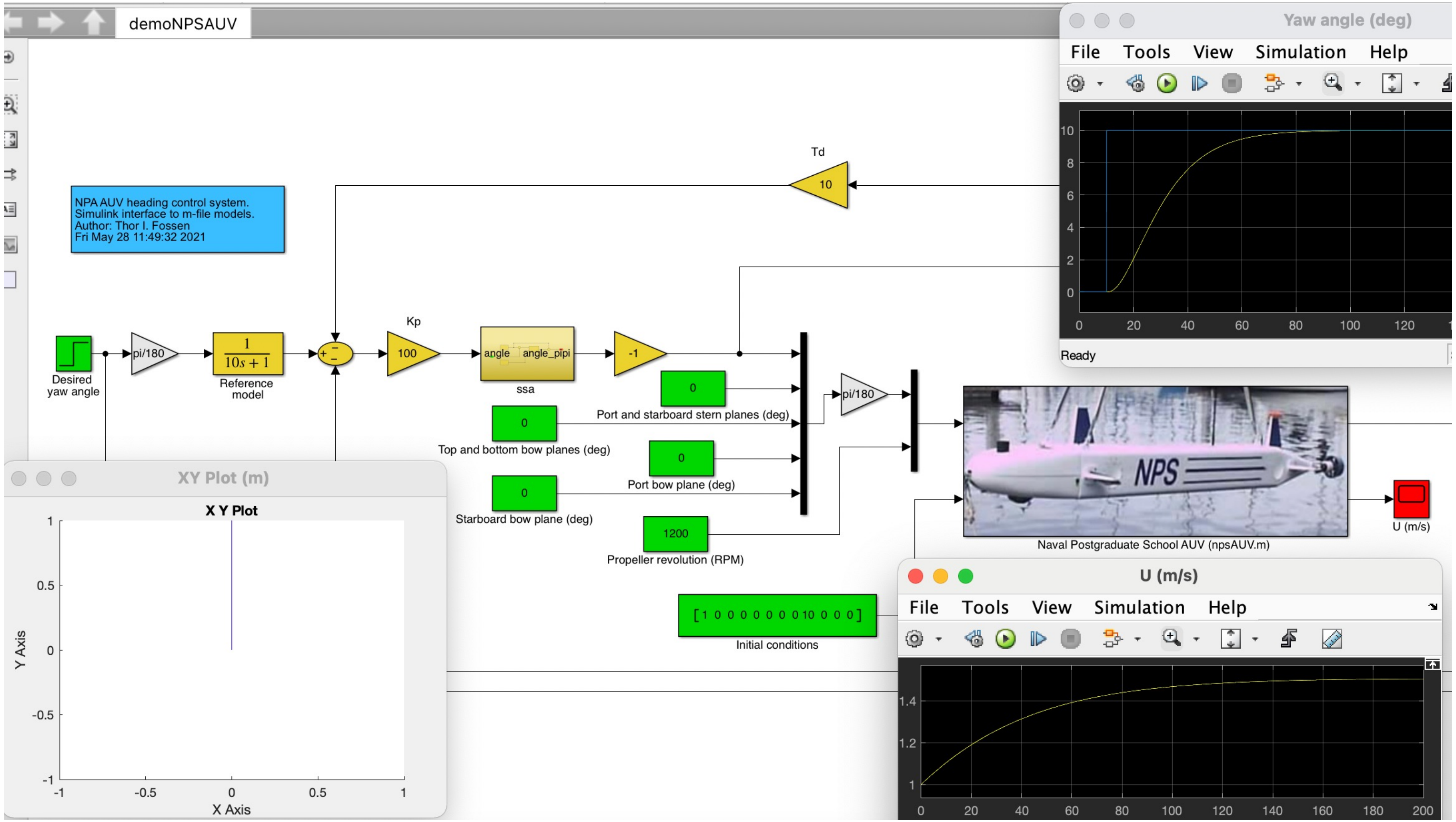
mssSImulink library block for numerical integration of the m-file function npsauv

Naval Postgraduate School AUV (npsAUV.m)



https://nps.edu/web/cavr/auv

MSS Simulink: demoNPSAUV



8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

Cylinder-shaped AUVs such as the

- **Light Autonomous Underwater Vehicle - LAUV** (Madureira et al. 2013)
- **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

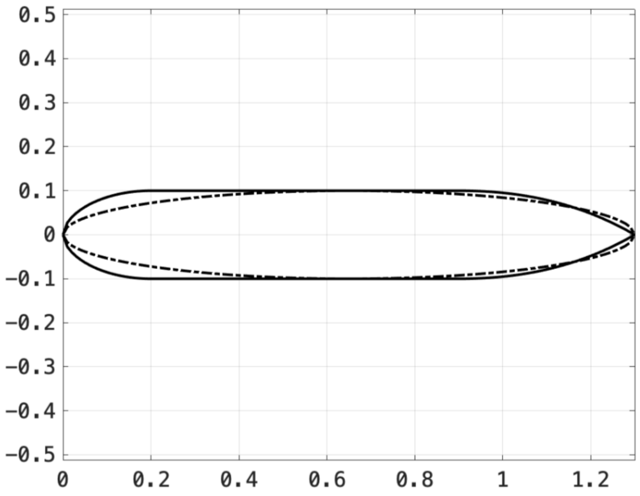
The Light AUV (LAUV):
https://www.oceanscan-mst.com
https://www.ntnu.edu/aur-lab/lauv-harald

- **Hydroid Inc. (2019).** Marine Robots (AUVs). www.hydroid.com
- **L., A. Madureira. A. Sousa, J. Braga, P. Calado, P. Dias, R. Martins, J. Pinto and J. Sousa (2013).** The Light Autonomous Underwater Vehicle: Evolutions and Networking. 2013 MTS/IEEE OCEANS. Bergen, Norway, 10–14 June 2013.
- **D. F. Myring (1976).** A Theoretical Study of Body Drag in Subcritical Axisymmetric Flow. Aeronautical Quarterly 27(3), 186–94.
- **Remus (2019).** Woods Hole Oceanographic Institution. www.whoi.edu

8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

Myring hull profile equations (Myring 1976)

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



Myring-type hull (solid) for D = 0.2 m, L = 1.3 m, Ln = 0.2 m, Lc = 0.7 m, Lt = 0.4 m, n = 1.8, and α_t = 30◦ approximated by a spheroid (dotted line).

Spheroid

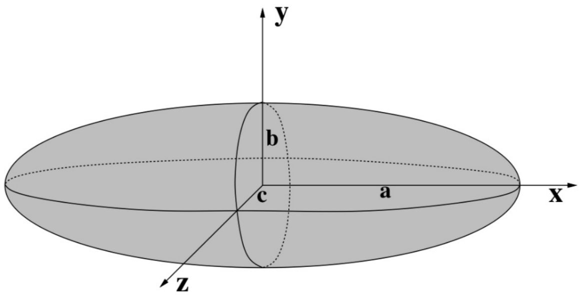
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

D. F. Myring (1976). A Theoretical Study of Body Drag in Subcritical Axisymmetric Flow. Aeronautical Quarterly 27(3), 186–94.

8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

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

A prolate spheroid is obtained by letting *b* = *c* and *a* > *b*



$$m = \frac{4}{3}\pi\rho abc = \frac{4}{3}\pi\rho ab^2$$

$$\text{CB} = \text{CO} \quad \boldsymbol{r}^b_{bb} = [0,\ 0,\ 0]^\top$$

$$\text{CG} \quad \boldsymbol{r}^b_{bg} = [x_g,\ y_g,\ z_g]^\top$$

Rigid-body inertia matrix about the CO

$$\boldsymbol{M}_{RB} = \boldsymbol{H}^\top(\boldsymbol{r}^b_{bg})\, \text{diag}\left\{m,\ m,\ m,\ \frac{2}{5}mb^2,\ \frac{1}{5}m(a^2+b^2),\ \frac{1}{5}m(a^2+b^2)\right\} \boldsymbol{H}(\boldsymbol{r}^b_{bg})$$

Rigid-body Coriolis-centrpetal matrix matrix about the CO

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}_r) = \boldsymbol{H}^\top(\boldsymbol{r}^b_{bg}) \begin{bmatrix} 0 & -mr & mq & 0 & 0 & 0 \\ mr & 0 & -mp & 0 & 0 & 0 \\ -mq & mp & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_z r & -I_y q \\ 0 & 0 & 0 & -I_z r & 0 & I_x p \\ 0 & 0 & 0 & I_y q & -I_x p & 0 \end{bmatrix} \boldsymbol{H}(\boldsymbol{r}^b_{bg})$$

$$I_x = \frac{1}{5}m(b^2+c^2) = \frac{2}{5}mb^2$$

$$I_y = \frac{1}{5}m(a^2+c^2) = \frac{1}{5}m(a^2+b^2)$$

$$I_z = \frac{1}{5}m(a^2+b^2) = I_y$$

MSS Toolbox (Rigid-Body Mass Matrices of a Spheroid)

$$\boldsymbol{M}_{RB} = \boldsymbol{H}^{\top}(\boldsymbol{r}^b_{bg})\,\mathrm{diag}\left\{m,\, m,\, m,\, \frac{2}{5}mb^2,\, \frac{1}{5}m(a^2+b^2),\, \frac{1}{5}m(a^2+b^2)\right\}\boldsymbol{H}(\boldsymbol{r}^b_{bg})$$

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}_r) = \boldsymbol{H}^{\top}(\boldsymbol{r}^b_{bg})\begin{bmatrix} 0 & -mr & mq & 0 & 0 & 0 \\ mr & 0 & -mp & 0 & 0 & 0 \\ -mq & mp & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_z r & -I_y q \\ 0 & 0 & 0 & -I_z r & 0 & I_x p \\ 0 & 0 & 0 & I_y q & -I_x p & 0 \end{bmatrix}\boldsymbol{H}(\boldsymbol{r}^b_{bg})$$

```
>> a = 1; b = 0.7; nu = [1 0 0 0.1 0.2 0.3]'; r_bg = [0 0 0.1]';
>> [MRB,CRB] = spheroid(a,b,nu(4:6),r_bg)
MRB =
   1.0e+03 *
    2.1059         0         0         0    0.2106         0
         0    2.1059         0   -0.2106         0         0
         0         0    2.1059         0         0         0
         0   -0.2106         0    0.4338         0         0
    0.2106         0         0         0    0.6486         0
         0         0         0         0         0    0.6275
CRB =
         0 -631.7617  421.1745   63.1762         0         0
  631.7617         0 -210.5872         0   63.1762         0
 -421.1745  210.5872         0  -21.0587  -42.1174         0
  -63.1762         0   21.0587         0  181.9474 -125.5100
         0  -63.1762   42.1174 -181.9474         0   41.2751
         0         0         0  125.5100  -41.2751         0
```

```
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] = spheriod(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)

$$k_1 = \frac{\alpha_0}{2-\alpha_0} \qquad \alpha_0 = \frac{2(1-e^2)}{e^3}\left(\frac{1}{2}\ln\frac{1+e}{1-e} - e\right) \qquad e := 1-(b/a)^2$$

$$k_2 = \frac{\beta_0}{2-\beta_0} \qquad \beta_0 = \frac{1}{e^2} - \frac{1-e^2}{2e^3}\ln\frac{1+e}{1-e}$$

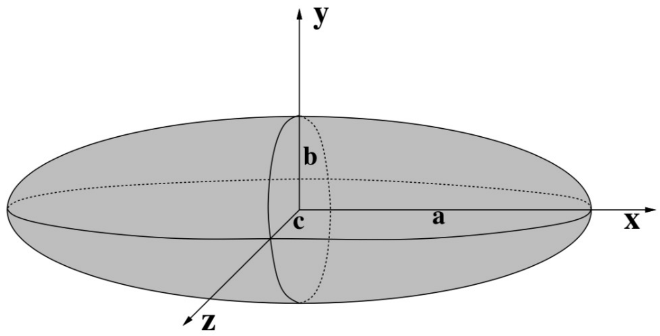
$$k' = \frac{e^4(\beta_0-\alpha_0)}{(2-e^2)[2e^2-(2-e^2)(\beta_0-\alpha_0)]}$$

Added inertia matrix about the CO

$$\begin{aligned}\boldsymbol{M}_A &= -\mathrm{diag}\{X_{\dot u}, Y_{\dot v}, Z_{\dot w}, K_{\dot p}, M_{\dot q}, N_{\dot r}\}\\ &= \mathrm{diag}\{mk_1,\ mk_2,\ mk_2,\ 0,\ k'I_y,\ k'I_y\}\end{aligned}$$

Added Coriolis-centripetal matrix about the CO

$$\boldsymbol{C}_A(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot w}w_r & Y_{\dot v}v_r \\ 0 & 0 & 0 & Z_{\dot w}w_r & 0 & -X_{\dot u}u_r \\ 0 & 0 & 0 & -Y_{\dot v}v_r & X_{\dot u}u_r & 0 \\ 0 & -Z_{\dot w}w_r & Y_{\dot v}v_r & 0 & -N_{\dot r}r & M_{\dot q}q \\ Z_{\dot w}w_r & 0 & -X_{\dot u}u_r & N_{\dot r}r & 0 & -K_{\dot p}p \\ -Y_{\dot v}v_r & X_{\dot u}u_r & 0 & -M_{\dot q}q & K_{\dot p}p & 0 \end{bmatrix}$$



MSS Toolbox (Added Mass Matrices by Imlay 1961)

```
>> a = 1; b = 0.7; nu = [1 0.1 0 0 0 0 ]';
>> [MA,CA] = imlay61(a,b,nu)
MA =
   1.0e+03 *
    0.8774         0         0         0         0         0
         0    1.1487         0         0         0         0
         0         0    1.1487         0         0         0
         0         0         0         0         0         0
         0         0         0         0    0.0091         0
         0         0         0         0         0    0.0091
CA =
         0         0         0         0         0 -114.8662
         0         0         0         0         0  877.4391
         0         0         0  114.8662 -877.4391         0
         0         0 -114.8662         0         0         0
         0         0  877.4391         0         0         0
  114.8662 -877.4391         0         0         0         0
```

Optional parameter to specify added moment in roll

```
>> [MA,CA] = imlay61(a,b,nu,0.5
MA =
   1.0e+03 *
    0.8774         0         0         0         0         0
         0    1.1487         0         0         0         0
         0         0    1.1487         0         0         0
         0         0         0    0.2064         0         0
         0         0         0         0    0.0091         0
         0         0         0         0         0    0.0091
CA =
         0         0         0         0         0 -114.8662
         0         0         0         0         0  877.4391
         0         0         0  114.8662 -877.4391         0
         0         0 -114.8662         0         0         0
         0         0  877.4391         0         0         0
  114.8662 -877.4391         0         0         0         0
```

$$
\begin{aligned}
\boldsymbol{M}_A &= -\mathrm{diag}\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\} \\
&= \mathrm{diag}\{mk_1,\ mk_2,\ mk_2,\ 0,\ k'I_y,\ k'I_y\}
\end{aligned}
$$

$$
\boldsymbol{C}_A(\boldsymbol{\nu}_r) = \begin{bmatrix}
0 & 0 & 0 & 0 & -Z_{\dot{w}}w_r & Y_{\dot{v}}v_r \\
0 & 0 & 0 & Z_{\dot{w}}w_r & 0 & -X_{\dot{u}}u_r \\
0 & 0 & 0 & -Y_{\dot{v}}v_r & X_{\dot{u}}u_r & 0 \\
0 & -Z_{\dot{w}}w_r & Y_{\dot{v}}v_r & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\
Z_{\dot{w}}w_r & 0 & -X_{\dot{u}}u_r & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\
-Y_{\dot{v}}v_r & X_{\dot{u}}u_r & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0
\end{bmatrix}
$$

```
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 semiaxes 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
% midtships. The length of the AUV is L = 2*a while the diamater is D = 2*b.
%
% Inputs: a, b: spheroid semiaxes a > b
%         nu = [u, v, w, p, q, r]': generalized velocity vector
%         r44: hydrodynamic added moment MA(4,4) = r44 * Ix in roll.
%              If r44 is not specified, MA(4,4) = 0.
%              Typicaly values 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)
%
% Refs: Lamb, H. (1932). Hydrodynamics. Cambridge University Press. London.
%       Imlay, F. H. (1961). The Complete Expressions for Added Mass of a
%          Rigid Body Moving in an Ideal Fluid. Technical Report DTMB 1528.
%          David Taylor Model Basin. Washington D.C.
```

8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

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

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

$$F_{\text{drag}} = \frac{1}{2}\rho V_r^2 S C_D(\alpha)$$
$$F_{\text{lift}} = \frac{1}{2}\rho V_r^2 S C_L(\alpha)$$

$$C_D(\alpha) \approx C_{D_0} + C_{D_\alpha}\alpha + C_{D_{\delta_e}}\delta_e$$
$$C_L(\alpha) \approx C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e$$
$$C_Y(\beta) \approx C_{Y_0} + C_{Y_\beta}\beta + C_{Y_{\delta_r}}\delta_r$$

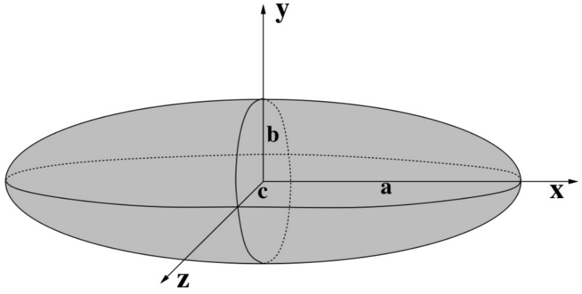
Nonlinear damping about the CO (Beard and McLain 2012)

$$\boldsymbol{d}(\boldsymbol{\nu}_r) = \frac{1}{2}\rho V_r^2 S \begin{bmatrix} C_D(\alpha)\cos(\alpha) - C_L(\alpha)\sin(\alpha) \\ C_Y(\beta) \\ C_D(\alpha)\sin(\alpha) - C_L(\alpha)\cos(\alpha) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Nonlinear damping is important for high-speed flying vehicles, while linear damping dominates at low speed

Linear damping about the CO

$$\boldsymbol{D} = -\text{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\}$$



MSS Toolbox (Lift and Drag of a Submerged "Wing Profile")

```
>> b = 0.8; S = 2; CD_0 = 0.1; alpha = 0.1; U_r = 1;
>> tau_liftdrag = forceLiftDrag(b,S,CD_0,alpha,U_r)
tau_liftdrag =
 -100.5905
         0
  -61.5965
         0
         0
         0
```

The function force_LiftDrag.m calls

[CL,CD] = coeffLiftDrag(b,S,CD_0,alpha,sigma,display)

which computes the hydrodynamic lift CL(alpha) and drag CD(alpha) coefficients as a function of alpha (angle of attack) of a submerged "wing profile".

```
function tau_liftdrag = forceLiftDrag(b,S,CD_0,alpha,U_r)
% tau_liftdrag = forceLiftDrag(b,S,CD_0,alpha,Ur) 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; nu = [1 0 0 0.1 0.2 0.3]'; r_bg = [0 0 0.1]';
>> [MRB,CRB] = spheroid(a,b,nu(4:6),r_bg)
MRB =
   1.0e+03 *
    2.1059         0         0         0    0.2106         0
         0    2.1059         0   -0.2106         0         0
         0         0    2.1059         0         0         0
         0   -0.2106         0    0.4338         0         0
    0.2106         0         0         0    0.6486         0
         0         0         0         0         0    0.6275
CRB =
         0 -631.7617  421.1745   63.1762         0         0
  631.7617         0 -210.5872         0   63.1762         0
 -421.1745  210.5872         0  -21.0587  -42.1174         0
  -63.1762         0   21.0587         0  181.9474 -125.5100
         0  -63.1762   42.1174 -181.9474         0   41.2751
         0         0         0  125.5100  -41.2751         0

>> g = 9.81; m = MRB(1,1); W = m*g; B = W;
>> T1 = 1; T2 = 1;
>> zeta4 = 0.3; zeta5 = 0.4;
>> T6 = 1;
>> r_bg = [ 0 0 0.02 ]'; r_bb = [ 0 0 0 ]';
>> D = Dmtrx([T1 T2 T6],[zeta4 zeta5],MRB,MA,[W r_bg' r_bb'])
D =
   1.0e+03 *
    2.9833         0         0         0         0         0
         0    3.2545         0         0         0         0
         0         0    3.2545         0         0         0
         0         0         0    0.3086         0         0
         0         0         0         0    0.4170         0
         0         0         0         0         0    0.6366
```

Surge, sway and yaw (T1, T2, T6)

$$d = \frac{m}{T}$$

Roll and pitch (zeta4, zeta5)

$$d = 2\zeta\sqrt{km}, \qquad \zeta = \sqrt{1-r^2}$$

Heave T3 = T2 since W = B

$$\boldsymbol{D} = -\text{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\}$$

```
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 rbody.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
```

8.4 Cylinder-Shaped Vehicles and Myring-Type Hulls

Gravitational/buoyancy forces about the CO

$$\text{CB} = \text{CO} \qquad \boldsymbol{r}_{bb}^{b} = [0,\ 0,\ 0]^{\top}$$

$$\text{CG} \qquad \boldsymbol{r}_{bg}^{b} = [0,\ 0,\ z_g]^{\top}$$

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

neutrally buoyant vehicle $W = B$.

6-DOF Equations of Motion (Relative Velocity)

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_k(\boldsymbol{\eta})(\boldsymbol{\nu}_r + \boldsymbol{\nu}_c)$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}\boldsymbol{\nu}_r + \boldsymbol{d}(\boldsymbol{\nu}_r) + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}$$

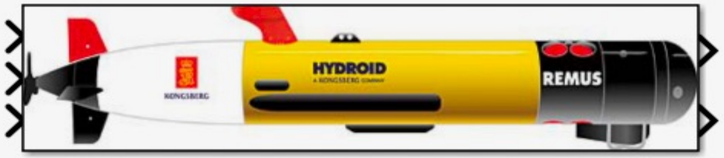
MSS Toolbox (Gravity and Buoyancy)

```
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);

g = [...
   (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       ];
```

$$
g(\eta) = \begin{bmatrix} (W-B)\sin(\theta) \\ -(W-B)\cos(\theta)\sin(\phi) \\ -(W-B)\cos(\theta)\cos(\phi) \\ -(y_g W - y_b B)\cos(\theta)\cos(\phi) + (z_g W - z_b B)\cos(\theta)\sin(\phi) \\ (z_g W - z_b B)\sin(\theta) + (x_g W - x_b B)\cos(\theta)\cos(\phi) \\ -(x_g W - x_b B)\cos(\theta)\sin(\phi) - (y_g W - y_b B)\sin(\theta) \end{bmatrix}
$$

MSS m-file function and Simulink Interface: Remus 100 AUV

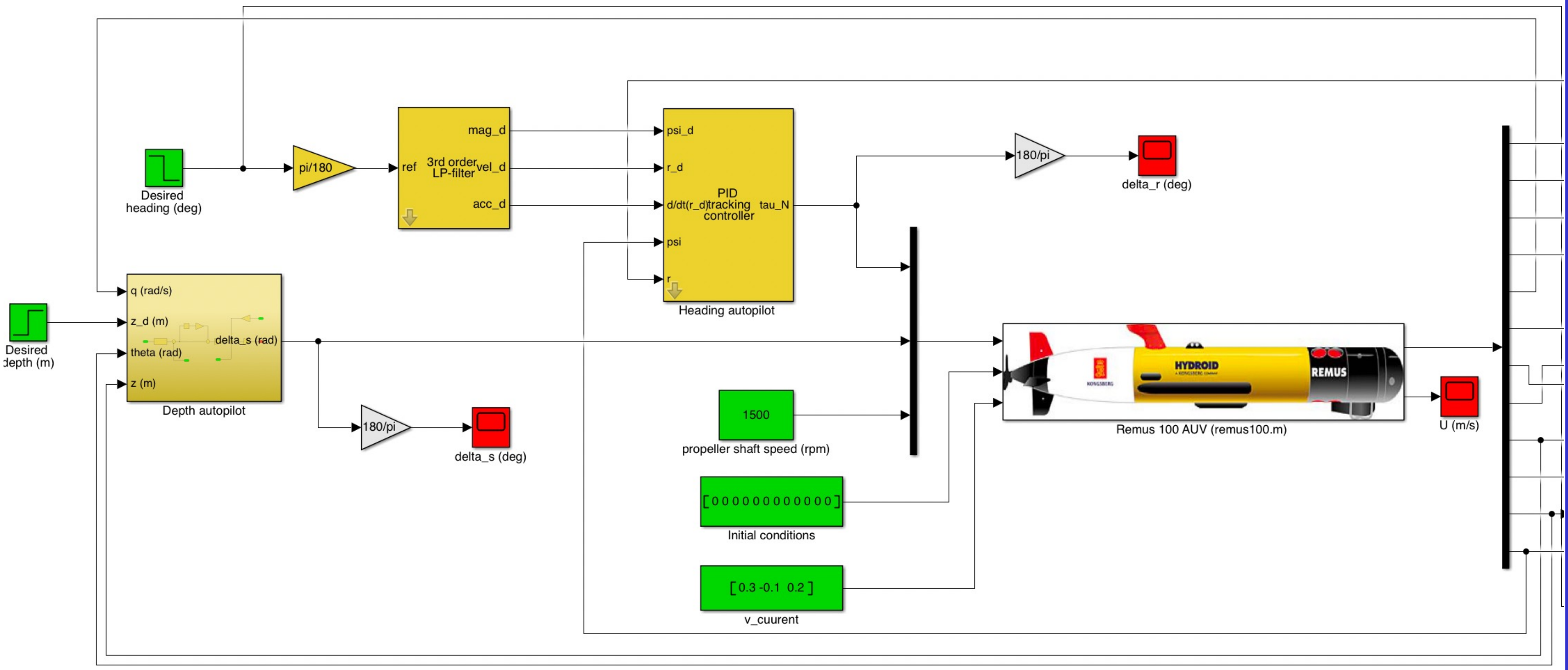
```
function [xdot,U] = remus100(x,ui,v_current)
% [xdot,U] = remus100(x,ui,v_current) returns the time derivative of the
% state vector: x = [ u v w p q r x y z phi theta psi ]' and speed U in m/s
% (optionally) for the Remus 100 autonomous underwater vehicle (AUV). The
% length of the AUV is L = 1.7 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.
```

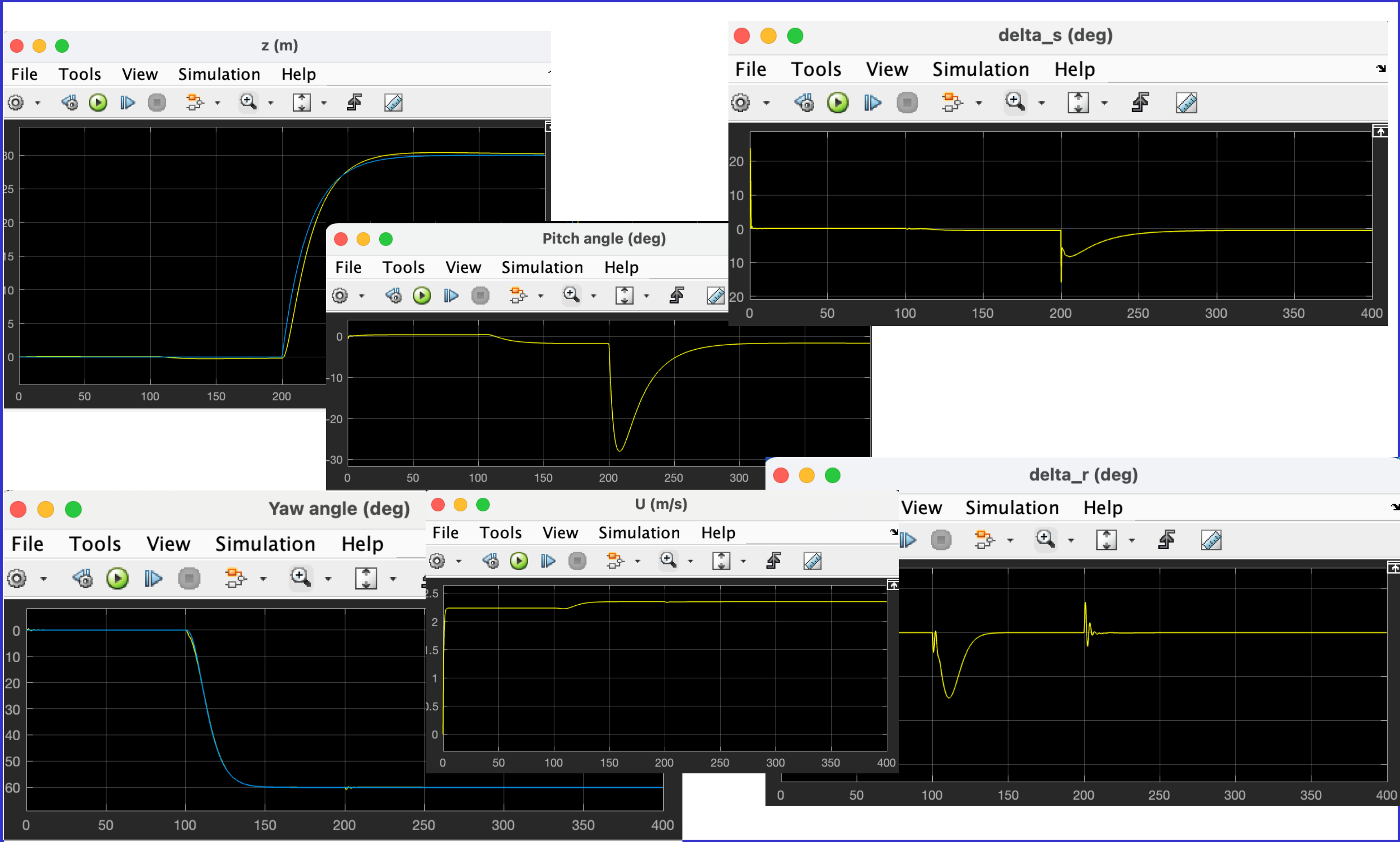


Remus 100 AUV (remus100.m)

mssSImulink library block for numerical integration of the m-file function remus100.m

MSS Simulink: demoAUVdepthHeadingControl





z (m)
Pitch angle (deg)
delta_s (deg)
Yaw angle (deg)
U (m/s)
delta_r (deg)

8.5 Spherical-Shaped Vehicles

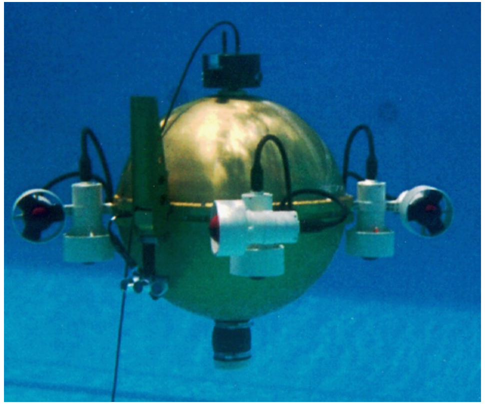
6-DOF Equations of Motion (Relative Velocity)

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_k(\boldsymbol{\eta})(\boldsymbol{\nu}_r + \boldsymbol{\nu}_c)$$
$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}\boldsymbol{\nu}_r + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}$$

Rigid-body matrices about the CO

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & 0 \\ 0 & m & 0 & -mz_g & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & -mz_g & 0 & I_x & 0 & 0 \\ mz_g & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix}$$

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & -mr & mq & mz_g r & 0 & 0 \\ mr & 0 & -mp & 0 & mz_g r & 0 \\ -mq & mp & 0 & -mz_g p & -mz_g q & 0 \\ -mz_g r & 0 & mz_g p & 0 & I_z r & -I_y q \\ 0 & -mz_g r & mz_g q & -I_z r & 0 & I_x p \\ 0 & 0 & 0 & I_y q & -I_x p & 0 \end{bmatrix}$$



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

CB = CO $\quad \boldsymbol{r}^b_{bb} = [0,\ 0,\ 0]^\top$

CG $\quad \boldsymbol{r}^b_{bg} = [0,\ 0,\ z_g]^\top$

$$I_x = I_y = I_z = \frac{2}{5} m R^2$$

Choi, H. T., A. Hanai, S. K. Choi and J. Yuh (2003). Development of an Underwater Robot, ODIN-III. IEEURSJ Int. Conf. on Intelligent Robots and Systems Las Vegas, NV.

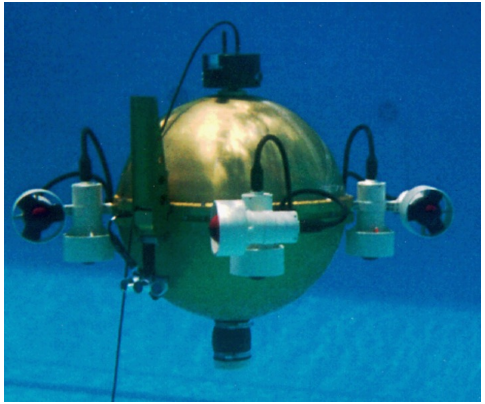
8.5 Spherical-Shaped Vehicles

Added mass about the CO

$$\boldsymbol{M}_A = -\mathrm{diag}\{X_{\dot{u}},\, Y_{\dot{v}},\, Z_{\dot{w}},\, 0,\, 0,\, 0\}$$

$$X_{\dot{u}} = Y_{\dot{v}} = Z_{\dot{w}} = -\rho\frac{2}{3}\pi R^2$$

$$\boldsymbol{C}_A(\boldsymbol{\nu}_r) = \rho\frac{2}{3}\pi R^2 \begin{bmatrix} 0 & 0 & 0 & 0 & w_r & -v_r \\ 0 & 0 & 0 & -w_r & 0 & u_r \\ 0 & 0 & 0 & v_r & -u_r & 0 \\ 0 & w_r & -v_r & 0 & 0 & 0 \\ -w_r & 0 & u_r & 0 & 0 & 0 \\ v_r & -u_r & 0 & 0 & 0 & 0 \end{bmatrix}$$



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

Linear damping (low-speed application)

$$\boldsymbol{D} = -\mathrm{diag}\{X_u,\, Y_v,\, Z_w,\, K_p,\, M_q,\, N_r\}$$

Gravitational/buoyancy forces about the CO

$$\boldsymbol{g}(\boldsymbol{\eta}) = \mathrm{diag}\{0,\, 0,\, 0,\, z_g W\cos(\theta)\sin(\phi),\, z_g W\sin(\theta),\, 0\}$$

neutrally buoyant vehicle $W = B$.

Choi, H. T., A. Hanai, S. K. Choi and J. Yuh (2003). Development of an Underwater Robot, ODIN-III. IEEURSJ Int. Conf. on Intelligent Robots and Systems Las Vegas, NV.

Chapter Goals – Revisited

- 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 are zero.
- 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 built 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