TTK4190 - Assignment 3

Group 27
Herman Kolstad Jakobsen (hermankj)
Aksel Heggernes (akselheg)
Sondre Holm Fyhn (sondrehf)
Iver Myklebust (iverom)

October 12, 2020

1 Problem 1 - Rigid-Body Kinetics of a Rectangular Prism

1.1 Problem 1.a

The numerical values of the rectangular prism are $L = 161, B = 21.8$ and $H = 15.8$. CG is located in the center of the body and $m = \rho_m LBH$, but the we are assuming the prism and ship have the same mass so $m = 17.0677 \cdot 10^6$. The moment of inertia I_z^{CG} about the the center of gravity is given by

$$I_z^{CG} = \int_V (x^2 + y^2)\rho_m dV = \frac{1}{12}m(L^2 + B^2) = 5.438 \cdot 10^{10} \tag{1}$$

Since CG is located in the volume center, the products of inertia $I_{xy}^{CG} = I_{yx}^{CG} = 0$ (this also yields for the other non-diagonal elements of the inertia matrix). This is because the body of the rectangular box is symmetrical about the axes of rotation.

1.2 Problem 1.b

The distance vector from CO on the body to CG is given by

$$r_{bg}^b = [x_g = -3.7, y_g = 0, z_g = H/2] \tag{2}$$

The parallell axis theorem states that y

$$I_z^{CO} = I_z^{CG} + m\sqrt{3.7^2 + \frac{H}{4}^2} = \frac{m}{12}(L^2 + B^2) + m\sqrt{3.7^2 + \frac{H}{4}^2} = 3.769 \cdot 10^{10} \tag{3}$$

The ratio between the moments of inertia for the prism and the real ship is

$$\frac{I_{z,prism}^{CO}}{I_{z,ship}^{CO}} = \frac{3.769 \cdot 10^{10}}{2.1732 \cdot 10^{10}} \approx 1.73 \tag{4}$$

There are approximately 73% difference between the moments of inertia, so it is up for discussion of how good the prism works as a proxy for the ship.

1.3 Problem 1.c

The state vector is generally given as

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{5}$$

and the rigid body kinetics about CO can then be expressed as

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu = \tau_{RB} \tag{6}$$

Hence, we are only interested in row 1,2 and 6 of M_{RB} and C_{RB}. Using eq (3.49) from Fossen [1], we get

$$M_{RB} = \begin{bmatrix} m & 0 & my_g \\ 0 & m & mx_g \\ -my_g & mx_g & I_z^{CO} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & -3.7m \\ 0 & -3.7m & I_z^{CO} \end{bmatrix} \tag{7}$$

The Coriolis matrix can be found using eq 3.62 in the same book

$$C_{RB} = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & 0 \end{bmatrix} \tag{8}$$

1.4 Problem 1.d

We have that $C_{RB} = -C_{RB}^T \implies$ skew-symmetric. This property is very useful when designing a nonlinear motion control system since the quadratic form $\nu^T C_{RB}(\nu)\nu \equiv 0$ and can be exploited in energy based designs and nonlinear observer design.

1.5 Problem 1.e

When ocean currents (irrotational) enter the equations of motion, the Coriolis matrix C_{RB}^{CO} does not depend on linear velocity ν_1. This is an useful property since we can do a linear velocity-independent parametrization.

2 Problem 2 - Hydrostatics

2.1 Problem 2.a

Archimedes law yields

$$mg = \rho g \nabla \tag{9}$$

which means

$$\nabla = \frac{m}{\rho} \tag{10}$$

Assuming that the real ship and the prism have equal mass, we get

$$\nabla = \frac{17.0677 \cdot 10^6}{1025} = 16651.41[m^3] \tag{11}$$

2.2 Problem 2.b

For a prism, the waterplane area for is simply given by

$$A_{wp} = L * B = 161 \cdot 21.8 = 3509.8[m^2] \tag{12}$$

The hydrostatic force in heave will be the difference between he gravitational and the buoyancy forces

$$z_{hs} = mg - \rho g(\nabla + \delta\nabla(z^n)) = -\rho g \delta\nabla(z^n) \tag{13}$$

where z^n is the displacement in heave and $\delta\nabla$ is change in displaced water. Assuming that the waterplane area is constant for small perturbations in vertical position, we get

$$\delta\nabla(z^n)) = A_{wp} z^n \tag{14}$$

The hydrostatic force can then be stated as

$$z_{hs} = -\delta g A_{wp} z^n \tag{15}$$

2.3 Problem 2.c

The heave period is given by

$$T_3 = 2\pi\sqrt{\frac{2T}{g}} = 2\pi\sqrt{\frac{2 \cdot 8.9}{9.81}} \approx 8.46s \tag{16}$$

2.4 Problem 2.d

The metacentric height GM_i, where $i \in \{T, L\}$, is the distance between the metacenter M_i and the CG. The second moment of areas are

$$I_L = \int\int_{A_{wp}} x^2 dA = \int_{-\frac{B}{2}}^{\frac{B}{2}} \int_{-\frac{L}{2}}^{\frac{L}{2}} x^2 dx dy = [y]_{\frac{-B}{2}}^{\frac{B}{2}} [\frac{1}{3}x^3]_{\frac{-L}{2}}^{\frac{L}{2}} = \frac{B \cdot L^3}{12} \tag{17}$$

Can derive from figure 1 that

$$BG = \frac{H}{2} - \frac{T}{2} = +\frac{15.8}{2} - \frac{8.9}{2} = 3.45 \tag{18}$$

For small roll and pitch angles, the tranverse and longitudinal radius of curvature can be approximated by

$$BM_T = \frac{I_T}{\nabla} = \frac{\rho}{m} \cdot \frac{B^3 L}{12} \tag{19}$$

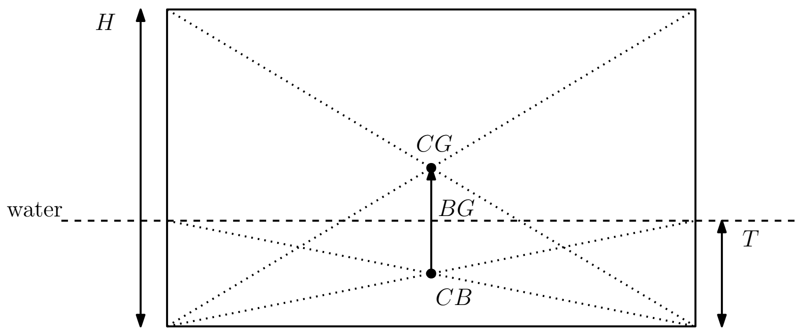

Figure 1: Metacenter

$$BM_L = \frac{I_L}{\nabla} = \frac{\rho}{m} \cdot \frac{BL^3}{12} \tag{20}$$

The metacenter height can then be computed as

$$GM_T = BM_T - BG = \frac{I_T}{\nabla} - 3.5 \approx 1 \tag{21}$$

$$GM_L = BM_L - BG = \frac{I_L}{\nabla} - 3.5 \approx 239 \tag{22}$$

2.5 Problem 2.e

Yes, with $GM_T \approx 1$ and $GM_L \approx 239$, the floating vessel is metacentrically stable since it is both tranverse metacentrically stable

$$GM_T \geq GM_{T,min} \geq 0 \tag{23}$$

and longitudinal metacentrically stable

$$GM_L \geq GM_{L,min} \geq 0 \tag{24}$$

3 Problem 3 - Added Mass and Coriolis

3.1 Problem 3.a)

By only looking at the surge, sway and yaw components, the added mass matrix is reduced to

$$\mathbf{M}_A = \mathbf{M}_A^T = -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix} \tag{25}$$

3.2 Problem 3.b)

The Coriolis force matrix due to added mass with components corresponding to surge, sway and yaw are found using *Property 6.2* in Fossen [1]

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & Y_{\dot{u}}u + Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u - X_{\dot{v}}v - X_{\dot{r}}r \\ -Y_{\dot{u}}u - Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{r}}r & 0 \end{bmatrix} \tag{26}$$

4 Problem 4 - Implementing the Maneuvering Model in Matlab

The references that are made in this section to the MATLAB code can be further studied in the attachments to this assignment, *p4a.m* and *p4b.m*

4.1 Problem 4.a)

The rigid body matrices $\mathbf{M}_{RB}$ and $\mathbf{C}_{RB}$ are already implemented in the template *project.m* that is handed out with the assignment. In this task we also include $\mathbf{M}_A$ and $\mathbf{C}_A$ in the code. The calculation of the Coriolis force matrix, $\mathbf{C}_A$, happens inside the main loop.

```
% define default values for other added masses
Xvdot = 0;
Xrdot = 0;
Yudot = 0;
Nudot = 0;

% rigid-body mass matrix
MRB = [ m 0     0
        0 m     m*xg
        0 m*xg Iz ];
MRBinv = inv(MRB);

% added mass
MA = -[ Xudot   Xvdot   Xrdot
        Yudot   Yvdot   Yrdot
        Nudot   Nvdot   Nrdot];

M = MRB + MA;
Minv = inv(M);
```

Listing 1: $\mathbf{M}_A$ implementation

```
    % added mass coriolis from property 6.2
    a1 = Xudot*nu(1)+Xvdot*nu(2)+Xrdot*nu(3);
    a2 = Yudot*nu(1)+Yvdot*nu(2)+Yrdot*nu(3);

    CA = [  0       0   a2
            0       0   -a1
           -a2      a1  0   ];

    C = CRB+CA;
```

Listing 2: $\mathbf{C}_A$ implementation

4.2 Problem 4.b)

In this task the linear damping matrix, the nonlinear surge damping and the cross-flow drag is added to the code from the previous task.

```
% linear damping
Xu = -(m-Xudot)/T1;
Yv = -(m-Yvdot)/T2;
Nr = -(Iz-Nrdot)/T6;

D = -diag([Xu Yv Nr]);  % linear damping matrix
```

Listing 3: Linear damping matrix $\mathbf{D}$ implementation

```
% nonlinear surge damping (no current => u_r = u)
k = 0.1;                                  % some magical constant
epsilon = 0.001;                          % small number to ensure Cf is well ...
    defined
visc = 1e-6;                              % kinematic viscosity (m/s^2)
Rn = L/visc*abs(nu(1));                   % reynolds number
Cf = 0.075/((log10(Rn)-2)^2+epsilon);     % flat plate friction

X = -1/2*rho*S*(1+k)*Cf*nu(1)*abs(nu(1));    % nonlinear surge damping

% cross-flow drag
% Strip theory: c r o s s flow drag integrals
Yh = 0;      % start value of integral
Nh = 0;      % start value of integral
dx = L/10;   % 10 strips
for xL = -L/2:dx:L/2
    Ucf = abs(nu(2) + xL * nu(3)) * (nu(2) + xL * nu(3));
    Yh = Yh - 0.5 * rho * T * Cd_2D * Ucf * dx;          % sway force
    Nh = Nh - 0.5 * rho * T * Cd_2D * xL * Ucf * dx;     % yaw moment
end

% nonlinear damping matrix
Dn = diag([X Yh Nh]);
```

Listing 4: Nonlinear surge damping and cross-flow drag implementation

We must also update $\dot{\nu}$ to include the linear and nonlinear damping matrices.

```
nu_dot = Minv * (tau - C * nu - D * nu - Dn * nu);
```

Listing 5: Updated calculation of $\dot{\nu}$

4.3 Problem 4.c)

We can see in fig. 2 that without the linear and nonlinear damping matrices we are not able to hold a steady course. The yaw angle, yaw rate and surge is also not as we want it. If we look at fig. 3, we see that we are able to hold a somewhat steady course. However, we can also see that the actual yaw angle is relatively far off the desired yaw angle (even though the actual and desired yaw rate equals over time). This is because we don't have guidance controllers in our system. This is also the reason why surge velocity is steadily increasing.

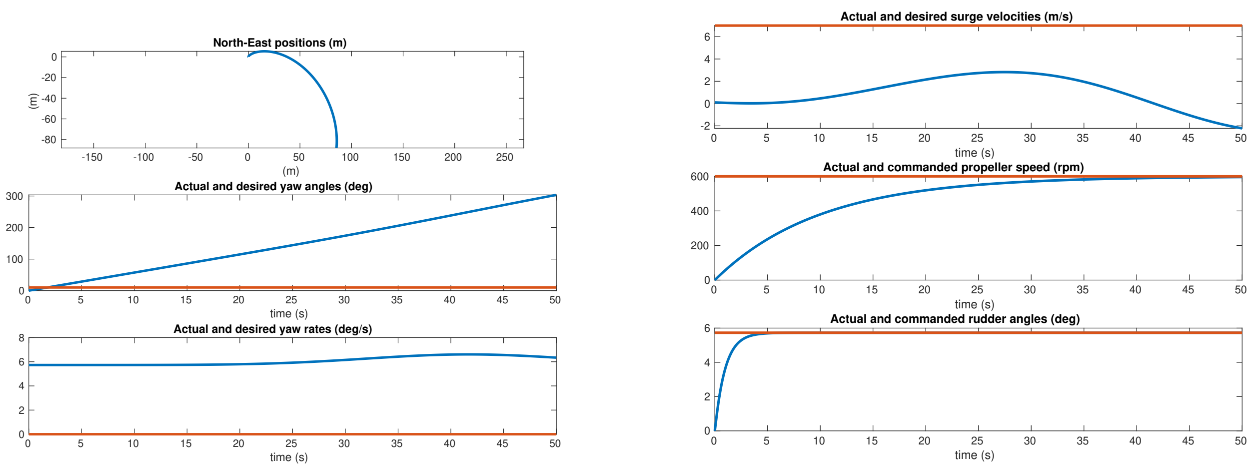

Figure 2: Plots for section 4.1

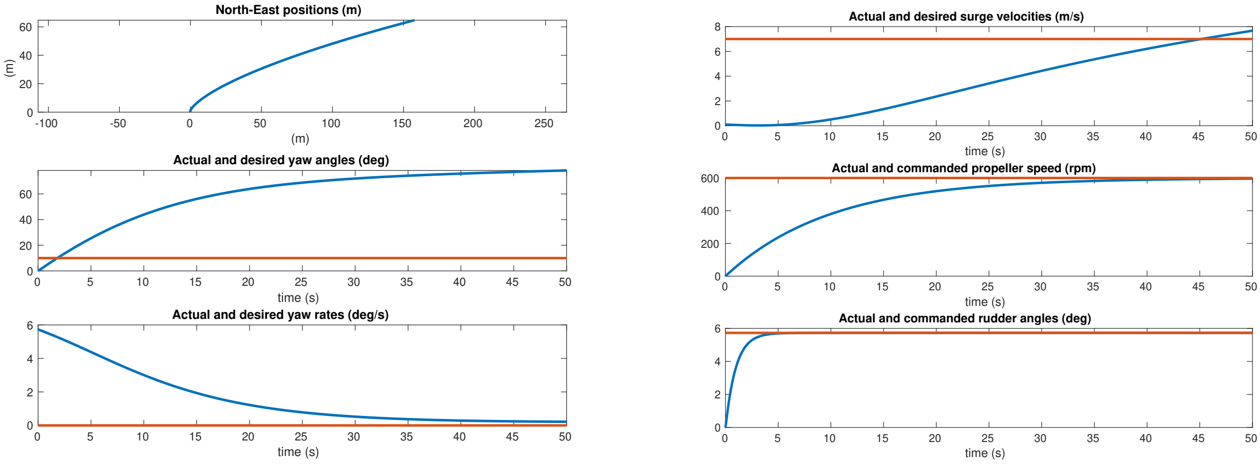

Figure 3: Plots for section 4.2

References

[1] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2011.