TTK4190 Guidance and Control of Vehicles

Assignment 2, Pt. 1

Written Fall 2023

Seknaya, Sacit Ali
sacitas@stud.ntnu.no

Kopperstad, Håvard Olai
haavarok@stud.ntnu.no

Almenningen, Elias Olsen
eliasoa@stud.ntnu.no

October 31, 2023

Problem 1 - Rigid-Body Kinetics of a Rectangular Prism

a)

The mass of the rectangular prism is given by $m = \rho_m LBH$, where ρ_m denotes the density of the body. The dimesions of the prism are given as: $L = 161, B = 21.8, H = 15.8$

The density can then be calculated as

$$\rho_m = \frac{m}{LBH} = 307.777\ldots \tag{1}$$

The moment of inertia I_z^{CG} about the prism's center of gravity (CG) is found in [1] as

$$I_z^{CG} = \int_V (x^2 + y^2)\rho_m dV \tag{2}$$

which is a tripple integral. The CG is found by dividing the dimensions of the prism by 2, since the mass is homogeneously distributed. Rewriting eq. 2 gives us

$$I_z^{CG} = \int_{\frac{-L}{2}}^{\frac{L}{2}} \int_{-\frac{B}{2}}^{\frac{B}{2}} \int_{-\frac{H}{2}}^{\frac{H}{2}} \left(x^2 + y^2\right) \rho_M dzdydx \tag{3}$$

and solving it symbolic and inserting numerical values gives us:

$$I_z^{CG} = \frac{1}{12} \underbrace{LBH\rho_m}_{m}(L^2 + B^2) = 3.7544e + 10 \tag{4}$$

b)

The distance from CO to CG is $r_{bg}^b = [x_g = -3.7, y_g = 0, z_g = H/2]$. The ineria about the z-axis I_z^{CO} is then calculated using the paralell-axis theorem (eq. 3.36 in [1]):

$$I_{CO}^b = I_{CG}^b - mS^2(r_{bg}^b) \tag{5}$$

where S is the skew-symmetric matrix operator. Extracting the element corresponding to I_z gives us in eq. (3.37) gives us the formula

$$I_z^{CO} = I_z^{CG} + m((r_{bg,x}^b)^2 + (r_{bg,y}^b)^2) = 3.7777e + 10 \tag{6}$$

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

$$\frac{I_{z,prism}^{CO}}{I_{z,boat}^{CG}} = \frac{3.7777e + 10}{2.1732e + 10} = 1.7383 \tag{7}$$

c)

$\boldsymbol{M}_{RB}$ is defined in eq. (3.49) in [1] as

$$\begin{aligned}
\boldsymbol{M}_{RB} &= \begin{bmatrix} m\boldsymbol{I}_3 & -m\boldsymbol{S}\left(\boldsymbol{r}_{bg}^b\right) \\ m\boldsymbol{S}\left(\boldsymbol{r}_{bg}^b\right) & \boldsymbol{I}_b^b \end{bmatrix} \\
&= \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix}
\end{aligned} \tag{8}$$

where $\boldsymbol{I}_b^b = I^{CO}$. Extracting the parts belonging to DOF 1,2 and 6 using MatLab:

```
% Find M_RB for DOF 1,2 and 6
% Eq 3.49 p.64 in Fossen
M_RB = [    m*eye(3)    -m*Smtrx(r_bg);
         m*Smtrx(r_bg)        I_CO ];
% Extracting DOF 1,2 and 6 from M_RB
M_RB_DOF = [M_RB(1,1) M_RB(1,2) M_RB(1,6);
            M_RB(2,1) M_RB(2,2) M_RB(2,6);
            M_RB(6,1) M_RB(6,2) M_RB(6,6)]
```

which results in

$$\begin{aligned} \boldsymbol{M}_{RB,3DOF} &= \begin{bmatrix} m & 0 & -my_g \\ 0 & m & mx_g \\ -my_g & mx_g & I_z \end{bmatrix} \\ &= \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \end{aligned} \tag{9}$$

The *linear velocity-independent parameterization* of $\boldsymbol{C}_{RB}$ is defined in eq. (3.63) in [1] as

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} m\boldsymbol{S}(\boldsymbol{\nu}_2) & -m\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{S}\left(\boldsymbol{r}_{bg}^b\right) \\ m\boldsymbol{S}\left(\boldsymbol{r}_{bg}^b\right)\boldsymbol{S}(\boldsymbol{\nu}_2) & -\boldsymbol{S}\left(\boldsymbol{I}_b^b\boldsymbol{\nu}_2\right) \end{bmatrix} \tag{10}$$

where $\boldsymbol{\nu}_2 = [p, q, r]^\top$. Redefining $\boldsymbol{I}_b^b = diag(I_x, I_y, I_z)$ and using MatLab

```
% Find C_RB for DOF 1,2 and 6
% Eq 3.63 p.68 in Fossen
C_RB = [    m*Smtrx(nu2)               -m*Smtrx(nu2)*Smtrx(r_bg);
        m*Smtrx(r_bg)*Smtrx(nu2)            -Smtrx(I_CO*nu2)];

% Extracting DOF 1,2 and 6 from C_RB
C_RB_lin_DOF = [C_RB(1,1) C_RB(1,2) C_RB(1,6);
                C_RB(2,1) C_RB(2,2) C_RB(2,6);
                C_RB(6,1) C_RB(6,2) C_RB(6,6)]
```

which results in

$$\begin{aligned} \boldsymbol{C}_{RB,3DOF}(\boldsymbol{\nu}) &= \begin{bmatrix} 0 & -mr & -mrx_g \\ mr & 0 & -mry_g \\ mrx_g & mry_g & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -mr & -mrx_g \\ mr & 0 & 0 \\ mrx_g & 0 & 0 \end{bmatrix} \end{aligned} \tag{11}$$

d)

The $C_{RB,3DOF}^{CO}(\boldsymbol{\nu})$ matrix can be verified to be skew-symmetric through Definition 2.2 in [1] :

$$\boldsymbol{S}(\boldsymbol{\lambda}) = -\boldsymbol{S}^T(\boldsymbol{\lambda}) \tag{12}$$

giving

$$C_{RB,3DOF}^{CO}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & -mr & -mrx_g \\ mr & 0 & -mry_g \\ mrx_g & mry_g & 0 \end{bmatrix} = - \begin{bmatrix} 0 & -mr & -mrx_g \\ mr & 0 & -mry_g \\ mrx_g & mry_g & 0 \end{bmatrix}^\top$$

"Having the coriolis matrix skew-symmetric is a desired property because when designing a non-linear motion conrol system, the quadratic form $\boldsymbol{\nu}^\top C_{RB}\boldsymbol{\nu} \equiv 0$. This is exploited in energy-based designs where Lyapunov functions play a role" [1].

e)

The other property of $C_{RB}^{CO}(\boldsymbol{\nu})$ that is useful when irrotational ocean currents enter the equations of motion is that the same matrix can be used as if there were no ocean currents. I.e.

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} \equiv \boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r \tag{13}$$

[1].

Problem 2 - Hydrostatics of a Rectangular Prism

a)

The buoyancy and weight of a floating vessel at rest is according to Arhchimedes

$$mg = \rho g \nabla \tag{14}$$

Solving for ∇ gives

$$\nabla = \frac{m}{\rho} = 1.6651e + 04 \tag{15}$$

$$\begin{aligned} \nabla &= A_{wp}T \\ &= LBT = 1.6636e + 04 \end{aligned} \tag{16}$$

b)

It is stated in [1] that $\nabla \approx A_{wp}T$, where T is the draft. Solving for A_{wp} gives

$$A_{wp} = \frac{\nabla}{T} = 3.5098e + 3 \tag{17}$$

Is this just the same as $A_{wp} = LB = 3.5098e + 3$

The hydrostatic force Z_{hs} is defined as eq. (4.12) in [1] as

$$Z_{hs} = -\rho g \delta \nabla (z^n) \tag{18}$$

It can also be approximated (eq. (4.14) in [1]) as

$$Z_{hs} \approx -\rho g A_{wp} z^n := Z_z z^n \tag{19}$$

where $Z_z = -\rho g A_{wp}$

c)

The period of Heave is calculated using eq. (4.78) in [1]

$$T_3 \approx 2\pi \sqrt{\frac{2T}{g}} = 6.1766[s] \tag{20}$$

d)

The analytical expressions for GM_T and GM_L are

$$\begin{aligned} GM_T &= BM_T - BG \\ GM_L &= BM_L - BG \end{aligned} \tag{21}$$

The Keel line for the prism is $KG = \frac{H}{2} = 7.9m$

The following method, known as Morrish's formula, gives a reasonably accurate estimate of KB [1]

$$KB = \frac{1}{3}\left(\frac{5T}{2} - \frac{\nabla}{A_{wp}}\right) \tag{22}$$

$$BG = KG - KB \tag{23}$$

For small inclinations ϕ and θ the transverse and longitudinal radii of curvature can be approximated by [1]

$$\begin{aligned} BM_T &= \frac{I_T}{\nabla} \quad where \quad I_T = \frac{B^3 L}{12} \\ BM_L &= \frac{I_L}{\nabla} \quad where \quad I_T = \frac{L^3 B}{12} \end{aligned} \tag{24}$$

Following Example 4.2 in [1] using MatLab

```
%% Problem 2c Finding GM_T and GM_L
% Following example 4.2 in Fossen
KG = 7.9;    % Disance from keel to CG(m)
% Eq. (4.38) in Fossen
KB = 1/3*((5*T)/2 - nabla/A_wp);
BG = KG-KB;

I_T = 1/12*B^3*L;
I_L = 1/12*L^3*B;

BM_T = I_T/nabla;
BM_L = I_L/nabla;

GM_T = BM_T - BG
GM_L = BM_L - BG
```

```
GM_T =

    2.8176
GM_L =

  449.7743
```

e)

According to Definition 4.2 in [1], a floating vessel is said to be tranvsverse metacentrically stable if

$$GM_T \geq GL_{T,min} > 0 \tag{25}$$

and longitudinal metacentrically stable if

$$GM_L \geq GL_{L,min} > 0 \tag{26}$$

[1] states that for most ships $GM_{T,min} > 0.5$ while for $GM_{L,min}$ its usually much larger (over 100). This is the case for our rectangular box, so its metacentrically stable.

Problem 3

a)

Several assumptions must be made to reduce the general 6x6 matrix of added mass coefficients into a 3x3 matrix consisting of surge, sway and yaw components only. The surge mode can be decoupled from the steering dynamics due to xz plane symmetry. Furthermore, the heave, roll and pitch modes are neglected under the assumption that these motion variables are small. Lastly, assuming that the sway-yaw subsystem have no coupling to the two others, the 3x3 matrix can be written as:

$$\boldsymbol{M}_A = -\boldsymbol{M}_A^T = -\begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix} = -\begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & Y_{\dot{r}} & N_{\dot{r}} \end{bmatrix} \tag{27}$$

where $N_{\dot{v}} = Y_{\dot{r}}$ from eq. (6.56) in [1].

b)

To find the Coriolis force matrix $\boldsymbol{C}_A(\boldsymbol{\nu}_r)$ due to added mass, the same assumptions as in problem 1a) stands, thus eq. (6.57) in [1] gives:

$$\boldsymbol{C}_A(\boldsymbol{\nu}_r) = -\boldsymbol{C}_A^T(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v_r + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u_r \\ -Y_{\dot{v}}v_r - Y_{\dot{r}}r & X_{\dot{u}}u_r & 0 \end{bmatrix} \tag{28}$$

Problem 4

a)

Utilizing eq. (6.54) and (6.55) in [1], the 3x3 mass matrix $\boldsymbol{M}$ and the 3x3 Coriolis matrix $\boldsymbol{C}(\boldsymbol{\nu}_r)$ can be implemented in the Matlab code as:

```
MA = - [Xudot    0      0;
            0    Yvdot  Yrdot;
            0    Nvdot  Nrdot];
M = MRB + MA;

CA = [            0                0       vdot*x(2)+Yrdot*x(3);
                  0                0           -Xudot*x(1);
       -Yvdot*x(2)-Yrdot*x(3)  Xudot*x(1)           0];

C = CRB + CA;
```

b)

A diagonal linear damping matrix $\boldsymbol{D}$ can be added by utilizing eq. (6.67) in [1] assuming that the heave, roll and pitch modes are neglected due to these motion variables being small, where $Y_r = N_v = 0$:

$$\boldsymbol{D} = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \tag{29}$$

In Matlab, the linear damping matrix is found by:

```
    T1 = 20; % Natural periods in surge, sway and yaw (Fossen Ch. 4.3)
    T2 = T1;
    T6 = 10;
    Xu = -(m-Xudot)/T1;
    Yv = -(m-Yvdot)/T2;
    Nr = -(Iz-Nrdot)/T6;
    D = -[Xu 0  0;
          0  Yv 0;
          0  0  Nr];
```

A nonlinear damping vector $\boldsymbol{d}(\boldsymbol{\nu}_r)$ containing the nonlinear surge damping and the two cross flow drags can be found utilizing eq. (6.81), (6.87), (6.88) and (6.116) in [1] and assuming $T(x) = T$ and $C_d^{2D}(x)$ is constant:

```
    % Nonlinear surge damping
    k = 1;
    C_R = 0;
    epsilon = 0.001;
    S = 2 * L * T + 2 * B * T + L * B;
    v = 1e-6;
    R_n = L / v * abs(x(1));
    CF = 0.075 / ((log10(R_n) - 2)^2 + epsilon);
    Cf = CF + C_R;
    X = -0.5 * rho * S * (1+k) * Cf * abs(x(1)) * x(1);

    % Adding cross-flow drag:
    Cd_2D = Hoerner(B,T);
    Ycf = 0;
    Ncf = 0;

    % Strip theory: cross-flow drag integrals
    dx = L/10;
    for xL = -L/2:dx:L/2
```

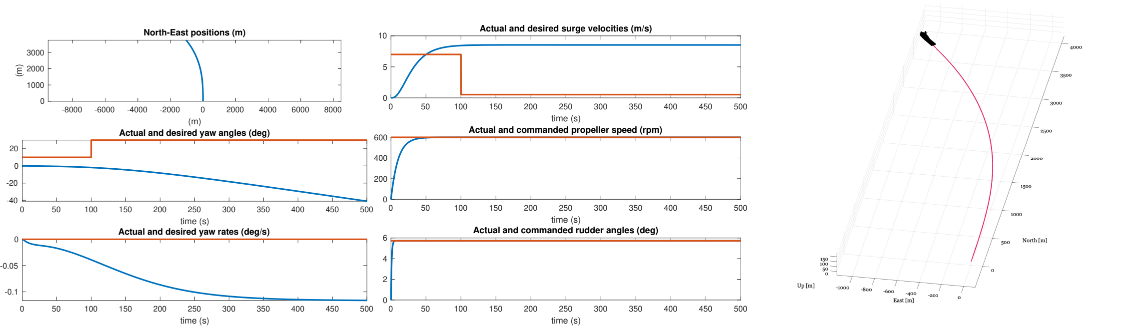
```
        Ucf = abs(x(2) + xL * x(3)) * (x(2) + xL * x(3));
        Ycf = Ycf - 0.5 * rho * T * Cd_2D * Ucf * dx;
        Ncf = Ncf - 0.5 * rho * T * Cd_2D * xL * Ucf * dx;
    end

    % Nonlinear damping given as vector
    d = [X -Ycf -Ncf]';

    % Ship dynamics
    u = [ thr Δ ]';
    tau = Bi * u;
    N = C + D;
    nu_dot = Minv * (tau - N * nu - d);
    eta_dot = R * nu;
```

c)

In Figure 1 the simulation of the maneuvering model is given with step responses in surge and yaw at $t = 100$:



(a) Step response in yaw (b) Step response in surge (c) The trajectory of the boat

Figure 1: The step responses

The step responses can be implemented in Matlab as:

```
    if t > 100
        psi_d = 30 * pi/180;
        u_d = 30 * pi/180;
    else
        psi_d = psi_ref;
        u_d = u_ref;
    end
```

It seems that step responses in surge and yaw have no effect on the trajectory of the ship.

References

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