TTK4190 - Assignment 3

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

October 30, 2020

Contents

1 Part 2: Heading Autopilot

1.1 Problem 1 - Environmental Disturbances

1.1.1 Problem 1.a

With $\beta_{V_c} = 45°$ and $V_c = 1$, and from using the following equations from [1]

$$u_c = V_c cos(\beta_{V_c} - \psi)$$
$$v_c = V_c sin(\beta_{V_c} - \psi)$$

```
for i=1:Ns+1

    t = (i-1) * h;                      % time (s)
    R = Rzyx(0,0,eta(3));

    % current (should be added here)
    nu_c(1) = Vc*cos(betaVc - eta(3));  % surge current
    nu_c(2) = Vc*sin(betaVc - eta(3));  % sway current

    nu_r = nu - nu_c;
```

Listing 1: MATLAB code for 2D irrotational ocean current in surge and sway

and the resulting plots become

1.1.2 Problem 1.b

With zero ocean currents (and wind) affecting the vessel, there will be no crab or sideslip angle. This will result in the vessel following a straight path.

1.1.3 Problem 1.c

The angle of attack between the wind and the boat is given as

$$\gamma_w = \psi - \beta_{V_w} - \pi \tag{1}$$

The wind-coefficients are approximated as in [1] in sway and yaw direction as

$$C_Y(\gamma_w) \approx c_y \sin(\gamma_w) \tag{2}$$

$$C_N(\gamma_w) \approx c_n \sin(2\gamma_w) \tag{3}$$

The moment

$$\tau_{wind} = \begin{bmatrix} \tau_{Y_{wind}} \\ \tau_{N_{wind}} \end{bmatrix} = \begin{bmatrix} \frac{\rho_a {V_{ra}}^2}{2} C_Y(\gamma_{rw}) A_{L_w} \\ \frac{\rho_a {V_{ra}}^2}{2} C_N(\gamma_{rw}) A_{L_w} L_{oa} \end{bmatrix} \tag{4}$$

```
    if t > 200
        % UNSURE ABOUT RELATIVE VS ABSOLUTE WIND SPEED AND ANGLE-OF-ATTACK
        gamma_w = eta(3) - betaVw - pi;
        C_Ywind = cy*sin(gamma_w);             % sway wind coefficient
        C_Nwind = cn*sin(2*gamma_w);           % yaw wind coefficient
        Ywind = 1/2*rho_a*Vw^2*C_Ywind*A_Lw;   % expression for wind moment in ...
            sway should be added.
        Nwind = 1/2*rho_a*Vw^2*C_Nwind*A_Lw*L; % expression for wind moment in ...
            yaw should be added.
    else
        Ywind = 0;
        Nwind = 0;
    end
    tau_env = [0 Ywind Nwind]';
```

Listing 2: Wind movement in sway and yaw.

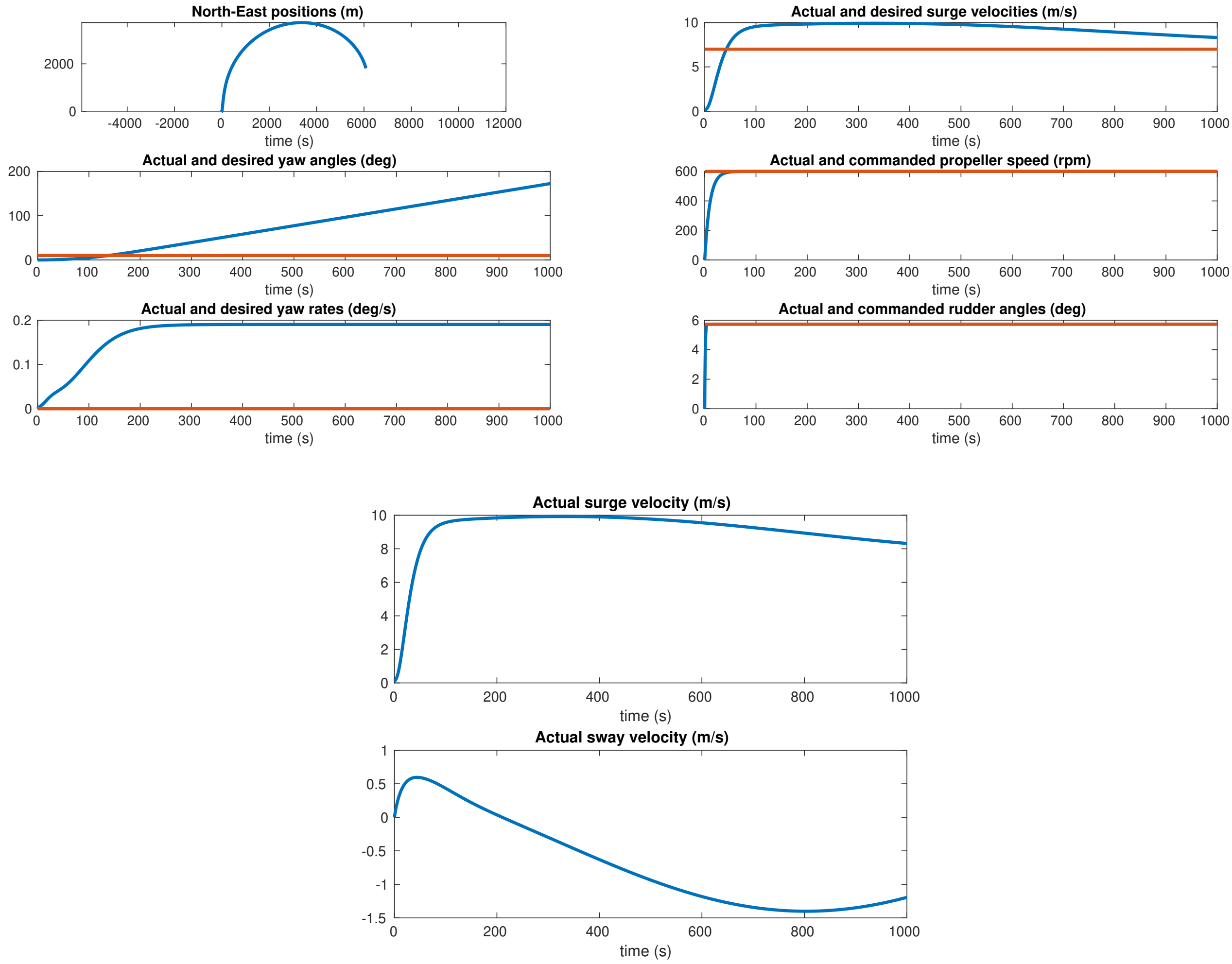

Figure 1: Boat behaviour with current disturbance

1.2 Problem 2 - Heading Autopilot

1.2.1 Problem 2.a

From the given matlab code we get that

$$\mathbf{C}_A = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{bmatrix} \tag{5}$$

$$C_{RB} = \begin{bmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{bmatrix} \tag{6}$$

and the nonlinear coriolis forces become

$$\mathbf{C}_A(\nu)\nu = \begin{bmatrix} Y_{\dot{v}}vr + Y_{\dot{r}}r^2 \\ -X_{\dot{u}}ur \\ -Y_{\dot{v}}vu - Y_{\dot{r}}ru + X_{\dot{u}}uv \end{bmatrix} \tag{7}$$

$$C_{RB}(\nu)\nu = \begin{bmatrix} -mrv - mx_g r^2 \\ mru \\ mx_g ru \end{bmatrix} \tag{8}$$

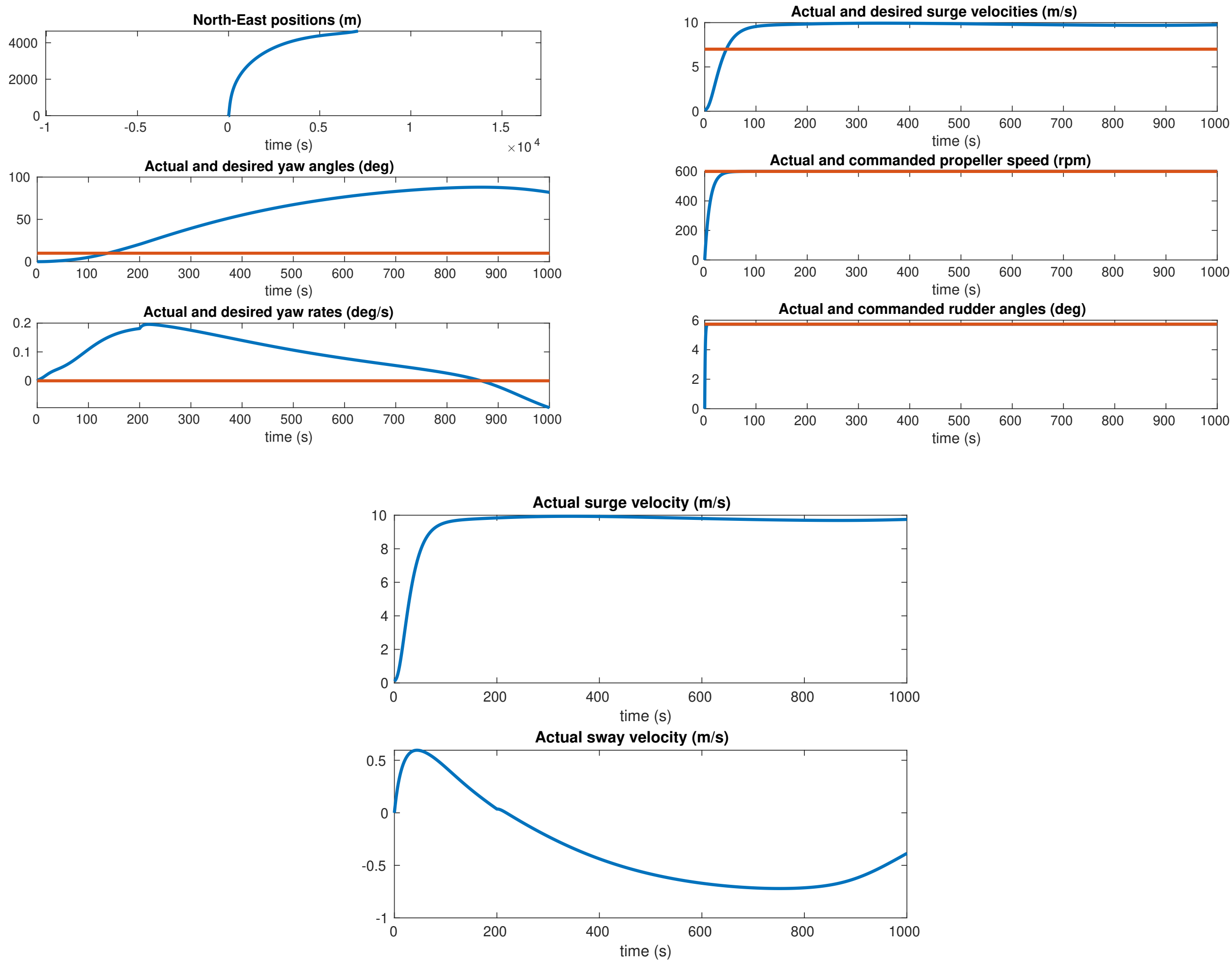

Figure 2: Boat behaviour with current disturbance and wind disturbance after 200 seconds

Linearized at $\nu = \nu^*$ we get

$$\mathbf{C}_A^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -X_{\dot{u}}U_d \\ 0 & -Y_{\dot{v}}U_d + X_{\dot{u}}U_d & -Y_{\dot{r}}U_d \end{bmatrix} \tag{9}$$

$$C_{RB}^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & mU_d \\ 0 & 0 & mx_gU_d \end{bmatrix} \tag{10}$$

1.2.2 Problem 2.b

The model is given in [1] as

$$M\dot{\nu_r} + N\nu_r = b\delta \tag{11}$$

and by rearranging we get that

$$\dot{\nu_r} = -M^{-1}N\nu_r + M^{-1}b\delta \tag{12}$$

The maneuvering model can be expressed as a state-space model as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

where

$$\mathbf{A} = -\mathbf{M}^{-1}N$$
$$\mathbf{B} = \mathbf{M}^{-1}b$$
$$\mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$
$$\mathbf{D} = 0$$

The resulting transfer function can be approximated by using ss2tf(.)

$$\frac{r}{\delta}(s) = \frac{8.04 \cdot 10^{-5}s + 6.15 \cdot 10^{-6}}{s^2 + 0.151s + 8.20 \cdot 10^{-4}} = \frac{K(T_3 s + 1)}{(T_1 s + 1)(T_2 + 1)} \tag{13}$$

With the numerical values yielding $K = 7.50 \cdot 10^-3$, $T_3 = 14.054$, $T_2 = 117.70$ and $T_1 = 6.90$.

1.2.3 Problem 2.c

Going from Equation (13) to the Nomoto model

$$\frac{r}{\delta}(s) = \frac{K}{Ts + 1} \tag{14}$$

where $T := T_1 + T_2 - T_3 = 169.55$ and $K = 7.50 \cdot 10^{-3}$

1.2.4 Problem 2.d

A PID controller for heading autopilot is created using pole placement. We can also see that the actual yaw angle and yaw rate is following the desired yaw angle and yaw rate. From the north east position we can see that the course angle is different from zero, however this is because we have created a heading controller, not a course controller. Because of the slow dynamics, the controller use some time to reach the desired angle. That is what causes the north east angle.

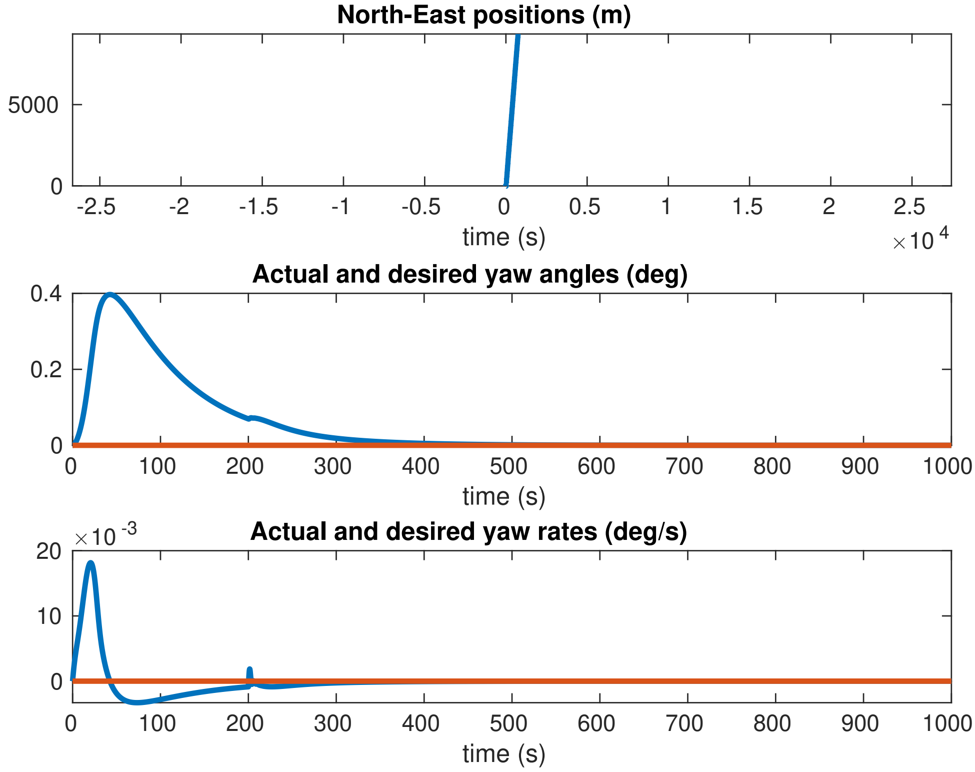

Figure 3: Boat behaviour with PID controller for heading autopilot.

1.2.5 Problem 2.e

The heading autopilot can compensate for the 10 deg to 20 deg maneuver, as can be seen in the North East position plot in fig. 4. Integral windup is not necessary in this case because we don't have saturation in the actuator. However it is good practice to implement.

```
M_lin = MRB + MA;
M_lin = M_lin(2:3,2:3);
M_lin_inv = inv(M_lin);

CRB_lin = [ 0     m*U_d
            0     m*xg*U_d   ];

CA_lin = [   0                       -Xudot*U_d
            -Yvdot*U_d+Xudot*U_d     -Yrdot*U_d   ];

N_lin = CRB_lin + CA_lin + D(2:3,2:3);
b_lin = [-2*U_d*Y_Δ -2*U_d*N_Δ]';

[num, den] = ss2tf(-M_lin_inv*N_lin, M_lin_inv*b_lin, [0 1], 0);

K_lin = num(3)/den(3);            % gain can be found by using the steady-state ...
    value (s=0) of the transfer function

poles_lin = roots(den);
zeros_lin = roots(num);

T3_lin = -1/zeros_lin;
T1_lin = -1/poles_lin(1);
T2_lin = -1/poles_lin(2);

```

```
T_lin = T1_lin+T2_lin-T3_lin;    % nomoto first-order time constant, eq. (7.24)

% controller gains (example 15.7)
Kp = wn^2*T_lin/K_lin;
Kd = (2*zeta*wn*T_lin-1)/K_lin;
Ki = wn/10*Kp;
```

Listing 3: Linearized C_a and C_{RB} and the gains for PID controller using pole placement

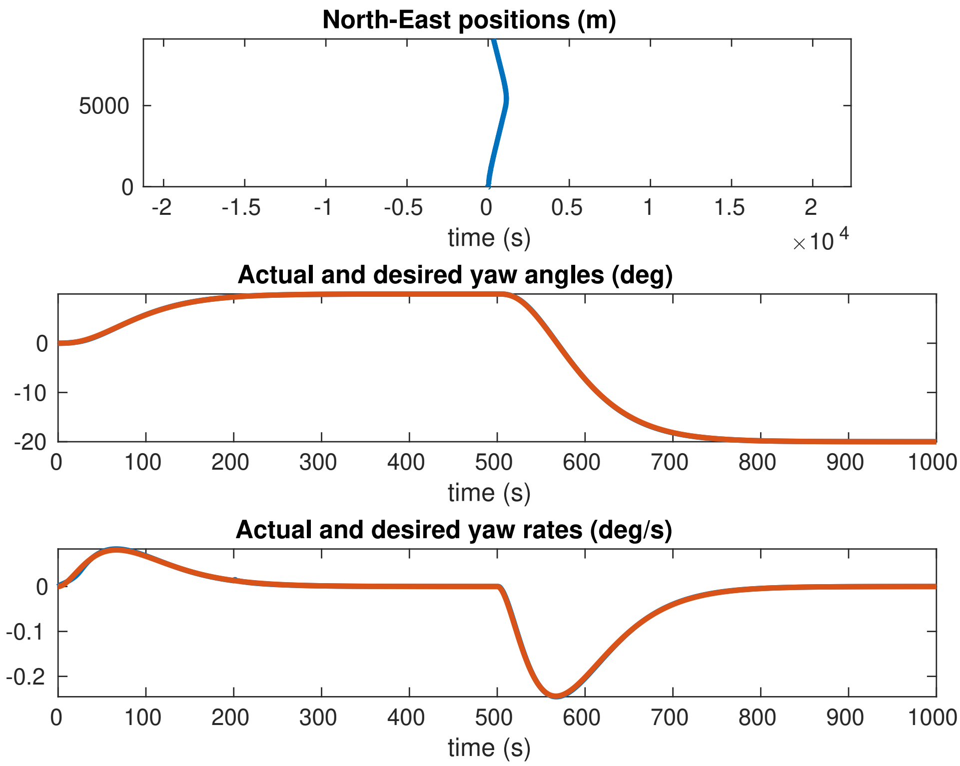

Figure 4: Boat behaviour when doing a 10-20 degree maneuver.

2 Part 3: Speed Control

2.1 Problem 1: Propeller Revolution and Speed Control

2.1.1 Problem 1a

After using the wageningen-function, the updated values are $K_T = 0.6367$ and $K_Q = 0.1390$

2.1.2 Problem 1b

The code is updated to include the dynamics of the prime mover system with the following equations.

$$\frac{Q_m}{Y}(s) = \frac{K}{Ts+1} \tag{15}$$

$$\dot{Q_m} = -\frac{Q_m}{T} + \frac{K}{t}(m_c - n) \tag{16}$$

The shaft speed dynamics $\dot{n}$ is also updated, and the code is shown below.

$$\dot{n} = \frac{Q_m - Q - Q_f}{I_m} \tag{17}$$

2.1.3 Problem 1c

```
% propeller dynamics
Im = 100000; Tm = 10; Km = 0.6;             % propulsion parameters

% added feedforward
Td = U_d*Xu/(t_thr-1); % desired thrust (N)

n_term = Td/(rho * Dia^4 * KT);
n_d = sign(n_term) * sqrt(abs(n_term));     % desired propeller speed (rps)

Qf = 0;                                     % friction torque (Nm)
Qd =  rho * Dia^4 * KQ * abs(n_d) * n_d;    % desired propeller moment (Nm)
Y = Qd/Km;                                  % control input to main motor

Qm_dot = -Qm/Tm + Km/Tm*Y;
n_dot = (Qm-Q-Qf)/Im;

% store simulation data in a table (for testing)
simdata(i,:) = [t n_d △_c n △ eta' nu' u_d psi_d r_d z];

% Euler integration
xd = euler2(xd_dot,xd,h);               % reference model
z = euler2(e_psi,z,h);                  % integral state
Qm = euler2(Qm_dot,Qm,h);
eta = euler2(eta_dot,eta,h);
nu  = euler2(nu_dot,nu,h);
△ = euler2(△_dot,△,h);
n  = euler2(n_dot,n,h);
```

Listing 4: Full code propeller dynamics

2.1.4 Problem 1d

Assuming $u_r = u$, $\dot{u} = \dot{u_r} = 0$, $u = U$, $x_{\delta\delta}\delta^2 = 0$, U is given by

$$U = \frac{(t-1)T}{x_u} \tag{18}$$

2.2 Problem 1e

With the equation from the last problem we get

$$T_d = \frac{U_d x_u}{t-1} \tag{19}$$

and

$$n_{term} = \frac{T_d}{\rho * d^4 * K_T} \tag{20}$$

$$n_d = sign(n_{term})\sqrt{abs(n_{term})} \tag{21}$$

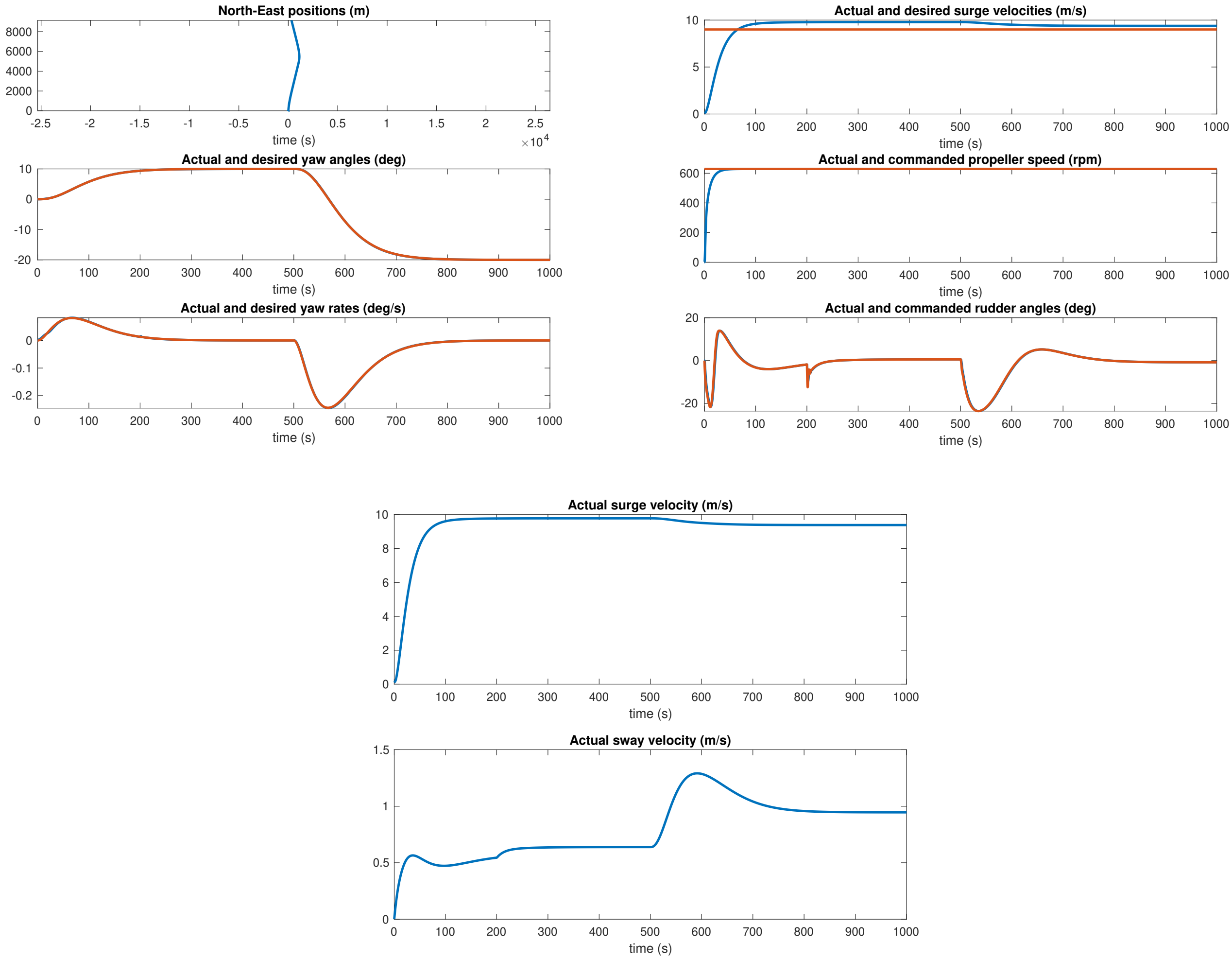

Figure 5: Ship behaviour with Propeller Revolution and Speed Control

No, we do not achieve the desired speed with constant heading angle. The surge velocity has a stationary error and does not reach the reference speed of $U_d = 9[m/s]$. This is because of the assumptions in the equation in problem 1d. To fix this we can add a feed-forward controller:

```
%Problem 1e: added feedforward
Td = (U_d-nu_c(1))*Xu/(t_thr-1);
```

Listing 5: Feed-forward

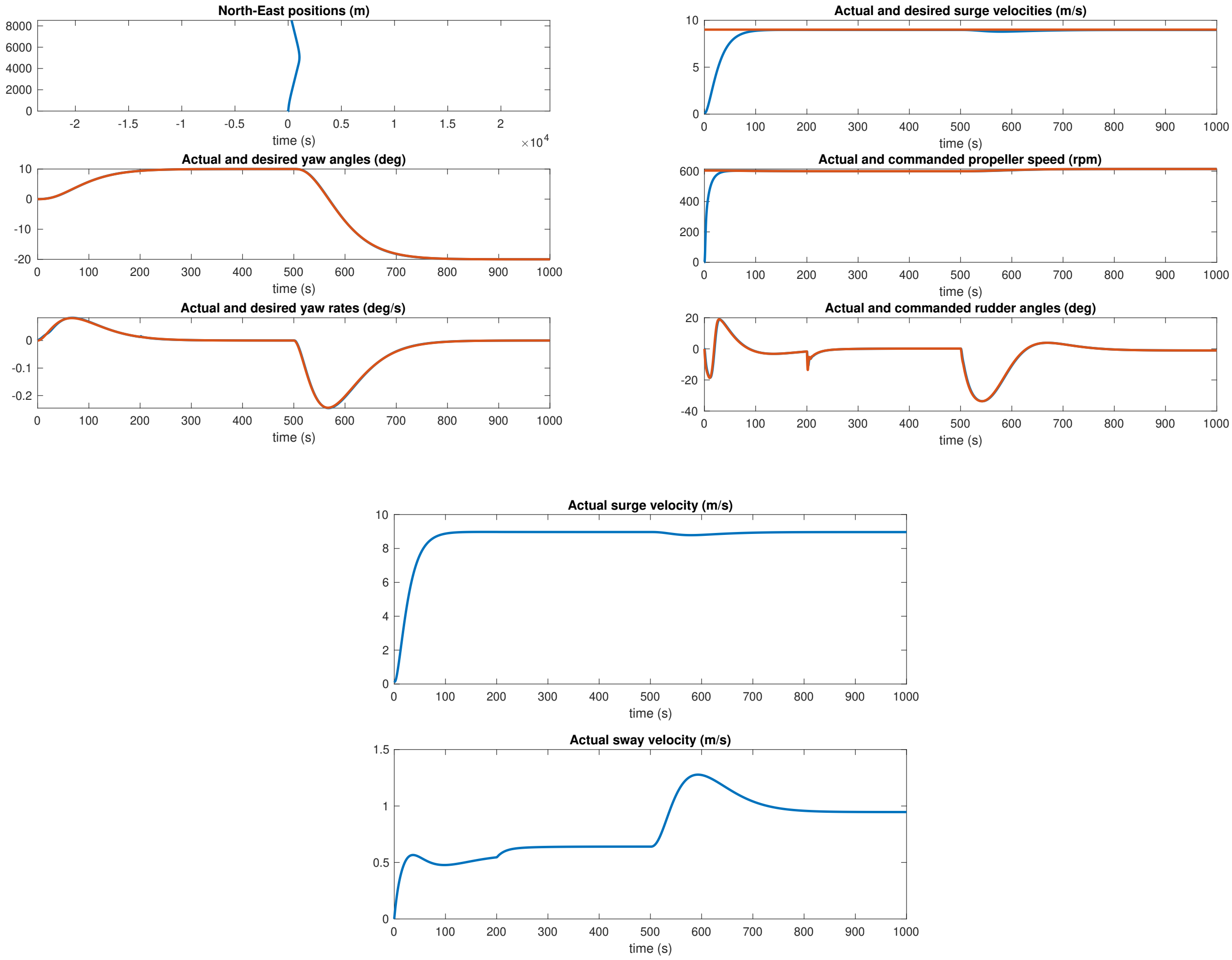

Figure 6: Ship behaviour with feed-forward

2.3 Problem 1f

Yes, with a $20[deg]$ setpoint change in heading, the speed will drop for a small period. This can be seen in figure 6. This is because of the sway velocity during the turn, so it need to be decomposed to be corrected to reference again.

References

[1] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2021.