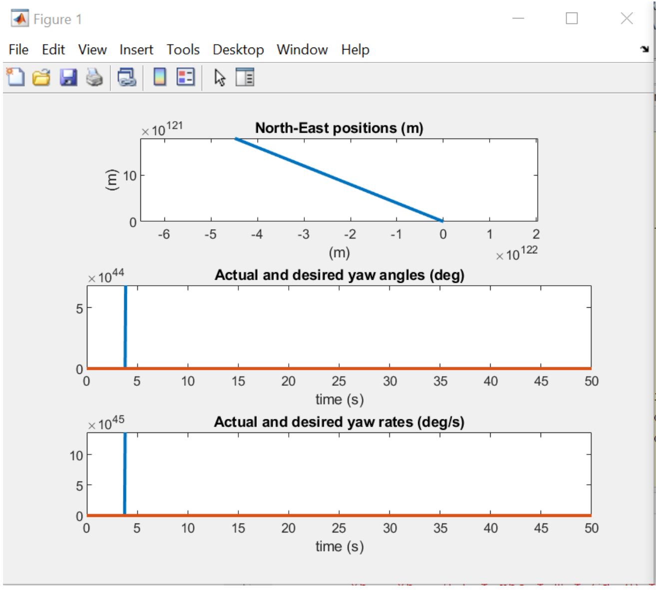

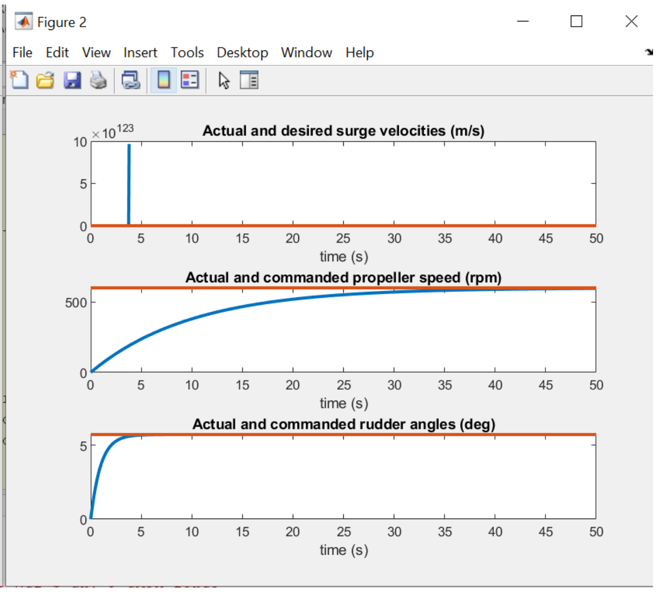


```
% Project in TTK4190 Guidance and Control of Vehicles
%
% Author:           Jacob Dahl
% Study program:    MTTK

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% USER INPUTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
addpath(genpath("C:\Users\jacob\Documents\Student\Fartøy\MSS"));
h  = 0.05;     % sampling time [s]
Ns = 1000;     % no. of samples

psi_ref = 10 * pi/180;  % desired yaw angle (rad)
u_ref   = 7;            % desired surge speed (m/s)

% ship parameters
m = 17.0677e6;          % mass (kg)
Iz = 2.1732e10;         % yaw moment of inertia (kg m^3)
xg = -3.7;              % CG x-ccordinate (m)
L = 161;                % length (m)
B = 21.8;               % beam (m)
T = 8.9;                % draft (m)
KT = 0.7;               % propeller coefficient (-)
```

```
Dia = 3.3;              % propeller diameter (m)
rho = 1025;             % density of water (m/s^3)

% rudder limitations
delta_max  = 40 * pi/180;        % max rudder angle      (rad)
Ddelta_max = 5  * pi/180;        % max rudder derivative (rad/s)

% added mass matrix
Xudot = -8.9830e5;
Yvdot = -5.1996e6;
Yrdot =  9.3677e5;
Nvdot =  Yrdot;
Nrdot = -2.4283e10;


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

% added mass matrices
MA = [-Xudot, 0, 0;
      0, -Yvdot, -Yrdot;
      0, -Nvdot, -Nrdot];

%Time constants (?)
T1 = 20;
T2 = 20;
T6 = 10;

% Damping matrix
Xu = (m - Xudot)/T1;
Yv = (m - Yvdot)/T2;
Nr = (Iz - Nrdot)/T6;

D = [Xu, 0, 0;
    0, Yv, 0;
    0, 0, Nr];


% input matrix
t_thr = 0.05;           % thrust deduction number
```

```matlab
X_delta2 = 0;            % rudder coefficients (Section 9.5)
Y_delta = 0;
N_delta = 1;
B_mat = [ (1-t_thr)  X_delta2
        0        Y_delta
        0        N_delta  ];

% initial states
eta = [0 0 0]';
nu  = [0.1 0 0.1]';
delta = 0;
n = 0;

%Idk
S = L*B + T*B + T*L; %is this a decent approx? Idk didnt find anything
uc = 0; %is this correct? No current?
vc = 0; % ditto
k = 0.1;
CR = 0;
ehtta = 0.001;
kin_visc = 10^(-6);
Lpp = L; %is this correct?
r = 0; %change this to correct value
```

```matlab

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% MAIN LOOP
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
simdata = zeros(Ns+1,14);                % table of simulation data

for i=1:Ns+1

    t = (i-1) * h;                       % time (s)

    % state-dependent time-varying matrices
    CRB = m * nu(3) * [ 0 -1 -xg
                        1  0  0
                        xg 0  0  ];
    R = Rzyx(0,0,eta(3));

    % added mass coriolis coeff
    a1 = Xudot * nu(1);
    a2 = Yvdot * nu(2) + Yrdot * nu(3);

    CA = [0, 0, a2;
```

```
      0,0, -a1;
      -a2, a1, 0];

    %Relative velocity
    ur = nu(1) - uc;
    vr = nu(2) - vc;
    %REynolds number
    Rn = (Lpp/kin_visc)*abs(ur);
    CF = (log10(Rn)-2)^2 + ehtta;

    %Nonlinear surge damping
    X_nonlin = -0.5*rho*S*(1+k)*(CF+CR)*abs(ur)*ur;

    %Integraaals
    Cd_2D = Hoerner(B,T);
    Yh = 0; %preallocate
    Nh = 0; %preallocate
    dx = Lpp/10; % 10 strips
    for xL = -Lpp/2:dx:Lpp/2
        Ucf = abs(vr + xL * r) * (vr + xL * r);
        Yh = Yh - 0.5 * rho * T * Cd_2D * Ucf * dx; % sway force
```

```
        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


    %Total damping
    D(1,1) = D(1,1)+X_nonlin;
    D(2,2) = D(2,2) + Yh;
    D(3,3) = D(3,3) + Nh;

    %added mass to matrices
    MRB = MRB + MA;
    Minv = inv(MRB);
    CRB = CRB + CA + D; %included damping

    % reference models
    psi_d = psi_ref;
    r_d = 0;
    u_d = u_ref;

    % thrust
    thr = rho * Dia^4 * KT * abs(n) * n;     % thrust command (N)
```

```matlab

    % control law
    delta_c = 0.1;              % rudder angle command (rad)
    n_c = 10;                   % propeller speed (rps)

    % ship dynamics
    u = [ thr delta ]';
    tau = B_mat * u;
    nu_dot = Minv * (tau - CRB * nu);
    eta_dot = R * nu;

    % Rudder saturation and dynamics (Sections 9.5.2)
    if abs(delta_c) >= delta_max
        delta_c = sign(delta_c)*delta_max;
    end

    delta_dot = delta_c - delta;
    if abs(delta_dot) >= Ddelta_max
        delta_dot = sign(delta_dot)*Ddelta_max;
    end

    % propeller dynamics
    n_dot = (1/10) * (n_c - n);

    % store simulation data in a table (for testing)
    simdata(i,:) = [t n_c delta_c n delta eta' nu' u_d psi_d r_d];

    % Euler integration
    eta = euler2(eta_dot,eta,h);
    nu  = euler2(nu_dot,nu,h);
    delta  = euler2(delta_dot,delta,h);
    n  = euler2(n_dot,n,h);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% PLOTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t       = simdata(:,1);                 % s
n_c     = 60 * simdata(:,2);            % rpm
delta_c = (180/pi) * simdata(:,3);      % deg
n       = 60 * simdata(:,4);            % rpm
delta   = (180/pi) * simdata(:,5);      % deg
x       = simdata(:,6);                 % m
y       = simdata(:,7);                 % m
psi     = (180/pi) * simdata(:,8);      % deg
u       = simdata(:,9);                 % m/s
v       = simdata(:,10);                % m/s
r       = (180/pi) * simdata(:,11);     % deg/s
u_d     = simdata(:,12);                % m/s
psi_d   = (180/pi) * simdata(:,13);     % deg
r_d     = (180/pi) * simdata(:,14);     % deg/s

figure(1)
```