TTK4190 Guidance and Control of Vehicles

Assignment 3

Written Fall 2018 By
Anders Haver Vagle,
Ada Skarsholt Larsen,
Sondre Aleksander Bergum,
Martin Madsen

Department of Engineering Cybernetics

Problem 1 - Autopilot Design

Problem 1.1

A model often used to describe the heading dynamics of a ship is the Nomoto model. The choice is between first order, second order and nonlinear Nomoto model. Because we want to avoid a nonlinear model unless absolutely necessary, we look at first and second order.

The second order model is known for having a better phase margin and is a typical choice when dealing with complex dynamics. However the second order Nomoto model is known for being ill-conditioned and the first order model is therefore preferred [1].

The first order Nomoto model can be found by defining the equivalent time constant:

$$T = T_1 + T_2 - T_3 \tag{1}$$

Such that,

$$\frac{r}{\delta}(s) = \frac{K}{(1 + Ts)} \tag{2}$$

Problem 1.2

The parameters can be identified by drawing inspiration from the "`ExKT.m`"-script in the MSS toolbox. We do a nonlinear least squares approximation of the step response of the plant. With a chosen step-input $\delta_r = 5$ degrees this gives the following:

$$T = 122.6001 \tag{3}$$
$$K = -0.0594 \tag{4}$$

Problem 1.3

To start designing a control law, we use a standard PID-controller on the form of equation 5, with controller gains tuned by specifying bandwith, ω_b, and damping ratio, ζ, in accordance with 6.

$$\delta_c = -K_p\tilde{\psi} - K_d\dot{\psi} - K_i\int_0^t \tilde{\psi}(\tau)d\tau \tag{5}$$

$$\begin{aligned} \omega_n &= \frac{1}{\sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}} \\ K_p &= \frac{\omega_n^2 T}{K} \\ K_d &= \frac{2\zeta\omega_n T - 1}{K} \\ K_i &= \frac{\omega_n^3 T}{10K} \end{aligned} \tag{6}$$

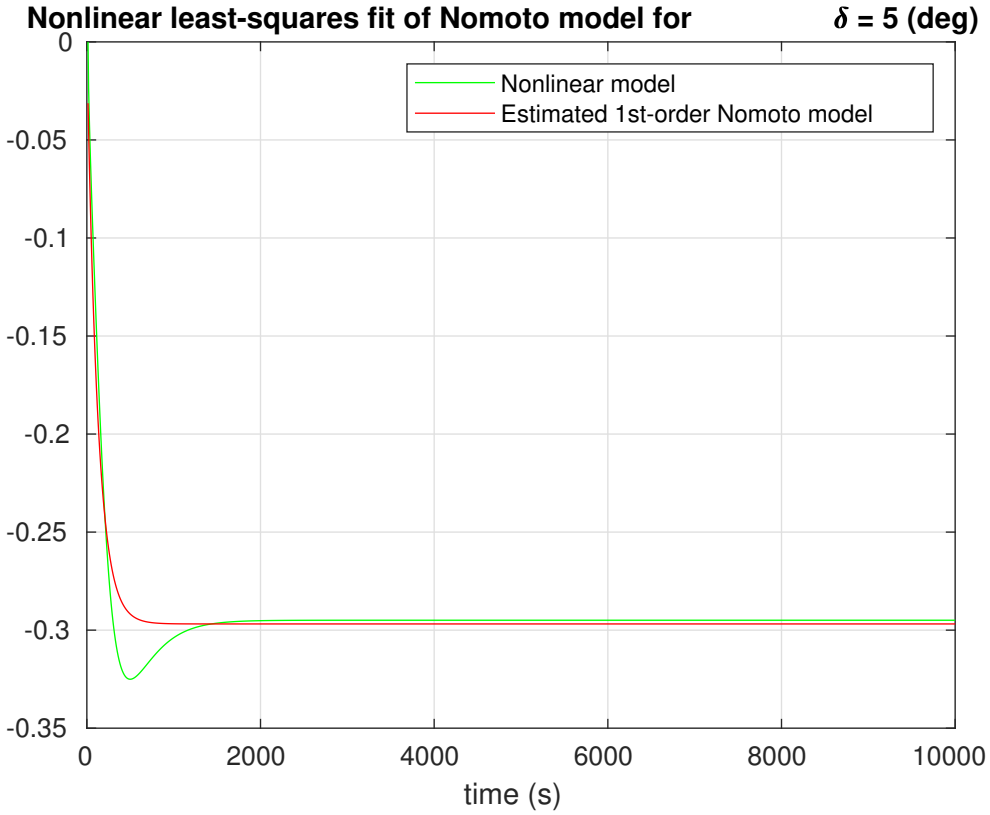

Figure 1: Fitting our estimated 1st-order Nomoto model to the nonlinear ship model.

With a slowly varying heading-reference, ψ_d, the current can be considered a piecewise constant disturbance and can be integrated away by the I-term in a standard controller. Thus, we should be able to follow the slow time-varying reference with a PI-controller. The D-term is added to ensure a faster response with less oscillations, as we will see in the next problem.

A limitation of the PID controller is that if the rudder is in saturation over a longer period of time, then the integral gain will cause integral windup. Without anti-integral windup, we can expect to see some less than optimal behaviour if we often saturate the rudder.

Problem 1.4

Figure 2 shows the ships response to a slowly-varying sinusoidal heading reference, ψ_d, in the presence of a current with PI control. Though the ship eventually follows the reference closely, the initial response has significant oscillations. The D-term is introduced with the expectation of a smoother response, which is seen to be achieved in figure 3. From the figure we note that the heading error stays close to zero after the initial turning of the ship, and the rudder input stays far away from its saturation limit and is a smooth signal that minimizes actuator strain. With this we conclude that the heading controller is acceptably tuned. This result was achieved by choosing the controller bandwidth of $\omega_b = 0.05$, and a damping ratio of $\zeta = 0.8$, giving the controller gains found in equation 7.

$$K_p = 6.8032 \quad K_d = 172.7604 \quad K_i = 0.0391 \tag{7}$$

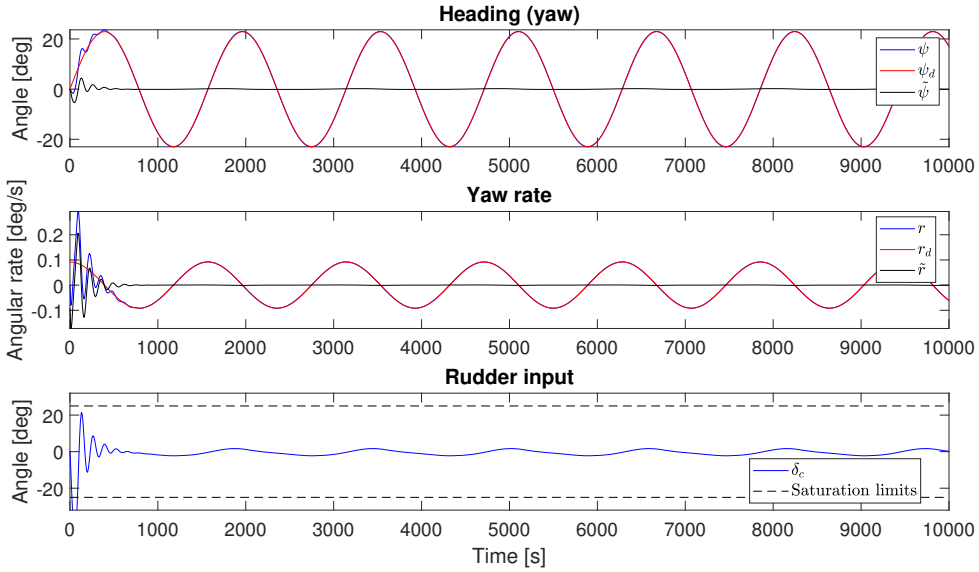

Figure 2: The ship response from the PI heading controller.

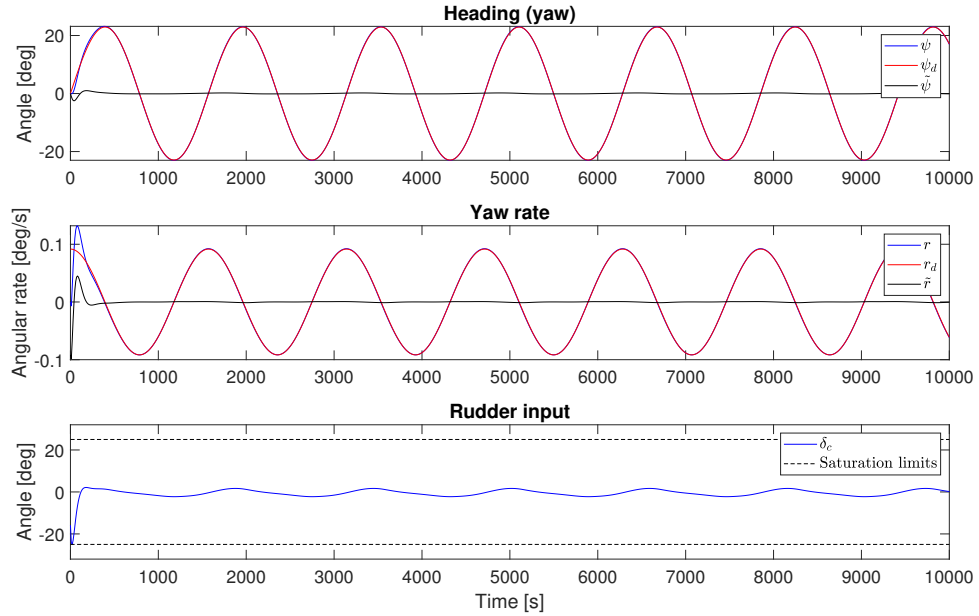

Figure 3: The ship response from the PID heading controller.

Problem 1.5

We choose the model from eq. 7.32 in [1] as shown in eq. 8

$$(m - X_{\dot{u}})\dot{u} - X_u u_r - X_{|u|u}|u_r|u_r = \tau \tag{8}$$

where

$$\tau = K_{thrust} n|n|$$

to simplify, we define

$$\begin{aligned} m_u &:= \frac{m - X_{\dot{u}}}{K_{thrust}} \\ d_1 &:= -\frac{X_u}{K_{thrust}} \\ d_2 &:= -\frac{X_{|u|u}}{K_{thrust}} \end{aligned}$$

to arrive at eq. the model in 9

$$m_u \dot{u} + d_1 u + d_2 |u|u = |n|n \tag{9}$$

Problem 1.6

To determine the model parameters, we run some simulations and analyze the results.

Looking at a steady-state setting, we know $\dot{u} = 0$ so in this case our surge model reduces to eq. 10

$$d_1 u + d_2 |u|u = |n|n \tag{10}$$

Looking at the steady-state surge speed u at two different shaft speeds n gives a set of 2 equations with 2 unknowns so we can determine d_1 and d_2, yielding

$$d_1 \approx 0, \quad d_2 = 1.0375 \tag{11}$$

To determine the parameter m_u we inspect the step response of the model compared to that of the ship. With trial and error, $m_u = 5500$ gave a good approximation as seen in fig. 4.

Problem 1.7

As we define

$$\tilde{u} := u - u_d$$

From section 13.2.3 in [1] we get the

$$\ddot{u} = \dot{u_d} - K_p \tilde{u} - K_i \int_0^t \tilde{u} d\tau \tag{12}$$

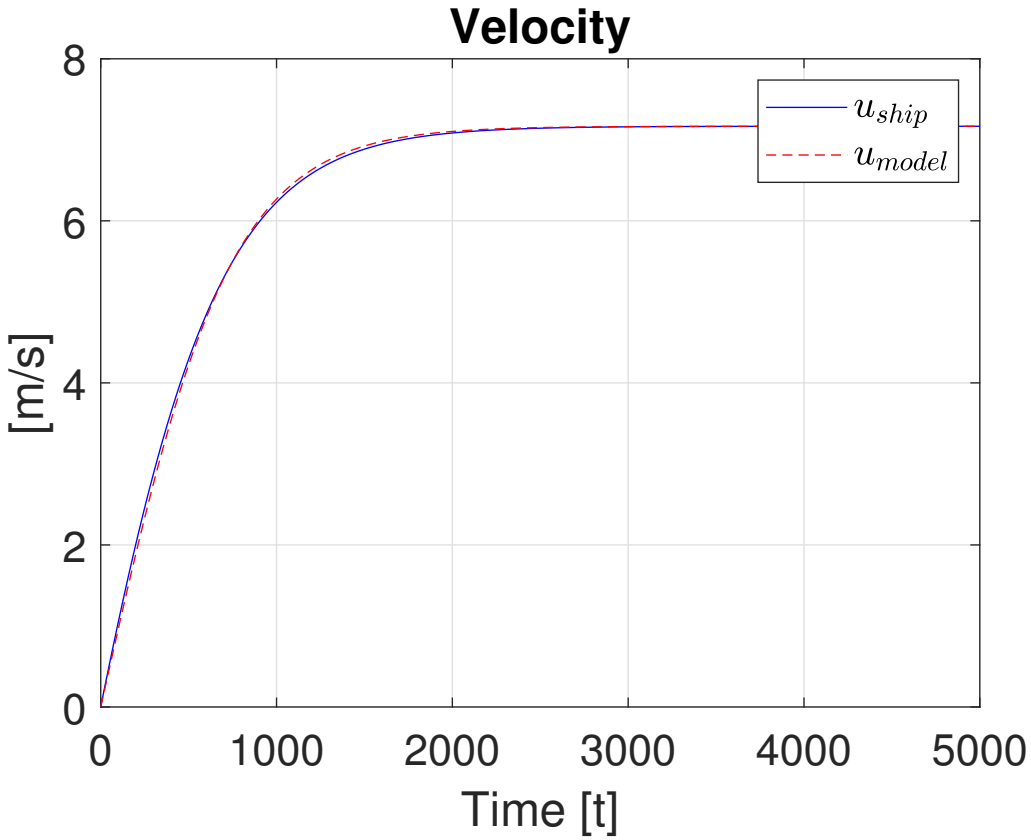

Figure 4: Step response of theoretical surge model compared to that of MSfaroystyring

By using this in our chosen surge-model, we get the speed controller in eq. 13

$$m_u[\dot{u_d} - K_p\tilde{u} - K_i\int_0^t \tilde{u}(\tau)d\tau] + d_1u + d_2|u|u = \tau \tag{13}$$

which is a feedback linearizing speed PI controller with feedforward acceleration. We then know that the linear system's equilibrium point given by

$$\dot{\tilde{u}} + K_p\tilde{u} + K_i\int_0^t \tilde{u}d\tau = 0 \tag{14}$$

is globally exponentially stable if the gains are chosen as

$$K_p = 2\lambda \tag{15}$$

$$K_i = \lambda^2 \tag{16}$$

where $(\lambda > 0)$, ensuring that $lim_{t\to\infty}\tilde{u} = 0$.

Problem 1.8

To implement the speed controller we can use the reference model from eq. 13.203 in [1] shown in eq.

$$\ddot{u}_d + 2\zeta\omega\dot{u}_d + \omega^2u_d = \omega^2r^b$$

with the transfer function

$$\frac{u_d}{u_r}(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \tag{17}$$

where $\zeta > 0$ and $\omega > 0$ are the reference models damping coefficient and natural frequency while r^b is the set-point specifying the desired surge speed.

With trial and error we arrived at the parameters for the reference model and controller as shown in eq. 18.

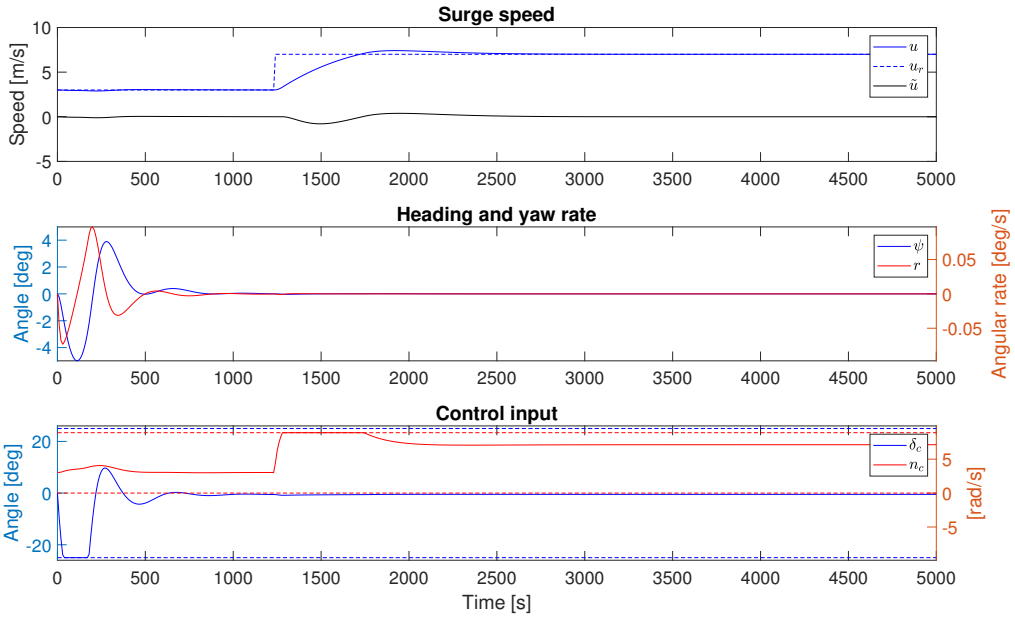

Figure 5

$$\zeta = 1, \quad \omega = 0.01 \quad \lambda = 0.001 \tag{18}$$

The resulting behavior is reasonable, however the control input saturates as we are doubling our speed.

Problem 2 - Straight-line path following in the horizontal plane

Problem 2.1

For the guidance system, we chose *enclosure-based steering*. This method is based upon enclosing the vessel's current position $\mathbf{p^n} = [x, y]^T$ by a circle with radius R and setting the desired course χ_d towards the intersection between the circle and the desired path. Using this point as the LOS-target we force the cross-track error from the desired path to zero. We chose

$$R = \min\{2.5L_{pp},\ ||\mathbf{p^n} - WP(k)||\} \tag{19}$$

This way the vessel will converge to the desired path when

$$2.5L_{pp} < ||\mathbf{p^n} - WP(k)|| \tag{20}$$

and use the next waypoint as the LOS-target otherwise. This is done to ensure that the vessel approaches the waypoint even when they are close to each other, though this is not a major issue here. The desired speed, U_d, was simply set to a constant $5m/s$. It is possible that a reference signal that would slow down before reaching the waypoints and speeding up after would improve the performance somewhat, but was not deemed necessary.

Problem 2.2

Using the guidance system outputs as reference for heading and speed autopilots, the simulated system behavior can be found in Figure 6 and the corresponding cross track error in Figure 7.

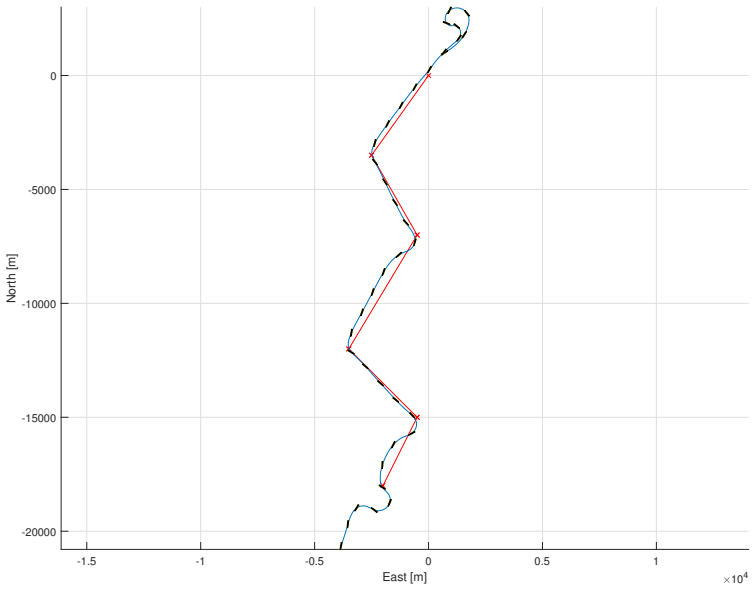

Figure 6: Path Simulation

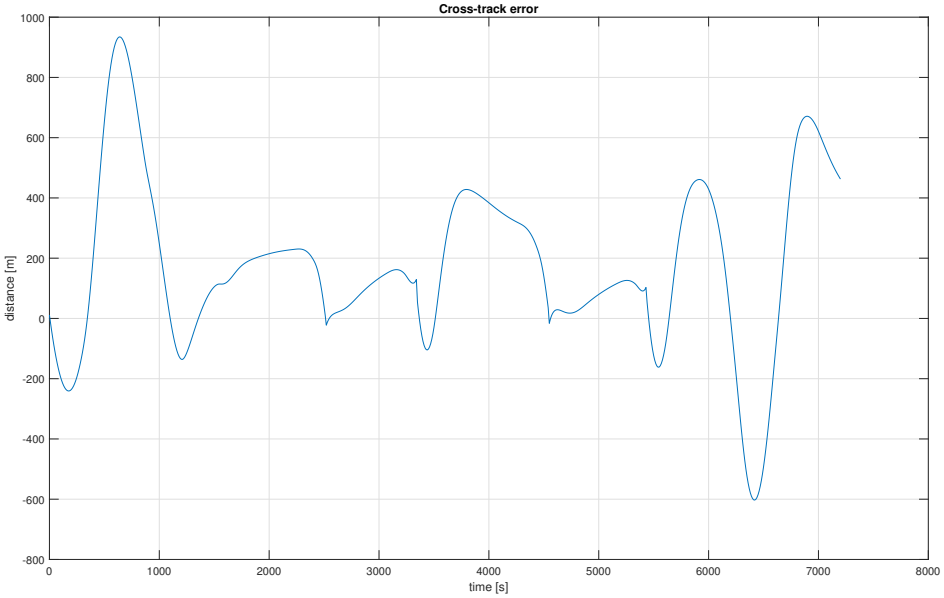

Figure 7: Cross Track Error

Problem 2.3

Figures 6 and 7 show the results of the waypoint guidance. The path-following behavior is approximately as expected. There is a deviation between the reference path and the actual path but mostly in the north-west direction. This might be explained by the current acting on the boat in this direction, inducing a crab angle. What is more unexpected is the first turn the vessel does to face the first waypoint. We see that the vessel turns too far around before finding the correct course. This could potentially be due to integrator wind-up, as the desired course angle is very different from the initial, meaning the rudder will be saturated for a significant amount of time.

Problem 2.4

In the previous section we assumed that $\chi = \psi$. The desired course χ_d, was therefore used as input for the heading controller. From figure 8, we see that this assumption does not hold. We know that $\chi = \psi + \beta$, and with β being non-zero there will be an offset between the course and the heading. While we see that the heading follows the desired course, the actual course is offset by the crab angle. As we want to steer the course and not the heading, a crab angle compensation is needed. Figure 8 also illustrates nicely how the integrator windup affects the vessel. This was noticed in hindsight, and could easily have been avoided with anti-windup. The effect is very obvious in the first turn, but also plays a role in the subsequent turns, causing the overshoots we observe in figure 6.

Problem 2.5

To include crab angle compensation we use

$$\chi = \psi + \beta \tag{21}$$

Figure 8: Course and heading with desired course and crab angle

and

$$\beta = \arcsin \frac{v}{U} \tag{22}$$

$$\tag{23}$$

such that

$$\beta = \arccos \frac{u}{U} \tag{24}$$

$$\rightarrow u_d = U_d \cos(\beta) \tag{25}$$

$$\psi_d = \chi_d - \beta \tag{26}$$

Problem 2.6

Comparing the results from 2.2 with the addition of crab angle compensation the resulting path following is a clear improvement. The effect of the current disturbance is almost not noticeable. The cross track error with crab compensation is generally smaller than without, but with more oscillations.

In Problem 2.4, the heading followed the desired course and the actual course was offset by the crab angle. With the crab angle compensation, the actual course now follows the desired course.

Problem 2.7

We want to track a target with constant speed $U_t = 3$ m/s along a straight line. Added to the current speed controller and heading controller is pure pursuit tracking. The target is simulated by using $\tilde{p}^n = p^n(t) - p_t^n(t)$. The chosen velocity assignment is therefore

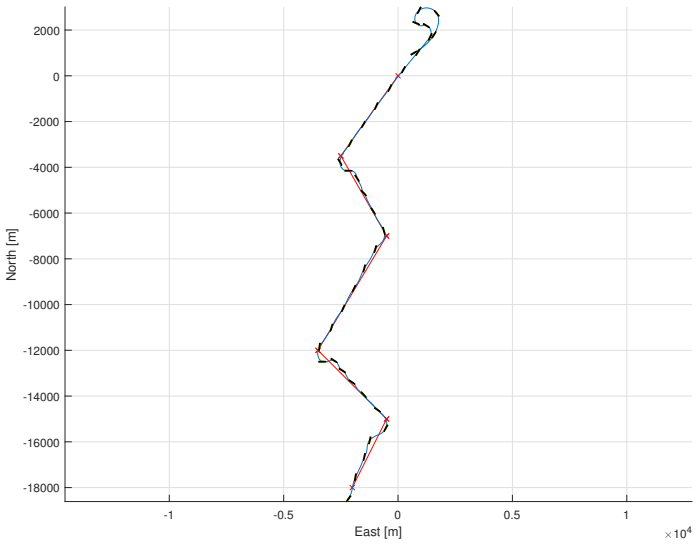

Figure 9: Path Simulation

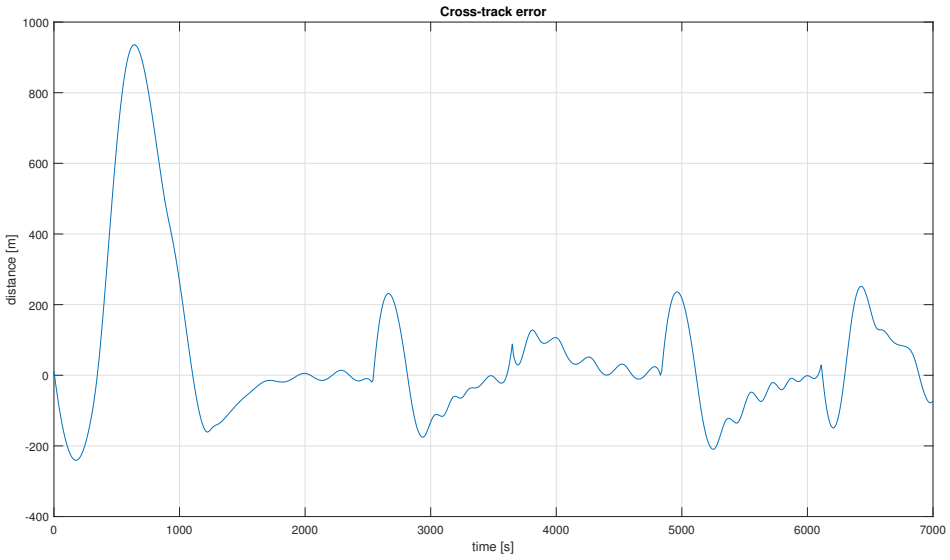

Figure 10: Cross Track Error

$$\kappa = (U_{max} - U_t)\frac{||\tilde{p}||}{\sqrt{\tilde{p}^T\tilde{p} + \Delta_{\tilde{p}}^2}} \tag{27}$$

$$v_d^n = -\kappa\frac{\tilde{p}^n}{||\tilde{p}^n||} \tag{28}$$

$$U_d = ||v_d^n|| \tag{29}$$

$$\chi_d = \arctan(v, u) \tag{30}$$

The parameter $U_{max} - U_t$ is the relative velocity and $\Delta_{\tilde{p}}$ is the transient interceptor-target rendezvous behavior. This means it decides how near the target the vessel will aim to be. We set

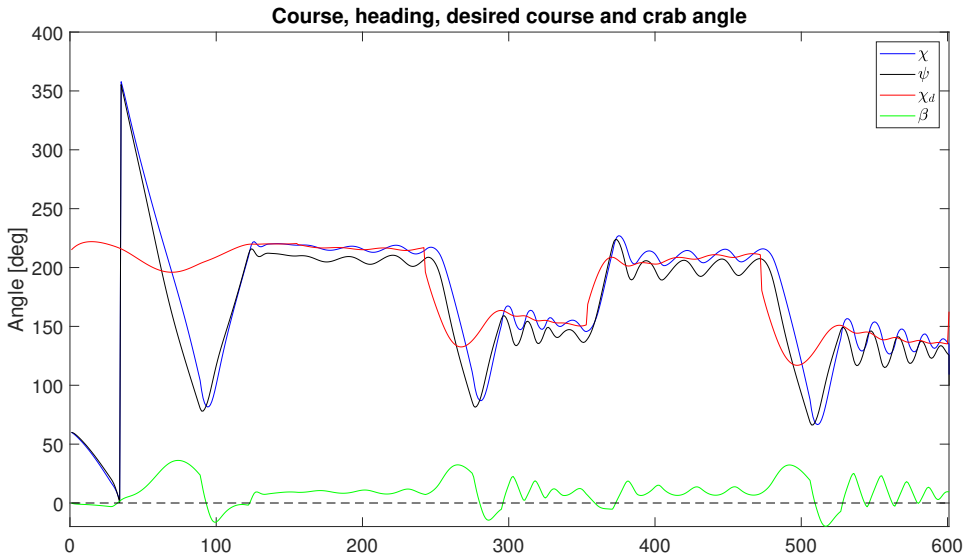

Figure 11: Course and heading with desired course and crab angle with crab angle compensation

$\Delta_{\tilde{p}} = 2000$, which from figure 13 we see results in a distance of $\approx 1200m$. Figure 12 shows the vessel approach the target with the pure pursuit tracking. From figure 14, we can see that the vessel starts following a slightly curved path in order to keep its distance to the target without getting too close as it approaches with a higher speed than the target.

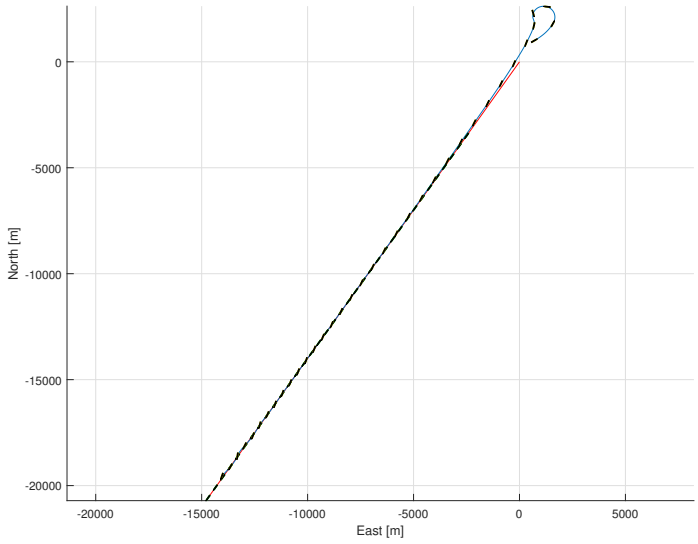

Figure 12: Path simulation with target tracking

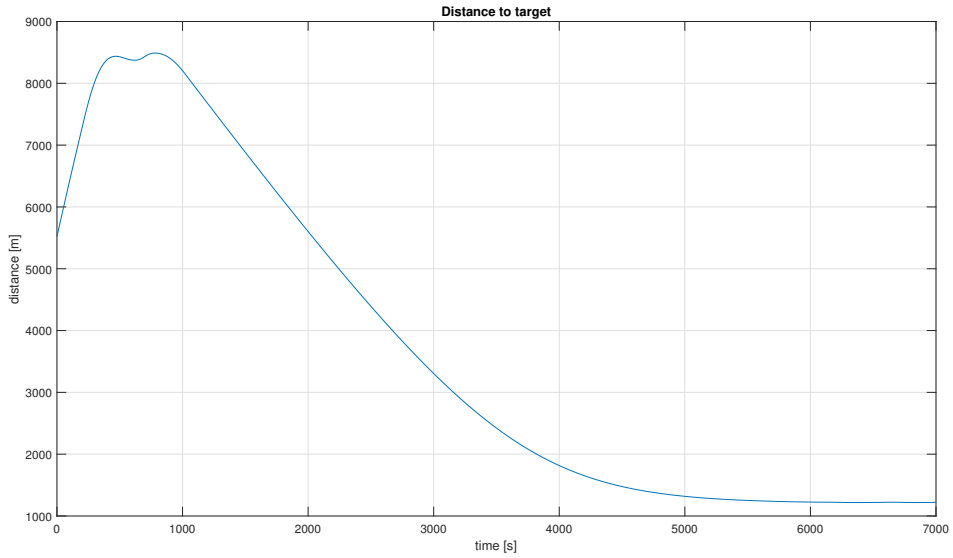

Figure 13: Distance to target

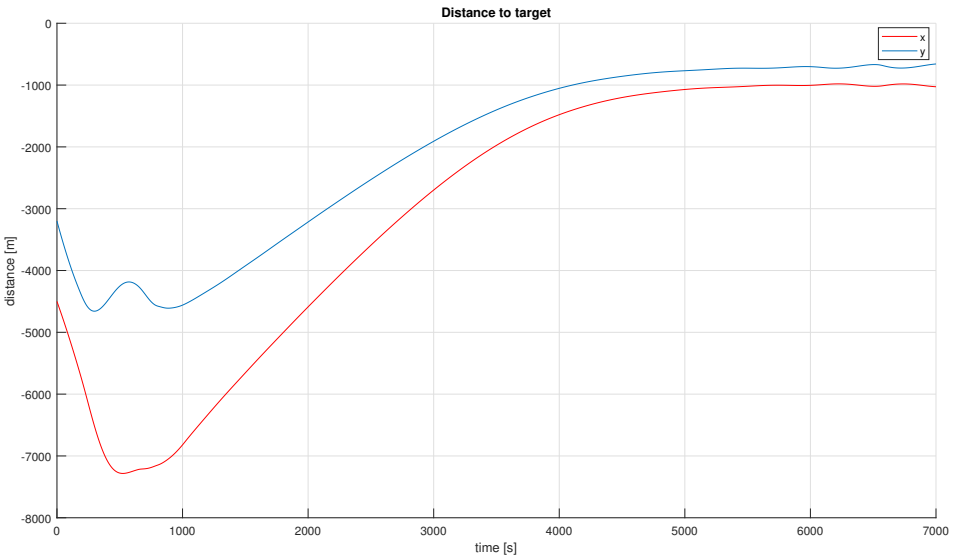

Figure 14: Distance to target in both directions

References

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