# Assignment 3, TTK4190

Shiv Jeet Rai
Arne Selle
Erik Liland

17. November 2015

## Contents

# 1 Autopilot design

## 1.1 Heading autopilot

### Analysis of ship characteristics

To be able to model the ship in a good way we ran many simulations with different rudder angle and measured the steady-state yaw rate. We then made a $\delta - r$ plot of the result. Since the ship was turning port while giving a positive rudder command, this plot and the rest of the assignment is made with a fixed gain of -1 on $\delta_c$.

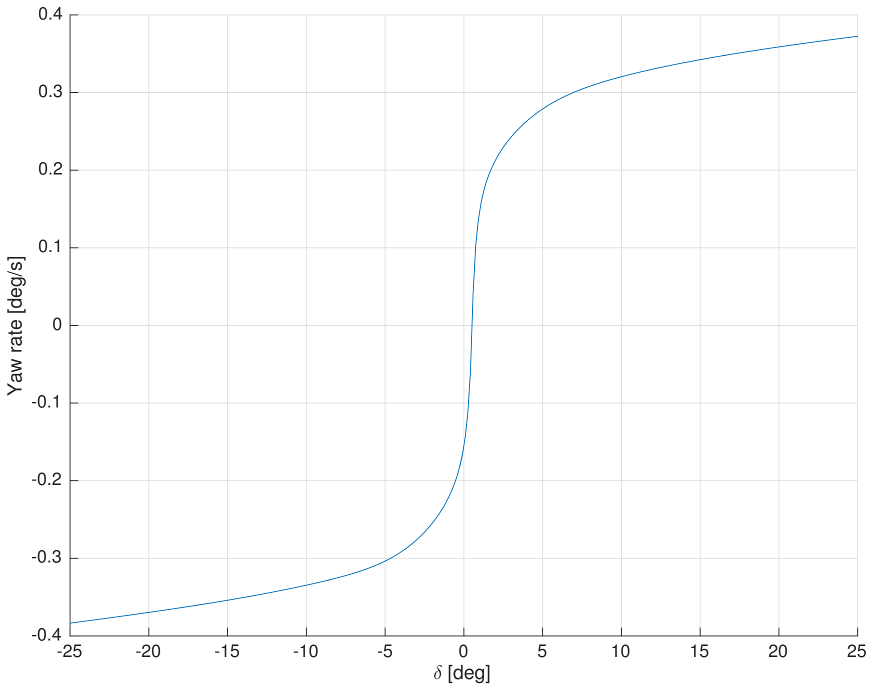


Figure 1: $\delta - r$ plot

From figure 1 we clearly see the non-linear characteristics of the ship. Since we only want to control heading and course, a 1-DOF heading model i.e. first- or second order Nomoto model with or without non-linear extensions can be used. To further investigate the effect of the non-linear characteristics of this ship, we compare the actual response with different models at different rudder angles. It should also be noticed that the ship has a constant drift to starboard when $\delta_c = 0$ as seen by the curve not passing through the origin. We compensate for this through the rest of the modeling part by adding a fixed rudder angle of $0.52^\circ$ to the rudder input. We only need this correction while estimating the model parameters. In a closed loop, the integral effect will cancel both this drift and drift caused by wind, current and waves.

### 2. order linear Nomoto model

$$\frac{r}{\delta}(s) = \frac{K(1+T_3 s)}{(1+T_1 s)(1+T_2 s)} \tag{1}$$

The second order Nomoto model follows the ships overshoot quite well.

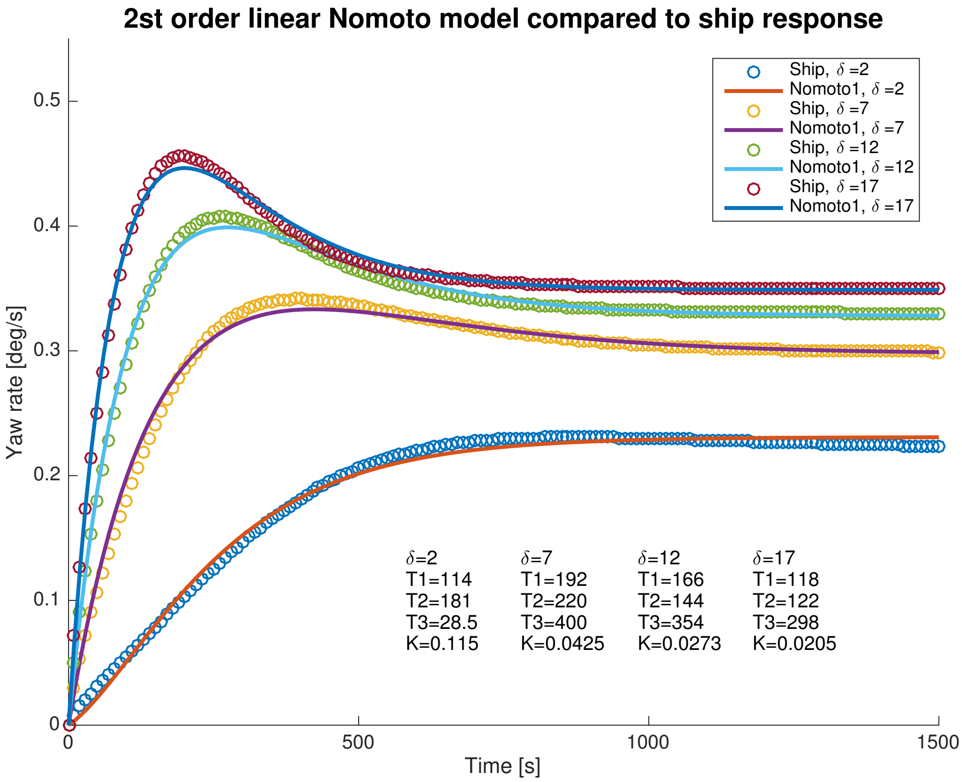


Figure 2: 2.order linear Nomoto model

**1. order linear Nomoto**

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

We also tried the first order version Nomoto, and as expected the model will only be accurate for small rudder angles, and is therefore not very good for modeling the non-linearities.

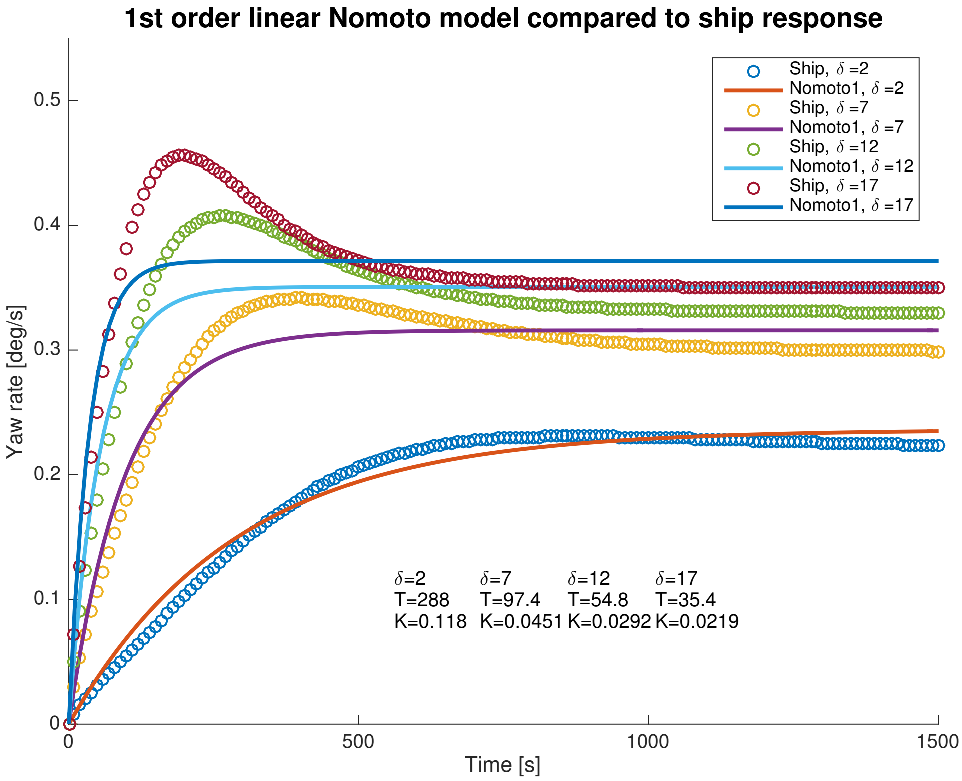


Figure 3: 1.order linear Nomoto model

**2. order non-linear Nomoto**

$$\begin{aligned} T_1T_2\ddot{r} + (T_1 + T_2)\dot{r} + KH_b(r) &= K(\delta + T_3\dot{\delta}) \\ H_B(r) &= b_3r^3 + b_2r^2 + b_1r + b_0 \end{aligned} \tag{3}$$

Where the steady state of $H_B(r) = \delta$. $b_0$ have already been taken care of in the fixed rudder offset, and by symmetry in the hull leads to $b_2 = 0$. We then only need the first- and third-order term to describe the maneuvering characteristics. By curve fitting $H_B(r) = b_3r^3 + b_1r = \delta$ to the obtained delta-r curve, we estimate the parameters $b_3$ and $b_1$.

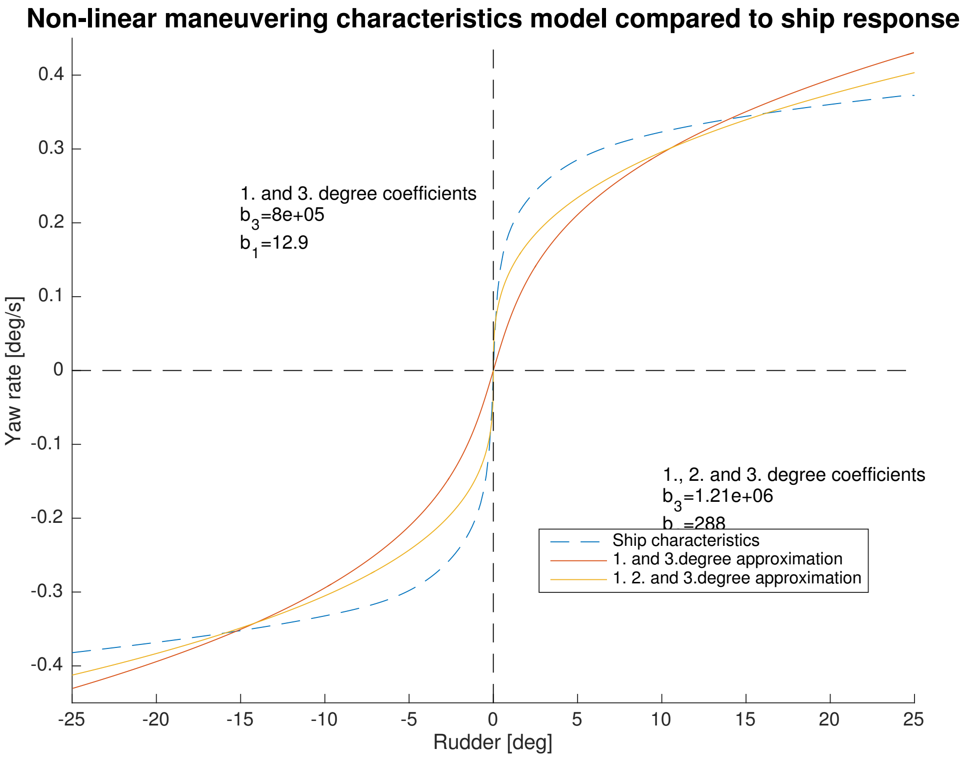


Figure 4: Third degree approximation of ship characteristics

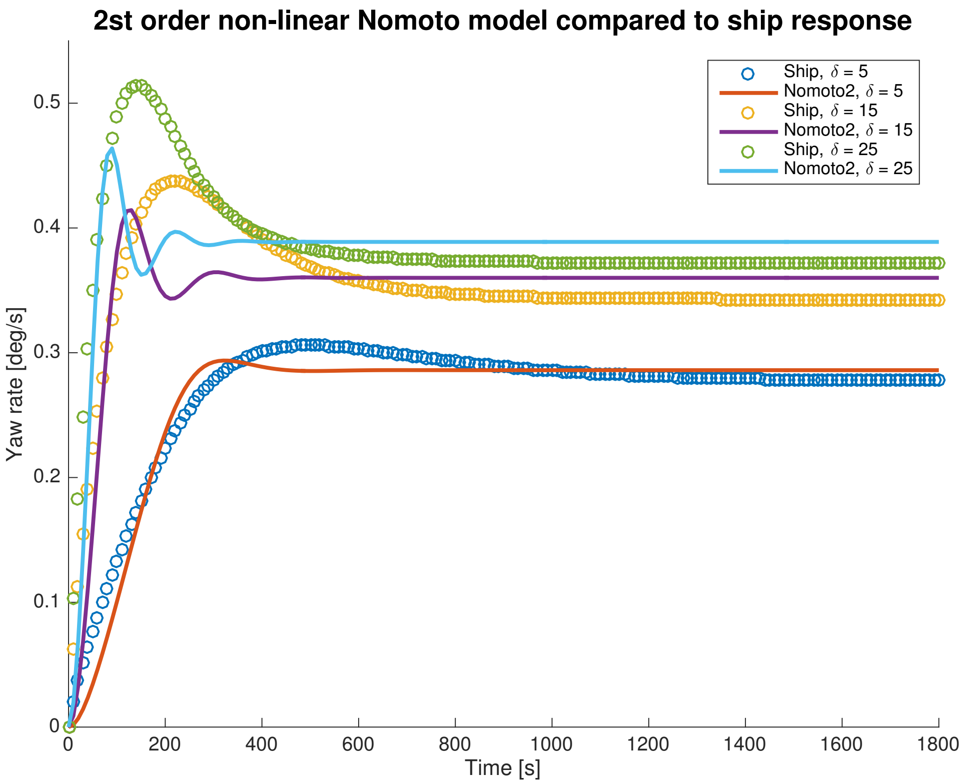


Figure 5: 2.order non-linear Nomoto model

### 1. order non-linear Nomoto

Norbins' extension of the linear first order model:

$$\begin{aligned} T\dot{r} + H_N(r) &= K\delta \\ H_N(r) &= n_3 r^3 + n_2 r^2 + n_1 r + n_0 \end{aligned} \tag{4}$$

Where the steady state of $H_N(r) = K\delta$. We know that $n_i = \frac{b_i}{|b_1|}$, and since our ship is stable we know that $n_1 = 1$, and $n_3 = sign(b_3) = 1$ thus resulting in following model.

$$T\dot{r} + r^3 + r = K\delta \tag{5}$$

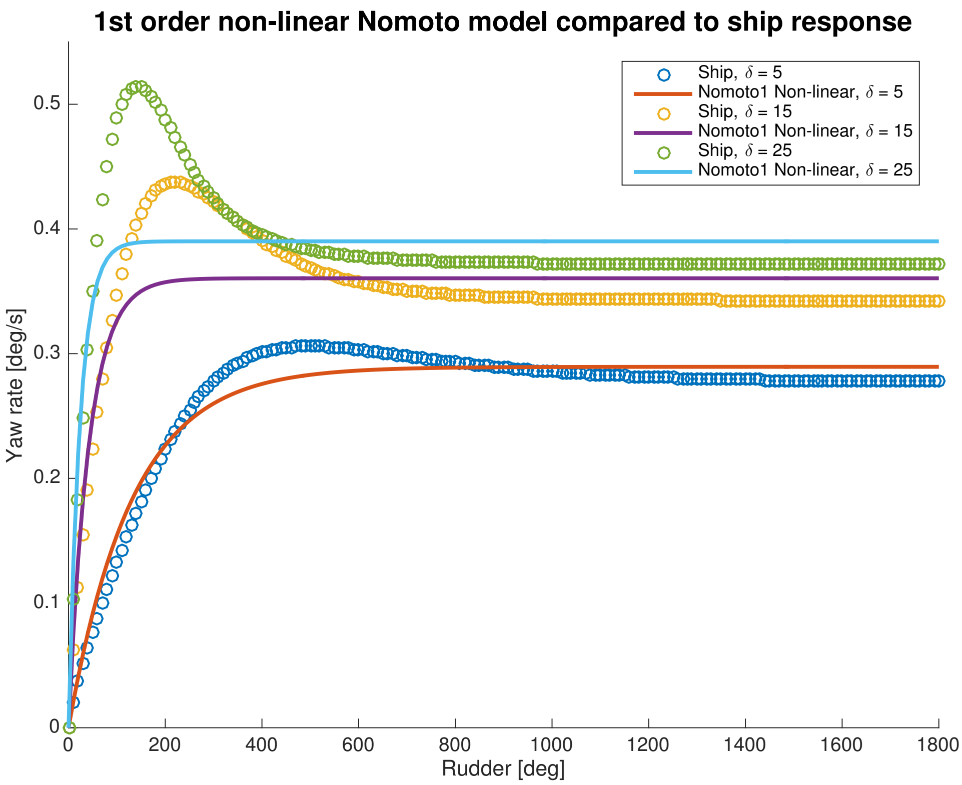


Figure 6: 1.order non-linear Nomoto model

## 1.2 Speed autopilot

To control the surge speed of MS Fartøystyring we suggest using a linearized model, where the surge speed is decoupled from the rest of the system. We are assuming

$$u >> v$$

which leads to

$$U = u$$

. We then use a quadratic speed model

$$K_1 \dot{u} - X_u u_r - X_{|u|u} |u_r| u_r = \tau \tag{6}$$

which leads to

$$\dot{u} = \frac{\tau + X_{|u|u} |u_r| u_r + X_u u_r}{m - X_{\dot{u}}} = \frac{X_{|u|u} |u_r| u_r + X_u u_r}{m - X_{\dot{u}}} + \tau_{nl} \tag{7}$$

where

$$\tau_{nl} = \frac{\tau}{m - X_{\dot{u}}} \Rightarrow \tau = \tau_{nl}(m - X_{\dot{u}}) \tag{8}$$

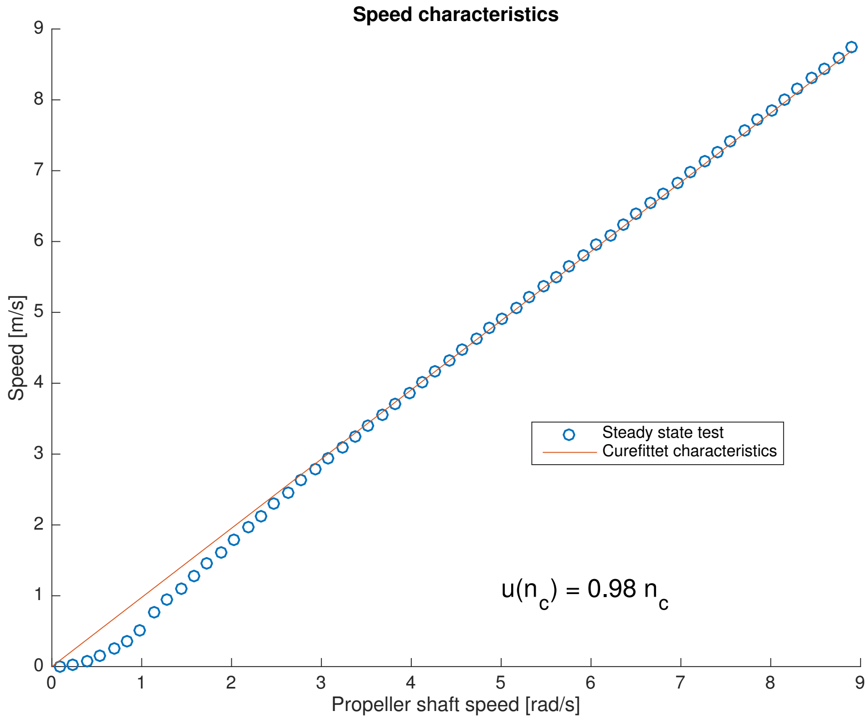


Figure 7: Speed characteristics

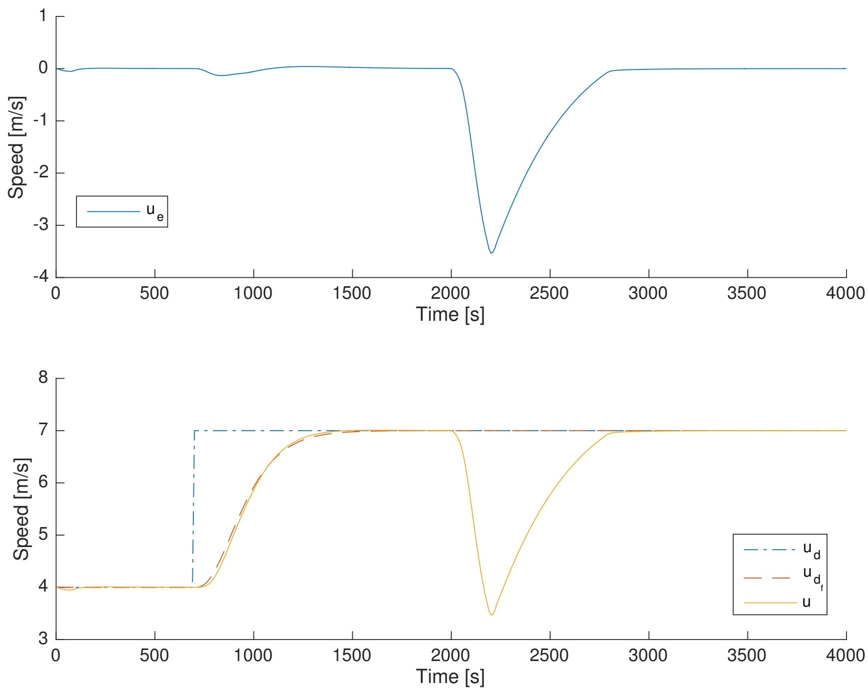


Figure 8: Speed step response

# 2 Path following and Path tracking

## 2.1 Path Generation

We can see from figure 9 that both methods based on continuous interpolation ("Piecewise Cubic Hermite Interpolating Polynomial" and "Cubic spline data interpolation") are very smooth, but they are more off-track than on-track. The pice-wise continuous interpolation are the crudest method of them all, since it in no way takes in to consideration the dynamics of the ship. The combination of circles and straight lines may look like the obvious best solution, but it does have a step in yaw rate ($r$), which means that the ship will not be able to follow the circle exactly while turning. Beside that, the lines and circles makes an excellent path for a ship to follow since it reduces the amount of time the ships actively uses its rudder, and therefore minimizes the drag on the ship. The turning radius set in the last method would be selected equal or larger to the ships turning radius. Preferably set it large enough to not loose to much speed, and small enough to not collide with someone/something. The nice about these turning radius, is that they can be adapted to the situation.

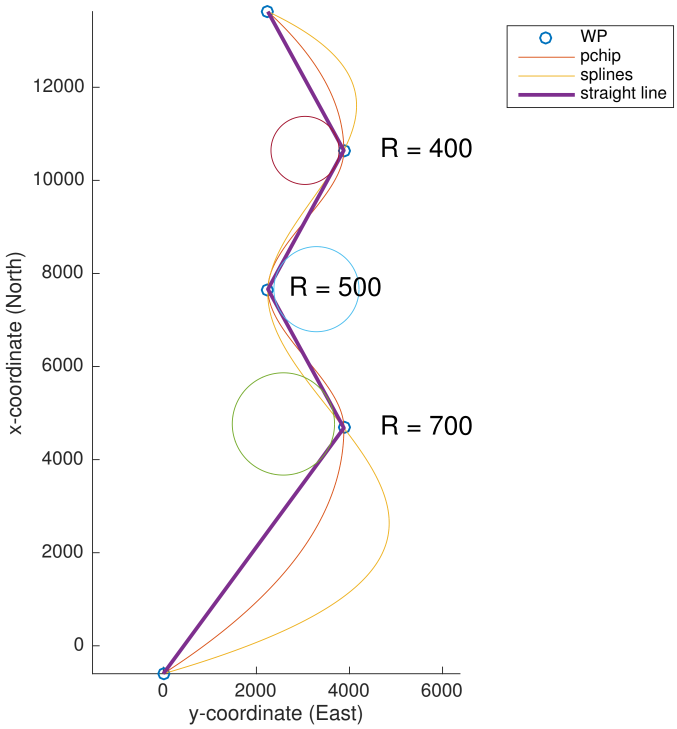


Figure 9: Different trajectories

## 2.2 Path following

We implemented a lookahead-based steering law, based on figure (10.10) in Fossen (2011). The desired course $\chi_d$ is made up of two parts, a path-angle $\chi_p$ and a cross-track error correction $\chi_r$. The cross-track error correction is essentially a PI-controller, normalized with an inverse tangent. The proportional gain is the inverse of the lookahead distance $\Delta_s$ which is a design parameter.

$$K_p = \frac{1}{\Delta_s} \tag{9}$$

The integral effect of the controller is quite complex, since we only want the integrator to compensate for small slow-changing disturbances like wind and current when the ship has come to a near steady state. To achieve this, we made an integral structure that takes into consideration the yaw rate of the ship $r$, the cross-track error $e$, a soft maximum integral effect and an anti-wind-up corrector and a regular gain on the end result. The total control law is listed in equation 10.

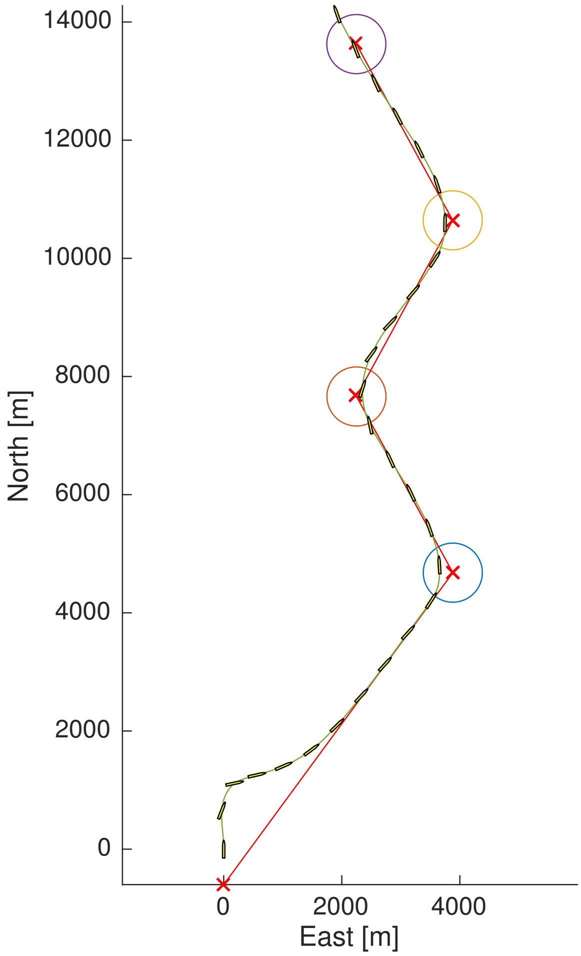

Figure 10: Path following

$$\begin{aligned}
\chi_d &= \chi_p + \chi_r \\
\chi_r &= atan(P - I) \\
P &= K_p e \\
I &= \int K_i e_i \\
e_i &= \begin{cases} 0, & \text{if } |r| > r_{lim} \text{ or } |e| > e_{lim} \\ \frac{e}{k_1 + k_2 \frac{|I|}{I_{max}}}, & \text{if } (I > I_{max} \text{ and } e > 0) \text{ or } (I < -I_{max} \text{ and } e < 0) \end{cases}
\end{aligned} \tag{10}$$

From the cross-track error (figure 11) it can be seen that when the ship first get aligned and on the track, the error stays within $\pm$200m.

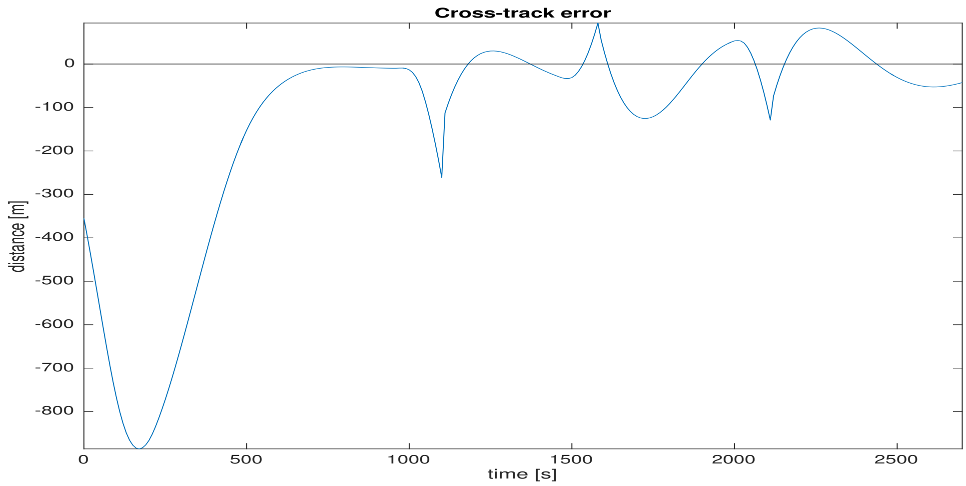


Figure 11: Cross-track error

## 2.3 Path Tracking

We want to follow a target with constant speed and course made up by the two first waypoints. This is essentially the same as path following with only one active segment, and a speed controller ensuring that we intercept the target and keep a constant distance from it. The heading and speed controller in use is the same as in section 2.2, and we have added a computation of $\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}_t$ which is a two dimensional vector from the target to our position. This vector is used by the speed guidance block to determine the distance to the target, and calculate the desired speed vector.

$$\begin{aligned} \mathbf{v}_d &= \mathbf{v}_t + \mathbf{v}_a \\ \mathbf{v}_a &= -\kappa \frac{\tilde{\mathbf{p}}}{||\tilde{\mathbf{p}}||} \\ \kappa &= U_{a,max} \frac{||\tilde{\mathbf{p}}||}{\sqrt{(\tilde{\mathbf{p}})^T \tilde{\mathbf{p}} + \Delta_s^2}} \end{aligned} \tag{11}$$

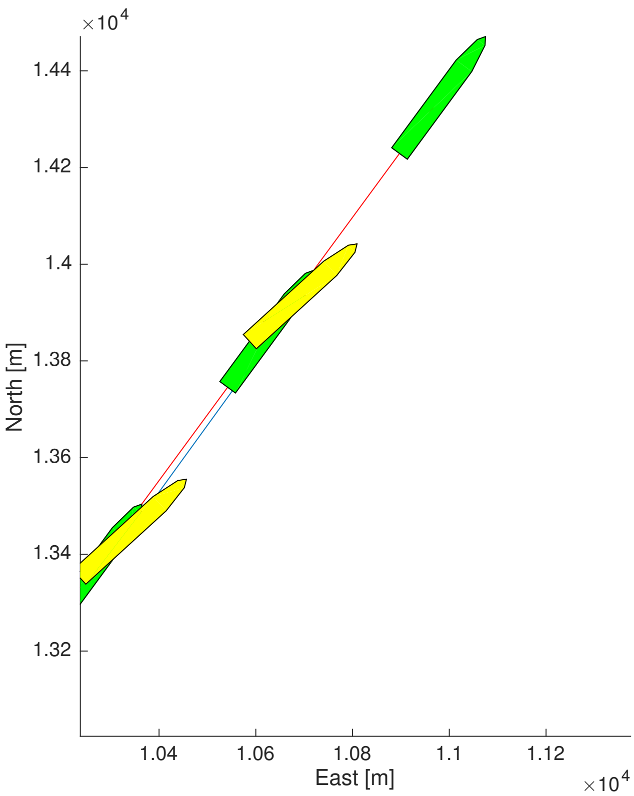


Figure 12: Target tracking (zoom)

Where $U_{a,max}$ and $\Delta_s^2$ are design parameters describing the maximum relative velocity between the intercepting ship and target and transient interceptor-target rendezvous behavior respectively.

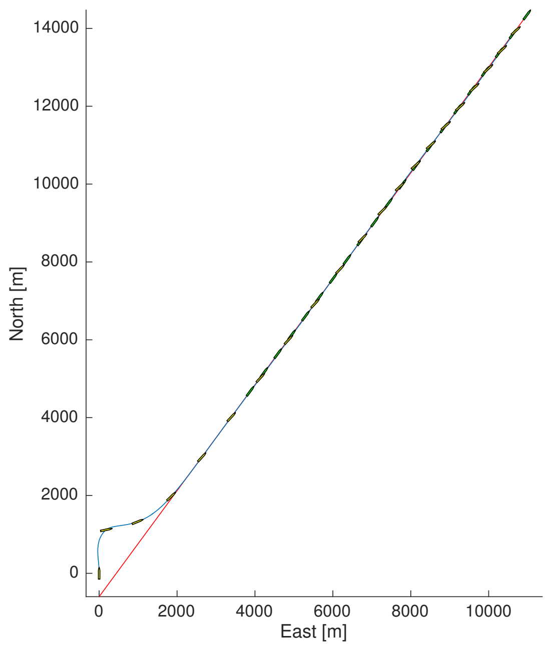


Figure 13: Target tracking

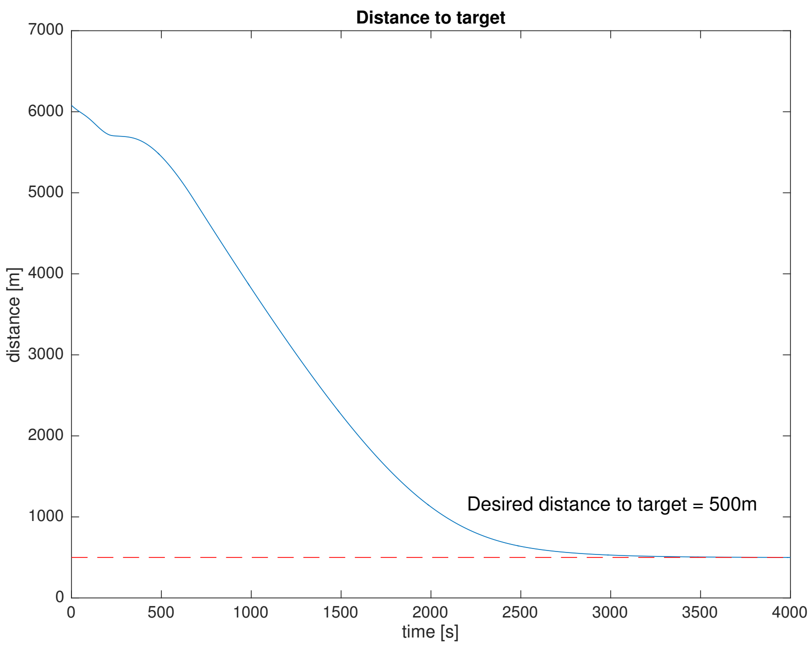


Figure 14: Interception of target