TTK4190 Guidance and Control of vehicles
Assignment 3

Name	Student number
Matthias Monnier	764085
Hubert Woszczyk	764091
Patrick Feiring	742442
Roy Angelsen	739457

November 17, 2015

Department of Engineering Cybernetics
Norwegian University of Science and Technology

Contents

1 Autopilot Design

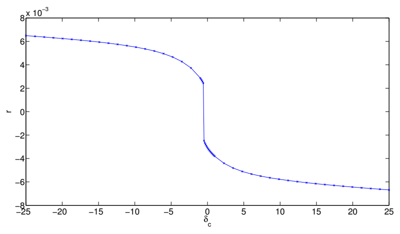
1.1 Heading Autopilot

1.1.1 Task 1.1

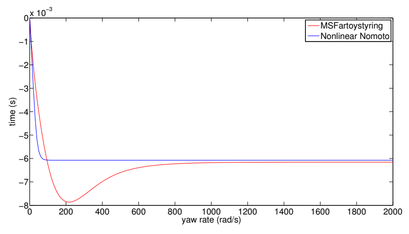
The heading is modelled as a nonlinear first order Nomoto model because

1. The step response is second order, but it is fairly close to a first order step response as shown in figure 1b. For simplicity and for control purposes a simpler model was chosen.

2. The relationship between the rudder angle and the corresponding steady state heading angle is clearly nonlinear, as illustrated by figure 1a.



(a) Nonlinear relationship between δ_c and r



(b) Step response of r and its first order approximation

Figure 1: Dynamics of the yaw rate r

Hence, the model can be written as

$$T\dot{r} + KH_B(r) = K\delta \tag{1}$$

where T, K are parameters to be found by model fitting and $H_B(r)$ is a polynomial which can be found by performing Bech's reverse spiral maneuver.

1.1.2 Task 1.2

By performing Bech's maneuver, $H_B(r)$ is found by curve fitting. The result is the following nonlinear function

$$H_B(r) = -1825700r^3 - 484.9072r^2 + 20.8534r - 0.0027 \tag{2}$$

K and T are determined through least square fitting of a step response (with rudder input 8° and thrust 80rpm) cf. figure 1b.

- $K = -0.0417$
- $T = 80.0360$

The resulting model is

$$80.036\dot{r} + 76131.69r^3 + 20.22r^2 - 0.8695r + 0.0001 = -0.0417\delta \tag{3}$$

where the constant offset is dropped due to its small value and integral action will also compensate for constant disturbances when the controller is introduced. The final model is

$$80.036\dot{r} + 76131.69r^3 + 20.22r^2 - 0.8695r = -0.0417\delta \tag{4}$$

1.1.3 Task 1.3

The controller is derived through feedback linearization of the model obtained earlier.

We define $\tilde{\psi} = \psi - \psi_d$ as the error on the heading and $\hat{d}$ as the disturbances generated by the current and the modelling uncertainties. By choosing $\delta_c = H_B(r) + K_p\tilde{\psi} + K_i\int_0^t \tilde{\psi}(\tau)d\tau + K_d\dot{\tilde{\psi}}$ the dynamics of r are

$$\dot{r} = \frac{K}{T}(K_p\tilde{\psi} + K_i\int_0^t \tilde{\psi}(\tau)d\tau + K_d\dot{\tilde{\psi}})$$

Defining $\tilde{r} = r - r_d, \dot{\tilde{r}} = \dot{r}$ and adding a new state $q = \int_0^t \tilde{\psi}$ the dynamics of the error on the heading are

$$\begin{pmatrix} \dot{\tilde{r}} \\ \dot{\tilde{\psi}} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{KK_d}{T} & \frac{KK_p}{T} & \frac{KK_i}{T} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \tilde{r} \\ \tilde{\psi} \\ q \end{pmatrix} + \begin{pmatrix} 1 & -\frac{KK_d}{T} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{d} \\ r_d \end{pmatrix} \tag{5}$$

The system will be stable if all the eigenvalues of the update matrix are negative.

$$\begin{aligned} \det(A - \lambda I) &= (\frac{KK_d}{T} - \lambda)\lambda^2 - 1(-\frac{KK_p}{T}\lambda - \frac{KK_i}{T}) = 0 \\ &= -\lambda^3 + \frac{KK_d}{T}\lambda^2 + \frac{KK_p}{T}\lambda + \frac{KKi}{T} = 0 \end{aligned}$$

By choosing

$$K_p = 2000 \quad K_i = 0.6 \quad K_d = 100$$

we have the following characteristic polynomial ($K = -0.0417$, $T = 80.360$)

$$-\lambda^3 - 0.052\lambda^2 - 1.04\lambda - 0.0003126$$

which has poles in $-0.0003, -0.0258497 - 1.01947i, -0.0258497 + 1.01947i$, which proves that the error will converge to 0 and thus the system is stable.

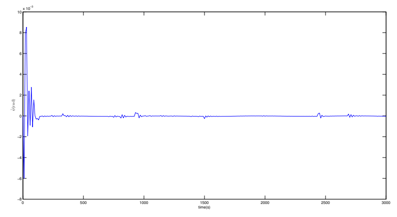
Since the controller includes integral action it will also converge to the desired state in presence of current.

1.1.4 Task 1.4

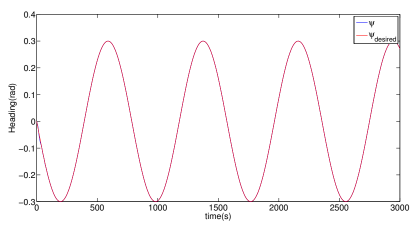
Figures 2a, 2b, 2c and 2d show how the controller behaves when following the reference signal $\psi_d = -0.3\sin(0.008t)$. The control input was not changed so it is still

$$\delta_c = H_B(r) + 2000\tilde{\psi} + 0.6\int_0^t \tilde{\psi}(\tau)d\tau + 100\dot{\tilde{\psi}} \tag{6}$$

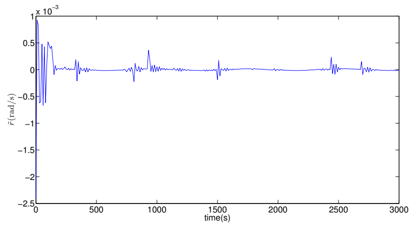
The ship's heading follows the reference perfectly.



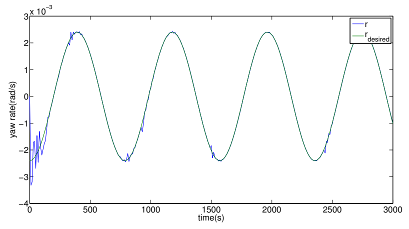
(a) Plot of $\tilde{\psi}$



(b) Plot of ψ and ψ_d



(c) Plot of $\tilde{r}$ for course following



(d) Plot of r and r_d for course following

Figure 2: Heading following

1.2 Speed Autopilot

1.2.1 Task 1.5

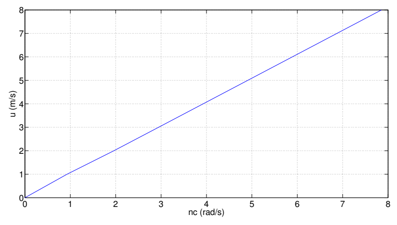
The surge is modelled as a linear first order system because

1. The relationship between thrust and steady state surge speed is linear, as seen on figure 3a.
2. The step response is first order, as visible on figure 3b.

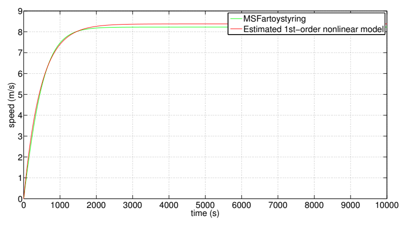
Consequently, the model can be written as

$$T\dot{u} + u = K\tau \tag{7}$$

with parameters K and T, the gain and the time constant of the system.



(a) Linear relationship between n_c and u, for a constant heading



(b) Step response of u and its model to input $n_c = 80$rpm

Figure 3: Dynamics of the surge speed u

1.2.2 Task 1.6

The parameters K and T are found by least square fitting on the step response to $n_c = 80$rpm cf. figure 3b

- $K = 0.96$
- $T = 458.76$

1.2.3 Task 1.7

The surge speed will be controlled with a PI controller. Performing a root locus the gains were initially set to

$$K_p = -10 \quad K_i = -0.05$$

but K_p were tuned to have a better response, which resulted in $K_p = -300$.

Stability can be proven by determining the eigenvalues of the update matrix of the system. If we define $\hat{d}$ the disturbances generated by the current and the uncertainties in the the modelling of the system, we have

$$T\dot{u} + u = K\delta_c + \hat{d}$$

Defining $\tilde{u} = u - u_d$ as the error on the surge speed we can write the error dynamics as

$$\dot{\tilde{u}} = \frac{KK_p - 1}{T}\tilde{u} + \frac{KK_i}{T}\int_0^t \tilde{u}(\tau)d\tau + \frac{\hat{d} - u_d}{T}$$

If we augment the system with the state $q = \int_0^t \tilde{u}(\tau)d\tau$ we have

$$\begin{pmatrix} \dot{\tilde{u}} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{KK_p-1}{T} & \frac{KK_i}{T} \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tilde{u} \\ q \end{pmatrix} + \begin{pmatrix} -\frac{1}{T} & \frac{1}{T} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_d \\ \hat{d} \end{pmatrix}$$

Now let's find the eigenvalues

$$\begin{aligned}\det(A-\lambda I) &= -\lambda\left(\frac{KK_p-1}{T}-\lambda\right)-\frac{KK_i}{T}=0\\ &= \lambda^2-\frac{KK_p-1}{T}\lambda-\frac{KK_i}{T}\end{aligned}$$

replacing with $K = 0.96$, $T = 458.76$, $K_p = -300$ and $K_i = -0.05$ we have

$$\begin{aligned}\lambda^2+0.6256\lambda+0.0001=0\\ (\lambda+0.00016)(\lambda+0.625)=0\end{aligned}$$

The eigenvalues are both negative so we can conclude that the error will be globally asymptotically stable at the origin. Moreover, integral action will make the controller work in presence of constant disturbance (current and modelling errors).

1.2.4 Task 1.8

Figures 4 and 5 show the behaviour of the controller when following a reference signal that is first equal to 4m/s until 700s have elapsed when it will be equal to 7m/s until the end of simulation. The simulation is conducted in presence of current. The figures show surge speed follows the desired speed.

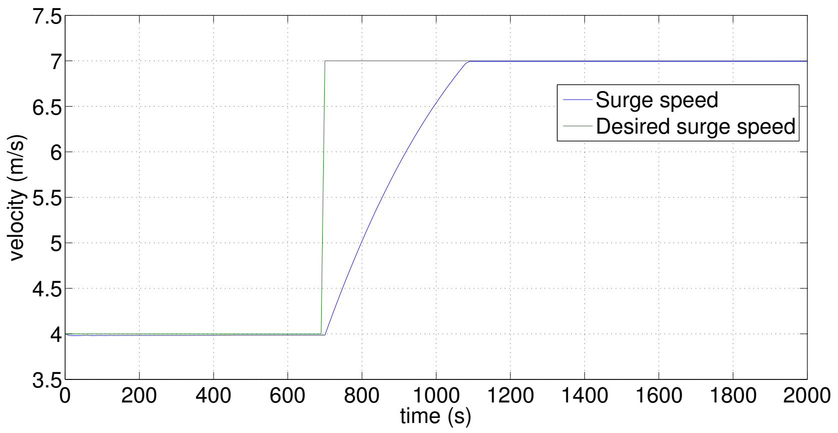


Figure 4: Plot of u and u_d

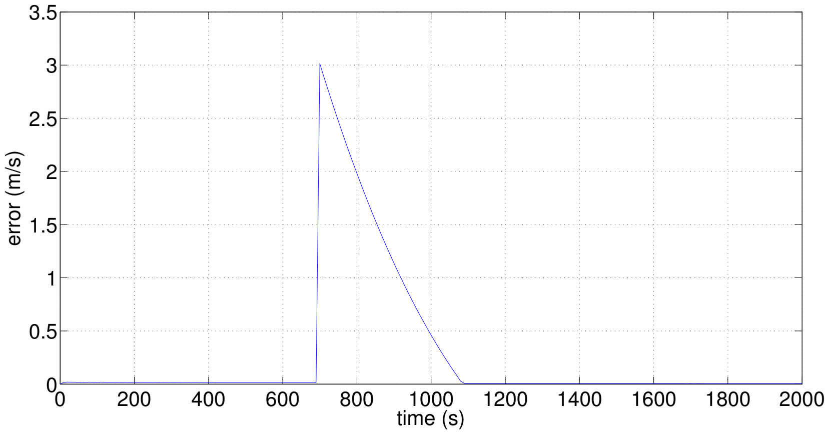


Figure 5: Plot of $\tilde{u}$

2 Path Following and Path Tracking

2.1 Path Generation

2.1.1 Task 2.1

Figure 6 shows paths generated by various methods. The first path is found using continuous interpolation of the waypoints, more specifically cubic Hermitian interpolation. The polynomial coefficients describing the path is generated using the *pchip* function found in the marine systems simulator(mss) toolbox. The second path, piecewise continuous interpolation is found by just drawing straight lines between two waypoints. For the third method, circles and straight lines, the lines are the same as in the case of piecewise continuous interpolation. The circles are chosen such that the ship is able to hold the path with a desired speed, i.e. the turning radius of the ship at the operating speed must not exceed that of the circle. In that case the ship would not be able to hold the path, and thereby drift away. Another element to take into consideration is passenger and crew comfort. Rapid turns, even though the boat is capable, is not always a good idea, suggesting that the radii are set with some margin with respect to the capabilities of the ship.

For a system deployed in a real setting the continuous interpolation would be the preferred option. This is due to the fact that it has continuous derivatives, making for smoother transitions between the different segments of the path. One could also argue that due to its simplicity and the option for land stationed operators to specify desired circle radii the circles and straight lines approach could be favorable.

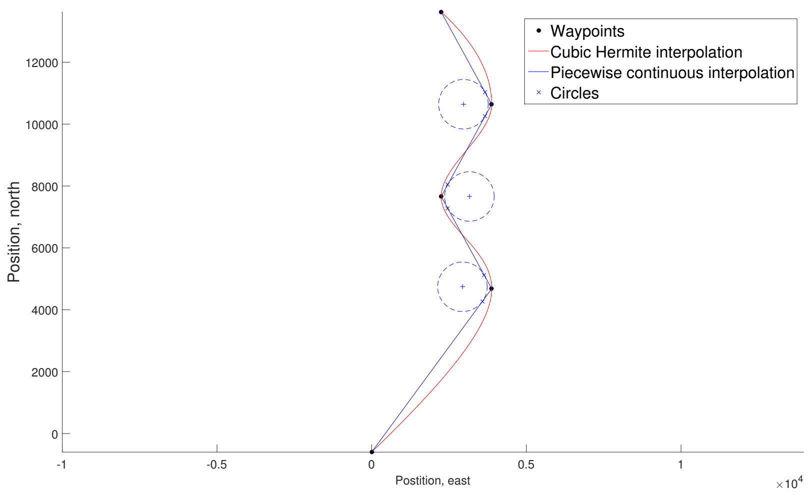


Figure 6: Generated paths using cubic Hermite interpolation, piecewise continuous interpolation, and circles and straight lines

2.2 Path Following

2.2.1 Task 2.2

The guidance system is based on a lookahead-based line-of-sight(LOS) system presented in the book. The ship is set to follow a straight line path given by the previous waypoint and next waypoint. The active line segment is switched when the ship is a desired distance from the next waypoint. The ship is then controlled to follow the next line.

The algorithm can be summed up as

1. Compute the path tangential angle.
2. Compute the cross-track and along-track errors.
3. Compute the desired course from the errors.
4. Compute the desired speed from the errors.
5. Check if the active line segment should be changed.

The desired course is given by

$$\chi_d = \chi_p + \chi_r$$

where χ_p is the path tangential error computed in the first step of the algorithm and the angle χ_r is given by

$$\chi_r = arctan(-\frac{e}{\Delta})$$

where e is the cross-track error and δ is the lookahead distance. This corresponds to setting the course to a point δ ahead of the direct projection down to the active line segment. The lookahead distance works as the gain in a saturated P-controller and chosen is to be 600. Assuming a ship length of 300 meters, a common value for δ is usually between 1.5 and 2.5 of the ship length, simulations showed that a value of 2 gave a good response in terms of aggressiveness in the course towards the line.

The desired speed was modelled as

$$U_d = U_r \frac{s}{\sqrt{s^2 + \Delta_s^2}}$$

where s is the along track error. U_r is the nominal speed of the ship, and was set to 7. The fraction involving the along track error corresponds to a high speed when far from the next waypoint, and a slow down when approaching the waypoint. This is because the Δ_s will dominate when s is small, and yield a small U_d. When far from the next waypoint s will dominate and the fraction is approximately 1, hence giving a speed of approximately U_d. Δ_s was chosen to be 1000.

The determination of when to switch line segment was found by using the intuition from the circles and straight lines and seeing that turning requirements are almost identical for the turns we are considering. Thus the rule for switching line segments is chosen to be when $s < R$, for some $R > 0$. This R was chosen to be 800, which resulted in satisfactory behaviour. It should be noted that it is possible to be more precise here, we did however consider the waypoints as mere guidelines and not destination. For these purposes the turning rules gave satisfactory results.

2.2.2 Task 2.3

See figure 7 and 8.

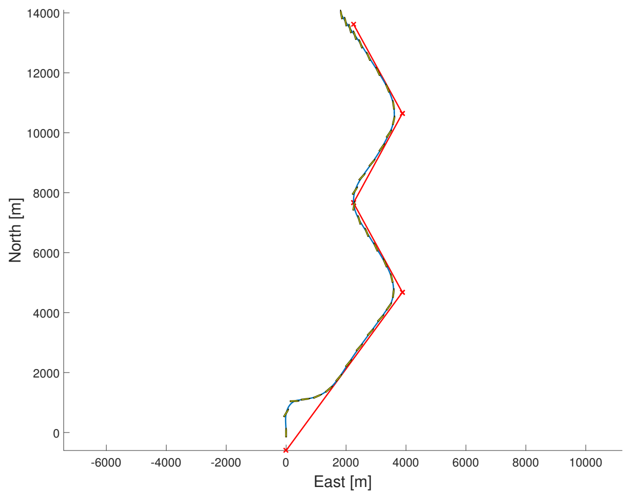


Figure 7: MS Fartøystyring tracking the path given by the waypoints

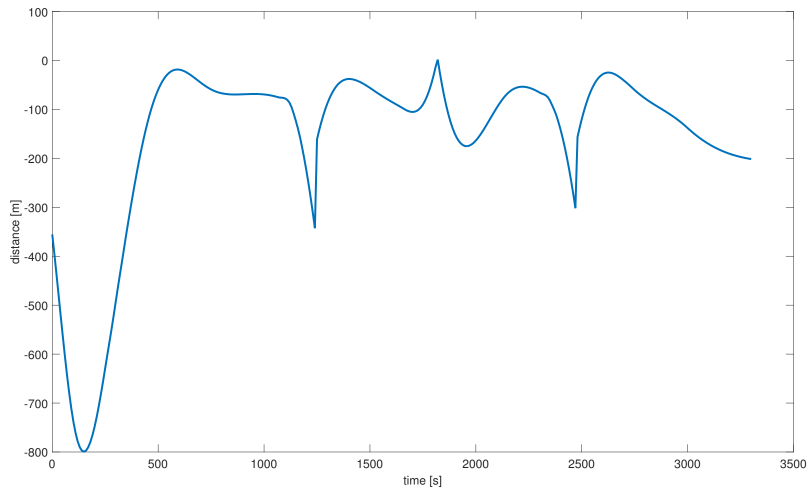


Figure 8: Cross-track error for MS Fartøystyring when tracking the specified path.

2.2.3 Task 2.4

From the figure 7 it can be seen that the ship is not able to stay on the line. The ship is drifting due to the current. The relationship between the course, heading and crab angle is

$$\chi = \psi + \beta. \tag{8}$$

If we assume a perfect heading controller, the heading obtained is

$$\psi = \psi_d = \chi_d$$

Substituting this in (8) gives

$$\chi = \chi_d + \beta.$$

From this equation it can be seen that when applying the course reference to the heading controller the course which is then followed deviates from the desired course by the crab angle, β. This results in the ship not following the proper course.

The speed of the ship is given by

$$U^2 = u^2 + v^2, \tag{9}$$

From turning and currents the sway component, v is clearly nonzero for most cases, thus there is no direct coupling between U_d and u, resulting in the deviation observed.

2.2.4 Task 2.5

For transforming the desired course to a desired heading for the heading controller the relationship presented in (8) can be exploited. Note that the crab angle β is given by

$$sin(\beta) = \frac{v}{U}.$$

Because the controller has perfect state knowledge, the true crab angle can be computed. Rearranging (8) and substituting the true values for the desired values results in the following transformation rule from the desired course to the desired heading

$$\psi_d = \chi_d - \beta = \chi_d - arcsin(\frac{v}{U})$$

The formula (9) can be used to design a transformation from the desired speed to a desired surge speed which is fed into the surge controller. Rearranging the terms, taking the square root and finally substituting the true values for the desired values, the following relationship is obtained

$$u_d = \sqrt{U_d^2 - v^2},$$

where the sway speed, v, is known.

2.2.5 Task 2.6

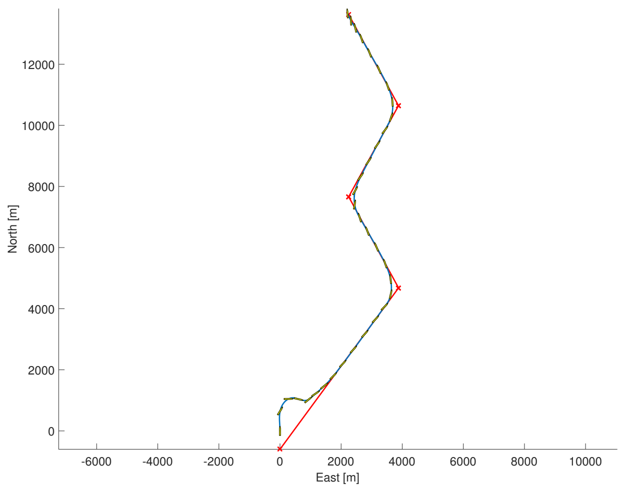


Figure 9: MS Fartøystyring tracking the path given by the waypoints, using transformations.

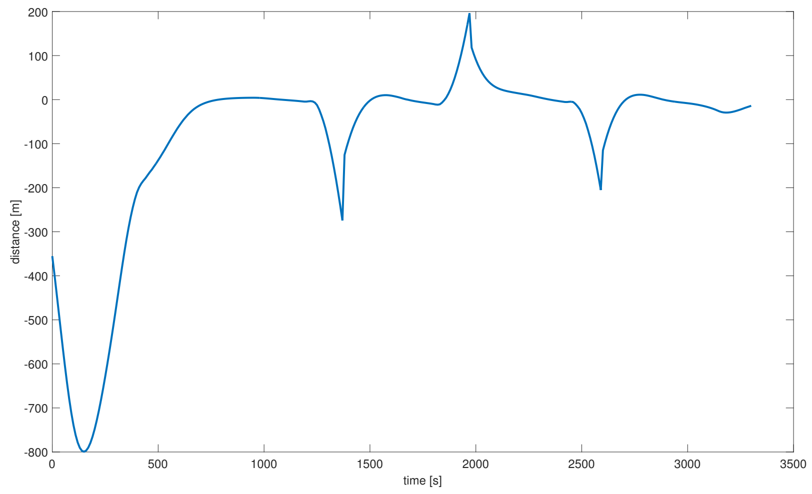


Figure 10: Cross-track error for MS Fartøystyring when tracking the specified path, using transformations.

From figures 9 and 10 it can be seen that the performance of the controller increases significantly when the transformations are applied. The ship is able to counter the effects of the current and get onto the line. In the curves as well the performance is better as it does not slide out. The reason for this is as shown in previous sections using the relationship (8), while now the crab angle is compensated for. Additionally, a reference model is used between the heading transformation and the heading controller to improve performance of the system. The reference model structure for the course angle ψ resembles a low-pass filter and prevents the ship from overcompensating for high frequency crab angle components. The time derivative of the low-pass-filtered course angle ψ used as a reference for the yaw rate $r_d = \dot{\psi}_d$

2.3 Path Tracking

2.3.1 Task 2.7

In order to track the target following a straight line from the first waypoint through the second waypoint, the same path tracking system as in the previous task is used, only without changing waypoints. This is implemented in combination with a target distance controller. That is, the ship will continue to track the path straight path and additionally track to a point 1000m behind the target along the straight line.

In order to control the distance to the target, a speed command U_d is computed using equation (10.15) from the book

$$U_d = U_t + \kappa \frac{s}{\sqrt{s^2 + \Delta_s^2}}$$

where $U_t = 3$ is the target speed, $\Delta_s = 1000$ is the desired distance from the target ship in the along-track direction and $\kappa = 4$ can be interpreted as the approach speed. This is the same concept as employed in previous task. U_t can be viewed as a reference feed forward term, while the fraction involving s will control the speed. If the along track error to the desired tracking point is greater than zero, the ship will increase its speed and try to catch up. On the other hand if the ship is in front of the desired tracking point the non feed forward term will be negative and the ship will slow down in order to keep the specified distance.

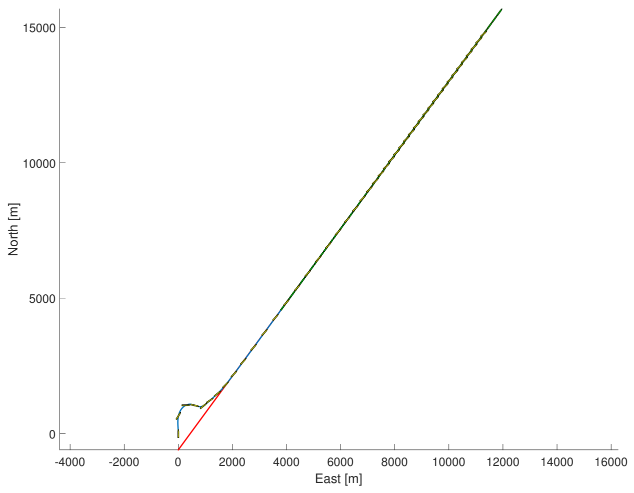


Figure 11: MS Fartøystyring tracking to a position a fixed distance behind the target ship.

Figure 11 and 12 shows that the ship reaches the specified distance of 1000m from the target within a reasonable amount of time and remains at this distance for the remainder of the simulation.

For the sake of demonstration, MS Fartøystyring could also track a fixed distance away from the target ship in the cross-track direction as shown in figure 13. That is, hold a specified position left or right of the target if viewed in the path parallel coordinate system. This is done in the same manner as above by simply setting the cross-track error reference different than zero. This is equivalent to parallel shifting the target path to the LOS line.

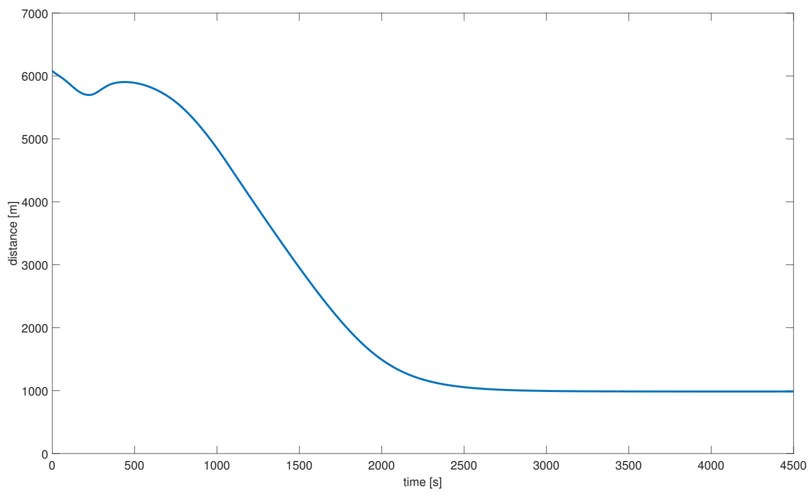


Figure 12: Distance from MS Fartøystyring to the target ship.

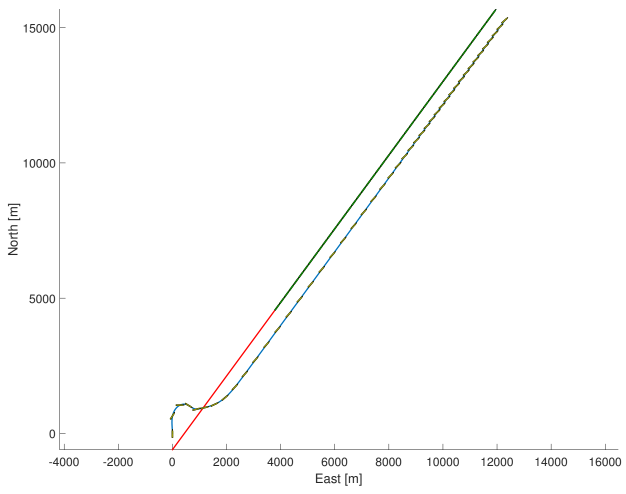


Figure 13: MS Fartøystyring tracking to a position a fixed distance to the side of the target ship.