TTK4190 Guidance and Control of Vehicles

Assignment 3

Autopilots for Path Following/Tracking

Fall 2018

Deadline: Friday the 16th of November at 23:59:00

Objective

The main challenges of this assignment are to develop heading and speed autopilots for the ship MS *Fartøystyring*, and subsequently use those autopilots to perform various path-following tasks. The book and lecture notes from Fossen [1] will be your main source for information, but you may need to use knowledge from previous courses in some problems.

Grading

This assignment must be passed to get access to the final exam. The overall impression of how well you have understood the problems will be the basis for the evaluation. You need at least 60 % of the assignment correct to pass. **You are encouraged/supposed to work in groups of 2-4 people, but are allowed to do the assignment individually if that is preferred for some reason**. Note that the grading will be equally severe if you do the work individually. The participants in the group will receive the same feedback. Because someone might want to change groups, a new set of groups should be used and are available on Blackboard. Use the sign-up sheet called "Assignment 3 - Group" when you register in a group. It is obviously allowed to cooperate with the same people as on the first assignment, but you still need to register again to get access to the delivery. Every group member must be enrolled in the group before you deliver.

Deadline and Delivery details

The assignment must be handed in by **23:59:00 Friday November 16th**. Late submissions will not be accepted. Simulations should be started via a m-file and not the Simulink window. Matlab code and Simulink models should not be included in the report. The MSS Toolbox (available from the course website) is required for this assignment.

This assignment requires a great number of plots/figures to be included in the report. Use common sense and do not place every plot in separate figures. Use "subplot" or "hold on" whenever it is appropriate. Moreover, put figures side-by-side as long as the main result in the figure is still visible. You should look into the function "hgexport" in Matlab to export figures in a nice and easy manner. You can for instance define your own style to increase the line-width and font sizes in your figures and then use that style every time you export a figure. Use degrees and degrees/s as unit in all of your figures showing angles or angular rates.

Introduction

The MS *Fartøystyring* is restricted to movement in the horizontal plane, and is equipped with a main propeller and a main rudder. MS *Fartøystyring* has a length L_{pp} of 304.8 meters. Position, heading, and linear and angular velocities are assumed perfectly measured. The ship is subjected to a constant current. The input-output structure of MS *Fartøystyring* can be seen in Figure 1, where δ_c is the commanded rudder angle and n_c is the commanded shaft speed. The relationships between the commanded inputs and the true inputs are given by

$$\delta = \delta_{\max}\text{sat}(\delta_c/\delta_{\max})$$
$$n = n_{\max}\text{sat}(n_c/n_{\max})$$

where

$$\text{sat}(x) = \begin{cases} 1 & \forall \quad x \geq 1 \\ x & \forall \quad x \in [-1, 1] \\ -1 & \forall \quad x \leq -1 \end{cases}$$

The rudder has a maximum deflection of ± 25 degrees and the shaft has a maximum velocity of $(85 \cdot 2\pi)/60$ rad/s. This is handled in the ship model so you do not have to add saturation blocks to your system.

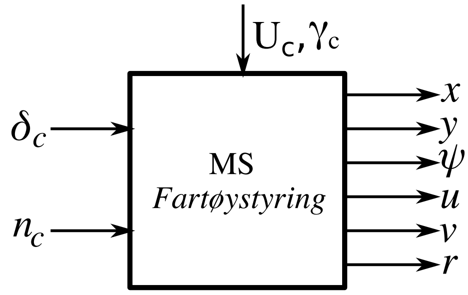

Figure 1: Input-output structure of the ship.

The ship position is given by $\mathbf{p} = [x, y]^{\mathrm{T}}$, represented in NED, with the linear velocity $\mathbf{v} = [u, v]^{\mathrm{T}}$, represented in body. The heading (yaw angle) is represented by ψ, with angular speed (yaw rate) represented by r. The direction of the linear velocity $\mathbf{v}$ is represented by the course χ, while the size of $\mathbf{v}$ (speed) is defined by $U \triangleq \sqrt{u^2 + v^2}$. Since a current is present, the angle $\beta \triangleq \chi - \psi$ is called crab angle and $\beta_r \triangleq \chi_r - \psi$ is called sideslip angle. The current velocity is identified by U_c and its direction by the angle γ_c. The relative velocity is defined by U_r and its direction by χ_r. See Figure 2 for an illustration of these variables.

Simulink code similar to the block shown in Figure 1 is included with this assignment. The internal workings of the block can obviously not be changed. The relationships between inputs and outputs are nonlinear. The current can be switched on and off. Base your code on the supplied code.

1 Autopilot Design (50 %)

In order to design a controller, a model of the involved process is usually helpful as an aid at a preliminary design stage. There are several different models that can be considered, both linear and nonlinear.

1.1 Heading Autopilot

Models for use when designing heading autopilots are described in Chapter 7 of the textbook.

Task 1.1. *If you were to control the MS* Fartøystyring*'s heading, what kind of model would you use? Justify your answer, and write down the model structure.*

Task 1.2. *By employing δ_c and recording the corresponding ψ and r, identify the parameters of the model you chose in Task 1.1. Write down the parameters, and describe the method you used to find them.*
Hints: *You may switch off the current for this task. Note that a negative δ_c gives a positive r, opposite of what is assumed for, e.g., the Nomoto model.*

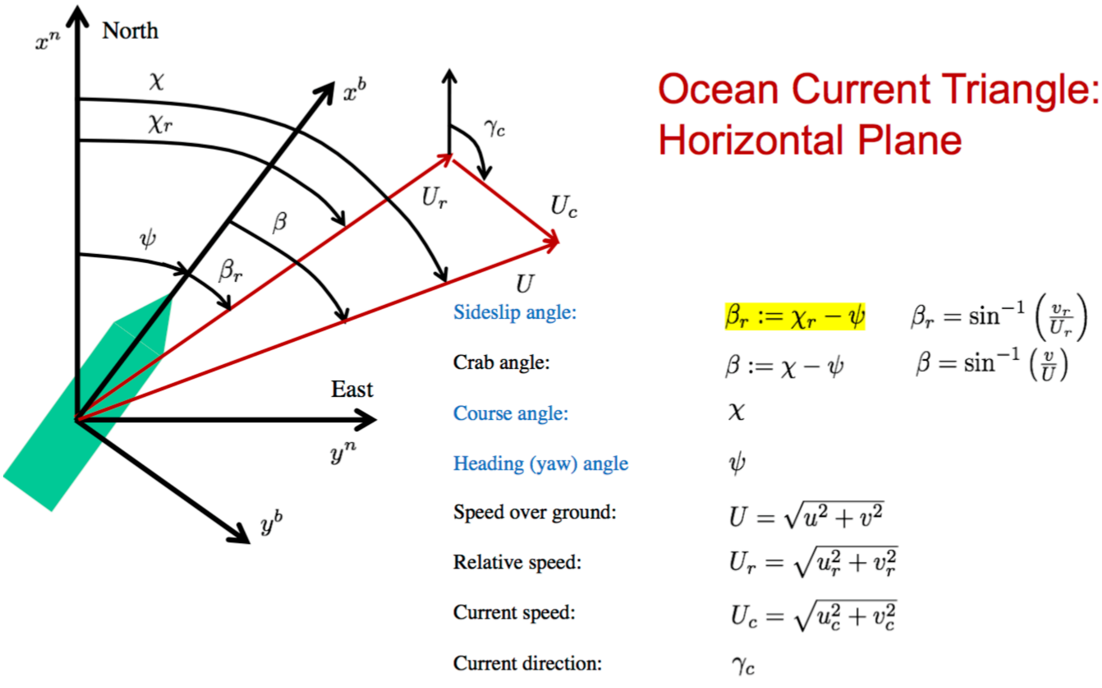

Figure 2: Variable definitions.

We define $\tilde{\psi} \triangleq \psi - \psi_d$ and $\tilde{r} \triangleq r - r_d$, where ψ_d and r_d are desired, time-varying yaw angle and yaw rate reference signals, respectively.

Task 1.3. *Design a control law that enables you to follow constant or slow time-varying reference signals (see Figure 3). The controller must work in the presence of current. Justify your choice of controller and argue for why it works in the presence of current. Write down the controller structure and implement it in Matlab/Simulink. Does the controller have any limitations?*

Task 1.4. *Consider* $\psi_d(t) = 0.4\sin(0.004t)$ rad *and* $r_d(t) = \dot{\psi}_d(t)$*. Simulate the system and use* $u(0) = 6.63$ m/s, $v(0) = \psi(0) = r(0) = 0$ *and* $n_c \equiv 7.3$ rad/s*. Tune the controller until it behaves satisfactorily. Write down the final choice of controller parameters. Comment on the results and add the following figures to the report:*

1. ψ, ψ_d *and* $\tilde{\psi}$
2. r, r_d *and* $\tilde{r}$
3. δ_c *(rudder input) and saturation limits*

For this task, have the current switched on.

1.2 Speed Autopilot

Task 1.5. *If you were to control the MS* Fartøystyring*'s surge speed, what kind of model would you use? Justify your answer, and write down the model structure.*

Task 1.6. *By employing* n_c *and recording the corresponding* u*, identify the parameters of the model you chose in Task 1.5. Write down the parameters, and describe the method you used to find them.* ***Hint:*** *You may switch off the current for this task.*

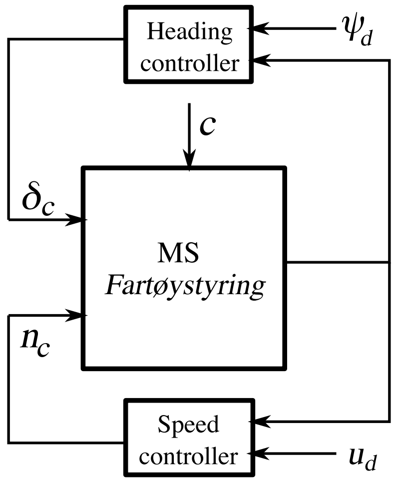

Figure 3: Controller structure for Section 1.

We define $\tilde{u} \triangleq u - u_d$, where u_d is a desired, time-varying surge reference signal.

Task 1.7. *Design a control law that ensures you can follow constant or slow time-varying surge speed commands (see Figure 3). The controller must work in the presence of current. Justify your choice and argue for why it works in the presence of current. Write down the controller structure and implement it in Matlab/Simulink. Does the controller have any limitations?*

Task 1.8. *Consider u_d as a step from 3 to 7 m/s at $t = 1250$. Use $u(0) = 3$ m/s, $v(0) = \psi(0) = r(0) = 0$ and the controller from Task 1.4 to keep $\psi \approx 0$. Tune the control law until it behaves satisfactorily. Write down the final choice of controller parameters. Comment on the results and add the following figures to the report:*

1. *u, u_d and $\tilde{u}$*
2. *ψ and r*
3. *n_c (shaft velocity) and δ_c (rudder input) with saturation limits*

For this task, have the current switched on.

Hint: *It might be advantageous to include a reference model.*

2 Path Following and Target Tracking (50%)

One of the main objectives of the autopilots is to be employed for the purpose of following a path. Based on the current location of the ship and available path information, a guidance system computes a desired velocity and course that, if properly tracked, ensures path following for the ship. Useful guidance laws for path following can be found in the textbook [1].

Included with the assignment is a set of six waypoints, WP. These waypoints can be considered as a sampled selection from an infinite amount of possible paths. The first row of WP consists of the north coordinates of the waypoints and the second row of the east coordinates. By defining a series of straight-line segments between pairs of consecutive waypoints, a piece-wise continuous, non-smooth path can be generated.

2.1 Path Following

In the following tasks, your goal is to get the MS *Fartøystyring* to follow a piece-wise linear path that can be generated from the waypoints. The ship is starting at coordinates $[x(0) = 1000m, y(0) =$

$700m]^\top$, with $u(0) = 6.63$ m/s, $\psi(0) = 60^o$ and $v(0) = r(0) = 0$. Use the straight line defined by the first two waypoints as the first active path segment. You can choose the desired total speed U_d yourself.

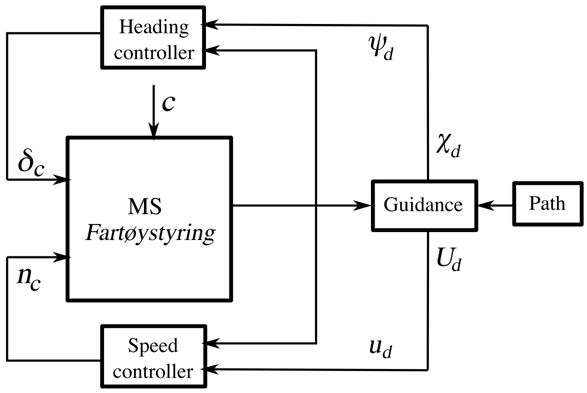

Figure 4: Controller structure for Section 2, Tasks 2.1–2.3.

Task 2.1. *Design a guidance system that employs the ship states and path information as inputs and generates time-varying reference signals χ_d (desired course) and U_d (desired speed). Justify your choices of parameters in the guidance system and explain the principle you are using. Implement the guidance system in Matlab/Simulink.*

Task 2.2. *Use the guidance system outputs as reference signals for the heading and speed autopilots, by directly assigning $\psi_d = \chi_d$ and $u_d = U_d$ (see Figure 4). This is a simulation without crab angle compensation. Simulate the system until you think the MS* Fartøystyring *is sufficiently close to the final way-point. Plot the behavior of the closed-loop system with the included Matlab function pathplotter.m. Include both figures from pathplotter.m in your report. For this task, have the current switched* on.

Task 2.3. *Is the path-following behavior observed in the previous task as expected? Explain the results.*

2.2 Path Following with Crab angle Compensation

Crab angle compensation was not included in the previous task. Therefore, the reference sent into the heading controller may be wrong because the heading angle was directly assigned from the desired course. Moreover, the reference sent into the surge controller was the total desired speed and not necessarily the desired surge speed.

Task 2.4. *Show the problem by displaying a figure with the course χ, heading ψ, desired course χ_d (from the guidance system) and crab angle β from the simulation results acquired in Task 2.2. Include the figure in the report and comment on the results.*

Task 2.5. *We will now include crab angle compensation in our system. Design transformations that purposefully convert χ_d to ψ_d and U_d to u_d, see Figure 5. Justify your conversion methods, and implement them in Matlab/Simulink.*

Task 2.6. *Repeat Tasks 2.2, 2.3 and 2.4 with crab angle compensation. Comment on the difference of the results with and without crab angle compensation.*

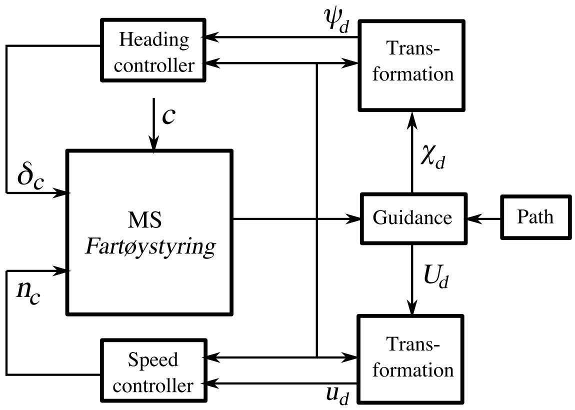

Figure 5: Controller structure for Section 2, Tasks 2.5–2.7.

2.3 Target Tracking

A target is moving with constant speed $U_t = 3$ m/s along the straight line passing through the first two waypoints of WP, with initial position at the second waypoint. It is heading away from the first waypoint. The goal for the following task is to track this target. Use a suitable velocity assignment from Chapter 10 in [1] to track the target. You can assume that you know the target position and velocity at all times, but you have to simulate the target motion yourself. MS *Fartøystyring* has the same initial conditions as stated in Section 2.1.

Task 2.7. *In addition to your path-following system from Task 2.6, include feedback in the speed assignment U_d so that the MS* Fartøystyring *can track the target. Implement and simulate this target-tracking system in Matlab/Simulink from $t = 0$ until you think the ship is sufficiently close to the target, plus a further 200 seconds to show that it remains on target. Make sure you clearly specify the desired distance to target. Plot the behavior of the closed-loop system with the included Matlab function pathplotter.m. Include all figures in your hand-in. For this task, have the current switched* on.

3 Course Evaluation

Task 3.1. *Do the course evaluation on Blackboard :-) The evaluation will be available a few days before the deadline of the assignment.*

References

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