Guidance Function justification file

Author : Gérard Grigore Bousquet

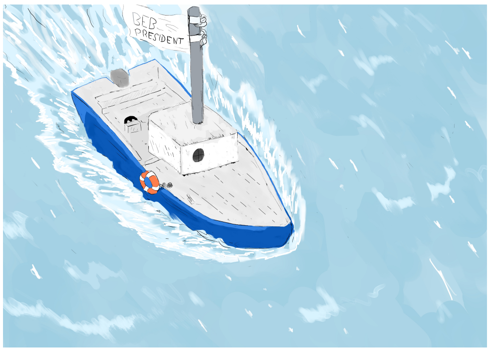

Figure 1: copyright - Jordi Oprisa Bousquet

Object : GN&C Guidance Function design justification file
Version : Simulator version 9
GitHub Release :
https://github.com/GerardGrigore/Casteldos_GNC/releases/tag/0.0.4

This document is the result of a rigorous reflection. It constitutes my intellectual property, although it is based on several bibliographic references.

Contents

Acronym list

DoF Degree of Freedom

ECEF Earth Centered Earth Fixed

ECI Earth Centered Inertial

EFK Extended Kalman Filter

ENU East North Up

GNC Guidance Navigation and Control

GNSS Global Navigation Satellite System

GPS Global Positioning System

I-LOS Integral-Line-of-Sight

INS Inertial Navigation System

KF Kalman Filter

LLA Latitude Longitude Altitude

LOS Line-Of-Sight

LS Least Squares

MAG Magnetometer

NED North East Down

P-LOS Proportional-Line-of-Sight

ReLS Regularized Least Squares

WLS Weighted Leasat Squares

Chapter 1

The Guidance function in the GN&C loop

This file gathers the mathematical equations and algorithms for the Guidance function embedded in the Software and used by the computer to guide the ship. In a general way, the Guidance function receives the information about the estimated current states of the ship, and attempt to answer to the question "given my estimated states and the reference trajectory, which control action corrects my deviation ?". This function will then generate the high level inputs of the Control function, the aimed or reference states, required to correct the trajectory of the ship.

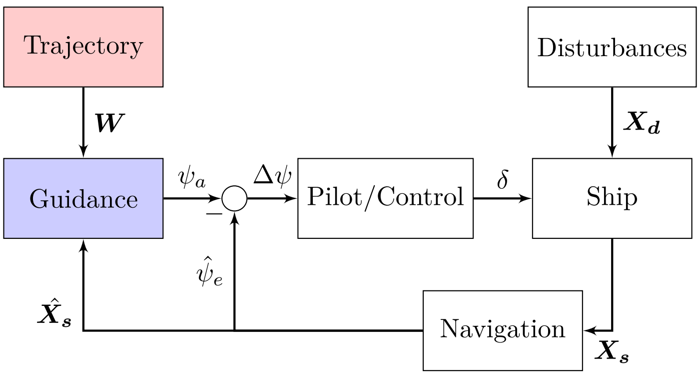

Figure 1.1: GN&C loop for heading control and path-following.

Without loss of generality, one can assume that several environmental perturbations can act on the ship, such as wind, current, waves and wave-induced actions for example. Therefore the vector $\boldsymbol{X_d}$ was used on the previous figure, assuming that several components can act on the vessel. In the last version of Simulator Case 9 for instance, only the waves are considered to perturb the motion of the ship by a contribution ψ_w on the total heading. In this particular case, the vector $\boldsymbol{X_d}$ becomes a scalar ψ_w. Additionally, the motion of the ship is captured through the state vector $\boldsymbol{X_s}$. It is in fact composed by the total heading of the ship ψ_t, its longitudinal position x_s and its lateral position y_s in the local horizontal plane.

In function of the estimated states of the ship $\hat{\boldsymbol{X}}_{\boldsymbol{s}}$; the heading $\hat{\psi}_e$, the local longitudinal estimated position $\hat{x}_s$ and the local lateral estimated position $\hat{y}_s$, the Guidance function computes the required aimed heading to reach the following waypoint. Indeed, the usual Guidance schemes involve the use of a predefined trajectory and an online real-time comparison between the current estimated position of the ship and the one from the trajectory. Then the heading error $\Delta\psi$ is elaborated, and the controller computes the aimed rudder angle δ to be applied by the ship in order to correct its trajectory and aim for the new point of the trajectory (waypoints gathered in $\boldsymbol{W}$).

1.1 Guidance scheme for the ship motion control

There are many different Guidance schemes that can be found in the relevant vessel control literature. In the case of our project, the goal is to perform path-following based on a predefined trajectory taking the form of a list of waypoints $\boldsymbol{W}$. Note that a waypoint is an intermediate point along a route where the course may change, or the vehicle may briefly stop before continuing [1].

Note also the difference between trajectory-tracking an path-following. Indeed, in a path-following problem, the time dependence of the trajectory is removed. Instead of requiring the ship to be at a certain position at a specific time, the objective is simply to converge to and follow the geometric path, moving along it at any speed [1]. In our case, we can even add that the velocity of the ship is an uncontrollable input handled by the user of the ship. Therefore, our algorithms must be tuned accordingly and be robust to velocity variations.

The structure of the Guidance algorithm is then based on the following. In function of the predefined waypoint trajectory and the current estimated location of the ship, the Guidance function generates the required heading angle to reach the next waypoint of the vector $\boldsymbol{W}$. As a consequence, it is mandatory for a specific mission to generate the preconceived aimed trajectory and to inject it into the on-board Software of the embedded computer. In the next section, the simple design of one trajectory will be detailed.

Chapter 2

Trajectory generation

In this section, the Trajectory block from Figure 1.1 will be further described. Since no refreshment nor online update of the trajectory is needed to be done in real-time, it is created on-ground before the mission.

2.1 Trajectory generation before the mission

As mentioned previously, the embedded computer must have a trajectory stored for comparison with the current estimated states of the ship. Let us detail here how a trajectory can be simply generated.

The goal of this whole GN&C project is to design an entire autopilot system, in the sense of software and hardware, for a surface ship. The objective for the system is then to pilot the ship from a reference initial specified point on the Earth's surface, to another one. Here, the trajectory is generated on a small salted lake near the Mediterranean Sea in the very south of France and close to the Spanish border. The ship shall start from the local town's nautical base *Port-Mahon* (marked in green in the top left corner of the following figure) and orient itself towards a small isle called *Île-De-La-Nadière*, an old disused local fisherman village (marked with a grey pin in the bottom right corner of the following Figure).

Figure 2.1: Saled lake/pond used for the mission.

For a better visualization, a theoretical trajectory has been generated on Google Earth. This will not be the effective trajectory, but it is of a very similar shape. Note that there is approximately 3800 meters to be traveled.

Figure 2.2: Saled lake/pond used for the mission from Google Earth's view.

As detailed in the Navigation function justification file [2], a GPS module and a magnetometer are used in order for the ship to estimate its current heading and position on the Earth's surface. The GPS provides with position measurements expressed in the LLA frame. As a consequence, the waypoint vector containing the trajectory path has been initially created in this coordinate frame. As the heading is computed using the positions in the local horizontal frame ENU, these waypoints must be converted to the same reference frame. A specific function has been designed in order to do this conversion in real-time, and can be found in the repository of the project through: conversion from LLA to ENU. The mathematical justification is provided in [2] in section 2.1.1.

NOTA - In the LLA frame, the Altitude term had to be specified as the height in meters above the WGS84 reference ellipsoid. The WGS84 stands for the standard World Geodetic System of 1984 (WGS84). As a first approximation for trajectory generation, and since the project deals only with the 3-DoF without the altitude, the Altitude of the waypoints are fixed to be 0.

The aimed trajectory corresponding to the one described on the previous Figures is therefore:

$$\boldsymbol{W} = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{pmatrix} \tag{2.1}$$

With:

$$\begin{aligned}
W_1 &= \begin{pmatrix} 43.0618927 & 3.0062474 & 0 \end{pmatrix} \\
W_2 &= \begin{pmatrix} 43.0633876 & 3.0100340 & 0 \end{pmatrix} \\
W_3 &= \begin{pmatrix} 43.0614052 & 3.0174674 & 0 \end{pmatrix} \\
W_4 &= \begin{pmatrix} 43.0553345 & 3.0241387 & 0 \end{pmatrix} \\
W_5 &= \begin{pmatrix} 43.0474289 & 3.0330359 & 0 \end{pmatrix} \\
W_6 &= \begin{pmatrix} 43.0406640 & 3.0391810 & 0 \end{pmatrix}
\end{aligned}$$

The reference point for the initial ENU trajectory is located at the Nautical *Port-Mahon* base:

$$W_0 = \begin{pmatrix} 43.059023 & 3.002915 & 0 \end{pmatrix} \tag{2.2}$$

The script allowing the conversion of the waypoints from LLA to ENU is accessible through the following repository: waypoint conversion from LLA to ENU.

Chapter 3

Guidance functions

The function is here to generate the command parameters that shall be realized in order to go from a point A to a point B and to follow the predefined trajectory. As for the Navigation and Control functions, mathematical models are used for this function too to produce realistic orders that must be done. Several state-of-the-art Guidance schemes exist in the frame of ship Guidance, and they will be briefly presented hereafter. Note that the majority of the following development is based on [1] with the detailed demonstrations done here.

3.1 The Guidance problem

The following mathematical description is used to establish the Guidance equations and algorithm for heading control and path-following:

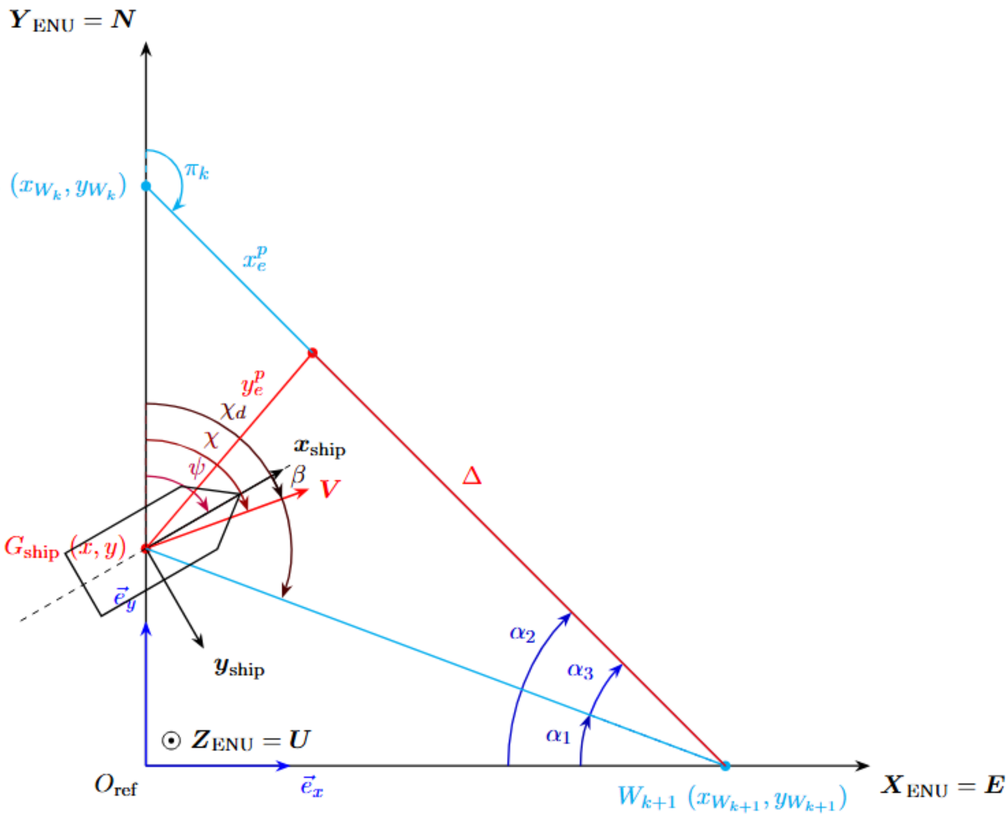

Figure 3.1: Mathematical modeling for heading and path-following Guidance.

The following parameters are on the Figure. ψ is the heading of the ship; the angle between the North direction and the body ship fixed longitudinal axis $\boldsymbol{x_{ship}}$. The course angle χ between the North direction and the current velocity vector $\boldsymbol{V}$ of the ship. The desired course angle, which is the wanted angle between the North direction and the line of sight. β is the crab angle in between the longitudinal ship axis $\boldsymbol{x_{ship}}$ and the velocity vector of the ship $\boldsymbol{V}$. The existence of the crab angle is due to perturbations acting on the ship, implying a misalignment between the velocity vector of the ship and its longitudinal axis. x_e^p is the along-track error, whereas y_e^p is the cross-track error. Δ is a positive number; the look-ahead distance. The angles α_i ,$\forall i \in [\![1, 3]\!]$, are defined as follows:

$$\tan(\alpha_3) = \frac{y_e^p}{\Delta} \Rightarrow \alpha_3 = \arctan\left(\frac{y_e^p}{\Delta}\right) \tag{3.1}$$

Note that here, an abuse of notation has been done in the previous figure. Indeed, the look-ahead distance Δ is not necesarily going to the next waypoint $k + 1$, and most of the times, it does not. In fact, it is a distance placed on the path trajectory and dictated by the direction α_3. Using the alternate interior angles, it is straightforward that:

$$\alpha_1 = \chi_d - \frac{\pi}{2} \tag{3.2}$$

And that:

$$\alpha_2 = \pi_k - \frac{\pi}{2} \tag{3.3}$$

The Guidance scheme used in this project is based on the Line-of-Sight. As extensively findable in the literature, this seems to be the most used concept of Guidance for surface vessels. The goal for the ship is to align its velocity vector $\boldsymbol{V}$ (in red) with the line of sight direction, i.e the cyan straight line from the ship center of gravity G_{ship} to the next waypoint W_{k+1}. The target point is then tuned by choosing the value of the first tuning parameter of the algorithm, namely, the look-ahead distance $\Delta > 0$.

When the ship is considered to be close enough to its aimed point, which will be the $k + 1$ waypoint most of the time, a switching mechanism must be engaged in order to select the next waypoint to aim for. This switching mechanism takes the form of a radius of acceptance, such that by letting $\boldsymbol{X_{W_k}}$ and $\boldsymbol{X_{W_{k+1}}}$ the position vectors of the waypoints k; W_k (x_{W_k}, y_{W_k}), and $k + 1$; W_{k+1} $(x_{W_{k+1}}, y_{W_{k+1}})$:

$$||\boldsymbol{X_{W_{k+1}}} - \boldsymbol{X_{W_k}}|| - |x_e^p| < R_{switch} \tag{3.4}$$

R_{switch} is therefore the second tuning parameter of the algorithm.

3.2 Guidance parameters and aimed heading

The goal is to compute, at a given frequency that characterizes the Guidance action, the aimed parameters. In our case, for the path-following problem, that would be the heading aimed and potentially, the yaw rate angular rate aimed. Based on the previous figure note that the relation between the heading ψ and the course angle χ is:

$$\psi + \beta = \chi \tag{3.5}$$

Therefore, in terms of desired aimed heading ψ_d:

$$\psi_d = \chi_d - \beta \tag{3.6}$$

On the other hand:

$$\alpha_2 = \alpha_1 + \alpha_3$$

Leading to:

$$\pi_k - \frac{\pi}{2} = \chi_d - \frac{\pi}{2} + \arctan\left(\frac{y_e^p}{\Delta}\right)$$

Using equation 3.6 leads to:

$$\pi_k = \psi_d + \beta + \arctan\left(\frac{y_e^p}{\Delta}\right) \tag{3.7}$$

Hence:

$$\psi_d = \pi_k - \beta - \arctan\left(\frac{y_e^p}{\Delta}\right) \tag{3.8}$$

Where π_k is the direction of the line connecting the waypoints k and $k + 1$, β is the crab angle and the last term is composed by the cross-track error and the look-ahead distance.

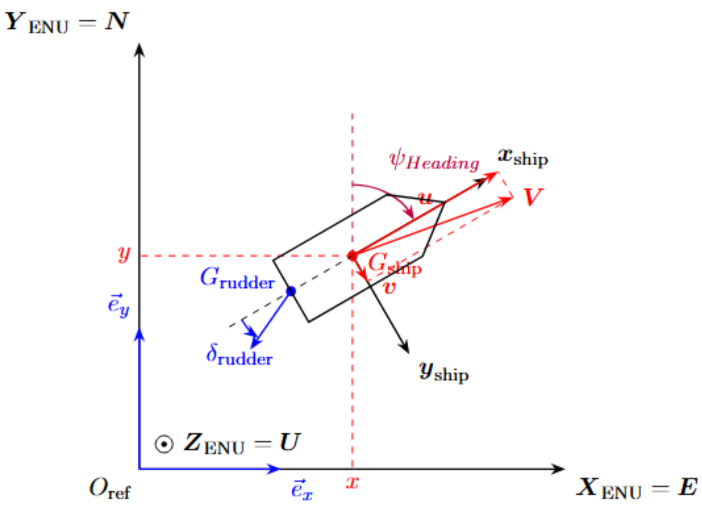

Figure 3.2: Ship parametrization and velocity vector and components.

π_k is known and can be computed at each step time using the embedded trajectory and the fact that:

$$\pi_k = \text{atan2}(x_{W_{k+1}} - x_{W_k}, y_{W_{k+1}} - y_{W_k}) \tag{3.9}$$

Concerning the crab angle, based on the previous figure and on the following one:

It corresponds to the angle between the longitudinal axis of the ship $\boldsymbol{x_{ship}}$ and its velocity vector $\boldsymbol{V}$. As a result, one can note that:

$$\tan(\beta) = \frac{v}{u} \;\Rightarrow\; \beta = \arctan\left(\frac{v}{u}\right) \tag{3.10}$$

NOTA - In many cases, β is treated as a disturbance acting upon the yaw components [1]. This allows to avoid retrieving the measurements on the velocities u and v. On the other hand, compared to the other terms in 3.8, this contribution can be judged to be small for the specific mission of exploring the pond during a sunny and windless day. designing a Guidance algorithm that works with this term is dimensioning, since we have also made the assumption that under the nominal conditions, $u \approx V$ and $v \approx 0$, hence $\beta \approx 0$ in the Navigation mathematical embedded models. As it will be explained, this specific value of β will also be the one implemented in the final Guidance algorithm.

The last unknown in equation 3.8 is the cross-track error y_e^p. The previous Figure 3.1 can be slightly modified by adding at the waypoint k a second frame such as:

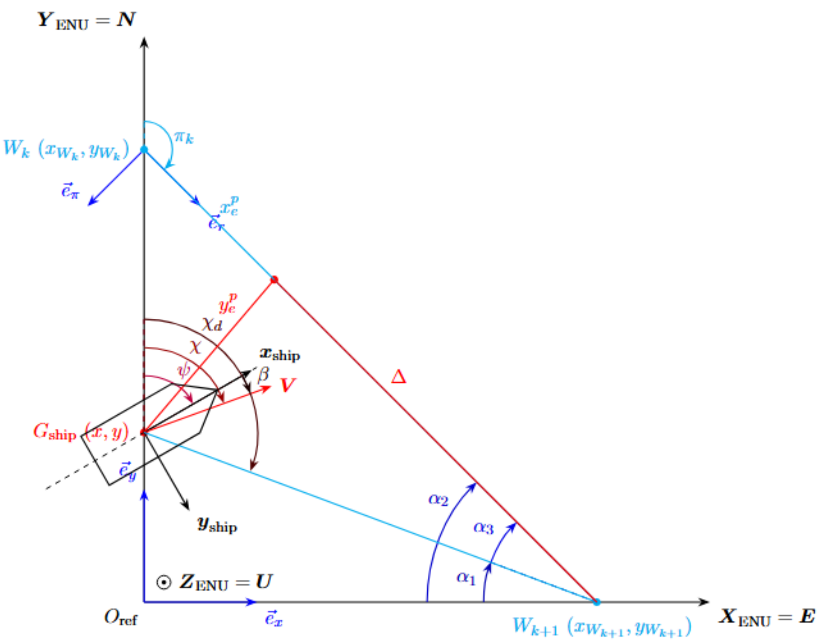

Figure 3.3: Mathematical modeling for heading and path-following Guidance updated.

Based on this second Figure, one can note that the along-track error is of the vectorial form:

$$\boldsymbol{x_e^p} = x_\epsilon.\boldsymbol{e_r}$$

$$\boldsymbol{x_e^p} = x_\epsilon.(\sin(\pi_k).\boldsymbol{e_x} + \cos(\pi_k).\boldsymbol{e_y}) \tag{3.11}$$

On the other hand, the usual vector relations allow us to write that, by denoting $P(x_p, y_p)$ to be the junction point of the y_e^p red straight line with the cyan one linking waypoints k and $k+1$:

$$\boldsymbol{x_e^p} = (x_p - x_{W_k}).\boldsymbol{e_x} + (y_p - y_{W_k}).\boldsymbol{e_y} \tag{3.12}$$

Additionally:

$$\boldsymbol{y_e^p} = y_\epsilon.\boldsymbol{e_\pi}$$

$$\boldsymbol{y_e^p} = y_\epsilon.(\cos(\pi_k).\boldsymbol{e_x} - \sin(\pi_k).\boldsymbol{e_y}) \tag{3.13}$$

Again, by its vectorial definition, it holds that:

$$\boldsymbol{y_e^p} = (x - x_p).\boldsymbol{e_x} + (y - y_p).\boldsymbol{e_y} \tag{3.14}$$

With (x, y) the coordinates of the ship COG G_{ship}. Using the previous relations 3.12 and 3.14, it holds that:

$$\boldsymbol{x_e^p} + \boldsymbol{y_e^p} = (x_p - x_{W_k}).\boldsymbol{e_x} + (y_p - y_{W_k}).\boldsymbol{e_y} + (x - x_p).\boldsymbol{e_x} + (y - y_p).\boldsymbol{e_y}$$

Hence:

$$\boldsymbol{x_e^p} + \boldsymbol{y_e^p} = (x - x_{W_k}).\boldsymbol{e_x} + (y - y_{W_k}).\boldsymbol{e_y} \tag{3.15}$$

Using equations 3.11 and 3.13, we also have:

$$\boldsymbol{x_e^p} + \boldsymbol{y_e^p} = x_\epsilon.(\sin(\pi_k).\boldsymbol{e_x} + \cos(\pi_k).\boldsymbol{e_y}) + y_\epsilon.(\cos(\pi_k).\boldsymbol{e_x} - \sin(\pi_k).\boldsymbol{e_y}) \tag{3.16}$$

Identifying the terms in equations 3.15 and 3.16 leads to:

$$x - x_{W_k} = x_\epsilon.\sin(\pi_k) + y_\epsilon.\cos(\pi_k) \tag{3.17}$$

$$y - y_{W_k} = x_\epsilon.\cos(\pi_k) - y_\epsilon.\sin(\pi_k) \tag{3.18}$$

This system is of the form:

$$\begin{pmatrix} x - x_{W_k} \\ y - y_{W_k} \end{pmatrix} = \begin{pmatrix} \sin(\pi_k) & \cos(\pi_k) \\ \cos(\pi_k) & -\sin(\pi_k) \end{pmatrix} . \begin{pmatrix} x_\epsilon \\ y_\epsilon \end{pmatrix} = \boldsymbol{A}. \begin{pmatrix} x_\epsilon \\ y_\epsilon \end{pmatrix} \tag{3.19}$$

And here, one can note that:

$$\boldsymbol{A} = \begin{pmatrix} \sin(\pi_k) & \cos(\pi_k) \\ \cos(\pi_k) & -\sin(\pi_k) \end{pmatrix} = \boldsymbol{A}^T$$

Hence $\boldsymbol{A}$ is symmetric. On the other hand:

$$\boldsymbol{A}.\boldsymbol{A}^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \boldsymbol{I_2}$$

Therefore, matrix $\boldsymbol{A}$ is a square, symmetric and orthogonal matrix. It leads to the fact that $\boldsymbol{A}^{-1} = \boldsymbol{A}^T$. The system in equation 3.19 can be solved, leading to:

$$\begin{pmatrix} x_\epsilon \\ y_\epsilon \end{pmatrix} = \boldsymbol{A}^T . \begin{pmatrix} x - x_{W_k} \\ y - y_{W_k} \end{pmatrix} \tag{3.20}$$

This leads to the along-track error norm:

$$x_\epsilon = (x - x_{W_k}). \sin(\pi_k) + (y - y_{W_k}). \cos(\pi_k) \tag{3.21}$$

And the cross-track error norm:

$$y_\epsilon = (x - x_{W_k}). \cos(\pi_k) - (y - y_{W_k}). \sin(\pi_k) \tag{3.22}$$

3.3 Classical Line-of-Sight Guidance function

In this section will be presented the classical equations that governs the classical LOS Guidance algorithm. The goal of the LOS algorithm is to provide the desired aimed heading to be sent to the controller/pilot. According to equation 3.8 recalled here:

Desired heading

$$\psi_{d_k} = \pi_k - \beta_k - \arctan\left(\frac{y^p_{e_k}}{\Delta}\right) \tag{3.23}$$

There is a need to have the trajectory path angle π_k, the crab angle β_k and the cross-track error $y^p_{e_k}$ at the current step time k.
One equation allowing to retrieve the cross-track error is the following:

Cross-track error time derivative

$$\dot{y}^p_e = V. \sin(\chi - \pi_k) \tag{3.24}$$

Indeed, according to our adopted ENU convention on Figure 3.2, the velocity vector expresses as:

$$\boldsymbol{V}^{[\boldsymbol{R_{ship}}]} = u.\boldsymbol{x_{ship}} + v.\boldsymbol{y_{ship}} \tag{3.25}$$

Which gives, in the inertial frame ENU:

$$\boldsymbol{V}^{[\boldsymbol{R_{ENU}}]} = u.(\sin(\psi).\boldsymbol{e_x} + \cos(\psi).\boldsymbol{e_y}) + v.(\cos(\psi).\boldsymbol{e_x} - \sin(\psi).\boldsymbol{e_y}) \tag{3.26}$$

$$\boldsymbol{V}^{[\boldsymbol{R_{ENU}}]} = (u. \sin(\psi) + v. \cos(\psi)).\boldsymbol{e_x} + (u. \cos(\psi) - v. \sin(\psi)).\boldsymbol{e_y} \tag{3.27}$$

On the one hand, we can let $u = V. \cos(\beta)$ and $v = V. \sin(\beta)$:

$$u. \sin(\psi) + v. \cos(\psi) = V. \cos(\beta). \sin(\psi) + V. \sin(\beta). \cos(\psi) = V. \sin(\beta + \psi) = V. \sin(\chi)$$

On the other hand:

$$u. \cos(\psi) - v. \sin(\psi) = V. \cos(\beta). \cos(\psi) - V. \sin(\beta). \sin(\psi) = V. \cos(\beta + \psi) = V. \cos(\chi)$$

Therefore, as stated by equation 3.10:

$$\frac{v}{u} = \frac{V. \sin(\beta)}{V. \cos(\beta)} = \tan(\beta) \;\Rightarrow\; \beta = \arctan\left(\frac{v}{u}\right) \tag{3.28}$$

Injecting those results in equation 3.27 leads to:

$$\boldsymbol{V}^{[\boldsymbol{R_{ENU}}]} = V.\sin(\chi).\boldsymbol{e_x} + V.\cos(\chi).\boldsymbol{e_y} \tag{3.29}$$

Furthermore, by adopting the convention of a positive cross-track error y_e^p at the right of the trajectory in Figure 3.1 along $\boldsymbol{e_\pi}$, it can be noted that:

$$\boldsymbol{y_e^p} = \cos(\pi_k).\boldsymbol{e_x} - \sin(\pi_k).\boldsymbol{e_y} \tag{3.30}$$

The projection of the velocity vector along the cross-track error vector can be expressed as:

$$\dot{y}_e^p = \boldsymbol{V}^{[\boldsymbol{R_{ENU}}]}.\boldsymbol{y_e^p} = V.\sin(\chi).\cos(\pi_k) - V.\cos(\chi).\sin(\pi_k) \tag{3.31}$$

$$\dot{y}_e^p = V.(\sin(\chi).\cos(\pi_k) - \cos(\chi).\sin(\pi_k))$$

$$\dot{y}_e^p = V.\sin(\chi - \pi_k) \tag{3.32}$$

Therefore, the following set of equations are used in the classical LOS Guidance algorithm:

Equations for the classical LOS Guidance algorithm

$$\dot{y}_e^p = V.\sin(\chi - \pi_k) \tag{3.33}$$

$$\beta = \arctan\left(\frac{v}{u}\right) \tag{3.34}$$

$$\pi_k = \mathrm{atan2}(x_{W_{k+1}} - x_{W_k}, y_{W_{k+1}} - y_{W_k}) \tag{3.35}$$

$$\psi_{d_k} = \pi_k - \beta_k - \arctan\left(\frac{y_{e_k}^p}{\Delta}\right) \tag{3.36}$$

NOTA - As previously stated, β is treated as a disturbance acting upon the yaw components [1]. Additionally, the discretization of equation 3.33 would lead to the following form to retrieve the cross-track error: $y_{e_k}^p = y_{e_{k-1}}^p + \delta t.V_{k-1}.\sin(\chi_{k-1} - \pi_{k-1}) = y_{e_{k-1}}^p + \delta t.V_{k-1}.\sin(\psi_{k-1} + \beta_{k-1} - \pi_{k-1})$. In fact, what will be implemented in the final Guidance algorithm will be an alternative way to calculate the cross-track error, based on equation 3.22: $y_\epsilon = (x - x_{W_k}).\cos(\pi_k) - (y - y_{W_k}).\sin(\pi_k)$.

3.4 Proportional Line-of-Sight Guidance function

In the Proportional LOS (P-LOS) Guidance algorithm, the look-ahead distance can be interpreted as a proportional tunable gain such that:

$$K_p = \frac{1}{\Delta} \tag{3.37}$$

Therefore leading to the following set of equations for the P-LOS algorithm:

Equations for the P-LOS Guidance algorithm

$$\dot{y}_e^p = V.\sin(\chi - \pi_k) \tag{3.38}$$

$$\beta = \arctan\left(\frac{v}{u}\right) \tag{3.39}$$

$$\pi_k = \mathrm{atan2}(x_{W_{k+1}} - x_{W_k}, y_{W_{k+1}} - y_{W_k}) \tag{3.40}$$

$$\psi_{d_k} = \pi_k - \beta_k - \arctan\left(K_p.y_{e_k}^p\right) \tag{3.41}$$

For more details about this algorithm, refer to [1]. This will not be the implemented one. The chosen Guidance algorithm is the Integral LOS (I-LOS) Guidance algorithm, presented in the next section.

3.5 Integral Line-of-Sight Guidance function

Two major issues with the classical LOS and the P-LOS algorithms are:

- The compensation of the unknown angle β which is hard to estimated.
- The fact that if the ship is on its trajectory, this implies that the cross-track error is null $y_e^p = 0$. Therefore, if $\psi_{LOS} = -\arctan\left(\frac{y_e^p}{\Delta}\right)$ then $\psi_{LOS} = 0$. This means that the heading of the ship is equal to 0, which is not necessarily true.

It seems that one of the solutions to tackle these problems is to consider the following for for the desired heading [1]:

$$\psi_{d_k} = \pi_k - \arctan\big(K_p.y_{e_k}^p + K_i.y_i\big) \tag{3.42}$$

With:

$$\dot{y}_i = \frac{\Delta.y_e^p}{\Delta^2 + (y_e^p + \kappa.y_i)^2} \tag{3.43}$$

A proposed explanation can be done as follows. Concerning the P-LOS algorithm, we showed that equations 3.38 and 3.41 holds, recalled respectively hereafter:

$$\dot{y}_e^p = V.\sin(\chi - \pi_k) = V.\sin(\psi + \beta - \pi_k)$$

$$\psi_{d_k} = \pi_k - \beta_k - \arctan\big(K_p.y_{e_k}^p\big)$$

Let us place ourselves in the equilibrium regime, by definition different than the transitional regime. If the ship is at the equilibrium, it means that it perfectly follows its trajectory, hence $\psi_d = \psi$. In that case, using the last equation, if β is known, then:

$$\psi = \pi_k - \beta_k - \arctan\big(K_p.y_{e_k}^p\big)$$

Hence:

$$\psi + \beta_k - \pi_k = -\arctan\big(K_p.y_{e_k}^p\big) \tag{3.44}$$

Injecting this into the first recalled time derivative of the cross-track error leads to:

$$\dot{y}_e^p = V.\sin(\psi + \beta - \pi_k) = V.\sin\big(-\arctan\big(K_p.y_{e_k}^p\big)\big) = -V.\frac{y_{e_k}^p}{\sqrt{y_{e_k^p}^2 + \Delta^2}} \tag{3.45}$$

At the equilibrium, $\dot{y}_e^p = 0$ which implies using the previous equation that $y_e^p = 0$. However, this result is only accurate when one supposed that the current contribution β is known. Which is not true in most of the times. Therefore, the equation 3.44 is not accurate. In fact, without the knowledge on the current, one has:

$$\psi = \pi_k - \arctan\big(K_p.y_{e_k}^p\big) \tag{3.46}$$

Leading to:

$$\psi + \beta - \pi_k = \beta - \arctan\big(K_p.y_{e_k}^p\big) \tag{3.47}$$

Again, injecting this into equation 3.38 recalled in the beginning of this aside leads to:

$$\dot{y}_e^p = V.\sin(\psi + \beta - \pi_k) = V.\sin\big(\beta - \arctan\big(K_p.y_{e_k}^p\big)\big) \tag{3.48}$$

As mentioned earlier, at the equilibrium, $\dot{y}_e^p = 0$, leading in our case to:

$$\sin\big(\beta - \arctan\big(K_p.y_{e_k}^p\big)\big) = 0$$

$$\beta - \arctan\big(K_p.y_{e_k}^p\big) \equiv 0 \pmod{\pi} \tag{3.49}$$

We can state that:

$$\beta = \arctan\big(K_p.y_{e_k}^p\big) \tag{3.50}$$

Or, equivalently:

$$y_{e_k}^p = \frac{\tan(\beta)}{K_p} \neq 0 \tag{3.51}$$

Therefore, in presence of unobserved currents, the cross-track error in the permanent equilibrium regime is not null. This is the main limit of the P-LOS Guidance algorithm. The I-LOS Guidance algorithm tackles this issue. Indeed, in this case, the desired heading takes the form of equation 3.42:

$$\psi_{d_k} = \pi_k - \arctan\big(K_p.y^p_{e_k} + K_i.y_i\big)$$

To begin, one sees that the current angle β is not taken into account here. As for the P-LOS description, let us consider that the ship is in its permanent regime, perfectly following its trajectory. Then, $\psi = \psi_d$ and:

$$\psi + \beta - \pi_k = \beta - \arctan\big(K_p.y^p_{e_k} + K_i.y_i\big) \tag{3.52}$$

Injecting this expression in equation 3.38 leads to:

$$\dot{y}^p_e = V.\sin(\psi + \beta - \pi_k) = V.\sin\big(\beta - \arctan\big(K_p.y^p_{e_k} + K_i.y_i\big)\big) \tag{3.53}$$

As previously stated, in the equilibrium regime, $\dot{y}^p_e = 0$, but this time, $\dot{y}_i = 0$ too. Using equation 3.43, letting $\dot{y}_i = 0$ implies that $y^p_e = 0$. The first conclusion is that the I-LOS action forces the cross-track error to converge to 0 in the permanent regime, even if the current angle β is unknown. Additionally:

$$\sin\big(\beta - \arctan\big(K_p.y^p_{e_k} + K_i.y_i\big)\big) = 0 \tag{3.54}$$

Implies that:

$$\beta = \arctan\big(K_p.y^p_{e_k} + K_i.y_i\big) \tag{3.55}$$

At equilibrium. Recall that in this regime, $y^p_e = 0$, leading to:

$$\beta = \arctan(K_i.y_i) \tag{3.56}$$

Where here y_i is at the equilibrium. This previous equation could be seen as an implicit observer of the current crab angle, only considering the permanent regime of the ship. The integral action tunes itself automatically to create a variation in the heading that compensates the current crab angle.

Concerning the choice of the expression of $\dot{y}_i$ in equation 3.43, a proposed explanation can be done as follows. From the previous development, we know that at the equilibrium, the following equivalence must hold:

$$\dot{y}_i = 0 \;\Leftrightarrow\; y^p_e = 0$$

We also note that to compensate the unknown current crab angle, equation 3.56 must hold, implying that $K_i.y_i$ must converge to $\tan(\beta)$. The law on $\dot{y}_i$ must then be as follows:

- y_i must vary as long as the cross-track error is not null.
- It must stop varying when the cross-track error is equal to zero.
- It must be stable and bounded for all y^p_e.

Naively, we could have stated that $\dot{y}_i = y^p_e$. However, it is obvious that this form does not take into account the LOS geometrical problem. In addition, it does not ensure and certify for global stability. This can be linked to the wind-up problem; what if the cross-track error becomes important. The integral term will increase and maybe diverge. The expression presented through equation 3.43 seems to be suitable due to the following elements:

- We can see that $\dot{y}_i = 0 \;\Leftrightarrow\; y^p_e = 0$.
- The denominator of $\dot{y}_i$, namely $\Delta^2 + (y^p_e + \kappa.y_i)^2$, is the same expression as when taking the time derivative of $\arctan\big(K_p.y^p_{e_k} + K_i.y_i\big)$. Therefore, the geometrical LOS problem is embedded in this expression and when y_i varies, then it creates a bounded variation of the desired heading.
- The local behavior when $y^p_e << 1$ implies that: $\dot{y}_i \approx \frac{y^p_e}{\Delta}$ which is a classical integrator action. As known, this type of action allows a suppression of the static error.
- This expression seems to behave as a natural anti wind-up system. Indeed, if the cross-track error y^p_e becomes too high, the denominator will be high, leading to a low value of $\dot{y}_i$. It therefore guarantees stability.
- Direct link with the learning of β; while the current is not counterbalanced, $y^p_e \neq 0 \;\Rightarrow\; \dot{y}_i \neq 0$. This means that y_i will be varying, therefore the desired heading too.

Eventually, the equations used for the I-LOS Guidance algorithm are the following:

Equations for the I-LOS Guidance algorithm

$$y_{e_k}^p = (x - x_{W_k}).\cos(\pi_k) - (y - y_{W_k}).\sin(\pi_k) \tag{3.57}$$

$$\pi_k = \text{atan2}(x_{W_{k+1}} - x_{W_k}, y_{W_{k+1}} - y_{W_k}) \tag{3.58}$$

$$\psi_{d_k} = \pi_k - \arctan\big(K_p.y_{e_k}^p + K_i.y_i\big) \tag{3.59}$$

$$\dot{y_i} = \frac{\Delta.y_e^p}{\Delta^2 + (y_e^p + \kappa.y_i)^2} \tag{3.60}$$

The final algorithm implemented in the Maritime Software can be found here: I-LOS Guidance algorithm.

3.6 Reference models for the Guidance function

By using this algorithm, a desired heading will be generated at each step time of the Guidance function. However, it can present steep variations similar to step commands and does not necessarily takes into account the limited available angular acceleration nor the constraints on the rudder angle for example, that will play a role in the possibility to reach a desired heading. For example, on the following Figure, two curves are plotted:

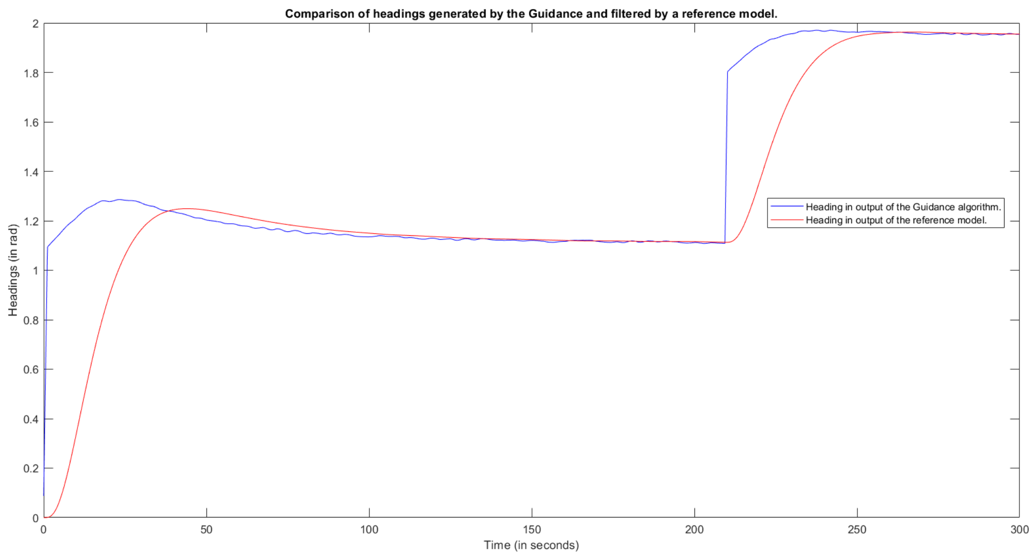

Figure 3.4: Heading from Guidance (in blue) and heading in output of a filter reference model (in red).

The blue plot corresponds to the heading generated by the I-LOS algorithm. It can be seen that the desired heading brutally varies, at the beginning of the simulation as well as at around 200 seconds with respect to the beginning of the simulation. Additionally, one can detect a slightly oscillatory motion in this raw desired heading, particularly in between 60 seconds to 200 seconds for example. These features, coupled with no physical constraint of the system embedded in the Guidance algorithm, can lead to bad motion control and eventually, to the divergence and failure of the GN&C chain.

To tackle this, the raw heading aimed generated by the Guidance function can be injected into a filter, usually called a reference model [1]. In the output of such a model, the heading is usually filtered and realistic with regard to what the actuators and the pilot can do. The red curve on the previous Figure is the filtered heading that will be sent to the Controller function.

The mathematical relation between the raw heading generated by the I-LOS Guidance algorithm and the reference heading in the output of the reference model can be expressed:

$$\psi_{d_{ref}} = H_m.\psi_{ILOS_{raw}} \tag{3.61}$$

The heading $\psi_{d_{ref}}$ will be sent to the pilot controller in the output of the Guidance function. H_m is the reference model transfer function typically chosen to be of order three [1] such that:

$$\frac{\psi_{d_{ref}}}{\psi_{ILOS_{raw}}} = H_m = \frac{\omega_n^2}{(1+T.s).(s^2+2.\xi.\omega_n.s+\omega_n^2)} \tag{3.62}$$

After development, this expression can take the form:

$$H_m = \frac{\frac{\omega_n^2}{T}}{s^3+(\frac{1}{T}+2.\xi.\omega_n).s^2+(2.\xi.\frac{\omega_n}{T}+\omega_n^2).s+\frac{\omega_n^2}{T}} \tag{3.63}$$

If one let $\omega_n = \frac{1}{T}$, then the expression can be further simplified, leading to:

$$H_m = \frac{\omega_n^3}{s^3+(1+2.\xi).\omega_n.s^2+(2.\xi+1).\omega_n^2.s+\omega_n^3} \tag{3.64}$$

For the reference model to be damped, the damping coefficient ξ shall be chosen to be inside of [0.6;1.0]. The natural pulsation of the reference model shall be lower than the autopilot one. This must be understood as the fact that the control action must be done most frequently than the guidance one. The following set of equations will be used as a reference model equivalent to the transfer function 3.64:

Reference model for heading changing turning maneuver

$$\dot{\psi}_d = sat(r_d) \tag{3.65}$$

$$\dot{r}_d = sat(\gamma_d) \tag{3.66}$$

$$\dot{\gamma}_d = -(2.\xi+1).\omega_n.sat(\gamma_d) - (2.\xi+1).\omega_n^2.sat(r_d) + \omega_n^3.(\psi_r - \psi_d) \tag{3.67}$$

As explained in [3], the use of a rate limitation in the reference model is justified since a turning maneuver can be decomposed into three phases:

- A constant angular acceleration, producing an affine increase in the angular velocity.
- A null acceleration, producing a constant angular velocity during the turning maneuver.
- A deceleration at the end of the turn.

It can be shown that equations 3.65 to 3.67 are in fact equivalent to the transfer function H_m of equation 3.64. Indeed, by neglecting the saturation and by switching to the Laplace domain, the equations are:

$$s.\psi_d = r_d \tag{3.68}$$

$$s.r_d = \gamma_d \tag{3.69}$$

$$s.\gamma_d = -(2.\xi+1).\omega_n.\gamma_d - (2.\xi+1).\omega_n^2.r_d + \omega_n^3.(\psi_r - \psi_d) \tag{3.70}$$

this leads to:

$$\gamma_d.(s+(2.\xi+1).\omega_n) = -(2.\xi+1).\omega_n^2.\frac{\gamma_d}{s} + \omega_n^3.\psi_r - \omega_n^3.\psi_d$$

$$\gamma_d.(s+(2.\xi+1).\omega_n + (2.\xi+1).\frac{\omega_n^2}{s}) = \omega_n^3.\psi_r - \omega_n^3.\frac{\gamma_d}{s^2}$$

$$\gamma_d.(s+(2.\xi+1).\omega_n + (2.\xi+1).\frac{\omega_n^2}{s} + \frac{\omega_n^3}{s^2}) = \omega_n^3.\psi_r$$

Since $\gamma_d = s^2.\psi_d$:

$$s^2.\psi_d.(s+(2.\xi+1).\omega_n+(2.\xi+1).\frac{\omega_n^2}{s}+\frac{\omega_n^3}{s^2}) = \omega_n^3.\psi_r = \psi_d.(s^3+(2.\xi+1).\omega_n.s^2+(2.\xi+1).\omega_n^2.s+\omega_n^3)$$

Since here ψ_r stands for the raw heading generated by the Guidance function and ψ_d is the final desired heading after the filter, one has:

$$\frac{\psi_d}{\psi_r} = \frac{\omega_n^3}{s^3+(2.\xi+1).\omega_n.s^2+(2.\xi+1).\omega_n^2.s+\omega_n^3} \tag{3.71}$$

Equation 3.71 is effectively equivalent to the equation of the third order reference model transfer function 3.64.

An embedded algorithm has also been created and is available through this link: reference model for heading Guidance. As mentioned earlier, since the reference model employs saturation function, the maximal rudder rate and rudder acceleration are the tuning parameters of this algorithm. For the rudder rate, the same value must be selected as the one settled in the saturation of the actuators in the control section. As it will be shown, a maximal rudder rate of around $8\,\deg/s$ to $11\,\deg/s$ is allowed. For the acceleration rate, several values have been tested and implemented. For now, a value of around $10\,\deg/s^2$ has been settled. Indeed, if one supposes that at a certain step time, the rudder rate of the ship is at a maximal value of around $10\,\deg/s$ then in the worst case, 2 seconds later, it could be at $-10\,\deg/s$. An approximation of the maximal rudder acceleration would then be $\gamma \approx \frac{10-(-10)}{2} = 10\,\deg/s^2$.

As mentioned in the Navigation function report, a performance study will be performed in which the tuning parameters of the Guidance algorithms will be tested in order to see their influence on the ship - as it will be for the Navigation and Control functions too. Nevertheless, the following short analysis of the implemented algorithms can be done. This will allow to discuss the influence on the ship trajectory of the choice of the look-ahead distance Δ and the radius switching R_{switch} for example. Let us consider a simulation of 700 seconds duration, with $\Delta = 100\ m$ and $R_{switch} = 50\ m$. This means that the look-ahead distance is quite high; the ship will therefore aim for a relatively distant point on the path trajectory. Also, since the switching radius is also quite high with respect to the distances between the waypoints of this trajectory, one shall expect a turning of the ship before reaching the waypoint. The following plots are the general desired heading produced by the Guidance algorithms:

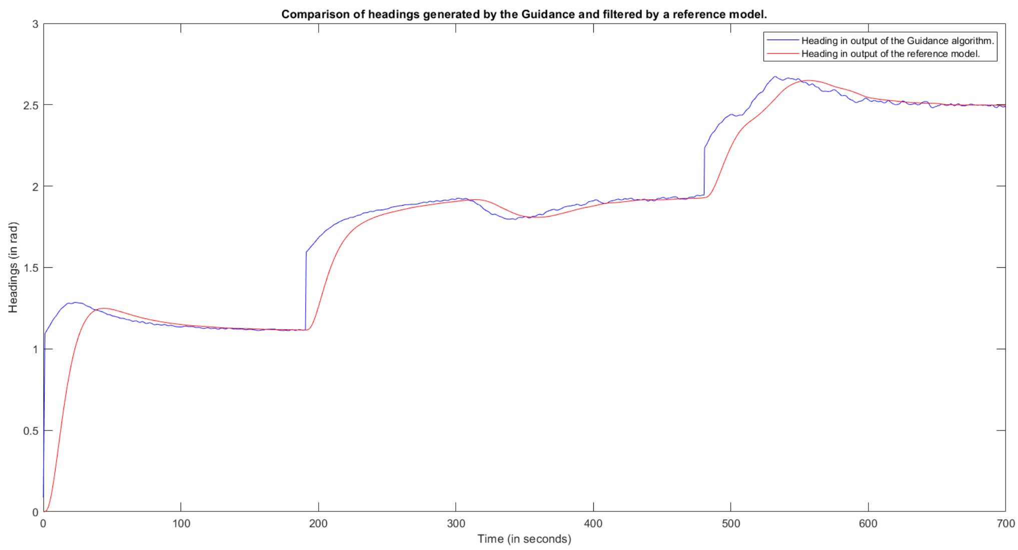

Figure 3.5: Heading from Guidance (in blue) and heading in output of the filter reference model (in red). For $\Delta = 100\ m$ and $R_{switch} = 50\ m$.

One sees that the desired heading is quite well filtered in the output of the I-LOS algorithm. Even if the raw desired heading sometimes exhibits strong oscillatory dynamics, mainly due to wave perturbations and to the activation of the online embedded wave feature estimation around 550 seconds, the filtered heading sent to the Pilot is cleansed. It does not present any significant oscillations that could cause difficulties for the controller.

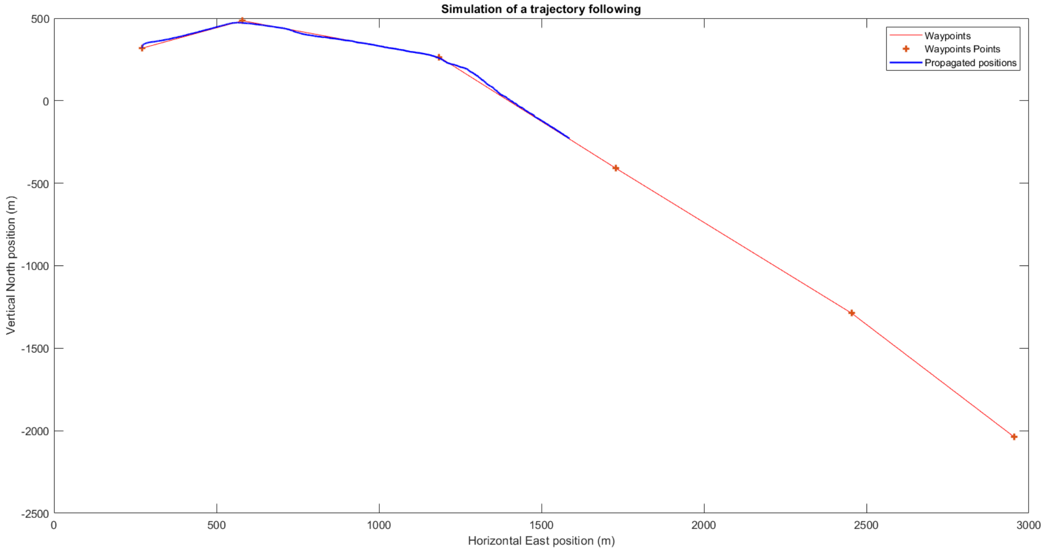

Figure 3.6: General trajectory view. For $\Delta = 100\ m$ and $R_{switch} = 50\ m$.

Since the simulation has been fixed to last 700 seconds, the velocity chosen does not allow the ship to reach its final waypoint, but this was not the goal here. An interesting zone for our comparison is the first waypoints:

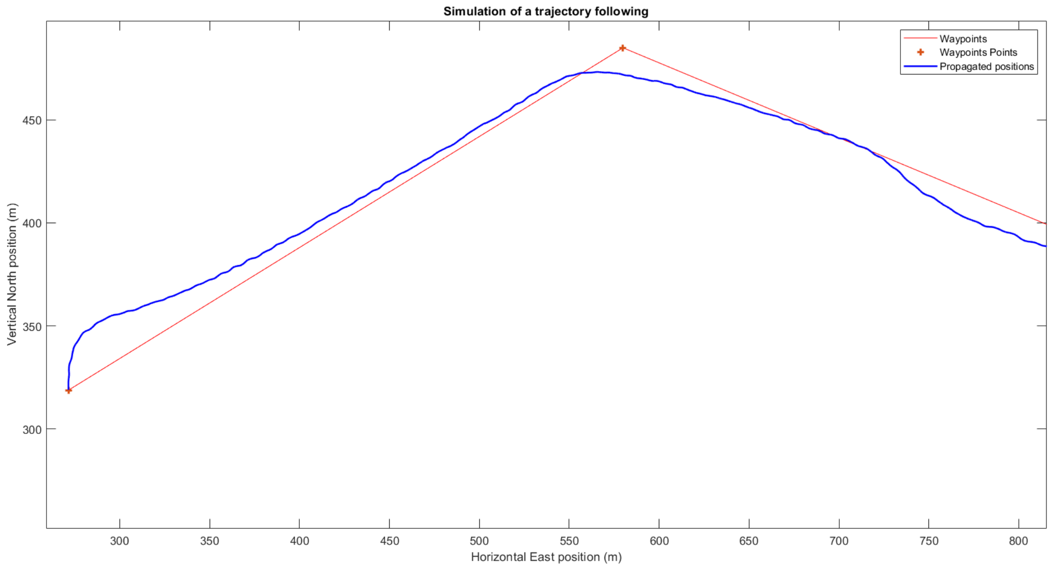

Figure 3.7: Beginning of the trajectory and path-following. For $\Delta = 100\ m$ and $R_{switch} = 50\ m$.

It seems that the vessel has begun its trajectory by aiming a certain distant look-ahead distance and that it went to the following waypoint without any major heading changes. This non-brutal and quite smooth path-following before the first turn can be seen through the desired heading angle from Figure 3.5 in between 50 seconds to around 200 seconds. Then, entering in the 50 meters radius of switching area of the waypoint number 2, its desired heading changed.

The same analysis can be done by fixing this time $\Delta = 30\ m$ and $R_{switch} = 15\ m$. This leads to the following results:

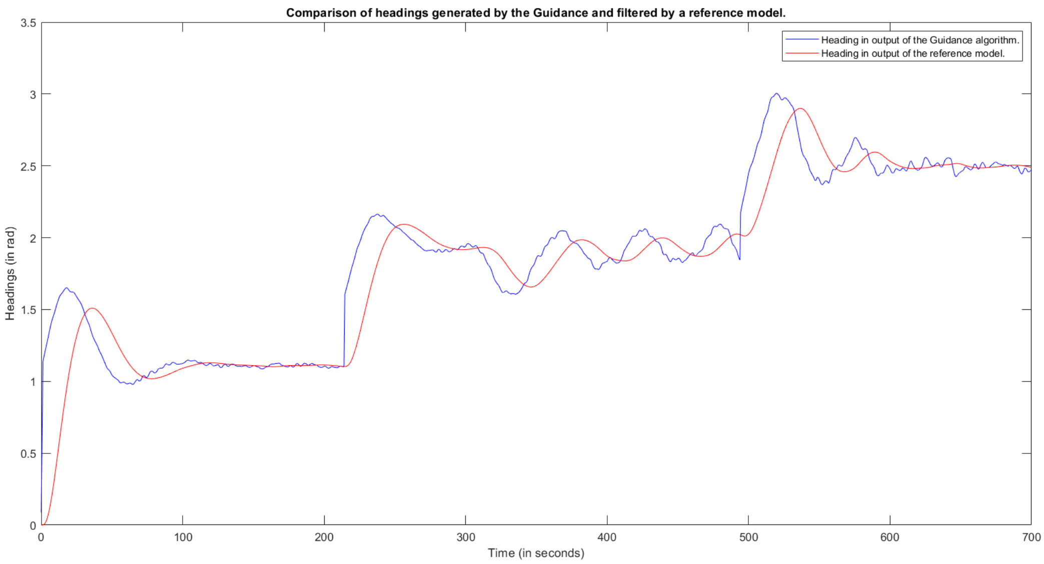

Figure 3.8: Heading from Guidance (in blue) and heading in output of the filter reference model (in red). For $\Delta = 30\ m$ and $R_{switch} = 15\ m$.

One can therefore anticipate that by reducing the look-ahead distance, the vessel will maneuver more in order to control its trajectory in a more effective manner. Since the look-ahead distance is smaller, it will try to stay as close as possible from its predefined trajectory. This is confirmed by the following plot:

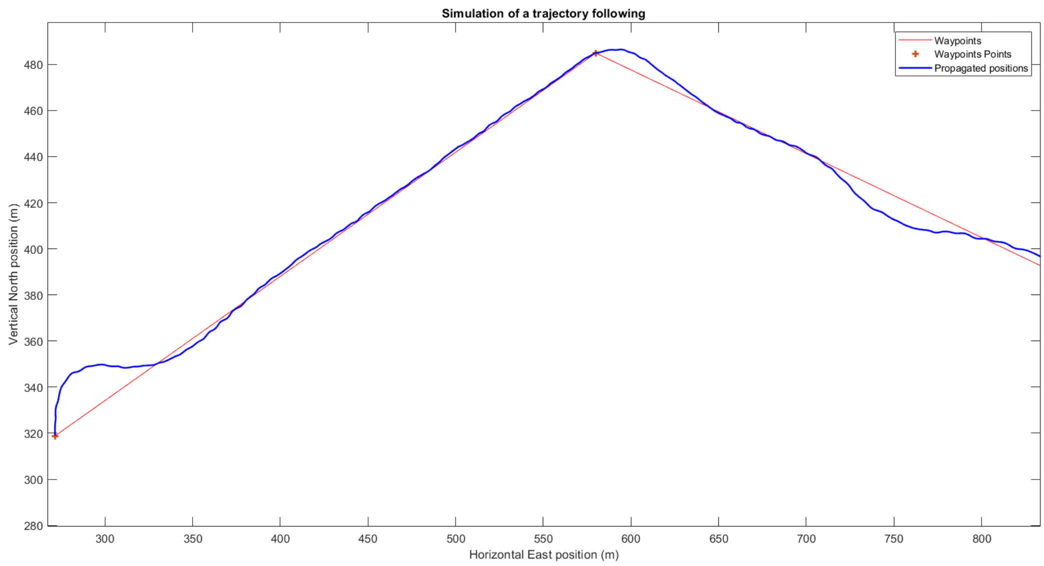

Figure 3.9: Beginning of the trajectory and path-following. For $\Delta = 30\ m$ and $R_{switch} = 15\ m$.

The simulated trajectory in this case is more aligned to the predefined one that in the case of $\Delta = 100\ m$ and $R_{switch} = 50\ m$. However, the control action is much more requested, and the final position difference with respect to the previous case seems to be not so different. A compromise in performance must therefore be chosen in order to tune the algorithm. This study will be done in the generic performance study associated with a specific mission.

Bibliography

[1] T. I. Fossen, "Guidance systems," NTNU, Book and lesson, 2025. [Online]. Available: https://github.com/cybergalactic/FossenHandbook.

[2] G. G. Bousquet, "Gn&c navigation function design," Casteldos-GNC, Technical Note, 2025. [Online]. Available: https://github.com/GerardGrigore/Casteldos_GNC/tree/main/Justification_Files/Navigation_Function_Design.

[3] T. I. Fossen, "Motion control system," NTNU, Book and lesson, 2025. [Online]. Available: https://github.com/cybergalactic/FossenHandbook.