“Virtual Waterway” – a traffic simulation environment for inland and coastal waterways

J. Beschnidt\textsuperscript{1} & E. D. Gilles\textsuperscript{2,3}
\textsuperscript{1}Barco Orthogon AG, Vaihinger Str. 169, 70567 Stuttgart, Germany
\textsuperscript{2}ISR, University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany
\textsuperscript{3}Max-Planck-Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany

Abstract

This paper presents structure, functionality and applications of a virtual navigation environment for the simulation of traffic on inland and coastal waterways. The simulation is focused on realistically modelling the dynamics of vessels in water and on generating typical sensor measurements such as radar or GPS. Simulated objects react on internal behaviour descriptions or external control signals. Due to the ability to simulate a large number of vessels and objects, even complex traffic scenarios can be created. The simulator can be linked to external systems such as radar processing systems, on-board navigation aids, or VTS and coastal surveillance systems, hence allowing one to extensively and safely test and validate their functionalities (e.g. integrated sensor signal processing, data presentation or course control and route planning) under real-time and close-to real-life conditions.

This article concentrates on describing the differential equations used to model the vessel dynamics. Furthermore, the emulation techniques used to generate realistic radar video, including various measurement artefacts, are presented. Finally this article will show setup and results of testing a multi-sensor vessel traffic surveillance system.

Keywords: vessel traffic simulation, sensor emulation, radar.
1 Motivation and objective

Development and installation of complex vessel traffic surveillance systems would be nearly impossible without extensive validation under realistic conditions. Especially, if new technologies or components are introduced to the system, a major part of expensive and time consuming real-life tests can be avoided, if initial testing can be performed using a suitable simulation tool. The simulation environment presented in this paper serves this purpose. It allows the real time simulation of all relevant aspects of modern vessel traffic and provides various means to extensively test and verify the sensor processing functions of the surveillance system under development. For a simulated entity – which can be a vessel, an aircraft or a shore-based station – the entire range of movement data and sensor signals – which in reality would be provided by the respective hardware components – can be generated and passed to the connected system under test for processing. In return, the simulated entity reacts to the control signals issued by the connected system by adjusting its movement and component state, taking into account its own dynamic properties as well as the influence of the navigational environment (Figure 1). Ideally, the system under test is unaware of whether it is processing real or simulated data.

Figure 1: Components of the simulation environment.
2 Structure and functionality of the simulation environment

2.1 Database

The simulator uses a set of digital charts to create its internal representation of the waterway. These charts are derived from the maritime ECDIS standard which has been extended by elements relevant for inland navigation (Inland-ECDIS [1]). In general, Inland ECDIS charts will be maintained and certified by the national waterway authorities, such that they will represent the localities of the waterway with sufficient accuracy. ECDIS charts may comprise a group of rivers or the coastal region of a sea or ocean. They provide the static data on the topography of the waterway, such as river banks, shore lines, bridges, depth and current profiles; furthermore, important navigational data, like the location of the fairway, way points or traffic regulations. The charts are not only used as a database but also to display the navigational environment of the simulated objects (Figure 2).

![Figure 2: Traffic simulation on the river Rhine near Speyer.](image)

Apart from the digital chart, the database of the simulator maintains various libraries providing object and component templates, which are used to create and equip the simulated objects.

2.2 Simulation module

Core of the simulation module is the Administrator, which is maintaining the simulated objects and coordinating the simulation process. The simulation module distinguishes between static and dynamic simulation objects. Static objects are immobile and are controlled by a behaviour description. This group comprises the
shore-based stations along the waterway and the coast line, such as control centers, locks, or signal and surveillance sites (radar site, AIS base station).

*Dynamic objects*, on the contrary, change their state of movement in the course of the simulation. Thus, apart from the behaviour description they require a dynamic model, which describes the movement dynamics and the reaction to control signals. The most prominent representatives of this group are the simulated vessels.

Without user interaction the vessels are guided according to their behaviour description along a typical traffic route. This mode is useful to build traffic scenarios in which vessels operate autonomously. Alternatively, vessels can be controlled manually via a control panel or by an external system, such as a navigation system connected to the simulator. In this case, the movement model is used to calculate the reaction of the vessel to the control signals. The differential equations used to describe the vessel dynamics are derived from the movement model proposed by Nomoto [2] and its adaptations by Neul [3] and Zimmermann [4]. They describe the longitudinal and lateral dynamics of the vessel movement as first order delay relations with nonlinear coefficients:

\[
\frac{d}{dt} \begin{bmatrix} v \\ \omega \\ x_r \\ y_r \\ \psi \end{bmatrix} = \begin{bmatrix} \frac{1}{T_l(v)} (-v + K_l(n_s, n_p)) \\ \frac{1}{T_q(v)} (-\omega + K_q(v) \delta + K_d(v, n_s, n_p)) \\ v \sin \psi + L_l \omega \cos \psi + L_q \omega \sin \psi \\ v \cos \psi - L_l \omega \sin \psi + L_q \omega \cos \psi \\ \omega \end{bmatrix}
\]

(1)

The first row describes the longitudinal dynamics as a dependency of the relative velocity against water, \(v\), from the starboard and portside machine revolutions, \(n_s\) and \(n_p\). The transfer function from revolutions to velocity is defined by a set of nonlinear characteristic curves, \(K_l\), which depends on the specific type of ship. \(T_l\) denotes the speed dependent time constant for the longitudinal movement.

The lateral dynamics (second row) describe the reaction of the vessel to the position of the rudder \(\delta\) and the difference between the starboard and portside machine’s revolutions. Again, sets of nonlinear curves (\(K_q\) and \(K_d\)) are used to account for the properties of different vessel types. \(T_q\) is, similar to \(T_l\), the velocity dependent time constant of the lateral movement.

The equations for \(\dot{x}_r\) and \(\dot{y}_r\) express the relative movement of the vessel with regard to longitudinal velocity \(v\), heading angle \(\psi\) and rate of turn \(\omega\). \(x_r\) and \(y_r\) are the coordinates in a global cartesian coordinate system. Since the rotational movement occurs around the center of gravity of the vessel, \(L_l\) and \(L_q\) denote the distance of the arbitrarily chosen ship reference point to the actual centroid.

The model parameters for various vessels have been identified during several trials, results can be found for instance in [3] and [4].

Modelling the vessel dynamics using a first order delay relation represents of course a simplification of reality. Studies in [4] however have shown, that the parameter values of higher order models are hard to identify and that the effects of
the resulting higher accuracy of the model on the system behaviour lie in the same order of magnitude as the disturbances, that affect the system from outside.

For certain applications, the model can be further simplified. Under the assumption of cruising with constant velocity, the nonlinear terms in \( \dot{\omega} \) are eliminated such that the equation for the relative velocity can be reduced to \( \dot{\nu} = 0 \).

In addition to the inertial dynamics of the vessel, the influence of the navigation environment has to be considered. Therefore, the weighted vectors of local water and wind currents, \( w_d \ast (\dot{x}_d, \dot{y}_d)(x,y) \) and \( w_w \ast (\dot{x}_w, \dot{y}_w)(x,y) \), are added to the relative motion of the vessel:

\[
\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} x_r \\ y_r \end{bmatrix} + w_d \ast \frac{d}{dt} \begin{bmatrix} x_d \\ y_d \end{bmatrix}(x,y) + w_w \ast \frac{d}{dt} \begin{bmatrix} x_w \\ y_w \end{bmatrix}(x,y)
\]

(2)

The weights \( w_d \) and \( w_w \) hereby depend on the shape of the under- and over-water parts of the vessel’s hull. The true current profiles depend on a range of factors, such as height and depth contours and shore line structures, and can presently not be quantified. Apart from using public current data from external sources, the water current profile of rivers can be simulated by assuming a flow parallel to the river axis. The local magnitude of the current velocity can be interpolated for a given river based on the established river coordinate system (river kilometer) and known local measurements. The value of the magnitude will have its maximum inside the fairway and will decrease towards each river bank and various obstacles, such as buoys and shore line constructions.

2.3 Radar emulation

Output of the model-based computation is an update to the vessel’s dynamic state, comprising exact values for position, velocity and rotational movement of the vessel in the simulation.

In practice, these values can only be obtained using measurements. Each measurement introduces an error to the measured state value, thus, the true value can only be obtained in approximation. A typical sensor processing system would use methods like averaging or Kalman filtering to cope with the stochastic properties of the measurements.

To validate the sensor processing functions of such a system under realistic conditions, the simulation has to closely emulate the properties of sensors, including their error and failure behaviours.

To facilitate such tests, the simulation environment provides a set of common navigation sensors, which can be equipped to the simulated vessels:

- Radar
- GPS (Standard, PDGPS, CPDGPS, Angular-GPS)
- Rudder position indicator
- Rate gyro (rate of turn)
- Gyro compass (heading)
- Log (Velocity relative to water or over ground)
- Depth sounder (depth below keel)
- Ultrasonic distance sensor (precision distance meter)
- Laser scanner distance sensor (precision distance meter)
- AIS transponder (position and voyage data of nearby vessels)

Among this list, the navigation radar is currently the most important marine surveillance sensor. Opposed to most other sensors it does not deliver a scalar but a two dimensional, image like measurement, making the emulation of a radar sensor particularly complex.

The antenna of a typical navigation radar is rotating in one to six seconds around its vertical axis, emitting periodic pulses at an interval of down to 100 µs. A target illuminated by the antenna beam will return an echo. The time \( \Delta t_l \) from transmitting the radar pulse to the reception of the first echo return is proportional to the distance \( r_{obj} \) between the reflecting object and the antenna:

\[
\Delta t_l = \frac{2 r_{obj}}{c_{air}} \tag{3}
\]

By measuring the delay of the return echo, knowing the speed of light \( c_{air} \) and considering that the wave must travel forth and back, the distance of the target can be computed.

The radial extension of the radar echo \( l_e \), called range resolution, can be computed in a similar fashion from the pulse duration or pulse width \( \Delta t_i \):

\[
l_e = \frac{c_{air} \Delta t_i (r_{max})}{2} \tag{4}
\]

The pulse width is usually fixed or limited to a small set of values, the value itself depends on the desired maximum detection range \( r_{max} \). On inland vessels a typical radar pulse has a duration of 50 ns, resulting in an echo length of about 7.5 m. The radar of a coastal vessel usually has to look farther (30 and more kilometers), thus the radar pulse may last as long as 1200 ns, which results in an echo length of 180 m and more.

The angular resolution of a conventional radar is equal to the horizontal aperture of the antenna beam, which, in turn, is related to the linear dimension of the antenna and to the signal wavelength. For an antenna having the linear dimension \( l_a \) and for an operating wavelength \( \lambda \), the beam aperture can be approximated by

\[
\Theta = \frac{1.2 \lambda}{l_a} \tag{5}
\]

Navigation radars typically deliver the video and synchronization data using four signals: Trigger, Video, Heading (ARP) and Sync (ACP). The trigger pulse marks the start of the transmission of a radar beam. The video signal contains the intensity progression of the received radar echo. Together with the corresponding trigger pulse the video signal informs on distance and reflectivity of the radar targets. The heading pulse marks the north direction of the antenna. The synchronization signal is a signal generated synchronously to the rotation of the antenna.
In combination with the heading pulse it serves as an indication of the current orientation of the radar antenna.

The radar simulation tries to emulate the described properties of the radar sensor as closely as possible. The beam pattern is generated using a modified ray tracing algorithm. To assure real time behaviour even under heavy load conditions, two optimising steps are performed to speed up the ray tracing. First, all radar reflecting objects are selected from the digital chart. The coordinates of these radar objects are transformed into the local coordinate system of the antenna (Figure 3, left). Since these coordinates are changing only marginally for slow moving objects, such as vessels, they are cached in order to be used for multiple consecutive radar images.

A further segmentation step fractionises the radar image into narrow sectors \((\alpha_{min}, \alpha_{max})\). Radar objects are reduced to their polygonal parts located inside the sector (Figure 3, right). During the ray tracing process, only those objects are considered, which have a polygon segment inside the sector currently probed by the radar beam. For each relevant intersection, a radar echo will be generated. Its length and intensity depends on the reflectivity and direction of the target surface and the properties of the radar ray at this distance from the antenna. The generated echoes are collected in rays and provided as a digital polar radar image to the external sensor processing system.

Navigation radars are rather inaccurate sensors and exhibit a range of unwanted effects and mismeasurements when used in practice. Apart from the properties inherent in the physical principle of scanning with focussed electromagnetic beams, such as beam aperture, reflectivity dependence, and finite duration of the radar pulse, radar applications have to cope with a series of disturbances and artefacts in the radar image, which complicate the radar processing procedure. The most notorious of these effects are

- clutter
- attenuation by rain
- multipath reflections
- side lobe effects
- multi radar interference
- refraction and superrefraction
This non-ideal behaviour has to be emulated likewise by the simulated radar sensor. Due to the ray tracing algorithm, which probes the simulated environment in a manner similar to an actual radar beam, most of these effects are implemented in a straightforward way, re-enacting the same physical process.

As an example, multipath reflections, which occur if the beam is mirrored between objects of high reflectivity, such as smooth metal surfaces, are modelled by following the beam beyond the initial reflection and creating additional echoes if a suitable surface in the new direction is hit. Figure 4 shows a recorded radar image of a traffic scene on the Rhine on the left and the corresponding simulated situation on the right. The scene contains both multipath echoes (M) and wave clutter (C).

Figure 4: Original and synthetic radar image with multipath echoes.

The directional radiation pattern of the radar beam, which causes a broadening of the radar lobe with increasing distance from the antenna as well as a pattern of smaller side lobes, causes objects to create an echo even if they lie outside the main lobe. This results in the broadening of the radar echo of small objects, the introduction of side lobe echoes and the merge of the echoes of nearby objects. The simulation emulates this behaviour by considering the lobe aperture of the radar beam.

Another typical nuisance during practical radar application is clutter. Clutter is typically caused by waves, rain or unstructured objects on land. Wave clutter, as an example, appears as a group of small undulating echoes with short life span, which causes false tracks and in general increases the time needed to process the radar video and to identify real targets. On the other hand, areas with rain produce a mid-level return for the entire affected region, making it harder to detect actual objects inside the region. Within the simulation, clutter objects are created and maintained using a dedicated clutter generator. During radar generation, they are ray traced in the same manner as normal radar objects. The echo intensity is chosen according to the specific clutter properties.
Accounting for the non-ideal radar properties does not impair the real time behaviour of the radar video generation. The average time needed to compute a complete image of 10 km range and about 300 radar objects of variable size on standard PC hardware (1 GHz COTS) amounts to less than 100 ms. Using this setup, up to 20 radar sensors, each rotating with 2 seconds per revolution, could be simulated in parallel.

The computation time can be further reduced for stationary radar sensors (radar site). Here, the location of the echoes of all immobile objects, like shore lines or bridges, can be precomputed. Figure 5 shows a fictive coastal radar station in the Finnish Archipelago. The radar generator scans a 100 km range with about 2,800 small islands in about 8 ms per revolution.

![Figure 5: Synthetic radar image of a coastal radar station.](image)

### 3 Applications

These performance reserves allow the simulator to be used to validate multi-sensor processing systems. In a typical vessel traffic surveillance system, as shown in Figure 6, multiple radar sensors, possibly in combination with information provided by RDF and AIS sensors, will monitor the area of interest, leading to multi coverage of certain regions. A state-of-the-art sensor processing system must be able to combine all available information using actual sensor data fusion, without producing false or duplicate data. The performance of such a system can be validated before installation only when using a suitable simulation tool, that is able to recreate a realistic multi-sensor environment.

As an example, the new coastal surveillance system for Estonia has been tested and validated continuously during the development phase using the simulation environment. The system includes a total of 20 radar sites, covering the entire coastline of Estonia. Four regional operation centers are connected to one national control center in the capital Tallinn. Each regional center runs a multi-sensor processing system, that collects and fuses the radar data of up to seven radar sites. After
having been thoroughly tested in simulation, the system is now in the process of installation and has already successfully passed a series of field trials, which have proven its suitability for intended task [5].

References


