Optimizing Control at Sea: The Experience of the Seapacer Project

Thomas Hellström

UMINF 02.08
ISSN-0348-0542
Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden
thomash@cs.umu.se

September 20, 2002

Abstract

This paper discusses the experiences of a challenging industrial research and development project: Design and implementation of a fuel saving control system for large vessels. The developed hardware and software has been successfully installed on around 20 ferries around Europe and has reduced the fuel consumption by as much as 5-15% in many cases. The fuel saving is achieved by optimizing control at three levels ranging from low level propeller and main engine control up to route planning for optimal speed profiles compensated for varying depth and weather conditions. The control problems involve classical control functions as well as numerical optimization. Other important issues that are discussed in the paper are the safety aspects in designing and building semi-autonomous control systems for large vehicles.

1 Introduction

The fuel costs are the second largest item (after salaries) on a big vessel's budget. The fuel consumption for a large ferry ranges between 1000 and 5000 liters per hour. This means that the ship consumes more oil per hour than a one-family house does for one whole year's heating (in northern Sweden). The annual fuel budget for a ferry running 20 hours per day is in the order of millions of dollars. Even small reductions of a few percent off the fuel consumption means considerable annual savings.

Seapacer is an integrated system for fuel saving and top-level control of a ship's performance. The system is operated from the central unit placed on the bridge (see Figure 1). The operator, normally the ship's First Officer or Captain, inputs the required values for speed, arrival times, and complete route plans from the keyboard. The main engines (10-40.000 horsepower) and propellers are then automatically adjusted, to reach and maintain the required speed at the lowest fuel consumption. Fuel saving is typically 5-10%, corresponding to at least 1 cubic meter of heavy fuel oil per day. The saving is achieved by optimizing control at three levels:

1. Pitch optimization. The pitch angle of the blades on a controllable propeller acts as a kind of gear box, and affects the ship's speed together with the main engine's revolutions (rpm). The optimal combination of pitch/rpm depends on a number of external and time-varying conditions, and therefore must be subjected to dynamic optimization to be optimal.

2. Dynamic control of speed to avoid sudden peaks in fuel consumption caused by low water depth, or unanticipated changes of the weather conditions.

3. Route planning. The fuel consumption for a ship depends not only on speed, but also on water depth and weather conditions. The optimal speed distribution along the route can be computed in advance, if weather forecast is available.

Figure 1: The Seapacer central unit placed on the bridge of the vessel. The left screen is used for real-time control tasks such as speed settings while the right screen is used for long term voyage analysis and follow-up.

The developed hardware and software have been successfully installed on around 20 ferries across northern Europe. The most recent installation was on MS Gotland (Gotlandsbolagen) in 1997. This paper discusses the experiences of the research and development of the Seapacer system. Section 2 describes the design and basic operation of the Seapacer system. The three levels of optimizing control described above are covered in more detail in Sections 3,4 and 5. Section 6 discusses general experiences and difficulties encountered during the project.

2 The Seapacer System

The basic layout of the Seapacer system is shown in Figure 2. Seapacer takes over control from the maneuvering handles of the main engine's speed (revolutions per second) and of the propeller pitch . The most important components are shown in the figure. More information is found in Sections 2.2, 2.3 and 2.4.

2.1 Basic Modes of Operation

From the user's perspective, Seapacer is a tool for high-level control of the vessel's speed and fuel consumption. The main functionality can be described as a number of control systems aiming at obtaining and keeping set values for speed, fuel consumption or arrival time. During operation, any of the following operational modes may be selected:

- SPEED Fixed speed.
- L/NM Fixed fuel consumption per nautical mile (liters/nautical mile).
- L/H Fixed fuel consumption per hour (liters/hour).
- POWER Fixed power from engine.
- POS/ARR Adapt speed to planned arrival and waypoint defined by latitude and longitude.
- DIST/ARR Adapt speed to planned arrival and distance.
- ROUTE Describe which mode to use at what time during the voyage. Seapacer does the alternation between modes automatically. This mode is described in detail in Section 5.

Besides the pure control functions, the system contains extensive support for data logging and analysis. Various types of lists and charts can be produced, enabling for instance, long-term analysis, comparative evaluations before and after a reconstruction, maintenance at the shipyard,

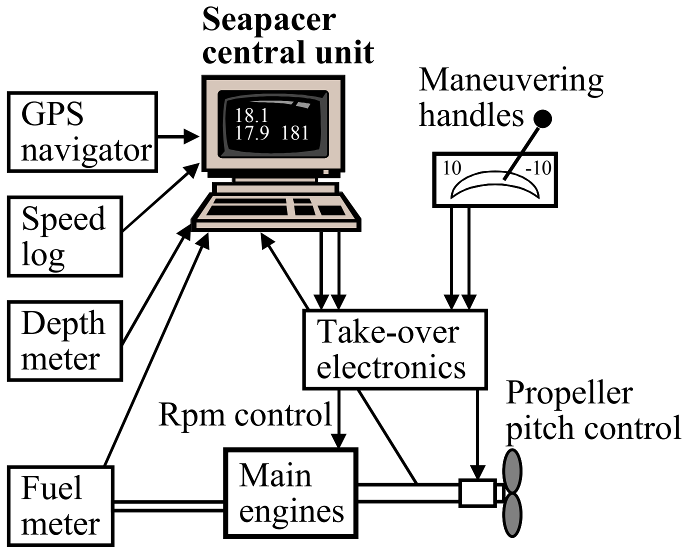

Figure 2: Block diagram of a basic Seapacer system. Seapacer takes over control from the maneuvering handles and controls the main engine speed and propeller pitch.

etc. Examples can be seen in Figure 3 and 4. In Figure 3, data from a large number of voyages is presented. The fuel consumption (liters/nautical mile) is shown as a function of the water track speed. Each dot represents the mean values for one voyage. The two clusters correspond to day and night trips, each having a different number of active main engines. The diagram in Figure 4 shows the relation between speed, fuel consumption, and wind strength for a two-month period. Diagrams like these ones aid in the crew's constant effort to minimize fuel consumption by route planning and scheduling of timetables.

For educational purposes, a simulation function is provided inside Seapacer. It is built up around similar parameters to those of the vessel in question. More information about the functionality of the Seapacer system can be found in the User's Guide: *Seapacer Optimizing System Mark II* [3].

2.2 Inputs

Seapacer interfaces with a large number of sensors and sub-systems on the vessel. The types of inputs vary from ship to another, and the software has to be easy-to-configure. Many sensors supply digital outputs with a frequency or pulse length proportional to the measured quantity. Other, more modern equipment, output data in ASCII format via an RS232 connection. Quite often, signals are already used for other purposes. It is therefore essential to connect Seapacer so that these other functions on the ship are not affected. All inputs are galvanically isolated from the computer (the voltage differences can be more than 200 volts between two different ground points). The following signals are normally connected to the Seapacer system:

- Ship's speed over ground (Bottom track speed)
- Ship's speed through water (Water track speed)
- Fuel consumption
- Propeller revs (revolutions per second)
- Main engine power (horsepower)
- Water depth below keel
- GPS navigator. Outputs position and optionally bottom track speed
- Trim

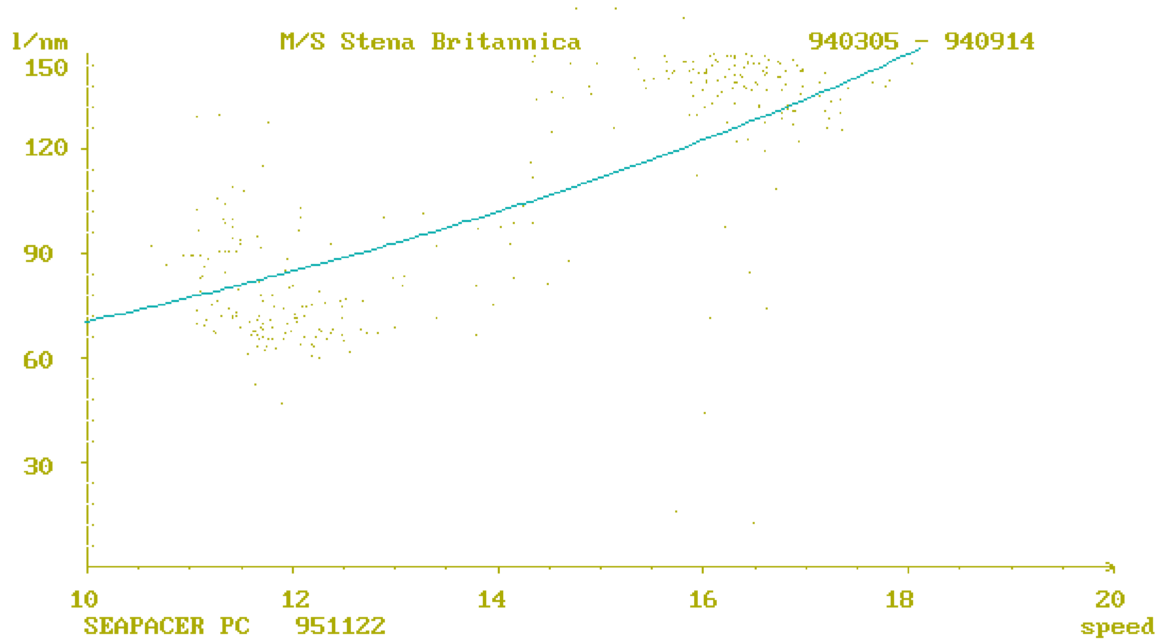

Figure 3: Off-line analysis of fuel consumption versus water track speed. Each dot corresponds to one voayage. The two clusters correspond to different numbers of main engines.

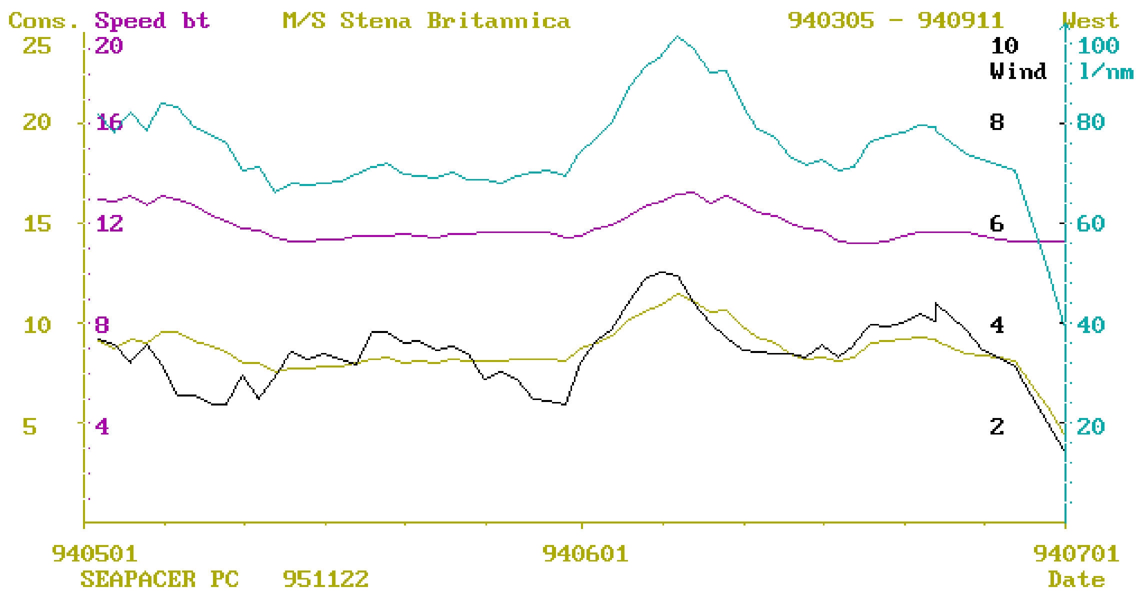

Figure 4: Diagram showing the relation between speed, fuel consumption, and wind strength (Beaufort). Each dot represents one night trip.

- Number of engaged main engines
- Shaft generator engaged (starboard and port)
- Handle position

Many of the signals are highly noisy and also give off completely incorrect signals from time to time. This has to be handled in a stable manner by the software by filtering and outlier detection. Sensor fusion is also utilized for the estimation of bottom track speed. The primary source for speed is the Doppler log, which measures the echoes of ultrasound pulses against the bottom. This normally works well, but can sometimes give off incorrect values due to false echoes or too large water depth. A differential GPS navigator provides an alternative speed source. In older systems, this signal is often updated too slowly or too much delayed to be useful as input in the actual control system. However, the GPS speed is useful as a backup for the speed log, if and when the speed log fails. Likewise, the speed log is used as complement to the navigator. The navigator sometimes loses the signal from the satellites. The speed log is then, in combination with the last estimate of the ship's course, used for dead reckoning to update the estimate of the ship's position.

2.3 Outputs

Seapacer is integrated into the vessel's existing control system by galvanically isolated 4-20 mA outputs (separate for starboard and port sides):

- Main engine revs control. Connected so that a zero signal corresponds to idle speed of the vessel (not zero revs). This is important for safety reasons, since a disconnected or failing system should not cause the engines to come to a complete stop.
- Propeller pitch control. The pitch of the propeller is the angle of the propeller blades, and acts as a gearbox for the propulsion of the vessel. Zero angle results in zero propulsion, even with the engines running at a high speed. The direction of the blades makes it possible for the vessel to move forwards or backwards without changing the rotation direction of the propellers. The way to connect Seapacer's control of the pitch is therefore very important, not least for safety reasons. The normal way to do it is by interfacing to an existing system, the load controller, responsible for the low-level control of the pitch. The load controller can be controlled in such a way that Seapacer only gets access to reduce the pitch from the current set point, down to around 80%. In this way the safety issues are left with the design of the load controller, and Seapacer has no possibility to output fatal commands.
- Takeover. Relay contacts, which close when Seapacer should take over rpm and pitch control from the manual maneuvering handles. The way this function is implemented is extremely important, since a failure in operation could cause the vessel to be completely inoperable. Also, the Takeover module must have intelligence to handle some of the safety issues described in the next Section.

2.4 Safety Issues

A typical vessel with Seapacer installed has a main engine power of 10-40.000 horsepower. The vessel itself may be well over 100 meters long, and has a considerable inertia, which makes the control problem both difficult and important. For safety reasons, Seapacer must be installed in such a way that it can be bypassed by the operators at any time. This bypassing functionality has to be implemented at many levels, and designed in such a manner that the takeover procedure is totally intuitive for the operator, even under mental stress. Control may be transferred back to the operator handles by any of the following procedures:

- Issuing a *Handle* command on the Seapacer keypad. This is the normal way to transfer control from Seapacer to the operator. However, it cannot be expected to be used by a stressed operator in a situation of emergency.
- Turning a designated *Seapacer In Control* switch to the OFF position. This disconnects Seapacer from the ship control functions electrically and is used as a safety precaution, e.g. when servicing Seapacer.

- Moving the maneuvering handles of the vessel below a certain set point. This is the most natural way for an operator at sea. In a situation of emergency, the speed of the vessel has to be reduced drastically in most cases, and the normal way to do this is to pull the maneuvering handles to zero or even to full backward speed (a.k.a. crash stop). By sensing the maneuvering handles, Seapacer can automatically disconnect itself and transfer all control back to the operator.

In addition to these manual ways of transferring control from Seapacer to the maneuvering handles of the vessel, the system automatically transfers control by two functions:

- Watch dog. An electrical timer function that automatically transfers control back to the handles if the timer is not periodically (e.g. every second) reset by the Seapacer software. This ensures that a computer error (software or hardware) does not cause the system to hang or issue unpredictable control signals to the main engines or propellers.
- Power failure. The takeover electronics automatically falls back to manual control in the case of a power failure.

Another related issue to bear in mind when designing a high-level control system such as Seapacer, is how to disconnect the entire system when the vessel is serviced or repaired. There should always be a simple means, by which the entire installation can be removed, leaving the ship in a fully operational mode. Just as the potential fuel saving is attractive to the ship owner, so an interruption in the operation of the ship is totally unacceptable.

3 Pitch Optimization

The pitch angle of the blades on a controllable propeller acts as a gearbox, and controls the ship's speed along with the main engine's revolutions (rpm). The optimal combination of pitch/rpm depends on a number of external conditions, and therefore must be subjected to dynamic optimization to be optimal. Seapacer minimizes the consumption of fuel by maintaining an optimal ratio between the propeller's pitch and the speed of rotation. The optimization aims at minimizing the fuel consumption, measured as consumed oil per nautical mile, for a given set speed s_{set}. The directly measurable entities are water track speed s_{wt} (nautical miles per hour) and fuel consumption c (liters per hour). Both s_{wt} and c are functions of the pitch p and main engine revs r. Hence, the pitch optimizer tries to solve

$$(r_{opt}, p_{opt}) = \arg\min \frac{c(r,p)}{s_{wt}(r,p)} \tag{1}$$

with the constraint

$$s_{wt}(r,p) = s_{set} \tag{2}$$

where s_{set} is the set speed for the vessel. s_{set} is given explicitly by the operator if running in SPEED MODE or implicitly if running in POS/ARR or DIST/ARR mode (see Section 4.3 for details). The optimization problem has to be solved in real-time with both c and s_{wt} being extremely noisy. Furthermore, the time constants involved in the processes generating c and s_{wt} are large. This means that a change in r or p not immediately causes a measurable change in neither c nor s_{wt}. By the time c and s_{wt} respond, the process may very well have a new characteristic, i.e. the optimal values (r_{opt}, p_{opt}) may have changed. Altogether the optimization problem is indeed very hard. The implemented solution uses a proprietary algorithm that first controls r and p such that constraint 2 is fulfilled. In the next stage, r and p are moved in one direction until a local min value for $c(r,p) \div s_{wt}(r,p)$ along this direction has been detected. The step sizes for r and p are set so the reduction in engines revs r is approximately balanced by the change in pitch p. In this way the constraint 2 is approximately fulfilled during the search operation. If necessary, r is finally adjusted so the constraint is not violated. The system then waits, either a predefined period of time, or until a detection algorithm signals that a new search may be fruitful. The search direction is now reversed.

The algorithm has worked well, but needs steady and fast responding fuel signals to be meaningful. This is seldom the case with ordinary fuel meters installed on the ship for ordinary purposes.

4 Dynamic Control

Seapacer's basic functions are a set of controllers for speed, fuel consumption liters/hour, fuel consumption liters/nautical mile, and shaft power. These controllers may be used as such by issuing set points from the keyboard. One example can be seen in Figure 5, where the user has issued a SET SPEED command that governs the main engines so that the bottom track speed maintains 17.8 knots. The controllers are ordinary PID controllers, which control a linearized version of the physical entity to be controlled. For example, to control the speed s of the vessel, a model f for the static dependency between r, issued rpm (main engines revs), and s, is utilized. The relation is given by $r = f(s)$ where the function f is approximated from sampled data and linear interpolation. Different functions have to be used for different numbers of engaged main engines. The speed controller acts on the f entity:

$$r = k_p E + k_i \int E \partial t + k_d \frac{\partial E}{\partial t} \tag{3}$$

with the control error E defined as

$$E = f(s_{set}) - f(s_{act}) \tag{4}$$

where s_{set} is the commanded set speed, and s_{act} is the ship's actual speed. In practice, the derivative part is not used, i.e.: $k_d = 0$ in most cases.

The control of the main engines has to be done in a gentle way to avoid unnecessary rapid thermal changes. Of course, this can be adhered to in the tuning of the PID controllers, but other functions have also been added. It is possible to limit the speed, by which the controllers are allowed to change the main engine revs (i.e. the unit for the limit is revs/sec^2). This causes the main engines to operate more smoothly than when run manually.

4.1 Handling Fuel and Power Limits

Figure 5 also illustrates some additional control functions in the system. In the example, the user has entered a fuel consumption limit of 180 l/nm (liters per nautical mile). This is treated as a constraint in the control algorithm, and has a higher priority than the set speed, which is the actual control entity. In the same manner, a lower (2000 kW) and a higher (19000 kW) power constraint was entered. The fuel limit and upper power limit serve as safeguards against temporary and unanticipated increases in the load, caused by changed weather conditions, or low water depth below the keel. The lower power limit is necessary to ensure acceptable working conditions for the main engines. In the control system, the constraints are handled as penalties by modifying the control error E to reflect the violated constraint, for example, excessive fuel consumption. The modification is done by a model function g that relates the constraint entity to the control entity (normally the speed). In speed mode with a fuel consumption limit, a function $s = g(c)$ is used. c denotes the fuel consumption and s denotes the speed that approximately corresponds to c. Like the function f in the previous section, the function g is approximated from sampled data and linearly interpolated. Assuming a set fuel consumption limit $c_{\max}$ and a sampled fuel consumption c_{act}, the control error E is now computed as

$$E = \begin{cases} f(g(c_{act})) - f(g(c_{\max})) & : \text{ if } c_{act} > c_{\max} \\ f(s_{set}) - f(s_{act}) & : \text{ otherwise.} \end{cases} \tag{5}$$

In practice, the sharp switch point in the definition of E is smoothed so the penalty starts to work already before the limit is violated. The effect of the modified control error is that too high a fuel consumption (i.e.: $c_{act} > c_{\max}$) makes the controller act as if the speed is too high, even if $s_{set} > s_{act}$. Since expressions 4 and 5 only compute differences of the functions f and g, the absolute values for these models are not critical. The purpose of using them is to linearize the control entities so the apparent process for the PID controller is more linear and easier to control.

4.2 Dynamic Set Points for Fuel and Power Limits

The fuel and power limits described in the previous section serve as safeguards against temporary increases in the load. The result of such an increase, e.g. caused by shallow waters, is a slowing

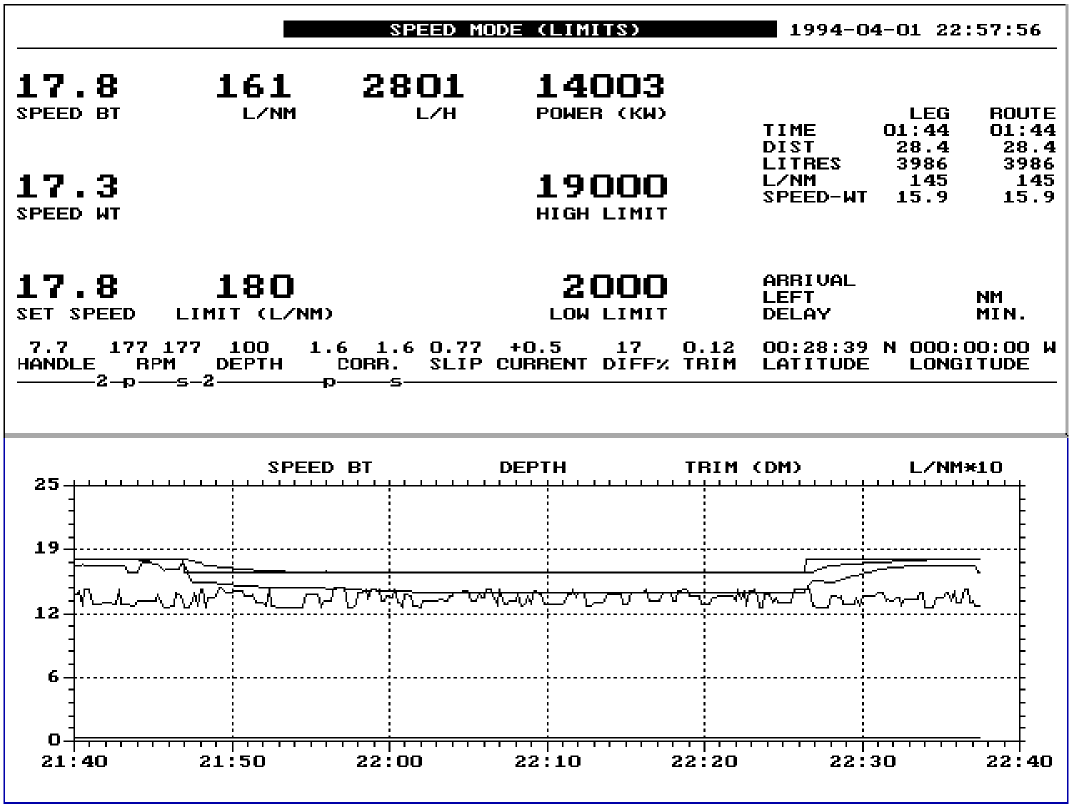

Figure 5: Seapacer screen showing real-time values for all connected sensors and given commands. The diagram part is updated in real-time.

down of the vessel to reduce the fuel consumption or power below the set upper limit. This can be a very useful function as such, provided the settings of the limits are done carefully. However, too hard limits prohibit Seapacer from keeping the arrival times, while too loose limits are without effect. To eliminate the need for manual choice of limits, a dynamic computation of suitable values has been developed. It works by slowly lowering the upper limit until the limit almost becomes active, i.e. when the actual fuel consumption, or the main engine power necessary to maintain the speed, almost reaches the dynamically set upper limit. In this way a sudden increase in load causes the limit to be violated and Seapacer to lower the speed. However, after a pre-defined delay time, the dynamically set limit is slowly adapted upwards, to allow the vessel to run at the necessary speed in the long run. The result of the dynamic set points for fuel and power limits is that of evening out the power over the entire route. This results in lower total fuel consumption.

4.3 Automatic Computation of Speed Set Points

The main objective for the crew of a ferry is normally to keep the set arrival times. For this purpose, a function that dynamically designates the SET SPEED of the speed controller, is implemented in the DIST/ARR running mode. The user enters the arrival time and distance to the goal. The vessel now runs at the lowest speed possible, while still arriving in time. The speed necessary to travel the distance is updated continuously. No updating takes place for the five last minutes ahead of estimated arrival time. When 0.5 NM remain of the stated distance, Seapacer automatically changes control-mode to SPEED MODE, using the last set speed as the new set speed. The distance is computed by integrating the log signal (speed relative to ground). A similar run mode POS/ARR works by using a given geographical location (waypoint), instead of a certain distance to travel. In this way the system is more tolerant for deviations from the originally intended routes than in DIST/ARR mode. A more complete handling of entire routes is implemented in the ROUTE PLANNING run mode described in more detail in Section 5.

5 Route Planning

The route planning of a ship with varying speed in different parts of the route is designed to keep the set arrival time, while reducing the total fuel consumption. Seapacer automatically optimizes the speed distribution between the route legs. Legs with different depths and/or weather conditions then run at different speeds to minimize the total fuel consumption.

5.1 What Affects the Fuel Consumption

Following are some of the most important factors that affect the fuel consumption of a ship:

- Ship-specific parameters such as form of the hull, weight, type of main engines, propellers, etc.
- Number of engaged main engines.
- The ship speed relative to the ground, measured in knots (denoted "bottom track speed").
- Water currents (direction and speed in knots).
- Water depth under the keel.
- The ship's draft (depends on the cargo).
- Wind and waves (direction and strength measured in Beaufort points).

5.2 How Can the Fuel Consumption Be Reduced?

Route planning consists of varying the ship's speed in different parts of a route. Since the external conditions (wind, current, and depth) vary, it is evident that fuel consumption cannot be maintained at a minimum, if a constant speed is kept throughout the route. Therefore, we get a minimization problem that has to be solved numerically: we have to find the speed distribution that minimizes the total fuel consumption within the constraint of keeping the scheduled arrival time. Route planning of some kind or another is done on all ships. Most often the "calculation" consists of manual estimates, based on previous experience from the same route. The Seapacer system contains functions for automatic route planning. Wind, current, and water depth can be input by the operator before departure or during the voyage. Seapacer then automatically calculates a speed profile that minimizes the total fuel consumption. Based on the computed speed profile, Seapacer regulates the ship's speed by controlling the main engines and the propellers. The arrival time is kept without unnecessary margins. Following is a description of the basic route planning system of Seapacer.

5.3 Models

The dependency of fuel consumption upon speed, wind, and water depth is essential for the route planning and optimization. Analytical models are rare and are not general enough to be used for all sorts of ships. Therefore, data is sampled at different running conditions, and gathered in tables. These tables serve as models for the optimization and route planning. The values shown in the following tables are examples from a ferry running between Hook van Holland and Harwich on the English Channel. The sampled values do vary from one ship to another, but they normally share the same characteristics. Values in-between points are estimated by a 1- or 2-dimensional linear interpolation.

5.3.1 Speed Models $F(x_w)$

Table 1 shows fuel consumption $Cons$ (liters/hour) for different speeds through water x_w (knots). The data has been sampled with no influence from limiting water depth. The function $F(x_w)$ is defined as linear interpolation in the table.

x_w	Cons
10.4	650
13.2	875
17	1300
20.1	2120
20.7	2900

Table 1: Fuel consumption (liters/hour) at different speeds

5.3.2 Depth Model $D(x, d)$

In limiting water depths the ship's speed decreases due to the increased water resistance. In very shallow waters (typically a few meters below the keel), the so-called "squat effect" pulls the ship downwards, thereby reducing its speed further. Table 2 shows the fuel consumption increase (%) for different water depths d and speeds x. The function $D(x, d)$ is defined as a 2-dimensional linear interpolation in the table.

x	d	Increase (%)
10	8	5
10	10	3
10	100	0
17	8	20
17	15	10
17	100	0
20	8	30
20	15	20
20	100	0

Table 2: Increase in fuel consumption (%) due to limited water depth at different speeds

5.3.3 Wind Model $W(w, w_d)$

The wind is specified by the direction relative to the ship's heading and the wind strength, expressed in Beaufort points. The parameter w_d is the wind direction (degrees). w is the Beaufort degree (0-10). Generally speaking, wind from the side and even slightly from behind increases the fuel consumption due to the large open surface on the sides of many ships. Table 3 describes a typical and approximate relation between increased wind strength, direction, and increased fuel consumption for each unit of Beaufort. The wind direction is measured clockwise, relative to the ship, with zero degrees defined as a wind blowing along the ship from bow to stern. The function W is defined as: $W(w, w_d) = w \cdot I(w_d)$, where $I(w_d)$ is determined by a linear interpolation in Table 3.

w_d	Type	Increase (%)
315-360, 0-45	Head wind	4.0
45-135, 225-315	Side wind	2.0
135-225	Tail wind	1.0

Table 3: Increase in fuel consumption (%) due to wind from different directions

5.3.4 Example:

The ship is running at 18 knots. The current is 1 knot along the direction of the ship. The wind blows 4 Beaufort points straight against the starboard side of the ship (w_d=90). The water depth d is 15 meters below the keel. The speed through water is then $18 - 1 = 17$ knots. Table 1 shows the fuel consumption, before the wind and limited depth effects have been taken into account, at 1300 liters/hour. Table 2 shows the fuel consumption increase by 10%, caused by the water depth.

Table 3 gives a $4 * 2\% = 8\%$ increase due to the side wind. The total estimated fuel consumption in the example is therefore $1300 * 1.10 * 1.08 = 1544$ liters/hour.

5.4 Formulating the Optimization Problem

The route is divided into n parts, each having a constant depth, wind, and current conditions. n is typically between 2 and 40. Each part is denoted "route leg" or just "leg". For each leg i, the following data is available:

- Length s_i (nautical miles)
- Longitudinal current component $c_{l,i}$, (unit: knots) parallel to the ship's direction
- Transversal current component $c_{t,i}$, (unit: knots) perpendicular to the ship's direction
- Wind strength w_i (Beaufort)
- Relative wind direction $w_{d,i}$. (0.-360) (see Table 3)
- Water depth d_i (meters) below keel
- Minimum allowed bottom track speed $xmin_i$ (knots)
- Maximum allowed bottom track speed $xmax_i$ (knots)
  For the route as a whole, the following data is provided:
- Total travel time T (unit: hours)

The ship's speed relative to the ground (also called bottom track speed) is given in knots (1 knot equals 1 nautical mile per hour). The speed relative to the ground, x_i, is related to the speed through water, $x_{w,i}$ (also called water track speed) and the current components $c_{t,i}$ and $c_{l,i}$ according to:

$$x_{w,i} = \sqrt{c_{t,i}^2 + (x_i - c_{l,i})^2}. \tag{6}$$

In other words: the water track speed $x_{w,i}$ is the vector difference between the bottom track speed and current vector. The fuel consumption (liters) on leg i is denoted C_i and is a function of speed over ground x_i and the properties of leg i ; the length s_i , the current ($c_{l,i}$, $c_{t,i}$) , the wind strength w_i, the wind direction $w_{d,i}$, and the water depth d_i:

$$C_i = \frac{s_i}{x_i} F(x_{w,i})(1 + D(x_i, d_i)/100)(1 + W(w_i, w_{d,i})/100 \tag{7}$$

where F, D, and W are given by Tables 1, 2 and 3 respectively. The factor $\frac{s_i}{x_i}$ is the time (hours) that the ship is on route leg i.

For the optimization algorithm it is practical to express the fuel consumption as a function of the bottom track speed x. We therefore define the fuel consumption (liters) on leg i for x knots bottom track speed as

$$C(x, i) = \frac{s_i}{x} F(x_w)(1 + D(x, d_i)/100)(1 + W(w_i, w_{d,i})/100 \tag{8}$$

where

$$x_w = \sqrt{c_{t,i}^2 + (x - c_{l,i})^2}. \tag{9}$$

The vector $\mathbf{x}$ is defined as $(x_1, x_2, ..., x_n)$, i.e. the unknown bottom track speeds on the n legs. The total fuel consumption for a voyage is given by:

$$\Phi(\mathbf{x}) = \sum_{i=1}^{n} C(x_i, i). \tag{10}$$

The objective of the optimization is to find the speed vector $\mathbf{x}$ that minimizes $\Phi(\mathbf{x})$.

5.4.1 Constraints

As constraints in the optimization of $\Phi(\mathbf{x})$ we have:

1. $\sum_{i=1}^{n} \frac{s_i}{x_i} = T$. I.e. the ship has to arrive on time
2. $xmin_i \leq x_i \leq xmax_i, \forall i$. These constraints can be used to define speed limits on parts of the route, and also to set the available speed register for the ship.

$\Phi(\mathbf{x})$ should now be minimized with respect to $\mathbf{x}$ under the above mentioned constraints.

5.4.2 Start Value Algorithms

As start value for $\mathbf{x}$, three methods have been considered:

1. Assign an equal speed to all legs. I.e.: $x_i = \sum_{j=1}^{n} \frac{s_j}{T}, \forall i$. This method hardly needs any calculations, but on the other hand does not take current, wind or water depth into account.
2. Compute one value x_w for the speed through water, same for all legs, that makes the ship arrive on time (i.e. $\mathbf{x}$ fulfills constraint 1 above). This means that legs with a counter current are run at a lower speed (through water) than legs with the current along. I.e.: Compute a value x_w that solves

$$\sum_{i=1}^{n} \frac{s_i}{x_i} = T \tag{11}$$

$$x_w = \sqrt{c_{t,i}^2 + (x_i - c_{l,i})^2}, \forall i. \tag{12}$$

   Expression 12 assigns values to all bottom track speeds x_i in such a way, that the water track speed becomes equal in all legs. If constraint 2 above out rules the necessary bottom track speed for a leg, assign the relevant end point in the constraint to x_i. This method gives an $\mathbf{x}$ vector that compensates for the current, but not for the water depth d or the wind w.

3. Compute a fuel consumption λ (liters/hour) that, if used on all legs on the route, makes the ship arrive on time (fulfills constraint 1 above). This means that legs with a counter current are run at a lower speed over ground than legs, in which the ship runs with the current. It also means that legs with a heavy load due to shallow waters and/or wind are run more slowly than other legs. The algorithm involves solving two nested equations:

   - Find the fuel consumption λ (liters/hour) that solves:

$$\sum_{i=1}^{n} \frac{s_i}{y_i(\lambda)} = T \tag{13}$$

     where $y_i(\lambda)$ is the bottom track speed achieved on leg i if the fuel consumption is λ liters/hour. I.e.: Each $y_i(\lambda), \forall i$ has to satisfy

   - $$C(y_i(\lambda), i) \cdot \frac{y_i(\lambda)}{s_i} = \lambda \tag{14}$$

     where C is given by equation 8.

   Equation 13 is solved by the secant method. Each term in the sum requires solving equation 14 for a particular value on λ. This is also done by the secant method. If constraint 2 above out rules the computed bottom track speed for a leg i, the relevant end point in the constraint to x_i is assigned to $y_i(\lambda)$.

Start value algorithm 3 gives an $\mathbf{x}$ vector most often very close to the optimum for $\Phi(\mathbf{x})$, and can actually be used to compute the speeds on the different legs on the ship's route. Further attempts to improve the reached optimum can be found in [1], where a number of optimization routines are applied to the problem. The tables with models are approximated with continous functions so the derivatives can be computed analytically. Both quasi-Newton and conjugated gradient methods are applied together with a variant of Fletcher's line searching algorithm [2] and also with a golden section search algorithm. The combination of quasi-Newton and Fletcher's line searching gives best results, but a general conclusion is that start value algorithm 3 most often give a good enough solution vector, and definitely much faster. Therefore, the route planning module in Seapacer uses this algorithm to compute optimal bottom track speed values for each leg in the route plan. The optimization is repeated at certain interval and also when new data for current, wind or depth is input during the voyage.

5.5 Using the Optimized Route

The input to the optimization consists of positions for the legs in the route. The following data is also given for each leg:

- **CURRENT -** Strength c and absolute direction c_{dir} of current. Unit: knots.
- **WIND -** Strength w and absolute direction w_{dir} of wind. Unit: Beaufort.
- **DEPTH -** Mean water depth below the keel. Unit: meters.

c_{dir} and w_{dir} are absolute values, and are entered as any of the following abbreviations: N, S, W, E, NE, NW, NNW, NNE, SE, SW, SSE, SSW, ENE, ESE, WNW, WSW. The relative directions are computed automatically by the system, depending on the ship's actual course at each moment. Additional inputs are the ship's mean draft and required departure/arrival times. The route optimizer computes the bottom track speeds $x_i, \forall i$, that minimize the total fuel consumption for the voyage. The arrival time is always kept as requested. The route plan shown in Table 4 has been computed with algorithm 3 described above. The set values x for speed are automatically executed as POSITION/ARRIVAL commands in Seapacer (see Section 4.3). The speed control is combined with the dynamic limits for fuel consumption (see Section 4.2). In this way the engines are controlled in a smooth and economical way throughout the route. The route is automatically re-optimized every 10 minutes or when new current or wind values are entered.

6 Conclusion

The propulsion of ships offer many interesting and challenging control and optimization problems. The control problems are characterized by high-time constants and noisy sensor signals. For reasons of robustness and generality, simple and intuitive solutions are often to be preferred. Furthermore, the noisy and time-affected nature of the problem makes the search for global and exact optima pointless, and even sub-optimal, if it involves a slower system with a higher risk for volatile behavior.

We have successfully implemented a number of control systems that aim at lowering the fuel consumption by optimizing control. The systems have worked particularly well for vessels with a wide speed-control range. This gives room for intelligent route planning, which really makes a difference for the total fuel consumption. The pitch optimization has worked best for older ships, where the initial rpm/pitch combinations are far from optimal. Newer ships have partly recognized the importance of having correct rpm and pitch at varying running conditions, and allow less room for a separate optimizer such as Seapacer. The general trend in bridge equipment has, for a number of years, been integrated systems, where the same manufacturer delivers integrated equipment for many bridge functions. Along this trend, some radar manufacturers are offering primitive route planning functions and speed control functions as options in their systems. Also, advanced electronic sea chart systems are likely to include more and more route planning options in the future. Advanced optimizing functions, such as the ones described in this report, have still not been implemented in other products, to the authors knowledge.

SEAPACER PC VOYAGE CONDITIONS SET UP											
MS Emmaräng											
Voyage conditions 1											
Route number 1: Göteborg - Kiel											
Mean draft: 6.3 meters											
leg		$Name$	c	c_{dir}	w	w_{dir}	$Depth$	ME	x	x_w
1	at	KAJ GÖTEBORG	0.0	N	3.0	SW	8	1	7.00	7.00
2	at	Göteborgsgrund	0.0	S	3.0	SW	10	3	13.00	13.00
3	at	Brandnäs	0.0	S	3.0	SW	22	4	17.81	17.81
4	at	Trubaduren	0.0	N	3.0	SW	40	4	19.13	19.13
5	at	Vanguardsgrund	0.0	N	3.0	SW	45	4	19.13	19.13
6	at	No. 4	0.0	N	3.0	SW	30	4	18.63	18.63
7	at	Anholt Knob	0.0	N	3.0	SW	30	4	18.49	18.49
8	at	No. 7	0.0	N	3.0	SW	15	4	17.32	17.32
9	at	No.11	0.0	N	3.0	SW	18	4	17.52	17.52
10	at	SNR	1.0	N	3.0	SW	22	4	17.08	17.94
11	at	No.15	1.0	N	3.0	SW	16	4	16.75	17.54
12	at	No.20	1.0	N	3.0	SW	20	4	16.97	17.96
13	at	No.23	1.0	N	3.0	SW	22	4	17.15	18.09
14	at	No.25	1.0	N	3.0	SW	16	4	17.17	18.00
15	at	No.26	1.0	N	3.0	SW	24	4	17.24	18.24
16	at	No.28	1.0	N	3.0	SW	30	4	18.16	19.08
17	at	Vengeance grund	1.0	N	3.0	SW	30	4	17.72	18.72
18	at	Ägersö	1.0	N	3.0	SW	20	4	16.96	17.80
19	at	No.2	1.0	N	3.0	SW	20	4	16.97	17.96
20	at	No.3	1.0	N	3.0	SW	20	4	17.01	17.95
21	at	DW 51	1.0	N	3.0	SW	20	4	17.18	17.76
22	at	DW 53	1.0	N	3.0	SW	20	4	16.89	17.82
23	at	DW 55	1.0	N	3.0	SW	30	4	17.72	18.72
24	at	DW 57	1.0	N	3.0	SW	18	4	16.77	17.69
25	at	DW 59	1.0	N	3.0	SW	18	4	16.86	17.83
26	at	DW 61	1.0	W	3.0	SW	14	4	17.88	17.10
27	at	KO 2	1.0	W	3.0	SW	10	4	17.54	16.65
28	at	Kieler Förde	1.0	W	3.0	SW	10	4	17.20	16.76
29	at	Fartbojen	1.0	W	3.0	SW	10	3	11.00	10.77
30	at	Friedrichsort	1.0	W	3.0	SW	8	1	8.00	7.50
31	at	Kiel 11 Reede	1.0	W	3.0	SW	7	1	7.00	6.72
32	at	Scwedenkai								
33	at	End of route plan								
Optimization data:											
Mean: 3078 l/h Total: 40320 litres Total dist: 234.6 nm Total time: 14:00											
Successful optimization											

Table 4: Optimized route plan for the route Göteborg-Kiel. The x values are optimized speed (bottom track) values compensated for the varying depth values and weather conditions on the 31 legs. The x values are used as set values for the speed control along the route.

7 Acknowledgments

I wish to acknowledge the invaluable help from electrical engineer Göran Ekesfors, and former technical director Gösta Kjellberg. The Seapacer system would most definitely not have come to existence without their dedication and excellent domain knowledge. Many thanks.

References

[1] E. Dotzauer and G. Rosendahl. Ruttplanering för fartyg. Bachelor thesis, Department of Mathematics and Physics, Mälardalen University, Sweden, 1995.

[2] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, New York, 2nd edition, 1987.

[3] T. Hellström. Seapacer Optimizing System Mark II. Technical report, http://www.cs.umu.se/ thomash/products/seapacer.pdf, Seapacer AB, Sweden, 1996.