10th International Conference on

Computer and IT Applications in the Maritime Industries

COMPIT'11

Berlin, 2-4 May 2011

Edited by Volker Bertram

TuTech
TUHH Technologie GmbH

A Machine-Learning Approach to Predict Main Energy Consumption under Realistic Operational Conditions

Jóan Petur Petersen, DTU Informatics, Lyngby/Denmark, joanpeturpetersen@gmail.com
Daniel J. Jakobsen, Decision3, Tórshavn/Faroe Islands, dj@decision3.com
Ole Winther, DTU Informatics, Lyngby/Denmark, owi@imm.dtu.dk

Abstract

We present a novel and publicly available data set of high quality sensory data collected from a ferry over a period of 2 months, and investigate state of the art machine-learning methods for prediction of main propulsion fuel efficiency. Neural networks are applied in both instantaneous and predictive settings. Performance results for the instantaneous models and examples and discussions of the practical advantage of the predictive models are given. The presented models have been successfully deployed in a trim optimization application onboard one of DS NORDEN's product tankers.

1. Introduction

Most modern ships have several measurement devices that keep track of its operations, such as the vessel's speed, fuel consumption, weather conditions, and so forth. Storing, analyzing, and acting upon this data could become a valuable asset for the ship owners, operators, and crew. In this article we present a freely available data set of data collected from a ferry, where the data has been used to develop the models used in a trim optimization application.

Already in 1987 the technical article "Marine Performance Surveillance with a Personal Computer" by *Journée (1987)* and *Journée et al. (2003)* described an on-line data collecting system and a mathematical model tuned by the collected data to predict ship performance. This mathematical model is based upon physics and hydrodynamics formulas. In the article by *Leifson et al. (2008)* model combinations of conventional physical models and artificial neural networks are tested. In the article by *Pedersen and Larsen (2009)* an Artificial Neural Network is used to predict the propulsion power. Both papers make use of collected operational data from the ship.

Ship resistance evaluation methods can be divided into four groups, from the traditional to the more advanced: Traditional and standard series methods, regression based methods, direct model tests, and computational fluid dynamics (CFD) *Molland (2008)*. The traditional, standard series, and regression based methods typically rely on a set of parameters that describe the hull, for example the wetted area, the Froude number, and so on. A well-know regression based approach to predict speed/power is the Holltrop-Mennen method, *Holtrop and Mennen (1982)*, *Holtrop (1984)*, which is widely applied in the initial design stage. Here the total resistance of the ship is subdivided into additive components, which are estimated based on data collected from model and full-scale tests. Other full-scale and model experiments have been carried out to investigate how different components affect the total resistance of the ship, such as fouling and wind load *Townsin (2003)*, *Blendermann (1996)*. The Holltrop-Mennen methods have been applied for operational optimizations *Leifson et al. (2008)*. These initial design methods may sacrifice details for the sake of fewer parameters, and more robust results. Because of this sacrifice, these methods might not be very well suited for analysis of the ship's performance after it has been constructed. Computational Fluid Dynamics, CFD, can be used to estimate hull resistance *Ruggiero et al. (2007)*, but these calculations are computationally demanding, and typically take several hours to complete. So it is infeasible to use them onboard directly for optimization applications. It is however possible to do the CFD calculations beforehand for a given set of parameters. This technique has been used successfully for an trim advisory system *Hansen and Freund (2009)*.

In contrast to the physical models, statistical approaches only require knowledge about the ship and the physical setup when it comes to the design of what sensory features to use in the system. The statistical model learns the relations between the measured signals from the collected training data. In the predictive machine learning approach the main agenda is generalization. This means the ability of the model to give sensible predictions for situations not identical to what has already been observed in the training data. The main factors determining the predictive performance are the relevance of the sensory features, the amount of training data available and the choice and tuning of the complexity of the model to the task at hand. Hybrid physical and statistical models with a proper assessment of the predictive uncertainty of each component may in principle improve over a pure model by suitably interpolating between the two *Leifson et al. (2008).*

In a system that can optimize the fuel consumption for propulsion, the two key quantities we need to be able to predict are the fuel consumption f and speed through water v. Ignoring route planning concerns the fuel efficiency is defined as the speed through water divided by the energy (fuel) consumed: $e = v / f$.

The ship's state is the set of variables that allow us to predict f and v. Depending on the application, a number of these quantities can be considered as control variables such as the trim, propeller pitch, RPM, i.e. these can be changed by the crew, while others like weather conditions or physical characteristics of the ship cannot. All relevant state variables are not necessarily measured - there can be several reasons for this, for example by choice, cost, or complexity. But arguably the most important factor in getting good model predictions is to have rich and accurate sensory data. An *instantaneous* model predicts the current value based upon current measured signals. It therefore does not treat the control variables differently than the remaining variables. Using the model directly as part of an advisory system for optimizing fuel efficiency on-line can be problematic, because a change in a control variable may affect several variables, including inputs to the model and not only the outputs of the model. For example changing the trim could affect the way the autopilot controls the rudder, as the dynamics of the ship are also affected, which would result in an incorrect estimate of the speed and fuel consumption. So a temporal state-space model, *Ghahramani and Hinton (1996)*, for the dynamics of all variables is a more adequate model for this purpose. We will not consider a complete state-space model with hidden states, but instead predictive model, which uses a non-linear neural network to capture the assumed Markovian dynamics of the system, *Weigend et al. (1990), Chakraborty et al. (1992), Svarer et al. (1993).* This so-called *tapped-delay neural network* is not explicitly defined in terms of a model with hidden states like the traditional state-space model, but can never the less capture the essential part of the dynamics of the system. Ideally, the tapped-delay model represents the deterministic part of the dynamics. With careful regularization in order to avoid over-fitting, we may also use the residual error of the model on the training to fit an additive term in the model to represent the stochastic part of the model due to noise and/or un-observed information. The full model consisting of the deterministic and stochastic part allows us to make efficient non-linear propagation of predictions and uncertainty through time. The instantaneous model and the tapped-delay neural network, which we will focus on in this article, may be considered as an important first step towards a complete state-space model since it actually models v and f, which in the complete state space view are just two of the state variables.

2. Data collection

The data is collected from a domestic ferry, M/S Smyril in the Faroe Islands, owned by Strandfaraskip Landsins. The ship serves a daily route from the capital Tórshavn to the southernmost island Suduroy. The ferry sails two to three trips back and forth each day, where the duration of each trip is 1 h 55 min. The ship is designed by Knud E. Hansen, and built on the IZAR shipyard, San Fernando, Spain (delivered in 2005). Fig. 1 gives some data on the vessel.

A computer system and some additional hardware is installed onboard the ferry. This system collects the data from the ship, which the models will be based upon. The data is made publicly available, and can be found here: http://cogsys.imm.dtu.dk/propulsionmodelling/.

Length overall	135.00 m
Breadth moulded	22.70 m
Depth to main deck	8.10 m
Design draught	5.60 m
No. of private cars	970 m / 200 cars
No. of passengers	800 / 975 pax
Service speed	21 kn
Main engines	4 x 3360 kW (13440 kW)
Aux. engines	4 x 515 kW
Identification number	KEH 00072
IMO number	9275218

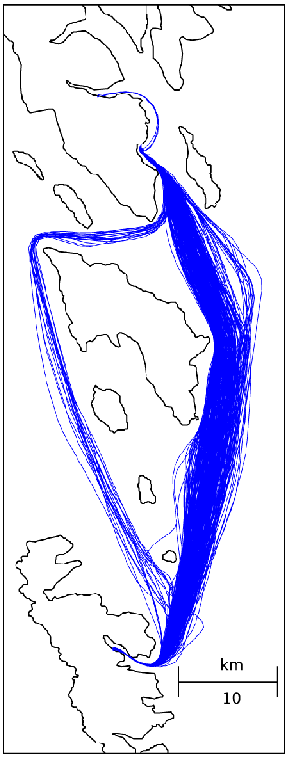

Fig.1: Information about the vessel Smyril, and the collected trips in the data set.

The data spans a period of almost two months, February 16th to April 12th, 2010. The map in Fig. 1 shows the routes taken by the vessel during the data collection period. We have made this data available in an attempt to encourage benchmarking within the ship propulsion field. Decision3 has provided the data for the project. To our knowledge, this is the first data set of its type that has been made publicly available. By releasing it to the community, we hope that it will be used for benchmarking similar models, and further work within this field.

The following signals are stored by the system: Port and starboard propeller pitch, port and starboard rudder angle, port and starboard level measurements, fuel density, fuel temperature, fuel volume rate, trim angle, longitude, latitude, speed through water, speed over ground, true heading, wind speed, and wind angle. Each signal is stored in a Comma Separated Values (CSV) file. For more details please refer to the homepage. All of the signals supplied in the data set are common onboard most ships. However the microwave based level measurements devices are quite novel, though they have been used before *Atwater (1990)*. With these devices it is possible to get information about the squat, trim, heeling, and draft of the ship, the waves generated by the motions of the ship, and sea waves around the ship. Much more work could be done interpreting these signals.

The system was setup to start storing data from the point when the ship starts to move and until it stops. Subsequently this might have been too restrictive, as the information just before and after a trip could also contain valuable information, which has not been stored. Smyril has a shaft generator, but it has not been used during the period of data collection. The engine has a constant RPM, and it has therefore not been measured.

If we look at the map in Fig. 1, we see that there are two short trips, where Smyril sails northward from Tórshavn. Here the ship is bunkering heavy fuel oil. The bunkering can be seen on the vessels draft. The variation in the routes taken is due to weather conditions.

3. Methods

3.1 Instantaneous/regression model

We use supervised learning methods for building a instantaneous/regression model to estimate the fuel consumption and speed from a set of measured features, *Bishop (2007)*. Formally the response (or output) variable, y, e.g. the fuel consumption, is modeled as

$$y = f(x;\theta) + \varepsilon, \tag{1}$$

where f denotes the model function depending upon parameters θ and explanatory (or input) variables x. The residual ε is the part of the measured signal y not explained by the model due to noisy measurements and/or model shortcomings. The parameters θ should be learned (inferred) from a training set $\mathbf{y} = \{(\mathbf{x}_n, y_n) \mid n = 1, \ldots, N\}$ of input output pairs. We will use a probabilistic formulation assuming that the residual has a Normal (or Gaussian) zero-mean and σ_ε^2 variance distribution: $Norm(\varepsilon|0,\sigma_\varepsilon^2)$. We will assume that measurements are independently identically distributed (iid) such that we can write the likelihood as

$$p(\mathbf{y}|\mathbf{X},\theta,\sigma_\varepsilon^2) = \prod_{n=1}^{N} Norm(y_n|f(\mathbf{x}_n;\theta),\sigma_\varepsilon^2), \tag{2}$$

where $\mathbf{y}$ and $\mathbf{X}$ are shorthand for the vector of output and matrix of inputs, respectively. We will use an Artificial Neural Network (ANN), *Bishop (2007)*, a non-linear model for f. We briefly review this method below including details for our use of the method, *Bishop (2007)* for detailed general description. We take a predictive rather than descriptive modeling approach reporting the test (or generalization) performance on a test set. In all our experiments with the instantaneous model we use one third of the data as test set and the remaining two-thirds as training set. We investigate two different ways to make the training-test split. In the first approach test data is selected at random and in the second we minimize the training-test redundancy (or cross talk) coming from adjacent time-windows using complete legs as test and training data units, see Section 4.1 for details.

3.2 Tapped-delay neural network

While the instantaneous model only predicts the current target value, the tapped-delay neural network setup can predict future values.

A tapped-delay neural network model can be described as

$$\mathbf{Y}_n = f(\mathbf{X}_n, \mathbf{X}_{n-1}, \ldots, \mathbf{X}_{n-d+1}, \mathbf{w}) + e(n, \mathbf{w}), \tag{3}$$

where f is in this case is a artificial neural network that maps the seen sequence of sample vectors, $\mathbf{X}_n$, into a predicted future sample, *w* are the model's parameters often also called weights, and *e* is the error in the prediction. The error is only considered to be additive here. The number of previous samples vectors available to the network or steps is given by *d*.

We will make some small modifications to the tapped-delay neural network, as to adapt it more to this application. The output of the network is set to predict the next ship state, so that we want the output of the network to be $\mathbf{Y}_n = \mathbf{X}_{n+1}$. The controls are added as inputs to the model. The controls are considered to be known, so these are not predicted by the model. The controls are represented by a vector $\mathbf{U}_n$. The variable $\mathbf{X}_n$ will express the dynamic state of the ship. It will contain the ship's speed through water, trim, draft, and so forth. The value from the current step, $\mathbf{X}_n$, is added to the right-hand side of Eq.(3). The argument for doing this, is that we suspect the next step, $\mathbf{X}_{n+1}$, to be quite similar to $\mathbf{X}_n$,, and therefore the model will only have to learn the difference between the steps. We will concentrate on the simplest tapped-delay neural network first, where $d = 1$. Applying these modification to Eq.(3) we obtain a new model given by

$$\mathbf{X}_{n+1} = f(\mathbf{X}_n, \mathbf{U}_n, \mathbf{w}) + \mathbf{X}_n + e(n, \mathbf{w}) . \tag{4}$$

3.3 Data pre-processing

The samples are arranged into intervals or windows. The windows are non-overlapping and there is no gap between consecutive windows on a trip. If measurements are missing completely from a device in a window, the window is discarded. A device might have malfunctioned or stopped providing data for some other reason. Based on the samples within a window, a number of features are generated. The feature extraction process consists mostly of simple mathematical operations such as taking the mean, variance, or derivative of an input. The mean, variance, and derivative features are calculated as

$$\Omega_{mean}(w)=\tfrac{1}{M}\sum_{n=0}^{M} x_n \ , \ \Omega_{\mathrm{var}}(w)=\tfrac{1}{M}\sum_{n=0}^{M}\left(x_n-\Omega_{mean}(w)\right)^2 \text{, and } \Omega_{der}(w)=\tfrac{1}{M-1}\sum_{n=0}^{M}\frac{x_{n-1}-x_n}{t_{n-1}-t_n}\ , \tag{5}$$

where w is a window identifier, M the number of samples within the window, and x_n samples (n being its index) from the selected input signal within a window, and t the corresponding time stamps. The variance feature will express how much the signals vary within a window. The variance of the surface distance measurements will for example give the model an idea of the waves surrounding the ship. The derivative feature will tell the model if a signal is increasing or decreasing within a window.

The choice of the window size depends on the application. We found that a window size of three minutes represent a good trade-off between robust estimation of data and time-scale for change in the variables for the instantaneous model. For the given data set using a window size of 3 min gives a total of 9001 windows. Others have found a window size of 10 min to be suitable *Pedersen et al. (2009)*, *Leifsson et al. (2008)*. The tapped-delay network example will have a much shorter length, as here we are interested in seeing the dynamic changes. The window size used for the tapped-delay neural network is 15 s.

Table I: Features extracted for the instantaneous model.

Description	Mean	Variance	Derivative
Speed through water *(target)*	√		
Fuel consumption *(target)*	√		
Trim	√	√	√
Port and starboard pitch	√	√	√
Port and starboard rudder	√	√	√
Heeling	√		
Draft	√		
Port and starboard level	√	√	√
Headwind and crosswind	√	√	√

Table I gives the features used for the instantaneous model tested. The features speed through water and fuel consumption are model outputs, while the rest are for inputs – there is a total of 29 inputs. The draft and heel are estimated from the trim and level measurements. For the tapped-delay neural network example, we will only use a small subset of these.

3.4 Artificial Neural Network

We will use a feed forward neural network with two layers of adaptive parameters and bias units[1]:

$$f(\mathbf{x};\theta) = \tilde{g}\left(\sum_{j=0}^{M} w_j^{(2)} g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right)\right). \tag{6}$$

The parameters of the model are the weights of the two layers $\theta = \left(\mathbf{W}^{(1)}, \mathbf{w}^{(2)}\right)$. Non-linearity is achieved through the activation functions $\tilde{g}$ and g, which are usual taken to be sigmoid-functions such as hyperbolic tangent or the logistic function. In our case we use a linear output activation $\tilde{g}(a) = a$, and non-linear activation for the hidden layer $g(a) = \tanh(a)$. The output of the network therefore becomes a linear combination of the nonlinear activation functions.

In order to control the complexity of the model we use regularization of the weights. This can be formulated in a number of ways. Here we view it as maximum a posteriori (MAP) estimation $\theta_{MAP} = \arg\max_{\theta} = p(\theta|D)$, where the posterior is given by

$$p\left(\theta \middle| D, \sigma_{\varepsilon}^2, \sigma_{\theta}^2\right) = \frac{p\left(\mathbf{y} \middle| \mathbf{X}, \theta, \sigma_{\varepsilon}^2\right) p\left(\theta \middle| \sigma_{\theta}^2\right)}{p\left(\mathbf{y} \middle| \mathbf{X}, \sigma_{\varepsilon}^2, \sigma_{\theta}^2\right)}. \tag{7}$$

and $p\left(\theta \middle| \sigma^2\right)$ is the prior distribution of the parameters. We take this distribution to be independent and identically distributed (i.i.d.) for weights and the prior for a single weight to be $Norm(w|0,\sigma_{\theta}^2)$. We can formulate the MAP estimation problem as an equivalent cost function minimization problem by taking the logarithm and omitting terms not depending upon θ:

$$E(\theta;\lambda) = \tfrac{1}{2}\sum_{n=1}^{N}\left(f(\mathbf{x}_n;\theta) - y_n\right)^2 + \lambda\|\theta\|^2, \tag{8}$$

where $\|\theta\|^2$ is the two-norm or sum of the squared elements of the weights and $\lambda = \sigma_{\varepsilon}^2 / \sigma_{\theta}^2$. Optimization of this non-linear cost function for constant hyper-parameter λ is performed by minimizing the error gradient found with back-propagation *Bishop (2007).* Model prediction is computed according to Eq. (6) using the optimized weights which will be a function of both the training data, the hyper-parameters and possibly the initialization of parameters and learning rates. The learning of the hyper-parameter λ is done by *k*-fold cross-validation, where *k*=10. In *k*-fold cross-validation we split the training data into k approximately equal sized sets. We perform *k* training runs using in turn one of the sets for validation. The average test error over the k runs is proxy for the test error for a model trained on $N - N/k$ examples. We scan over a range of λ values and select the one with the lowest cross-validation error, see Section 4.

---

[1] The bias terms are included in a compact formulation by letting the summation start from zero and clamping both the zeroth input and the zeroth output from the hidden unit to minus one.

3.5 Performance measures

The accuracy of statistical prediction models may be quantified in terms of for example squared residuals of the model predictions on a test set and should be judged relative to baselines such as the error using the mean fuel consumption or the variance of the fuel consumption. The effect of single variables (for example the control variables) can also be investigated by comparing performance of models with and without these variables. A model which is successful in terms of significant improvement over the baseline can be considered a simulator of the ship in term of fuel consumption and speed. For regression models a convenient measure is the root-mean-square error (RMSE). It has the same unit as the model output variable, and has a similar form to the standard deviation. Given a data set containing N windows, the RMSE can be calculated as follows

$$RMSE = \sqrt{\frac{\sum_n \left\| f(\mathbf{x}_n|\theta) - y_n \right\|^2}{N}}, \tag{9}$$

where f denotes the model function, and θ denotes a model parameter vector. The output of f for a given an input vector $\mathbf{x}_n$ is the predicted value by the model, and y_n is the value actually observed.

4. Results

4.1 Instantaneous model results

As mentioned in section 3.4 the data set has been split into two sets, a training set and a test set. The training set is 2/3 of the data. This has been done by i) selecting random windows from the original set and ii) randomly selecting whole trips (sets of windows). The test sets are not used in any way before evaluating the performance of the models. The weight decay parameter of the artificial neural network models are found using 10-fold cross validation as mentioned in section 3.4. The training set is split into 10 approximately equal sized sets; this split is done window wise. Fig. 2 shows the RMSE for the testing set using a different number of trips in the train set. Here the fuel consumption is predicted using the data split by whole trips. This is done in order to get an indication of how much data is required before the model can be used. Ten different trip sequences have been used - the variation resulting from this is illustrated by the box plots. The shape of the learning curve is quite normal: a steep initial decrease followed by a slower (typically power-law) decrease.

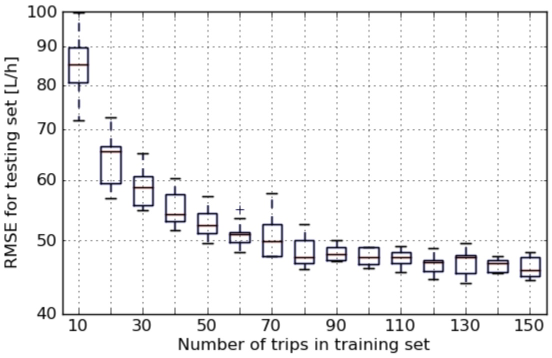

Fig.2: Test set RMSE as a function of the number of trips in the training set. The fuel consumption is predicted using the data sets split up by whole trips. The error bars are obtained by ordering the way the trips are included in the data set in 10 different ways.

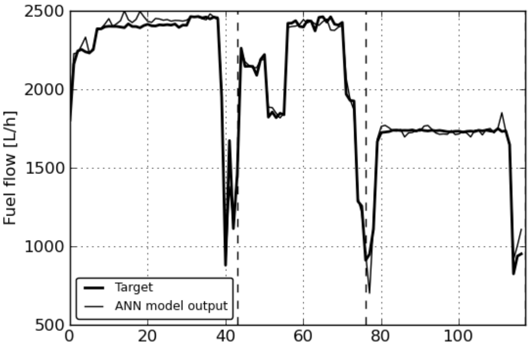

Fig.3: The output from the final ANN model and the corresponding target (true) values. The data points are from three selected trips from the test set (split trip-wise). The trips are separated by vertical dashed lines in the plot.

By splitting up the sets by trips, gives us the ability to examine whole trips from the test set, because otherwise it would be likely that data points from a trip would end up in both the training and testing sets. Three selected trips are shown in Fig. 3, and the predictions obtained from the ANN. It can be seen that the model is able to predict the changes within the trip. These plots give a qualitative impression of the models.

Table II gives the performance results obtained for the instantaneous models, along with the cross validation results, which are expressed as a mean value and a standard deviation. As one might have expected, the models perform a little better using the dataset where the windows are shuffled, due to the cross talk mentioned in section 3.1.

A direct comparison with similar work *Pedersen et al. (2009)*, *Leifsson et al. (2008)* is hard because the sensor data is different. *Pedersen et al. (2009)* report, as best result with an ANN model, a mean relative error on propulsion power of 1.65%. Our results for ANN models are 1.50% for the fuel consumption using the data set split by windows and 1.67% if split by trips. *Leifsson et al. (2008)* report 0.65 knots RMSE for the speed and 60 L/h for fuel. Using the mean relative error is a disadvantage to our model and data, because we have relatively many samples where the speed is low compared to the two other articles.

Table II: Performance results for the instantaneous ANN model

		Shuffled windows		Shuffled trip	
		Speed [kn]	Fuel [L/h]	Speed [kn]	Fuel [L/h]
Performance RMSE	test	0.32	41.1	0.38	47.2
	x-val	0.32 ± 0.012	41.06 ± 0.76	0.39 ± 0.011	46.8 ± 0.75

Figs. 4 and 5 show the histograms for the residuals for two modeled feature values. The distributions of the residuals have been approximated by a normal distribution and Student's t distribution. The Student's t distribution has longer ‘tails’ than a Gaussian. This makes the distribution much less sensitive than the Gaussian to the presence of a few data points which are outliers, *Bishop (2007)*. The probability density distribution for the speed and fuel residuals is narrower than the fitted normal distribution. Outliers often arise in practical applications either because the process that generates the data corresponds to a distribution having a heavy tail or simply through erroneous data.

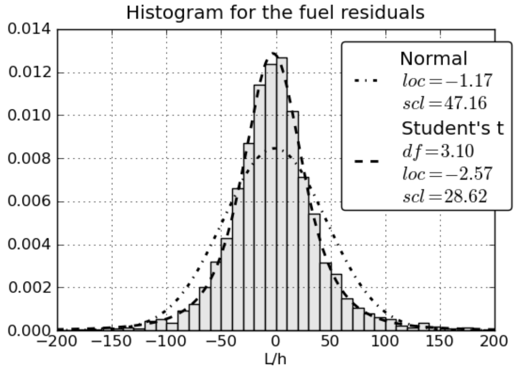

Fig.4: Histogram of the residuals for fuel consumption using the instantaneous model

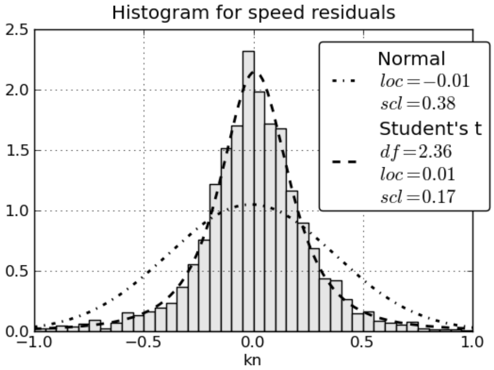

Fig.5: Histogram of the residuals for speed through water using the instantaneous model

4.2 Tapped-delay neural network model and residual modeling

We will keep the tapped-delay neural network example simple, and only include a few parameters in the model. The following control features have been used: Port and starboard pitch, port and starboard rudder, initial port and starboard level, difference between ground and speed through water, headwind and crosswind. The following dynamic variables have been used: Speed through water, port and starboard level, trim, draft, difference in heading. The model is trained on 2/3 of the data using a range of weight decays values, and the model with the best performance based on the rest of the data is selected (validation set).

The predictive model is evaluated using Eq.(4). The noise, $e(n, \mathbf{w})$, will be simulated using a Gaussian and Student's t distribution fitted to the residuals on the validation data. We will use the univariate distribution here for each variable. The model can be initialized with some reasonable settings for the dynamic values, and the model's response to a given input can be found. Including the noise in the dynamic states will give an idea of the sensitive the system is. By examining the plots in Figs. 6 and 7 where the histograms for two of the dynamic variables are given, it is evident that the student's t distribution is a good representative for the residual noise.

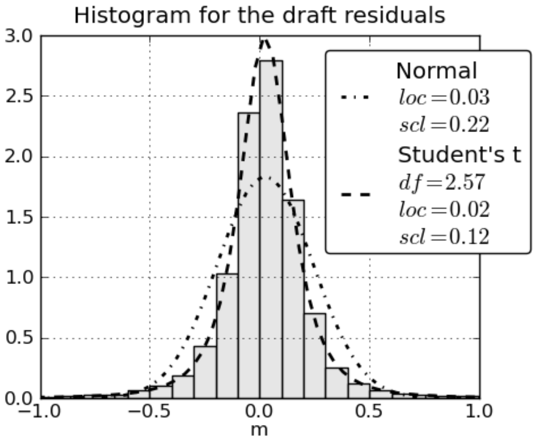

Fig.6: Histogram of the draft residuals using the predictive model

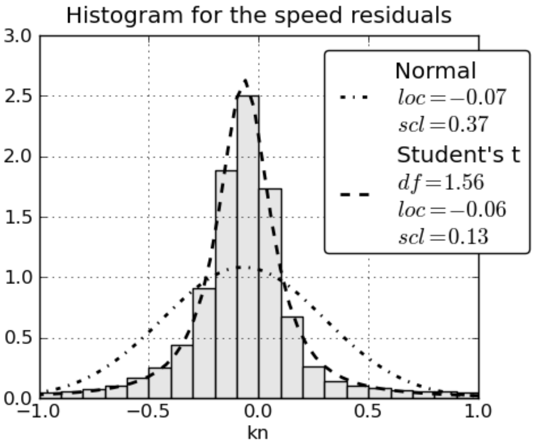

Fig.7: Histogram of the speed through water residuals using the predictive model

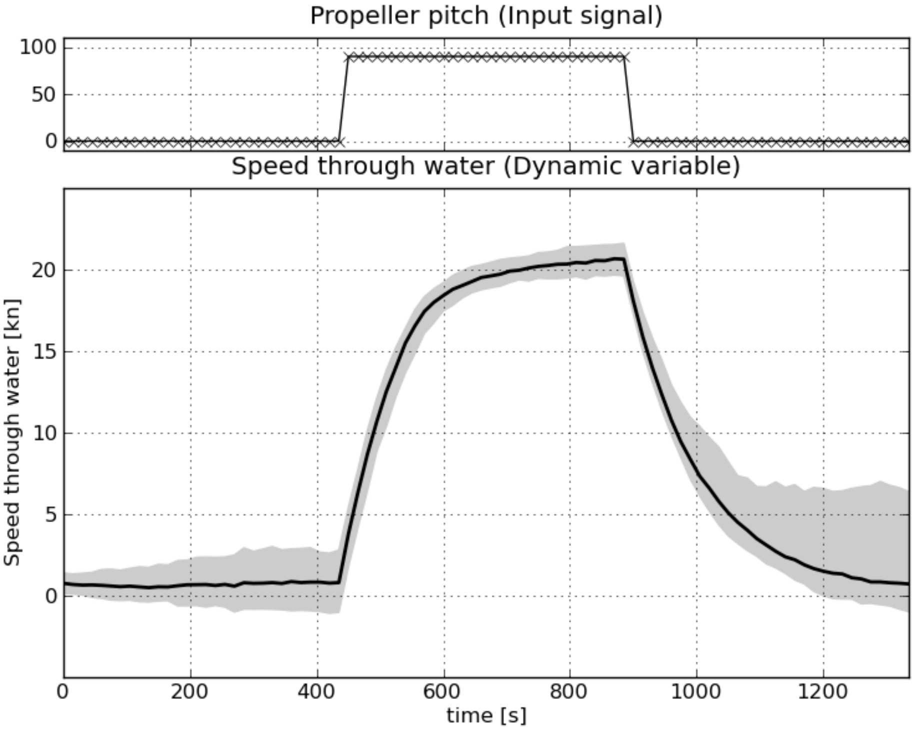

Fig.8: Tapped-delay neural network response to a (rectangular) step function for the propeller pitch. Both propeller pitches are set to the same value. The solid gray area gives the models sensitivity to added Gaussian multivariate noise.

As an example, we have generated 100 dynamic variable samples, and propagated these through the tapped-delay neural network for 90 steps, where each step is 15 seconds. The propeller pitches are changed from being 0 for 30 steps, to being 90 for 30 steps, and then 0 again for 30 steps. The control signal is drawn in the upper plot in Figs. 8 and 9, and the tapped-delay neural network's response is given in the lower plot. The samples have been ordered after the speed through water value, and the area between the 5 lowest and highest values is given by the solid gray interval in the plots. Fig. 8 gives the response from the tapped-delay network with added Gaussian noise, while the plot in Figur 9 gives the response with added noise from the Student's t distribution. If we look at the plots, we can see that the uncertainty in the speed prediction grows with time as the noise accumulates. It also seems like the noise is worse with lower speeds than higher speeds – a possible reason for this could be that we have much more data at the ship's cruising speed, and a smaller part of the data at lower speeds. The system also stops storing data just before the ship stops, so the system newer actually experiences the ship being totally still.

If used correctly this model approach gives us the ability to determine how a change in a control will affect all of the dynamics of the ship; for example how changes in propeller pitch would affect the trim of the vessel. This is a clear advantage compared to the instantaneous model.

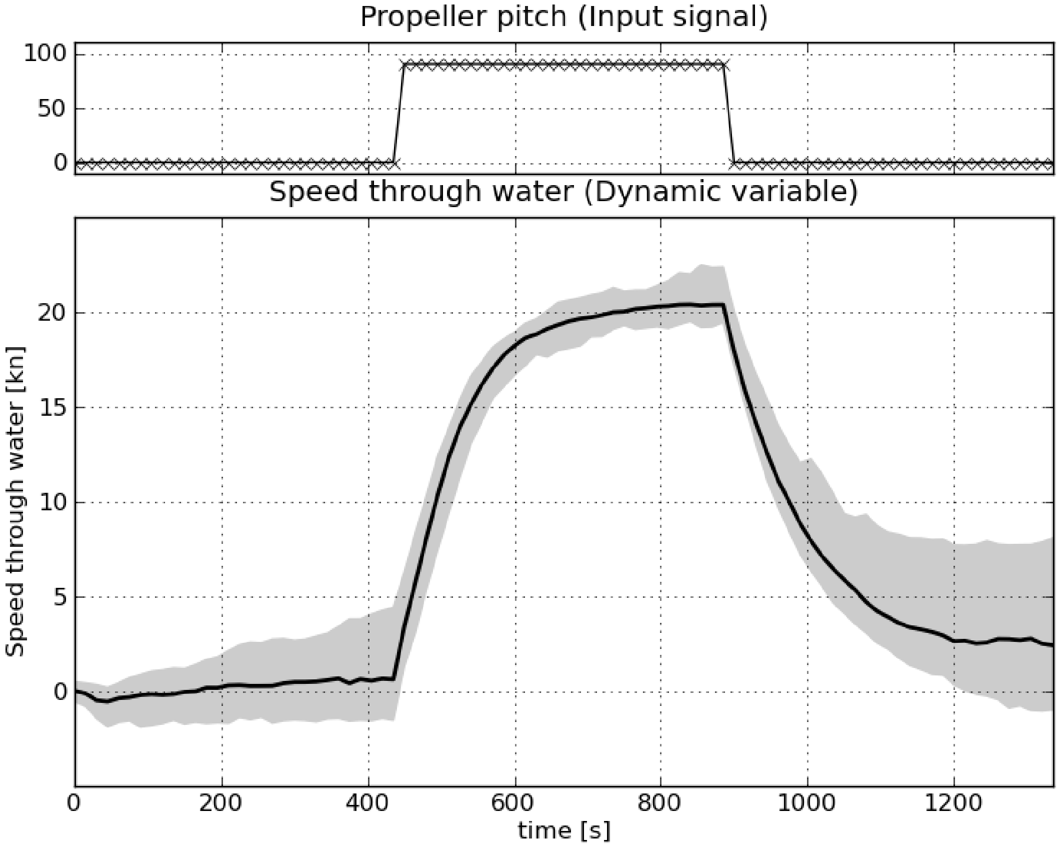

Fig.9: Using the same model and control signal sequence as in Fig. 8, but now the noise is drawn from the student's t distribution

5. Conclusion

A non-linear ANN model has been used for modeling of fuel efficiency in ship propulsion. The instantaneous model, using a artificial neural network model, has been used before in similar work *Pedersen et al. (2009)* and *Leifsson et al. (2008)*. The results obtained are similar to those of previous works; it is however difficult to compare these results, because the data used is different. Difficulties comparing our results with previous work might indicate that there is a need for some publicly available data, which can be used for benchmarking these models and methods. The high quality sensory data set presented here can hopefully fill this gap, and be used as a common reference, or encourage others to make their data publicly available.

Our ongoing work within this area will focus on improving these models, especially with regard to the problem that several state variables are affected when changing one of them. This is a problem that for example can occur when using these models in a trim optimization application. For example advising a trim that is far from the current trim, will make the inputs invalid when the new trim is reached, and the model will give different results. It is plausible that a state-space model or similar may be able to handle this problem.

References

ATWATER, R. (1990), *Shipboard wave height sensor*, Tech. rep., Ship Structure Committee

BLENDERMANN, W. (1996), *Wind loading of ships collected data from wind tunnel tests in uniform flow*, Tech. rep., Institut fur Schiffbau, Univ. Hamburg

BISHOP, C.M. (2007), *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer.

CHAKRABORTY, K.; MEHROTRA, K; MOHAN, C. K.; RANKA S. (1992), *Original contribution: Forecasting the behavior of multivariate time series using neural networks*, Neural Netw., Elsevier Science Ltd.

GHAHRAMANI, Z.; HINTON, G.E. (1996), *Parameter estimation for linear dynamical systems*. Tech. rep., Univ. Toronto

HANSEN, H.; FREUND, M. (2010), *Assistance tools for operational fuel efficiency*, 9th Conf. Computer and IT Applications in the Maritime Industries (COMPIT), Budapest, pp.356–366

HOLTROP, J. (1984), *A statistical re-analysis of resistance and propulsion data*. Int. Shipbuilding Progress 31, pp.272–276

HOLTROP, J.; MENNEN, G.G.J (1982), *An approximate power prediction method*, Int. Shipbuilding Progress 29, pp.166–170

JOURNÉE, J. (2003), *Review of the 1985 full-scale calm water performance tests onboard m.v.mighty servant 3,* Tech. Rep. DUT-SHL Report 1361, TU Delft, Ship Hydromechanics Laboratory

JOURNÉE, J.; RIJKE, R.; VERLEG, G. (1987), *Marine performance surveillance with a personal computer*. Tech. Rep. Report 753-P, TU Delft, Ship Hydromechanics Laboratory

LARSEN J. (1993), *Design of neural network filters*, Ph. D. thesis, TU of Denmark

LEIFSSON, L.T.; SÆVARSDÓTTIR, H.; SIGURDSSON, S.T.; VÉSTEINSSON, A. (2008), *Grey-box modeling of an ocean vessel for operational optimization.* Simulation Modelling Practice and Theory 16(8), pp.923–932

MOLLAND, A.F. (2008), *The maritime engineering reference handbook*. Butterworth-Heinemann

PEDERSEN, B.P., LARSEN, J. (2009), *Prediction of full-scale propulsion power using artificial neural networks*, 8th Int. Conf. Computer and IT Applications in the Maritime Industries, Budapest, pp.537–550

RUGGIERO, V.; FILARDI, V.; CUCINOTTA, F. (2007), *Mesh size influence in a CFD code on resistance evaluation of a motor yacht*, 6th Int. Conf. Computer and IT Applications in the Maritime Industries, Cortona, pp.458–466

SVARER, C.; HANSEN, L.K.; LARSEN, J. (1993), *On design and evaluation of tapped-delay neural network architectures*, IEEE Int. Conf. on Neural Networks

TOWNSIN, R. (2003), *The ship hull fouling penalty*, Biofouling 19, pp.9–15

WEIGEND, A.S; HUBERMAN, B.A.; RUMELHART, D.E. (1990), *Predicting the future: A connectionist approach.* Int. J. Neural Systems 1/3, pp.193-209