electronics

MDPI

Article

Black-Box Marine Vehicle Identification with Regression Techniques for Random Manoeuvres

Raul Moreno †, **David Moreno-Salinas** *,† and **Joaquin Aranda** †

Department of Computer Science and Automatic Control, University of Distance Learning Education, UNED Madrid, 28040 Madrid, Spain; raul.moreno.salinas@gmail.com (R.M.); jaranda@dia.uned.es (J.A.)

* Correspondence: dmoreno@dia.uned.es

† The authors contributed equally to this work.

Received: 8 April 2019; Accepted: 27 April 2019; Published: 30 April 2019

Abstract: As a critical step to efficiently design control structures, system identification is concerned with building models of dynamical systems from observed input–output data. In this paper, a number of regression techniques are used for black-box marine system identification of a scale ship. Unlike other works that train the models using specific manoeuvres, in this work the data have been collected from several random manoeuvres and trajectories. Therefore, the aim is to develop general and robust mathematical models using real experimental data from random movements. The techniques used in this work are ridge, kernel ridge and symbolic regression, and the results show that machine learning techniques are robust approaches to model surface marine vehicles, even providing interpretable results in closed form equations using techniques such as symbolic regression.

Keywords: Marine identification; ridge regression; symbolic regression; modelling

1. Introduction

System identification, also known in industrial design as *surrogate modelling*, is one of the most important phases in multiple engineering areas, where reliable mathematical models, and tools are needed for a wide range of applications [1,2]. The goal of system identification is to build mathematical models that, given the same inputs, yield the best fit between the measured response of the systems and the model outputs. Specifically, identification of mathematical models is of great importance for an efficient control design for autonomous vehicles, and therefore, this paper focuses on the identification of marine vehicles for control purposes.

It is important to remark that a real experiment in the sea involves a lot of people and infrastructures, with a high cost in terms of time and money. In this sense, the development of efficient models may provide accurate simulations which help to tackle real experiments with higher confidence, avoiding the realization of multiple and costly tests at sea. These models can be used to simulate new control systems and predict the behaviour of the real vehicles with high accuracy before real tests are carried out.

In control system applications, a closed-loop system is composed of the physical system or process, the feedback loop and the controller, as shown in Figure 1. The objective is to obtain a good performance of the closed-loop control system based on simulations carried out with the computed model, for example [3] where a course-keeping algorithm based on a knowledge base is developed for control tasks that can be applied for the simulation of nonlinear ship models. Therefore, the identification of the mathematical model should be sufficiently accurate and easily implementable so that these simulations can provide useful information. Although the performance of the vehicle can be improved when the control system is designed and implemented, the better the performance of the model computed in the identification phase, the better the performance of the closed-loop system once it is translated to the real world.

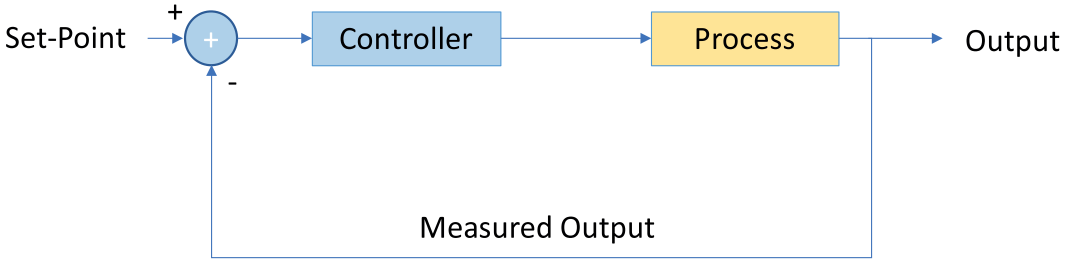

Figure 1. Diagram of a control loop.

A good survey on identification can be found in [4,5]. It is possible to find different kinds of models in the literature, such as the gray-box models, which estimate parameters from data given a generic model structure, i.e., the model is based on the physics of the system and data, or the black-box models, which determine the model structure and estimate the parameters from data. For a survey on black-box identification, the reader is referred to [6]. There are plenty of techniques that may be applied for black-box identification, for example, in the present work Symbolic Regression, Kernel Ridge Regression and also models computed with Support Vector Machines are considered. Among other techniques, an important technique for black-box system identification is the high resolution spectral estimation technique, where the modelling of time series is made over a limited window size of measurements. Different divergence measures can be employed, such as the beta divergence [7], which is used to develop highly robust models with respect to outliers in learning algorithms. These information-theoretic divergences are commonly used in fields such as pattern recognition, sound speech analysis, or image processing [8]. In this paper, the well known Root Mean Square Error (RMSE), which is closely related to L_2-norm, has been considered as a cost function to optimize the models.

Much work has been carried out in marine vehicle identification for different types of marine vehicles, for example, for surface marine vessels some of the most popular are Nomoto and Blanke models [9]. For the computation of an accurate model, a large experimental dataset is usually needed so that the model computed may be able to characterize the hydrodynamics of the vehicle. In this sense, the identification of an accurate model may be a very complex task that entails great computational effort. There are different methods to estimate these models in the state-of-the-art such as Kalman Filter (KF) [10], Extended Kalman Filter (EKF) [11], or AutoRegresive Moving Average (ARMA) [12]. For some other interesting related works, the reader is referred to [9,13–15], and the references therein.

Machine learning (ML) techniques have recently gained popularity in system identification and control problems, for example, we can find some works using Support Vector Machines (SVM) [16], Support Vector Regression (SVR) for linear regression models [17], the application of SVM to time series modelling [18], and Least-Squares Support Vector Machines (LS-SVM) for nonlinear system identification [19]. We can also find interesting works using Gaussian Processes for identification, such as [20], where Gaussian Processes are used to model non-linear dynamics systems [21], in which model identification is applied including prior knowledge as local linear models, or [22], where a new approach of Gaussian Processes regression is described to handle large datasets for approximation. For a survey on Gaussian Processes identification, see [23], and for an interesting survey on kernel methods for system identification, the reader is referred to [24].

We can find some works where neural networks have been employed in marine system identification [25–27], and other interesting works include the training of Abkowitz models using LS-SVM [28] and ϵ-SVM [29]. However, most of these works use synthetic data without noise, simulating basic trajectories for training and testing the models, without proving the efficiency of the model in real experiments.

As far as the authors know, the state-of-the-art shows only a few works using real datasets for marine vehicle identification with ML techniques. Some of the works that have modelled surface marine vehicles using real experimental datasets are, for example, [30] where LS-SVM is used for training a Nomoto second-order linear model, and a Blanke model in [31]. In [32], SVM is used for modelling a torpedo AUV, and symbolic regression is applied in [33]. Finally, in [34] a surface marine vehicle is modelled using Kernel Ridge Regression with Confidence Machine (KRR-CM). However, they use a small number of basic and simple trajectories to train and test the models.

In this work, the aim is to provide general and robust black-box models of a ship by applying different regression techniques. Unlike other studies in the state-of-the-art that employ synthetic data without noise or small experimental datasets of basic movements, the work described in this paper uses experimental data from different random manoeuvres and trajectories. In these previous works, models have been trained using specific trajectories, such as evolution circles or Zig Zags, and have been tested on the same type of movement. In contrast, in our study, we train and test the model with different random trajectories to ensure its robustness. Among the different techniques available from the Machine Learning field, we have selected the well-known Ridge Regression (RR) technique, since it is a basic regression technique widely used in many applications with good results. Alternatively, as an extension of RR, we have also applied Kernel Ridge Regression (KRR) with two different kernels: Radial Basis Function (RBF) and polynomial. The last approach applied is Symbolic Regression (SR), which is based on Genetic Programming (GP), obtaining a full expression for the equations of the model. While not as simple as RR models, it is more interpretable than kernel methods.

This paper is organized as follows. In Section 2, the experimental system and dataset are presented. Section 3 explains the techniques used to train the different models. In particular, parameter selection for RR and KRR are detailed, and the scheme followed for the SR models is described. The results of the different models are shown in Section 4, and finally the conclusions and future work are discussed in Section 5.

2. Surface Marine Vehicle Identification

This section introduces the formulation that will be followed for the model computation, and the experimental dataset used in this work.

2.1. Experimental System and Dataset

The experimental data have been collected from different experiments and tests from a scale ship of an operational vessel in a 1/16.95 scale, as shown in Table 1 and Figure 2.

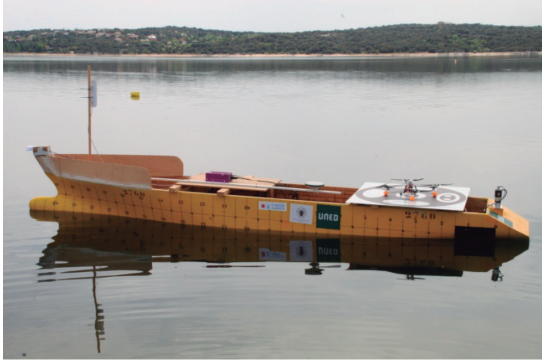

Figure 2. Scale ship used for the experiments.

The vehicle has been designed, constructed and tested before deployment by CEHIPAR (Canal of Hydrodynamic Experiences of El Pardo), a research and development centre adjoined to the Defence Ministry of Spain. This centre has designed and constructed the ship following marine vehicle construction standards and protocols, and has tested the vehicle before deployment, guaranteeing that it accurately reproduces, in scale, the behaviour of a real ship in open-loop.

This ship is an underactuated vehicle, i.e., the number of actuators is lower than the degrees of freedom. It is composed of a DC electric motor, which provides the turning speed of the propeller to control the surge speed, and by a servo motor, which controls the rudder angle, both commanded either using a long-range WiFi connection between the ship and the control station or computed by the control law programmed in the on-board computer. Therefore, there are two input signals, namely commanded speed (tension applied to the DC motor) and rudder angle, and different output signals measured from the Inertial Measurement Unit (IMU) on-board the ship, namely, surge speed, sway speed and yaw rate.

The data collected are significantly different from previous experiments and works in marine identification. Instead of using typical and basic manoeuvres such as Zig-Zags or evolution circles, these experiments have described different random manoeuvres and trajectories by giving different and random values to the rudder angle and commanded speed. The dataset is composed of three different sets with a sampling rate of 0.2 s: a training set (8752 samples), a validation set (9401 samples) and a test set (6550 samples), as shown in Figure 3.

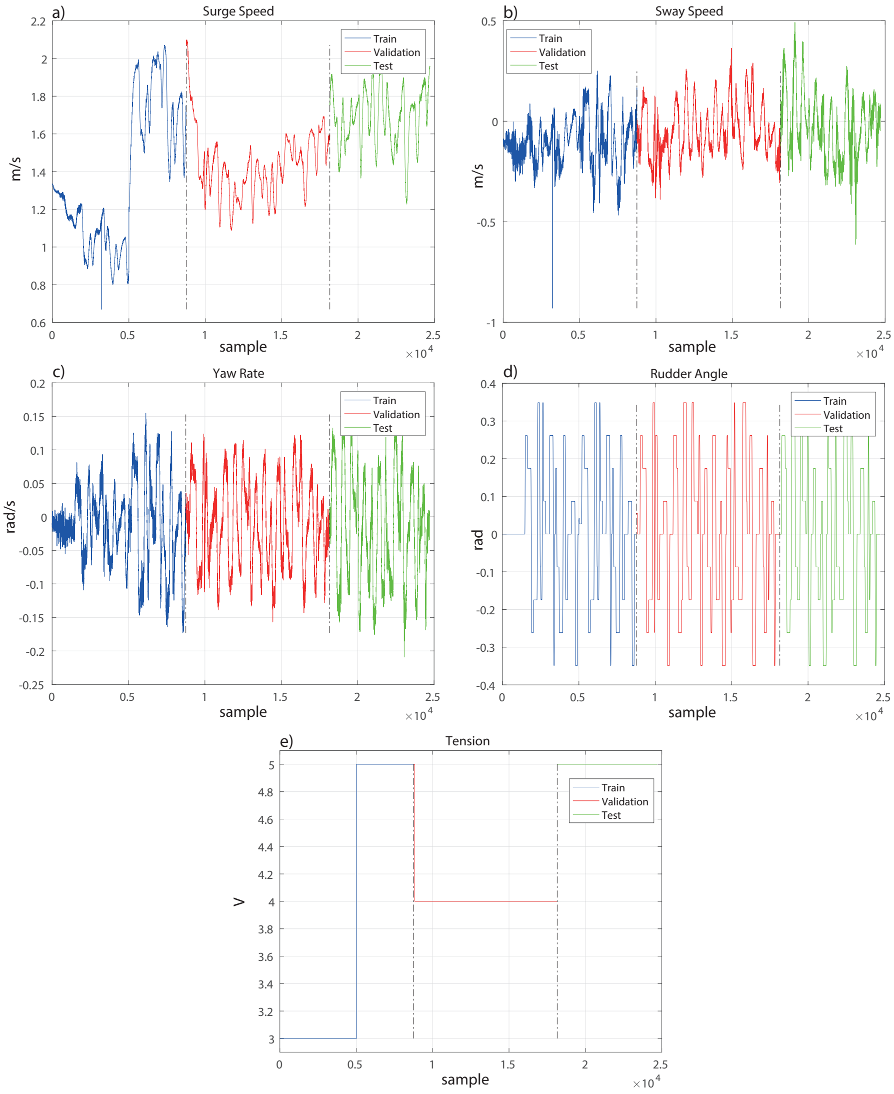

Figure 3. Training (blue), validation (red) and test (green) sets: (**a**) surge speed, (**b**) sway speed, (**c**) yaw rate, (**d**) rudder angle, and (**e**) tension of the motor.

Table 1. Parameters and dimensions of the vessel and the scale ship.

Parameter	**Vessel [m]**	**Scale Ship [m]**
Length between perpendiculars (L_{pp})	74.40	4.389
Maximum beam (B)	14.20	0.838
Mean depth to the top deck	9.05	0.534
Design draught (T_m)	6.30	0.372

2.2. Model Formulation

The cost of experimental tests in marine systems can be very high since it involves many people, transport, material, infrastructures, etc. Hence, an efficient mathematical model is essential to simulate the real system with high accuracy without constraints. This work uses different regression techniques to obtain black-box identification models, i.e., models that describe accurately the relationship between the inputs and outputs without assumptions or constraints on the mathematical model structures. The motivation of this work is to obtain simple and robust models joining the black-box identification and experimental data from random trajectories, and using different ML techniques.

The formulation proposed allows for the computation of the differential equations of the speeds (surge, sway and yaw rate) in the following form:

$$\dot{u}_t = f_u(u_t, v_t, r_t, \delta_t, \Delta_t) \tag{1}$$

$$\dot{v}_t = f_v(u_t, v_t, r_t, \delta_t, \Delta_t) \tag{2}$$

$$\dot{r}_t = f_r(u_t, v_t, r_t, \delta_t, \Delta_t) \tag{3}$$

where u, v, r are the surge speed, sway speed and yaw rate, respectively; δ is the rudder angle and Δ is the tension applied to the motor. The measurements at time step t are used to compute the accelerations and then, to predict the outputs at time step $t+1$. As mentioned above, the target variables to model are the differential equations of the speeds, $\dot{u}, \dot{v}, \dot{r}$. These accelerations are computed as the discrete time derivative of the speed variables from raw data (Equation (4)):

$$\dot{\xi}_t = \frac{\xi_t - \xi_{t-1}}{T}, \tag{4}$$

where T is the sampling rate of 0.2s, and $\xi = u, v, r$. There are more accurate approximations to avoid the numerical approximation errors, such as the free-derivative learning method, which avoids the use of numerical derivatives and its associated errors [35]. However, the numerical approximation in (4) is widely used in many engineering applications, and it is accurate enough for the problem at hand given the slow dynamics of the surface vehicle, and the accurate measurements of the speeds provided by the IMU on-board the ship.

The training process minimizes an appropriately defined cost function to find the best fitting of target data independently for the three models of the speeds. The cost function employed for the computation of the models will be the Root Mean Square Error (RMSE), i.e., the squared error between the predicted value by the model and the real value for each differential equation:

$$RMSE_u = \left(\sum_{t=1}^{N} \frac{(\hat{\dot{u}}_t - \dot{u}_t)^2}{N} \right)^{1/2},$$

$$RMSE_v = \left(\sum_{t=1}^{N} \frac{(\hat{\dot{v}}_t - \dot{v}_t)^2}{N} \right)^{1/2},$$

$$RMSE_r = \left(\sum_{t=1}^{N} \frac{(\hat{r}_t - \dot{r}_t)^2}{N} \right)^{1/2},$$

where $\hat{u}_t$, $\hat{v}_t$, and $\hat{r}_t$ are the predicted values; $\dot{u}_t$, $\dot{v}_t$, and $\dot{r}_t$ are the real measured values, and N is the number of samples used.

However, the validation and test are applied following a recursive regression, i.e., the commanded variables (δ and Δ) are provided in every step but the state variables (u, v, r) correspond to the predicted values in the previous step $(\hat{u}, \hat{v}, \hat{r})$. The recursive regression process, also known as freerun simulation, is shown in Figure 4. The inputs for all the models at $t = 0$ are $u_0, v_0, r_0, \delta_0, \Delta_0$, where the speed variables u_0, v_0, r_0 come from the initial measurement of the sensors (IMU), and δ, Δ are the control input variables given to the models for every time step $t \geq 0$. However, the speed inputs $\hat{u}_t, \hat{v}_t, \hat{r}_t$ for $t > 0$ will be the predicted values from the previous time step.

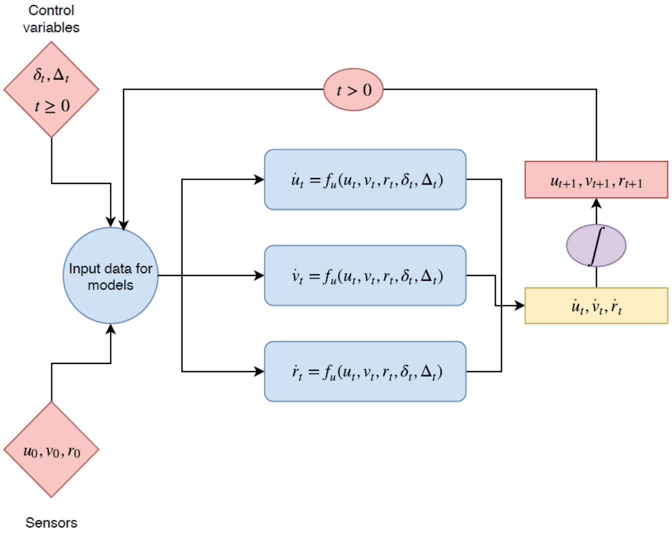
Control variables
δ_t, Δ_t
$t \geq 0$
$t > 0$
$\dot{u}_t = f_u(u_t, v_t, r_t, \delta_t, \Delta_t)$
$u_{t+1}, v_{t+1}, r_{t+1}$
Input data for models
$\dot{v}_t = f_u(u_t, v_t, r_t, \delta_t, \Delta_t)$
$\int$
$\dot{u}_t, \dot{v}_t, \dot{r}_t$
$\dot{r}_t = f_u(u_t, v_t, r_t, \delta_t, \Delta_t)$
u_0, v_0, r_0
Sensors

Figure 4. Freerun simulation diagram. At time t_0, the input variables are given from the sensors, and for $t > 0$ the speed inputs for $t + 1$ are the predicted values in t.

3. Machine Learning Techniques

In this section, an overview of the regression techniques applied is provided. First, an introduction on Ridge Regression (RR) is given, and then extended to Kernel Ridge Regression (KRR). Finally, Symbolic Regression (SR) using Genetic Programming (GP) is explained. *Notation:* In the following, matrices appear in bold and capital letters, while vectors appear in bold and lower case letters. We denote by $\mathbf{I}$ the identity matrix, and by $\mathbf{X}^T$ the transpose of matrix $\mathbf{X}$.

3.1. Ridge Regression

Ridge Regression (RR) is a well known regression shrinkage method proposed in [36]. The original motivation of this technique is to cope with the presence of too many predictors, i.e., the number of input variables exceeds the number of observations. Although this is not the case for our problem, applying a regularization on the predictor estimation nevertheless helps to achieve a better model. The parameters of standard linear regression can be obtained as $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$, where $\mathbf{X}$ is input data, $\mathbf{X}^T$ is its transpose matrix, and $\mathbf{Y}$ is output data. The estimates depend on $(\mathbf{X}^T\mathbf{X})^{-\mathbf{1}}$ and small changes in $\mathbf{X}$ produce large changes in $(\mathbf{X}^T\mathbf{X})^{-1}$ when $\mathbf{X}^T\mathbf{X}$ is nearly singular. The model estimated may fit the training data but may not fit the test data. The conditioning of the problem can be improved by adding a small constant λ to the diagonal of the matrix $\mathbf{X}^T\mathbf{X}$ before taking its inverse:

$$\hat{\boldsymbol{\beta}}_{RR} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \tag{5}$$

Then, RR is a modified version of least squares that adds an L_2 regularization to the estimators to reduce the variance. This shrinkage parameter λ controls the penalization over the variables ($\lambda \geq 0$). If $\lambda = 0$, we will have least squares, and if $\lambda \gg 0$, the constraints on the variables are very high.

3.2. Kernel Ridge Regression

Dual ridge regression was proposed in [37] using kernels. RR is combined with the kernel trick, i.e., the substitution of dot products in the optimization problem with kernel functions. This allows to transform a non-linear problem into a higher dimensional feature space where the problem is linearly separable. A kernel function is expressed as $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$, with $i, j = 1, \ldots, N$. Applying the kernel trick on the expression of RR, Equation (5), by substituting all $\mathbf{X}$ with $\phi(\mathbf{X})$ yields:

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{Y}, \tag{6}$$

where $\mathbf{K}$ is the kernel matrix. Then, the general form for Kernel Ridge Regression (KRR) becomes:

$$f_{KRR}(\mathbf{x}_j) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_j, \mathbf{x}_i), \tag{7}$$

where $\mathbf{x}_j$ is a sample, α_i with $i = 1, \ldots, N$ are the learned weights, $\mathbf{I}$ is the identity matrix, and $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function.

KRR and Support Vector Regression (SVR) are similar. Both use L_2 regularization but have different loss functions and KRR seems to be faster for medium datasets. The main motivation to use KRR, rather than SVR or LS-SVM, is the good results obtained in a previous work [34], where a KRR Confidence Machine (Conformal Predictors) was applied.

There are plenty of kernel functions available, however we have selected two of the main kernel functions widely used in the literature, namely, Radial Basis Function (RBF) and polynomial. The RBF kernel presents the following form:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma}\right), \tag{8}$$

where $||\cdot||$ denotes the euclidean distance, and the σ parameter should be carefully tuned since it plays a major role in the performance. If σ is overestimated, the exponential would behave linearly, losing its non-linear properties in the higher dimensional space; but if it is underestimated, the kernel will present a lack of regularization and will be sensitive to noise.

The polynomial kernel has the form:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\theta\mathbf{x}_i^T\mathbf{x}_j + c)^p, \tag{9}$$

where the parameters are the slope θ, a constant term c and the polynomial degree p. In general, polynomial kernels behave well for normalized data.

3.3. Symbolic Regression

Genetic Programming (GP) is a supervised learning method based on the idea of biological evolution, i.e., the survival of the fittest individuals [38,39]. In GP, the possible solutions of the problem are presented as individuals who are evaluated using a fitness function to select the best candidates. These candidates are then mixed to combine the genes and generate new solutions in an iterative process. The basic algorithm works as follows:

1. A random initial population is created.
2. The fitness of the members of the population is evaluated.
3. The 'stop if' condition is evaluated (a given accuracy, number of generations, etc.).
4. A next generation of solutions is formed by combining the selected members of the population using different operators (crossover, mutation, reproduction, etc.). Go back to step 2.

The operators basically mix the characteristics of the solutions, called genes, to generate new candidates. The combination of the best individuals along the generations should converge to an optimal solution. Symbolic Regression (SR) is a particular case of GP where the candidates are defined by functions and the objective is to fit empirical data without *a priori* assumptions on the model structure. The fitness function in SR is the accuracy of the individuals fitting the empirical data, in our case RMSE, and the possible solutions are presented in a tree-based structure where nodes represent operators and variables. See [40] for further description of SR. The library used for SR is the toolbox GPTIPS 1.0 in Matlab [41].

Following a similar process to [33], the operations allowed between variables are sum, difference, multiplication, protected division, protected root square and square. More operations can be used but the purpose is to keep the model simple while obtaining a good representation of the system.

In the following section, we have set the population to 600 and the generations to 500, with a tournament selection. The maximum number of genes per individual is fixed to 5 without the computation of the bias term, and the maximum depth of trees is 3.

These parameters could be changed with a large number of possible combinations. However, the values selected are a tradeoff election between very complex and very simple models after running a battery of experiments conducting a search grid over these parameters. By doing this, and with the selected parameters, the models proposed by the SR algorithms provide high accuracy while keeping the model structure relatively simple. For smaller values than those selected (genes or depth tree), the individuals (models) were quite simple, providing a reduced variety of individuals, thus constraining the optimal solutions to a few members. For larger values, the models became quite complex and also encountered the problem of over-fitting, modelling even the noise of the signals. Therefore, the above parameters have been set to the specified values to obtain interpretable models while not being excessively complex.

4. Results

To evaluate the performance of the models, Root Mean Squared Error (RMSE) has been used as a fitness function as mentioned in Section 2.2.

Two different approaches for the parameter selection of KRR with the RBF kernel have been applied. Firstly, each model was optimized independently, the parameters were tuned using the recursive regression only for the variable being optimized while the other variables were set to their true values from the training data, rather than their previous predicted values. In other words, if we are performing a freerun simulation, as shown in Figure 4, the recursive regression uses as initial input $(u_0, v_0, r_0, \delta_0, \Delta_0)$ and for $t > 0$ the inputs used are δ_t, Δ_t from the dataset and the predicted speed values $\hat{u}_t, \hat{v}_t, \hat{r}_t$. However, if we are training the surge speed model $f_u(u, v, r, \delta, \Delta)$, the inputs for $t > 0$ are $\delta_t, \Delta_t, v_t, r_t$ from the dataset and the predicted value for $\hat{u}_t$. This was done for each model f_u, f_v, f_r for the parameter selection. Nevertheless, this approach does not work once the optimal models are used together in a recursive regression for the validation and test.

Therefore, a second approach has been used. The λ_u, λ_v and λ_r parameters and σ_u, σ_v and σ_r hyperparameters selection has been carried out training the three models together by means of an exhaustive grid search. Training data are used to optimize our cost function and compute our model parameters $\boldsymbol{\alpha}$ given specific values of the kernel hyperparameters and λs parameters. These specific parameters are computed as follows: first, a quick search was done to evaluate the range of the parameters where models show a coherent behaviour, i.e., low values for RMSE. Then, an exhaustive grid search for each parameter is performed by fixing the rest of the parameters at the values obtained in the previous process, as shown in Figures 5 and 6. Each model obtained in this search is validated using the validation set, and the parameter values with the best performance are selected. Figure 5a,b show the selection of λ_u keeping fixed the rest of the parameters for the training and validation set. The best value obtained is $\lambda_u = 0.02$, then this parameter is kept fixed for tuning the others in the following optimization of λ_v and λ_r in Figure 5c,d and Figure 5e,f respectively.

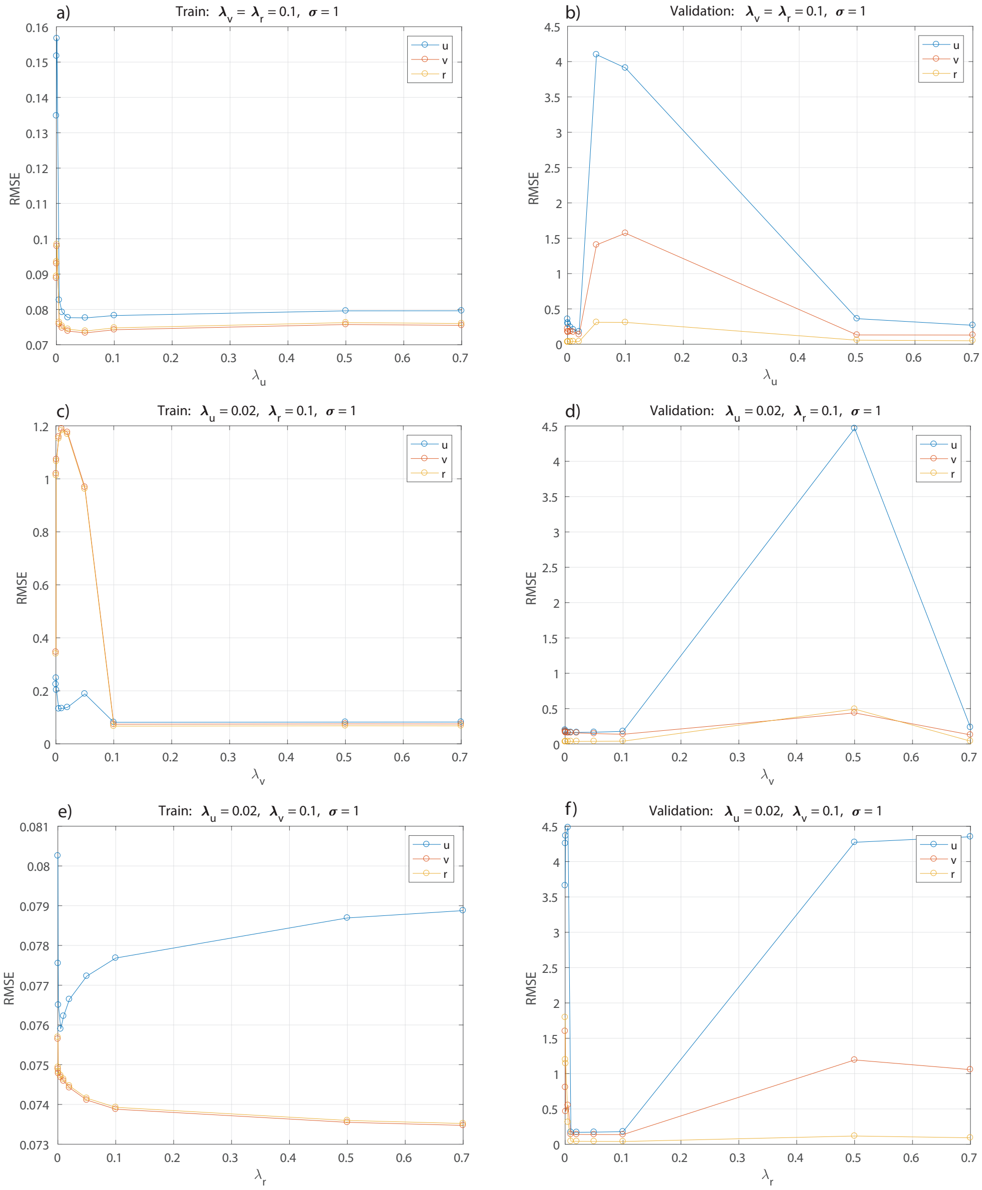

Figure 5. λ-parameter selection for kernel RBF. The units for the RMSE curves in all subplots are m/s for surge speed (u) and sway speed (v), and rad/s for the yaw rate (r).

The best values correspond to $\lambda_u = 0.02, \lambda_v = \lambda_r = 0.1$. Now, keeping these values fixed, a range of values for σ_u, σ_v, and, σ_r is tested; see Figure 6, with the best results obtained for $\sigma_u = 1$. For simplification, only the result for σ_u is shown as the analysis applies the same value for σ_v and σ_r, since changing the parameter for each model does not show a significant difference.

The polynomial kernel shows acceptable results with degrees $p = 1, p = 2$ using constant parameters $c = 1$ and $\theta = 1$. The values used for $\lambda_u, \lambda_v, \lambda_r$ are the same as the best result from the previous analysis for kernel RBF.

The results are shown in Table 2 with the best model obtained from the parameter selection of kernel RBF, polynomial kernel and SR. The polynomial kernel shows slightly better results than RBF for both cases, $p = 1$ and $p = 2$. The case of $p = 1$ corresponds to the standard RR approach.

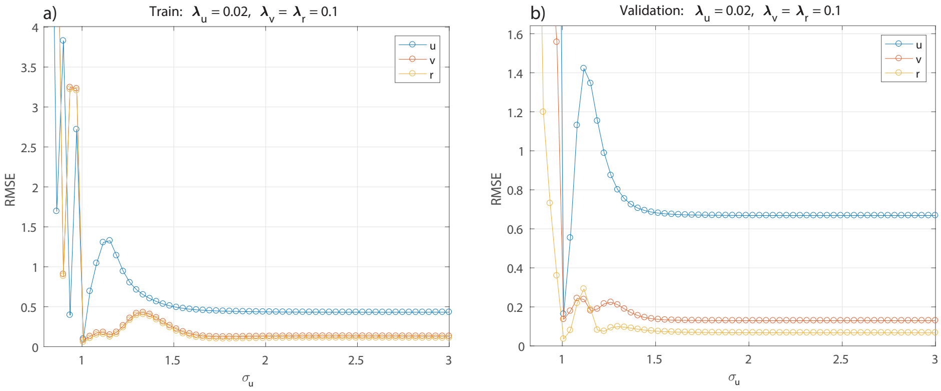

Figure 6. σ-parameter selection for kernel RBF. The units for the RMSE curves in all subplots are m/s for surge speed (u) and sway speed (v), and rad/s for the yaw rate (r).

Table 2. Best models for KRR and SR.

Model	**RMSE Train [u-v-r]**	**RMSE Validation [u-v-r]**
RBF	0.0777 − 0.0739 − 0.0137	0.1784 − 0.1372 − 0.0391
Polynomial $p = 1$	0.1059 − 0.0793 − 0.0209	0.1653 − 0.1050 − 0.0154
Polynomial $p = 2$	0.0773 − 0.0765 − 0.0146	0.1645 − 0.1148 − 0.0181
Symbolic Regression	0.2614 − 0.2422 − 0.1085	0.3482 − 0.2861 − 0.1150

The last model is computed applying GP with symbolic regression, where the following equations are obtained:

$$\dot{u} = 0.01779 \cdot \delta + 0.005287 \cdot \Delta - 0.01779\sqrt{|v|} - 0.001518 \cdot u^3 - \frac{0.7361 \cdot u \cdot r \cdot \delta}{\Delta}, \tag{10}$$

$$\dot{v} = 0.0776 \cdot r - 0.1217 \cdot v + 0.108 \cdot u \cdot r - \frac{0.0776 \cdot u}{\Delta} + 0.5737 \cdot r^2, \tag{11}$$

$$\dot{r} = 0.1862 \cdot \delta\sqrt{|u|} - 0.06615 \cdot \delta\sqrt{|\Delta|} - \frac{0.01165 \cdot v \cdot (v - 9.974)}{\Delta} - \frac{3.585 \cdot u \cdot r}{\Delta^2}. \tag{12}$$

These equations show a dependency that is inversely proportional to the applied tension. It is important to note that the operation applied is a protected division, i.e., the tension variable will never be 0. Then, this model is useful for manoeuvres when the tension applied to the motor is $\Delta \neq 0$, which makes perfect sense since there is no interest in modelling the boat with $\Delta = 0$, and very low values of Δ are not considered, since a minimum tension is needed to start or keep running the motor.

Table 3 shows the performance of the two best models obtained for validation and test sets. These models correspond to the polynomial kernel with degree $p = 2$ and the best model obtained with SR following the process explained in Section 3.3. It can be seen that both results are really close and the polynomial kernel with $p = 2$ obtains slightly better performance in the validation and test sets. Although the difference between the models and the real data is not so small, it is important to consider that we are modelling experimental data. The noise has not been removed from the data, since we try to simulate the real experiment as close as possible. Thus, if the performance had been very accurate, it would indicate that we are over-fitting the noise in the data.

Therefore, since we are not interested in modelling the experimental data completely to avoid fitting the noise, and we just search for fitting the real movement of the ship, the models should follow with small error the trend of the real movement. Under these considerations, both models are really close, as shown in Figures 7 and 8. For the validation set (see Figure 7), the surge speed (u) and sway speed (v) are very similar for both models. In the case of the yaw rate (r), both models are very efficient. For the test set (see Figure 8), we can see a similar result. However, the polynomial kernel shows a peak where only one sample has been extremely poorly fitted. Despite this, the models are robust and recover the trend of the real movement.

As mentioned, there are not many works with experimental data for marine vehicle identification, and a comparison with models computed from synthetic data without noise or basic manoeuvres of a different ship would not be a rigorous analysis. A reasonable comparison with models from the state-of-the-art would be to compare the models computed in the present work with those from two works using the same ship and experimental data with basic movements for training them. Therefore, we have considered the models trained in [33] with SR and in [31] with LS-SVM. We will call these models, from the state-of-the-art, SOA-SR and SOA-LSSVM, respectively. hlOn one hand, SOA-SR is trained using experimental data from a 20/20 degree Zig-Zag manoeuvre and a similar process in the GP setup. On the other hand, SOA-LSSVM uses a Blanke model with LS-SVM and a 20/20 degree Zig-Zag manoeuvre for the training. These models have been tested on the validation and test sets; see Table 3 where SOA-LSSVM shows a worse fitting on both datasets. However, SOA-SR shows results that are slightly worse than those of the best models selected in this work, and following the previous discussion, we analyze the fitting in both datasets in Figures 7 and 8, to show graphically the performance of the model from the state-of-the-art in comparison with the models computed in this paper.

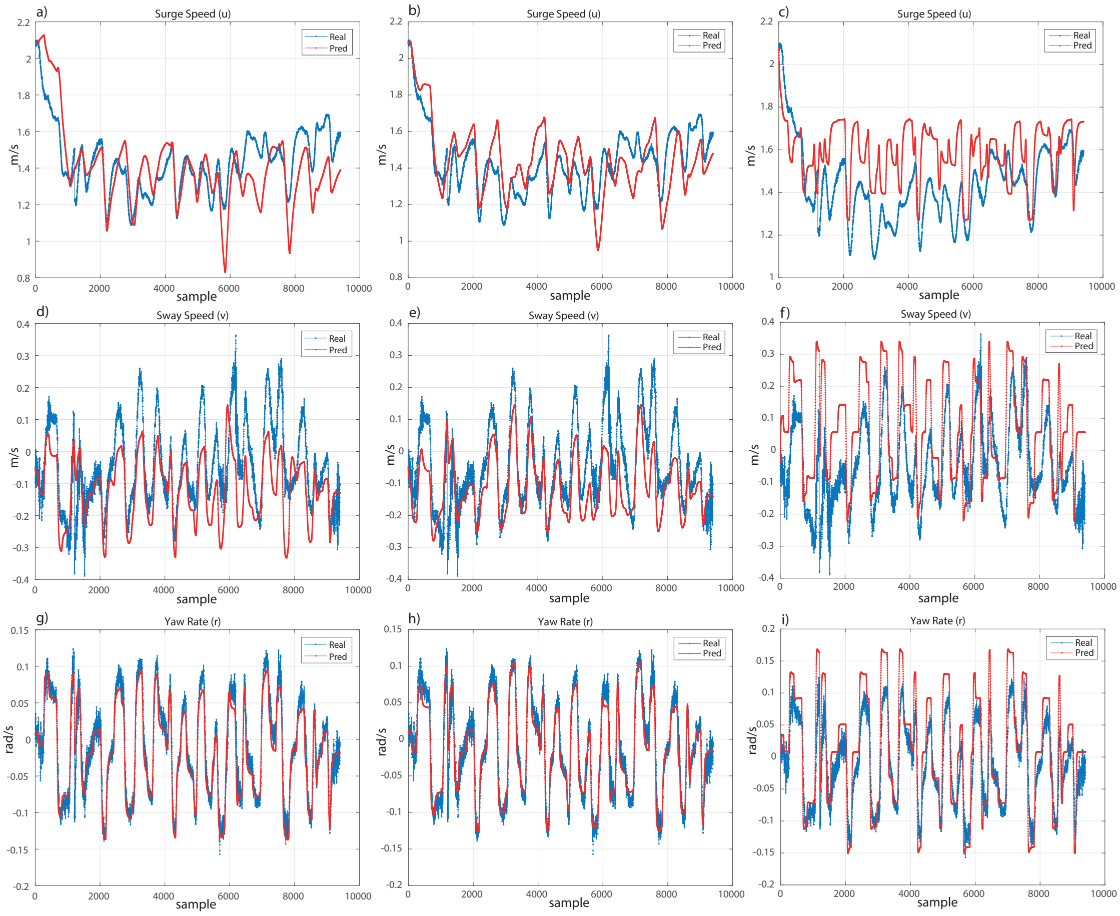

Figure 7. Validation set for the best models with the polynomial kernel (**a**) surge speed, (**d**) sway speed, (**g**) yaw rate; SR (**b**) surge speed, (**e**) sway speed, (**h**) yaw rate; and SOA-SR (**c**) surge speed, (**f**) sway speed, (**i**) yaw rate.

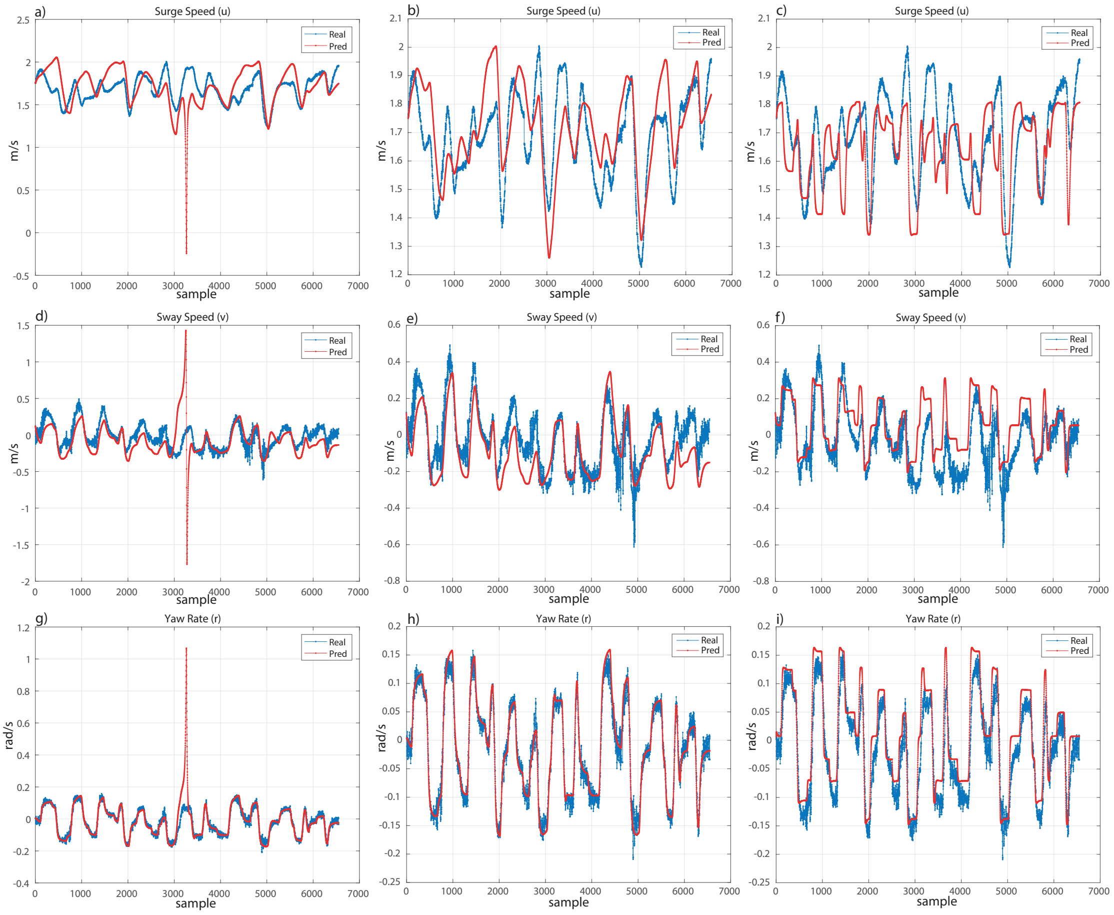

Figure 8. Test set for best models with the polynomial kernel (**a**) surge speed, (**d**) sway speed, (**g**) yaw rate; SR (**b**) surge speed, (**e**) sway speed, (**h**) yaw rate; and SOA-SR (**c**) surge speed, (**f**) sway speed, (**i**) yaw rate.

Table 3. Best models Validation and Test.

Model	**RMSE Validation [u-v-r]**	**RMSE Test [u-v-r]**
Polynomial $p = 2$	0.1645 – 0.1148 – 0.0181	0.2249 – 0.2211 – 0.0654
Symbolic Regression	0.3482 – 0.2861 – 0.1150	0.3279 – 0.3153 – 0.1077
SOA-SR	0.4255 – 0.3700 – 0.1895	0.3403 – 0.3302 – 0.1788
SOA-LSSVM (Blanke model)	0.8835 – 0.3166 – 0.3963	0.7350 – 0.2936 – 0.4169

5. Conclusions and Future Work

This paper showed alternatives for black-box marine identification using different regression techniques. The main objective was to develop robust and simple models from experimental data of a real ship using random manoeuvres instead of predefined and simple manoeuvres. Three different datasets from different random trajectories have been used for training, validation and test, respectively. On the one hand, we have used kernel ridge regression with RBF and polynomial kernels with their respective parameter selection to train the models. The standard RR has been treated as a particular case of polynomial kernel with $p = 1$. On the other hand, GP with SR has been used with some constraints on the operations and on the depth of the trees and nodes to keep the complexity of the model low and, at the same time, obtain a good performance. Among all the models obtained, SR and KRR with the polynomial kernel with $p = 2$ showed the best results. Although they present similar performance, we conclude that SR shows a slightly better approach since the equation form is not constrained to a polynomial, and it gives more flexibility to obtain different equations and to apply different operations. Furthermore, being able to generalize better a wider range of manoeuvres, the models provided seem to be more robust than classical models from the state-of-the-art such as SOA-LSSVM which is based on a Blanke model. Although the models presented in this work obtain better results than the SOA-SR model, it is important to note that this model obtains satisfactory results testing random trajectories. This shows how SR is an effective approach to obtain black-box identification models, and deserves further experimentation and efforts to further improve the modelling accuracy and its generalization performance.

An extended analysis on black-box identification models will be carried out as future work, where a wider variety of different approaches will be tested. Different techniques can be applied for black-box identification, and identifying which ones are more suitable for this problem would help to obtain more efficient models. The models presented in this work are formulated as differential equations of the speed variables, however it is interesting to develop discrete models based only on the speed variables. In addition, the identification is carried out using only the previous time step to predict the new incoming sample. The application of models trained with different temporal window sizes as input should be studied in the future.

Author Contributions: Conceptualization, R.M., D.M.-S. and J.A.; Formal analysis, R.M. and D.M.-S.; Investigation, R.M., D.M.-S. and J.A.; Methodology, D.M.-S.; Resources, J.A.; Supervision, J.A.; Validation, R.M.; Writing–original draft, R.M. and D.M.-S.; Writing–review & editing, D.M.-S. and J.A.

Funding: This work has been funded by Spanish Ministry of Economy and Competitiveness under projects CICYT DPI2014-55932-C2-2-R and CICYT DPI2017-84259-C2-2-R.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Forrester, A.; Sobester, A.; Kane, A. *Engineering Design Via Surrogate Modelling: A Practical Guide*; Jhon Wiley & and Sons: Hoboken, NJ, USA, 2008.
2. Belyaev, M.; Burnaev, E.V., Kapushev, E.; Panov, M.; Prikhodko, P.; Vetrov, D.; Yarotsky, D. GTApprox: Surrogate modeling for industrial design. *Adv. Eng. Softw.* **2016**, *102*, 29–39. [CrossRef]
3. Borkowski, P. and Zwierzewicz,Z. Ship Course-Keeping Algorithm Based On Knowledge Base. *Intell. Autom. Soft Comput.* **2011**, *17*, 149–163. [CrossRef]
4. Ljung, L. *System Identification: Theory for the User. Upper Saddle River*; Prentice-Hall: Upper Saddle River, NJ, USA, 1999; ISBN 0-13-656695-2.
5. Ljung, L. Identification of Nonlinear Systems. In Proceedings of the International Conference on Control, Automation, Robotics and Vision, Setúbal, Portugal, 1–5 August 2006.
6. Juditsky, A.; Hjalmarsson, H.; Benveniste, A.; Delyon, B.; Ljung, L.; Sjoberg, J.; Zhang, Q. Nonlinear black-box modeling in system identification: Mathematical foundations. *Automatica* **1995**, *31*, 1724–1750. [CrossRef]
7. Zorzi, M. A New Family of High-Resolution Multivariate Spectral Estimators. *IEEE Trans. Autom. Control* **2014**, *59*, 892–904. [CrossRef]
8. Cichocki, A.; Amari, S.-I. Families of Alpha- Beta- and Gamma- Divergences: Flexible and Robust Measures of Similarities. *Entropy* **2010**, *12*, 1532–1568. [CrossRef]
9. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control;* Wiley: London, UK, 2011; ISBN 978-1119991496.
10. Abkowitz, M.A. Measurements of hydrodynamic characteristic from ship manoeuvring trials by system identification. *Trans. Soc. Nav. Archit. Mar. Eng.* **1980**, *88*, 283–318.
11. Fossen, T.I.; Sagatun, S.I.; Sorensen, A.J. Identification of dynamically positioned ships. *Identif. Control* **1996**, *17*, 153–165. [CrossRef]
12. Velasco, F.J.; Revestido, E., López, E.; Moyano, E. Identification for a Heading Autopilot of an Autonomous In-Scale Fast Ferry. *IEEE J. Ocean. Eng.* **2013**, *38*, 263–274. [CrossRef]
13. Caccia, M.; Bruzzone, G.; Bono, R. A practical approach to modelling and identification of small autonomous surface craft. *IEEE J. Ocean. Eng.* **2008**, *33*, 133–145. [CrossRef]
14. Perez, T.; Sørensen, A.J.; Blanke, M. Marine Vessel Models in Changing Operational Conditions—A Tutorial. *IFAC Proc. Vol.* **2006**, *39*, 309–314. [CrossRef]

15. De la Cruz, J.M., Aranda, J., & Girón, J.M. Automática Marina: Una revisión desde el punto de vista de control. *Rev. Iberoam. Autom. Inform. Ind.* **2012**, *9*, 205–218.
16. Drezet, P.M.L.; Harrison, R.F. Support Vector Machines for system identification. In Proceedings of the International Conference on Control '98, Beijing, China, 18-21 August 1998.
17. Adachi, S.; Ogawa, T. A new system identification method based on support vector machines. In Proceedings of the IFAC Workshop Adaptation and Learning in Control and Signal Processing, Cernobbio-Como, Italy, 29–31 August 2001.
18. Jemwa, G.T.; Aldricht, C. Non-linear system identification of an autocatalytic reactor using least squares support vector machines. *J. S. Afr. Inst. Min. Metall.* **2003**, *103*, 119–126.
19. Wang, X.D.; Ye, M.D. Nonlinear dynamic system identification using Least Squares Support Vector Machine Regression. In Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, China, 26–29 August 2004.
20. Kocijan, J.; Girard, A.; Banko, B.; Murray-Smith, R. Dynamic systems identification with Gaussian processes. *Math. Comput. Model. Dyn. Syst.* **2007**, *11*, 411–424. [CrossRef]
21. Ažman, K.; Kocijan, J. Dynamical systems identification using Gaussian process models with incorporated local models. *Eng. Appl. Artif. Intell.* **2011**, *24*, 398–408. [CrossRef]
22. Belyaev M., Burnaev E., Kapushev Y. Gaussian Process Regression for Structured Data Sets. In *Statistical Learning and Data Sciences*; Gammerman A., Vovk, V., Papadopoulos, H., Eds.; SLDS 2015, Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2015; Volume 9047.
23. Kocijan, J. Modelling and Control of Dynamic Systems Using Gaussian Process Models. In *Advances in Industrial Control*; Springer International Publishing: Cham, Switzerland, 2016. doi:10.1007/978-3-319-21021-6_2.
24. Pillonetto, G.; Dinuzzo, F.; Chen, T.; De Nicolao, G.; Ljung, L. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica* **2014**, *50*, 657–682. [CrossRef]
25. Haddara, M.; Wang. Y. Parametric identification of manoeuvring models for ships. *Int. Shipbuild. Prog.* **1999**, *46*, 5–27.
26. Haddara, M.R.; Xu, J.S. On the identification of ship coupled heave-pitch motions using neural networks. *Ocean. Eng.* **1999**, *26*, 381–400. [CrossRef]
27. Mahfouz, A.B. Identification of the nonlinear ship rolling motion equation using the measured response at sea. *Ocean. Eng.* **2004**, *31*, 2139–2156. [CrossRef]
28. Luo, W.L.; Zou Z.J. Parametric identification of ship manoeuvring models by using Support Vector Machines. *J. Ship Res.* **2009**, *53*, 19–30.
29. Zhang, X.G., Zou Z.J. Identification of Abkowitz model for ship manoeuvring motion using ϵ-Support Vector Regression. *J. Hydrodyn.* **2011**, *23*, 353–360. [CrossRef]
30. Moreno-Salinas, D.; Chaos, D.; de la Cruz, J.M.; Aranda, J. Identification of a Surface Marine Vessel Using LS-SVM. *J. Appl. Math.* **2013**, *2013*, 803548. [CrossRef]
31. Moreno-Salinas, D.; Chaos, D.; Besada-Portas, E.; Lopez-Orozco, J. A.; de la Cruz, J.M.; Aranda, J. Semiphysical Modelling of the Nonlinear Dynamics of a Surface Craft with LS-SVM. *Math. Probl. Eng.* **2013**, *2013*, 890120. [CrossRef]
32. Xu, F.; Zou, Z.J.; Yin, J.C.; Cao, J. Identification modeling of underwater vehicles' nonlinear dynamics based on support vector machines. *Ocean. Eng.* **2013**, *67*, 68–76. [CrossRef]
33. Moreno-Salinas, D.; Chaos, D.; Besada-Portas, E.; Lopez-Orozco, J. A.; de la Cruz, J.M.; Aranda, J. Symbolic Regression for Marine Vehicles Identification. *IFAC-PapersOnLine* **2015**, *48*, 210–216. [CrossRef]
34. Moreno-Salinas, D.; Moreno, R.; Pereira, A.; Aranda, J.; De la Cruz, J.M. Modelling of a surface marine vehicle with kernel ridge regression confidence machine. *Appl. Soft Comput.* **2019**, *76*, 237–250. [CrossRef]
35. Romeres, D.; Zorzi, M.; Camoriano, R.; Traversaro, S.; Chiuso, A. Derivative-Free Online Learning of Inverse Dynamics Models. *IEEE Trans. Control. Syst. Technol.* **2019**. [CrossRef]
36. Hoerl, A. E.; Kennard, R. W. Ridge Regression—Biased Estimation for Nonorthogonal Problems. *Technometrics* **1970**, *12 Pt 55*, 55–67. [CrossRef]
37. Saunders, C.; Gammerman, A.; Vovk, V. Ridge Regression Learning Algorithm in Dual Variables. In Proceedings of the Fifteenth International Conference on Machine Learning, New York, NY, USA, 13–18 July 2019; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 1998; pp. 515–521.
38. Koza, J.R. *Genetic Programming*; MIT Press: Cambridge, MA, USA, 1998; ISBN 0-262-11189-6.

39. Langdon, W.B.; Poli, R. *Foundations of Genetic Programming*; Springer: Berlin, Germany, 2002; ISBN 3-540-42451-2.
40. Sette, S.; Boullart, L. Genetic programming: Principles and applications. *Eng. Appl. Artif. Intell.* **2001**, *14* 727–736. [CrossRef]
41. Searson, D.P.; Leahy, D.E.; Willis, M.J. GPTIPS: An open source genetic programming toolbox for multigene symbolic regression. In Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 (IMECS 2010), Hong Kong, China, 17–19 March 2010.