On-line Identification of Autonomous Underwater Vehicles Through Global Derivative-free Optimization

George C. Karras[1], Charalampos P. Bechlioulis[1], Matteo Leonetti[2], Narcís Palomeras[3], Petar Kormushev[2], Kostas J. Kyriakopoulos[1] and Darwin G. Caldwell[2]

***Abstract*— We describe the design and implementation of an on-line identification scheme for Autonomous Underwater Vehicles (AUVs). The proposed method estimates the dynamic parameters of the vehicle based on a global derivative-free optimization algorithm. It is not sensitive to initial conditions, unlike other on-line identification schemes, and does not depend on the differentiability of the model with respect to the parameters. The identification scheme consists of three distinct modules: a) System Excitation, b) Metric Calculator and c) Optimization Algorithm. The System Excitation module sends excitation inputs to the vehicle. The Optimization Algorithm module calculates a candidate parameter vector, which is fed to the Metric Calculator module. The Metric Calculator module evaluates the candidate parameter vector, using a metric based on the residual of the actual and the predicted commands. The predicted commands are calculated utilizing the candidate parameter vector and the vehicle state vector, which is available via a complete navigation module. Then, the metric is directly fed back to the Optimization Algorithm module, and it is used to correct the estimated parameter vector. The procedure continues iteratively until the convergence properties are met. The proposed method is generic, demonstrates quick convergence and does not require a linear formulation of the model with respect to the parameter vector. The applicability and performance of the proposed algorithm is experimentally verified using the AUV Girona500.**

I. INTRODUCTION

Underwater vehicles usually operate under difficult circumstances and perform complex tasks such as ship hull inspection, surveillance of underwater facilities (e.g oil platforms) and handling of underwater equipment (e.g control panels, valves, etc.). These tasks require motion and force control schemes with enhanced robustness. In order to design such schemes, a suitable dynamic model of the vehicle must first be considered and identified. The complexity of the dynamic model may vary depending of the vehicle configuration, actuation capabilities, operating speed and planes of symmetry [1].

System identification can be either an off-line or on-line procedure. Off-line identification involves the acquisition of large amount of input-output data sets, which are appropriately filtered and fed to high computational power processing units. On the contrary, on-line identification is an on-the-fly process, where the estimation of the dynamic parameters occurs concurrently with the data acquisition process during the real-time operation of the system. Thus, the dynamic model of the system can be automatically updated by the estimation algorithm, and modified for optimal performance according to the task and the operational environment.

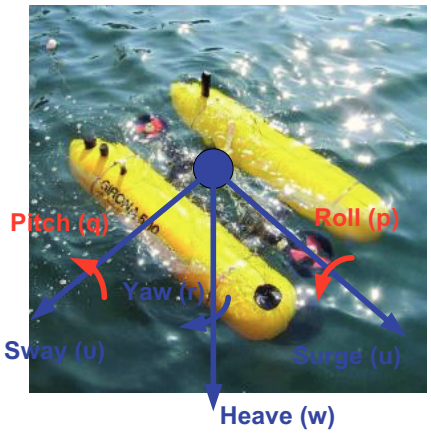

Fig. 1. The Girona500 AUV. Blue color indicates actuated DoFs. Red color indicates under actuated DoFs.

Many works can be found in the literature regarding the identification of underwater robotic vehicles. Ridao et al. [2] presented a comparison between two methods for off-line identification of an Unmanned Underwater Vehicle (UUV). The first method is based on minimization of the acceleration prediction error, while the second one is based on minimization of the velocity prediction error. Caccia et al. presented a Least-Squares (LS) based identification of a lumped parameter model of an open-frame UUV, where the effects of propeller-hull and propeller-propeller interactions are considered. Furthermore, Panagou et al. [3] presented the identification of the dynamic model of an under-actuated underwater vehicle, through an off-line LS technique considering the presence of slowly varying unknown disturbances. Most of the aforementioned off-line techniques can provide sufficiently accurate results, but they usually require a large amount of data, expensive sensor suites and dedicated experimental setups. Generally, off-line identification is a time consuming process that takes place in-situ and if the

This work was supported by the EU funded project PANDORA: Persistent Autonomy through learNing, aDaptation, Observation and ReplAnning", FP7-288273, 2012-2014.

[1]School of Mechanical Engineering, National Technical University of Athens, Athens 15780, Greece {karrasg, chmpechl, kkyria}@mail.ntua.gr

[2]Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego 30, 16163 Genova {matteo.leonetti, petar.kormushev, darwin.caldwell}@iit.it

[3]University of Girona, Edifici Politecnica IV, Campus Montilivi, Girona 17071, Spain {palomer@eia.udg.edu

configuration of the vehicle changes (i.e addition or removal of sensors or tools) the procedure must be repeated.

Smallwood and Whitcomb [4] describe an on-line adaptive technique for the identification of finite dimensional dynamical models of dynamically positioned underwater robotic vehicles. Furthermore, Van de Ven et al. [5] discuss the use of neural networks in the identification of models for underwater vehicles. The most efficient methods for on-line system identification are based on the augmentation of the well known Unscented Kalman Filter (UKF) algorithm for state estimation [6]. A description of UKF's applicability for the state and parameter estimation of non–linear systems, as well as an analytical justification of the superiority of the UKF over the Extended Kalman Filter (EKF) parameter estimation techniques are provided by der Merwe and Wan [7], and VanDyke et al. [8]. Finally, Karras et al. [9] described a dual UKF algorithm for the on-line state and parameter estimation of an underwater vehicle. Although UKF-based algorithms can prove quite efficient, they are extremely sensitive to the initialization of the parameter vector. Indeed, the filter may diverge very fast in case the initial vector is chosen far from the real parameter values. Thus, a rough approximation of the system parameters must be known a priori, to ensure convergence of the filter, which reduces the applicability of these algorithms.

In this paper, we present a novel method for the on-line identification of underwater robotic vehicles. The method is based on a global derivative-free optimization algorithm. The optimization algorithm has three key characteristics that enhance its applicability in identification schemes: it is global, derivative-free and iterative. Since it is a global algorithm, unlike UKF and EKF it does not depend on the initialization and does not require the designer to know a good starting point. On the other hand, if the designer does know a good parameter vector, the algorithm can benefit from it. Since it is derivative-free, unlike LS and EKF, it does not necessitate that the dynamic system should be either linear or differentiable with respect to the parameter vector. Finally, since it is an iterative algorithm, it can be easily integrated in an on-line identification scheme, as the one we implemented for the AUV Girona500 (see Fig. 1).

II. PRELIMINARIES

A. AUV Kinematics and Dynamics

The prior step before system identification is the modeling of the underwater vehicle. According to the standard underwater vehicles' modeling properties [1], the vehicle can be modeled as a rigid body subject to external forces and torques:

$$\begin{array}{c} M\dot{\mathbf{v}}+C(\mathbf{v})\,\mathbf{v}+D(\mathbf{v})\,\mathbf{v}+g(\eta)=\tau \\ \dot{\eta}=J(\eta)\,\mathbf{v} \end{array} \tag{1}$$

where:

- $M = M_{RB} + M_A$, where M_{RB} and M_A are the inertia matrix for a rigid body and added mass respectively;
- $C(\mathbf{v}) = C_{RB}(\mathbf{v}) + C_A(\mathbf{v})$, where $C_{RB}(\mathbf{v})$ and $C_A(\mathbf{v})$ are the Coriolis and centripetal matrix for a rigid body and added mass respectively;
- $D(\mathbf{v}) = D_{quad}(\mathbf{v}) + D_{lin}(\mathbf{v})$, where $D_{quad}(\mathbf{v})$ and $D_{lin}(\mathbf{v})$ are the quadratic and linear drag matrix respectively;
- $g(n)$ is the hydrostatic restoring force vector;
- $J(\eta)$ is the Jacobian matrix transforming the velocities from the body-fixed to the earth-fixed frame;
- $\eta = [x\ y\ z\ \phi\ \theta\ \psi]^T$ is the pose (position and orientation) vector;
- $\mathbf{v} = [u\ v\ w\ p\ q\ r]^T$ is the body velocity vector;
- τ is the input (force/torque) vector.

The dynamic model of the vehicle can be written in the following generic form:

$$\tau = \mathbf{Y}(\eta, \mathbf{v}, \dot{\mathbf{v}}, \pi) \tag{2}$$

where the elements of the vector π involve the unknown parameters of the inertia matrix M, the Coriolis and centrifugal matrix $C(\mathbf{v})$, the drag matrix $D(\mathbf{v})$ as well as of the hydrostatic force vector $g(n)$. Depending on the configuration of the vehicle, existing planes of symmetry and operational speed, the parameter vector π may have fewer elements. Nevertheless, in either case any underwater vehicle can be presented by Eq. 1 or equivalently by Eq. 2.

B. Navigation Module

The navigation module estimates the vehicle position ($[x\ y\ z]$) and linear velocity ($[u\ v\ w]$). The fusion algorithm in charge of merging all the navigation sensor measurements is an extended Kalman filter (EKF) [10]. Vehicle orientation ($[\phi\ \theta\ \psi]$) and angular velocity ($[p\ q\ r]$) are not estimated but directly measured by an Inertial Measurement Unit (IMU). Details of the EKF implementation are presented in the following.

1) State vector: The information to be estimated by the EKF algorithm is stored in the following state vector:

$$\mathbf{x}_k = [x\ y\ z\ u\ v\ w]^T \tag{3}$$

where $([x\ y\ z])$ represents the vehicle position in the world coordinate frame and $([u\ v\ w])$ is the vehicle linear velocity represented in the vehicle coordinate frame.

2) System model: A constant velocity kinematics model is used to determine how the vehicle state will evolve from time $k-1$ to k. The following prediction equation is used:

$$\mathbf{x}_k^- = f(\mathbf{x}_{k-1}, \mathbf{n}_{k-1}, \mathbf{u}_k, t). \tag{4}$$

$$\mathbf{x}_k^- = \begin{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} + \mathbf{R}(\phi_k \theta_k \psi_k) \left(\begin{bmatrix} u_{k-1} \\ v_{k-1} \\ w_{k-1} \end{bmatrix} t + \begin{bmatrix} n_{u_{k-1}} \\ n_{v_{k-1}} \\ n_{w_{k-1}} \end{bmatrix} \frac{t^2}{2} \right) \\ \begin{array}{c} u_{k-1} + n_{u_{k-1}} t \\ v_{k-1} + n_{v_{k-1}} t \\ w_{k-1} + n_{w_{k-1}} t \end{array} \end{bmatrix} \tag{5}$$

where t is the time period, $\mathbf{u} = [\phi\ \theta\ \psi]$ is the control input determining the current vehicle orientation and $\mathbf{n} = [n_u\ n_v\ n_w]$ is a vector of zero-mean white Gaussian noise

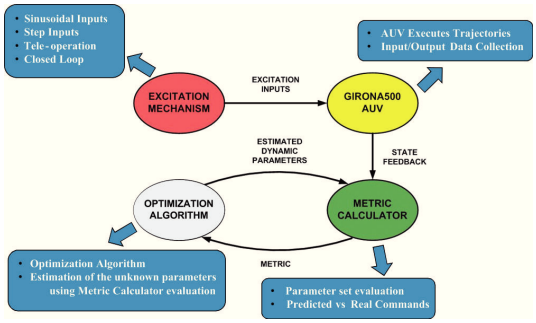

Fig. 2. On-line identification scheme based on an optimization algorithm.

representing vehicle accelerations. Finally, the covariance values of the noise, represented by the system noise matrix $\mathbf{Q} = diag(\sigma_{n_u}^2, \sigma_{n_v}^2, \sigma_{n_w}^2)$, were set empirically.

3) Measurements: Two linear measurement updates are applied in the filter: pose and velocity updates. Both updates follow equation:

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k(z_k - \mathbf{H}\mathbf{x}_k^-) \tag{6}$$

where $\mathbf{K}_k$ is the Kalman gain, z_k is the measurement itself and $\mathbf{H}$ is the observation matrix.

Two sensors provide pose information: the global positioning system (GPS) gives the vehicle position in the plane $(x,\ y)$ while the vehicle is at the surface, and the pressure sensor transforms pressure values into depth measurements (z). Velocity updates are provided by a doppler velocity log (DVL). This sensor is able to measure linear velocities with respect to the sea bottom or the water around the vehicle. Hence, we obtain:

$$z_k = [x_{gps}\ y_{gps}\ z_{depth}\ u_{dvl}\ v_{dvl}\ w_{dvl}] \tag{7}$$

$$\mathbf{H} = I_{6\times 6} \tag{8}$$

where $\mathbf{I}_{6\times 6}$ denotes the 6×6 identity matrix. If only the GPS or the depth sensor data is available, z_k and $\mathbf{H}$ have to be properly arranged.

III. METHODOLOGY

The on-line identification method is based on a global derivative-free optimization algorithm, and unlike other on-line identification schemes it is not sensitive to the initialization vector. The identification scheme consists of three distinct modules: a) System Excitation, b) Metric Calculator and c) Optimization Algorithm. The System Excitation module sends excitation inputs (sinusoidal, ramp, step) to the vehicle in the form of body forces and torques. Concurrently, the Metric Calculator and the Optimization Algorithm modules operate in a closed loop form. More specifically, the Optimization Algorithm calculates a candidate parameter vector that is fed to the Metric Calculator. Then, the Metric Calculator evaluates the candidate parameter vector using a metric based on the residual of the actual and predicted commands. The predicted commands are calculated through Eq. 2 utilizing the candidate parameter vector and the vehicle state vector (output). The state vector is obtained from the aforementioned navigation module (the sensor suite and the optimal estimation state filter). This metric is directly fed to the optimization algorithm to be used for the correction of the estimation and the calculation of a new parameter vector. The procedure continues iteratively until the optimization algorithm meets its convergence properties. The process is depicted in Fig. 2. Each module is described in detail in the following subsections.

A. Excitation Module

This module is responsible for providing the appropriate excitation inputs to the vehicle in the form of body force and torque commands. These commands are transformed to thruster commands through the thruster allocator matrix which is implemented in the control software of the vehicle. The excitation forces and torques τ are of two different types:

- Various step inputs for the identification of the linear and quadratic drag coefficients.
- Various sinusoidal inputs (different amplitudes, frequencies, single sinuses, sum of sinuses) for the identification of the inertial and Coriolis/centripetal coefficients.

Finally, it should be noted that the identification of the unknown parameters can be performed only in the actuated DoFs of the vehicle.

B. Metric Calculator

This module is responsible for evaluating the parameter vector provided by the Optimization Algorithm module. The Metric Calculator module samples the input and output data sets and stores them to separate buffers. Each output set (buffer) contains measurements of the state vectors $\mathbf{n}$, $\mathbf{v}$, $\dot{\mathbf{v}}$, as provided by the navigation module. The output data sets are then filtered using a Savitzky-Golay smoother to remove any noise and prepare the data for the evaluation process. After filtering, the buffers:

- $\mathbf{n^b}$ for the 3D position and orientation
- $\mathbf{v^b}$ for the 3D linear and angular velocities
- $\dot{\mathbf{v}}^\mathbf{b}$ for the 3D linear and angular acceleration
- $\tau^\mathbf{b}$ for the input body forces and torques

are created for each data set. More specifically, for the j^{th} data set the following buffers are created:

$$\begin{aligned}
\mathbf{n_j^b} &= \begin{bmatrix} \mathbf{n_0^j} & \mathbf{n_1^j} & \dots & \mathbf{n_n^j} \end{bmatrix}^\mathbf{T} \\
\mathbf{v_j^b} &= \begin{bmatrix} \mathbf{v_0^j} & \mathbf{v_1^j} & \dots & \mathbf{v_n^j} \end{bmatrix}^\mathbf{T} \\
\dot{\mathbf{v}}_\mathbf{j}^\mathbf{b} &= \begin{bmatrix} \dot{\mathbf{v}}_\mathbf{0}^\mathbf{j} & \dot{\mathbf{v}}_\mathbf{1}^\mathbf{j} & \dots & \dot{\mathbf{v}}_\mathbf{n}^\mathbf{j} \end{bmatrix}^\mathbf{T} \\
\tau_\mathbf{j}^\mathbf{b} &= \begin{bmatrix} \tau_\mathbf{0}^\mathbf{j} & \tau_\mathbf{1}^\mathbf{j} & \dots & \tau_\mathbf{n}^\mathbf{j} \end{bmatrix}^\mathbf{T}
\end{aligned} \tag{9}$$

While collecting the subsequent data set $j+1^{th}$, the previous and the already preprocessed output data set are used in combination with the $\hat{\pi}$ parameter vector estimated by the optimization algorithm, to calculate the predicted input set $\hat{\tau}_\mathbf{j}^\mathbf{b} = \begin{bmatrix} \hat{\tau}_\mathbf{0}^\mathbf{j} & \hat{\tau}_\mathbf{1}^\mathbf{j} & \dots & \hat{\tau}_\mathbf{n}^\mathbf{j} \end{bmatrix}^\mathbf{T}$, where:

$$\begin{aligned}
\hat{\tau}_{\mathbf{0}}^{\mathbf{j}} &= \mathbf{Y}\left(\begin{array}{cccc} \mathbf{n}_{\mathbf{0}}^{\mathbf{j}}, & \mathbf{v}_{\mathbf{0}}^{\mathbf{j}}, & \dot{\mathbf{v}}_{\mathbf{0}}^{\mathbf{j}}, & \hat{\pi} \end{array} \right) \\
\hat{\tau}_{1}^{\mathbf{j}} &= \mathbf{Y}\left(\begin{array}{cccc} \mathbf{n}_{1}^{\mathbf{j}}, & \mathbf{v}_{1}^{\mathbf{j}}, & \dot{\mathbf{v}}_{1}^{\mathbf{j}}, & \hat{\pi} \end{array} \right) \\
&\ldots \\
\hat{\tau}_{n}^{\mathbf{j}} &= \mathbf{Y}\left(\begin{array}{cccc} \mathbf{n}_{n}^{\mathbf{j}}, & \mathbf{v}_{n}^{\mathbf{j}}, & \dot{\mathbf{v}}_{n}^{\mathbf{j}}, & \hat{\pi} \end{array} \right)
\end{aligned} \tag{10}$$

and $\mathbf{Y}(\cdot)$ is the function that describes the dynamics of the vehicle as described in Eq. 2. The evaluation metric for the estimated parameter is given by the norm of the residual:

$$R_j^2 = \left\| \tau_{\mathbf{j}}^{\mathbf{b}} - \hat{\tau}_{\mathbf{j}}^{\mathbf{b}} \right\|^2. \tag{11}$$

Subsequently, this evaluation metric is fed to the Optimization Algorithm module in order to correct the estimated parameter vector.

The evaluation process is significantly accelerated since it is carried out asynchronously with the data acquisition session. Every new parameter estimate is evaluated with the previous input/output datasets while simultaneously new data sets are stored in the background for future evaluations. Thus, the evaluation can be performed immediately after a new parameter estimate vector arrives from the Optimization Algorithm module, which is computationally inexpensive as discussed in the next section.

C. Optimization algorithm

We introduce in system identification a global derivative-free black-box optimization algorithm, borrowed from policy search in reinforcement learning [11]. The algorithm is a composition of a global and a local derivative-free method, designed for the optimization of non-linear, multi-modal, multi-variate functions. We chose to employ this algorithm because of its theoretical properties, for it is: global, derivative-free, and iterative (i.e., suitable for on-line identification). We consider the problem

$$\min_{\theta \in D} J(\theta)$$

of minimizing a cost function J over a convex set $D \subset \mathbb{R}^n$. From this point onward we indicate by n the number of dimensions of the domain. The algorithm is divided into two phases: a controlled global random-search phase, and a deterministic local line-search phase. The algorithm used in the global phase has been introduced by Brachetti et al. [12], and we report it in Algorithm 1.

The global phase is population-based, and the initial population is drawn at random over D (line 3). It is also possible to add to the initial population any good point known in advance by the designer. The population at any time is composed by the best m points ever sampled, where m is a parameter of the algorithm. The bigger m, the more likely the algorithm is to avoid non-global minima. The algorithm terminates when the difference between the best and the worst point of the population is less then the parameter ε. The population evolves by sampling a random family of $n+1$ points from the population itself, and computing the weighted centroid (lines 9–10). The next trial point is computed as a weighted reflection of the worst point of the population with respect to the weighted centroid (line 11).

Algorithm 1 Controlled Random Search Phase

1: Input: a positive integer $m \geq n+1$, $\varepsilon > 0$
2: $k = 0$
3: compute the initial set: $S^k = \{\theta_1^k, \ldots, \theta_m^k\}$ where the points θ_i^k, $i = 1, \ldots, m$ are chosen at random over a box D
4: evaluate J at each point θ_i^k, $i = 1, \ldots, m$.
5: determine the points θ_{max}^k, θ_{min}^k and the values J_{max}^k, J_{max}^k such that: $J_{max}^k = J(\theta_{max}^k) = \max_{\theta \in S^k} J(\theta)$ and $J_{min}^k = J(\theta_{min}^k) = \min_{\theta \in S^k} J(\theta)$
6: **if** $J_{max}^k - J_{min}^k \leq \varepsilon$ **then**
7: STOP
8: **end if**
9: choose at random $n+1$ points $\theta_{i_0}^k, \theta_{i_1}^k, \ldots, \theta_{i_n}^k$ over S^k, where $J(\theta_{i_0}^k) \geq J(\theta_{i_j}^k), \quad j = 1, \ldots, n$
10: determine the centroid $c^k = \sum_{j=0}^n w_j^k \theta_{i_j}^k$
11: determine the trial point $\bar{\theta}^k$ given by

$$\bar{\theta}^k = c^k - \alpha^k (\theta_{i_0}^k - c^k)$$

where

$$w_j^k = \frac{\eta_j^k}{\sum_{r=0}^n \eta_r^k}, \qquad \eta_j^k = \frac{1}{J(\theta_{i_j}^k) - J_{min}^k + \phi^k},$$

$$\alpha^k = 1 - \frac{J(\theta_{i_0}^k) - \sum_{j=0}^n w_j^k J(\theta_{i_j}^k)}{J_{max}^k - J_{min}^k + \phi^k}$$

and

$$\phi^k = n \frac{(J_{max}^k - J_{min}^k)^2}{J_{max}^0 - J_{min}^0};$$

12: **if** $\bar{\theta}^k \notin D$ **then**
13: go to 9
14: **else**
15: compute $J(\bar{\theta}^k)$
16: **end if**
17: **if** $J(\bar{\theta}^k) \geq J_{max}^k$ **then**
18: $S^{k+1} = S^k$, $k = k+1$, go to 9
19: **else**
20: $S^{k+1} = S^k \cup \{\bar{\theta}^k\} - \{\theta_{max}^k\}$, $k = k+1$, go to 5
21: **end if**

This algorithm has been proved to converge to the global minimizer if uniform random sampling is performed together with the weighted centroid reflection [12]. Since this step guarantees the convergence in the limit by assigning a non-zero probability to the neighborhood of any point on the domain, it often compromises the performance on most functions in practice. Therefore, we chose not to perform the uniform sampling, and to rely only on the heuristic provided by the centroid reflection. This approach has been extensively numerically evaluated in the literature [11], [12], [13].

We followed the approach presented by Leonetti et al. [11] and combined this global search with a deterministic local

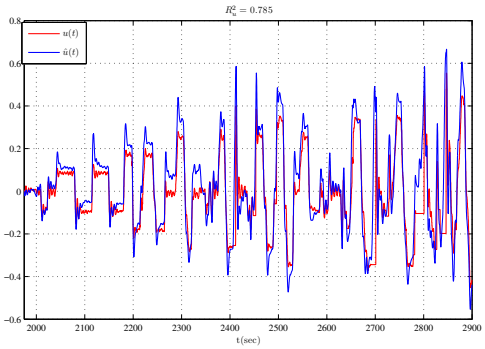

Fig. 3. Validation in surge velocity.

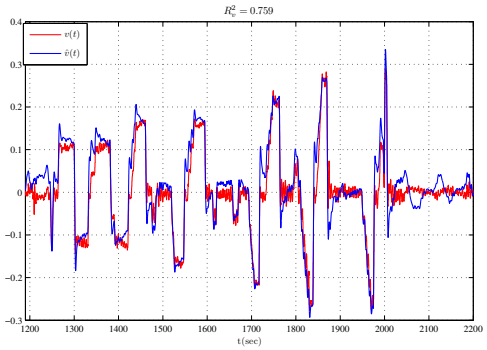

Fig. 4. Validation in sway velocity.

search. Therefore, instead of employing Algorithm 1 with a very small ε (typically in the order of 10^{-6}) we let $\varepsilon = 10^{-2}$, and performed a deterministic local search in the neighborhood represented by the population at the time the global search is terminated. While global stochastic search is a powerful method to avoid being trapped in local minima, the random nature of its sampling makes it less effective when the region of the global minimizer has been identified. Thus, the best point from the first global phase is used as the starting point of the following local search, which employs a coordinate-search algorithm with line-search expansions [14] reported in Algorithm 2.

Algorithm 2 Line Search Phase

1: Input: $\theta^0 \in \mathbb{R}^n$, $\tilde{\alpha}_1^0, \ldots, \tilde{\alpha}_{2n}^0 > 0$, $\sigma \in (0,1)$, $\gamma \in (0,1)$, $\delta > 1$, $\varepsilon > 0$
2: $k = 0$
3: **if** $\max_{i=1,\ldots,2n} \{\tilde{\alpha}_i^k\} \leq \varepsilon$ **then**
4: STOP
5: **end if**
6: $i = 1$, $y_1^k = \theta^k$, $x^k = \theta^k$
7: **if** $J(y_i^k + \tilde{\alpha}_i^k e_i) \leq J(y_i^k) - \gamma(\tilde{\alpha}_i^k)^2$ and $J(y_i^k + \tilde{\alpha}_i^k e_i) < J(x^k)$ **then**
8: $\alpha_i^k = \tilde{\alpha}_i^k$, $x^k = y_i^k + \alpha_i^k e_i$
9: **while** $J(y_i^k + \delta\alpha_i^k e_i) \leq J(y_i^k) - \gamma(\delta\alpha_i^k)^2$ and $J(y_i^k + \delta\alpha_i^k e_i) < J(x^k)$ **do**
10: $\alpha_i^k = \delta\alpha_i^k$, $x^k = y_i^k + \alpha_i^k e_i$
11: **end while**
12: $\tilde{\alpha}_i^{k+1} = \alpha_i^k$
13: **else**
14: $\alpha_i^k = 0$
15: $\tilde{\alpha}_i^{k+1} = \sigma\tilde{\alpha}_i^k$
16: **end if**
17: $y_{i+1}^k = y_i^k + \alpha_i^k e_i$
18: **if** $i < 2n$ **then**
19: $i = i + 1$, go to 7
20: **end if**
21: $\theta^{k+1} = x^k$, $k = k + 1$, go to 3

The algorithm uses positive step sizes $\alpha_j, j = 1, \ldots, 2n$ along the cardinal directions $\{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$ to search for a point that improves the current best point starting from θ^0. If the largest step $\max_{i=1,\ldots,2n}\{\tilde{\alpha}_i^k\}$ is smaller than the parameter ε the algorithm terminates (line 3). Trial points are subsequently generated, along the direction e_i and with steps α_i. If a point along a direction e_i starting at y_i^k improves y_i^k of at least $\gamma \cdot (\alpha_i^k)^2$, and also improves the current best point x_i^k, then α_i^k is increased by a factor δ, and a farther point along e_i is tried (lines 7–12). This is the expansion phase. If the trial point is not sufficiently improving, α_i^k is reduced by a factor σ in a contraction phase (lines 14–15), and other directions are polled. Typically $\delta = 2$ and $\sigma = 1/2$. When all the directions have been contracted up to ε the algorithm terminates. Coordinate-search algorithms poll the function on a finite number of points, and are guaranteed to provide a locally optimum value at the required precision ε. No point closer to the current best point than ε is polled.

IV. EXPERIMENTAL RESULTS

In order to prove the effectiveness of the proposed methodology, the on-line identification procedure was carried out for the AUV Girona500 (see Fig. 1). According to the method analyzed in Section III, we provide sinusoidal and step excitation inputs in the form of body forces and torques in the Excitation Module. The vehicle's state vector is measured via the Navigation Module presented in Subsection II-B. The identification scheme was applied on-line with the Metric Calculator and the Optimization Algorithm modules cooperating in a closed loop form.

The vehicle is under-actuated about surge axis (roll). Also, in this work we consider no actuation about the sway axis (pitch) because large pitch angles may cause the Navigation Module (Section II-B) to diverge, due to false DVL measurements. Thus, with the existing setup we cannot provide adequate excitation inputs to properly identify the vehicle in pitch either. However, the vehicle is statically stable, so without actuation about surge and sway axes the roll and pitch angles are always close to zero. According to the configuration of the vehicle, existing planes of symmetry and operational speed, we can safely assume that Girona500 can be described by the following dynamic model:

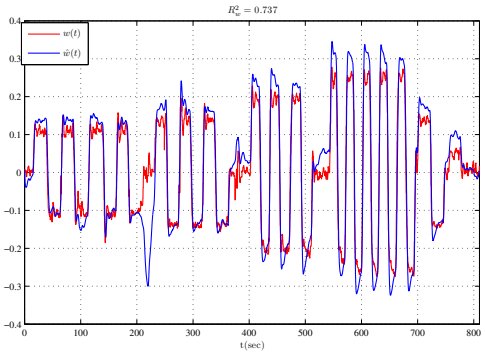

Fig. 5. Validation in heave velocity.

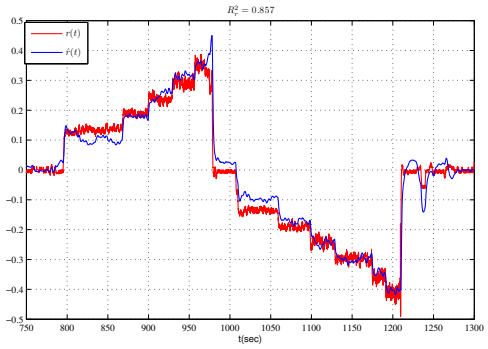

Fig. 6. Validation in yaw velocity.

$$m_u\dot{u} - m_v vr + m_w wq + (W-B)sin(\theta) - X_u u - X_{|u|u}|u|u = X \quad (12)$$

$$m_v\dot{v} - m_w wq + m_u ur - (W-B)cos(\theta)sin(\phi) - Y_v v - Y_{|v|v}|v|v = Y \quad (13)$$

$$m_w\dot{w} - m_u uq + m_v vp - (W-B)cos(\theta)cos(\phi) - Z_w w - Z_{|w|w}|w|w = Z \quad (14)$$

$$m_r\dot{r} - (m_u - m_v)uv - N_r r - N_{|r|r}|r|r = N \quad (15)$$

The dynamic model is highly non-linear and can be written in the generic form as described in Eq. 2 where X, Y, Z, N are the input commands, m_u, m_v, m_w, m_r, $W-B$, X_u, $X_{|u|u}$, Y_v, $Y_{|v|v}$, Z_w, $Z_{|w|w}$, N_r, $N_{|r|r}$ correspond to the unknown parameters and u, v, w, p, q, r, ϕ, θ are provided by the navigation module. The optimization algorithm has been executed with the nominal parameter values as presented in the previous section, with bounds $[-1000, 1000]$ for all the unknown parameters, and converged to the following values: $m_u = 249.5384$, $m_v = 367.7126$, $m_w = 659.9799$, $m_r = 74.9024$, $W-B = -37.3058$, $X_u = -42.4181$, $X_{|u|u} = -125.3578$, $Y_v = -75.7673$, $Y_{|v|v} = -447.6195$, $Z_w = -44.0561$, $Z_{|w|w} = -325.0138$, $N_r = -20.5833$, $N_{|r|r} = -60.9373$. In order to prove the accuracy of the estimated parameters, an additional input/output data set was acquired to be used for validation purposes only. Using the estimated parameters and applying the validation inputs X, Y, Z, N, we solved the differential Equations 12–15 to calculate the predicted states $\hat{u}$, $\hat{v}$, $\hat{w}$, $\hat{r}$. The yielded responses are given in Figures 3–6 along with the corresponding measured signals u, v, w, r. As it can be seen, satisfactory coefficients of determination R^2 are calculated for all DoFs. Finally, it should be noted that the values of the estimated parameters are reasonable in engineering terms, since (added) masses and (added) inertias were found positive as well as greater than the vehicle's mass ($\approx$ 175kgr). The hydrodynamic friction coefficients of first and second order were correctly found negative.

V. CONCLUSIONS

We presented an on-line identification scheme for autonomous underwater vehicles, that estimates the unknown dynamic parameters based on a global derivative-free optimization algorithm. The proposed algorithm has the following significant attributes in comparison to other on-line identification schemes: a) it is not sensitive to initial conditions, b) it does not depend on the differentiability of the model with respect to the parameters, and does not require a linear formulation, c) demonstrates quick convergence. The applicability and performance of the proposed algorithm were experimentally verified with the Autonomous Underwater Vehicle Girona500.

REFERENCES

[1] T. Fossen, "Guidance and control of ocean vehicles," *Wiley, New York*, 1994.

[2] P. Ridao, A. Tiano, A. El-Fakdi, M. Carreras, and A. Zirilli, "On the identification of non-linear models of unmanned underwater vehicles," *Control Engineering Practice 12*, 2004.

[3] D. Panagou, G. Karras, and K. Kyriakopoulos, "Towards the stabilization of an underactuated underwater vehicle in the presence of unknown disturbances," *MTS/IEEE OCEANS*, 2008.

[4] D. Smallwood and L. Whitcomb, "Adaptive identification of dynamically positioned underwater robotic vehicles," *IEEE Transcactions on Control Systems Technology*, pp. 505–515, 2003.

[5] P. van de Ven, T. Johansen, A. Sorensen, C. Flanagan, and D. Toal, "Neural network augmented identification of underwater vehicle models," *Control Engineering Practice 15*, 2007.

[6] S. Julier and J. Uhlmann, "A new extension of the kalman filter to nonlinear systems," *Proc.of the Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.

[7] R. V. der Merwe and E. Wan, "The square-root unscented kalman filter for state and parameter-estimation," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3461–3464, 2001.

[8] M. VanDyke, J. Schwartz, and C. Hall, "Unscented kalman filtering for spacecraft attitude state and parameter estimation," *Advances in the Astronautical Sciences*, pp. 217–228, 2004.

[9] G. Karras, S. Loizou, and K. Kyriakopoulos, "Towards semi-autonomous operation of under-actuated underwater vehicles: sensor fusion, on-line identification and visual servo control," *Autonomous Robots*, pp. 67–86, 2011.

[10] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep., 1995.

[11] M. Leonetti, P. Kormushev, and S. Sagratella, "Combining local and global direct derivative-free optimization for reinforcement learning," *Cybernetics And Information Technologies*, vol. 12, no. 3, pp. 53–65, 2012.

[12] P. Brachetti, M. De Felice Ciccoli, G. Di Pillo, and S. Lucidi, "A new version of the price's algorithm for global optimization," *Journal of Global Optimization*, vol. 10, no. 2, pp. 165–184, 1997.

[13] W. Price, "Global optimization by controlled random search," *Journal of Optimization Theory and Applications*, vol. 40, no. 3, pp. 333–348, 1983.

[14] S. Lucidi and M. Sciandrone, "On the global convergence of derivative-free methods for unconstrained optimization," *SIAM Journal of Optimization*, vol. 13, no. 1, pp. 97–116, 2002.