Coastal Current Prediction using CMA Evolution Strategies

Andrea G. B. Tettamanzi
Università degli Studi di Milano
DTI, via Bramante 65
26013 Crema (CR), Italy
andrea.tettamanzi@unimi.it

Christel Dartigues-Pallez
Université de Nice Sophia Antipolis
I3S (CNRS UMR-6070), 2000 route des Lucioles
06903 Sophia Antipolis, France
christel.dartigues-pallez@unice.fr

Célia da Costa Pereira
Université de Nice Sophia Antipolis
I3S (CNRS UMR-6070), 2000 route des Lucioles
06903 Sophia Antipolis, France
celia.pereira@unice.fr

Denis Pallez
Université de Nice Sophia Antipolis
I3S (CNRS UMR-6070), 2000 route des Lucioles
06903 Sophia Antipolis, France
denis.pallez@unice.fr

Philippe Gourbesville
Université de Nice Sophia Antipolis
Polytech’Nice-Sophia, 930 route des Collès
06903 Sophia Antipolis, France
gourbesv@unice.fr

ABSTRACT

We propose a data-driven evolutionary approach to the modeling of marine currents in the Bay of Monaco. The CMA (Covariance Matrix Adaptation) evolution strategy is used to optimize the parameters of a predictive model that may be used as a surrogate of expensive and time-consuming finite-element simulations. The models obtained are reasonably accurate and easy to interpret.

Categories and Subject Descriptors

1.2.6 [Artificial Intelligence]: Learning—parameter learning; I.6.5 [Simulation and Modeling]: Model Development; J.2 [Physical Sciences and Engineering]: Earth and atmospheric sciences; G.1.6 [Numerical Analysis]: Optimization—global optimization; G.3 [Probability and Statistics]: Probabilistic algorithms

General Terms

Algorithms

Keywords

Evolution Strategies, Hydrology

1. INTRODUCTION

During the last years, the monitoring of continental and marine environments has known a tremendous improvement with the combined evolution of the monitoring technologies and the numerical modeling tools. The most impressive improvements are linked to the growing efficiency of sensors associated with a limited cost which allows to design and implement large data collection networks. This new type of approach will allow to have a more accurate and in-depth view of the physical processes. The evolution of technology, with new types of sensors such as acoustic Doppler current profilers (ADCPs), multibeam sonars, light detection and ranging (LIDAR), and pushbroom cameras, has deeply modified the quality and the quantity of data available for modeling tools [5].

ADCPs use sound to measure their environment instead of a spinning propeller, transmitting ultrasounds into a flowing steam of water. After a sound burst is emitted by the device, echoes are returned from the particles carried by currents and from the streambed. By analyzing these sound echoes, the device makes several different measurements. Some of those measurements concern the speed and direction of water currents, which are determined at many levels through the water depth (the current profile).

The ADCP has replaced the traditional current meters moored on a vertical line. The use of a single ADCP can replace several current meters and provides a perfect view of the velocity distribution in the water depth. The size of the cells can be adjusted and, in most cases, this size can be set to only a few meters. With this accuracy, the ADCP can provide a perfect picture—temporal and spatial—of the physical processes.

Even if ADCPs offer a better precision, there still remain some problems to solve. One of them concerns the complexity of the marine currents. The analysis of marine currents is the result of a complex mix of regional and local currents.
This particular situation explains why some of current measurements could be neglected or rejected according to the objective of the study. In the same way, this fact means that the simultaneous use of several ADCPs is preferred in order to understand the different scales of the processes.

Hydroinformatixs systems — last generation of modeling tools — now have the possibility of investigating the details of the physical hydrodynamic process as well as the complexity of the geometry of the continental and marine environments. This new situation represents a unique situation where all hydroinformatics approaches (data-driven and physically based) can be associated and combined in order to achieve the most efficient model for the simulation of the dynamic of the environment.

This paper describes a data-driven, evolutionary approach to the modeling of the hydrodynamics of a specific, large-scale marine environment in the framework of several study campaigns carried out in the Bay of Monaco.

Since 2006, the Principality of Monaco has promoted a sea extension project, with the idea to extend the 2 km² urban domain, following various approaches which envisage the construction of floating platforms secured to the land.

To avoid the water stagnation problems and eutrophication processes encountered in similar projects like in Qatar and Dubai, to preserve the exceptional quality of the coast line which is still covered with red coral and protected sea grass species, the Monaco government has initiated an in-depth investigation of the marine environment in order to have an exhaustive and detailed knowledge of currents patterns taking place along the Monaco coastline and around the future extension of the Principality.

Detailed historical series of current data have been acquired during several campaigns carried out in the sea section in front of Monaco by means of eight ADCPs deployed in strategic positions for catching the main characteristics of currents. The time series comprise measures of the direction and speed of current for each sensor at different depths, as well as data on the water temperature.

These data could be used to calibrate a 3D hydrodynamic model, based on the Telemac system which is using a finite element method, based on the integration of partial differential equations describing the fluid dynamics of a body of sea water, with boundary conditions given by the specific bathymetry of the bay and by a region model covering a large extent of the Mediterranean sea. In such deterministic approach, the results of the physical model are validated with the measurements and provide a simulation of the current patterns for the measurement periods.

The purpose of the research work described in this paper is to approach this problem from a completely different angle, by using the available data to learn a predictive model of water circulation which captures the specificities of the Bay of Monaco in a succinct and actionable representation. The model should allow to predict with reasonable accuracy the current speed and direction at any arbitrary point of the bay from the current speed and direction in the (fixed) locations of the sensors, without having to perform an expensive and time-consuming simulation.

Recently, successful attempts at doing something similar have been made by using artificial neural networks (ANNs) in the somehow related hydrological settings of river stage forecasting in Bangladesh [9] and of tsunami forecasting [12]. ANNs have also been used, with promising results, in the same setting of the Bay of Monaco, to predict a profile of the current based on measurement of the wind at a number of nearby meteorological stations [1]. Although we do not use ANNs in this work, our proposal draws inspiration from such data-driven approaches to hydrological forecasting. The main advantage of our approach with respect to the ANN-based approaches is that we obtain more intuitive models that are easily interpreted by the experts.

The paper is organized as follows: Section 2 defines the modeling problem and Section 3 describes the data available for approaching it. Section 4 proposes a method whereby the modeling problem is reformulated as a parameter optimization problem that can be solved by means of an evolution strategy. Experimental results are discussed in Section 5 and some conclusions are drawn in Section 6.

2. PROBLEM DESCRIPTION

Two measurement campaigns were organized in 2006 in summer and winter conditions in order to obtain an accurate view of the currents along the coast during those periods. For this purpose, eight ADCP’s were moored from the 7th of July to the 7th of August and from the 16th of November and the 17th of December. The location of the various ADCP’s were selected in order to provide an optimal picture of the complex current patterns generated by the regional and local circulation.

The problem is to find a function that, for any time \( t \), given the measurements of the current magnitude and direction in the fixed locations \( x_1, \ldots, x_8 \) of the sensors, \( v(t, x_1), \ldots, v(t, x_8) \), computes (i.e., predicts) the unknown magnitude and direction \( v(t, x) \) at an arbitrary point \( x \) of the sea section of interest.

Here, we are just interested in modeling the current at a constant depth of 20 m below the sea level. In other words, of the whole 3D hydrodynamics of the bay, we choose to study a 2D section that is relevant to the problem of characterizing the circulation of water near the surface. Of course, this choice is justified by the specific application (understanding the environment the floating platforms would be placed in). However, the method we describe in the following sections could be generalized to approach a full 3D modeling problem.

The current magnitude and direction, in this 2D section, is represented as a vector \( v = (u, v) \), where \( u \) is the longitudinal component and \( v \) is the latitudinal component of the speed, measured in m/s, of a hypothetical water particle. A point of the sea section of interest is represented, as well, by a vector \( x = (x, y) \) in the Lambert III France metric coordinate system of the Lambert conformal conic projection. Notice that, in this system, both the longitudinal component \( x \) and the latitudinal component \( y \) are expressed in m, which makes it easy to compute distances between points.

The positions of the eight ADCPs deployed in the Bay of Monaco are shown in Figure 1.

3. MATERIAL

For the purpose of the study described in this paper, eight ADCPs have been placed throughout the coastal in order to collect data from different positions and different depths. Two datasets of vectors which represents the current at the positions of the ADCPs at given moments in time have been extracted. The first dataset corresponds to the data col-
4. METHOD

To approach the problem described above, the idea is to adopt an ad hoc representation of a predictive model for marine currents, suggested by the nature of the phenomenon under study. A prediction by interpolation of current \( v \) at an arbitrary position \( x \) could be obtained by means of a weighted average of the currents measured by the eight sensors at their respective positions,

\[
v(x) = \frac{\sum_{i=1}^{C} v(x_i) \frac{f_i(x)}{d(x,x_i)}}{\sum_{i=1}^{C} f_i(x)}
\]

where \( x_i \) is the position of the \( i \)th sensor, \( C = 8 \) is the number of sensors, and \( d(\cdot, \cdot) \) is a norm of the difference of its arguments.

In general, an interpolative model like the one in Equation 1 may be a good starting point to make predictions when nothing else is known about the context in which the current is measured. However, it will fail to take into account any specific knowledge of how currents behave in the particular environment that is being studied.

The idea, then, is to replace the generic weighting factor \( 1/d(x,x_i) \) in Equation 1 with several weighting functions \( f_i(x) \), one for each available sensor, whose definition captures the influence the measures read by the \( i \)th sensors have on the prediction of the current at position \( x \). The predictor, thus, becomes

\[
v(x) = \frac{\sum_{i=1}^{C} v(x_i) f_i(x)}{\sum_{i=1}^{C} f_i(x)}.
\]

According to such representation, the specific knowledge of the bathymetry of the bay would be represented by the shapes of functions \( f_i \).

These weighting functions must have a large enough granularity in order to avoid the risk of overfitting the data. Furthermore, it is reasonable to assume that the weight of an ADCP on the prediction of current at a given position decreases as its distance from that position increases. This naturally suggests to model the influence of each ADCP’s readings as a two-dimensional bell-shaped function, such as the general two-dimensional elliptical Gaussian function centered in \( x_0 \), the position of the sensor,

\[
f(x) = h \cdot e^{-r^T C r},
\]

where \( h \) is the height, i.e., the maximum of \( f(\cdot) \), \( r = x - x_0 \), and the matrix

\[
C = \begin{bmatrix} a & b \\ b & c \end{bmatrix}
\]

is positive-definite. Equation 3 may thus be rewritten as

\[
f(x,y) = h \cdot e^{-r_x^2 - 2br_x r_y + r_y^2},
\]

where \( r_x = x - x_0 \) and \( r_y = y - y_0 \).

To deal with missing values, the summations in Equation 2 are carried out over the sensors whose data are available; if
data for a sensor is missing, the relevant term in the summations is skipped.

A predictive model of the form of Equation 2, where the \( f_i(\cdot) \) are defined as in Equation 3, is completely determined by 4\( C \) real-valued parameters:

\[
m = (h_1, a_1, b_1, c_1, \ldots, h_C, a_C, b_C, c_C). \tag{5}
\]

Given the current vectors read by the ADCPs at a certain moment, the predictor may be used to compute an estimate of the current in any arbitrary position. To calibrate the model, we may use a dataset of past readings of the current vectors.

The problem of finding a good current predictor of this form may thus be formulated as an unconstrained continuous minimization problem, whose objective function computes the root mean square deviation (RMSD) of the predicted current vectors with respect to the historical vectors recorded in the training set.

The RMSD of the predictions made by Equation 2 with parameters given by vector \( m \) of Equation 5 is computed as

\[
\text{RMSD}(m) = \sqrt{\frac{1}{Cn} \sum_{i=1}^{n} \sum_{j=1}^{C} ||v_{ij} - \text{predict}(x_j; i; m)||^2}, \tag{6}
\]

where \( n \) is the number of records in the dataset considered, \( x_j \) is the position of the \( j \)-th sensor, \( v_{ij} \) is the reading of the \( j \)-th sensor recorded in the \( i \)-th row of the dataset, and

\[
\text{predict}(x_j; i; m) = \frac{\sum_{k \neq j} v_{ik} f_k(x_i; j; m)}{\sum_{k \neq j} f_k(x_i; j; m)}. \tag{7}
\]

4.1 Representation

Instead of encoding parameters \( a \), \( b \), and \( c \) directly in the genotype, we encode the angle \( \theta \) and the standard deviations \( \sigma_x \) and \( \sigma_y \) along the two axes (longitude and latitude), from which parameters \( a \), \( b \), and \( c \) may be computed based on the following equations:

\[
a = \frac{\cos^2 \theta}{2\sigma_x^2} + \frac{\sin^2 \theta}{2\sigma_y^2}, \tag{8}
\]

\[
b = \frac{\sin 2\theta}{4\sigma_x^2} - \frac{\sin 2\theta}{4\sigma_y^2}, \tag{9}
\]

\[
c = \frac{\sin^2 \theta}{2\sigma_x^2} + \frac{\cos^2 \theta}{2\sigma_y^2}. \tag{10}
\]

This choice has the advantage that all values for \( \theta \), \( \sigma_x \), and \( \sigma_y \), even negative ones, yield a positive-definite matrix \( C \).

As a consequence, a candidate solution to the modeling problem may be expressed as a vector

\[
(h^{(1)}, \theta^{(1)}, \sigma_x^{(1)}, \sigma_y^{(1)}, \ldots, h^{(C)}, \theta^{(C)}, \sigma_x^{(C)}, \sigma_y^{(C)}) \in \mathbb{R}^C. \tag{11}
\]

4.2 The CMA Evolution Strategy

Evolutionary strategies (ESs) [13, 11, 14] are a family of stochastic methods for real-parameter (continuous domain) optimization of non-linear, non-convex functions, that are one component of the broader class of evolutionary algorithms.

Evolutionary algorithms (EAs) [2, 3] are a broad class of stochastic optimization algorithms, inspired by biology and in particular by those biological processes that allow populations of organisms to adapt to their surrounding environment: genetic inheritance and survival of the fittest. Each individual of the population represents a point in the space of the potential solutions for the considered problem. The evolution is obtained by iteratively applying a (usually quite small) set of stochastic operators, known as mutation, recombination, and selection. Mutation randomly perturbs a candidate solution; recombination decomposes two distinct solutions and then randomly mixes their parts to form novel solutions; selection replicates the most successful solutions found in a population at a rate proportional to their relative quality. The initial population may be either a random sample of the solution space or may be seeded with solutions found by simple local search procedures, if these are available. The resulting process tends to find, given enough time, globally optimal solutions to the problem much in the same way as in nature populations of organisms tend to adapt to their surrounding environment.

Evolution strategies approach function optimization problems in the \( n \)-dimensional real space by exploiting a real encoding of the objective function parameters. Candidate solutions are \( n \)-dimensional vectors. An individual is made up of the same vector as the associated candidate solution, plus up to \( n \) variances \( c_{ii} = \sigma_i^2 \), with \( i = 1, \ldots, n \), and up to \( n(n - 1)/2 \) covariances \( c_{ij} \), with \( i, j = 1, \ldots, n \), of the \( n \)-dimensional normal joint distribution having mean \( 0 \) and density function, for all \( z \in \mathbb{R}^n \),

\[
p(z) = \frac{\det C^{-1}}{(2\pi)^n} e^{-\frac{1}{2} y^T C^{-1} y}, \tag{12}
\]

where \( C = (c_{ij}) \) is the variance/covariance matrix. The choice of a normal distribution, that will be used to perturb the candidate solutions, is obviously arbitrary. Overall, an individual will contain \( k \leq n(n+1)/2 \) parameters relevant to the “strategy” besides the \( n \) parameters relevant to the object problem; often, however, only variances are considered, whereas sometimes it is sufficient to consider one variance for all the object problem parameters.

An interesting variant of ESs that may be regarded as essentially an estimation of distribution algorithm (EDA) is the covariance-matrix-adapting evolution strategy (CMA-ES) [8]. CMA-ES is an evolutionary strategy whose aim is to detect and exploit the local structure of a function to be optimized. Although the algorithm has proved to be very effective when used to solve real-world problems, its design (based on the evolution of the covariance matrix) can make its application to high dimensional optimization problems very expensive.

However, our experience (see Section 5) is that the CMA-ES is anyway less computationally expensive than popular deterministic numerical optimization methods like BFGS (Broyden, Fletcher, Goldfarb and Shanno).

5. EXPERIMENTS

We applied the CMA-ES to the optimization of the predictor parameters, represented as explained in Section 4.1. In particular, we used the implementation in R provided by the “cmaes” package [15].

We compare the results with BFGS [4], a well-known quasi-Newton numerical optimization method, and a simple Monte Carlo method.

5.1 Experimental Protocol

The external parameter setting for CMA-ES is the default strategy parameters recommended in [7], except that \( \lambda = 20 \).
Table 1: Summary of the comparison of CMA-ES, BFGS, and Monte Carlo.

<table>
<thead>
<tr>
<th>Method/Dataset</th>
<th>avg</th>
<th>stdev</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMA-ES/training (Summer)</td>
<td>0.056457</td>
<td>0.002260</td>
<td>0.052756</td>
<td>0.061005</td>
</tr>
<tr>
<td>CMA-ES/test (Summer)</td>
<td>0.056735</td>
<td>0.002455</td>
<td>0.053288</td>
<td>0.062262</td>
</tr>
<tr>
<td>CMA-ES/training (Autumn)</td>
<td>0.061989</td>
<td>0.002471</td>
<td>0.058034</td>
<td>0.069365</td>
</tr>
<tr>
<td>CMA-ES/test (Autumn)</td>
<td>0.052154</td>
<td>0.001902</td>
<td>0.049150</td>
<td>0.055932</td>
</tr>
<tr>
<td>CMA-ES/training (Summer + Autumn)</td>
<td>0.060170</td>
<td>0.001260</td>
<td>0.058693</td>
<td>0.062720</td>
</tr>
<tr>
<td>CMA-ES/test (Summer + Autumn)</td>
<td>0.054942</td>
<td>0.000837</td>
<td>0.053559</td>
<td>0.056296</td>
</tr>
<tr>
<td>BFGS/training (Summer)</td>
<td>0.057639</td>
<td>0.002174</td>
<td>0.053336</td>
<td>0.061856</td>
</tr>
<tr>
<td>BFGS/test (Summer)</td>
<td>0.057756</td>
<td>0.002388</td>
<td>0.053868</td>
<td>0.062925</td>
</tr>
<tr>
<td>BFGS/training (Autumn)</td>
<td>0.064292</td>
<td>0.005092</td>
<td>0.058413</td>
<td>0.084768</td>
</tr>
<tr>
<td>BFGS/test (Autumn)</td>
<td>0.053964</td>
<td>0.004281</td>
<td>0.049144</td>
<td>0.073654</td>
</tr>
<tr>
<td>BFGS/training (Summer + Autumn)</td>
<td>0.061281</td>
<td>0.001941</td>
<td>0.059614</td>
<td>0.064712</td>
</tr>
<tr>
<td>BFGS/test (Summer + Autumn)</td>
<td>0.055831</td>
<td>0.001945</td>
<td>0.052835</td>
<td>0.058870</td>
</tr>
<tr>
<td>Monte Carlo/training (Summer)</td>
<td>0.056667</td>
<td>0.000563</td>
<td>0.055254</td>
<td>0.057709</td>
</tr>
<tr>
<td>Monte Carlo/test (Summer)</td>
<td>0.056904</td>
<td>0.000915</td>
<td>0.054976</td>
<td>0.058496</td>
</tr>
<tr>
<td>Monte Carlo/training (Autumn)</td>
<td>0.062398</td>
<td>0.000683</td>
<td>0.061274</td>
<td>0.063883</td>
</tr>
<tr>
<td>Monte Carlo/test (Autumn)</td>
<td>0.052503</td>
<td>0.000733</td>
<td>0.051331</td>
<td>0.054474</td>
</tr>
<tr>
<td>Monte Carlo/training (Summer + Autumn)</td>
<td>0.060898</td>
<td>0.000463</td>
<td>0.059929</td>
<td>0.061476</td>
</tr>
<tr>
<td>Monte Carlo/test (Summer + Autumn)</td>
<td>0.055160</td>
<td>0.000604</td>
<td>0.053849</td>
<td>0.055774</td>
</tr>
</tbody>
</table>

and $\mu = 10$ instead of $\lambda = 4 + \lfloor 3 \ln n \rfloor = 14$ and $\omega = \lfloor \frac{\lambda}{2} \rfloor = 7$. An increased $\lambda$ usually improves the global search capability and the robustness of the CMA-ES, at the price of a reduced convergence speed [6]. We stop each run after 200 generations, which take between 1h40 and 3h to complete on the machines we used. This results in exactly 4,000 objective function evaluations per run.

The same number of objective function evaluations is used for the Monte Carlo method as well. The probability distribution used to randomly sample the parameter space is

$$h_i^{(i)} \sim \mathcal{E}(1),$$

$$\theta_i^{(i)} \sim \mathcal{U}(0, 2\pi),$$

$$\sigma_i^{(i)} \sim \mathcal{U}(0, 10^{-3}),$$

where $\mathcal{E}(\lambda)$ stands for an exponential distribution with mean $\frac{1}{\lambda}$ and $\mathcal{U}(a, b)$ stands for a uniform distribution over the interval $[a, b]$.

The BFGS is run with the default parameters of its implementation under the R system [10], which include stopping after 100 iterations. Each iteration of the BFGS method performs one evaluation of the objective function and one evaluation of its gradient. Since a closed form for the gradient of the objective function is not available, BFGS must estimate each partial derivative numerically, by sampling two points along the relevant dimension; therefore, each gradient evaluation requires 64 evaluations of the objective function, yielding a total of 6,500 objective function evaluations per run, 62.5% more than a CMA-ES or Monte Carlo run. This “bonus” is compensated, in our opinion, by the handicap of BFGS not being a global optimization method like CMA-ES and the Monte Carlo method.

We have applied the methods to three modeling tasks:

- Summer—using the training dataset from the July–August 2006 campaign, learn a predictive model of the currents typical of the hot season;
- Autumn—using the training dataset from the November–December 2006 campaign, learn a predictive model of the currents typical of the cold season;
- Summer + Autumn—using the two training dataset combined, learn a predictive model of the currents for all seasons.

Since we know that the current patterns are very different for the hot and the cold season, the last task is considerably harder. Furthermore, since more data have to be taken into account, the Summer + Autumn task is also more demanding in terms of computing resources.

5.2 Results

Table 1 shows statistics of the results obtained by the three methods considered over 39 independent runs for the Summer 2006 data, 43 independent runs for the Autumn 2006 data, and 9 independent runs for the Summer + Autumn data.

We applied the independent samples heteroskedastic $t$-test to the following hypotheses:

1. “CMA-ES performs better than BFGS”: $p$-value = 0.033197 based on the Summer test set, $p$-value = 0.007010 based on the Autumn test set, $p$-value = 0.116982 based on the Summer + Autumn test set;
2. “CMA-ES performs better than Monte Carlo”: $p$-value = 0.344348 based on the Summer test set, $p$-value = 0.016280 based on the Autumn test set, $p$-value = 0.116982 based on the Summer + Autumn test set;
3. “Monte Carlo performs better than BFGS”: $p$-value = 0.268121 based on the Summer + Autumn test set;
4. “CMA-ES performs worse on the test dataset than on the training set”: $p$-value = 0.302423 based on Summer
data, \( p\)-value = 1.00000 based on Autumn data; \( p\)-value = 1.00000 based on Summer + Autumn data;

5. “BFGS performs worse on the test dataset than on the training set”: \( p\)-value = 0.410173 based on Summer data, \( p\)-value = 1.00000 based on Autumn data, \( p\)-value = 0.99990 based on Summer + Autumn data;

6. “Monte Carlo performs worse on the test dataset than on the training set”: \( p\)-value = 0.086651 based on Summer data, \( p\)-value = 1.00000 based on Autumn data, \( p\)-value = 1.00000 based on Summer + Autumn data.

Therefore, if we base acceptance or rejection of the first three hypotheses on the test set alone, i.e., on data that has not been used to train the models, we can say that CMA-ES performs better than BFGS with 95% confidence for the Summer and Autumn tasks and Monte Carlo performs better than BFGS with 95% confidence on the same two tasks. The difference in performance between CMA-ES and Monte Carlo is not significant for the Summer and Autumn tasks, nor do the three methods show any significant difference between one another when applied to the Summer + Autumn task. This is probably due to the fact that no single model can be found that works equally well for all seasons.

However, since the standard deviation of the RMSD of the results is much larger for CMA-ES than for Monte Carlo, in practice the former is guaranteed to yield much better models than the latter over multiple runs, as witnessed by the “\( p\)" column in Table 1. This causes CMA-ES to be the preferred method to fit the parameters of the predictive models for practical purposes.

Nevertheless, it should be noted that, in some rare cases, BFGS achieves the best results, as is the case with the Autumn dataset, although by a very narrow margin over the CMA-ES. This is likely due to the fact that, if the initial solution happens to fall within the basin of attraction of a good local optimum, BFGS is very effective at finding it. However, the performance of BFGS is not consistent over multiple runs, as the scatter plot in Figure 3 reveals.

Hypotheses 3 to 6 cannot be accepted with a high confidence, with the notable exception of Monte Carlo performing significantly worse on the Summer test than on the relevant training, with 91% confidence. This means that the modeling approach we propose tends to produce models with satisfactory generalization capabilities, regardless of the method used to fit their parameters.

5.3 Visualization of the Results

To allow the interpretation of the models thus obtained, we propose four types of diagrams. Figures 4 and 6 show images of the superposition of the eight weighting functions: lighter areas correspond to higher weights and darker areas to lower weights. The weight of the superposition in a given position is the maximum of the weights of the weighting functions. This type of image gives an overall idea of the model structure. In particular, the model shown in Figure 4 is the best model found by the CMA-ES for the Summer dataset; the one shown in Figure 6 is the best model found by BFGS for the Autumn dataset—the best model found by the CMA-ES is almost identical.

Figures 5 and 7 give an idea of how the eight sensors distribute their influence on the prediction of the current all over the area considered. The zones where the contributions of one sensor prevail over the others are filled with the color associated with that sensor. Zones not covered by any sensor would appear in red. However, all the models found by the three methods we have used cover the whole area.

From these diagrams, it can be observed that the models obtained capture relevant aspects of the phenomenon under study and represent them in an intuitive way. For instance, Figure 5 shows that ADCPs 4 and 5 are well positioned to measure the coastal current during the summer, as their influence extends toward open sea, whereas ADCPs 1 and 3 capture turbulences caused by the coastline. The Gaussian weighting functions, which are mostly parallel to the coast in the summer model of Figure 4, have different orientations in the autumn model (Figure 6), which determine a less regular partition of the zones of influence of the sensors, with ADCPs 4 and 5 still accounting for the coastal current.

Figure 8 shows an example of the current pattern predicted by the best Summer model based on the measurement of the eight sensors on July 8, 2006, at 0:00 AM.

Finally, Figure 9 shows a simulation of the trajectory that would have been followed by eight particles dropped at the positions of the sensors on July 11, 2006 at midnight. The trajectories have been integrated with an integration step of 10 minutes, by linear interpolation of the currents predicted by the model of Figure 4. This type of diagram gives an idea of the water circulation patterns under different conditions, which is the main objective of the study.

6. CONCLUSION

We have shown that models of marine currents that may be used as surrogates of expensive and time-consuming finite-element simulations can be obtained for specific environments by means of a powerful evolutionary optimization method. The models obtained are reasonably accurate and have good generalization properties. Furthermore, the seem to capture relevant aspects of the phenomenon in an intuitive and easy to interpret representation.

Possible future work would consist of trying alternative
Figure 4: Superposed Gaussian functions of the best model for the Summer dataset.

Figure 5: Zones of influence of the sensors for the model in Figure 4. Each color corresponds to the sensor whose number is given in the right bar, and shows the zone where the weight of the sensor is greater than the weight of all the other sensors.

Figure 6: Superposed Gaussian functions of the best model for the Autumn dataset.

Figure 7: Zones of influence of the sensors for the model in Figure 6. Each color corresponds to the sensor whose number is given in the right bar, and shows the zone where the weight of the sensor is greater than the weight of all the other sensors.
parametric shapes for the weight functions, possibly asymmetric.

7. ACKNOWLEDGMENTS
The authors are grateful to Prof. Liong Shie-Yui for fruitful discussion and suggestions.

8. REFERENCES