Neural Network Modeling of Nearshore Sandbar Behavior

L. Pape, B. G. Ruessink, M. A. Wiering, and I. L. Turner

Abstract— **The temporal evolution of nearshore sandbars (alongshore ridges of sand fringing coasts in water depths less than 10 m and of paramount importance for coastal safety) is commonly predicted using process-based models. These models are autoregressive and require offshore wave characteristics as input, properties that find their neural network equivalent in the NARX (Nonlinear Auto-Regressive model with eXogenous input) architecture. Earlier literature results suggest that the evolution of sandbars depends nonlinearly on the wave forcing and that the sandbar position at a specific moment contains 'memory', that is, time-series of sandbar positions show dependencies spanning relatively long time periods. Using observations of an outer sandbar collected daily for about 3.5 years at the double-barred Surfers Paradise, Gold Coast, Australia we find, however, little difference in performance between a NARX, an autoregressive multilayer perceptron (without long-term dependencies), and a linear NARX. It is uncertain whether these results generalize to the inner Gold Coast bar or to other field sites.**

I. Introduction

The study of the coastal zone and the processes that lead to the morphological behavior of sandbars are important issues in coastal management. As the coast is under constant exposure to continuous incoming wave attack, nearshore sandbars are the most important factors responsible for the dissipation of wave energy, protecting the shoreline from coastal erosion. The prediction of the evolution of these morphological features is required to plan and evaluate coastal defenses, and to determine the effect of changes in the current conditions. Field data are essential for studying the evolution of nearshore sandbars and the calibration and validation of constructed models. Extensive data are available in the Argus program [1], providing hourly video images of the nearshore zone together with hydrodynamic data, for several locations worldwide.

Models of the nearshore constructed using knowledge of physical processes are common in physical geography, civil engineering and coastal oceanography [2]. As these process-based models come with their own practical problems and limitations, other classes of models are applied in some situations. An important class of models is the class of data-driven models. In contrast to process-based models, data-driven models need no process knowledge to operate, but are constructed by establishing statistical relations in field data. Neural networks, becoming increasingly popular in many sciences, are members of the class of data-driven models. Most process-based models describing the behavior of the nearshore morphology are autoregressive, a property that finds its neural network equivalent in the NARX (Nonlinear Auto-Regressive model with eXogenous input) neural network architecture.

The behavior of sandbars in terms of the cross-shore position has been shown to depend nonlinearly on the factors forcing the migration of the sandbar (for instance, [3]). Furthermore, 'memory' is sometimes assumed to be present in the time-series of cross-shore sandbar positions, meaning dependencies in the position of the sandbar spanning longer time periods. An example of modeling temporal dependencies is [4], in which the weighted hydrodynamic forcings over several days are included in a variable to provide for a measure of relaxation time in morphology.

The NARX neural network architecture is known to be capable of representing nonlinear dynamics as well as long-term dependencies in time-series data [5]. The presence of memory and nonlinearity in the time-series of cross-shore positions of sandbars are studied here to identify if these claims justify the application of a recurrent neural network such as the NARX neural network architecture.

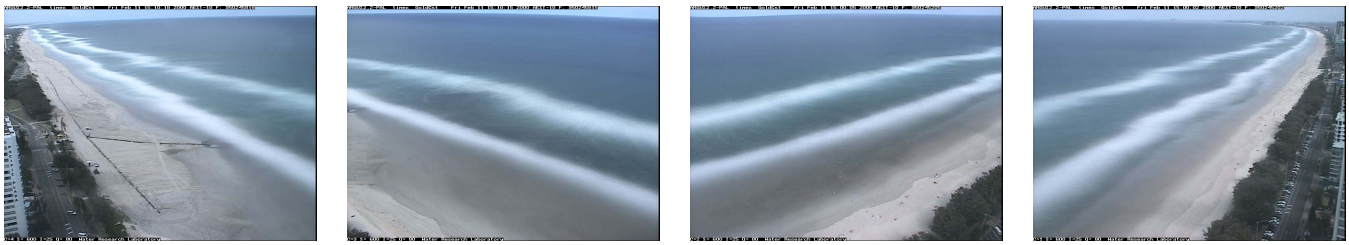
II. Observations

The sandbar data set used in the present work was acquired with an Argus coastal imaging station located at the double-barred Surfers Paradise, northern Gold Coast, Queensland, Australia [6]. The station consists of four cameras pointed obliquely along the beach (Fig. 1(a)), providing 180° uninterrupted coverage of the beach and nearshore zone. Each daylight hour, the cameras acquire a time-exposure image (Fig. 1(a)), created by averaging over 600 consecutive images (snapshots) collected at 1 Hz. This will remove moving objects such as ships, vehicles and people. Furthermore, the time-exposure images reveal one or more smooth white bands of breaking waves, which serve as a reasonable estimate for the submerged sandbars [7], [8]. All four images can be rectified [9] and merged to yield a single planview image (e.g., Fig. 1(b)). As detailed in [8], the crest lines of the inner and outer bar can be extracted from a planview image by the automated alongshore tracking of the intensity maxima across each bar (Fig. 1(c)). The alongshore average of a a crest line, in the present work based on planview images with an alongshore extent of 1800 m, is referred to in the following as the barcrest position ($\bar{X}$). The actual (in situ) bar position, however, is known to deviate from the position of the breaking waves by a factor $\mathcal{O}$(10 m), and

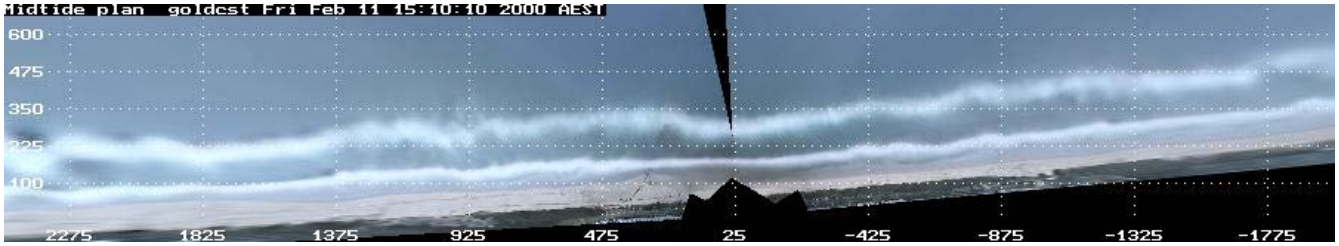
Leo Pape and Gerben Ruessink are with the Institute for Marine and Atmospheric research Utrecht, Department of Physical Geography, Faculty of Geosciences, Utrecht University, P.O. Box 80.115, 3508 TC Utrecht, Netherlands (phone: +31 30 2532982; fax: +31 30 2531145; email: l.pape@geo.uu.nl).

Marco Wiering is with the Institute of Information and Computing Sciences, Utrecht University, Netherlands (email: marco@cs.uu.nl).

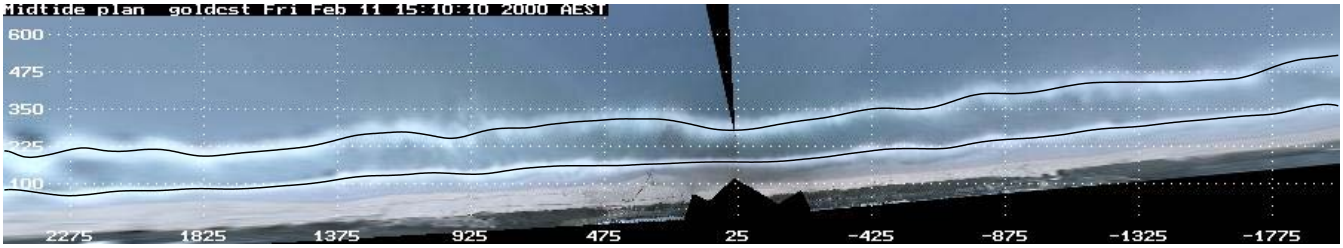
Ian Turner is with the Water Research Laboratory, School of Civil and Environmental Engineering, University of New South Wales, Manly Vale, Australia (email: ian.turner@unsw.edu.au).



(a) Time-exposure Argus images of all cameras. The high-intensity bands in each image are due to persistent wave breaking on the inner and outer bar.



(b) Merged plan view image



(c) Tracked barlines in plan view

Fig. 1. Argus camera images, merged plan view, and tracked barlines

varies in time and alongshore distance with the wave height, the water level, and the bathymetry [8].

Given the availability and quality of the available image data set, a period of 1388 days was selected from September 20, 2000 to July 9, 2004 to construct the time-series of alongshore average bar positions. To eliminate the time-varying nature of the difference between the measured and the actual bar position caused by the semi-diurnal tide, the data set was reduced to a single observation each day, at the lowest tide of that day (when the breaking patterns are most pronounced in the images). During the selected period, three different bars can be distinguished (Fig. 2(a)): (1) outer bar existing at the start of the period, but moving seaward, degenerating and finally disappearing in April 2001, (2) inner bar existing at the start of the period, and becoming the outer bar at the time the former outer bar degenerates, (3) inner bar which forms after the degeneration of the first bar. For several days it was not possible to compute accurate bar positions due to poor image quality (fog or rain droplets on the camera lens) or conditions when waves were too low to break, leaving a total of 1106 observations over this period. To create a continuous time-series data set, the gaps were filled with observations from the last breaking-based observation, assuming that bar migration is insignificant under low-energy conditions. The resulting data set consists of the alongshore average bar position on each day t: $\bar{X}^t$ (note: superscript will be used for time indexing).

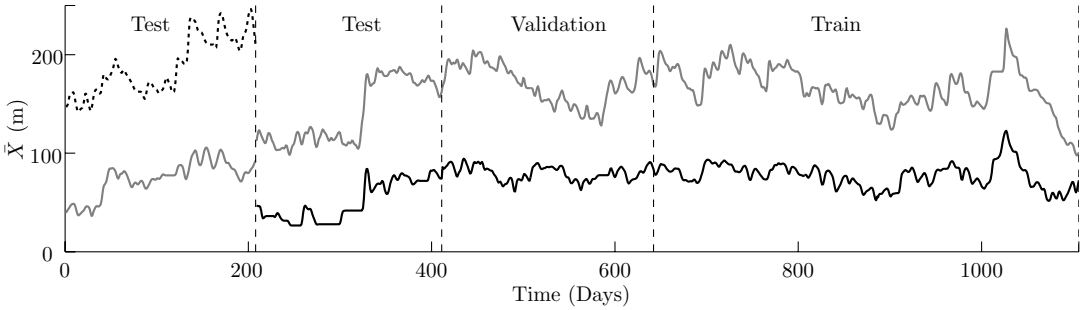
The predominant exogenous inputs for driving sandbar variability in process-based models are the offshore waves, which are represented by their root-mean-square wave height (H_{rms}), peak wave period (T_{peak}) and wave direction relative to the shore normal (θ). To reduce the number of inputs for the neural network, these variables can be combined (using shallow-water linear wave theory) into the wave height at breaking (H_b)

$$H_b = \left(\frac{\gamma}{g}\right)^{\frac{1}{5}} \left[H_{rms}^2 \cdot c_g \cdot \cos(\theta)\right]^{\frac{2}{5}}, \tag{1}$$

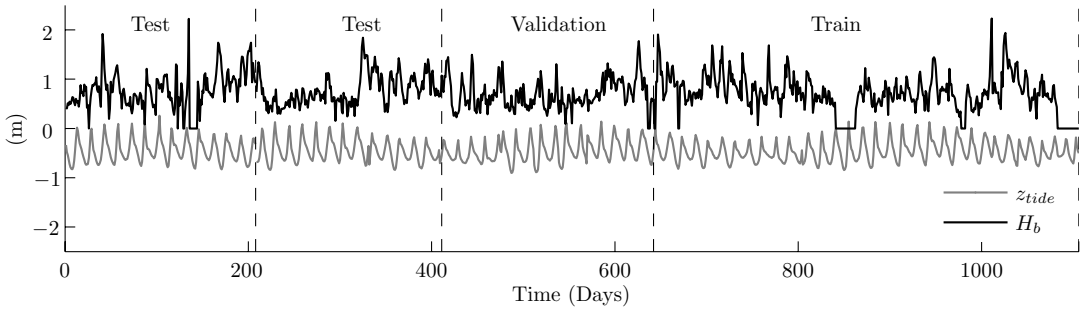
where c_g is the offshore group velocity, g the gravitational acceleration, and γ the breaker parameter. Since the variables are meant for use with a neural network instead of a process-based model, the exact scale of H_b is not important. Therefore any constant in (1) can be left out. Furthermore, according to linear wave theory it holds that $c_g \propto T_{peak}$. Leaving out the constants and replacing c_g by T_{peak} results in a simpler formula to compute the variable proportional to H_b, which we shall call H_{b_s}

$$H_{b_s} = H_{rms}^2 \cdot T_{peak} \cdot \cos(\theta). \tag{2}$$

The variables H_{rms} and T_{peak} were obtained hourly from the Gold Coast waverider buoy, located approximately 2 km offshore of the study area, in 16 m water depth. Directional



(a) $\bar{X}$ for the different bars: dashed (1), gray (2) and black (3)



(b) Hydrodynamic data

Fig. 2. Overview of the variables in the data set

information (θ) was collected by the Brisbane waverider buoy located some 10 km offshore in 70 m water depth, about 100 km north of the study area. Because the wave variables don't change very much on a daily timescale, the values at the same time as the (low-tide) collection of the image data were used.

As pointed above, another possible factor in determining the actual, and especially the observed bar migration, is the tide (z_{tide}). The value of the tide used here is the astronomical tide which is known to deviate from the actual water level due to atmospheric conditions, such as air pressure and temperature, and extreme weather conditions causing a storm surge. Because no data on the atmospheric conditions and the weather were available, the predicted astronomical tide was used for an estimate of the actual water level. The time-series of the hydrodynamic data are depicted in Fig. 2(b).

For application in neural networks, the alongshore average bar positions were scaled between 0 and 1, while H_{b_s} was scaled between 0 and 1, and z_{tide} between -1 and 1.

III. Models

A. Autoregressive Models

The presently most common approach to the modeling of nearshore bathymetry is by means of process-based models [2]. A formal description of these models can be given in terms of inputs, outputs, states, functions and parameters. In discrete time, the internal state of the model at a certain time t, called the system state ($\vec{U}^t$), depends on the states of the model in the past, the external forcing ($\vec{I}^t$) to the model at time t, and the external forcing in the past

$$\vec{U}^t = \mathcal{M}\left(\vec{U}^{t-1}, \ldots, \vec{U}^{t-M}; \vec{I}^t, \vec{I}^{t-1}, \ldots, \vec{I}^{t-N} | \vec{\Theta}\right), \quad (3)$$

where $\mathcal{M}$ represents the model's process knowledge, $\vec{\Theta}$ the adjustable model parameters, M the autoregressive order of the system state, and N the order of the external forcings. The parameters in $\vec{\Theta}$ must be identified by calibration from observed model behavior or the literature. A model in which the output is fed back into the model, together with external forcing, is known as an Auto-Regressive model with eXogenous Input (ARX). In case the transfer functions of the model are nonlinear, it is called a Nonlinear Auto-Regressive model with eXogenous Input (NARX).

Instead of incorporating all forms of process knowledge in a model, a statistical model can be constructed from available data. Various methods exist to build simple or more complicated statistical models, but because of the supposed nonlinear nature of the nearshore dynamics such a model should preferably be nonlinear. Several feedforward neural network architectures, using nonlinear transfer functions show such nonlinear behavior, as well as recurrent neural networks [10]. Replacing the process knowledge ($\mathcal{M}$) in (3) by a neural network ($\mathcal{N}$) and the adjustable model parameters ($\vec{\Theta}$) by the weight matrix ($\mathbf{W}$) of the neural network, (3) can be written as

$$\vec{U}^t = \mathcal{N}\left(\vec{U}^{t-1}, \ldots, \vec{U}^{t-M}; \vec{I}^t, \vec{I}^{t-1}, \ldots, \vec{I}^{t-N} | \mathbf{W}\right). \quad (4)$$

The transfer function ($\mathcal{N}$) is now a neural network, and the weight matrix ($\mathbf{W}$) is to be determined by learning from the data, as is the case for data-driven models. This learning involves the minimization of the squared differences between the observations ('targets') and neural network output.

B. NARX Neural Network Architecture

A multilayer perceptron (MLP) [11] is an appropriate candidate neural network architecture to replace the process-based model. Each time step t the set of external inputs, which are defined for $t, \ldots, t-N$, is provided to the network, together with the set of outputs of the network from previous steps: $t-1, \ldots, t-M$. The alongshore average position of the sandbar is the value that is to be predicted, so the output of the network ($\vec{U}^t_{out}$) is

$$\vec{U}^t_{out} = \left(\bar{X}^t\right), \tag{5}$$

where $\bar{X}^t$ is the observed alongshore average bar position at time t. The external inputs to the network, being the factors that force the bar migration, are the hydrodynamic data

$$\vec{I}^t = \left(H^t_{b_s}, z^t_{tide}\right). \tag{6}$$

While backpropagation [11] is a commonly used technique for training feedforward neural networks such as MLPs, several interpretations exist for the application of this method in recurrent neural networks. RealTime Recurrent Learning (RTRL) [12] and BackPropagation Through Time (BPTT) [11] are well-known training algorithms. Whereas RTRL is of more theoretical interest, BPTT is the algorithm described by [5] to train the NARX neural network architecture. Using BPTT to train a NARX neural network, the recurrent connections are unfolded in time, and the resulting network is treated as an MLP with injected errors. Instead of unfolding the temporal operation of a network into a multilayer feedforward network that grows by one layer each step, as is common practice in BPTT [11], the entire network is unfolded at the recurrent connections. In the unfolded network, the recurrent connections appear as jump-ahead connections, providing for a shorter path for backpropagating the error through the network.

Because the weight changes are accumulated during the presentation of a certain number of samples and applied after the computation of all errors in the unfolded network, this approach is called batch-wise backpropagation. Since the time-series data of alongshore average positions of the sandbars are not naturally segmented into independent batches, the BPTT algorithm as described above cannot be used on these data. [12] provide a solution to this problem, using a modified BPTT algorithm that can handle time-series of any size, dividing it into a number of batches with arbitrary size h. The resulting algorithm, called BPTT($h; h'$), is exactly the same as batch-wise backpropagation through time in case h is chosen to equal h'.

After presenting one batch to the network, the error is backpropagated through the unfolded network. In the output units of the recurrent network, the local error is computed and added to the backpropagated value from the subsequent input unit. As the error in the present time step is reduced while taking into account the errors made in the future, the NARX neural network is able to learn long-term dependencies in the data.

It is possible to disregard these long-term dependencies when the error is not backpropagated over the recurrent connections. In such a network the errors made in the future are not considered in the computation of the present weight updates. This autoregressive MLP still uses the previous predicted outputs as input, but does not backpropagate any error information through the recurrent connections. When using this modified learning algorithm, the network will not be able to learn any long-term dependencies in the data.

C. Linear NARX Neural Networks

Even the complex NARX architecture can be reduced to a very simple recurrent linear network, using only a few units and linear functions. The linear network used in the experiments is a NARX neural network with two input units, one for H_{b_s} and one for z_{tide}, an additional recurrent input for the previous predicted $\bar{X}$, and one output unit. All units use linear squashing functions. This network does not contain any hidden layers, yielding a linear equation for the trained network

$$\bar{X}^t = p_0 + p_1 \cdot H^t_{b_s} + p_2 \cdot z^t_{tide} + p_3 \cdot \bar{X}^{t-1}, \tag{7}$$

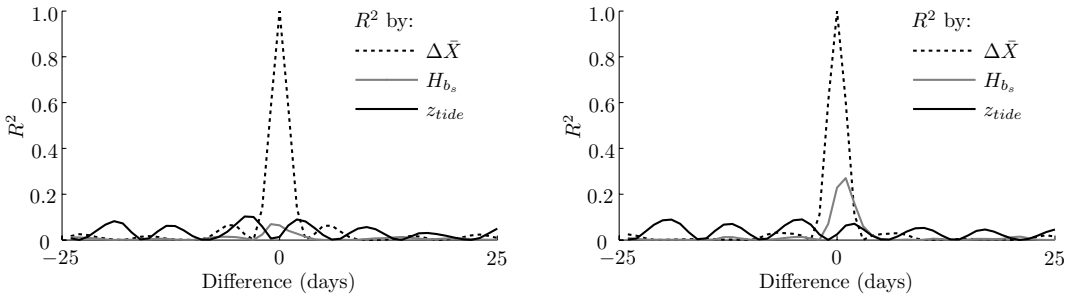
where p_0 denotes the bias value of the output unit and $p_1 \ldots p_3$ the values of the weights from the input units to the output unit. This network can be used to test if the dependence of the bar migration on the hydrodynamic forcings is nonlinear, in which case the linear network should perform very bad. However, if the processes that force the migration of the sandbar can be learned by this simple network, it yields a very simple linear model in the form of (7).

IV. Experiments

A. A Preliminary Analysis

To gain more insight in the processes that force the migration of the sandbar and the timescale on which they operate, a simple linear analysis is performed first. A reasonable measure of the contribution of the different variables to the bar migration, is the proportion of explained variance of these variables in the bar migration according to a linear regression model, as shown in Fig. 3. The difference in days between the time of the measurement of the variables and the observed bar migration is on the horizontal axis, whereas the proportion of explained variance (R^2) in bar migration ($\Delta\bar{X}$) by the hydrodynamic variables is on the vertical axis, together with the autoregressive proportion of the variance in $\Delta\bar{X}$.

As becomes clear from Fig. 3 there is a striking difference between the two bars. In the outer bar more than thirty percent of the variance in observed bar migration is explained by the variable H_{b_s} with a linear model, whereas this value is well below ten percent for the inner bar. The tides however, contribute not much to the variance in observed bar migration; the tides measured at the same moment as the observation of the bar position explain a value close to zero of the bar migration. The timescale on which the



(a) Inner bar (b) Outer bar

Fig. 3. Proportion of explained variance in bar migration over several days for the inner and outer bar

hydrodynamic variables cause the sandbar migration, as well as the timescale of the autocorrelation in bar migration is very small. According to this linear model there will be little or no long-term dependencies in the data.

Because in this linear analysis the proportion of explained variance in bar migration by H_{b_s} is much larger for the outer than for the inner bar, it is likely that a neural network can learn to capture the behavior of the outer bar better. As such, all further experiments will be performed on the outer bar only.

Linear regression can also be applied to assess the variation between different parts of the data. Although not shown here in detail, the proportion of explained variance in bar migration of the outer bar by H_{b_s} explains as much as forty percent of the observed outer bar migration in the first halve of the data set, whereas this proportion is only twenty percent in the second halve. Being just a linear model, the full dynamics of the interactions between the bar migration and the hydrodynamics is not covered. It is however an indication that there can be a lot of variation between different parts of the data set, possibly resulting in bad generalization when using neural networks.

B. Batch Size

As noted in section III-B, the time-series data of the alongshore average sandbar positions are not naturally segmented into independent batches, but the data can be arbitrary divided into batches of a certain size. It is however important to notice that the NARX neural network will not be able to learn long-term dependencies spanning different batches, so the batch size has to be larger than the longest-term dependency in the data. From Fig. 3 it becomes clear that the longest-term dependency according to a linear model will not be much larger than three or four days, but it cannot be used to determine what this value should be for nonlinear models. While the computation time for the data set decreases as the batch size grows larger, but the learning speed decreases as well for larger batch sizes, an optimal batch size has to be found empirically.

An experiment on a small part of the data (200 days) was performed to investigate how the learn speed changes with the batch size. Fig. 4 shows the course of the root-mean-square (RMS) error during learning, averaged over the training of thirty NARX neural networks initialized with different random weights, for different batch sizes. Whereas these networks had one hidden layer containing three units, different numbers of hidden units and different sizes of the total data set yielded similar results. The smallest batch sizes with the largest learning speed are found between four and thirty, so anywhere between these values is an appropriate choice for the batch size.

As becomes clear from Fig. 4, training the network takes an enormous amount of epochs, generally larger than two million. To train the network at a higher speed, a momentum factor can be used. This increased the learning speed with a factor ten, but resulted in less accuracy. Another method to speed up the learning by reducing the possibility to oscillate, is to fix the bias after a number of epochs, while the weights continue to be trained. This did indeed speed up the learning, but also reduced the learning capacity of the network. Other

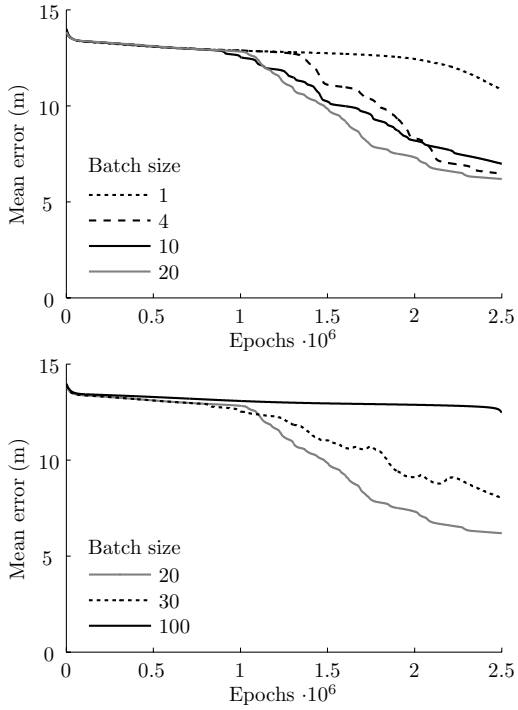
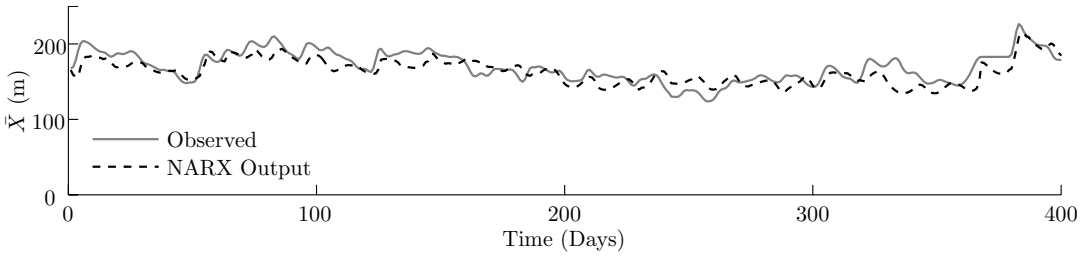
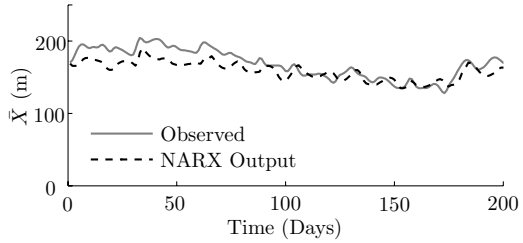


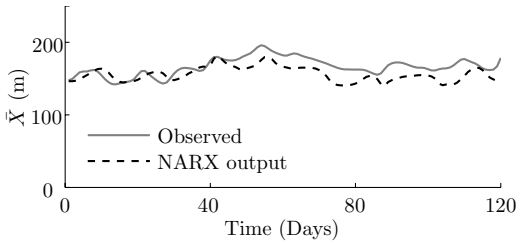
Fig. 4. Decrease in RMS error for different batch sizes



(a) Results on the train set

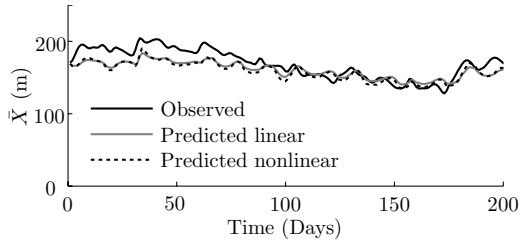


(b) Results on the validation set

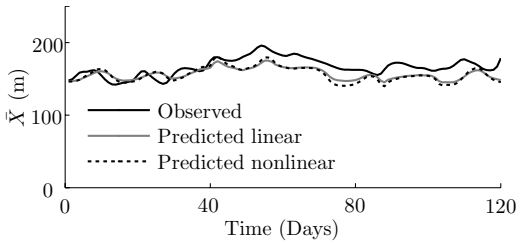


(c) Results on the test set

Fig. 5. Model output from NARX neural network versus observed positions for different sets



(a) Results on the validation set



(b) Results on the test set

Fig. 6. Model output from linear and nonlinear models versus observed positions for different sets

learning methods using second or higher order derivatives or approximates, such as quasi-Newton methods, can improve learning speed, but also decrease the learning capacity of recurrent architectures because of fast-vanishing gradient information over recurrent connections. Since the ability of the NARX neural network to learn long-term dependencies is studied, we will only use gradient descent based on first-order derivative.

C. NARX Results

In the next experiment we measure the performance of a NARX neural network to predict the migration of a sandbar given the hydrodynamic forcings. The network has two input units, one for H_{b_s} and one for z_{tide}, an additional recurrent input for the previous predicted $\bar{X}$, one output unit, and two hidden layers with eight units each (other configurations with one or two hidden layers and up to eight units in each hidden layer gave essentially the same results). The units in the input and output layer use linear squashing functions whereas the units in the hidden layers use sigmoid squashing functions. The network was trained with BPTT using gradient descent, for a number of epochs on one part of the data set; the training set, and subsequently tested on another part of the data set; the validation set. When the error in the validation set stopped to decrease, the network was supposed to start overfitting and the training was stopped. This process was repeated for fifty networks with different initialized weights.

Fig. 5 shows the observed bar positions together with the model output. The RMS error on the training set is 17 m, on the validation set 14 m, and on the test set 15 m. Note that the error on the training set is larger than on the validation and test sets, because position is 'clamped' only at the initial value and the training set is much larger than the validation and test sets.

D. Linear versus Nonlinear Networks

It is suggested in the literature that the interaction between the hydrodynamics and the morphology is highly nonlinear. To test this claim, the results of linear and nonlinear models can be compared to each other. If the aforementioned interaction is in itself nonlinear, a linear model has more difficulty to model the system than a nonlinear model. The linear and nonlinear models used here are two NARX neural networks, the former using linear squashing functions in all units, and the latter using linear functions in the input and output layers, but sigmoid functions in the hidden layers. The structure of the linear network is as described in section III-C, and that of the nonlinear network as described in section IV-C. To

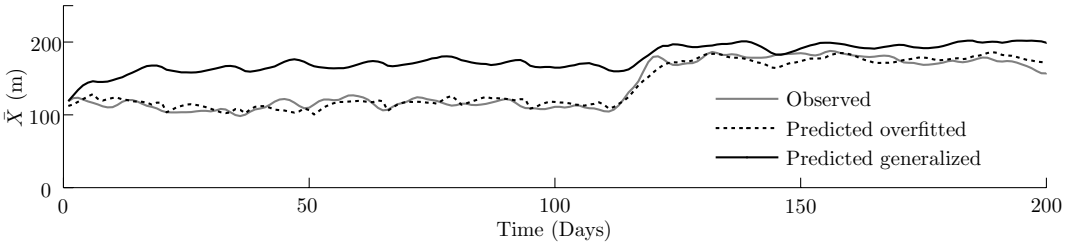


Fig. 7. Model output from generalized and overfitted models together with observed positions

obtain comparable results, both architectures were trained in the same fashion as the networks in section IV-C. The results of the model outputs together with the observed outer bar positions are depicted in Fig. 6.

As becomes clear from Fig. 6, there is not much difference between the results of the linear and the nonlinear networks. Both achieved a RMS error of 17 m on the training set, 14 m on the validation set and 15 m on the test set, with no significant difference in the 99 percent confidence interval.

E. NARX Neural Network versus Autoregressive MLP

The last experiment is performed to analyze the difference in learning between the NARX neural network architecture, and an autoregressive MLP as described in section III-B. The MLP is not capable of learning long-term dependencies in the data because the error is not backpropagated over the recurrent connection, in contrast to the NARX neural network. Both networks in this experiment use one hidden layer with five units and all other training parameters are the same. Training is performed in the same fashion as in the previous sections.

As in the previous experiment, there was no significant difference in performance between the two architectures on the training, the validation or the test set. This suggests that there are no important long-term dependencies in the time-series of alongshore average sandbar positions, or that the difference between the sets does not allow the networks to learn them, by complicating the learning process too much.

V. Conclusion and Discussion

The nearshore morphology at Gold Coast is a highly dynamic system that is very responsive to the hydrodynamic forcing. This behavior is reflected in the observed cross-shore positions of the sandbars. The preliminary analysis showed that the behavior of the sandbars in terms of linear dependence on the hydrodynamic forcings can be different for separate parts of the data set. One of the causes that might partially be responsible for this variation, is the small part of the bar (1.8 km) that was tracked from the total part of the bar ($>$ 4 km) available on the images.

Although all neural network architectures and training algorithms used in the experiments were able to capture some of the dynamics of the outer bar, generalization proved to be difficult. All models learned to predict approximately the same course of the bar positions, but the error remained quite large on all parts of the data set. However, the predicted positions of the models never showed any values beyond the bar zone (between 0 and 250 m) while this was not imposed by the linear nature of the output neurons.

The method used in the experiments in which training was terminated when the error in the *validation* set stopped to decrease, resulted in approximately equal errors for the training, test and validation set. Training the networks on a smaller part of the data set ($<$ 250 days) until the error on the *training* set did not decrease any further, resulted in much better performance on the training set, but decreased performance on other parts of the data. In Fig. 7 the predicted alongshore average position of the outer bar for a small part of the data set is depicted for several models. The generalized model, which was trained on another part of the data, performed rather poorly with a RMS error of 25 m, while the overfitted model, which was trained on the same data as shown in this figure, did much better with RMS errors of approximately 10 m.

A possible solution for the problem of bad generalization is to train a number of networks on different parts of the data set, and use each of the networks' output with a certain weight dependent on an analysis of the data set, comparing it with other data sets which were learned by the networks, to compute a weighted prediction. As becomes clear from Fig. 7 this would probably result in better performance, but is might also be the case that the dynamic character of the Gold Coast data does not even allow for such a generalization.

While the difference between the results of the models using several architectures and training algorithms is small and the experiments concern only one bar, the answer to the question whether the supposed nonlinearity and long-term dependencies in the Gold Coast data require a NARX neural network to capture the dynamics of the cross-shore sandbar migration remains uncertain. In this project, more simple models, such as the linear and the autoregressive MLP performed as well or sometimes even better. Although this might suggest a more complex neural network such as the NARX architecture is not required to capture the dynamics of the nearshore sandbars, other bars and other locations show possible less linear behavior. For example, the inner bar, which was not studied here, shows less direct response to the factors forcing the migration than the outer bar (see Fig. 2), indicating that its behavior might be difficult to capture with a simple model.

The neural networks had to be trained for an enormous

amount of epochs ($> 2 \cdot 10^6$) before reasonably good results were obtained. The gradient descent method used in the experiments is known to converge slowly. As mentioned before, other methods using second or higher order derivatives or approximates, can dramatically improve the learning speed, but also decrease the learning capacity of recurrent architectures. Feedforward networks suffer less from the problem of vanishing gradients because they don't employ recurrent connections. Considering the performance of the network architectures and algorithms in the experiments, simple models are better than complex models. Further research has to demonstrate if the performance of data-driven models, such as neural networks, can be improved by architectures that are nonlinear and can learn long-term dependencies.

ACKNOWLEDGMENTS

This work was supported by the Netherlands Organization for Scientific Research (NWO) under contract 864.04.007.

REFERENCES

[1] S. G. J. Aarninkhof and R. A. Holman, "Monitoring the nearshore with video," *Backscatter*, vol. 10, pp. 8–11, 1999.

[2] J. A. Roelvink and I. Brøker, "Cross-shore profile models," *Coastal Engineering*, vol. 21, pp. 163–191, 1993.

[3] N. G. Plant, R. A. Holman, M. H. Freilich, and W. A. Birkemeier, "A simple model for interannual sandbar behavior," *Journal of Geophysical Research*, vol. 104, pp. 15,755–15,776, 1999.

[4] L. D. Wright, S. May, A. D. Short, and M. Green, "Beach and surf zone equilibria and response times," in *Proceedings of the 19th International Conference on Coastal Engineering*, 1985, pp. 2150–2164.

[5] T. Lin, B. G. Horne, P. Tiño, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1329–1338, November 1996.

[6] I. L. Turner, S. G. J. Aarninkhof, T. D. T. Dronkers, and J. McGrath, "CZM applications of Argus coastal imaging at the Gold Coast, Australia," *Journal of Coastal Research*, vol. 20, pp. 739–752, 2004.

[7] T. C. Lippmann and R. A. Holman, "Quantification of sand bar morphology: a video technique based on wave dissipation," *Journal of Geophysical Research*, vol. 94(C1), pp. 995–1011, 1989.

[8] I. M. J. Van Enckevort and B. G. Ruessink, "Effects of hydrodynamics and bathymetry on video estimates of nearshore sandbar position," *Journal of Geophysical Research*, vol. 106, pp. 16 969–16 979, 2001.

[9] K. T. Holland, R. A. Holman, T. C. Lippmann, J. Stanley, and N. Plant, "Practical use of video imagery in nearshore oceanographic field studies," *Journal of Oceanic Engineering*, vol. 22, pp. 81–92, 1997.

[10] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, 1990.

[11] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, vol. 1, pp. 318–362, 1986.

[12] R. J. Williams and J. Peng, "An efficient gradient–based algorithm for on–line training of recurrent network trajectories," *Neural Computation*, vol. 2, no. 4, pp. 490–501, 1990.