

Detecting behaviours in ecological models

Fabio Boschetti

CSIRO, Marine and Atmospheric Research
e-mail: Fabio.Boschetti@csiro.au

Abstract

Given an ecological model, I address the question of how to identify the different dynamical behaviours it can exhibit. This requires three steps: the discrimination of different behaviours, the detection of novel behaviours in the model space given current knowledge on model behaviour and the display of the results for visual inspection. I propose simple heuristic algorithms to carry out these steps in the case of models generating time series. I test the method on three models of increasing complexity, analysing both local and global structures in the time series and demonstrating the flexibility of the approach.

Keywords: Modelling; anomaly detection; signal processing; time series analysis.

1 Introduction

Numerical modelling is increasingly being used to inform policy-making with examples including resource management, biodiversity conservation, global warming mitigation and economic policy. The interpretation of modelling results thus has the potential to profoundly affect our environment and millions of people.

When the modelled process is complex (that is almost always) and our understanding of the process is not complete (that is almost always) a model is at best an approximation of the underlying dynamics; in this situation the considerable amount of uncertainty in the model implementation, parameterisation and input translates in an equally considerable amount of uncertainty in the model output. This has led to an on-going discussion among modelling practitioners on what a model output represents, how it should be interpreted and what its overall scientific significance is: views cover a continuum between two extremes: one suggests that models can provide only a qualitative understanding of the modelled process and their output simply offers insight into general trends; another sees a model as a virtual laboratory in which real processes are roughly mimicked and whose outcome can be interpreted as predictions.

Somewhere in between these views, a number of practitioners suggest that the purpose of modelling is to explore the potential behaviours a system can display. In this work I adopt this view and in particular I aim to discriminate among model behaviours which, given a problem, appear to be qualitatively different. This approach fits nicely within a pre-cautionary approach to ecological problems aimed to inform policy-makers on the range of scenarios a policy may need to address.

There are three main challenges in implementing this approach: the first one is how to define and discriminate different behaviours. This is clearly problem-specific and depends not only on the purpose of the analysis but also on the kind of output a model produces. In this work I

assume a model generates a time series and I embed the time series in delayed-coordinate space (Takens, 1981; Kantz and Schreiber, 1999), in which I define a number of simple measures able to detect both local and global features in the model output; these features are then classified, discriminated and clustered into a smaller number of representatives of heuristically-defined, qualitatively different behaviours.

The second challenge is how to *detect* or discover different behaviours. When a model can be written in an analytical form, techniques to do so are well developed though not always easy to implement. It is not so when the model can not be written in closed-form but has to be solved numerically. This is the case I analyse here and I resolve to address it via a search in a high-dimensional input space.

Finally, once a set of different behaviours have been found, these need to be presented to the user and if many of such behaviours have been detected some sort of classification and simplification is also needed; also since these features may belong to high-dimensional spaces visualising the results is not trivial. I employ a clustering algorithm to simplify the results and a Self-Organised Map (Kohonen, 2001) to allow an approximate visual representation of the results. While these tools do not provide for an 'exact' analysis of the results, I believe they allow a potential decision-maker to obtain a rough picture of the variability and the range of behaviours the policies may need to address.

Algorithmically, the crucial component for the approach is the processing of the model output in the delayed-coordinate space. This is a common procedure in time series analysis, information theory and dynamical system analysis (Takens, 1981; Kantz and Schreiber, 1999), from which a number of ideas can be borrowed and adapted to different needs. I devise three different measures able to detect different kinds of behaviours, both local and global ones and I apply the method to three problems. The first one is chosen merely as demonstration; I simulate a population dynamics via a θ -Ricker map and define both search space and feature space in 2D, which allows to visualise each step of the approach and the details of its implementation. I then apply the method to the analysis of an NPZ model with the purpose of detecting different global behaviours and to demonstrate how the method can be extended to higher dimensional problems. Finally, I address a real-world application: the study of the recreational fishery of a marine park in Western Australia; this is carried out via a multi-species numerical model aimed at informing policy-making for the multi-use management of the park.

The approach results from the combined use of models, algorithms and heuristics, all of which are readily available in the literature. My main contribution lies in two items: first, in the details of the parameter space search aimed at detecting different model behaviours; second, and probably most important, in striving to simplify the presentation of the result in a graphical and intuitive form.

2 The approach

In this section I describe the numerical implementation of the approach. In order to make the description less abstract I take the reader through a working example: I employ a θ -Ricker map as a population dynamics model and I simplify the analysis so that all steps can be easily visualised in 2D.

I define the θ -Ricker map as in (Ellner and Turchin, 2005):

$$N_{t+1} = N_t \exp(r(1 - N_t^\theta) + \sigma W_t) \quad \text{Eq 1}$$

where N_t is the population at time t , r is the growth rate, W is a normally distributed random variable with mean 0 and variance 1, and σ is the intensity of the additive noise.

I aim to study the behaviour of the time-series (ts) generated by the θ -Ricker map (Rm in the following) as a function of a 2D input parameter space represented by r and θ , that is:

$$ts = Rm(r, \theta) \quad \text{Eq 2}$$

For this application I also assume that we are not interested in the overall behaviour of ts , rather in some of its local features (extension to the analysis of global features is discussed in the following section). This addresses the classes of problems for which local extreme values or local trends are important.

I assume I have a signal processing method which allows to extract local features from the time-series. Here I use possibly the simplest such processing method: I embed ts in a 2D delayed-coordinate space represented by the pairs $[ts_t, ts_{t-1}]$ (Takens, 1981; Kantz and Schreiber, 1999) and I call these *extracted features* in the rest of the document. Because of their construction, these *extracted features* inform us about extreme values of the time-series as well as of local 2-step trends, that is, local first derivatives. More complex features could be analysed by embedding ts in higher dimensional spaces as will be shown in the next examples, but I limit to 2D in this section to facilitate the visualisation of the procedure. This description leads to a natural way to define the difference between two *extracted features* as their Euclidian distance in the embedding space.

With this representation I can now describe and visualise the procedure used to extract features of interest. This is divided into three stages: in the first stage I attempt to describe the most likely model behaviour, where likelihood is understood, albeit loosely, in a Bayesian sense: these are the features we expect to be present in the output of the model given our knowledge and expectation on the input parameter (more informally, these are the features we expect to see in Nature or which we are used to seeing in the model output and represent what we *currently* ‘know’ of the model behaviour). In the rest of the document I call this ‘standard’ model behaviour. In the second stage I actively look for model behaviours which are as different as possible from the standard one and I define the behaviours I find as ‘anomalous’. Once again this should be understood within a Bayesian framework: these behaviours are anomalous only in relation of our previous knowledge of the model. Finally, in the third stage of the procedure I update our knowledge of the system behaviour by including the anomalous features into the set of standard ones and iterate, thereby adaptively expanding the scope of the search into the model behaviour space as well as redefining the understanding of the model and consequently the understanding of what represents standard and anomalous behaviour.

2.1 Defining ‘standard’ system behaviour

Problem-specific knowledge and experience may inform us on expected model behaviours and these can be represented as a collection of time-series generated by the model under expected input parameters. If this information is not available, we can run the model a number of times under random input parameters chosen within expected ranges and collect these time series. From these time-series I obtain a collection of *extracted features* as described above. In the case of the θ -Ricker map this results in a set of points in the 2D embedded space as can be seen in Figure 1.

Figure 1b shows the input parameter space of dimension θ, r containing 16 points used to generate 16 random time-series. From the time-series I obtain the 2D *extracted features* displayed in Figure 1a which shows the feature space of dimensions ts_t, ts_{t-1} . The dark points show all features contained in the initial 16 time series corresponding to the 16 input parameters in Figure 1b.

In order to summarise the information contained in Figure 1a, I group the points into a number of clusters (for the choice of the number of clusters see the Discussion section). I carry out the clustering via the VSH (Frey and Dueck, 2007) algorithm and the cluster centres are displayed in Figure 1a as 4 large white markers. I call the centres of these clusters *characteristic behaviours*, since by construction each represent a set of similar model behaviours; the set of these characteristic behaviours represent the ‘standard’ behaviour displayed by the Rm model according to the information so far collected. In Figure 1c I show the time series from which the characteristic behaviours have been extracted and the exact location of the characteristic behaviours along these time series. These correspond to the locations along the time series from which the extracted features, later selected as clusters by the VSH algorithm, were originally detected. The 4 characteristic behaviours seem to capture the structures contained in the time-series, which cover both average values (extracted features 1 and 4) and ‘extreme’ values (features 2 and 3) like peaks, valleys and sudden jumps (as seen in Figure 1c).

To be consistent all extracted features in Figure 1b should be defined as ‘standard’; to do so, I take the maximum intra cluster distance and draw a circle around each cluster centre with radius equal to such distance. Obviously, by construction, all points found so far fall inside one of the circles or, equivalently, so far we have no information about the existence of any features outside the circles. I then define as ‘anomalous’ features which fall outside such circles and I devise a procedure to search for those features.

2.2 Detecting ‘anomalous’ system behaviour

According to the definition in the previous section, searching for anomalous system behaviours coincides with searching for features in the extracted feature space which lie outside the circles in Figure 1a. We can carry out this search via a numerical optimisation method in which the fitness function is the distance between one feature and the closest characteristic behaviour (the fitness function needs to be maximised). Since I expect that more than one such feature may exist, a natural choice is to use a population-base search algorithm and in this work I use a real-coded Genetic Algorithm (GA, see (Davis, 1991; Forrest and Mitchell, 1993; Mitchell, 1998)) with a population size of 16 individuals (details of the specific Genetic algorithm used in this work can be found in (Boschetti *et al.*, 1996)).

After two iterations, the GA finds an anomalous feature, that is a feature not included within the ranges of the characteristic behaviours as defined above. This feature can be seen as a grey marker both in the extracted feature space in Figure 2a and in the corresponding time-series in Figure 2c. It represents an extreme value of the time series which had not been seen previously.

The presence of this new feature affects the optimisation process: first, it adds a new feature to the characteristic behaviours and consequently it changes the shape of the solution surface. Second, I want the newly discovered anomalous features to represent a new class of behaviours: if in the subsequent search another feature is found in the proximity of this anomalous feature, it will be disregarded if it is closer to an existing characteristic behaviour than this anomalous features and it will replace it otherwise. This procedure allows us to avoid having to re-cluster the points in the feature space every time a new anomalous feature is detected; this has the potential to considerably reduce the computational effort of the approach, given that the number of points in the extracted feature space, and consequently the clustering effort, increases at each iteration (obviously this benefit needs to be weighted against the computational effort needed to run the model (R_m in this case) which in ecological applications may vary from a fraction of a second to several hours and even days).

The optimisation then proceeds in this new mode. Figure 3a shows the result after 20 iterations. Two items are worth noticing: first, a new characteristic behaviour has been detected (marker 6 in Figure 3a and in the time series in Figure 3c). Second, the characteristic behaviour in marker 5 has been replaced by a new feature, very close to the previous characteristic behaviour 5 but further away from the characteristic behaviours found at the previous iteration (that is more ‘extreme’).

These two new characteristic behaviours add information to our previous knowledge of the system behaviour by informing us that larger values in the population dynamics are possible than previously thought. This information may have relevance for decision-making: if the modelled species was a pest, this result would warn us of a potential bigger threat than previously expected.

As a final consideration, the plot in Figure 3b illustrates the sampling of the input parameter space carried out by the Genetic Algorithm; as can be seen the sampling is skewed, the search has guided the GA towards area of the parameter space were anomalous behaviours are likely to be found, which in this case correspond to high values of both θ and r .

This was a simple test case since the full range of behaviours of the θ -Ricker map could be easily discovered without the need of the specific state space exploration I described. The purpose of this test case was merely to illustrate the proposed approach, which is summarised into a pseudo algorithm in the next section. More challenging test cases are then discussed in the rest of the document.

2.3 The algorithm

I summarise this approach by describing it in the form of a pseudo-algorithm and generalising it to higher dimensions:

- 1) choose the input parameter space I (define its dimensionality and the expected ranges for each parameter) and an embedding dimension M ;
- 2) if not already available, generate a set of output time-series by running the model on a number of random points from the input parameter space I ;
- 3) embed the time-series in a M -dimensional delayed-coordinate space; I call this feature space;
- 4) define a measure of similarity in feature space (in this work I used the Euclidian distance);
- 5) run the clustering algorithm and detect N initial characteristic behaviours (with N defined by the clustering algorithm or set a priori by the user);
- 6) calculate the range, that is the maximum intra cluster distance, which defines the separation between ‘standard’ and ‘anomalous’ features;
- 7) start the Genetic Algorithm optimisation; at each iteration:
 - a. for each new model output time-series;
 - i. embed the time series into the feature space;
 - ii. if any of the extracted featured is outside the range of the characteristic behaviours, accept it as a potential new characteristic behaviour;
 - iii. if the potential new characteristic behaviour is not within the range of another new potential characteristic behaviours, add it to the list of characteristic behaviours;
 - iv. if the potential new characteristic behaviour is within the range of another new potential characteristic behaviours, choose the one farther away from an existing characteristic behaviour;
 - b. end
- 8) end

A flowchart of the pseudo algorithm is displayed in Figure 4.

3 Detecting behaviours in an NPZ model

In this second example I employ a more complicated model, the NPZ model from (Edwards and Brindley, 1999); a description of the model is included in Appendix A, together with the list of input parameters and their ranges. Unlike the previous example, here I focus on the global structure of the output time series, rather than on local features; in particular, for global structure I refer to the ensemble of features present in a time series and develop a measure to evaluate how different two time series are based on these ensembles. Similarly to (Boschetti, 2008), the final aim is to determine how many different behaviours the NPZ model may display and roughly partition the input state space based on these output behaviours. To study the behaviour of the NPZ model I focus on the time series of phytoplankton biomass (Taken’s theorem (Takens, 1981) guarantees that, provided I embed the time series in a sufficiently high dimensional space, the study of the phytoplankton biomass provides information about the overall behaviour of the NPZ system).

As in the previous analysis of the θ -Ricker map, I embed the time series into a delayed-coordinate space. In this case I choose an embedding dimension of 7, which allows us to analyse longer structures and higher derivatives. Unlike the previous example however, I do not analyse the extracted features in isolation, rather I consider the ensemble of all the extracted features from an individual time series, which corresponds to a set of points in the embedded space (the concept is summarised in Figure 5). In order to carry out the algorithm

as described in the previous section I thus need a measure of the difference between ensembles of extracted features rather than between individual extracted features.

As before, I define this difference as a distance in delayed-coordinate space. This difference can be defined in many different ways: in (Boschetti, 2008) I used the difference between the statistical complexity (Crutchfield and Young, 1989; Shalizi and Shalizi, 2004) of the time series, which is calculated from the ensemble. Many other information-theoretic measures could similarly be employed (Kantz and Schreiber, 1999; Schreiber, 2000; Ray, 2004). Most of these measures require the time series to be symbolised (that is discretised), either before or after embedding. As discussed in (Daw *et al.*, 2003; Kennel and Buhl, 2003) the symbolisation imposes an arbitrary distortion on the time series which I try to avoid in this work; consequently, here I define a distance which is based solely on the geometrical location of the extracted features in embedded space. Given two ensembles $e1$ and $e2$ of extracted features, the distance from ensemble $e1$ to ensemble $e2$ is defined as the mean distance from each feature in $e2$ to the closest among the features in the $e1$; to clarify:

- 1) for each extracted feature $f2_i$ belonging to ensemble $e2$
- 2) look for the feature $f1_j$ in ensemble $e1$ with shortest Euclidean distance $|f1_j, f2_i|$, call it $d_{i,j}$
- 3) the distance from ensemble $e1$ to ensemble $e2$ is then defined as:

$$D_{e1 \rightarrow e2} = \text{mean}(d_{i,j}) \quad \text{Eq 3}$$

A geometrical explanation of this procedure is given in Figure 5. In general, this distance is not symmetric, that is $D_{e1 \rightarrow e2} \neq D_{e2 \rightarrow e1}$: as can be seen from Figure 5 the distance from the ‘more complex’ time series (which generates the ensemble with more diverse extracted features) tends to be shorter than the distance in the opposite direction. This measure can be interpreted as the amount of distortion we need to impose to ensemble $e1$ in order to reconstruct ensemble $e2$. This lack of symmetry is not a concern in our application since the clustering algorithm I adopt is able to account for it (Frey and Dueck, 2007).

Given this distance, I can now proceed with the algorithm as described in section 2.3. I run the NPZ model 40 times with random input from a 6 dimensional parameter space (the chosen input parameters are marked with an asterisk in Table 1, where their range is also reported). From these runs the clustering algorithm selected 4 time series summarising the ‘standard’ behaviour of the model. I then run the GA for 260 further model run (13 iterations of a population of 20 individuals) looking for time series as different as possible from the currently stored time series. The algorithm found an additional 8 time series which I define as ‘anomalous’. To summarise, I sampled 300 points from the input parameters space, stored 300 time series and from them extracted 12 which characterise the main behaviours arising from the model.

In this test case both the input parameter and the feature space are high dimensional and consequently it is difficult to visualise the results. To circumvent the problem, I employ a self-organised map (SOM), (Kohonen, 2001) as in (Boschetti, 2008). A SOM maps vectors in a high-dimensional space into a lower dimensional space (2D in our case) by respecting the vector neighbourhood topology, that is, by plotting along side points which are close in the original high-dimensional space. The result can be seen in Figure 6. The central plate shows the SOM U-Matrix (Kohonen, 2001): this does not have a specific physical interpretation, which is why the axis are not labelled, rather it should be understood as the display of an

acceptable arrangement of the 300 points sampled in the input parameter space such that their distance in the 2D plate is as close as possible to their distance in the original 6D parameter space. Over the U-Matrix I plotted the 12 points in the input parameter space which generated the 12 time series detected by the search process; I interpret these 12 points as representative of the behaviour of the NPZ model and partition the U-Matrix by assigning the remaining points to one of these representatives according to their ensemble distance.

In Figure 6 I plotted the time series corresponding to each of the 12 selected points, which shows the different behaviours we may expect from the NPZ model. In many applications it also of interest to know what parameters are responsible for these different behaviours. Since I have sampled 300 values from the input parameters space I may employ this sampling to extract some rough information from them. I attempt to do this visually in Figure 7.

Following a method tested in several other studies involving searches in high dimensional spaces (Boschetti et al., 2003; Wijns et al., 2003a; Wijns et al., 2003b; Boschetti, 2005; 2008), the value for each of the 6 input parameters is mapped over the U-Matrix and interpolated; this results in the 6 plates in Figure 7 which give us an approximate visual description of how each parameter varies in the domain characterised by different behaviours. Obviously, these plates are the result of a process prone to error: first points are arranged from 6D to 2D, then an interpolation is performed in the resulting 2D space (interpolating in 2D points originally lying in a 6D space imposes a further distortion to the image; this is carried out for the sole purpose of facilitating visual analysis); we thus can not expect an exact outcome from this analysis, rather only a rough impression which may lead us to insights for further analysis. This also will be further discussed below.

Figure 7 suggests that the Predation on P (plate b) seems to be responsible for the partition between the behaviours in domains 1, 10 and 11 versus the rest of the domain. Similarly, plate (e) seems to suggest the role of concentration of N on the behaviour of domains 2 and 11. Domains 3, 5 and 7 seems to share fairly similar values in plates a, b and e, suggesting that their different behaviour may be due to the value of the parameters in the other plates, particularly c and f. Similarly, the high value in plate c seems to be responsible of the behaviour in domain 9. Because of the distortions in the generation of these maps these insights should not be taken as conclusive, rather may suggest directions for further enquiry.

4 An example application: fisheries management in a marine park

Our final example is taken from a real world problem: the management of the fishery in the Ningaloo Marine Park, a 300 km long fringing coral reef in the northwest coast of Western Australia. The model has been developed as part of a study aimed at devising a set of fishing regulations that will ensure a sustainable future for the park, in which recreational fishing currently represents one of the main drivers for local tourism. Previous initiatives have established a number of sanctuary zones and imposed a set of recreational fishing regulations, as well as a complete ban on any form of commercial fishing within the park. The need for an evaluation of the effectiveness of current regulations and sanctuary zones and for the assessment of the possible need to adjust or redefine management objectives has lead to the development of a model to support management decisions.

This model subdivides the park into a number of zones in accordance with both ecological and administrative constraints and within each zone simulates ecological processes as well as

recreational fishing carried out according to the modelled regulations. A brief description of the model is given in Appendix B, while more details can be found in (Boschetti *et al.*, 2008).

The approach proposed in this paper can be used to inform decision-makers of the possible regimes of biomass evolution of a number of species of interest as a function of variables under management control. In order to address this specific requirement I modify slightly the algorithm described in the previous sections and in doing so I further demonstrate the flexibility of the method.

Current studies on the marine park are focussed on the assessment of two species of interest: Spangled Emperor (*Lethrinus nebulosus*) and Chinaman Cod (*Epinephelus rivulatus*). The output time series of biomass from this model display less ‘structure’ or local variability than the ones arising from the NPZ model, that is, local trends are much smoother. Consequently the biomass value at the end of the simulation is fairly representative of the global biomass evolution during the modelled period (no sudden crashes followed by recovers have been noticed in our modelled runs). I thus considered as output of our model a 2D embedding consisting of the values of biomass for Spangled Emperor and Chinaman Cod at the end of the modelled period. This is a considerable simplification, of course, which no longer guaranties that the embedding satisfies the requirements of Taken’s theorem; thus these embedded features no longer represent a compression of the model behaviour over its entire dynamics (over the full output time series) but simply provide a representation of the final status of the system in term of the two species of interest.

I also assume that the two species, which underwent several decades of fishing, are at 50% of the carrying capacity at the beginning of the simulation. I then run the model for 5 year and check the level of final biomass compared to carrying capacity for both species. Figure 8 shows the outcome the analysis of model behaviour (involving 1000 runs controlled by the Genetic Algorithm) as described above. The horizontal and vertical lines show the 50% of carrying capacity threshold (at the beginning of each run the biomass of each species is at the centre of the picture). The GA searched an 8D space characterised by variables under the control of fishery management, which include: the total size of sanctuary zones, the number of fishing licences allowed in the park and the quota and minimum and maximum legal length for both species of interest. In this parameter space the GA searched for standard and anomalous features as described above.

Two features are obvious from the figure: first, there is a fairly strong correlation between the final biomass of the two species; second, the output of the model varies considerably, from complete crash of both species to both species reaching full capacity. In our analysis I want to try to understand which of the variables under management control has the largest effect on the final biomass.

I try to obtain a qualitative answer to this question via a visual analysis which takes advantage of the fact that the plot of the model behaviour (Figure 8) is two-dimensional. Take a model input variable, the total size of closure areas for example (plate a in Figure 9), that is one of the eight variables under the management control; in correspondence of each sample of the model output (that is each point in the Spangled Emperor - Chinaman Cod biomass plot in Figure 8) I plot the value held by that variable in the simulation which resulted in that specific output; I then interpolate these values in the 2D space. The result, for all 8 input variables, can be seen in Figure 9; this helps to obtain an approximate understanding of the relation

between input and output variables in our simulation (as discussed in relation to Figure 7 this interpolation imposes a distortion and is carried only to facilitate visual analysis).

The interpretation of Figure 9 suggests the following: first, many different combinations of input parameters seem to be responsible for fairly similar final biomass values; this appears to be true both for low and high final output biomass. This can be deduced from the texture of the plots which is much less smooth than that of the plates in Figure 7. This may be due to two different effects: the compensating effect of the other variables and the inherently stochastic behaviour of the model (see Appendix B). Second, of all variables under management control, the one which shows the clearer correlation with output biomass is the number of allowed fishing licences (plate b), which strongly influences the fishing effort. This is the only plate which shows a recognisable trend from the lower left corner of the Spangled Emperor - Chinaman Cod biomass plot (mostly red colours = high values) to the top right corner (mostly blue colours = low values). Even in this case though, the trend is not very smooth, showing that the effect of the number of fishing licences can also be compensated for by combination of values of other variables.

Two clear messages arise from Figure 9; first, in choosing among a number of options for management intervention, priority should be placed on intervention on the number of licences allowed in the park since fishing effort is the main factor controlling the amount of biomass in the park for the two species under analysis; the management has considerable power to influence this outcome. Second, consideration needs to be given to monitor other factors affecting the status of the stock since, as shown in Figure 9, natural variability and different combinations of other parameters may be able to cancel out or considerably reduce the efficacy of management intervention; for example, loosening control on minimum and maximum legal size may considerably reduce the effect of limiting the allowed number of licences. As a consequence of the latter, more simulations may be required to further explore the above mentioned variability; for example, the starting biomass of Spangled Emperor and Chinaman Cod (assumed to be at 50% of the carrying capacity in the current simulation) may be varied to evaluate whether management intervention is robust to stock status; if not, a stock assessment may be required before further decision making. Finally, as a result of the level of uncertainty in the interpretation of the plots in Figure 9, an adaptive management approach may be chosen, whereby monitoring is carried out after the initial decision on the suitable number of allowed licences to see whether the status of the stock, together with other restrictions on quota and legal sizes are indeed affecting the expected results and whether further adjustments are needed. The prioritization of intervention on the number of licences and the realisation that no option for intervention can be safely discarded are the main outcomes of the analysis, which would not have been obvious with a satisfactory exploration of the model input space.

5 Discussion

As mentioned in the Introduction, the approach here described employs a number of algorithms. Most of them have been proposed in the literature to address very difficult problems for which experts do not yet agree on a 'best practise'. These include the embedding method, the clustering algorithm (VSH), the global search algorithm (Genetic Algorithm) and the dimensionality reduction algorithm (SOM). The same applies to a number of parameters controlling the functioning of the above algorithms, the choice of which requires heuristic tuning; this includes: the choice of the number of initial clusters, the proper embedding

dimension, the time lag in the choice of the delays used in the embedding (above this is assumed to coincide with the discretisation interval in the time series, which implies that we assume the time series has been discretised in order to minimise the mutual information between samples), the stopping criteria for each algorithm (including the stopping criterion for the GA), the population size of the GA and the initial number of model runs with random input used to generate the initial clusters. Obviously, each of these components could be changed and affect the outcome of the method. The results here presented have been obtained by tuning some of these parameters, but should it be judged worthwhile to extend the general approach to other problems more testing and problem-specific tuning will be necessary.

When the model is not deterministic, as in the case of the fishery model in the previous section, a given set of input parameters may result in different output values or time series. This can complicate all steps of the analysis: the definition of the clusters, the search in the model parameter space as well the visualisation of the results and the use of error bars to account for this may be beneficial, since they would provide a manager with an idea of how reliable the expectation of a certain result might be.

In general, the proposed approach should not be seen as an avenue to obtain firm problem-independent answers on a model behaviour, rather as a tool to highlight difference in model behaviours and provide their rough categorization. This information can then be used to guide a more focussed search of the model space aimed at answering problem-dependent specific questions in more details.

I applied this approach only to model output in 1D representing time series of single values. Extensions to 2D and 3D are in principle possible (Shalizi et al., 2004), though they involve much heavier computation, provided these consist of sequences of profiles or images ordered in time (animations). Should the output not have this time component, issues may arise in implementing the algorithms meaningfully and care is needed (Feldman and Crutchfield, 2003). In particular, the problem of how to analyse static images is closely related to current applications in image processing, image recognition and artificial intelligence aimed at classifying large images data bases or recognise specific events requiring human attention, for example in surveillance. Improvements in those fields are slow but steady and may be soon relevant to ecological applications.

6 Conclusions

By embedding the time series generated by an ecological model into delayed-coordinate spaces of various dimensionality, we can define simple measures of structure similarity which allow to discriminate both local and global features, thus suggesting a basic classification of model behaviour. Given that classification, we can then explore the model input parameter space for novel structures, thereby carrying out a search specifically targeted at detecting novel behaviours. Depending on the dimensionality of the problem, different visualisation techniques can then be used to provide the user an approximate, qualitative understanding of the model's different dynamical behaviours. The approach has been tested on three models of increasing complexity and dimensionality and it can be easily modified to study specific features in model input and outputs.

7 References

- Boschetti, F., 2005. Controlling and investigating cellular automaton behavior via interactive inversion and visualization of the search space. *New Generation Computing*, 23:157-169.
- Boschetti, F., 2008. Mapping the complexity of ecological models. *Ecological Complexity*, 5:37-47.
- Boschetti, F., de La Tour, A., Fulton, E. and Little, R., 2008. Interactive modelling for natural resource management Submitted.
- Boschetti, F., Dentith, M.C. and List, R.D., 1996. Inversion of seismic refraction data using genetic algorithms. *Geophysics*, 61:1715-1727.
- Boschetti, F., Wijns, C. and Moresi, L., 2003. Effective exploration and visualization of geological parameter space. *Geochemistry Geophysics Geosystems*, 4:-.
- Crutchfield, J.P. and Young, K., 1989. Inferring Statistical Complexity. *Physical Review Letters*, 63:105-108.
- Davis, L., 1991. *Handbook on genetic algorithms*. Van Nostrand Reinhold.
- Daw, C.S., Finney, C.E.A. and Tracy, E.R., 2003. A review of symbolic analysis of experimental data. *Review of Scientific Instruments*, 74:916-930.
- Edwards, A.M. and Brindley, J., 1999. Zooplankton mortality and the dynamical behaviour of plankton population models. *Bulletin of Mathematical Biology*, 61:303-339.
- Ellner, S.P. and Turchin, P., 2005. When can noise induce chaos and why does it matter: a critique. *Oikos*, 111:620-631.
- Feldman, D.P. and Crutchfield, J.P., 2003. Structural Information in Two-Dimensional Patterns: Entropy Convergence and Excess Entropy,. *Physical Review E*, 67:051104.
- Forrest, S. and Mitchell, M., 1993. What makes a problem hard for a genetic algorithm? Some anomalous results and their explanation. *Machine Learning*, 13:285-319.
- Frey, B. and Dueck, D., 2007. Clustering by Passing Messages Between Data Points. *Science*, 319:972-976.
- Kantz, H. and Schreiber, T., 1999. *Nonlinear Time Series Analysis*. Cambridge University Press, Cambridge.
- Kennel, M.B. and Buhl, M., 2003. Estimating good discrete partitions from observed data: symbolic false nearest neighbors. *Physical Review Letters* 91:084102.084101-084102.084104.
- Kohonen, T., 2001. *Self-organizing maps*. Springer, New York, 5001 p.
- Mitchell, M., 1998. *An Introduction to Genetic Algorithms*. MIT Press.
- Ray, A., 2004. Symbolic dynamic analysis of complex systems for anomaly detection. *Signal Processing* 84:1115—1130.
- Schreiber, T., 2000. Measuring Information Transfer. *Phys. Rev. Lett.*, 85:461-465.
- Shalizi, C., Shalizi, K. and Haslinger, R., 2004. Quantifying Self-Organization with Optimal Predictors. *Physical Review Letters*, 93:118701
- Shalizi, C.R. and Shalizi, K.L., 2004. Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences. *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference*. AUAI Press, Arlington, Virginia.
- Takens, F., 1981. Detecting strange attractors in turbulence. *Dynamical systems and turbulence*, Warwick pp. 366-381.
- Wijns, C., Boschetti, F. and Moresi, L., 2003a. Inverse modelling in geology by interactive evolutionary computation. *Journal of Structural Geology*, 25:1615-1621.
- Wijns, C., Poulet, T., Boschetti, F., Griffiths, C. and Dyt, C., 2003b. Interactive Methodology Applied to Stratigraphic Inverse Modelling. In: A.C.a.R.W. (Eds.) (Editor), *Geological Prior Information: Value and Quantification*. Geol. Soc. of London Special Publication, London.

Appendix A

NPZ model. Here I briefly describe the nutrient-phytoplankton-zooplankton (NPZ) model used in this study. For more details I refer the reader to Edwards and Brindley (1999). The specific equations used are:

$$\begin{aligned}\frac{dN}{dt} &= -\frac{N}{e+N} \frac{a}{b+cP} P + rP + \frac{\beta\lambda P^2}{\mu^2 + P^2} Z + \gamma qZ + k(N_0 - N) \\ \frac{dP}{dt} &= \frac{N}{e+N} \frac{a}{b+cP} P - rP - \frac{\lambda P^2}{\mu^2 + P^2} Z - (s+k)P \\ \frac{dZ}{dt} &= \frac{\alpha\lambda P^2}{\mu^2 + P^2} Z - qZ\end{aligned}\quad (F1)$$

where N , P and Z are nutrient, phytoplankton and zooplankton respectively, with units of gCm^{-3} . The model parameters, units and ranges are described in Table 1. The parameter ranges have been selected by Edwards and Brindley after extensive literature review (see Edwards and Brindley, 1996, pp 351-353).

Table 1. Parameter used in the NPZ model. The asterisks refer to the 6 parameters employed in the 6D parameter space search. For these parameters we give the allowed range. All other parameters are kept constant at the indicated default value.

Parameter	Symbol	Default value and ranges
a/b gives maximum P growth rate	a	0.2 $m^{-1} day^{-1}$
Light attenuation by water	b	0.2 m^{-1}
P self-shading coefficient	c	0.4 $m^2 (g C)^{-1}$
Half-saturation constant for N uptake	e	0.03 $g C m^{-3}$
Cross-thermocline exchange rate*	k	0.0008–0.13 day^{-1}
Higher predation on Z *	q	0.015–0.150 day^{-1}
P respiration rate*	r	0.05–0.15 day^{-1}
P sinking loss rate*	s	0.032–0.08 day^{-1}
N concentration below mixed layer*	N_0	0.1–2.0 $C m^{-3}$
Z growth efficiency*	α	0.2–0.5
Z excretion fraction	β	0.33
Regeneration of Z predation excretion	γ	0.5
Maximum Z grazing rate	λ	0.6 day^{-1}
Z grazing half-saturation coefficient	μ	0.035 $g C m^{-3}$

Appendix B

Fishery model. The purpose of the fishery model used in Section 4 is to estimate the expected effect of a number of fishery regulation options which are available to the management team. These management options control the number of dispensed fishing licences, the total areas reserved as sanctuary zones, daily catch limits, and minimum and maximum legal lengths for two species of interest. Once such regulations are defined, the model mimics the behaviour of a fishing fleet (in this case of recreational fishers) and the effect of their catch and by-catch on the modelled species. The output is then presented to the user in the form of time series of biomass for the different species.

The model run is stochastic, in the sense that certain ecological parameters, for I have no firm a priori information, are allowed to vary within given ranges. The purpose is to capture, to some extent, the uncertainty in the model parameterization resulting from the lack of precise biological data as well as the inherent uncertainty of biological and ecological processes. This choice aims to inform the users of the range of variability which I may expect in the modelled response even given precisely defined fishery regulations.

A detailed description of the Simulation Model implementation can be found in (Boschetti *et al.*, 2008); here I briefly describe the main modules.

Ecological module accounts for trophic relations among different species; at present five fish species within a 3-level food-web are modelled for each environment/zone. This includes:

- 1) a lower trophic level prey, representing the basic food source for the entire foodweb.
- 2) 3 intermediate species; these include the two species targeted by recreational fishing, whose dynamics are the focus of the study, as well as a competitor which undergoes possible by-catch;
- 3) a top predator, which is also accounted for as by-catch.

The size of the foodweb is constrained for the sake of computational speed.

A fishing module models the fishing behaviour, the access to the fishing zones, the sharing of the catch among vessels targeting the same zone, the effect of gear selection and the choice of the target species.

An economic module models the fishers' decision making; fishers store their past record of catches and choose which fishing zone to target according to a prediction of what the most profitable zone might be in the next iteration. The prediction can be carried out either by attempting to maximise the catch of each individual vessel or by accounting for the behaviour of the overall fleet.

A fishing regulation module; this defines the fishing regulations at each fishing zone and represents the management strategy options (input space of the model as discussed in Section 4). These include the extent of sanctuary zones, the number of fishing licences allowed, and the bag limits, and legal minimum and maximum length for two species of interest: Spangled Emperor (*Lethrinus nebulosus*) and Chinaman Cod (*Epinephelus rivulatus*). The module can be extended easily to include other regulatory and assessment criteria.

Figures

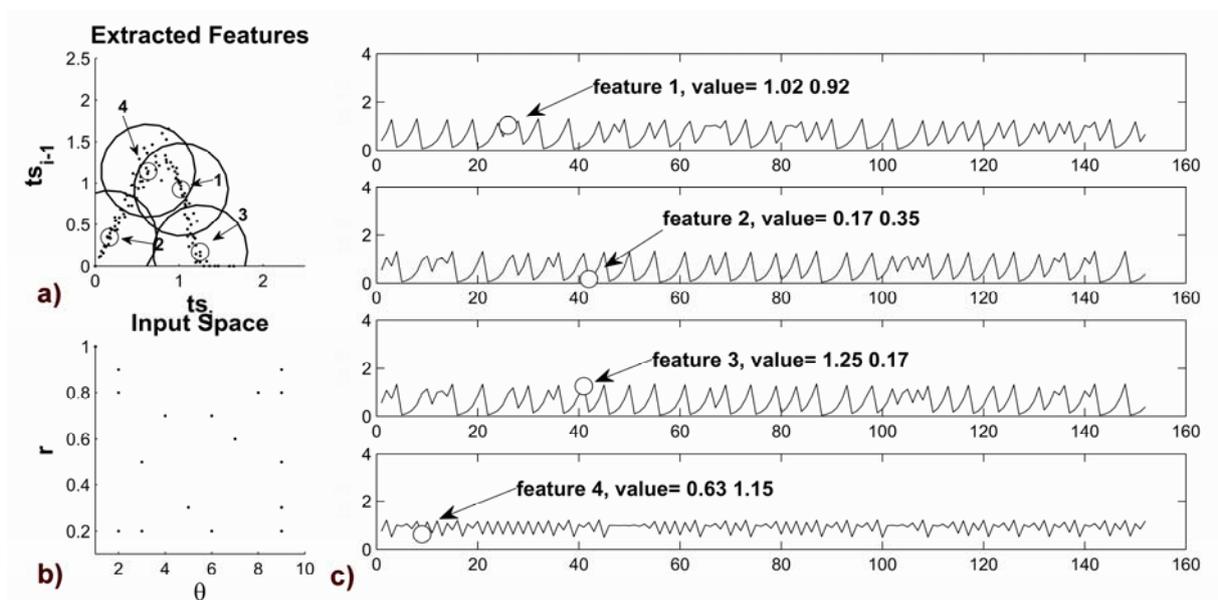


Figure 1. Detection of ‘standard’ features in the θ -Ricker map; b) 16 random points from the θ - r input parameter space; a) 2D delayed-coordinate feature space with all features extracted from the 16 time series; the white markers show the representatives selected by the clustering algorithm; c) time series from which the 4 representatives have been selected, and their locations along the time series themselves.

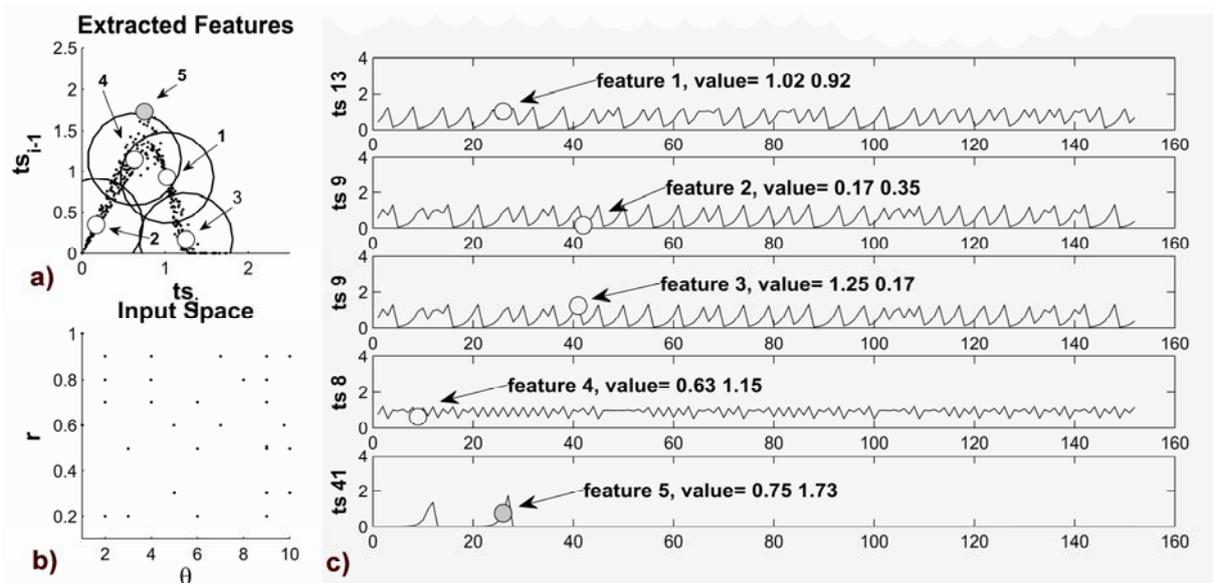


Figure 2. Detection of ‘anomalous’ features in the θ -Ricker map; a) point 5 (grey marker) has been detected outside the range defined by the 4 ‘standard’ features; c) its position along the time series.

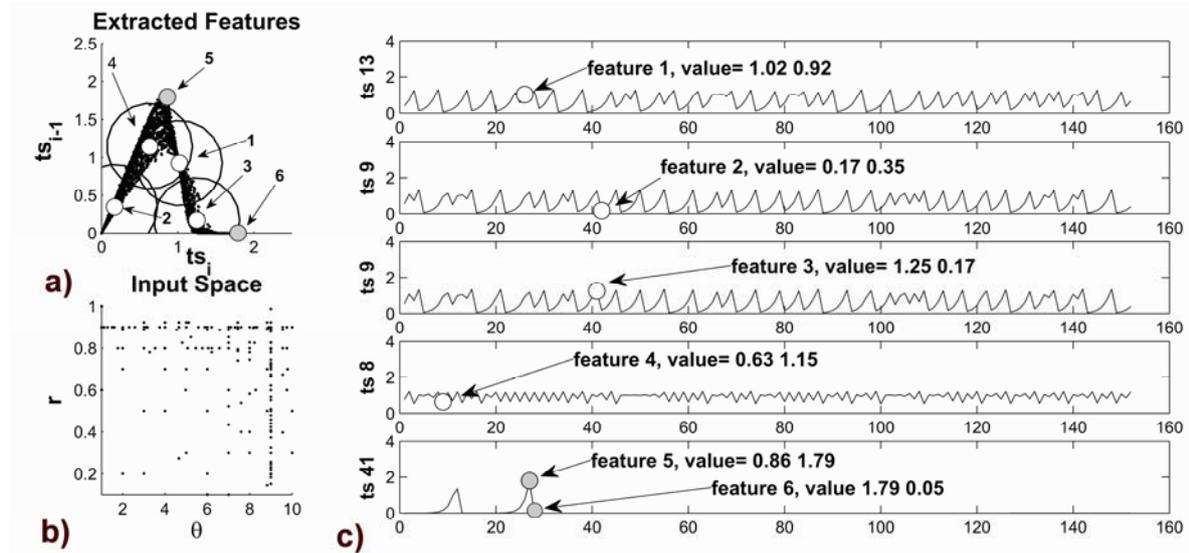


Figure 3. Final result of the analysis of the θ -Ricker map; a) a further ‘anomalous’ point (6) has been found while point 5 has been moved further away from the 4 characteristic behaviours; b) the final sampling of the parameter space is not uniformly spaced rather focussed on extreme values of θ and r ; c) locations of the detected features along the time series they belong to.

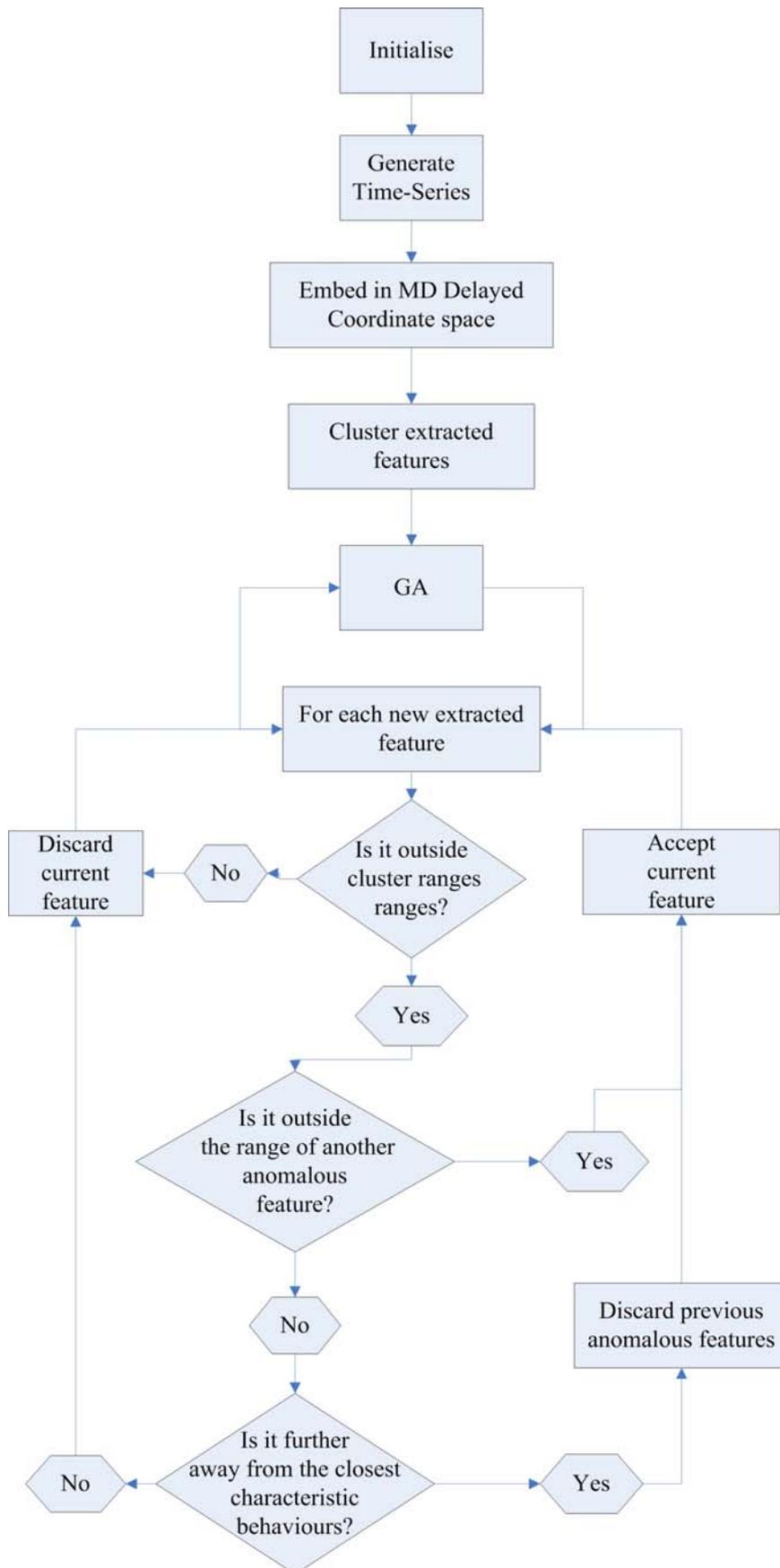


Figure 4. Flowchart of the proposed algorithm to detect different model behaviours.

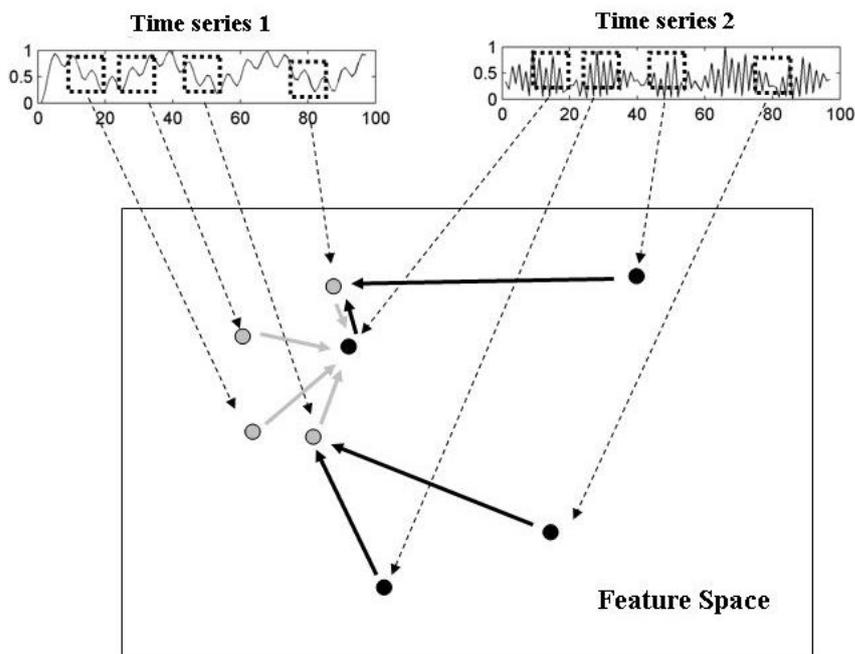


Figure 5. Time series distance used in the analysis of the NPZ model: 1) two time series are embedded in the N-dimensional delayed-coordinates Feature Space (gray dots = features from time series 1, black dots features from time series 2); 2) from each feature belonging to time series 2 (black dots) I measure the shortest distance to a features in time series 1 (gray dots); the black arrows give the set of these shortest distances; $D_{e1 \rightarrow e2}$ is then defined as the mean length of these arrows. 3) Similarly, $D_{e2 \rightarrow e1}$ is defined as the mean length of the gray arrows. I notice that $D_{e2 \rightarrow e1} < D_{e1 \rightarrow e2}$, that is a) the distance is not symmetric and b) a simple time series tends to be farther from complex one than the distance in the opposite direction.

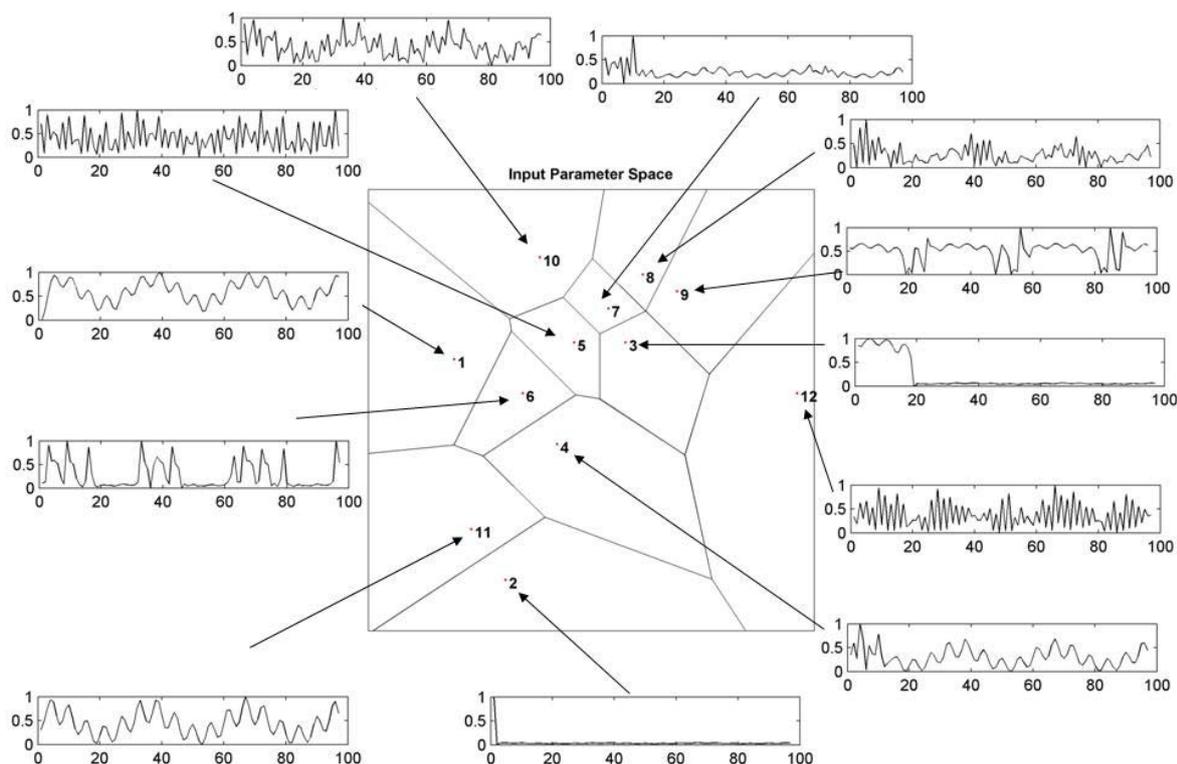


Figure 6. 12 different behaviours displayed by the NPZ model and their location on the SOM 2D representation of the model input parameter space.

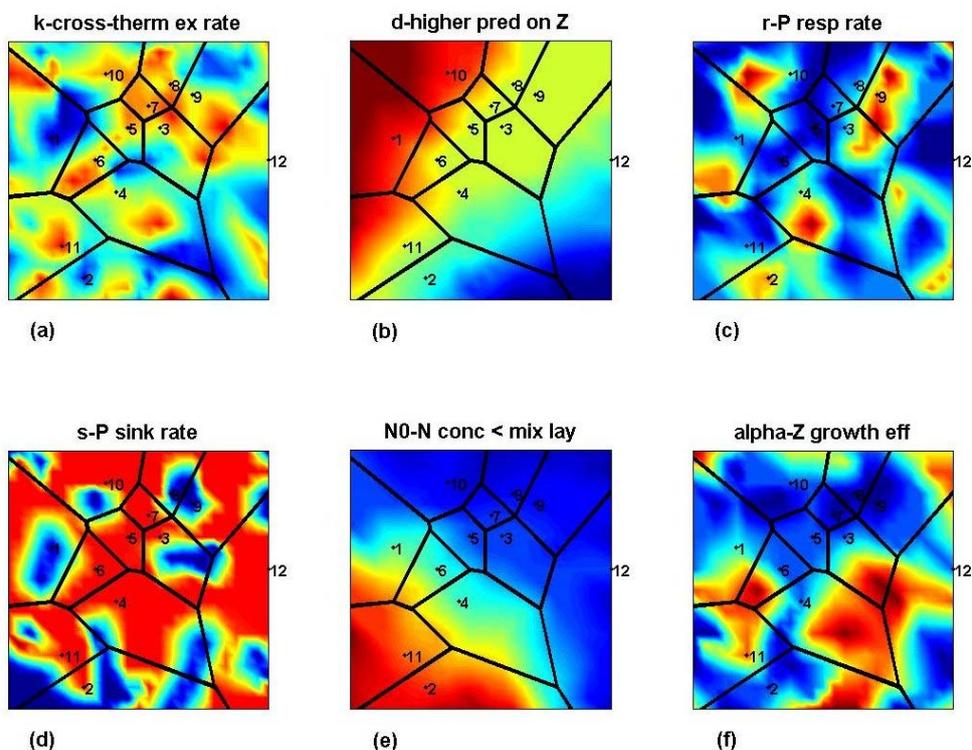


Figure 7. Values of each of the 6 NPZ model input parameters interpolated over the SOM 2D representation on the input parameter space. The axis are not labelled, since they have no physical interpretation, see description of SOM in the text for details.

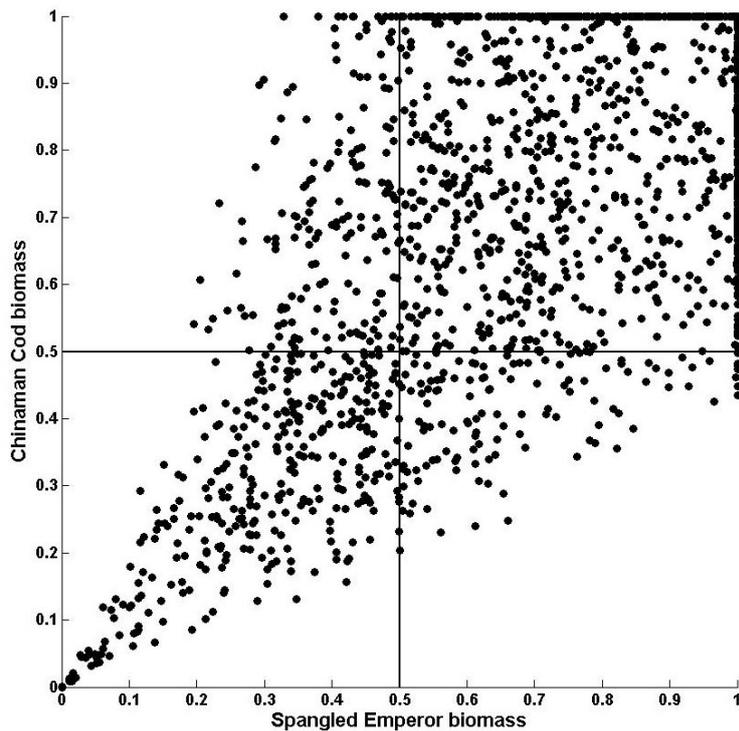


Figure 8. Final values of Spangled Emperor and Chinaman Cod biomass normalised to carry capacity. Starting biomass was 50% of carrying capacity for both species. Results from 1000 runs. The search was carried out in a parameter space including only variables under management control.

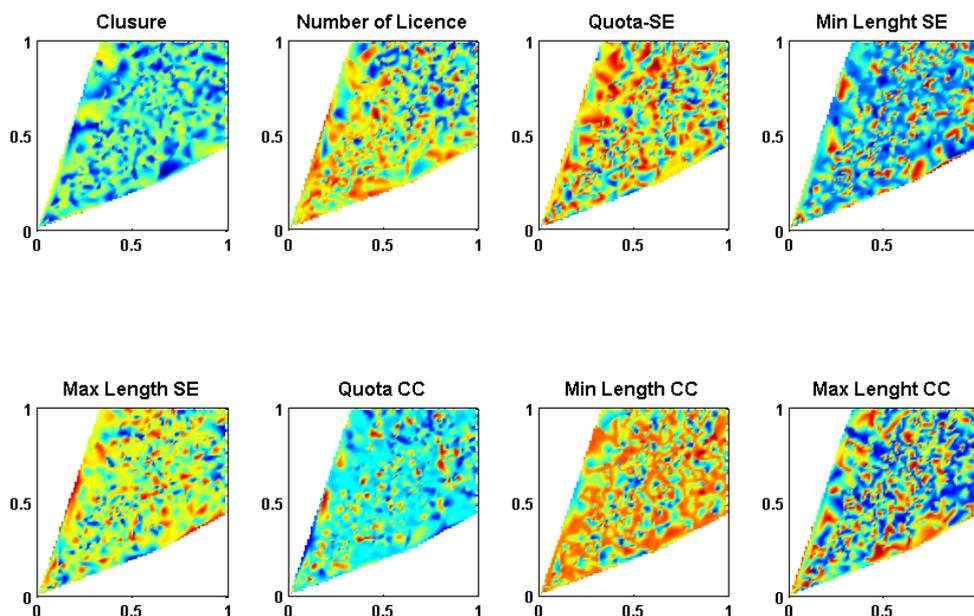


Figure 9. Values of the eight variables under management control mapped over the plot of Spangled Emperor-Chinaman Cod final relative biomass. The plates allow to gauge an approximate understanding of the effect of the input variables on the output biomass. The axes are not labelled, since they have no physical interpretation, see description of SOM in the text for details. In the plate labels SE stays for Spangled Emperor and CC for Chinaman Cod.