

# A Delay-Embedding Approach to Multi-Agent System Construction and Calibration

Campbell, Alex and Pham, Binh and Tian, Yu-Chu

Computational Intelligence Group, Faculty of Information Technology  
Queensland University of Technology, GPO Box 2434, Brisbane Qld 4001, Australia. E-Mail: ab.campbell@qut.edu.au

*Keywords: Delay-Embedding, Simulation, Calibration*

## EXTENDED ABSTRACT

Agent-based modelling offers a way to break from the crude assumptions of mean-field type models, which ignore space correlations between elements of the system and replace local interactions with uniform long-range ones. Multi-Agent Systems (MAS) explicitly model spatially distributed individuals; however the richness of such a model can also be a liability due to the sensitive dependence of such high-dimensional systems. This has implications for choice of MAS architecture, programming of rules, confidence in predictions, and calibration of model parameters.

*Delay-embedding*, also known as *geometry from a time series*, provides a deep theoretical foundation for the analysis of time series generated by nonlinear deterministic dynamical systems. The profound insight of embedding is that an accessible variable can explicitly retrieve unseen internal degrees of freedom. [3].

In the domain of complex systems modelling, however, there typically exist an abundance of observables, in which case reconstructing hidden degrees of freedom may be problematic or even nonsensical. Also, many observables often implies high dimensionality, which generally precludes a dynamical systems approach in the first instance. Uncautious use of delay-embedding, from which it is easy to get a result regardless of physical justification, has in the past led to a degree of negative press for this idea.

However, the recent extensions of Takens' delay-embedding theorem to deterministically and stochastically forced systems [9, 8, 10] provide a rigorous framework in which to reconstruct using multiple observables. This holds great significance for pattern discovery in complex data series, which we define to be more than one series - spatial, temporal or a mixture - of an underlying complex system. In particular, the concept of a *bundle embedding* highlights a way to usefully employ the 'surplus' observables in the embedding process. More generally, forced embeddings provide a methodology to breakdown complex system data sets in a modular

fashion, while still retaining nonlinear relationships.

Cluster-Weighted Modelling is a sophisticated approach to density estimation that, when applied to the output of a delay-embedding process, is able to obtain a statistical representation of the dynamics. These two concepts - forced embeddings and density estimation - provide a promising theory and a practical probabilistic interpretation respectively to the 'inverse problem' of system identification.

Expert-knowledge based MAS construction and density-estimation of delay-embedded data can therefore be thought of as two complementary approaches to the goal of bottom-up, complex systems modelling. The original contribution of this paper is to present the latter as a highly data-driven approach to MAS construction in its own right, and, perhaps more importantly, as an aid to constructing and calibrating the more expert-knowledge rule-driven approach. The emphasis is on a solid theoretical and conceptual foundation.

To illustrate the feasibility of our approach, preliminary implementation results for an ecological modelling scenario are presented and discussed.

## 1. INTRODUCTION

The analysis of time series data has benefited greatly from the concept of *delay-embedding*, also known as *geometry from a time series* after the paper in which it was introduced by Packard *et al.* in 1980 [6]. Under reasonable technical conditions this allows the reconstruction of a multi-dimensional phase space from a series of a single observable only [11].

Recent progress in extending delay-embedding theory to ‘forced’ systems (perturbed or driven by some external influence) provides a platform for more general application of this principle. One significant result coming from this work is a sound justification for modelling high-dimensional spatially extended systems as a collection of subsystems at different spatial locations coupled together, in a sense driven by the ‘noise’ of the neighbouring systems.

Expert-knowledge based MAS construction and density-estimation of delay-embedded data can be thought of as two different approaches to the goal of bottom-up, complex systems modelling. Furthermore, they are complementary in a number of specific ways. For example, one problem for the application of delay-embedding is that generally the training data requirements are exponential in the dimensionality of the dynamics. In a MAS the data is generated by a simulation, it is therefore possible to analyse higher dimensional data.

Delay-embedding, forced delay-embedding, and cluster-weighted modelling are introduced and integrated into a single spatial-temporal systems identification approach in Section 2. A conceptual framework for multi-agent system construction and calibration is presented in Section 3. We apply this to an ecological scenario with some real data in Section 4. In Section 5 we conclude and discuss future work .

## 2. DELAY-EMBEDDING AND CWM

### 2.1. Standard Delay-Embedding

Assume that at each point in time a system is determined completely by a point  $\mathbf{x}$  lying in a  $m$ -dimensional *phase space*, which we denote  $M$ . Technically  $M$  needs to be a *manifold*<sup>1</sup>, and is often some subset of  $\mathbb{R}^n$ . The time evolution of such a system (the movement of the point through phase space) is given by a map  $f^\tau : M \rightarrow M$  such that  $\mathbf{x}_{t+\tau} = f^\tau(\mathbf{x}_t)$ .

Takens’ theorem [11] says that ‘almost every’ smooth map  $\Phi : M \rightarrow \mathbb{R}^d$ ,  $d \geq 2m + 1$ , is an *embedding*. This means that the map  $\Phi$  can get to every point

in its ‘image’,  $\Phi(\mathbf{x}_t)$ , that each point reached has a unique starting point in  $\mathbf{x}_t$ , and that it (the map) is differentiable in both directions. Essentially this imposes a certain degree of equivalence between the spaces. The term ‘almost every’ captures what is meant by *generic* in the field of dynamical systems, and is defined formally, but in practice this turns out not to be a major obstacle. Because  $d \geq 2m + 1$  is a sufficient condition, an embedding may be possible for  $d < m \leq 2m$ .

The consequences of this are far-reaching. We can define the map,

$$F = \Phi \circ f \circ \Phi^{-1}, \quad (1)$$

which can be seen as the same dynamical system as  $F$  under the coordinate change given by  $\Phi$ . Because of the equivalence mentioned above, coordinate-independent quantities on  $F$  and  $f$  such as Lyapunov exponents and correlation dimension (and many others) will be identical.

Given a smooth observation function  $h : M \rightarrow \mathbb{R}$ , and setting  $o_t = h(\mathbf{x}_t)$ , we define an instance of  $\Phi$  as

$$\begin{aligned} \Phi_{f,h,\tau}(\mathbf{x}_t) &= (h(\mathbf{x}_t), h(f^{-\tau}(\mathbf{x}_t)), \dots, h(f^{-(d-1)\tau}(\mathbf{x}_t))) \\ &= (o_t, o_{t-\tau}, \dots, o_{t-(d-1)\tau}) \end{aligned} \quad (2)$$

This provides a snapshot of the system  $\mathbf{z}_t = \Phi_{f,h,\tau}(\mathbf{x}_t)$  in this new space,  $\mathbb{R}^d$ , the dynamics of which are given by  $F$ . Remarkably,  $F$  is defined entirely in terms of a  $\tau$ -shift in the observations,

$$F^\tau(o_t, o_{t-\tau}, \dots, o_{t-(d-1)\tau}) = (o_{t+\tau}, o_t, \dots, o_{t-(d-2)\tau})$$

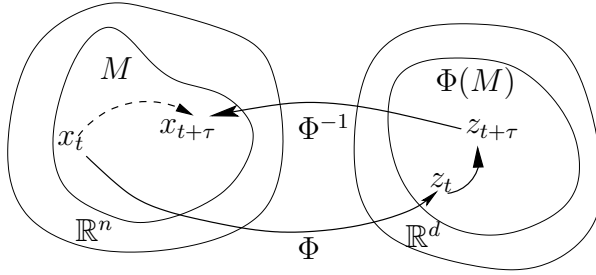
This means that we can deduce many of the properties of  $F$ , and hence of  $f$ , purely in terms of the observed time series, and despite not knowing any of the dynamics *a priori*. From a forecasting perspective this is also very strong result: the next state has only one unknown,  $o_{t+\tau}$ , which is a function of the  $d$  previous values,

$$o_{t+\tau} = G(o_t, o_{t-\tau}, \dots, o_{t-(d-1)\tau}), \quad (3)$$

and can be estimated from training data to forecast future values the time series. The coordinate transform is illustrated in Figure. 1.

To make practical use of delay-embedding, suitable values for the delay  $\tau$ , and the embedding dimension  $d$ , need to be identified. There is a large body of literature on this. The first minimum of the *mutual information* is popular for finding a suitable delay [2], and the embedding dimension can be estimated via the *correlation dimension* or a multi-dimensional extension of mutual information called the *redundancy* [3].

<sup>1</sup>A manifold is a space that locally looks like a ball in  $\mathbb{R}^n$ .



**Figure 1.** The coordinate transform  $\Phi$

## 2.2. Forced Delay-Embedding

The theory outlined above is a remarkable result, however the assumptions that the system is deterministic (it has no random aspect) and autonomous (it is not ‘driven’ or ‘forced’ by any external events) are not justified for many real world applications. In fact real world systems are almost always forced by something: even if you are considering the whole planet to be your dynamical system, there would still be external forcing from, for example, solar activity and meteors.

There have since been a number of generalisations of the theory, most significantly a method for reconstructing input-output systems conjectured by Casdagli [1] and subsequently proved by Stark *et al.* [9]. Building on this, Stark and colleagues give an thorough presentation of a number of forced embedding theorems in [8] and [10], covering both deterministic and stochastic forcing. These extensions dramatically increase the degree to which real world data can be considered from a formal basis. The forced theorems are given for the discrete time case, so we will use  $i \in \mathbb{Z}$  rather than  $t \in \mathbb{R}$  and,  $f^i$  is just  $f$  composed  $i$  times.

A standard approach to modelling a non-autonomous system in dynamical systems theory is to replace it with a new, autonomous system, constructed by including the driving dynamics. In other words we are expanding the phase space to include the driving. A standard model is

$$\begin{aligned} x_{i+1} &= f(x_i, y_i) \\ y_{i+1} &= g(y_i), \end{aligned} \quad (4)$$

where the pair  $f : M \times N \rightarrow M$  and  $g : N \rightarrow N$  is known as a *skew product* on  $M \times N$ .

Stark *et al.* prove that the map  $\Phi_{f,g,h}$  is generically an embedding, provided some conditions on the periodic orbits of the forcing are satisfied. This requirement may cause problems in an experimental situation where the sampling is done commensurate with the forcing, but in a real-world data situation it is not an

issue.

This is a very interesting result, implying that even if we have only one observable it is possible to reconstruct dynamics of the whole (expanded) phase space. But the dimensionality of the reconstructed phase space is necessarily increased by the dimension of  $N$ , which is a problem for complex data, and it does not usefully employ knowledge of more than one observable.

Rather than requiring  $\Phi_{f,g,h}$  to embed  $M \times N$ , Stark *et al.* asked what if we simply require that we can embed each  $M \times \{y\}$ . If knowledge of  $y_i = g^i(y)$  is available, then

$$\begin{aligned} z_i &= (o(x_i), o(f^{-1}(x_i, y_i)), \dots, o(f^{-(m+1)}(x_i, y_i))) \\ &= \Phi_{f,g,h}(x_i, y_i) \end{aligned}$$

This is what Stark *et al.* call a *bundle embedding*. The name ‘bundle’ arises from a technical concept in topology which do not go into details of, but essentially it means that it is possible to just reconstruct  $M$ , as long as each embedded point is indexed by the value of the forcing at that time. This means that the map

$$o_{i+1} = G_{y_i}(o_i, o_{i-1}, \dots, o_{i-d+1}) \quad (5)$$

is identical to the forecasting map of the standard delay-embedding theorem, except for an index of the forcing. By using our additional data it is then possible to estimate  $G_{y_i}$  as a function from  $\mathbb{R}^d \times N$ . Most significantly, the required dimensionality of the reconstructed phase space remains  $\geq 2m + 1$ .

One final problem with this model is the assumption that the forcing is deterministic and that  $g : N \rightarrow N$  is known. For real world data the types of forcing that we want to consider are best modelled as stochastic variables. For details on the proof of this and the other embedding theorems we refer the reader to [8] and [10]. Practically speaking, a powerful way to deal with (and learn) probability densities is provided by the machine learning technique Cluster-Weighted Modelling.

## 2.3. Cluster-Weighted Modelling

Despite the theoretical implications of delay embedding - that it is possible to handle highly non-linear behaviour of arbitrary physical systems with hidden dynamics - it is still very difficult to deal with the errors and uncertainty in real data. This is where a probabilistic approach is essential. Cluster-Weighted Modelling is a sophisticated approach to learning probability density functions in the presence of nonlinear, and even nonstationary dynamics. The framework is based on density estimation around

Gaussian kernels which contain simple local models describing the system dynamics of a data subspace [3].

The essential idea underlying the CWM procedure is to approximate an unknown model,  $M$ , of the dynamical system by a set of local nonlinear models denoted  $C_k$ ,  $k = 1, 2, \dots, K$ . The model,  $M$ , outputs the time series  $y$  given input data  $\mathbf{x}$ , and is constructed as a Gaussian mixture over a suitably chosen set  $\{C\}$ . Each local model  $C_k$  is obtained via fitting a nonlinear function, namely, by obtaining a set of parameters  $\beta_k$  in suitably chosen nonlinear maps

$$y = f(\mathbf{x}, \beta_k) \quad (6)$$

For generality, the maps  $f$  are taken to be polynomial functions. The joint probability distribution  $p(y, \mathbf{x})$  is expressed as a sum over the densities coming from each local model, — where  $\mu_k$  are the cluster means. Since the input and output data are related via Eq. 6, this gives an output distribution of the form

$$\begin{aligned} p(y, \mathbf{x}) &= \sum_{1 \leq m \leq M} p(\mathbf{x}, y, c_m) \\ &= \sum_{1 \leq m \leq M} p(y|\mathbf{x})p(\mathbf{x}|c_m)p(c_m) \end{aligned} \quad (7)$$

The probability of a given local model is denoted  $p(C_k)$ , with the usual normalization  $\sum_{k=1}^K p(C_k) = 1$ . Denote by  $\mathbf{P}_{\mathbf{x},k}$  and  $\mathbf{P}_{y,k}$  the covariance matrices for the input and output data, and by  $\mathbf{D}_x$  and  $\mathbf{D}_y$  the input and output dimensions. A useful choice for the input distribution  $p(\mathbf{x}|C_k)$  is a Gaussian density,

$$p(\mathbf{x}|c_k) = \frac{|\mathbf{P}_{\mathbf{x},k}|^{1/2}}{(2\pi)^{\mathbf{D}_x/2}} e^{-(\mathbf{x}-\mu_k)^T \mathbf{P}_{\mathbf{x},k} (\mathbf{x}-\mu_k)/2} \quad (8)$$

where  $\mu_k$  are the cluster means. Since the input and output data are related via Eq. 6, this gives an output distribution of the form

$$p(y|\mathbf{x}, C_k) = \frac{\mathbf{P}_{y,k}}{(2\pi)^{\mathbf{D}_y/2}} e^{-[y-f(\mathbf{x},\beta_k)]^T \mathbf{P}_{y,k} [y-f(\mathbf{x},\beta_k)]/2} \quad (9)$$

The posterior probability, which can be computed through Eqs. 6,8, and 9 as

$$p(C_k|y, \mathbf{x}) = \frac{p(y, \mathbf{x}|C_k)p(C_k)}{\sum_{j=1}^K p(y, \mathbf{x}|C_j)p(C_j)}, \quad (10)$$

is maximized with respect to model parameters  $\beta_k$  which can be recursively estimated by an iterative search procedure, the expectation-maximization algorithm. For further details (especially regarding how this provides a measure of the conditional uncertainty), see [3].

Perhaps the most important advantage CWM brings is a transparent architecture: the parameters are

easy to interpret, and by definition the clusters find ‘interesting’ places to represent the data. Another is its flexibility: it is possible to include more than one type of local model and let the clustering find where they are most relevant. This has particular relevance for hypothesis testing, as we will see below. Also, it can model nonstationary data sets by including absolute time as another input: the clusters will then spread out to describe different periods in time. This concept extends to space as well, again, this is particularly relevant for our purposes.

## 2.4. Spatially Extended Systems

The basic idea, presented in [4], is that a spatially extended system can be modelled as a number of local spatial subsystems weakly coupled to, and driven by, the ‘noise’ at their boundaries. In other words  $M$  is a local spatial region, and it forms the part of the whole system from which we can take observables. Therefore our delays may also have a spatial component - this corresponds to recording values of neighbours. The forcing by the rest of the lattice is modelled similar to the standard forced system above, except that the update of the forcing is now itself dependent on  $\mathbf{x}$ . This makes intuitive sense: if the point in  $M$  is updated, the forcing from the lattice must be updated, because  $M$  is part of the lattice. This means that it is no longer possible to have an *embedding* in the rigorous, technical sense of the word. However, because the effect  $M$  has on the forcing dynamics is generally small, Orstavik and Stark felt that the delay embedding technique still provides the correct intuition. This is supported by experimental results in their paper, and in a number of other studies before and since, many of which have found that the reconstruction technique to work surprisingly well. Interestingly, it is delays in both space and time (rather than just time or just space) for which the best results were found. The forced delay-embedding theorems of the previous sections began as an outline of a generalisation of Takens’ theorem to account for this and apply to systems which are only ‘approximately’ low dimensional.

The consequences of all this is a unified framework for dealing with arbitrary forced dynamical systems, that has a truly holistic treatment of space and time. To see this, consider an example prediction map  $G$ :

$$h(\mathbf{x}_i^r) = G_{\mathbf{x}_i, \omega_i}(h(\mathbf{x}_{i-1}^r), h(\mathbf{x}_{i-1}^{r-1}), h(\mathbf{x}_{i-1}^{r+1})) \quad (11)$$

where we are using superscripts to denote spatial delay. The function we would like to learn,  $G$ , treats spatial delays and temporal delays indiscriminately (though of course it does not have to, practically speaking weighting the number of delays and their distance differently for space vs time may make more sense).

In the next section we show how these features - forced ‘bundle’ delay embeddings, cluster-weighted modelling, and spatial forcing - can be combined to provide a prescription for constructing a MAS.

### 3. EMBEDDINGS FOR MULTI-AGENT SYSTEM CONSTRUCTION AND CALIBRATION

The prescription for MAS construction by Delay-Embedding and Cluster-Weighted Modelling (the DECWM approach) is as follows. Using CWM (or some other form of function approximation technique), and given a sufficient amount of low noise training data, it is possible to learn the mapping  $G$  given above (even if only for a region of its input-output space). This is done initially for one variable, which we call the ‘target’ variable. For example this may be house-prices on a grid in an urban economic modelling scenario. Once learned, this will then act as a ‘rule’ of a cellular automata type model. Following the bundle embedding concept, we extend  $G$  to include other ‘explanatory’ variables as forcings, for example spatially explicit demographic data. For each additional dimension induced by the spatial or temporal delays, the forcing must be considered. However, crucially this does not increase the dimensionality of the space into which we are embedding. The dimensionality of the dynamics and the delay magnitude must be determined by the methods mentioned earlier, for example the mutual information and correlation dimension. Lower dimensionality indicates greater ability to make predictions.

The novel aspect to the use of the bundle embedding concept is that the more data that is ‘explanatory’ of the target variable is available, the less dynamics has to be explained through target variable itself, and the lower the dimensionality.

Generally speaking this will be inadequate as a stand-alone approach for real world data. Firstly it just provides a model for one spatiotemporal variable. It therefore must be trained again on other ‘target’ variables to build models for them (although the possibility of incrementally synthesising ‘coherent’ models by doing so is an interesting prospect we intend to focus on in future work). Secondly it obviously is limited by the availability and quality of data. However, the combination of this with the conventional Programmed Rules (PR) approach can be useful in three distinct ways:

1. As a validation tool. If there is high enough quality data to get decent predictions on real world data then the DECWM method can be used to validate a PR model by training it on the output of a PR model and running it to show

similar probabilistic behaviour. This is useful because the PR rules may be much easier to interpret.

2. As a complementary approach to PR MAS construction when there is limited and/or poor quality data. In this guise it is designed to be used in parallel with PR modelling in an iterative fashion. See Figure 2
3. As an analysis tool for ‘artificial’ MAS, for which data is not available, or is not necessary. Here the link between the PR and DECWM approach is closest.

Both approaches have their strengths and weaknesses: consciously programmed rules are of course easier to understand and interpret, however inference from data often is more accurate in terms of predictions if there is sufficient good quality data available.

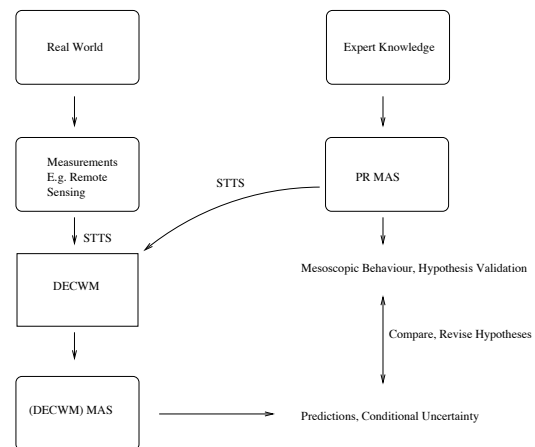


Figure 2.

### 4. AN EXAMPLE

We consider a continuous version of a cellular automata model known as a coupled map lattice. Although cellular automata are highly simplified compared to most MAS, they do retain the essential elements. They are a prototypical form of MAS, and therefore well suited to this preliminary study.<sup>2</sup>

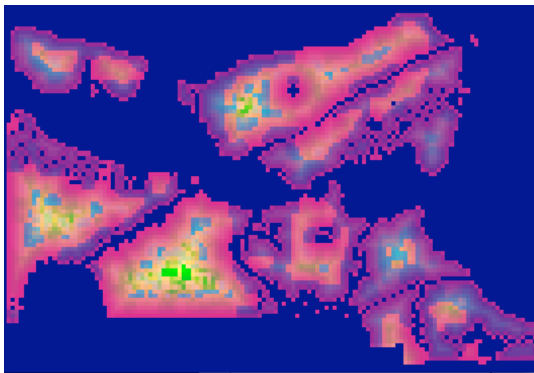
Data for the ecological scenario we consider comes from a study on wading birds in North America. A particular data set was first considered by Ozesmi [5] in an ecological modelling context, and then taken up as a spatial data mining problem by Shekhar et. al. [7]. It consists of three explanatory variables: “distance to open water”, “water depth”, and “vegetation durability”, though we only use the distance to open water variable in this study.

<sup>2</sup>More images and implementation details can be accessed at <http://sky.fit.qut.edu.au/~campbeab/decwm.html>

**MAS Simulation** The MAS we simulate to generate our space-time data is based on the nonlinear Hénon map:

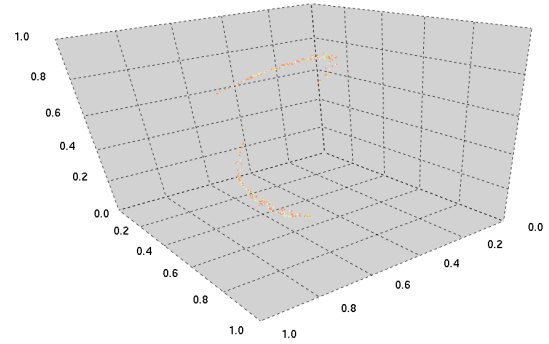
$$x_{i+1}^r = 1 - \alpha((1 - \mu)x_i^r + \mu \frac{N(x_i^r)}{4})^2 + \beta y_i^r \quad (12)$$

where  $x$  and  $y$  correspond to a two-dimensional target system at site  $r$  and time step  $i$ .  $N(x_i)$  is a neighbourhood function, returning the value of  $x$  at the four sites adjacent to  $r$ .  $\mu$  is the spatial coupling, and for model parameters  $\alpha = 1.45$  and  $\beta = 0.3$  we are in the chaotic region. The target system is mildly forced by the “distance to open water” variable through the beta parameter - i.e. the parameter beta will vary across the lattice. A snapshot of the simulation is given in Figure 3.



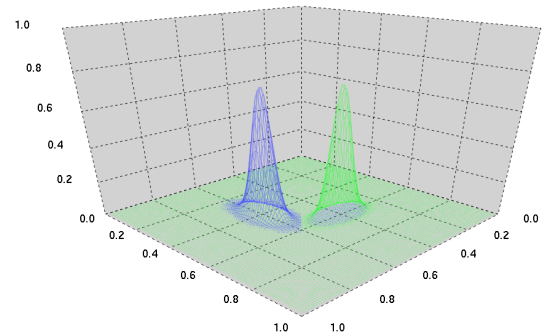
**Figure 3.** Snapshot of simulation. State variables and landscape variables represented by red-green-blue level of each cell. The ‘agent’ variables are red and blue, the distance to open water is green.

**Delay-Embedding.** In Figure 4, the observed variable is being measured with only one delay in time, i.e. we are synthesising a two-dimensional system. Any two of the axes represent these two degrees of freedom, with the third recording the value of the one dimensional time series at the next time-step. In fact this redundancy was just done to give us training data - you can see that the third dimension is not really required here. Experiments with spatial delays are being carried out - we do not have any visualisations for them as yet, but we still make some comments in this regard. Because the dynamics are translationally invariant it is not necessary for one spatial delay to give us four additional dimensions. For more complicated dynamics that are still rotationally invariant (an assumption which is valid for many systems) we suggest using a form of averaging or even principal component analysis on the spatial delays. Technically the time and space dimensions should multiply to give us a four-dimensional embedding space, however in many cases an approximation of three (the more ‘distant’ space-time dimension is not used) seems to give comparable results.



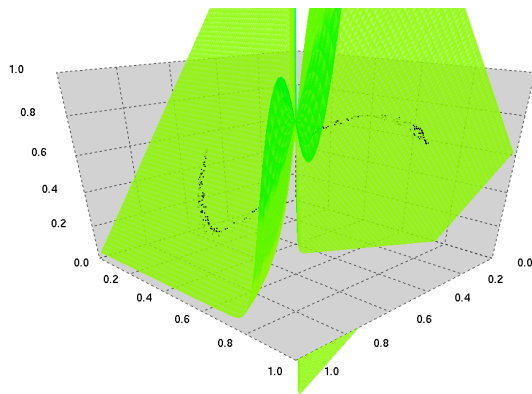
**Figure 4.** Delay Embedding

**Cluster-Weighted Modelling.** Cluster-weighted modelling of the delayed variables using two clusters is shown in Figure 5. As mentioned, it is possible



**Figure 5.** CWM with two clusters

to include absolute space as an input to the model; clusters will then spread out to describe different spatial areas. This combines nicely with the ability to program different functional relationships between inputs and outputs for the different clusters: the cluster or clusters with that feature should move to that spatial region. This provides a good (perhaps rather challenging) test for that knowledge/intuition. For example in the current scenario we may believe that a special relationship exists along the boundaries between the open water and the edge of the marshland, and that this particular spatial region (even though it is a distributed spatial region) can be described very simply, though very nonlinearly. An example would be that the probability for the existence of the agent is high only if the three or more of the surrounding sites are not open water, else it is low. In these preliminary stages, we only are interested in showing qualitative results. Nevertheless, it is encouraging that good predictions are being obtained. In Figure 6, the original data in our extended embedding space is overlaid with the ‘prediction surface’ obtained by the CWM procedure. Note the closeness of the surface to the data points, and the nonlinearity afforded by the use of two clusters.



**Figure 6.** Prediction Surface

**Implementation** The iterative, interactive nature of the construction process, and the fact that it is based on an inherently parallel model (cellular automata), means that significant computational resources are required. Additionally, the approximation we have made here that the target variable (the ‘agent’) is continuous valued most probably has to be relaxed with the use of a Monte-Carlo approach. With this in mind, programmable graphics hardware has been used to accelerate the computations. Graphics hardware is well suited to the complex systems paradigm as many concepts and models therein are based on the parallel execution of many local models; this is more or less a summary of the architecture of such hardware. It is also a good visualisation tool, which becomes useful when contemplating three-D (and higher) embedding spaces.

## 5. CONCLUSION AND FUTURE WORK

In this paper we have proposed a forced delay-embedding framework for the analysis of spatiotemporal time-series and begun to compose a methodology for its use in construction and calibration of MAS models. The major focus has been on presenting the connection between a powerful body of theory and its implications for practical model building.

As this is a preliminary study, there are a number of simplifications and approximations we have used. Perhaps the most significant one is treating ‘agent’ variables as continuous valued, and spatially fixed. The latter is actually not as much of an issue as it may seem - CA are capable of generating much, if not all, of the behaviour of ostensibly ‘richer’ models such as artificial ecologies, simply requiring more iterations to get there. The former will be addressed through the use of Monte-Carlo techniques, as mentioned above. Future work will focus on building a more thorough understanding of a methodology for the use of this approach through a case study.

## 6. ACKNOWLEDGEMENTS

This project is supported by an Australian Research Council Linkage grant in collaboration with the Built Environment Research Unit (BERU), Queensland Department of Public Works. Thanks also to Alex Streit for Graphics API expertise.

## 7. REFERENCES

- MJ Casdagli. A dynamical systems approach to modeling input-output systems. In M.J. Casdagli and S.G. Eubank, editors, *Nonlinear Modeling and Forecasting*. Addison-Wesley, Santa-Fe Institute, 1992.
- AM Fraser and HL Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2):1134–1140, 1986.
- N Gershenfeld. *The nature of mathematical modeling*. Cambridge University Press, Cambridge, United Kingdom, 1999.
- S Orstavik and J Stark. Reconstruction and cross-prediction in coupled map lattices using spatio-temporal embedding techniques. *PHYSICS LETTERS A*, 247(1-2):145 – 160, 1998.
- Uygar Ozesmi and William J. Mitsch. A spatial habitat model for the marsh-breeding red-winged blackbird (*agelaius phoeniceus* l.) in coastal lake erie wetlands. *Ecological Modelling*, 101(2-3):139–152, 1997. TY - JOUR.
- NH Packard, JP Crutchfield, JD Farmer, and RS Shaw. Geometry from a time-series. *Physical Review Letters*, 45:712–716, 1980.
- S Shekhar, P Schrater, R Vatsavai, W Wu, and S Chawla. Spatial contextual classification and prediction models for mining geospatial data. *IEEE Transactions on Multimedia*, 4(2):174–188, 2002.
- J Stark. Delay embeddings for forced systems. i. deterministic forcing. *JOURNAL OF NONLINEAR SCIENCE*, 9(3):255 – 332, 1999.
- J Stark, DS Broomhead, ME Davies, and J Huke. Takens embedding theorems for forced and stochastic systems. *NONLINEAR ANALYSIS-THEORY METHODS AND APPLICATIONS*, 30(8):5303 – 5314, 1997.
- J Stark, DS Broomhead, ME Davies, and J Huke. Delay embeddings for forced systems. ii. stochastic forcing. *JOURNAL OF NONLINEAR SCIENCE*, 13(6):519 – 577, 2003.
- F Takens. Detecting strange attractors in turbulence. *Lecture Notes in Mathematics*, 898, 1981.