

# Modelling vertical fish migration using mixed Ornstein-Uhlenbeck processes

Erik Natvig\*, Sam Subbey†

## Abstract

Based on vertical movement data derived from electronic storage tags (DST) attached to fish, we construct a stochastic model that aims at capturing the main characteristics of the observations over one year. We use a mixed Ornstein-Uhlenbeck process to model attraction to a limited number of concentration points on the depth axis. A methodology for determining states and transition probabilities between them, and for setting model parameters for the process, is discussed. We show some examples of simulations using the model, and compare the simulations to the original data.

In general, the model appears to capture the main characteristics of the vertical dynamics, except in cases where the data is characterized by a long-lasting transient state at the start of the time series.

## 1 Introduction

Electronic data storage tags (DSTs) attached to fish have the potential of providing long term, high resolution observation data on individual fish behaviour. DSTs typically record values of depth and ambient temperature (and occasionally, salinity) sampled at prescribed time frequencies (e.g., every 10 minutes). The release-recapture period for an individual fish may range from a few months to a couple of years in some exceptionally few cases.

Thus, DSTs provide a large volume of data and depth of knowledge about individual fish movement. A major drawback, however, has been the lack of adequate methodology to efficiently handle the large amount of data recorded by DSTs. The vertical movement, for instance, may vary significantly in frequency and magnitude, as well as between months, seasons and years. This characteristic of the data poses a challenge to classical data analysis methodologies. Further, whereas information on an individual fish may be attractive, using such information as a basis for inference on fish behaviour at stock (population) level is more desirable. Upscaling of the dynamics of a single fish, extracted from DST analysis, to stock levels, has not been addressed in the literature.

This article presents a stochastic modelling approach that seeks to mimic the vertical movement of fish as seen in an example data set, assuming that the fish makes transitions between several *behavioural states*. The aim is to capture the main features of water column usage for a single fish such as depth-homing, which might encapsulate several responses. The next step in future work is to extend the model to several other data

---

\*Master student, Department of Informatics, University of Bergen, P.O. Box 7803, 5020 Bergen, Norway. E-mail: erik.natvig@student.uib.no. URL: <http://eriknatvig.net/>

†Institute of Marine Research, P.O. Box 1870, Nordnes, 5817 Bergen, Norway. E-mail: samuels@imr.no. URL: <http://uncertainty.imr.no/>

*This paper was presented at the NIK-2011 conference; see <http://www.nik.no/>.*

series for fish tagged and released either in the same period or in the same location in the ecosystem. The ultimate aim is to extend inference on individual fish to population levels, based on the variability in model parameters obtained by analysis of large sets of observations.

Section 2 gives an overview of the methodological approach, assumptions, and presents some related work. Section 3 presents the theory needed for the modelling and simulation procedures, while in Section 4 we suggest a method for determining Markov states for the vertical dynamics, and formulation of the associated transition matrix. Section 5 presents the data and inference results, and shows examples of simulations for a given number of months. Section 6 concludes the paper, considers validity and suggests points for improvement of the model. All methods have been implemented in MATLAB.

## 2 Overview and related work

Within short time intervals, one can assume that the individual fish moves according to a single diffusion (Ornstein-Uhlenbeck) process. However, the depth data encapsulates switches between different behavioural and physiological processes, which include, among others, feeding, swimming, passive tidal transport and spawning. Thus the vertical movement, over a long time period, involves switches between several states representing the varying behavioural/physiological processes, and thus a mixed diffusion process. We will need the following definitions in the rest of the paper:

**Definition 1.** A *concentration point* is a unique point in the space of the data being considered, with a high density of actual data points around it. Concentration points will be denoted by  $\mu_i$ . The space of the data will be partitioned into subsets which contain one concentration point each, and the neighbourhood of a concentration point will be defined as a symmetric region around the point.

**Definition 2.** A (behavioural) *state* is uniquely defined by the combination of the concentration point the fish was attracted to last and the one the fish is currently attracted to, i.e. a state with movement from  $\mu_i$  to  $\mu_j$  is denoted by  $\pi_{ij} = \{i \rightarrow j\}$ .

The idea is to model the transitions between behavioural states as a discrete-time Markov chain  $S_t$ , and model the fish trajectories using a *mixed Ornstein-Uhlenbeck* (OU) process with parameters determined by the state of the Markov chain. The OU process is a mean-reverting process, such that it can model random movement but with attraction towards a fixed point  $\mu$ , and alternating the behavioural states makes it suitable for capturing the data characteristics mentioned above.

Simulations generated using the model we create will be compared to the original data, to see if important characteristics have been captured. This poses four challenges which will be addressed:

- Determining which state a given observation belongs to.
- Forming a transition matrix for a discrete-time Markov chain controlling the behavioural state.
- Simulating the mixed OU process that mimics the behaviour of the fish.
- A metric for comparing simulations to observed data.

## Related work

Bivariate Ornstein-Uhlenbeck models are used for modelling wildlife movement in Dunn and Gipson [2], Preisler et al. [7] and Blackwell [1], though only Blackwell actually used a *mixed* OU process and a set of several behavioural states (corresponding to resting, feeding and travelling for wood mice). The procedure we use in this paper is directly inspired by the work on modelling movements of mobile phone users by Rosenblum [8].

## 3 Modelling and simulation theory

### Markov Chains

A *Markov stochastic process* is special in that the outcome/change in the system after the passage of some time is dependent *only* on the state of the system before the passage of time. This means that a Markov process is without memory – wherever the state has been before the previous point does not matter (see [6]). A Markov *chain* is a stochastic process with a countable set of states. Transition probabilities for finite-state discrete-time Markov chain can be given by a *transition matrix*  $\mathbf{P}$  whose entries are the probabilities for transitions from each state to all others in one time step. A state of a Markov chain is called *transient* if there is zero probability for the chain of returning to it once the state has been left once, and *recurrent* otherwise.

### Ornstein-Uhlenbeck process

A simple model for attraction towards a concentration point  $\mu$  is the Ornstein-Uhlenbeck (OU) process (originally used to describe Brownian motion in physics, see [10] and [4]). It is defined as the stochastic process  $X(t)$  which satisfies the stochastic differential equation (SDE)

$$dX(t) = b(\mu - X(t)) dt + c dW(t) \quad (1)$$

where  $dW(t)$  denotes an infinitesimal increment of a Wiener or white-noise process and  $b$  and  $c$  are parameters. Using the relevant expressions from [4], and substitution for the case when  $\mu \neq 0$ , we get the long-term expectation and variance of the process, with  $X(0) = x_0$  as initial condition:

$$E[X(t)] = \mu + (x_0 - \mu)e^{-bt} \rightarrow \mu \text{ as } t \rightarrow \infty, \quad (2a)$$

$$\text{Var}(X(t)) = \frac{c^2}{2b} \left(1 - e^{-2bt}\right) \rightarrow \frac{c^2}{2b} \text{ as } t \rightarrow \infty. \quad (2b)$$

This clearly shows that the process is attracted towards  $\mu$ . The *drift term parameter*  $b$  determines the intensity of attraction. The second term (diffusion term) provides the randomness in the process. A first-order approximation for use in simulating this SDE is the updating formula from [4]:

$$X(t + \Delta t) \approx X(t) + b(\mu - X(t)) \Delta t + cn\sqrt{\Delta t}, \quad (3)$$

where  $n$  is a sample value of a standard normally distributed variable  $N$  ( $N \sim \mathcal{N}(0, 1)$ ).

### Mixed Ornstein-Uhlenbeck process

To model the attraction of the fish towards concentration points, we follow the idea in [8], and use a *mixed Ornstein-Uhlenbeck* (OU) process. Let  $X_t$  be the stochastic process that is a solution of the stochastic differential equation (1) above that governs the movement of the fish in the model, with a time index  $t$  running from 1 to  $T$ . Next, assume that there are  $m$  different concentration points on the depth axis, with labels  $\mu_1, \dots, \mu_m$ , that the fish may be attracted to.

We further assume that a discrete-time Markov chain  $S_t$  can model which concentration point the fish is attracted to at any time step. The state-space of  $S_t$  is denoted  $\Pi$ , and for each ordered pair of concentration points  $\mu_i$  and  $\mu_j$  we define a state of the Markov chain as  $\pi_k = \{i \rightarrow j\}$ , representing movement from  $\mu_i$  to  $\mu_j$ . The reason for including one state for each *pair* of concentration points, instead of just one for each, is that we might want to model different variabilities and intensities of attraction towards the concentration point depending on where the fish came from. For instance, diving down in the sea might be quicker than moving towards the surface or vice versa, so the model should be able to capture that by varying the drift term coefficient  $b$  between states (although in practice this possibility will not be explored in this paper).

Now, for each state  $\pi_k \in \Pi$ , we define a set of parameters for the OU process:  $\{\mu^{(k)}, b^{(k)}, c^{(k)}\}$ . In the model we let  $X_t$  follow a OU process with parameters given by the state of the Markov chain  $S_t$ . When the chain transitions to a new state, the new parameters are used instead of the old, so that we get a new OU process starting in the end point of the previous process.

The parameters for the OU process for each state of the Markov chain, along with the selection of concentration points, must be estimated using the data collected from actual fish. In the following sections we suggest a method for this.

## 4 Determining states

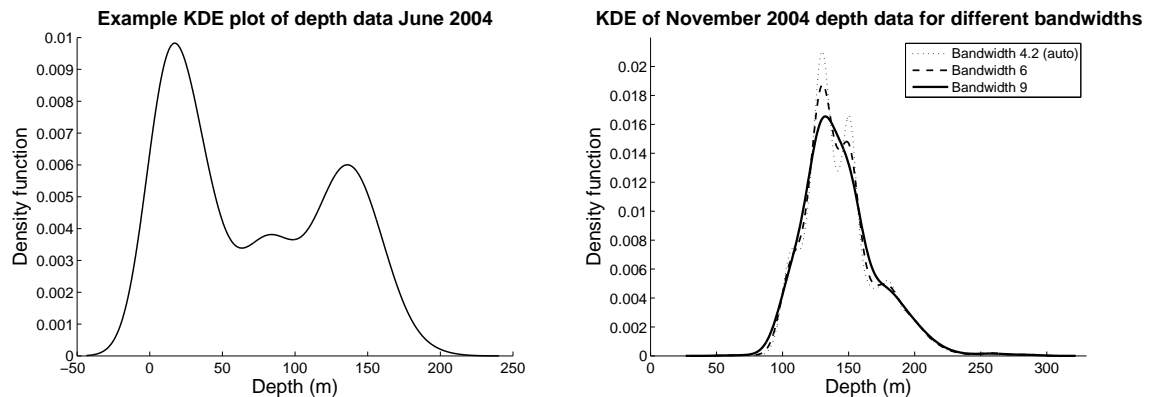
### Kernel density estimation

Kernel density estimation (KDE) is a non-parametric method used to estimate an unknown probability density function from  $n$  data points assumed to be realizations of  $n$  independent and identically distributed random variables. A kernel is a symmetric function that integrates to one (usually taken to be the standard normal density function). By placing a kernel at each data point and summing them up, dividing by the number of data points, we get a new function that integrates to one and has peaks at regions with high density of data points. Formally, the estimated function  $\hat{f}_h$  of the data  $d$  is

$$\hat{f}_h(d) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{d-d_i}{h}\right), \quad (4)$$

where  $h$  is a smoothing parameter known as the bandwidth,  $K()$  is the kernel and  $d_i$  are the observed data points. The bandwidth that will be used by default is one optimal for estimating normal densities, but we will see that it might need to be adjusted to reveal the main features of interest.

**Figure 1:** Kernel density estimate examples



A kernel density estimate for an example time series is shown in Figure 1 on the left. Observe that the estimate has three peaks that can be considered as concentration points. The next step involves defining neighbourhoods around such points.

## Persistence analysis

Common methods for discretizing continuous values in a dataset into discrete symbolic values, such as the histogram, do not take the time aspect into account. They may also place cuts in regions of the data with high density. The temporal structure, i.e. the order in which observations occur in a time series, is valuable information that should be used. The Persist Algorithm [5] uses this information to find cuts such that the resulting symbolic values, or states, are persisting. This means that for a point in time there is high probability to observe the same symbol as in the previous point in time, when using the discretization given by the algorithm. The bins given by the algorithm often coincide with peaks in the KDE plot, such that each peak has its own bin.

## Combining persistence analysis and kernel density estimation

### *Choosing concentration points*

We wish to determine, for a slice  $\tilde{X}_t, t = 1, \dots, T$  of the time series (we will be using whole months), which concentration points to use in the model. The following method is used:

1. Run the Persist algorithm in order to find bin boundaries for the data.
2. Make KDE plot and find local maxima (peaks in the graph).
3. Adjust KDE bandwidth if needed:
  - (a) Some of the local maxima may lie too close to each other (rule of thumb: less than 10 meters apart), or have too low value on the y-axis compared to other local maxima, so we consider smoothing the plot by increasing the bandwidth manually until the KDE has desirable properties. This can be improved in future work by adjusting the bandwidth more systematically. See Figure 1 on the right for an example of bandwidth adjustment.
  - (b) One should also check that the bin boundaries from Persist are such that each concentration point has its own bin – e.g. if there are two peaks in one bin, the bandwidth should probably be increased so that the two peaks join into one peak.
4. Discard peaks that are less than some percentage in height compared to the highest peak. This paper uses a cut-off of 10%. However, this value is arbitrary. Future work will seek to develop a more heuristic approach to determining a reasonable cut-off value.
5. Store the  $m$  remaining peaks as concentration points  $\mu_1, \dots, \mu_m$  and define a possible state of the Markov chain for each ordered pair of points to define the set of states  $\Pi$ .

### *Neighbourhoods and states*

Having defined the state  $\pi_k = \{i \rightarrow j\}$ , we define the *neighbourhood* around  $\mu_j$  as the interval  $(\mu_j - n\sigma_j, \mu_j + n\sigma_j)$ , where  $\sigma_j$  is the standard deviation of the data in the same bin as the concentration point, and  $n = 1$  (for the moment). Define a number  $k$  for each state in  $\Pi$ , then a mapping  $\Psi(i, j)$  from the numbering of the concentration points to the numbering of the states, such that if state  $\{i \rightarrow j\}$  corresponds to state  $\pi_k$ , then  $\Psi(i, j) = k$ .

We define two vectors  $\boldsymbol{\phi}$  and  $\boldsymbol{\tau}$ , which store origin and destination concentration points, respectively, for each data point. We use the following algorithm:

```

1:  $\boldsymbol{\phi} \leftarrow \mathbf{0}$  {zero vector of length  $T$ }
2:  $\boldsymbol{\tau} \leftarrow \mathbf{0}$ 
3: for all  $\tilde{X}_t$  do
4:   if  $\mu_i - n\sigma_i \leq \tilde{X}_t \leq \mu_i + n\sigma_i$  for some concentration point  $\mu_i$  then
5:      $\tau_t \leftarrow i$ 
6:   end if
7: end for
8: if  $\tau_T = 0$  then {take special care of the end of time series}
9:   find the last time-point  $t_{last}$  such that  $\tau_{t_{last}} \neq 0$ 
10:   $\tau_t \leftarrow \tau_{t_{last}}$  for  $t = t_{last} + 1, \dots, T$ 
11: end if
12: for all  $1 \leq t \leq T-1$  do
13:   if  $\tau_{T-t} = 0$  then {attraction to next point visited}
14:      $\tau_{T-t} \leftarrow \tau_{T-t+1}$ 
15:   end if
16: end for
17:  $\phi_1 \leftarrow \tau_1$ 
18: for  $2 \leq t \leq T$  do
19:   if  $\tau_t \neq \tau_{t-1}$  then {attraction to new point, i.e. state change}
20:      $\phi_t \leftarrow \tau_{t-1}$ 
21:   else {same attraction as before}
22:      $\phi_t \leftarrow \phi_{t-1}$ 
23:   end if
24: end for
25:  $\tilde{S}_t \leftarrow \Psi(\phi_t, \tau_t)$  for  $t = 1, \dots, T$ 

```

Even though the process assigns to each point  $\tilde{X}_t$  a state  $\tilde{S}_t$ , there will be unencountered states. We discard unencountered states and renumber those that remain, so that we get a new set of states  $\Pi$ . Note that the first data point, and those that follow until a new attraction is observed, are assigned a special state  $\pi_{\text{initial}} = \{i \rightarrow i\}$  with the same destination and origin concentration point  $\mu_i$ . This state is transient by construction, since once it has been left there will always be a previously visited concentration point, different from the one currently being sought.

## Counting transitions and forming the transition matrix

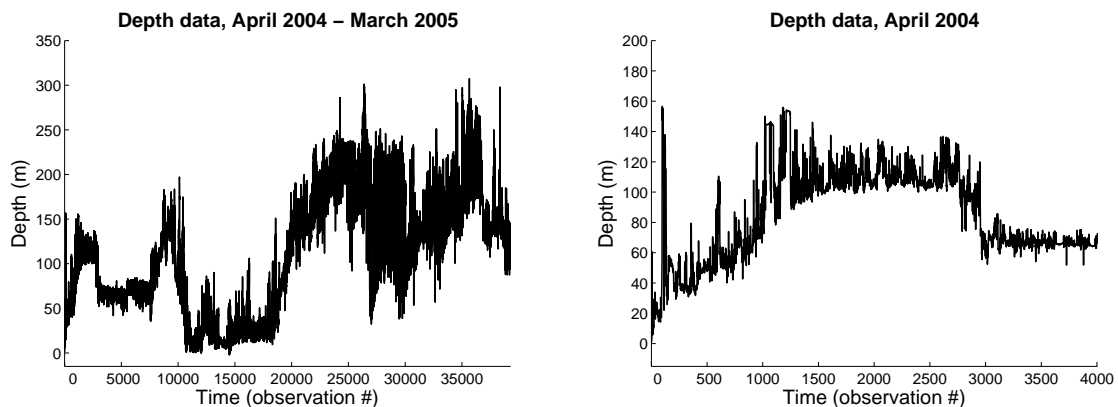
A consistent estimator for the elements of the transition matrix  $\mathbf{P}$  for a finite-state discrete-time Markov chain  $S_t$  is, according to [6],

$$\hat{p}_{ij} = \left( \frac{\sum_{n=0}^{N-1} \mathbf{1}_{\{S_n=i, S_{n+1}=j\}}}{\sum_{n=0}^{N-1} \mathbf{1}_{\{S_n=i\}}} \right), \quad (5)$$

which is the proportion of all transitions from state  $i$  that go to state  $j$ . Here,  $\mathbf{1}_{\{\cdot\}}$  is the indicator function which is equal to 1 if the subscripted expression in braces is true, and 0 if it is false. As  $N$  tends to infinity,  $\hat{p}_{ij}$  will tend to  $p_{ij}$  with probability 1, by the strong law of large numbers.

We now consider  $\tilde{S}_t$ , our time series of states, as a sample path of a discrete-time Markov chain  $S_t$ . We wish to estimate the  $|\Pi| \times |\Pi|$  transition matrix  $\mathbf{P}$  for this chain. In order to do this, we summarize the chain by a matrix  $\mathbf{M}$  with entries  $m_{kl}$  counting transitions from  $\pi_k$  to  $\pi_l$ . Next, we form a matrix  $\hat{\mathbf{P}}$  which has the normalized rows of  $\mathbf{M}$ , so that each entry  $\hat{p}_{kl}$  of  $\hat{\mathbf{P}}$  has the proportion of all transitions from  $\pi_k$  that go to  $\pi_l$ , according to the estimator (5) above.

**Figure 2:** Example time series, the whole year April 2004 - March 2005 (left) and April 2004 (right)



## 5 Data and problem description

In order to test the methodology introduced in the previous section, we use the first year of the depth data for a fish (cod) that was tagged and released in April 2004 and recaptured in May 2006. The data is sampled at 10 minute intervals. This sampling frequency is sufficient, and actually far too high, considering the ambient environment. The dominant tidal frequencies are clustered around 6 and 24 hour periods. It has been established in the literature that fish may use tidal transport for horizontal/vertical migration (see [11] and [3]). Given this, it makes sense to expect that the vertical dynamics would encapsulate signals with periodicity 6 and 12 hours. Further, it is natural to expect diurnal periodic behaviour due to the influence of sunlight (see [9]).

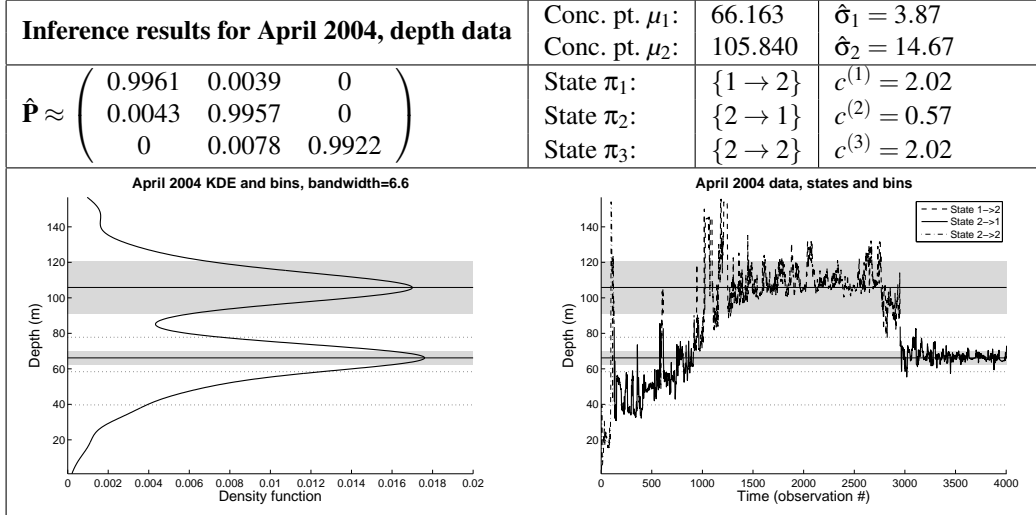
In Figure 2 we show on the left the whole time series, while on the right we show the data for April 2004. A visual inspection shows that there are basically three or four regions on the depth axis with a high concentration of data points. In this section we will determine concentration points and transition matrices for this data, present simulations using these and compare the simulations to the original data.

### Example of inference results

Running the procedure for determining states and transition matrices on this data gives us the results shown in Table 1. The left-hand figure shows the (flipped) kernel density estimate with the concentration points as lines, and the neighbourhoods as gray backgrounds. Notice that the used bandwidth is indicated in the title of the figure. The right-hand figure also shows the concentration points and neighbourhoods, but also a line for the data with different line styles indicating which state the points are considered as being in. (The data in the plot, and in all the similar plots following, is slightly smoothed using moving averages for better visual quality.)

### Simulation details

The modelling process requires both the simulation of the Markov chain and OU process at the same time. Thus, the Markov chain  $S_t$  is considered first. It is simulated using numbers  $u$  from a pseudo-random number generator on  $[0, 1]$ , where the interval is partitioned into a set of disjoint subintervals for each state, as described in [6]. The OU parameters belonging to the state at each time step are used in the updating formula (3) for  $X(t)$ .



**Table 1:** Inference results for April 2004, depth data

### Parameters for the Ornstein-Uhlenbeck process

For each state of the Markov chain  $S_t$  we determine the parameters  $b$  (drift term coefficient) and  $c$  (diffusion term coefficient) for the Ornstein-Uhlenbeck process.  $b$  is in this paper chosen empirically and equal for all states. This is done by visually inspecting the plots of the simulated paths and adjusting  $b$  so that they resemble the paths in the data ( $b = 0.05$  is used throughout). A systematic way of doing this is suggested in [8], but this has not been implemented in this paper due to time constraints. The analysis shows that the results have low sensitivity to the accurate determination of  $b$ , especially in terms of picking the main dynamics of the migratory patterns. From the expression (2b), using  $\sigma^2$  to denote the long-term variance of the OU process, we get the relation  $c = \sigma\sqrt{2b}$ . We estimate the long-term variance of the process for a state  $\pi_k = \{i \rightarrow j\}$  as the squared standard deviation  $\hat{\sigma}_j^2$  of all data points that are within the neighbourhood of concentration point  $\mu_j$ , so we set  $c^{(k)} = \hat{\sigma}_j\sqrt{2b^{(k)}}$ . Thus, all states with the same destination concentration points have the same OU-parameters. Nevertheless, keeping a state for each combination of concentration points in the model allows for further experiments in future work by larger variations of  $b$  and  $c$  between states.

### Simulation time step

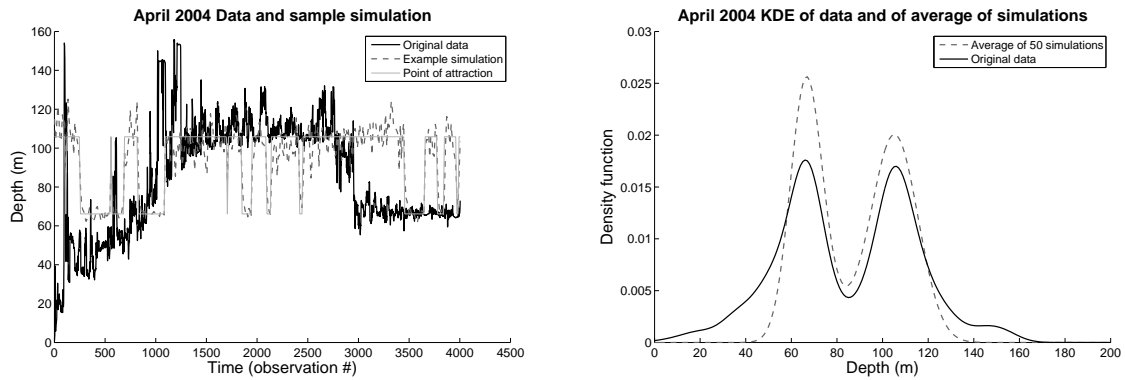
The time step used in the simulation should not be the same as in the data, because the random effects in the fish behaviour occur over a larger time-scale than just one time step of 10 minutes. For instance, one may observe a steady increase in depth over an hour before the depth decreases again, all while the fish is in the same state. Had we used a time step of only 10 minutes in the simulation, we would get large oscillations in the values for very short time intervals, since the random part of the model is very dominant once the process is close to a concentration point. Thus, we choose for now to simulate using a 2 hour time step instead, corresponding to 12 time steps in the data. Since the transition matrix we have estimated is for time steps of 10 minutes, we need to use the 12-step transition probabilities for  $S_t$ , which are the entries of the matrix  $\hat{\mathbf{P}}^{12}$ . So in practice, the updating formula above is used with  $\Delta t = 12$  and with OU parameters that depend on the state of the Markov chain in the new time-point. We define

$$X_{t+1} = X_t + b^{(k)} \left( \mu^{(k)} - X_t \right) \Delta t + c^{(k)} n \sqrt{\Delta t}, \quad (6)$$

with  $k$  given by the Markov chain in the sense that  $S_{t+1} = \pi_k$ .



**Figure 3:** Example simulation for April 2004 and KDE plot



### *Markov chain and OU process initial states, and burn-in*

We choose as initial state  $S_1$  the special, transient state  $\pi_{\text{initial}} = \{i \rightarrow i\}$  for the first concentration point  $\mu_i$  visited. Also, we let the OU process start in this concentration point by letting  $X_1 = \mu_i$ . The initial state chosen for a Markov chain will slightly affect the mean time spent in each state over the whole simulation period, but the longer the chain is allowed to evolve, the less this “noise” from the start of the process affects it. To get results that are not affected too much by the initial state, we let the process evolve for 200 time steps before recording the data. This is known as “burn-in”.

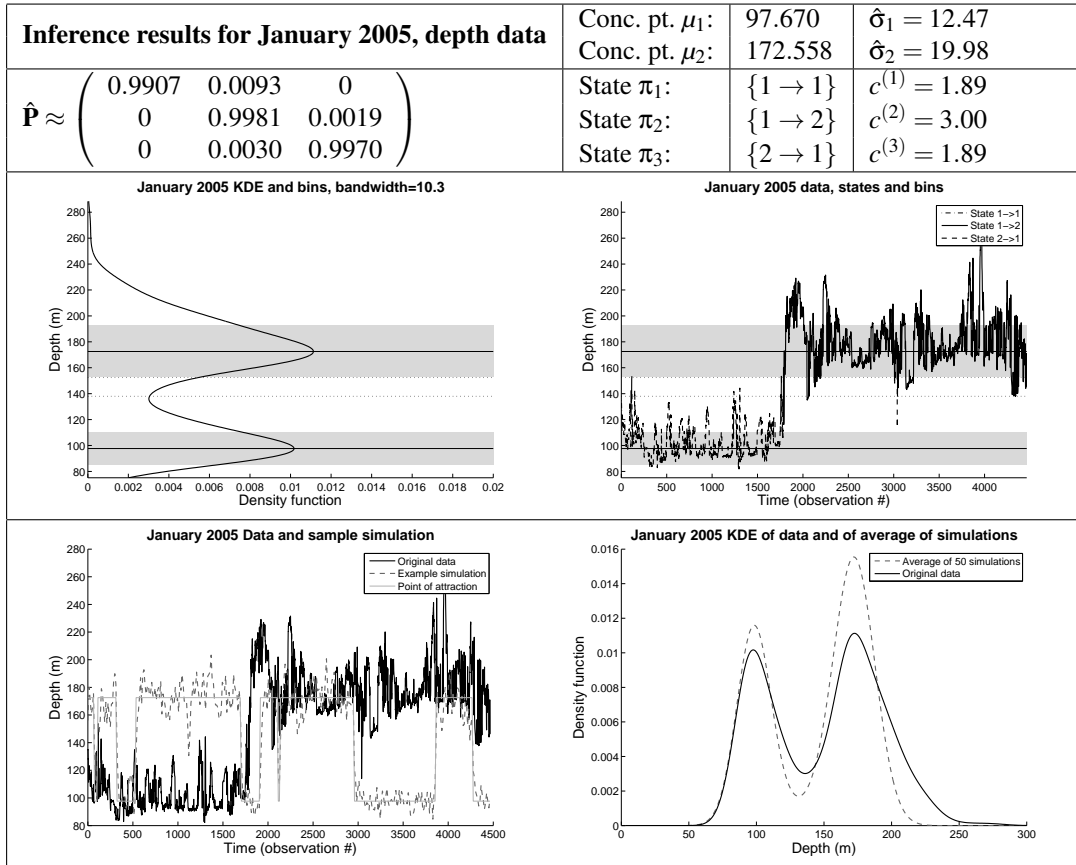
### **Simulation examples and comparing to data**

In Figure 3 we show on the left an example of a simulation using the states, the transition matrix and the OU parameters inferred from the April 2004 depth data shown in Table 1. Note the solid gray line indicating which concentration point the process is attracted towards. The number of data points in the simulation are 1/12 of those in the data, but we have plotted the simulation such that the two paths are aligned in time, the units of the x-axis corresponding to the original numbering of the observation data points.

We see that some features of the original data for April 2004 are recreated in the simulation, in the sense that the process spends time around the same concentration points, and with more variability around the concentration point  $\mu_2$  at 106 meters than  $\mu_1$  at 66 meters. Comparing kernel density estimates for the data and the simulation (making sure to use the same bandwidth in order to keep the plots comparable) confirms this. But since this is a random process, both due to the randomness in the Markov chain and due to the random term in the updating formula, the simulated path will be different every time. To get an idea of the “mean behaviour” of the model, we take the average value of the KDEs for 50 different simulations and compare that with the KDE for the data.

The resulting mean KDE plot for the April 2004 simulations is shown with a dashed line on the right in Figure 3 together with a solid plot of the KDE of the data. We see that the time spent around  $\mu_1$  is overestimated by the model. But this is not surprising – there is a lot of area under the solid curve to the left that corresponds to data points whose characteristics we have not made any attempt to capture in the model. The peak around  $\mu_1$  at 66 meters could have been a little wider to capture more of the original data spread, indicating that the diffusion term coefficient  $c$  used in the state that generated those data points was too small. This indicates that the method for choosing  $c$  should be improved.

Table 2 shows complete inference and simulation results for a different month in the example time series: January 2005. Here we also see that the distribution of the data in the simulation matches quite well to the distribution of the original data.



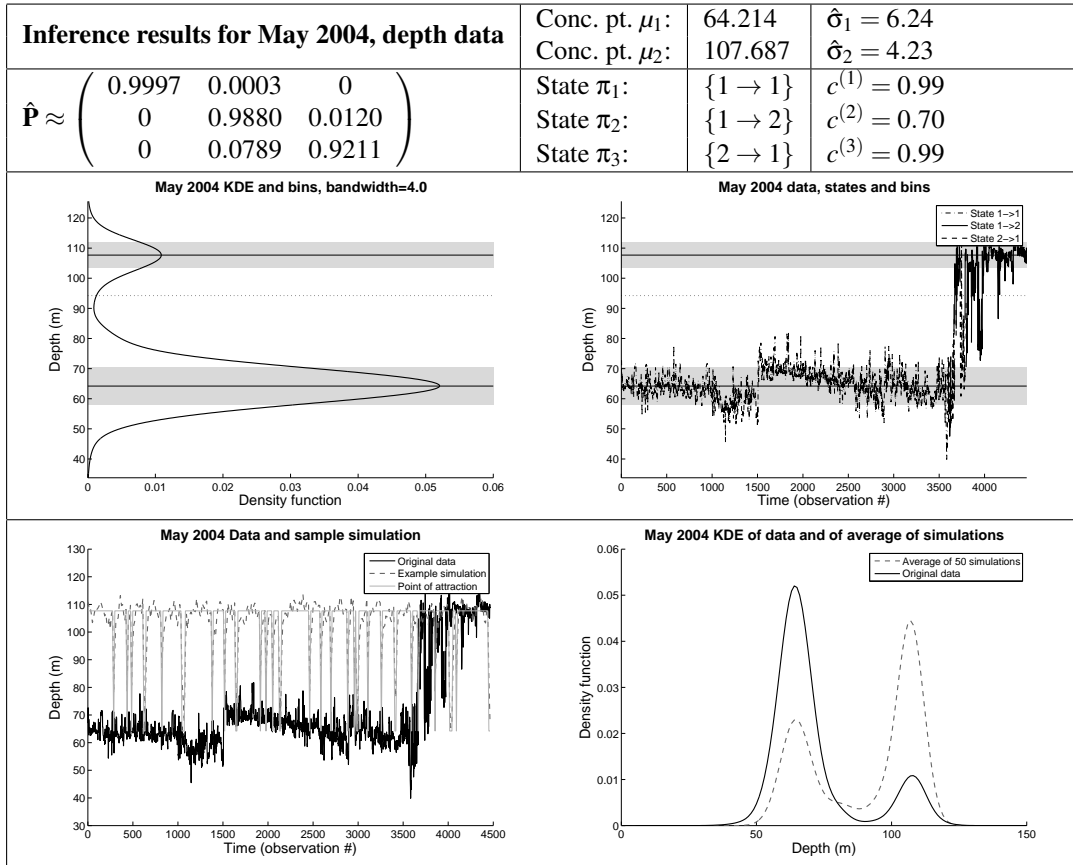
**Table 2:** Inference results and simulations for January 2005, depth data

### Simulations matching poorly

We see that the plot of the average of the KDEs matches pretty poorly with the data KDE in the results for May 2004 in Table 3. The two peaks from the simulations have almost opposite heights, which is very different from the situation in the data. Not all such average KDEs of simulation series using the parameters of this data have shown as poor results as this, but none of them show good model fit. Poor model fit may occur whenever the fish has stayed close to one concentration point  $\mu_i$  for a long time at the start of the month. The problem in such cases is that the initial state  $\pi_{\text{initial}} = \{i \rightarrow i\}$  accounts for a large portion of the time spent around  $\mu_i$ . Randomness might bring the Markov chain away from this state earlier than observed in the data, (possibly during burn-in) and since it is transient, the chain will never revisit the state. Then, the proportion of time spent around each concentration point for the rest of the simulated path will depend on the corresponding proportion in the data *after*  $\pi_{\text{initial}}$  was left, which might not correspond to the total proportion of time spent around each point in the whole month. This problem needs to be solved in order to make the model usable for months where this occurs. A possible solution is to avoid using whole months as cut-points for the data.

## 6 Conclusion

The model does quite a good job of capturing the main aspects of the vertical dynamics for time series where the initial transient state is left early in the data. This is a good starting point for modelling vertical migration of the fish. However, for months such as May 2004, starting with a long-lasting transient state, the model fails to capture the main characteristics.



**Table 3:** Inference results and simulations for May 2004, depth data

## Possible improvements to the model

Obtaining better model fit to data requires a method for estimating the OU process parameters and a systematic approach for varying the parameters between states. More systematic adjustment of KDE bandwidth must be considered. One might argue that a transition between states should only be recorded if the state change lasts for some time. A short excursion of 4-5 data points towards a different concentration point should perhaps not be counted as a transition, since it often will affect the transition matrix and thus the model dramatically. It is also necessary to find a way to avoid the problem with long-lasting transient states. Finally, attempts would be made to use second-order or exact updating formulas for  $X_t$  in the simulations.

## Validity and reliability

The procedures in this paper have not been tested with other data than those mentioned, and conditions for them to work well for other data will now be considered. The model depends on data having a mean-reverting tendency. The data should be characterized by more than one concentration point. Otherwise the model will simplify to a single-state OU process starting at the mean, making it a white-noise process. No attempts have been made to enforce biological plausibility in the model, for instance “speed limits” or behavioural restrictions. The time aspect (i.e. the time spent between transitions) of the model versus original data has not been considered.

## Extensions

Plans for further work include:

- Improve model fit by predefining a transition matrix and OU parameters, and trying to recover them using the inference methods described here on simulations produced using these settings.
- Testing the model also on temperature data and extending it to two dimensions: depth and temperature, where concentration points are defined in the depth/temperature plane.
- Using the model as a basis for species discrimination (North-East Arctic cod versus Norwegian coastal cod).
- Attempting geolocation of the fish by combining the model with a depth and temperature atlas.
- Using the model as a basis for making inferences on behaviour on the population level.

## References

- [1] P.G. Blackwell. Random diffusion models for animal movement. *Ecological Modelling*, 100(1-3):87 – 102, 1997.
- [2] J.E. Dunn and P.S. Gipson. Analysis of Radio Telemetry Data in Studies of Home Range. *Biometrics*, 33(1):85–101, 1977.
- [3] R.N. Gibson. Go with the flow: tidal migration in marine animals. *Hydrobiologia*, 503(1-3):153–161, 2003.
- [4] Daniel T. Gillespie. Exact numerical simulation of the Ornstein-Uhlenbeck process and its integral. *Physical Review E*, 54(2):2084–2091, 1996.
- [5] Fabian Mörchen and Alfred Ultsch. Optimizing time series discretization for knowledge discovery. In *Proceedings The Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 660–665, 2005, Chicago, IL, USA.
- [6] J.R. Norris. *Markov Chains*. Cambridge Series on Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.
- [7] Haiganoush K. Preisler, Alan A. Ager, Bruce K. Johnson, and John G. Kie. Modeling animal movements using stochastic differential equations. *Environmetrics*, (15):643–657, 2004.
- [8] Michael Rosenblum. Mobility modeling with a mixed Ornstein-Uhlenbeck process. Found on website <http://people.csail.mit.edu/mrosenblum/work/>.
- [9] Sam Subbey, Kathrine Michalsen, and Geir Nilsen. A tool for analyzing information from data storage tags: the continuous wavelet transform (CWT). *Reviews in Fish Biology and Fisheries*, 18:301–312, 2008. 10.1007/s11160-007-9078-2.
- [10] G.E. Uhlenbeck and L.S. Ornstein. On the theory of the brownian motion. *Physical Review*, 36(5):0823–0841, Sep 1930.
- [11] D. Weihs. Tidal stream transport as an efficient method for migration. *Journal du Conseil International pour l'Exploration de la Mer*, (38):92–99, 1978.