Supplementary material for "Varying-coefficient stochastic differential equations with applications in ecology"

Théo Michelot, Richard Glennie, Len Thomas, Catriona Harris

March 4, 2021

Appendix A Special cases with applications in ecology and finance

The specification of the functions μ and σ in the SDE determines the type of diffusion process (Equation 2 of the paper). In practice, this choice depends on the characteristics of the observed process. In this section, we propose several choices of μ and σ, corresponding to varying-coefficient variants of standard diffusion processes, and possible applications to the analysis of animal movement and other ecological data. In all the examples that we consider here, the drift function μ and the diffusion function σ each depend on only one time-varying parameter, i.e. $\mu(Z_t, \boldsymbol{\theta}_t) = \mu(Z_t, \theta_t^{(1)})$ and $\sigma(Z_t, \boldsymbol{\theta}_t) = \sigma(Z_t, \theta_t^{(2)})$, where $\boldsymbol{\theta}_t = (\theta_t^{(1)}, \theta_t^{(2)})$. We denote $r_t = \theta_t^{(1)}$ and $s_t = \theta_t^{(2)}$ for simplicity.

Brownian motion with drift The simplest model is the Brownian motion (with drift), where $\mu(Z_t, \boldsymbol{\theta}_t) = r_t$ and $\sigma(Z_t, \boldsymbol{\theta}_t) = s_t$. Here, r_t and s_t are time-varying drift and diffusion parameters for the process, respectively. Based on the Euler-Maruyama discretization, the approximate transition density of this process is

$$[Z_{t+\Delta} = z_{t+\Delta} | Z_t = z_t] = \phi(z_{t+\Delta};\ z_t + r_t\Delta,\ s_t^2\Delta),$$

where $\phi(x;\ m,\ v)$ is the pdf of the normal distribution with mean m and variance v.

Geometric Brownian motion The process Z_t is called geometric Brownian motion if $\log(Z_t)$ follows a Brownian motion with drift. The varying-coefficient geometric Brownian motion is a diffusion process with $\mu(Z_t, \boldsymbol{\theta}_t) = r_t Z_t$ and $\sigma(Z_t, \boldsymbol{\theta}_t) = s_t Z_t$. Geometric Brownian motion is a popular choice to model population growth in ecology, where Z_t is the population size at time t, and r_t is the growth rate (Dennis et al., 1991), which could be modelled as a function of covariates in the framework that we present.

This process is also used in finance to describe asset prices under the Black-Scholes model. In addition, stochastic volatility models are typically based on geometric Brownian motion, where the variance parameter s_t^2 is modelled with an SDE (Aït-Sahalia and Kimmel, 2007). The varying-coefficient model introduced here is an alternative approach to specify a process with time-varying volatility.

The standard geometric Brownian motion has a closed form transition density, and so we can obtain the transition density of the varying-coefficient process by substituting r_t and s_t for the SDE parameters,

$$[Z_{t+\Delta} = z_{t+\Delta} | Z_t = z_t] = \frac{1}{\sqrt{2\pi\Delta}} \frac{1}{z_{t+\Delta} s_t} \exp\left[-\frac{\left\{\log(z_{t+\Delta}) - \log(z_t) - (r_t - 0.5 s_t^2)\Delta\right\}^2}{2 s_t^2 \Delta} \right].$$

Ornstein-Uhlenbeck (OU) The varying-coefficient OU process is obtained with $\mu(Z_t, \boldsymbol{\theta}_t) = r_t(\zeta - Z_t)$ and $\sigma(Z_t, \boldsymbol{\theta}_t) = s_t$. It is mean-reverting, i.e. it tends to revert to its mean value ζ with a rate measured by r_t and diffusion measured by s_t. The standard OU process was proposed by Uhlenbeck and Ornstein (1930) to describe the velocity of a particle subject to friction, and it has also been proposed as a model for interest rates in finance (Vasicek, 1977). Figure S1 shows a realisation from the varying-coefficient OU process, to illustrate how time-varying parameters can induce time-varying dynamics. The approximate transition density of the varying-coefficient OU process is

$$[Z_{t+\Delta} = z_{t+\Delta} | Z_t = z_t] = \phi \left\{ z_{t+\Delta};\ \zeta + e^{-r_t\Delta}(z_t - \zeta),\ \frac{s_t^2}{2r_t}(1 - e^{-2r_t\Delta}) \right\}.$$

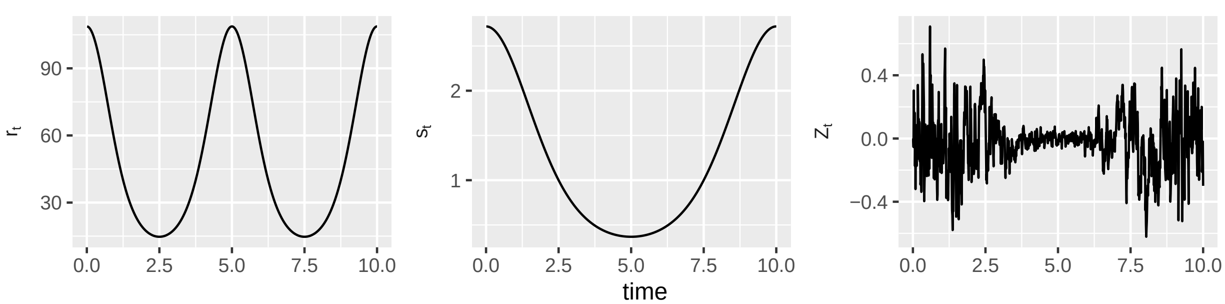


Figure S1: *Simulated example for the time-varying Ornstein-Uhlenbeck process with mean $\zeta = 0$, showing the reversion parameter r_t (left), the diffusion parameter s_t (middle) and the simulated process Z_t (right).*

Dunn and Gipson (1977) suggested the bivariate OU process as a model of movement for animals displaying home range behaviour: in two dimensions, mean reversion is analogous to attraction to a point in space, such as the centre of an animal's home range. The long-term distribution of the OU process is Gaussian, and can be derived in closed form. In that context, r_t and s_t can be linked to the size of the home range, and the strength of attraction to a central location. In practice, V_t and Z_t are usually two-dimensional (representing Easting and Northing), but it is most common to assume that the process is isotropic, i.e. the same process describes the movement in both dimensions. Blackwell (1997) argued that the OU process may often be too simplistic to model the movement of animals, and proposed a mixture model, where an animal switches between discrete behaviours through time, each associated with an OU process. In the framework presented here, r_t and s_t are specified as flexible functions of covariates, as described in Section 2 of the manuscript, and the dynamics of the OU process can therefore change smoothly over time and space. The number of time-varying parameters in the model is not limited to two (r_t and s_t), and the centre of attraction ζ could in principle also be formulated as a function of covariates.

The OU process is also a popular model for the *velocity* of a moving animal, as proposed by Johnson et al. (2008). In the varying-coefficient variant of this model, the velocity V_t is specified as $dV_t = r_t(\zeta - V_t)dt + s_t dW_t$, and the location Z_t of the animal is obtained as the integral $Z_t = Z_0 + \int_0^t V_s ds$, where Z_0 is the initial location. Here, the two parameters r_t and s_t can be linked to the speed and persistence of the movement, and ζ is usually set to zero to indicate that there is no systematic bias in the velocity. Several approaches have been developed to allow for time-varying dynamics in the velocity OU model. In particular, Michelot and Blackwell (2019) presented a state-switching formulation, where an animal transitions between different velocity OU processes through time, characterised by different movement characteristics. Russell et al. (2018) used a time-varying term for the centre of attraction ζ, to include covariate effects on the direction of movement of the animal. The model presented here generalises that approach to the case where any parameter of the velocity process can be written as a GAM of spatiotemporal (or other) covariates.

Potential-based models In ecology, diffusion processes have also been used to study the response of animals to their environment. For this purpose, the drift of the process can be specified as a function of the gradient of a "potential" function H, which measures habitat suitability over space (Preisler et al., 2004). In that model, we have $\mu(Z_t, \boldsymbol{\theta}_t) = -\nabla H(Z_t, \boldsymbol{\theta}_t)$, where ∇ is the spatial gradient operator. Preisler et al. (2004) estimated H with smoothing splines of spatial covariates, using methodology similar to that presented in this paper. Within the framework that we propose, their model could be extended to include non-spatial covariates, and to investigate covariate effects on the diffusion parameter of the process (which they assumed constant).

Appendix B Implementation

B.1 Model matrices using mgcv

The mgcv R package can be used to define the design matrices (including basis functions) and the penalty matrix for basis-penalty smooths, with the function `gam` (Wood, 2017). As an example, consider the data frame shown below, where 'ID' is the time series identifier, 'Z' is the response variable, and 'x1' and 'x2' are two covariates.

```
##   ID           Z       x1         x2
## 1  1 -1.06520105 1.139137 -0.3088276
## 2  1 -1.36064381 2.159093 -0.1593682
## 3  1 -1.04660349 2.610276 -0.6394209
## 4  1  0.35250424 1.830601 -0.7986991
## 5  1 -0.07465397 1.905763  0.4524077
## 6  1 -0.91468379 2.124810  0.3858297
```

Then, consider that `Z` is to be modelled with a varying-coefficient SDE, where the relationship of each SDE parameter with the covariates is specified by the following formula,

```
# Formula for SDE parameter, using mgcv syntax for smooth
# terms and random effects
form <- ~ x1 + s(x2, k = 5, bs = "ts") + s(ID, bs = "re")
```

i.e. `x1` has a linear effect, `x2` has a smooth effect modelled using thin-plate regression splines, and a random normal intercept is included for `ID`. (See the documentation of mgcv for additional detail on the syntax.) The design matrices can then be derived as follows,

```
# Create smooth object using mgcv
smooth <- gam(formula = update(form, dummy ~ .),
              data = cbind(dummy = 1, data),
              fit = FALSE)

# Design matrix
X <- smooth$X

# Number of non-smooth model terms (i.e. fixed effects)
nsdf <- smooth$nsdf

# Design matrix for fixed effects
X_fe <- X[, 1:nsdf, drop = FALSE]
```

```r
# Design matrix for random effects (including smooth model terms)
X_re <- X[, -(1:nsdf), drop = FALSE]
```

The design matrix for the fixed effects is

```r
head(X_fe)
```

```
##   (Intercept)       x1
## 1           1 1.139137
## 2           1 2.159093
## 3           1 2.610276
## 4           1 1.830601
## 5           1 1.905763
## 6           1 2.124810
```

and the design matrix for the random effects is

```r
head(X_re)
```

```
##                                                ID1 ID2 ID3 ID4
## 1 -0.2064242 0.6333093 0.4155266 -0.12492927   1   0   0   0
## 2 -0.1186081 0.6690554 0.4211488 -0.06103063   1   0   0   0
## 3 -0.3930194 0.5225471 0.3865907 -0.26626842   1   0   0   0
## 4 -0.4775210 0.4553437 0.3648461 -0.33436490   1   0   0   0
## 5  0.2464654 0.7122938 0.3947111  0.20052304   1   0   0   0
## 6  0.2071078 0.7157652 0.4013989  0.17205880   1   0   0   0
```

where the first four columns correspond to the four basis functions for `x2`, and the four last columns are dummy indicator variables for `ID`. Then, the linear predictor for the SDE parameter is

```r
lp <- X_fe %*% coeff_fe + X_re %*% coeff_re
```

where `coeff_fe` and `coeff_re` are the coefficients for the fixed effects and the random effects, respectively. The SDE parameter (at each time point) is then obtained by applying the inverse link function to this linear predictor.

Similarly, the penalty matrix for the smooth terms can be extracted from the GAM object,

```
S <- smooth$S
```

B.2 Laplace approximation using TMB

The model fitting procedure proposed in the paper requires the evaluation of the marginal likelihood, where the random effects (including basis coefficients for smooth terms) have been integrated out. We suggest using the R package Template Model Builder (TMB) to implement the marginal likelihood, based on the Laplace approximation Kristensen et al. (2016). Here, we broadly explain how TMB can be used to evaluate the objective function (i.e., the negative log-likelihood) of the model, and to obtain point and uncertainty estimates for the model parameters.

The joint log-likelihood must first be written in C++, following the TMB syntax (for examples, see `kaskr.github.io/adcomp/examples.html`). For our model, it has two main components:

1. the joint log-likelihood of the fixed and random effect parameters, given by the sum of the log-pdf of the observed transitions, $\log[Z_{i+1}|Z_i]$, e.g. obtained using the Euler-Maruyama discretization of the process. This part requires the SDE parameters on a time grid, which can be computed based on the model matrices provided by mgcv, as described in Appendix B.1.

2. the log-pdf of the basis coefficients (and other random effects) given the smoothness parameter, $\log[\boldsymbol{\beta}|\boldsymbol{\lambda}]$, obtained as the log of a multivariate normal pdf with block-diagonal precision matrix, where the i-th block is $\lambda_i \boldsymbol{S}_i$. The smoothness matrix $\boldsymbol{S}_i$ is provided by mgcv, as described in Appendix B.1.

The C++ function takes the following arguments as data:

- times of observations $(t_1, \dots, t_n)$;
- observations $(z_1, \dots, z_n)$
- design matrix for fixed effects (`X_fe` in Appendix B.1), provided by mgcv;
- design matrix for random effects (`X_re` in Appendix B.1), provided by mgcv;
- smoothness matrix (`S` in Appendix B.1), provided by mgcv;

and the following arguments as parameters:

- coefficients for fixed effects (`coeff_fe` in Appendix B.1);
- coefficients for random effects (`coeff_re` in Appendix B.1);
- smoothness parameters $\boldsymbol{\lambda}$.

The objective function (and its gradient) can then be defined in R with the TMB function `MakeADFun` applied to the joint negative log-likelihood, using the argument 'random' to specify that the basis coefficients `coeff_re` should be treated as random effects. It can then be passed to a numerical optimiser, e.g. `optim` or `nlm`, to perform maximum likelihood estimation of the fixed effect parameters. After optimisation, the function `sdreport` can be used to obtain estimates of the random effect parameters, as well as a joint precision matrix for the fixed and random effects. Posterior samples of all model parameters can be generated from a multivariate normal distribution, where the covariance matrix is the inverse of this precision matrix.

B.3 Package smoothSDE

This method is implemented in the R package `smoothSDE`, available at `github.com/TheoMichelot/smoothSDE`. The package provides functions for model fitting, uncertainty estimation, model checking, and model plots, and we hope that it will greatly facilitate the application of varying-coefficient SDEs. Here, we present the code required to fit the varying-coefficient model applied to the elephant data set from Wall et al. (2014), in Section 3.2 of the main text, to showcase its use.

First, we download the data from the Movebank data repository, and create a data frame with columns for ID, time (as numeric), response variables (here, easting and northing), and covariates (here, temperature),

```r
# Load package and set seed for reproducibility
library(smoothSDE)
set.seed(58652)

# Load data and keep relevant columns
URL <- paste0("https://www.datarepository.movebank.org/bitstream/handle/",
              "10255/move.373/Elliptical%20Time-Density%20Model%20%28Wall%",
              "20et%20al.%202014%29%20African%20Elephant%20Dataset%20%",
              "28Source-Save%20the%20Elephants%29.csv")
raw <- read.csv(url(URL))
keep_cols <- c(11, 13, 14, 17, 6)
raw_cols <- raw[, keep_cols]
colnames(raw_cols) <- c("ID", "x", "y", "date", "temp")

# Only keep five months to eliminate seasonal effects
track <- subset(raw_cols, ID == unique(ID)[1])
track$date <- as.POSIXlt(track$date, tz = "GMT")
track$time <- as.numeric(track$date - min(track$date))/3600
```

```r
keep_rows <- which(track$date > as.POSIXct("2009-05-01 00:00:00") &
                       track$date < as.POSIXct("2009-09-30 23:59:59"))
track <- track[keep_rows,]

# Convert to km
track$x <- track$x/1000
track$y <- track$y/1000
```

```r
# Data set including ID, time, responses (x, y), and covariate (temp)
head(track)
```

```
##                ID        x        y                date temp time
## 9689 Salif Keita 572.3427 1675.424 2009-05-01 00:00:00   33 9703
## 9690 Salif Keita 572.5443 1675.392 2009-05-01 01:00:00   32 9704
## 9691 Salif Keita 572.6159 1675.339 2009-05-01 02:00:00   31 9705
## 9692 Salif Keita 572.7745 1675.101 2009-05-01 03:00:00   31 9706
## 9693 Salif Keita 572.8844 1675.065 2009-05-01 04:00:00   31 9707
## 9694 Salif Keita 573.6659 1674.322 2009-05-01 05:00:00   30 9708
```

For this analysis, we use the velocity Ornstein-Uhlenbeck model (also called continuous-time correlated random walk, "CTCRW"), which has two parameters: 'beta' (mean reversion parameter) and 'sigma' (variance parameter), defined in Johnson et al. (2008). We define formulas for the SDE parameters to express covariate dependence, using the syntax from mgcv for smooth terms and random effects. Both parameters are specified as functions of the temperature covariate, using thin-plate regression splines,

```r
# Model formulas for CTCRW parameters
formulas <- list(beta = ~ s(temp, k = 10, bs = "ts"),
                 sigma = ~ s(temp, k = 10, bs = "ts"))
```

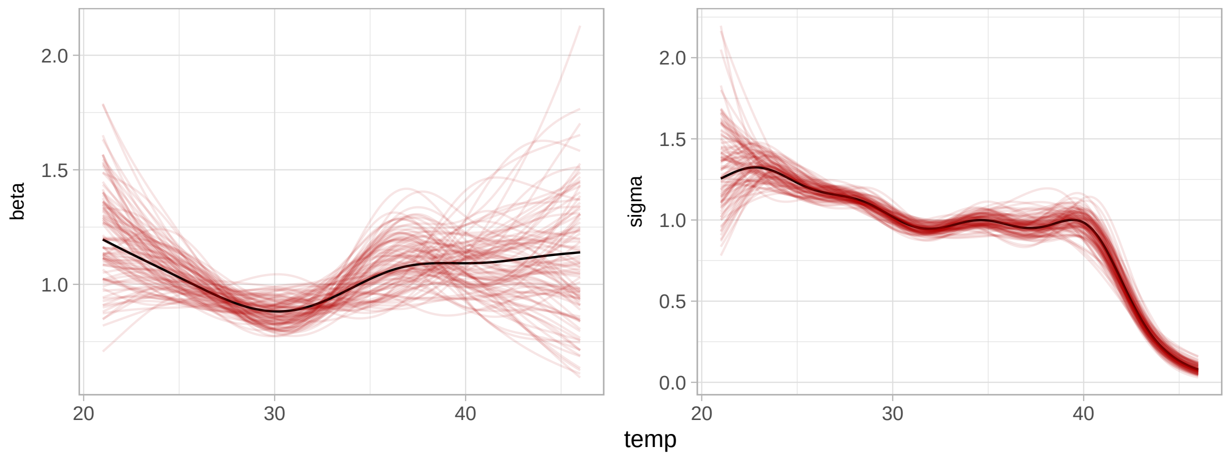
We then create the SDE as an object, which encapsulates the data and model formulas,

```r
# Type of SDE model: continuous-time correlated random walk
type <- "CTCRW"

# Create SDE model object
my_sde <- SDE$new(formulas = formulas, data = track, type = type,
                  response = c("x", "y"))

# Fit model
```

```
my_sde$fit()
```

After model fitting, estimates of the SDE parameters, and posterior samples, can be plotted as functions of the covariates,

```
my_sde$plot_par("temp", n_post = 100)
```



Appendix C Simulation study

We ran simulations for two different model formulations, to investigate the performance of the inference method presented in Section 3 of the manuscript.

Scenario 1 We considered a Brownian motion with drift, where the parameters r_t (drift) and s_t (diffusion) were functions of a covariate x_{1t}. We generated the covariate as Brownian motion (with zero drift) over the time period of the simulations, and then scaled it to $[0, 1]$. For a choice of functions r_t and s_t (shown in Figure S2), we ran 50 simulations. For each, we simulated 10^5 data points at a fine time resolution ($\Delta = 0.01$) to make sure that the discretization error of simulation was negligible. We then downsampled the time series by keeping 2000 observations at random, to assess the performance of the method to recover the model parameters from data collected at irregular time intervals. We fitted the model to each downsampled data set, and derived estimates of r_t and s_t. Figure S2 shows the true r_t and s_t and the 50 estimates as functions of the covariate x_{1t}. The splines generally fitted the true functions well, although most of them did not capture the detailed oscillations in the drift parameter r_t over the lower end of the covariate range. This is because the smoothness of the true function varied over the covariate range (less smooth for low values, more smooth for high values), and the estimated smoothness can therefore be viewed as an average. This could be addressed with adaptive smoothing, i.e. by letting the degree of smoothness vary across values of the covariate (Wood, 2017, Section 5.3).

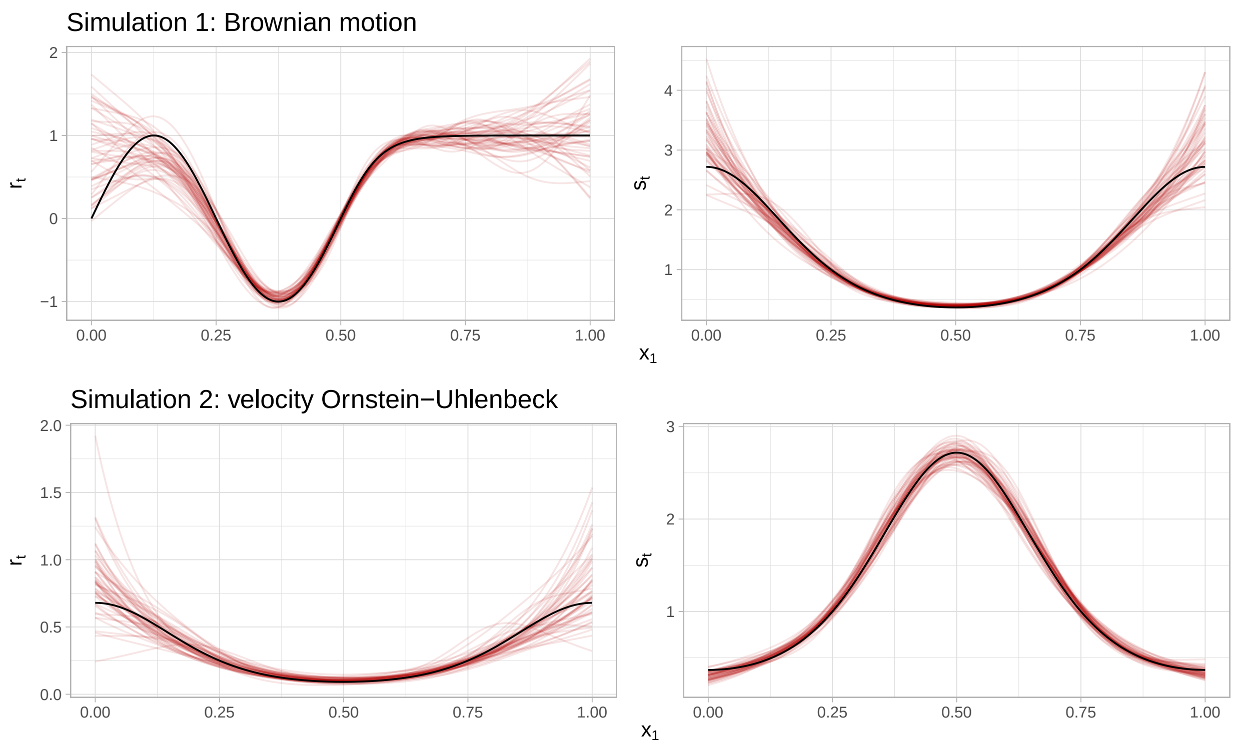


Figure S2: *Results of simulation scenario 1 (top row) and scenario 2 (bottom row). The red lines are the 50 estimated drift parameters r_t and diffusion parameters s_t as functions of the covariate x_1, and the black lines are the true functions.*

Scenario 2 In the second scenario, we assessed the performance of maximum likelihood estimation in the case where the diffusion process is not directly observed, and the Kalman filter needs to be implemented, as described in Section 2.2.1. We simulated 50 trajectories from the velocity OU model presented in Appendix A, following the same procedure as in the first scenario to obtain irregular observations. The parameters r_t and s_t used in the simulations were functions of one covariate, like in the previous scenario. We implemented the Kalman filter, to make inference about the parameters of the latent velocity process, based on the simulated observations. Figure S2 shows the results from the 50 simulations. The smoothness and general shape of both parameters r_t and s_t were recovered in all experiments.

Confidence interval coverage We ran another set of simulations to assess the coverage of Wald confidence intervals obtained using the joint precision matrix given by TMB (using `sdreport` as described in Appendix B.2). We simulated data from a varying-coefficient Brownian motion with drift, using the same drift and diffusion functions as in Scenario 1. Then, we fitted the model using TMB, and generated 1000 posterior samples for r_t and s_t on a grid over the covariate range. We derived pointwise 95% confidence intervals on that grid, and checked whether they included the true value. We repeated this experiment 1000 times, and derived the proportions of confidence intervals that included the true value. Average coverage over the covariate range was 95.3% for r_t and 96.7% for s_t, close to the expected 95%.

Appendix D Application to oil prices

This application involves the analysis of a time series of oil prices, inspired by García et al. (2017). We downloaded daily prices on WTI crude oil between 2 January 1986 and 30 March 2020 from the US Energy Information Administration website (`eia.gov/dnav/pet/pet_pri_spt_s1_d.htm`, accessed on 31 March 2020), and derived the log-returns as $R_i = \log(P_{i+1}/P_i)$ for day i, where P_i is the price of a barrel of crude oil. There were occasional missing values in this daily time series (easily accommodated in this continuous-time framework), resulting in a total of 8628 data points. We then fitted a varying-coefficinet Brownian motion with drift to the log-returns R_i, where the drift and diffusion coefficients were specified as functions of the process value R_i. Model fitting took around 1 min on a 1.3GHz Intel i7 CPU.

Estimates of the drift parameter r_t and diffusion parameter s_t are shown as functions of the log-returns in Figure S3. Similarly to García et al. (2017), we found that the drift r_t was negative for positive log-returns, and positive for negative log-returns, i.e., the process was attracted to zero. The diffusion parameter s_t increased with the absolute value of the log-returns, meaning that the process was more volatile when it took large (positive or negative) values. Based on a similar observation, García et al. (2017) suggested that s_t may be expressed as a quadratic function of the log-returns. Interestingly, our results indicate that this may be inappropriate, because the rate of increase of s_t decreases below -0.07 and above 0.07.

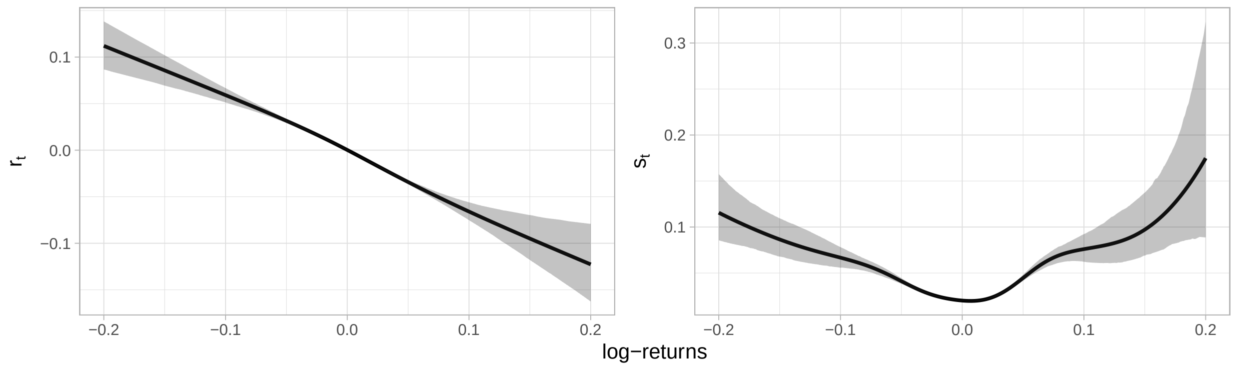


Figure S3: *Results of crude oil price analysis, showing the drift parameter r_t and diffusion parameter s_t as functions of the log-returns. The black lines are the mean estimates, and the grey shaded areas are 95% confidence bands.*

Appendix E Supplementary information for beaked whale analysis

E.1 Euler angles

Figure S4 shows an illustration of the three Euler angles derived from accelerometer data in the beaked whale analysis of Section 3.3: pitch, roll, and heading. They quantify the posture of the whale in the water.

E.2 Heavy-tailed model formulation

The three variables used in the beaked whale analysis of Section 3.3 of the main paper (pitch, roll, and heading) had heavy-tailed increments, which did not satisfy the assumption of normality made by Brownian motion. Here, we describe how we modified that model to accommodate the heavy tails.

Formulation Let $D_i = Z_{t_{i+1}} - Z_{t_i}$ denote the increment in the process between two successive times of observation t_i and t_{i+1}. We write each process increment as

$$D_i = r_{t_i}\Delta_i + \tilde{s}_{t_i}\sqrt{\Delta_i}X_i$$

where

- $\Delta_i = t_{i+1} - t_i$ is the time interval;

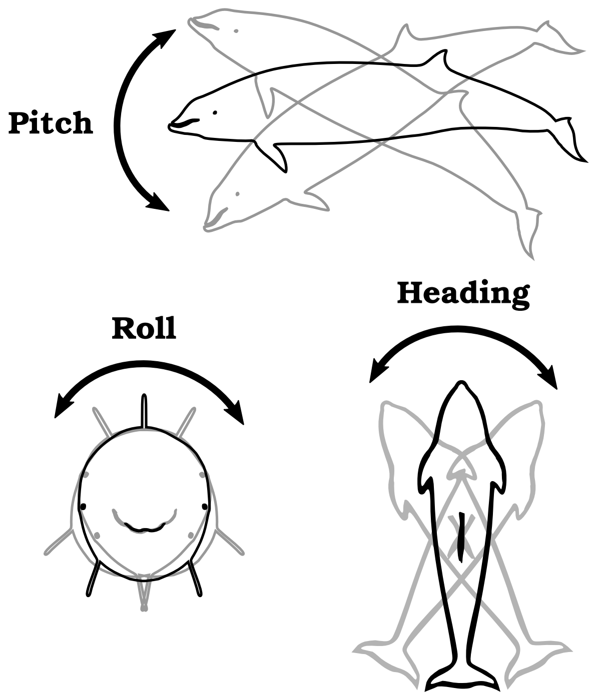


Figure S4: *Illustration of Euler angles.*

- r_t is a time-varying location parameter;
- $\tilde{s}_t$ is a time-varying scale parameter;
- for all i, X_i follows a Student's t distribution with $\nu > 2$ degrees of freedom.

Then, increments of the process follow a heavy-tailed generalized Student's t distribution, with mean $r_{t_i}\Delta_i$ and standard deviation $\tilde{s}_{t_i}\sqrt{\Delta_i}\sqrt{\nu/(\nu-2)}$. This is based on the standard assumption of Brownian motion that the increment mean scales linearly with the time interval, and the increment standard deviation scales linearly with the square root of the time interval. For ease of interpretation, in the paper we present results for the standard deviation parameter $s_t = \tilde{s}_t\sqrt{\nu/(\nu-2)}$ rather than for the scale parameter $\tilde{s}_t$.

Likelihood We consider n observations $(z_1, \dots, z_n)$ from the process, collected at times $t_1 < \dots < t_n$. Under the assumption that increments are independent, we can write the full approximate likelihood as

$$L(\theta|z_1, \dots, z_n) = \prod_{i=2}^{n} [Z_{t_i} = z_i | Z_{t_{i-1}} = z_{i-1}, r_i, \tilde{s}_i],$$

where $r_i = r_{t_i}$ and $\tilde{s}_i = \tilde{s}_{t_i}$.

The transition density, i.e., the pdf of an increment of the process, is given by

$$[Z_{t_i} = z_i | Z_{t_{i-1}} = z_{i-1}, r_i, \tilde{s}_i] = f_X\left(\frac{z_{i+1} - z_i - r_i\Delta_i}{\tilde{s}_i\sqrt{\Delta_i}}\right) \times \frac{1}{\tilde{s}_i\sqrt{\Delta_i}},$$

where f_X is the pdf of a t distribution with ν degrees of freedom, and the term $1/(\tilde{s}_i\sqrt{\Delta_i})$ is the Jacobian of the transformation from X_i to the increment D_i.

Residuals We define the residuals of this model as

$$\epsilon_i = \frac{z_{i+1} - z_i - r_i\Delta_i}{\tilde{s}_i\sqrt{\Delta_i}}$$

using the notation from the previous sections. Under the assumptions of the model and the discretization, these residuals follow a t distribution with ν degrees of freedom.

E.3 ACF plots of residuals

Figure S5 shows autocorrelation function plots of the residuals for the three data variables in the beaked whale analysis (pitch, roll, and heading). Pitch and roll both display some negative autocorrelation over a few time lags (20-30 sec), which may be due to cycles in the motions of whales that were not captured by the model. The high positive autocorrelation in the residuals for heading suggest a strong tendency for whales to persist in the direction of rotation in the horizontal plane (i.e., either circling to the left or to the right).

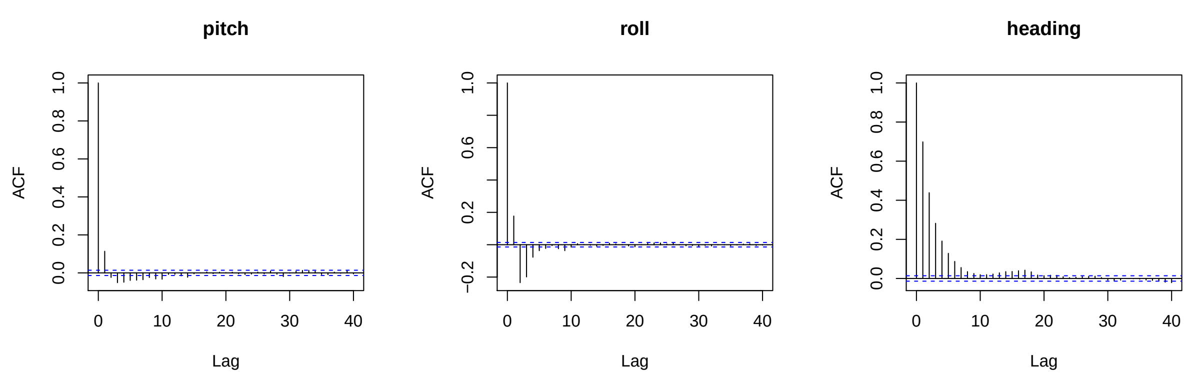


Figure S5: *Autocorrelation function plots of residuals for beaked whale analysis. One lag unit corresponds to five seconds.*

References

Aït-Sahalia, Y. and Kimmel, R. (2007). Maximum likelihood estimation of stochastic volatility models. *Journal of financial economics*, 83(2):413–452.

Blackwell, P. G. (1997). Random diffusion models for animal movement. *Ecological Modelling*, 100(1-3):87–102.

Dennis, B., Munholland, P. L., and Scott, J. M. (1991). Estimation of growth and extinction parameters for endangered species. *Ecological monographs*, 61(2):115–143.

Dunn, J. E. and Gipson, P. S. (1977). Analysis of radio telemetry data in studies of home range. *Biometrics*, pages 85–101.

García, C. A., Otero, A., Felix, P., Presedo, J., and Marquez, D. G. (2017). Nonparametric estimation of stochastic differential equations with sparse Gaussian processes. *Physical Review E*, 96(2):022104.

Johnson, D. S., London, J. M., Lea, M.-A., and Durban, J. W. (2008). Continuous-time correlated random walk model for animal telemetry data. *Ecology*, 89(5):1208–1215.

Kristensen, K., Nielsen, A., Berg, C., Skaug, H., and Bell, B. (2016). TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software, Articles*, 70(5):1–21.

Michelot, T. and Blackwell, P. G. (2019). State-switching continuous-time correlated random walks. *Methods in Ecology and Evolution*, 10(5):637–649.

Preisler, H. K., Ager, A. A., Johnson, B. K., and Kie, J. G. (2004). Modeling animal movements using stochastic differential equations. *Environmetrics*, 15(7):643–657.

Russell, J. C., Hanks, E. M., Haran, M., Hughes, D., et al. (2018). A spatially varying stochastic differential equation model for animal movement. *The Annals of Applied Statistics*, 12(2):1312–1331.

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the Brownian motion. *Physical review*, 36(5):823.

Vasicek, O. (1977). An equilibrium characterization of the term structure. *Journal of financial economics*, 5(2):177–188.

Wall, J., Wittemyer, G., LeMay, V., Douglas-Hamilton, I., and Klinkenberg, B. (2014). Elliptical time-density model to estimate wildlife utilization distributions. *Methods in Ecology and Evolution*, 5(8):780–790.

Wood, S. N. (2017). *Generalized additive models: an introduction with R*. CRC press. Second Edition.