Switch Point Algoirthm Specifics

This note is a reference for how we select the priors and hyperparameters for our MCMC switch point algorithm. Moreover, we use the ideology to select the priors and hyperparametrers for our MCMC algorithm under the assumption of a Brownian Motion and Ornstein-Uhlenbeck process.

Model: OU with drift and k switches

For $t \in [0, T = N\delta]$, we assume that X_t satisfies the following SDE

$$\mathrm{d}X_t = -\kappa_j(X_t - \nu_j(t - \tau_{i-1}) + X_{\tau_{i-1}})dt + \sqrt{2D_j}\mathrm{d}W_t \qquad \text{for } \tau_{j-1} < t \leq \tau_j$$

for $j = 1, \ldots, k$ where $\tau_0 = 0$ and $\tau_{k+1} = T$.

We further assume that process has been observed at discrete time measurements, $t_n = n\delta$ where $n = 0, \ldots, N,$. Since solutions to SDEs are Markov Processes, the likelihood of observing $\mathbf{x} = (x_0, \ldots, x_N)$ is given by

$$p_{\boldsymbol{\theta}}(\mathbf{X} = \mathbf{x}) = \prod_{n=1}^{N} p_{\boldsymbol{\theta}}(X_n = x_n | X_{n-1} = x_{n-1})$$

where

$$X_{n+1}|_{X_n} \sim N\left(X_n e^{-\Delta_{n+1}\kappa_j} + \nu_j A_{n+1}(\kappa_j) + (1 - e^{-\kappa_j \Delta_{n+1}})(X(\tau_{j-1}) - \nu_j \tau_{j-1}), \sigma^2_{n+1}(D_j, \kappa_j)\right)$$

$$A_{n+1}(\kappa_j) = \int_{t_n}^{t_{n+1}} \kappa_j s e^{-\kappa_j(t_{n+1}-s)} ds = t_{n+1} - t_n e^{-\kappa_j \Delta_{n+1}} - \frac{1}{\kappa_j}(1 - e^{-\kappa_j \Delta_{n+1}})$$

$$\sigma^2_{n+1} = \frac{D_j}{\kappa_j}(1 - e^{-2\kappa_j \Delta_{n+1}})$$

In practice, we use the log likelihood rather than the likelihood.

In [2]:
```
## logLikelihood Function for 2D process that is projected onto the micr
otubule.
## so mydata is the longitudinal direction. (ie the position along the m
ictrotuble)
## or for a 1d switching process

ll_switch = function(mydata, time_seq, my_D_vec, my_kg_vec, my_nu_vec, m
y_tau_vec){

    #If the value for kappa is the same for each state
    if(length(my_kg_vec) ==1){my_kg_vec = rep(my_kg_vec, length(my_tau_v
ec) + 1)}
    #If the value for D is the same for each state
    if(length(my_D_vec) ==1){my_D_vec = rep(my_D_vec, length(my_tau_vec)
+1)}
    #If the value for nu is the same for each state
    if(length(my_nu_vec) ==1){my_nu_vec = rep(my_nu_vec, length(my_tau_v
ec)+1)}


    X_switch_vec = rep(mydata[c(1,my_tau_vec[-c(1, length(my_tau_vec
))])], diff(my_tau_vec))
    time_switch_vec = rep(time_seq[c(1,my_tau_vec[-c(1, length(my_tau_ve
c))])], diff(my_tau_vec))

    nu_vec = rep(my_nu_vec, diff(my_tau_vec))
    kg_vec = rep(my_kg_vec, diff(my_tau_vec))
    D_vec = rep(my_D_vec, diff(my_tau_vec))

    rho_plus = exp(-kg_vec*Delta_plus)
    sigma_sq_plus = (D_vec/kg_vec)*(1- exp(-2* kg_vec*Delta_plus))

    A_plus = time_seq[2:num_steps] - time_seq[1:(num_steps-1)]*rho_plus[
2:num_steps] - (1/kg_vec[2:num_steps])*(1 - rho_plus[2:num_steps])
     ##Note that kg_vec[1] corresponds to the starting position
    mu_long = mydata[1:(num_steps-1)]*rho_plus[2:num_steps] + nu_vec[2:n
um_steps]*A_plus+ (X_switch_vec[2:num_steps]- nu_vec[2:num_steps]*time_s
witch_vec[2:num_steps])*(1 - rho_plus[2:num_steps])

    log_like = -1*((num_steps-1)/2*log(2*pi) + (1/2)*sum(log(sigma_sq_pl
us[2:num_steps]))) -(1/2)*sum(( (mydata[2:num_steps] - mu_long )^2/sigma
_sq_plus[2:num_steps]))

    }
```

Prior Selection and Candidate Generating functions

For ease of notation, we drop the subscript j for each parameter.

Diffusivity Constant, D

Prior: Gamma

$$D \sim Gamma(\alpha_D, \beta_D)$$
$$\alpha : \text{shape parameter;} \quad \beta : \text{rate parameter}$$

where $f_D(d) = \frac{\alpha^\beta}{\Gamma(\alpha)} d^{\alpha-1} \exp(-d\beta)$. Here the hyperparameters are α_D, β_D.

Transistion Function: Normal Random Walk

Let σ_D^2 be given. The transition function of going from x to y is given by

$$T(x, y) = \left(\frac{1}{2\pi\sigma_D^2}\right)^2 \exp\left(-\frac{1}{2\sigma_D^2}(y-x)^2\right)$$

(ie $y|x \sim N(x, \sigma_D^2)$)

Spring Constant, κ

Remark: For ease of notation we denote κ/γ by κ where κ is the spring constant and γ is drag coefficent.

Prior: lognormal for $\rho = e^{-\delta\kappa}$

Because we do not know the magnitude of the coefficient κ, we use the parameter $\rho = \exp(-\delta\kappa)$ in our algorithm. We place a log normal prior of ρ,

$$\rho \sim logNorm(loc, shape)$$

where

$$loc = \log\left(\frac{\mathbb{E}(\rho)^2}{\sqrt{\mathbb{E}(\rho)^2 + \mathbb{V}(\rho)}}\right) \quad \text{and} \quad shape = \sqrt{\log(1 + \frac{\mathbb{V}(\rho)}{\mathbb{E}(\rho)^2}}.$$

In this case the hyperparameters, $\mathbb{E}(\rho)$ and $\mathbb{V}(\rho)$.

Transistion Function: lognormal

Let σ_ρ^2 be given, then the transition function fo going from ρ_0 to ρ_1 is given by

$$T(\rho_0, \rho_1) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(\frac{-(\ln x - \mu)^2}{2\sigma^2}\right)$$

where $\mu = \log \frac{\rho_0}{\sqrt{\rho_0^2 + \sigma_\rho^2}}$ and $\sigma = \left(\log(1 + \frac{\sigma_\rho^2}{\rho_0^2})\right)^{1/2}$.

Motor Velocity: ν

Prior: Uniform(-1,1)

$$\nu \sim Uniform(a, b)$$

for $a < b$. Note: We use this as a semi-informative prior because we known that the motors cannot travel more than $1\mu m/s$, ie $a = -1$ and $b = 1$.

Transition function: Normal Random Walk

Let σ_ν^2 be given. The transition function of going from x to y is given by

$$T(x, y) = \left(\frac{1}{2\pi\sigma_\nu^2}\right)^2 \exp\left(-\frac{1}{2\sigma_\nu^2}(y-x)^2\right)$$

(ie $y|x \sim N(x, \sigma_\nu^2)$)

Switch Points, τ

Prior: Discrete uniform on $1, \ldots, N$,

Transistion function: Discrete Uniform Random Walk

Let σ_τ be given. Then the transisition function of going from τ_0 to τ_1 is given by

$$T(\tau_0, \tau_1) = \begin{cases} 1 & \text{if } |\tau_0 - \tau_1| \leq 2 * \sigma_\tau \\ 0 & \text{if } |\tau_0 - \tau_1| > 2 * \sigma_\tau \end{cases}$$

Picking the Hyperparmeters

Diffusivity Constant

Hyperpameters for Prior

While our likelihood can handle different diffusivity constants in each state, we set all the diffusivity constant for each state to be same. This is because the diffusivity constant is a property of the tagged particle (cargo) not of the motors.

We use maximum likelihood estimator for D (assuming Brownian Motion) and for $j = 1, \ldots, k$ set

$$\alpha_{D_j} = \hat{D}_{mle} \quad \text{and} \quad \beta_{D_j} = 1.$$

Hyperparameter for the transition function

$\tau_D^2 = \hat{D}_{mle}^2$

In [3]:
```
## MLE for diffuvity
D_mle_function = function(mydata, dt){(sum(diff(mydata)^2))/(2*dt*length
(diff(mydata)))}
```

Switch Points τ

Initializing

We arbitrarily set the switch points be uniform spaced. See R code snippet below this block.

Hyperparameter for the transition function

Because we want the chain for τ to be explore the parameter space $(1, 2, \ldots, N)$, we set $\sigma_\tau = 0.1 * \frac{N}{k}$ for all switch points. See R code snippet below this block.

In [ ]:
```
num_obs = 100 # Number of measurements
num_switch = 4 # Number of switch points

## Initializing tau
tau_est = floor(num_obs/(num_switch + 1))*(1:num_switch)

##
sigma_tau = ceiling((length(num_obs)/num_switch)*0.1)
```

Autoregressive parameter ρ

Let κ_M be a large value of κ that is selected by the user and $\kappa_m \geq 0$ be a small value of κ that is selected by the user. If κ_m is not specified by the user the default is the time step.

Currently we have $\kappa_M = 20$ and $\kappa_m = dt$.

Hyperpameters for Prior

Case 1: When inference is done using the longitudinal and transverse direction rather than the Euclidean coordinate system. We model the tranvserse direction by a centered OU process ($\nu = 0$), which can be written as an AR(1) process with autoregressive parameter $e^{-\delta\kappa}$. Using this fact, we estimate the hyperparameters by considering the ACF(1) value of the tranvserse position process. We set the mean of the prior lognormal distribution to $ACF(1)$

$$\mathbb{E}(\rho_j)^{(0)} = \begin{cases} exp(-\delta * \kappa_M) & \text{if } ACF_{x^\perp}(1) < 0 \\ ACF_{x^\perp}(1) & \text{if } 0 < ACF_{x^\perp}(1) < 1 \\ exp(-\delta * \kappa_m) & \text{if } 1 < ACF_{x^\perp}(1) \end{cases}$$

and we set $\mathbb{V}(\rho)^{(0)} = 0.1$ for all j. Let $\rho_{j,0} = \mathbb{E}(\rho_j)^{(0)}$. We obtain the location and shape hyperparameters for the prior by the following transformation:

$$\mu = \log \frac{\rho_{j,0}^2}{\sqrt{\rho_{j,0}^2 + 0.1}} \quad \text{and} \quad \sigma = \left(\log \left(1 + \frac{0.1}{\rho_{j,0}^2} \right) \right)^{1/2}$$

Case 2: Set $\mathbb{E}\{\rho_j\}^{(0)} = \exp(-\delta\kappa_M)$ for all j.

Hyperparameter for the transition function

For all j, we set variance of the transistion function to the same value denoted $\sigma_\rho^2 = 0.1$. This value is picked to obtain a proper acceptance rate.

Initializing

We initialize $\rho_i^{(0)} = \mathbb{E}(\rho_i)^{(0)}$.

Velocity ν

Hyperpameters for Prior

We use domain knowledge to select $a = -1 \mu m/s$ and $b = 1 \mu m/s$.

Hyperparameter for the transition function

Let $\nu_j^{(0)}$ be the initial velocity for the jth state. Then for $j = 1, \ldots, k+1$ we set $\sigma^2_{\nu_j} = \min\left(\frac{\nu_j^{(0)}}{2}, 0.005\right)$.

Remark: This was done in the case where the velocity is approximately zero. In actuality we should probably figure out something out for the case when the velocity should be zero.

Initializing ν_j

Currently I have $\nu_j^{(0)}$ as the slope of LSQR in the jth state. In the case that $\nu_j^{(0)} \notin (-1, 1)$ then we randomly sample from unif($-1, 1$). It might just be better to sample all of them from the prior distribution and pick the