

Stochastic Volatility

Example 4 reproduced from the Doucet, Johansen paper (15 years later).

Prior

$$X_0 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1 - \alpha^2}\right)$$

Transition

$$X_t = \alpha X_{t-1} + \sigma V_t, \quad \text{where } V_t \sim N(0, 1)$$

Emission

$$Y_t = \beta \exp(X_t/2) W_t, \quad \text{where } W_t \sim N(0, 1)$$

The parameters for generating the data are $\alpha = 0.91$, $\sigma = 1$, $\beta = 0.5$, and we generate $T = 500$ states.

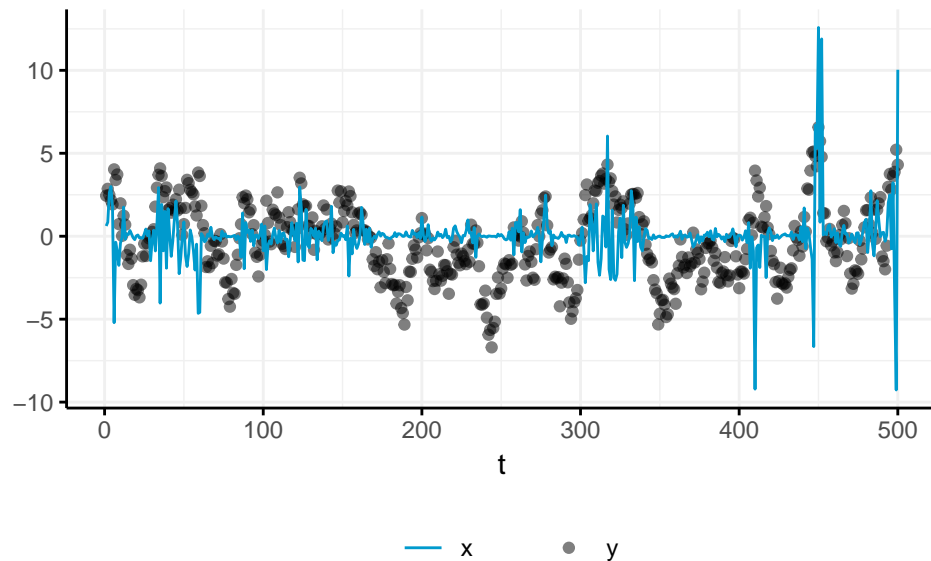
Generate data

```
set.seed(42)
alpha_true <- 0.91
sigma_true <- 1
beta_true <- 0.5
T <- 500

generate_sv_data <- function(alpha = alpha_true,
                              beta = beta_true,
                              sigma = sigma_true,
                              n_states = T) {
  x <- rep(0, T + 1)
  y <- rep(0, T + 1)
  x[1] <- rnorm(1, mean = 0, sd = sigma / sqrt(1 - alpha^2))
  y[1] <- NA
  for (t in 2:(T + 1)) {
    x[t] <- rnorm(1, mean = alpha * x[t - 1], sd = sigma)
    y[t] <- rnorm(1, mean = 0, sd = beta * exp(x[t] / 2))
  }
  return(list(x = x, y = y))
}

## Generate data
data <- generate_sv_data()
x <- data$x
y <- data$y
df <- data.frame(x = x, y = y, t = 0:T)
df %>%
  filter(row_number() != 1) %>% # ignore the first values (x=x_0, y=NA)
  ggplot() +
  geom_point(aes(x=t, y=x, fill='y'), alpha=0.5) +
  geom_line(aes(x=t, y=y, color='x')) +
  scale_color_manual(values=c('x'='deepskyblue3')) +
  scale_fill_manual(values=c('y'='black')) +
```

```
theme(legend.position = 'bottom')+
labs(y='',fill='',color='')
```

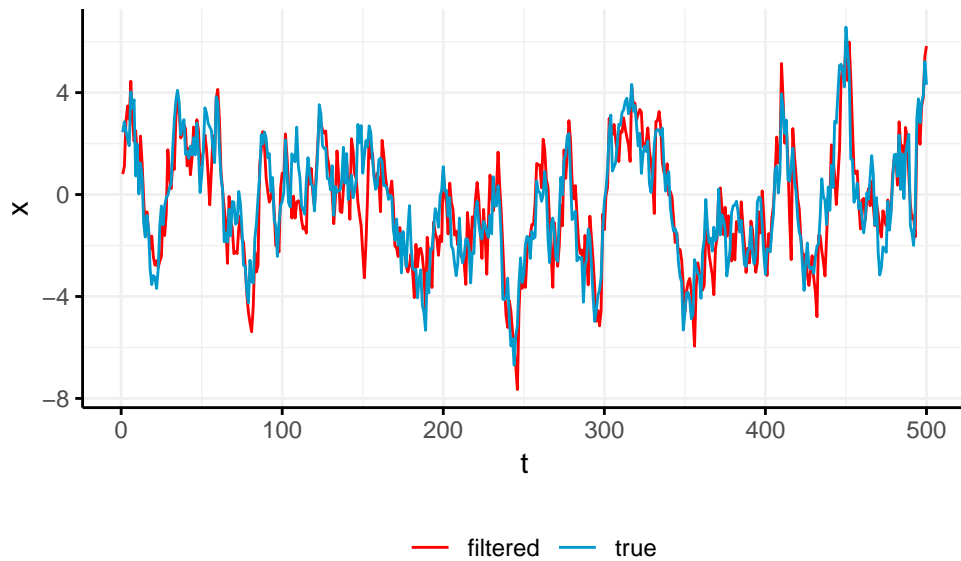


Apply the auxiliary particle filter to the data.

```
## LOAD RCPP CODE
sourceCpp("SV.cpp")

## COMPUTE FILTERED STATES
x_hat_filtered <- ap_filter_sv(y,
  theta = c(alpha_true, beta_true, sigma_true),
  n_particles = 400
)

## PLOT THE REAL VS FILTERED STATES
data.frame(true = x, filtered = x_hat_filtered, t = 0:T) %>%
  gather(key = "state", value = "x", -t) %>%
  filter(t != 0) %>% # ignore the first values (x=x_0, y=NA)
  ggplot(aes(x = t, y = x, color = state)) +
  geom_line() +
  labs(color = "", group = "") +
  scale_color_manual(values = c(true = "deepskyblue3", filtered = "red"))+
  theme(legend.position = "bottom")
```



Equivalent code in R.

```
ap_filter_sv_R <- function(y,
                           theta = c(alpha_true, beta_true, sigma_true),
                           n_particles = 400) {
  alpha <- theta[1]
  beta <- theta[2]
  sigma <- theta[3]

  particles <- matrix(0, nrow = n_particles, ncol = T + 1)
  x_hat_filtered <- rep(0, T + 1)
  initial_particles <- rnorm(n_particles, 0, sqrt(sigma^2 / (1 - alpha^2)))

  particles[, 1] <- initial_particles
  for (t in 2:(T + 1)) {
    ## RESAMPLE
    weights_1 <- dnorm(y[t], 0, beta * exp(alpha * particles[, t - 1] / 2))
    weights_1_norm <- weights_1 / sum(weights_1)
    particles[, t - 1] <- sample(particles[, t - 1], n_particles, TRUE, weights_1_norm)

    ## PROPAGATE
    particles[, t] <- alpha * particles[, t - 1] + sigma * rnorm(n_particles, 0, 1)

    ## RESAMPLE
    weights_2 <- dnorm(y[t], 0, beta * exp(particles[, t] / 2))
    weights_2_norm <- weights_2 / sum(weights_2)
    particles[, t] <- sample(particles[, t], n_particles, TRUE, weights_2_norm)
    x_hat_filtered[t] <- mean(particles[, t])
  }
  return(x_hat_filtered)
}
```