Lab3 SSM

Milda Poceviciute

4 October 2018

Question 1

A Hidden Markov model is implemented with the following features:

Transition model:

$$p(z_t|z_{t-1}) = \frac{1}{3}[N(z_t|z_{t-1}, 1) + N(z_t|z_{t-1} + 1, 1) + N(z_t|z_{t-1} + 2, 1)]$$

Emission model:

$$p(x_t|z_t) = \frac{1}{3}[N(x_t|z_t, 1) + N(x_t|z_t + 1, 1) + N(x_t|z_t - 1, 1)]$$

Initial model:

$$p(z_1) = Uniform(0, 100)$$

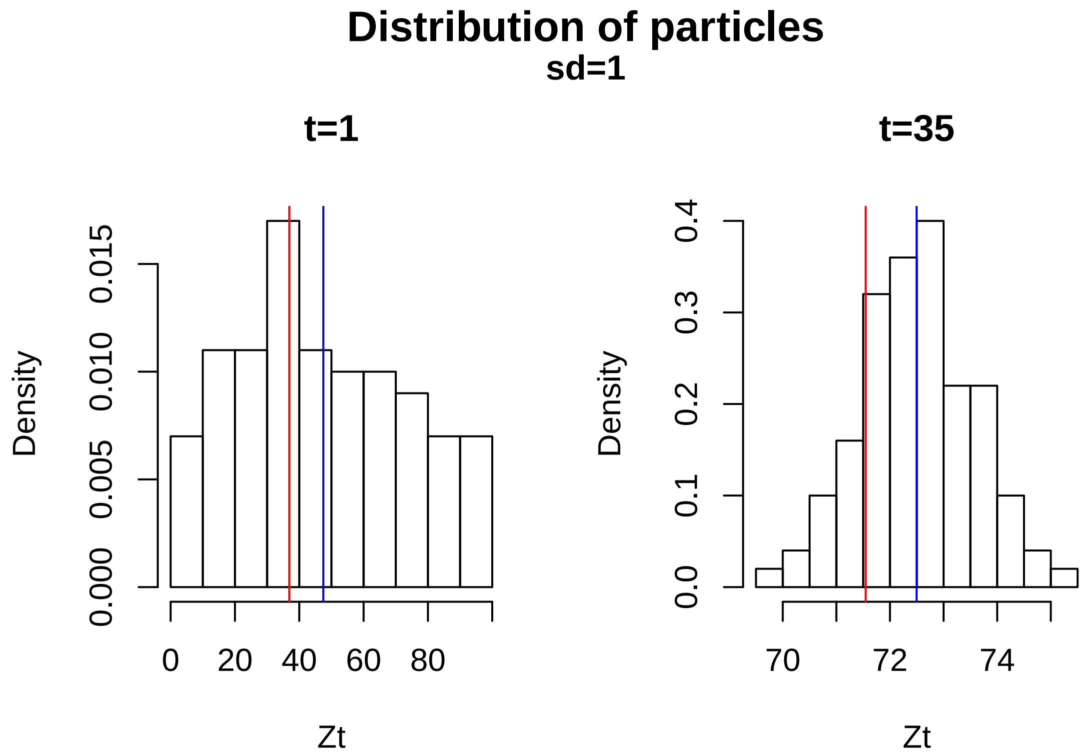
The functions are as follows:

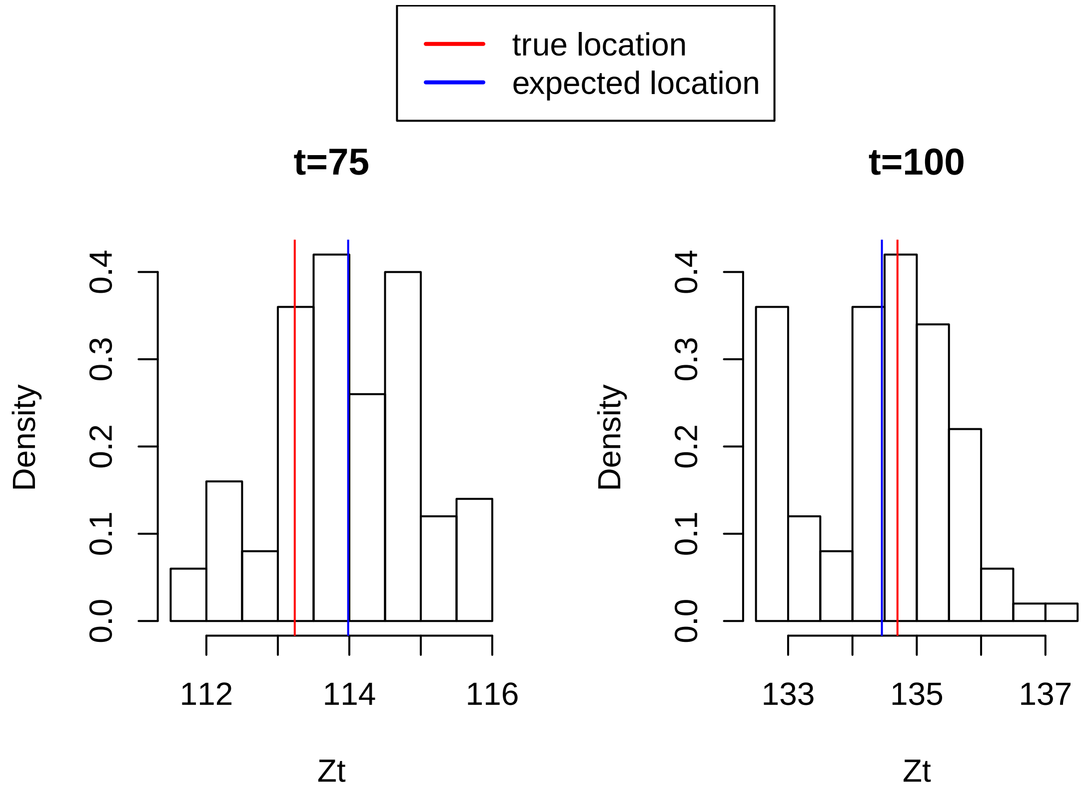
```
# M - number of particals Iter - number of time steps T z is
# latent variable x is the observation

# Function for sampling from a mixture model
sampleMix <- function(mean1, mean2, mean3, sdX) {
    u <- sample(1:3, 1)
    means <- c(mean1, mean2, mean3)
    return(rnorm(1, mean = means[u], sd = sdX))
}

## Function that creates the SSM model
create_SSM <- function(z0, TT, sdX = 1) {
    Zt <- c(z0)

    Xt <- sampleMix(mean1 = z0, mean2 = (z0 - 1), mean3 = (z0 +
        1), sd = 1)

    for (i in 2:TT) {
        Zt[i] <- sampleMix(mean1 = Zt[i - 1], mean2 = (Zt[i -
            1] + 1), mean3 = (Zt[i - 1] + 2), sdX = 1)
        Xt[i] <- sampleMix(mean1 = Zt[i], mean2 = (Zt[i] - 1),
            mean3 = (Zt[i] + 1), sdX = sdX)
    }
    return(list(sates = Zt, observations = Xt))

}

generateX <- function(Zt, TT, sdX = 1) {
    Xt <- c()
    for (i in 1:TT) {
        Xt[i] <- sampleMix(mean1 = Zt[i], mean2 = (Zt[i] - 1),
            mean3 = (Zt[i] + 1), sdX = sdX)
```

```
    }
    return(Xt)
}

## Particle filter function
particles <- function(Obs, M, Iter, sdX = 1, correction = TRUE) {
    Xt <- Obs

    # Initialise particles for the latent variable
    Zt <- matrix(ncol = Iter, nrow = M)
    wt <- matrix(ncol = Iter, nrow = M)
    Zt_bar <- matrix(ncol = Iter, nrow = M)
    # fill the initial time step
    Zt[, 1] <- runif(M, 0, 100)
    # Add arbitrary values in order for the indexing to work
    # later
    Zt_bar[, 1] <- rep.int(1, M)
    wt[, 1] <- rep.int(1, M)

    for (t in 2:Iter) {
        # Prediction
        for (m in 1:M) {
            # sample from the transition model (current Z_t depends on
            # the Z_{t-1}) Zt_bar[m,t] <-
            # (rnorm(1,mean=Zt[m,t-1],sd=1)+rnorm(1,mean=(Zt[m,t-1]+1),sd=1)+rnorm(1,mean=(Zt[m,t-1]+2)
            Zt_bar[m, t] <- sampleMix(mean1 = Zt[m, t - 1], mean2 = (Zt[m,
                t - 1] + 1), mean3 = (Zt[m, t - 1] + 2), sdX = 1)
            # Calculate weights
            wt[m, t] <- (dnorm(Xt[t], mean = Zt_bar[m, t], sd = sdX) +
                dnorm(Xt[t], mean = (Zt_bar[m, t] - 1), sd = sdX) +
                dnorm(Xt[t], mean = (Zt_bar[m, t] + 1), sd = sdX))/3
        }
        # Correction
        if (correction) {
            Zt[, t] <- sample(Zt_bar[, t], M, replace = TRUE,
                prob = wt[, t])
        } else {
            Zt[, t] <- Zt_bar[, t]
        }

    }
    return(list(Zt = Zt, Zt_bar = Zt_bar, wt = wt))

}
```

Afterward the states $z_{1:T}$ (using the initial and transition models) and observations $x_{1:T}$ of sensor readings (using the emission model) are simulated for $T = 100$ time steps. The observations are used in the particle filter algorithm with 100 particles in order to predict the possible states (location of the robot).

I chose time points 1, 35, 75, and 100 for which I plot the distribution of the particles (a histogram), and I indicate on them what is the true value of the z_t (the red line) and what is the expected value by the particle filter (the blue line) at that particular time step.

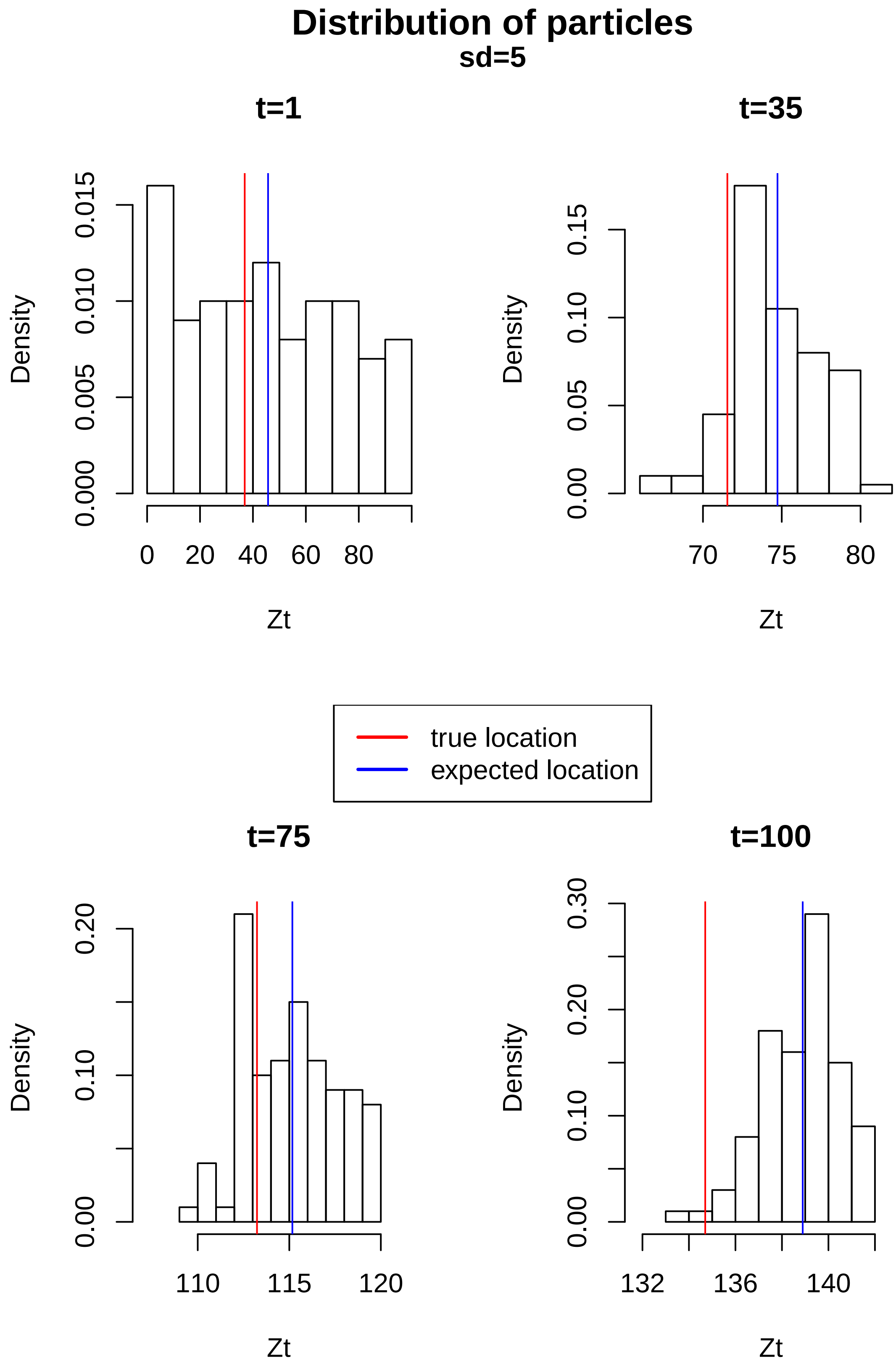

Distribution of particles
sd=1
t=1
t=35
Density
0.000
0.005
0.010
0.015
0
20
40
60
80
Zt
0.0
0.1
0.2
0.3
0.4
70
72
74
Zt



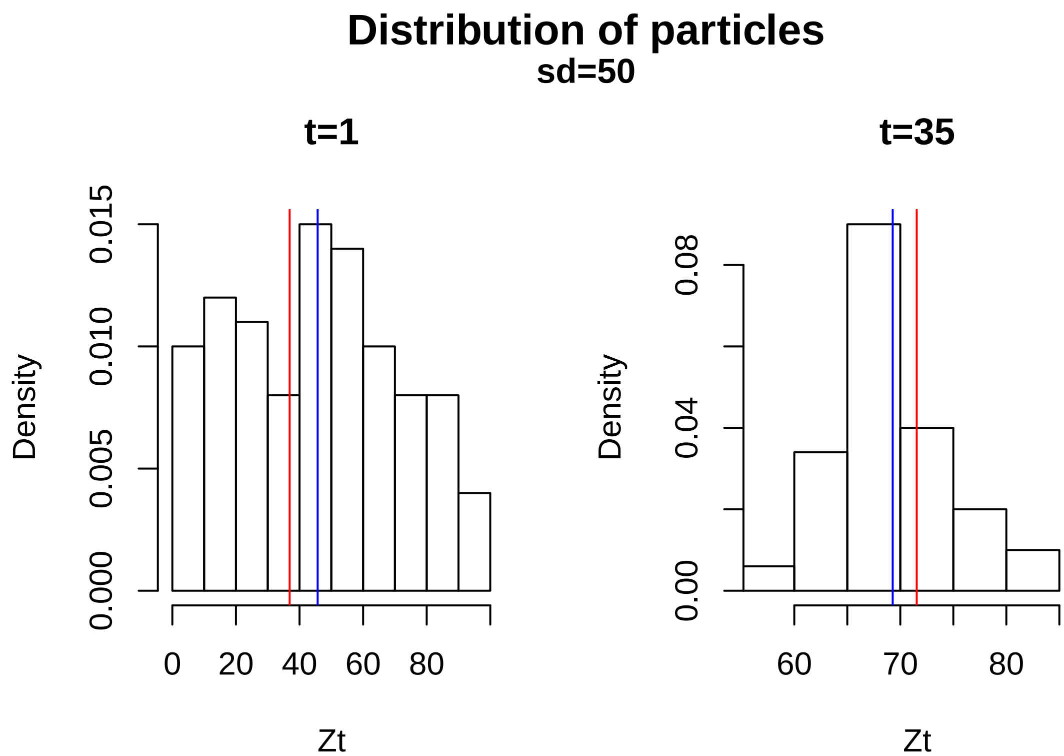
From the histograms above we can see that the algorithm converges pretty well: the spread of the distribution decreases from 100 to below 20 in the last time step. This indicates that the algortihm is more certain about the position of the robot. In all four plots the true position (the red line) is pretty close to the expected position based on the mean of the particles distribution (the blue line).

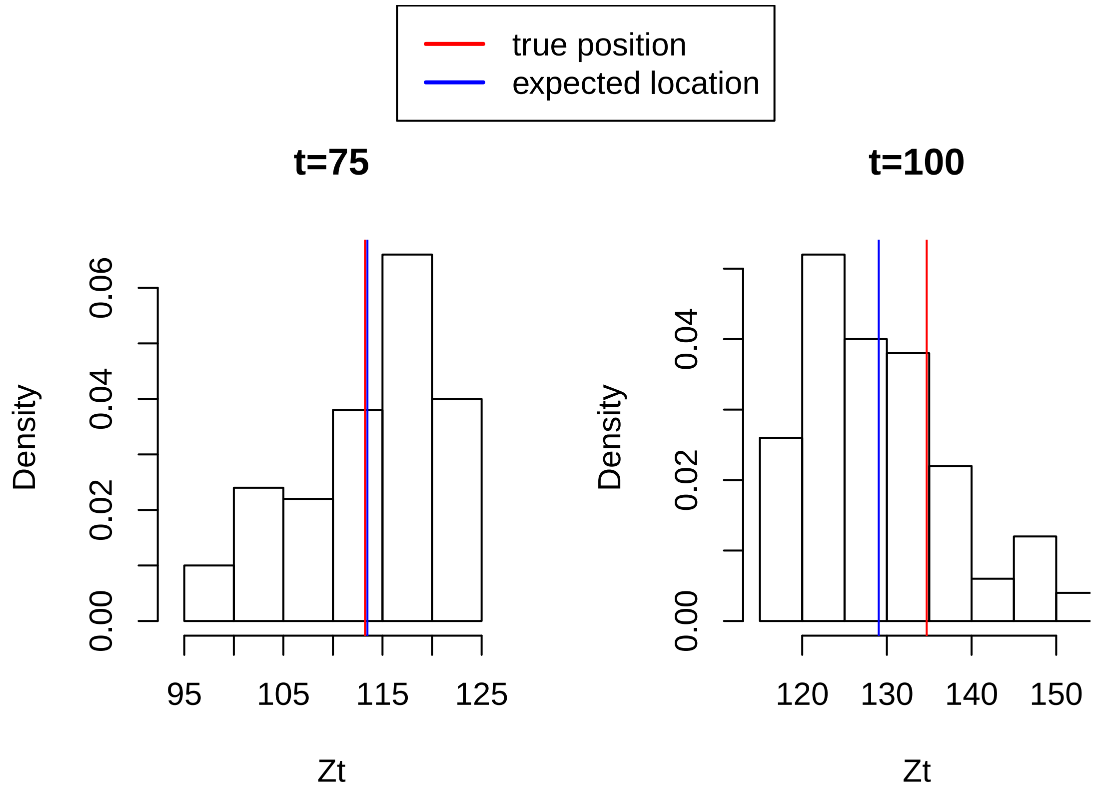
Question 2

In this question the HMM is modified by changing the emission models. Firstly, the standard deviation is set to 5, and later it is increased to 50. The states $z_{1:T}$ are kept the same, but new observations $x_{1:T}$ are generated. The particle filter algorithm is applied again on both cases.


Distribution of particles
sd=5
t=1
Density
Zt
t=35
true location
expected location
t=75
t=100

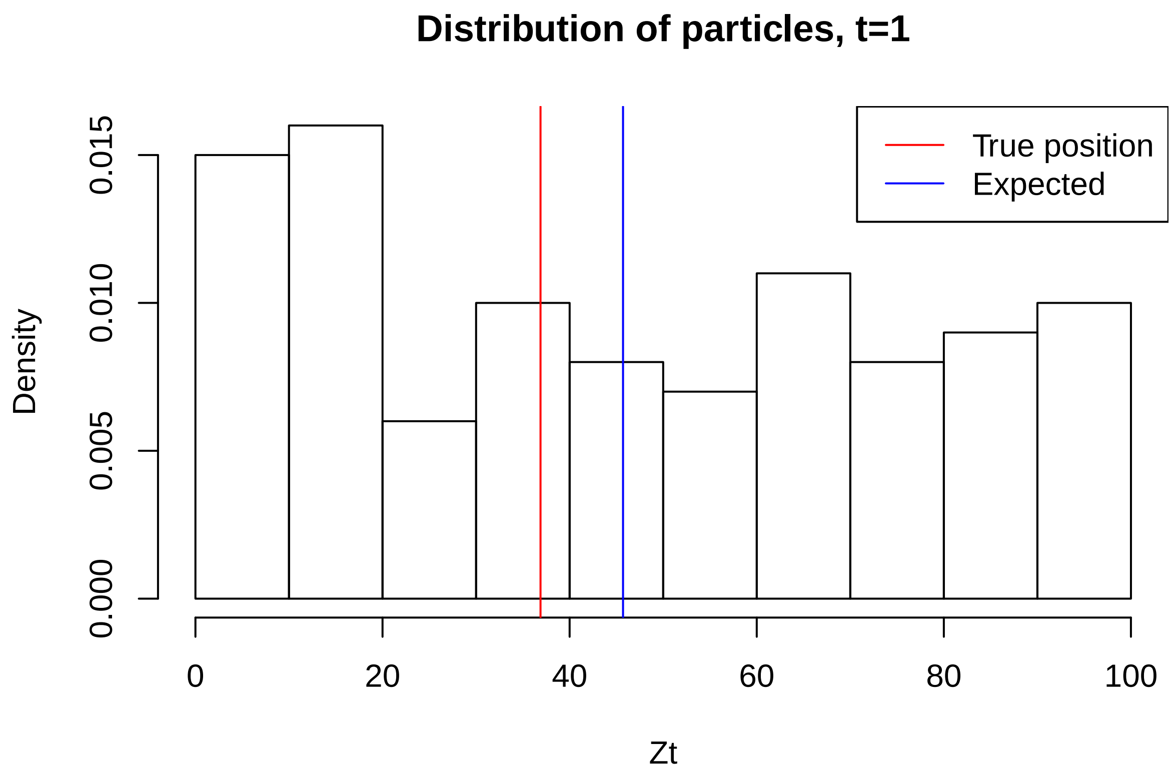
The histogram plots above show the distribution of the particles when standard deviation is 5 (for time steps 1, 35, 75, and 100). This time the algorithm still converged, but it possibly performed worse. Even though the spread of the distribution is quite narrow at the time step 100, the prediction of the robot's location is further from the true location. However, the overall performance cannot be deduced by looking at only 4 time points: each new step introduces some new uncertainty, and the results at it might be worse then the results from the previous time step. In a comparison section I use some more plots to draw better conclusions.



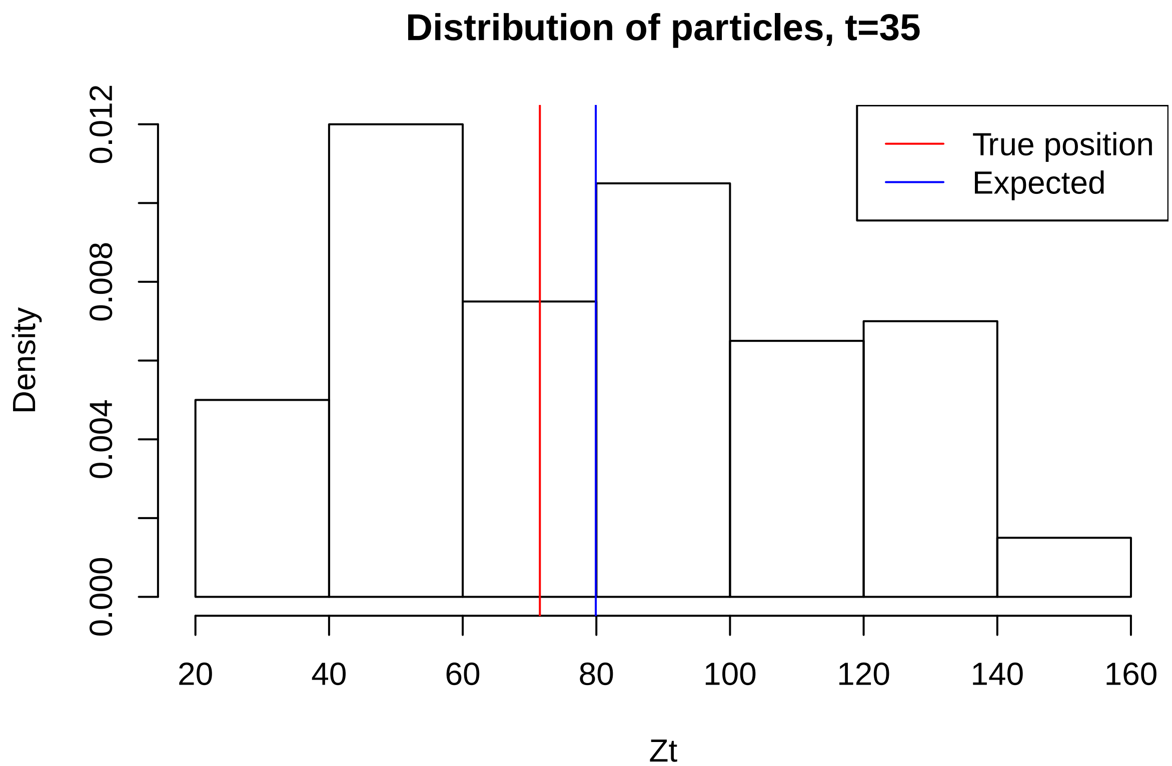


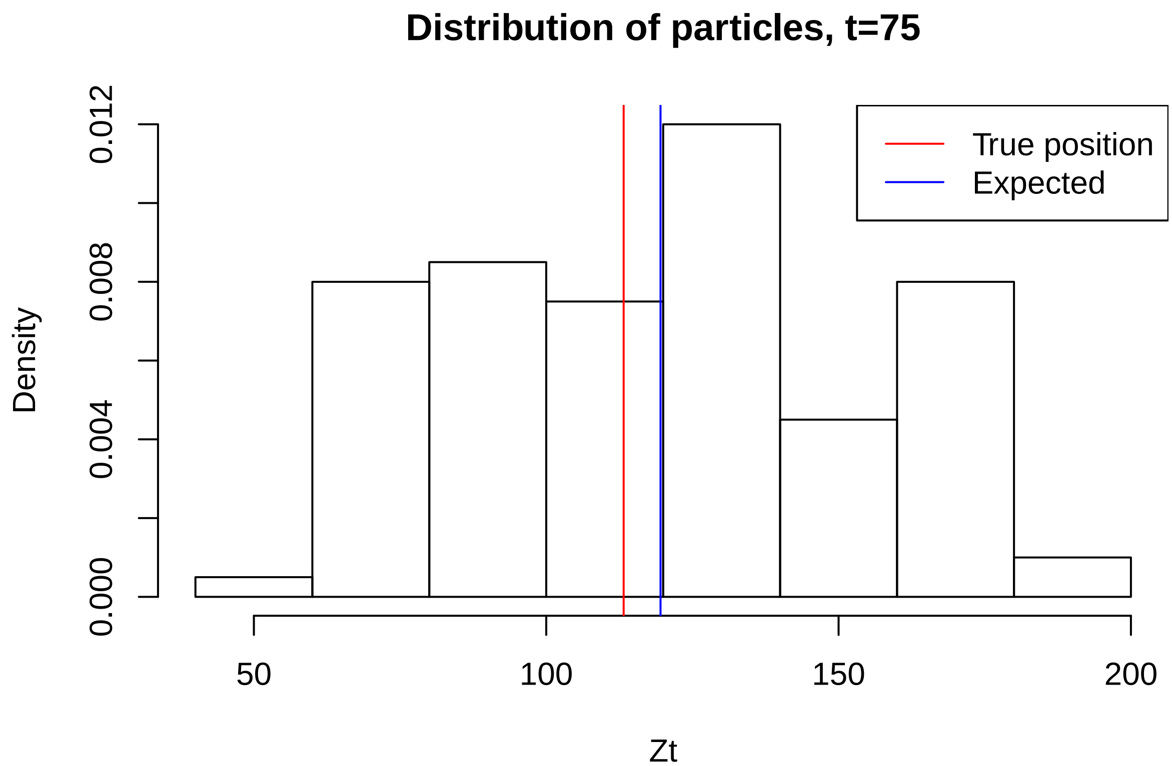
The histogram plots above show the distribution of the particles when standard deviation is 50 (for time steps 1, 35, 75, and 100). This time the spread of the particle distribution is much wider than in the previous cases. The increased standard deviation in the emission model adds more noise to the data, hence the algortihm is less sure about the actual location of the robot. However, the algorithm still converged and predicted reasonable values for the location of the robot.

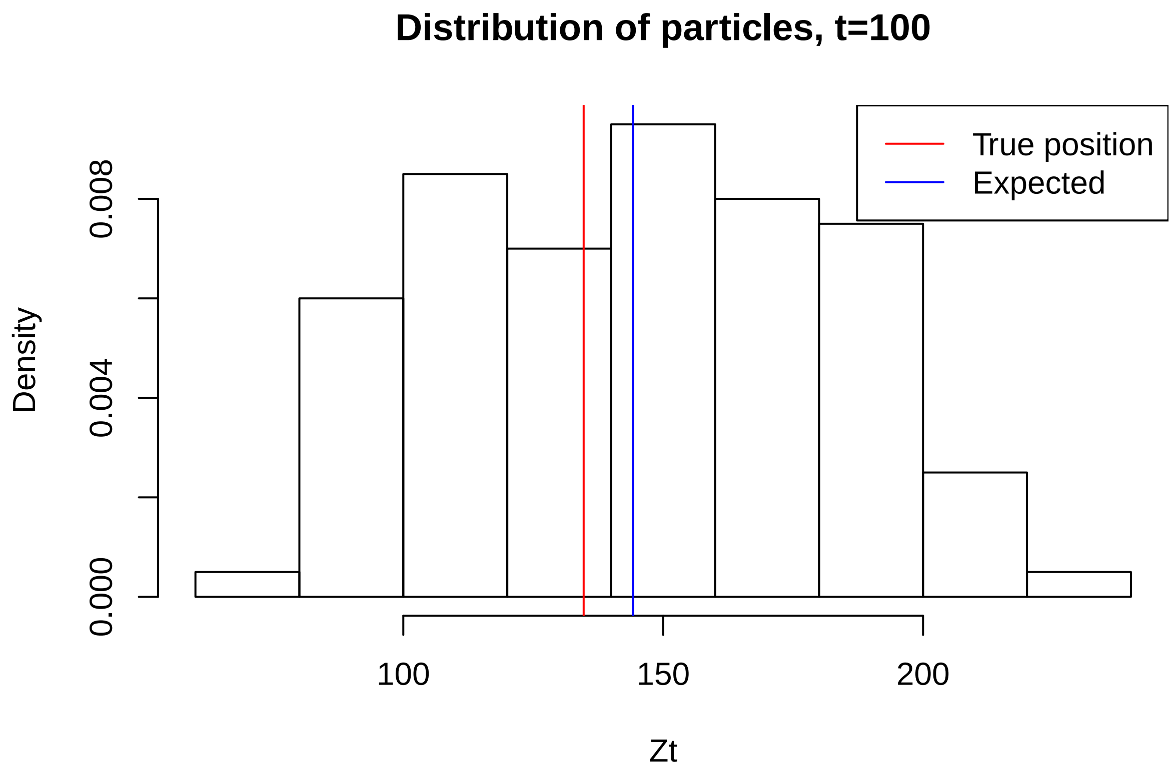
Question 3



Distribution of particles, t=1
True position
Expected
Density
0.000
0.005
0.010
0.015
0
20
40
60
80
100
Zt


Distribution of particles, t=35
True position
Expected
Density
0.000
0.004
0.008
0.012
20
40
60
80
100
120
140
160
Zt

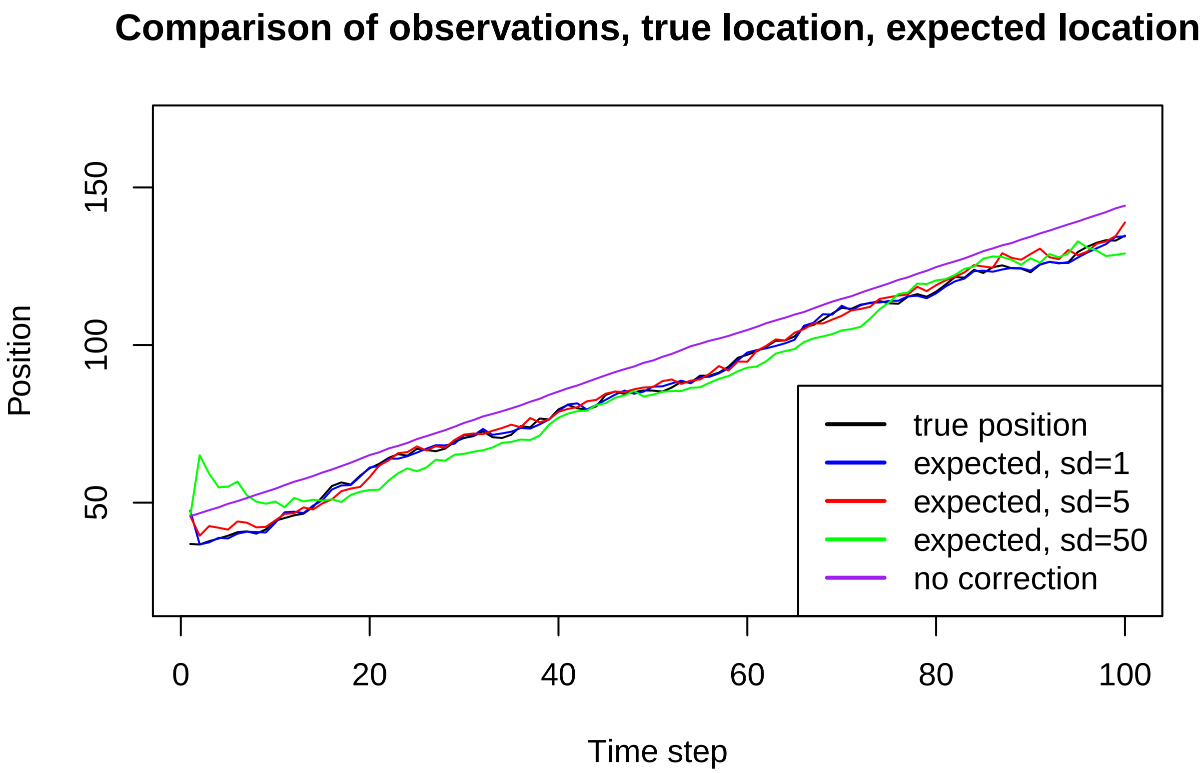

Distribution of particles, t=75
True position
Expected
Density
0.000
0.004
0.008
0.012
50
100
150
200
Zt



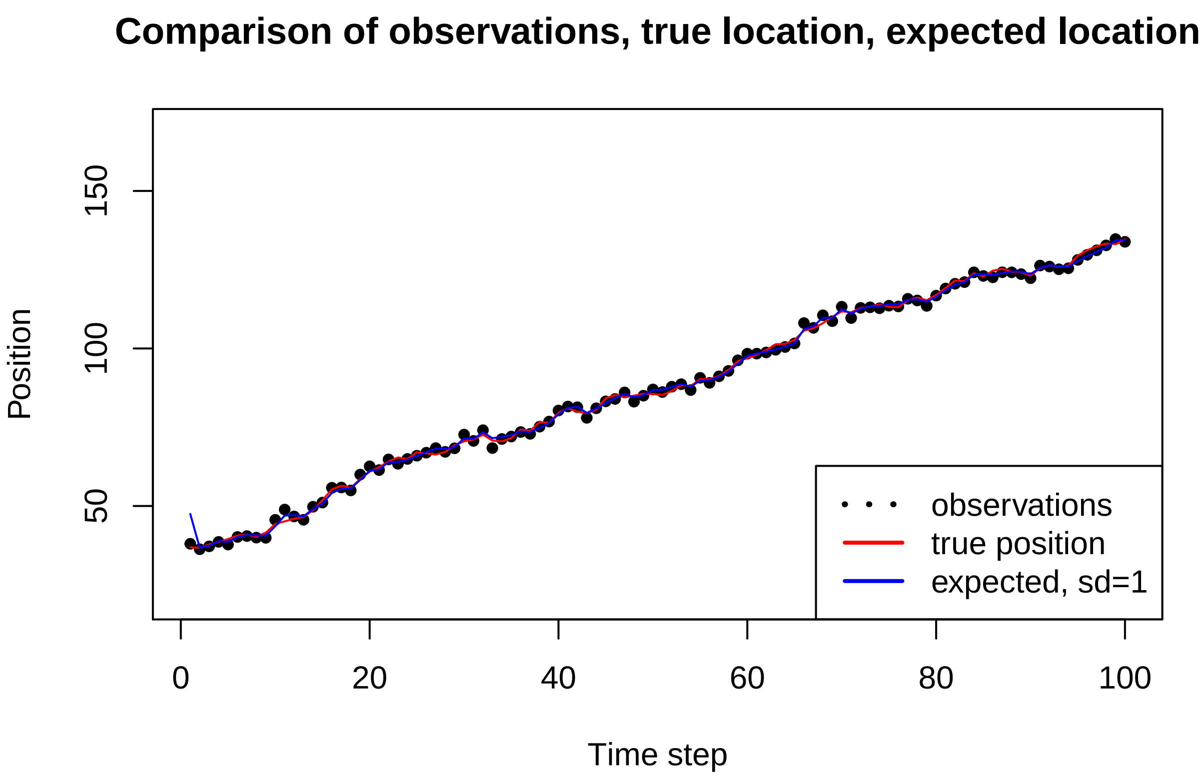
When correction step is missed, the algortihm prediction is no longer based on the observations of the sensors. This means that the algorithm just uses the transition model to predict what is the likely outcomes after the initial z_t values. Therefore, the spread of the distribution is very wide: this indicates that a lot of uncertainty is involved.

Comparison of all methods

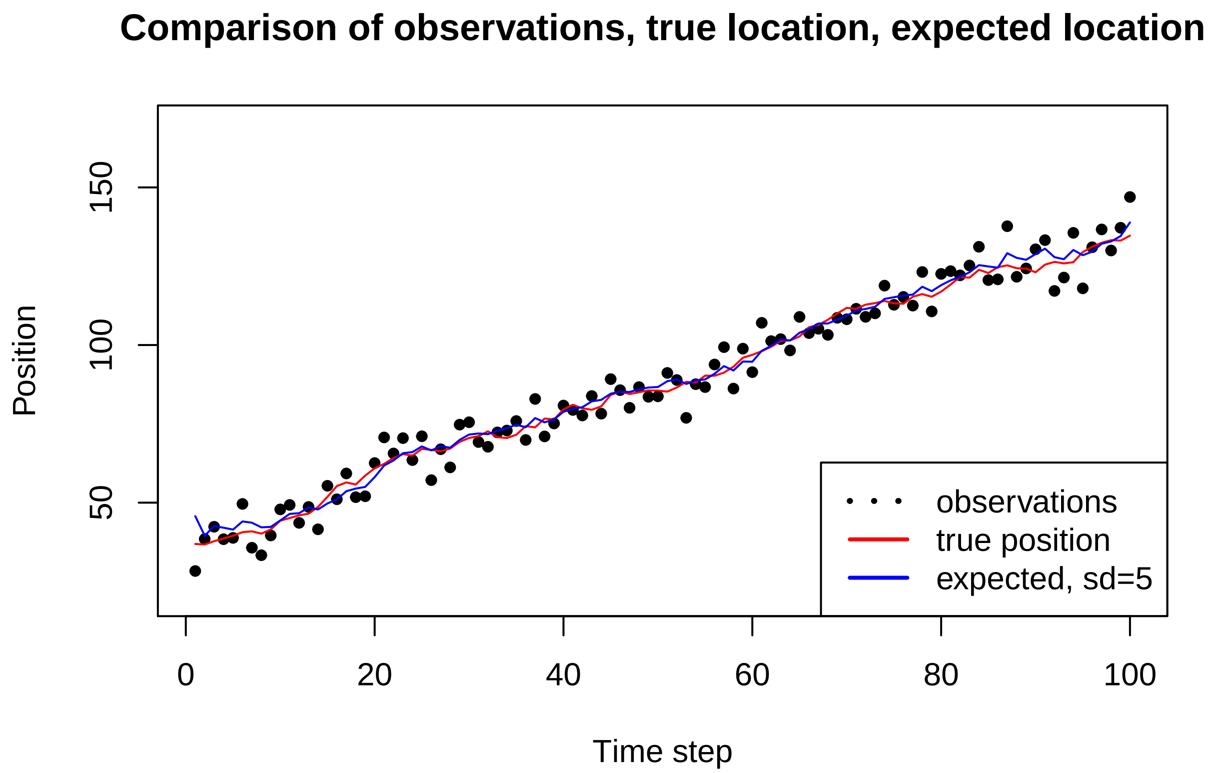
I have taken the mean value of particles for each time step and plotted the true value of z_t with likely predictions from the different models. Also the plots of observations, the true value of z_t and the predictions from each model is plotted separately.

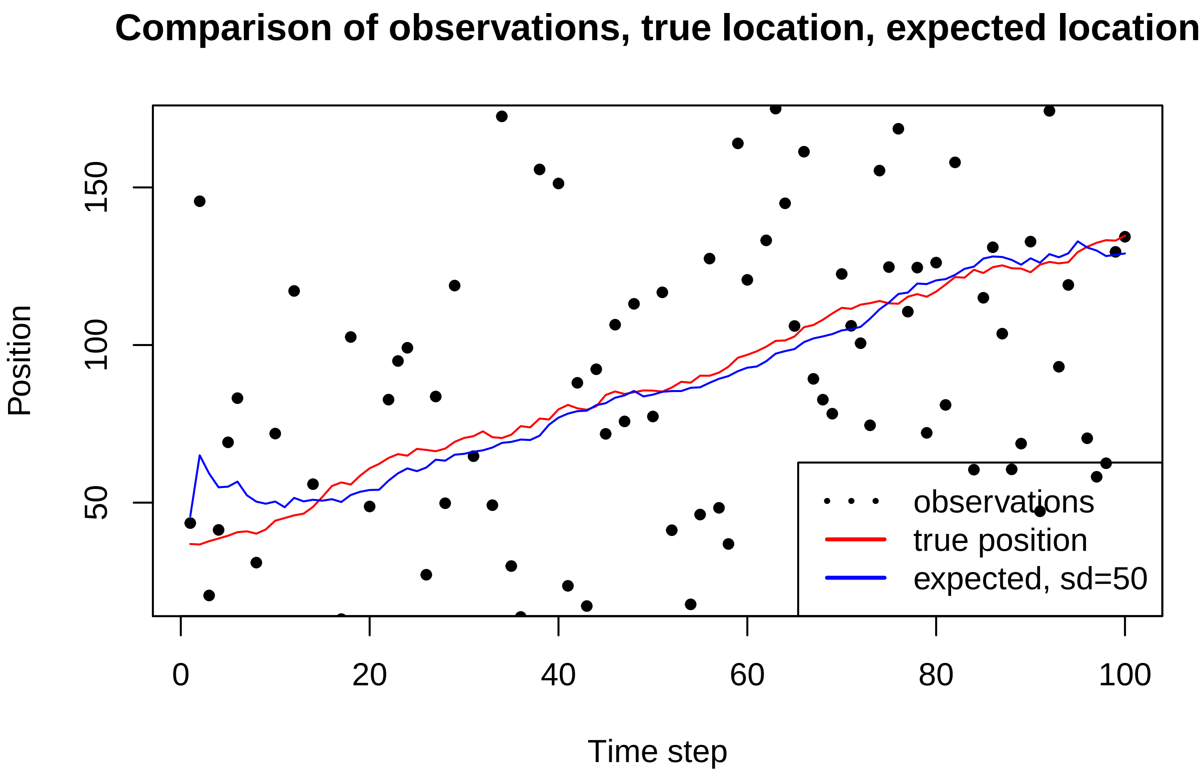


Comparison of observations, true location, expected location
Position
Time step
true position
expected, sd=1
expected, sd=5
expected, sd=50
no correction

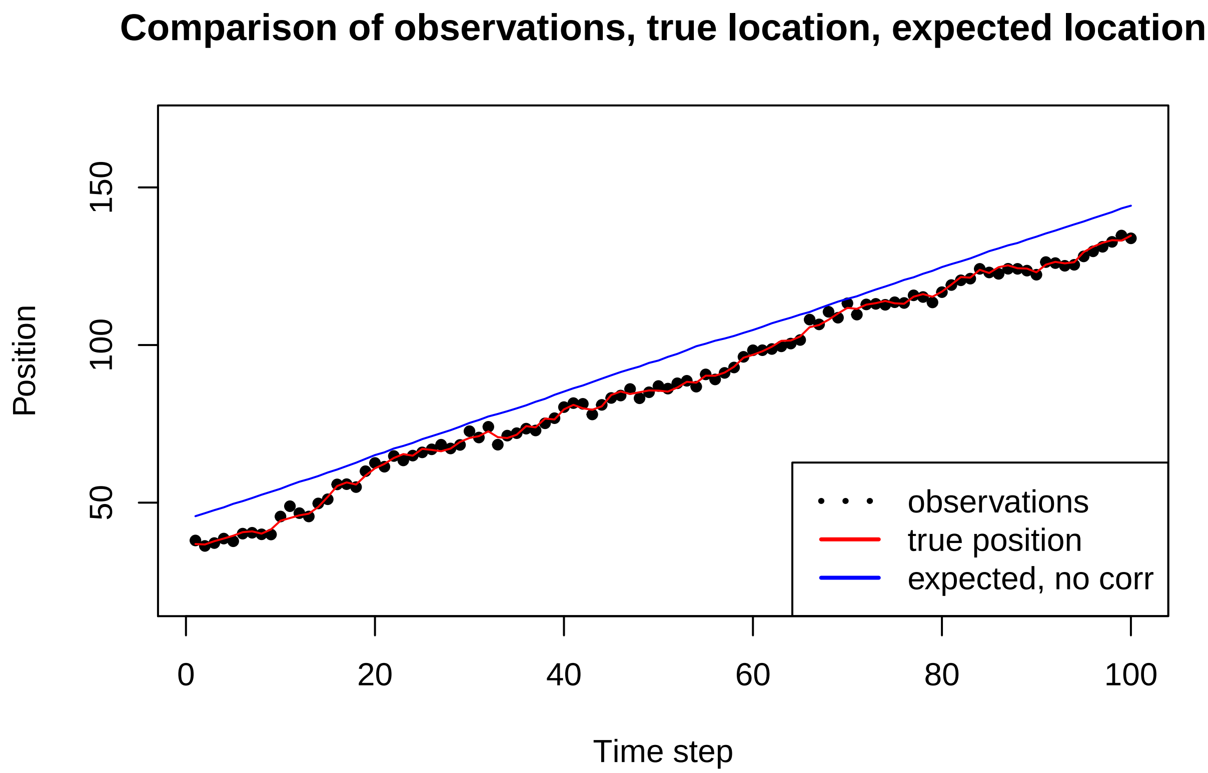


Comparison of observations, true location, expected location
Position
Time step
observations
true position
expected, sd=1


Comparison of observations, true location, expected location
Position
150
100
50
0
20
40
60
80
100
Time step
observations
true position
expected, sd=5



Comparison of observations, true location, expected location
Position
150
100
50
0
20
40
60
80
100
Time step
observations
true position
expected, sd=50



All algorithms exept the one without the correction, converged to the true value of the z_t. As expected, the lower the standard deviation is in the emission model, the faster the particle filter converges. The performance of the particle filter decreased as the standard deviation in the emission model was increased. This is logical, as the more observations deviate from the true location, the more difficult it is to predict correctly what was the true location.

When the correction step is skipped in the particle filter, it seems that the prediction is highly influenced by the random choice of the first state value, and then the consequent time steps is just a gradual increase (in most cases) from that. It does not update the predicted z_t values by how likely they are for each observation x_t. Hence it does include the information about the sensor readings, and just models the possible outcomes from the initial z_t and the transition model application multiple times.

Appendix

```
knitr::opts_chunk$set(echo = TRUE, tidy.opts = list(width.cutoff = 60),
    tidy = TRUE)
set.seed(987654321)
# M - number of particals Iter - number of time steps T z is
# latent variable x is the observation

# Function for sampling from a mixture model
sampleMix <- function(mean1, mean2, mean3, sdX) {
    u <- sample(1:3, 1)
    means <- c(mean1, mean2, mean3)
```

```r
    return(rnorm(1, mean = means[u], sd = sdX))
}

## Function that creates the SSM model
create_SSM <- function(z0, TT, sdX = 1) {
    Zt <- c(z0)

    Xt <- sampleMix(mean1 = z0, mean2 = (z0 - 1), mean3 = (z0 +
        1), sd = 1)

    for (i in 2:TT) {
        Zt[i] <- sampleMix(mean1 = Zt[i - 1], mean2 = (Zt[i -
            1] + 1), mean3 = (Zt[i - 1] + 2), sdX = 1)
        Xt[i] <- sampleMix(mean1 = Zt[i], mean2 = (Zt[i] - 1),
            mean3 = (Zt[i] + 1), sdX = sdX)
    }
    return(list(sates = Zt, observations = Xt))

}

generateX <- function(Zt, TT, sdX = 1) {
    Xt <- c()
    for (i in 1:TT) {
        Xt[i] <- sampleMix(mean1 = Zt[i], mean2 = (Zt[i] - 1),
            mean3 = (Zt[i] + 1), sdX = sdX)
    }
    return(Xt)
}

## Particle filter function
particles <- function(Obs, M, Iter, sdX = 1, correction = TRUE) {
    Xt <- Obs

    # Initialise particles for the latent variable
    Zt <- matrix(ncol = Iter, nrow = M)
    wt <- matrix(ncol = Iter, nrow = M)
    Zt_bar <- matrix(ncol = Iter, nrow = M)
    # fill the initial time step
    Zt[, 1] <- runif(M, 0, 100)
    # Add arbitrary values in order for the indexing to work
    # later
    Zt_bar[, 1] <- rep.int(1, M)
    wt[, 1] <- rep.int(1, M)

    for (t in 2:Iter) {
        # Prediction
        for (m in 1:M) {
            # sample from the transition model (current Z_t depends on
            # the Z_{t-1}) Zt_bar[m,t] <-
            # (rnorm(1,mean=Zt[m,t-1],sd=1)+rnorm(1,mean=(Zt[m,t-1]+1),sd=1)+rnorm(1,mean=(Zt[m,t-1]+2)
            Zt_bar[m, t] <- sampleMix(mean1 = Zt[m, t - 1], mean2 = (Zt[m,
                t - 1] + 1), mean3 = (Zt[m, t - 1] + 2), sdX = 1)
            # Calculate weights
```

```
            wt[m, t] <- (dnorm(Xt[t], mean = Zt_bar[m, t], sd = sdX) +
                dnorm(Xt[t], mean = (Zt_bar[m, t] - 1), sd = sdX) +
                dnorm(Xt[t], mean = (Zt_bar[m, t] + 1), sd = sdX))/3
        }
        # Correction
        if (correction) {
            Zt[, t] <- sample(Zt_bar[, t], M, replace = TRUE,
                prob = wt[, t])
        } else {
            Zt[, t] <- Zt_bar[, t]
        }

    }
    return(list(Zt = Zt, Zt_bar = Zt_bar, wt = wt))

}

# Create the model
my_SSM <- create_SSM(runif(1, 0, 100), 100)

# Filter particles
results_q1 <- particles(my_SSM$observations, 100, 100)
par(mfrow = c(1, 2), oma = c(1, 1, 2, 1))
hist(results_q1$Zt[, 1], freq = FALSE, xlab = "Zt", main = "t=1")
abline(v = my_SSM$sates[1], col = "red")
abline(v = mean(results_q1$Zt[, 1]), col = "blue")

hist(results_q1$Zt[, 35], freq = FALSE, xlab = "Zt", main = "t=35")
abline(v = my_SSM$sates[35], col = "red")
abline(v = mean(results_q1$Zt[, 35]), col = "blue")

mtext("Distribution of particles", side = 3, line = 0, outer = TRUE,
    font = 2, cex = 1.3)
mtext("sd=1", side = 3, line = -1, outer = TRUE, font = 2, cex = 1.1)

hist(results_q1$Zt[, 75], freq = FALSE, xlab = "Zt", main = "t=75")
abline(v = my_SSM$sates[75], col = "red")
abline(v = mean(results_q1$Zt[, 75]), col = "blue")

hist(results_q1$Zt[, 100], freq = FALSE, xlab = "Zt", main = "t=100")
abline(v = my_SSM$sates[100], col = "red")
abline(v = mean(results_q1$Zt[, 100]), col = "blue")

par(fig = c(0, 1, 0, 1), oma = c(0, 0, 0, 0), mar = c(0, 0, 0,
    0), new = TRUE)
plot(0, 0, type = "n", bty = "n", xaxt = "n", yaxt = "n")
legend("top", c("true location", "expected location"), col = c("red",
    "blue"), lty = c(1, 1), lwd = c(2, 2), xpd = TRUE)
# Create observations for the model with sd=5
my_SSM_2a <- list()
my_SSM_2a$sates <- my_SSM$sates
my_SSM_2a$observations <- generateX(my_SSM$sates, 100, sdX = 5)
# Filter particles, sd=5
```

```r
results_q2a <- particles(my_SSM_2a$observations, 100, 100, sdX = 5)

# Create the model, sd=50
my_SSM_2b <- list()
my_SSM_2b$sates <- my_SSM$sates
my_SSM_2b$observations <- generateX(my_SSM$sates, 100, sdX = 50)
# Filter particles, sd=50
results_q2b <- particles(my_SSM_2b$observations, 100, 100, sdX = 50)

# Plot results from 2a, sd =5
par(mfrow = c(1, 2), oma = c(1, 1, 2, 1))
hist(results_q2a$Zt[, 1], freq = FALSE, xlab = "Zt", main = "t=1",
    xlim = c(min(min(results_q2a$Zt[, 1], my_SSM_2a$sates[1])) -
        2, max(max(results_q2a$Zt[, 1]), my_SSM_2a$sates[1]) +
        2))
abline(v = my_SSM_2a$sates[1], col = "red")
abline(v = mean(results_q2a$Zt[, 1]), col = "blue")

hist(results_q2a$Zt[, 35], freq = FALSE, xlab = "Zt", main = "t=35",
    xlim = c(min(min(results_q2a$Zt[, 35], my_SSM_2a$sates[35])) -
        2, max(max(results_q2a$Zt[, 35]), my_SSM_2a$sates[35]) +
        2))
abline(v = my_SSM_2a$sates[35], col = "red")
abline(v = mean(results_q2a$Zt[, 35]), col = "blue")

mtext("Distribution of particles", side = 3, line = 0, outer = TRUE,
    font = 2, cex = 1.3)
mtext("sd=5", side = 3, line = -1, outer = TRUE, font = 2, cex = 1.1)

hist(results_q2a$Zt[, 75], freq = FALSE, xlab = "Zt", main = "t=75",
    xlim = c(min(min(results_q2a$Zt[, 75], my_SSM_2a$sates[75])) -
        2, max(max(results_q2a$Zt[, 75]), my_SSM_2a$sates[75]) +
        2))
abline(v = my_SSM_2a$sates[75], col = "red")
abline(v = mean(results_q2a$Zt[, 75]), col = "blue")

hist(results_q2a$Zt[, 100], freq = FALSE, xlab = "Zt", main = "t=100",
    xlim = c(min(min(results_q2a$Zt[, 100], my_SSM_2a$sates[100])) -
        2, max(max(results_q2a$Zt[, 100]), my_SSM_2a$sates[100]) +
        2))
abline(v = my_SSM_2a$sates[100], col = "red")
abline(v = mean(results_q2a$Zt[, 100]), col = "blue")

par(fig = c(0, 1, 0, 1), oma = c(0, 0, 0, 0), mar = c(0, 0, 0,
    0), new = TRUE)
plot(0, 0, type = "n", bty = "n", xaxt = "n", yaxt = "n")
legend("top", c("true location", "expected location"), col = c("red",
    "blue"), lty = c(1, 1), lwd = c(2, 2), xpd = TRUE)

# Plot results from 2b, sd =50
par(mfrow = c(1, 2), oma = c(1, 1, 2, 1))
hist(results_q2b$Zt[, 1], freq = FALSE, xlab = "Zt", main = "t=1",
    xlim = c(min(min(results_q2b$Zt[, 1], my_SSM_2b$sates[1])) -
```

```
        2, max(max(results_q2b$Zt[, 1]), my_SSM_2b$sates[1]) +
        2))
abline(v = my_SSM_2b$sates[1], col = "red")
abline(v = mean(results_q2b$Zt[, 1]), col = "blue")

hist(results_q2b$Zt[, 35], freq = FALSE, xlab = "Zt", main = "t=35",
    xlim = c(min(min(results_q2b$Zt[, 35], my_SSM_2b$sates[35])) -
        2, max(max(results_q2b$Zt[, 35]), my_SSM_2b$sates[35]) +
        2))
abline(v = my_SSM_2b$sates[35], col = "red")
abline(v = mean(results_q2b$Zt[, 35]), col = "blue")

mtext("Distribution of particles", side = 3, line = 0, outer = TRUE,
    font = 2, cex = 1.3)
mtext("sd=50", side = 3, line = -1, outer = TRUE, font = 2, cex = 1.1)

hist(results_q2b$Zt[, 75], freq = FALSE, xlab = "Zt", main = "t=75",
    xlim = c(min(min(results_q2b$Zt[, 75], my_SSM_2b$sates[75])) -
        2, max(max(results_q2b$Zt[, 75]), my_SSM_2b$sates[75]) +
        2))
abline(v = my_SSM_2b$sates[75], col = "red")
abline(v = mean(results_q2b$Zt[, 75]), col = "blue")

hist(results_q2b$Zt[, 100], freq = FALSE, xlab = "Zt", main = "t=100",
    xlim = c(min(min(results_q2b$Zt[, 100], my_SSM_2b$sates[100])) -
        2, max(max(results_q2b$Zt[, 100]), my_SSM_2b$sates[100]) +
        2))
abline(v = my_SSM_2b$sates[100], col = "red")
abline(v = mean(results_q2b$Zt[, 100]), col = "blue")

par(fig = c(0, 1, 0, 1), oma = c(0, 0, 0, 0), mar = c(0, 0, 0,
    0), new = TRUE)
plot(0, 0, type = "n", bty = "n", xaxt = "n", yaxt = "n")
legend("top", c("true position", "expected location"), col = c("red",
    "blue"), lty = c(1, 1), lwd = c(2, 2), xpd = TRUE)

# Filter particles
results_q3 <- particles(my_SSM$observations, 100, 100, correction = FALSE)


hist(results_q3$Zt[, 1], freq = FALSE, xlab = "Zt", main = "Distribution of particles, t=1")
abline(v = my_SSM$sates[1], col = "red")
abline(v = mean(results_q3$Zt[, 1]), col = "blue")
legend("topright", c("True position", "Expected"), col = c("red",
    "blue"), lty = c(1, 1))


hist(results_q3$Zt[, 35], freq = FALSE, xlab = "Zt", main = "Distribution of particles, t=35")
abline(v = my_SSM$sates[35], col = "red")
abline(v = mean(results_q3$Zt[, 35]), col = "blue")
legend("topright", c("True position", "Expected"), col = c("red",
    "blue"), lty = c(1, 1))
```

```r
hist(results_q3$Zt[, 75], freq = FALSE, xlab = "Zt", main = "Distribution of particles, t=75")
abline(v = my_SSM$sates[75], col = "red")
abline(v = mean(results_q3$Zt[, 75]), col = "blue")
legend("topright", c("True position", "Expected"), col = c("red",
    "blue"), lty = c(1, 1))



hist(results_q3$Zt[, 100], freq = FALSE, xlab = "Zt", main = "Distribution of particles, t=100")
abline(v = my_SSM$sates[100], col = "red")
abline(v = mean(results_q3$Zt[, 100]), col = "blue")
legend("topright", c("True position", "Expected"), col = c("red",
    "blue"), lty = c(1, 1))


exp_loc_q1 <- colMeans(results_q1$Zt)
exp_loc_q2a <- colMeans(results_q2a$Zt)
exp_loc_q2b <- colMeans(results_q2b$Zt)
exp_loc_q3 <- colMeans(results_q3$Zt)

plot(my_SSM$sates, col = "black", type = "l", ylab = "Position",
    xlab = "Time step", main = "Comparison of observations, true location, expected location",
    ylim = c(20, 170))
lines(exp_loc_q1, col = "blue")
lines(exp_loc_q2a, col = "red")
lines(exp_loc_q2b, col = "green")
lines(exp_loc_q3, col = "purple")
legend("bottomright", c("true position", "expected, sd=1", "expected, sd=5",
    "expected, sd=50", "no correction"), col = c("black", "blue",
    "red", "green", "purple"), lty = c(1, 1, 1, 1, 1), lwd = c(rep(2,
    5)))


plot(my_SSM$observations, pch = 20, ylab = "Position", xlab = "Time step",
    main = "Comparison of observations, true location, expected location",
    ylim = c(20, 170))
lines(my_SSM$sates, col = "red")
lines(exp_loc_q1, col = "blue")
legend("bottomright", c("observations", "true position", "expected, sd=1"),
    col = c("black", "red", "blue"), lty = c(3, 1, 1), lwd = c(3,
        rep(2, 2)))


plot(my_SSM_2a$observations, pch = 20, ylab = "Position", xlab = "Time step",
    main = "Comparison of observations, true location, expected location",
    ylim = c(20, 170))
lines(my_SSM$sates, col = "red")
lines(exp_loc_q2a, col = "blue")
legend("bottomright", c("observations", "true position", "expected, sd=5"),
    col = c("black", "red", "blue"), lty = c(3, 1, 1), lwd = c(3,
        rep(2, 2)))
```

```r
plot(my_SSM_2b$observations, pch = 20, ylab = "Position", xlab = "Time step",
    main = "Comparison of observations, true location, expected location",
    ylim = c(20, 170))
lines(my_SSM$sates, col = "red")
lines(exp_loc_q2b, col = "blue")
legend("bottomright", c("observations", "true position", "expected, sd=50"),
    col = c("black", "red", "blue"), lty = c(3, 1, 1), lwd = c(3,
        rep(2, 2)))

plot(my_SSM$observations, pch = 20, ylab = "Position", xlab = "Time step",
    main = "Comparison of observations, true location, expected location",
    ylim = c(20, 170))
lines(my_SSM$sates, col = "red")
lines(exp_loc_q3, col = "blue")
legend("bottomright", c("observations", "true position", "expected, no corr"),
    col = c("black", "red", "blue"), lty = c(3, 1, 1), lwd = c(3,
        rep(2, 2)))
```