Lab 3

Kristian Sikiric (krisi211)

The lab

The purpose of the lab is to put in practice some of the concepts covered in the lectures. To do so, you are asked to implement the particle filter for robot localization. The robot moves along the horizontal axis according to the following SSM:

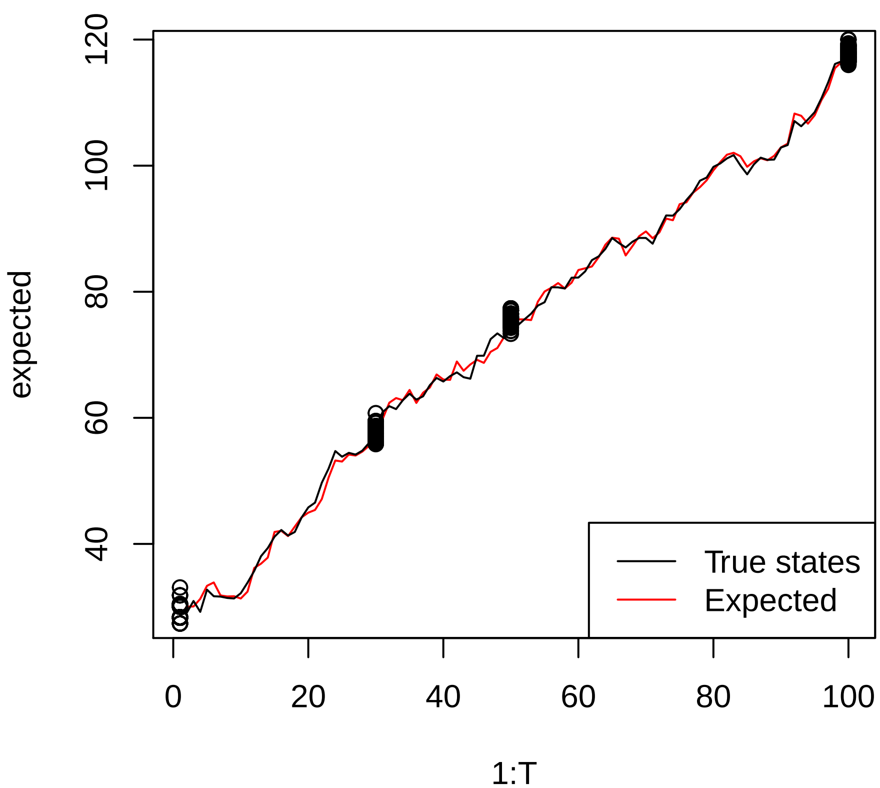
Transition model: $p(z_t|z_{t1}) = (N(z_t|z_{t-1}, 1) + N(z_t|z_{t-1} + 1, 1) + N(z_t|z_{t-1} + 2, 1))/3$

Emission model: $p(x_t|z_t) = (N(x_t|z_t, 1) + N(x_t|z_t - 1, 1) + N(x_t|z_t + 1, 1))/3$

Initial model: $p(z_1) = Uniform(0, 100)$

Assignment 1

The first assignment was to implement the SSM described above and simulate it for 100 time steps to obtain states and observations. Then the observations were to be used to identify the states via particle filtering. Below a graph with the expected states, true states and particles for time steps 1, 100, 30 and 50 are shown (represented as dots in the graph).

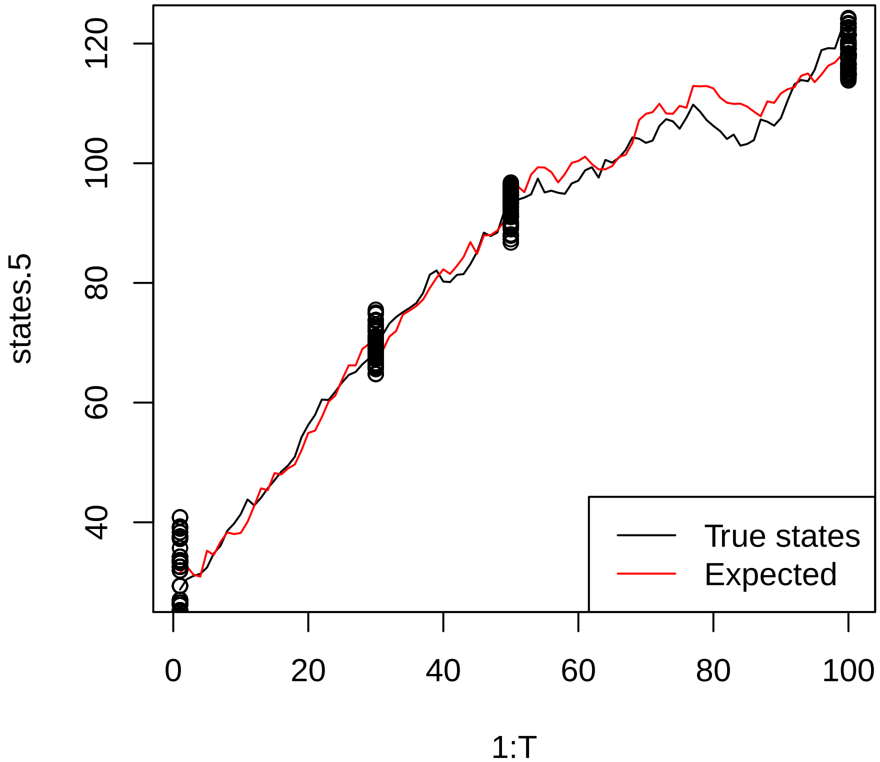



```
## [1] "MSE:  0.929973385302431"
```

Assignment 2

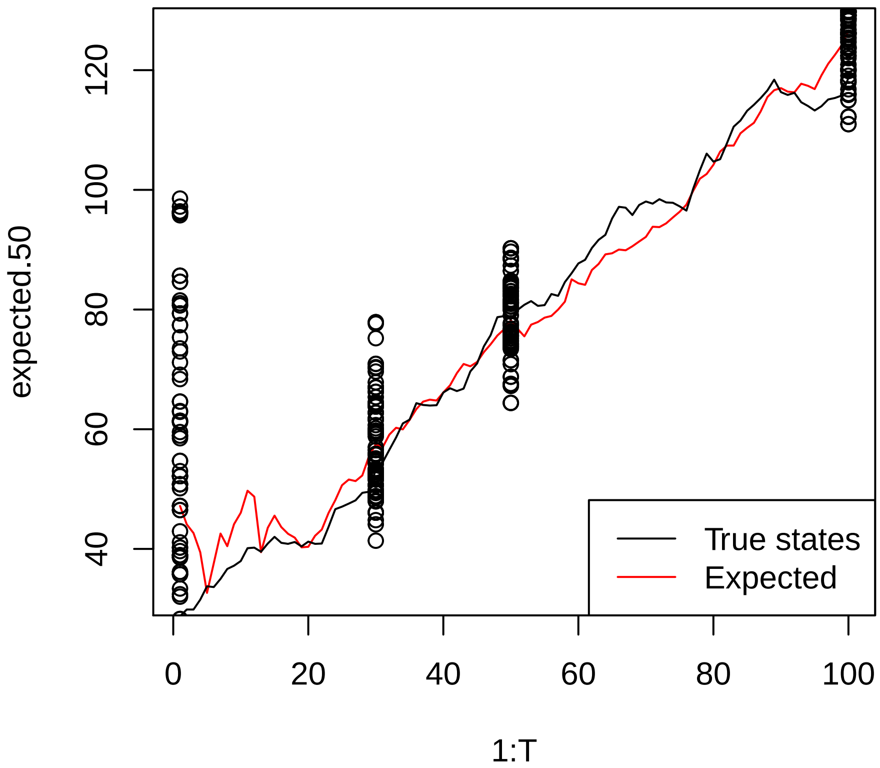
Now the above assignment was to be redone with different standard deviations for the emission model. First the standard deviation was increased to 5, then to 50. By doing this, we introduce more uncertenty and noise, which makes it harder to identify the correct states. This is shown below.

Standard deviation = 5.




```
## [1] "MSE:  7.61674968437069"
```

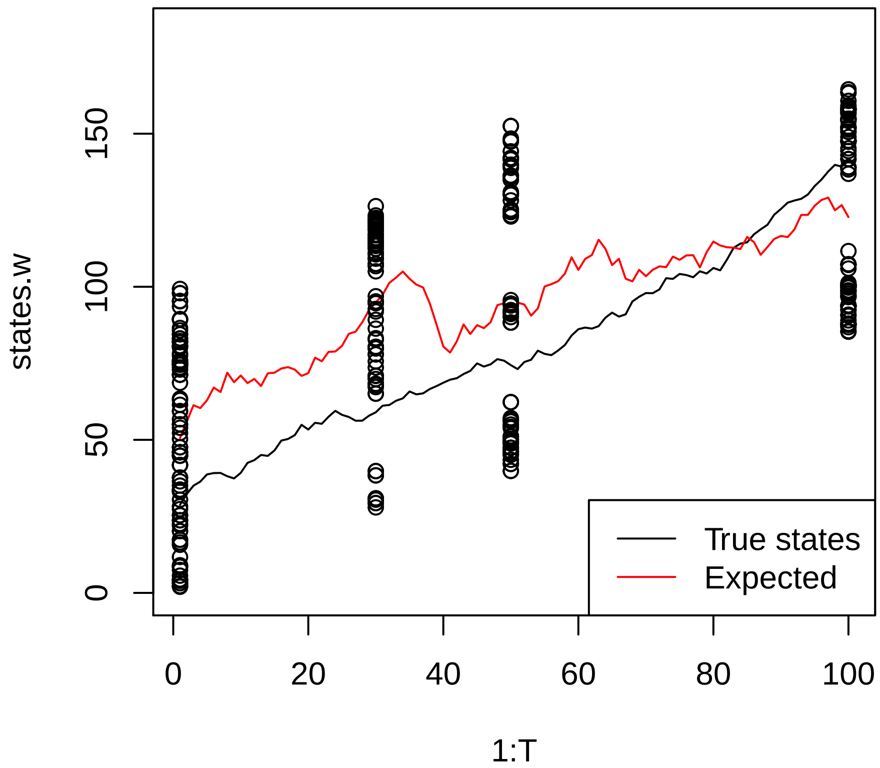
Standrd deviation = 50.




```
## [1] "MSE:  21.3881765525824"
```

Assignment 3

In this assignment, we were to show and explain what happens when the weights are constant and set to 1. This means that we see all particles as equally probable. This means that even bad particles will be taken into account even if they are not part of the true distribution.




```
## [1] "MSE:  432.717864465798"
```

Appendix - Code

```
####### Innit ######
set.seed(123)
z1 = runif(1,0,100)
T = 100
M = 100
sd_emission = 1
####### Functions ######
transition = function(z,sd = 1){
  comp = sample(0:2,1)
  return (rnorm(1,z+comp,sd))
}

emission = function(z,sd = sd_emission){
  comp = sample(-1:1,1)
  return (rnorm(1,z+comp,sd))
}

density.emission = function(z,x,sd = sd_emission){
  p = dnorm(x,z,sd) + dnorm(x,z-1,sd) + dnorm(x,z+1,sd)
  return(mean(p))
}

particle_filter = function(weight = TRUE, obs){
```

```r
  w = rep(1,100)
  particles = runif(M,0,100)
  expected = rep(0,100)
  X = matrix(NA,nrow = T, ncol = M)
  for (t in 1:T){
    if(weight){
      for (m in 1:M){
        w[m] = density.emission(z = particles[m],x = obs[t])
      }
    }
    particles = sample(particles, size = 100, replace = TRUE, prob = w)
    X[t,] = particles

    w = w / sum(w)
    tmp = NULL
    for(m in 1:M){
      tmp[m] = w[m]*particles[m]
    }

    expected[t]= sum(tmp)
    for(m in 1:M){
      particles[m] = transition(particles[m])
    }
  }
  return (list("Expected" = expected,"Particles" = X))
}

SSM = function(){
  states = rep(0,T)
  observations = rep(0,T)
  states[1] = z1

  observations[1] = emission(states[1])
  for (t in 2:T){
    states[t] = transition(states[t-1])
    observations[t] = emission(states[t])
  }
  return (list("States" = states, "Observations" = observations))
}

####### Assignment 1 ######
#SSM
ssm = SSM()
states = ssm$States
observations = ssm$Observations

kalman = particle_filter(obs = observations)
expected = kalman$Expected
X = kalman$Particles

plot(1:T,expected,type = "l", col = "red")
lines(1:T,states, col = "black")
points(rep(1,100), X[1,],pch = 1)
```

```r
points(rep(30,100), X[30,],pch = 1)
points(rep(50,100), X[50,],pch = 1)
points(rep(100,100), X[100,],pch = 1)

mean((states-expected)^2)


###### Assignment 2 ######
sd_emission = 5
ssm.5 = SSM()
states.5 = ssm.5$States
observations.5 = ssm.5$Observations

kalman = particle_filter(obs = observations.5)
expected.5 = kalman$Expected
X.5 = kalman$Particles

plot(1:T,states.5, col = "black", type = "l")
lines(1:T,expected.5,col = "green")
points(rep(1,100), X.5[1,],pch = 1)
points(rep(30,100), X.5[30,],pch = 1)
points(rep(50,100), X.5[50,],pch = 1)
points(rep(100,100), X.5[100,],pch = 1)
mean((states.5-expected.5)^2)

sd_emission = 50
ssm.50 = SSM()
states.50 = ssm.50$States
observations.50 = ssm.50$Observations

kalman = particle_filter(obs = observations.50)
expected.50 = kalman$Expected
X.50 = kalman$Particles

plot(1:T,expected.50, col = "blue", type = "l")
lines(1:T,states.50,col = "black")
points(rep(1,100), X.50[1,],pch = 1)
points(rep(30,100), X.50[30,],pch = 1)
points(rep(50,100), X.50[50,],pch = 1)
points(rep(100,100), X.50[100,],pch = 1)
mean((states.50-expected.50)^2)

###### Assignment 3 ######
set.seed(123)
sd_emission = 1
ssm.w = SSM()
states.w = ssm.w$States
observations.w = ssm.w$Observations

kalman = particle_filter(weight = FALSE,obs = observations.w)
expected.w = kalman$Expected
X.w = kalman$Particles
```

```r
plot(1:T,states.w, col = "black", type = "l", ylim = c(0,max(X.w)))
lines(1:T,expected.w,col = "yellow")
points(rep(1,100), X.w[1,],pch = 1)
points(rep(30,100), X.w[30,],pch = 1)
points(rep(50,100), X.w[50,],pch = 1)
points(rep(100,100), X.w[100,],pch = 1)
mean((states.w-expected.w)^2)
```