example

Packages needed

```
#devtools::install_github("Peprah94/nimMCMCSMCupdates")
library(nimMCMCSMCupdates)
library(nimble)
library(nimbleSMC)

set.seed(1)
```

Simulating data

```
# Setting up MCMC configuration values and variation of parameters
nIterations = 1000
nBurnin = 200
nChains = 3
nThin = 1
nyears = 50
aVars <- c(0.1, 0.8) # changing the intercept
#High and small values of a
iNodePrev <- c(49, 45, 20) # The number of years for reduced model
aVarstag = 2
iNodetag = 2
mcmcRun <- FALSE #use mcmc or nimbleSMC for reduced Model
pfTypeRun = "auxiliary"

sim2 <- function(a, b, c, t, mu0){
  x <- y <- numeric(t)
  x[1] <- rnorm(1, mu0, 1 )
```

```
  y[1] <- rnorm(1, x[1], 1)

  for(k in 2:t){
    x[k] <- rnorm(1, a*x[k -1] + b, 1)
    y[k] <- rnorm(1, x[k-1]*c, 1)# + (sigOE * (sqrt(df -2)/df) * rt(1, df))
  }
  return(list(x=x, y=y))
}

message("simulating data for a = ", aVars[aVarstag])
```

```
simulating data for a = 0.8
```

```
simData <- sim2(a = aVars[aVarstag],
                b = 1,
                c = 1.5,
                t = nyears,
                mu0 = 0.2)

str(simData)
```

```
List of 2
 $ x: num [1:50] -0.426 -0.177 1.188 2.438 3.526 ...
 $ y: num [1:50] -0.243 0.956 -1.086 2.52 3.351 ...
```

Define the NIMBLE model

```
stateSpaceCode <- nimbleCode({
  x[1] ~ dnorm(mu0, 1)
  y[1] ~ dnorm(x[1], 1)
  for(i in 2:t){
    x[i] ~ dnorm(x[i-1] * a + b, 1)
    y[i] ~ dnorm(x[i] * c, 1)
  }
  a ~ dunif(0, 1)
  b ~ dnorm(0, 1)
  c ~ dnorm(1,1)
  mu0 ~ dnorm(0, 1)
```

```
})
#
# ## define data, constants, and initial values
data <- list(
  #   #y = c(0.213, 1.025, 0.314, 0.521, 0.895, 1.74, 0.078, 0.474, 0.656, 0.802)
  y = simData$y
)
constants <- list(
  t = nyears
)
inits <- list(
  a = 0.1,
  b = 0,
  mu0= 0.2,
  c = 1
)
#
#
# ## build the model
stateSpaceModel <- nimbleModel(stateSpaceCode,
                               data = data,
                               constants = constants,
                               inits = inits,
                               check = FALSE)
```