

Circular Drift Diffusion Model on JAGS: Full example

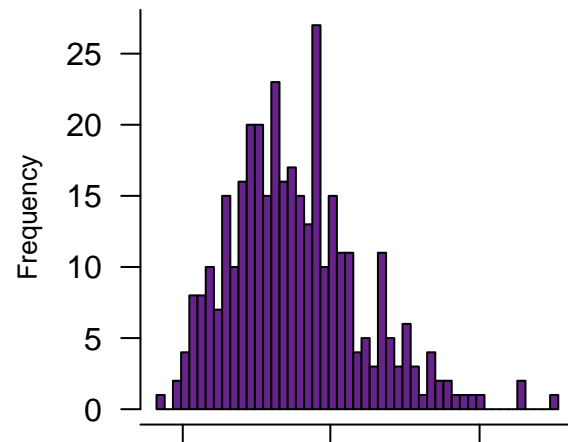
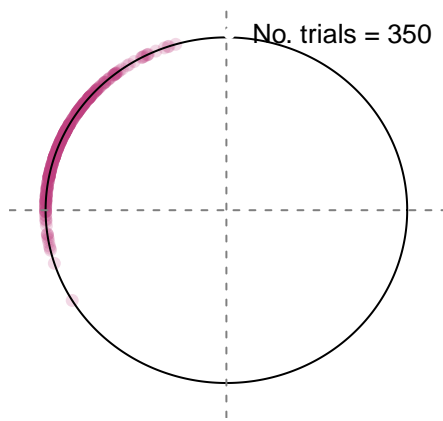
06 September, 2022

1. Generate/load simulated data

```
# Establish no. of trials
trials <- 350
# Call Rscript to generate simulated data / load it if already existing
source("./getData.R")
dim(data)
```

```
## [1] 350  2
```

```
# Plot data
cddm.plotData(data)
```



```
# Print parameter values used to generate this data
par
```

```
## true.theta0 true.drift true.bound true.ter0
##      2.685      2.870      3.990      0.230
```

2. Write JAGS model

```
modelFile <- "cddm.bug"
write('
    model{
        # Likelihood
        for (i in 1:N) {
            X[1:2,i] ~ dcddm(drift, bound, ter0, theta0)
        }

        # Priors
        drift ~ dnorm(0, 1)
        theta0 ~ dnorm(0, 1)T(-3.14, 3.14)
        bound ~ dgamma(3, 2)
        ter0 ~ dexp(1)T(, tmin)
    },
    modelFile)
```

where:

- **drift** is the magnitude of the drift vector composed by the individual drift rates related to the average motion observed across the x and y axes, according to the CDDM.
- **bound** is the threshold (i.e. the radius of the circle)
- **ter0** is the non-decision time (a.k.a. “time for encoding and response”)
- **theta0** is the direction of the drift vector, in radians.

Prepare Settings to be passed to JAGS

```
n.chains = 4
n.iter = 2500
n.burnin = 500
n.thin = 1
perParticipant = FALSE
perTask = FALSE

sampling.Settings <- list(n.chains,n.iter,n.burnin,n.thin,perParticipant,perTask)
names(sampling.Settings) <- c("n.chains","n.iter","n.burnin","n.thin","perParticipant","perTask")
```

Run JAGS model

```
source("../Functions/runCDDMjags.R")

samplesFile <- "samples.RData"

if(file.exists(samplesFile)){
    load(file=samplesFile)
}else{
    myJAGSsampling.CDDM(sampling.Settings,modelFile,samplesFile,data)}

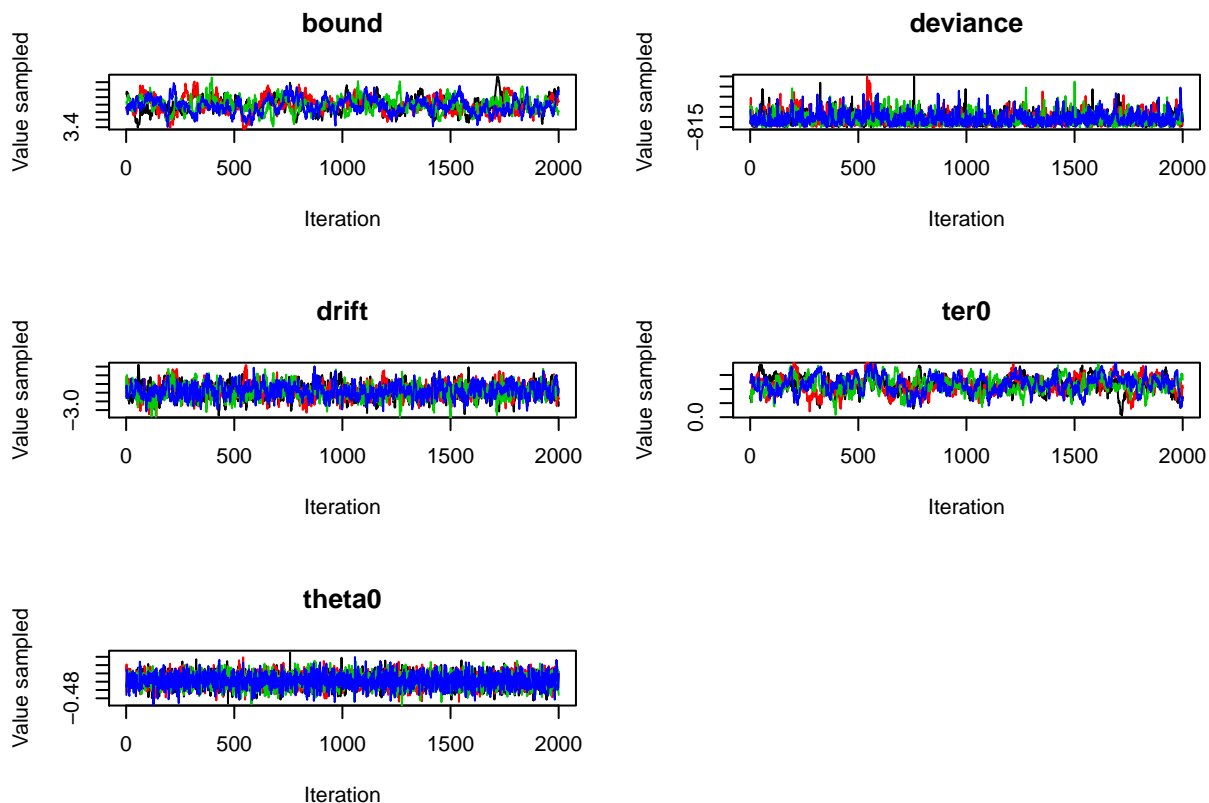
source("../Functions/processJAGSsamples.R")
drift <- myJAGSsampling.extractSamples("drift",samples)
bound <- myJAGSsampling.extractSamples("bound",samples)
ter0 <- myJAGSsampling.extractSamples("ter0",samples)
theta0 <- myJAGSsampling.extractSamples("theta0",samples)

# Get descriptive statistics for posterior samples
#####
map.theta0 <- JAGSoutput.maxDensity(theta0)
map.drift <- JAGSoutput.maxDensity(drift)
```

```
map.bound <- JAGSoutput.maxDensity(bound)
map.ter0 <- JAGSoutput.maxDensity(ter0)
MAPS <- c(map.theta0,map.drift,map.bound,map.ter0)
names(MAPS) <- c("map.theta0","map.drift","map.bound","map.ter0")

mean.theta0 <- mean(theta0)
mean.drift <- mean(drift)
mean.bound <- mean(bound)
mean.ter0 <- mean(ter0)
means <- c(mean.theta0,mean.drift,mean.bound,mean.ter0)
names(means) <- c("mean.theta0","mean.drift","mean.bound","mean.ter0")
```

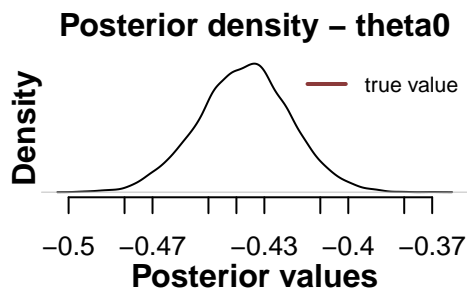
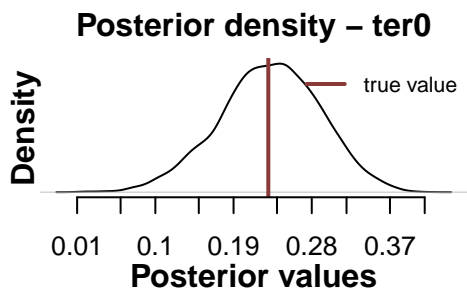
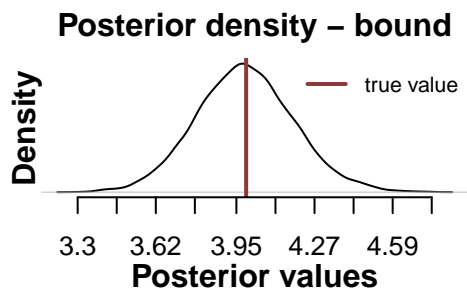
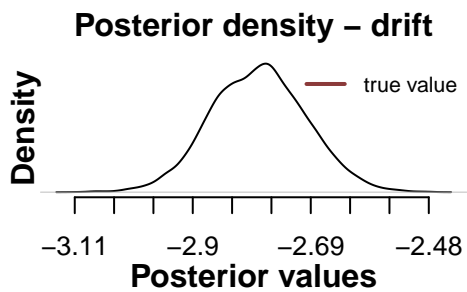
```
source("../Functions/plotJAGSsamples.R")
par(mfrow = c(3,2))
plot.ShowAllChains(samples)
```



```
source("../Functions/processJAGSsamples.R")
myJAGSsampling.Rhat.max(samples)
```

```
## [1] "The maximum value of Rhat observed was 1.0056 which corresponds to: bound"
```

```
par(mfrow = c(2,2))
plot.PosteriorDensity(drift,par["true.drift"])
plot.PosteriorDensity(bound,par["true.bound"])
plot.PosteriorDensity(ter0,par["true.ter0"])
plot.PosteriorDensity(theta0,par["true.theta0"])
```



Check against EZ-estimates

```
source("../Functions/ezcdm.R")
EZ <- ezcdm.fit(data$Choice,data$RT)

## First, compare EZ estimates against true parameter values
EZ

## EZ.theta0  EZ.drift  EZ.bound  EZ.ter0
## 2.7043808 2.8427654 4.0235061 0.2442079
par

## true.theta0  true.drift  true.bound  true.ter0
##      2.685      2.870      3.990      0.230

# Then, compare against point descriptors for posterior samples
#####
MAPS

## map.theta0  map.drift  map.bound  map.ter0
##    -0.434    -2.770     3.976     0.243
means

## mean.theta0  mean.drift  mean.bound  mean.ter0
## -0.4372454 -2.7828805  3.9850560  0.2316032
```