

Universidade Federal de Uberlândia

Lista do Módulo 4

Levy Gabriel da Silva Galvão

Uberlândia
2021

Módulo 4

Exercício 1

Faça uma coleta simultânea de dados eletromiográficos e inerciais. Você deverá seguir o seguinte protocolo experimental:

- Posicionar os sensores de eletromiografia no músculo tibial anterior e nos músculos do tríceps sural. O acelerômetro deve ser posicionado nos dois terços proximais da parte lateral da perna, com o eixo y contra a gravidade.
- Com o sujeito na posição ortostática deve-se realizar o movimento de dorsiflexão e flexão plantar. No retorno da flexão realizar um contato brusco do calcanhar com o solo.
- Realizar 60 repetições da tarefa, sem descanso.
- Salvar o arquivo de dados resultante com o nome "DadosM4-1.txt".

Resposta: como os dados já estão coletados, resta realizar o processamento.

Exercício 2

Faça a estimativa de parâmetros estatísticos dos sinais eletromiográficos disponíveis no arquivo "DadosM4-1.txt". Os seguintes passos devem ser executados:

- Faça a detecção dos picos no sinal AccY. Devem ser considerados picos aqueles valores que tenham amplitude de pelo menos 90% do valor do pico máximo do sinal.
- Gere um gráfico com os picos encontrados na letra a).
- Calcule as estatísticas abaixo para cada janela de sinal do músculo tibial anterior. O tamanho da janela deve ser de 500 ms, e o início da mesma deve ser a partir de cada um dos picos detectados na letra a). Média, Mediana, Moda, Amplitude, Variância, Coeficiente de Variação, Distância Interquartil
- Faça um gráfico que descreva a variação da Mediana em função do tempo (dica: observe como o valor rms foi calculado em função do tempo, no tutorial)

```
# step 1 - read data in .txt
```

```
library(openxlsx)
df <- read.table("Dados/DadosM4-1.txt", header = FALSE, sep = " ", dec = ".", skip = 6)
colnames(df) <- c("AccX", "AccY", "MuscAnterior", "MuscPosterior")
splRate <- 500
df$Time <- seq(0, nrow(df)/splRate-1/splRate, by=1/splRate)
```

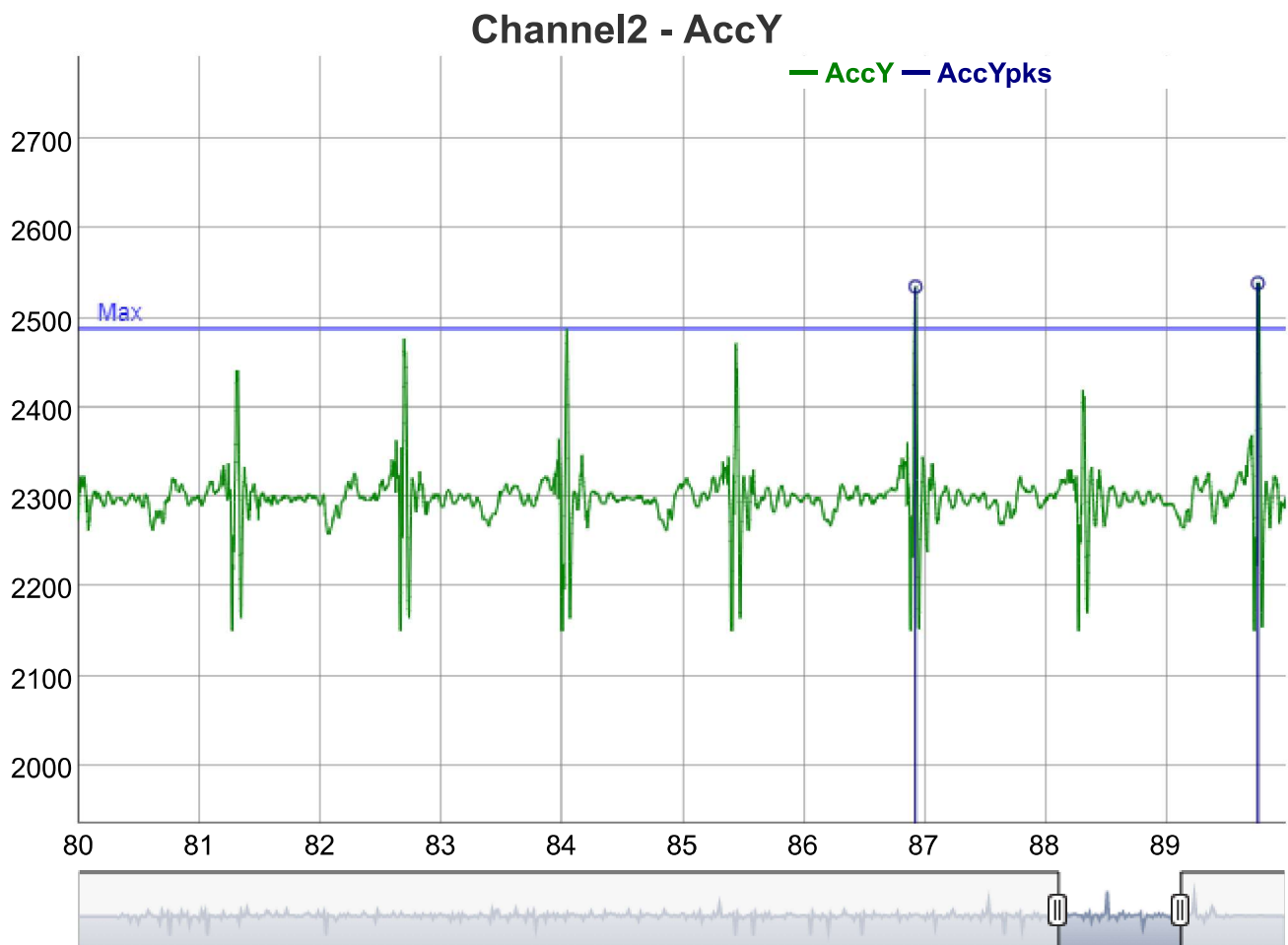
```
# step 2 - track peaks in AccY
```

```
library(pracma)
threshold = 0.98 # >90% of max
th <- max(df$AccY, na.rm = TRUE)*threshold # apply threshold as 98% of max value
pp <- findpeaks(df$AccY, minpeakheight=th) # find peaks in all dataframe based in the threshold
idxpk <- pp[, 2] # return indexes for peaks
```

```
# step 3 - plot peaks in time series
```

```
df$AccYpks <- NA # peaks for AccY # initialize columns to store peak value  
df$AccYpks[idxpk] <- df$AccY[idxpk] # attribute peak value
```

```
library(dygraphs)  
df[c("Time", "AccY", "AccYpks")] %>%  
  dygraph(group="ch2", main="Channel2 - AccY") %>%  
  dySeries("AccYpks", stemPlot=TRUE) %>%  
  dyAxis("y", valueRange = c(min(df$AccY)*0.9, max(df$AccY)*1.1)) %>%  
  dyLimit(th, "Max", strokePattern = "solid", color = "blue") %>%  
  dyRangeSelector(c(80,90))
```



```
# step 4 - generate statistics
```

```
data <- df$MuscAnterior # data for analysis  
wTime <- 500/1000 # window time in seconds  
wLen <- splRate*wTime # window length in samples  
idxIni <- idxpk # index for start of each data window  
nWin <- length(idxpk) # number of windows  
dt <- 1/splRate # time resolution  
  
st <- data.frame(time=rep(NA,times=nWin)) # data frame of statistics  
  
st$time <- NA # time  
st$mean <- NA # mean  
st$median <- NA # median
```

```

st$mode      <- NA # mode
st$amplitude <- NA # amplitude
st$var       <- NA # variance
st$covar     <- NA # variance coefficient
st$intqt     <- NA # interquartil distance

for(i in 1:nWin) {
  wData      <- data[idxIni[i]:(idxIni[i]+wLen)] # data window extract for calculations
  st$time[i]  <- (idxIni[i] + wLen/2) * dt # time vector centralized in each window
  st$mean[i]  <- mean(wData) # mean
  st$median[i] <- median(wData) # median
  st$mode[i]  <- as.numeric(names(table(wData))[table(wData)==max(table(wData))]) # mode
  st$amplitude[i] <- max(wData)-min(wData) # amplitude
  st$var[i]   <- var(wData) # variance
  st$covar[i] <- 100*sd(wData)/st$mean[i] # variance coefficient
  st$intqt[i] <- diff(quantile(wData, names=FALSE)[c(2,4)]) # interquartil distance
}

```

```

## Warning in st$mode[i] <- as.numeric(names(table(wData))[table(wData) == : número
## de itens para para substituir não é um múltiplo do comprimento do substituto

## Warning in st$mode[i] <- as.numeric(names(table(wData))[table(wData) == : número
## de itens para para substituir não é um múltiplo do comprimento do substituto

## Warning in st$mode[i] <- as.numeric(names(table(wData))[table(wData) == : número
## de itens para para substituir não é um múltiplo do comprimento do substituto

## Warning in st$mode[i] <- as.numeric(names(table(wData))[table(wData) == : número
## de itens para para substituir não é um múltiplo do comprimento do substituto

## Warning in st$mode[i] <- as.numeric(names(table(wData))[table(wData) == : número
## de itens para para substituir não é um múltiplo do comprimento do substituto

## Warning in st$mode[i] <- as.numeric(names(table(wData))[table(wData) == : número
## de itens para para substituir não é um múltiplo do comprimento do substituto

```

```

# step 5 - plot of median variation

data      <- df$MuscAnterior # data for analysis
wTime     <- 500/1000 # window time in seconds
wLen      <- splRate*wTime # window length in samples
idxIni    <- seq(from=1, to=length(data), by=wLen) # index for start of each data window
nWin      <- length(idxIni) # number of windows
dt        <- 1/splRate # time resolution

plt        <- data.frame(time=rep(NA, times=nWin-1)) # data frame for meadian plot
plt$time   <- NA
plt$median <- NA

for(i in 1:(nWin-1)){
  plt$time[i] <- (idxIni[i] + (idxIni[i+1]-idxIni[i])/2) * dt # time vector
  plt$median[i] <- median(data[idxIni[i]:idxIni[i+1]])
}

```

```
library(dygraphs)
dygraph(plt, group="dMedian", main="Variation of median accross time - MuscAnterior") %>%
  dyRangeSelector()
```

