

AllTonesPlot

Henry Heyden

2025-10-25

```
# Libraries
library(tidyverse)
library(mgcv)
library(sjlabelled)

# Data loading helper function!
high_tones <- list('b', 'j', 'g')
med_low_tones <- list('0', 's', 'v', 'm', 'd') # Redundant but good info

prepareData <- function(df, duration_dataframe) {
  df <- as_factor(df, Filename, Speaker, Tone, Word, NormalizedWord, ui, uf, p, df, PreviousTone)
  newdata <- df %>%
    left_join(duration_dataframe, by = 'token_id') %>%
    mutate(
      UtterancePosition = case_when(
        p == 1 ~ 'X', # do not use (pause)
        df == 1 ~ 'X', # do not use (disfluency)
        ui == 1 ~ 'INITIAL', # initial
        uf == 1 ~ 'FINAL', # final
        TRUE ~ 'MEDIAL' # medial
      ),
      PreviousToneHeight = case_when(
        PreviousTone %in% high_tones ~ 'High',
        TRUE ~ 'MED/LOW'
      )
    )
  newdata <- newdata %>% filter(!(UtterancePosition == 'X')) %>% filter(!(is.na(F0))) %>% select(!(ui:d
  return(as_factor(newdata, Filename, Speaker, Tone, Word,
                    NormalizedWord, PreviousTone, PreviousToneHeight, UtterancePosition))
}

# Load Data for All Tones
hmongData0 <- prepareData(read_csv("normalized_AVG_R/hmongData-0-ST_AVG-R.csv"),
                           read_csv("duration_csvs/hmongData-0.csv"))
hmongDataB <- prepareData(read_csv("normalized_AVG_R/hmongData-b-ST_AVG-R.csv"),
                           read_csv("duration_csvs/hmongData-b.csv"))
hmongDataD <- prepareData(read_csv("normalized_AVG_R/hmongData-d-ST_AVG-R.csv"),
                           read_csv("duration_csvs/hmongData-d.csv"))
hmongDataG <- prepareData(read_csv("normalized_AVG_R/hmongData-g-ST_AVG-R.csv"),
                           read_csv("duration_csvs/hmongData-g.csv"))
hmongDataJ <- prepareData(read_csv("normalized_AVG_R/hmongData-j-ST_AVG-R.csv"),
                           read_csv("duration_csvs/hmongData-j.csv"))
```

```

hmongDataM <- prepareData(read_csv("normalized_AVG_R/hmongData-m-ST_AVG-R.csv"),
                           read_csv("duration_csvs/hmongData-m.csv"))
hmongDataS <- prepareData(read_csv("normalized_AVG_R/hmongData-s-ST_AVG-R.csv"),
                           read_csv("duration_csvs/hmongData-s.csv"))
hmongDataV <- prepareData(read_csv("normalized_AVG_R/hmongData-v-ST_AVG-R.csv"),
                           read_csv("duration_csvs/hmongData-v.csv"))

gam_0 = gam(F0 ~ s(NormalizedTime), data = hmongData0, method = 'REML')
gam_b = gam(F0 ~ s(NormalizedTime), data = hmongDataB, method = 'REML')
gam_d = gam(F0 ~ s(NormalizedTime), data = hmongDataD, method = 'REML')
gam_g = gam(F0 ~ s(NormalizedTime), data = hmongDataG, method = 'REML')
gam_j = gam(F0 ~ s(NormalizedTime), data = hmongDataJ, method = 'REML')
gam_m = gam(F0 ~ s(NormalizedTime), data = hmongDataM, method = 'REML')
gam_s = gam(F0 ~ s(NormalizedTime), data = hmongDataS, method = 'REML')
gam_v = gam(F0 ~ s(NormalizedTime), data = hmongDataV, method = 'REML')

pd <- data.frame(NormalizedTime = seq(0, 1, by = 0.001))
pr_0 <- predict(gam_0, newdata = pd, type = "response")
pr_b <- predict(gam_b, newdata = pd, type = "response")
pr_d <- predict(gam_d, newdata = pd, type = "response")
pr_g <- predict(gam_g, newdata = pd, type = "response")
pr_j <- predict(gam_j, newdata = pd, type = "response")
pr_m <- predict(gam_m, newdata = pd, type = "response")
pr_s <- predict(gam_s, newdata = pd, type = "response")
pr_v <- predict(gam_v, newdata = pd, type = "response")

quartzFonts(times = quartzFont(c("Times-Roman", "Times-Bold", "Times-Italic", "Times-BoldItalic")))

plot(NULL, xlim=c(0,1), ylim=c(-6,5), ylab="Fundamental Frequency (st)", xlab="Normalized Time", cex.lab=1.5)

lines(pd$NormalizedTime, pr_d, col = 'gray11', lwd = 2.2, lty = "2262")
lines(pd$NormalizedTime, pr_s, col = 'steelblue1', lwd = 2.2, lty = "dotted")
lines(pd$NormalizedTime, pr_0, col = 'seagreen2', lwd = 2.2, lty = "44")
lines(pd$NormalizedTime, pr_b, col = 'indianred2', lwd = 2.2)
lines(pd$NormalizedTime, pr_g, col = 'darkorchid2', lwd = 2.2, lty = "dotdash")
lines(pd$NormalizedTime, pr_j, col = 'sandybrown', lwd = 2.2, lty = "13")
lines(pd$NormalizedTime, pr_m, col = 'palevioletred3', lwd = 2.2, lty = "longdash")
lines(pd$NormalizedTime, pr_v, col = 'green4', lwd = 2.2, lty = "dashed")

op <- par(cex = 0.6)

legend("bottomleft", legend=c("Mid", "b", "d", "g", "j", "m", "s", "v"),
      fill=c(
        'seagreen2',
        'indianred2',
        'gray11',
        'darkorchid2',
        'sandybrown',
        'palevioletred3',
        'steelblue1',
        'green4'), col = c(
        'seagreen2',
        'indianred2',
        'gray11',

```

```

'darkorchid2',
'sandybrown',
'palevioletred3',
'steelblue1',
'green4'), lty=c("44", "solid", "2262", "dotdash", "13", "longdash", "dotted", "dashed"), lwd

```

