HASP Monthly Status Report - October 2014

Balloons over Volcanoes Team

October 27, 2014

Student Leader: Daniel C. Bowman (phone: 1-575-418-8555, email: daniel.bowman@unc.edu)

Faculty Advisor: Jonathan M. Lees (phone: 1-919-962-1562, email: jonathan.lees@unc.edu)

1 Synopsis

- Started work on scientific report
- Determined frequency response of microphones
- Acoustic spectra analysis ongoing
- Video analysis ongoing

2 Activity Summary

Daniel Bowman measured the corner frequency of the mechanical filters used during flight and wrote up the methodology. We extrapolated the results to stratospheric pressures and found that the corner frequency decreases by several orders of magnitude. The writeup is attached as an appendix to this report. Spectra of signals recorded when the balloon was near Albuquerque, New Mexico, contain peaks that correlate with a local ground infrasound station (see Figure 1). Jonathan Lees has investigated signal coherency, and we have confirmed that the 17 Hz peak on the spectrum is coherent with respect to the ground station, indicating that the signal propagates from the surface to the stratosphere. Team member Patrick Gauge continues to analyze video data recorded during flight to determine if there are any connections between signals on our sensors and events on the gondola.

3 Issues Encountered

Due to the sparsity of ground stations and lack of horizontal control on the balloon array, we cannot determine the source of the signal recorded simultaneously on the surface and the stratosphere.

4 Milestones Achieved

We confirmed the observation of simultaneous infrasound on the ground and the balloon from a common source.

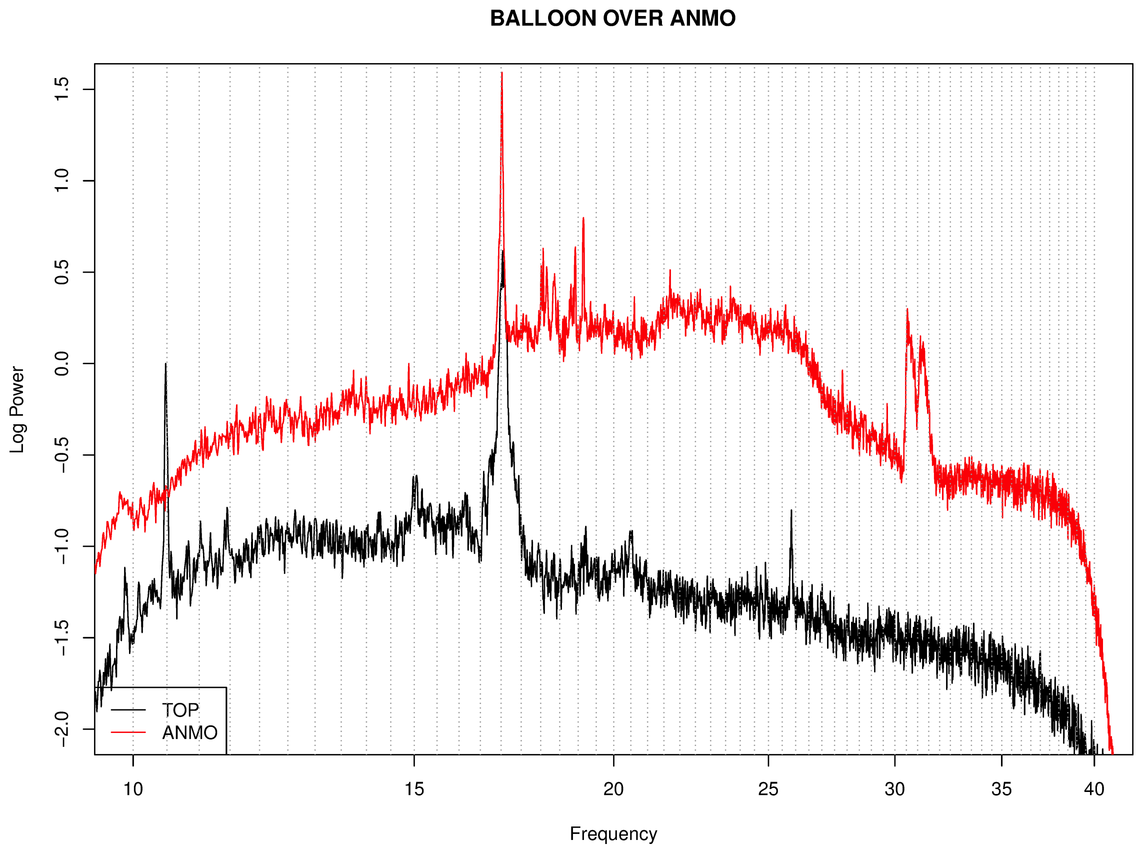

Figure 1: Acoustic spectrum of station ANMO (south of Albuquerque, NM) compared with the spectrum of station TOP (top of balloon ladder) during the hour of closest approach to ANMO.

5 Team

The student team consists of Daniel C. Bowman and Patrick Gouge (University of North Carolina at Chapel Hill), Jacob F. Anderson (Boise State University), and Tierney Larson (Yale University). Jonathan M. Lees (UNC Chapel Hill) serves as Faculty Advisor. Paul Norman and Kyle Jones are outside advisors.

6 Appendix

Determining the Frequency Response of a Differential Pressure Transducer

Daniel C. Bowman

October 27, 2014

1 Description of Instrument

A differential pressure transducer consists of two chambers separated by a diaphragm. One chamber has a port open to the environment, and the other has a port with a mechanical low pass filter attached. Pressure difference between the open chamber and the filtered chamber cause the diaphragm to deflect. Sensors on this diaphragm produce a voltage in response to the deflection (Figure 1). Additional details can be found in Marcillo et al. (2012) and Mutschlecner and Whitaker (1997).

The mechanical low pass filter consists of a backing volume connected to the environment via a capillary tube. The filter is conceptually equivalent to an electrical circuit consisting of a resistor and a capacitor in series. In this analogy, the capillary tube acts as a resistor and the backing volume is a capacitor. Pressure is equivalent to voltage and fluid flow through the capillary tube is equivalent to current. The result is a single pole low pass filter between the environment and one port of the transducer. Since the transducer only produces a signal when there is a pressure difference across the diaphragm, the end result is a high pass filter that is sensitive to transient fluctuations (i.e. infrasound) but does not detect broader pressure changes due to barometric or thermal forcing.

The acoustic resistance of the capillary tube is derived from the expression for Poiseuille flow (Mutschlecner and Whitaker, 1997):

$$R = \frac{8\eta}{\pi r^4} l \tag{1}$$

where η is the shear viscosity of the fluid, r is the radius of the capillary tube, and l is the length of the capillary tube. This expression assumes equilibrium laminar fluid flow in long cylinders.

The acoustic capacitance is

$$C = \frac{V}{\gamma \bar{P}} \tag{2}$$

where V is the backing volume, γ is the adiabatic gas constant, and $\bar{P}$ is the average background pressure. In Earth's atmosphere, the value of γ varies between 1.403 in pure adiabatic and 1 in pure isothermal conditions; this is dependent on the frequency of pressure fluctuations, air conditions, and chamber geometry (Marcillo et al., 2012). The adiabatic regime dominates at high frequencies, and isothermal effects appear at low frequencies.

The response of the microphone to a time varying pressure $p(t)$ is

$$\Delta p(t) = p(t) - h(t) * p(t) \tag{3}$$

where the impulse response of the low pass filter is represented by $h(t)$. The frequency response of the system is thus

$$\Delta p(\omega) = P(\omega) H(\omega) \tag{4}$$

where $H(\omega)$ is the frequency response of the low pass filter. The frequency response is a function of the acoustic resistance and capacitance via

$$H(\omega) = \frac{1}{1 + \imath \omega RC} \tag{5}$$

and the corner frequency of the filter is

$$f_{corner} = \frac{1}{RC}. \tag{6}$$

2 Calculating the Frequency Response

The expressions in the previous section depend on knowledge of RC, the product of acoustic resistance and acoustic capacitance in the mechanical filter. This product can be extracted by considering the time evolution a single pole low pass filter responding to a step function:

$$p(t) = p_0 e^{-\frac{t}{RC}}. \tag{7}$$

The expression above suggests two approaches. The first method involves measuring the time interval between an initial pressure p_0 and a final pressure $\frac{p_0}{e}$:

$$p_0 e^{-1} = p_0 e^{\frac{-t}{RC}}$$

Dividing through by p_0 and taking the logarithm gives

$$-1 = \frac{-t}{RC},$$

resulting in

$$t = RC.$$

Another, more statistically rigorous, approach involves taking the logarithm of both sides of Equation 7, yielding a linear expression with slope $\frac{-1}{RC}$:

$$\log p(t) = \log p_0 - \frac{1}{RC}t. \tag{8}$$

Performing a linear regression on this data provides an estimate and confidence intervals for RC. The quality of the model fit also allows a useful check on the performance of the mechanical filter with regards to its expected time evolution.

3 Testing Procedure

Pressure step functions can be generated by attaching a syringe to the mechanical filter and depressing the plunger slightly. Another simple method is to remove the mechanical filter from the transducer port and then reattach it. The second method tends to produce large overpressures. The result is a decaying exponential curve per Equation 7 (Figure 2). After flipping the time series around the x-axis and subtracting off the minimum value (if necessary), take the logarithm and fit a linear model using least squares regression (Figure 3). There is a small but systematic (nonrandom) misfit between the linear model and the data; this appears in most tests. This indicates that an RC-circuit model does not perfectly describe the behavior of these filters, and thus should be treated as a useful approximation rather than a comprehensive description.

An example test series is shown in Figure 4. These filters were attached to differential pressure transducers on board a high altitude balloon as part of the NASA HASP project. Stations BOT and TOP had the same type of filter, but station MID had a different capillary tube and a modified backing volume. BOT and TOP were tested 3 times each by recording the pressure decay when the filter was plugged into the transducer. MID was tested twice in this manner, and a further 3 times using a syringe. The similarities between BOT and TOP are apparent, but MID had a much lower corner period (approximately 10 seconds). In general, plug tests give more consistent results than syringe tests. This is evident on this plot as well as during tests not described here.

4 Environmental Effects on Corner Period

Differential pressure transducers should be tested in an environment similar to the one present during data acquisition. However, in some cases (such as stratospheric balloon flights), this may not be feasible. In these types of situations, changes to the RC term should be investigated to get a sense of the changes in frequency response that will occur. The two atmospheric parameters that affect RC are shear viscosity η and ambient pressure $\bar{P}$. An increase in η leads to an increase in R (see Equation 1) and thus an increase in corner period. Conversely, an increase in $\bar{P}$ leads to a decrease in C (see Equation 2) and thus a decrease in corner period.

Shear viscosity is nearly independent of pressure, but it does vary with temperature. Sutherland's formula

$$\eta = \eta_0 \frac{T_0 + 110.3}{T + 110.3} \left(\frac{T}{T_0} \right) \tag{9}$$

where η_0 is a reference viscosity at temperature T_0 (in Kelvin), describes the shear viscosity of air between about 100 and 1900 Kelvin (Ames Research Staff, 1953). The expected variation in corner period due to temperature alone is shown in Figure 5. Since pressure decreases by two orders of magnitude between the surface and the stratosphere, it will have a greater effect on the corner period than temperature. A plot of corner period versus altitude for a typical day in Chapel Hill is shown in Figure 6. The extreme increase in corner period in the stratosphere implies that microphones on high altitude balloon flights should have small backing volumes.

References

Ames Research Staff (1953). Equations, tables, and charts for compressible flow. Technical report, National Advisory Committee for Aeronautics. Report 1135.

Marcillo, O., Johnson, J. B., and Hart, D. (2012). Implementation, characterization, and evaluation of an inexpensive low-power low-noise infrasound sensor based on a micromachined differential pressure transducer and a mechanical filter. *Journal of Atmospheric and Oceanic Technology*, 29:1275–1284.

Mutschlecner, J. P. and Whitaker, R. W. (1997). The design and operation of infrasonic microphones. Technical report, Los Alamos National Laboratories.

5 Figures

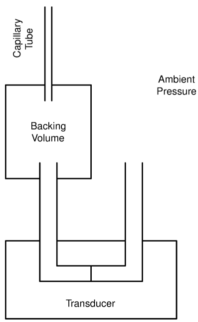

Figure 1: Schematic diagram of InfraNMT mechanical high pass filter, modified after Figure 1b in Marcillo et al. (2012)

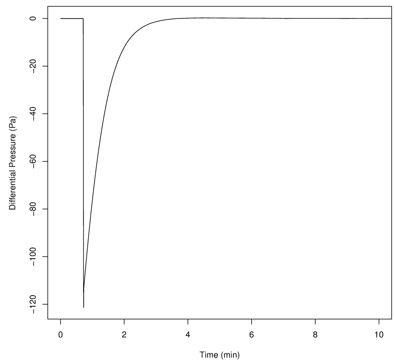

Figure 2: Response of differential pressure transducer when the mechanical filter is unplugged, then plugged back in. This pressurizes the backing volume, resulting in a negative differential pressure.

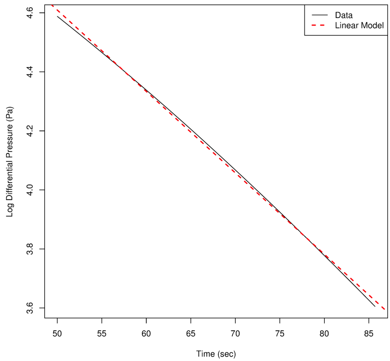

Figure 3: A linear least squares regression model applied to time versus the logarithm of pressure. The slope of the regression line is related to the product of acoustic resistance and acoustic capacitance per Equation 8.

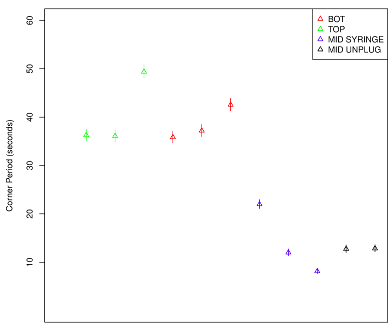

Figure 4: Example of test data from the 2014 NASA HASP project. Filters BOT and TOP were the same model, but filter MID had a different backing volume and capillary tube. BOT and TOP were both tested by unplugging and plugging in the filters, MID SYRINGE was tested using a syringe to pressurize the filter. The lines represent 95% confidence intervals.

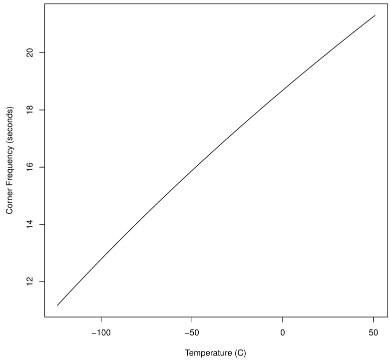

Figure 5: Corner frequency variation with temperature.

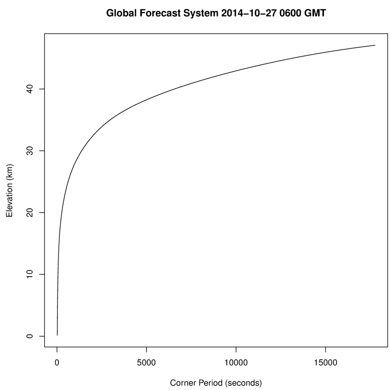

Figure 6: Corner period with height over Chapel Hill, North Carolina. The filter has a corner period of 20 seconds at 25 Celsius and 1000 millibars.

6 R Code for Generating Figures

```
> ####
> #   SETUP PLOTTING FUNCTIONS
> ####
>
> dir.create("figures", showWarnings = FALSE)
> DPostscript <- function(file.name, width = 8, height = 8)
+ {
+     postscript(file=file.name, width = width,
+        height = height, paper = "special",
+        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
+ }
> EPS2PNG <- function(fig.dir) { #Convert eps files to png so you can use pdflatex
+     oldwd <- getwd()
+     setwd(fig.dir)
+     system("perl ps2png.prl *eps")
+     system("sh png_transform.sh")
+     setwd(oldwd)
+ }
> fig.dir <- "figures/"
```

```
> ####
> # TRANSDUCER SCHEMATIC
> ####
> DPostscript("figures/transducer.eps")
> plot(c(0, 1.1), c(0, 1.1), type = "n", xlab = "", ylab = "", axes = FALSE)
> lty = 1
> lwd = 2
> #Make pressure transducer box
> x <- c(0.35, 0.25, 0.25, 0.75, 0.75, 0.65)
> y <- c(0.25, 0.25, 0.00, 0.00, 0.25, 0.25)
> lines(x, y, lty = lty, lwd = lwd)
> #Make filter-end piping
> x <- c(0.35, 0.35, 0.50, 0.50, 0.40, 0.40)
> y <- c(0.50, 0.12, 0.12, 0.17, 0.17, 0.50)
> lines(x, y, lty = lty, lwd = lwd)
> lines(1 - x, y, lty = lty, lwd = lwd)
> lines(c(x[6], 1 - x[6]), rep(0.25, 2), lty = lty, lwd = lwd)
> #Make backing volume
> x <- c(0.35, 0.25, 0.25, 0.365)
> y <- c(0.45, 0.45, 0.75, 0.75)
> lines(x, y, lty = lty, lwd = lwd)
> lines(0.375 * 2 - x, y, lty = lty, lwd = lwd)
> #Make capillary tube
> x <- rep(0.365, 2)
> y <- c(0.70, 1.00)
> lines(x, y, lwd = lwd, lty = lty)
> lines(0.375 * 2 - x, c(0.70, 1.00), lwd = lwd, lty = lty)
> #Text labels
> text(0.50, 0.05, "Transducer")
> text(0.375, 0.6, "Backing\nVolume")
> text(0.30, 0.90, "Capillary\nTube", srt = 90)
> text(0.75, 0.75, "Ambient\nPressure")
> dev.off()
```

```
null device
          1

> ####
> # TIME DECAY
> ####
>
> data <- scan("data/TOP1.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> DPostscript("figures/step_response.eps")
> plot(tt/60, xt, type = "l", xlim = c(0, 10), xlab = "Time (min)",
+     ylab = "Differential Pressure (Pa)")
> dev.off()

null device
          1

> ####
> # FITTING LINEAR MODEL
> ####
>
> data <- scan("data/TOP1.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 50)
> sig <- log(xt2[ind]) #Flip and subtract min so the series ends in 0
> tt2 <- tt[ind]
> sl <- lm(sig ~ tt2)
> DPostscript("figures/linear_model.eps")
> plot(tt2, sig, type = "l", xlab = "Time (sec)", ylab = "Log Differential Pressure (Pa)")
> abline(sl, col = "red", lty = 2, lwd = 2)
> legend("topright", lty = c(1, 2), lwd = c(1, 2),
+     col = c("black", "red"), legend = c("Data", "Linear Model"))
> dev.off()

null device
          1

> ####
> # CORNER FREQUENCY CALCULATION
> ####
>
> test.data <- list()
> filter.name <- NULL
> dt <- 0.01 #Sampling rate
> conf <- 0.95 #Confidence interval for regression
> ttail <- c((1 - conf)/2, 1 - (1 - conf)/2)
> #Subset and orient data
> data <- scan("data/TOP1.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 50)
```

```
> test.data[[1]] <- xt2[ind]
> filter.name[1] <- "TOP"
> data <- scan("data/TOP2.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt[1000:1500]))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 53)
> test.data[[2]] <- xt2[ind]
> filter.name[2] <- "TOP"
> data <- scan("data/TOP3.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 46)
> test.data[[3]] <- xt2[ind]
> filter.name[3] <- "TOP"
> data <- scan("data/BOT1.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 49)
> test.data[[4]] <- xt2[ind]
> filter.name[4] <- "BOT"
> data <- scan("data/BOT2.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 23)
> test.data[[5]] <- xt2[ind]
> filter.name[5] <- "BOT"
> data <- scan("data/BOT3.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 60)
> test.data[[6]] <- xt2[ind]
> filter.name[6] <- "BOT"
> data <- scan("data/MID_SYRINGE1.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt <- xt[tt > 63]
> tt <- tt[tt > 63]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 > max(xt2)/exp(1))
> test.data[[7]] <- xt2[ind]
> filter.name[7] <- "MID_SYRINGE"
> data <- scan("data/MID_SYRINGE2.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt <- xt[tt > 60]
> tt <- tt[tt > 60]
> xt2 <- xt - min(xt)
> ind <- which(xt2 > max(xt2)/exp(1))
> test.data[[8]] <- xt2[ind]
```

```
> filter.name[8] <- "MID_SYRINGE"
> data <- scan("data/MID_SYRINGE3.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt <- xt[tt > 60 & tt < 80]
> tt <- tt[tt > 60 & tt < 80]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 > max(xt2)/exp(1))
> test.data[[9]] <- xt2[ind]
> filter.name[9] <- "MID_SYRINGE"
> data <- scan("data/MID_UNPLUG1.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 65)
> test.data[[10]] <- xt2[ind]
> filter.name[10] <- "MID_UNPLUG"
> data <- scan("data/MID_UNPLUG2.ascii", skip = 1)
> tt <- data[seq(1, (length(data) - 1), by = 2)]
> xt <- data[seq(2, length(data), by = 2)]
> xt2 <- -(xt - max(xt))
> ind <- which(xt2 < 100 & xt2 > 100/exp(1) & tt > 65)
> test.data[[11]] <- xt2[ind]
> filter.name[11] <- "MID_UNPLUG"
> #Calculate corner periods and confidence intervals
>
> corner.period <- rep(NA, length(test.data))
> conf.intervals <- array(NA, dim = c(length(test.data), 2))
> for(k in 1:length(test.data)) {
+    x <- (1:length(test.data[[k]])) * dt
+    y <- log(test.data[[k]])
+    sl <- lm(y ~ x)
+    corner.period[k] <- -1/sl$coefficients[2]
+    mse <- sum((y - mean(y))^2)/length(x)
+    conf.intervals[k,] <- -1/(sl$coefficients[2] +
+        qt(ttail, length(x) - 2) * sqrt(mse/sum((x - mean(x))^2)))
+ }
> #Plot them
> DPostscript("figures/balloon_filters.eps")
> filters <- c("BOT", "TOP", "MID_SYRINGE", "MID_UNPLUG")
> cols <- c("red", "green", "blue", "black")
> plot(c(1, length(test.data) + 1), c(0, 60), type = "n",
+    xlab = "", ylab = "Corner Period (seconds)", axes = FALSE)
> axis(2, at = seq(10, 60, by = 10))
> box()
> for(k in 1:length(corner.period)) {
+    points(k + 1, corner.period[k], pch = 2,
+        col = cols[which(filter.name[k] == filters)])
+    segments(k + 1, conf.intervals[k, 1], k + 1,
+        conf.intervals[k, 2], col = cols[which(filter.name[k] == filters)])
+ }
> legend("topright", pch = 2, col = cols,
+    legend = c("BOT", "TOP", "MID SYRINGE", "MID UNPLUG"))
> dev.off()
```

```
null device
          1

> ####
> # Corner frequency variation with temperature
> ####
>
> #Assume 20 second period at 25 degrees Celsius
> t0 <- 25 + 274.15
> t <- seq(150, 325)
> cp <- 20 * ((t0 + 110.3)/(t + 110.3)) * (t/t0)^(3/2)
> DPostscript("figures/viscosity.eps")
> plot(t - 274.15, cp, xlab = "Temperature (C)", ylab = "Corner Frequency (seconds)", type = "l")
> dev.off()

null device
          1

> ####
> # Corner frequency variation with altitude
> ####
>
> library(rNOMADS)
> lat <- 35.907553
> lon <- -79.052104
> lons <- seq(0, 359.5, by = 0.5)
> lats <- seq(-90, 90, by = 0.5)
> lon.diff <- abs(lon + 360 - lons)
> lat.diff <- abs(lat - lats)
> model.lon.ind <- which(lon.diff == min(lon.diff)) - 1
> model.lat.ind <- which(lat.diff == min(lat.diff)) - 1
> model.urls <- GetDODSDates("gfs_hd")
> latest.model <- tail(model.urls$url, 1)
> model.runs <- GetDODSModelRuns(latest.model)
> latest.model.run <- tail(model.runs$model.run, 1)
> variables <- c("hgtprs", "tmpprs")
> lev <- c(0, 46) #All levels in atmosphere
> time <- c(0,0) #Analysis model
> raw.data <- DODSGrab(latest.model, latest.model.run,
+    variables, time, rep(model.lon.ind, 2), rep(model.lat.ind, 2),
+    levels = lev, display.url = FALSE)
> pressure <- 100 * rev(c(1, 2, 3, 5, 7,
+ 10, 20, 30, 50, 70,
+ seq(100, 1000, by = 25)))
> hgt <- raw.data$value[which(raw.data$variables == "hgtprs")]
> tmp <- raw.data$value[which(raw.data$variables == "tmpprs")]
> prs.spline <- splinefun(hgt, pressure, method = "natural")
> tmp.spline <- splinefun(hgt, tmp, method = "natural")
> synth.hgt <- seq(min(hgt), max(hgt), length.out = 1000)
> synth.prs <- prs.spline(synth.hgt)
> synth.tmp <- tmp.spline(synth.hgt)
> #Assume 20 sec corner at 25 C and 1000 mb
> t0 <- 25  + 274.15
> suth <- ((t0 + 110.3)/(synth.tmp + 110.3)) * (synth.tmp/t0)^(3/2)
> cp <- suth * 2000000 / synth.prs
```

```
> n <- nchar(latest.model)
> n1 <- nchar(latest.model.run)
> DPostscript("figures/period_profile.eps")
> plot(cp, synth.hgt/1000, xlab = "Corner Period (seconds)", ylab = "Elevation (km)", type = "l",
+    main = paste0("Global Forecast System ", substr(latest.model, n - 7, n - 4), "-",
+    substr(latest.model, n - 3, n - 2), "-", substr(latest.model, n - 1, n), " ",
+    substr(latest.model.run, n1-2, n1-1), "00 GMT"))
> dev.off()

null device
          1

> #Convert figures to PNG for plotting
> EPS2PNG("figures")
```