Calibration of USRP radar digitizers; Nov. 2015

John Brzustowski

Summary

USRPs with custom front-end boards and firmware adapted to radar digitizing were calibrated with a radar. Microwave pulses of known power were fed into the radar via 3 m of LMR 400 ultraflex cable, then into a cable-waveguide adapter, then through a 30cm length of flexible waveguide attached to the radar. Components downstream of the cable-waveguide adapter are the same as used in field radar studies.

For each digitizer, the results are plotted, and an R data file (.rds) is generated. It contains a list with two items:

- **data**: a data.frame of measurements. It has these columns:
    - *attendB*: level of attenuator, if present. 0 otherwise.
    - *sigdBm*: power level of generated microwave pulses
    - *vidGain*: USRP video gain setting (dB, but this is voltage gain; see notes below)
    - *usrpVal*: mean of maximum sampled value for each of 50 pulses, obtained as 5 groups of 10 consecutive pulses, with 1 second between groups
    - *sdUsrpVal*: standard deviation (not standard error) of usrpVal across 50 pulses
    - *indBm*: power level of signal reaching waveguide (sigdBm - attendB)
    - *adjUsrpVal*: usrpVal adjusted to take vidGain into account; i.e. what the usrpVal would have been if vidGain were zero.
- **map**: a function mapping USRP values to dBm entering waveguide. It has these parameters:
    - *val*: the USRP values
    - *gain*: USRP video gain setting (defaults to 0)

This function is valid over the range 2048..4095 of USRP values. (Because the video signal is measured against ground, we only end up using the top half of the analog to digital converter's sample range; samples 0..2047 would correspond to a video signal at a level below ground. Ideally, the USRP front-end board would bias the incoming signal to cover the entire range.)

Equipment:

- Furuno FR 1965 X-band radar
- Wavetek 907A-S-792 Microwave signal generator
- Tektronix 2445 scope
- Marconi RF power meter 6960
- USRP with DC front end board and custom FPGA / host build as used by radR's usrp plugin

Procedure:

0. R preliminaries:

```
library(dplyr, quietly=TRUE, verbose=FALSE, warn.conflicts=FALSE)
library(lattice)

dB = function(x) 10*log10(x) ## convert linear units to dB
undB = function(x) 10^(x/10) ## convert dB to linear units
```

1. Calculate calibration factor for RF probe @ 9.41 GHz

There is a table printed on the probe label. We use the nearby values.

```
probeCalib = '\
freqMHz    calibFactor
  8000        0.9797
  9000        0.9759
 10000        0.9600
 11000        0.9411\
' %>% textConnection %>% read.table (header=TRUE)

dBmProbeLoss = probeCalib %>%
    with(spline(freqMHz, calibFactor, xout=9410) $y) %>%
    dB
dBmProbeLoss
```

```
## [1] -0.130818
```

2. Validate output from microwave generator against RF meter

- set internal pulse generator on microwave source to PRF=2100, PLEN = 160 ns (the shortest pulse it can generate; still twice the nominal length of the Furuno's SP mode); set frequency to 9.41 GHz

```
PRF = 2100
```

- use RF meter's internal source to calibrate it.
- attach probe cable: RF Meter <–> Signal generator
- auto-zero the RF Meter
- read power meter level for different output settings:

```
readings1 = '\
  plen   dBmGen   dBmMeter
 160e-9   -5.0      -38.25
 160e-9   -7.5      -41.02
 160e-9  -10.0      -43.89
 160e-9  -12.5      -46.67
 160e-9  -15.0      -49.5
 160e-9  -17.5      -52.3
 160e-9  -20.0      -55.4
 160e-9  -22.5      -58.5
 160e-9  -25.0      -62.8\
' %>% textConnection %>% read.table (header=TRUE)
```

- at this duty cycle, lower power levels drop out of the meter's range, so we use a 10 x wider pulse (1600 ns = 1.6e-6s)

```
readings2 = '\
  plen  dBmGen   dBmMeter
 1.6e-6  -22.5    -47.8
 1.6e-6  -25.0    -50.6
 1.6e-6  -27.5    -53.5
 1.6e-6  -30.0    -56.8
 1.6e-6  -32.5    -61.0
 1.6e-6  -34.0    -65.0\
' %>% textConnection %>% read.table (header=TRUE)
```

- again, use a wider pulse (16 us = 16e-6s)

```
readings3 = '\
 plen  dBmGen   dBmMeter
16e-6  -22.5     -37.6
16e-6  -25.0     -40.3
16e-6  -27.5     -42.8
16e-6  -30.0     -45.4
16e-6  -32.5     -48.1
16e-6  -35.0     -50.8
16e-6  -37.5     -53.6
```

```
16e-6  -40.0     -56.8
16e-6  -42.5     -60.8\
' %>% textConnection %>% read.table (header=TRUE)
```

- switch to continuous wave (plen = 1 / PRF = 4.762e-4s), and sometimes use a (nominally) 30 dB attenuator

```
readings4 = '\
  plen   dBmGen  dBmMeter  dBmAtten
4.762e-4    0.0   -25.6     30.0
4.762e-4   -2.5   -28.2     30.0
4.762e-4   -5.0   -30.5     30.0
4.762e-4   -7.5   -33.1     30.0
4.762e-4  -10.0   -35.7     30.0
4.762e-4  -12.5   -38.1     30.0
4.762e-4  -15.0   -40.6     30.0
4.762e-4  -17.5   -43.2     30.0
4.762e-4  -20.0   -45.8     30.0
4.762e-4  -22.5   -48.3     30.0
4.762e-4  -25.0   -51.2     30.0
4.762e-4  -27.5   -53.9     30.0
4.762e-4  -30.0   -57.2     30.0
4.762e-4  -19.0   -19.26     0.0
4.762e-4  -20.0   -20.32     0.0
4.762e-4  -22.5   -22.80     0.0
4.762e-4  -25.0   -25.42     0.0
4.762e-4  -27.5   -27.97     0.0
4.762e-4  -30.0   -30.51     0.0
4.762e-4  -32.5   -33.09     0.0
4.762e-4  -35.0   -35.50     0.0
4.762e-4  -37.5   -37.91     0.0
4.762e-4  -40.0   -40.39     0.0
4.762e-4  -42.5   -42.95     0.0
4.762e-4  -45.0   -45.42     0.0
4.762e-4  -47.5   -47.84     0.0
4.762e-4  -50.0   -50.45     0.0
4.762e-4  -52.5   -52.8      0.0
4.762e-4  -55.0   -55.4      0.0
4.762e-4  -56.0   -56.6      0.0
4.762e-4  -57.0   -57.9      0.0
4.762e-4  -57.5   -58.9      0.0
4.762e-4  -60.0   -64.5      0.0\
' %>% textConnection %>% read.table (header=TRUE)
```

- combine all readings into a master table

```
genCal = rbind (
    cbind(readings1, dBmAtten=0),
    cbind(readings2, dBmAtten=0),
    cbind(readings3, dBmAtten=0),
    cbind(readings4)) %>% as.tbl

genCal = genCal %>% mutate(
    dutyCycle = plen * PRF,
    trueMeter = ((dBmMeter + dBmProbeLoss) %>% undB / dutyCycle ) %>% dB + dBmAtten,
    attenFactor = as.factor(dBmAtten)
    )
```

The attenuator, rated at 30 dB (at 50 MHz) is not as effective at 9.4 GHz, as can be seen in this plot:

```
    xyplot(trueMeter~dBmGen,genCal,groups=attenFactor, auto.key=list(corner=c(0.2,1), title=
```

The meter reading is higher than it should be when the attenuator is present and corrected for with the nominal value. A simple model estimates the attenuator's offset from nominal as:

```
summary(lm(dBmGen~trueMeter*attenFactor, subset(genCal, plen = 1/2100)))
```

```
##
## Call:
## lm(formula = dBmGen ~ trueMeter * attenFactor, data = subset(genCal,
##     plen = 1/2100))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5373 -0.5340 -0.1512  0.1766  5.0131
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -0.45240    0.42905  -1.054    0.296
## trueMeter              0.95461    0.01186  80.465  < 2e-16 ***
## attenFactor30         -3.94156    0.64338  -6.126 1.14e-07 ***
## trueMeter:attenFactor30  0.00889    0.03476   0.256    0.799
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.143 on 53 degrees of freedom
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9942
## F-statistic:  3194 on 3 and 53 DF,  p-value: < 2.2e-16
```

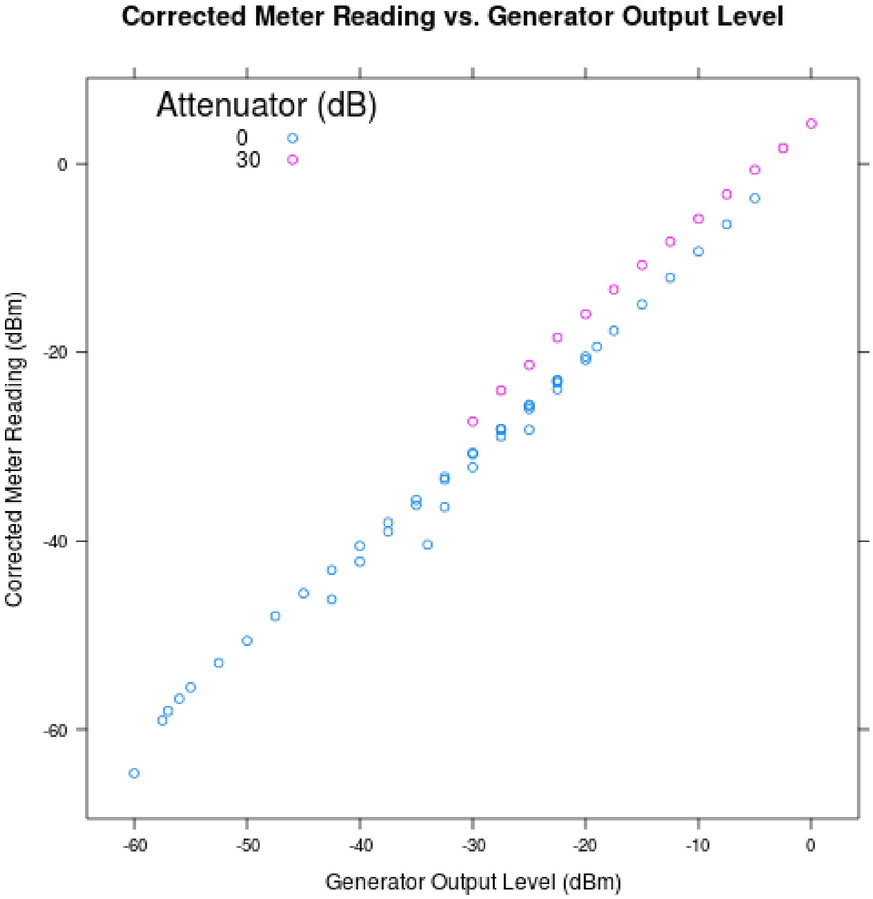


Figure 1: plot of chunk unnamed-chunk-9

From this we see that the readings in the presence of the attenuator are approximately 3.94 dB too high. So the true attenuation is closer to 26.06 dB. Indeed, using that value leads to this plot and model:

```
genCal2 = genCal %>% mutate(
    dBmAtten = ifelse(dBmAtten == 30, 26.06, 0),
    trueMeter = ((dBmMeter + dBmProbeLoss) %>% undB / dutyCycle ) %>% dB + dBmAtten,
    attenFactor = as.factor(dBmAtten)

)

xyplot(trueMeter~dBmGen,
genCal2,
groups=attenFactor, auto.key=list(corner=c(0.2,1), title="Attenuator (dB)"), main="Corre


summary(lm(dBmGen~trueMeter*attenFactor, subset(genCal2, plen = 1/2100)))
```

```
##
## Call:
## lm(formula = dBmGen ~ trueMeter * attenFactor, data = subset(genCal2,
##     plen = 1/2100))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5373 -0.5340 -0.1512  0.1766  5.0131
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -0.45240    0.42905  -1.054    0.296
## trueMeter                   0.95461    0.01186  80.465   <2e-16 ***
## attenFactor26.06           -0.14537    0.72326  -0.201    0.841
## trueMeter:attenFactor26.06  0.00889    0.03476   0.256    0.799
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.143 on 53 degrees of freedom
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9942
## F-statistic:  3194 on 3 and 53 DF,  p-value: < 2.2e-16
```

These plots also show that at the meter's linearity degrades quickly below readings of approximately -58 dBm. Dropping those from the table gives this plot and model:

```
genCal3 = genCal2 %>% filter(dBmMeter > -58) %>%
mutate (
```

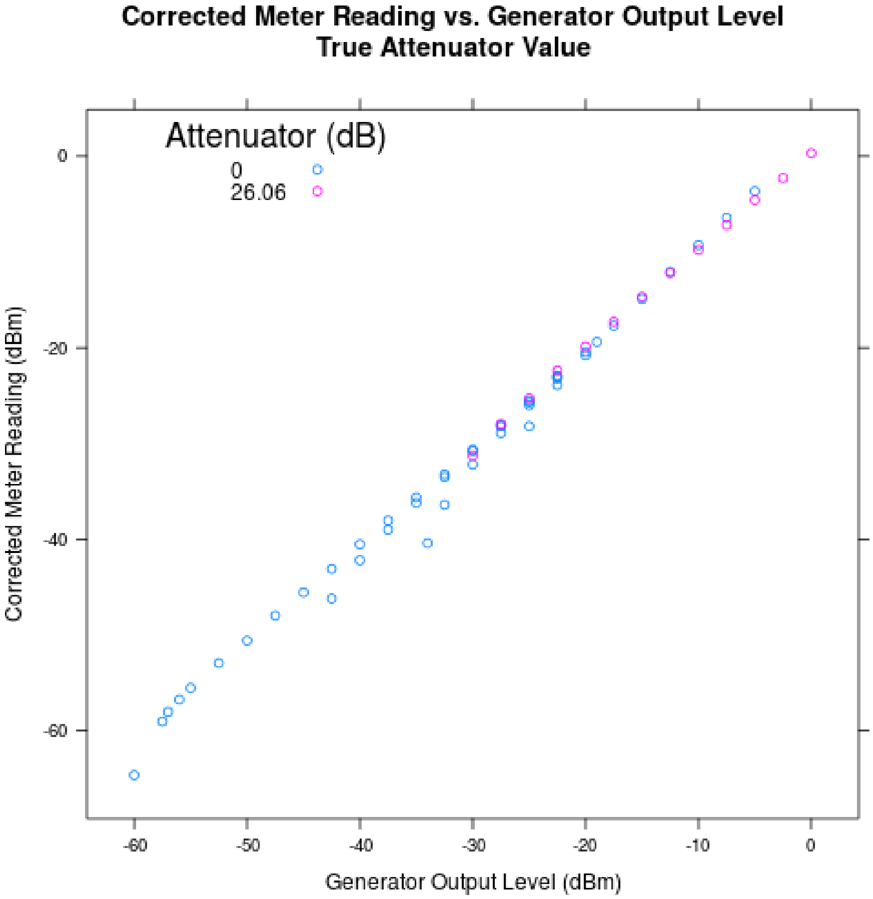


Figure 2: plot of chunk unnamed-chunk-11

```
   trueMeter = ((dBmMeter + dBmProbeLoss) %>% undB / dutyCycle ) %>% dB + dBmAtten
)

xyplot(trueMeter~dBmGen,
genCal3,
groups=attenFactor, auto.key=list(corner=c(0.2,1), title="Attenuator (dB)"), main="Corre
```

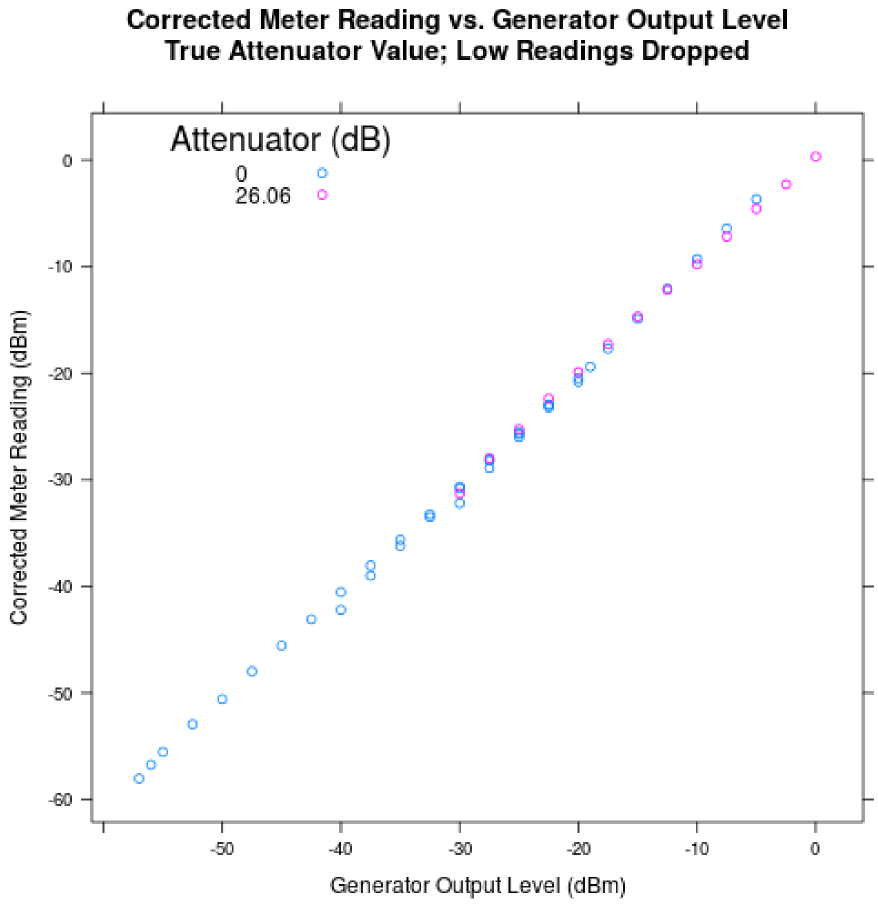


Figure 3: plot of chunk unnamed-chunk-12

```
summary(lm(dBmGen~trueMeter*attenFactor, subset(genCal3, plen = 1/2100)))
```

```
##
## Call:
## lm(formula = dBmGen ~ trueMeter * attenFactor, data = subset(genCal3,
##     plen = 1/2100))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.18021 -0.30453  0.00949  0.25673  1.58880
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -0.275268   0.219333  -1.255    0.216
## trueMeter                   0.972645   0.006402 151.929   <2e-16 ***
## attenFactor26.06           -0.322504   0.352403  -0.915    0.365
## trueMeter:attenFactor26.06 -0.009145   0.016749  -0.546    0.588
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5415 on 46 degrees of freedom
## Multiple R-squared:  0.9987, Adjusted R-squared:  0.9986
## F-statistic: 1.169e+04 on 3 and 46 DF,  p-value: < 2.2e-16
```

The point of all this is that the signal generator shows good linearity over the range we can check with our (old, not very sensitive, but self-calibrated) RF meter, namely around 0 to -55 dBm. Moreover, with the attenuator, we can drop the signal another 26 dB, so we are assured of good linearity from 0 to -81 dBm. But the noise threshold of the radar receiver appears to be around -80 dBm, so for this calibration, we appear to have decent linearity.

3. Set up R code for reading raw samples from the USRP.

We use the *test_usrp_bbprx* utility which forms part of the custom build of gnuradio-based software for the USRP as a radar digitizer. This utility reads sequences of raw samples from any of the USRP's input lines (video, trigger, Heading, Azimuth) at the specified decimation rate. It can also read pulsed data gated by the trigger pulse.

```
getUSRPdata = function(channel=c("vid", "trig", "arp", "acp"), decim=1, gain=0, ns=1024, np=

#' @param channel: which channel to read data from
#'
#' @param decim: rate at which sample clock should be decimated down from 64MHz
#' The effective sampling rate will be 64 / (1 + decim) MHz
#'
#' @param gain: gain setting for channel.  Only for channel = "vid" or "trig";  0 to 20 dB
#'
#' @param ns: number of samples to read
#'
```

```
#' @param np: number of pulses to read, when onTrig = TRUE; otherwise, just multiplies ns.
#'
#' @param onTrig: if TRUE, read ns samples each time a trigger pulse is detected; if FALSE,
#' continuously from the channel.
#'

    usrpDir = "/home/john/proj/ocean-currents/usrp_code/gnuradio-3.3.0/usrp/host/apps"
    trigArgs = "--trig-thresh-excite 31 --trig-thresh-relax 31 -G 10"
    channel = match.arg(channel)
    ## sanity check on parameters; choose channel number
    if (onTrig) {
        if (channel !=  "vid")
          stop("When reading samples gated by a trigger, you can only look at the video char
        chanNum = 0
    } else {
        chanNum = list(vid=1, trig=2, arp=3, acp=4)[[channel]]
    }

    fout = tempfile()
    cmd = "sudo su -c 'cd %s; ./test_usrp_bbprx -d %d -m %d %s %d -n %d -P %d %s %s' root >
        sprintf( usrpDir, decim, chanNum,
        ifelse(channel=="vid", "-g", ifelse(channel=="trig", "-G", "--heading-gain")),
        gain,
        ns, np, if(onTrig) trigArgs else "", fout) %>%
        system
    readBin(fout, integer(), s=2, n=file.info(fout)$size/2, signed=FALSE) %>% return
}
```

Fine tune the signal generator frequency

We use a simple loop to read and plot data from the USRP, watching the plot while adjusting the frequency control on the signal generator to maximize the peak value. This was done only once, to mimic the autotune function on the radar's own receiver.

We then run a loop to grab pulses and plot the peaks, recording values:

```
go = function(...) {
   a = list(...)
   if ("gain" %in% names(a))
       gain = a$gain
   else
       gain = 0
   mm = c()
   while(TRUE) {
      x = matrix(getUSRPdata(np=10, ...), ncol=10)
      ## get the max for each pulse
      mx = apply(x, 2, max)
      mm = c(mx, mm)
      plot(c(x), ylim=c(2047,4096), type="l", main=sprintf("Max Sig: %d +/- %d; Mean of 
      abline(h=c(mean(mx), mean(mx)-sd(mx), mean(mx)+sd(mx)), lty=c(1, 2, 2))
      ## sleep to prevent re-running test_usrp_bbprx so quickly it hangs
      ## waiting on an unclosed USB channel (or something)
      cat(sprintf("%2.0f    %6.1f    %6.1f", gain, mean(mm), sd(mm)), "\n")
      Sys.sleep(1)
   }
}
```

Results by USRP

Eagle

```
eagle = '\
sigdBm  usrpVal  attendB  vidGain
-5.0     3976       0.0    0.0
-7.5     3963       0.0    0.0
-10.0    3950       0.0    0.0
-12.5    3932       0.0    0.0
-15.0    3909       0.0    0.0
-17.5    3870       0.0    0.0
-20.0    3819       0.0    0.0
-22.5    3758       0.0    0.0
-25.0    3684       0.0    0.0
-27.5    3607       0.0    0.0
-30.0    3517       0.0    0.0
-32.5    3428       0.0    0.0
-35.0    3343       0.0    0.0
-37.5    3267       0.0    0.0
-40.0    3187       0.0    0.0
-42.5    3100       0.0    0.0
-45.0    3013       0.0    0.0
-47.5    2932       0.0    0.0
-50.0    2845       0.0    0.0
-52.5    2781       0.0    0.0
-55.0    2715       0.0    0.0
-57.5    2631       0.0    0.0
-60.0    2541       0.0    0.0
-62.5    2452       0.0    0.0
-65.0    2381       0.0    0.0
-67.5    2309       0.0    0.0
```

```
-70.0    2243       0.0    0.0
-72.5    2188       0.0    0.0
-75.0    2144       0.0    0.0
-77.5    2123       0.0    0.0
-80.0    2118       0.0    0.0
-82.5    2114       0.0    0.0
-85.0    2109       0.0    0.0
-87.5    2113       0.0    0.0
-30.0    2659      26.06   0.0
-35.0    2497      26.06   0.0
-40.0    2351      26.06   0.0
-45.0    2216      26.06   0.0
-50.0    2131      26.06   0.0
-55.0    2117      26.06   0.0
-60.0    2113      26.06   0.0
-50.0    2187      26.06   5
-50.0    2284      26.06   10
-50.0    2453      26.06   15
-50.0    2670      26.06   20
-45.0    3447      26.06   20
-45.0    2542      26.06   10
-40.0    3604      26.06   15
-40.0    3775      26.06   16
-40.0    3001      26.06   10
-35.0    3468      26.06   10
-30.0    3970      26.06   10
-27.5    3614      26.06    7
-25.0    3787      26.06    7
-22.5    3982      26.06    7
-22.5    3585      26.06    5
-20.0    3736      26.06    5
-17.5    3895      26.06    5
-15.0    4046      26.06    5
-15.0    3650      26.06    3
-12.5    3761      26.06    3
-10.0    3873      26.06    3
-7.5     3996      26.06    3
-7.5     3780      26.06    2
-5.0     3894      26.06    2
-2.5     3998      26.06    2
0.0      4094      26.06    2\
' %>% textConnection %>% read.table (header=TRUE)
```

We sort these records according to effective incoming signal:

```
    eagle = eagle %>% mutate (indBm = sigdBm - attendB) %>% arrange (indBm)
```

To compare curves with video gain set at different levels, we need to treat the video gain carefully. The scale of USRP values is nominally from 0 to 4095, but this is actually a remapping of the true values coming from the analog to digital converter (ADC), which are in the range -2048 to 2047. The ADC converts incoming voltages in the range -1.0 to +1.0 volts to this integer range, and the usrp plugin then adds -2048 to map the range to [0, 4095]. The gain is a multiplier, scaling voltages in both directions away from zero. A voltage gain of 10 dB multiplies voltages by a factor of 10. So a signal that leads to a USRP value of 2148 when the voltage gain is 0 dB leads to a USRP value of 3048 when the voltage gain is 10 dB. (i.e. the difference between the USRP value and 2048 is multiplied by 10 when the voltage gain is 10). The last complication is that the USRP gain setting is a power gain; i.e. 10 dB represents a 10-times increase in power. Because power scales with the square of voltage, this means a 10 dB power gain is really only a 5 dB gain in voltage (doubling dB values corresponds to squaring the linear value).

So to take video gain setting into account in this calibration, we must undo its effect on the USRP value:

```
   eagle = eagle %>% mutate (adjUsrpVal = 2048 + (usrpVal - 2048) / undB(vidGain / 2))
```

which amounts to removing the “expansion-away-from-zero” caused by the video (power) gain.

If we plot adjusted USRP value against incoming signal value (in dBm), we get an s-shaped curve, which shows good linearity in the mid-scale. The points with positive video gain appear to lie a bit below the overall curve, but it’s hard to be sure as we only looked at weaker signals.

```
 xyplot(adjUsrpVal~indBm, eagle, groups=as.factor(vidGain), auto.key=list(corner=c(0.2,1),ti
```

Finally, we generate an approximation function to map USRP values for a given gain to dBm. First, we remove records for USRP values below 2115, as these are below the noise threshold of the receiver.

```
eagleGood = eagle %>% filter(usrpVal > 2115)

eagle0gainMap = with(eagleGood, approxfun(lowess(adjUsrpVal, indBm, f=0.1)))

eagleUSRPvalTodBm = function(val, gain=0) {
    eagle0gainMap(2048 + (val - 2048) / 10^(gain / 20))
}

## write the full data and smoothed map to an rds file
saveRDS(list(data=eagle, map=eagleUSRPvalTodBm), "eagleUSRPCalibration.rds")
```

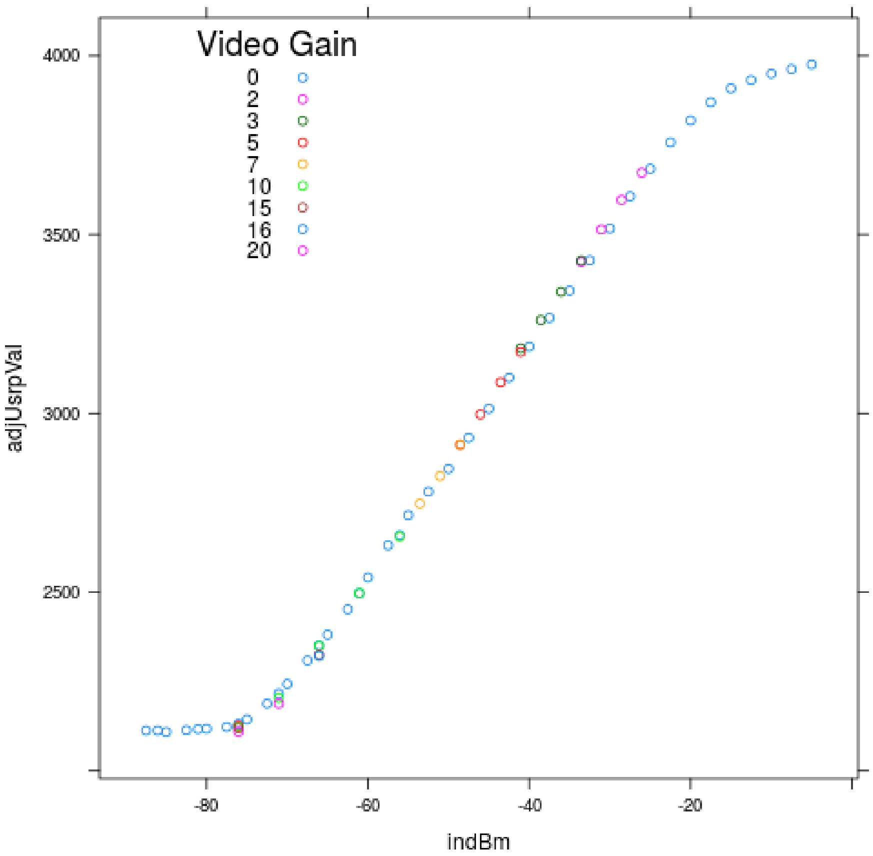
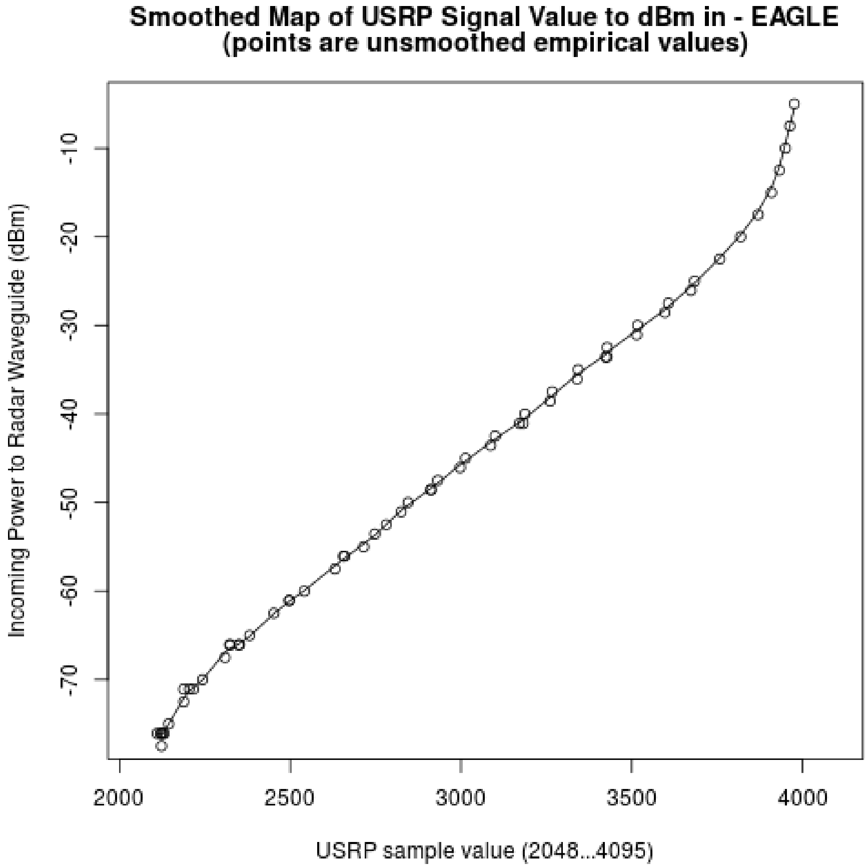


Figure 4: plot of chunk unnamed-chunk-18

```
## demonstrate the curve over the useful USRP sample value range:

usrpRange = 2048:4095

plot(usrpRange, eagleUSRPvalTodBm(usrpRange), main=c("Smoothed Map of USRP Signal Value to
points(eagleGood$adjUsrpVal, eagleGood$indBm)
```



#### Bunting2

We repeat the measurement with Bunting2, this time retaining SDs of USRP values.

```
bunting2 = '\
attendB  sigdBm   vidGain  usrpVal  sdUsrpVal
26.06      0.0       0      3653.8       2.5
26.06     -2.5       0      3577.1       1.6
```

26.06	-5.0	0	3496.0	1.9
26.06	-7.5	0	3406.3	1.4
26.06	-10.0	0	3321.0	1.3
26.06	-12.5	0	3241.8	1.8
26.06	-15.0	0	3164.8	1.6
26.06	-17.5	0	3078.7	2.1
26.06	-20.0	0	2987.3	2.2
26.06	-22.5	0	2903.7	2.6
26.06	-25.0	0	2818.5	3.2
26.06	-27.5	0	2741.6	4.0
26.06	-30.0	0	2659.2	6.1
26.06	-32.5	0	2580.3	6.5
26.06	-35.0	0	2499.7	8.7
26.06	-37.5	0	2422.9	9.0
26.06	-40.0	0	2349.5	11.2
26.06	-42.5	0	2282.9	13.7
26.06	-45.0	0	2220.9	18.7
26.06	-47.5	0	2173.6	20.7
26.06	-50.0	0	2140.4	17.5
26.06	-52.5	0	2124.1	15.6
26.06	-55.0	0	2119.7	15.2
26.06	-57.5	0	2119.2	14.4
26.06	-60.0	0	2118.8	13.2
0.0	-60.0	0	2527.9	8.3
0.0	-62.5	0	2441.9	9.7
0.0	-65.0	0	2370.0	12.6
0.0	-67.5	0	2302.8	14.0
0.0	-70.0	0	2240.4	16.6
0.0	-72.5	0	2185.9	18.5
0.0	-75.0	0	2143.6	17.5
0.0	-77.5	0	2125.6	16.9
0.0	-80.0	0	2123.8	15.3
0.0	-82.5	0	2122.4	17.2
0.0	-85.0	0	2119.2	17.4
0.0	-87.5	0	2115.4	13.3
0.0	-90.0	0	2119.6	17.6
0.0	+0.5	0	3920.5	2.1
0.0	0.0	0	3920.2	2.1
0.0	-2.5	0	3912.6	2.2
0.0	-5.0	0	3899.9	1.9
0.0	-7.5	0	3884.9	2.1
0.0	-10.0	0	3868.2	2.2
0.0	-12.5	0	3852.7	2.8
0.0	-15.0	0	3833.5	2.3
0.0	-17.5	0	3805.2	2.0
0.0	-20.0	0	3763.1	1.5

```
 0.0     -22.5       0      3711.0       1.7
 0.0     -25.0       0      3640.4       1.3
 0.0     -27.5       0      3565.7       1.4
 0.0     -30.0       0      3477.8       1.6
 0.0     -32.5       0      3390.9       1.5
 0.0     -35.0       0      3308.4       1.5
 0.0     -37.5       0      3234.5       1.4
 0.0     -40.0       0      3155.0       1.5
 0.0     -42.5       0      3071.2       2.2
 0.0     -45.0       0      2982.1       2.7
 0.0     -47.5       0      2899.8       2.8
 0.0     -50.0       0      2818.5       3.4
 0.0     -52.5       0      2755.6       3.9
 0.0     -55.0       0      2690.0       5.8
 0.0     -57.5       0      2610.6       6.5
 0.0     -60.0       10     3477.2      21.2
 0.0     -55.0       10     3986.0      15.0
 0.0     -65.0       10     2993.8      32.3
 0.0     -70.0       10     2579.1      46.7
 0.0     -75.0       10     2294.7      60.3
 0.0     -80.0       10     2238.2      49.3
 0.0     -55.0       10     3981.6      12.6
 0.0     -70.0       20     3588.8     173.6
 0.0     -67.5       20     4087.4      23.3
 0.0     -75.0       20     2720.1     176.0
 0.0     -72.5       20     3136.2     170.2
 0.0     -60.0       5      2867.3      17.2
 0.0     -65.0       5      2591.6      22.9
 0.0     -55.0       5      3164.4       9.2
 0.0     -50.0       5      3389.4       5.7
 0.0     -45.0       5      3677.3       4.1
 0.0     -40.0       5      3981.1       4.1
 0.0     -38.0       5      4091.6       2.5\
' %>% textConnection %>% read.table (header=TRUE)

bunting2 = bunting2 %>% mutate (indBm = sigdBm - attendB) %>% arrange (indBm)

bunting2 = bunting2 %>% mutate (adjUsrpVal = 2048 + (usrpVal - 2048) / undB(vidGain / 2))

bunting2Good = bunting2 %>% filter(usrpVal > 2115)

xyplot(adjUsrpVal~indBm,bunting2, groups=as.factor(vidGain), auto.key=list(corner=c(0.2,1),t


bunting20gainMap = with(bunting2Good, approxfun(lowess(adjUsrpVal, indBm, f=0.1)))
```

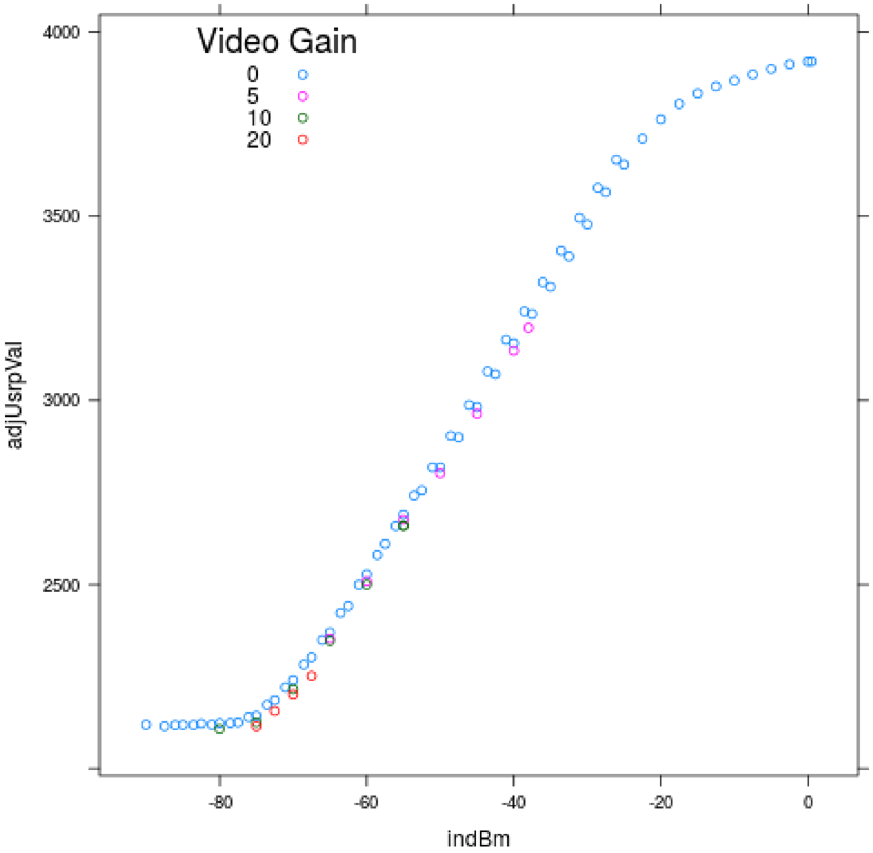


Figure 5: plot of chunk unnamed-chunk-20

```
bunting2USRPvalTodBm = function(val, gain=0) {
    bunting20gainMap(2048 + (val - 2048) / 10^(gain / 20))
}

## write the full data and smoothed map to an rds file
saveRDS(list(data=bunting2, map=bunting2USRPvalTodBm), "bunting2USRPCalibration.rds")

## demonstrate the curve over the useful USRP sample value range:

usrpRange = 2048:4095

plot(usrpRange, bunting2USRPvalTodBm(usrpRange), main=c("Smoothed Map of USRP Signal Value t
points(bunting2Good$adjUsrpVal, bunting2Good$indBm)
```

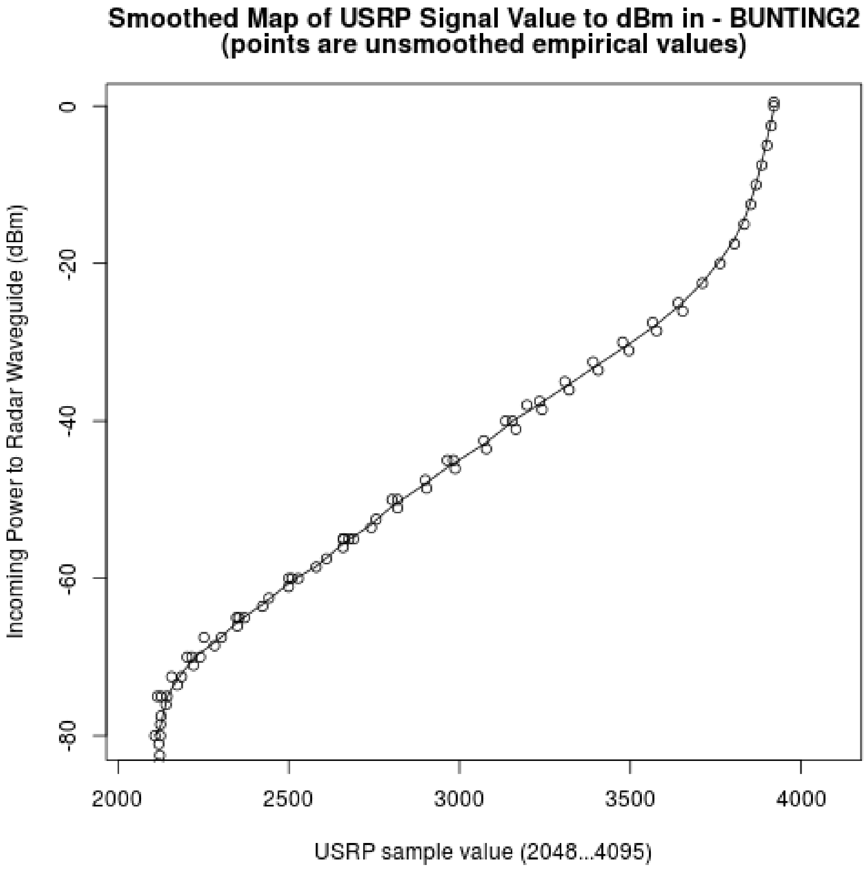


Figure 6: plot of chunk unnamed-chunk-20

Raven We repeat the measurement with Raven.

```
raven = '\
attendB  sigdBm   vidGain  usrpVal  sdUsrpVal
26.06     +0.5       0      3695.9       1.7
26.06      0.0       0      3684.3       1.9
26.06     -2.5       0      3607.6       1.7
26.06     -5.0       0      3522.2       1.6
26.06     -7.5       0      3432.8       1.7
26.06    -10.0       0      3347.7       1.7
26.06    -12.5       0      3268.5       2.0
26.06    -15.0       0      3188.9       1.9
26.06    -17.5       0      3101.3       2.3
26.06    -20.0       0      3009.9       2.7
26.06    -22.5       0      2923.2       2.5
26.06    -25.0       0      2838.3       2.9
26.06    -27.5       0      2761.5       4.1
26.06    -30.0       0      2678.6       5.7
26.06    -32.5       0      2597.1       6.5
26.06    -35.0       0      2512.7       8.6
26.06    -37.5       0      2442.9      10.6
26.06    -40.0       0      2363.4      11.8
26.06    -42.5       0      2296.6      13.7
26.06    -45.0       0      2235.4      17.1
26.06    -47.5       0      2186.4      16.9
26.06    -50.0       0      2150.3      18.6
26.06    -52.5       0      2141.0      17.2
26.06    -55.0       0      2133.3      16.0
26.06    -57.5       0      2132.2      15.3
26.06    -60.0       0      2130.5      14.3
 0.0     -60.0       0      2533.1       8.2
 0.0     -62.5       0      2446.5      11.4
 0.0     -65.0       0      2377.8      11.4
 0.0     -67.5       0      2309.9      13.1
 0.0     -70.0       0      2242.2      16.1
 0.0     -72.5       0      2194.6      20.8
 0.0     -75.0       0      2153.3      20.7
 0.0     -77.5       0      2138.7      13.3
 0.0     -80.0       0      2133.8      15.0
 0.0     -82.5       0      2136.0      16.5
 0.0     -85.0       0      2136.3      15.6
 0.0      +0.5       0      3960.1       2.6
 0.0       0.0       0      3958.3       2.3
 0.0      -2.5       0      3950.7       2.6
 0.0      -5.0       0      3937.1       2.2
 0.0      -7.5       0      3920.7       2.6
 0.0     -10.0       0      3902.1       2.4
 0.0     -12.5       0      3885.1       2.2
 0.0     -15.0       0      3865.7       2.2
 0.0     -17.5       0      3837.1       2.1
 0.0     -20.0       0      3794.9       1.6
 0.0     -22.5       0      3743.0       1.7
 0.0     -25.0       0      3674.4       1.6
 0.0     -27.5       0      3596.9       1.3
 0.0     -30.0       0      3508.6       1.8
 0.0     -32.5       0      3420.9       1.9
 0.0     -35.0       0      3338.4       1.5
 0.0     -37.5       0      3262.0       1.6
 0.0     -40.0       0      3180.6       1.8
 0.0     -42.5       0      3094.0       2.0
 0.0     -45.0       0      3005.8       2.5
 0.0     -47.5       0      2921.7       2.8
 0.0     -50.0       0      2840.6       3.4
 0.0     -52.5       0      2777.8       3.9
 0.0     -55.0       0      2709.9       6.1
 0.0     -57.5       0      2630.5       6.4
 0.0     -60.0       10     3527.9      26.5
 0.0     -55.0       10     4053.1      14.6
 0.0     -65.0       10     3052.5      39.9
 0.0     -70.0       10     2625.2      52.0
 0.0     -75.0       10     2346.5      62.1
 0.0     -80.0       10     2279.1      49.9
 0.0     -55.0       10     4053.5      15.8
 0.0     -70.0       20     3736.8     160.1
 0.0     -68.5       20     4040.8      86.5
 0.0     -75.0       20     2890.8     169.5
 0.0     -72.5       20     3269.1     186.8
 0.0     -60.0       5      2901.0      14.8
 0.0     -65.0       5      2624.9      20.4
 0.0     -55.0       5      3202.0      11.2
 0.0     -50.0       5      3432.8       5.8
 0.0     -45.0       5      3724.1       4.6
 0.0     -40.0       5      4028.8       2.6
 0.0     -39.0       5      4089.1       2.9\
' %>% textConnection %>% read.table (header=TRUE)

raven = raven %>% mutate (indBm = sigdBm - attendB) %>% arrange (indBm)

raven = raven %>% mutate (adjUsrpVal = 2048 + (usrpVal - 2048) / undB(vidGain / 2))

ravenGood = raven %>% filter(usrpVal > 2115)
```

```
xyplot(adjUsrpVal~indBm,raven, groups=as.factor(vidGain), auto.key=list(corner=c(0.2,1),titl
```

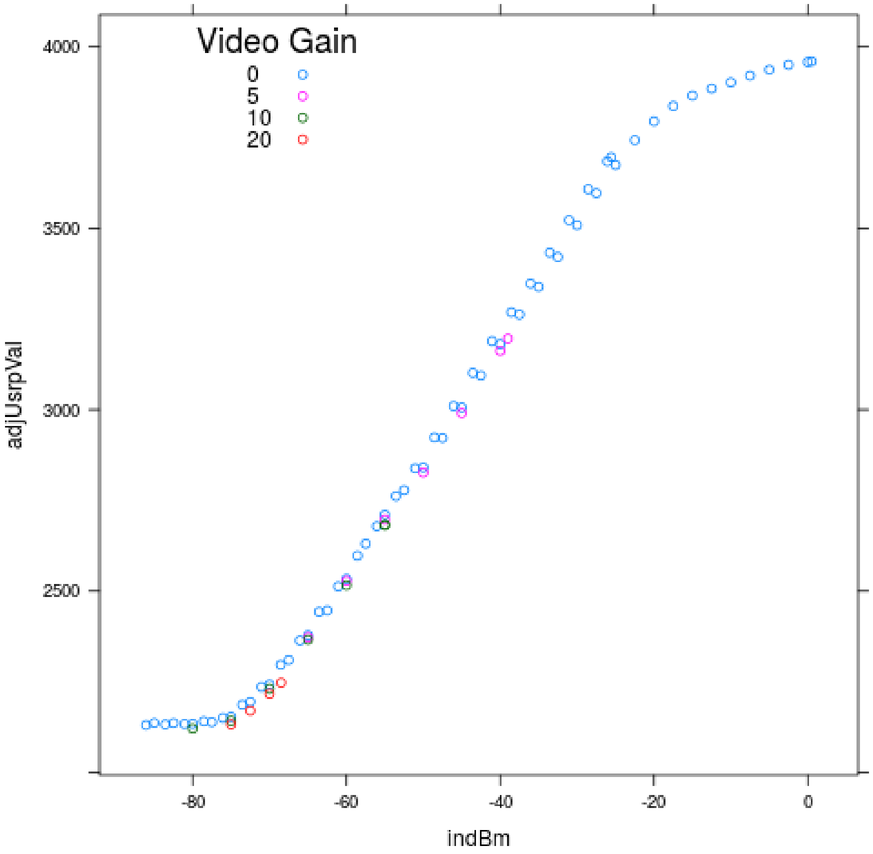


Figure 7: plot of chunk unnamed-chunk-21

```
raven0gainMap = with(ravenGood, approxfun(lowess(adjUsrpVal, indBm, f=0.1)))

ravenUSRPvalTodBm = function(val, gain=0) {
    raven0gainMap(2048 + (val - 2048) / 10^(gain / 20))
}

## write the full data and smoothed map to an rds file
saveRDS(list(data=raven, map=ravenUSRPvalTodBm), "ravenUSRPCalibration.rds")

## demonstrate the curve over the useful USRP sample value range:
```

```
usrpRange = 2048:4095

plot(usrpRange, ravenUSRPvalTodBm(usrpRange), main=c("Smoothed Map of USRP Signal Value to d
points(ravenGood$adjUsrpVal, ravenGood$indBm)
```

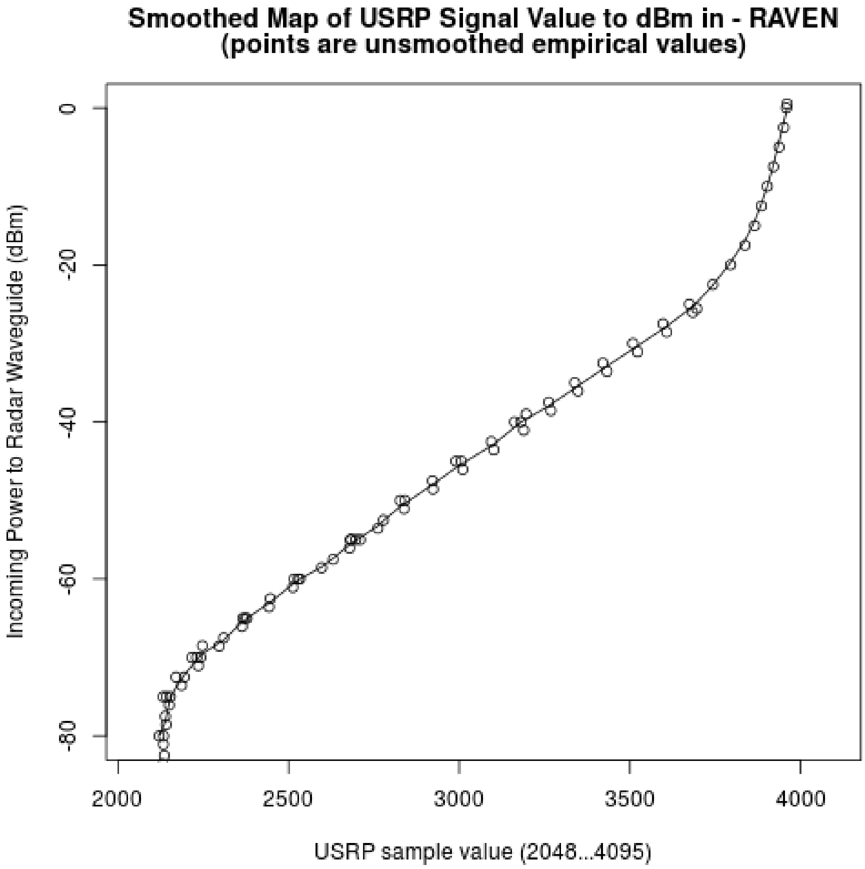


Figure 8: plot of chunk unnamed-chunk-21

Bunting We repeat the measurement with Bunting.

```
bunting = '\
attendB  sigdBm   vidGain  usrpVal  sdUsrpVal
26.06     +0.5       0      3723.8       1.4
26.06      0.0       0      3709.1       1.3
26.06     -2.5       0      3622.9       1.4
```

26.06	-5.0	0	3532.7	1.2
26.06	-7.5	0	3436.9	1.3
26.06	-10.0	0	3348.7	1.3
26.06	-12.5	0	3269.8	1.6
26.06	-15.0	0	3189.0	1.6
26.06	-17.5	0	3099.5	2.3
26.06	-20.0	0	3007.5	2.1
26.06	-22.5	0	2922.0	2.6
26.06	-25.0	0	2834.2	3.7
26.06	-27.5	0	2759.2	4.5
26.06	-30.0	0	2676.9	5.6
26.06	-32.5	0	2593.7	7.5
26.06	-35.0	0	2506.2	9.4
26.06	-37.5	0	2430.2	10.8
26.06	-40.0	0	2352.7	11.8
26.06	-42.5	0	2280.3	16.2
26.06	-45.0	0	2222.4	17.2
26.06	-47.5	0	2171.2	18.9
26.06	-50.0	0	2147.9	17.0
26.06	-52.5	0	2129.5	13.1
26.06	-55.0	0	2130.0	15.2
26.06	-57.5	0	2128.8	16.6
26.06	-60.0	0	2125.9	15.2
0.0	-60.0	0	2537.0	8.5
0.0	-62.5	0	2449.2	9.9
0.0	-65.0	0	2372.2	13.6
0.0	-67.5	0	2303.0	15.1
0.0	-70.0	0	2230.7	15.6
0.0	-72.5	0	2189.1	19.8
0.0	-75.0	0	2151.4	19.6
0.0	-77.5	0	2136.8	16.4
0.0	-80.0	0	2130.6	15.6
0.0	-82.5	0	2126.9	15.0
0.0	-85.0	0	2128.0	15.4
0.0	-7.5	0	4082.1	1.4
0.0	-10.0	0	4057.3	1.8
0.0	-12.5	0	4026.7	1.5
0.0	-15.0	0	3980.8	1.3
0.0	-17.5	0	3920.5	1.5
0.0	-20.0	0	3850.5	1.4
0.0	-22.5	0	3776.9	1.3
0.0	-25.0	0	3695.1	1.3
0.0	-27.5	0	3610.2	1.5
0.0	-30.0	0	3516.6	1.5
0.0	-32.5	0	3426.9	1.2
0.0	-35.0	0	3342.6	1.3

```
 0.0     -37.5      0       3260.7       1.5
 0.0     -40.0      0       3180.5       2.3
 0.0     -42.5      0       3093.8       2.1
 0.0     -45.0      0       3006.3       2.1
 0.0     -47.5      0       2923.7       2.4
 0.0     -50.0      0       2839.2       3.0
 0.0     -52.5      0       2773.5       3.2
 0.0     -55.0      0       2708.4       4.4
 0.0     -57.5      0       2629.5       5.7
 0.0     -60.0      10      3515.1      25.2
 0.0     -55.0      10      4043.4      14.0
 0.0     -65.0      10      3000.4      37.8
 0.0     -70.0      10      2574.2      54.2
 0.0     -75.0      10      2319.0      59.0
 0.0     -80.0      10      2262.2      54.9
 0.0     -55.0      10      4041.6      13.9
 0.0     -70.0      20      3599.8     172.6
 0.0     -68.5      20      3936.7     144.7
 0.0     -75.0      20      2782.0     156.5
 0.0     -72.5      20      3145.0     140.0
 0.0     -60.0      5       2891.1      15.9
 0.0     -65.0      5       2600.1      20.5
 0.0     -55.0      5       3196.7       8.2
 0.0     -50.0      5       3426.0       6.1
 0.0     -45.0      5       3724.5       4.7
 0.0     -40.0      5       4029.6       2.9
 0.0     -39.0      5       4089.2       2.6\
' %>% textConnection %>% read.table (header=TRUE)

bunting = bunting %>% mutate (indBm = sigdBm - attendB) %>% arrange (indBm)

bunting = bunting %>% mutate (adjUsrpVal = 2048 + (usrpVal - 2048) / undB(vidGain / 2))

buntingGood = bunting %>% filter(usrpVal > 2115)

xyplot(adjUsrpVal~indBm,bunting, groups=as.factor(vidGain), auto.key=list(corner=c(0.2,1),ti


bunting0gainMap = with(buntingGood, approxfun(lowess(adjUsrpVal, indBm, f=0.1)))

buntingUSRPvalTodBm = function(val, gain=0) {
    bunting0gainMap(2048 + (val - 2048) / 10^(gain / 20))
}

## write the full data and smoothed map to an rds file
saveRDS(list(data=bunting, map=buntingUSRPvalTodBm), "buntingUSRPCalibration.rds")
```

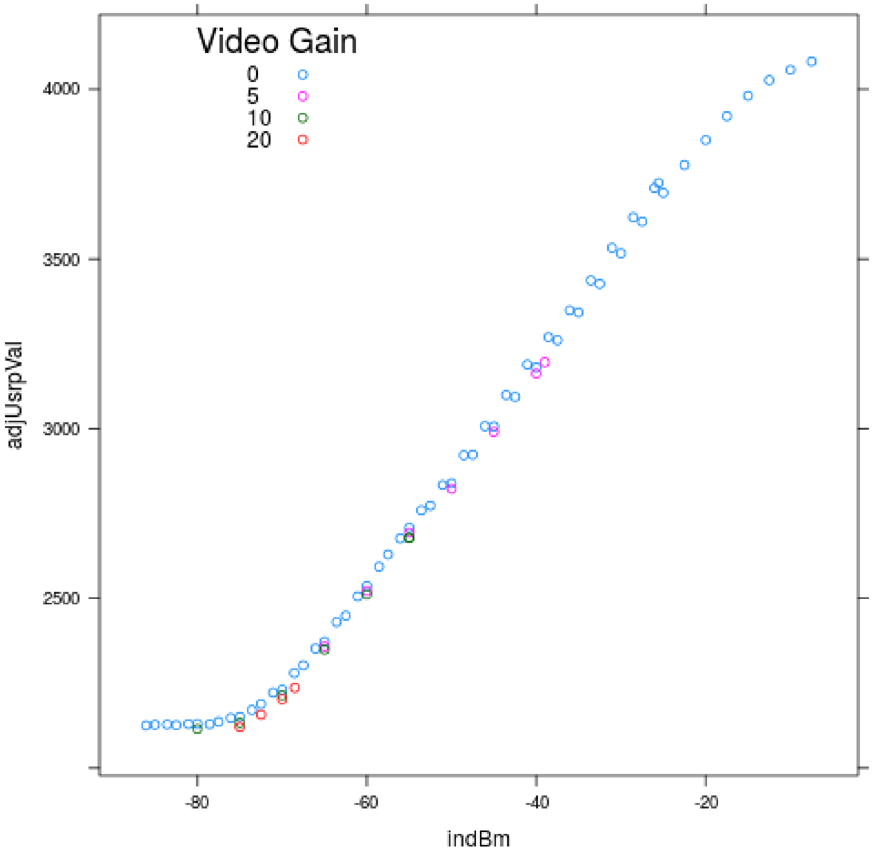


Figure 9: plot of chunk unnamed-chunk-22

```
## demonstrate the curve over the useful USRP sample value range:

usrpRange = 2048:4095

plot(usrpRange, buntingUSRPvalTodBm(usrpRange), main=c("Smoothed Map of USRP Signal Value to
points(buntingGood$adjUsrpVal, buntingGood$indBm)
```

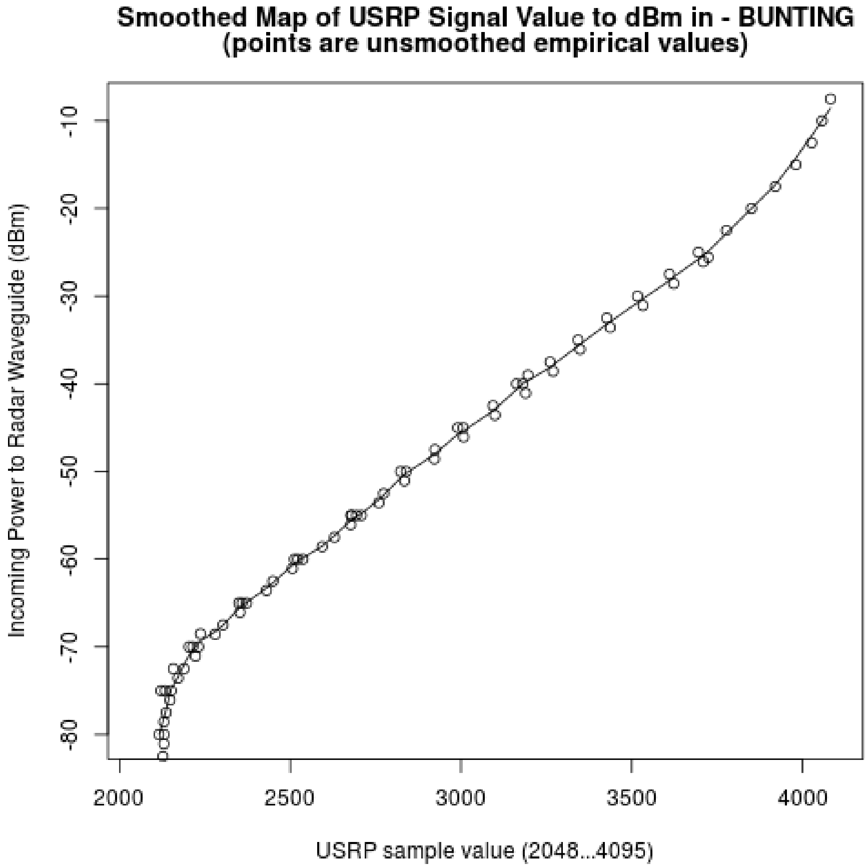


Figure 10: plot of chunk unnamed-chunk-22

Plover We repeat the measurement with Plover.

```
plover = '\
attendB  sigdBm   vidGain  usrpVal  sdUsrpVal
```

26.06	+0.5	0	3731.8	1.4
26.06	0.0	0	3717.6	1.8
26.06	-2.5	0	3635.2	1.3
26.06	-5.0	0	3546.6	1.3
26.06	-7.5	0	3455.0	1.2
26.06	-10.0	0	3365.4	1.5
26.06	-12.5	0	3286.3	1.5
26.06	-15.0	0	3205.6	1.7
26.06	-17.5	0	3117.5	1.9
26.06	-20.0	0	3023.9	2.0
26.06	-22.5	0	2938.7	3.0
26.06	-25.0	0	2851.4	3.2
26.06	-27.5	0	2772.2	4.8
26.06	-30.0	0	2692.9	5.1
26.06	-32.5	0	2608.8	8.4
26.06	-35.0	0	2526.9	9.1
26.06	-37.5	0	2447.4	10.7
26.06	-40.0	0	2373.6	13.5
26.06	-42.5	0	2308.5	15.7
26.06	-45.0	0	2246.5	17.0
26.06	-47.5	0	2191.9	23.2
26.06	-50.0	0	2151.7	16.5
26.06	-52.5	0	2140.3	15.5
26.06	-55.0	0	2138.4	17.5
26.06	-57.5	0	2139.8	17.1
26.06	-60.0	0	2139.7	18.0
0.0	-60.0	0	2571.3	8.6
0.0	-62.5	0	2481.3	8.4
0.0	-65.0	0	2407.3	12.5
0.0	-67.5	0	2335.6	12.8
0.0	-70.0	0	2266.9	14.2
0.0	-72.5	0	2213.9	19.9
0.0	-75.0	0	2165.2	19.8
0.0	-77.5	0	2145.4	17.0
0.0	-80.0	0	2135.8	17.3
0.0	-82.5	0	2137.5	15.3
0.0	+0.5	0	4034.9	1.9
0.0	0.0	0	4033.1	2.2
0.0	-2.5	0	4024.6	2.1
0.0	-5.0	0	4014.0	1.8
0.0	-7.5	0	4000.1	1.7
0.0	-10.0	0	3985.1	2.0
0.0	-12.5	0	3968.5	2.1
0.0	-15.0	0	3944.5	1.8
0.0	-17.5	0	3905.9	1.5
0.0	-20.0	0	3855.1	1.5

```
 0.0     -22.5      0      3794.9       1.4
 0.0     -25.0      0      3720.4       1.9
 0.0     -27.5      0      3720.4       1.9
 0.0     -30.0      0      3551.7       1.6
 0.0     -32.5      0      3462.3       1.9
 0.0     -35.0      0      3374.8       1.6
 0.0     -37.5      0      3295.5       1.7
 0.0     -40.0      0      3215.1       1.5
 0.0     -42.5      0      3131.6       1.9
 0.0     -45.0      0      3041.4       2.2
 0.0     -47.5      0      3041.4       2.2
 0.0     -50.0      0      2873.6       2.9
 0.0     -52.5      0      2806.3       3.4
 0.0     -55.0      0      2739.3       4.4
 0.0     -57.5      0      2664.7       6.5
 0.0     -60.0      10     3619.2      21.9
 0.0     -57.5      10     3895.4      18.8
 0.0     -65.0      10     3118.8      37.4
 0.0     -70.0      10     2669.0      48.5
 0.0     -75.0      10     2372.8      54.9
 0.0     -80.0      10     2272.5      33.4
 0.0     -70.0      20     3914.2     139.7
 0.0     -68.5      20     4089.7      24.5
 0.0     -75.0      20     2975.5     210.2
 0.0     -72.5      20     3384.3     187.1
 0.0     -60.0      5      2958.9      11.9
 0.0     -65.0      5      2669.8      22.5
 0.0     -55.0      5      3255.6       7.4
 0.0     -50.0      5      3488.2       5.3
 0.0     -45.0      5      3784.8       4.6
 0.0     -40.0      5      4087.9       2.8\
' %>% textConnection %>% read.table (header=TRUE)

plover = plover %>% mutate (indBm = sigdBm - attendB) %>% arrange (indBm)

plover = plover %>% mutate (adjUsrpVal = 2048 + (usrpVal - 2048) / undB(vidGain / 2))

ploverGood = plover %>% filter(usrpVal > 2115)

xyplot(adjUsrpVal~indBm,plover, groups=as.factor(vidGain), auto.key=list(corner=c(0.2,1),tit


plover0gainMap = with(ploverGood, approxfun(lowess(adjUsrpVal, indBm, f=0.1)))

ploverUSRPvalTodBm = function(val, gain=0) {
    plover0gainMap(2048 + (val - 2048) / 10^(gain / 20))
```

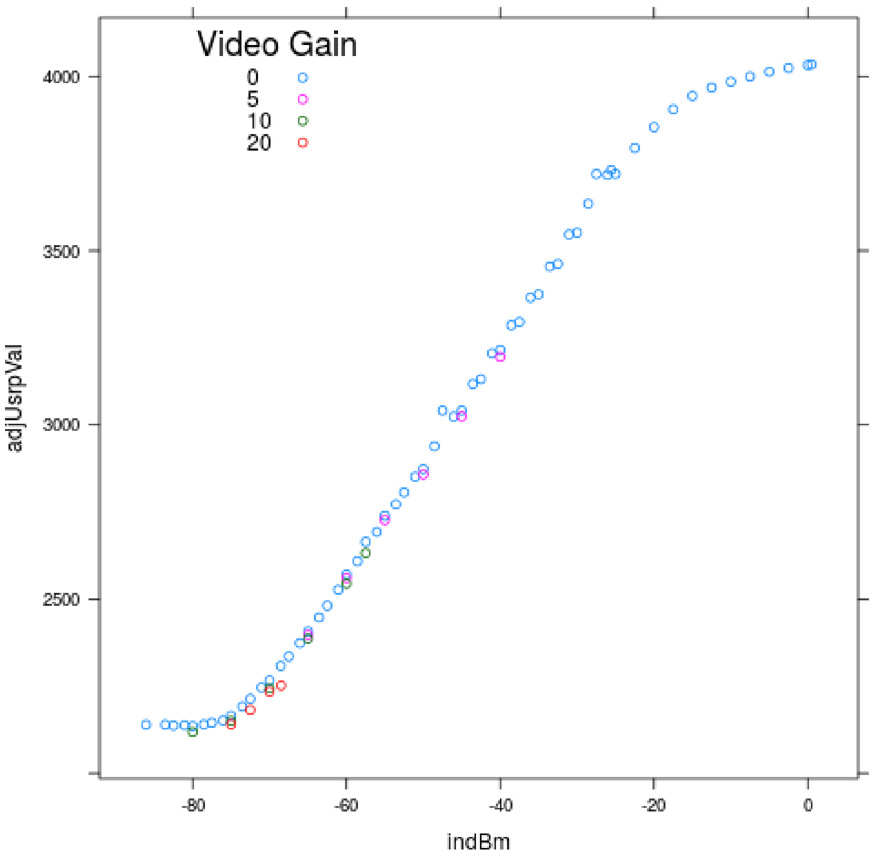


Figure 11: plot of chunk unnamed-chunk-23

```
}

## write the full data and smoothed map to an rds file
saveRDS(list(data=plover, map=ploverUSRPvalTodBm), "ploverUSRPCalibration.rds")

## demonstrate the curve over the useful USRP sample value range:

usrpRange = 2048:4095

plot(usrpRange, ploverUSRPvalTodBm(usrpRange), main=c("Smoothed Map of USRP Signal Value to
points(ploverGood$adjUsrpVal, ploverGood$indBm)
```

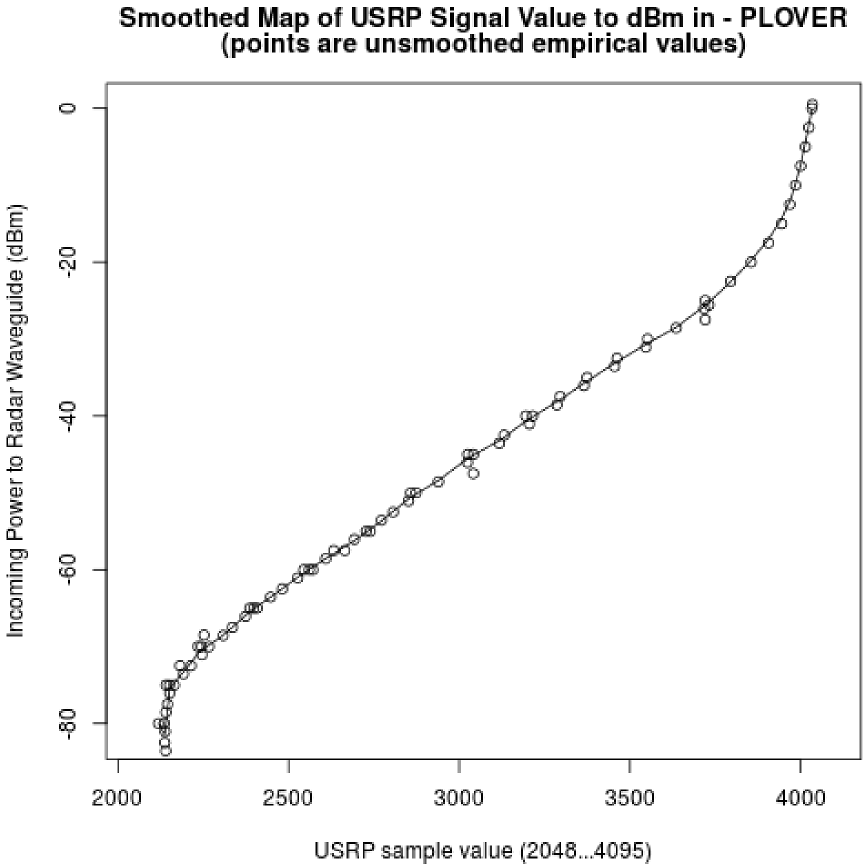


Figure 12: plot of chunk unnamed-chunk-23

Phoebe We repeat the measurement with Phoebe.

```
phoebe = '\
attendB  sigdBm   vidGain  usrpVal  sdUsrpVal
26.06     +0.5       0      3721.0       1.4
26.06      0.0       0      3704.0       4.5
26.06     -2.5       0      3623.8       1.5
26.06     -5.0       0      3536.8       1.4
26.06     -7.5       0      3445.4       1.3
26.06    -10.0       0      3357.6       1.5
26.06    -12.5       0      3276.2       1.5
26.06    -15.0       0      3196.4       2.4
26.06    -17.5       0      3109.2       1.9
26.06    -20.0       0      3017.6       2.9
26.06    -22.5       0      2929.7       2.7
26.06    -25.0       0      2842.6       3.1
26.06    -27.5       0      2765.0       4.4
26.06    -30.0       0      2683.6       4.8
26.06    -32.5       0      2599.6       6.2
26.06    -35.0       0      2518.5       8.0
26.06    -37.5       0      2441.3      10.8
26.06    -40.0       0      2363.1       9.3
26.06    -42.5       0      2290.3      15.6
26.06    -45.0       0      2230.7      19.5
26.06    -47.5       0      2177.7      21.2
26.06    -50.0       0      2141.8      19.1
26.06    -52.5       0      2128.0      20.9
26.06    -55.0       0      2124.9      15.6
26.06    -57.5       0      2119.8      16.8
26.06    -60.0       0      2116.8      14.0
 0.0     -60.0       0      2558.0       6.3
 0.0     -62.5       0      2471.3       8.4
 0.0     -65.0       0      2400.8      12.0
 0.0     -67.5       0      2326.0      15.5
 0.0     -70.0       0      2254.9      17.7
 0.0     -72.5       0      2197.8      17.7
 0.0     -75.0       0      2154.1      21.1
 0.0     -77.5       0      2127.7      13.3
 0.0     -80.0       0      2119.4      15.3
 0.0     -82.5       0      2123.3      15.3
 0.0      +0.5       0      4023.1       2.0
 0.0       0.0       0      4023.3       2.0
 0.0      -2.5       0      4013.2       2.0
 0.0      -5.0       0      4003.4       2.1
 0.0      -7.5       0      3990.0       1.8
 0.0     -10.0       0      3974.2       1.8
 0.0     -12.5       0      3956.0       1.5
 0.0     -15.0       0      3931.5       1.6
 0.0     -17.5       0      3892.5       1.8
 0.0     -20.0       0      3840.4       1.3
 0.0     -22.5       0      3779.4       1.4
 0.0     -25.0       0      3706.9       1.3
 0.0     -27.5       0      3626.5       1.3
 0.0     -30.0       0      3538.9       1.6
 0.0     -32.5       0      3449.3       1.6
 0.0     -35.0       0      3361.4       1.5
 0.0     -37.5       0      3284.5       1.5
 0.0     -40.0       0      3203.5       1.8
 0.0     -42.5       0      3117.4       1.7
 0.0     -45.0       0      3031.7       2.5
 0.0     -47.5       0      2948.2       2.6
 0.0     -50.0       0      2861.7       3.7
 0.0     -52.5       0      2794.4       3.1
 0.0     -55.0       0      2729.0       4.4
 0.0     -57.5       0      2650.1       5.8
 0.0     -60.0       10     3592.1      19.0
 0.0     -57.5       10     3872.4      14.4
 0.0     -65.0       10     3093.4      28.4
 0.0     -70.0       10     2638.8      35.3
 0.0     -75.0       10     2350.5      69.1
 0.0     -80.0       10     2233.6      40.4
 0.0     -70.0       20     3865.1     156.8
 0.0     -68.5       20     4085.0      34.0
 0.0     -75.0       20     2857.6     193.7
 0.0     -72.5       20     3312.2     181.1
 0.0     -60.0       5      2931.8      14.5
 0.0     -65.0       5      2651.5      20.4
 0.0     -55.0       5      3234.8       7.7
 0.0     -50.0       5      3469.6       5.0
 0.0     -45.0       5      3765.7       3.6
 0.0     -40.0       5      4068.3       3.1\
' %>% textConnection %>% read.table (header=TRUE)

phoebe = phoebe %>% mutate (indBm = sigdBm - attendB) %>% arrange (indBm)

phoebe = phoebe %>% mutate (adjUsrpVal = 2048 + (usrpVal - 2048) / undB(vidGain / 2))

phoebeGood = phoebe %>% filter(usrpVal > 2115)

xyplot(adjUsrpVal~indBm,phoebe, groups=as.factor(vidGain), auto.key=list(corner=c(0.2,1),tit


phoebe0gainMap = with(phoebeGood, approxfun(lowess(adjUsrpVal, indBm, f=0.1)))
```

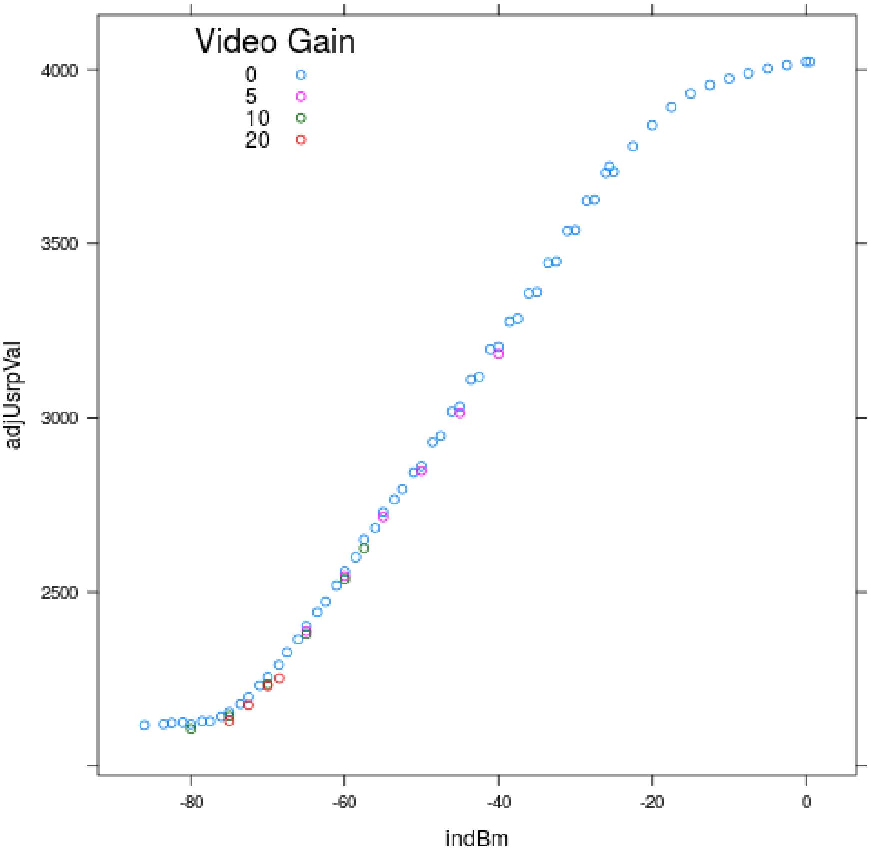


Figure 13: plot of chunk unnamed-chunk-24

```
phoebeUSRPvalTodBm = function(val, gain=0) {
    phoebe0gainMap(2048 + (val - 2048) / 10^(gain / 20))
}

## write the full data and smoothed map to an rds file
saveRDS(list(data=phoebe, map=phoebeUSRPvalTodBm), "phoebeUSRPCalibration.rds")

## demonstrate the curve over the useful USRP sample value range:

usrpRange = 2048:4095

plot(usrpRange, phoebeUSRPvalTodBm(usrpRange), main=c("Smoothed Map of USRP Signal Value to
points(phoebeGood$adjUsrpVal, phoebeGood$indBm)
```

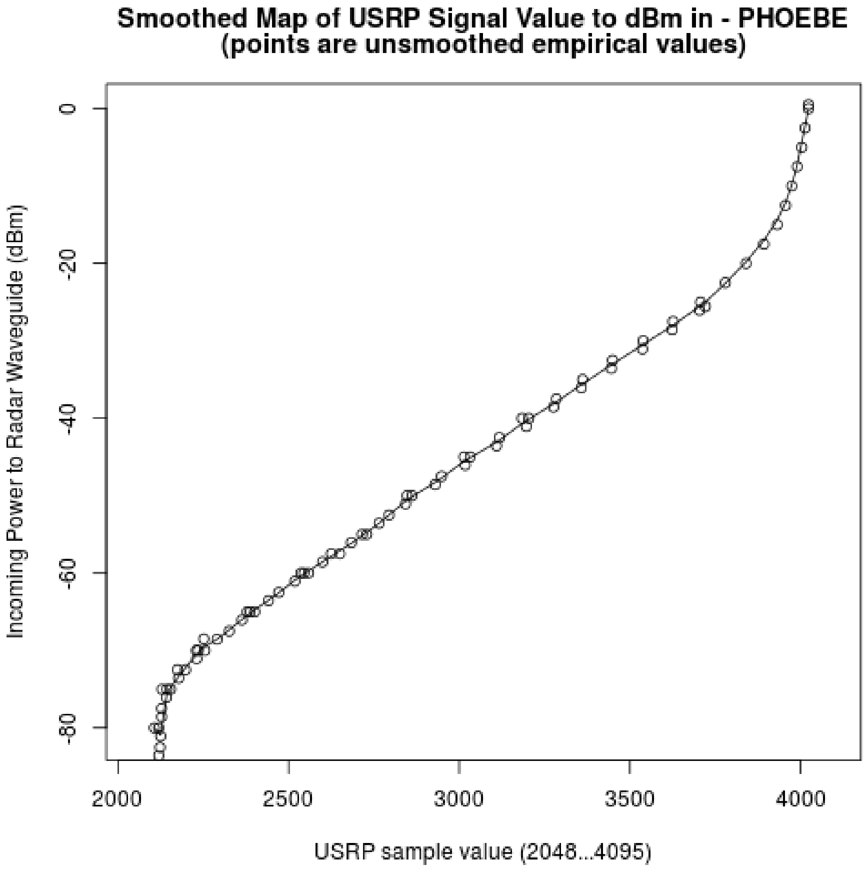


Figure 14: plot of chunk unnamed-chunk-24