Tire Bridging Algorithm

Brian Michalk, Texas DOT
RPUG 2006
Des Moines, Iowa
September 22, 2006

Overview

- References and definitions
- Filtering
- Experiment
- Tire Bridge Algorithm

Bandwidth

- A measure of frequency range and is typically measured in hertz. Bandwidth is a central concept in many fields, including information theory, radio communications, signal processing, and spectroscopy. Bandwidth also refers to data rates when communicating over certain media or devices.
- **Explanation**
- There is no *single* universal precise definition of bandwidth, as it is vaguely understood to be a measure of how wide a function is in the frequency domain.
- According to the Shannon–Hartley theorem, the data rate of reliable communication is directly proportional to the frequency range of the signal used for the communication. In this context, the word bandwidth can refer to either the data rate or the frequency range of the communication system (or both).
- **Bandwidth of a signal is a measure of how rapidly its parameters (e.g. amplitude and phase) fluctuate with respect to time. Hence, the greater the bandwidth, the faster the variation in the signal parameters may be.**

References

- Little Book Of Profiling
- The Scientist and Engineer's Guide to Digital Signal Processing
  - http://www.dspguide.com

PSD (Position Sensitive Device)

- A single axis PSD device has two anode connections, Y1 and Y2. The beam's position is calculated from the ratios of these signals.
- Assume Y is the coordinate of the **centroid** of the beam spot on the PSD surface; and L is the size of the detector's sensitive surface, then the distance is reported as: $Y=((Y1-Y2)/(Y1+Y2))/(L/2)$

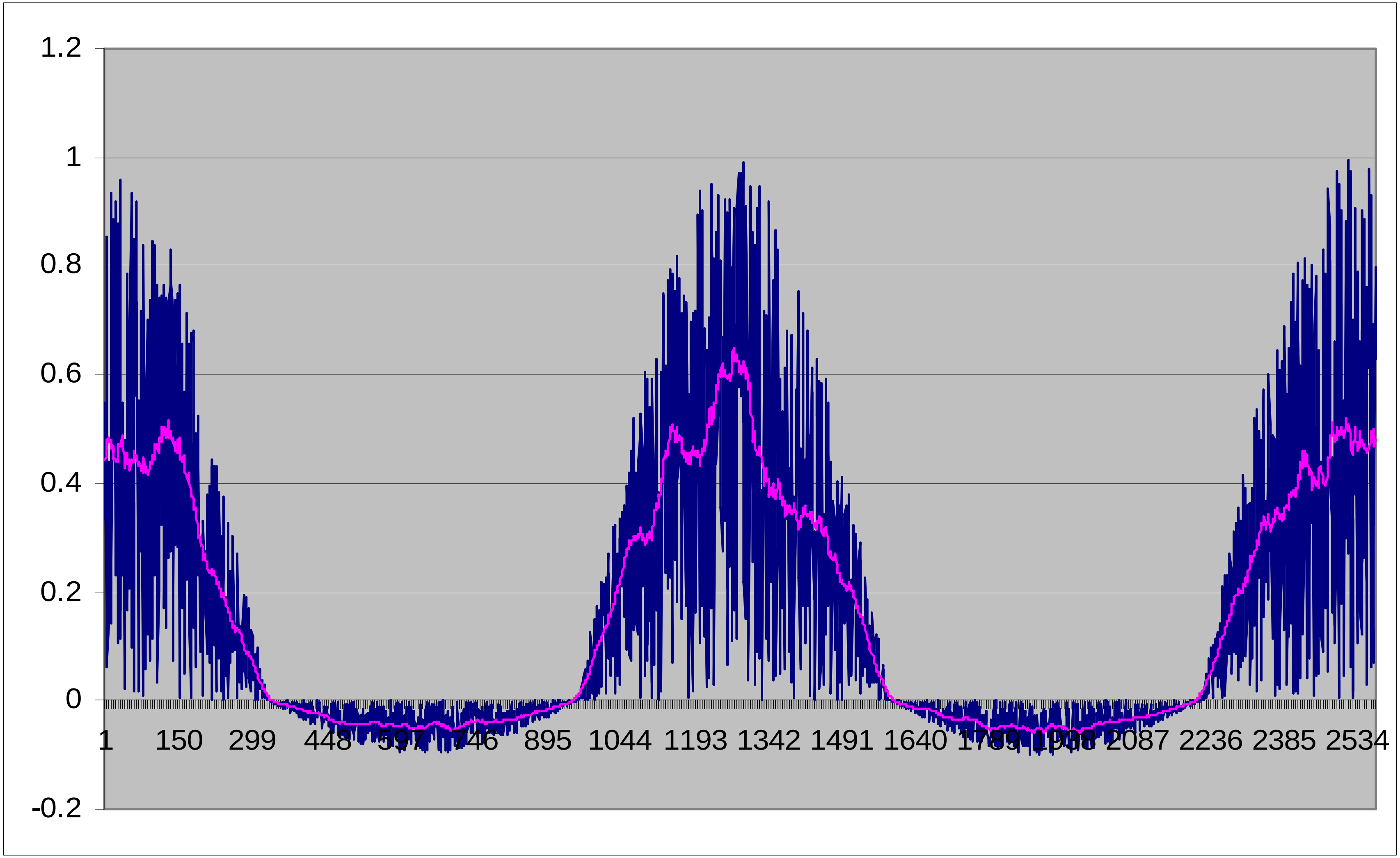
Texture Issues (bandwidth)

- Why do we care about texture?
- Want to “remove texture”.
  - Reduce the bandwidth.
    - Modify spot size
      Grade 3 aggregate has a maximum particle size of 5/8 inch. Requires a laser spot size of greater than 1.25 inches.
    - Filter the laser data
      Filter with low pass at 1.25 inches
    - What filter length?
      - Mean Profile Depth (MPD) has about 50mm bandwidth
      - IRI passes wavelengths longer than 250mm
  - Acquire texture to deal with it later.
    Requires higher bandwidth.
    - Properly characterize 6mm features. Requires higher bandwidth, about 3mm. Preferably 1mm. (32Khz at 50mph = 1.4samples/mm)
- What speed of laser? Speak in bandwidth.
  - Some lasers have report interval different from bandwidth
  - Should be stated in specifications at a reference velocity. Instead of 16Khz, minimum, should state:
    - Samples per cm?
    - Bandwidth in distance?

Filters

- Anti-alias filters
  - Determined not by laser, but by analog acquisition system
  - Not required for digital systems
  - Should be at least half the sampling frequency
- Texture Filters
  - Standard filters destroy information in the signal that could be used to remove texture
  - Fix noise by other non-filter methods.
    - Limit slew rate to expected amplitude of texture
- Analog lasers should be sampled at least the frequency of the laser.

Averaging Texture

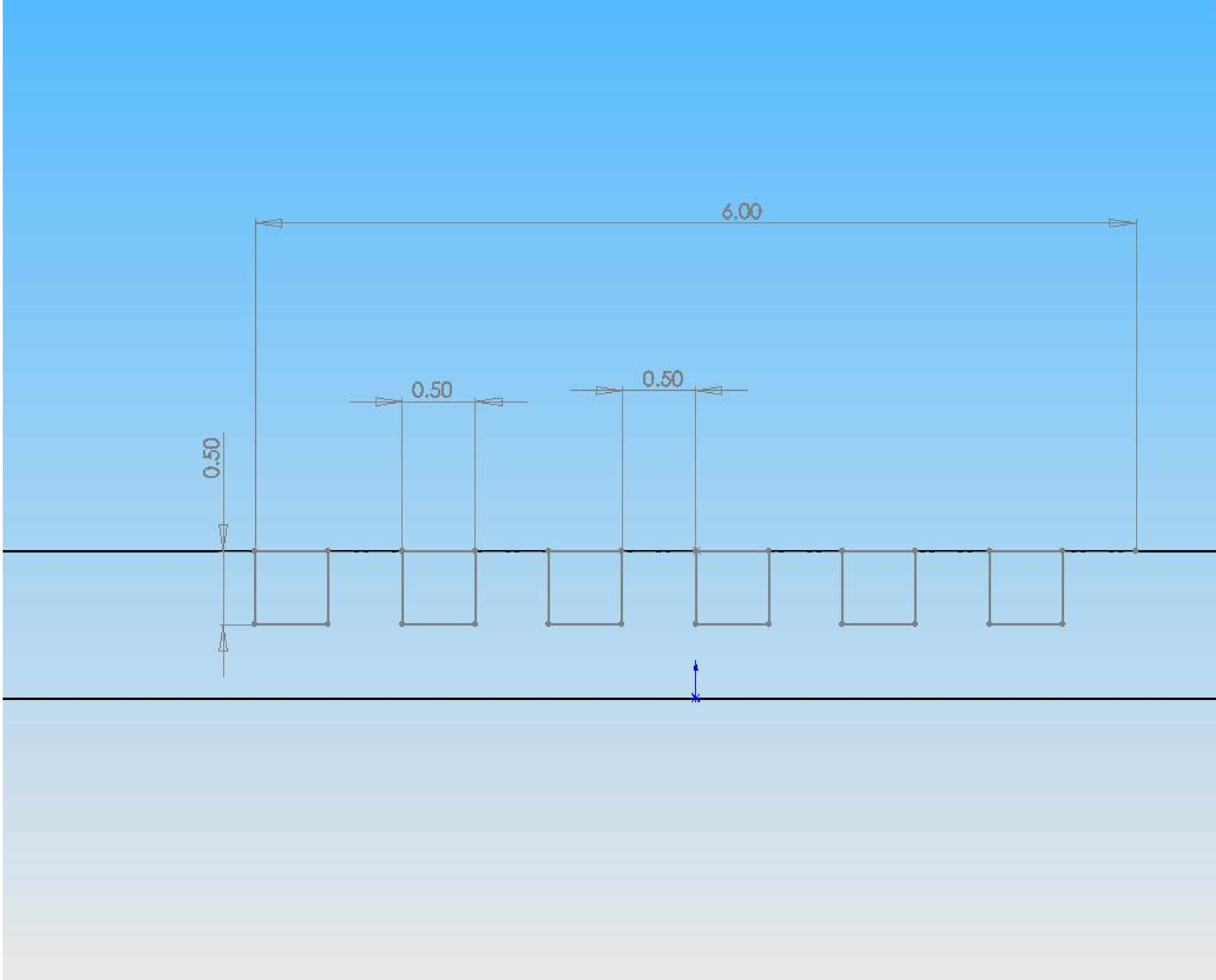


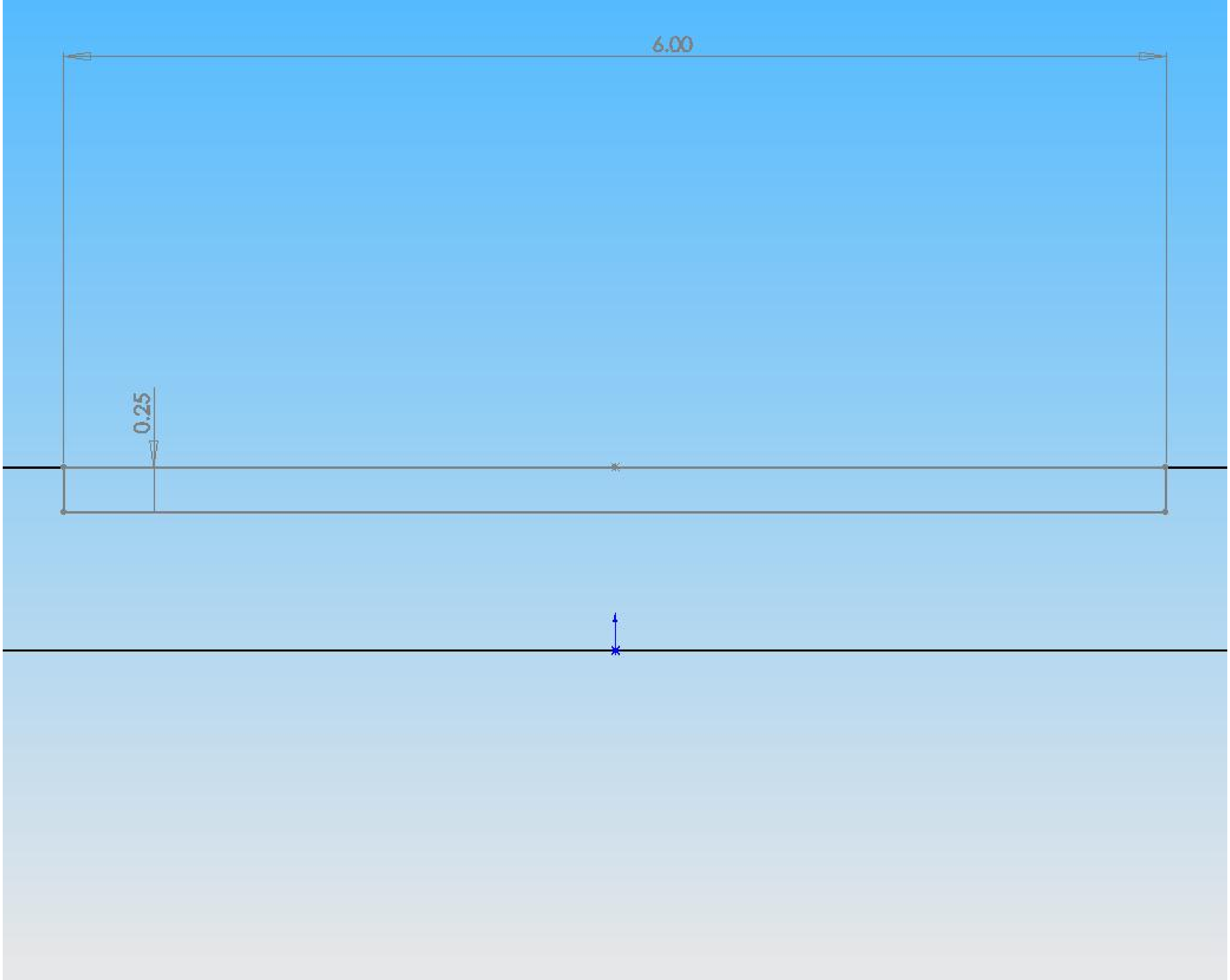
Experiment

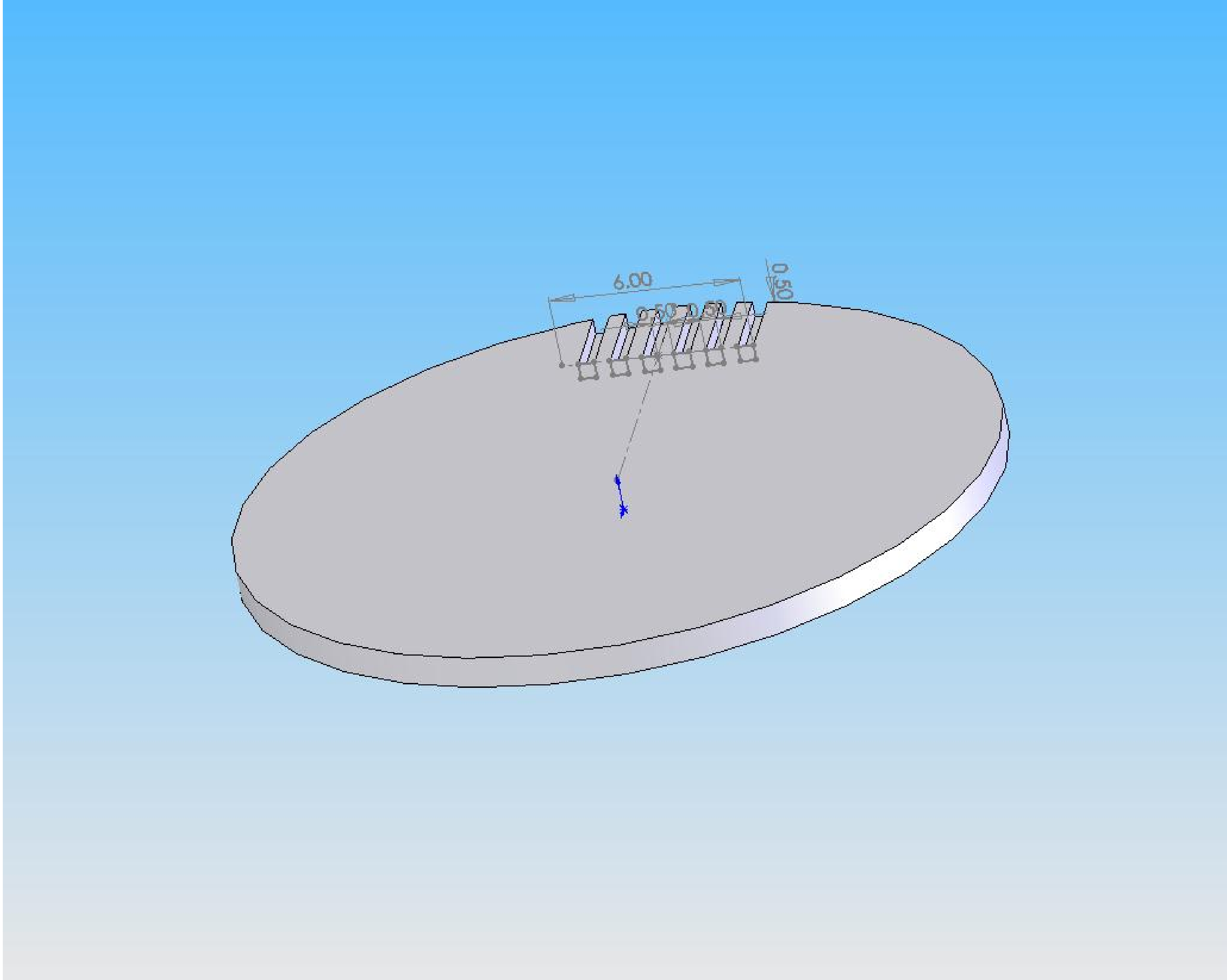
- Purpose
  - Determine effect of variation of texture on IRI
- Hypothesis
  - Analog filtered data for pothole and tined traces will result in identical IRI values.
- Conclusion
  - Tined surfaces cause a tire to contact the top 50% of the road profile. Any deviation from this behavior will result in IRI errors when encountered repetitively.
- Conditions
  - Two disks, each with circumference of 2.2 meters
  - Mounted to turntable
  - Assume constant 1G acceleration
  - One sample per millimeter
  - 100Hz 8pole analog filter

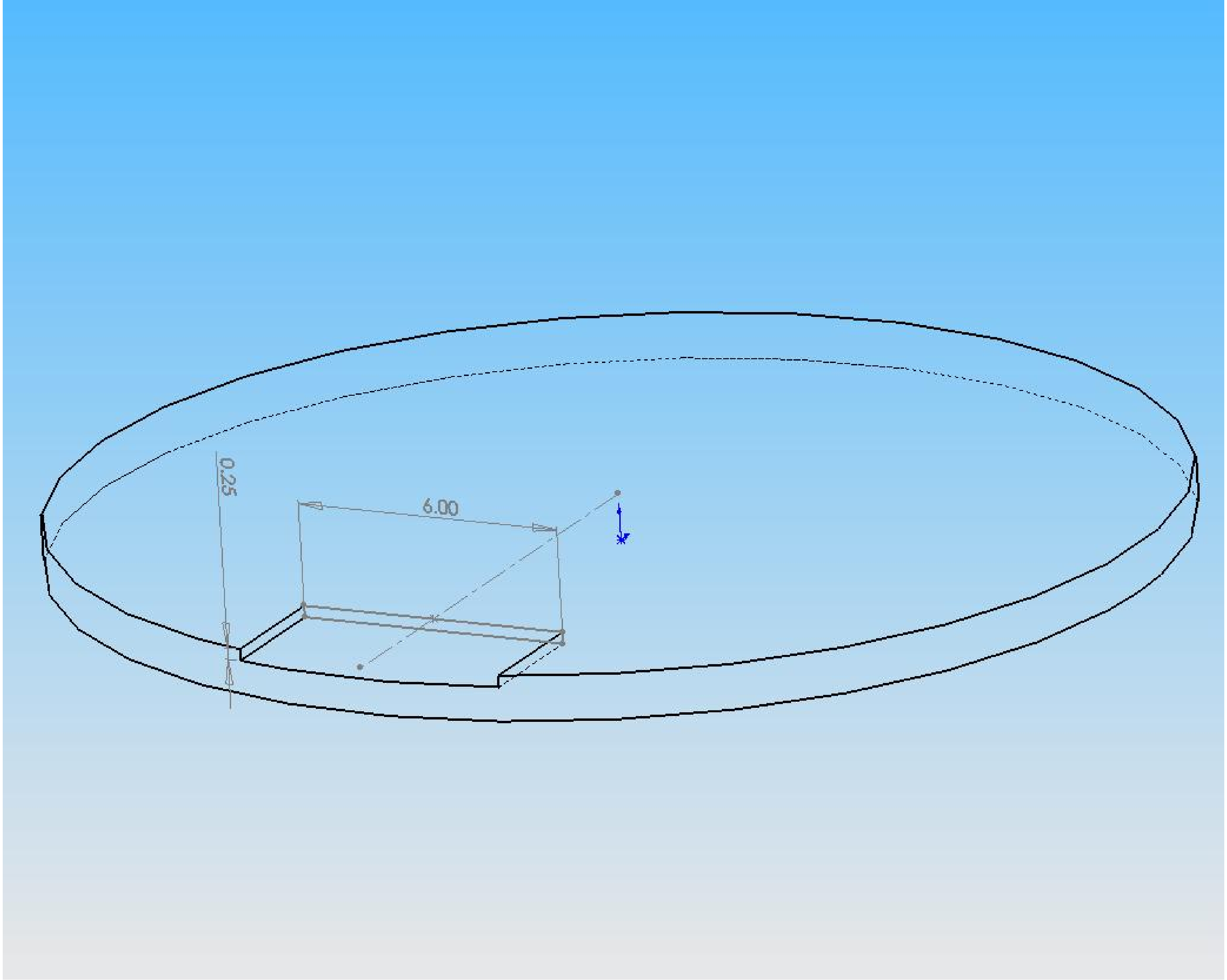
Experiment (Cont)

- Test cases
  - 50mph, .5” tined, 0” radius wheel (no wheel)
  - 3mph, .5” tined, 0” wheel
  - 3mph, .5” tined, 6” wheel
  - 50mph, .25” pothole, 0” wheel
  - 3mph, .25” pothole, 0” wheel
  - 3mph, .25” pothole, 6” wheel


6.00
0.50
0.50
0.50

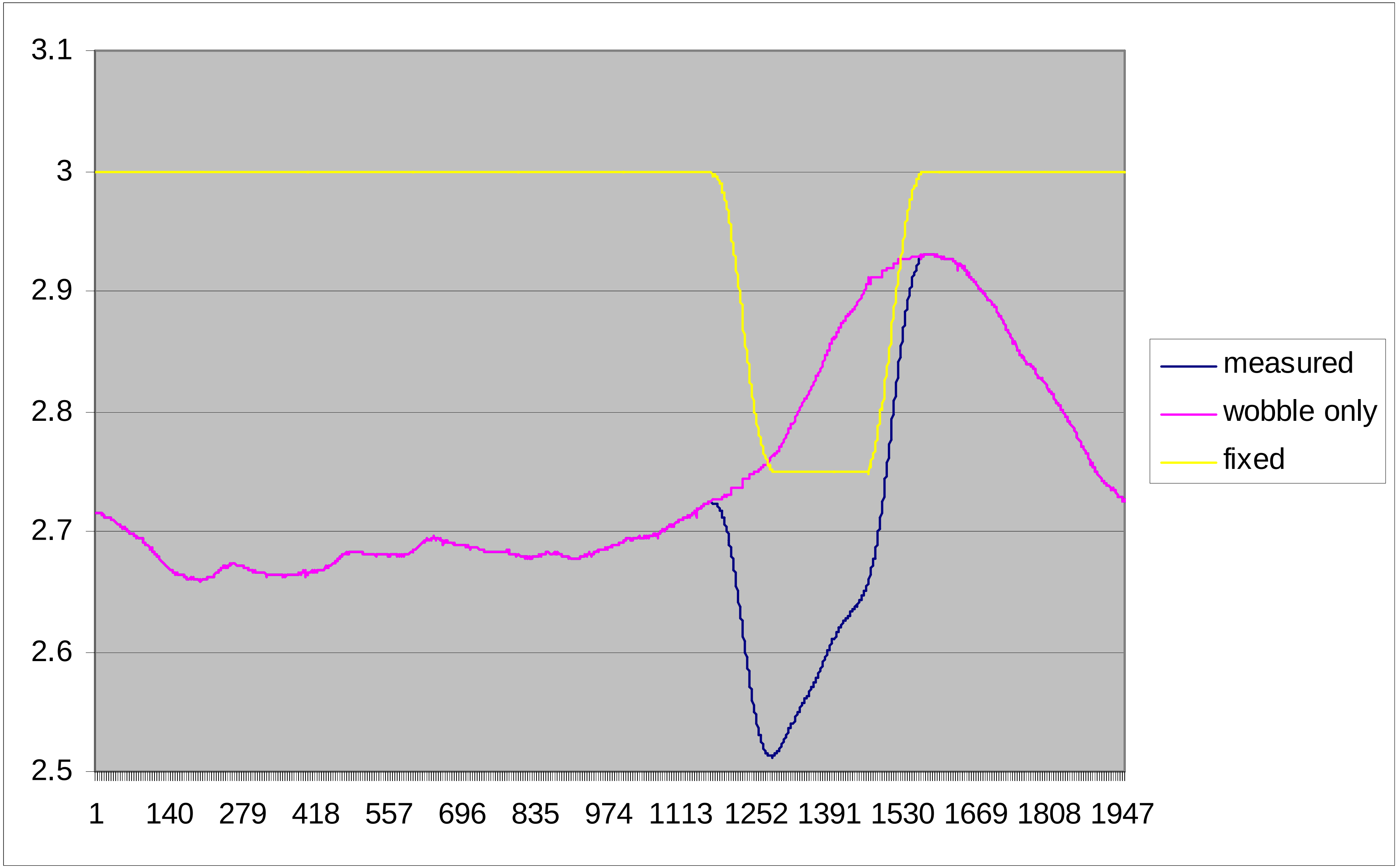

6.00
0.25

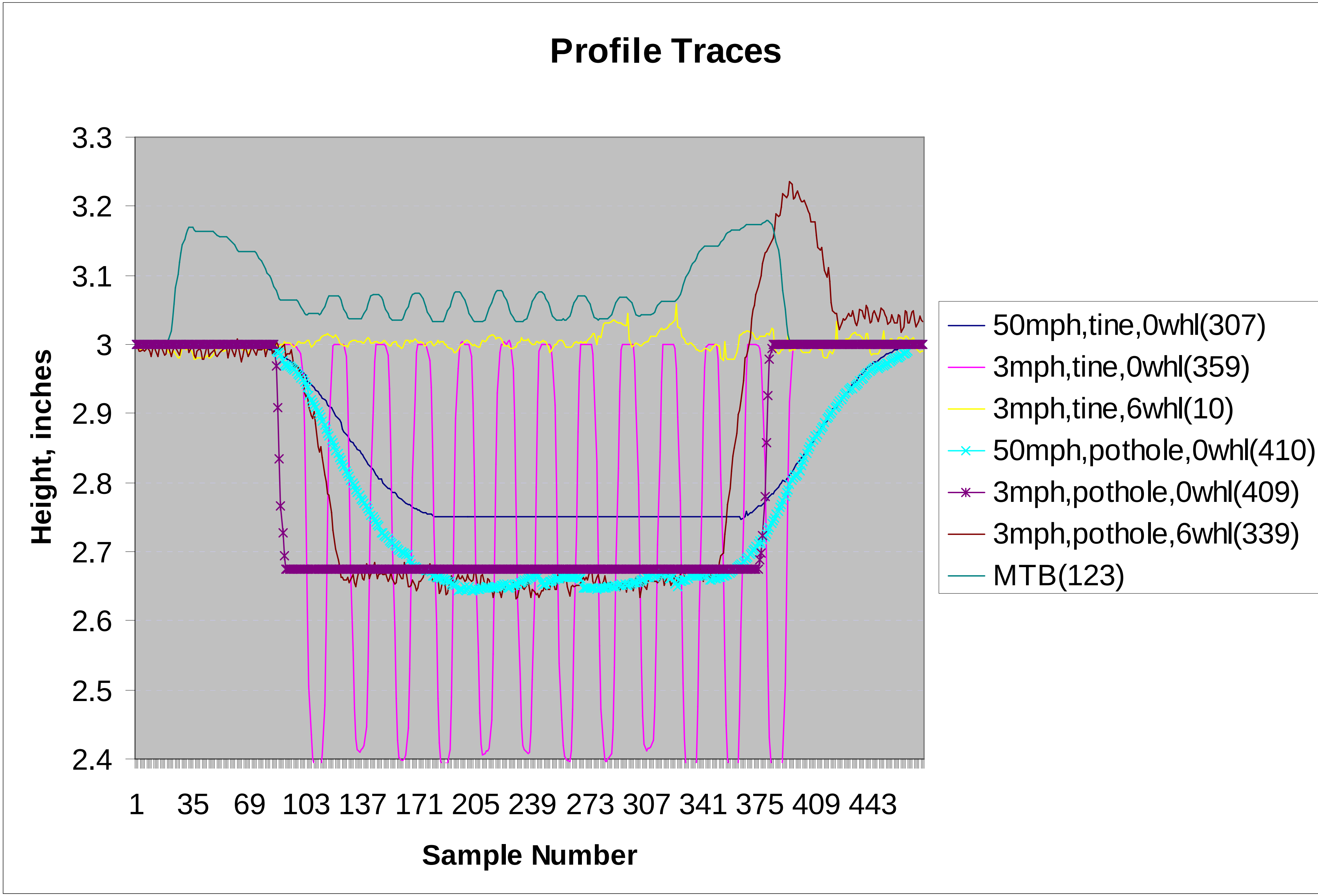

6.00
0.50
0.50
0.50

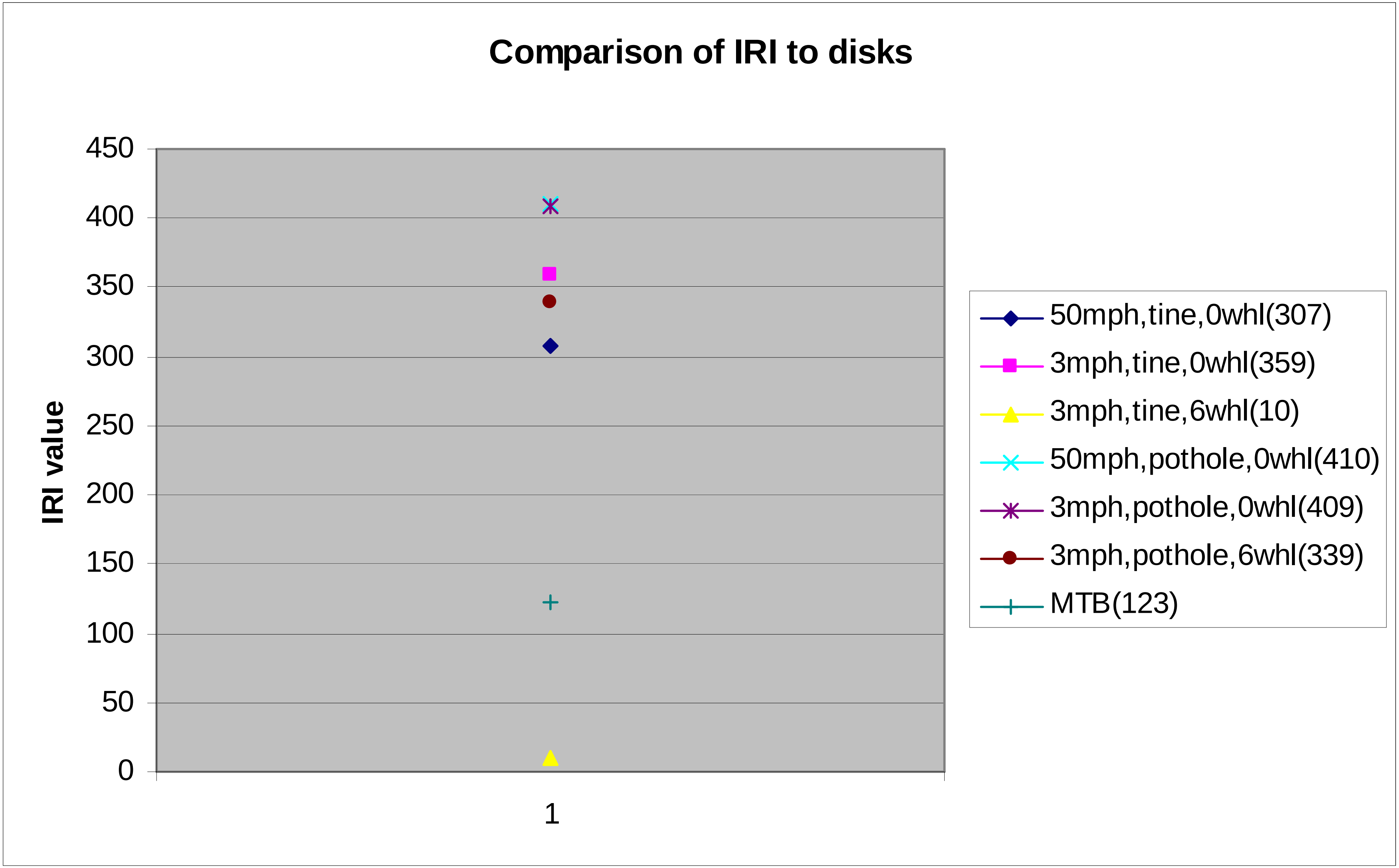


0.25
6.00

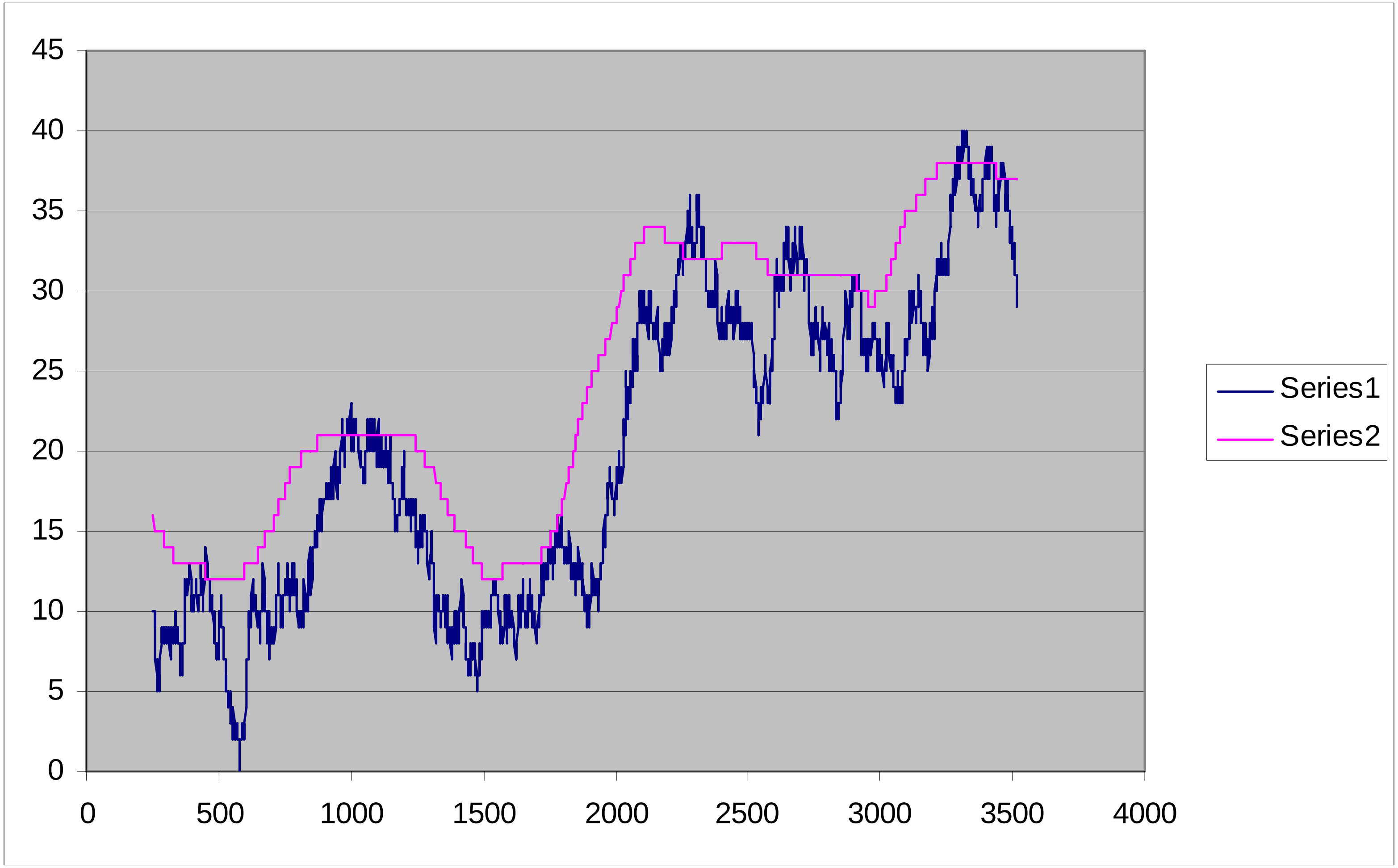
Repairing the “pothole” disk data



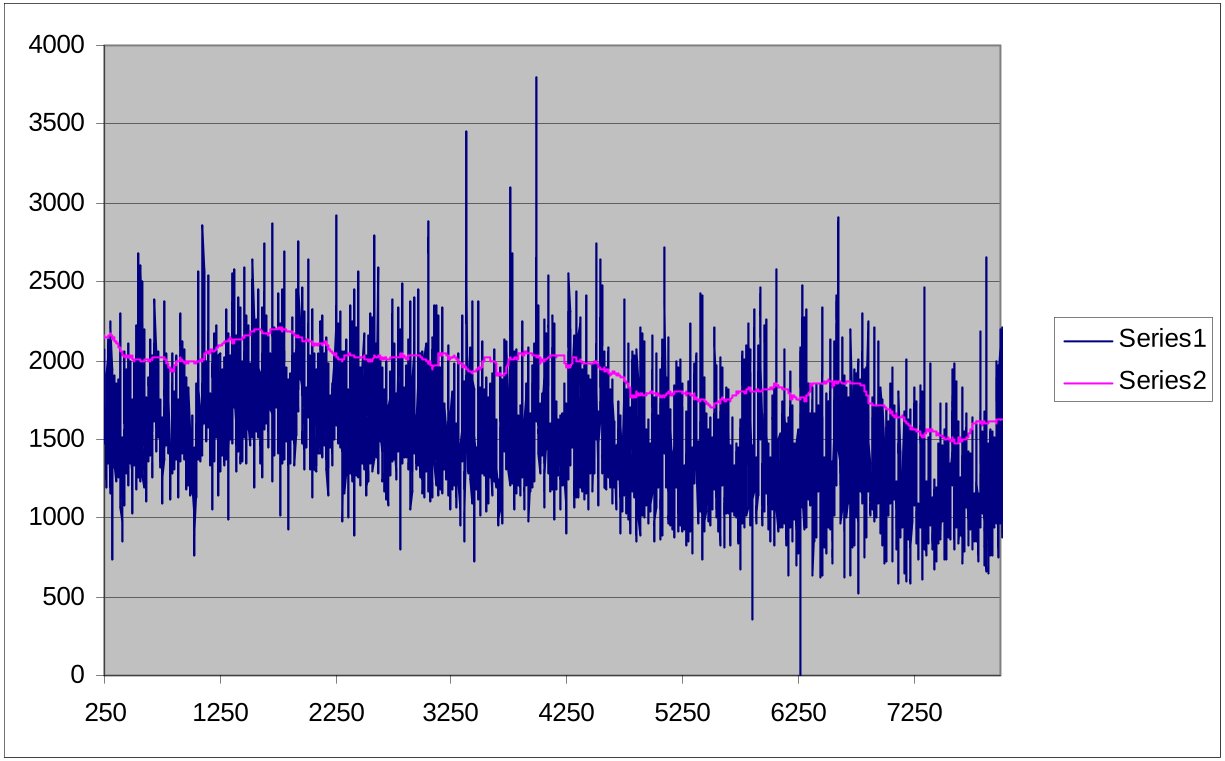

Profile Traces
Height, inches
3.3
3.2
3.1
3
2.9
2.8
2.7
2.6
2.5
2.4
1
35
69
103
137
171
205
239
273
307
341
375
409
443
Sample Number
50mph,tine,0whl(307)
3mph,tine,0whl(359)
3mph,tine,6whl(10)
50mph,pothole,0whl(410)
3mph,pothole,0whl(409)
3mph,pothole,6whl(339)
MTB(123)


Comparison of IRI to disks
IRI value
450
400
350
300
250
200
150
100
50
0
1
50mph,tine,0whl(307)
3mph,tine,0whl(359)
3mph,tine,6whl(10)
50mph,pothole,0whl(410)
3mph,pothole,0whl(409)
3mph,pothole,6whl(339)
MTB(123)

Tire Bridge on Excel Data



Tire Bridge on 1 meter Texture (8 samples / mm)



Michalk Tire Bridging Algorithm

- 18 floating operations per sample
- Suitable for real time use
- Requires memory size equal to sample set
- 33 lines of code not including de-glitching or dc-offsetting
- Gory details
  - Remove DC offset for overflow prevention
  - Remove glitches
  - Calculates average, standard deviation over selected length
  - Using Z-transform, assume normal data to select the elevation which represents the 10% line.
  - Report the sum of the average and Z-transform.

```
int calculate_michalk_tire_datum(double zx,
                   int winsamps,
                   int insize,
                   int *indata,
        int *return_dat
                   ) {

  int inidx = 0;               // current pointer into input array
  double t1=0;        // term1 used for standard deviation
  long long t2=0;
  float avg=0.0;              // average used for standard deviation
  register int back_idx=0;    // array index pointing to first element in averaging window
  register int backward_val=0;// value of first element in averaging window
  register int forward_val  =0;// value at end of window
  int endsize=0;              // array index of last point in window to process
  int offset=0;               // calculated offset to apply to incomning data
  long long sum=0;            // sum applied to subtracted data for standard deviation
  long long sumsq=0;
  register float sdev=0;
  int i;

  // Offset all incoming data.
  // This is important so that the standard deviation calculation is
  // less likely to overflow.
  //deglitch(insize, indata, winsamps);
  dc_offset(insize, indata);

  inidx = 0;
  endsize = insize;

  if (winsamps >= endsize) {
    fprintf(stderr, "Window size is larger than sample set.\n");
    return 1;
  }

  // Prime the filter
  while (inidx < winsamps) {
    forward_val = indata[inidx];
    sum   += forward_val;
    sumsq += ((long long)forward_val * forward_val);
    inidx++;
  }
  avg = (float)sum / (float)winsamps;

  // calculate the tire datum
  while (inidx < endsize) {
    back_idx = inidx-winsamps;
    backward_val = indata[back_idx];
    forward_val = indata[inidx];
    sum   -= backward_val;
    sum   += forward_val;
    avg = (float)sum / (float)winsamps;

    sumsq -= ((long long)backward_val * backward_val);
    sumsq += ((long long)forward_val  * forward_val);

    t1 = winsamps*sumsq;
    t2 = (long long)sum * (long long)sum;
    if ((sumsq < 0) || (t1 < 0) || (t2 < 0))  {
      fprintf(stderr, "Overflow\n");
      // return 1;
    }
    if (t1 > t2) {
      sdev = sqrt((t1-t2)/(winsamps*winsamps));
    } else {
      fprintf(stderr, "bad standard deviation\n");
      // return 1;
    }
    // determine offset
    offset = (zx * sdev) + avg;
    fprintf(stderr, "%d,%d,%f,%f\n",inidx,offset,sdev,avg);

    return_dat[back_idx] = offset;

    inidx++;
  }
  return 0;
}
```

Future Work

- Locate a section at TTI grounds
- Establish reference profile with walking profiler
- Establish variation in texture
- Establish variations in the inertial profile which deviate from reference profile, and correlate to the variation in texture.