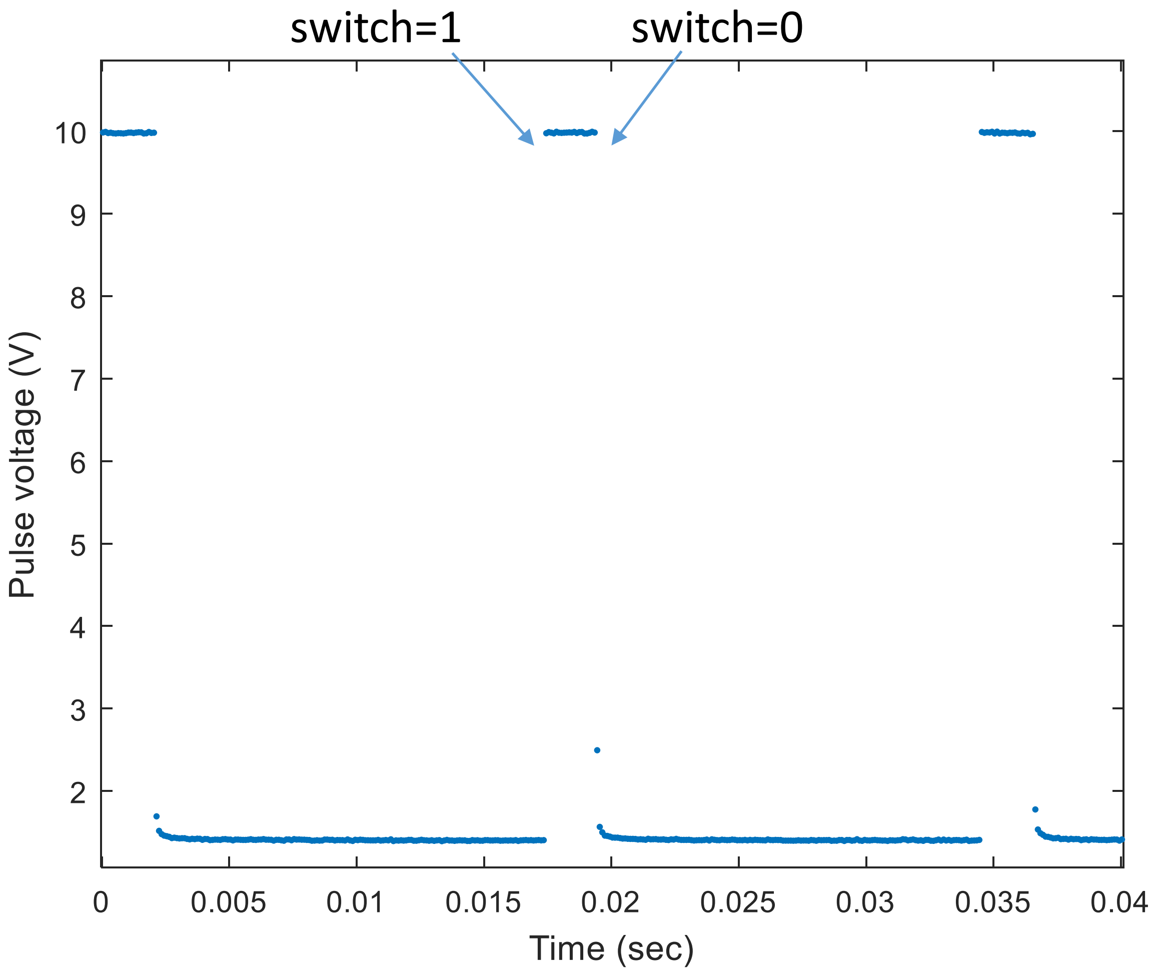
Steps to analyzing the engine data

Converting p vs. t and crank position to p-V - Creating the volume and pressure vectors

- Input the values of the compression ratio, the displacement, the connecting rod length, and the stroke. You must use these values to calculate the clearance volume and the radius of the crank, a.
- Convert the pressure sensor voltage to pressure and add another plot below the above plot to show that the peak pressure is near TDC.

Converting p vs. t and crank position to p-V: - Creating the θ vs. t vector

- The crank position sensor indicates top dead center (TDC) of the piston position by the leading edge of voltage pulse.
- Search through the crank position sensor vector for the leading edge of the pulse by taking the difference between adjacent points to find a step above a threshold. Since the pulse is large, you can use a large e.g. ~1 V threshold. Define a switch variable to 0 before you start searching and a initialize the pulse counter variable j=1;. Your “if” statement should have both the threshold and the switch = 0 as the condition.
- Once you find a positive difference, store the row in an array corresponding to TDC (e.g. pulse(j) = i; where i is the variable for the whole time array), toggle the switch variable to one and increment j by 1 (j=j+1). This means you will ignore any positive adjacent differences above the threshold as the pulse continues to rise.
- Continue to calculate differences between adjacent points until you find a value below a negative threshold. When you find the downside of the pulse, reset the switch variable to zero so you can find the next leading edge of the pulse.
- Create a θ vs. time vector (array) based on the index values for the leading edge you just stored. Use the leading edge as θ =0 and the last point before the next pulse will be a little less than 2π. You need to use nested loops The outer loop steps through the vector that has the leading edge locations. The inner loop steps through the leading edge locations e.g. (pulse(j):pulse(j+1)-1).
- The length of the pulses are not fixed so this is the way to adjust for each cycle. You will use the vector indices to compute θ instead of the actual time values because they are essentially giving the same information in this normalized equation.
- Use the θ array to calculate a volume array using the equation from the PowerPoint.
- Test how well this has worked by plotting the crank position sensor output and θ vs. t on the same plot. The minimum volume should correspond to θ = 0;


switch=1
switch=0
10
9
8
7
6
5
4
3
2
Pulse voltage (V)
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
Time (sec)

Array index

Integrating the p-V curves

- Use trapz on the p and V vectors over a full cycle (two up and down motions of the piston) to calculate the mechanical work per cycle per piston. Start from the $\theta = 0$ location for the trapz function to work properly. Check your values to make sure they are of the correct magnitude by comparing to the engine power. Using SI units will make things a bit easier but are not required.
- Store these values in appropriate arrays so you can provide the requested statistical data.
- You need to sort out the fact that you are only recording p vs. t data for one piston when you compare to engine power and work.

Common problems

- θ not correct, missing pulse edge, off by π, etc.
- p vs. V integration only corresponds to one piston, p-V integration only over half the cycle.
- Transmission reduces RPM by 3.16 but increases torque by 3.16. Must convert dynamometer reading to engine reading.
- English units. Confusion regarding whether you are computing for one or two cylinders.
- Remembering that there are two revolutions per cycle and that they are interleaved with one another.