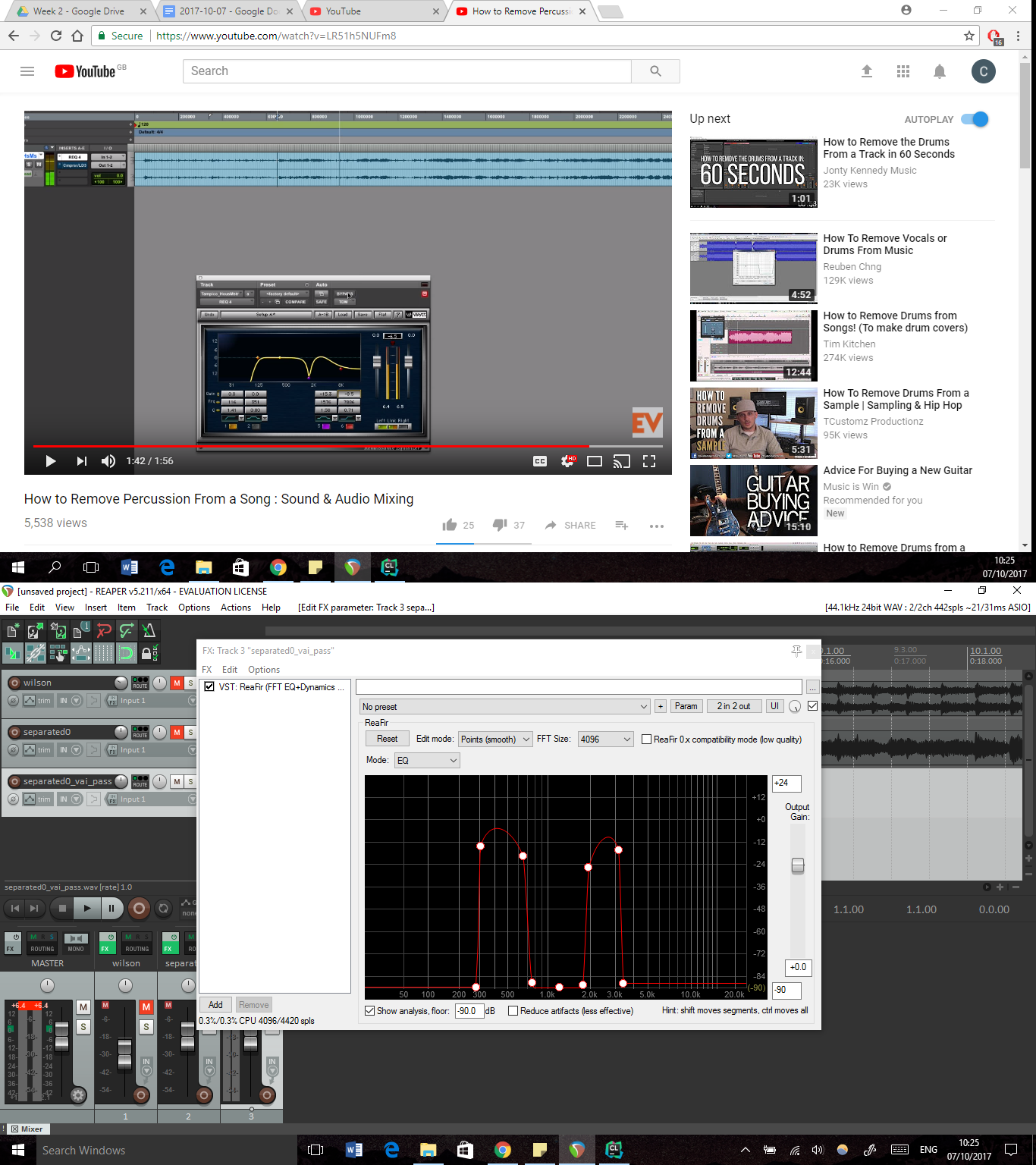
7th October 2017

My first experiment of the day is to look a little further at applying filters post-processing.

As was seen yesterday, applying band pass filters to the ‘minimum’ area of a guitar’s frequency spectrum seemed to do the best part of bugger all I believe with the steve vai track. I might try it with the steven wilson one to see what I get out.

Experiment with Steven Wilson track: No luck either, one IC seems to remove the lead guitar for the most part but it is still audible.

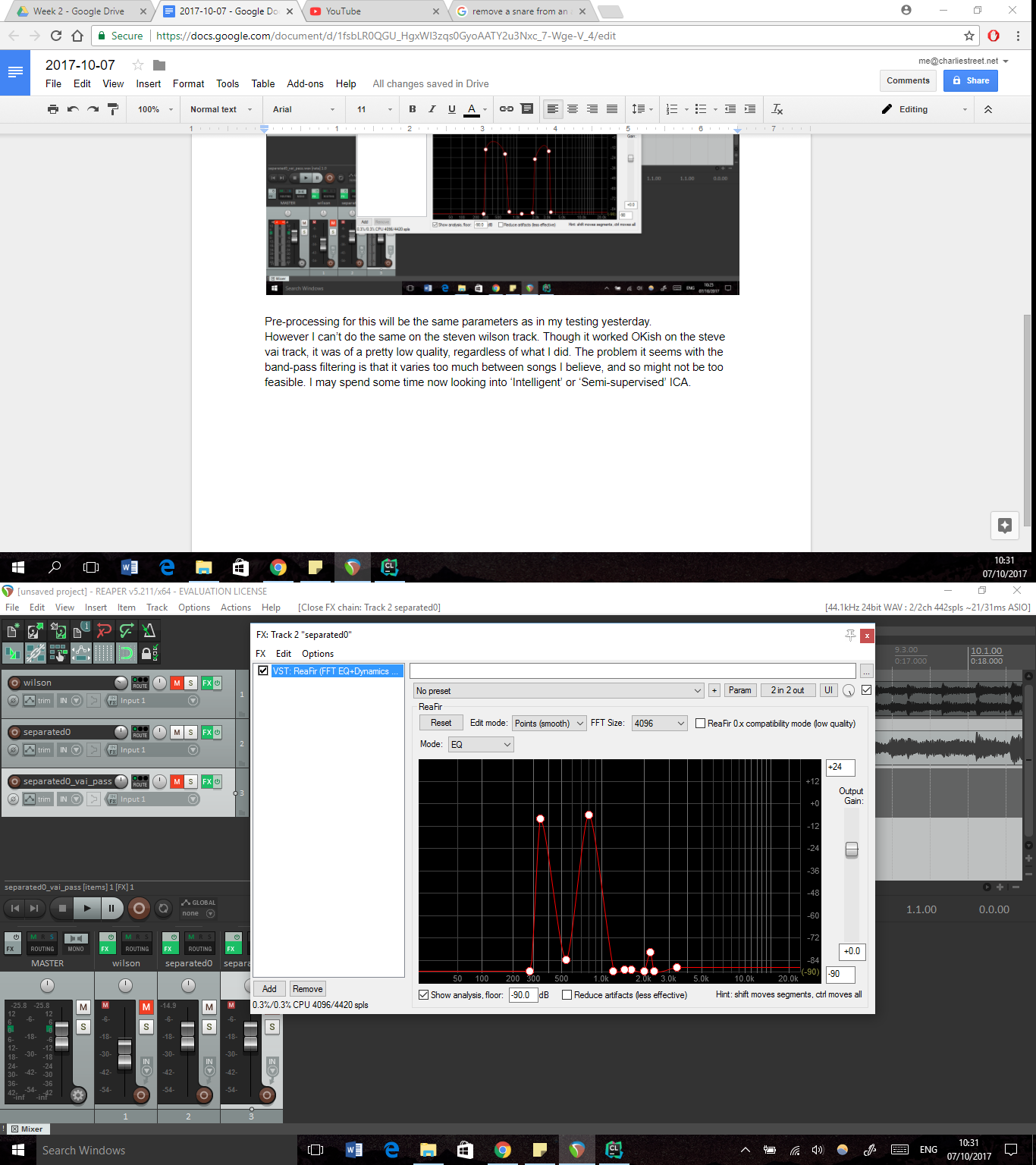
This is probably the best I have from post processing:



Pre-processing for this will be the same parameters as in my testing yesterday.

However I can’t do the same on the steven wilson track. Though it worked OKish on the steve vai track, it was of a pretty low quality, regardless of what I did. The problem it seems with the band-pass filtering is that it varies too much between songs I believe, and so might not be too feasible. I may spend some time now looking into ‘Intelligent’ or ‘Semi-supervised’ ICA.

For the wilson track, the best I could get was by:

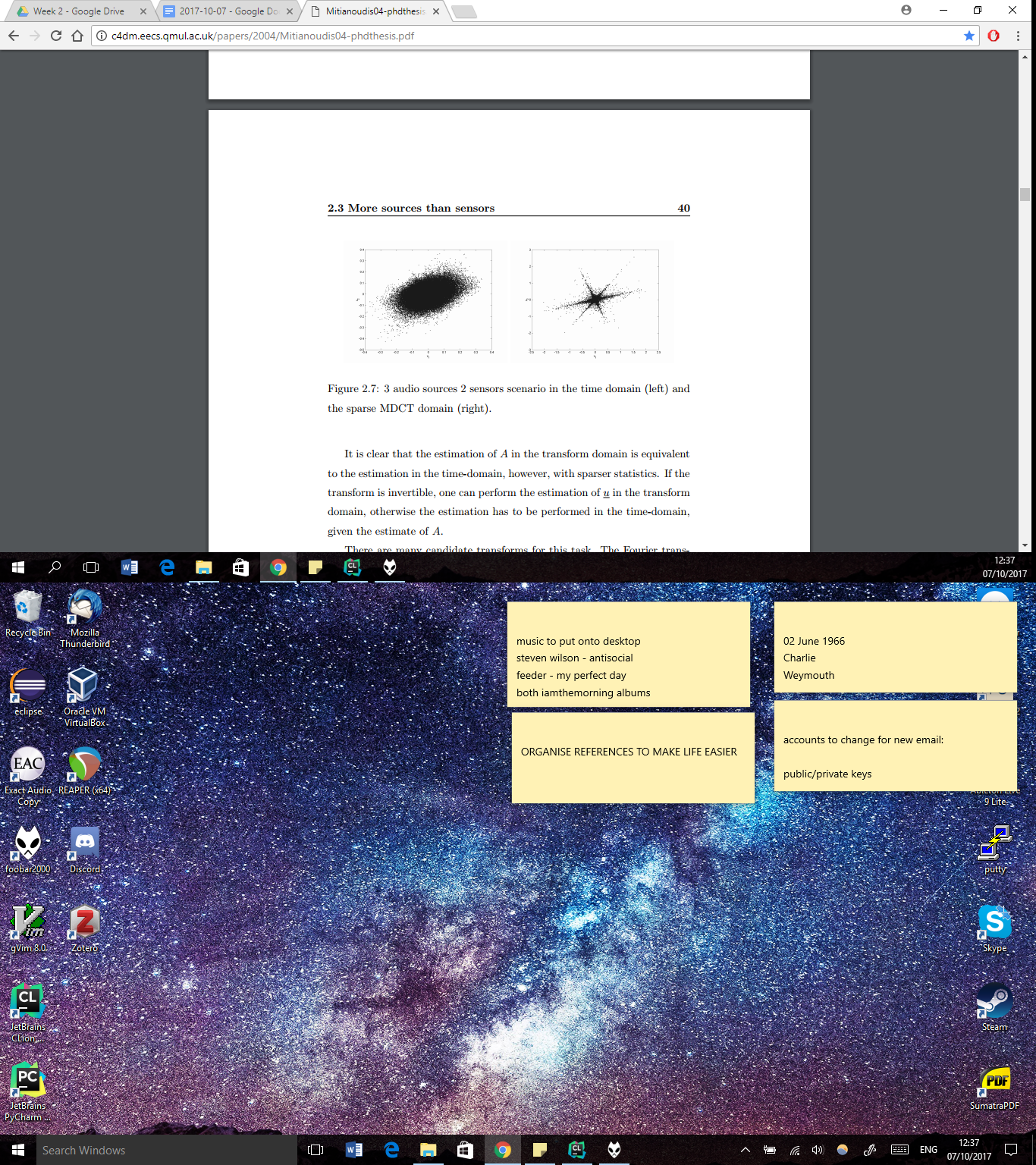


Thom, B., 2001. BoB: An Improvisational Music Companion. Carnegie Mellon University.

Looking at this paper, I have got a few interesting ideas:

<http://c4dm.eecs.qmul.ac.uk/papers/2004/Mitianoudis04-phdthesis.pdf>

The first is to essentially make my data sparser. This may look like this:



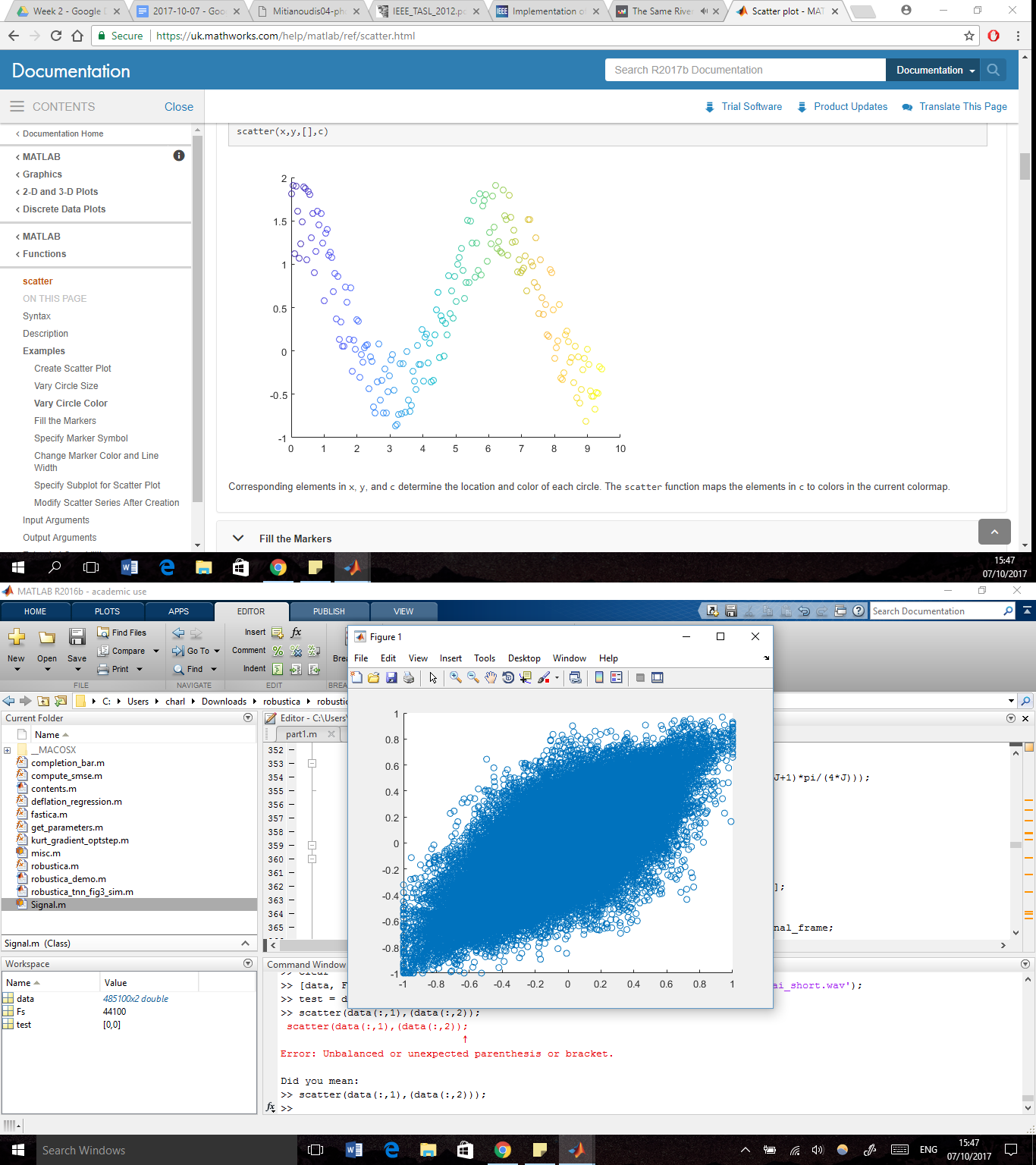
This can be achieved using a modified discrete cosine transform (or possibly a wavelet transform). This would allow me to solve for more sources. However, I may first attempt to do the discrete transform on its own and its inverse. If this doesn’t provide much, I can then add in more sources and see what happens (though the solver may take longer, the code shouldn’t be too complicated by the looks of it, it's just dependent on how horrible this MDCT is).

Because the MDCT is even lower on resources than FFT I might try it in matlab first and see what happens.

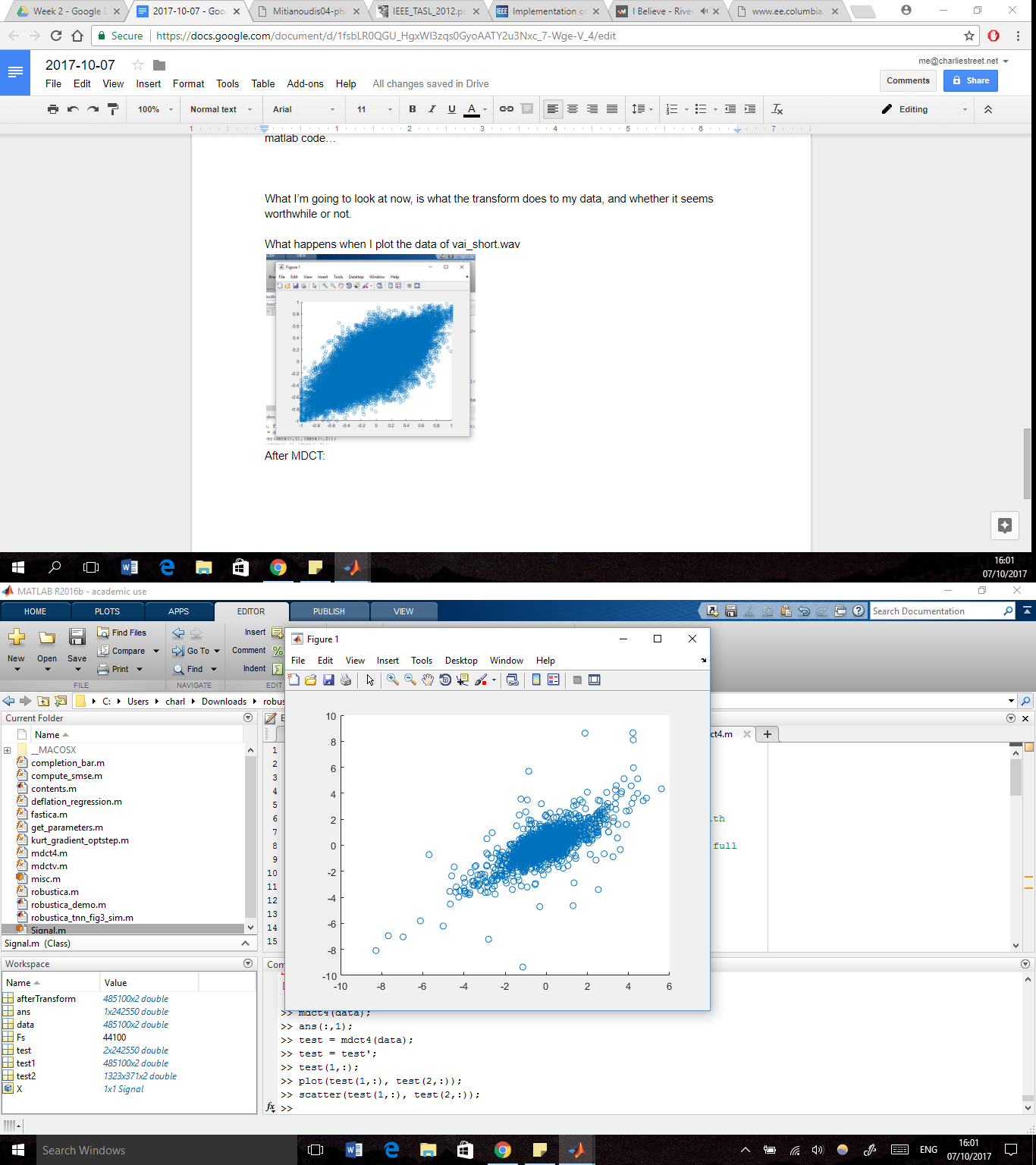
After trying it in Matlab with the MDCT on its own, not a lot happens other than the usual. My other option then is to try it with more sources available to fill in. That is something only my code can probably accommodate. However this looks like it is my fault not knowing how to use the matlab code…

What I’m going to look at now, is what the transform does to my data, and whether it seems worthwhile or not.

What happens when I plot the data of vai\_short.wav



After MDCT (I think):



So this hasn’t really appeared to do a whole lot…

However, it could be that its not doing the lapping for me?

From this, I’m starting to think the penalty method may be my best idea…

I could always try the intelligent ICA implementation and find an instrument recogniser?

I also found something called the DUET algorithm - may be worth looking into