Design Automation & Optimisation for Multi-Material 3D Printing

Benjamin Yi

August 3, 2018

Contents

List of Figures

1 Algorithm comparisons

Current work

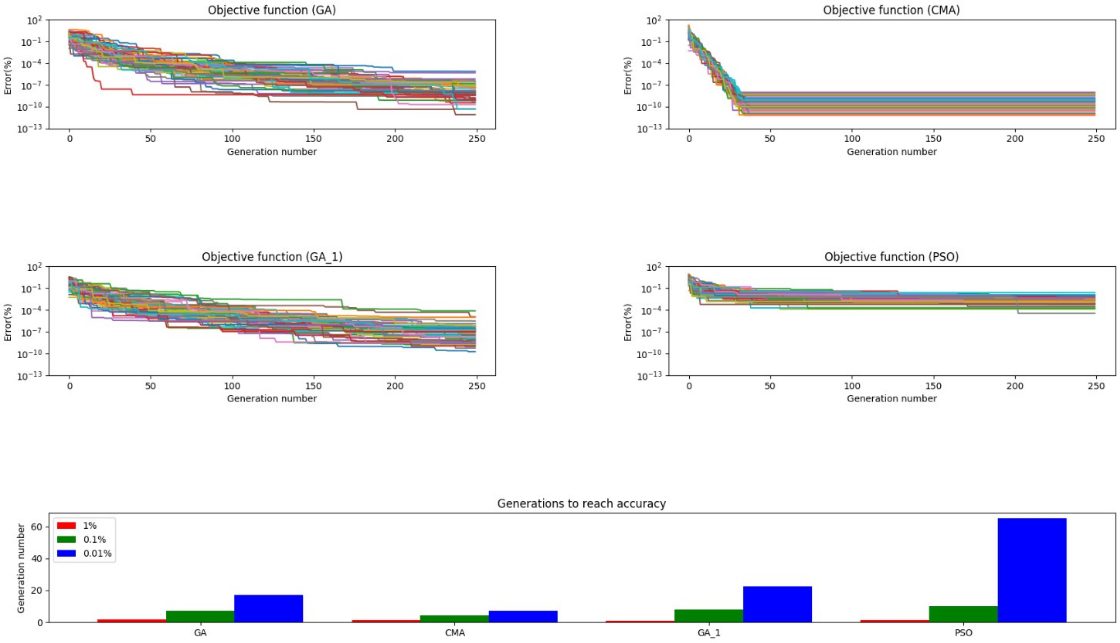


Figure 1: Comparison of algorithms for a 1D beam

- 1D 40 voxel beam using PLA and Al
- Objective function: Sum of absolute differences to target frequencies
- Trying to match a known beam - comparing GA, CMA-ES (binary encoding), PSO, GA (adaptive) in 250 generations
- All algorithms perform fairly well, reaching $< 1\%$ error in 250 generations.
- CMA-ES performed the best, but hard to draw conclusions.

Future work

- Test using 2D 40 voxel and flex/PLA/steelfill
- Test using custom GA crossover
- Test using time as stop criterion, not number of generations
- Test using sum of squares as objective function

2 Population size testing

Current work

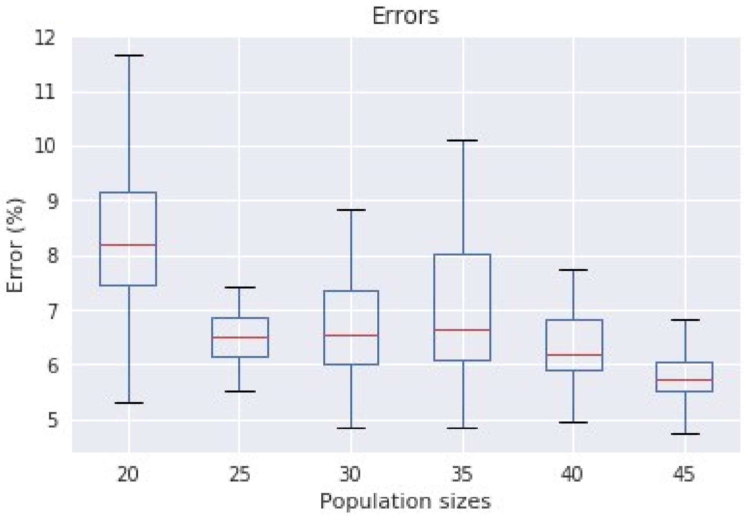


Figure 2: Population size vs final error

- 2D 40 voxel beam using PLA and Al
- Objective function: Sum of absolute differences to (f1 = 200, df = 600)
- Testing population size for GA
- Population size ≥ 25 seems to be the best, with minimal gains after (6.5% vs 8.5%)
- This ignores time taken - assumes number of cores ≥ 45

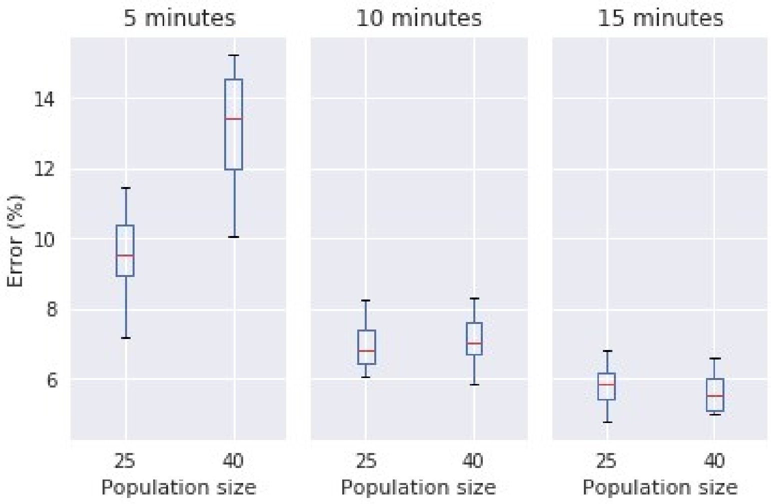


Figure 3: Population size vs time vs final error

- 2D 40 voxel beam using PLA and Al
- Objective function: Sum of absolute differences to (f1 = 200, df = 600)
- Testing population size for GA, using time as stop criterion
- Difficult to draw any conclusion here

Future work

- Test using flex/PLA/steelfill
- Test using custom GA crossover
- Test using machine with large number of cores.
- Repeat second experiment with a larger range of populations
- Test using sum of squares as objective function

3 Uniform objective function

Current work

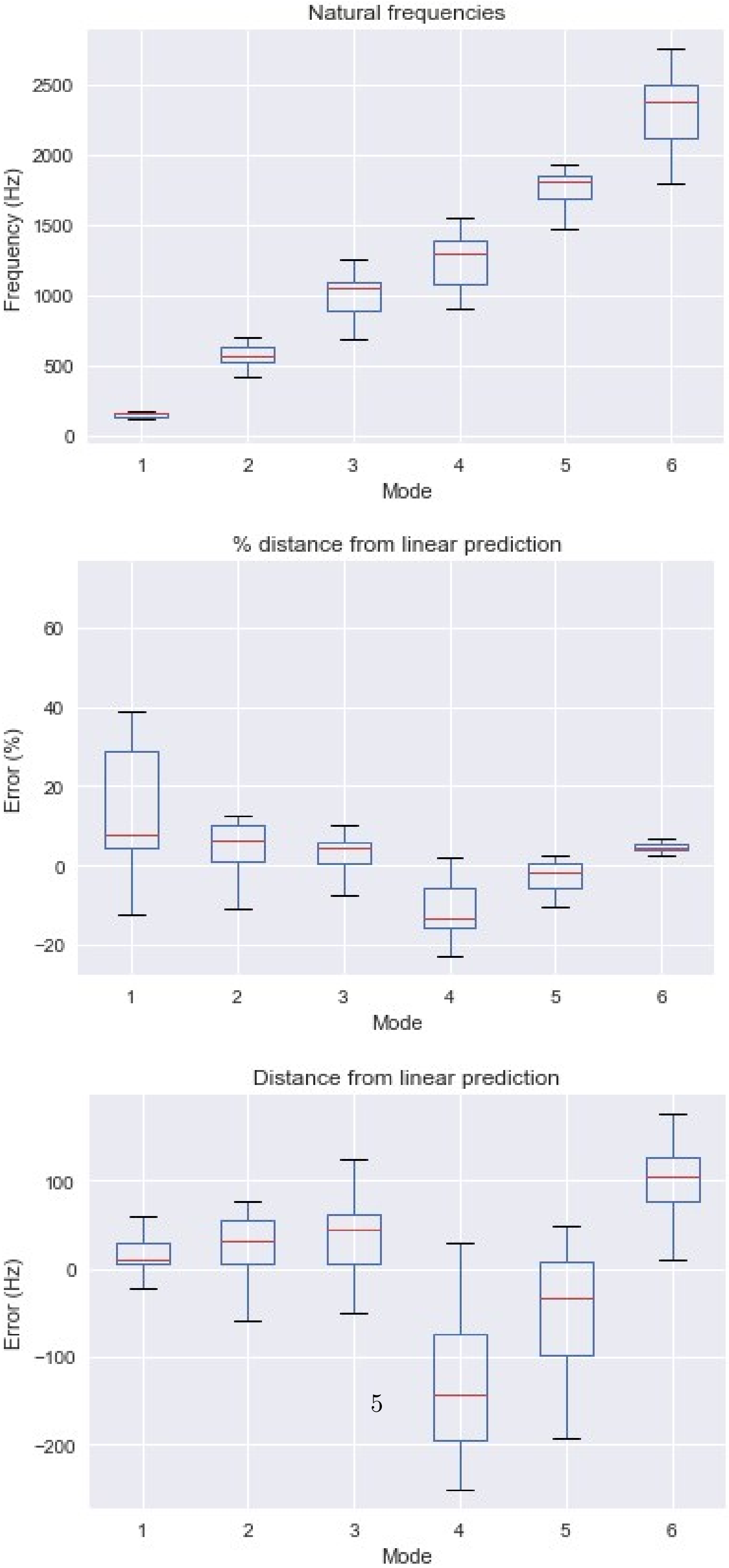


Figure 4: Linear fit as objective function

- 2D 40 voxel beam using PLA and Al
- Objective function: Fit to a line, minimizing r^2
- Using GA for 50 generations
- $r^2 = 0.98 \pm 0.008,\ \text{f1} = 140 \pm 30,\ \text{df} = 420 \pm 40$[1]

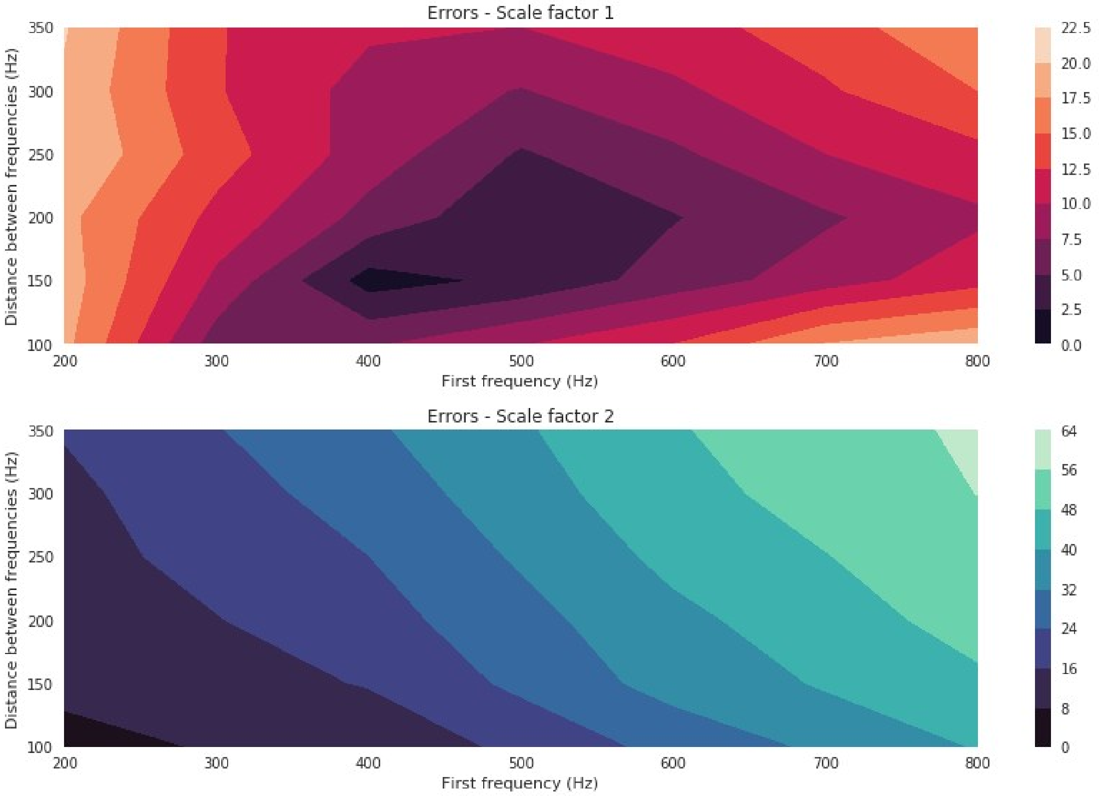
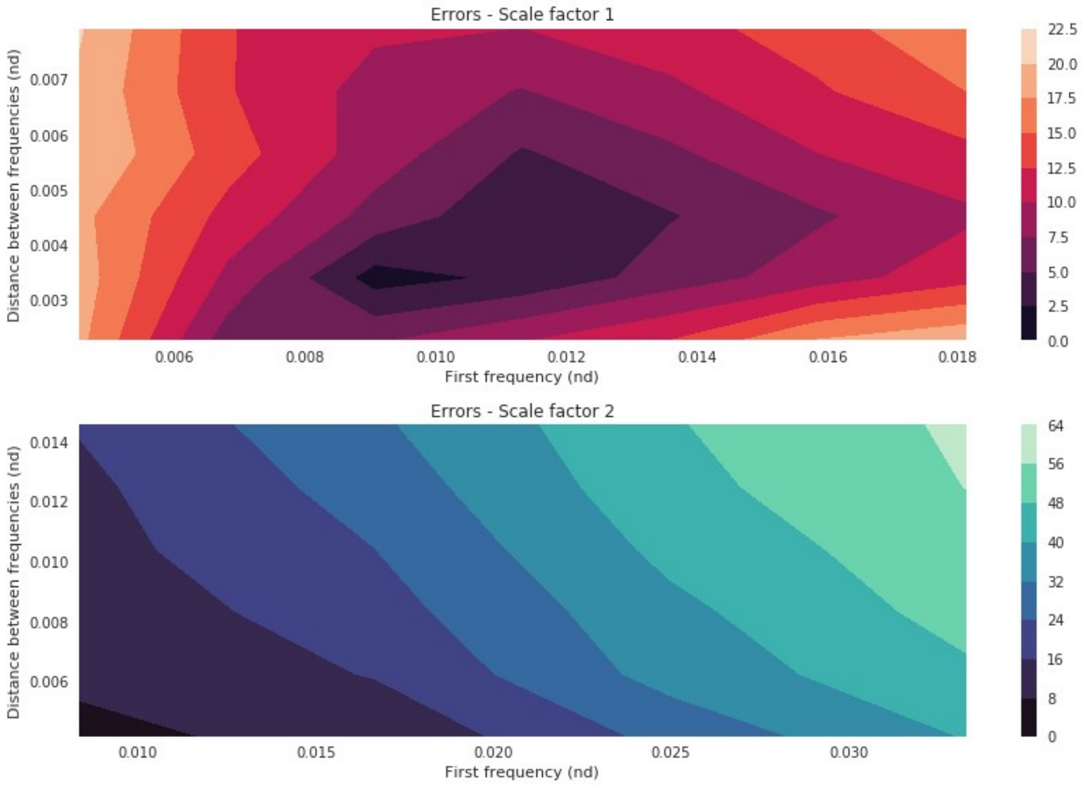


Figure 5: Uniform fit over a range of frequencies

- 2D 40 voxel beam using PLA and Al, doubled dimensions for second graph
- Objective function: Sum of absolute differences to (f1, df)
- Using GA for 1 hour
- Best spot around f1 = 200, df = 600[2]

[1] https://github.com/byi649/Design-Automation-Optimisation-for-Multi-Material-3D-Printing/blob/master/Results/2018-05-02/Statistical%20analysis%20-%20uniform.ipynb

[2] https://github.com/byi649/Design-Automation-Optimisation-for-Multi-Material-3D-Printing/blob/master/Results/2018-07-06/Statistical%20analysis%20uniform.ipynb


Errors - Scale factor 1

Errors - Scale factor 2

Figure 6: Uniform fit over a range of frequencies (non-dimensionalised)

- Same as above, but non-dimensionalised by average wave speed
- Best spot around f1 = 0.009, df = 0.0035

Future work

- Test using flex/PLA/steelfill
- Test using custom GA crossover
- Test using time as stopping criterion
- Test more beam sizes
- Non-dimensionalise by individual wave speed
- Test using sum of squares as objective function

4 Voxel number

Current work

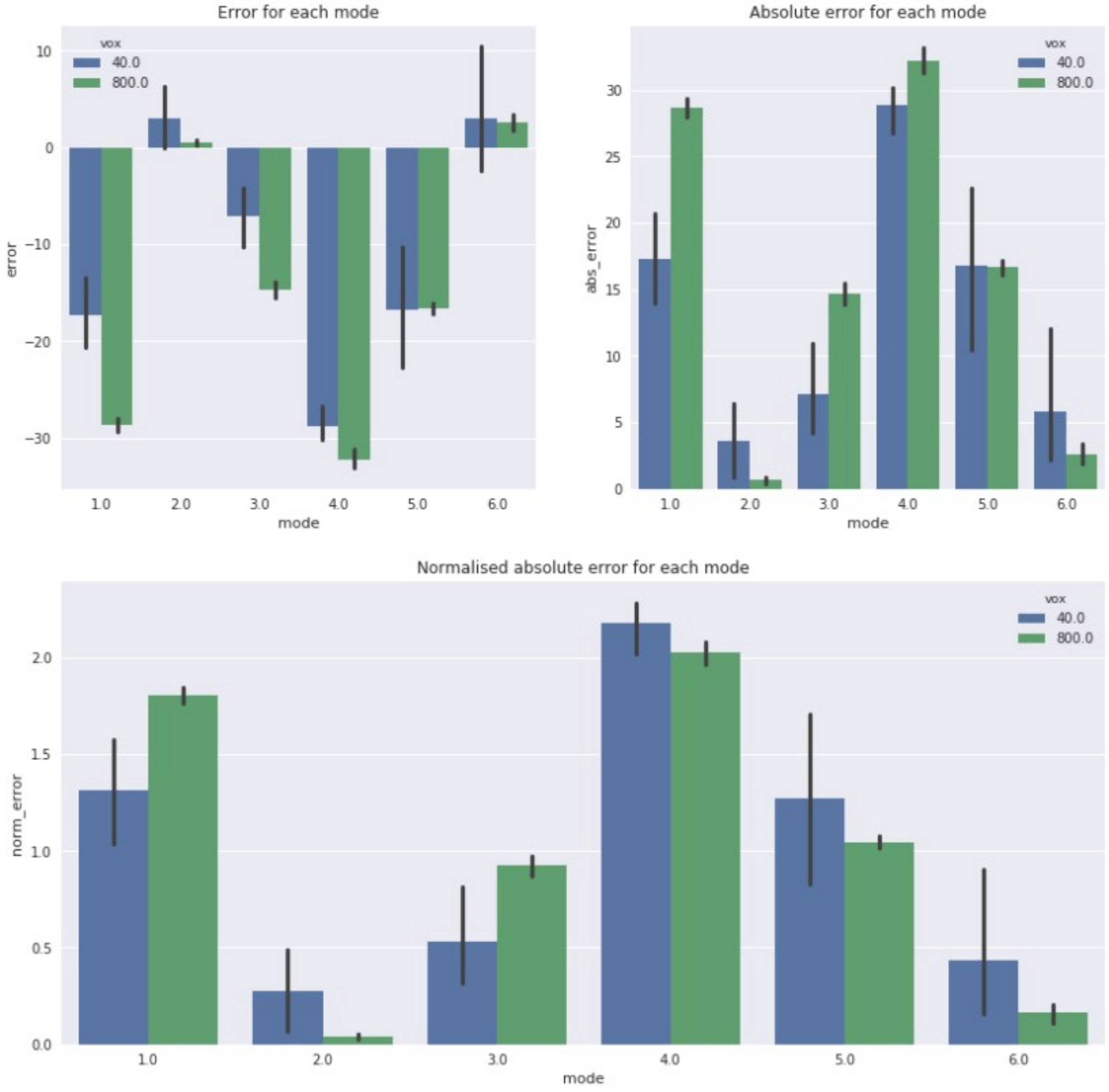


Figure 7: 40 voxel vs 800 voxel comparison

- 2D 40 voxel beam using PLA and Al, 3D 800 voxel beam using PLA and Al
- Objective function: Sum of absolute differences to (f1 = 200, df = 600)
- Using GA for 1 hour

- 40 voxel performs better than 800 voxel[3], no other patterns found

Future work

- Test using flex/PLA/steelfill
- Test using custom GA crossover
- Test using sum of squares as objective function

[3] https://github.com/byi649/Design-Automation-Optimisation-for-Multi-Material-3D-Printing/blob/master/Results/2018-06-08/Statistical%20analysis.ipynb

5 Custom crossover

Current work

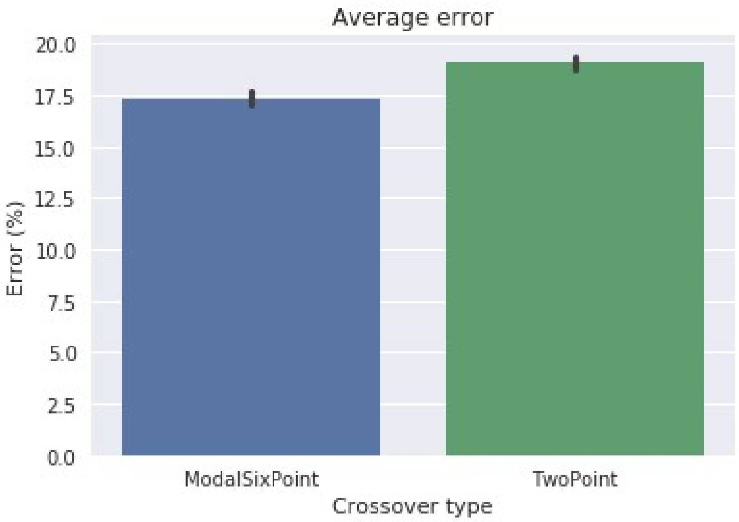


Figure 8: Custom crossover vs standard 2 point crossover

- 3D 800 voxel beam using PLA and Al
- Objective function: Sum of absolute differences to (f1 = 200, df = 600)
- Using GA for 1 hour
- 6-point crossover performs better than stock 2-point crossover (17.5% vs 19% error)[4], p-value = 3.8e-8.

[4] https://github.com/byi649/Design-Automation-Optimisation-for-Multi-Material-3D-Printing/blob/master/Results/2018-06-29/Statistical%20analysis.ipynb

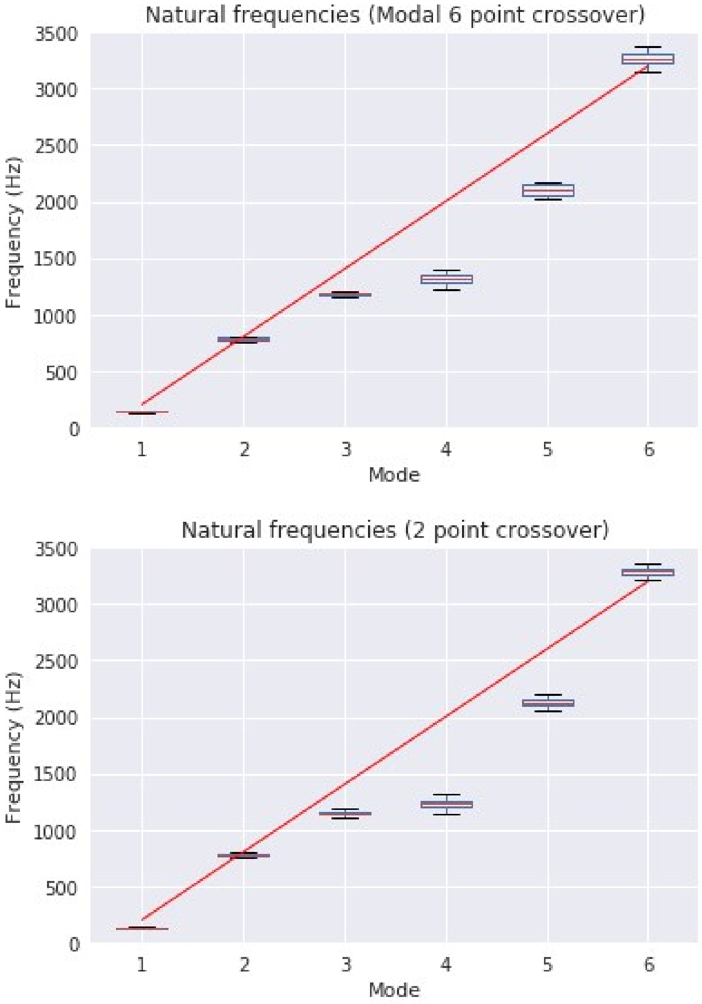


Figure 9: Custom crossover mode frequencies

- Same as above, looking into individual frequencies matched
- Mode 4 perhaps slightly better?

Future work

- Test using flex/PLA/steelfill
- Test using sum of squares as objective function
- More indepth tests on custom crossover usefulness (testing individual frequencies etc)

6 Clustering

Current work

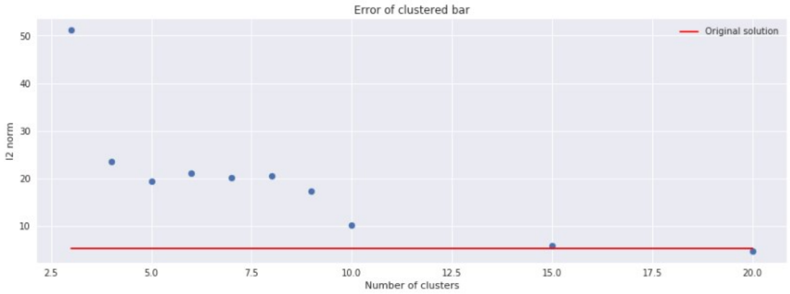
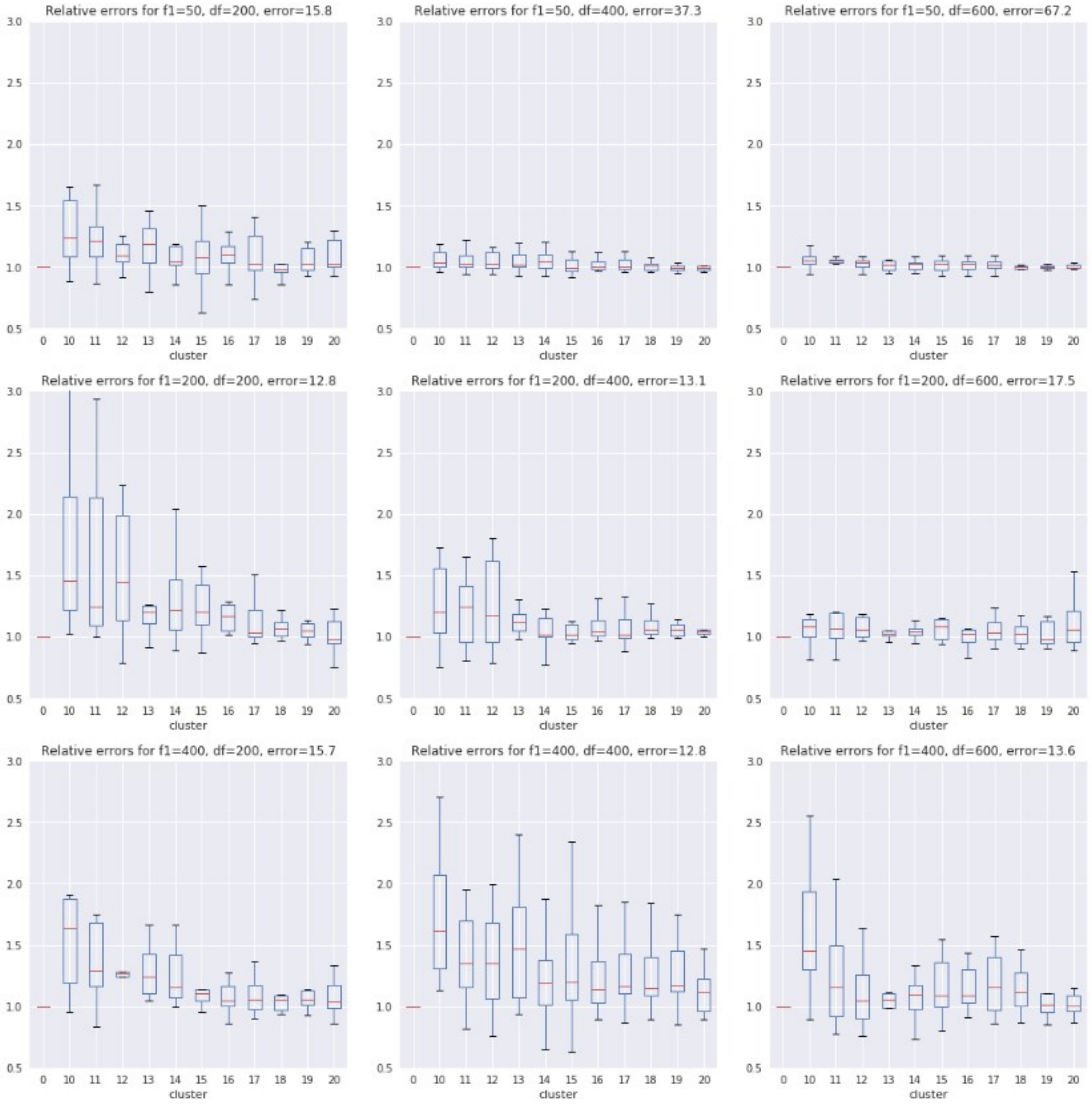


Figure 10: Clustering introduced error

- 2D 40 voxel beam using flex and steelfill
- Objective function: Sum of squares to (f1 = 200, df = 600)
- Using CMA-ES for one hour, then clustering via scikit-learn
- Breakpoints at n.clusters = 5, 10, 15 (final error = 20%, 10%, 5%)



Relative errors for f1=50, df=200, error=15.8

Relative errors for f1=50, df=400, error=37.3

Relative errors for f1=50, df=600, error=67.2

Relative errors for f1=200, df=200, error=12.8

Relative errors for f1=200, df=400, error=13.1

Relative errors for f1=200, df=600, error=17.5

Relative errors for f1=400, df=200, error=15.7

Relative errors for f1=400, df=400, error=12.8

Relative errors for f1=400, df=600, error=13.6

Figure 11: Clustering introduced error for many frequencies

- 2D 40 voxel beam? using ? and ?
- Objective function: Sum of squares to (f1, df)
- Using CMA-ES for one hour, then clustering via scikit-learn
- Errors are normalised within each plot

Future work

- Test using flex/PLA/steelfill
- Test using sum of squares as objective function