Hi there,

This section of the file repository is dedicated to the simulation developed for TorPeDO’s NCal prototype. This simulation was then developed to assist in optimising the prototype’s architecture for TorPeDO’s calibration. The aim of the simulation was to model the setup demonstrated in Figure 1 where *m1* and *m2* are the slug masses to be used as previously discussed and simulate the two rotating point masses from Figure 5. *Mx1*, *Mx2*, *My1* and *My2* are the masses on TorPeDO’s arms respectively. The right image of Figure 1 provides an overview visual of TorPeDO with Fabry-Perot cavity length *d*. Located distance *L* to the right in the same plane is the rotational axis of point masses *m1* and *m2*, following a circular path of radius r with oscillation period *T*. The simulation was developed in MatLab. When using the simulation, the user inputs values for *m1, m2, r, L* and *T*. The most recent version of the simulation (Mark 6) outputs d over three rotation periods of *m1* and *m2* as demonstrated in Figure 2.

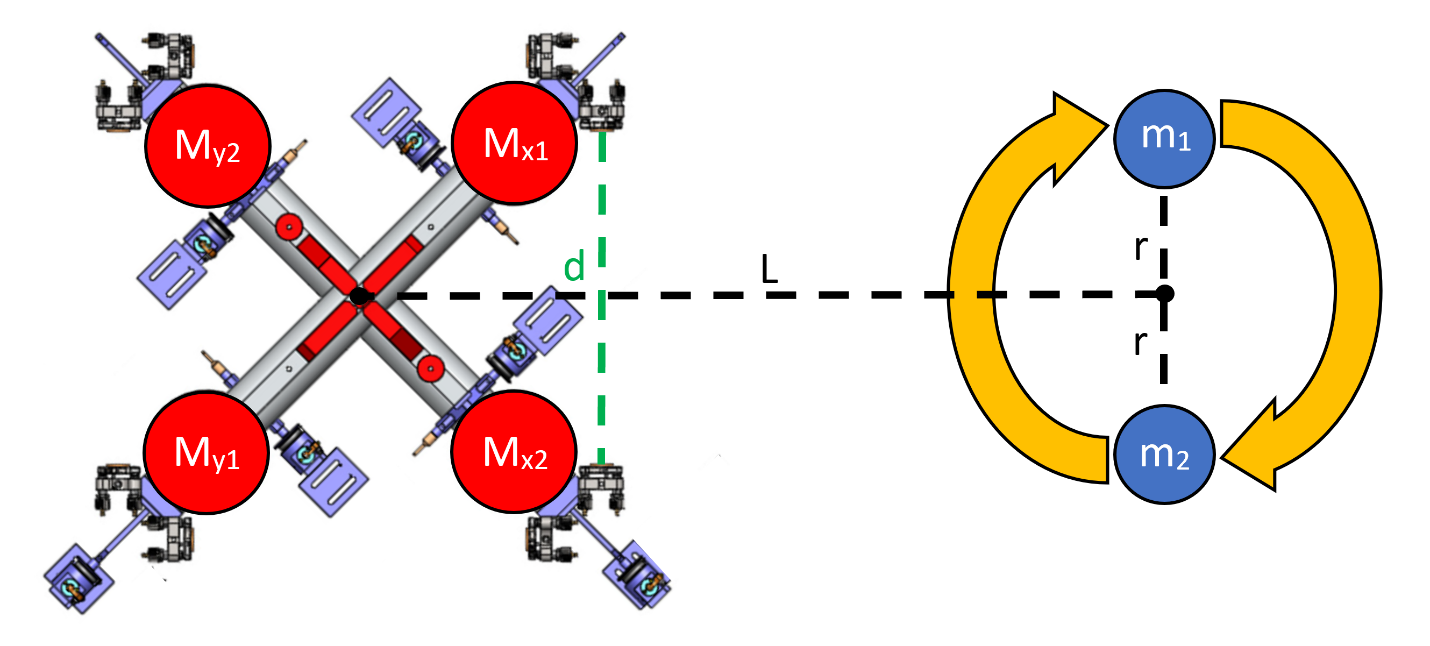


Figure . The setup the simulation is based upon as described above.

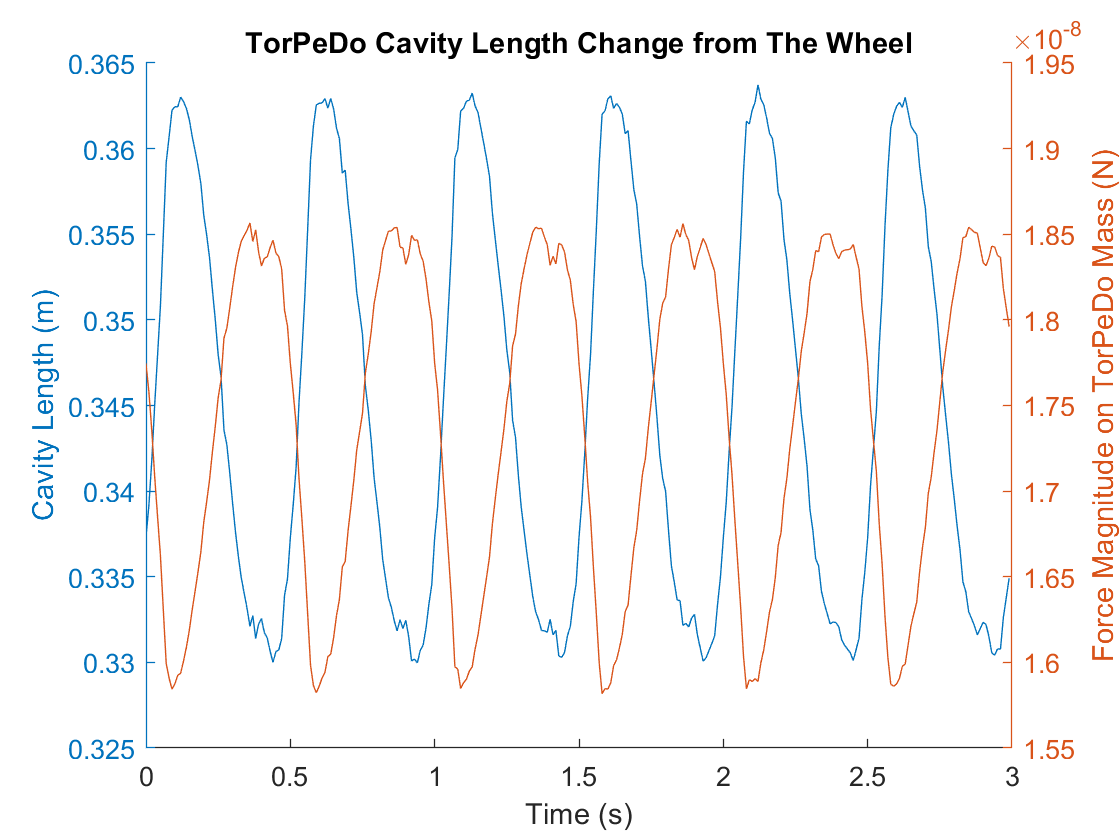


Figure . The output of Mark 6, the cavity length d, plotted alongside the total force magnitude exhibited on Mx1 from Figure 1.

The simulation development took place over six iterations. Mark 1 began as a simplified adaption of Figure 1 and each successive version increased in complexity over the previous. The simulation iterations can be summarised as follows:

[**Mark 1:**](https://thebestportfolio.squarespace.com/s/gravity_sim_MArk1.m)The first simulation was based on the model shown in Figure 3 where the idea was to plot the gravitational signal over a single rotation period of *m1* and *m2* from a point *d* away from their circular path. *m1* and *m2* were made equivalent to a tonne and were 1 meter apart.

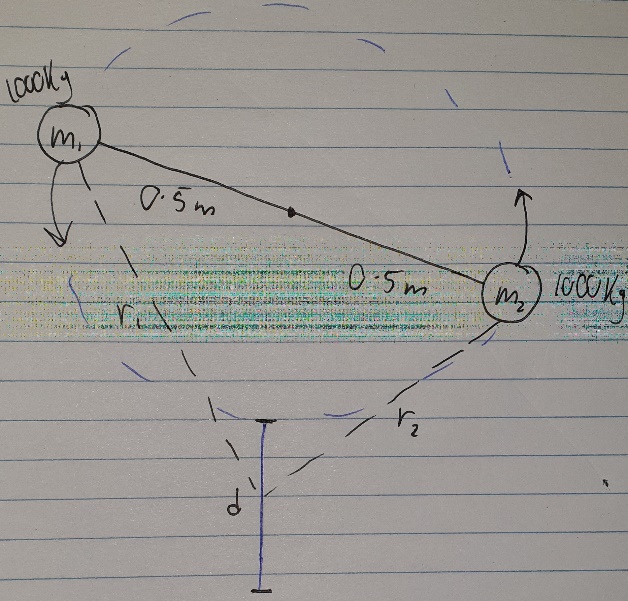


Figure . The model which Mark 1 of the simulation was based upon.

[**Mark 2:**](https://thebestportfolio.squarespace.com/s/gravity_sim_MArk2.m)This version was the same as Mark 1 but allowed the user to set the mass values, their orbiting radii, the maximum distance of the point of analysis and output a surface plot of the gravitational signal as a function of time and along *d*.

[**Mark 3:**](https://thebestportfolio.squarespace.com/s/gravity_sim_MArk3.m)This version took the basis of the previous iteration and applied it to *Mx1* of Figure 9. Mark 3 took in the specific dimensions of TorPeDO (i.e. the arm length, *Mx1* and *Mx2*), allowed the user to set *m1, m2, r* and *L* and output the magnitude of gravitational force *Mx1* exhibited over a rotation period of the wheel. Note that *My1* and *My2* were not accounted for as a simplification of the model.

[**Mark 4:**](https://thebestportfolio.squarespace.com/s/gravity_sim_MArk4.m)This version was the same as Mark 3, except the output was d from Figure 9 over three oscillation periods. *d* was difficult to calculate, requiring approximation methods in MatLab to do so meaning the output was noisy and questionable in accuracy. The problem with the output is that (besides the noise) it is bimodal in frequency and *d* is changing up to 7cm in length (which is a lot!). However, the plot does follow the same signal of the magnitude of gravitational force on *Mx1* which also shouldn’t be bimodal, meaning the analysis of the model used in Figure 1 to generate this simulation is incorrect.

[**Mark 5:**](https://thebestportfolio.squarespace.com/s/gravity_sim_MArk5.m)This version is the same as Mark 4, however is no longer bimodal in its output as the model analysis was readjusted. Here I accounted for *My1* and *My2* instead of just *Mx1* and *Mx2*. However, extreme changes in d are still present. After much digging through the code I concluded that determining the miniscule values d would realistically take are beyond MatLab’s approximation function capabilities.

[**Mark 6:**](https://thebestportfolio.squarespace.com/s/gravity_sim_MArk6.m)This is the same version as Mark 5, however the output is the rotational acceleration of TorPeDo’s arms (the output the actual device uses).

Note that each MatLab script is accompanied by many comments to explain what each segment of code is doing. This document should be of valuable use also.