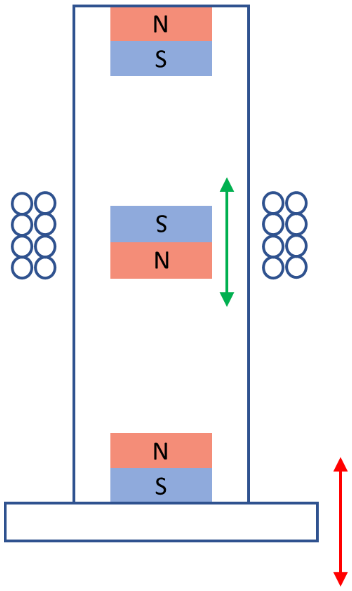
Magnetic Levitation Energy Harvesting

CBP 15-02-23

Energy harvesting from vibrating structures is a recent and active area of research. Many mechanical systems vibrate (car suspension, hair clippers, buildings, aircraft wings, your shoes when walking) and it could be useful to extract energy from these systems, in the form of electricity. One such system is the magnetic energy harvester shown below. On the left is the basic diagram and on the right its incarnation in Unreal.



The top and bottom magnets are fixed inside a cylinder and arranged so the centre magnet, which is free to move, is levitated between them by repulsive forces.

When the base vibrates, then the cylinder and top and bottom magnets oscillate, and this forces the centre magnet to oscillate, though there is relative motion between the centre magnet and the cylinder.

A coil of wire is fixed to the cylinder. So, there is relative motion between the centre magnet and the coil, and this induces a voltage in the coil which can drive a current through a load resistor. So power is harvested.

The interesting feature of this system is the way the force on the central magnet changes with the displacement of this magnet from its levitated position. It turns out that this is ***non-linear*** and has the characteristics of a Duffing oscillator where the force is expressed as

$$F = -k_1 x - k_3 x^3$$

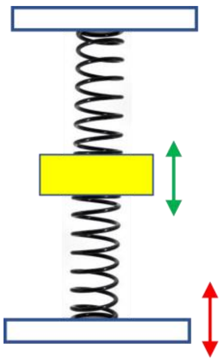
So, what does this mean? Well, let's compare the maglev-VEH with a simple forced oscillator, like the coil spring suspension you might find on your car (see the diagram below). In this case the force-displacement relationship is ***linear*** which means it varies like this,

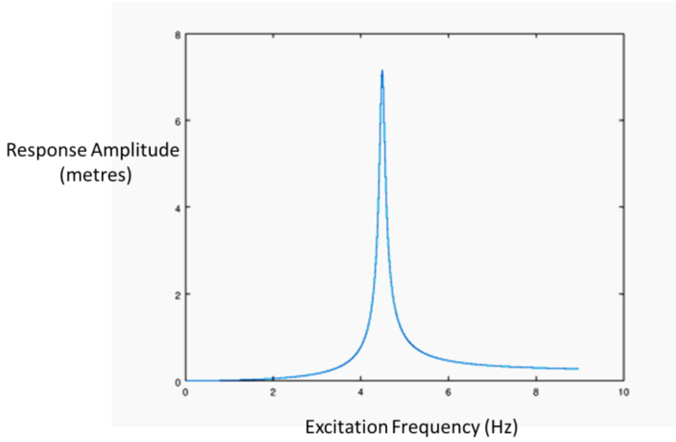
$$F = -k_1 x$$

which says that the force is proportional to displacement x and acts in the opposite direction; it is a *restoring* force. The maglev force has a second term which is nonlinear, proportional to displacement *x-cubed*.

For our maglev, we are interested in how the amplitude of the voltage generated depends on the frequency of vibration of the base plate. Most structures vibrate with a particular 'natural' frequency, so we are interested in 'tuning' our Vibration Energy Harvester (VEH) to this frequency. When the VEH frequency is the same as the structure frequency, then maximum energy is extracted.

Back to the simpler system, the coil suspension. When you increase the driving frequency and measure the car response (height) then you get a graph with a peak shown below. This is called 'resonance'.





So when the excitation frequency hits this value (around 4.5) then the car's response is the greatest. Nonlinear systems show behaviour which is much more complex.

The following discussion of the maglev is taken from a couple of papers authored by Krysztov Kecik (2018,2020) which are available to you. Many of the following graphs are attributed to these papers. The graphs presented below show the amplitude response of the maglev for various forcing frequencies. The latter are expressed in radians per second where $\omega = 2\pi f$ where f is the frequency in Hertz.

Three curves are presented for increasing maximum amplitudes of excitation. The black curve for excitation amplitude 0.005 m has the same shape as the curve for the linear system shown above. There is a clear peak, and we conclude that the cubic non-linearity is not large in this situation. However, at larger values, the curve folds over to the right. This is a direct consequence of the nonlinearity. The blue curve is for an amplitude of 0.01m and the green is for amplitude 0.014. The green curve is very interesting since it shows the *jump phenomenon* characteristic of a cubic nonlinearity (see below).

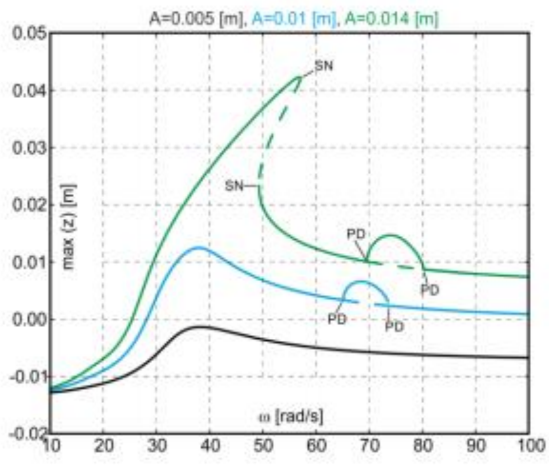


Fig. 5. Resonance curves of the magnet, for k=38.7N/m and R=2.3kΩ

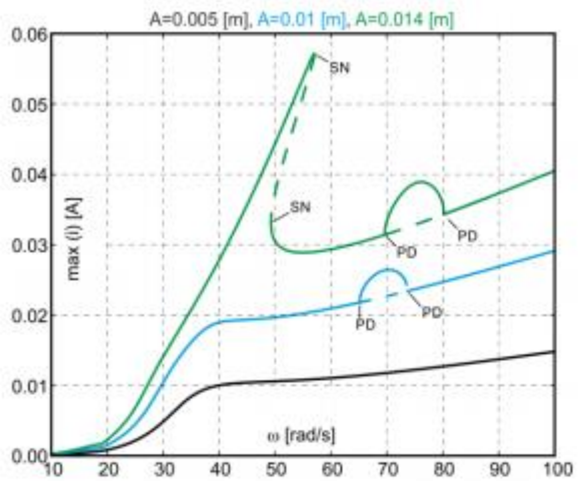
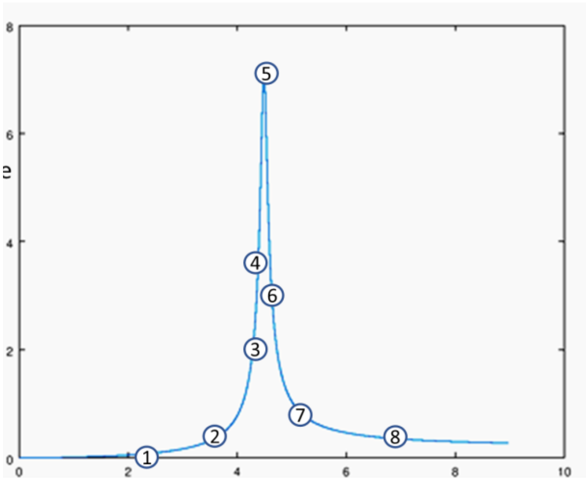
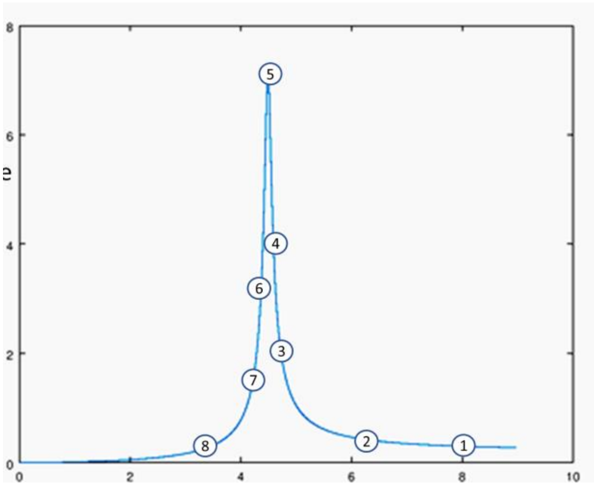


Fig. 6. Recovered current for the various excitation amplitude, for k=38.7N/m and R=2.3kΩ

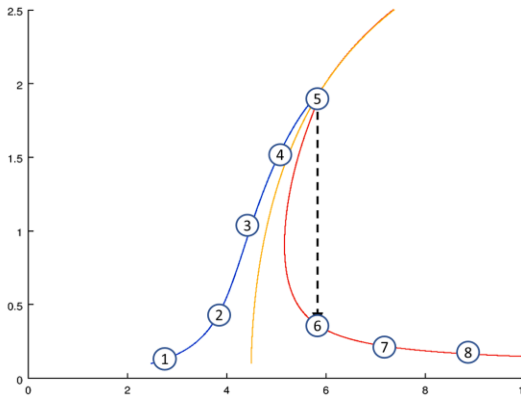
The Jump Phenomenon

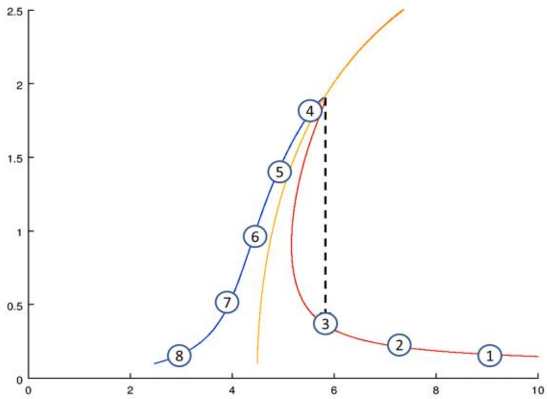
To understand the jump phenomenon, let's take the case of the *linear* system where this does not happen. Say we take a particular forcing amplitude and slowly increase the forcing frequency, from points 1, 2, 3, …8 shown below left. The response amplitude passes through a peak. If we repeat the experiment starting with a high forcing frequency and steadily reduce it, we get the same curve, but with points in a different order, bottom right,





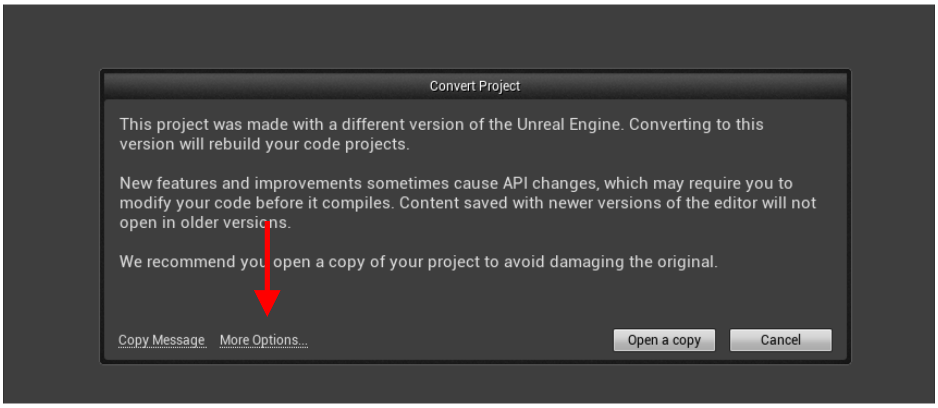
Now, a system that shows the jump phenomenon behaves differently. When we steadily increase the forcing frequency we get the curve below left which has two 'branches' the blue then the red. Here the response amplitude increases through point 1, 2, ..5 on the blue branch, then suddenly drops and continues on the red branch 6, 7, 8. If we start at a high frequency then reduce it, we traverse the red branch until we hit point 3 where we jump up to point 4 then continue down the blue branch. Interesting stuff.



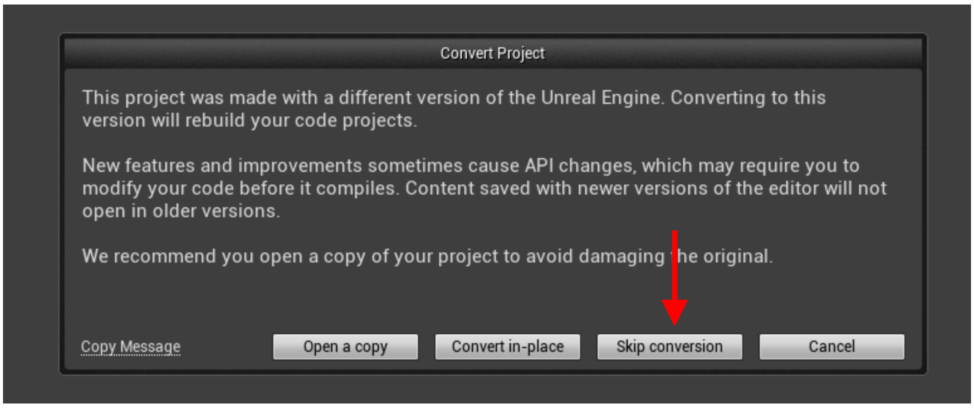


Unreal on the Lab Machines

Download the Unreal 4 Assets zip. Click on the zip and drag the folder onto the Desktop which will unzip. Open Unreal 4, and hit **More** then **Browse** and find the file **MAS22_SciencePark.uproject** which you should open. A number of things may happen depending on which computer you are working on. Most likely you will be faced with this dialogue box, in which case select **More Options**.



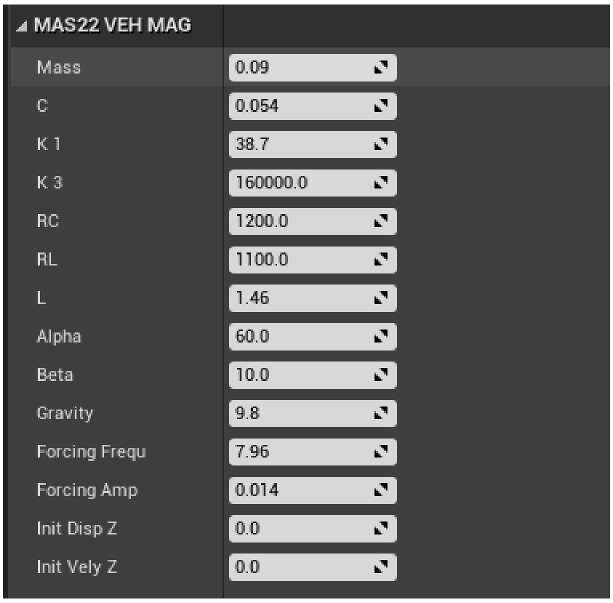
Now a second dialogue will pop up, and choose **Skip conversion**.



Hopefully the level will open.

Default Parameters

You will find the following defaults in Unreal-Ed.



Maths Model

This is quite straightforward and consists of two parts, one is the mechanics of the oscillating magnet, the second is the induction of the voltage in the coil. Here's the magnet dynamics

$$\frac{dz}{dt} = v$$

$$\frac{dv}{dt} = \frac{1}{m}(-k_1 z - k_3 z^3 - cv - \alpha i) - g + \Omega^2 Asin(\Omega t)$$

and here's the induced current

$$\frac{di}{dt} = \frac{1}{L}\big(\alpha v - (R_C + R_L)\big)$$

If we look at the frequencies in Fig.5 $\omega = 40$ corresponds to about 6.4 oscillations per second. This is too large to visualize using UDK so we need to slow down the oscillations, say by a factor of 10. We can do this by *scaling* the above equations. We do this by replacing *t* by βt where the scaling factor $\beta = 10$. The above equations transform to

$$\frac{dz}{dt} = \frac{1}{\beta} v$$

$$\frac{dv}{dt} = \frac{1}{\beta}\left[\frac{1}{m}(-k_1 z - k_3 z^3 - cv - \alpha i) - g + \Omega^2 Asin(\Omega t)\right]$$

$$\frac{di}{dt} = \frac{1}{\beta}\frac{1}{L}\big(\alpha v - (R_C + R_L)\big)$$

and it's these equations we code. You can see that since $\beta = 10$ is on the bottom of the rhs's, then these are smaller by a factor 10 which slows everything down by this factor.