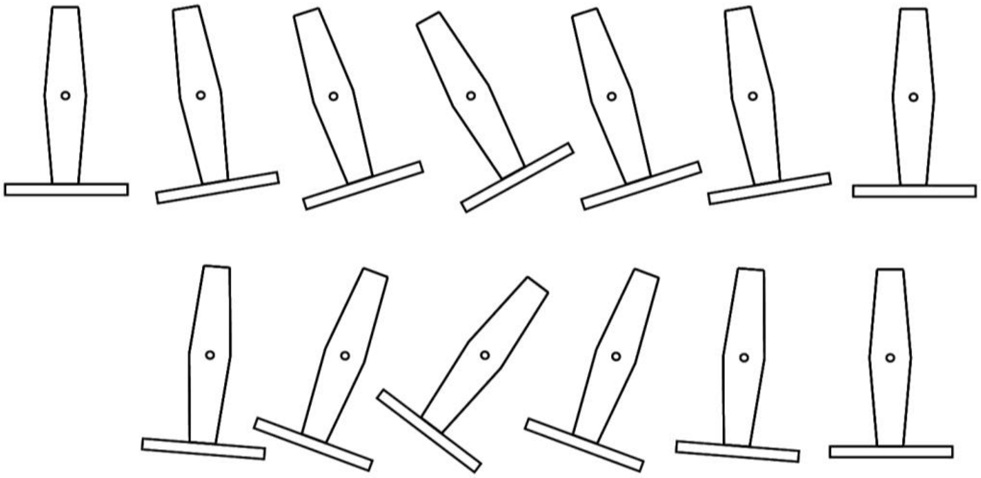
The Skymaster

The Mathematical Model

The SkyMaster is rather like a pendulum. Give it a push and it will swing back and fro. Here's a sketch showing several steps in the oscillation. Note here the oscillation is a rotation, unlike the displacement of the monster truck.



The *amplitude* of the oscillation is the maximum amount of movement of the object, here rotation. For the above figure I guess it's around 15 degrees. The *period* of the oscillation is the time taken to complete one whole swing, from any starting angle back to that angle and back again (Think – what is a "whole swing"?). Without any friction or *damping* the SkyMaster will oscillate for ever, we need to include *damping*.

What makes the Syymaster move? Well, it's the guy in the control cabin. He's got a joystick so when he moves to the left and the Skymaster will get a push (an *impulse*) in a clockwise direction and when he moves it to the right the Skymaster will get an impulse in an anti-clockwise direction. So the operator can build up the amplitude of the oscillations by applying successive impulses, and of course send the Skymaster over the top.

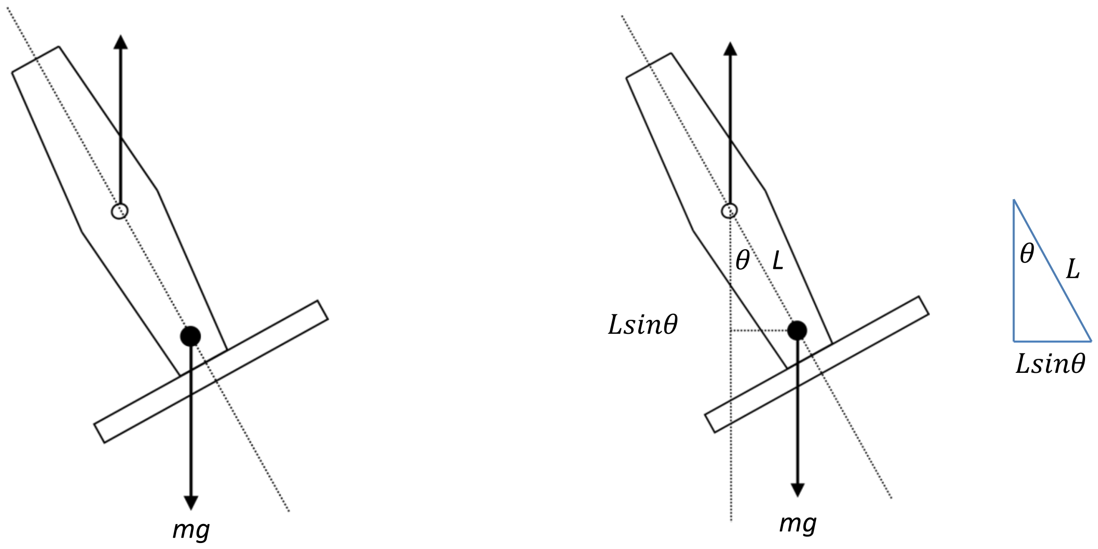
Now let's develop the equations for rotation. I'll write these alongside the equations for displacement in 1D. You'll notice they are **identical** apart from the use of different symbols of course. Now let's write down the equations for the motion of the Skymaster. Since we have rotation instead of translation we need some different symbols. Let's make a comparison

Translation		**Rotation**	
Force	F	Torque	T
Acceleration	a	Angular Acceleration	α
Velocity	v	Angular Velocity	ω
Displacement	z	Angle	θ
Mass	m	Moment of Inertia	I

The symbol θ (theta) is the rotation (here calculated in *radians*). The symbol ω is the "angular velocity", that means how fast theta, the rotation changes. Finally, α is the "angular acceleration", how fast the angular velocity changes.

When we are considering rotating objects, we have to use “torque” T instead of force, and “moment of inertia” I instead of mass. That will get a little more clearer as we move on.

In the case of translation, we started with force, calculated acceleration then used this in the computation of the speed changes and hence the displacement changes. Now we start must start with the torque then calculate the angular acceleration. Before we look at the torque, let’s look at the forces on the skymaster. We assume that the mass is effectively located at the lower point below. The downward force on the skymaster is *F = mg* as we have used below on the left.



Now to work out the torque on the ride due to the mass of the ride, we must multiply the force by the distance from the force arrow to a line parallel through the centre of rotation. This is shown in the centre diagram, with detail on the right. From this we find the expression for torque to be

$$\tau = -mg(Lsin\theta)$$

In addition to this torque, there is an additional torque from the motor which drives the ride when the operator presses a button, this is τ_m and there is an extra torque due to damping which is proportional to the angular velocity of the ride. This is $-b\omega$. To the total torque on the ride is

$$\tau = -mg(Lsin\theta) - b\omega + \tau_m$$

Now the angular acceleration is just torque divided by the moment of inertia of the rotating object

$$\alpha = -\frac{T}{I}$$

and since the moment of inertia is, for this situation $I = mL^2$ then the expression for angular acceleration becomes

$$\alpha = -\frac{T}{mL^2}$$

Now we have an expression for the angular acceleration we can find the angular velocity and the angle of the ride.
Finally we must consider τ_m the torque on the ride due to the motor. We shall take the simplest case of sinusoidal driving (a sine wave) with Amplitude A_d and period T_d seconds, like this

$$\tau_m = A_d \cos\left(\frac{2\pi t}{T_d}\right)$$

So **let's summarize**. Let's return to the equation for the total torque on the ride

$$\tau = -mg(Lsin\theta) - b\omega + \tau_m$$

The third term (from the motor) is positive and makes the ride accelerate in rotation. The second term is damping and as expected, reduces the effect of the motor torque. The first term also works against the motor, causing a torque in the opposite direction.

Analytical Solutions

1. The Period of Free Oscillation of an undamped pendulum. This is the case where there is no driving, but the Skymaster starts with an initial angle (theta). Then the period is *for small angles is*

$$T = 2\pi\sqrt{\frac{L}{g}}$$

Interestingly, if you model your leg as a pendulum, this will give the time for a stride or two. Try it out. When the angle of the pendulum is large, then the period is given by

$$T = 2\pi\sqrt{\frac{L}{g}}\left(1 + \frac{1}{16}\theta_i^2\right)$$

where θ_i is the initial angle. This is applicable to the Skymaster when it is not driven, but when it starts from some initial angle.

2. The Period of Free Oscillation of a damped pendulum. With damping coefficient *b* the period now becomes

$$T = \frac{2\pi}{\sqrt{\frac{g}{L} - \left(\frac{b}{2mL^2}\right)^2}}$$

Again you can use this only very approximately to the Skymaster, but could be a useful comparison.

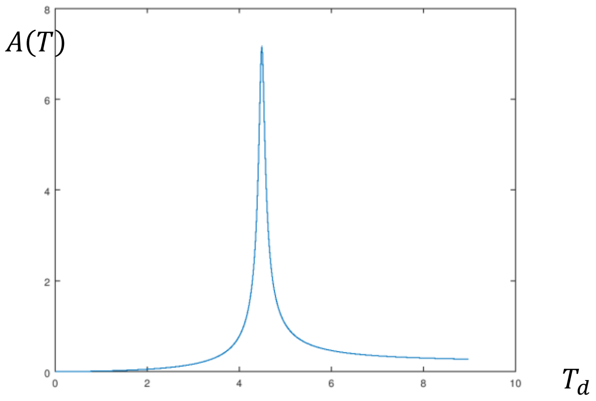
3. Forced Oscillations of a damped pendulum. When the pendulum is driven with a force which is periodic with amplitude $A_d$and period T_d this is represented as a new torque like this (where $\omega = 2\pi/T_d$)

$$\frac{A_d}{mL^2}\cos(\omega t)$$

The amplitude of the steady oscillations is then given by the following expression

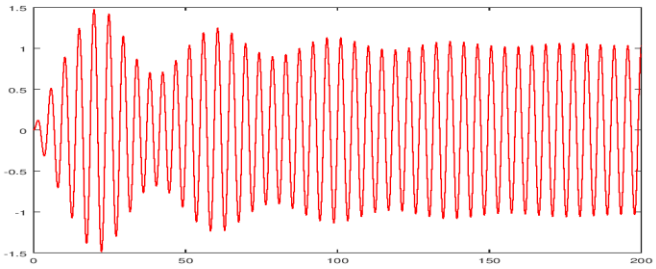
$$A(T) = \frac{A_d}{mL^2}\frac{1}{\sqrt{\left(\frac{g}{L} - \omega^2\right) + \left(\frac{b}{mL^2}\right)^2 \omega^2}}$$

This looks quite scary, but plotting it out produces the following curve where you can see that the amplitude of the Skymaster's oscillations peaks at some value of period.



Please note **this result only applies for small angles of oscillations, so does not apply to the Skymaster** which of course makes it more interesting.

One final piece of information which you will experience when you collect data and make your plots. When the drive is first applied to the Skymaster there is a *transient* oscillation where the angle varies wildly. After a while, the angle settles down to a *steady-state* value. Here's an example



The transient appears as a low-frequency 'modulation' of the higher forcing frequency.

Possible Investigations (Straightforward)

Here I will make suggestions, and these will be "staged" to help you develop a great investigation.

Stage 1.

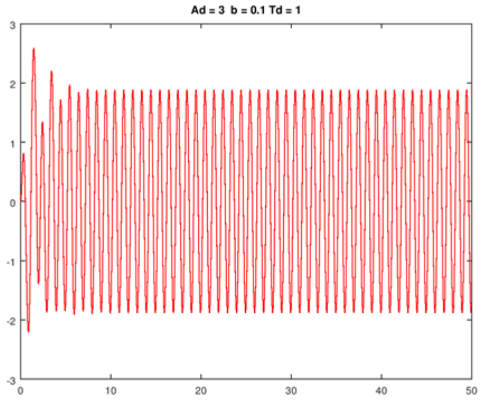
Use the default parameters from the following table.

armLength	0.2824
mmass	1.0
gravity	9.8
armDamp	0.1
drivePeriod	1.0

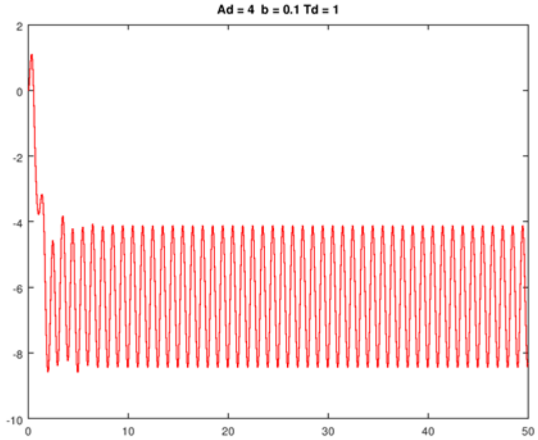
Change the **driveAmp** and record the **maximum angle**. The HUD will give you angles in ‘radians’ To convert to degrees, then multiply by 180 and divide by pi (3.141529) if you like, but not required. You could do this in your HUD code. Here’s a table for your results

driveAmp	max angle
0.5	
1.0	
2.0	
…	
5.0	

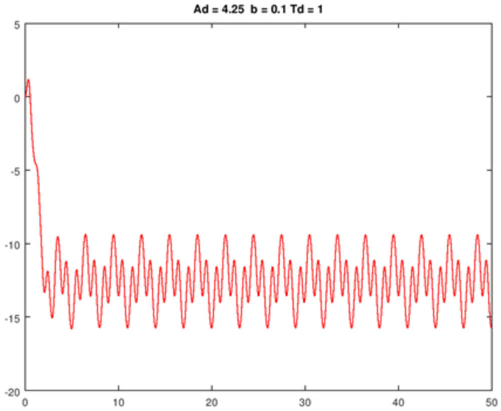
Try values of driveAmp from 0.5 to 5.5. You will find a range of results. Here are some examples of behaviour you might find: The figures show the angle (vertical) versus time (horizontal). You might just be able to read the parameters I used.



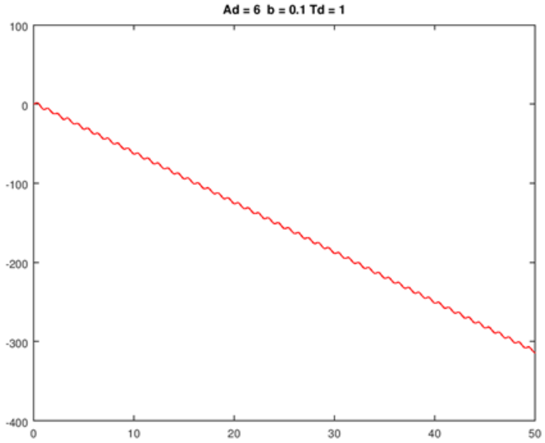
Simple result. The Skymaster oscillates with an amplitude just under 2.0 after an initial ‘transient’



The Skymaster rotates ‘over the top’ then settles down to nice oscillation



Very Very interesting. There are 3 cycles in each repeated motion. Stuff for “Phase 2”



Here it’s going over the top again and again, just rotating (with a bit of a wiggle)

So what should you do? Well, collect diagrams like the ones above, and see if you can find any pattern as you increase driveAmp as suggested above. To calculate the "max Angle" measure the difference between the top and bottom of the oscillation and divide by 2. E.g., for the top-right example the calculation (approx) is (-4 - -8)/2 = 4/2 = 2.

The Jump Phenomenon (advanced investigation)

Systems that oscillate such as musical instruments have a natural period of oscillation. At the piano, as you move up the keyboard to the right, the period of the note gets smaller. Also, a garden swing has a natural period. If you are pushing someone with a drive-period below the value of its natural period of oscillation (i.e. too quickly), then the swing will not build up a large amplitude of swing. That's on the left of the diagram (below left). Also if you push it too slowly, so the drive-period is above its natural period (on the right of the left diagram) then the same happens. Only if you push the swing close to its natural period, then the amplitude becomes enormous, that's the peak of the left diagram. You will have found something similar in your investigations. So if you slowly increase the period from 0.0 towards 10.0, then the amplitude would follow the curve smoothly, rising up, peaking and then falling.

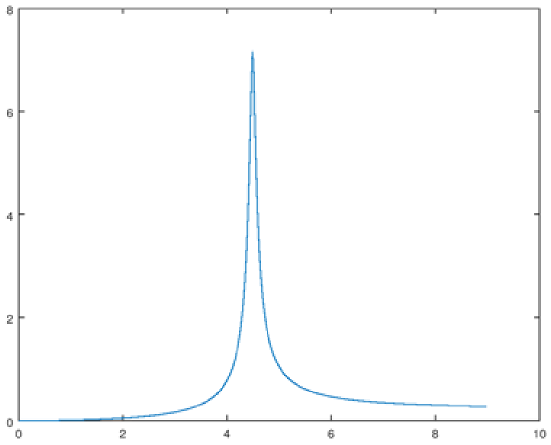
But the left diagram applies to *linear* systems. That means systems where the restoring force is proportional to the displacement. If the Skymaster were linear then its expression for torque would look like this. The first term on the right shows the restoring force is proportional to the Skymaster angle θ

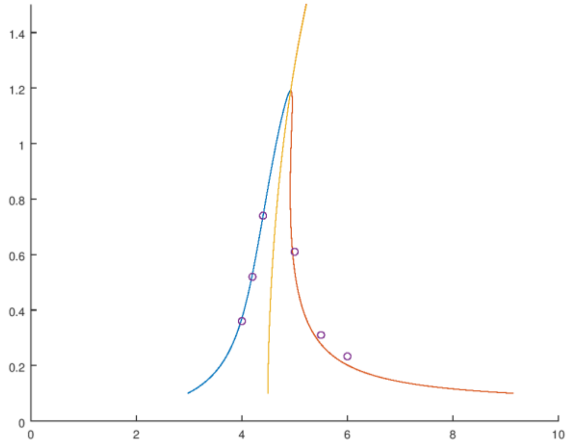
$$\tau = -mgL\theta - b\omega + \tau_m$$

But we know this is not true, the torque is proportional to the sine of theta $sin\theta$ which is a non-linear function like this

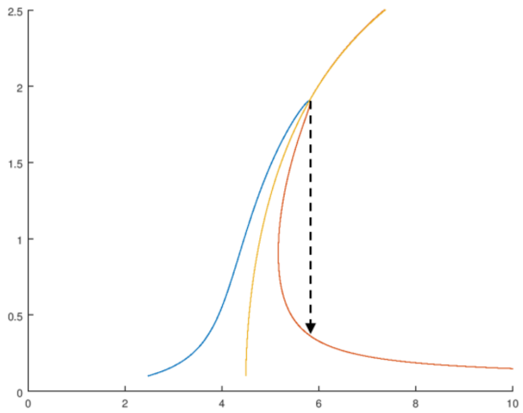
$$\tau = -mg(Lsin\theta) - b\omega + \tau_m$$

This non-linearity changes the period amplitude curve from the one on the left (below) to the one on the right. The curve has been "bent over" to the right.





The consequences of this are quite interesting. Look at the diagram below where the bend has been exaggerated. If we start with period 2.5 and gradually increase this, then we move up the blue curve, the oscillation amplitude is increasing from 0.1 towards 2.0. But we run out of the blue curve, so where do we go? The only possibility is that the amplitude suddenly jumps down as shown by the dotted arrow. This is called the "jump effect" and the Skymaster should display this behaviour.



To investigate this, I suggest using the following parameter set, “slow parameters”. This will produce data which lie on the ‘bent over’ curve reasonably well. The standard parameter set will show the effect, but the agreement with the theory (presented below) will not be so good. The above graph (on the right) was created with these parameters.

armLength	5.0
mmass	1.0
gravity	9.8
armDamp	3.0
driveAmp	5.0

[Some theory here based on the reference below]

http://elib.sci.am/2011_4/03_4_2011.pdf

The process of collecting data for the jump phenomenon can be automated. The drive amp can be progressively increased with time in the solver file. For example in Octave this would start off like this:

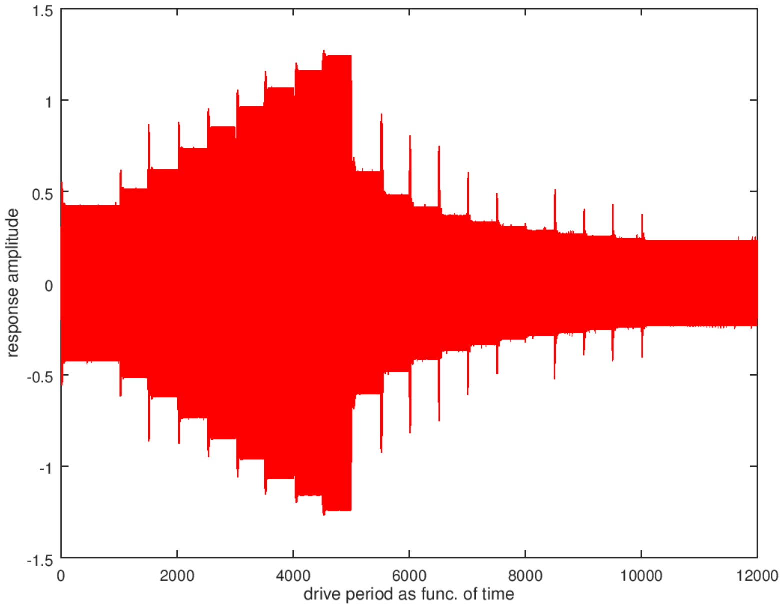
```
if(t < 1000) Td = 4.1;
elseif( (t > 1000) && (t < 1500))  Td = 4.2;
elseif( (t > 1500) && (t < 2000)) Td = 4.3;
elseif( (t > 2000) && (t < 2500)) Td = 4.4;
elseif( (t > 2500) && (t < 3000)) Td = 4.5;
```

and end like this

```
elseif( (t > 8500) && (t < 9000)) Td = 5.7;
elseif( (t > 9000) && (t < 9500)) Td = 5.8;
elseif( (t > 9500) && (t < 10000)) Td = 5.9;
else Td = 6.0;
endif
```

Similar code would work for the UDK solver.

The output, plotted with Octave will look like this



where you get the drive period from the time, and read off the drive amp. This plot clearly shows the jump phenomenon, where the drive amp initially increases then suddenly drops. Data from this plot can be extracted and placed on the "bent over" curve.

Files Required (Octave)

solve_DDHO_Sweep.m	main program
rhs_ddho_1.m	solver called by main program
jump.m	plots the jump curve and your data

Period-doubling bifurcation and the route to chaos (advanced investigations)

Perhaps you met the idea behind this work in the extension worksheet on oscillators. Here you will investigate period-doubling bifurcations in the context of the Skymaster. There is a separate "From Maths to Code" document which links to the lecture slides from Jen-Hao Yeh which beautifully explains these concepts. You can do this in **UDK** or you can choose to use the Octave files

solve_DDHO_Chaos.m
rns_ddho_ch.m