Beyond the spring-mass-damper...

This Application Note has been written to give the reader some suggestions for extending the classic Spring-Mass-Damper model to a more complex real-world problem – bungee jumping!

This is not a definitive "How To" guide for creating a bungee jumping model but is a set of suggestions for self-guided investigations into the problem. As with most analog computer setups, there is no "right" way of solving a problem, only setups which work and those that don't – although some will be more elegant than others!

1 Basic spring-mass-damper setup

Figure 1 shows a general representation of the spring-mass-damper problem. The "THAT – First Steps" booklet includes, in section 8.1, an excellent discussion of the basic spring-mass-damper problem and includes a worked example which assumes that:

- There is no forcing function to drive the body.
- The damping is proportional to the body's velocity.
- The Spring is linear and acts equally in tension and compression.
- The body is given non-zero initial conditions for velocity and/or position.

Figure 2 shows a modified version of the "First Steps" model, in which a forcing function, gravity, has been introduced and the initial values of velocity and position have been set to zero.

Set up this model and get it working in REP OP mode with nice, damped oscillations. Note that it should reach a steady state in which the downwards force exerted by gravity equals the upwards force exerted by the spring. Also, notice that a spare inverter has been used to give a +velocity signal.

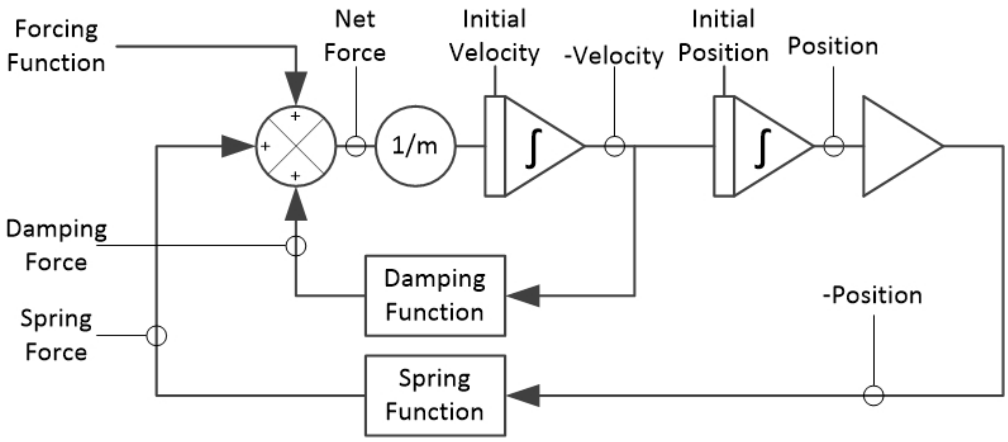

Figure 1: Generalized spring-mass-damper problem

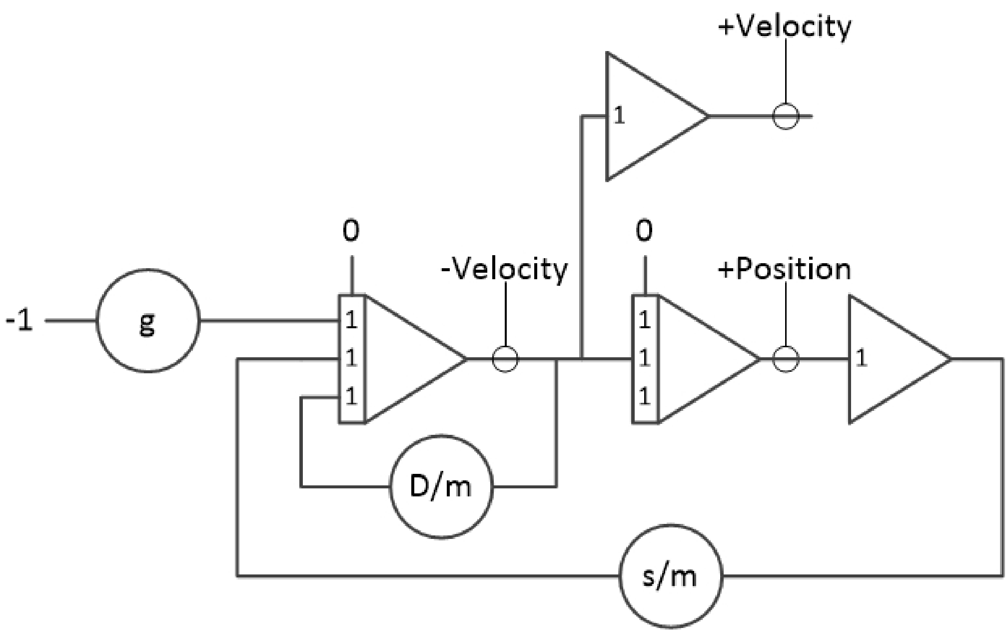

Figure 2: Setup for the spring-mass-damper problem

It much easier to see, at a glance, what is happening if the velocity and position traces on the display device have the same sign, rather than having to mentally invert one.[1]

This model is a good basis for the subsequent investigations.

2 The bungee jumping problem

The coordinate system used here assumes that the jumper starts at an elevation of zero and that the ground is at -1 machine unit. This means that downwards forces, velocities and displacements have a negative sign, which makes sense while developing the model. If the jumper reaches -1, then he is in trouble!

As the bungee jumper simply steps off the platform, the initial position and velocity are both zero using this coordinate system. This means that no initial conditions need to be specified for the two integrators. The jumper is subject to three forces:

- The gravitational pull downwards, $-mg$, which is constant.
- Aerodynamic drag, approximately proportional to the square of the velocity, Dv^2, where D is a drag coefficient. Note that this force always opposes the instantaneous direction of motion.
- The force exerted upwards by the bungee cord, which is proportional to the extension of the cord, $-s\delta y$, where s is the spring constant of the cord and δy is the extension of the cord.

These three forces, scaled by (1/mass), are the inputs to the first integrator.

[1] Alternatively, the signal could normally be inverted in the display device, but it can cause confusion if the operator forgets about this inversion when the probe is connected to a different signal!

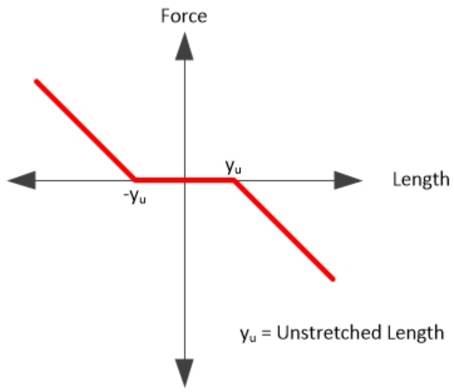

Figure 3: Force vs. length for bungee cord

3 The bungee cord model

With a bungee jumper, the cord is slack when he jumps and only goes into tension once the jumper has fallen past the cord's unstretched length. If the cord is unstretched, it cannot exert any force on the jumper, unlike a spring which produces forces in both compression and extension. The upwards force exerted by the spring is given by $-s(y-y_u)$, where s is the spring constant, y is the instantaneous position of the jumper and y_u is the cord's unstretched length. Although in this example the jumper's position can only be below the starting position of $y=0$, if the jumper were above $y=0$, the cord would exert a downwards force once the position was above the cord's unstretched length.

Figure 3 shows how the force varies with position. The graph exhibits classic deadband behaviour – and there are many published analog computer models for simulating a deadband.[2] However, as most of these involve the use of an ungrounded potentiometer, which is not provided in THAT, we need to find an alternative solution.

Making use of the fact that y can never go positive in this problem (i. e. the

[2] EAI – Handbook of Analog Computation, 1967.

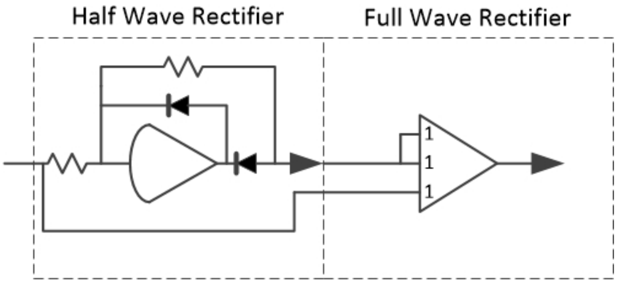

Figure 4: Ideal rectifiers

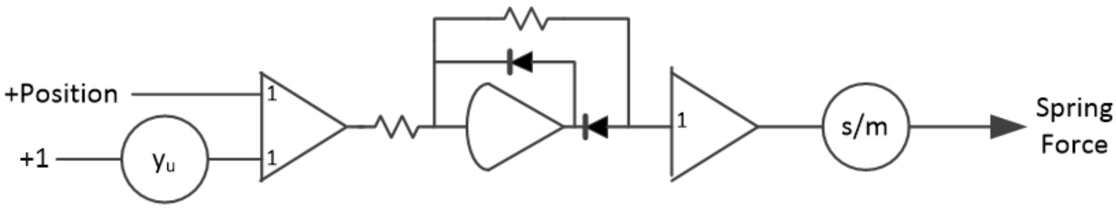

Figure 5: Bungee cord model

jumper cannot end up higher than the position he started from – simple conservation of energy), the function can be modelled with $y = 0$ for $y > -y_u$ and the force then becomes a half-waved rectified version of $-(y - y_u)$.

Figure 4 shows the classic implementation of ideal half-wave and full wave rectifiers. They use an open amplifier (i. e. an op amp without any feedback element) and this can be implemented on THAT by linking a summer's FB and ground connections, which removes the normal feedback resistor. THAT has sufficient diodes to implement two rectifiers and summer input resistors can be used as the feedback elements. Consequently no external components are required for this setup.

Figure 5 shows the bungee cord model. The unstretched length of the cord is subtracted from the instantaneous length and is the extension of the cord, δy. This is fed through a half wave rectifier and then inverted and scaled to give the cord force which is acting on the jumper.

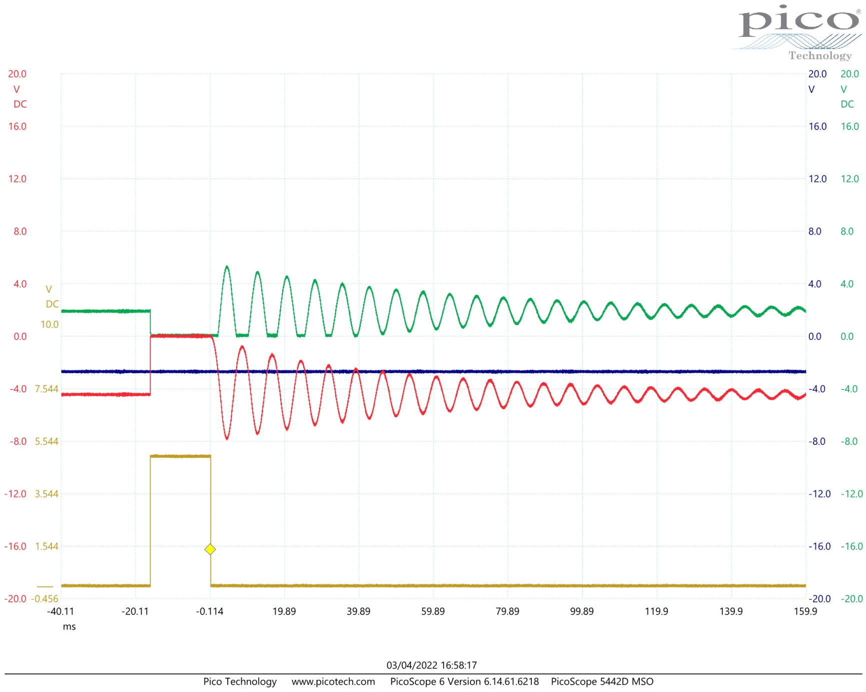

Figure 6: Bungee cord behaviour

Figure 6 shows:

Red trace: the position of the jumper.

Blue trace: the unstretched length of the cord.

Green trace: The cord force acting on the jumper.

Note that the cord only exerts a force when the position is below the unstretched length.

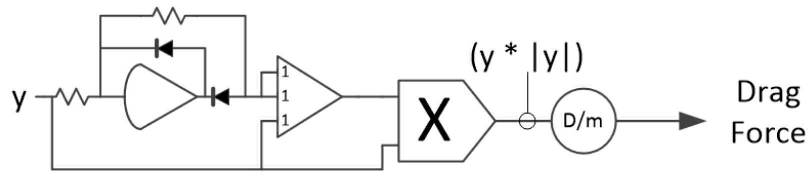

Figure 7: Aerodynamic drag model

4 The drag model

The aerodynamic drag is a function of the square of the velocity. Whilst it is simple to calculate the square – the velocity is fed into both inputs of a multiplier – the squaring operation looses the sign information, so it is necessary to have a method of determining if the drag is acting as an upwards or downwards force.

The simplest method is to calculate $v|v|$. As $|v|$ is always positive, the sign of the result is determined by the sign of the velocity. This can be implemented using the setup shown in Figure 7, where a full wave rectifier is used to generate $|v|$, which is then multiplied by the velocity and the result scaled to give the drag force.

In Figure 8, the green trace shows the output of this setup when it is driven with the triangular wave shown by the red trace. The quadratic shape can clearly be seen.

5 Using comparators

It is also possible to use comparators to create the drag and bungee cord models. For instance, to create the drag force, the velocity could just be squared and a comparator used to select either a positive or inverted version of the squared signal.[3]

[3] I have been told that some people abhor the use of "digital" computing elements such as comparators. . .

Figure 8: Drag model behaviour

6 Assembling the bungee jumping set-up

A confident user of THAT could take the information contained in this note, scale the problem and implement it directly! However, a more experimental approach is recommended for those new to analog computing. This will not only make fault finding simpler but will also give better insight into what is happening. This approach is outlined below:

- Implement the basic spring-mass-damper setup with linear drag and spring forces and get it working.
- Implement the drag model and connect its input to the output of the first inte-

grator (−velocity). Observe the output from the drag model and compare it with the linear drag signal. Check that the phase is correct and invert something if it isn't. Replace the linear drag signal into the first integrator with the output from the drag model and see what happens. Adjust pots if necessary to get nice, damped oscillations.

- Implement the bungee cord model and connect its input to the output of the second integrator (+position). Observe the output from the bungee cord model and compare it with the linear spring signal. Check that the phase is correct and invert something if it isn't. With the unstretched length pot set to zero, the signals should be identical. See how the bungee cord signal changes as the unstretched length is increased. Replace the linear spring signal into the second integrator with the output from the bungee cord model and see what happens. Adjust pots if necessary to get nice, damped oscillations. See what happens when the unstretched length and/or the elasticity of the bungee cord are changed.

This is the complete bungee jumping setup.

7 The x/y view

These investigations will yield sets of damped oscillations – and the shape of the oscillations will vary depending on which non-linearities are included in the setup.

The effects of these non-linearities are more apparent if velocity is plotted against position. The linear case will give a nice spiral, while the different non-linearities will change the shape of the spiral. Figures 9 and 10 show velocity plotted against position for the linear case and for the quadratic drag. In the latter plot, it is clear that the damping effect diminishes significantly as the amplitude falls, resulting in a dense set of loops in the centre as the oscillations take longer to diminish.

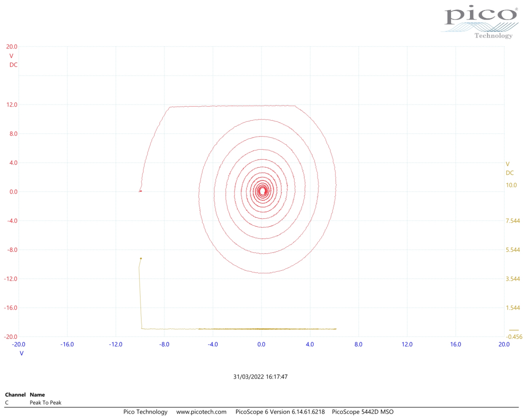

Figure 9: First simulation result

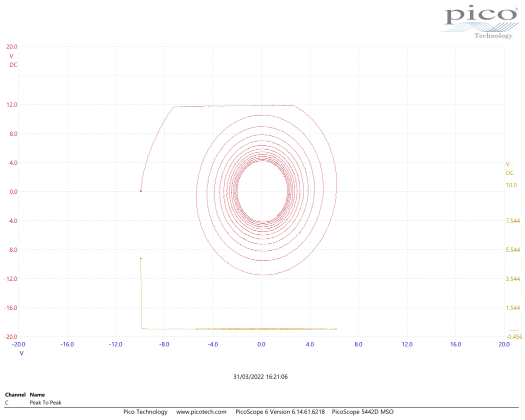

Figure 10: Second simulation result

8 Further investigations

Further investigations into the bungee jumping problem can be carried out using the setup which has now been developed. Possible avenues of exploration include:

- Calculate a spring constant and unstretched length of cord which won't result in the jumper being subjected to intolerable deceleration.
- Plot the energy in the system during the computer run. This would be the sum of:
  - The body's potential energy.
  - The body's kinetic energy.
  - The energy stored in the bungee cord.
- It has been assumed that no energy is lost in the bungee cord. Work out how to incorporate such a loss in the bungee cord model.

Happy analog computing!