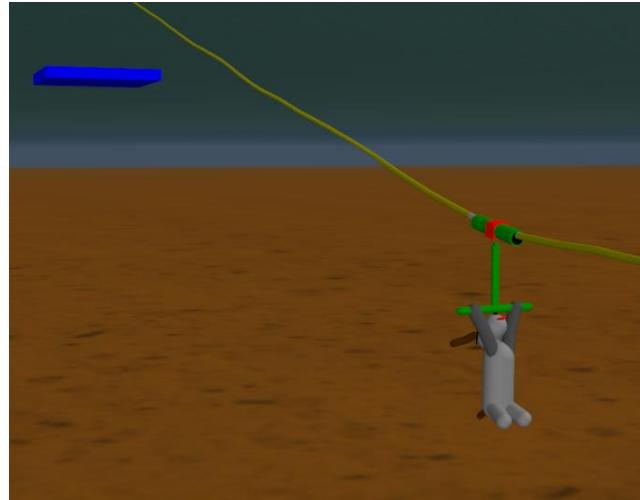


Z02 Line on line slide



Introduction

In this example, we introduce OrcaDog, our adventurous office canine.

We've sent OrcaDog to play on a zip wire, so that we can demonstrate the two types of line-to-line contact that are available in OrcaFlex.

On opening the simulation file, the default workspace will set up two shaded graphics views of the model. Run the replay to see OrcaDog slide down the zip wire. If the simulation replays too quickly or too slowly, you can change the speed relative to real time via the *View menu (edit replay parameters)*.

Building the model

The system is submerged but the seawater density has been set at 1.28kg/m³, the density of air. We have hidden the sea in the model view by right mouse clicking on the *environment* object in the model browser and selecting *hide sea*. Alternatively, we could have modelled the whole system above the sea surface.

OrcaDog is modelled as a 6D buoy. Properties for a typical small-medium sized dog have been applied, but drag and added mass are ignored as he is moving in air, so generated loads will be negligible. The front paws, ears and tail are modelled using lines, so that they can flex. A number of drawing-type shapes are attached to the buoy to make the eyes, nose, sunglasses and tongue. Adding this level of detail is not necessary (as the fact that OrcaDog is wearing sunglasses will not affect his dynamic behaviour (!)), but the use of *drawing*-type shapes in this way will not adversely impact the simulation time, as they are only providing a visual shape and will not interact with any other objects in the model.

The ropeway is a simple line inclined and held at both ends. The travelling block is modelled as a tube (another line), which slides along the ropeway, with a T-bar handle arrangement hanging from

it. The lines used to make the travelling block are connected together using two buoy objects to act as connection points. This is because you can only connect to a line at the nodes, and in this case using single segment lines is appropriate, so there are no nodes at the mid-points. Alternatively, the connecting buoys could be removed and more segments added to the lines so that connections could be made in the required places. A 3D buoy is used to connect the two parts of the T-bar together because this is a rigid connection, so rotations about the connection point are not important. A 6D buoy is used to connect the T-bar to the block because we *do* want rotations to be modelled about this point.

Line contact is used to create the relationship between the travelling block's *Tube* line and the ropeway. Open up the *line contact data* form (accessible from the model browser) to see this. *Line contact* differs from *line clashing* in a number of ways. Firstly, *line contact* is active in statics. This means that contact between the tube and the ropeway is recognised and included in the static calculation, rather than suddenly becoming active at the start of dynamics.

In addition, *line contact* allows friction effects to be included between the two lines. In this case, a friction coefficient of 0.15 has been applied. We have identified the ropeway as being the penetrating line and the tube as the splined line. We have also said that the ropeway is 'Inside' the tube. This means that the tube is modelled with a smooth splined bore surface, which makes contact with penetrator 'balls' that are located at each node on the ropeway line.

More details on the available line contact relationships, and some discussion about the differences between the *line contact* and *line clashing* methods, can be found in the help file, [Modelling, data and results | Lines | Line contact | Modelling](#).

The second type of line-on-line contact, *line clashing*, is used to model the brake line, which catches the T-bar of the travelling block in order to slow it down. This type of contact is used to model clashing events such as this, or for analysing interference between riser and mooring lines etc., but it is worth noting that *line clashing* is not active during statics, and doesn't allow friction to be applied between the two lines.

Both of the line-on-line contact methods require 'switching on' for them to be active in the model. *Line contact* is activated by setting up the relationship between two specific lines on the *line contact* data form, *line clashing* is activated by ticking the *clash check* boxes on each line's data form, and by ensuring that the line types involved have non-zero line clashing stiffnesses assigned to them.

Therefore, to prevent the T-bar passing through the *BrakeLine*, both lines must have a *line clashing stiffness* assigned and *clash check* selected for the relevant line sections. The clashing stiffness should be sufficiently high that one line cannot push through the other, but the actual value is not important in this case. Matters become more difficult if accurate loads are needed for design purposes. For further details, see the help file [Theory | Line theory | Clashing](#).

Note that the clash check will result in slower simulation run time, so should only be applied where required. Because you can have contact applied section by section, it is possible to have parts of the line active and other parts with contact turned off.

A link is used to hold the travelling block at the top of the ropeway until the start of dynamics, when it releases allowing the block to slide. This is done on the data form for the link called *Stop*, where the link is set to *release at start of stage 0*.

This model is run with the explicit time integration scheme. Contact between lines will also work with implicit time integration schemes but in this example the lines are in air so have little damping. The resulting sudden changes in loading are better captured with the explicit scheme, which is more robust with discrete events.

The right hand of the two views that the default workspace shows (select *Use file default* from the *Workspace* menu if you have changed the views at all) has the viewpoint moving with the dog, rather than staying fixed in space, as the left hand view does. This has been set up by adding a vessel object (called *Camera*) that has been given a prescribed path to follow via an internal time history file. The view point of any OrcaFlex view can be made relative to any vessel, 6D buoy or shape object in the model. This means that if the object moves then the viewpoint will move with it. In this case the view has been set relative to the *Camera* object. Open the *edit view parameters* data form (shortcut *Ctrl+W*) to see where this is done.

Results

Observe the animation through the whole simulation. Notice that the blue-coloured *BrakeLine* highlights in white when the T-bar clashes with it – this is *line clashing* in action. The green tube sliding down the yellow ropeway line is *line contact*.

Load up the workspace file [Z02 Line on line slide results.wrk](#). The upper left hand graph shows an instantaneous range graph of *line contact force* for the *Ropeway* line. If you run the replay you will see the contact force progressing along the line until contact is made with the brake. The upper right hand graph shows the *clash force* between the T-bar and the *BrakeLine*, and the lower left hand graph shows the *time history* plot of the *clash force* from that event, showing that (as you would expect) the initial impact gives the highest clash force followed by intermittent contact as the T-bar rebounds repeatedly.

Now go to the *results* form and select *line clashing report* for the *BrakeLine* for the *whole simulation*. This gives details of the magnitude and duration of each clash event.