

## Analysing rotational motion using Tracker

This document assumes you have worked through *Analysing linear motion using Tracker* and can set axes, change the quantities displayed on graphs etc.

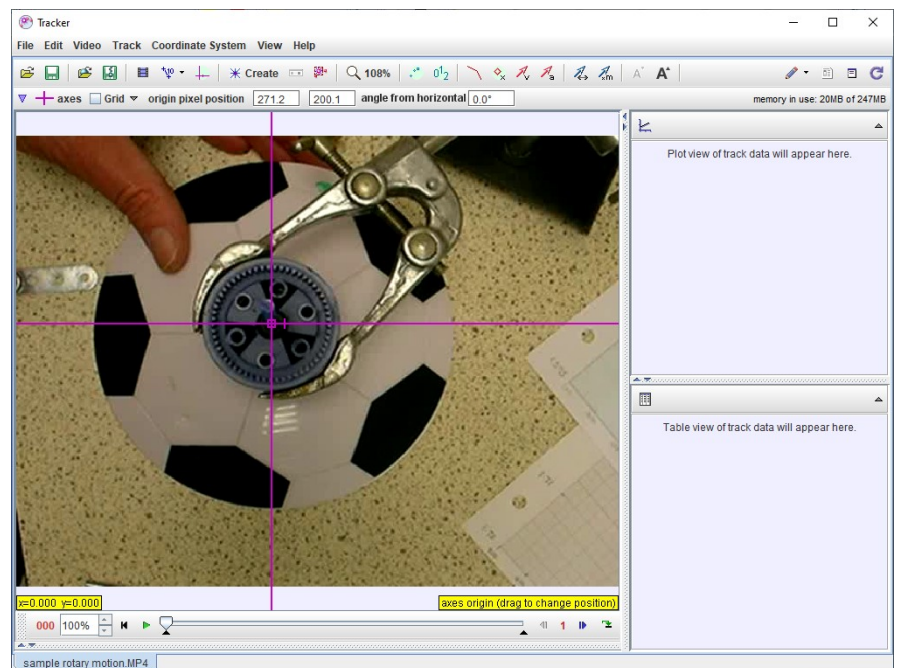
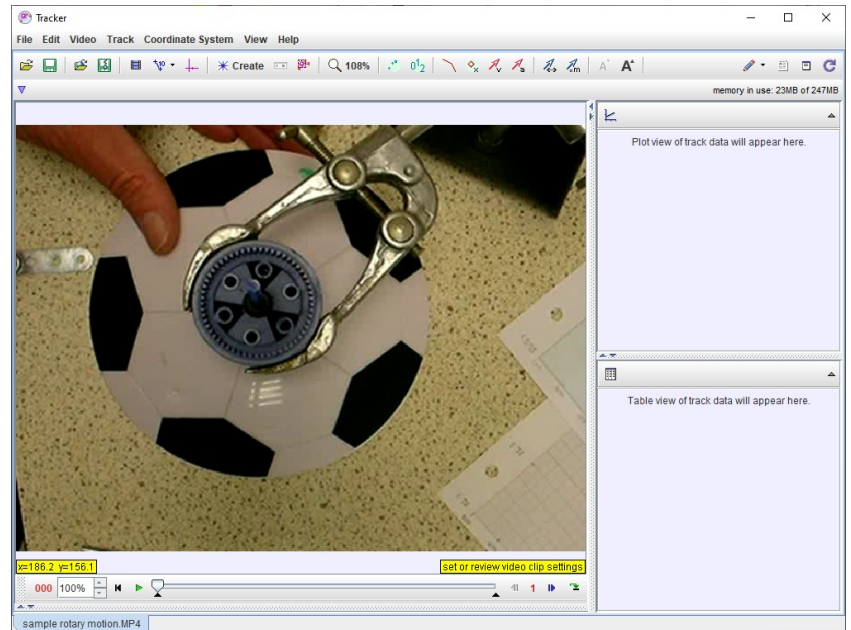
Launch Tracker and import the video *Sample rotary motion.mp4*.

This is a video of an ingenious experiment devised by a delegate at a SSERC course. This is an overhead view of a 'hover puck' with an axle through its centre. A string connected to a falling mass via a pulley has been wrapped round it. As the mass falls, the string unwinds, causing the puck to rotate.

The relationship between torque and angular acceleration can be investigated by finding the angular acceleration for different falling masses (and hence different torques).

We will determine angular acceleration from the video clip.

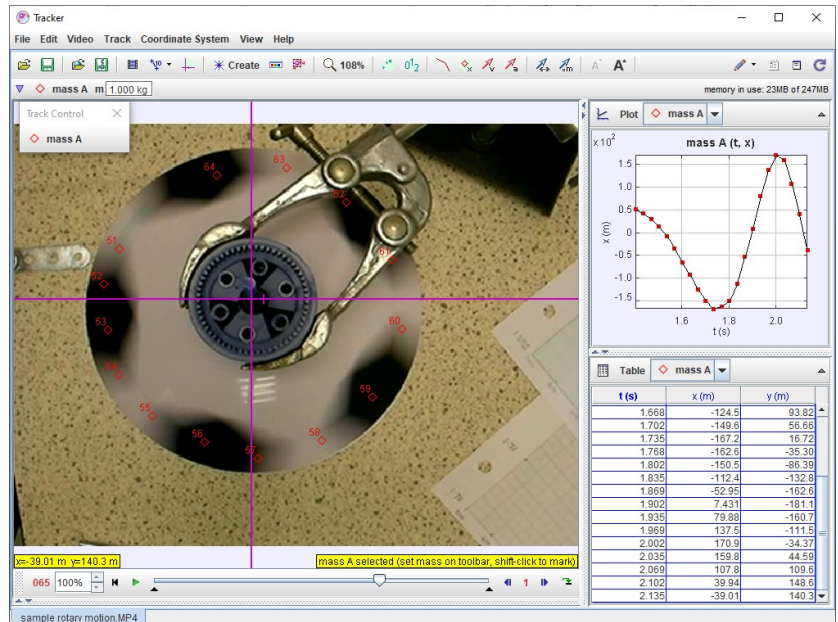
Click the axes icon and drag the origin to the centre of rotation.



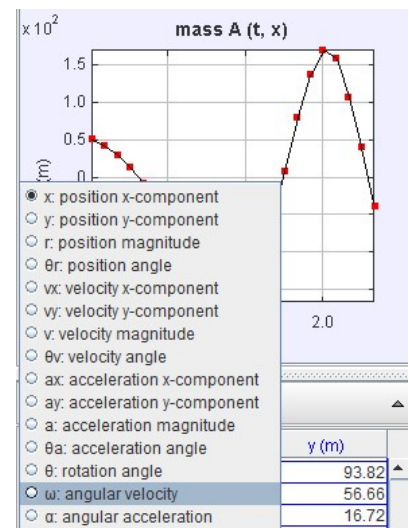
Click **Create** and choose **Point Mass**. Advance the video, using the slider and nudge buttons, until the puck begins to rotate.

Select a point on the puck (e.g. the green dot, one of the black 'peaks') and shift+click in successive frames.

Once you have tracked several points, your display should look something like this:



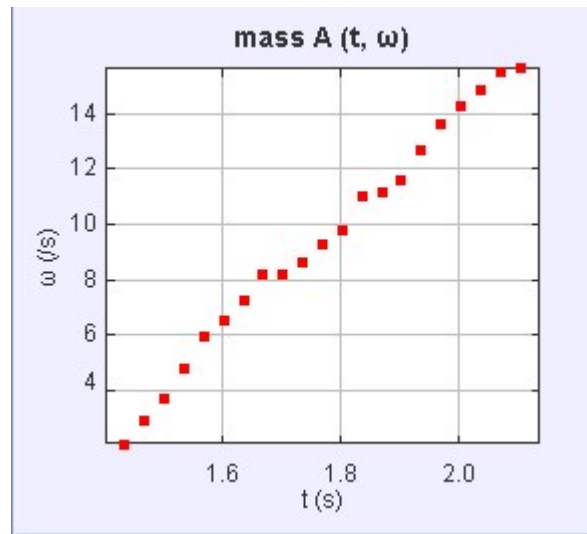
Experiment with graphing different angular quantities (remember – we click  $x$  (m) on the  $y$ -axis of the graph to change the quantity).



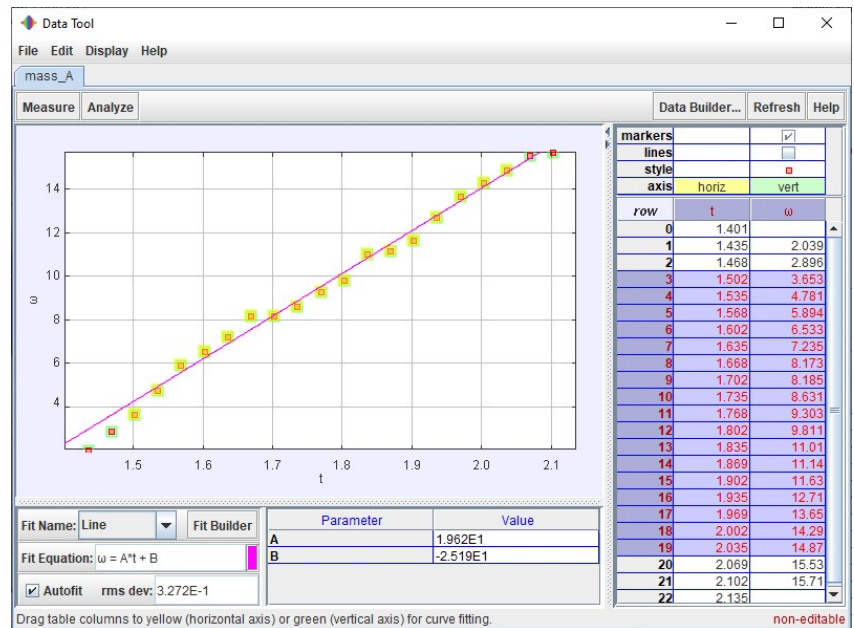
When plotting angular quantities, Tracker will default to degrees. If you go to the menu bar, select **Coordinate System** and choose **Units...** you can switch to radians.

Change the y-axis quantity to angular velocity ( $\omega$ ).

A constant torque should produce a constant angular acceleration. In other words, our graph of  $\omega$  versus  $t$  should be a straight line.



As we did before, we can double click on the graph to get a larger one, then drag over a group of points and analyze the gradient of the best-fit line. This should give us the angular acceleration.



## Why not plot angular acceleration versus time?

This gets a bit technical. Feel free to ignore it.

Why not just cut to the chase and plot angular acceleration ( $\alpha$ ) versus time? Tracker can do that. It's just not easy to make Tracker do it very well in this case. Here's why.

A video is made up of a series of still images. Depending on the video format, there may be, say, 15, 25, 30 or 60 images per second. You would think that if there were 30 images per second, each image would last  $1/30$  second. However, not every camera has a stable frame rate. There might be 30 images per second but some may last more than  $1/30$  second, some less.

Actually capturing the image might take the full  $1/30$  second, or, if the image is brightly lit, it may take less time. The longer it takes to capture the image, the more it will have moved in that time. That's why our moving objects in videos tend to be blurred rather than sharp.

When we track an object, we have to decide where we are going to click. The more blurred the object is, the harder it is to be consistent regarding where you click.

Here's a subtle point. Tracker can't *measure* velocity. It can only *calculate* velocity from frame rate and position.

Suppose we have an object captured by a camera that takes 30 frames per second. We track its position and get the following results:

Frame	Position (metres)
20	0.51
21	0.55
22	0.60

Tracker would calculate the velocity at frame 20 in the following way:

Velocity at frame 20 = position at frame 21 – position at frame 19

Time for 2 frames

Now, Tracker assumes that the time for 2 frames will be  $2 \times 1/30$  second but we've seen that it might not be.

We've done our best to click these positions accurately, but we've had to use our judgement. These factors mean we have a bit of uncertainty in our velocity. It's especially uncertain if the positions don't differ by much because we'll be left with a small number when we subtract one position measurement from another.

Tracker calculates accelerations in a similar manner to the way it finds velocities. The velocities it uses to calculate accelerations will, as we've seen, probably have significant uncertainties associated with them. This leads to even greater uncertainties in accelerations.

Using the gradient of an  $\omega/t$  graph is a way of averaging these individual accelerations. If the points on your  $\omega/t$  graph do not lie on a straight line, there's something else going on in your set up that needs investigating.

You can experiment with Tracker's **Bounce Detection** algorithm. Use this with care – it sometimes has unwanted effects. If you have a 'bumpy' graph, right click on the graph, select **Algorithms** and click the **Bounce detection** button.