Unbalanced Wheel (Drum)

Objective
- User observations to calculate the radius of gyration of a wheel
- Apply energy models to the dynamics of an unbalanced rotating wheel
- Learn computing skills

No cover sheet, abstract, table of contents, description of experiment or conclusions are expected or required. Treat this like an extended homework, give us derivation of equations, plots and questions as asked for and Matlab code.

Overview
Ideally, this lab would have a “point” mass attached to a wheel whose moment of inertia was known. Using your knowledge of energy methods, you could predict the motion of the assembly and compare it to measurements. Real life is not so simple. First, we don’t have a wheel whose moment of inertia is known. Second, without some extra non-rotating hardware, we can’t measure the rotation of the wheel. Therefore, before we can predict the motion of the unbalanced wheel, we must first do some work to calculate the moment of inertia (radius of gyration) of the balanced wheel.

Walt Lund has built an apparatus consisting of a drum with mass attachments, and a ramp. The drum has a support wheel on a support arm that also has attached a rotary encoder. Brad Dunkin has built a LabView VI that task as input the rotary encoder pulses and produces an angle and rate measurement of the wheel. Although the moment of inertia is not known, you are provided with drawings of the drum along with the material properties. The slope of the ramp is 4.30° and take g to be g=9.81 m/sec/sec.

Equations

The Balanced Wheel

The kinetic energy is the sum of the translational and rotational energies:

\[ KE = \frac{1}{2} (M + M_{o})V_{G}^2 + \frac{1}{2} I_{G} \omega^2 \]

\( V_{G} \) is the velocity of the center of mass of the wheel and \( I_{G} \) is the moment of inertia, defined as \( Mk^2 \). If we assume that the drum is rolling without slipping, we can make the usual assumptions about the relationship between \( V_{G} \) and the angular velocity \( \omega \). The
work done (equivalent to the change in potential energy) after the wheel has rotated through an angle \( \theta \) is

\[
work = (M + M_o)gR \theta \sin \beta
\]

One can then show that \( k \) is defined as

\[
k^2 = \frac{work - \frac{1}{2}(M + M_o)R^2 \omega^2}{\frac{1}{2}M\omega^2}
\]

### The Unbalanced Wheel

Calculating the motions of the drum with an extra mass, \( m \), is more difficult. For this derivation we will use conservation of energy rather than work-energy. The kinetic energy equation now includes another term:

\[
KE = \frac{1}{2}(M + M_o)V_G^2 + \frac{1}{2}I_c \omega^2 + \frac{1}{2}mv_b^2
\]

We have assumed that the attached mass is a particle and thus has no moment of inertia or rotational energy. The velocity of the attached mass, \( v_b \), is \( b \omega \) where \( b \) is the distance between the contact point and the mass. By using the law of cosines, you can derive \( b \) in terms of \( r \), \( R \), and \( \theta \). We then define the kinetic energy:

\[
KE = \frac{1}{2}(M + M_o)R^2 \omega^2 + \frac{1}{2}MK^2 \omega^2 + \frac{1}{2}m(R^2 + r^2 + 2rR \cos \theta)\omega^2
\]

We can define the potential energy by putting our x-y axis at the center of the wheel at \( \theta = 0 \). The potential energy at arbitrary \( \theta \) is:

\[
PE = -(M + M_o)gR \theta \sin \beta - mR \theta \sin \beta + mgr \cos (\theta + \beta)
\]

You can then easily solve for \( \omega \) as a function of \( \theta \) using these expressions for kinetic and potential energy.

### Data Collection

As a group, we will collect several trials of data for the drum, both with and without the extra masses. The files of data saved from the VI will be saved in the ASEN2003 shared directory on the ITLL computers. Make sure you find out what is contained in the file and the units. As in the previous labs, you’ll be importing these data into MATLAB using the `load` command. The files may include useless data, such as when the wheel left the ramp; so use Matlab to restrict your analysis to the first three revolutions of data.
Analysis

The Balanced Wheel

In homework and lecture, we showed that the angular acceleration $\alpha$ of a wheel rolling down a ramp is constant. This means that $\omega$ should increase linearly with time and $\theta$ should increase as a quadratic function of time. By plotting $\theta$ vs. time you should see this quadratic dependence.

Use the MATLAB command `polyfit` to fit a second order polynomial to the ($\theta$,t) data. Then use the MATLAB command `polyval` to define an array of smooth ($\theta$,t) values for each trial. The usage of `polyfit` and `polyval` is explained in the MATLAB help section.

Once we have corrected the $\theta$ data for the measurements, we can calculate the radius of gyration, $k$. As discussed in the background section, $k$ depends on both $\theta$ and $\omega$. Which $\theta$ and $\omega$ should you use? For this lab you should find a $k$ value for each ($\theta,\omega$) and then compute an average for each trial.

Turn In:
- Plot $\theta$ vs. time for all trials
- Plot $\omega$ vs. time for all trials
- For trial 1, plot $\theta$ and the polynomial fit
- Plot $\omega$ vs. $\theta$ for all trials and compare with the model. Include $k$ values on the plots.
- Report the average $k$ over all the trials. Use this average $k$ value in the next section

The Unbalanced Wheel

Unfortunately, there is not a simple polynomial that we can fit to the unbalanced data. You can still compare the model to the data.

Turn In:
- Plot $\omega$ vs. $\theta$ for all trials
- Plot $\omega$ vs. $\theta$ for the model and the average of the 4 trials
- How well does the theoretical model agree with the measurements, qualitatively and quantitatively?
- Describe at least one way the theoretical model could be improved.

Matlab Code

Your main program must call at least two functions – one to compute $k$ and the other to read in and plot the unbalanced wheel trials. The constant of the experiment must be defined in you main code. If these values are needed in the functions, then they need to
be sent to the functions as arguments. This way you can just change the constants in the main code if the values change on a later experiment. You are not limited to just two functions – you can use more if you like.

On your plots, make sure to distinguish measure values as points and modeled or fitted functions as lines. Also, if you use a black and white printer, make sure you use different styles of lines and symbols instead of color since this is lost.