

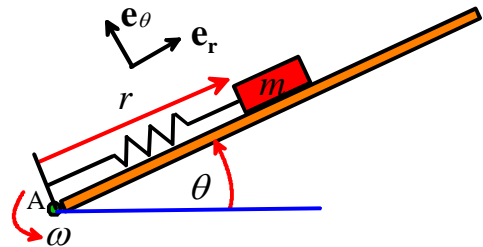


School of Engineering
Brown University

EN40: Dynamics and Vibrations

Homework 3: Solving equations of motion for particles Due Friday Feb 15th

1. The figure shows a simple idealization of a type of [MEMS gyroscope](#) (way over-simplified, actually...). It consists of a mass that slides up and down a guide, which spins at constant angular speed ω about A. The mass is attached to a spring with stiffness k and unstretched length L_0 . The goal of this problem is to find, and solve, the equation governing the radial distance $r(t)$ of the mass from the pivot. Friction and gravity should be neglected.



1.1 Write down the acceleration vector, in polar coordinates, in terms of time derivatives of r and ω

[2 POINTS]

1.2 Draw a free body diagram showing the forces acting on the mass.

[2 POINTS]

1.3 Use Newton's law $\mathbf{F}=\mathbf{ma}$ (and the spring force law) to show that $r(t)$ satisfies the differential equation

$$\frac{d^2 r}{dt^2} + \Omega^2 r = \frac{k}{m} L_0$$

where $\Omega^2 = \frac{k}{m} - \omega^2$ is a constant (introduced to simplify the solution)

[2 POINTS]

1.4 Assume that at time $t=0$, the mass is not moving radially (so $dr/dt=0$) and the length of the spring is equal to its unstretched length (so $r=L_0$). Use Mupad to solve the differential equation, following the procedure described in class. To stop Mupad giving complex exponentials in the solution, you can use

`assume(`Ω`>0)`

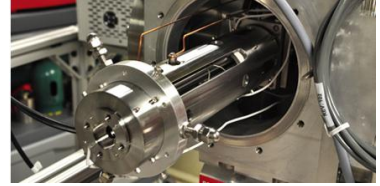
(Here, ``Ω`` is the Mupad for the Greek symbol Ω - if you use some other variable to denote Ω in your calculation you will have to change the ``Ω`` to be your variable). Why this should work is a complete mystery to me, since Ω is squared, but it does work.... You only need to submit the solution to the ODE (Copy it from Mupad), and do not need to hand in your mupad.

[2 POINTS]

1.5 Hence, write down a formula for the frequency of vibration of the mass. Does the frequency increase or decrease with the angular speed ω ? What happens if the bar spins very rapidly, so that $\omega^2 > k/m$?

[3 POINTS]

2. The picture shows a ‘quadrupole mass filter’ (from the [Rowland Institute](#)). It consists of four parallel rods, which are charged to induce a special time-dependent electric field in the space between them. Ions are fed into the space between the rods at one end. As this problem will show, only ions with a charge to mass ratio within a range are able to pass through the gap from one end of the rods to the other without hitting one of the rods.



2.1 The motion of the ion will be described by the components of its position vector (x,y,z) . Write down the velocity and acceleration of the ion in terms of these variables.

[1 POINT]

2.2 The ion has mass m , and is subjected to a force

$$\mathbf{F} = Q\mathbf{E}$$

where Q is its charge, and

$$\mathbf{E} = E_0 \frac{1 + \beta \cos(\omega t)}{d} (x\mathbf{i} + y\mathbf{j})$$

is a time dependent electric field vector. Here, E_0, β, d, ω are constants that describe the magnitude and geometry of the electric fields. Use Newton’s law to show that the components of the position vector of the ion satisfies the following equations of motion

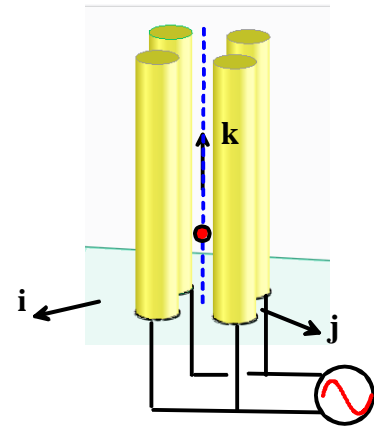
$$\frac{d^2x}{dt^2} = \Omega^2(1 + \beta \cos(\omega t))x \quad \frac{d^2y}{dt^2} = \Omega^2(1 + \beta \cos(\omega t))y \quad \frac{d^2z}{dt^2} = 0$$

where

$$\Omega = \sqrt{\frac{QE_0}{md}}$$

is a parameter (which has units of frequency).

[2 POINTS]



2.3 Show that the equations can be re-written in the following form (which MATLAB can solve)

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \Omega^2(1 + \beta \cos(\omega t))x \\ \Omega^2(1 + \beta \cos(\omega t))y \\ 0 \end{bmatrix}$$

[1 POINT]

2.4 Write a MATLAB script that will solve the equations of motion to determine x,y,z, v_x, v_y, v_z as a function of time (values for all the variables are given in the next section). You need not submit a solution to this problem. (There is no need to submit a solution to this problem)

[0 POINTS]

2.5 Analyze the motion of a charged particle that enters the rods at $z=0$ with initial position very slightly off the axis (take the initial position and velocity to be $(0.005, 0.005, 0)$ and $(0, 0, 0.01)$, respectively) with the following parameters: $\omega=10, \beta=20$. Run the computation for 100 time units. Try a solution with $\Omega=1.1$, and plot the trajectory of the ion (hand in this plot). Recall that to plot an x-y-z trajectory you can use

```
plot3(sols(:,1),sols(:,2),sols(:,3))
```

where `sols` is the solution variable computed eg using `[times,sols] = ode45(...etc)`.

If you like, you can download a script that will animate the motion of the ion as it moves through the filter (the script plots the trajectory as well, so you could use this to do your trajectory plot). To see the animation use the following syntax

```
[times,sols] = ode45(@eom,[0:1:100],w0);
animate_quadrupole(times,sols)
```

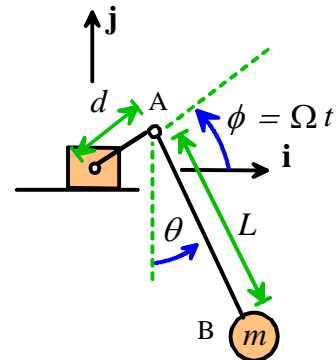
(this assumes that the function you are using to compute the time derivatives is called 'eom').

Show that with these settings, the ion will arrive at $z=1$ without a large radial deflection if $0.73 < \Omega < 1.55$ (you can do this by trial and error, or a better way would be to plot the value of $r = \sqrt{x^2 + y^2}$ at the end of the simulation as a function of Ω . Plotting this graph is quite challenging, however, so just plotting trajectories for Ω a bit below and above the range is OK too.). This is how the filter works – since Ω depends on the mass of the particle, only ions with a mass in some range can make it to the far end. **Hand in a plot of the trajectory for $\Omega=1.1$, and either two more trajectory plots showing the operation of the filter, or else the challenge graph)**

[6 POINTS]

3. The figure shows a pendulum (AB) mounted on a crank that rotates at constant angular speed Ω . The pendulum shaft AB pivots freely at A, so the angle θ varies with time as the pendulum swings. The goal of this problem is to derive the differential equation governing θ and solve it with Matlab.

3.1 Write down the position vector of the mass m in terms of d, L, Ω, θ . Hence, find an expression for its acceleration, in terms of d, L, Ω and θ and its time derivatives. Mupad can do the derivatives for you if you like -



[3 POINTS]

3.2 Draw a free body diagram showing the forces acting on the mass m , and hence write down $\mathbf{F}=\mathbf{ma}$ in terms of d, L, Ω, g, m and θ and its time derivatives, as well as the unknown tension force in the pendulum shaft.

[2 POINTS]

3.3 Eliminate the unknown tension force in the crank from the two equations in the preceding problem to show that θ is governed by the following differential equation

$$\frac{d^2\theta}{dt^2} - \frac{d}{L}\Omega^2(\cos\phi\cos\theta + \sin\phi\sin\theta) + \frac{g}{L}\sin\theta = 0$$

[3 POINTS]

3.4 Show that the equation of motion can be re-arranged into the following form

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \omega \end{bmatrix} = \begin{bmatrix} \omega \\ \frac{d}{L} \Omega^2 (\cos \phi \cos \theta + \sin \phi \sin \theta) - \frac{g}{L} \sin \theta \end{bmatrix}$$

[1 POINT]

3.5 Write a MATLAB script that will solve the equations of motion, and plot a graph of θ as a function of time. Values for all the constants are given in the next problem. You can download a script that will animate the motion of the forced pendulum to help visualize its behavior. To call the function, use the syntax

`animate_pendulum(times, sols, L, d, Omega)`

where `times, sols` are the solutions returned by the ODE solver, and `L, d, Omega` are the values of L, d, Ω . **(There is no need to submit a solution to this problem)**

3.6 Hence, plot graphs showing the variation of θ with time for $L=1m$, $d=0.1m$, and the f values of Ω listed below. Plot each solution for $0 < t < 100$, and use $\theta = d\theta/dt = 0$ at time $t=0$.

1. $\Omega = 0.25\sqrt{g/L}$
2. $\Omega = \sqrt{g/L}$
3. $\Omega = 1.85\sqrt{g/L}$. Try this solution with the default value of the tolerance for the ODE solver, then try setting the 'tolerance' parameter to `RelTol = 0.0001` and `RelTol = 0.00001` (the Matlab tutorial has info about how to do this). Notice that MATLAB gives an entirely different solution as the tolerance is made smaller – in fact the solution goes on changing no matter how accurate you try to make the computations. This is because the behavior with this value of Ω is *chaotic* – very small changes in parameter values lead to completely different solutions if you let the simulation run for a while.

Hand in 3 graphs as a solution to this problem.

[6 POINTS]

4 OPTIONAL – EXTRA CREDIT PROBLEM. In this problem we will set up a more accurate calculation of Felix Baumgartner's space jump. We made two crude approximations in our earlier calculation: (i) the first phase of the jump was approximated using a rough guess for the variation of acceleration with time; and (ii) the variation of air density with altitude was approximated. With MATLAB, we can correct both these approximations. We will make the following assumptions:

- Experiments show that the air temperature varies with altitude y (approximately) according to the equation

$$\theta(y) = \begin{cases} 298 - 120y/14000 & y < 14000 \\ 178 & 14000 < y < 20000 \\ 178 + 140(y - 20000)/30000 & y > 20000 \end{cases}$$

with y in meters, and θ in Kelvin.

- If air is modeled as an ideal gas, the variation of air density with altitude y satisfies the equation

$$\frac{d\rho}{dy} = -\frac{\rho g}{R\theta(y)}$$

Where g is the gravitational acceleration, R is the gas constant for air, and θ is the air temperature (in Kelvin)

- The air drag force is given by $F_D = \frac{1}{2}\rho C_D A v^2$

1.1 Draw a free body diagram for the jumper during free-fall, and hence use Newton's law to find an equation for his acceleration in terms of $m, g, \rho, C_D, \theta, A, v$. Do not assume that the drag force equals the weight in this problem (that was an approximation – here we can do the calculation properly)

[2 POINTS]

1.2 Show that the equations governing the jumper's altitude y , vertical speed v and the air density at the jumper's altitude can be expressed in the following form

$$\frac{d}{dt} \begin{bmatrix} y \\ v \\ \rho \end{bmatrix} = \begin{bmatrix} v \\ -g + \frac{1}{2m} \rho C_D A v^2 \\ -\frac{\rho g v}{R\theta(y)} \end{bmatrix}$$

[3 POINTS]

1.3 Hence, set up a MATLAB script that will calculate y, v, ρ as functions of time. Use the following parameter values $g=9.81 \text{ms}^{-2}$, $R=286.9 \text{J/kgK}$, $A=1.1 \text{m}^2$, $C_D = 0.8$, $m=80 \text{kg}$, and take $y = 39000 \text{m}$, $v = 0$, $\rho = 0.002 \text{kg m}^{-3}$ at time $t=0$. You will need to use a conditional statement to calculate the temperature. Add an 'event' function to your code to stop the calculation when $y=1524 \text{m}$ (this is when the parachute was deployed). **There is no need to submit a solution to this problem**

[0 POINTS]

1.4 Hence plot graphs showing (i) the variation of speed with time; (ii) the variation of altitude with time; and (iii) the variation of density with altitude. Hand in your graphs as a solution to this problem.

[6 POINTS]