



School of Engineering  
Brown University

## EN40: Dynamics and Vibrations

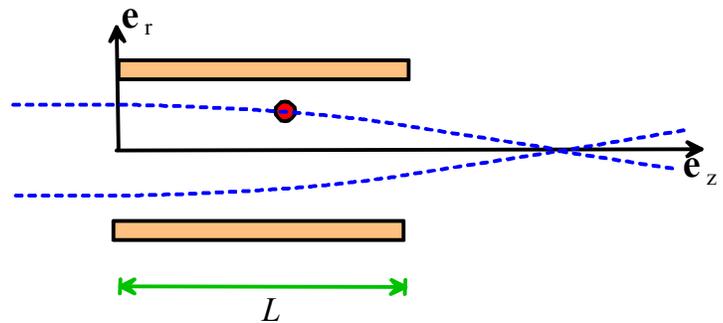
### Homework 3: Solving equations of motion for particles Due Friday Feb 20<sup>th</sup>

Your solution should include (1) a hand-written solution, submitted to 7<sup>th</sup> floor of B&H, with printed plots; and (2) a Matlab code uploaded to CANVAS. Please put all your matlab solutions in the same matlab file

1. The figure shows an electrostatic lens, whose purpose is to focus a parallel beam of charged particles to a point. An electrostatic field

$$\mathbf{E} = E_0 \left( -\frac{r}{2L} \mathbf{e}_r + \frac{z}{L} \mathbf{e}_z \right)$$

acts in the cylindrical region  $0 < z < L$ . A representative charged particle enters the lens at a position  $(r, 0)$  traveling with speed  $V$  parallel to the  $\mathbf{e}_z$  direction. It is subjected to a force  $q \mathbf{E}$ .



1.1 Show that the  $r, z$  coordinates of the particle satisfy the equations

$$\frac{d^2 r}{dt^2} = -\frac{qE_0}{2mL} r \quad \frac{d^2 z}{dt^2} = \frac{qE_0}{mL} z \quad z < L$$

$$\frac{d^2 r}{dt^2} = \frac{d^2 z}{dt^2} = 0 \quad z > L$$

1.2 Use Mupad to find solutions for  $z$  and  $r$  as functions of time, for the region  $z < L$ . Use as initial conditions  $\mathbf{v} = \mathbf{e}_z, \mathbf{r} = \mathbf{e}_r$  at time  $t=0$ . It is best to solve the two equations separately (they are independent) (You don't need to submit the Mupad solution to this problem)

1.3 Show that the equations can be re-written in MATLAB form as

$$\frac{d}{dt} \begin{bmatrix} r \\ z \\ v_r \\ v_z \end{bmatrix} = \begin{bmatrix} v_r \\ v_z \\ -cr/2 \\ cz \end{bmatrix} \quad c = \frac{qE_0}{mL} \quad z < L \quad c = 0 \quad z > L$$

1.4 Write a MATLAB script that will solve these equations of motion (use a conditional statement in your differential equation function to compute the value of  $c$ ).

1.5 Using MATLAB, Plot (on the same figure) the particle trajectories for  $\frac{qE_0}{mL} = 5, L=1$ , and initial conditions  $\mathbf{v} = \mathbf{e}_z, \mathbf{r} = 0.5\mathbf{e}_r, \mathbf{e}_r, 1.5\mathbf{e}_r$ . Make the  $z$  axis horizontal and the  $r$  axis vertical, as in the sketch, and use a time period of about 1.2 units so you can see that the trajectories cross. Find the focal length of the lens (measured from  $z=0$ ).

2. The goal of this problem is to compute the trajectory of a spacecraft during reentry. Assume that

(i) The air density at a distance  $h$  above the earth's surface is given by  $\rho = \rho_0 \exp(-h/d)$

(ii) The drag force acting on the vehicle is given by

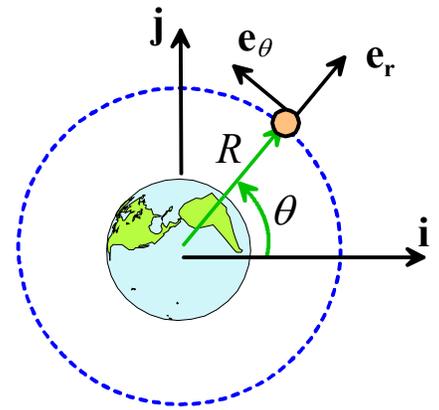
$$\mathbf{F}_D = -\frac{1}{2} \rho C_D A V \mathbf{v}$$

where  $V$  is its speed,  $\mathbf{v}$  is its velocity vector,  $A$  is the frontal area of the vehicle and  $C_D$  is the drag coefficient.

(iii) The gravitational force acting on the vehicle is given by

$$\mathbf{F}_G = -mg \frac{R_e^2}{R^3} \mathbf{r}$$

where  $\mathbf{r}$  is the position vector of the vehicle relative to the earth's center,  $R$  is the magnitude of the position vector and  $R_e$  is the radius of the earth.



We can assume that the vehicle moves in the  $(x,y)$  plane. We will solve this problem using polar coordinates – so the position of the vehicle is specified by its distance  $R$  from the origin and the angle  $\theta$ .

2.1 Write down the velocity and acceleration vectors of the vehicle in polar coordinates, in terms of time derivatives of  $R$  and  $\theta$

2.2 Using Newton's laws, show that the differential equations for  $R$  and  $\theta$  and their time derivatives

$v_r = \frac{dR}{dt}$       $\omega = \frac{d\theta}{dt}$  can be expressed as

$$\frac{d}{dt} \begin{bmatrix} R \\ \theta \\ v_r \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \\ \omega \\ -B + R\omega^2 - Cv_r \\ -\frac{2v_r\omega}{R} - C\omega \end{bmatrix} \quad B = g \frac{R_e^2}{R^2} \quad C = \frac{1}{2} \frac{\rho_0 C_D A V}{m} \exp\left(-\frac{(R - R_e)}{d}\right)$$

$$V = \sqrt{v_r^2 + (R\omega)^2}$$

2.3 Write a MATLAB script that will solve the equations listed in 2.2, using the following parameters (representative of the Orion crew module; see also [Tiggles et al](#))

- Capsule mass  $m=8600$  kg
- Diameter of heat shield (specifies the area  $A$ )  $5m$
- Drag coefficient  $C_D = 1$
- Earth radius  $R_e = 6371$  km
- Density decay length  $d=5$  km
- Air density at sea level  $1.2$  kg/m<sup>3</sup>
- Re-entry altitude  $h_0 = 122$  km
- Velocity at re-entry  $V_0 = 32000$  km/hr

Start the calculation with initial conditions  $R = R_e + h_0$ ,  $\theta = 0$ ,  $v_r = -V_0 \sin \phi$     $\omega = V_0 \cos \phi / R$  where  $\phi$  is the re-entry angle (a parameter that will be changed), and run the calculation for 240 minutes.

2.4 Add an ‘event’ function to your code that will stop the calculation when the vehicle reaches the ground.

2.5 Add lines to your code that will use the solution computed by ode45 to calculate the  $x, y$  coordinates of the position vector. You will need to loop over each point in the solution vector, extract the values of  $(R, \theta, v_r, \omega)$  from the solution, and then use the polar coordinate formulas to calculate the  $(x, y)$  coordinates. As a check on your calculation, plot the trajectory  $(x-v-y)$  for  $C_D = 0$  and re-entry angle  $\phi = 0$ . The trajectory should be a weakly elliptical orbit around the earth. It is best to get this working before proceeding any further....

2.6 Add lines to your code that will use the solution computed by ode45 to calculate the magnitude of the acceleration of the vehicle, for each value of time. You will need to use the polar-coordinate formulas to calculate the velocity and acceleration components in the  $\{\mathbf{e}_r, \mathbf{e}_\theta\}$  basis, and then compute the magnitude of this vector. You can use your ‘differential equation’ function to calculate the values of  $dv_r/dt$  and  $d\omega/dt$ .

2.7 Plot graphs that show the variation of (i) the altitude of the vehicle  $(R - R_e)$ ; and (ii) the magnitude of the acceleration, as a function of time (with  $C_D = 1$ ), for  $\phi = 3^\circ$  (the spacecraft skips off the atmosphere and never lands)  $\phi = 4.12^\circ$  (this gives a skipped reentry)  $\phi = 5^\circ$  (a normal re-entry)  $\phi = 9^\circ$  (this is too steep and gives a very large deceleration)

**3.** The goal of this problem is to investigate the performance of a feedback controller that is intended to hold a quadcopter at a prescribed altitude  $h$ . We will assume that the vehicle moves only vertically, to keep things simple. At some instant of time, the aircraft has height  $y$  and vertical speed  $v_y$ .

The magnitude of the lift force is given (very roughly) by  $F_L = 4P(t) / \left( (v_y/2) + \sqrt{mg / (\pi\rho L^2)} \right)$  where  $L$  is the length of the rotor blades,  $\rho = 1.2 \text{ kg/m}^3$  is air density and  $P(t)$  is the power developed by one of the four motors.

The magnitude of the drag force  $F_D = cV^2$  where  $V$  is the speed of the vehicle and  $c$  is a constant. Drag acts opposite to the direction of motion.

The feedback controller adjusts the engine power according to the difference between the actual height of the aircraft and the desired altitude. (In the general terminology of control theory the engine power is the ‘input’ to the system; the altitude  $y$  is the ‘output’; and  $e = h - y$  is the ‘error’). Specifically,

$$P(t) = K_p e(t) + K_D \frac{de}{dt} + K_I \int e(t) dt$$

where  $K_p, K_D, K_I$  are three constants that can be tuned to give the best performance of the system. This type of controller is called a ‘PID’ feedback loop, which stands for ‘Proportional, Integral, Derivative.’

3.1 Use  $F=ma$  to show that the equations for  $y$  and  $v_y$  can be written in the form

$$\frac{d}{dt} \begin{bmatrix} y \\ v_y \\ Q \end{bmatrix} = \begin{bmatrix} v_y \\ 8P / (v_y m + C) - cv_y^2 / m - g \\ h - y \end{bmatrix} \quad \begin{aligned} P &= K_p(h - y) + K_D \left( \frac{dh}{dt} - \frac{dy}{dt} \right) + K_I Q \\ C &= 2m\sqrt{mg / (\pi\rho L^2)} \end{aligned}$$

Here,  $Q$  is  $\int e(t)dt$ .

3.2 Write a MATLAB script that will calculate  $y$ ,  $v_y$  and  $Q$  as a function of time. Use parameters representative of the crazyflie –

- $m=10$  grams
- $L=2$ cm
- $c=0.0012\text{Ns}^2/\text{m}^2$
- $\rho = 1.2\text{kg}/\text{m}^3$

and initial conditions  $y = v_y = Q = 0$

3.3 Test the performance of the control system by calculating and plotting the height  $y(t)$  as a function of time as the aircraft rises to a preset, fixed altitude of  $h=1\text{m}$ , with the following values for the control parameters (a)  $K_p=1, K_D=K_I=0$ ; (b)  $K_p=1, K_D=0.2, K_I=0$  (c)  $K_p=1, K_D=0.2, K_I=0.2$ . Run the simulation for a 20 second time-interval.