16.07 Lab I

Issued: October 16, 2009
Due: Monday, October 26, 2009

Introduction

This laboratory is the first in a series that deal with celestial and spacecraft dynamics. In this lab, you will create a MATLAB code that will allow you to numerically confirm Kepler's laws as well as simulate the motion of the earth and the other planets in the solar system as well as spacecraft in orbit about or on the way to one of them. Since this lab will focus on the earth and moon system, the presence of the sun will be ignored. In order to accomplish these tasks, your MATLAB code will integrate the equations of motion in time, starting from an initial condition. In later labs, you will add features like spacecraft thrusting, which may allow you to simulate orbital transfer, rendezvous, descent and ascent from planetary surfaces and other more complicated situations.

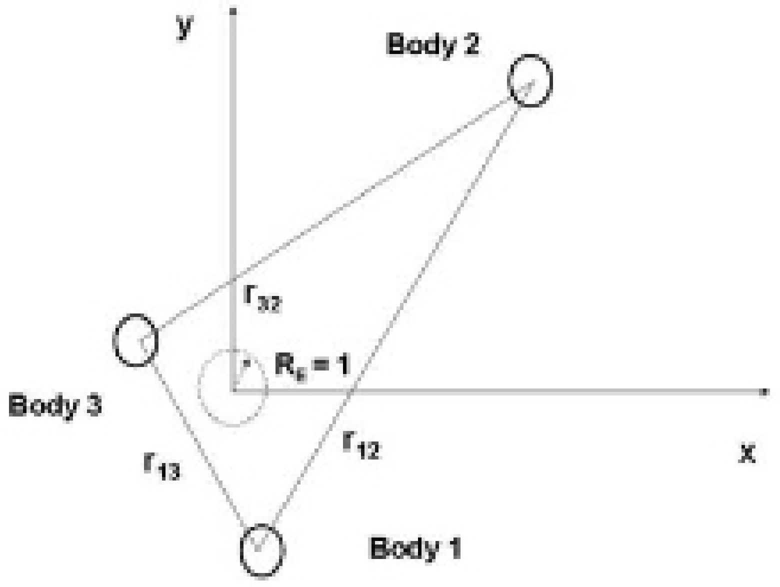

Figure 1: 3 Body coordinate system.

Figure 1 shows the coordinate system for a generic problem involving the three planetary/spacecraft bodies. The bodies are labeled as Body 1, Body 2 and Body 3, but these names are just an example. These bodies will turn into the Earth, the Moon and a spacecraft. The important thing is to choose a convention that you understand and can be consistent with.

The first step is to formulate the equations we would like to solve with the MATLAB code. In the general case, we desire to calculate the positions and velocities of each body as a result of its interaction with the other members of the system. For our purposes, we only consider the gravitational forces resulting from the mutual attraction of body pairs. Thus, the motion of any of the three moving bodies must consider the gravitational influence of the other two moving bodies. For instance, the attractive force that Body 2 exerts on Body 1 is given by,

$$\boldsymbol{F}_{12} = \frac{\mu_2 m_1}{r_{12}^3}\boldsymbol{r}_{12}. \tag{1}$$

Here, $\mu_2 = Gm_2$, the gravitational parameter of Body 2, where G is the universal gravitational constant[1] and m_2 is the mass of Body 2. The mass of Body 1 is denoted by m_1, and r_{12} is the magnitude of the distance vector that points from Body 1 to Body 2, $\boldsymbol{r}_{12} = \boldsymbol{r}_2 - \boldsymbol{r}_1$.

Celestial bodies move in three dimensions, so in general you would write your equations for the x, y,and z positions of each body. However, for this lab, we will take the position of the earth, moon and spacecraft system and their motions as confined to the x, y plane. Therefore, the z equations are not required.

For numerical purposes it is convenient to write the equation of motion $\ddot{\boldsymbol{r}} = \boldsymbol{F}/m$ for each body as a system of first order equations. That is, we write,

$$\dot{\boldsymbol{r}} = \boldsymbol{v}, \tag{2a}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{F}/m. \tag{2b}$$

Therefore, to describe the motion of a body in two dimensions, four state variables are required – two for position and two for velocity. For three moving bodies confined to planar motion in the x, y plane,we need only four for each body for a total of twelve. We will use a Cartesian coordinate system which means that the equations should be separated into Cartesian x,and y components. The state vector $\mathbf{X}$ contains all variables that completely describe the state of the system. For purposes of uniformity, you should order the states as follows,

$$\mathbf{X} = [x_1\ y_1\ x_2\ y_2\ x_3\ y_3\ u_1\ v_1\ u_2\ v_2\ u_3\ v_3]^T, \tag{3}$$

where x,and y are position components, u is the x-component of velocity, and v is the y-component of velocity.

You will be solving several different problems in this lab and building on your code in future labs. Thus, it will be beneficial to code in a flexible way. For example, you might want to define variables for the masses of the objects, instead of hard-coding the values. Some problems in this or future labs may only deal with two bodies such a planet in orbit around the sun, or in others, the gravitational pull of one of the bodies may be neglected[2]. *However, your code must have the capability to handle up to three bodies* . An effective way to remove the effect of a body on the other bodies in your code is to set its gravitational parameter to zero.

For later work, we present physical parameters for 5 celestial bodies: the sun, the earth, Mars, Jupiter and the moon. Because of the extremely large mass of the sun, it is usual to consider it to be stationary in our coordinate system. The planets, including the earth, orbit around the sun; and the moon orbits around the earth. When considering planetary orbits around the sun or the moon's orbit around the earth, the effect of other celestial bodies are ignored. These simplifications are referred to as the two-body and the three-body problems. The two body problem has an analytic solution in term of the Kepler planetary orbit problem. The three-body solution has no analytic solution and as we shall later see can exhibit quite wild behavior.

---

[1] $G = 6.673 \times 10^{-11}$ m^3/kg-s^2

[2] For example, the pull of the space shuttle on the Earth could certainly be neglected.

None of the planetary orbits are circular but are slightly elliptical. In addition, the moon's orbit is slightly out of the plane formed by the earth's orbit around the sun. However, when expressing motion of bodies in orbit around the sun or in the case of the moon in orbit about the earth, we assume these orbits to be **circular** and **co-planer**. Also, the planets are not spheres so that their radius is not constant; we will use the equatorial radius. Because, of these approximations, if you look up these physical parameters from various sources, you may notice slight discrepancies in the 3rd or 4th digit.

These parameters are not independent since the orbital velocity depends on the orbital radius. The orbital velocity is presented only for interest; you should obtain it from Kepler's laws and use it as your boundary condition.

Table 1: Physical and Orbital Data

	mass (kg)	semimajor axis (km)	radius (km)	orbital velocity (m/s)
Sun	1.989×10^{30}	–	—	—
Earth	5.975×10^{24}	149×10^{6}	6378	29,790
Moon	7.36×10^{22}	384×10^{3}	1737	1023
Mars	6.441×10^{23}	228×10^{6}	3397	24,140
Jupiter	1.899×10^{27}	778×10^{6}	71,492	13,060

When working with problems involving the sun and the planets, it is usual to non-dimensionalize distances with respect to the distance between the sun and the earth: 149×10^6 km, defined as 1 AU (astronomical unit.), and time by 24*60*60 sec.=1day. (In some studies, the time scale chosen is 365.25*24*60*60 sec.=1 year.)

When working with problems involving the earth and the moon and a spacecraft traveling between them or in orbit about them, normalize the distances by the radius of the earth E_R=6378 km, and time by 24*60*60 sec = 1 day.

Problems

In the following problems, the items you must turn in – either the written answers to questions, or required plots – are highlighted in **bold**. The ordering of these deliverables in your report should match the ordering that appears below.

1. The Equations of Motion

Making use of the state vector (Equation 3) and the system of first order equations (Equation 2), we can write the equations of motion for our system in the form,

$$\dot{\mathbf{X}} = \mathcal{F}(\mathbf{X}), \tag{4}$$

where the right-hand side is a non-linear function of the state. It turns out that when expressed in the SI system, the units of the distances and masses in our system have very large exponents if we're considering planetary systems. Because of finite machine precision, trying to do calculations with either very large or very small numbers can result in accumulated round-off errors. Thus, when