Abstraction of Mechanical Systems

Hopper Example

November 11, 2014

This is an in-class exercise meant to help you work through some of the issues that arise in going from a system in the physical world to a set of differential equations that are implemented in MATLAB. In each section we’ll ask you to spend some time talking to your neighbor about a strategy, and then we’ll ask you to share your thoughts with the rest of the class.

A simple hopper design

You have all designed hoppers in Design Nature; the designs you’ve pursued cover an enormous amount of space.

One simple version of a hopper is the linear reaction mass: a relatively light weight rod, with a heavier “reaction mass” which is attached via a spring or latex tube to the rod. The hopper is loaded by stretching the tubing; when the hopper is triggered, the reaction mass accelerates up, and the hopper leaps (hopefully to incredible heights).

Thinking about it

On your own, sketch out the key frames for the hopper during a launch. Note that there is space provided for about 9 frames here – use the space so that you can capture details! Assume the first frame shown is at the instant that the trigger has released. Then discuss your prediction with your neighbor. Do you agree or disagree? How many phases of motion are there here?
System boundaries, subsystem boundaries, and nature of the parts

Now that we’ve though through the behavior of the hopper, we need to think about where the system boundaries are, how many parts there are in the system, and how those parts act (i.e., are they particles? Rigid bodies? Deformable bodies?). Decide what you think are reasonable answers to these questions. Then discuss your answers with your neighbor. Do you agree or disagree? What are the different options here? Why would you choose one option over another?
Interactions, qualitative

What forces and torques might be present between each part of the system and the rest of the universe? What forces and torques might present between the each part of the system and the other part(s) of the system? Think about the different phases of motion: do the interactions change in different phases? Which of these forces and/or torques do you think it’s important to take into account? Why? In collaboration with your neighbor, identify, qualitatively, the different interactions that you think will be important in the system (assume a number of parts / type of parts consistent with what you think is appropriate).
Free body diagrams

For each phase of motion, and for each part of the system you have identified, draw the relevant free body diagram. Then discuss with your neighbor. Be sure to include all the interactions that you identified as important!
Coordinate systems, parameters, and interactions

Here is a picture of the system. On the picture draw an appropriate coordinate system that can be used to describe the motion of the system. Label relevant positions, velocities, lengths, parameters, etc.,

Use your coordinate system and labels to develop mathematical expressions for the forces. We recommend that you abstract the system to two point masses connected by a nonlinear spring. Work with your neighbor and discuss.
**Differential equations**

For a dynamical system, the **state variables** are a set of variables which fully describe the state of the system. State variables will feature in the third project in the same role that stocks played in the first two projects:

- For each state variable there will be one differential equation.
- We must specify the initial value of each state variable in order for the system of differential equations to have one and only one solution.

For example, for a single particle moving freely in three dimensions, there are six state variables: the position in three dimensions, and the velocity (or momentum) in three dimensions. The resulting differential equations are:

\[
\frac{dx}{dt} = v_x, \quad \frac{dy}{dt} = v_y, \quad \frac{dz}{dt} = v_z, \quad \frac{dv_x}{dt} = F_x/m, \quad \frac{dv_y}{dt} = F_y/m, \quad \frac{dv_z}{dt} = F_z/m,
\]

where \((x(t), y(t), z(t))\) give the position of the particle, \((v_x(t), v_y(t), v_z(t))\) give the time-dependent velocity of the particle, \((F_x, F_y, F_z)\) tell you the force at any given instant (and position and velocity) and \(m\) is the mass of the particle. In addition to the force and the mass we need to know both the initial position and the initial velocity in order to determine the future state of the system.

If you are familiar with vector notation, you can write the equations above more compactly as:

\[
\frac{d\vec{r}}{dt} = \vec{v}(t), \quad \frac{d\vec{v}}{dt} = \vec{F}/m
\]

where \(\vec{r} = xi + yj + zk\), \(\vec{v} = v_xi + v_yj + v_zk\), and so forth. But it’s still six differential equations.

**How many state variables** are there for the different abstractions we discussed back on page 2?
For the abstraction we’re working with (page 5), what are the differential equations needed to describe the time evolution of the system? Discuss with your neighbor, and propose some DE’s for each phase of motion. Also identify the condition for changing from one phase to another!

The wonderful world of MATLAB

Time permitting, we will ask you to translate your differential equations into a function appropriate for use with ode45. More likely, we’ll demo this for you. Code is available on the website.