

Introduction to Computers and Engineering Problem Solving Spring 2005

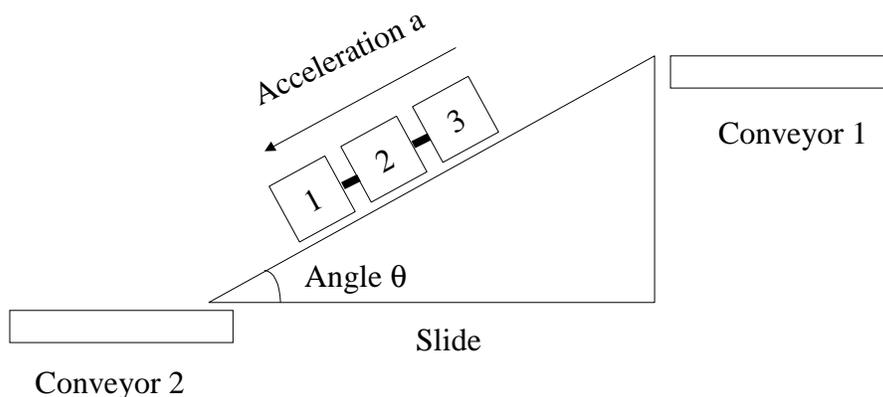
Problem Set 1: Moving stereo systems in a plant

Due: 12 noon, Session 5

1. Problem statement

At a manufacturing facility, stereo systems are produced, put in boxes and placed on a conveyor belt. An individual stereo system is put in three boxes. Box 1 holds the tuner/amp and weighs, for example, (has a mass of) 10 kg. The second box holds the woofer and two other speakers, and weighs, for example, 5 kg. The third box holds the remaining small speakers and miscellaneous components, and weighs, for example, 2.5 kg. The three boxes are taped together: box 1 to box 2, and box 2 to box 3.

At the end of one conveyor belt, the boxes go down a slide to the next conveyor belt, which takes the boxes to a shrink wrap machine on a lower floor of the factory building that will package them for shipment.



The elevation difference between the two conveyor belts is 4 m. The coefficient of sliding friction on the slide varies with the mass of the box:

$m > 8$ kg: 0.25

$4 \leq m \leq 8$ kg: 0.30

$m < 4$ kg : 0.35

2. Problem Objective

Your program will find the length and angle of the slide so that the velocity of the boxes when they reach the bottom of the slide is less than the maximum speed (say 2 m/s) to avoid damage to the speaker components when they reach the bottom of the slide and decelerate rapidly. Ignore the length of the boxes when computing the length of the slide. Report the angle to the nearest 0.01 radian. Convert the result to degrees as well, to make it easier for you to check your answer.

You will also need to find the tension on the tape between each pair of boxes (to make sure the tape is strong enough), and the acceleration of the boxes (all three will have the same acceleration).

The user will input the mass of the three boxes, the elevation difference between the conveyor belts, and the maximum speed. Check that the mass of the boxes decreases; if not, put them in decreasing (non-increasing) order and write an error message that you did so. You do not need to check the validity of any of the other inputs.

The three coefficients of friction should be constants in your program, and the mass ranges to which they apply are also constants. Gravity = 9.8 m/sec^2 is also a constant.

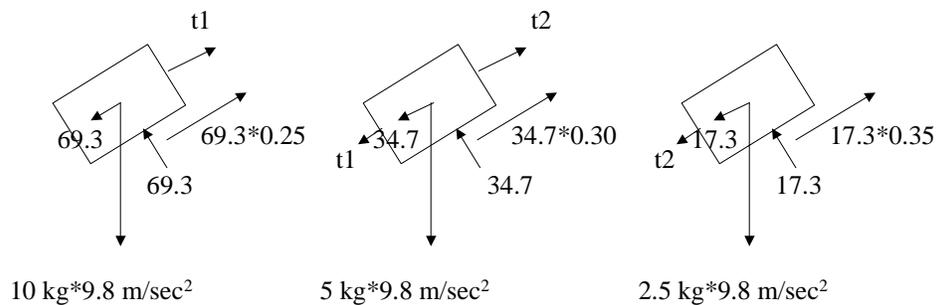
You will need to search through the possible angles to find the one that meets the maximum velocity criterion. You will need a loop construct in your program, and also a branch construct when you've found the solution. When you've found and output the solution, your program terminates.

Write your program as a `main()` method in a `SlideLength` class. If you know about methods, you may use them but they are not required. Comment your code. Points will be taken off for insufficient comments.

3. Approach

Model the three taped-together boxes as three masses connected by strings. The heaviest box will always go first, so the forces on the tape between the boxes will be tension, not compression, and thus modeling the connections as strings is acceptable.

Example: The free body diagram of the three boxes when the slide angle is 45 degrees is:



Free body diagrams at $\theta = 45^\circ$

For box 1: Gravity acts at 9.8 m/sec^2 . The component of gravity along the 45° angle is $mg \sin \theta$, or 69.3 kg-m/sec^2 . Positive acceleration on the box is down and to the left. The friction force resists the acceleration; it is 0.25 times 69.3 or 17.3 kg-m/sec^2 . Tension t_1 from box 2 also resists the acceleration. Force balance is written as:

$$10a = 69.3 - 69.3 \cdot 0.25 - t_1$$

For boxes 2 and 3 the corresponding equations are:

$$5a = 34.7 - 34.7 \cdot 0.30 + t_1 - t_2$$

$$2.5a = 17.3 - 17.3 \cdot 0.35 + t_2$$

You have three equations in three unknowns; solve for a , t_1 and t_2 . In this example, $a = 5.0$, $t_1 = 1.965$ and $t_2 = 1.255$.

After you've solved for a , t_1 and t_2 :

You know the length of the slide $s = h/\sin\theta$, where h is the elevation of the slide. $s = 5.7$ in the example.

Recall that $v = at$ and $s = 0.5at^2$. Once you know a , you can find the time t to travel a distance s and then you can find the velocity v at that time t . You then compare the velocity with the maximum allowable velocity. In the example, with $s = 5.7\text{m}$ and $a = 5.0\text{ m/sec}^2$, $t = 1.51\text{ sec}$ and $v = 7.55\text{ m/sec}$

General case: The equations for boxes 1, 2 and 3 with a slide angle θ are:

$$m_1 a + t_1 = m_1 g (\sin\theta - \mu_1 \cos\theta)$$

$$m_2 a - t_1 + t_2 = m_2 g (\sin\theta - \mu_2 \cos\theta)$$

$$m_3 a - t_2 = m_3 g (\sin\theta - \mu_3 \cos\theta)$$

where μ_i = coefficient of friction for mass i

For a given angle θ you must compute the coefficients of the left hand side of these three equations. Write expressions (derive them by hand) for a , t_1 and t_2 ; your program will evaluate these expressions, given the left hand side coefficients, for a , t_1 and t_2 . It's easiest to solve the first equation for t_1 as a function of a , the third equation for t_2 as a function of a , and then substitute t_1 and t_2 into the middle equation and solve for a . Last, find t_1 and t_2 trivially from the first and third equations.

Your program will then find total travel distance s ($s = h/\sin\theta$), time t at distance s ($t = \sqrt{2a/s}$), and then the velocity v at time t ($v = at$). Report a , t_1 , t_2 , time t , v and s at the maximum angle θ to meet the velocity constraint. Note that Java's `sin` and `cos` functions take angles in radians, not degrees, as their input.

For small values of θ , the friction force will be larger than the gravitational force, and will result in a negative (upward) acceleration, which is not possible. When this occurs, ignore the result, increment θ and keep searching for a valid solution.

If you use the most straightforward approach, your program will terminate when the velocity is slightly higher than the maximum velocity, but the angle will be within 0.01 radians of the desired angle. This is a satisfactory solution.

In general, we derive as much of the solution for a computational procedure analytically (by hand) as possible, and then use numerical methods for the test.

Turn In

1. Place a comment with your full name, MIT server username, section, TA name and assignment number at the beginning of all `.java` files in your solution. In this homework, you will have just one `.java` file.
2. Place all of the files in your solution in a single zip file. In this homework, you will have just a single file to put in your zip file.
 - a. Do not turn in electronic or printed copies of compiled byte code (`.class` files) or backup source code (`.java~` files)
 - b. Do not turn in printed copies of your solution.
3. Submit this single zip file.
4. Your solution is due at noon. Your uploaded files should have a timestamp of no later than noon on the due date.

Penalties

- 30% off if you turn in your problem set after Friday noon but before noon on the following Monday. You have one no-penalty late submission per term for a turn-in after Friday noon and before Monday noon.
- No credit if you turn in your problem set after noon on the following Monday.