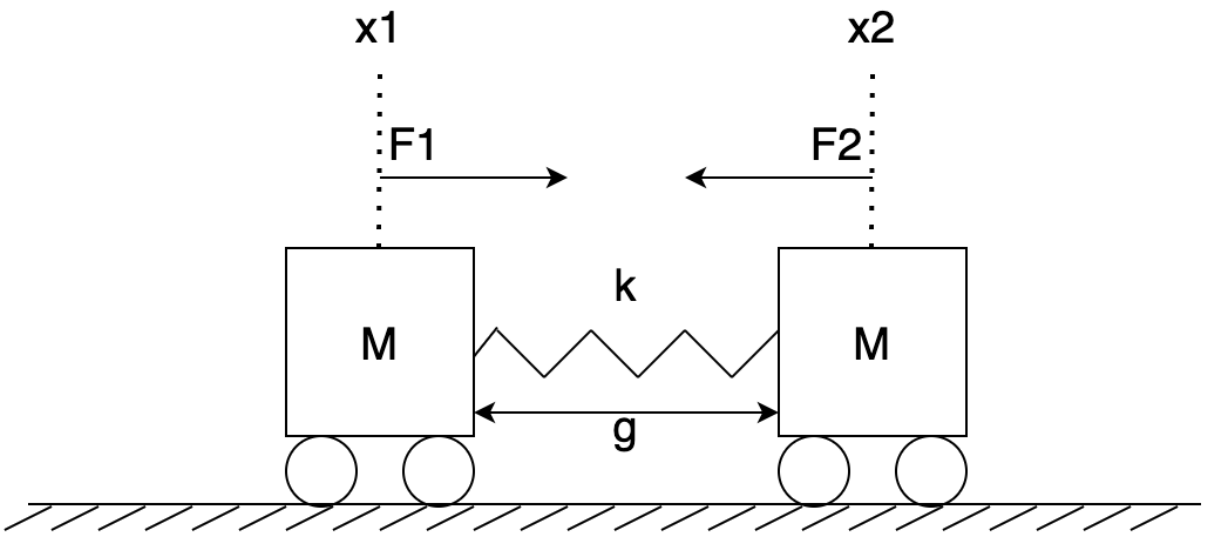
Controls Test

Q1) We have two blocks of mass “M” on a frictionless surface connected by a spring. They are acted upon by two forces F1 and F2.



1) What is the equation of motion for the system? The spring is at net 0N force when the gap g is at 1m.
    a. List the poles and zeros of the system

2) Simulate the motion of the two blocks for the following parameters
    a. M = 15 kg
    b. k = 100 N/m
    c. F1 = 100, F2 = 200N
    d. What do you notice about the damping behavior of the system? (undamped, overdamped, etc)

3) Design a controller which has the following sensors/actuators/reference
    a. Sensors: Assume you have sensors which measure the position of both masses.
    b. Reference: Target of the controller is to preload the spring to a target compression of g_target = 0.1m. In other words, the gap between the masses should be 0.1m.
    c. Actuators: F1 and F2 as control actuation
    d. Show a plot of the position, velocity of x1, x2
    e. Use initial condition of 1 meter separation between masses and no velocity for either mass

4) Update the controller from 3) to additionally target the following

a. Impart a net velocity on the system v_target = 0.3m/s. v_target is the average speed of both masses.
b. Should be robust to disturbances to the two masses and sensor noise
c. Show a plot of the position, velocity of x1, x2.
d. Show response to added gaussian sensor noise on the position and velocity measurement (mu = 0m, sigma = 0.25m)
e. Attempt to have the system settle at its target within 10s
f. How does your controller change the damping behavior of the plant?

5) Now, for the controller at 4)
    a. Once it has hit steady state, model what happens to the system if the spring snaps? i.e k -> 0
    b. Name a couple techniques which can be used to estimate if the spring has snapped.
    c. Can you recommend updates to the controller such that it will prevent the masses from colliding even if the spring snaps?
        i. Are these updates realistic. Will they always work? If not, what are the corner cases?

State any assumptions made for any of the above steps.

Q2) Low pass filter implementation in C

```
////////////////////////////////////////////////////////////////////////////
//    Low Pass Filter (10 points)
//    Implement a function that will be called at 10hz (every 100 ms) and returns
//    an exponentially weighted average. The latest sample is given 1/10
weighting
//    and previous filtered value a weighting of 9/10. The function should
//    initialize the filter to the first sample value received if it is the first
//    time the function has run.

float lowPassSamples_10hz(float sample)
{
    // Answer: TODO
    return 0.0f;
}
```