 ***MSIM 603: Simulation Design***

***Department of Modeling, Simulation and Visualization Engineering***

***Old Dominion University***

**Programming Assignment Two**

February 28, 2017

Mark Brosche

Major: MSVE - Master of Engineering

Online Distance Graduate - Outside Virginia (Saratoga Springs, NY)

[mbros006@odu.edu](mailto:mbros006@odu.edu)

407-325-2818

***Problem:*** You are to simulate a magnetic levitation system. The simulation should be based on the system discussed in class.

***Software Approach:*** Your software should use a procedural programming approach. You are to utilize the Euler method for integral evaluation, creating reusable Euler code.

***Deliverables:***

* Simulation design. This deliverable should include a description of the organization of your software (consider how you partition the program into .h files), a behavioral description of your major methods, and an algorithm description (either pseudocode, mathematical equations, or flowcharts).
* Simulation implementation (Visual Studio project directory).
* A discussion (based on plots of current and position relative to time) on the impact of the PID parameters.

# INTRODUCTION

The goal of a magnetic levitation system is to maintain a controlled distance from the object being levitated and the boundaries in which it is confined. The problem inherent with a moving system is dealing with displacements from outside forces, whether increased mass (train passengers), weather, or structural flexing/imperfections. To simulate the control of such a system, the use of feedback loops is required to maintain the desired levitation distance. A PID controller is used in this assignment.

# SIMULATION DESIGN

Similar to the Car Following example shown in class, the PID controller implementation uses a procedural programming approach that updates variables in a PID structure for a given time step.

***PID Controller Model***

***Controller***

Operations:

InitController

ComputeNS

UpdateCS

PrintState

Hidden

Operations:

k

Data:

accelerationCS

accelerationNS

velocityCS

velocityNS

positionCS

positionNS

currentCS

currentNS

errorCS

errorNS

integralErrorCS

integralErrorNS

derivativeErrorCS

derivativeErrorNS

P

I

D

m

g

x\_0

i\_0

Controller.cpp also uses the Euler.cpp to carry out calculations for integralErrorNS, velocityNS, and positions.

,

# DISCUSSION

The output of the PID controller can be found in the Excel files:

\\P2\_Brosche\_Mark\P2\_Brosche\_Mark\

MagLev Controller Data1

MagLev Controller Data\_P-6

MagLev Controller data\_I-.4

MagLev Controller Data\_D-.5

The graphs below show the output of the unaltered PID parameters on the left, juxtaposed with variations to P, I, and D on the right.



The output with unaltered parameters shows with an initial displacement of 5.1 (0.1 from x\_0), the damping effect of the controller oscillates with the application of current and settles out after about 6 seconds.

By increasing P, the output oscillations are increased by 6/5 of the unaltered P and take longer to settle out, roughly 8 seconds.

By increasing I by 0.2, the output oscillations do not visibly change much.

By increasing D by 0.2, the output shows an abrupt return to equilibrium with markedly fewer oscillations.

Further testing of the parameter coefficients would reveal a more complete picture of their impact on controller behavior.