PH 222 Digital Electronics and Microprocessors

Group Code: G21

Project Name: Simulation of Master-Slave Oscillations

Group Members:

Student 1: *Aditya Choudhury*

Student 2: *Arnab Choudhury*

Student 3: *Bhagyaraju Akepogu*

Pre – final report

Abstract: Our group altogether is interested to perform simulations on previously learned theoretical concepts and using electronic equipment as tangible system to further enable our appreciation of the concept. The idea is based on the concept of Non-Linear Dynamics and we'd like to demonstrate the simulation of non - linear Differential Equations (Master – Slave Oscillations) using physical systems which would further solidify the concept. As expected, this necessitates the Arduino to act as a medium for converting our coded differential equation simulation into real electronic results, essentially enabling us to carry out a tangible simulation.

Introduction: After brainstorming, the whole group brought out some of the coolest ideas we could think of and during the filtering process, we considered the feasibility of the project, time constraint and applications of our theoretical knowledge. We ended up with the idea of simulating the Master-Slave Oscillations which are driven by non – linear differential equations. We felt that since all of us had studied the relevant theory in our some of our previous courses, it was the ideal opportunity for us to implement this knowledge to further enhance the understanding of these concepts.

Design concept and implementation: In our project we are simulating a standard simple harmonic oscillator (as a precursor) and a coupled master slave oscillator. We are demonstrating our simulations using a LED matrix by lighting up LEDs as the oscillators oscillate with time. The LED matrix is controlled using an Arduino IDE code authored by us. This code utilizes the 4^{th} order Runge Kutta method of numerically solving differential equations. Since these non – linear equations do not have analytical solutions, we chose to use numerical methods to solve them.

The initial conditions and parameters for these equations will be given by the user. The interface we created for the user consists of a breadboard on which we use switches, potentiometers and a D Flip-Flop to control the input. These apparatuses essentially are used for the purpose of setting the Initial Conditions and parameters by the user. These values are read by the Arduino, via Jumper Wires, and they set the parameters for the Differential Equations to-be-executed. The user can see what initial values/parameters they are entering on the same LED Matrix, we will specify in more details below. We are using the first switch for two functions. Firstly, we use it to select the "display mode of the LED matrix, pressing the switch down for a prolonged time enables it to switch the display from showing the oscillations to the input parameters and vice versa. The second purpose of this switch enables us to choose which oscillator we want to give initial conditions for as well as the parameters we give to the system. FWe use a second switch to choose which one of the initial conditions (position or velocity) the user wants to give to the Oscillator. The same switch enables the user to choose the "Oscillator type" by long pressing, that is the user can choose which of the two simulations they want to demonstrate. Finally, we utilize 2 potentiometers (coarse knob and fine knob) to fix the value of initial conditions, the value of which will be reflected on the digital display. For this purpose, we have encoded a "seven segment display" algorithm on to our LED matrix. These switches are controlled through a d flip flop, using the fact that the memory only updates every clock pulse, to prevent multiple contacts from causing issues while pressing the switch.

List of components used: Arduino Uno, Jumper Wires, Breadboard (1), Switches (2), Potentiometers (2), 32x8 LED Matrix (1), 9V battery (1), 74ls175 D flip flop ic (1)

List of fabrication used: Our project didn't require us to fabricate anything.

Results: The final envisioned result is having an interface for the user to control all variable parameters of our oscillators, enabling them to simulate any arbitrary oscillations, and verify that the theory studied previously holds generally. Our final project will have the LED Matrix as a display for both the oscillations and assist the user by displaying the values of the desired initial condition as the potentiometer is rotated. Finally, the breadboard will have 3 switches and 1 potentiometer, which will be where the user can control the outputs.

Summary: As of 5th of April 2024, we still are continuing with our project, dealing with the Interface on the breadboard and the code it requires. We started off great with breaking down our whole idea of project into 3 parts for 3 successive weeks. As mentioned earlier, our week 1 tasks on figuring the entire apparatuses, concepts, and the code has been done seamlessly. During the 2nd week, we also completed exhibiting the Simple Harmonic Motion and Damped

Oscillations accordingly. Then we thought of interchanging our 3rd and 4th week tasks since the pace of the project was well-going, we thought of not to give it a break. Hence during the third week, we were successful with the Master-Slave simulations too. At current, we are dealing with the Interface and we hope to finish it as previously mentioned.

Final report

Original Aim:

1) Electronically simulating Damped Oscillations using a LED matrix
2) Electronically simulating Master Slave Oscillations using a LED matrix
3) Using electronic components to make an interface for users to be able to control parameters and initial conditions for the Oscillators

Major Milestones:

1) Simulation of Simple Harmonic and Damped Oscillations on LED Matrix achieved
2) Simulation of Master Slave Oscillations on LED Matrix achieved
3) Interfacing of switches and buttons (along with the code for their various purposes) completed
4) Letter and Numerical displays on LED Matrix completed
5) Integration of all the modules (under certain conditions) achieved
6) Fixing minor issues to get a completely functional final project

Bare Minimum Modules:

1) The Simple Harmonic and Damped Oscillators simulations
2) The Master Slave Oscillator simulations
3) The electronic interfacing for user control
4) The repurposing of LED matrix to display parameter values during user input

How much was achieved: As we set practical goals, all our aims for the project were achieved.

Role of Arduino: The Arduino was essential to our project, the most useful feature of it being that instructions can be given dynamically, that is new instructions are given by it every clock cycle. Our methods of simulating differential equations are also iterative, making the Arduino an ideal tool for taking in new data every iteration to calculate the next point to be displayed.

Digital Modules: MAX7219 LED Display, 74LS175 D flip flop

Testing Methodology of each module: Module 1 and 2 were easily tested by digitally varying parameters in the code. We tested the LED Display by attempting to display arbitrary LED's using our code.

The testing of module 3 required several steps, first we tested each component on the bread board using DMM'S to check of they were functioning properly, we then tested the effects of pressing the switches and their selections by analyzing the outputs we were obtaining on different lengths of the switch being pressed, and editing the code/circuit to get desired results. We also tested the stability of our input from the potentiometer by analyzing the outputs and bringing about necessary changes.

Module 4 was tested by varying inputs from the Arduino and checking if the results printed in the IDE matches the ones displayed.

Problems faced and solutions attempted:

1) Oscillations were out of bound on the LED Matrix.
   Solution: Introduced normalization on LED Matrix
2) "Button Mashing" (rapid pressing of switches) effecting the inputs and modes of our system
   Solution: Inclusion of several checkers and flags to prevent any successive presses from effecting the inputs
3) Multiple contacts on pressing the switch
   Solution: Utilization of D flip flops for clear inputs
4) Some parts of the code had very high run time which caused major delay in the display of oscillations
   Solution: Faster (but slightly less accurate) replacement code for removal of delays

Interesting work – around: The value of parameters was difficult to control using a single potentiometer (due to noise and fluctuations), hence we decided to add a "fine knob" as a second potentiometer to give us better control on the parameter selections

Individual Contributions:

Aditya Choudhury – Responsible for all the coding required for modules 1, 2, 3, partially responsible for coding module 4

Arnab Choudhury – Shared responsibility for circuit building and coding of module 4 and module 3, analyzed theory behind the simulations

Bhagyaraju Akepogu – Shared responsibility for circuit building and coding of module 4

Learnings: Since we had already learnt the over – arching concepts of oscillations in previous courses, our major learnings were in the control and flow of data through electronic components. We learnt about the drastic effects which multiple contacts of switches can have on a system, and methods to reduce them. We also learnt methods of repurposing the same components, making our systems small and compact. We learnt the effect of noise in the data and how to improve the accuracy of the data.

Resources: *Synchronization of Master-Slave Oscillators: Analysis and experimental results by Shyam Krishnan, Shaunak Sen, Indra Narayan Kar*

https://www.sciencedirect.com/science/article/pii/S2405896320300574

Final Code:

Since the code is very large, it is uploaded on a separate file