



University of Kentucky  
UKnowledge

---

Theses and Dissertations--Mechanical  
Engineering

Mechanical Engineering

---


2019

## SINGLE-DEGREE-OF-FREEDOM EXPERIMENTS DEMONSTRATING ELECTROMAGNETIC FORMATION FLYING FOR SMALL SATELLITE SWARMS USING PIECEWISE-SINUSOIDAL CONTROLS

Ajin Sunny

*University of Kentucky*, [ajin.sunny@gmail.com](mailto:ajin.sunny@gmail.com)

Author ORCID Identifier:

 <https://orcid.org/0000-0002-7164-6074>

Digital Object Identifier: <https://doi.org/10.13023/etd.2019.466>

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

### Recommended Citation

Sunny, Ajin, "SINGLE-DEGREE-OF-FREEDOM EXPERIMENTS DEMONSTRATING ELECTROMAGNETIC FORMATION FLYING FOR SMALL SATELLITE SWARMS USING PIECEWISE-SINUSOIDAL CONTROLS" (2019). *Theses and Dissertations--Mechanical Engineering*. 146.  
[https://uknowledge.uky.edu/me\\_etds/146](https://uknowledge.uky.edu/me_etds/146)

This Master's Thesis is brought to you for free and open access by the Mechanical Engineering at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Mechanical Engineering by an authorized administrator of UKnowledge. For more information, please contact [UKnowledge@lsv.uky.edu](mailto:UKnowledge@lsv.uky.edu).

## **STUDENT AGREEMENT:**

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

## **REVIEW, APPROVAL AND ACCEPTANCE**

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Ajin Sunny, Student

Dr. Jesse B. Hoagg, Major Professor

Dr. Alexandre Martin, Director of Graduate Studies

SINGLE-DEGREE-OF-FREEDOM EXPERIMENTS  
DEMONSTRATING ELECTROMAGNETIC FORMATION  
FLYING FOR SMALL SATELLITE SWARMS USING  
PIECEWISE-SINUSOIDAL CONTROLS

---

THESIS

---

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering in the College of Engineering at the University of Kentucky

By  
Ajin Sunny

Lexington, Kentucky

Director: Dr. Jesse B. Hoagg, Professor of Mechanical Engineering

Lexington, Kentucky

2019

Copyright© Ajin Sunny 2019  
<https://orcid.org/0000-0002-7164-6074>

## ABSTRACT OF THESIS

### SINGLE-DEGREE-OF-FREEDOM EXPERIMENTS DEMONSTRATING ELECTROMAGNETIC FORMATION FLYING FOR SMALL SATELLITE SWARMS USING PIECEWISE-SINUSOIDAL CONTROLS

This thesis presents a decentralized electromagnetic formation flying (EMFF) control method using frequency-multiplexed sinusoidal control signals. We demonstrate the EMFF control approach in open-loop and closed-loop control experiments using a single-degree-of-freedom testbed with an electromagnetic actuation system (EAS). The EAS sense the relative position and velocity between satellites and implement a frequency-multiplexed sinusoidal control signal. We use a laser-rangefinder device to capture the relative position and an ARM-based microcontroller to implement the closed-loop control algorithm. We custom-design and build the EAS that implements the formation control in one dimension. The experimental results in this thesis demonstrate the feasibility of the decentralized formation control algorithm between two satellites.

KEYWORDS: Electro Magnetic Formation Flying, Small Satellites, Decentralized Control

Author's signature: Ajin Sunny

Date: December 6, 2019

SINGLE-DEGREE-OF-FREEDOM EXPERIMENTS  
DEMONSTRATING ELECTROMAGNETIC FORMATION  
FLYING FOR SMALL SATELLITE SWARMS USING  
PIECEWISE-SINUSOIDAL CONTROLS

By  
Ajin Sunny

Director of Thesis: Prof. Jesse B. Hoagg

Director of Graduate Studies: Prof. Alexandre Martin

Date: December 6, 2019

*To my parents and my sister,  
for their love and sacrifice*

## ACKNOWLEDGMENTS

First and foremost, I would like to thank my parents for their constant support, love, and their financial sacrifice for me and my sister to support our education. I could not have accomplished any of this without them. Throughout the last two and a half years of my graduate school, I have been through severe hardships, and by the grace of my parent's support, I was able to persist through many hurdles with courage and willpower.

Second, I would like to thank my advisor, Professor Jesse Hoagg, for his continued guidance, support, and encouragement throughout my studies. His efforts in shaping and molding me into a stronger disciplined professional for the future have instilled in me to be academically, a stronger individual. I can assure myself and to the rest of the world that I am a person with higher intellect now than on the first day of my graduate school.

Third, I would like to thank all the staff and administrative members of the Department of Mechanical Engineering at the University of Kentucky for their administrative and logistical support for the research lab.

Finally, I would like to thank my labmates Roshan Chavan, Zachary Lippay, Chris Heintz, Zahra Abbasi, Pedram Rabiee, Shaoqian Wang, and Ali Mousavi who embraced the spirit of the lab with collaboration, communication, and insightful brainstorming sessions. To these friendships that will last long, I am very grateful.

# CONTENTS

<b>Acknowledgments</b> . . . . .	<b>iii</b>
<b>Contents</b> . . . . .	<b>iv</b>
<b>List of Figures</b> . . . . .	<b>vi</b>
<b>List of Additional Files</b> . . . . .	<b>vii</b>
<b>Chapter 1 — Introduction and Motivation</b> . . . . .	<b>1</b>
1.1 Electromagnetic Formation Flying . . . . .	2
1.2 Summary of Contributions . . . . .	4
<b>Chapter 2 — Equations of Motion and Formation Control</b> . . . . .	<b>5</b>
2.1 Three-Dimensional Problem Formulation . . . . .	5
2.2 Specialization to One Dimension and Approximate Dynamics . . . . .	8
2.3 Formation Control Algorithm . . . . .	10
<b>Chapter 3 — Experimental Hardware and Software</b> . . . . .	<b>13</b>
3.1 Electromagnetic Actuation System Hardware . . . . .	13
3.1.1 EAS Base Platform Design . . . . .	15
3.1.2 EAS Power and Electronics . . . . .	15
3.1.3 Coil Design . . . . .	18
3.2 EAS Sensing and Control . . . . .	22
3.2.1 EAS Software . . . . .	23
3.2.2 Data Acquisition . . . . .	23
3.2.3 Velocity Estimation . . . . .	25
3.2.4 Sinusoidal Signal Generation . . . . .	26
3.2.5 XBee Radio Communication . . . . .	26
<b>Chapter 4 — Experimental Results and Discussion</b> . . . . .	<b>28</b>
4.1 Open-Loop Experiments . . . . .	29
4.2 Closed-Loop Experiments . . . . .	30
4.3 Discussion of Experimental Results . . . . .	36
<b>Chapter 5 — Conclusions and Future Work</b> . . . . .	<b>38</b>
<b>Appendix A</b> . . . . .	<b>40</b>
<b>Appendix B</b> . . . . .	<b>42</b>
<b>Bibliography</b> . . . . .	<b>58</b>



Vita . . . . . 62

## LIST OF FIGURES

2.1	Each satellite is equipped with an electromagnetic actuation system consisting of three orthogonal coils. The relative positions of the satellites are controlled by the interaction of equal-and-opposite electromagnetic forces produced by the actuation systems. . . . .	7
3.1	Isometric view of a testbed modeled in CAD software. . . . .	14
3.2	Side view of the testbed placed on an optical table. The two EAS sit on a low-friction linear air track. . . . .	15
3.3	EAS platform design. . . . .	16
3.4	Top-down view of the EAS on the air track displaying the four main electrical components of the EAS. . . . .	17
3.5	Electrical schematic of the EAS power electronics with red wires depicting positive input and output current and blue wires depicting negative input and output current. . . . .	18
3.6	Electromagnetic coils with attractive and repulsive magnetic field lines that represents the dipole interaction of a bar magnet. . . . .	19
3.7	Force exerted by the EM coil at $x = 0.1$ m for different radius of coil in meters represented using different colors in the legend. . . . .	20
3.8	Force exerted by the EM coil at $x = 0.5$ m for different radius of coil in meters represented using different colors in the legend. . . . .	21
3.9	Force exerted by the EM coil at $x = 0.9$ m for different radius of coil in meters represented using different colors in the legend. . . . .	21
3.10	Side view and front view of the EAS on the air track displaying the Arduino Due microcontroller, the wireless SD shield, reflector, and laser-ranging sensor. . . . .	23
3.11	Software flowchart. . . . .	24
3.12	Block diagram that shows the network configuration of the experiment. . . . .	27
4.1	Open-loop control, where each EAS coil is actuated with a 10 Hz sinusoidal current with amplitude 1.25 A. The sinusoidal currents on the EAS are $180^\circ$ out of phase, which results in a repulsion force. . . . .	30
4.2	Open-loop control, where each EAS coil is actuated with a 10 Hz sinusoidal current with amplitude 1.25 A. The sinusoidal currents on the EAS are $180^\circ$ out of phase, which results in a repulsion force. . . . .	31
4.3	Closed-loop formation control, where $r_{ij}(0) = 0.25$ m and $d_{ij} = 0.35$ m. . . . .	32
4.4	Closed-loop formation control, where $r_{ij}(0) = 0.45$ m and $d_{ij} = 0.35$ m. . . . .	33
4.5	Closed-loop formation control, where $r_{ij}(0) = 0.30$ m and $d_{ij} = 0.35$ m. . . . .	34
4.6	Closed-loop formation control, where $r_{ij}(0) = 0.40$ m and $d_{ij} = 0.35$ m. . . . .	35

## LIST OF ADDITIONAL FILES

1. clsat1.pdf - Program used for first unit of the electromagnetic actuation system
2. clsat2.pdf - Program used for second unit of the electromagnetic actuation system
3. EXP\_PHOTO.jpg - Photo of the experimental testbed

# Chapter 1

---

## Introduction and Motivation

Small satellites have the potential to play a vital role in future space missions. In comparison to conventional large satellites, small satellites are less expensive to build and launch. In addition, multiple small satellites operating in coordination could have operational capabilities beyond those of large conventional satellites. Small-satellite swarms have application to a broad range of space missions, including space exploration, surveillance, comet detection, cosmological and biological studies, and space-weather monitoring [1, 2]. For example, small satellites flying in formation could provide high accuracy in measuring reflected energy from the Earth [3]. Unlike large monolithic satellites, small satellites are small light weight, which makes them easier and less expensive to place in space. However, many traditional actuation systems are not well suited to small satellites. Traditional actuation systems use propellant thrusters, which deplete over time and limit the mission life. In addition, propellant thrusters are a source of contamination for applications such as sparse aperture telescopes (e.g., NASA's Terrestrial Planet Finder). Contaminants from propellants can be deposited on optical surfaces of telescopes, and they can create plumes that may obstruct the telescope's line of sight. Over the last two decades, research efforts have been devoted to developing new small-satellite technologies, which extend on mission life, such as actuation systems that use renewable energy resource [4, 5]. There has also been substantial work on developing formation control

methods for small satellites. [1–3, 6–9].

Electromagnetic formation flying (EMFF) approaches were introduced by the Space Systems Lab at Massachusetts Institute of Technology [9]. EMFF could potentially replace or supplement conventional propulsion systems based on propellant thrusters [6, 7]. The main advantage of EMFF over conventional propellant thrusters is that their power source is renewable. EMFF uses electromagnetic (EM) coils to generate forces and torques between different satellites in a formation [8]. Small-satellite swarms that incorporate EMFF techniques could simplify the difficulties in coordinated control and enable applications such as holding position and orientation of telescopic mirrors in distributed aperture telescopes. However, formation control using EMFF is challenging. The intersatellite force and moment between two EMFF satellites is a nonlinear function of the EM fields produced by both satellites, as well as their relative position and orientation. Moreover, the EM field of one satellite interacts with the field of all other satellites.

### 1.1 Electromagnetic Formation Flying

Initially, EMFF was developed using a linearized models of the satellite dynamics and optimal controller designs, which are useful despite the nonlinearities of systems' dynamics [10]. The EMFF concept was demonstrated as a proof-of-concept for small satellites using a ground-based testbed in [11]. The testbed uses direct current (DC) to generate magnetic fields for control. EMFF can be implemented on a spacecraft by driving DC through three orthogonal electromagnetic coils to create a steerable magnetic dipole in three dimensions [12]. However, for multiagent spacecraft formations, EM actuation with DC generally requires a centralized formation control strategy. Centralized control strategies have been developed in [8, 12, 13] for small-satellites formation control. However, centralized formation control approaches require that feedback of relevant states of each satellite be transmitted to a centralized algorithm, which determines the controls for each satellite. For example [8], derives the dynamics for an  $N$ -satellite EM formation (in 2D) for deep space missions and presents a nonlinear control law using potential functions for the formation control and re-

configuration with collision avoidance. Alternatively, decentralized formation-flying algorithm allow each satellite to compute its own control signals based on feedback from its self and neighboring satellites. However, decentralized EMFF control is challenging, because the EM forces used for control are coupled across the entire satellite formation [14].

Decentralized formation control is commonly addressed using single or double integrator consensus algorithms. Consensus-based formation control algorithms have been presented for agents with linear dynamics [15–18], rotational dynamics [19–24], and certain types of nonlinear dynamics [25–29]. However, these consensus algorithms are based on a decoupled forcing model, where each agent can be controlled independent of the other agents. Controlling satellites independently with EMFF is not possible, and interagent force coupling cannot be arbitrarily specified. Thus, consensus algorithms cannot be directly applied for EMFF.

Decentralized EMFF can be achieved using frequency-multiplexed alternating current (AC) control signals to reduce the coupling effects [14, 30]. Unlike DC, AC with multiple frequencies can be used to energize the EM coils for interaction among specific pairs of satellites that are energized at the same AC frequency. Frequency multiplexing is a decoupling strategy that uses multi-frequency sinusoidal control signals to allow the interaction between specific satellite pairs, while limiting the interaction with others [14]. Decentralized EMFF control based on approximate formation dynamics is derived in [14]. In [14], a decentralized EMFF controller is designed using information from only the satellites in its neighbor set. The multiplexed AC approach in [14] creates a decoupling effect that enables interaction forces between pairs of satellites, while limiting the interaction with other satellites.

Another demonstration in [31] uses the frequency-multiplexed AC approach in an experiment with 2D motion of 4 EM coils. The electrical and electronics system used in this demonstration is offboard, stationary, and tethered to the satellite. The experiment in [31] successfully demonstrates the relative position reading with a sub-micron level resolution. However, it lacks an onboard lightweight power electronics for EM actuation.

## 1.2 Summary of Contributions

This thesis presents a multiplexed AC approach for decentralized EMFF and demonstrates this algorithm in experiments using a single-degree-of-freedom testbed. This testbed includes linear airtracks for low-friction motion and multiple small satellites equipped with an electromagnetic actuation system (EAS). The experiments in this thesis use two EAS-equipped satellites, which perform formation control maneuvers using a decentralized frequency-multiplexed AC formation control algorithm. Note that for two satellites, DC formation control approaches can be implemented in a decentralized fashion. Nevertheless, the experimental results in this thesis demonstrate the feasibility of EMFF using decentralized frequency-multiplexed AC algorithms.

The experimental results include open-loop and closed-loop control experiments that demonstrates EMFF with multiplexed AC control signals. We use a decentralized closed-loop algorithm that control the two EAS. The controller uses feedback of the relative position and relative velocity of the two satellites. We present the results of two open-loop and six closed-loop experiments. The two open-loop experiments test the attractive and repulsive force generated by the EM coils on the EAS. The six closed-loop experiments successfully demonstrates closed-loop formation control.

We use a simple and light-weight design for feasibility of translational motion of the satellites in the open-loop and closed-loop experiments. The closed-loop control algorithm implemented on a microcontroller uses onboard sensing to detect the relative position in real-time. The microcontroller estimates the relative velocity using a backward-Euler and low-pass-filter approach.

# Chapter 2

---

## Equations of Motion and Formation Control

### 2.1 Three-Dimensional Problem Formulation

Bold symbols are used to denote physical vectors. If  $x \in \mathbb{R}^3$ , then  $\|x\|$  is the 2-norm of  $x$ . If  $y \in \mathbb{R}$ , then  $|y|$  is the absolute value of  $y$ . Let  $\mathcal{J} \triangleq \{1, \dots, N\}$ , where  $N$  is a positive integer, and define  $\mathcal{P} \triangleq \{(i, j) \in \mathcal{J} \times \mathcal{J} : i \neq j\}$ , which is the set of ordered pairs. Unless otherwise stated, all statements in this chapter that involve the subscript  $i$  are for all  $i \in \mathcal{J}$ , and all statements that involve the subscript  $ij$  are for all  $(i, j) \in \mathcal{P}$ .

Consider a system of  $N$  identical satellites. The vector  $\mathbf{r}_i$  locates the mass center of satellite  $i$  relative to the origin of an inertial reference frame  $\mathcal{F}_I$ . Define the relative position vector  $\mathbf{r}_{ij} \triangleq \mathbf{r}_i - \mathbf{r}_j$ , which locates the mass center of satellite  $i$  relative to the mass center of satellite  $j$ . Let  $r_i \in \mathbb{R}^3$  and  $r_{ij} \in \mathbb{R}^3$  denote  $\mathbf{r}_i$  and  $\mathbf{r}_{ij}$  resolved in  $\mathcal{F}_I$ .

Next, let  $\mathcal{F}_{ij}$  be a reference frame whose orientation depends on the direction of  $\mathbf{r}_{ij}$ . Specifically, let  $\mathbf{i}_{ij}$ ,  $\mathbf{j}_{ij}$ , and  $\mathbf{k}_{ij}$  be the orthogonal unit vectors of  $\mathcal{F}_{ij}$ . Then,  $\mathcal{F}_{ij}$  is aligned such that  $\mathbf{r}_{ij} \cdot \mathbf{i}_{ij} = \|\mathbf{r}_{ij}\|$ . Thus, resolving  $\mathbf{r}_{ij}$  in  $\mathcal{F}_{ij}$  yields  $[\|\mathbf{r}_{ij}\| \ 0 \ 0]^T$ . Let  $R_{ij} : [0, \infty) \rightarrow \text{SO}(3)$  be the rotation matrix from  $\mathcal{F}_{ij}$  to  $\mathcal{F}_I$ . It follows that  $r_{ij} = R_{ij}[\|\mathbf{r}_{ij}\| \ 0 \ 0]^T$ . Since  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$ , it follows that  $\mathcal{F}_{ji}$  is aligned such that



$\mathbf{i}_{ji} = -\mathbf{i}_{ij}$ . We assume that  $\mathbf{j}_{ji} = -\mathbf{j}_{ij}$  and  $\mathbf{k}_{ji} = \mathbf{k}_{ij}$ , which implies that  $R_{ji} = R_{ij}\Phi$ , where

$$\Phi \triangleq \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

See Fig. 2.1 for a depiction of vectors and reference frames.

Each satellite is equipped with an electromagnetic actuation system (EAS), which is used to generate intersatellite forces. Let the vector  $\mathbf{F}_{ij}$  denote the normalized electromagnetic force per unit mass applied to satellite  $i$  by satellite  $j$ , and let  $\mathcal{F}_{ij} \in \mathbb{R}^3$  denote  $\mathbf{F}_{ij}$  resolved in  $\mathcal{F}_{ij}$ . The intersatellite forces are modeled by

$$F_{ij} \triangleq \frac{1}{\|r_{ij}\|^4} U_{ij}, \quad (2.1)$$

where

$$U_{ij} \triangleq \begin{bmatrix} 2X_i X_j - Y_i Y_j + Z_i Z_j \\ -X_i Y_j - X_j Y_i \\ X_i Z_j - X_j Z_i \end{bmatrix} \quad (2.2)$$

is the *intersatellite control* on satellite  $i$  between satellite  $i$  and  $j$ , and  $X_i, Y_i, Z_i$  are the *actuator controls*. Note that  $U_{ji} = -\Phi U_{ij}$  and  $F_{ji} = -\Phi F_{ij}$ .

The force model (2.1) and (2.2) is based on an EAS consisting of three orthogonal electromagnetic coils, where each coil is modeled as an electric dipole. Each coil of satellite  $i$  is actuated by an independent current signal, which is dependent on  $X_i, Y_i$ , and  $Z_i$  and the orientation of the coil relative to  $\mathcal{F}_{ij}$ . More details on electromagnetic actuation systems for satellites and the electromagnetic force model (2.1) can be found in [32] and [14].

Assuming that there are no other applied forces, the position of satellite  $i$  satisfies

$$\ddot{r}_i(t) = \sum_{(i,j) \in \mathcal{P}} \frac{1}{\|r_{ij}(t)\|^4} R_{ij}(t) U_{ij}(t). \quad (2.3)$$

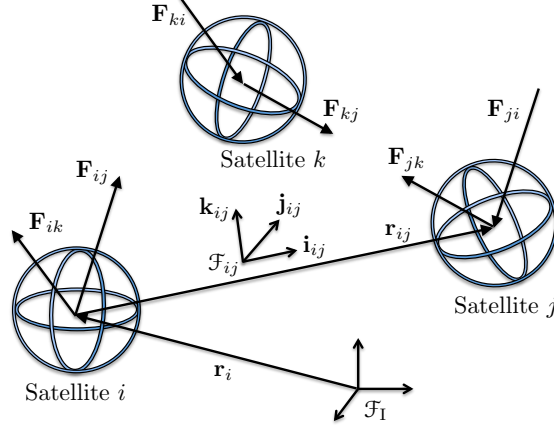


Figure 2.1: Each satellite is equipped with an electromagnetic actuation system consisting of three orthogonal coils. The relative positions of the satellites are controlled by the interaction of equal-and-opposite electromagnetic forces produced by the actuation systems.

Since  $F_{ji} = -\Phi F_{ij}$  and  $R_{ji} = R_{ij}\Phi$ , it follows that  $R_{ij}F_{ij} + R_{ji}F_{ji} = 0$ . Thus, the electromagnetic forces have equal magnitude and opposite direction. It then follows from (2.3) that  $\sum_{i \in \mathcal{J}} \dot{r}_i(t) = 0$ , which implies that sum of the linear momentum of all satellites is conserved. Thus, the relative positions of the satellites can be altered by the electromagnetic forces, but the mass center of the formation is unaffected.

Next, we describe the interagent communication (i.e., feedback) structure. Let  $\mathcal{G} = (\mathcal{J}, \mathcal{E})$  be an undirected graph, where  $\mathcal{J}$  is the vertex set and  $\mathcal{E} \subset \mathcal{J} \times \mathcal{J}$  is the undirected edge set. The neighbor set of satellite  $i$  is defined as  $\mathcal{N}_i \triangleq \{j \in \mathcal{N}_i : (j, i) \in \mathcal{E}\}$ . Moreover, we assume that the undirected graph  $\mathcal{G}$  is connected and that satellite  $i$  has access to  $\{r_{ij}\}_{j \in \mathcal{N}_i}$  and  $\{\dot{r}_{ij}\}_{j \in \mathcal{N}_i}$  and for feedback. We note that  $\{R_{ij}\}_{j \in \mathcal{N}_i}$  can be determined from  $\mathbf{r}_{ij}$  based on a predefined convention for specifying the directions of  $\mathbf{j}_{ij}$  and  $\mathbf{k}_{ij}$  relative  $\mathcal{F}_I$ . For practical implementation using electromagnetic coils, each satellite also requires knowledge of its coils' orientations relative to  $\mathcal{F}_{ij}$ . The coil actuation signals of satellite  $i$  are determined from  $X_i$ ,  $Y_i$ , and  $Z_i$  and the rotation matrix from  $\mathcal{F}_{ij}$  to a body-fixed frame of satellite  $i$  that is aligned with the orthogonal coils.

Let  $d_{ij} \in \mathbb{R}^3$  be the desired position of satellite  $i$  relative to satellite  $j$ , and define the formation error  $e_{ij} \triangleq r_{ij} - d_{ij}$ . The objective is to design actuator controls  $X_i$ ,  $Y_i$ , and  $Z_i$  that rely solely on local feedback of neighboring satellites and makes  $\|e_{ij}\|$

small. Our approach is based on actuator controls that are sums of sinusoidal signals, each having a piecewise-constant amplitude. These piecewise-sinusoidal controls (on average) decouple all non-neighbor satellite interactions and yield a decentralized average model of (2.3). The average model is used to design actuator controls that rely on sampled local feedback signals.

## 2.2 Specialization to One Dimension and Approximate Dynamics

In this section, we specialize the three-dimensional dynamics (2.2) and (2.3) to single-degree-of-freedom (ie., one-dimensional) motion, which is the focus of this thesis and the focus of the experiments presented in Chapter 4. Next, we let the controls  $X_i$  be piecewise sinusoidal, where the amplitudes of the piecewise sinusoids are updated at discrete instants in time. The more general three-dimensional case is considered in [14] and [33].

First, we specialize (2.2) and (2.3) to one dimension, where we assume without loss of generality, that the motion is in the  $\mathbf{i}_{ij}$  direction. In this case, (2.2) and (2.3) yields

$$\ddot{r}_i(t) = \sum_{(i,j) \in \mathcal{P}} \frac{2}{|r_{ij}(t)|^4} X_i(t) X_j(t). \quad (2.4)$$

Note that (2.4) is the mass-normalized dynamics. More generally, the dynamics are

$$m\ddot{r}_i(t) = \sum_{(i,j) \in \mathcal{P}} \frac{3\mu_0}{2\pi|r_{ij}(t)|^4} X_i'(t) X_j'(t), \quad (2.5)$$

where  $m$  is the mass of the satellite,  $\mu_0$  is the magnetic constant, and  $X_i'(t) = \sqrt{\frac{4\pi m}{3\mu_0}} X_i(t)$

Next, we consider the case where  $X_i$  is piecewise sinusoidal. For all  $(i, j) \in \mathcal{J} \times \mathcal{N}_i$ , let  $\omega_{ij} > 0$  be the interaction frequency between the  $i$ th and  $j$ th satellite, where  $\omega_{ij} = \omega_{ji}$ , and for all  $i_1, i_2 \in \mathcal{J}$  such that  $i_1 \neq i_2$ ,  $\omega_{i_1 j} \neq \omega_{i_2 j}$ . Let  $T_s > 0$  be a common multiple of  $\{2\pi/\omega_{ij}\}_{(i,j) \in \mathcal{J} \times \mathcal{N}_i}$ , and note that  $T_s$  is the update period for the piecewise

sinusoidal control. Then, for each  $k \in \mathbb{N}$  and for all  $t \in [kT_s, (k+1)T_s)$ , consider the control

$$X_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij,k} \sin \omega_{ij} t, \quad (2.6)$$

where for all  $(i, j) \in \mathcal{J} \times \mathcal{N}_i$ ,  $a_{ij,k} \in \mathbb{R}$  are the amplitudes of the sinusoids in (2.6). In the next section, we present a feedback control for updating  $a_{ij,k}$  using feedback of  $\{r_{ij}\}_{j \in \mathcal{N}_i}$  and  $\{\dot{r}_{ij}\}_{j \in \mathcal{N}_i}$ . Before presenting the feedback control, we show that the piecewise-sinusoidal control (2.6) results in an intersatellite control  $X_i(t)X_j(t)$  that on average depends on only  $a_{ij,k}$  and  $a_{ji,k}$ .

For all  $k \in \mathbb{N}$  and all  $t \in [kT_s, (k+1)T_s)$ , define

$$\bar{U}_{ij}(t) \triangleq \frac{1}{T_s} \int_{kT_s}^{(k+1)T_s} X_i(\tau) X_j(\tau) d\tau, \quad (2.7)$$

which is the average intersatellite control over the interval  $[kT_s, (k+1)T_s)$ . The following result shows that the average intersatellite control  $\bar{U}_{ij}(t)$  depends on only  $a_{ij,k}$  and  $a_{ji,k}$ .

**Proposition 2.1.** For all  $k \in \mathbb{N}$  and all  $t \in [kT_s, (k+1)T_s)$ ,

$$\bar{U}_{ij}(t) = \begin{cases} \frac{a_{ij,k} a_{ji,k}}{2}, & \text{if } j \in \mathcal{N}_i, \\ 0, & \text{if } j \notin \mathcal{N}_i. \end{cases} \quad (2.8)$$

*Proof.* Since  $T_s$  is a common multiple of  $\{2\pi/\omega_{ij}\}_{(i,j) \in \mathcal{J} \times \mathcal{N}_i}$  and for  $i_1 \neq i_2$ ,  $\omega_{i_1 j} \neq \omega_{i_2 j}$ , it follows from (2.6) and (2.7) that

$$\begin{aligned} \bar{U}_{ij}(t) &= \frac{1}{T_s} \int_{kT_s}^{(k+1)T_s} \left[ \sum_{l \in \mathcal{N}_i} a_{il,k} \sin \omega_{il} t \right] \left[ \sum_{l \in \mathcal{N}_j} a_{jl,k} \sin \omega_{jl} t \right] dt \\ &= \frac{1}{T_s} \int_{kT_s}^{(k+1)T_s} a_{ij,k} a_{ji,k} \sin^2 \omega_{ij} t dt \end{aligned}$$

$$\begin{aligned}
&= \frac{a_{ij,k}a_{ji,k}}{T_s} \left[ \frac{t}{2} - \frac{\sin 2\omega_{ij}t}{4\omega_{ij}} \right]_{t=kT_s}^{(k+1)T_s} \\
&= \frac{a_{ij,k}a_{ji,k}}{2},
\end{aligned}$$

where for all  $j \notin \mathcal{N}_i$ ,  $a_{ij,k} = 0$ . □

Proposition 2.1 implies that the piecewise-sinusoidal control (2.6) yields an average intersatellite control  $\bar{U}_{ij}(t)$  that is nonzero if and only if  $j \in \mathcal{N}_i$ . Therefore, for all  $k \in \mathbb{N}$  and all  $t \in [kT_s, (k+1)T_s)$ , we approximate the dynamics (2.4) and control (2.6) by the average dynamics

$$\ddot{\bar{r}}_i(t) = \sum_{j \in \mathcal{N}_i} \frac{2}{|\bar{r}_{ij}(t)|^4} \bar{U}_{ij}(t), \tag{2.9}$$

where  $\bar{r}_i(t) \in \mathbb{R}$  and  $\bar{r}_{ij}(t) \in \mathbb{R}$  are the approximations of  $r_i(t)$  and  $r_{ij}(t)$ , respectively. In the next section, we present a feedback control for the amplitudes  $a_{ij,k}$  of the piecewise-sinusoidal control (2.6) such that formation control is approximately achieved, that is,  $r_{ij}(t)$  tends to approximately  $d_{ij}$ .

### 2.3 Formation Control Algorithm

For all  $k \in \mathbb{N}$  and all  $(i, j) \in \mathcal{J} \times \mathcal{N}_i$ , consider the feedback control

$$a_{ij,k} = \gamma_{ij,k} r_{ij}^2(kT_s) \sqrt{|\alpha_{ij}(r_{ij}(kT_s) - d_{ij}) + \beta_{ij} \dot{r}_{ij}(kT_s)|}, \tag{2.10}$$

where for all  $(i, j) \in \mathcal{J} \times \mathcal{N}_i$ ,  $\alpha_{ij} > 0$  and  $\beta_{ij} > 0$ ; for all  $(i, j) \notin \mathcal{J} \times \mathcal{N}_i$ ,  $\alpha_{ij} = 0$  and  $\beta_{ij} = 0$ ; for all  $(i, j) \in \mathcal{P}$ ,  $\alpha_{ij} = \alpha_{ji}$  and  $\beta_{ij} = \beta_{ji}$ ; and for all  $(i, j) \in \mathcal{J} \times \mathcal{N}_i$ ,

$$\gamma_{ij,k} \triangleq \begin{cases} -1 & , \quad \text{if } i > j, \\ \text{sgn} \left[ \alpha_{ij}(r_{ij}(kT_s) - d_{ij}) + \beta_{ij}\dot{r}_{ij}(kT_s) \right] & , \quad \text{if } i < j. \end{cases} \quad (2.11)$$

Thus, it follows from Proposition 2.1, (2.10), and (2.11) that for all  $k \in \mathbb{N}$  and all  $t \in [kT_s, (k+1)T_s)$ ,

$$\bar{U}_{ij}(t) = \frac{-r_{ij}^4(kT_s)}{2} [\alpha_{ij}(r_{ij}(kT_s) - d_{ij}) + \beta_{ij}\dot{r}_{ij}(kT_s)]. \quad (2.12)$$

To analyze the closed-loop performance with the control (2.10) and (2.11), define the *ideal average intersatellite control*

$$\bar{U}_{ij*}(t) \triangleq \frac{-\bar{r}_{ij}^4(t)}{2} [\alpha_{ij}(\bar{r}_{ij}(t) - d_{ij}) + \beta_{ij}\dot{\bar{r}}_{ij}(t)]. \quad (2.13)$$

Note that the average intersatellite control (2.12) is a sampled-data implementation of the ideal average intersatellite control (2.13), where  $\bar{r}_{ij}(t)$  and  $\dot{\bar{r}}_{ij}(t)$  are replaced by the sampled data  $r_{ij}(kT_s)$  and  $\dot{r}_{ij}(kT_s)$ , and a zero-order hold is applied.

Next, define the approximate formation error

$$\bar{e}_{ij}(t) = \bar{r}_{ij}(t) - d_{ij}, \quad (2.14)$$

which is an approximation of  $e_{ij}(t)$ . If  $\bar{U}_{ij} = \bar{U}_{ij*}$ , then it follows from (2.9) and (2.13) that

$$\ddot{\bar{r}}_i(t) = - \sum_{j \in \mathcal{N}_i} [\alpha_{ij}\bar{e}_{ij}(t) + \beta_{ij}\dot{\bar{e}}_{ij}(t)], \quad (2.15)$$

which is equivalent to the closed-loop double integrator with a standard linear con-

sensus algorithm [34]. The following theorem is the main stability result in this thesis. A proof of this result is in [34].

**Theorem 2.1.** Consider the closed-loop dynamics (2.15), which consists of the average dynamics (2.9) and the intersatellite control  $\bar{U}_{ij} = \bar{U}_{ij^*}$ , where  $\bar{U}_{ij^*}$  is given by (2.13). Assume that the graph  $\mathcal{G} = (\mathcal{J}, \mathcal{E})$  is connected. Then, for all initial conditions and all  $(i, j) \in \mathcal{P}$ ,  $\lim_{t \rightarrow \infty} \bar{e}_{ij}(t) = 0$

Theorem 2.1 demonstrates that the ideal average intersatellite control (2.13) forces the average formation error  $\bar{e}_{ij}$  to zero asymptotically, which suggests that the sampled-data formation control (2.10) and (2.11) should achieve approximate formation control.

# Chapter 3

---

## Experimental Hardware and Software

In this chapter, we present the structural and mechanical design of the electromagnetic actuation system (EAS) testbed. Figure 3.1 shows a schematic of the single-degree-of-freedom (SDOF) testbed with three EAS. The testbed has the following components: four Eisco quiet air sources, two Eisco linear air tracks (PH0362A), and two custom-designed EAS. The four quiet air sources supply regulated air to the two linear air tracks. We connect the air sources to the linear air tracks using the air track inlets as shown in Figure 3.2. The EAS are the components of the testbed responsible for electromagnetic (EM) actuation.

### 3.1 Electromagnetic Actuation System Hardware

In this section, we describe the EAS hardware platform required to perform open-loop and closed-loop control experiments. Appendix A documents the names of the hardware components. Appendix B provides the software that runs the open-loop and closed-loop control algorithms for the two EAS.

The EAS includes an EM coil and the electronics platform. The EM coil is responsible for passing alternating currents (AC) and the electronics platform provides power and runs the closed-loop sinusoidal control algorithm (2.6), (2.10), and (2.11). The frame for the EM coil is 3D printed using standard polylactic acid material.



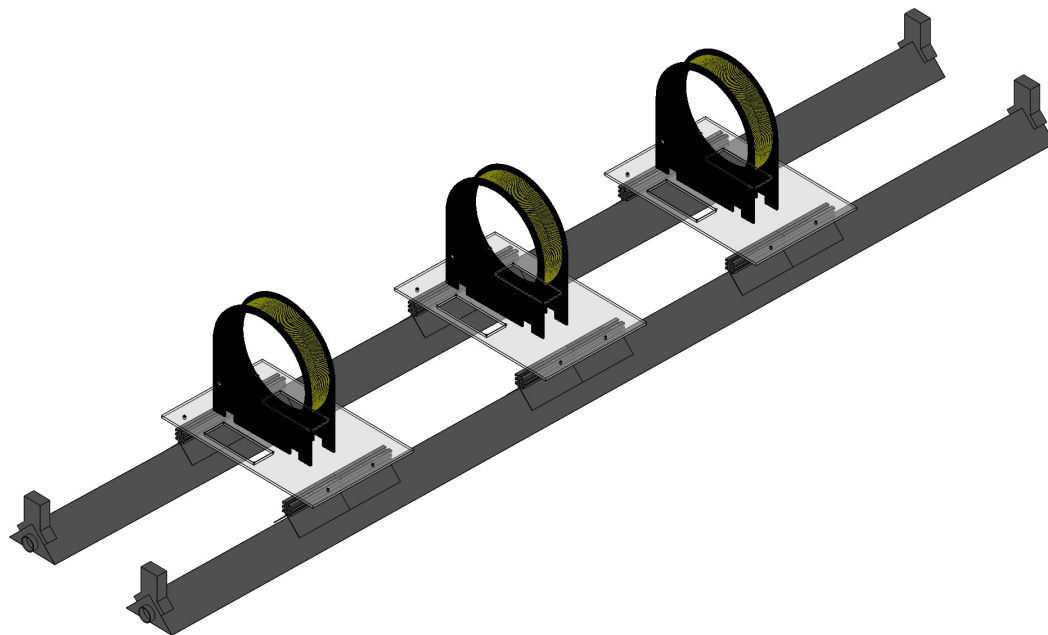


Figure 3.1: Isometric view of a testbed modeled in CAD software.

The frame is designed with a minimum area of contact at its underside to ensure uniform distribution of weight across the platform. On the frontside and backside of the frame, there are two openings at the base to route the electrical wires of 14 AWG that can carry high current.

We wind the coil using a multilayer multirow winding technique because of its advantages over other conventional winding techniques. Multilayer multirow winding design approach is useful in the case of reducing power loss in AC circuits. Additionally, they have advantages regarding skin and proximity effects in comparison to the single-layer winding process [35, 36]. We use 22 AWG magnet wire with a polyamide overcoat as the material for current-carrying conductor of the coil. The polyamide overcoat is used to prevent short circuits.

The following subsections describe the steps to design and build the base platform,

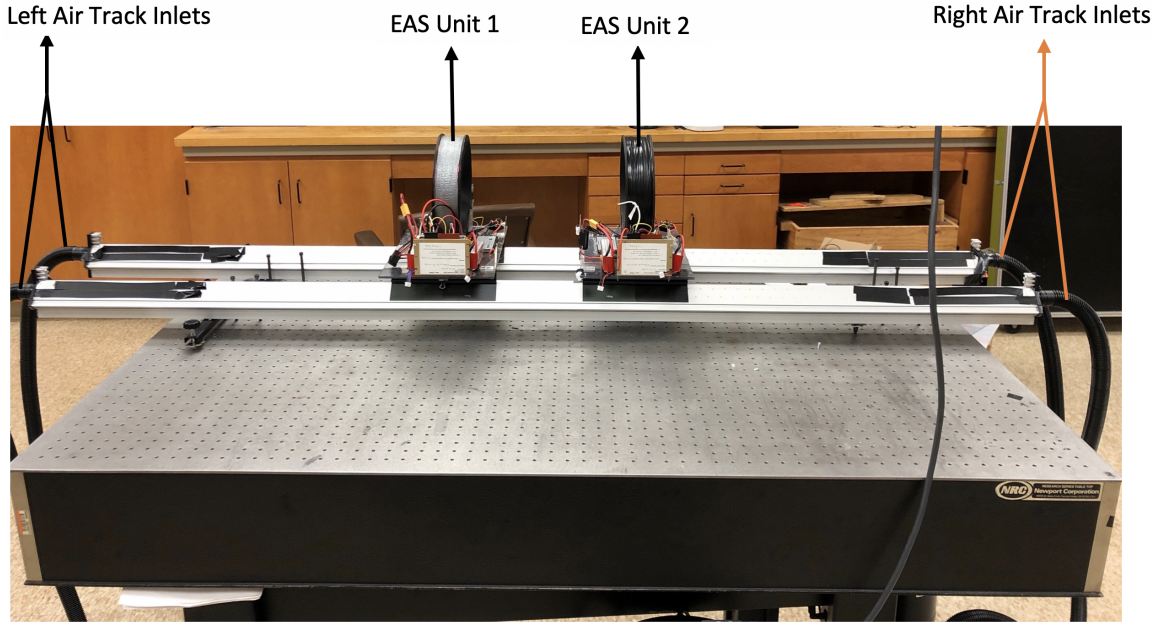


Figure 3.2: Side view of the testbed placed on an optical table. The two EAS sit on a low-friction linear air track.

EM coil, and direct current (DC) power source.

### 3.1.1 EAS Base Platform Design

The base platform is made of polycarbonate material because of its light weight. For rigidity, we use a double-layered polycarbonate sheet. The length, width, and thickness of the double-layered sheet are 0.356 m, 0.229 m, and 0.005 m, respectively.

Two air-track gliders are placed side by side on each linear airtrack, and the platform is mounted on the set of four airtrack gliders. We mount the electronics on the platform such that the center of gravity lies at the centroid of the sheet as shown in Figure 3.3.

### 3.1.2 EAS Power and Electronics

The power electronics of the EAS supply DC power, perform power conversion, compute the closed-loop sinusoidal control, and amplify the control signal to appropriate levels for the EM coil. The power electronics include: two lithium-ion polymer batteries, two isolated DC boost converter modules, two standard single-phase bridge

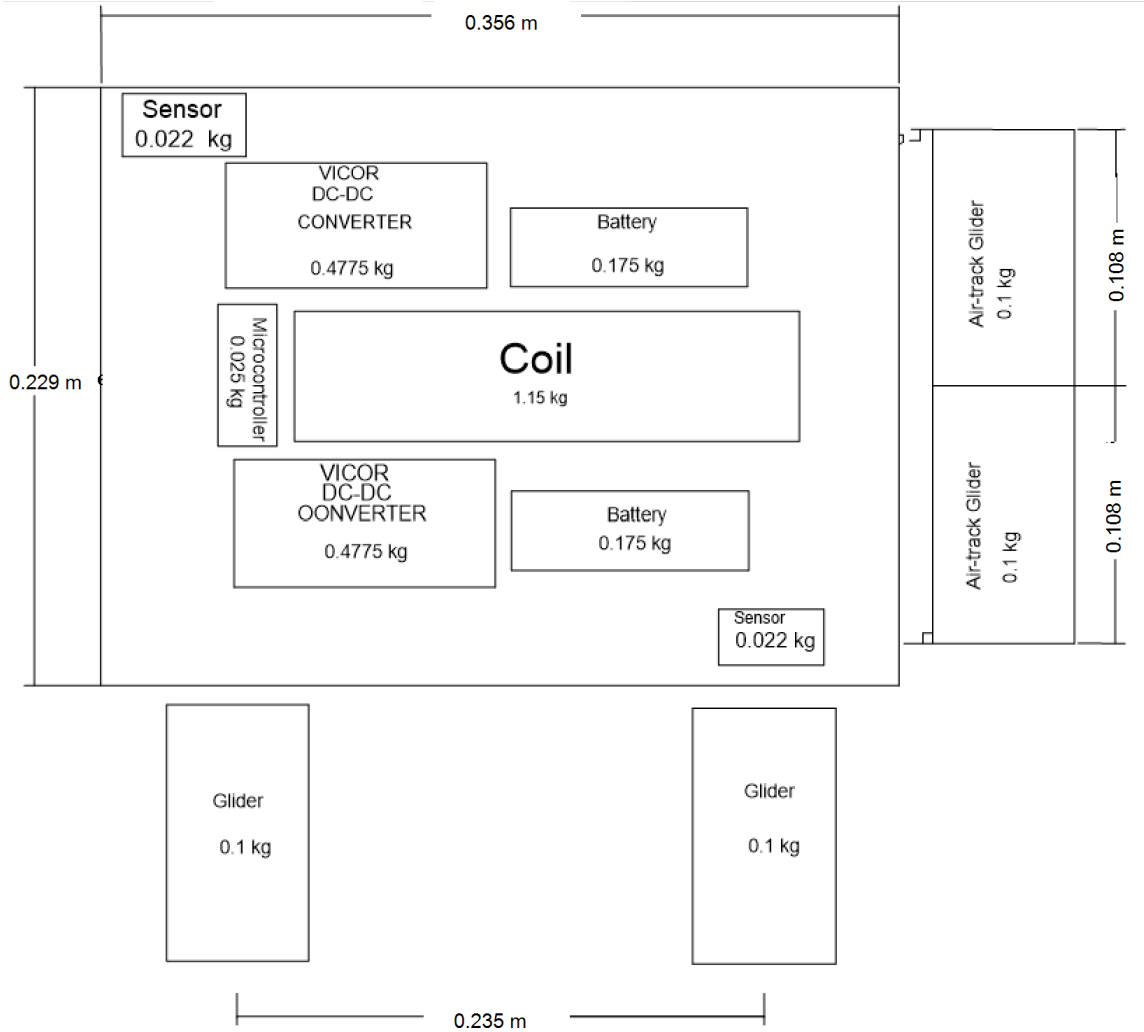


Figure 3.3: EAS platform design.

rectifier rated at 600 V and 50 A, an Arduino Due microcontroller, and a Copley Controls DC brush servo amplifier (4122Z) as shown in Figure 3.4.

These components mount to the double-layered polycarbonate sheet. We mount the lithium-ion polymer batteries to the top side of the sheet using velcro strips for easier removal to recharge. The remaining components screws into the platform at laser-cut perforations on the double-layered sheet. The Copley Controls DC brush servo amplifier amplifies a low-powered AC control signal from the digital to analog conversion (DAC) pins of Arduino to a high-current AC signal that is sent to the EM coils. To satisfy this functionality, we require a 70.5 V DC input to the amplifier.

We use the fixed high output voltage circuit configuration from VICOR Inc. and

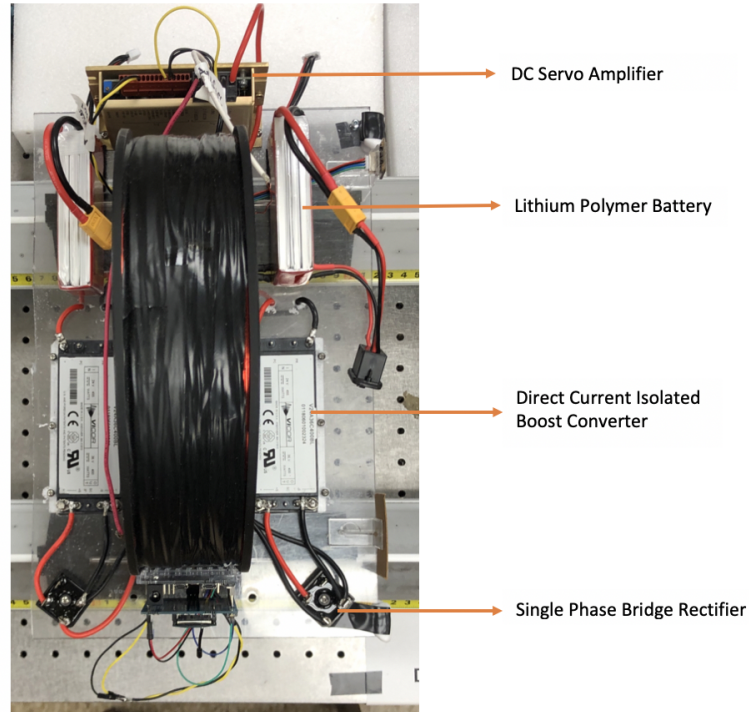


Figure 3.4: Top-down view of the EAS on the air track displaying the four main electrical components of the EAS.

customize the configuration to meet the input voltage requirements of the amplifier. We connect the battery packs and isolated DC boost converter modules in a series configuration, as shown in Figure 3.5. Note that the circuit shown in Figure 3.5 was partially designed by VICOR Inc. We connect and solder electrical wire connections of higher line-weight using a 14 AWG colored coded wires, while the remaining components use 18 AWG wires as shown in Figure 3.5.

In this series configuration, the OUT+ terminal is connected to SENSE+ terminal and OUT- terminal is connected to SENSE- terminals for each module. The output from these connections connects to two diodes in a single-phase bridge rectifier (MPS506-BPMS-ND). These rectifiers are rated at a voltage higher than the output voltage, and a backward current higher than the maximum load current. The diodes operate in forward-bias to ensure there is no reverse voltage drive from the amplifier to the converter modules during turn-off mode.

We initialize the power sequence to the EAS using a manual high current single

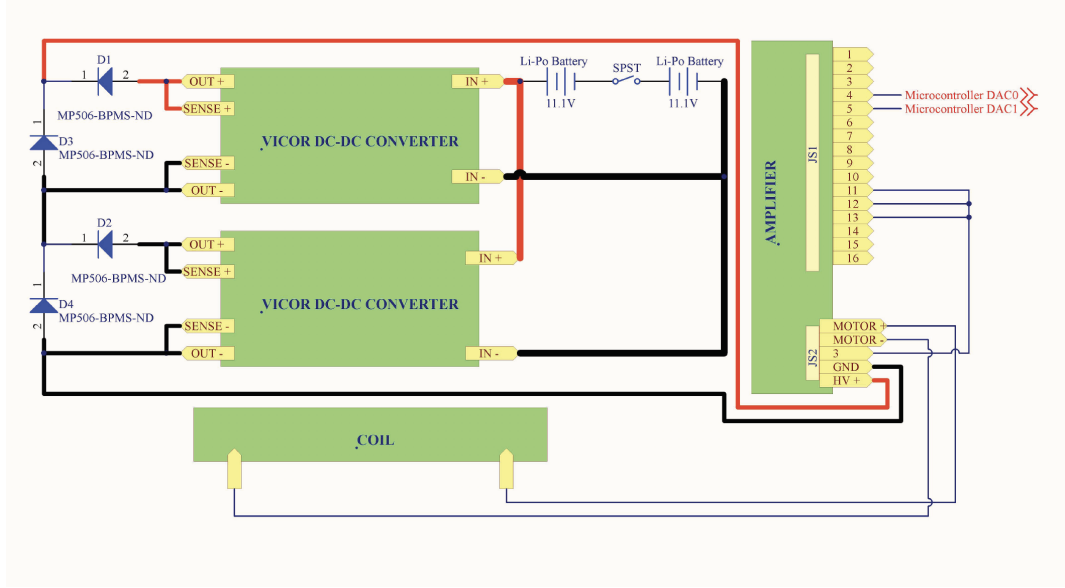


Figure 3.5: Electrical schematic of the EAS power electronics with red wires depicting positive input and output current and blue wires depicting negative input and output current.

pole single throw (SPST) switch. The circuit turns on with an initial voltage of 22.2 V DC from the fully charged batteries. We boost this low DC voltage to a high voltage of 70.5 V DC from the IN+ and IN- terminals to the OUT+ and OUT- terminals of the converter, respectively. After passing through the reverse protection circuit from the diodes, we connect the boosted 70.5 V DC output to HV+ and GND terminals of the amplifier as shown in Figure 3.5.

### 3.1.3 Coil Design

We design the coil by determining the magnetic force required to allow a frictionless motion of the EAS. First, we consider the EM coils as magnetic dipoles with magnetic moments generated by AC current.

Figure 3.6 shows the magnetic field  $B$  and the current  $I$  that pass through the coils. Figure 3.6 shows how each set of coils can be considered as a magnetic dipole and how they are similar to the dipole moment of a bar magnet that attracts or repels.

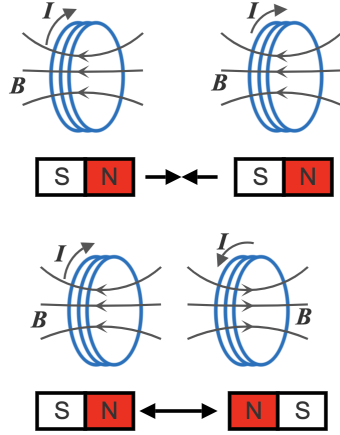


Figure 3.6: Electromagnetic coils with attractive and repulsive magnetic field lines that represent the dipole interaction of a bar magnet.

The force exerted by a magnetic dipole can be approximated by

$$F = -\nabla(uB) \quad (3.1)$$

where  $u$  is the magnetic moment and  $B$  is the magnetic field. The magnetic moment  $u$  of a dipole generated by an EM coil of  $N$  turns is defined as

$$u = NIA \quad (3.2)$$

where  $A = \pi R^2$  is the cross-sectional area of the EM coil of radius  $R$ . The magnetic field  $B$  of an EM coil is given by

$$B = -\frac{3\mu_0}{2\pi|x|^4}u^2 \quad (3.3)$$

We use this expression to calculate the force exerted on one EM coil by the other EM coil. The force  $F$  exerted on one EM coil of the EAS with weight  $m$ , due to the other, separated by a distance  $x$  is given by

$$m\ddot{x} = F \quad (3.4)$$

where  $F$  is defined

$$F \triangleq -\frac{3\mu_0}{2\pi|x|^4}N^2A^2I^2, \quad (3.5)$$

where  $\mu_0$ , is the magnetic constant,  $N$  is the number of turns, and  $A$  is the cross-sectional area. We use (3.5) to calculate intersatellite force on coils for different values of  $N$  and  $R$ , at different distance of separation  $x$  ranging from 0.1 m to 0.9 m.

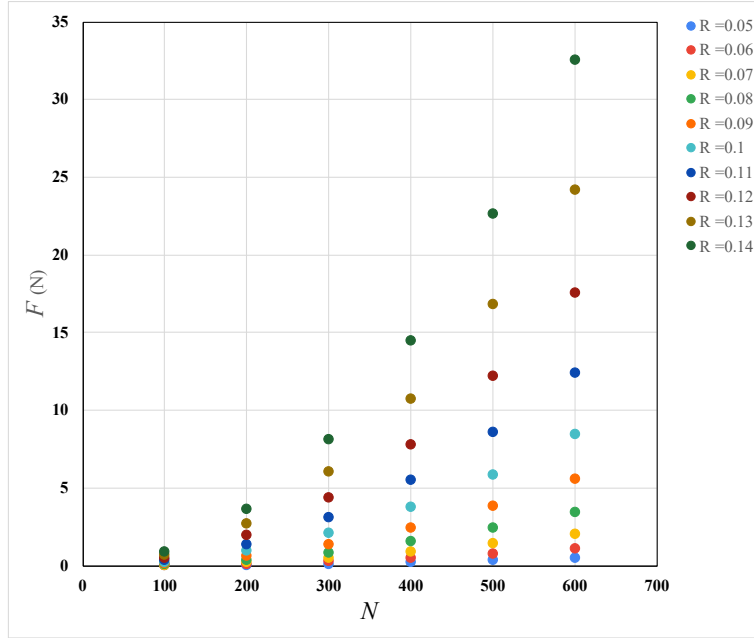


Figure 3.7: Force exerted by the EM coil at  $x = 0.1$  m for different radius of coil in meters represented using different colors in the legend.

We first conduct a weight test to determine the maximum weight the two linear airtracks can withstand. We obtain a result of 2.75 kg. The total weight of the power electronics is 1.6 kg. Thus, the remaining allowance is 1.15 kg. Next, we design a coil for the weight constraint of 1.15 kg with different radius, number of turns  $N$  and force  $F$  exerted by the EM coil. We examine  $F$  at  $x = 0.1$  m,  $x = 0.5$  m and  $x = 0.9$  m as shown in Figure 3.7–3.9 respectively for different radius.

For a set number of turns  $N$  of the current carrying conductor, the current flowing through the EM coil is limited due to the inductance of the EM coil. Increase in  $N$  leads to increase in inductance of the EM coil, thus reducing the maximum AC that

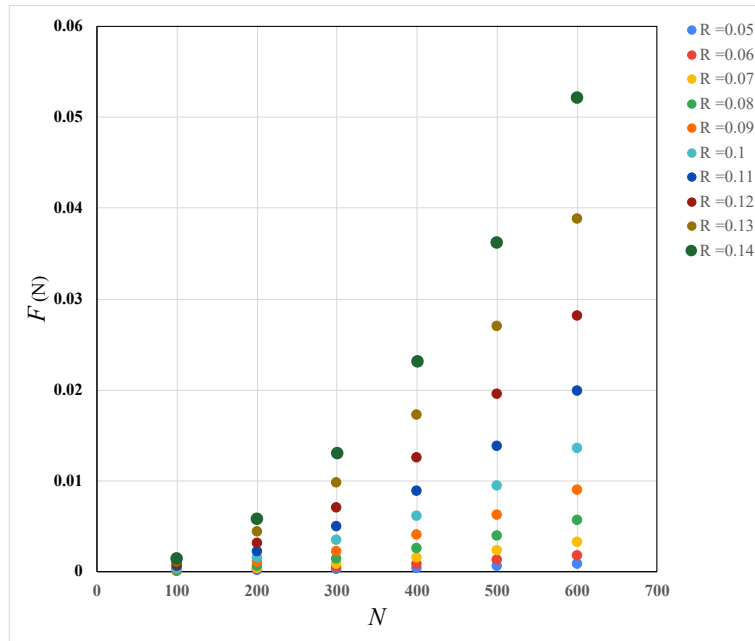


Figure 3.8: Force exerted by the EM coil at  $x = 0.5$  m for different radius of coil in meters represented using different colors in the legend.

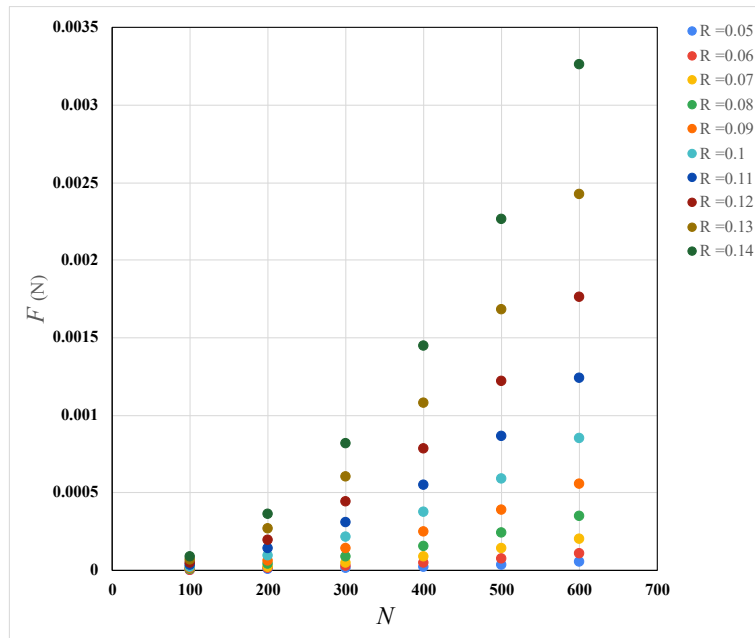


Figure 3.9: Force exerted by the EM coil at  $x = 0.9$  m for different radius of coil in meters represented using different colors in the legend.



can be sent to the EM coil. The magnetic moment  $u$  that generates the EM force  $F$  is also dependent on  $N$  of the EM coil. Thus, there exist a trade off between maximum AC, magnetic force  $u$  generated by the EM coil, and the weight constraint  $m$  in order to choose the optimum value of  $N$  and  $R$  to design the EM coil. Additionally, the width of the base platform is 0.356 m. This limits  $R$  to 0.178 m.

We choose  $N = 500$  and  $R = 0.1$  as the design for this thesis, which satisfies the remaining 1.15 kg allowance of weight. The calculated and measured inductance for this design is 71 mH and allows an AC of 2.5 A. An AC of 2.5 A generates sufficient magnetic force  $F$  for  $x = 0.1$  m,  $x = 0.5$  m, and  $x = 0.9$  m to initiate a translational motion.

### 3.2 EAS Sensing and Control

The EAS sensing and control system includes an Arduino Due microcontroller, Arduino wireless secure digital (SD) shield, XBee 802.15.4, and a STM VL53L0X laser-ranging module. The Atmel SAM38XE ARM Cortex-M3 chip on the microcontroller is the onboard processor that produces the sinusoidal control signal. The XBee radio frequency (RF) module that connects to the wireless SD shield is responsible for the communication between the EASs and a computer. The wireless SD shield between the microcontroller and the XBee RF module, shown in Figure 3.10 facilitates radio communication and data logging. The STM VL53L0X and the white reflector shown in Figure 3.10b are the sensing devices that provides measurements of relative position between the EAS.

We retrieve relative position data from the sensor modules and directly log this data to an onboard SD card present on the wireless SD shield. The software uploaded to the microcontroller includes programs `clsat1.ino` and `clsat2.ino`. These two programs generates a low voltage sinusoidal signal from DAC0 and DAC1 pins of the microcontroller to pins 4 and 5 of the servo amplifier respectively as shown in Figure 3.5. The retrieved relative position data and the low voltage sinusoidal signal are used simultaneously to generate a feedback control signal. We write a function for the closed-loop control algorithm described in Chapter 2 in each of the programs

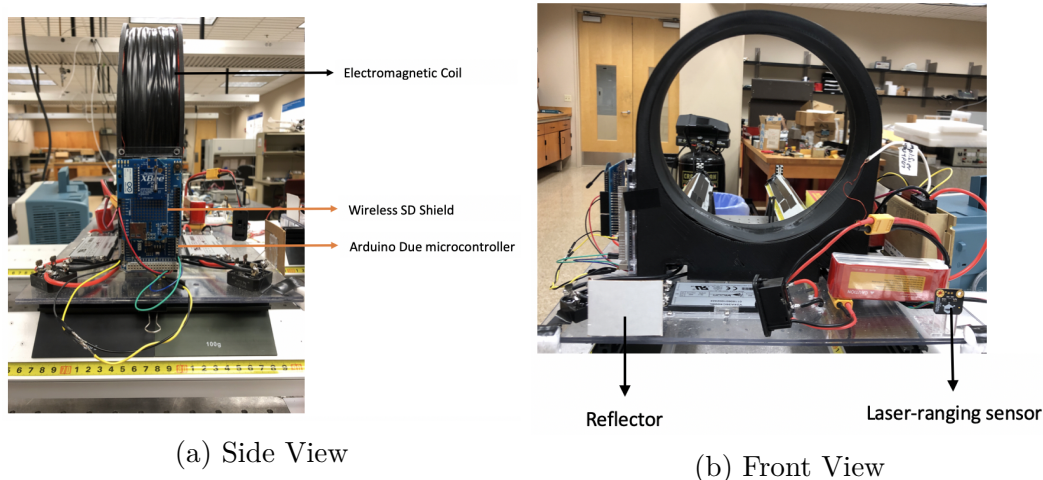


Figure 3.10: Side view and front view of the EAS on the air track displaying the Arduino Due microcontroller, the wireless SD shield, reflector, and laser-ranging sensor.

clsat1.ino and clsat2.ino that generates the feedback control signal. This feedback signal is sent to the amplifier.

The following section describes the onboard software and the radio network configuration for the open-loop and closed-loop controls experiment discussed and observed in Chapter 4.

### 3.2.1 EAS Software

We use the open-source Arduino integrated development environment (IDE) 1.8.10 to program the closed-loop control algorithm in Chapter 2. Arduino IDE 1.8.10 is built using Java and other open-source environments. The IDE supports languages such as C and C++ with special rules of structuring.

Figure 3.11 shows the flowchart for the program that runs all the required tasks in a single continuous loop with a setup function that predefines the control parameters before the experiment.

### 3.2.2 Data Acquisition

We use floating-point arrays as data structures in the program to store sensor information for feedback control. The program stores parameters such as relative

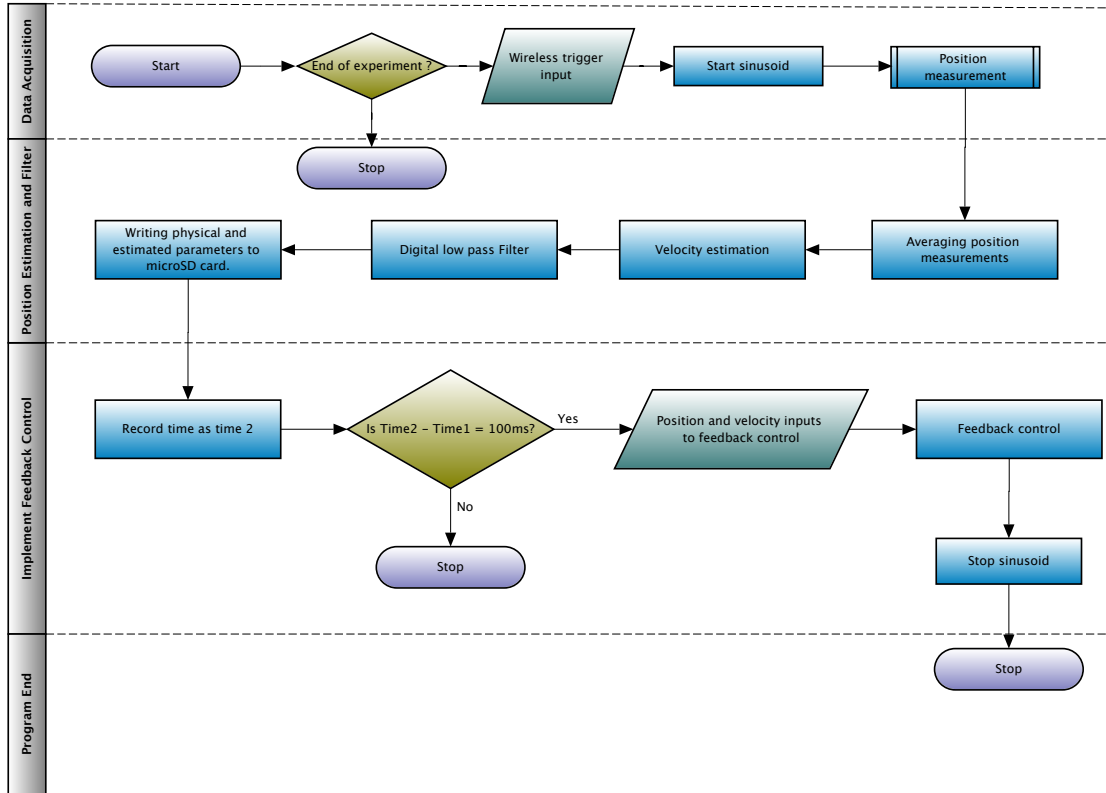


Figure 3.11: Software flowchart.

position, estimated velocity, and feedback amplitude in separate data structures.

The VL53L0X laser-ranging module obtains relative position measurements of EAS. The VL53L0X sensor uses a 940 nm vertical cavity surface emitting laser using a single-photon avalanche diode to obtain accurate distance measurements. This new generation of sensors allows us to capture such measurements with high immunity to ambient light and low crosstalk issues.

The acquisition process begins with the simultaneous trigger input from a personal computer connected with an XBee RF module. We connect the XBee network in the form of a point-to-multipoint communication network from the computer to the EAS. The sensor transmits these position measurements to the Arduino Due module using the I<sup>2</sup>C protocol. We retrieve these measurements from a function called `sensor.getDistance()` predefined in the libraries of the STM VL53L0X. We define a

function called `sensorDistRead()` that acquires the relative position measurements from the `sensor.getDistance()` function and performs an average of seven position measurements to obtain a better approximation of relative distance measurement.

### 3.2.3 Velocity Estimation

In this section, we describe the backward-Euler approximation and low-pass filter techniques used to better estimate the relative velocity of the EAS.

Let  $T_s = 0.1$  s be the update period for the amplitude of the sinusoidal control, and let  $\Delta T \triangleq \frac{T_s}{7} \approx 0.0143$  s be the sample time for the position data obtained from the STM VL53L0X sensor. For each  $n \in \mathbb{N}$ , let  $x(n\Delta T)$  be the relative position measured from the STM VL53L0X sensor at time  $n\Delta T$ . Next, we average 7 position measurements taken over each update interval to obtain

$$x_{\text{av}}(kT_s) \triangleq \frac{1}{7} \sum_{i=0}^6 x((k-1)T_s + i\Delta T), \quad (3.6)$$

which is the relative position at time  $kT_s$  and is used for feedback.

The STM VL53L0X sensor does not provide relative velocity measurements; however, the feedback control requires an estimate of the relative velocity. We use the relative positions  $x_{\text{av}}(kT_s)$  and  $x_{\text{av}}((k-1)T_s)$  to obtain a backward-Euler estimate of relative velocity, and we pass this estimate through a low-pass filter to obtain the relative velocity estimate used for feedback. More specifically, the relative velocity estimate at time  $kT_s$  is given by,

$$v_e(kT_s) = av_e((k-1)T_s) + (1-a) \left[ \frac{x_{\text{av}}(kT_s) - x_{\text{av}}((k-1)T_s)}{T_s} \right], \quad (3.7)$$

where  $a \in [0, 1]$ . For the experiments in this thesis, we select  $a = 0.97$ .

### 3.2.4 Sinusoidal Signal Generation

In this section, we describe the process of sinusoidal signal generation, which generates the necessary AC sinusoids for the control experiment. The program `clsat1.ino` and `clsat2.ino` generates the sinusoidal signal from the Arduino Due microcontroller by digital to analog conversion of the digital signal. We generate the digital sinusoids from a predefined program and its library, `SineWaveDuemaster.cpp` and `SineWaveDue.h`, respectively [37]. This digital sinusoidal signal uses the DAC interface of the embedded system inside ARM processor of the microcontroller to generate an analog signal.

Initially, we trigger the digital sinusoidal generation using the coordinator XBee radio module connected to the personal computer. Once the trigger is applied, wireless data packets received on the EAS from the personal computer starts an amplitude updating sinusoidal function. This function calls the predefined function that generates the sinusoids from `SineWaveDuemaster.cpp`, which generates a recursive sine wave output to the main program in Arduino IDE. The DAC of Arduino Due generates the recursive sinusoidal signal using the unit step function.

We use volatile floating-point registers to store the digital value of the sinusoid. As the main loop in the Arduino program starts running, the predefined function generates the sinusoids recursively using a timer interrupt. This process repeats until the end of the experiment.

### 3.2.5 XBee Radio Communication

The XBee radio modules mounted in the electronics of the two EAS is the primary key to trigger the program wirelessly from a personal computer. We use a Digi XBee 1 mW wire antenna series 1 (802.15.4) radio module on each system to connect the two units and the computer in a peer to peer network wirelessly. The control experiments require the two units to trigger the control simultaneously. We use a multi-point network topology, with a broadcast transmission protocol to satisfy this requirement for the experiment.

We program the XBee modules in a multi-point network such that they associate in a coordinator and end device fashion. The coordinator XBee module transmits the data packets to the end device XBee module mounted on Arduino Due microcontroller to trigger a simultaneous serial communication to the two EAS. This trigger is what enables the Arduino program that runs the experiment. Figure 3.12 depicts the radio network configuration used for this experiment.

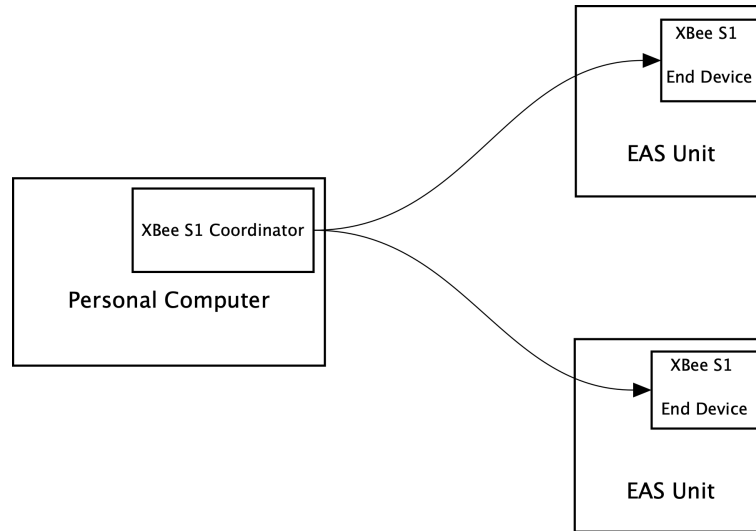


Figure 3.12: Block diagram that shows the network configuration of the experiment.

The overall communication process involves two types of communication, serial communication, and wireless communication. Serial communication between the personal computer and the coordinator radio module enables wireless communication. The wireless communication simultaneously enables the serial communication between the end device XBee and the EAS. The program interrupt receives the serial data that triggers the main software loop.

# Chapter 4

---

## Experimental Results and Discussion

In this chapter, we present the results for open-loop and closed-loop control experiments using the EAS introduced in Chapter 3. In all of the closed-loop experiments, feedback is provided by the laser range-finding sensor (STM VL53L0X). These sensors provide relative position feedback at 70 Hz, and we average every 7 measurements using (3.6) to obtain the relative-position feedback  $r_{ij}$  used by the control algorithm. The relative-velocity feedback  $\dot{r}_{ij}$  is estimated using the backward-Euler low-pass filter (3.7).

We conduct 6 experiments. Two experiments demonstrate open-loop control using 2 identical EAS, and the remaining four experiments demonstrate closed-loop control. The two open-loop experiments demonstrate the feasibility of data acquisition and the translational motion of the EAS. For all closed-loop experiments, the desired relative position  $d_{ij}$  remains the same, but we use different initial conditions.

For these experiments, the EAS translates from an initial position marked on the scale attached to the linear air track. We place the EAS at their respective initial positions on a perfectly leveled air track. Next, we upload the software to the microcontroller. Next, we switch on the quiet air source and regulate the air pressure to maintain a stationary position for the EAS. We then switch on the DC supply from battery packs to the electrical system. The microcontroller is powered using a 9 V rechargeable lithium-ion battery. Upon supplying power to the microcontroller,

they wait on a standby mode to receive a command signal from the host computer. Once the computer triggers the command signal, the EM coils are energized with a sinusoidal current. Finally, at end of the experiment, the software stops data acquisition, data logging, and control signal. We then turn off the power for all systems.

The following 2 experiments demonstrate open-loop control, where each EAS coil is actuated with a 10 Hz sinusoidal current. These experiments demonstrate that the experimental testbed is capable of generating attractive and repulsive forces that move the EAS.

#### 4.1 Open-Loop Experiments

**Experiment 4.1.** This experiment demonstrates open-loop repulsion. The initial relative position between the units is  $r_{ij}(0) = 0.25$  m. The EM coils are actuated by sinusoidal currents, which are out of phase by  $180^\circ$ . The currents' frequency is 10 Hz and the currents' amplitude is 1.25 A.

Figure 4.1 shows relative positions and velocities measured from the sensors for each EAS units. At  $t = 0$  s, the control is turned on leading to repulsion of the units. At  $t = 10$  s, the relative position is approximately  $r_{ij}(10) = 0.53$  m. As seen, the relative position and velocity measurements for each EAS are not identical. The STM VL53L0X laser ranging sensor has an accuracy of  $\pm 3\%$ , which helps to explain the difference between the measurements.  $\triangle$

**Experiment 4.2.** This experiment demonstrates open-loop attraction. The initial relative position is  $r_{ij}(0) = 0.45$  m. The EM coils are actuated by sinusoidal currents which are in phase and have an amplitude of 1.25 A. The currents' frequency is 10 Hz.

Figure 4.2 shows relative position and velocity of the two EAS units. At  $t = 0$  s, the control is turned on resulting in the attraction of the units. At  $t = 10$  s, the relative position is approximately  $r_{ij}(10) = 0.275$  m.  $\triangle$



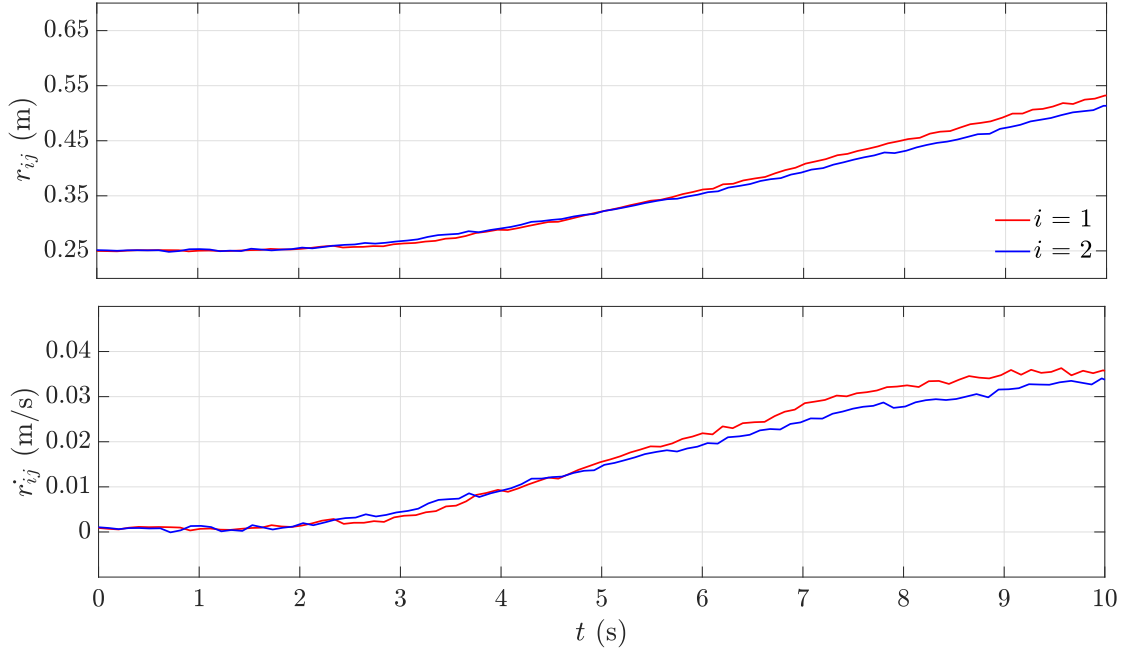


Figure 4.1: Open-loop control, where each EAS coil is actuated with a 10 Hz sinusoidal current with amplitude 1.25 A. The sinusoidal currents on the EAS are  $180^\circ$  out of phase, which results in a repulsion force.

The next 4 experiments demonstrate closed-loop control for several different initial conditions. For all closed-loop experiments, the control is given by (2.6), (2.10), and (2.11), where  $\alpha_{ij} = 900$ ,  $\beta_{ij} = 6882$ , and  $d_{ij} = 0.35$  m. We also let  $\omega_{ij} = 20\pi$  rad/s (or 10 Hz) for all experiments.

## 4.2 Closed-Loop Experiments

**Experiment 4.3.** In this experiment, the initial relative position between the units is  $r_{ij}(0) = 0.25$  m. Figure 4.3 shows the relative position  $r_{ij}$ , relative velocity  $\dot{r}_{ij}$ , control amplitude  $a_{ij}$ , and the piecewise-sinusoidal control  $X_i$ .

We turn on the control at  $t = 0$  s. The EAS initially repel one another. They accelerate towards each other between  $t = 0$  and  $t = 7.5$  s. The relative position  $r_{ij}$  overshoots the desired relative position  $d_{ij} = 0.35$  m at  $t = 7.5$  s, reaches a maximum of approximately 0.41 m. Note that  $a_{12}$  is always positive, whereas  $a_{21}$

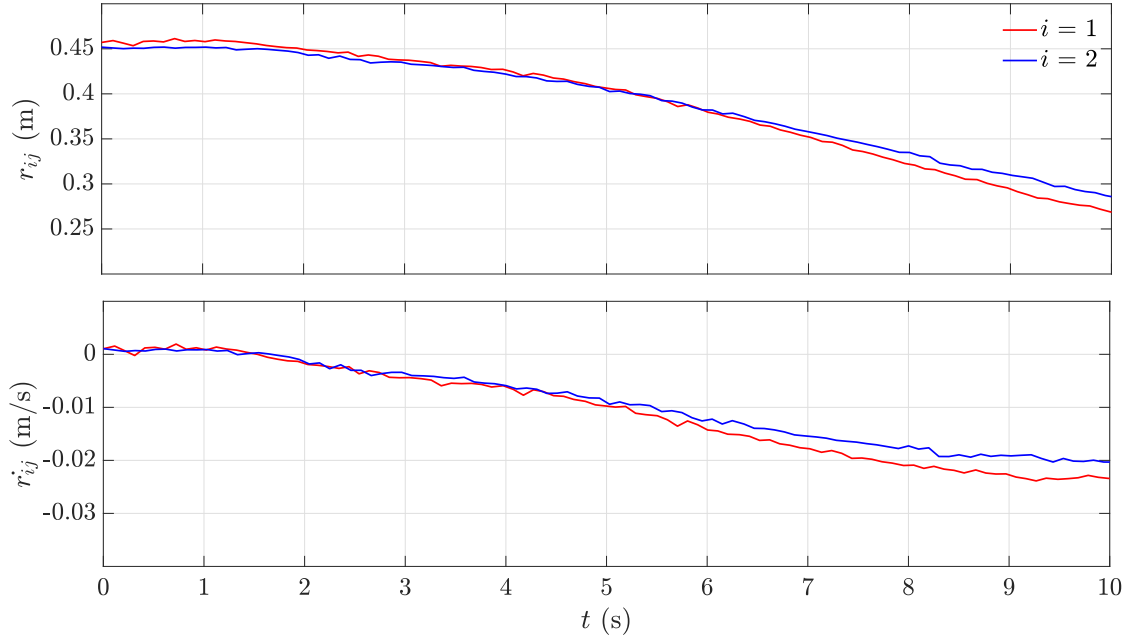


Figure 4.2: Open-loop control, where each EAS coil is actuated with a 10 Hz sinusoidal current with amplitude 1.25 A. The sinusoidal currents on the EAS are  $180^\circ$  out of phase, which results in a repulsion force.

changes its sign at  $t = 5$  s. This indicates a change in feedback, which attracts EAS towards each other. Note that  $a_{12}$  reaches a maximum of approximately 2 A when  $r_{ij}$  is approximately 0.41 m. At  $t = 17$  s,  $a_{21}$  changes its sign again indicating repulsive force. The EAS approximately converge to the desired relative position  $d_{ij}$  by  $t = 20$  s. Between  $t = 20$  s and  $t = 50$  s,  $r_{ij}$  is equal to  $d_{ij}$ ,  $\dot{r}_{ij}$  is approximately 0 m/s, and  $a_{12}$  is approximately 100 mA, where as  $a_{21}$  switches between  $-100$  mA and 100 mA.  $\triangle$

**Experiment 4.4.** In this experiment, the initial relative position between the units is increased to  $r_{ij}(0) = 0.45$  m. Figure 4.4 shows  $r_{ij}$ ,  $\dot{r}_{ij}$ ,  $a_{ij}$ , and  $X_i$ .

We turn on the control at  $t = 0$  s. The EAS initially attract one another. They accelerate towards each other between  $t = 0$  and  $t = 7.5$  s. The relative position  $r_{ij}$  overshoots the desired relative position  $d_{ij} = 0.35$  m at  $t = 7.5$  s, reaches a minimum of approximately 0.30 m. Note that  $a_{12}$  and  $a_{21}$  are positive between  $t = 0$  and  $t = 5$  s, where as at  $t = 5$  s,  $a_{21}$  changes its sign. Note that  $a_{12}$  reaches a maximum

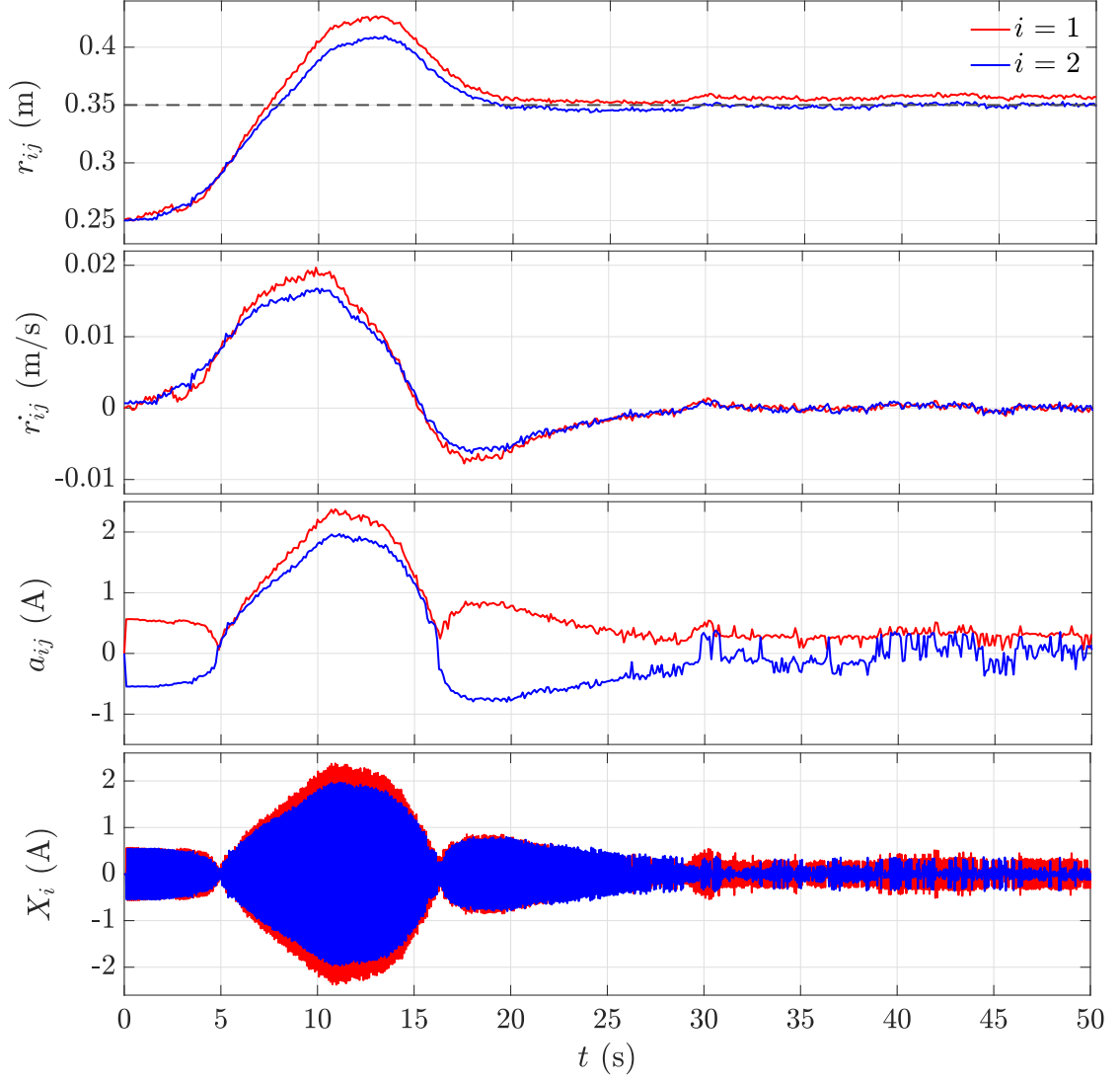


Figure 4.3: Closed-loop formation control, where  $r_{ij}(0) = 0.25$  m and  $d_{ij} = 0.35$  m.

of approximately 1 A when  $r_{ij}$  is approximately 0.30 m. This indicates a change in feedback, which repels EAS from each other. At  $t = 17$  s,  $a_{21}$  changes its sign again to positive indicating an attractive force between EAS. The EAS approximately converge to the desired relative position  $d_{ij}$  by  $t = 20$  s. Between  $t = 20$  s and  $t = 50$  s,  $r_{ij}$  is equal to  $d_{ij}$ ,  $\dot{r}_{ij}$  is approximately 0 m/s, and  $a_{12}$  is positive, whereas  $a_{21}$  switches between positive and negative.  $\triangle$

The next 2 experiments examine the inter-EAS behavior for the case where the initial relative position is closer to the desired relative position (i.e.,  $\pm 5$  cm as opposed

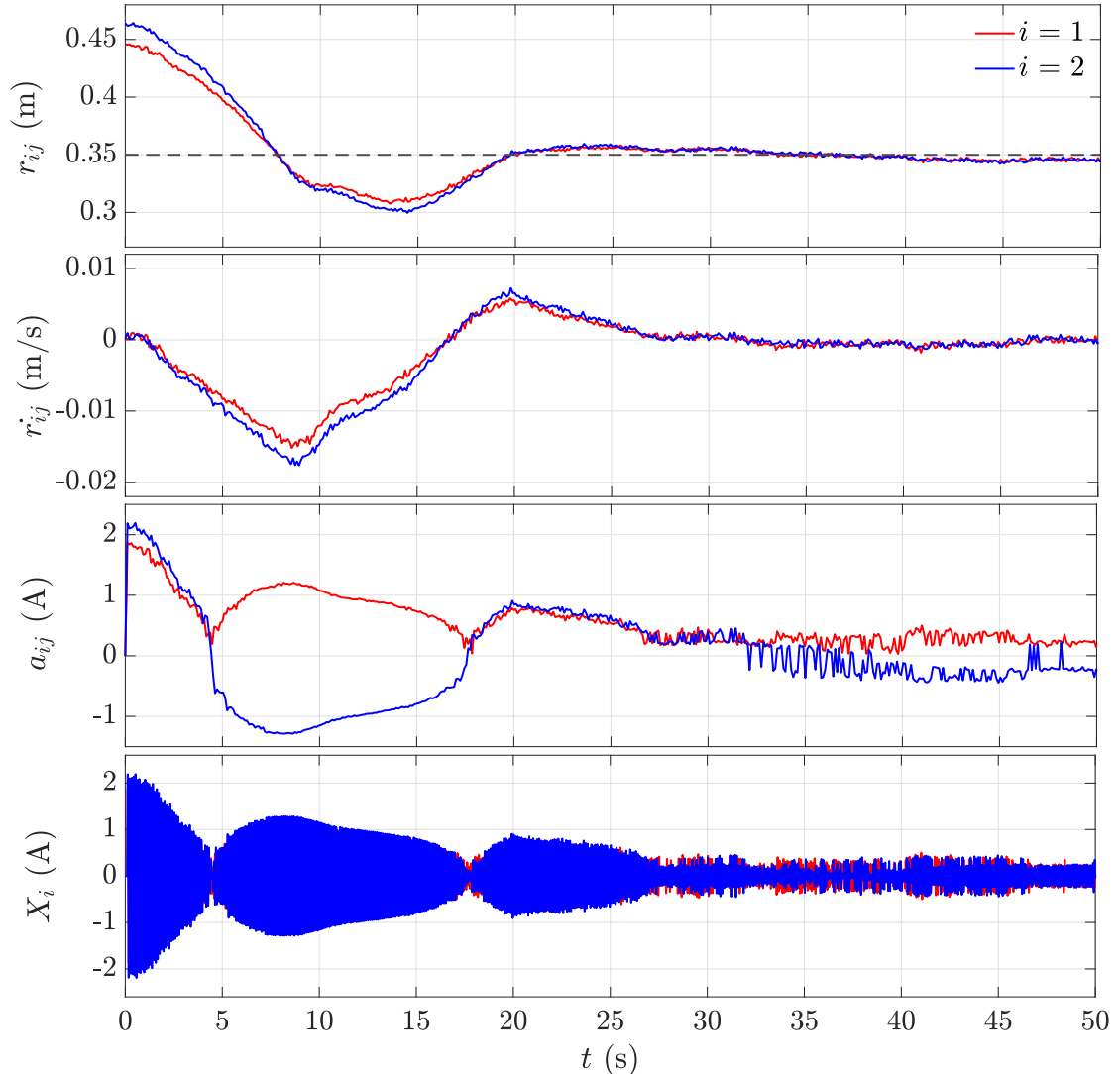


Figure 4.4: Closed-loop formation control, where  $r_{ij}(0) = 0.45$  m and  $d_{ij} = 0.35$  m.

to the  $\pm 10$  cm initial error considered in Experiments 4.3 and 4.4).

**Experiment 4.5.** In this experiment, the initial relative position between the units is  $r_{ij}(0) = 0.30$  m, which is closer to the desired relative position. Figure 4.5 shows  $r_{ij}$ ,  $\dot{r}_{ij}$ ,  $a_{ij}$ , and  $X_i$ .

We turn on the control at  $t = 0$  s. The EAS initially repel one another. They accelerate away from each other between  $t = 0$  and  $t = 7.5$  s. The relative position  $r_{ij}$  overshoots the desired relative position  $d_{ij} = 0.35$  m at  $t = 7.5$  s and reaches a maximum of approximately 0.39 m. Note that  $a_{12}$  is always positive, where as

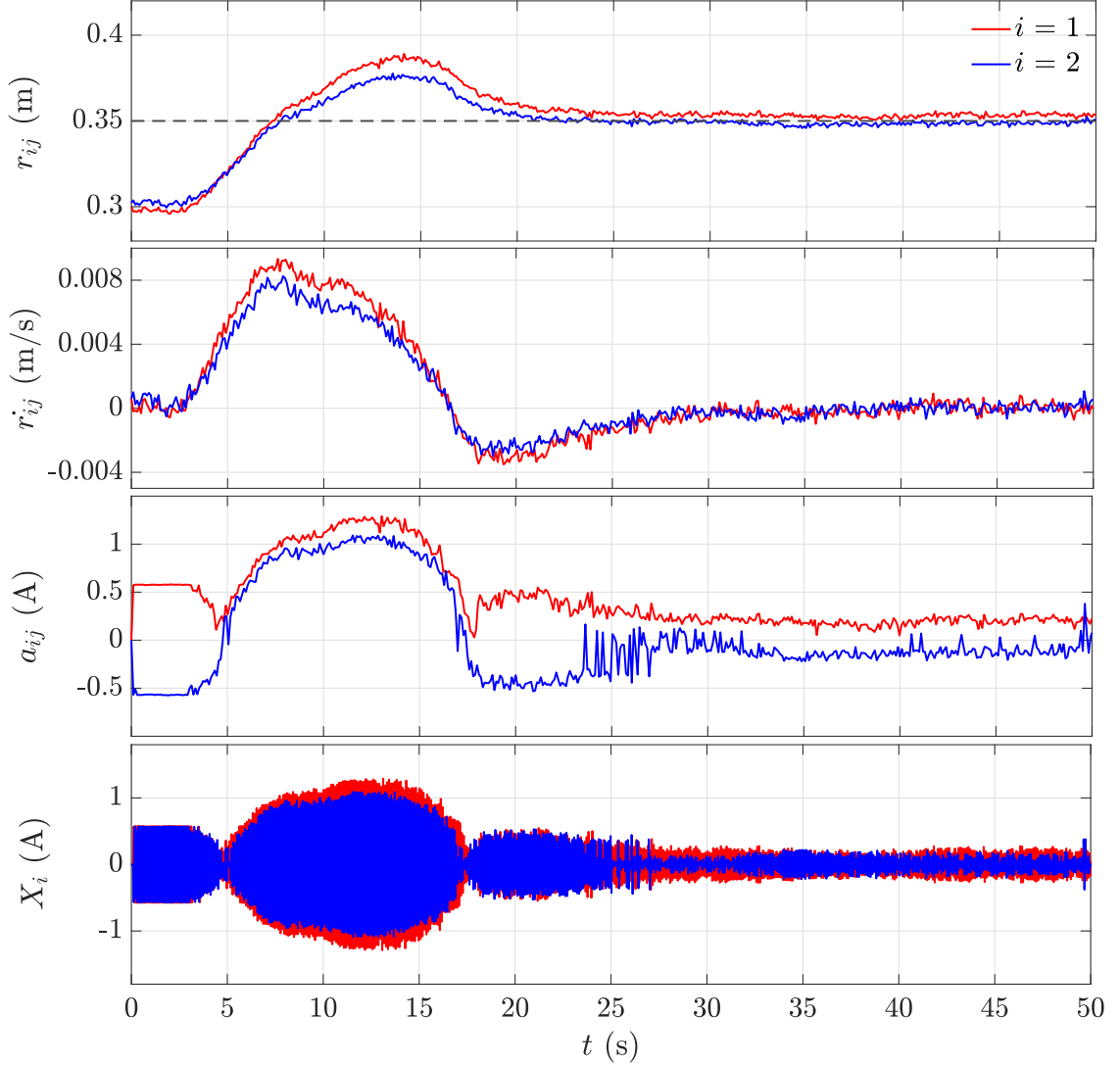


Figure 4.5: Closed-loop formation control, where  $r_{ij}(0) = 0.30$  m and  $d_{ij} = 0.35$  m.

$a_{21}$  is initially negative but changes to positive at  $t = 5$  s. This indicates a change in feedback, which attracts the EAS towards each other. Note that  $a_{12}$  reaches a maximum of approximately 1 A when  $r_{ij}$  is approximately 0.39 m. At  $t = 17$  s,  $a_{21}$  changes its sign again indicating a repulsive force. The EAS approximately converge to the desired relative position  $d_{ij}$  by  $t = 20$  s. Between  $t = 20$  s and  $t = 50$  s,  $r_{ij}$  is equal to  $d_{ij}$ ,  $\dot{r}_{ij}$  is approximately 0 m/s. Note that  $a_{12}$  reduces from 500 mA to 100 mA and maintains this value, whereas  $a_{12}$  switches its sign.  $\triangle$

**Experiment 4.6.** In this experiment, the initial relative position between the

units is  $r_{ij}(0) = 0.40$  m, which has +5 cm initial error. Figure 4.6 shows  $r_{ij}$ ,  $\dot{r}_{ij}$ ,  $a_{ij}$ , and  $X_i$ .

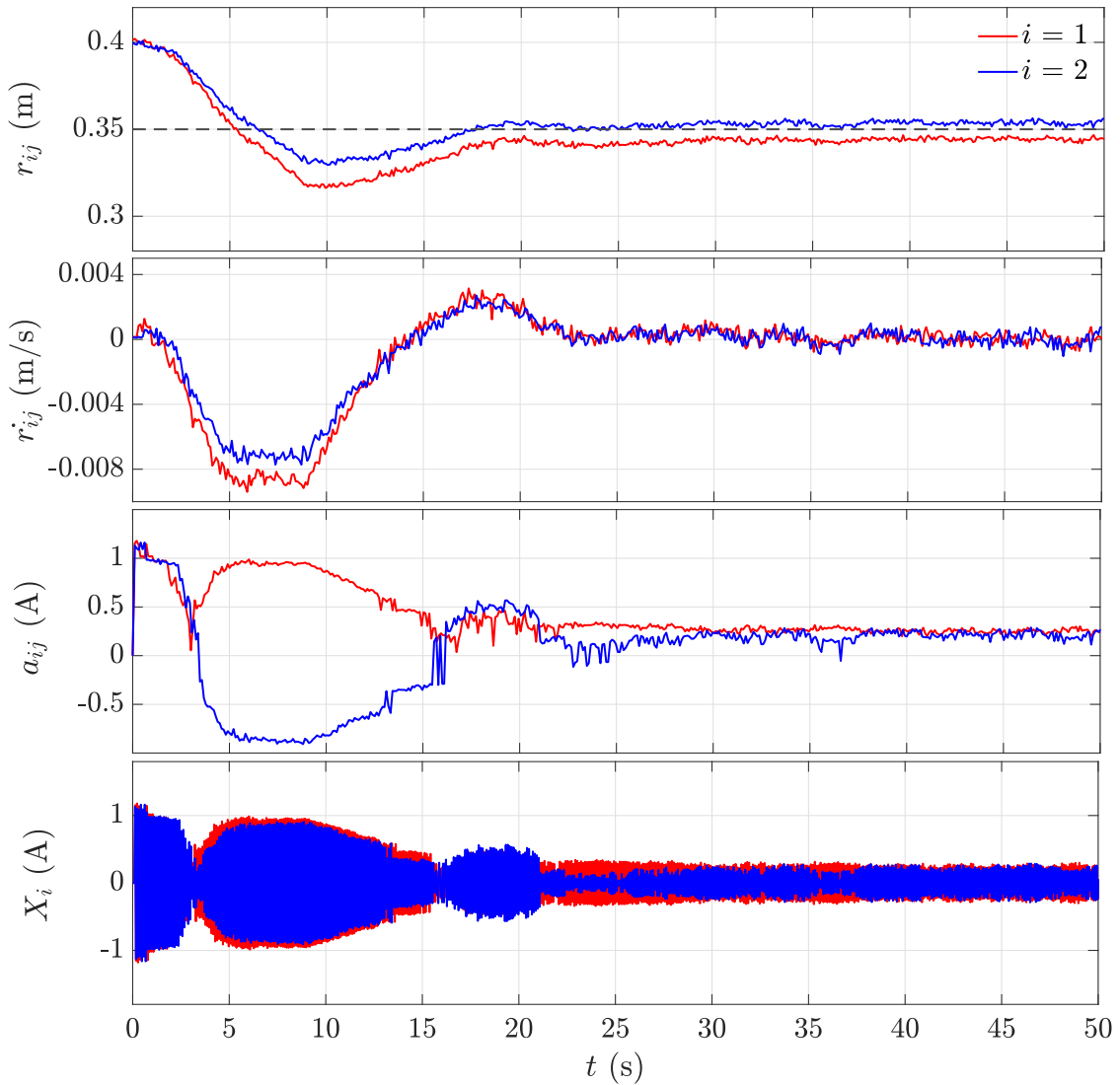


Figure 4.6: Closed-loop formation control, where  $r_{ij}(0) = 0.40$  m and  $d_{ij} = 0.35$  m.

We turn on the control at  $t = 0$  s. The EAS initially attract one another. The relative position  $r_{ij}$  overshoots the desired relative position  $d_{ij} = 0.35$  m at  $t = 5$  s, and reaches a minimum of approximately 0.32 m. Note that  $a_{12}$  and  $a_{21}$  are positive between  $t = 0$  and  $t = 2.5$  s, where as at  $t = 2.5$  s,  $a_{21}$  changes its sign. This indicates a change in feedback, which repels the EAS away from each other. Note that  $a_{12}$  approaches a maximum of approximately 1 A when  $r_{ij}$  is approximately

0.32 m. At  $t = 16$  s,  $a_{21}$  changes its sign again to positive indicating an attractive force between the EAS. The EAS approximately converge to the desired relative position  $d_{ij}$  by  $t = 20$  s. Between  $t = 20$  s and  $t = 50$  s,  $r_{ij}$  is equal to  $d_{ij}$ ,  $\dot{r}_{ij}$  is approximately 0 m/s, and  $a_{12}$  is positive, where as  $a_{21}$  switches between positive and negative.  $\triangle$

### 4.3 Discussion of Experimental Results

In the open-loop experiments, we tested the functionality and capability of the EAS to generate an attractive and repulsive force. In the closed-loop experiments, we demonstrated the closed-loop control introduced in Chapter 2. Initially, we conducted the open-loop experiments where we repelled the two EAS with an initial relative position  $r_{ij}(0)$  less than the desired relative position  $d_{ij}$  on the linear air track. In Experiment 4.2, we attracted the two EAS from an initial relative position  $r_{ij}(0)$  greater than the desired relative position  $d_{ij}$  on the linear air track. In Experiment 4.3 and Experiment 4.4 we introduce the results of the closed-loop control using an initial error of  $\pm 10$  cm between  $r_{ij}(0)$  and  $d_{ij}$ . In Experiment 4.5 and Experiment 4.6, we introduce the results of the closed-loop control using an initial error of  $\pm 5$  cm between  $r_{ij}(0)$  and  $d_{ij}$ . In all of the closed-loop experiments, we conduct the experiments using the gain parameters of the control  $u_{ij}$  tuned prior to the control experiment. For all of the experiments, we notice that measurements and estimations for each EAS are not identical due to the inaccuracy of the sensor, but the inaccuracy in measurements did not cause the EAS to produce results with significant bias. We show the results of estimated filtered velocity  $\dot{r}_{ij}$  for all experiments, but with a significant delay. Finally, the amplitude of the control and generated sinusoidal signal depicts expected but inaccurate results. The following discussion below explains the cause of such errors and the plan of action that can be taken to improve future work for reducing measurement and estimation errors.

As pointed out in Chapter 3 the estimated filtered velocity  $\dot{r}_{ij}$  is computed using the backward-Euler method and low-pass filter with an exponential smoothing factor, The input to the backward-euler function is an averaged value of relative position

measurements. Our averaged value from the sensors has a ranging accuracy issue, which may likely cause the inaccuracies in the estimated velocity, such as a noisy velocity. This implies that our filtered estimated velocity is also imperfect. In addition to this, we are also not compensating for the delay and distortions caused by the low-pass filter. To remediate and fix all the issues mentioned above for future work, we could implement a few solutions. For example, we can use a faster micro-controller with a better software strategy to perform computations in concurrence with the EM actuation, unlike the Arduino micro-controller, which performs the tasks and commands in a loop sequence. We could use a shorter-range sensor with no ranging accuracy issues or use an offboard vision system to track the relative position and velocity of the EAS. Implementing the above two strategies could also remove the need for the introduction of a low-pass filter for velocity estimation.



# Chapter 5

---

## Conclusions and Future Work

We presented specialized dynamics and control of interacting EAS for single-degree-of-freedom motion of satellites. The thesis also consists of series of experiments that demonstrate the specialized dynamics and piecewise-sinusoidal control of the satellites in one-dimension. The formation control algorithm uses the relative position and estimated velocity to update the control amplitude to converge to the desired relative position. We implemented the intersatellite control in a decentralized formation method using averaged intersatellite control. We proved intersatellite control authority by maintaining a desired relative position between two satellite units in closed-loop experiments with different initial conditions.

For the open-loop and closed-loop control experiments, we presented a custom-built EAS on an experimental testbed, equipped with EM coils, power-electronics and sensing devices that provides the relative position and velocity feedback for the formation control algorithm. An onboard ARM based microcontroller is the main processing center for data acquisition and feedback control of the EAS. In the first two experiments, we determine the operational capability of one-dimensional translation of the EAS in two direction. In the remaining four experiments, we demonstrate the result of the closed-loop control algorithm defined in Chapter 2. From the open-loop experiments we showed that the EAS repels and attraction appropriately with the results depicting a reasonable margin of error due to the ranging accuracy caused by the

sensor. From the closed-loop experiments, we showed that the EAS approaches the desired relative position using the formation control algorithm defined in Chapter 2 for different initial conditions.

In the future, we could redesign the EM coils using high temperature superconductors (HTS) to drive large AC with negligible inductance to generate stronger magnetic force for actuation. The current actuation method is current limited due to the inductance of the EM coil manufactured using copper material for the wire. Future work on this testbed could involve introducing a third satellite unit equipped with an EAS to implement frequency multiplexing AC for desired intersatellite interactions. An offboard vision sensing system with estimation and sensing techniques could be implemented to compare with the current onboard devices that tracks and implements the desired control. A high quality sensor with higher data resolution and faster sampling rate could be used to better estimate the relative position and velocity compared to the current estimation techniques. Additionally, a faster onboard microprocessor capable of sensor fusion could be used for future work in 2D and 3D.

## Appendix A

### Hardware Terminology

- I2C – inter-integrated circuit: a communication protocol used by microcontrollers and peripheral integrated circuits.
- DC Boost Converter – direct-current boost converter: an isolated power converter that increases a variable low voltage to a high voltage output.
- DAC – Digital to Analog Converter: In electronics, a digital-to-analog converter is a system that converts a digital signal into an analog signal.
- DC – Direct current: In electrical engineering direct current is defined the flow of charge in one direction.
- AC – Alternating current: In electrical engineering alternating current is defined the flow of charge in multiple direction over a period of time.
- SD – secure digital: It is a proprietary non-volatile memory card format developed by the SD Card Association for use in portable devices.
- EAS – electromagnetic actuation system: A system that consists of one EM coil and onboard power electronics required for electromagnetic actuation.
- RF – Radio frequency: The oscillation rate of the electromagnetic field in the frequency range 20 kHz to around 300 GHz.
- IDE – integrated development environment: An environment for the development of software applications that includes source code editor, compiler, build packages, and debugger.

## Hardware Components

1. STM VL53L0X laser-rangefinder
2. Floureon 3S Li-Polymer Lipo RC Battery Pack
3. Remington Industries 22SNSP Magnet Wire
4. 3D printed coil frame
5. Lexan polycarbonate sheet
6. PH0362A Eisco linear airtrack
7. PH0362B Eisco quiet air source
8. PH0362GLD2 Airtrack Glider
9. V24A36C400BL Vicor Maxi DC DC Converter
10. ARDUINO DUE ARM core microcontroller
11. 4122Z Copley Controls DC brush servo amplifier

## Software Libraries

1. DFRobot VL53L0X – Software library by sensor module manufacturer for I2C communication between the Arduino Due and the laser range-finder connected on the sensor module.
2. SineWaveDue – Software library for generating sinusoidal signals from DAC pins of Arduino Due microcontroller.
3. SD-Arduino – Arduino library allows for reading from and writing to SD cards onto the wireless SD shield connected to Arduino microcontroller modules.
4. SPI-Arduino – Arduino library allows communication with SPI devices, with the Arduino as the master device.

## Appendix B

clsat1.ino

---

```
/*  
  
    Small Satellite Relative Position and Velocity Control Software.  
    Filename: clsat1.ino  
    Author: Ajin Sunny  
    Last Modified by: Ajin Sunny  
  
    Written for Thesis: Single-Degree-of-Freedom Experiments Demonstrating  
        ↪ Electromagnetic Formation Flying for Small Satellite Swarms Using  
        ↪ Piecewise-Sinusoidal Controls  
    Version: 1.0  
    Date: 02-25-2019  
    Last Updated: 10-03-2019  
  
*/  
  
//HEADER FILES  
#include <DueTimer.h>  
#include <SineWaveDue.h>  
#include <SD.h>  
#include <SPI.h>  
#include <VL53LOX.h>  
#include "Arduino.h"  
#include "DFRobot_VL53LOX.h"  
#include "math.h"  
  
//OBJECTS FOR THE PROGRAM  
DFRobotVL53LOX sensor; // Sensor File Object  
File myFile;           // File Objec  
File raw_File;        // Raw File Object  
  
//VARIABLES REQUIRED FOR THE EXPERIMENT
```

```

unsigned long period = 50000; // Experiment time in milliseconds
unsigned long starttime;
unsigned long endtime;
long loops = 0;
double dist[8] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
const float c = 8.5;
float t1;
float t2;
double k1a = 28.5;
double kr = 1;
double kv = 1;
double vel[9] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
double dist_time[8] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
double dist_filtered[8] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
double velocity_final[4] = {0.0,0.0,0.0,0.0};
double return_vel;
double V_final;
double V_sat;
float a1 = 0;
float a2 = 0;
double desired_dist = 0.350;
double A_v = 0.00;
double A_d = 0.00;
double digital_vsine = 0.0;
double Amplitude = 0.00;
unsigned int i = 1;
unsigned int j = 1;
unsigned int k = 1;
char incomingByte;
double relative_dist = 0.0;
double total_dist = 0.0;
double total_relative_dist = 0.00;
double velocity_final_final = 0.00;
double a = 0.97;
double previous_velocity = 0.00;
double current_velocity = 0.00;

```

```

/*
↳ -----
↳ SETUP
↳ -----
↳ */
void setup() {

```

```

analogReadResolution(10);
analogWriteResolution(10);

//initialize serial communication at 9600 bits per second:
Serial.begin(115200);
//join i2c bus (address optional for master)
while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
}

Serial.print("Initializing SD card...");

    // see if the card is present and can be initialized:
    if (!SD.begin(4)) {
        Serial.println("Card failed, or not present");
        // don't do anything more:
        while (1);
    }
    Serial.println("card initialized.");

Wire.begin();
//Set I2C sub-device address
sensor.begin(0x50);
//Set to Back-to-back mode and high precision mode
sensor.setMode(Continuous, High);
//Laser rangefinder begins to work
sensor.start();
myFile = SD.open("sat1.csv", FILE_WRITE);

myFile.print("Time");
myFile.print(",");
myFile.print("Distance");
myFile.print(",");
myFile.print("Velocity");
myFile.print(",");
myFile.print("Saturated_Velocity");
myFile.print(",");
myFile.print("Amplitude");
myFile.print(",");
myFile.println("Amplitude Digital");

while (Serial.available() == 0) {}
incomingByte = Serial.read();

```

```

    if(incomingByte == 'A')
    {
        Serial.println(incomingByte);
    }
}

/*
↪ -----
↪ MAIN LOOP
↪ -----
↪ */

void loop()
{
    while (millis() < period)
    {
        S.startSinusoid_update(10,A_d);
        if(myFile)
        {
            starttime = millis();

            Serial.print("Start: ");
            Serial.println(starttime);

            V_final = velocity_func();

            if(abs(V_final) <= 0.001)
            {
                V_sat = 0;//The velocity variable that passes to the control
                ↪ function {feedback_algorrithm}
            }

            else{
                V_sat = V_final;
            }

            //Print Time to SD Card
            Serial.print("Time: ");
            Serial.println(millis());
            myFile.print(millis());
            myFile.print(",");

            //Print Relative position to SD Card

```



```

Serial.print("Distance: ");
Serial.println(dist[i-1],8);
myFile.print(dist[i-1],8);
myFile.print(",");

//Print Pre-saturated velocity to SD Card
Serial.print("Velocity: ");
Serial.println(V_final,8);
myFile.print(V_final,8);
myFile.print(",");

//Print saturated velocity(ie. deadzoned velocity) to SD Card
Serial.print("Saturated Velocity: ");
Serial.println(V_sat,8);
myFile.print(V_sat,8);
myFile.print(",");

//Print Feedback Amplitude (u_1) to the SD card and Serial Monitor
Serial.print("Amplitude: ");
Serial.println(A_v,8);
myFile.print(A_v,8);
myFile.print(",");

//Print Feedback Digital Amplitude (in digital For Arduino) to SD
    ↪ Card and Serial Monitor
Serial.print("Amplitude Digital: ");
Serial.println(A_d,8);
myFile.println(A_d,8);

endtime = millis();
Serial.print("End: ");
Serial.println(endtime);
Serial.print("Diff1: ");
Serial.println(endtime-starttime);

if((endtime-starttime) < 100)
{
    delay(100-(endtime-starttime));
    Serial.println("Action1");
    A_v = feedback_algorithm(dist[i-1],V_sat);
    A_d = (A_v*490)/2.75; // Converting voltage to digital voltage
    unsigned int endtime2 = millis();
    Serial.print("Diff2: ");
    Serial.println(endtime2-starttime);
}

```

```

        else{
            A_v = feedback_algorithm(dist[i-1],V_sat);
            A_d = (A_v*490)/2.75;
        }

    }
    S.stopSinusoid();
}
myFile.close();
exit(0);
}

/*
↪ -----
↪ VELOCITY FUNCTION
↪ -----
↪ */

double velocity_func()
{

    dist[i] = sensordistRead();
    if(dist[i] > 0.60 | dist[i] < 0.25 )
    {
        dist[i] = dist[i-1];
    }
    dist_time[i] = (double)millis()/1000;

    vel[j] = (dist[i]-dist[i-1])/(dist_time[i]-dist_time[i-1]);

    if(vel[j] > 0.20 | vel[j] < -0.20)
    {
        vel[j]=vel[j-1];
    }

    velocity_final[i] = a*velocity_final[i-1] + (1-a)*vel[j];
    return_vel = velocity_final[i];

    if (i == 3)
    {
        dist[0]=dist[i]; //shifts the array back to the 0th element of the
        ↪ array.
    }
}

```

```

        dist_time[0] = dist_time[i];
        velocity_final[0] = velocity_final[i];
        i = 0;           // sets the counter back to the first position.
        j = 1;
    }
    i++;
    j++;

    return return_vel;
}

```

```

/*
↳ -----
↳ SENSOR READ FUNCTION
↳ -----
↳ */
double sensordistRead()
{

    double sum = 0;
    double final_relative_dist = 0;
    for(int i=1;i<=7;i++)
    {
        double relative_dist;
        relative_dist = ((sensor.getDistance()/1000.00)+0.195005029);
        sum = sum + relative_dist;
    }
    final_relative_dist = sum/7;
    return final_relative_dist;

}

```

```

/*
↳ -----
↳ FEEDBACK ALGORITHM FUNCTION
↳ -----
↳ */
double feedback_algorithm(double dist, double V_final)
{

```

```
Amplitude = k1a * pow(dist,2) * (pow(abs(tanh(kr * (dist - desired_dist))
↪ + c*tanh(kv * V_final)),0.5));

if(Amplitude > 3.50)
{
return 3.50;
}
else if(Amplitude < -3.50)
{
return -3.50;
}

else{
return Amplitude;
}
}
```

---

## clsat2.ino

---

```
/*
  Small Satellite Relative Position and Velocity Control Software.
  Filename: clsat2.ino
  Author: Ajin Sunny
  Last Modified by: Ajin Sunny

  Written for Thesis: Single-Degree-of-Freedom Experiments Demonstrating
    ↪ Electromagnetic Formation Flying for Small Satellite Swarms Using
    ↪ Piecewise-Sinusoidal Controls
  Version: 1.0
  Date: 02-25-2019
  Last Updated: 10-03-2019

*/

//HEADER FILES
#include <DueTimer.h>
#include <VL53LOX.h>
#include <SineWaveDue.h>
#include <SD.h>
#include <SPI.h>
#include "Arduino.h"
#include "DFRobot_VL53LOX.h"
#include "math.h"

//OBJECTS FOR THE PROGRAM
DFRobotVL53LOX sensor; // SENSOR OBJECT
File myFile;           // SAT DATA FILE OBJECT
File raw_File;        // RAW FILE OBJECT

//VARIABLES REQUIRED FOR THE EXPERIMENT
unsigned long period = 50000; // Experiment time in milliseconds
unsigned int  starttime = 0;
unsigned int  endtime = 0;
unsigned int  stamp_time;
unsigned int  time1 = 0;
long loops = 0;
double dist[8] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0};
const float c = 8.5;
float t1;
float t2;
float delta_pos;
```

```

float velocity;
double k2a = 28.5;
float kr = 1;
float kv = 1;
double vel[9] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
double dist_time[8] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
double dist_filtered[8] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
double velocity_final[4] = {0.0,0.0,0.0,0.0};
double return_vel;
double V_final;
double V_sat;
float a1 = 0;
float a2 = 0;
double desired_dist = 0.350;
double Amplitude = 0.00;
double A_v = 0.00;
double A_d = 0.00;
unsigned int i = 1;
unsigned int j = 1;
unsigned int k = 1;
char incomingByte;
double relative_dist = 0.00;
double total_relative_dist = 0.00;
double velocity_final_final = 0.00;
double a = 0.97;
double previous_velocity = 0.00;
double current_velocity = 0.00;

/*
↳ -----
↳ SETUP
↳ -----
↳ */
void setup() {

    analogReadResolution(10);
    analogWriteResolution(10);

    //initialize serial communication at 115200 bits per second:
    Serial.begin(115200);
    //join i2c bus (address optional for master)
    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
    }

    Serial.print("Initializing SD card...");
    // see if the card is present and can be initialized:

```

```

    if (!SD.begin(4)) {
        Serial.println("Card failed, or not present");
        // don't do anything more:
        while (1);
    }
    Serial.println("card initialized.");

Wire.begin();
//Set I2C sub-device address
sensor.begin(0x50);
//Set to Back-to-back mode and high precision mode
sensor.setMode(Continuous, High);
//Laser rangefinder begins to work
sensor.start();
myFile = SD.open("sat2.csv", FILE_WRITE);

myFile.print("Time");
myFile.print(",");
myFile.print("Distance");
myFile.print(",");
myFile.print("Velocity");
myFile.print(",");
myFile.print("Saturated_Velocity");
myFile.print(",");
myFile.print("Amplitude");
myFile.print(",");
myFile.println("Amplitude Digital");

while (Serial.available() == 0) {}
incomingByte = Serial.read();

    if(incomingByte == 'A')
    {
        Serial.println(incomingByte);
    }
}

/*
↪ -----
↪ MAIN LOOP
↪ -----
↪ */

```

```

void loop()
{
  while (millis() < period)
  {
    S.startSinusoid_update(10,A_d);
    if(myFile)
    {
      starttime = millis();

      Serial.print("Start: ");
      Serial.println(starttime);

      V_final = velocity_func();

      if(abs(V_final) <= 0.001)
      {
        V_sat = 0;
      }

      else{
        V_sat = V_final;
      }

      //Time
      Serial.print("Time: ");
      Serial.println(millis());
      myFile.print(millis());
      myFile.print(",");

      //Distance
      Serial.print("Distance: ");
      Serial.println(dist[i-1],8);
      myFile.print(dist[i-1],8);
      myFile.print(",");

      //Velocity
      Serial.print("Velocity: ");
      Serial.println(V_final,8);
      myFile.print(V_final,8);
      myFile.print(",");

      //Saturated Velocity
      Serial.print("Saturated Velocity: ");
      Serial.println(V_sat,8);
      myFile.print(V_sat,8);
      myFile.print(",");
    }
  }
}

```



```

//Feedback Amplitude (u_2)
Serial.print("Amplitude: ");
Serial.println(A_v,8);
myFile.print(A_v,8);
myFile.print(",");

//Feedback Digital Amplitude (For Arduino)
Serial.print("Amplitude Digital: ");
Serial.println(A_d,8);
myFile.println(A_d,8);

endtime = millis();
Serial.print("End: ");
Serial.println(endtime);
Serial.print("Diff1: ");
Serial.println(endtime-starttime);
if((endtime-starttime) < 100)
{
    delay(100-(endtime-starttime));
    Serial.println("Action1");
    A_v = feedback_algorithm(dist[i-1],V_sat);
    A_d = (A_v*490)/2.75; // Converting voltage to digital voltage
    unsigned int endtime2 = millis();
    Serial.print("Diff2: ");
    Serial.println(endtime2-starttime);
}

else{
    A_v = feedback_algorithm(dist[i-1],V_sat);
    A_d = (A_v*490)/2.75;

}

}
S.stopSinusoid();
}
myFile.close();

exit(0);
}

```

```
/*
```



```

    ↪ VELOCITY FUNCTION
    ↪ -----
    ↪ */

double velocity_func()
{

    dist[i] = sensordistRead();
    if(dist[i] > 0.60 | dist[i] < 0.25)
    {
        dist[i] = dist[i-1];
    }

    dist_time[i] = (double)millis()/1000;

    vel[j] = (dist[i]-dist[i-1])/(dist_time[i]-dist_time[i-1]);

    if(vel[j] > 0.20 | vel[j] < -0.20)
    {
        vel[j]=vel[j-1];
    }
    velocity_final[i] = a*velocity_final[i-1] + (1-a)*vel[j];
    return_vel = velocity_final[i];

    if (i == 3)
    {
        dist[0]=dist[i]; //shifts the array back to the 0th element of the
            ↪ array.
        dist_time[0] = dist_time[i];
        velocity_final[0] = velocity_final[i];
        i = 0; // sets the counter back to the first position.
        j = 1;
    }
    i++;
    j++;

    return return_vel;

}

/*
    ↪ -----
    ↪ SENSOR READ FUNCTION
    ↪ -----

```

```

    ↪ */
double sensordistRead()
{
    double sum = 0;
    double final_relative_dist = 0;
    for(int i=1;i<=7;i++)
    {
        double relative_dist;
        relative_dist = ((sensor.getDistance()/1000)+0.210612583);
        sum = sum + relative_dist;
    }
    final_relative_dist = sum/7;
    return final_relative_dist;
}

/*
    ↪ -----
    ↪ FEEDBACK ALGORITHM FUNCTION
    ↪ -----
    ↪ */

double feedback_algorithm(double dist, double V_final)
{
    if((tanh(kr * (dist - desired_dist)) + c*tanh(kv * V_final)) > 0)
    {
        Amplitude = k2a * pow(dist,2) * (pow(abs(tanh(kr * (dist - desired_dist
            ↪ )) + c*tanh(kv * V_final)),0.5));
    }
    else{
        Amplitude = -1 * k2a * pow(dist,2) * (pow(abs(tanh(kr * (dist -
            ↪ desired_dist)) + c*tanh(kv * V_final)),0.5));
    }

    if(Amplitude > 3.50)
    {
        return 3.50;
    }
    else if(Amplitude < -3.50)
    {
        return -3.50;
    }

    else{
        return Amplitude;
    }
}

```

}

}

---

## Bibliography

- [1] H. Helvajian and S. Janson, “Small satellites: Past,” *Present, and Future*, AIAA, Reston, Virginia, 2009.
- [2] S. S. Board and N. R. Council, *The role of small satellites in NASA and NOAA Earth observation programs*. National Academies Press, 2000.
- [3] S. Nag, C. K. Gatebe, D. W. Miller, and O. L. De Weck, “Effect of satellite formations and imaging modes on global albedo estimation,” *Acta Astronautica*, vol. 126, pp. 77–97, 2016.
- [4] A. K. Porter, D. J. Alinger, R. J. Sedwick, J. Merk, R. A. Opperman, A. Buck, G. Eslinger, P. Fisher, D. W. Miller, and E. Bou, “Demonstration of electromagnetic formation flight and wireless power transfer,” *Journal of Spacecraft and Rockets*, vol. 51, no. 6, pp. 1914–1923, 2014.
- [5] A. K. Porter, D. Alinger, R. J. Sedwick, J. Merk, R. Opperman, A. Buck, G. Eslinger, P. Fisher, D. Miller, and E. Bou, “Dual-purpose resonate actuators for electromagnetic formation flight and wireless power transfer,” in *AIAA Guidance, Navigation, and Control Conference*, 2014, p. 0449.
- [6] I. J. Gravseth and R. D. Culp, *Guidance and Control 2003: Proceedings of the Annual AAS Rocky Mountain Guidance and Control Conference Held February 5-9, 2003, Breckenridge, Colorado, USA*. Amer Astronautical Society, 2003, vol. 113.
- [7] E. M. Kong, D. W. Kwon, S. A. Schweighart, L. M. Elias, R. J. Sedwick, and D. W. Miller, “Electromagnetic formation flight for multisatellite arrays,” *Journal of spacecraft and Rockets*, vol. 41, no. 4, pp. 659–666, 2004.
- [8] U. Ahsun and D. W. Miller, “Dynamics and control of electromagnetic satellite formations,” in *2006 American Control Conference*. IEEE, 2006, pp. 6–pp.
- [9] D. Miller and R. Sedwick, “Electromagnetic formation flight phase ii study,” Massachusetts: Massachusetts Institute of Technology, Tech. Rep., 2005.
- [10] L. M. Elias, “Dynamics of multi-body space interferometers including reaction wheel gyroscopic stiffening effects: structurally connected and electromagnetic formation of flying architectures,” Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

- [11] D. W. Kwon, R. J. Sedwick, S.-i. Lee, and J. Ramirez-Riberos, “Electromagnetic formation flight testbed using superconducting coils,” *Journal of Spacecraft and Rockets*, vol. 48, no. 1, pp. 124–134, 2011.
- [12] D. W. Kwon, “Propellantless formation flight applications using electromagnetic satellite formations,” *Acta Astronautica*, vol. 67, no. 9-10, pp. 1189–1201, 2010.
- [13] U. Ahsun, D. W. Miller, and S. Ahmed, “A hybrid systems approach to closed-loop navigation of electromagnetically actuated satellite formations using potential functions,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6810–6814, 2008.
- [14] Z. Abbasi, J. Hoagg, and T. M. Seigler, “Decentralized position and attitude control for electromagnetic formation flight,” in *AIAA Scitech 2019 Forum*, 2019-0908.
- [15] G. Lafferriere, A. Williams, J. Caughman, and J. Veerman, “Decentralized control of vehicle formations,” *Systems & Control Letters*, vol. 54, no. 9, pp. 899–910, 2005.
- [16] W. Ni and D. Cheng, “Leader-following consensus of multi-agent systems under fixed and switching topologies,” *Systems & Control Letters*, vol. 59, no. 3-4, pp. 209–217, 2010.
- [17] C. Tan and G.-P. Liu, “Consensus of discrete-time linear networked multi-agent systems with communication delays,” *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2962–2968, 2013.
- [18] G. Gu, L. Marinovici, and F. L. Lewis, “Consensusability of discrete-time dynamic multiagent systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2085–2089, 2011.
- [19] R. Chavan, S. Wang, T. M. Seigler, and J. B. Hoagg, “Consensus on  $SO(3)$  with piecewise-continuous sinusoids,” in *2018 American Control Conference (ACC)*. IEEE, 2018, pp. 1635–1640.
- [20] W. Song, J. Thunberg, X. Hu, and Y. Hong, “Distributed high-gain attitude synchronization using rotation vectors,” *Journal of Systems Science and Complexity*, vol. 28, no. 2, pp. 289–304, 2015.
- [21] J. Thunberg, W. Song, E. Montijano, Y. Hong, and X. Hu, “Distributed attitude synchronization control of multi-agent systems with switching topologies,” *Automatica*, vol. 50, no. 3, pp. 832–840, 2014.
- [22] J. R. Lawton and R. W. Beard, “Synchronized multiple spacecraft rotations,” *Automatica*, vol. 38, no. 8, pp. 1359–1364, 2002.
- [23] D. V. Dimarogonas, P. Tsiotras, and K. J. Kyriakopoulos, “Leader–follower cooperative attitude control of multiple rigid bodies,” *Systems & Control Letters*, vol. 58, no. 6, pp. 429–435, 2009.

- [24] A. Sarlette, R. Sepulchre, and N. E. Leonard, “Autonomous rigid body attitude synchronization,” *Automatica*, vol. 45, no. 2, pp. 572–577, 2009.
- [25] W. Yu, G. Chen, M. Cao, and J. Kurths, “Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 881–891, 2009.
- [26] Q. Song, J. Cao, and W. Yu, “Second-order leader-following consensus of nonlinear multi-agent systems via pinning control,” *Systems & Control Letters*, vol. 59, no. 9, pp. 553–562, 2010.
- [27] D. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin, “Decentralized overlapping control of a formation of unmanned aerial vehicles,” *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.
- [28] L. Moreau, “Stability of multiagent systems with time-dependent communication links,” *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.
- [29] G. Wen, Z. Duan, W. Yu, and G. Chen, “Consensus of multi-agent systems with nonlinear dynamics and sampled-data information: a delayed-input approach,” *International Journal of Robust and Nonlinear Control*, vol. 23, no. 6, pp. 602–619, 2013.
- [30] R. C. Youngquist, M. A. Nurge, and S. O. Starr, “Alternating magnetic field forces for satellite formation flying,” *Acta Astronautica*, vol. 84, pp. 197–205, 2013.
- [31] M. A. Nurge, R. C. Youngquist, and S. O. Starr, “A satellite formation flying approach providing both positioning and tracking,” *Acta Astronautica*, vol. 122, pp. 1–9, 2016.
- [32] C. Zhang and X.-L. Huang, “Angular-momentum management of electromagnetic formation flight using alternating magnetic fields,” *Journal of Guidance, Control, and Dynamics*, pp. 1292–1302, 2016.
- [33] Z. Abbasi, A. Sunny, J. Hoagg, and T. M. Seigler, “Relative-position formation control of satellites using electromagnetic actuation with piecewise-sinusoidal controls,” in *2020 American Control Conference*, 2020.
- [34] W. Ren, “Distributed attitude alignment in spacecraft formation flying,” *International Journal of Adaptive Control and Signal Processing*, vol. 21, no. 2-3, pp. 95–113, 2007.
- [35] M. E. Dale and C. R. Sullivan, “General comparison of power loss in single-layer and multi-layer windings,” in *2005 IEEE 36th Power Electronics Specialists Conference*. IEEE, 2005, pp. 582–589.

- [36] —, “Comparison of single-layer and multi-layer windings with physical constraints or strong harmonics,” in *2006 IEEE International Symposium on Industrial Electronics*, vol. 2. IEEE, 2006, pp. 1467–1473.
- [37] Cmasenas, “ajinsunny/sinewavedue: First release,” 2019. [Online]. Available: <https://zenodo.org/record/3497626>



## **Vita**

Name: Ajin Sunny

Degree Awarded: Bachelor of Science in Electrical Engineering

University of Kentucky, Lexington, Kentucky

Scholastic Honor: University of Kentucky Dean's List Fall 2016