Survey of Controller Designs and Implementations on a MagLev-Based Floating Light Bulb

Fernando Zigunov

***Abstract*— In this short paper, a working floating light bulb prototype will be presented along with simulations on its controller design. The light bulb contains a permanent magnet that is used in conjunction with an electromagnet solenoid to perform the levitation task. The energy is transferred to the light bulb with near-field wireless power transmission, where efficiencies of the order of 25% were achieved.**

Details on the failed implementations will be provided along with simulations that provide insight on the reasons why some of the controller topologies applied failed. Three controller topologies will be presented: An LQR-Observer based controller, a digital hysteresis controller and an analog hysteresis controller.

I. INTRODUCTION

Magnetic levitation is a very interesting control problem that has several potential applications in the industry. As Earnshaw's theorem demonstrates, there is no configuration of permanent magnets in space that is statically stable. Therefore, in order to achieve magnetic levitation for a specific application involving permanent magnets, it is necessary to employ some level of active control that changes the system poles to the left-half of the Laplace plane.

Although applications of magnetic levitation are usually related to high-technology industrial applications such as levitating trains and heavy-duty shaft bearings, devices accessible to the general population are already becoming ubiquitous and being mass-produced by several manufacturers as decorative household items. This work will focus on the engineering design of a device that targets a similar decorative application, where a functional light bulb is kept floating through a closed-loop magnetic levitation system.

II. HARDWARE DESIGN AND IMPLEMENTATION

A. Wireless Energy Transfer

In order to provide the floating LED with the energy required for light emission, a wireless transmitter/receiver circuit was designed and implemented. A simple resonant air-cored transformer was employed for near-field energy transfer. This form factor was chosen primarily due to its implementation simplicity and the ability to package into a lightweight form factor, which is required for levitation within the electromagnet force capacity.

The driver consists of a low-side MOSFET driver, as shown in Figure 1, driven by a 600kHz square wave generated by a dedicated Arduino Nano (ATMega 328P). The frequency corresponds to the LC tank fundamental resonant frequency. Since the coupling coefficient of the TX/RX coils is low, the effect of the RX circuit on the resonant frequency is negligible. The microprocessed waveform generation was particularly chosen due to its temperature stability, crucial for high Q-factor transformers.

The capacitors C1 and C2 are polyester with very low ESR, which is also crucial for the formation of a high Q tank circuit, necessary for the strong coupling of the two coils [1]. The coils L1 and L2 were four manually wound, 20 mm diameter loops of AWG24 wire. The inductance indicated in Figure 1 was measured experimentally by finding the tank circuit resonant frequency with a function generator and an oscilloscope. Although a design study on the coil parameters was not performed, the guidelines used involved maximizing the Q factor of the tank circuits by minimizing the parasitic ESR from the coil wire resistance. This was accomplished by increasing the wire diameter and minimizing the number of loops. On the other hand, these actions will reduce the self-inductance of the coil, increasing the resonant frequency. Although in the general case it is not an issue to resonate at higher frequencies, the efficiency of the amplifier circuit would be affected by the parasitic capacitance of the MOSFET, requiring faster-switching drivers and a more complex architecture to support the higher switching current.

Although the design was settled by experimentally building a few different coil configurations with the target LED load, simulations on the coupling coefficients that could be achieved with the coils used were performed to understand the performance levels achieved and whether a strong resonant coupling regime was achieved with the described design. Figure 2 shows the resulting coupling coefficients obtained through FEMM [2]. The desired distance between the the TX and RX coil was about 13 millimeters, which corresponds to a coupling coefficient of $\kappa = 0.065$, however simulations were performed across a larger range to understand the behavior as the RX coil is moved farther from the TX coil. Both tank circuits have a damping ratio of $\zeta_{TX} = \zeta_{RX} = 0.01$, estimated from the ESR values of the components. As defined by [1], the strong coupling regime where efficient wireless energy transfer happens when:

$$\frac{\kappa^2}{\zeta_{TX}\zeta_{RX}} > 1 \tag{1}$$

The strong coupling regime happens throughout the range plotted as long as the coil axes are aligned, however the voltage generated in the RX is only sufficient to light up the LED for $x < 15$ mm. Unfortunately, the test equipment required to characterize the true power transfer efficiency of this implementation was not available, however it is believed that the overall efficiency was less than 25% based on the

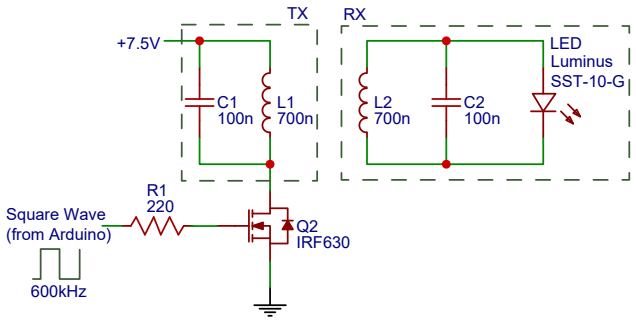

Fig. 1. Air-cored transformer driver circuit and receiver circuit.

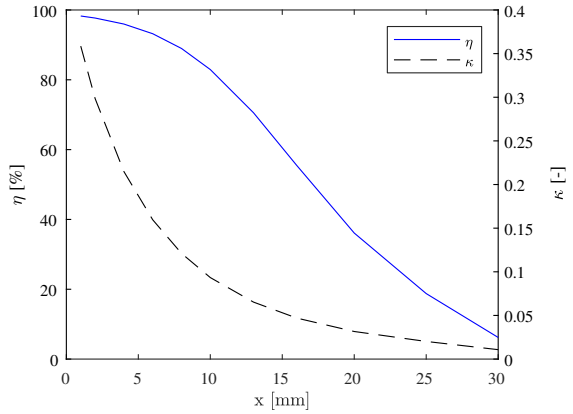

Fig. 2. Coupling coefficients obtained through FEMM simulations and their corresponding power transfer efficiencies for a 30Ω load, considering the driver resonates the transmitter coil.

power supply power consumption and the LED specifications.

B. Magnetic Levitation

The magnetic levitation hardware implementation used materials available to an electronics hobbyist. During the controller implementation phase it was found that this decision introduced a large amount of uncertainty in the model and led to the failure of the theoretical approaches used. However, since this is a very realistic scenario in engineering design, it serves as a benchmark to understand the requirements for control robustness.

The hardware, depicted in Figure 3, consisted of a hand-wound electromagnetic coil attached to a structure in the form of a lamp shade. The lamp shade, however, has the sole purpose of making space to house the electronics for both the magnetic levitation driver and the wireless transmitter. A Hall effect sensor (Honeywell SS41) was used to measure the position of the light bulb by sensing the magnetic field of the neodymium magnet that was embedded on the light bulb construction. An undesired side effect, however, is that the hall effect sensor also senses the magnetic field of the electromagnet, producing an output for the controller that contains two states encoded (electromagnet current and light bulb position).

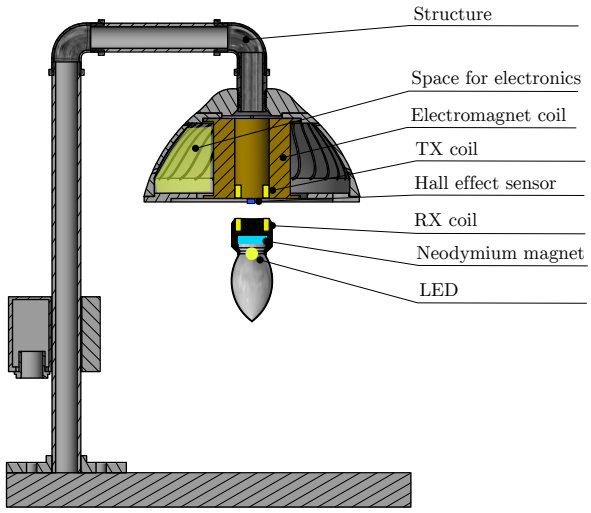

Fig. 3. Mechanical arrangement of the hardware in the lamp shade form factor.

The electromagnet consisted of 1 Kg of AWG24 copper wire wound over a spool of approximately 25 ID x 75 OD x 55 L mm. The coil resistance was measured to be 20 Ω at ambient temperature and its inductance $L = 33.6$ mH. The inductance was easily measured from the step response of the coil in series with a 1 kΩ resistor.

A low-side driver of similar architecture shown in Figure 1 was used to drive the electromagnet through high frequency PWM switching. The full circuit implementation will be discussed in Section IV, since the circuit topology was different for the analog and digital implementations.

III. CONTROLLER DESIGN AND SIMULATIONS

A. Traditional Equations of Motion

Although the floating magnet plant is a true six-degree-of-freedom (6-DOF) system with all spatial and rotational freedom, it is generally sufficient to restrict the model to the unstable degree of freedom for control purposes. The only unstable mode is associated with the motion of the permanent magnet in the axis of the solenoid, which is going to be called z in this work. The positive direction of z is aligned with the gravity vector. In the Appendix, a model based on an infinitesimal magnetic dipole pair is shown to indicate the unstable eigenvalues of the linearized 6-DOF plant.

It is widespread in the literature ([3], [4]) the use of the same simplified non-linear model for a solenoid-based floating magnet, and it is customary to write the equations of motion in the following form:

$$\begin{cases} \ddot{z} = g - k\dfrac{I^2}{z^2} \\ \dot{I} = \dfrac{u}{L} - \dfrac{IR}{L} \end{cases} \tag{2}$$

Where k is an arbitrary constant proportional to the electromagnet strength, g is the gravitational acceleration, I

is the solenoid current and R, L are the solenoid inductance and equivalent series resistance. The control input u is the voltage given to the solenoid. The equations simply model a force balance between the gravity and the electromagnet force, as well as the linear dynamics of the electromagnet circuit.

B. Identification of System Non-linearities

Although the model presented in the last section is based on theoretical results, the simplifying assumptions required to build the model are too restrictive and the real device did not follow the non-linearity $F \propto z^{-2}$, according to measurements made.

In order to build a model that more accurately describes the real device, measurements of the relevant parameters were performed and converted into polynomial fits that described the non-linear functions observed. Then the equations of motion for the controllable modes can be modified, given that only the solenoid intensity (m_s in the Appendix) is accessible through the control input.

1) Electromagnet Force: The first non-linearity analyzed was the electromagnetic pulling force exerted on the permanent magnet, both as a function of the magnet distance and the current input. A measurement of the force versus distance was performed with a scale (resolution 0.1 g) and an micrometric traverse. The magnet to be levitated was placed in the scale with an extra plate that added 120 grams of mass to prevent the magnet from being lifted off the traverse during the test. With the scale tared, data points were then taken at discrete locations with the magnet running the maximum current available and a polynomial curve fit was performed, as displayed in Figure 4. Since the weight of the permanent magnet was 24.2 grams, data points at greater distances were not taken as the system would just drop the magnet. The polynomial fit chosen was a quadratic, mostly due to its low VC dimension (to prevent overfit). A comparison with the theoretical $F \propto z^{-2}$ is shown in a red dashed line, showing how inappropriate the theoretical model is to describe the real device behavior. It is believed that the main source of discrepancy is the permanent magnet being on the near field of the electromagnet (z^{-2} decay assumes far field of two infinitesimal magnetic dipoles).

Since the measurements were performed manually, it was noticed that temperature effects due to the electromagnet warming up when mounted in the enclosure were significantly affecting the curves obtained as detailed measurements took around 10 minutes to perform. Because of this added variable, the number of data points was limited to minimize the effect of warming up on the results. After this limitation, the measurements took approximately 1 minute to perform, which significantly improved the repeatability of the measurements. All measurements shown in this section were started with a cold coil ($T_{coil} \approx 25°C$).

The non-linearity model for the magnetic attraction force as a function of current, which is usually regarded as a function $F \propto I^2$ (or, if following the Biot-Savart law, $F \propto I$), was also found to be inadequate. A measurement with the

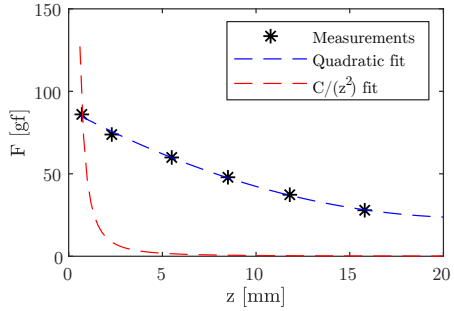

Fig. 4. Measured electromagnet force and comparison of a polynomial fit versus a C/z^4 fit, showing the traditional model inadequacy. The maximum control input was given to the electromagnet.

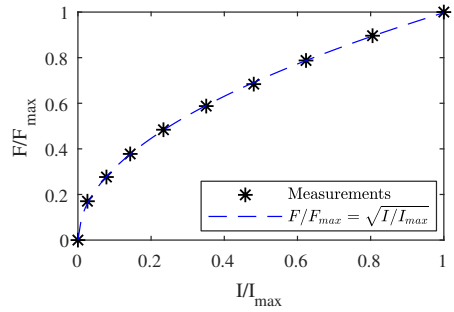

Fig. 5. Measured electromagnet force for $z = 10$ mm and comparison with a square-root fit showing excellent agreement. Force is normalized by the maximum pulling force F_{max} measured at full current I_{max}.

permanent magnet fixed at $z = 10$ mm was performed and, as shown in Figure 5, a square root fit $F \propto \sqrt{I}$ was found to show excellent agreement. The traditional $F \propto I^2$ is an analytical solution that does not correspond to the physical design, as it assumes a high permeability material in the solenoid core with only a small air gap. The Biot-Savart current loop ($F \propto I$) does not consider the effect of finite length of the solenoid. Limited discussion on the physics will be provided here to explain the measured $F \propto \sqrt{I}$ dependence, as a more comprehensive analytical solution would be necessary.

Although the plot shown in Figure 4 uses for simplicity millimeters and grams-force as length and force units, respectively, the curve fits were performed in SI units to ensure consistency during modeling and controller implementation. The polynomial coefficients for force as a function of current $F_1(I)$ and as a function of magnet distance $F_2(z)$ in SI units were, then:

$$\begin{aligned} F_1(I) &= 1.14\sqrt{I} \\ F_2(z) &= 1367z^2 - 59.2z + 0.87 \end{aligned} \tag{3}$$

It was verified that the magnet force $F_m = F_1(I)F_2(z)$ predicted the force correctly for other combinations of (I, z).

2) Hall Effect Sensor Reading: Due to the placement of the Hall effect sensor and its principle of operation, the sensor will respond to both the intensity of the permanent

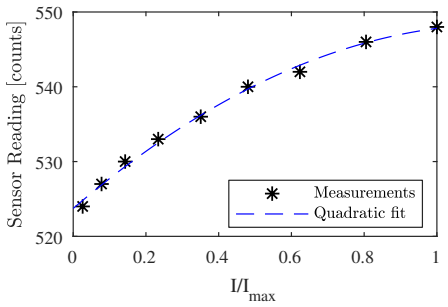

Fig. 6. Raw signal from the hall effect sensor (from the Arduino 10-bit ADC) as a function of electromagnet current.

magnet field (the desired measurement, z) and the intensity of the electromagnet field (which is a function of the control input and the system dynamics). This is inevitable as the Hall effect sensor is required to be placed between the two components in order to have a meaningful signal-to-noise ratio. Placing the Hall sensor between the permanent magnet and the structure base was an option that was ruled out due to aesthetical constraints. Another alternative would be to use a sensor that does not rely on magnetic effects to measure the coil position, where an example would be a time-of-flight optical sensor. These sensors, however, are prohibitively expensive for the small distances that the electromagnet could operate in this implementation, especially at the sampling rates necessary for control.

For these reasons, it is imperative to measure the influence of the electromagnet on the sensor reading. Such measurement is shown in Figure 6 as a raw signal from the Arduino ADC, which was fit to a quadratic polynomial. For reference, the reading at $I/I_{max} = 1$ when the permanent magnet is at $z = 10$ mm is 681 counts, which shows the interference generated by the electromagnet is not negligible. Other measurements not shown in this report showed that the readings from the Hall effect sensor when both the electromagnet and the permanent magnet were present could accurately be described as a superposition of the independent readings from each component. Therefore, a model for the output as measured by the microcontroller was defined as:

$$y = A_1(I) + A_2(z) \tag{4}$$

Where both $A_1(I)$ and $A_2(z)$ are non-linear functions. $A_1(I)$ is the polynomial fit shown in Figure 6. For brevity, the chart for $A_2(z)$ was omitted as a similar procedure was used. The functions $A_1(I)$ and $A_2(z)$ were measured to be:

$$\begin{aligned} A_1(I) &= -29.7I^2 + 54.14I \\ A_2(z) &= 8.94 \times 10^5 z^2 - 3.73 \times 10^4 z + 954.4 \end{aligned} \tag{5}$$

3) Generalized Non-linear Model: Since most of the measurements made resulted in non-linear curve fits that are not of the form shown in Section III-A, the equations of motion need to be generalized to account for the arbitrary non-linear force and input functions. These functions will then be replaced with the polynomial coefficients fitted. Furthermore, the linear system solenoid-resistor is also going to be introduced in the model:

$$\begin{cases} \ddot{z} = g - \frac{1}{M} F_1(I) F_2(z) \\ \dot{I} = \frac{u}{L} - \frac{RI}{L} \\ y = A_1(I) + A_2(z) \end{cases} \tag{6}$$

Where R is the solenoid wire parasitic resistance, L is the solenoid inductance and u is the voltage applied to the solenoid (which is also the control input given by the controller).

C. Linearization Around an Equilibrium Point

A linear controller can be built based on the non-linear equations of motion (6) if they are linearized around an equilibrium point (x_0, u_0). First, we define the system state vector x as:

$$x = \begin{bmatrix} \dot{z} \\ z \\ I \end{bmatrix} \tag{7}$$

The equations of motion (6) can then be linearized by defining the Jacobian matrix of the system dynamics without input:

$$\mathcal{J} = \begin{bmatrix} \partial\ddot{z}/\partial\dot{z} & \partial\ddot{z}/\partial z & \partial\ddot{z}/\partial I \\ \partial\dot{z}/\partial\dot{z} & \partial\dot{z}/\partial z & \partial\dot{z}/\partial I \\ \partial\dot{I}/\partial\dot{z} & \partial\dot{I}/\partial z & \partial\dot{I}/\partial I \end{bmatrix} \tag{8}$$

Which, evaluated for Equation (6), results in:

$$\mathcal{J} = \begin{bmatrix} 0 & -F_1(I)\frac{1}{M}\frac{dF_2}{dz} & -F_2(z)\frac{1}{M}\frac{dF_1}{dI} \\ 1 & 0 & 0 \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \tag{9}$$

Now, assuming there is an equilibrium point for (x_0, u_0), it is possible to linearize the equations by assuming the equilibrium point is disturbed by Δx and Δu:

$$\begin{aligned} \Delta\dot{x} &= A\Delta x + B\Delta u \\ \Delta y &= C\Delta x \end{aligned} \tag{10}$$

Where $A = \mathcal{J}$ is the uncontrolled system Jacobian. Since there is no non-linearity in the input, B is simply:

$$B = \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix} \tag{11}$$

As discussed in Equation 4, the output also is non-linear in relation to the system state. A similar linearization process is used to find C:

$$C = \begin{bmatrix} 0 & \frac{dA_2}{dz} & \frac{dA_1}{dI} \end{bmatrix} \tag{12}$$

Note all derivatives in A and C are evaluated at the equilibrium point x_0.

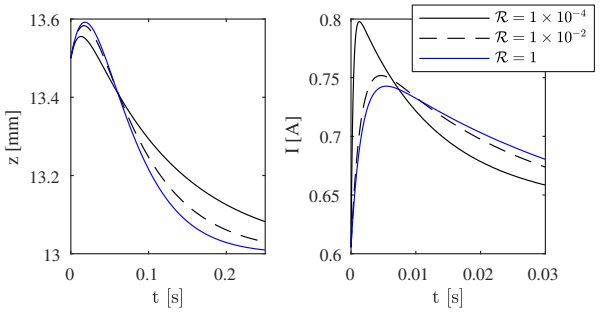

Fig. 7. Effect of the input cost matrix $\mathcal{R}$ on the system states for an initial disturbance $\Delta z(t=0) = 0.5mm$ and $\Delta \dot{z}(t=0) = 10mm/s$.

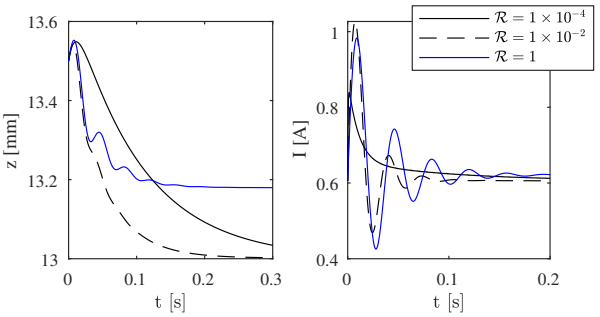

Fig. 8. Same control laws used for Figure 7 applied in the non-linear system modeled in Section III-B.

D. LQR Controller Design

With a linearized system equation in hands, the design of a controller can be done to place the system poles at stable frequencies. Defining the system variables and the equilibrium point for the analysis:

$$\begin{cases} R = 20\Omega \\ L = 33.6 \times 10^{-3} H \\ M = 30 \times 10^{-3} Kg \\ z_0 = 13 \times 10^{-3} m \\ \dot{z_0} = 0 m/s \\ I_0 = 0.605 A \end{cases} \tag{13}$$

Calculating the eigenvalues of A for the equilibrium point given, one unstable eigenvalue ($\lambda_1 = 26.4$) and two stable eigenvalues ($\lambda_2 = -26.4$ and $\lambda_3 = -595.2$) are found. Therefore, the uncontrolled system is unstable, as expected from the physics. It is worth noting that, for the values provided, the system is controllable and observable.

Since the desired equilibrium point is somewhat close to saturation ($I_{max} = 0.81$ A), the control input during the transient cannot exceed $\Delta I = 0.205$ A, corresponding to a maximum control input of $\Delta u = 4.05$ V. The LQR algorithm provides a means to tune the aggressiveness of a control law by varying the input cost matrix $\mathcal{R}$. In Figure 7 the linear system response for different $\mathcal{R}$ values is shown for the linear case, for an initial disturbance of $\Delta z(t=0) = 0.5mm$ and $\Delta \dot{z}(t=0) = 10mm/s$. The initial disturbance was arbitrarily chosen, but it reflects the inaccuracy of the light bulb placement during the initialization of the system. The $\mathcal{Q}$ matrix shown in Equation 14 was used for the optimization. Its form was chosen to bring all states (which have different units) to similar orders of magnitude such that the LQR algorithm would weight deviations in all states equally.

$$\mathcal{Q} = \begin{bmatrix} 1 \times 10^2 & 0 & 0 \\ 0 & 1 \times 10^3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

To understand the impact of the system non-linearities in the controller performance, the same controllers obtained for Figure 7 were simulated again for the full non-linear system using the Matlab ODE45 integrator, and their responses are plotted in Figure 8. It is very interesting how the controller responses are vastly different due to the non-linearity. This behavior, however, is expected because the non-linearity is very strong considering the displacements analyzed. Even though the system stability was not compromised, the controllers with $\mathcal{R} = 1$ and $\mathcal{R} = 1 \times 10^{-2}$ now display saturation considering the system capacity, which means they would fail in the physical device. Surprisingly, the only controller that does not display saturation is the most aggressive ($\mathcal{R} = 1 \times 10^{-4}$), where ΔI peaked at 0.2 A. On the other hand, the control input is very close to saturation, leaving little margin for inaccuracies. Other controllers were designed with pole placement at different locations in the s-plane, with similar performance degradation when simulated in the non-linear system.

The controller gain matrix K obtained for $\mathcal{R} = 1 \times 10^{-4}$ is shown below. It will be further analyzed in the next sections, as it was one of the few candidates that did not saturate the control input when applied to the non-linear system.

$$K = \begin{bmatrix} -1214 & -18467 & 85.2 \end{bmatrix} \tag{15}$$

With this controller matrix, the eigenvalues of the linear system were placed at $\lambda_1 = -3034 s^{-1}$, $\lambda_2 = -87.3 s^{-1}$ and $\lambda_3 = -8.5 s^{-1}$. However, the first eigenvalue places a very strong requirement when the system is digitized for microcontroller implementation, as it sets the order of magnitude of the sampling rates necessary to avoid aliasing.

Reducing the magnitude of λ_1 is possible at the expense of saturating the controller and even making the system unstable. For example, for $\lambda_1 = 510 s^{-1}$ and maintaining the other two pole locations, the non-linear system becomes unstable for the initial disturbance studied, as shown in Figure 9. As will further be discussed later, this is believed to be one of the reasons why the digital implementations of this controller (as well as many other attempts not reported here) failed.

E. Observer Design

The controller designed in the former subsection requires full state feedback. However, as discussed in Section III-B, the Hall effect sensor will only provide indirect information about the system state, requiring the design of an observer to

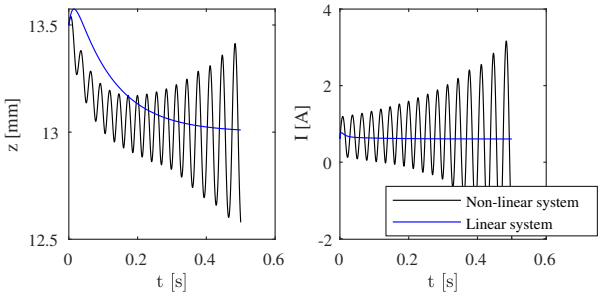

Fig. 9. Failed pole placement in non-linear system for linearized system poles at $\lambda_1 = -510s^{-1}$, $\lambda_2 = -87.3s^{-1}$ and $\lambda_3 = -8.5s^{-1}$.

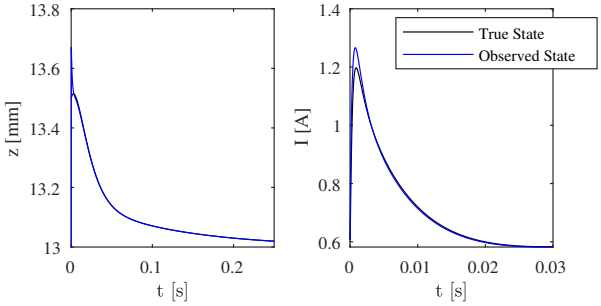

Fig. 10. Performance degradation of Luenberger observer implementation with poles 10 times faster than the LQR controller poles. Degradation observed against Figure 7 for the fully linearized system. Blue traces indicate observed state, which is used for control input computation.

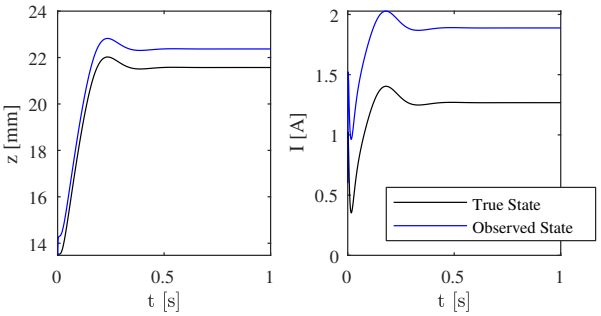

Fig. 11. Further degradation of performance for the same observer implemented in Fig. 10 when simulated in the non-linear system. Output equation is still simulated as fully linear.

enable the use of the LQR controller in the practical device. Only the case without noisy measurements will be examined here, as the linear observer was found to lack the robustness necessary to deal with the system non-linearities.

A linear system observer can be simply designed by using Ackermann's method on the linearized system matrix A^T and the linearized output matrix C^T. Pole placement can then be performed arbitrarily, but in general as a rule of thumb it is desired to have the observer settle around 5 to 10 times faster than the controller poles.

A Luenberger observer L was designed with pole placement at locations 10 times faster than the controller poles ($\lambda_{o,i} = 10\lambda_{c,i}$). This observer was applied together with the control law K from Equation 15 for the linear system. As can be observed from Figure 10 (compare against Figure 7), the observer degrades the performance but is still stable.

If this observer is now applied to the fully non-linear system of Equation 6, a very fast instability develops (not shown). In order to investigate the effect of the non-linearity that makes the system unstable, the output non-linearity was removed by replacing the y equation in Equation 6 with its linear version at the equilibrium point. This change makes the system stable again, and the results are ploted in Figure 11. The system settles in a different equilibrium point because its linear observer applied to the non-linear system makes the controller believe it is elsewhere in the state space, which is clearly observed by the steady-state error between the true and observed state lines. When the input non-linearity is added to the simulation, the error in the observed state becomes so much larger that it destabilizes the controller.

Because the simplified, noiseless version of the linearized controller/observer already saturates the control input (not shown in Figure 11, but can be inferred from $I > I_{max}$), the linearization technique is unlikely to produce any meaningful results for the physical system. Therefore, effects of noise, time discretization, output digitization, control input discretization and PWM will not be further analyzed as increased performance degradation from these effects is expected.

F. Hysteresis Controller

Perhaps on the opposite side of the spectrum of controller complexity lies the hysteresis controller, sometimes also called the Bang-Bang controller. It is very common to use a hysteresis controller in magnetic levitation apparatuses, especially in hobby level projects. Apart from its obvious simplicity, it seems to be popular belief that this kind of controller is unconditionally stable under high enough sampling frequencies. However, no theoretical evidence supporting such claim was found in the literature review performed in this work.

The hyteresis controller has, in its simplest form, only one parameter. A signal s is generated as a function of the sensor inputs and compared with a threshold value t. Then, the controller is simply defined as:

$$u = \begin{cases} 0, & \text{for } s \leq t \\ K_u, & \text{for } s > t \end{cases} \tag{16}$$

In the system studied, the signal s is a function of the only input ($s = -y$) and the threshold is a function of the equilibrium point input value ($t = -y_0 = -642$). The signs are reversed because y decreases as the permanent magnet falls.

Simulation results of this control rule applied on the fully non-linear controller for the same initial conditions presented in Section III-D are shown in Figure 12. Apart from the effect state and output non-linearity, this simulation was also performed with digitization effects (sensor bit rounding).

In this simulation, the physics were resolved with a time step of 1×10^{-6} s and the control input was updated

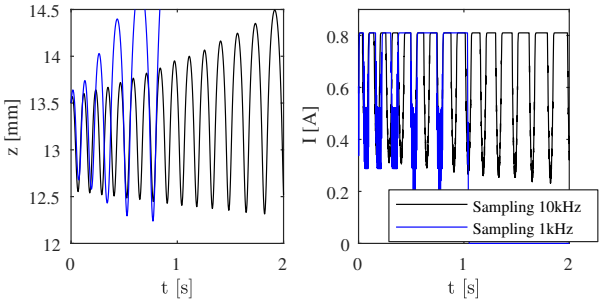

Fig. 12. Failed hysteresis controller simulated at initial conditions $\Delta\dot{z}(t = 0) = 10$ mm/s, $\Delta z(t = 0) = 0.5$ mm and $\Delta I(t = 0) = 0$ A at two different sampling rates.

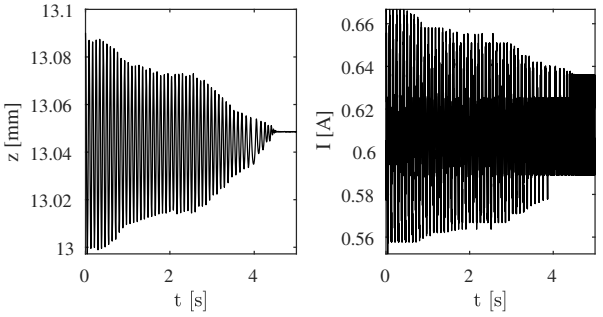

Fig. 13. Successful hysteresis controller simulated at initial conditions $\Delta\dot{z}(t = 0) = 0$ mm/s, $\Delta z(t = 0) = 0.2$ mm and $\Delta I(t = 0) = 0$ A. In the limit cycle, the control input waveform is a 2470 Hz square wave.

every 1×10^{-4} s or 1×10^{-3} s (the two sampling rates shown in Figure 12)). Lower sampling rates for the controller make the system more unstable, but increasing the sampling frequency does not improve the system stability for the initial conditions used. The results show that the hysteresis controller is also unstable for this initial condition. However, it is worth noting the time scale of the simulation, which is two seconds long. The LQR-observer based controller, for comparison, exceeded $z = 20$ mm in 1 millisecond for the same conditions with the full non-linear system.

Since the hysteresis-based controller was more promising on its stability and robustness characteristics, another simulation was performed with a more lenient initial condition of $\Delta\dot{z}(t = 0) = 0$ mm/s, $\Delta z(t = 0) = 0.2$ mm and $\Delta I(t = 0) = 0$, for which the LQR controller applied to the linear system still produces an unstable output. The results for the hysteresis controller, shown in Figure 13, display that the controller is now stable to these initial conditions. The longer simulation time of 5 seconds show that the controller reaches a stable point, which is at $z = 13.05$ mm due to the rounding of the sensor reading. A somewhat narrow-band limit cycle switching at 2470 Hz is reached at this equilibrium point. However, the control input is not a pure square wave, having noticeable frequency jitter.

Finally, the effect of bit noise was assessed for the same initial conditions of $\Delta z(t = 0) = 0.2$ mm and the other two states being the equilibrium conditions. The bit noise was observed to be an issue during the practical implementation,

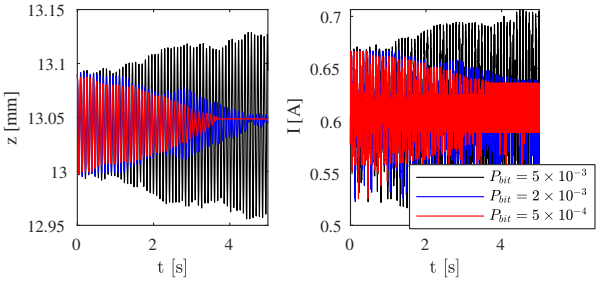

Fig. 14. Effect of random bit noise in the hysteresis controller stability for three bit change probabilities P_{bit}. Initial conditions are the same as Fig. 13.

and for this study it was defined as a uniformly distributed random process that would add or subtract one unit to the sensor reading y with a probability P_{bit}. The adding or subtracting would have equal probabilities. Other models, like introducing white noise in the analog signal prior to digitization had similar effects, however not reported here. The results of this study are summarized in Figure 14 and show that the system is robust to a bit noise probability of approximately $P_{bit} = 2 \times 10^{-4}$. Above this noise level, the system becomes unstable. Unfortunately, it was observed empirically from reading the Hall effect sensor to the Arduino ADC that the probability of last bit flipping with the coil and the wireless transmitter off was of $P_{bit} \approx 0.1$ when reading 1×10^5 samples. Similar tests at arbitrary magnet positions (still with the electromagnet off) were performed to remove the bias that an analog level between the two bit values could be the cause of bit flipping, with similar probability levels.

IV. PHYSICAL IMPLEMENTATION

Since the observer-based controller was found to lack the robustness necessary to be implemented in the real hardware, experimental data on its instability will not be presented herein for brevity. Many experiments were performed with the digital version of the observer-based controller, but all resulted in unstable systems. The deeper analysis shown in Section III-E reveals the causes of the controller failure, which in summary are related to the high non-linearity of the physical system and the non-linear coupling of two state variables in the input. The remaining non-idealities (digitization, sampling, noise) only exacerbated the effect of implementing an inadequate controller algorithm.

In the following subsections the results from the implementation of the hysteresis controller will be shown and analyzed.

A. Digital Hysteresis Controller

The digital implementation using an Arduino Nano board was implemented for both the LQR and the hysteresis controller. Its electrical circuit is shown in Figure 15 (a). The only input is the Hall effect sensor, and it drives the output MOSFET with a digital output. For the failed LQR implementations, the output signal to the MOSFET was an 16kHz, 8-bit PWM signal generated from the ATmega

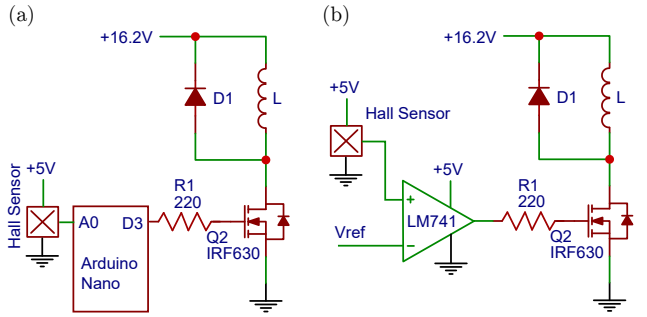

Fig. 15. Schematics of the driver circuits for the electromagnet. (a) Digital controller using an Arduino Nano. (b) Analog controller using an operational amplifier operating as a comparator.

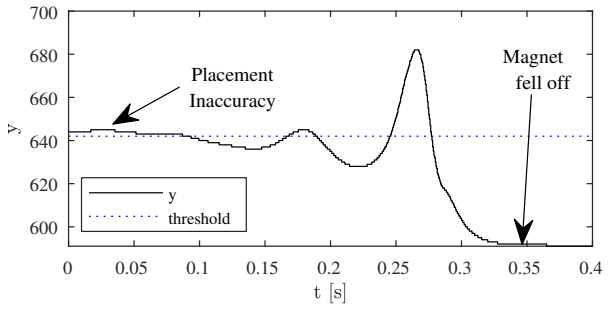

Fig. 16. Input seen by the Arduino controller operating a hysteresis controller at 1950 Hz. Input plateau at final time is the signal when the electromagnet is on but the permanent magnet is far.

Timer0 clock. Every iteration of the main loop would update the register that controls the duty cycle of the PWM signal, which had 8 bits of resolution, for the LQR controller case.

For the digital hysteresis controller, however, the implementation was much simpler. The signal read directly from the Hall effect sensor was directly compared with the threshold value and updated the digital output according to Equation 16. Since the sampling rate of the microcontroller depends heavily on the main loop overhead, the fastest controller obtained (operating at approximately 3.2kS/s) could not send signals to the computer for recording and analysis. However, by using the USB to Serial converter embedded in the Arduino, the signals could be sent with increased overhead, which reduced the sampling rate to 1.95 kS/s.

Both hysteresis controllers, however, were unstable at the sampling rates achieved. An example signal from the controller that used the USB to Serial interface (slower) is shown in Figure 16. It displays the raw Hall effect sensor signal, as read directly by the Arduino from its 10-bit ADC. This signal can be directly compared with the function for y given in Equation (4). As the figure displays, the digital hysteresis controller is unstable. The simulations performed in Section III-F point to two main causes for the instability: The ADC noise and the very low sampling rate. The problem with the sampling rate, specifically, is that the controller cannot capture the effect of the electromagnet dynamics, which is very fast (about 1.7 ms time constant). Since a faster sampling rate is not viable with the Arduino hardware, a decision was made to change the hardware and implement an analog controller.

B. Analog Hysteresis Controller

The analog controller implemented is a simple comparator operational amplifier and replaces the Arduino, as shown in Figure 15 (b). The unit gain bandwidth of the LM741 is very large, typically 1.5 MHz, but for the input changes expected (order 50mV) the output is expected to have a bandwidth in the order of 15 kHz. Therefore, a performance improvement in relation to the digital controller is expected, obviously at the cost of controller flexibility. The voltage V_{ref} to the comparator was supplied by the Arduino board, as there was

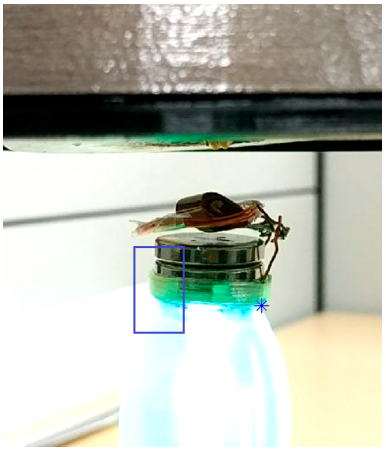

Fig. 17. Snapshot of the tracking video representing the tracking performed in the physical device. Blue box represents the feature window. Blue star is an arbitrarily placed point for visual confirmation of tracking success. Video of the tracking point available here. A more complete video of the working device is provided in this link

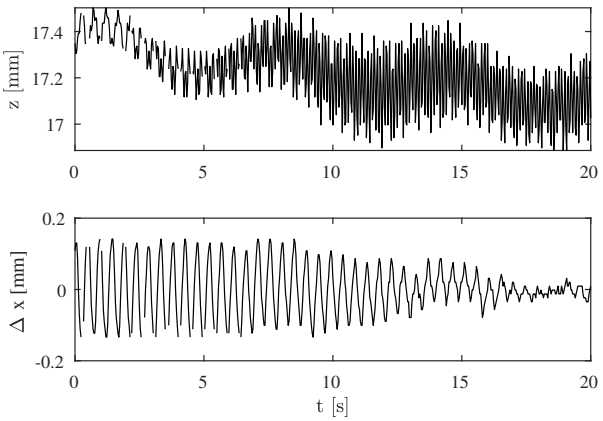

Fig. 18. Tracked image data for the magnet location as a function of time for the analog hysteresis controller based on the LM741 operational amplifier. Δx represents the sideways motion. Note the very long time scale.

no reason to remove it from the hardware. It was generated by supplying a higher PWM rate (64kHz) through an RC low-pass filter. The Arduino Nano hardware does not have any digital-to-analog converters.

In order to assess and report the efficacy of the analog controller, a digital image correlation algorithm was developed to track images of a video of the floating light bulb in operation, giving quantitative evidence of the results achieved. A simple dot pattern was used to re-scale the images to physical coordinates.

A frame of the tracking video is shown in Figure 17 with the light bulb successfully floating and lighting up. The full video is available online in this link. The figure also shows the feature window used, which provided good tracking accuracy due to its contrasting colors. It also provided an axisymmetric surface, which was necessary as the lamp rotated during the capture due to the expected misalignment during initial placement. The tracking was performed in grayscale images, and the z and x displacements are shown for 20 seconds of a video in Figure 18. They showcase the effectiveness of the analog hysteresis controller, which in practice holds the lamp for very long times (about 2-10 minutes). Some discussion on the reasons why the controller becomes unstable will be provided in the next section.

V. CONCLUSIONS AND PRACTICAL CONSIDERATIONS

In summary, this practical controller design exercise showed several engineering challenges that must be overtaken in order to successfully build a magnetic levitator. In this last section, some practical considerations and lessons learned will be briefly discussed to summarize the difficulties encountered during the physical implementation of the controller.

1) The Danger of Simplifications: It is very attractive to attempt to simplify the real-world applications to tractable mathematical constructs in order to gain insight about a problem. However, it is more often than not that the practicing engineer will find the models insufficient to provide useful solutions in real applications.

In the case studied, the linearizations performed proved inadequate even in simulated models. Thus, one must be extremely careful in new applications when carrying analytical models to real applications to ensure they describe the real system with sufficient accuracy. However, it could have been possible, with further modeling and the application of non-linear control techniques, to control the fully non-linear plant.

2) Model Complexity / Robustness Trade-off: In this study, opposite ends of the spectrum in model complexity were examined. On one hand, the LQR-Observer implementation requires the model of the system and its state-space equation. On the other hand, the hysteresis controller is model-free and requires a single parameter, its set-point threshold.

The increased model complexity of the LQR controller provides more confidence on the placement of the system poles (if the system was linear), at the expense of narrowing down the range of its applicability. Uncertainty in the curve fits performed to get the non-linear functions, as well as in the plant characteristics due to external factors will lead the controller to unknown regions of the Laplace plane. For both cases, robustness needs to be assessed if the application is critical. However, due to the lack of extra parameters in the model-free approach, it is very likely that the set of parameters for which the model is stable constitutes a larger fraction of the parameter space.

3) Effect of Coil Temperature: Although very briefly discussed in this work, it was found the coil power dissipation was found to be the major reason why the analog hysteresis controller eventually becomes unstable. The warming up of the coil causes its wire resistance to increase, changing not only its current but also its dynamics. The ratio of resistances of the cold coil versus its equilibrium temperature (which is very specific to this construction geometry) was found to be approximately $R_{hot}/R_{cold} = 1.2$.

One can tune the voltage V_{ref} in the analog controller to be stable when the coil is cold and change it as the coil heats up, which would require a temperature sensor input and fine-tuning to get the temperature curve stable across all temperatures. In this work, however, the reference voltage was constant, meaning that the stable operation time is limited.

4) Presence of the Wireless Transmitter: Another interesting practical observation specific to levitating objects where power transfer takes place is that the wireless transmitter signal will be captured by the Hall effect sensor. If the Hall effect sensor is sufficiently slow, the high frequency signal will be low-passed and not be a problem. However, the small signal can still contribute to noise at the digitization levels, systematically biasing the rounding of the digital readings and potentially causing the destabilization of the system.

Specifically to the ATmega328P microprocessor, the absence of analog filters in its analog inputs will cause the high frequency signal to be aliased, characterizing a non-white sensor noise reading. A signal of 2 mV amplitude at 600 kHz was measured at the analog input port where the Hall effect sensor was connected when the TX coil was on. This was also a reason why the analog hysteresis controller potentially performed better than its digital counterpart, as the comparator amplifier provides another layer of low frequency filtering.

REFERENCES

[1] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljačić, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, no. 5834, pp. 83–86, 2007.

[2] D. C. Meeker, "Finite element method magnetics, version 4.2 (28feb2018 build)."

[3] P. Suster and A. Jadlovska, "Modeling and control design of magnetic levitation system," pp. 295–299, 01 2012.

[4] F. Gómez-Salas, Y. Wang, and Q. Zhu, "Design of a discrete tracking controller for a magnetic levitation system: A nonlinear rational model approach," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–8, 03 2015.

[5] P. B. L. Kar W. Yung and D. D. Villani, "An analytic solution for the force between two magnetic dipoles," *Magnetic and Electrical Separation*, vol. 9, no. 1, pp. 39–52, 1998.

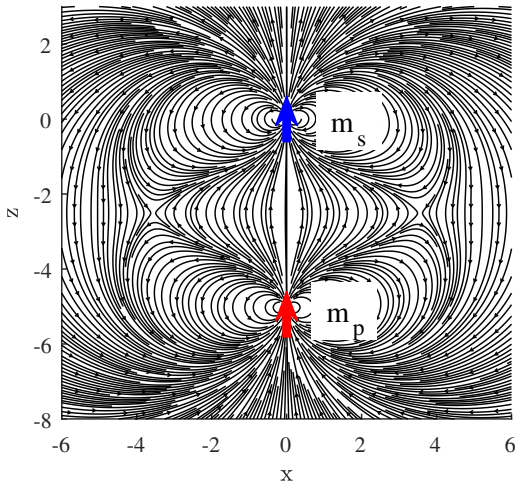

Fig. 19. Slice along the axis of the solenoid magnetic dipole m_s displaying the magnetic field $\vec{B}$ of two aligned magnetic dipoles according to the Biot-Savart law.

[6] D. D. V. Peter B. Landecker and K. W. Yung, "An analytic solution for the torque between two magnetic dipoles," *Magnetic and Electrical Separation*, vol. 10, no. 1, pp. 29–33, 1999.

APPENDIX

A. Three-dimensional system stability

As briefly discussed in Section III-A, the floating magnet problem is inherently a three-dimensional control problem, where the floating element has all 6 degrees of freedom. In order to facilitate an analytical understanding of the control problem of the real system, both the solenoid and the permanent magnet will be simplified as magnetic dipoles of dipole moment $\vec{m_s}$ and $\vec{m_p}$, respectively. A depiction of the $\vec{B}$ field generated by these magnetic dipoles is shown in Figure 19 for reference.

An analytical solution for the force between two magnetic dipoles based on the Biot-Savart law is provided by [5] and shown in Equation 17, where $\vec{r}$ is the distance vector between the two dipoles, $\hat{r}$ is its unit vector and r is its magnitude. Furthermore, μ_0 is the permeability of free space.

$$\vec{F} = \frac{3\mu_0}{4\pi r^4}\Big((\hat{r}\times\vec{m_s})\times\vec{m_p} + (\hat{r}\times\vec{m_p})\times\vec{m_s} - 2\hat{r}(\vec{m_s}\cdot\vec{m_p}) + 5\hat{r}[(\hat{r}\times\vec{m_s})\cdot(\hat{r}\times\vec{m_p})]\Big) \tag{17}$$

The same authors later [6] provided a calculation for the torque between dipoles, where m_s and m_p are the magnitude of the dipole moments:

$$\vec{\tau} = \frac{\mu_0 m_s m_p}{4\pi r^3}[3(\hat{m_s}\cdot\hat{r})(\hat{m_p}\times\hat{r}) + (\hat{m_p}\times\hat{m_s})] \tag{18}$$

Equations 17 and 18 are approximate under the assumption that the distance r between the dipoles is large. Let's further consider the configuration of Figure 3, where the solenoid is above the permanent magnet in a Cartesian coordinate system where $\hat{z}$ is downwards. In the desired state, the two dipoles are aligned in the vertical direction ($\hat{m_s} = \hat{m_p} = \hat{z}$) and positioned directly under another ($\vec{r} = r\hat{z}$).

In order to perform a linearized disturbance analysis, let's first add a small rotation to the dipole moment of the permanent magnet $\vec{m_p}$ and a small perturbation to its position vector $\vec{r}$. The disturbed vectors $\vec{m_p^*}$ and $\vec{r^*}$ would then be expressed as:

$$\vec{m_p^*} = R_x(\theta)R_y(\phi)\vec{m_p} = m_p\begin{bmatrix}\zeta_x\\ \zeta_y\\ 1\end{bmatrix} \tag{19}$$

$$\vec{r^*} = \begin{bmatrix}\epsilon_x\\ \epsilon_y\\ z\end{bmatrix} \tag{20}$$

Where $\zeta_x = \theta$ and $\zeta_y = \phi$ are small because θ and ϕ in the rotation matrices R_i are small angles. No rotation around the z axis was considered. Similarly, ϵ_x and ϵ_y are also small compared to z. It is straightforward to replace the disturbed vectors in the force and torque Equations 17 and 18 and drop the higher order terms to obtain:

$$\vec{F} = \begin{bmatrix}\frac{3\mu_0 m_s m_p(z\zeta_x - 4\epsilon_x)}{4\pi z^4|z|}\\ \frac{3\mu_0 m_s m_p(z\zeta_y - 4\epsilon_y)}{4\pi z^4|z|}\\ -\frac{3\mu_0 m_s m_p}{2\pi z^3|z|}\end{bmatrix} \tag{21}$$

$$\vec{\tau} = \begin{bmatrix}-\frac{m_s m_p \mu_0}{4\pi z^4|z|}(3\epsilon_y z - 2\zeta_y z^2)\\ \frac{m_s m_p \mu_0}{4\pi z^4|z|}(3\epsilon_x z - 2\zeta_x z^2)\\ 0\end{bmatrix} \tag{22}$$

Where the z component of torque is zero because it only contains higher order terms. To assess the stability of the system, it is more convenient to lump the positive constant $C = m_p\mu_0/4\pi$ in the force and torque vectors. It is also further assumed for now that m_s, the solenoid dipole, is constant:

$$\vec{F} = \begin{bmatrix}\frac{m_s}{z^4|z|}(3Cz\zeta_x - 12C\epsilon_x)\\ \frac{m_s}{z^4|z|}(3Cz\zeta_y - 12C\epsilon_y)\\ -\frac{2Cm_s}{z^3|z|}\end{bmatrix} \tag{23}$$

$$\vec{\tau} = \begin{bmatrix}-\frac{m_s}{z^4|z|}(3C\epsilon_y z - C\zeta_y z^2)\\ \frac{m_s}{z^4|z|}(3C\epsilon_x z - C\zeta_x z^2)\\ 0\end{bmatrix} \tag{24}$$

All dependences are now linearized, except for the z dependence. In order to keep the equations of motion of the floating magnet analytically tractable, the permanent magnet will be assumed to be spherical, such that its products of inertia are zero and its tensor of inertia $\mathbf{I}$ is a diagonal matrix where all non-zero elements are equal ($I_{xx} = I_{yy} = I_{zz} = I$). The classical equations of motion:

$$\begin{cases} M\ddot{\vec{r}}^{*} = \vec{F} \\ \mathbf{I}\dot{\vec{\omega}} = \vec{\tau} \end{cases} \tag{25}$$

Can then be used for assessing the system stability. Note the angular acceleration $\dot{\vec{\omega}}$ is the second time derivative of $\vec{m}_p^*/m_p$, which means the z component equation is already identically satisfied by the zero torque τ_z.

In order to perform a linearization of the z displacement around an equilibrium point z_0, let's define a position vector $\mathbf{x}$ as the combination of the disturbance displacement and rotation, where ϵ_z is a z disturbance around the equilibrium point z_0.

$$\mathbf{x} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \zeta_x \\ \zeta_y \end{bmatrix} \tag{26}$$

The linearization of the force and torque equations gives the following Jacobian matrix $\mathcal{J}$:

$$\mathcal{J} = \begin{bmatrix} \frac{-12Cm_s}{Mz_0^4|z_0|} & 0 & 0 & \frac{3Cm_s}{Mz_0^3|z_0|} & 0 \\ 0 & -\frac{-12Cm_s}{Mz_0^4|z_0|} & 0 & 0 & \frac{3Cm_s}{Mz_0^3|z_0|} \\ 0 & 0 & \frac{-8Cm_s}{Mz_0^4|z_0|} & 0 & 0 \\ 0 & \frac{-3Cm_s}{Iz_0^3|z_0|} & 0 & 0 & \frac{Cm_s}{Iz_0^2|z_0|} \\ \frac{3Cm_s}{Iz_0^3|z_0|} & 0 & 0 & \frac{-Cm_s}{Iz_0^2|z_0|} & 0 \end{bmatrix} \tag{27}$$

The equations of motion (25) then can be written in state-space form:

$$\begin{bmatrix} \ddot{\mathbf{x}} \\ \dot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathcal{J} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{x} \end{bmatrix} \tag{28}$$

Where $\begin{bmatrix} \mathbf{0} & \mathcal{J} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} = A$ is the system matrix.

Finding the eigenvalues of A then allows one to assess the stability of this 3D system for small disturbances. The expression forms for the eigenvalues are non-trivial and will not be reproduced herein. Numerical analysis of the expressions lead to unstable eigenmodes that are related only to the ϵ_z motion, as well as lateral modes that combine the $\zeta_{x,y}$ and $\epsilon_{x,y}$ displacement. Figure 20 displays how the eigenvalues related to these modes distribute in the Laplace plane, for a range of C values between 0.1 and 1 when the remaining constants are maintained equal to unity. Similar numerical studies have been performed when the other constants are varied, resulting in very similar eigenvalue plots. For each case, 4 eigenvalues related to the lateral modes are always unstable, apart from the unstable ϵ_z eigenmode.

These results lead to believe that, apart from the expected unstable z mode, there are also unstable lateral and torsional modes that even though are fully decopuled from the z mode from analysis of the eigenvectors of A, can also become unstable. Further analysis needs to be performed to understand if the instability of the lateral modes is related to the z mode instability or if the nature of the instability is related to the precession motion arising from the angular disturbances introduced.

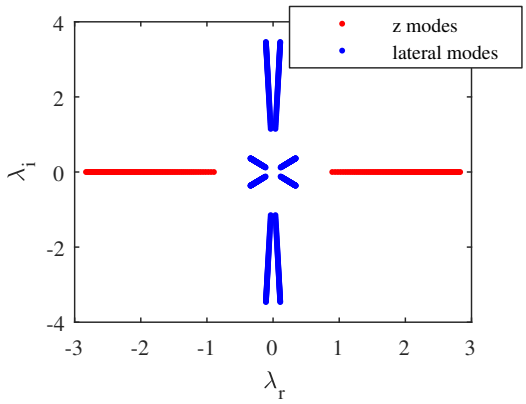

Fig. 20. Eigenvalues of A obtained by numerically sweeping $0.1 < C < 1$ for $m_p = m_s = \mu_0 = M = I = z_0 = 1$ plotted in the complex plane.