

MagnetoShield: Prototype of a Low-Cost Magnetic Levitation Device for Control Education

Gergely Takács*, Jakub Mihalík, Erik Mikuláš and Martin Gulán

Institute of Automation, Measurement and Applied Informatics

Faculty of Mechanical Engineering

Slovak University of Technology in Bratislava

Bratislava, Slovakia

*gergely.takacs@stuba.sk

Abstract—This article presents a prototype reference design for a low-cost miniature magnetic levitation experiment. The proposed device is built as an Arduino expansion module, thus may be installed on a broad range of microcontroller prototyping boards. The open-source hardware design uses off-the-shelf and widely available components and 3D printing technology, thus its overall material cost is minimal. This way, the magnetic levitation experiment is transformed into a pocket laboratory that can be borrowed by students for take-home experiments. In addition to the device itself, we present an open-source application programming interface and the outline of classroom examples in modeling, system identification and closed-loop control.

Index Terms—magnetic levitation, control engineering education, student experiments, open educational resources, microcontrollers, mechatronics, educational technology, system modeling, system identification

I. INTRODUCTION

The dynamics of magnetic levitation systems is nonlinear, open-loop unstable and with fast dynamics; rendering modeling, estimation and eventually closed-loop control a particularly challenging task [1]. These challenges are welcome in controller or observer design, which is why numerous academic research papers use magnetic levitation systems for testing and verification purposes. The engaging visual feedback and the gravity-defying dynamics also make these magnetic levitation systems attractive for classroom use. This way, control engineering and mechatronics students gain valuable laboratory experience in modeling, system identification, signal processing and closed-loop control.

Unfortunately, the price of commercial magnetic levitation devices ranges from several thousands up to even several tens of thousands of US dollars. The vast investment cost means that some schools find it difficult to equip laboratories by off-the-shelf solutions. Even if commercial equipment is readily available, students can only use them under close supervision and their borrowing for take-home experiments is practically unimaginable. On the upside, commercial equipment is well

The authors gratefully acknowledge the contribution of the Slovak Research and Development Agency (APVV) under the contracts APVV-18-0023, APVV-17-0214 and APVV-14-0399. The authors appreciate the financial support provided by the Cultural and Educational Grant Agency (KEGA) of the Ministry of Education of Slovak Republic under the contract 005STU-4/2018.

made, comes with customer support and a selection of educational examples. One of the most prominent features of commercial solutions is their hardware and software consistency, thus making the verification, replication and comparison of results among various research teams possible. Some examples of commercial magnetic levitation devices and their utilization in academic literature includes Feedback Instruments 33-210 [2], [3], Educational Control Products (ECP) Model 730 [4], [5], Amira MA 400 Magnetic Suspension [6], InTeKo Magnetic Levitation System [7], Humusoft CE152 [8] and Quanser Consulting Magnetic Levitation [9], [10].

An alternative route chosen by many researchers and educators to equip a controls or mechatronics laboratory with a magnetic levitation device is to construct an improvised do-it-yourself device, e.g. [11]–[17]. Lilienkamp et al. have encouraged student groups to come up with low-cost improvised designs themselves [18], [19], thus integrating hardware design to their course. The economy of individual designs is certainly a motivation [18], [20], yet in certain cases the auxiliary equipment, such as amplifiers, laboratory measurement cards and computers, adds a considerable overhead to the base device, c.f. [1]. The downside of custom designs is that they are hardly universal, since most devices remain in the stage of one-off prototypes with closed hardware design that is not available to the public. Creating open curricula or comparing and verifying results by various research teams is thus impossible.

The immense success of the open-source hardware movement and the so-called “maker” subculture has shaped the Arduino microcontroller prototyping boards into a standard and widely accepted tool not only for hobbyists but for many institutions of higher education as well. The electronic connection of the header pins of Arduino boards is kept consistent across various models and even for third-party products. This way, expansion modules known as “shields” may be connected to the base prototyping board this way, broadening its functionality by additional hardware such as Ethernet connection, GPS receiver, SD card expansion, etc.

Previously we have proposed that these Arduino shields could be also used to as a platform to build pocket laboratories for control engineering and mechatronics courses [21]–[23]. We have recently introduced reference designs for miniaturized open-source trainers implementing an optical experiment



Fig. 1. Low-cost magnetic levitation experimental device prototype.

[21], a thermal experiment using a 3D printer hot-end [22], and an air levitation device [23]. By making the hardware and software design of these devices fully open-source, we get the advantages of both improvised and commercial hardware. The reference designs are kept as simple as possible, using cheap and easily available electronic and mechanical components and 3D printing technology. Thus, the hardware is low-cost and small, making it ideal for take-home experiments or long-term student projects. The open-source nature of hardware makes standardization possible, thus software and even course material may be openly shared amongst educators or researchers.

We will present a prototype reference design of a low-cost open-source magnetic levitation experiment in this article (Fig. 1). The magnetic levitation system is miniaturized to a size that fits on a standard Arduino shield, hence we shall refer to it as the “MagnetoShield” in the upcoming sections. Besides the hardware itself, we will introduce an open-source application programming interface (API) and outline possible classroom examples in modeling, system identification and closed-loop control.

II. HARDWARE DESIGN

The schematic representation of the proposed prototype is shown in Fig. 2. The position of a disc-shaped levitating permanent magnet (a) is monitored indirectly by a Hall effect sensor (b). The analog voltage signal from the sensor is passed through an analog-to-digital converter (ADC) (c) to the microcontroller (d). Inputs computed in the microcontroller are turned to an analog voltage signal through a digital-to-analog converter (DAC) (e) then passed to an amplification circuit (f) to the solenoid (g). A current sensor (h) is employed to gain additional insight to dynamic processes, mainly for system identification purposes.

A. Schematic design

The design assumes that the microcontroller prototyping board such as the Arduino Uno, Mega, Zero or Due uses the Arduino R3 pin layout with the standardized electronic and mechanical layout.

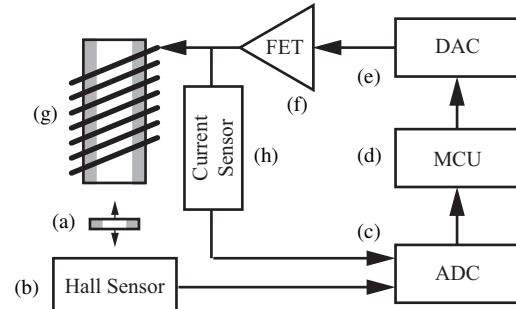


Fig. 2. Schematic representation of device functionality.

Let us begin with the input signal path and describe the electronic design of the MagnetoShield with the help of the schematic shown in Fig. 3. The microcontroller ((a), U1)—for example an atMega 328P in case of an Arduino Uno—is connected to an NXP Semiconductors PCF8591T external 8-bit DAC chip ((b), U3) via the I²C bus, pulled up to 5 V ((c), R1, R2). Although some actuators, such as motors, can be conveniently supplied by a pulse width modulated (PWM) signal, tests have shown that the fast dynamic response of the levitation process is affected by this type of input significantly. The 12 V supply voltage (d) to the electromagnet is pulled from the common V_{IN} pin of the Arduino layout. This means that the barrel jack connector on the Arduino can be used to power the electromagnet by a wall-plug adapter. The input voltage from the DAC chip is then fed to an IRF520 N-channel MOSFET ((e), Q2) in a low-side configuration. The analog output of the pin is protected in transients by a small-value series resistor, while open floating states are prevented with a parallel resistor ((f), R3, R4). The solenoid ((g), L1) goes to the high-side of this configuration, thus when there is gate-source voltage on the MOSFET, the channel starts to conduct and the electromagnet is energized. Transient effects are filtered by a capacitor ((h), C2), while the MOSFET and voltage source is protected from back-EMF by a diode ((i), D1).

Let us now inspect the indirect distance sensing circuitry by the Hall sensor. The device uses an Allegro MicroSystems A1302KUA continuous-time ratiometric linear Hall-effect sensor ((j), U2). Since the magnetic levitation experiment is designed to be compatible with ARM Cortex M-series boards as well, a 3.3 V Zener diode is used to protect analog inputs from overvoltage. The hall sensor is bipolar and is supplied from the 5 V rail, therefore an incidental swap of the magnet polarities could result in overvoltage on these devices. Other sensing circuits use the same type of Zener clippers ((k), D3–D5) too. The output of the Hall sensor is connected directly to the Arduino, which contains built-in ADC peripheries.

Although a non-essential functionality, sensing the current passing through the electromagnet may aid the mathematical modeling of the system, providing data for estimation and system identification tasks. The electromagnet is powered through a precise shunt resistor with a known resistance ((l),

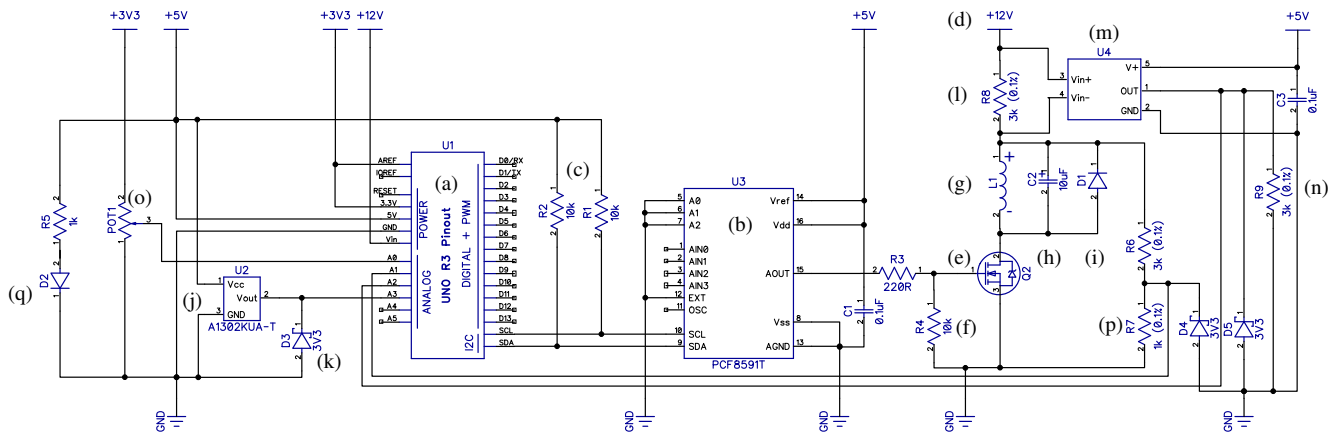


Fig. 3. Electronic schematics of the device.

R8). This differential input signal is then amplified using a Texas Instruments INA169 high-side measurement current shunt monitor ((m), U4) with a gain configured by a precise resistor ((n), R9). The output of this amplifier is then fed to the ADC of the Arduino, which is also protected by the aforementioned Zener clamp.

There are simple auxiliary circuits extending the functionality of the MagnetoShield. A potentiometer ((o), POT1) can be programmed to perform a number of roles, most notably it may supply a manually adjusted reference trajectory for position. There is a voltage divider circuit ((p), R6, R7) that monitors supply voltage, and finally, a LED signals power to the board ((q), D6).

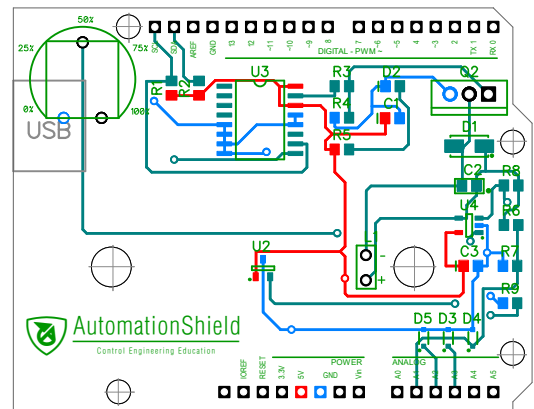
B. Printed Circuit Board and Assembled Device

The schematics and printed circuit board of the device have been designed in the DipTrace CAD/CAE environment. As the device is relatively simple, a two-layer PCB is sufficient and economic, and most online manufacturers offer 10 pieces under \$5. The final PCB is illustrated in Fig. 4. All components are mounted on the top layer, while the bottom one contains only traces. The schematic file, board layout including the ready-to-manufacture production files are available for further modification and use [24].

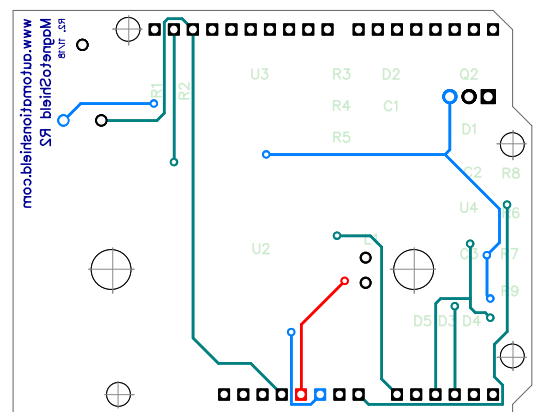
The surface mounted components can be easily soldered by hand with little experience. The assembly of the board may thus hold an educational value in itself.

The structure for holding the solenoid was designed in Autodesk Fusion 360 and printed by a Prusa i3 MK3/S 3D printer using PETG filament in 80 minutes time. It requires 6 g of filament, rendering its cost to less than \$0.20 including the 0.2 kWh electricity consumption of the printer itself. The model for this structure is included in the open-source documentation of the MagnetoShield [24].

A top view of the assembled device is shown in Fig. 5. The circuit board contains long stacking headers (I) which connect the device to a compatible Arduino. The magnet is held in place by a 3D printed superstructure (II). The 12 V rail for the magnet is powered through the barrel connector of the



(a) Top layer.



(b) Bottom layer.

Fig. 4. Printed circuit board, 1:1 scale.

development board underneath, while the USB programming port of an Arduino Uno is also shown here. The equivalents of the circuit components described above for the schematics are marked on the assembled device with the same letters.

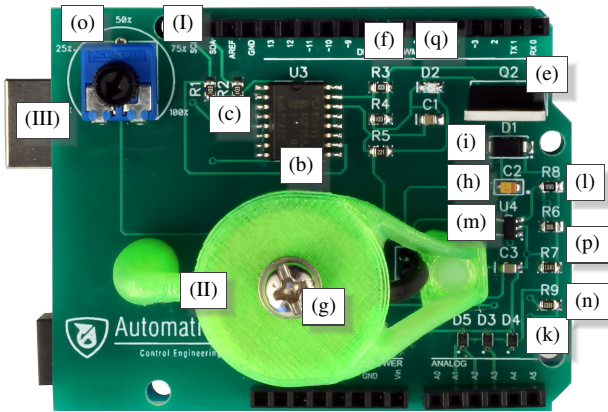


Fig. 5. Top view of the assembled device

C. Hardware Cost

An exhaustive component list and manufacturing cost estimate is provided in Tab. I. Component unit prices are given in US dollars for low purchase quantities that are required for building a single device. Prices in the table are selected from the offering of several suppliers and vendors to minimize overall cost. All prices shown in Tab. I exclude postage, packing, material (solder, flux, etc.) and labor costs. In case of certain components, such as parts of the superstructure, the price has been figured for a single unit based on the required amount of material.

As it is evident, even the small-batch manufacturing price is under \$10, which is affordable to most students and educational institutes. With production volume increased, the authors believe the unit price can be kept at this level even when labor, postage and material are accounted for.

We have excluded the price of the Arduino board itself and the 12 V wall adapter from this calculation, since the microcontroller board and the adapter can be reused for other purposes.

III. APPLICATION PROGRAMMING INTERFACE

The goal of the application programming interface is to create an abstraction layer between the hardware and the control systems engineering problem itself. Although creating an API may hold an educational value on its own—for example in courses aimed at microcontrollers and embedded hardware—the readily available interface allows students to start working on their control or mechatronics assignments right away.

The API is available for the Arduino IDE as a standard library, and may be installed through the usual means, e.g. the graphical user interface. The MagnetoShield functionality is a part of the larger AutomationShield library, which provides an interface for other similar devices as well [24].

A. Initialization and calibration

The entire interface to the hardware is managed in the `MagnetoShield` header and implementation file, by the `MagnetoClass` class, which is pre-initialized into the

`MagnetoShield` object. The device should then be started by the

```
MagnetoShield.begin()
```

method, which sets the analog reference to external. This way the 3.3 V rail connected to the reference pin acts as the maximum for the 10-bit built-in ADC ensuring compatibility with non-AVR boards. In addition to this, the method calls the `Wire` library for the I2C bus functionality required for the DAC chip.

Just as other architecture-specific parts of the library, this method is conditionally compiled depending on the specific board and architecture configured by the user. The library is compatible to and has been tested on Atmel AVR (e.g. Uno, Mega), Atmel SAMD (e.g. Zero) and Atmel SAM (e.g. Due) microcontrollers.

It is recommended to start the applications by self-calibrating the device, by calling the

```
MagnetoShield.calibration() ;
```

method, which measures the output of the Hall effect sensor with the solenoid off and on full power. The calibrated ADC levels can be accessed by the `getMinCalibrated()` and `getMaxCalibrated()` methods. Direct distance measurement is not available on this device and it is not even needed for simple feedback control, but first-principle models require a distance signal. The distance of the permanent magnet from the solenoid is known at the two extremes; e.g. when it rests on the surface of the PCB directly over the sensor and when it hits the roof of the enclosure. Let us assume that there is a power relationship between the magnetic flux $B(k)$ measured on the Hall sensor and the distance $h(k)$ at time $t = kT_s$, so that

$$h(k) = p_1 B(k)^{p_2}. \quad (1)$$

The calibration routine thus performs a two-point calibration searching for constants p_1 and p_2 based on this idea. In case the user does not run the calibration routine, default values are given as well.

B. Input

For general feedback control experiments, it does not actually matter what units we use to power the solenoid: it may be percentage of full power, voltage, current, etc. However, if we wish to use first-principle models or model-based control, the input to the solenoid shall be

```
MagnetoShield.actuatorWrite(u) ;
```

which applies the input voltage $u(k)$ ranging from 0 to 12 V at the given sample.

The method uses the results of an offline multi-point calibration procedure, in which the voltage measured on the solenoid by the Digilent Analog Discovery 2 probe was fit to the 8-bit DAC input levels by the `MATLAB` Curve Fitting Toolbox. The best match to this measurement was obtained by assuming that the DAC levels $u_{DAC}(k)$ depend on the desired voltage level $u(k)$ by the following relation:

$$u_{DAC}(k) = p_3 u(k)^{p_4} + e^{p_5 u(k)}, \quad (2)$$

TABLE I
COMPONENT LIST AND COST CALCULATION IN U.S. DOLLARS.

Name	Part no., value.	Designator	Mark	Pcs.	Price	Total
3D print	5.7 g $\phi=1.75$ mm PETG filament, bright green, at 240°C (90°C bed)	-	-	1	0.18	0.18
Capacitor	0805, ceramic, 0.1 μ F (e.g. KEMET C0805C104K3RACTU, 25 V)	C1,C3	-	2	0.13	0.26
Capacitor	0805, tantalum, 10 μ F (e.g. AVX TAJR106M010RNJ, 10 V)	C2	(h)	1	0.10	0.10
Enclosure top	clear acrylic; e.g. h=2 mm, stamped to the outer diameter of the tube	-	-	1	0.05	0.05
Current sensor	Texas Instruments INA169	U4	(m)	1	1.07	1.07
DAC	NXP Semiconductors PCF8591T	U3	(b)	1	0.79	0.79
Diode	DO214AC (e.g. Vishay Semiconductor BYG20J, 1.5 A, 600 V)	D1	(i)	1	0.15	0.15
Hall sensor	Allegro Microsystems A1302ELHLT-T	U2	(j)	1	0.71	0.71
Header	6x1, female, 2.54 mm pitch	-	-	1	0.07	0.07
Header	8x1, female, 2.54 mm pitch	-	-	2	0.10	0.20
Header	10x1, female, 2.54 mm pitch	-	-	1	0.10	0.10
LED	0805, red	D2	(q)	1	0.16	0.16
Magnet	NdFeB, disc, $\phi=8$ mm, h=2 mm, N38	-	-	1	0.09	0.09
MOSFET	IRF520	Q2	(e)	1	0.22	0.22
PCB	2 layer, FR4, 1.6 mm thick	-	-	1	0.23	0.23
Pot	10 k, 250 mW (e.g. ACP CA14NV12,5-10KA2020)	POT1	(o)	1	0.10	0.10
Resistor	10 k Ω , 0805	R1,R2,R4	(c),(f)	3	0.01	0.03
Resistor	3 k Ω , 0805, 0.1%, 0.125 W (e.g. Viking AR05BTCW3001)	R6, R9	(n),(p)	2	0.35	0.71
Resistor	1 k Ω , 0805, 0.1%, 0.125 W (e.g. Viking AR05BTCW1001)	R7	(p,q)	1	0.08	0.16
Resistor	220 Ω , 0805	R3	(f)	1	0.01	0.01
O-Ring	rubber, M12, h=1 mm, e.g. $\phi=18$ mm (outer),	-	-	1	0.03	0.03
Screws	polyamid, M3x8	-	-	2	0.01	0.03
Shaft	ACP CA9MA9005	-	-	1	0.12	0.12
Shunt	10 Ω , 0805, 0.1%, 0.1 W (e.g. ROYAL OHM TC0525B0100T5)	R8	(l)	1	0.47	0.47
Solenoid	ELE-P20/15, $\phi=20$ mm, h=15 mm, 12/24 V, 25 N	L1	(g)	1	2.67	2.67
Enclosure tube	clear, Plexiglas XT, h=8 mm, $\phi=10$ mm (inner), $\phi=12$ mm (outer)	-	-	1	0.03	0.03
Zener diode	3.3V, SOD323 (e.g. NEXPERIA BZX384-C3V3.115)	D3-D5	(k)	3	0.02	0.06
					Total:	\$8.79

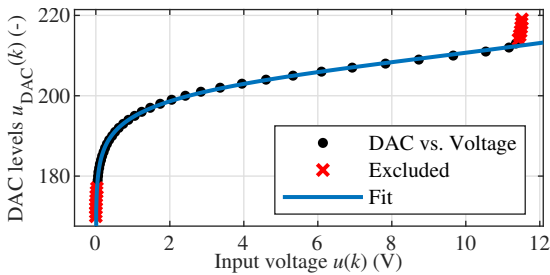


Fig. 6. Curve fitting results for input voltage vs. DAC levels.

resulting in a $R^2 = 0.9991$ coefficient of determination for the area of interest (see Fig. 6). The appropriate DAC levels are thus recalculated in the `voltageToDac()` method, which takes a floating point number and returns a byte. This is then passed onto the `dacWrite()` method, addressing the DAC chip, engaging the DAC register and writing the desired output byte. Alternatively, input to the solenoid can be also sent by employing the `actuatorWritePercents()` method.

C. Output

The distance of the permanent magnet from the solenoid can be determined by calling the

```
y=MagnetoShield.sensorRead();
```

method that returns the distance $y(k) = h(k)$ in millimeters as a floating point number. The method reads the ADC levels from the Hall sensor and calls `adcToGauss()`, converting the levels to voltage related to the 3.3 V reference, removing the 2.5 V bias for zero magnetic flux and multiplying by the

manufacturer given sensitivity of 1.3 mV/G. The magnetic flux is then recalculated by the two-point calibration in Eq. (1) implemented in the `gaussToDistance()` method. Sensor readings can be also acquired in percents, by employing the `sensorReadPercents()`.

D. Auxiliary functions

There are some non-essential auxiliary functions provided in the device API. First of these is the method reading the current sensor that is useful when one considers a first-principle model of the process. The current sensor is accessed by

```
i=MagnetoShield.auxReadCurrent();
```

which returns the current reading $i(k)$ in milliamperes. The method polls for ADC levels, which are converted to voltage and this in turn to current by a proportional gain defined by the current sensor chip.

The external reference potentiometer is read by

```
r=MagnetoShield.referenceRead();
```

which returns a floating point number in the range of 0–100%. Finally, calling

```
vin=MagnetoShield.auxReadVoltage();
```

will return the supply voltage of the 12 V rail, being mainly useful for self-diagnostics.

IV. DIDACTIC EXAMPLES

The following section describes experiments carried out on the proposed device in system identification and closed-loop control.

TABLE II
LIST OF SYMBOLS USED IN MODELING.

Symbol	Description	Unit
$h(t)$	Instantaneous distance from solenoid	m
m	Mass of the magnet	kg
$i(t)$	Instantaneous solenoid current	A
K	Magnetic force constant	Nm^2A^{-2}
L	Solenoid inductance	H
R	Solenoid resistance	Ω
$u(t)$	Instantaneous solenoid voltage	V
F	Magnetic force	N
$B(t)$	Instantaneous magnetic flux density	G
N	Number of turns on the solenoid	-
A	Active cross sectional area of the solenoid	m^2
μ	Permeability of air	NA^{-2}
μ_0	Vacuum permeability	NA^{-2}
g	Gravitational acceleration	m s^{-2}

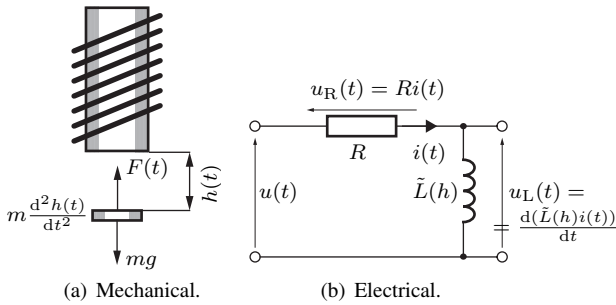


Fig. 7. Mechanical and electrical first principles to represent the device.

Our primary aim is to demonstrate and validate the functionality of the hardware, but we also suggest possible laboratory exercises for control engineering and mechatronics courses. In order to preserve the educational value of the discussion, we have chosen to include details that are usually omitted from academic articles. These sample experiments are included with the AutomationShield library for students to repeat and explore.

The reader shall note that it is not possible to include an entire spectrum of sample exercises here. Nevertheless, the hardware itself poses no practical limitations.

A. Modeling

Let us analyze the physical background of the magnetic levitation experiment by inspecting the forces acting on the permanent magnet. These forces are illustrated in Fig. 7(a). To simplify the model, we shall disregard non-essential forces such as air drag. A list of symbols is given in Tab. II.

In this case, we can express the sum of forces acting on a levitating point mass in a magnetic field by

$$m \frac{d^2h(t)}{dt^2} = mg - F(t), \quad (3)$$

where m denotes the mass of the object, $h(t)$ is the distance measured from the surface of the electromagnet to the object and g is the gravitational acceleration.

The electromagnetic force $F(t)$ can be expressed by

$$F(t) = \frac{B(t)^2 A}{2\mu_0}, \quad (4)$$

where $B(t)$ is the magnetic flux, A is the area of the electromagnet and μ_0 is the magnetic permeability of free space. The magnetic flux in turn depends on the current passing through the coil, that is

$$B(t) = \frac{N\mu i(t)}{2\mu_0}, \quad (5)$$

where N is the number of turns, μ is magnetic permeability, and finally, $i(t)$ is the current. This can be substituted back to Eq. (4) to obtain

$$F = \frac{\mu^2 N^2 A i(t)^2}{2\mu_0 h(t)^2} = K \frac{i(t)^2}{h(t)^2}, \quad (6)$$

where, to simplify further notation, we may introduce the constant K expressing the electromagnetic force for a unit area and current, given as

$$K = \frac{\mu^2 N^2 A}{2\mu_0}. \quad (7)$$

The dynamic equation for the position will thus reduce to

$$m \frac{d^2h(t)}{dt^2} = mg - K \frac{i(t)^2}{h(t)^2}. \quad (8)$$

Let us now assume that the electromagnet can be modeled by a series RL circuit and ignore transient effects caused by circuit capacitance. An illustration of the equivalent circuit is then shown in Fig. 7(b). According to Kirchhoff's voltage law, the total voltage across the electromagnet is contributed by the change of current in the inductor and ohmic resistance, that is

$$u(t) = \frac{d(\tilde{L}(h)i(t))}{dt} + Ri(t). \quad (9)$$

Instead of assuming the inductance as a constant value, as it is often the case in the literature [16], and because a permanent magnet is used in our device, it is beneficial to employ a distance-dependent inductance parameter $\tilde{L}(h)$. Now, let us assume that $\tilde{L}(h)$ is composed of a base inductance L which is modified by the moving permanent magnet. Suppose that the permanent magnet modifies the inductance by L_0 , related to its relative position from the equilibrium h_0 . This idea can be represented mathematically as follows [13]:

$$\tilde{L}(h) = L + \frac{L_0 h_0}{h(t)}. \quad (10)$$

Substituting this distance-dependent inductance model back to Eq. (9) and differentiating it yields

$$u(t) = L \frac{di(t)}{dt} + Ri(t) - L_0 h_0 \frac{i(t)}{h(t)^2} \frac{dh(t)}{dt}. \quad (11)$$

Subsequently, after re-arranging and differentiating the product $\tilde{L}(h)i(t)$, we obtain

$$\frac{d^2h(t)}{dt^2} = g - \frac{K i(t)^2}{m h(t)^2}, \quad (12a)$$

$$\frac{di(t)}{dt} = -\frac{R}{L} i(t) + \frac{L_0 h_0}{L} \frac{i(t)}{h(t)^2} \frac{dh(t)}{dt} + \frac{1}{L} u(t). \quad (12b)$$

Furthermore, because of the conservation of energy it is valid that

$$K = \frac{L_0 h_0}{2}, \quad (13)$$

and thus we may avoid using the unknown L_0 parameter and arrive at the nonlinear model of magnetic levitation given by the differential equations:

$$\frac{d^2 h(t)}{dt^2} = g - \frac{K i(t)^2}{m h(t)^2}, \quad (14a)$$

$$\frac{di(t)}{dt} = \frac{2K}{L} \frac{i(t)}{h(t)^2} \frac{dh(t)}{dt} - \frac{R}{L} i(t) + \frac{1}{L} u(t). \quad (14b)$$

1) *Linear Transfer Function:* Let us linearize by expressing the first-order term of the Taylor expansion of Eq. (14) in the equilibrium position h_0 and the corresponding input voltage u_0 and resulting current i_0 , that is, compute

$$\frac{d^2 \Delta h(t)}{dt^2} \approx \left. \frac{\partial \left(g - \frac{K i(t)^2}{m h(t)^2} \right)}{\partial h(t)} \right|_{h_0, i_0} \Delta h(t) + \quad (15a)$$

$$+ \left. \frac{\partial \left(g - \frac{K i(t)^2}{m h(t)^2} \right)}{\partial i(t)} \right|_{h_0, i_0} \Delta i(t),$$

$$\frac{d \Delta i(t)}{dt} \approx \left. \frac{\partial \left(\frac{2K}{L} \frac{i(t)}{h(t)^2} \frac{dh(t)}{dt} - \frac{R}{L} i(t) + \frac{1}{L} u(t) \right)}{\partial \left(\frac{d \Delta h(t)}{dt} \right)} \right|_{h_0, i_0} \frac{d \Delta h(t)}{dt} + \quad (15b)$$

$$+ \left. \frac{\partial \left(\frac{2K}{L} \frac{i(t)}{h(t)^2} \frac{dh(t)}{dt} - \frac{R}{L} i(t) + \frac{1}{L} u(t) \right)}{\partial i(t)} \right|_{h_0, i_0} \Delta i(t) +$$

$$+ \left. \frac{\partial \left(\frac{2K}{L} \frac{i(t)}{h(t)^2} \frac{dh(t)}{dt} - \frac{R}{L} i(t) + \frac{1}{L} u(t) \right)}{\partial u(t)} \right|_{h_0, i_0} \Delta u(t),$$

to obtain the linear perturbation model

$$\frac{d^2 \Delta h(t)}{dt^2} \approx \left(\frac{2K i_0^2}{m h_0^3} \right) \Delta h(t) - \left(\frac{2K i_0}{m h_0^2} \right) \Delta i(t), \quad (16a)$$

$$\frac{d \Delta i(t)}{dt} \approx \left(\frac{2K i_0}{L h_0^2} \right) \frac{d \Delta h(t)}{dt} - \frac{R}{L} \Delta i(t) + \frac{1}{L} \Delta u(t). \quad (16b)$$

To simplify notation, we will further assume that the linearized approximation sufficiently represents the real dynamics, thus we will use the equality sign from now on.

Applying the Laplace transform to Eq. (16a) we get

$$\left(s^2 - \left(\frac{2K i_0^2}{m h_0^3} \right) \right) \Delta H(s) = - \left(\frac{2K i_0}{m h_0^2} \right) \Delta I(s), \quad (17)$$

while doing the same to Eq. (16b) we obtain the s -domain transform of the electrical equation

$$s \Delta I(s) = \left(\frac{2K i_0}{L h_0^2} \right) s \Delta H(s) - \frac{R}{L} \Delta I(s) + \frac{1}{L} \Delta U(s), \quad (18)$$

and after rearranging the terms we can express the current as

$$\Delta I(s) = \frac{\left(\frac{2K i_0}{L h_0^2} \right) s \Delta H(s) + \frac{1}{L} \Delta U(s)}{\left(s + \frac{R}{L} \right)}. \quad (19)$$

Let us now substitute Eq. (19) to the transfer function for the position of the point mass in dependence of the current in Eq. (17). After some algebraic manipulation we arrive at

$$\left[\left(s^2 - \left(\frac{2K i_0^2}{m h_0^3} \right) \right) \left(s + \frac{R}{L} \right) + \left(\frac{4K^2 i_0^2}{m L h_0^4} \right) s \right] \Delta H(s) = - \left(\frac{2K i_0}{m L h_0^3} \right) \Delta x U(s), \quad (20)$$

from which the transfer function that relates position to the voltage across the electromagnet is

$$G(s) = \frac{\Delta H(s)}{\Delta U(s)} = \frac{- \left(\frac{2K i_0}{m L h_0^3} \right)}{\left(s^2 - \left(\frac{2K i_0^2}{m h_0^3} \right) \right) \left(s + \frac{R}{L} \right) + \left(\frac{4K^2 i_0^2}{m L h_0^4} \right) s} =$$

$$= \frac{- \left(\frac{2K i_0}{m L h_0^3} \right)}{s^3 + \left(\frac{R}{L} \right) s^2 + \left(\frac{2K i_0^2 (2K - L h_0)}{m L h_0^3} \right) s - \left(\frac{2K R i_0^2}{m L h_0^3} \right)}. \quad (21)$$

2) *Nonlinear State-Space:* The rate of change in the state of the magnetic levitation system is a nonlinear function of the current state and input given by $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t))$. Let us now choose the states of the system $\mathbf{x}(t) = [x_1(t) \ x_2(t) \ x_3(t)]^T$ as the position and velocity of the magnet and the current in the coil, that is $\mathbf{x}(t) = [h(t) \ \dot{h}(t) \ i(t)]^T$. Breaking down the mechanical equation (14a) into first-order equivalents and including the electrical equation (14b) results in the nonlinear state-space representation given by

$$\dot{x}_1(t) = x_2(t), \quad (22a)$$

$$\dot{x}_2(t) = g - \frac{K x_3(t)^2}{m x_1(t)^2}, \quad (22b)$$

$$\dot{x}_3(t) = \frac{2K}{L} \frac{x_2(t) x_3(t)}{x_1(t)^2} - \frac{1}{L} R x_3(t) + \frac{1}{L} u(t). \quad (22c)$$

3) *Linearized State-Space:* We may linearize the nonlinear state-space model (22) around the expected working setpoint to obtain a perturbation model of states, that is $\Delta \mathbf{x}(t) = [\Delta x_1(t) \ \Delta x_2(t) \ \Delta x_3(t)]^T$, or in our case taking a perturbation of position, velocity and current from the linearization point as $\Delta \mathbf{x}(t) = [\Delta h(t) \ \Delta \dot{h}(t) \ \Delta i(t)]^T$, resulting in [25]:

$$\Delta \dot{x}_1(t) = \Delta x_2(t), \quad (23a)$$

$$\Delta \dot{x}_2(t) = \frac{2K}{m} \frac{x_{3(0)}^2}{x_{1(0)}^3} \Delta x_1(t) - \frac{2K}{m} \frac{x_{3(0)}}{x_{1(0)}^2} \Delta x_3(t), \quad (23b)$$

$$\Delta \dot{x}_3(t) = \frac{2K x_{3(0)}}{L x_{1(0)}^2} \Delta x_2 - \frac{1}{L} R \Delta x_3(t) + \frac{1}{L} \Delta u(t), \quad (23c)$$

or in a compact matrix form given by

$$\Delta \dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2K}{m} \frac{x_{3(0)}^2(t)}{x_{1(0)}^3(t)} & 0 & -\frac{2K}{m} \frac{x_{3(0)}(t)}{x_{1(0)}^2(t)} \\ 0 & \frac{2K x_{3(0)}(t)}{L x_{1(0)}^2(t)} & -\frac{1}{L} R \end{bmatrix} \Delta \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \Delta u(t), \quad (24)$$

where $x_{1(0)} = h_0$ and $x_{3(0)} = i_0$ are the linearization points for the perturbation model.

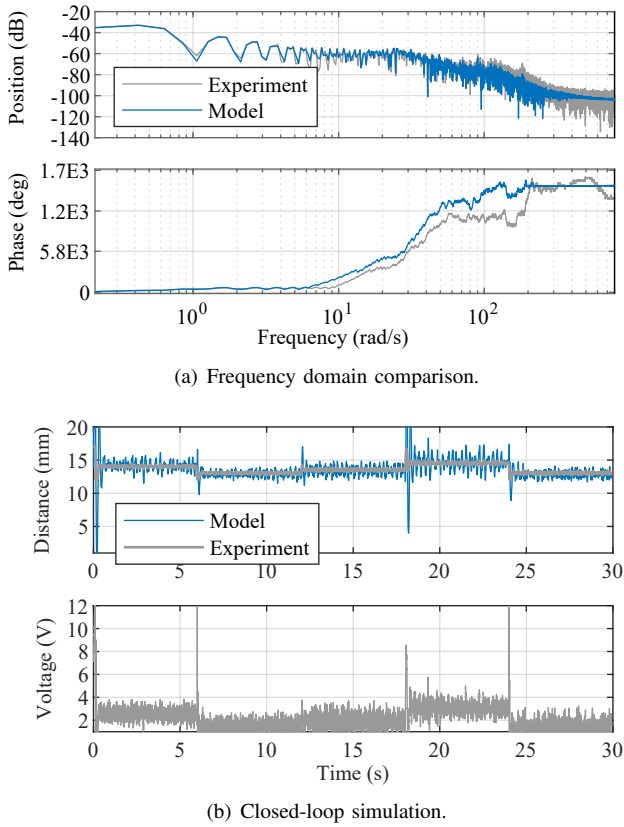


Fig. 8. Comparison of the transfer function model to experimental data.

B. System Identification

1) *Data Acquisition Experiment*: The system is open-loop unstable, therefore experimental data has been gathered in a closed loop using a PID controller. The discrete-time controller was set to $K_P = 2.2$, $T_I = 0.1$ and $T_D = 0.02$, with a $T_s = 4$ ms sampling period (see Sect. IV-C). Note that, including serial data communication, the microcontroller on the Arduino Uno could not reach real-time sampling of $T_s = 3$ ms, therefore this sampling is close to the limits of the hardware. A reference vector of $\mathbf{r} = [14.0 \ 13.0 \ 13.5 \ 14.5 \ 13.0]^T$ has been followed, each section for a thousand samples. In order to create a rich probe signal, the inputs computed by the controller were superimposed by a 1.5 V pseudo-random signal generated in the device. The resulting input and corresponding output has been logged. The experiment setup, data and the following identification results are available as a part of the API [24].

2) *Transfer Function Parameter Identification*: The sample data have been converted to a perturbation format with the linearization point chosen as $h_0 = 13.6$ mm and $u_0 = 2.1$ V, and subsequently transformed to the frequency domain. Based on laboratory measurements the mass of the magnet was initialized as $m = 0.76\text{E-}3$ kg, inductance of the coil $L = 0.239$ H, its resistance $R = 198.3$ Ω . The magnetic force constant was assumed as $K = 5\text{E-}3$. By taking the transfer

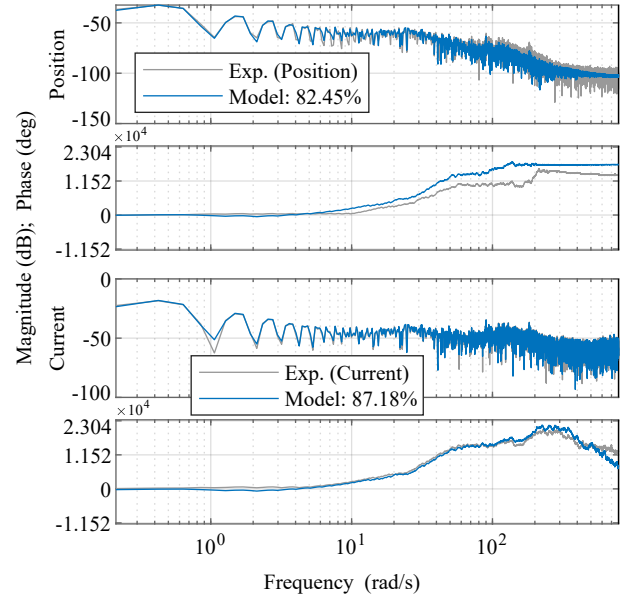


Fig. 9. Model and experimental data in the frequency domain for measured outputs.

function in Eq. (21), the initial guess is

$$G(s) = \frac{\Delta H(s)}{\Delta U(s)} = \frac{-3033}{s^3 + 829.7s^2 + 1126s - 4.52\text{E}5}$$

This initial model has been fitted to the experimental data using the MATLAB System Identification Toolbox. The identification procedure converged reliably to a solution after 12 iterations, resulting in the following transfer function:

$$G(s) = \frac{\Delta H(s)}{\Delta U(s)} = \frac{-332.6}{s^3 + 200.8s^2 + 1297s - 3.977\text{E}5}$$

Comparison of the resulting transfer function to the experimental data is shown in Fig. 8(a). The transfer function provides a decent match to experimental data and this is confirmed by a 83.3 % fit to estimation data by the normalized root mean squared error (NRMSE) metric, with a $1.07\text{E-}08$ m² mean squared error (MSE) and $1.07\text{E-}08$ Akaike's final prediction error (FPE). Figure 8(b) shows a closed-loop simulation of the model response in the time domain. As it is evident, the linearized model over-values the random noise in the probe signal, which in reality is much less stated.

3) *Linearized State-Space Representation and Parameter Estimates*: To simplify further notation, we take the linear perturbation model (24) as

$$\Delta \dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ \alpha & 0 & -\beta \\ 0 & \gamma & -\delta \end{bmatrix} \Delta \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ \epsilon \end{bmatrix} \Delta u(t), \quad (25)$$

TABLE III
INITIAL GUESS AND ESTIMATE OF VARIOUS MODEL PARAMETERS

Symbol	Measured or initialized	Identified	Unit
m	7.60E-04	-	kg
K	2E-6, 5E-3	1.6E-6, 4.9E-5	Nm ² A ⁻²
L	0.239	0.175	H
R	198	236	Ω
α	6300	1853	Nm ⁻²
β	46.4	365.1	NAm ⁻¹
γ	7.24E-04	2.537	NA ⁻¹ H ⁻¹
δ	829.7	1345	Ω H ⁻¹
η	4.184	5.71	H ⁻¹

where

$$\alpha = \frac{2K x_{3(0)}^2(t)}{m x_{1(0)}^3(t)}, \quad \beta = \frac{2K x_{3(0)}(t)}{m x_{1(0)}^2(t)},$$

$$\gamma = \frac{2K x_{3(0)}(t)}{L x_{1(0)}^2(t)}, \quad \delta = \frac{1}{L} R,$$

$$\epsilon = \frac{1}{L}.$$

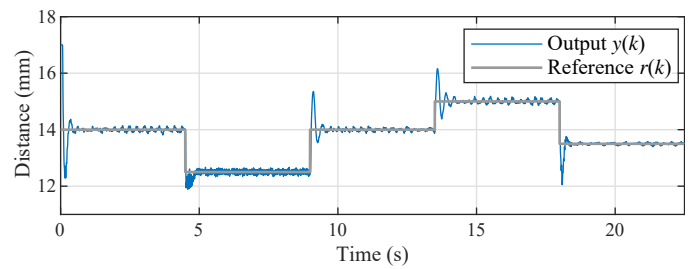
Initializing this model with the previously listed parameters and $K = 2\text{E-}6$ results in an initial guess of the parameters: $\alpha = 9063$, $\beta = 59.13$, $\gamma = 9.221\text{E-}4$, $\delta = 829.7$, $\eta = 4.184$.

The linearized state-space model parametrized with the five unknown parameters was identified from the same frequency domain data given before. The only difference here is that the current sensor is used to gather dynamic current measurements at each sample. The parameters converged robustly to a solution after 7 iterations, resulting in the model parameters: $\alpha = 1853$, $\beta = 365.1$, $\gamma = 2.537$, $\delta = 1345$, $\eta = 5.71$. Note that the initial guess is quite accurate, except for the γ parameter. This is because γ relies on the variable inductance model and that the MATLAB System Identification Toolbox cannot enforce parameter equality constraints. The K parameter guess cannot be reliably measured or guessed, therefore we get two estimates: 1.6E-06 and 4.9E-05. A dynamically equivalent resistance and inductance for the linearized model can be reliably computed as $R = 236 \Omega$ and $L = 0.175 \text{ H}$, which strongly correlates to instrument measured values. The identified model parameters are summarized in Tab. III as well.

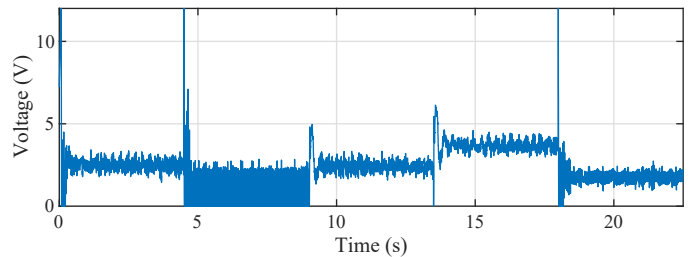
Figure 9 shows the comparison of the model outputs with experimental measurements in the frequency-domain. The model fit to measurement according to NRMSE is 82% for the position measurement and 87% for the current signal. For the overall metrics this is MSE 2.9E-07 and FPE 3.0E-15, suggesting an adequate model that is suited for simulation, estimator and observer design, and model-based control.

C. PID Control

In addition to the hardware interfaces, the AutomationShield library contains a PID controller method as well. This PID controller assumes the absolute formulation, with hard input saturation constraints and integral windup protection by clamping. Students shall be encouraged to formulate their



(a) Magnetic disc position.



(b) Input voltage.

Fig. 10. Closed-loop control using PID.

own—possibly more advanced—control algorithms, but here we shall rely only on our pre-programmed routines for demonstration.

A $T_s = 3.2 \text{ ms}$ sampling period is assumed for closed-loop control. The external reference potentiometer may be engaged to set the desired position of the levitated magnet, here, we chose to pre-set the reference trajectory to $\mathbf{r} = [14.0 \ 12.5 \ 14.0 \ 15.0 \ 13.5]^T \text{ mm}$. The hard input saturation was set to 0–12 V, while the integrator clamp to 0–20 mm. The PID controller was tuned empirically to $K_P = 2.3$, $T_I = 0.1$ and $T_D = 0.03$ for best results, this procedure has been confirmed by repeated experiments accounting for the integral sum of the control error.

A typical and robustly repeatable closed-loop experiment is shown in Fig. 10. At the beginning of the experiment, the permanent magnet laying at rest is quickly pulled up close to the level of the solenoid, then settles on the first reference level $r = 14 \text{ mm}$ with some overshoot (Fig. 10(a)). Other reference settings are followed closely. The levitation of the object along with the changes of reference levels are pronounced and visually engaging. The corresponding input signal is shown in Fig. 10(b). The nonlinear nature of its closed-loop dynamics can be hypothetically further improved by better—possibly natively nonlinear—control methods.

V. CONCLUSION

We have proposed a reference design for a prototype of a low-cost magnetic levitation experiment that can be used for control engineering and mechatronics courses with potential in academic research as well. The hardware design is open-source and uses electronic and mechanical parts that are globally and easily available. This, in combination with the material cost kept under \$10, means that the MagnetoShield

is an affordable laboratory aid for universities and students alike. The open-source application programming interface aids the learning process by taking care of the input and output functionality. A comprehensive system identification problem and a basic closed-loop experiment demonstrate the versatility of our design.

We have identified several hardware and software improvements that shall form a basis for future enhancements to this concept while working on the demonstration examples featured here. The MOSFET of our choice severely reduces the effective resolution of the DAC, while its gate-source characteristics are inconsistent among individual components as well. This means that each shield must be individually calibrated before use. In future versions the input chain shall be upgraded to gain consistency amongst hardware, possibly also using a higher resolution DAC chip. With regards to software, an API for Simulink would certainly improve classroom usability.

REFERENCES

- [1] C. I. Muresan, C. Ionescu, S. Folea, and R. De Keyser, "Fractional order control of unstable processes: the magnetic levitation study case," *Nonlinear Dynamics*, vol. 80, no. 4, pp. 1761–1772, 2015. [Online]. Available: <https://doi.org/10.1007/s11071-014-1335-z>
- [2] B. Bidikli and A. Bayrak, "A self-tuning robust full-state feedback control design for the magnetic levitation system," *Control Engineering Practice*, vol. 78, pp. 175–185, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0967066118302028>
- [3] S. K. Pradhan and B. Subudhi, "Nonlinear control of a magnetic levitation system using a new input-output feedback linearization," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 332–336, 2016, 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896316300751>
- [4] D. Khimani, S. Karnik, and M. Patil, "Implementation of high performance nonlinear feedback control on magnetic levitation system," *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 13 – 18, 2018, 5th IFAC Conference on Advances in Control and Optimization of Dynamical Systems. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896318301630>
- [5] G. S. Seyboth and F. Allgöwer, "Synchronized model matching: a novel approach to cooperative control of nonlinear multi-agent systems," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1985 – 1990, 2014, 19th IFAC World Congress. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S147466701641904X>
- [6] P. Horáček, "Laboratory experiments for control theory courses: A survey," *Annual Reviews in Control*, vol. 24, pp. 151 – 162, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1367578800900294>
- [7] J. de Jesús Rubio, L. Zhang, E. Lughofer, P. Cruz, A. Alsaedi, and T. Hayat, "Modeling and control with neural networks for a magnetic levitation system," *Neurocomputing*, vol. 227, pp. 113 – 121, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231216313674>
- [8] M. Klaučo, M. Kalúz, and M. Kvasnica, "Real-time implementation of an explicit mpc-based reference governor for control of a magnetic levitation system," *Control Engineering Practice*, vol. 60, pp. 99 – 105, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0967066117300011>
- [9] S. Y. Oh and H. L. Choi, "Robust observer-based output feedback controller for nonlinear systems with uncertain triangular and nontriangular nonlinearities and diagonal terms," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 4, pp. 1182–1200, 2019. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.4425>
- [10] H. N. Truong and X. K. Ngo, "Tuning parameters of fuzzy logic controller using PSO for Maglev system," *International Journal of Computer Applications*, vol. 178, no. 18, 09 2019.
- [11] J.-H. Li and T.-H. S. Li, "Multiloop control of thyristor-driven magnetic levitation system," *Mechatronics*, vol. 5, no. 5, pp. 469–481, 1995. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/095741589500018Z>
- [12] W. Yu and X. Li, "A magnetic levitation system for advanced control education," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 9032 – 9037, 2014, 19th IFAC World Congress. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474667016430399>
- [13] T. H. Wong, "Design of a magnetic levitation control system: An undergraduate project," *IEEE Transactions on Education*, vol. E-29, no. 4, pp. 196–200, Nov 1986.
- [14] G. P. Grodzicki, "Postgraduate magnetic levitation experience," in *2011 Australian Control Conference*, Nov 2011, pp. 130–135.
- [15] U. Sadek, A. Sarjaš, A. Chowdhury, and R. Svečko, "Improved adaptive fuzzy backstepping control of a magnetic levitation system based on symbiotic organism search," *Applied Soft Computing*, vol. 56, pp. 19 – 33, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494617301229>
- [16] M. H. Yaseen and H. J. Abd, "Modeling and control for a magnetic levitation system based on SIMLAB platform in real time," *Results in Physics*, vol. 8, pp. 153 – 159, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2211379717320065>
- [17] W. Wei, W. Xue, and D. Li, "On disturbance rejection in magnetic levitation," *Control Engineering Practice*, vol. 82, pp. 24 – 35, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0967066118305653>
- [18] K. A. Lilienkamp, "Low-cost magnetic levitation project kits for teaching feedback system design," in *Proceedings of the 2004 American Control Conference*, vol. 2, 2004, pp. 1308–1313 vol.2.
- [19] K. H. Lundberg, K. A. Lilienkamp, and G. Marsden, "Low-cost magnetic levitation project kits," *IEEE Control Systems Magazine*, vol. 24, no. 5, pp. 65–69, Oct 2004.
- [20] J. I. Artigas, L. A. Barragán, S. Llorente, A. Marco, and O. Lucía, "Low-cost magnetic levitation system for electronics learning," in *2010 4th IEEE International Conference on E-Learning in Industrial Electronics*, 2010, pp. 55–60.
- [21] G. Takács, M. Gulan, J. Bavlina, R. Köplinger, M. Kováč, E. Mikuláš, S. Zarghoon, and R. Salíni, "HeatShield: a low-cost didactic device for control education simulating 3D printer heater blocks," in *Proceedings of the 2019 IEEE Global Engineering Education Conference*, Dubai, United Arab Emirates, April 2019, pp. 374–383.
- [22] G. Takács, T. Konkoly, and M. Gulan, "Optoshield: A low-cost tool for control and mechatronics education," in *Proceedings of the 12th Asian Control Conference*, Kitakyushu-shi, Japan, Jun 2019, pp. 1001–1006.
- [23] G. Takács, P. Chmurčiak, M. Gulan, E. Mikuláš, J. Kulhánek, G. Penzinger, M. Vdoleček, M. Podbielančík, M. Lučan, P. Šálka, and D. Šroba, "FloatShield: An Open Source Air Levitation Device for Control Engineering Education," in *Proceedings of the 21st IFAC World Congress*, Berlin, Germany, July 2020, pp. 1–7, preprint submitted for review.
- [24] G. Takács *et al.*, "AutomationShield: Control Systems Engineering Education," Online, 2019, [cited 3.12.2019]; Available from <http://www.automationshield.com>.
- [25] W. Barie and J. Chiasson, "Linear and nonlinear state-space controllers for magnetic levitation," *International Journal of Systems Science*, vol. 27, no. 11, pp. 1153–1163, 1996. [Online]. Available: <https://doi.org/10.1080/00207729608929322>