

Modeling and Control Design of an Educational Magnetic Levitation System

Anton Haumer¹

¹ Faculty of Electrical Engineering and Information Technology, Ostbayerische Technische Hochschule Regensburg, Germany, anton.haumer@oth-regensburg.de

Abstract

A magnetic levitation system is a perfect educational example of a nonlinear unstable system. Only with suitable control, a small permanent magnet can be held floating stable below a coil. After modeling and simulation of the system, control of the system can be developed. At the end, the control algorithm can be coded on a microcontroller, connected to a pilot plant.

Keywords: mechatronics, magnetic levitation, time-discrete control, functional mockup interface

1 Introduction

Using a ready-to-use system like Zeltom's depicted in **Figure 1** allows to set the focus on the development of the control algorithm and gives quick results.

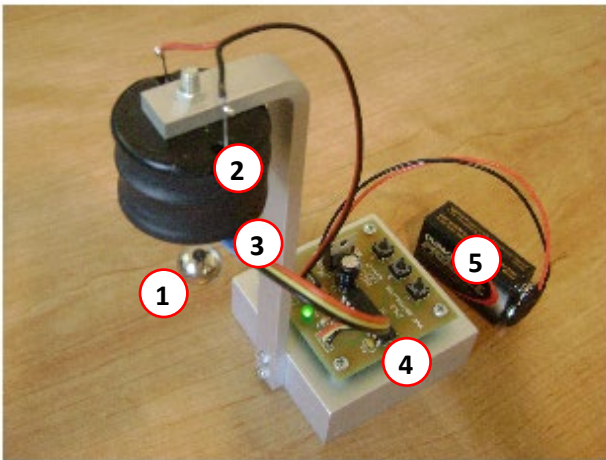


Figure 1 Magnetic Levitation System
<https://www.zeltom.com/emls.html>

1.1 Description of the System

Zeltom's system (**Figure 1**) consists of:

1. Permanent magnet (disc or sphere)
2. Coil (with iron core)
3. Hall effect sensor (below coil)
4. PCB with controller and power electronics
5. Voltage source (9 V battery)

The permanent magnet is attracted by the iron core of the magnet and the magnetic field excited by the current flowing through the coil. Gravitational force acts in opposite direction. Inverting the coil current wouldn't result in repelling the magnet, but causes the magnet to flip by 180°.

The magnet's position is detected by a Hall effect sensor, placed at the bottom of the coil. Not only does the magnetic field of the permanent magnet affect the Hall effect sensor, but the magnetic field caused by the coil current, too.

1.2 Equations of the System

The transient behavior of the coil current is described by Equation 1:

$$v = R \cdot i + L \cdot \frac{di}{dt} \quad (1)$$

Induced voltage due to the moving permanent magnet can be neglected.

The equation of motion of the permanent magnet can be written as Equation 2:

$$m \cdot \ddot{d} = f(d, i) - m \cdot g \quad (2)$$

The position of the magnet is measured along the d-axis in upward direction as shown in **Figure 2**. Position $d = 0$ is located at the bottom of the coil. Therefore, only positions of the permanent magnet along the negative half of the d-axis are meaningful.

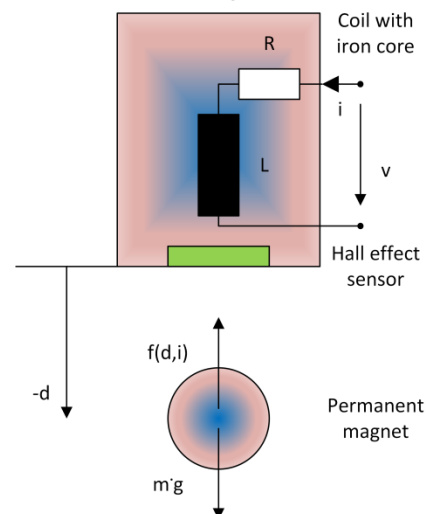


Figure 2 Illustration of the system

The force between coil and magnet is strongly nonlinear and either the function has to be determined by finite element simulations of the magnetic field or by measurements at the real system. Fortunately Zeltom has performed such measurements and provides a set of equations by (Zeltom 2009) - Equation 3 for the force:

$$f(d, i) = k \cdot \frac{i}{d^4} \quad (3)$$

Unfortunately Equation 3 doesn't take into account the force between the permanent magnet and the iron core even in absence of coil current $i = 0$. Although the influence on control around the equilibrium is small, the equation can be enhanced as shown in Equation 4:

$$f(d, i) = k' \cdot \frac{i_C + i}{d^4} \quad (4)$$

Equation 4 introduces a new parameter i_C which describes the force between the permanent magnet and the iron core and adapts parameter k such way that the original equilibrium at position d_0 with coil current i_0 described by Equation 5 remains unchanged:

$$k \cdot \frac{i_0}{d_0^4} = k' \cdot \frac{i_C + i_0}{d_0^4} = m \cdot g \quad (5)$$

Specifying the position d_C where the force between the permanent magnet and the iron core just meets the gravitational force (Equation 6) allows calculation of the new parameters (Equations 7 and 8):

$$k' \cdot \frac{i_C}{d_C^4} = m \cdot g \quad (6)$$

$$k' = k \cdot \left(1 - \frac{d_C^4}{d_0^4}\right) \quad (7)$$

$$i_C = \frac{m \cdot g}{k'} \cdot d_C^4 \quad (8)$$

The output of the Hall effect sensor has been approximated by Zeltom in Equation 9:

$$e = \alpha + \frac{\beta}{d^2} + \gamma \cdot i \quad (9)$$

This voltage e will be corrupted by some noise. The influence of this effect can easily be investigated by adding a noise signal from `Modelica.Blocks.Noise` as described by (Klöckner 2014).

The parameters of the system are summarized in **Table 1** and taken from (Zeltom 2009) as well as (Thiele 2019).

Equilibrium is investigated with a stationary model, setting all derivatives in Equations 1 and 2 to zero, additionally exploiting Equation 4 (and 9, if the output of the Hall effect sensor is of interest).

Figure 3 compares the steady-state characteristic $i_0 = f(d_0)$ according to Equation 3 (blue) and 4 (red, dashed). Bear in mind that equilibrium for these points of operation is unstable without appropriate control.

Table 1 Parameters of the system

Parameter	Value	Unit
R	2.41	Ω
L	15.03	mH
m	3.02	g
k	$1.731 \cdot 10^{-8}$	$\text{N} \cdot \frac{\text{m}^4}{\text{A}}$
d_0	-2	cm
$\rightarrow i_0$	273.75	mA
d_C	-1	cm
$\rightarrow i_C$	18.25	mA
$\rightarrow k'$	$1.623 \cdot 10^{-8}$	$\text{N} \cdot \frac{\text{m}^4}{\text{A}}$
α	2.48	V
β	0.292	$\text{mV} \cdot \text{m}^2$
γ	0.48	S

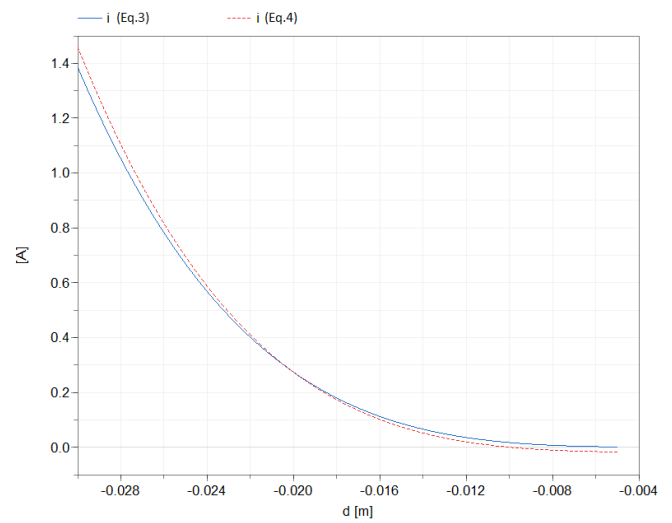


Figure 3 Steady-state characteristic of the system

2 Controller Design

Splitting the system into three control loops allows simple and stable control, similar to an electric drive:

- A series connection of resistor and inductor, with or without induced voltage (Equation 1).
- The current causes force (Equation 4). The force accelerates the mass (Equation 2). Integral of acceleration gives velocity.
- Integral of velocity gives position.

Table 2 Comparison between DCPM and MagLev

DCPM	MagLev
$v_A = R_A \cdot i_A + L_A \cdot \frac{di_A}{dt} + v_i$	$v = R \cdot i + L \cdot \frac{di}{dt}$
$v_i = k\phi \cdot \omega$	$v_i \approx 0$
$\tau = k\phi \cdot i_A$	$f = f(i, d)$
$J \cdot \frac{d\omega}{dt} = \tau - \tau_L$	$m \cdot \ddot{d} = f - m \cdot g$
$\omega = \frac{d\phi}{dt}$	$d = \int \dot{d} \cdot dt$

Table 2 shows a comparison between a DC machine excited by permanent magnets and the magnetic levitation system under investigation. Despite the fact that the force formula is nonlinear and the velocity is not directly accessible, the equations are equivalent.

According to (Schröder 2020) such a system can be controlled using cascaded control. **Figure 14** shows a block diagram of the whole system including control.

2.1 Current Controller

Input: Current control error

Output: Reference voltage

Taking into account the dead time caused by the time discrete communication between controller and hardware based on switching frequency $T_d = 1/f_{switch}$, the controllers can be designed as for continuous control. The dead time is replaced by the first element of its series expansion, i.e. a first order delay.

Optimal control for the innermost control loop results in a PI controller, parameterized according to the magnitude optimum (Dierk Schröder 2020, Equation 10 and 11):

$$k_p = \frac{L}{2 \cdot T_d} \quad (10)$$

$$T_i = \frac{L}{R} \quad (11)$$

The output of the controller is limited to the source voltage (battery), therefore an anti-windup measure has to be implemented.

The transfer function of the closed loop can be approximated by a first order delay with substitute time constant $T_{sub} = 2 \cdot T_d$.

2.2 Speed Controller

Input: Speed control error

Output: Reference force \rightarrow Reference current

Assume the actual position d is known, we can establish a transformation from force to current inverting Equation 4. The output of the speed controller is limited by a force dependent on the maximum admissible current. Therefore an anti-windup measure has to be implemented. Using the actual position d , the maximum admissible force can be calculated from the maximum admissible current using Equation 4.

Optimal control of the speed control loop results in a PI controller, parameterized according to the symmetrical optimum (Dierk Schröder 2020, Equation 12 and 13):

$$k_p = \frac{m}{2 \cdot T_{sub}} \quad (12)$$

$$T_i = 4 \cdot T_{sub} \quad (13)$$

Furthermore, a low-pass filter for the reference speed has to be implemented. This fulfills the physical law that a step in speed would require infinite force.

2.3 Position Controller

Input: Position control error

Output: Reference speed

A simple P-controller should be sufficient, since the speed controlled system subsequently integrates speed to position, which guarantees accuracy without permanent deviation between reference and actual position. It is possible to define an upper limit for the proportional gain (Equation 14) to avoid overshooting position. In reality we will have to reduce proportional gain to find the optimal setting.

$$k_p \leq \frac{1}{16 \cdot T_{sub}} \quad (14)$$

2.4 Observer

Since neither position nor velocity is measured directly, both of them have to be calculated from the output of the Hall effect sensor, using some sort of observer. Using the measured current, we can evaluate Equation 15 after inverting Equation 9, neglecting noise:

$$d = -\sqrt{\frac{\beta}{e - \alpha - \gamma \cdot i}} \quad (15)$$

Velocity is the derivative of position.

2.5 Time Discrete Control

According to (Latzel 1995), the controller blocks may be transformed from a continuous version to a time discrete version without changing the parameterization, as long as the sample period – which is chosen as the inverse of the switching frequency – is short compared to the system's time constants. The shortest time constant is the coil's time constant. Using a switching frequency of 1 kHz (or higher) this constraint (Equation 16) is fulfilled:

$$\frac{1}{f_{switch}} = 1 \text{ ms} \ll \frac{L}{R} = 6.23 \text{ ms} \quad (16)$$

All controller tasks are triggered once per sample period, but additionally the desired sequence has to be kept:

- A/D-conversion (sample)
- Position controller
- Speed controller
- Current controller
- D/A-conversion (hold)

Two versions of the control blocks have been implemented, a triggered and a clocked one.

The triggered blocks (**Figure 15**) use slightly time shifted triggers to guarantee the order of execution of the blocks.

The clocked blocks (Figure 16) based on Modelica.Clocked (Otter 2012) are executed within the same clock partition. The order of execution relies on the tool, sorting the blocks according to the signal flow.

3 The MagLev Library

The library developed by the author contains examples, components, DC/DC-converter and control blocks. It is available on github under the BSD 3-Clause Revised License:

<https://github.com/AHaumer/MagLev>

The complete system model is depicted in Figure 15 and Figure 16. The discrete block e2d calculates position from the output of the Hall effect sensor and the actual sampled current (Equation 15) and performs a time discrete differentiation of the position to obtain velocity.

The discrete block adda samples sensed and holds actuating variables with $sampleTime = \frac{1}{f_{switch}}$:

- ← Source voltage
- ← Coil current
- → Reference voltage

When the control part (grey background in Figure 15) communicates with a co-simulation FMU of the system (light blue background in Figure 15) this block can be omitted.

All parameters are summarized in a parameter record which also calculates steady state position and controller parameter. This ensures consistent parameterization of all components and makes it easy to switch to a different system, e.g. Zeltom offers additionally a MagLevPlus system with an enhanced coil providing higher force acting on the magnet.

3.1 Components

This sub-library contains the coil and the magnet, modeled in an object-oriented style.

The coil has an additional translational mechanical connector to determine the magnet’s position and provide the force caused by the magnetic field of the coil current, as shown in Figure 4. This force is calculated according to Equation 4. Additionally, Equation 9 is evaluated to determine the output signal of the Hall effect sensor, including noise which is modeled using Modelica.Blocks.Noise (Klößner2014).

The magnet model is a simple point mass whose vertical acceleration is determined by the magnetic force provided by the translational mechanical connector and the gravitational force.

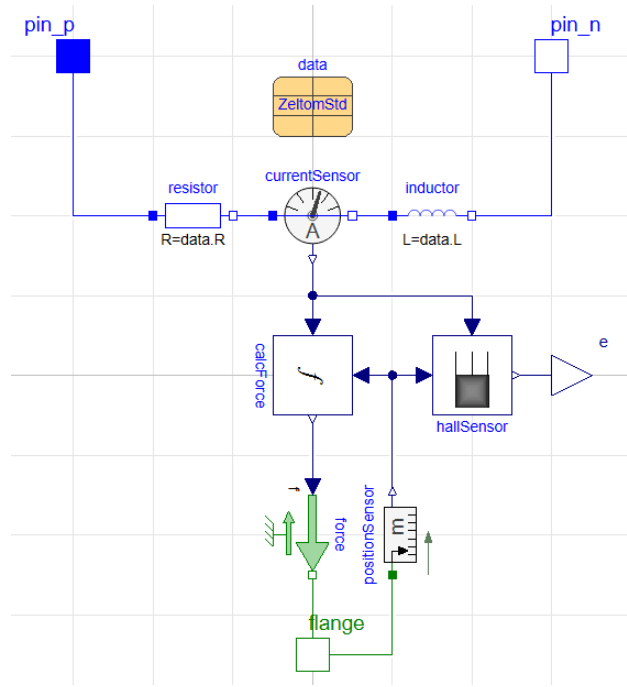


Figure 4 Model of the coil

3.2 DC/DC-Converter

The DC/DC-converter is provided as an averaging implementation and a switching version. The averaging version avoids switching effects and provides higher performance. For a detailed proof of concept the switching version is used.

The averaging version shown in Figure 5 just prescribes the reference voltage to the output and pulls current from the input such way that power on both sides is equal. This is achieved with a fast integral controller. Of course reference voltage is limited between actual source voltage and zero in case of a buck converter.

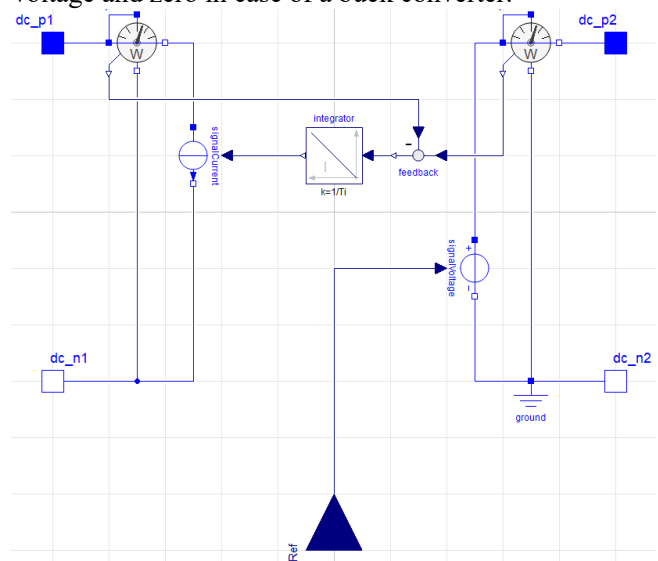


Figure 5 Averaging DC/DC-converter

The switching H-bridge shown in **Figure 6** has the same functionality as the model implemented in `Modelica.Electrical.PowerConverters.DCDC` but modeled with a different layout. It is used as a buck converter. The `SignalPWM` is taken from `Modelica.Electrical.PowerConverters.DCDC` but is adapted by replacing the sawtooth PWM reference signal by a triangular one. This allows to sample the current at the beginning (or in the middle) of the switching period, finding the average of the current with good accuracy as shown in **Figure 7**. The block C in the controller time line designates the time span to execute the control algorithm.

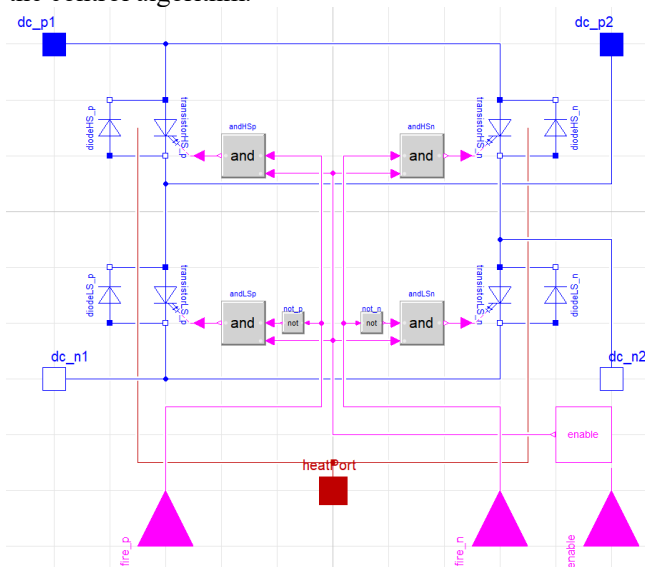


Figure 6 Switching DC/DC-converter (H-bridge)

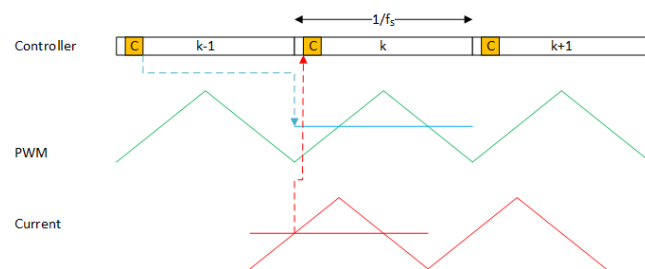


Figure 7 Timing: Switching period and PWM

3.3 Control Blocks

For the anti-windup measure necessary in both the continuous and time discrete PI controller, two solutions have been implemented:

- Back-Calculation: The difference between limited and unlimited output multiplied by the inverse of the proportional gain is subtracted from the integrator's input.
- Clamped: The integral part is stopped when the output exceeds the limit.

The observer block `e2d` implements Equation 15 and takes an approximate derivative.

The continuous version of the limited PI controller is taken from `Modelica.Electrical.Machines.Examples.ControlledDCDrives.Utilities` with some enhancements. The equations of the continuous PI controller do not avoid iteration.

The time discrete version implements an equivalent algorithm as shown in **Listing 1**, using an explicit (forward) Euler. Iteration is avoided by calculating a prediction of the unlimited output. This function is called both from the triggered block and the clocked block within when-clauses.

Listing 1 Time discrete PI controller

```
function piStep
  input Real u "Reference signal";
  input Real u_m "Measured signal";
  input Real kp "Proportional gain";
  input SI.Time Ti "Integral time constant";
  input SI.Time Ts "Sample period";
  input Real kFF "Gain of feed-forward";
  input Real ff "Feed-forward signal";
  input AntiWindup antiWindup;
  input Real yMin "Lower limit of output";
  input Real yMax "Upper limit of output";
  input Real pre_x "Previous state";
  output Real x "State";
  output Real y "Result";
protected
  Real e "Control error";
  Real predict "Prediction of output ";
  Real cropped "Cropped part of output";
algorithm
  e := u - u_m;
  predict := kp*e + kp*(pre_x + Ts/Ti*e) +
    kFF*ff;
  cropped := predict -
    min(max(predict, yMin), yMax);
  x := pre_x + Ts/Ti*
    (if antiWindup == BackCalc then
      (e - cropped/kp)
    else (if abs(cropped) > small then 0
      else e));
  y := min(max(kp*e+kp*x+kFF*ff, yMin), yMax);
end piStep;
```

4 Simulation Results

Simulation of the time discrete position controlled system shows satisfactory results:

In **Figure 8** the initial position of the magnet is $d_0 = -2\text{ cm}$, the initial velocity is zero. The reference position is given by a trapezoid which is nearly a pulse series. The real position of the magnet follows the reference position very well. The observer's result `e2d.d` shows only negligible deviation from the real position taken from the magnet model.

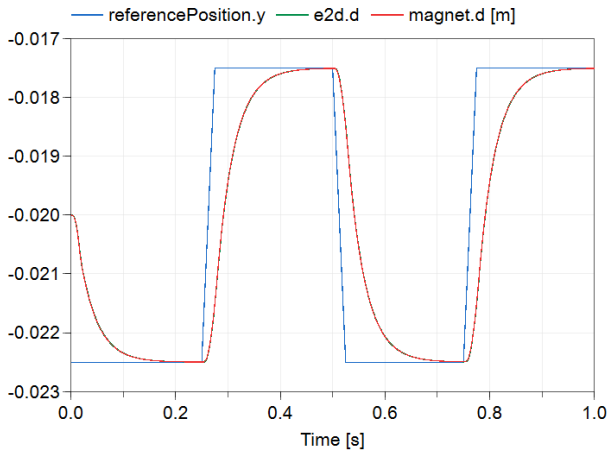


Figure 8 Time discrete position control: Position trajectory

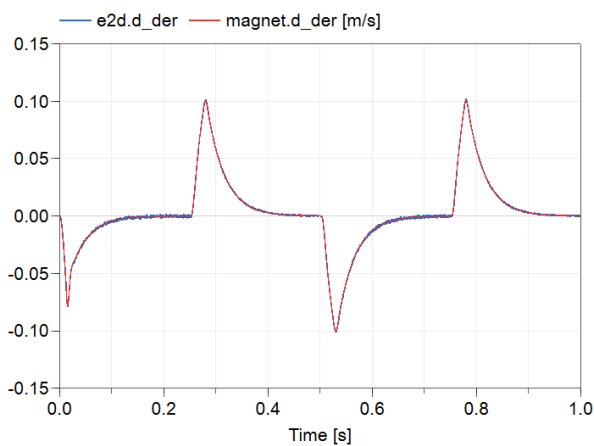


Figure 9 Time discrete position control: Speed trajectory

Figure 9 compares velocity obtained in the observer block with velocity in the magnet model. The influence of the noise added to the output signal of the Hall effect sensor can be seen at the trajectory of the velocity signal $e2d.d_der$ reconstructed by the observer. Thus it is possible to estimate the impact of noise on the control performance.

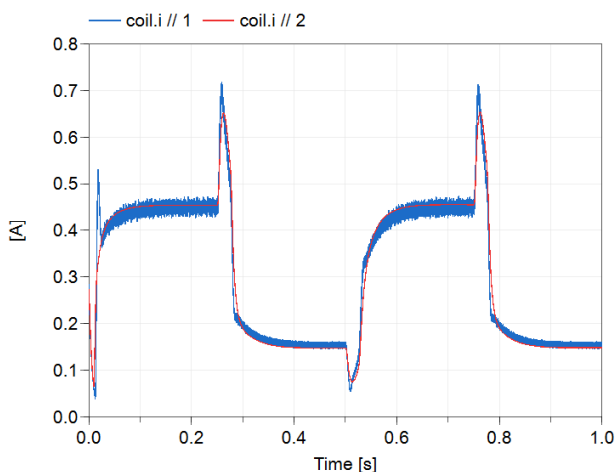


Figure 10 Coil current of discrete and continuous version

The coil current simulated with discrete control and switching DC/DC converter is shown in blue in **Figure 10**. The same figure shows the coil current obtained with continuous control and averaging DC/DC converter in red. The discrete results have been obtained using a switching frequency $f_{swtch} = 2.5 \text{ kHz}$. For the continuous results additionally noise of the Hall sensor signal has been set to zero.

Differences between the two trajectories:

- Switching versus averaging DC/DC-converter
- Time discrete versus continuous control
- Sample/hold versus first order delays
- Absence of noise in the continuous model

The continuous model neglects some effects but is capable of being simulated in real-time. It shows a good estimation of the behavior under control and the energy consumption from the source. The energy consumption over the shown cycle differs only by 0.6% from the discrete version.

Both the triggered and the clocked version of the discrete models provide a more detailed insight in the systems behavior, but the triggered version takes 7.5-times and the clocked version 6-times the as long as the continuous version to simulate.

5 Controller and Power Electronics

Zeltom’s printed circuit board (No. 4 in **Figure 1**) is replaced by a rapid prototyping system available in the lab:

The TI F28069M LaunchPad shown in **Figure 11** contains a TI C2000 μC and can be programmed either using a software tool provided by TI or utilizing Simulink.

The latter one has the advantage that the system model can be exported from Modelica as functional mockup unit. This FMU is subsequently imported to Simulink, the control algorithm can get thoroughly tested under realistic conditions before downloading it to the μC .

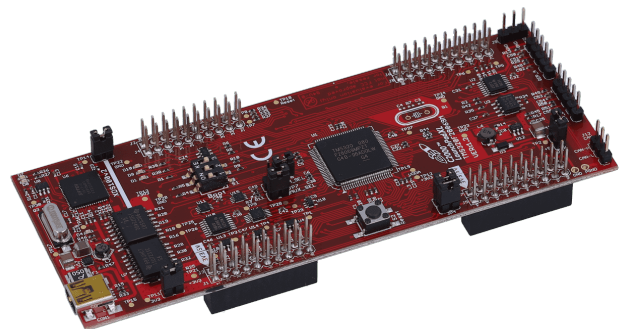


Figure 11 TI 28069M LaunchPad
<https://www.ti.com/tool/LAUNCHXL-F28069M>

The TI BoosterPack DRV8848 shown in **Figure 12** provides two H-bridges suitable for 4.18 V/1 A RMS. One half of one of the two H-bridges is used as a step-down converter to supply the coil with variable voltage.

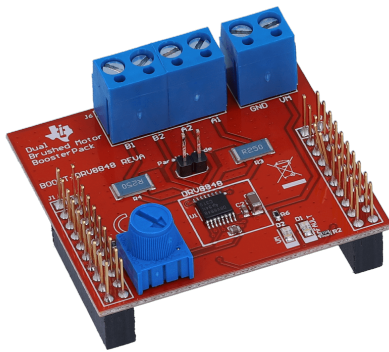


Figure 12 BoosterPack DRV8848
<https://www.ti.com/tool/BOOST-DRV8848>

For sure this choice of controller and DC/DC converter is overkill, but it is available in the lab and the workflow is well documented.

6 Conclusions and Outlook

This project proves the advantages of using Modelica, especially for educational purposes:

1. Develop a clearly arranged open source model of the physical system using Modelica.
2. Design a control concept, testing it with Modelica.
3. Export a functional mockup unit of the physical system from Modelica.
4. Import the FMU to Simulink. Implement the controller in Simulink and test it acting on the FMU.
5. Download the control algorithm to the embedded controller, proving the control concept in reality.

Steps 1 – 3 can be performed by a teacher, preparing a students' project.

A physical model of the magnetic levitation system has been implemented, as well as both averaging and switching versions of a DC/DC-converter.

For the concept of control a conventional cascaded control with current controller, speed controller and position controller has been chosen. Final simulations show satisfactory results.

Furthermore, the system model can be exported as functional mockup unit shown in **Figure 13**. This FMU can be imported in Simulink.

It is a students' project to implement the control structure in Simulink, subsequently downloading the control algorithm to the embedded controller. Thus the concept

can be proven: The magnet should follow the reference position signal.

Since the students' project is ongoing work, it is not sure that the magnetic levitation system with the new control board can be presented at the conference, but following the results of the Modelica simulations it can be expected to be a success.

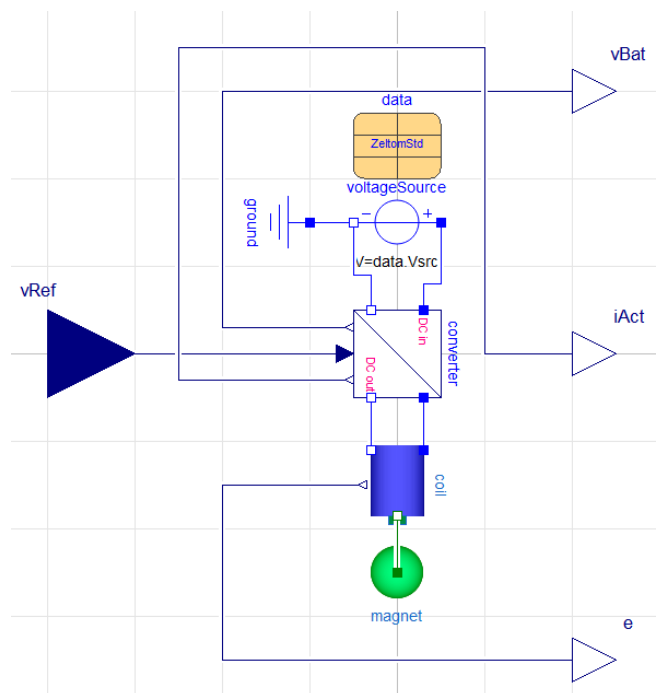


Figure 13 Functional mockup unit of the system

Acknowledgement

Without the discussions with Martin Otter it wouldn't have been possible to develop the discrete control blocks based on `Modelica.Clocked`.

References

- Hilding Elmquist, Martin Otter, Sven-Erik Mattsson (2012). "Fundamentals of Synchronous Control in Modelica". Proceedings of the 9th International Modelica Conference 2012. DOI:10.3384/ecp1207615.
- Andreas Klöckner, Franziskus L. J. van der Linden, Dirk Zimmer (2014). "Noise Generation for Continuous System Simulation". Proceedings of the 10th International Modelica Conference 2014, pp. 837-846. DOI 10.3384/ecp14096837.
- Wolfgang Latzel (1995). "Einführung in die digitalen Regelungen" (in German). VDI. ISBN 978-3-642-95779-6.
- Martin Otter, Bernhard Thiele, Sven-Erik Mattsson (2012). "A Library for Synchronous Control Systems in Modelica". Proceedings of the 9th International Modelica Conference 2012. DOI:10.3384/ecp1207627.
- Dierk Schröder (2020). "Elektrische Antriebe – Regelung von Antriebssystemen" (in German). 5th Edition. Springer Vieweg. ISBN 978-3-662-62700-6.

Bernhard Thiele, Bernt Lie, Martin Sjölund, Adrian Pop, Peter Fritzson (2019). “Controller Design for a Magnetic Levitation Kit using OpenModelica’s Integration with the Julia Language”. Proceedings of the 13th International Modelica Conference 2019, pp. 303-311. DOI 10.3384/ecp19157303.

Zeltom LLC (2009). “Electromagnetic Levitation System - Mathematical model”. URL: https://www.zeltom.com/documents/emls_md.pdf

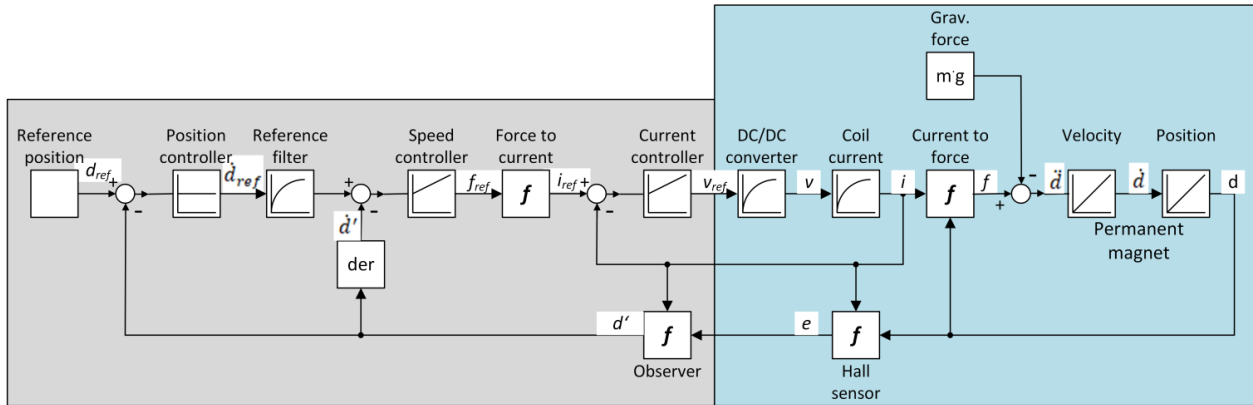


Figure 14 Block diagram of the system

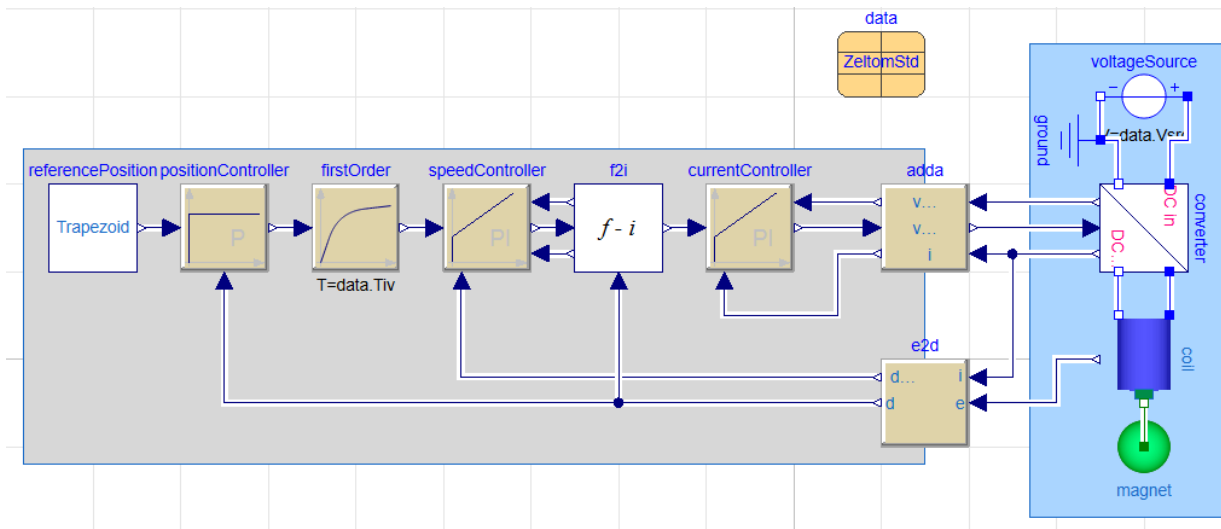


Figure 15 Complete System Model using triggered blocks

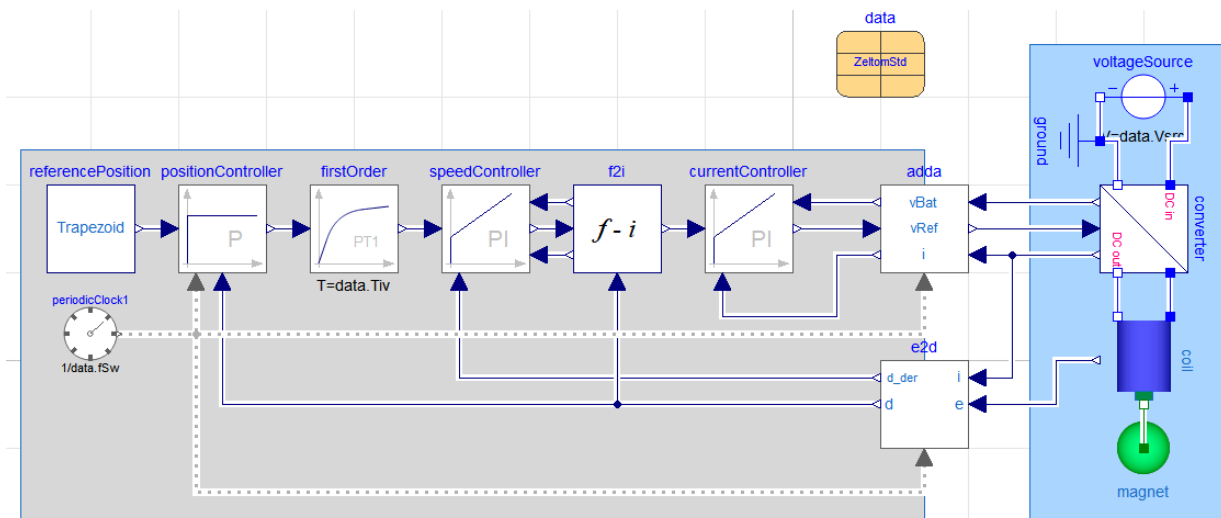


Figure 16 Complete System Model using clocked blocks