UNIVERSIDAD SANTO TOMÁS

MASTER THESIS

DEVELOPMENT OF A PLATFORM USING HARDWARE IN THE LOOP TECHNIQUE TO VALIDATE THE DYNAMIC BEHAVIOR OF A COUPLED-INDUCTOR BUCK-BOOST DC/DC CONVERTER

Author:
Eng. Jhon Erik Navarrete Gómez

Director:
Dr. Eng. Carlos Torres Pinzón

Supervisors:
M. Eng. Carlos Mojica Casallas
Dr. Eng. Harrynson Ramírez Murillo

A thesis submitted in fulfillment of the requirements for the degree of Master of Engineering

in the

MEM - Modeling, Electronics and Monitoring
Engineering Department

September, 2020

Declaration of Authorship

I, Jhon Erik Navarrete Gómez, declare that this thesis titled, "DEVELOPMENT OF A PLATFORM USING HARDWARE IN THE LOOP TECHNIQUE TO VALIDATE THE DYNAMIC BEHAVIOR OF A COUPLED-INDUCTOR BUCK-BOOST DC/DC CONVERTER" and the work presented in it are my own. I confirm that:

1. I did this work or while in candidature for a research degree at this University.
2. Where I have consulted the published work of others, I always attribute this.
3. Where I have quoted from the work of others, I always give the source. Except for such quotations, this thesis is my work.
4. I have acknowledged all key sources of help.
5. I base the thesis on work by myself with others; I have made clear what was done by others and what I have contributed myself.

Eng. Jhon E. Navarrete Gómez
Author

Likewise, the work mentioned above was revised and approved by:

Dr. Eng. Carlos Andrés Torres Pinzón
Director

Dr. Eng. Harrynson Ramírez Murillo
Supervisor

M. Eng. Carlos Javier Mojica Casallas
Supervisor

UNIVERSIDAD SANTO TOMÁS

Abstract

Faculty of Electronics Engineering
Engineering Department

Master of Engineering

DEVELOPMENT OF A PLATFORM USING HARDWARE IN THE LOOP TECHNIQUE TO VALIDATE THE DYNAMIC BEHAVIOR OF A COUPLED-INDUCTOR BUCK-BOOST DC/DC CONVERTER

by Eng. Jhon Erik Navarrete Gómez

Briefly explained, systems modeling is an equivalent exchange of variables such as Speed, precision and effort. When modeling a system, we may have to choose two of these three attributes, leaving one side. A model with a top level of precision and speed will always require an impressive deal of effort. The opposite is rare. This research project will implement a model on a Hardware in the Loop platform that allows improving the design and implementation times of Power Electronics prototypes.

Acknowledgements

I thank my advisors, Professor Carlos A. Torres and Professor Carlos J. Mojica, for their encouragement and advice during writing this thesis.

I wish to thank Eng. Davis Montenegro, my teacher of automation when I started my first approach to the LabVIEW programming environment, for their training and support throughout the years.

Most importantly, I thank my mother, Isabel Gómez, and my brother Freddy Navarrete. Without their support and understanding, my master's study would never have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Preamble

1.1 Introduction

The systems exponentially increase their complexity in both software and hardware each year. What started out as a conventional car was moved to an electric one and today leaps and bounds are being made towards the stand-alone. The example given allow to affirm that more lines of code are required to implement a reliable system, so a greater number of micro-controllers is also necessary, and those manufactured with a greater number of transistors.

Model-based system development offers an early exploration into design, which involves prototyping and verification in a quick time so system design can be further optimized. A reliable model would allow engineers to develop software and hardware in parallel, thus reducing the development cycle. It is necessary to take into account that the model of the system itself is important, but verify the correct operation of the model is more important.

1.2 Problem Statement

Traditional simulation has limited the testing capabilities of engineers designing control systems, forcing them to rely heavily on expensive instruments.

During designing power electronics systems, it is possible to encounter problems related to a stage that requires including changes in the Hardware or even tests or experiments to validate its behavior. Sometimes because of the rigidity of the prototype it is not practical to develop the field tests, in fact, some of them will destroy the hardware if they are carried out incorrectly.

To find an opportunity for improvement in the design and implementation process, the following research question is proposed:

How can experimental results be got in real-time in power electronics systems without having access to an existing prototype?

1.3 Objectives

In this section, the proposed objectives that will answer the research question will be shown.

1.3.1 General Objective

To develop a platform using Hardware in the Loop technique that allows validation of the dynamic behavior of a coupled inductor Buck-Boost DC/DC converter model on FPGA technology.

1.3.2 Specific Objectives

1. To create a parameterizable model of the coupled inductor Buck-Boost DC/DC converter that is compatible with National Instruments VeriStand software.
2. To build a scalable and flexible Hardware in the Loop system that allows loading the model described in the previously with a highly deterministic calculation speed.
3. To design and implement a control system that allows regulating a variable of the coupled inductor Buck-Boost DC/DC converter to the set-point established.
4. To verify the real-time behavior of the closed-loop system executed by the Hardware in the Loop platform.

1.4 Justification

The increasing adoption of unknown technologies at all levels in the industry is a challenge for engineers developing integrated control systems. Until a few years ago, it had been very difficult to have an accurate model combined with top execution speed. Today test systems developed with Hardware in the Loop must run simulation models in hundreds of nanoseconds. To develop tests with Hardware in the Loop will allow to study the behavior of the system under fault conditions impractical to perform with actual systems, and that can destroy the Hardware.

Improving the speed and accuracy of the simulation would help engineers who design and implement systems in power electronics to perform more tests at the signal level, which would reduce the time and cost of physical testing.

Chapter 2

Converter Analysis

2.1 Introduction

Power factor correction, battery charging and discharging, and maximum power point tracking in photovoltaic panels are some many applications that switching converters have in research and industry [1–4]. In all the above applications, the goal is to obtain a regulated output voltage from an unregulated voltage source[5–8].

A common feature of basic converter, both Buck and Boost topologies, is that a single active switching element exists. We use this basic topologies in applications where it is important to have a small size of the elements and a low cost. However, a great disadvantage has also been noted in them, when the voltages are increased, the size of the capacitors becomes an important variable to consider [9–12].

The Boost DC/DC converters operated in their continuous conduction mode have a zero in the right half plane (RHP) on their transfer function. When this characteristic occurs, it refers to the converter as a non-minimum phase converter. It is also possible to affirm that this characteristic hinders the design of the controller, in other words, it limits the band-width of the control loop and, as mentioned before, increases the size of the output capacitor.

When the converter hinders the design of the controller, it refers to the fact that, to guarantee a sufficient margin of stability in the frequency behavior, it is necessary that the frequency of the zero found in the positive half-plane is greater than the frequency of 0 dB crossing of the closed loop gain. Because of this, the dynamic performance of a Boost converter is very limited compared to that of a Buck converter, which has a minimal transfer function that allows you to design your controller without having so many considerations present.

They have developed many studies that focus on develop strategies that reduce the influence of zero in the right half plane of the transfer functions of the converters. One of them mentions that if we design the Boost Converter with an output capacitor with a large equivalent series of resistance (ESR); it is possible to achieve excellent dynamic behavior [13]. However, this condition doesn't take into account the requirement for high efficiency.

Another study suggests that if the converter operate on a point very close to discontinuous conduction mode to get an outstanding performance in dynamic behavior [14]. However, the previous consideration penalizes efficiency by increasing the voltage on the semiconductor elements that are switching.

This work aims to present an unusual converter topology with the ability to raise and lower the voltage, with the same conversion properties of the Buck and Boost converters. The converter shown

in Figure 2.1 has magnetically coupled inductors at the input and output. With these inductors, the input and output currents are not pulsating, the noise level is lower and both the control and the limit of the currents can be easier to deal with compared to the case in which they were pulsating[15, 16].

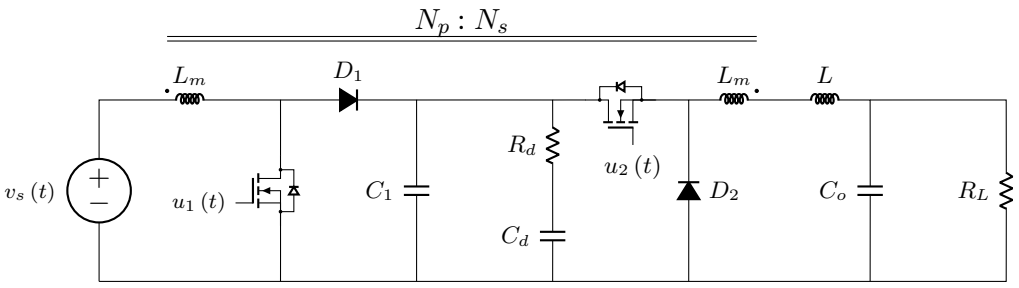


FIGURE 2.1: Schematic circuit diagram of the Non-inverter Buck-Boost converter with magnetically coupled inductors.

Magnetic coupling on inductors at the input and output is an approach that has not been studied deeply. It has carried studies out showing that magnetic coupling makes it possible to obtain a transfer function with two zeros. But the most important thing is that by appropriate selection of the coefficients of the numerator, the zeros can be placed in the left half plane so that the resulting transfer function is of minimal phase. The previous characteristic is maintained even when the possibility of developing the topology using capacitors with an ESR close to zero is evaluated, without implying a high current ripple. This topology therefore provides a filtering of the input and output current, reducing noise and the need for additional filters [17–19].

This converter can have great applications in systems that need backup batteries since they must be charged and discharged correctly. In other words, its application is possible in the space, automotive and telecommunications industries[20, 21].

2.2 Magnetically Coupled Circuits

When two meshes with or without contact between them are mutually affected by the magnetic field, then those meshes are said to be magnetically coupled. With the proposed converter, magnetically coupled inductors are used to transfer energy between them. To begin the converter's analysis, it is necessary to know the concept of mutual inductance that will be dealt with throughout this section. The dot convention will determine the voltage polarities of the coupled components.[22]

2.2.1 Mutual Inductance

In the converter topology we can see it that there are 2 inductors with proximity to each other, that implies that the magnetic flux of one inductor is related to the other. When a current i flows through an inductor with N number of turns, a magnetic flux ϕ occurs. The voltage v induced in the coil is proportional to the number of turns and the rate of change of the magnetic flux regarding time, as showed by Faraday's law:

$$v = N\frac{d\phi}{dt} \tag{2.1}$$

Any change in flux will produce a change in current and therefore a change in voltage, that we can write a close relationship between variables as:

$$v = L\frac{di_{L}}{dt} \tag{2.2}$$

Starting from the previous relationships, it can establish an additional relationship between the variables, the relationship proposed below is called self-inductance because it relates the induced voltage of the inductor with a current variable in time in the same inductor:

$$L = N\frac{\phi}{dt} \tag{2.3}$$

We will show an example that can summarize what we have mentioned so far below. In Figure 2.2 2 coils with self-inductance it will show L_1 and L_2. Coil 1 will have N_1 turns, while coil 2 will have N_2 turns.

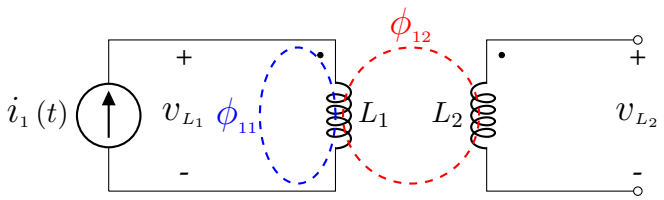


FIGURE 2.2: Mutual inductance M_{21} of coil 2 with respect to coil 1.

We should emphasize it that the ϕ_1 magnetic flux has two components, one that only covers coil 1, while the other that will pass through both coil 1 and coil 2. we can describe the above as:

$$\phi_1 = \phi_{11} + \phi_{12} \tag{2.4}$$

Because magnetic flux ϕ_1 passes through the coil 1, the voltage induced there can be expressed as:

$$v_1 = N_1\frac{d\phi_1}{dt} \tag{2.5}$$

Taking into account that the magnetic flux depends on the current that passes through the inductor, it can also express the voltage of coil 1 as:

$$v_1 = L_1\frac{di_1}{dt} \tag{2.6}$$

L_1 refers to the self-inductance of coil 1. With coil 2, the magnetic flux that passes through it is ϕ_{12}, so its voltage is of the form:

$$v_2 = N_2\frac{d\phi_{12}}{dt} \tag{2.7}$$

Which can also be expressed as:

$$v_2 = M_{21}\frac{di_1}{dt} \tag{2.8}$$

Here the new variable M_{21} was introduced, which is also known in the literature as Mutual Inductance. Subscript 21 refers to the relationship between coil 2 voltage and coil 1 current. We can express the Mutual inductance as:

$$M_{21} = N_2 \frac{d\phi_{12}}{di_1} \tag{2.9}$$

In the same way as with self-inductance L, the units of mutual inductance are measured in Henrys (H). It is necessary to remember that mutual inductance only exist when the related inductors are in proximity and also when they are excited by time-varying sources. In conclusion, there is a mutual inductance when voltage is induced in another inductance because of a time-varying current.

To determine the polarity of the induced voltage when mutual inductance exists, the marks convention will take into account some points that accompany the inductances when they are magnetically coupled. The mark provides the magnetic flux direction if a current enters to the marked terminal. Next figure 2.3 will show some examples that allow to understand the point convention concept discussed above.

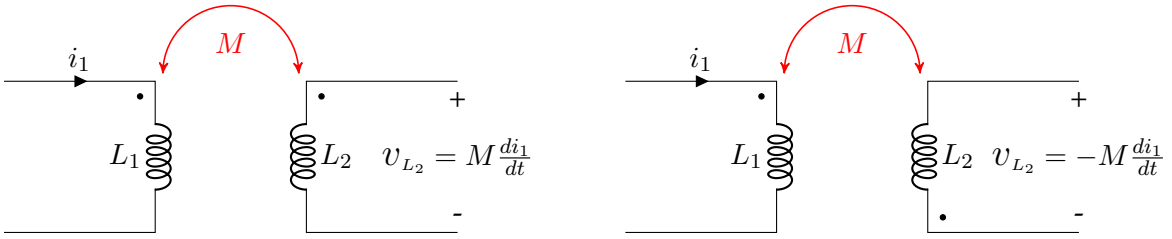


FIGURE 2.3: Point convention examples.

For the last examples, we have used only the variable M, without making use of the subscripts. This is because M_{12} is equal to M_{21}. We will explain this phenomenon below.

2.2.2 Energy behavior in coupled inductors

As mentioned in several books, the inductor is a passive element that aims to store energy in its magnetic field. It is possible to describe the stored energy of the inductor with the following expression:

$$w = \frac{1}{2} L i^2 \tag{2.10}$$

We now use that relationship to describe the energy stored in coupled inductors. To do this, the circuit shown in figure 2.4 will be studied.

To begin the analysis, initial conditions equal to 0 must be assumed in each of the inductors. We will analyze the case studies under 2 scenarios.

1. **Scenario 1:**
   Current i_2 will have a constant value of zero. Current i_1 will be a time-varying current that will start at 0 and grow to a steady state value of I_1.

   We will analyze the power in both inductors:

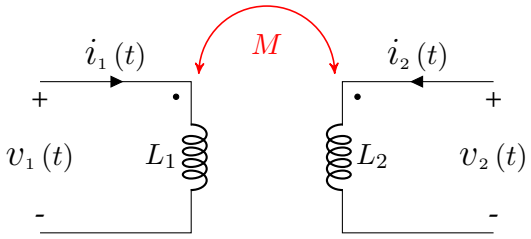


FIGURE 2.4: Example to find the relationship between M_{12} and M_{21}.

Inductor 1:
i_1 is a time dependent function.

$$v_{L_1} = L_1 \frac{di_{L_1}}{dt} \tag{2.11}$$

$$p_1(t) = L_1 \frac{di_{L_1}}{dt} \tag{2.12}$$

Inductor 2:
The current i_2 is zero Amps.

$$v_{L_2} = M_{21} \frac{di_{L_1}}{dt} \tag{2.13}$$

$$p_2(t) = 0 \tag{2.14}$$

The total power by the elements in this scenario is:

$$\begin{aligned} p_{T_1}(t) &= p_1(t) + p_2(t) \\ &= L_1 i_{L_1} \frac{di_{L_1}}{dt} \end{aligned} \tag{2.15}$$

The energy stored in the circuit is:

$$\begin{aligned} w_1 &= \int p_{T_1}(t)\, dt = L_1 \int_0^{I_1} i_{L_1} di_{L_1} \\ &= \frac{1}{2} L_1 I_1^2 \end{aligned} \tag{2.16}$$

2. **Scenario 2:**
Now that i_1 has reached its steady state value I_1, current i_2 will increase from 0 to steady state value I_2.

Again, we will analyze the power in both inductors.

Inductor 1:
i_1 is constant and its value es I_1.

$$v_{L_1} = M_{12} \frac{di_{L_2}}{dt} \tag{2.17}$$

$$p_1(t) = I_{L_1} M_{12} \frac{di_{L_2}}{dt} \tag{2.18}$$

Inductor 2:
The current i_2 a time dependent function.

$$v_{L_2} = L_2 \frac{di_{L_2}}{dt} \tag{2.19}$$

$$p_2(t) = i_{L_2} L_2 \frac{di_{L_2}}{dt} \tag{2.20}$$

The total power by the elements in this scenario is:

$$\begin{aligned} p_{T_2}(t) &= p_1(t) + p_2(t) \\ &= I_{L_1} M_{12} \frac{di_{L_2}}{dt} + i_{L_2} L_2 \frac{di_{L_2}}{dt} \end{aligned} \tag{2.21}$$

The energy stored in the circuit is:

$$\begin{aligned} w_2 &= \int p_{T_2}(t)\, dt = L_2 \int_0^{I_2} i_{L_2} di_{L_2} \\ &= M_{12} I_1 I_2 + \frac{1}{2} L_2 I_2^2 \end{aligned} \tag{2.22}$$

After completing the analysis of the previous scenarios, we need to find the total stored energy of the circuit, it will achieve this with the sum of the energy of each scenario as shown below:

$$\begin{aligned} w_T &= w_1 + w_2 \\ &= \frac{1}{2} L_1 I_1^2 + M_{12} I_1 I_2 + \frac{1}{2} L_2 I_2^2 \end{aligned} \tag{2.23}$$

However, it is necessary to repeat the procedure if we want to understand the phenomenon. Before repeating it, we will reverse the order in which the currents reach their last values. It will first increase i_2 starting from 0 and reaching I_2 and then increase i_1 from 0 to I_1. By repeating the entire procedure we get that the total stored energy is:

$$w_T = \frac{1}{2} L_1 I_1^2 + M_{21} I_1 I_2 + \frac{1}{2} L_2 I_2^2 \tag{2.24}$$

Starting from the principle that the total stored energy must be the same regardless of the order in which the currents reach their initial values, then:

$$M = M_{12} = M_{21} \tag{2.25}$$

And therefore, regardless of the order in which the currents reach their steady state value, the total energy stored in the circuit is:

$$\begin{aligned} w_T &= w_1 + w_2 \\ &= \frac{1}{2} L_1 I_1^2 + M I_1 I_2 + \frac{1}{2} L_2 I_2^2 \end{aligned} \tag{2.26}$$

Now we know the relationship that each of the inductors has with the mutual inductance. However, one concern remains: What is the value of mutual inductance (M)? The reason we did the above development with energy is to find that value. We must observe the circuit, it is a passive circuit which corresponds to that the stored energy must be greater than or equal to zero. Therefore, we can propose the following inequality:

$$\frac{1}{2} L_1 I_1^2 + M I_1 I_2 + \frac{1}{2} L_2 I_2^2 \geq 0 \tag{2.27}$$

To find the value of M starting from the inequality, we must complete the square with the term without affecting it, so we get:

$$\frac{1}{2}\left(i_{L_1}\sqrt{L_1} - i_{L_2}\sqrt{L_2}\right)^2 + i_{L_1} i_{L_2}\left(\sqrt{L_1 L_2} - M\right) \geq 0 \tag{2.28}$$

Since the squared term is never negative and we have an additive expression, the only way left is to take the remaining term and apply the inequality to it. By doing so, we can clear the mutual inductance, as shown below:

$$M \leq \sqrt{L_1 L_2} \tag{2.29}$$

Since the mutual inductance cannot be greater than the geometric mean of the inductances, it is necessary to include a new variable. We will call this new variable the Coupling Factor, which will have values between 0 and 1. The *Coupling Factor* (k) refers to the amount of flux from one coil linked to another, if the complete flux that is produced by one inductor passed through another inductor, they will have a coupling factor of 1. In other words, 100% of the magnetic flux produced by one inductor is linked to another [23].

Everything mentioned above allows us to establish the relationship:

$$M = k\sqrt{L_1 L_2} \tag{2.30}$$

As a conclusion, the value of the coupling coefficient depends on construction factors such as core, orientation, winding and proximity of the coils.

2.3 Possible States Analysis

The semiconductor devices in the proposed converter are transistors and diodes that are controlled to turn on and off to perform the function of an ideal switch, as necessary. Mathematically the switching frequency is equal to the multiplicative inverse of the switching time and depending on the semiconductor device, usually in the 1 kHz to 1 MHz range.

The duty ratio D is the fraction of time spent in position 1 by the switch and is a number between zero and one. It specifies the duty ratio complement as $(1 - D)$.

To analyze the converter, we will separate its operation by states, U_1 and U_2 will select the state of the switches. I will show the states below:

State	U_1	U_2
1	0	0
2	0	1
3	1	0
4	1	1

TABLE 2.1: States of the Non-inverter Buck-Boost converter with magnetically coupled inductors.

I did the simplification of the equations for each state using the **MATLAB** "Symbolic Variables" tool. Algorithm 2.1 shows the declaration corresponding to each variable.

```
syms Vs
    % Source Voltage

% Circuit Elements
syms Lm1 Lm2 L M Rd Cd Ci Co RL
    % Mutual Inductor 1
    % Mutual Inductor 2
    % Inductor
    % Mutual Inductance
    % Damping Resistor
    % Damping Capacitor
    % Input Capacitance
    % Output Capacitance
    % Load Resistor

% State Variables
syms Ilm1 Il Vcd Vci Vco
VarArray = [Ilm1 Il Vcd Vci Vco];
    % Mutual Inductor 1 Current
    % Inductor Current
    % Damping Capacitor Voltage
    % Input Capacitor Voltage
    % Output Capacitor Voltage

% Derivate Variables
syms dIlm1_dt dIl_dt dVcd_dt dVci_dt dVco_dt
DerArray = [dIlm1_dt dIl_dt dVcd_dt dVci_dt dVco_dt];
```

ALGORITHM 2.1: Variables Declaration on MATLAB of Mathworks.

I will show the analysis of each of the converter states below.

2.3.1 First State

Figure 2.5 shows the schematic circuit of the Buck-Boost converter of coupled inductors under the conditions U_1 and U_1 equal to zero.

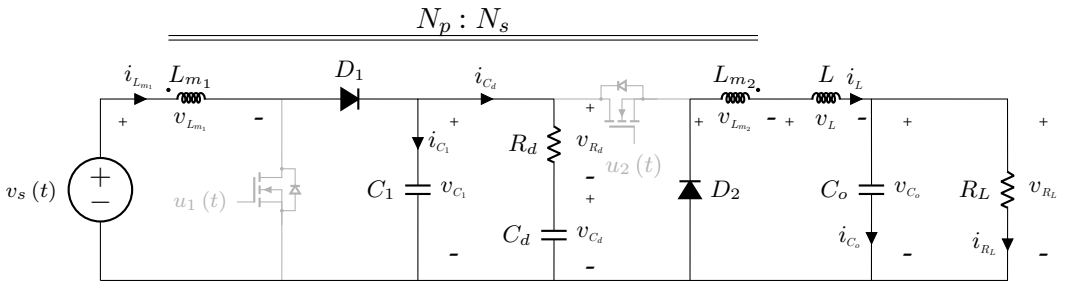


FIGURE 2.5: Schematic circuit diagram of the Non-inverter Buck-Boost converter with magnetically coupled inductors in the first switching state.

Under these conditions, I will show the equations that are derived below:

$$
\begin{aligned}
V_s &= L_{m_1}\frac{di_{L_{m_1}}(t)}{dt} - M\frac{di_L(t)}{dt} + v_{C_i}(t) \\
v_{C_i}(t) &= R_d C_d \frac{dv_{C_d}(t)}{dt} + v_{C_d}(t) \\
i_{L_{m_1}}(t) &= C_i\frac{dv_{C_i}(t)}{dt} + C_d\frac{dv_{C_d}(t)}{dt} \\
-v_{C_o}(t) &= \left(L_{m_2} + L\right)\frac{di_L(t)}{dt} - M\frac{di_{L_{m_1}}(t)}{dt} \\
i_L(t) &= C_o\frac{dv_{C_o}(t)}{dt} + \frac{1}{R_L}v_{C_o}(t)
\end{aligned}
\tag{2.31}
$$

Having found the relationships in equations 2.31, we find the derivatives of each state variable. In this regard, we use the algorithm 2.2 in MATLAB.

```
% Equations First State
Equ_1 = Vs == Lm1*dIlm1_dt - M*dIl_dt + Vci;
Equ_2 = Vci == Rd*Cd*dVcd_dt + Vcd;
Equ_3 = Ilm1 == Ci*dVci_dt + Cd*dVcd_dt;
Equ_4 = (Lm2 + L)*dIl_dt - M*dIlm1_dt == -Vco;
Equ_5 = Il == Co*dVco_dt + Vco/RL;

Equations = [Equ_1, Equ_2, Equ_3, Equ_4, Equ_5];
[dIlm1_dt dIl_dt dVcd_dt dVci_dt dVco_dt] = solve(Equations,DerArray);
```

ALGORITHM 2.2: Algorithm for searching the state variables of the first converter state.

The results of the algorithm are shown below:

$$
\begin{aligned}
\frac{di_{L_{m_1}}(t)}{dt} &= \frac{1}{L_{m_1}(L+L_{m_2}) - M^2}\left[-(L+L_{m_2})\ \boldsymbol{v_{C_i}}(\mathbf{t}) - M\ \boldsymbol{v_{C_o}}(\mathbf{t}) + (L+L_{m_2})\ \mathbf{V_s}\right] \\
\frac{di_L(t)}{dt} &= \frac{1}{L_{m_1}(L+L_{m_2}) - M^2}\left[-M\ \boldsymbol{v_{C_i}}(\mathbf{t}) - L_{m_1}\ \boldsymbol{v_{C_o}}(\mathbf{t}) + M\ \mathbf{V_s}\right] \\
\frac{dv_{C_d}(t)}{dt} &= \frac{1}{C_d R_d}\left[-\boldsymbol{v_{C_d}}(\mathbf{t}) + \boldsymbol{v_{C_i}}(\mathbf{t})\right] \\
\frac{dv_{C_i}(t)}{dt} &= \frac{1}{C_i}\left[\boldsymbol{i_{L_{m_1}}}(\mathbf{t}) + \frac{1}{R_d}\boldsymbol{v_{C_d}}(\mathbf{t}) - \frac{1}{R_d}\boldsymbol{v_{C_i}}(\mathbf{t})\right] \\
\frac{dv_{C_o}(t)}{dt} &= \frac{1}{C_o}\left[\boldsymbol{i_L}(\mathbf{t}) - \frac{1}{R_L}\boldsymbol{v_{C_o}}(\mathbf{t})\right]
\end{aligned}
\tag{2.32}
$$

2.3.2 Second State

Figure 2.6 shows the schematic circuit of the Buck-Boost converter of coupled inductors under the conditions U_1 equal to zero and U_2 equal to one.
Under these conditions, I will show the equations that are derived below:

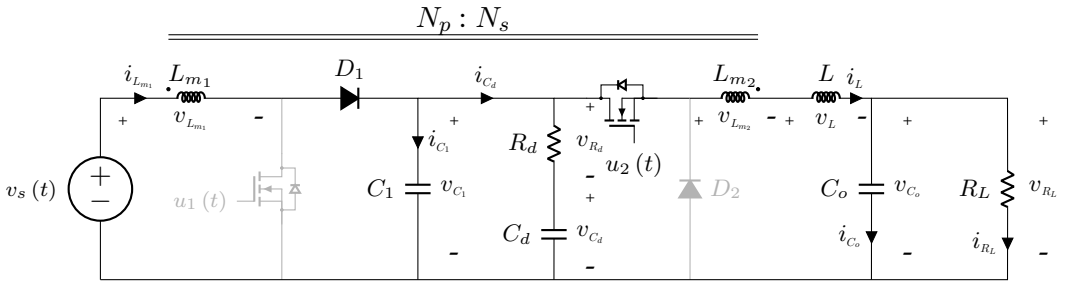


FIGURE 2.6: Schematic circuit diagram of the Non-inverter Buck-Boost converter with magnetically coupled inductors in the second switching state.

$$
\begin{aligned}
V_s &= L_{m_1}\frac{di_{L_{m_1}}(t)}{dt} - M\frac{di_L(t)}{dt} + v_{C_i}(t) \\
v_{C_i}(t) &= R_d C_d \frac{dv_{C_d}(t)}{dt} + v_{C_d}(t) \\
i_{L_{m_1}}(t) &= C_i\frac{dv_{C_i}(t)}{dt} + C_d\frac{dv_{C_d}(t)}{dt} + i_L(t) \\
-v_{C_o}(t) &= L_{m_2}\frac{di_L(t)}{dt} - M\frac{di_{L_{m_1}}(t)}{dt} + L\frac{di_L(t)}{dt} - R_d C_d\frac{dv_{C_d}(t)}{dt} - v_{C_d}(t) \\
i_L(t) &= C_o\frac{dv_{C_o}(t)}{dt} + \frac{1}{R_L}v_{C_o}(t)
\end{aligned}
\tag{2.33}
$$

Having found the relationships in equations 2.33, we find the derivatives of each state variable. In this regard, we use the algorithm 2.3 in MATLAB.

```
% Equations Second State
Equ_1 = Vs == Lm1*dIlm1_dt - M*dIl_dt + Vci;
Equ_2 = Vci == Rd*Cd*dVcd_dt + Vcd;
Equ_3 = Ilm1 == Ci*dVci_dt + Cd*dVcd_dt + Il;
Equ_4 = Rd*Cd*dVcd_dt + Vcd == Lm2*dIl_dt - M*dIlm1_dt + L*dIl_dt + Vco;
Equ_5 = Il == Co*dVco_dt + Vco/RL;

Equations = [Equ_1, Equ_2, Equ_3, Equ_4, Equ_5];
[dIlm1_dt dIl_dt dVcd_dt dVci_dt dVco_dt] = solve(Equations,DerArray);
```

ALGORITHM 2.3: Algorithm for searching the state variables of the second converter state.

The results of the algorithm are shown below:

$$
\begin{aligned}
\frac{d\dot{i}_{L_{m_1}}(t)}{dt} &= \frac{1}{L_{m_1}(L+L_{m_2})-M^2}\Big[-(L+L_{m_2}-M)\ \boldsymbol{v}_{\boldsymbol{C_i}}(\mathbf{t})-M\,\boldsymbol{v}_{\boldsymbol{C_o}}(\mathbf{t})+(L+L_{m_2})\ \mathbf{V_s}\Big]\\
\frac{d\dot{i}_{L}(t)}{dt} &= \frac{1}{L_{m_1}(L+L_{m_2})-M^2}\Big[(L_{m_1}-M)\ \boldsymbol{v}_{\boldsymbol{C_i}}(\mathbf{t})-L_{m_1}\,\boldsymbol{v}_{\boldsymbol{C_o}}(\mathbf{t})+M\,\mathbf{V_s}\Big]\\
\frac{dv_{C_d}(t)}{dt} &= \frac{1}{C_dR_d}\Big[-\boldsymbol{v}_{\boldsymbol{C_d}}(\mathbf{t})+\boldsymbol{v}_{\boldsymbol{C_i}}(\mathbf{t})\Big]\\
\frac{dv_{C_i}(t)}{dt} &= \frac{1}{C_i}\Big[\boldsymbol{i}_{\boldsymbol{L_{m_1}}}(\mathbf{t})-\boldsymbol{i}_{\boldsymbol{L}}(\mathbf{t})+\frac{1}{R_d}\boldsymbol{v}_{\boldsymbol{C_d}}(\mathbf{t})-\frac{1}{R_d}\boldsymbol{v}_{\boldsymbol{C_i}}(\mathbf{t})\Big]\\
\frac{dv_{C_o}(t)}{dt} &= \frac{1}{C_o}\Big[\boldsymbol{i}_{\boldsymbol{L}}(\mathbf{t})-\frac{1}{R_L}\boldsymbol{v}_{\boldsymbol{C_o}}(\mathbf{t})\Big]
\end{aligned}
\tag{2.34}
$$

2.3.3 Third State

Figure 2.7 shows the schematic circuit of the Buck-Boost converter of coupled inductors under the conditions U_1 equal to one and U_2 equal to zero.

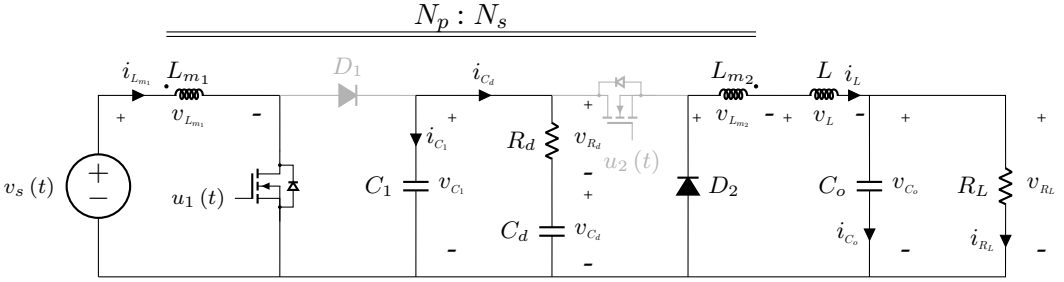


FIGURE 2.7: Schematic circuit diagram of the Non-inverter Buck-Boost converter with magnetically coupled inductors in the third switching state.

Under these conditions, I will show the equations that are derived below:

$$
\begin{aligned}
V_s &= L_{m_1}\frac{d\dot{i}_{L_{m_1}}(t)}{dt}-M\frac{d\dot{i}_L(t)}{dt}\\
v_{C_i}(t) &= R_dC_d\frac{dv_{C_d}(t)}{dt}+v_{C_d}(t)\\
0 &= C_i\frac{dv_{C_i}(t)}{dt}+C_d\frac{dv_{C_d}(t)}{dt}\\
-v_{C_o}(t) &= L_{m_2}\frac{d\dot{i}_L(t)}{dt}-M\frac{d\dot{i}_{L_{m_1}}(t)}{dt}+L\frac{d\dot{i}_L(t)}{dt}\\
\dot{i}_L(t) &= C_o\frac{dv_{C_o}(t)}{dt}+\frac{1}{R_L}v_{C_o}(t)
\end{aligned}
\tag{2.35}
$$

Having found the relationships in equations 2.35, we find the derivatives of each state variable. In this regard, we use the algorithm 2.4 in MATLAB.

```
% Equations Third State
Equ_1 = Vs == Lm1*dIlm1_dt - M*dIl_dt;
Equ_2 = Vci == Rd*Cd*dVcd_dt + Vcd;
Equ_3 = 0 == Ci*dVci_dt + Cd*dVcd_dt;
Equ_4 = (Lm2 + L)*dIl_dt - M*dIlm1_dt == -Vco;
Equ_5 = Il == Co*dVco_dt + Vco/RL;

Equations = [Equ_1, Equ_2, Equ_3, Equ_4, Equ_5];
[dIlm1_dt dIl_dt dVcd_dt dVci_dt dVco_dt] = solve(Equations,DerArray);
```

ALGORITHM 2.4: Algorithm for searching the state variables of the third converter state.

$$
\begin{aligned}
\frac{d\dot{i}_{L_{m_1}}(t)}{dt} &= \frac{1}{L_{m_1}(L+L_{m_2}) - M^2}\left[(L+L_{m_2})\ \mathbf{V_s} - M\ \boldsymbol{v_{C_o}}\right] \\
\frac{d\dot{i}_{L}(t)}{dt} &= \frac{1}{L_{m_1}(L+L_{m_2}) - M^2}\left[M\ \mathbf{V_s} - L_{m_1}\ \boldsymbol{v_{C_o}}(\mathbf{t})\right] \\
\frac{dv_{C_d}(t)}{dt} &= \frac{1}{C_d R_d}\left[-\boldsymbol{v_{C_d}}(\mathbf{t}) + \boldsymbol{v_{C_i}}(\mathbf{t})\right] \\
\frac{dv_{C_i}(t)}{dt} &= \frac{1}{C_i R_d}\left[\boldsymbol{v_{C_d}}(\mathbf{t}) - \boldsymbol{v_{C_i}}(\mathbf{t})\right] \\
\frac{dv_{C_o}(t)}{dt} &= \frac{1}{C_o}\left[\boldsymbol{i_L}(\mathbf{t}) - \frac{1}{R_L}\boldsymbol{v_{C_o}}(\mathbf{t})\right]
\end{aligned}
\tag{2.36}
$$

2.3.4 Fourth State

Figure 2.8 shows the schematic circuit of the Buck-Boost converter of coupled inductors under the conditions U_1 and U_2 equal to one.

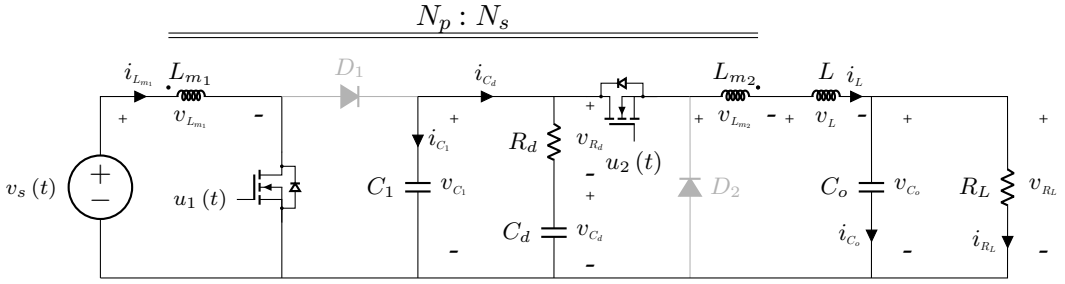


FIGURE 2.8: Schematic circuit diagram of the Non-inverter Buck-Boost converter with magnetically coupled inductors in the fourth switching state.

Under these conditions, I will show the equations that are derived below:

$$
\begin{aligned}
V_s &= L_{m_1}\frac{d\dot{i}_{L_{m_1}}(t)}{dt} - M\frac{d\dot{i}_L(t)}{dt} \\
v_{C_i}(t) &= R_d C_d \frac{dv_{C_d}(t)}{dt} + v_{C_d}(t) \\
0 &= C_i\frac{dv_{C_i}(t)}{dt} + C_d\frac{dv_{C_d}(t)}{dt} + \dot{i}_L(t) \\
0 &= L_{m_2}\frac{d\dot{i}_L(t)}{dt} - M\frac{d\dot{i}_{L_{m_1}}(t)}{dt} + L\frac{d\dot{i}_L(t)}{dt} + v_{C_o}(t) - v_{C_d}(t) - R_d C_d\frac{dv_{C_d}(t)}{dt} \\
\dot{i}_L(t) &= C_o\frac{dv_{C_o}(t)}{dt} + \frac{1}{R_L}v_{C_o}(t)
\end{aligned}
\tag{2.37}
$$

Having found the relationships in equations 2.37, we find the derivatives of each state variable. In this regard, we use the algorithm 2.5 in MATLAB.

```
% Equations Second State
Equ_1 = Vs == Lm1*dIlm1_dt - M*dIl_dt;
Equ_2 = Vci == Rd*Cd*dVcd_dt + Vcd;
Equ_3 = 0 == Ci*dVci_dt + Cd*dVcd_dt + Il;
Equ_4 = Lm2*dIl_dt - M*dIlm1_dt + L*dIl_dt + Vco - Vcd - Rd*Cd*dVcd_dt == 0;
Equ_5 = Il == Co*dVco_dt + Vco/RL;

Equations = [Equ_1, Equ_2, Equ_3, Equ_4, Equ_5];
[dIlm1_dt dIl_dt dVcd_dt dVci_dt dVco_dt] = solve(Equations,DerArray);
```

ALGORITHM 2.5: Algorithm for searching the state variables of the second converter state.

The results of the algorithm are shown below:

$$
\begin{aligned}
\frac{d\dot{i}_{L_{m_1}}(t)}{dt} &= \frac{1}{L_{m_1}(L+L_{m_2}) - M^2}\left[M\,\boldsymbol{v_{C_i}}(\mathbf{t}) - M\,\boldsymbol{v_{C_o}}(\mathbf{t}) + (L+L_{m_2})\,\mathbf{V_s}\right] \\
\frac{d\dot{i}_L(t)}{dt} &= \frac{1}{L_{m_1}(L+L_{m_2}) - M^2}\left[L_{m_1}\,\boldsymbol{v_{C_i}}(\mathbf{t}) - L_{m_1}\,\boldsymbol{v_{C_o}}(\mathbf{t}) + M\,\mathbf{V_s}\right] \\
\frac{dv_{C_d}(t)}{dt} &= \frac{1}{C_d R_d}\left[-\boldsymbol{v_{C_d}}(\mathbf{t}) + \boldsymbol{v_{C_i}}(\mathbf{t})\right] \\
\frac{dv_{C_i}(t)}{dt} &= \frac{1}{C_i}\left[-\boldsymbol{i_L}(\mathbf{t}) + \frac{1}{R_d}\boldsymbol{v_{C_d}}(\mathbf{t}) - \frac{1}{R_d}\boldsymbol{v_{C_i}}(\mathbf{t})\right] \\
\frac{dv_{C_o}(t)}{dt} &= \frac{1}{C_o}\left[\boldsymbol{i_L}(\mathbf{t}) - \frac{1}{R_L}\boldsymbol{v_{C_o}}(\mathbf{t})\right]
\end{aligned}
\tag{2.38}
$$

2.3.5 Switched Model

As given by the variations in energy accumulation, this model explains fundamental low-frequency dynamics, and it also captures the switching dynamics of power among electronic converters.

A valuable research method, which emphasizes the existence of external control behavior, is the switched model. We can use this model for simulation and control design purposes in its bilinear

form. To get other types of models, such as average or reduced-order models, the switched model provides a starting point.

A power electronic converter experiences a regularly repeated series of configurations during its operating time interval, also called switching time. In fact, each such configuration represents a unique circuit containing sources and passive elements that a series of differential equations can mathematically describe.

Once the analysis of each of the converter states has been achieved, D will replace the input signal U, to take into account the useful cycle. To find the switched model, it will use the following code:

```
% DerArray = [dIlm1_dt dIl_dt dVcd_dt dVci_dt dVco_dt];
syms D1 D2

dIlm1_dt =  FirstState.dIlm1_dt*(1-D1)*(1-D2) + ...
            SecondState.dIlm1_dt*(1-D1)*(D2) + ...
            FourthState.dIlm1_dt*(D1)*(D2);
            SwitchedModel.dIlm1_dt = simplify(dIlm1_dt);

dIl_dt =    FirstState.dIl_dt*(1-D1)*(1-D2) + ...
            SecondState.dIl_dt*(1-D1)*(D2) + ...
            FourthState.dIl_dt*(D1)*(D2);
            SwitchedModel.dIl_dt = simplify(dIl_dt);

dVcd_dt =   FirstState.dVcd_dt*(1-D1)*(1-D2) + ...
            SecondState.dVcd_dt*(1-D1)*(D2) + ...
            FourthState.dVcd_dt*(D1)*(D2);
            SwitchedModel.dVcd_dt = simplify(dVcd_dt);

dVci_dt =   FirstState.dVci_dt*(1-D1)*(1-D2) + ...
            SecondState.dVci_dt*(1-D1)*(D2) + ...
            FourthState.dVci_dt*(D1)*(D2);
            SwitchedModel.dVci_dt = simplify(dVci_dt);

dVco_dt =   FirstState.dVco_dt*(1-D1)*(1-D2) + ...
            SecondState.dVco_dt*(1-D1)*(D2) + ...
            FourthState.dVco_dt*(D1)*(D2);
            SwitchedModel.dVco_dt = simplify(dVco_dt);
```

ALGORITHM 2.6: Algorithm for searching the equations of switched model.

The results of the algorithm are shown below.

The mathematical expression that models the behavior mutual inductance current is:

$$\begin{aligned}\frac{d\dot{i}_{L_{m_1}}(t)}{dt} = \frac{1}{L_{m_1}(L+L_{m_2}) - M^2}\Bigg[&\Big[D_1(L+L_{m_2}) + D_2(M) - (L+L_{m_2})\Big]\boldsymbol{v}_{C_i}(\mathbf{t}) \\ &- \Big[M(D_1D_2 - D_1 + 1)\Big]\boldsymbol{v}_{C_o}(\mathbf{t}) + \Big[(L+L_{m_2})(D_1D_2 - D_1 + 1)\Big]\mathbf{V_s}\Bigg]\end{aligned} \tag{2.39}$$

I will show the way in which it can reflect this mathematical function in a Simulink model in Figure 2.9.

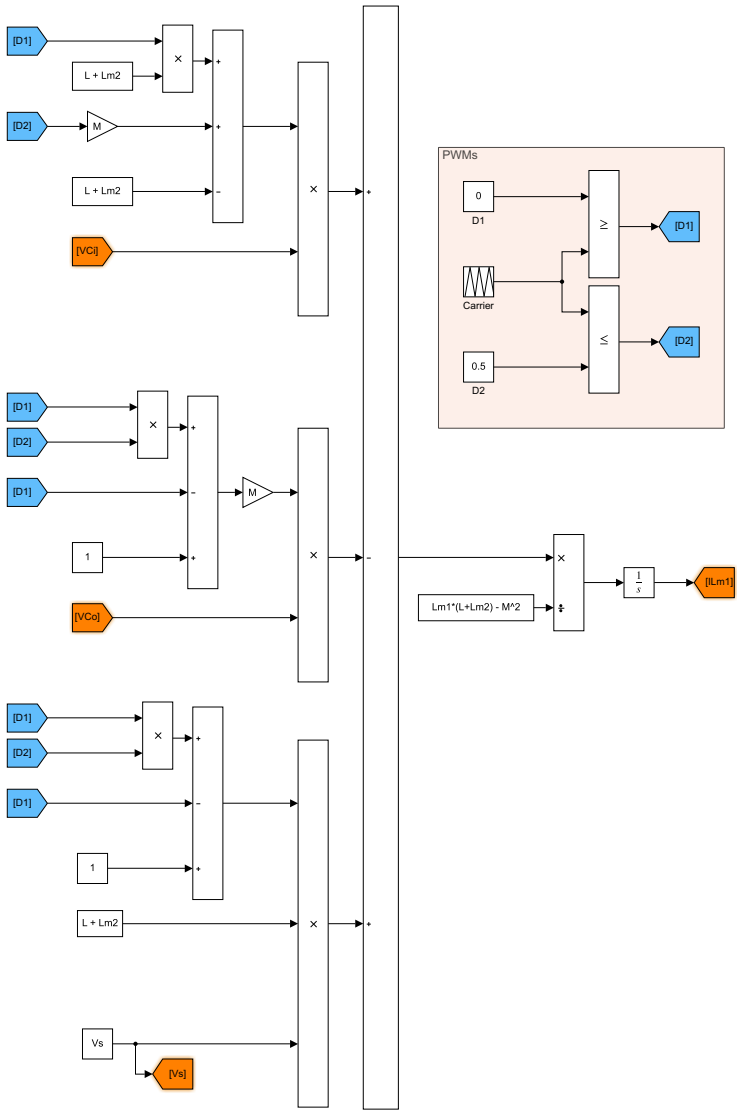


FIGURE 2.9: Mutual Inductance Current Model on Simulink/MATLAB.

The mathematical expression that models the behavior inductance current is:

$$\frac{d\dot{i}_L(t)}{dt} = \frac{1}{L_{m_1}(L+L_{m_2}) - M^2}\Big[\Big[D_1M + D_2Lm_1 - M\Big]\boldsymbol{v}_{C_i}(\mathbf{t}) - \Big[L_{m_1}(D_1D_2 - D_1 + 1)\Big]\boldsymbol{v}_{C_o}(\mathbf{t}) + \Big[M(D_1D_2 - D_1 + 1)\Big]\mathbf{V_s}\Big] \tag{2.40}$$

I will show the way in which it can reflect this mathematical function in a Simulink model in Figure 2.10.

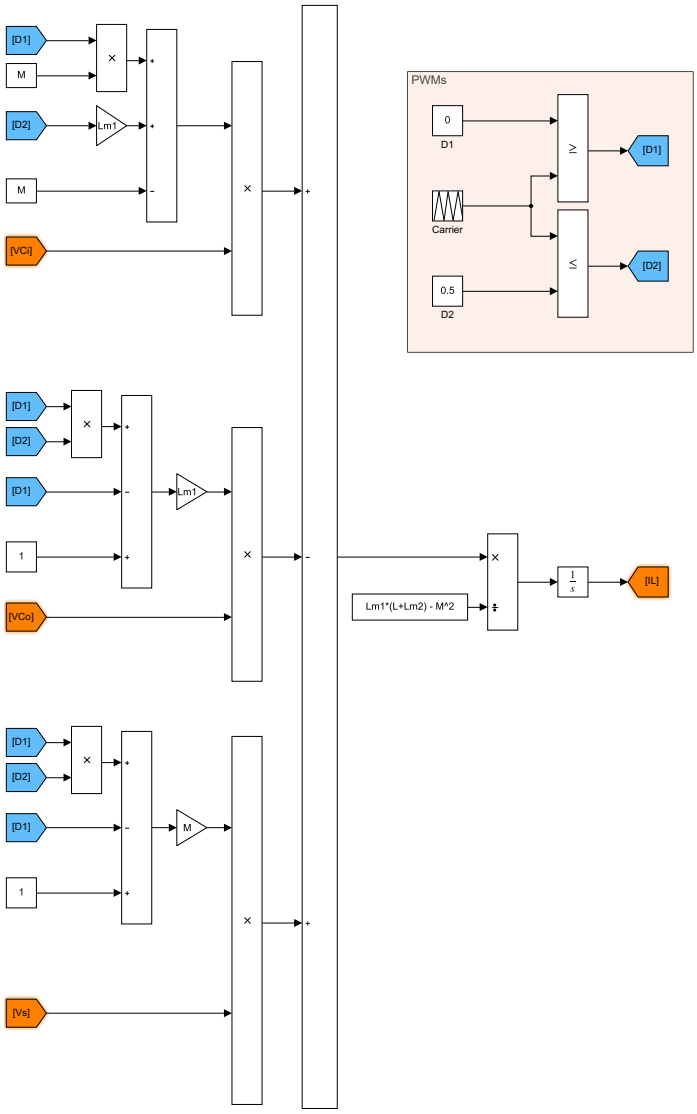


FIGURE 2.10: Inductance Current Model on Simulink/MATLAB.

The mathematical expression that models the behavior of the damping capacitor voltage is:

$$\frac{dv_{C_d}(t)}{dt} = \frac{1}{C_d R_d}\left[-\Big[D_1 D_2 - D_1 + 1\Big]\boldsymbol{v_{C_d}}(\mathbf{t}) + \Big[D_1 D_2 - D_1 + 1\Big]\boldsymbol{v_{C_i}}(\mathbf{t}) \right] \tag{2.41}$$

I will show the way in which it can reflect this mathematical function in a Simulink model in Figure 2.11.

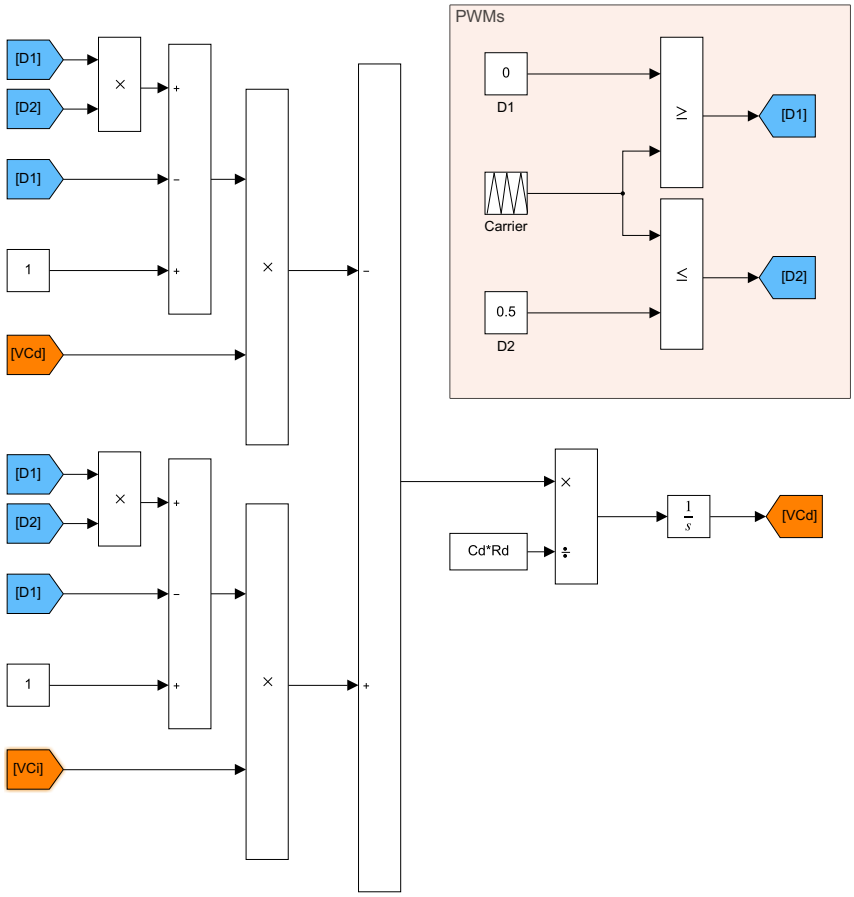


FIGURE 2.11: Inductance Current Model on Simulink/MATLAB.

The mathematical expression that models the behavior of the input capacitor voltage is:

$$\begin{aligned}\frac{dv_{C_i}(t)}{dt} = \frac{1}{C_i}\Bigg[&\Big[1-D_1\Big]\boldsymbol{i}_{L_{m_1}}(\mathbf{t}) - \Big[D_2\Big]\boldsymbol{i}_{L}(\mathbf{t}) \\ &+ \frac{1}{R_d}\Big[D_1D_2 - D_1 + 1\Big]\boldsymbol{v}_{C_d}(\mathbf{t}) - \frac{1}{R_d}\Big[D_1D_2 - D_1 + 1\Big]\boldsymbol{v}_{C_i}(\mathbf{t})\Bigg]\end{aligned} \tag{2.42}$$

I will show the way in which it can reflect this mathematical function in a Simulink model in Figure 2.12.

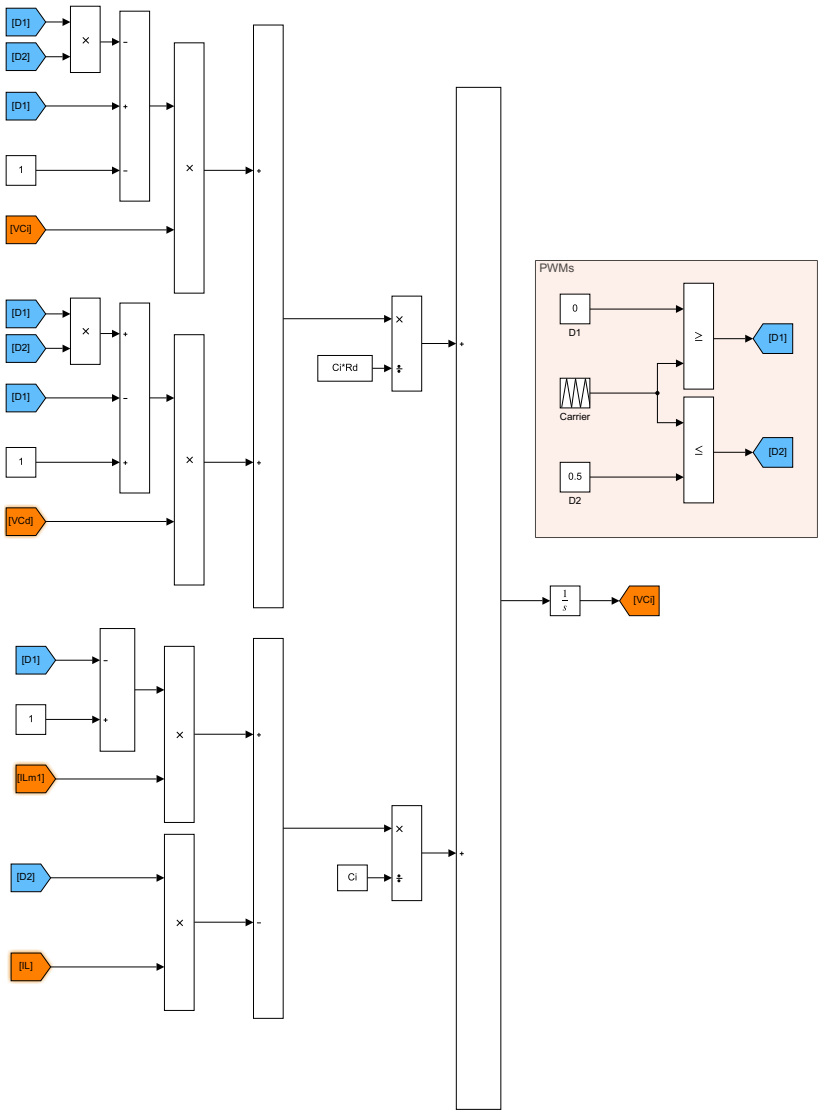


FIGURE 2.12: Inductance Current Model on Simulink/MATLAB.

The mathematical expression that models the behavior of the output capacitor voltage is:

$$\frac{dv_{C_o}(t)}{dt} = \frac{1}{C_o}\left[\Big[D_1D_2 - D_1 + 1\Big]\boldsymbol{i_L}(\mathbf{t}) - \frac{1}{R_L}\Big[D_1D_2 - D_1 + 1\Big]\boldsymbol{v_{C_o}}(\mathbf{t})\right] \tag{2.43}$$

I will show the way in which it can reflect this mathematical function in a Simulink model in Figure 2.13.

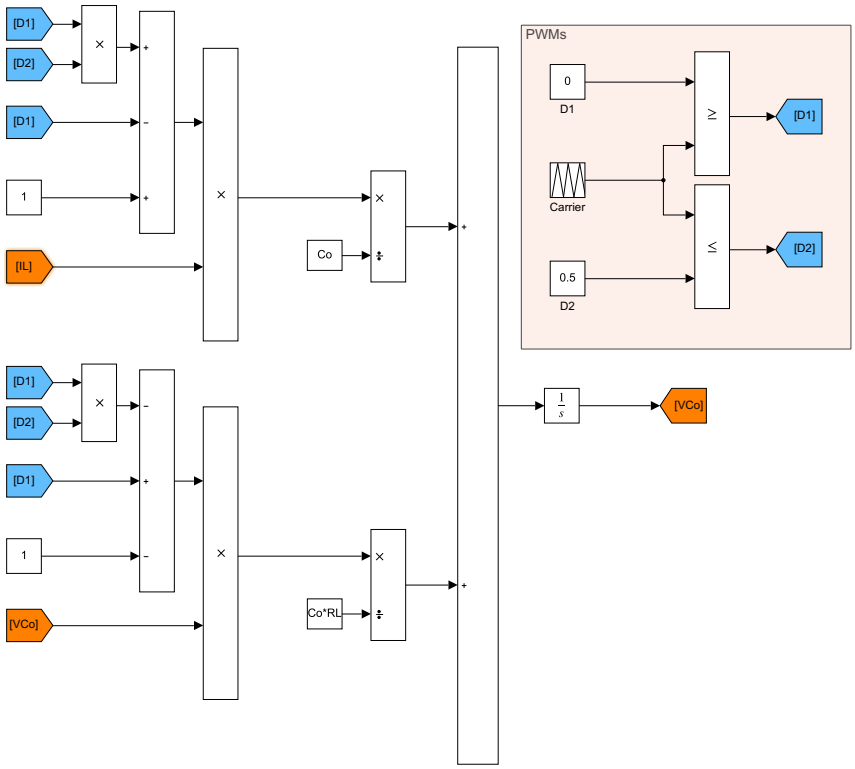


FIGURE 2.13: Inductance Current Model on Simulink/MATLAB.

2.4 Steady-State Converter Analysis

It would be possible to derive an expression for the output voltage of the Coupled-Inductor Buck-Boost Converter using some simple arguments based on Fourier analysis. However, it may not be obvious how to apply these arguments to find the DC output voltage.

In this part, I will develop a more general method for analyzing. The principles of inductor volt-second balance and capacitor charge balance will be used. These can solve for the inductor currents and capacitor voltages of switching converters.

```
1     Equations = [   SwitchedModel.dIlm1_dt == 0,...
2                     SwitchedModel.dIl_dt == 0,...
3                     SwitchedModel.dVcd_dt == 0,...
4                     SwitchedModel.dVci_dt == 0,...
5                     SwitchedModel.dVco_dt == 0];
6
7     [Ilm1 Il Vcd Vci Vco] = solve(Equations,[Ilm1 Il Vcd Vci Vco])
```

ALGORITHM 2.7: Algorithm used to find the steady state value of each state variable in MATLAB.

The results of the algorithm are:

$$\begin{aligned} I_{L_{m_1}} &= \frac{V_s {D_2}^2}{R_L (1-D_1)^2} \\ I_L &= \frac{V_s D_2}{R_L (1-D_1)} \\ V_{C_d} &= \frac{V_s \left[D_1 D_2 - D_1 + 1\right]}{1-D_1} \\ V_{C_i} &= \frac{V_s \left[D_1 D_2 - D_1 + 1\right]}{1-D_1} \\ V_{C_o} &= \frac{V s D_2}{1-D_1} \end{aligned} \tag{2.44}$$

Chapter 3

Hardware in the Loop

3.1 Background

The aviation industry encouraged the development of HIL techniques over 25 years ago with real-time simulations of aircraft [24], however the infrastructure for implementing HIL was expensive since It involved a high level of processing, so the advances in this method were affected, it was not until 2005 that W. Ren and his partners made use of the extensive real-time processing capacity that the University of Florida had Developing and testing for the first time an industrial grade power controller focused on ships of the United States Navy, this research began the development of ships that implement a control system for compensating electrical loads on the distribution bus. By having the level of processing necessary for the development of HIL tests, they could reduce the simulation time even when the complexity of the system was increased. Although they increased the speed by distributing the computational load, they found it necessary to feed back the system with an analog signal, because the switching frequency was very high, so the PWM signal needs more computation time. Which disadvantages the system [25].

In this same year, the article [26] shows an alternative in the development of HIL starting from a virtual platform called Virtual Test Bed Real-Time (VTB-RT), the focus of this article is to provide the tools to the engineers who design control systems, to show the HIL method in the development of any system before its implementation in reality. These researchers started from the simulation interface problem, addressing two methods to reduce computation times, the first method was the implementation of a technique called Firing Signal Averaging (FSA), the second consisted in the Time's use Variant First Order Approximation (TFA) technique. These methods increased the response time of the system so that by using the HIL method, more complex systems could be simulated.

The results of the RT platform for the development of HIL techniques, different approaches emerged for implementing real-time systems, one of these is shown in the article [27] elaborating a digital power controller for aerospace applications to model the energy conversion system composed of solar panels and batteries, the drawback for the development of the digital controller was the costs involved in implementing real tests in aerospace ships, so techniques such as HIL's were A simple and reliable option, the results that the platform is an effective tool for implementing digital control systems, which makes it suitable for real-time testing.

After the popularity of RT, the article [28] was published, which mentions the advances of the platform showing an economic solution to the development of HIL, since the hardware necessary to implement this method was still very expensive , for which virtual systems that guaranteed real-time tests with high reliability had a high demand, so the platform was achieving better processing

times around 100us-300us, which compared to real Hardware was a very attractive solution for systems without high switching frequencies.

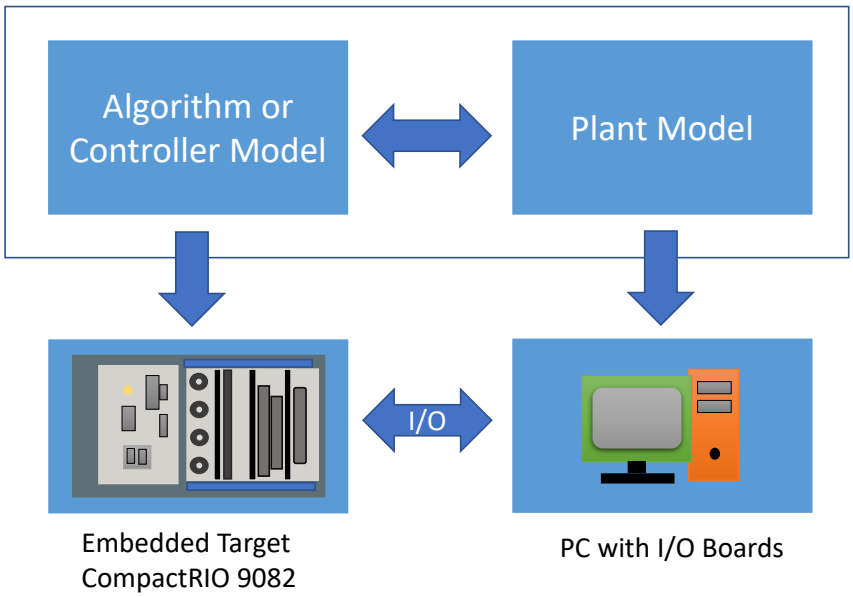


FIGURE 3.1: General Architecture.

Although the virtual platforms were used for the development of HIL, it was necessary to know how to implement this methodology correctly. [29] presents the steps to follow for the execution of Hardware-in-the-Loop tests, in order to Correctly determine the simulator requirements, among the procedures is the development of the simulation where prototypes of the system are created, to later develop the HIL tests, so that the modeled system is implemented in the digital simulator, in addition to implementing the control on a development card. The digital simulator must comply with the level of processing necessary to execute the simulation in real time, for this research a National instrument controller (PXI 8196) was used that provided a high capacity for data acquisition, making HIL tests more flexible, the results obtained correspond to a good implementation, however it was concluded that the processing speed of this development card should still be investigated for more complex systems.

In the article [30] the techniques for the development of HIL are described, the first is called Controller Hardware-in-the-Loop (CHIL) which consists of the development of tests focused on the controller, so that the signals are of low power, on the other hand, the second is called Power Hardware-in-the-Loop (PHIL), it consists of the development of tests to test the power devices, so that it is focused on the emulation of the plant to generate signals that are adapted through an interface that allows simulating the power signals, as if the real system were being tested.

With technological advances, processing systems are faster, so that the HIL methodology can be applied covering greater fields of action, in the article [31] the problem of implementing solutions in the field of converters is shown. power, they start from the analysis of working with analog

systems, saying that it is an alternative for the development of real tests at small scales, however it is not the solution for more complex projects, since the design of analog prototypes for complex systems requires longer planning time, this to ensure a behavior very similar to the real one, so that simulations with the HIL technique is a more profitable and safe option for developments, the results show that the control method is effective under steady-state conditions , being evaluated with the HIL technique which has demonstrated stability over a long period of time, likewise, the Using this technique allows you to reduce the error by ensuring that the converter behaves as in the real system.

With regard to technological advances, the article [32] shows the application of CHIL to a complex system, covering the problems of a control system composed of an electric motor, an inverter and batteries, the results obtained show that advances in processing have increased, so that the use of the FPGA to simulate electrical models is a viable option, since these lead to a longer computation time, the time achieved was around 200ns to 500ns On the other hand, the mechanical systems were simulated in a CPU since the dynamics of the system is slower, the time was around 10us to 50us, therefore the implementation of the FPGA and the CPU to simulate these systems is a reliable choice Due to the high level of processing it is possible to simulate these highly complex systems.

3.2 Overview

We easily update the platform with additional features using a modular architecture to future proof the test framework and meet the specifications of challenging embedded device test applications. The Figure 1 underneath appears the different hardware and software it amasses components into an HIL system. The framework incorporates a program for test robotization in expansion to the HIL test execution and interface. Underneath the computer program are two distinctive hardware platforms for measured I/O and a third that's used for application-specific exchanging, loads, and flag conditioning.

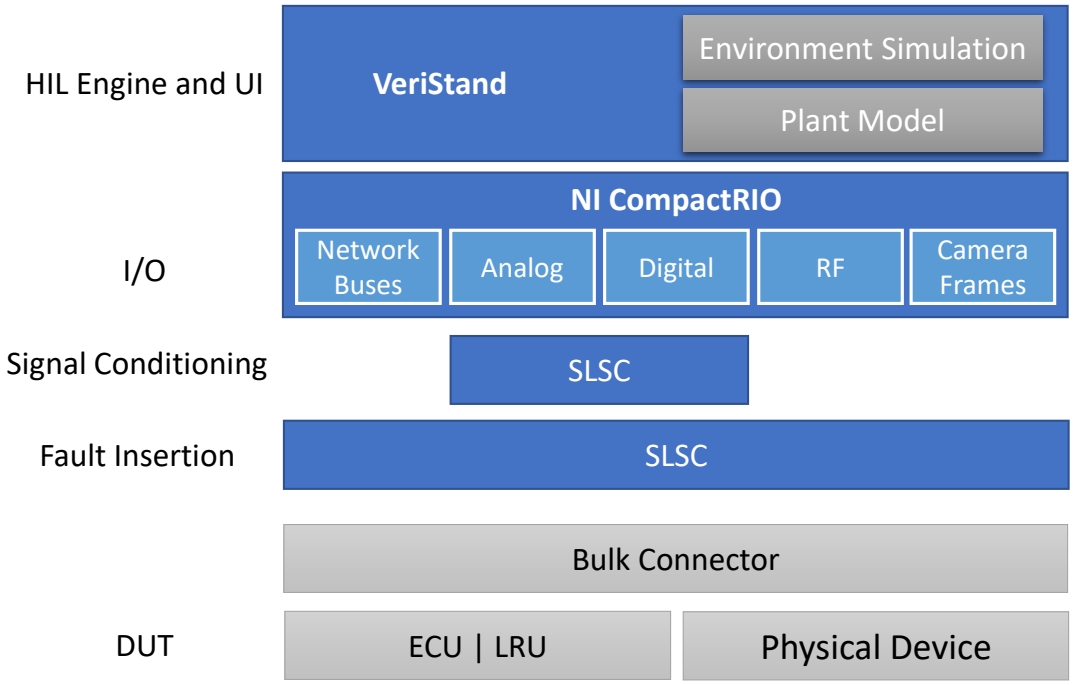


FIGURE 3.2: Flexible and modular Hardware in the Loop System.

3.3 Programming Environment

NI VeriStand offers a platform to more effectively construct real-time testing applications. The real-time test systems consider factors as durability test cells, environmental test systems, or Hardware-in-the-loop (HIL) simulators; we need to build many of the following features in our real-time test software for the application. Some of them are: execution of control algorithms, analytical routines, or models of simulation, among others.

This out-of-the-box functionality incorporated in a well-tested architecture would speed up the production and reduces the cost of support and maintenance for design project. While NI VeriStand provides most of the features needed by a real-time testing application, we design a software that allows it to be changed and expanded to ensure that it always meets the unique application specifications using LabVIEW. Figure 3.3 will allow us to know how NI VeriStand works.

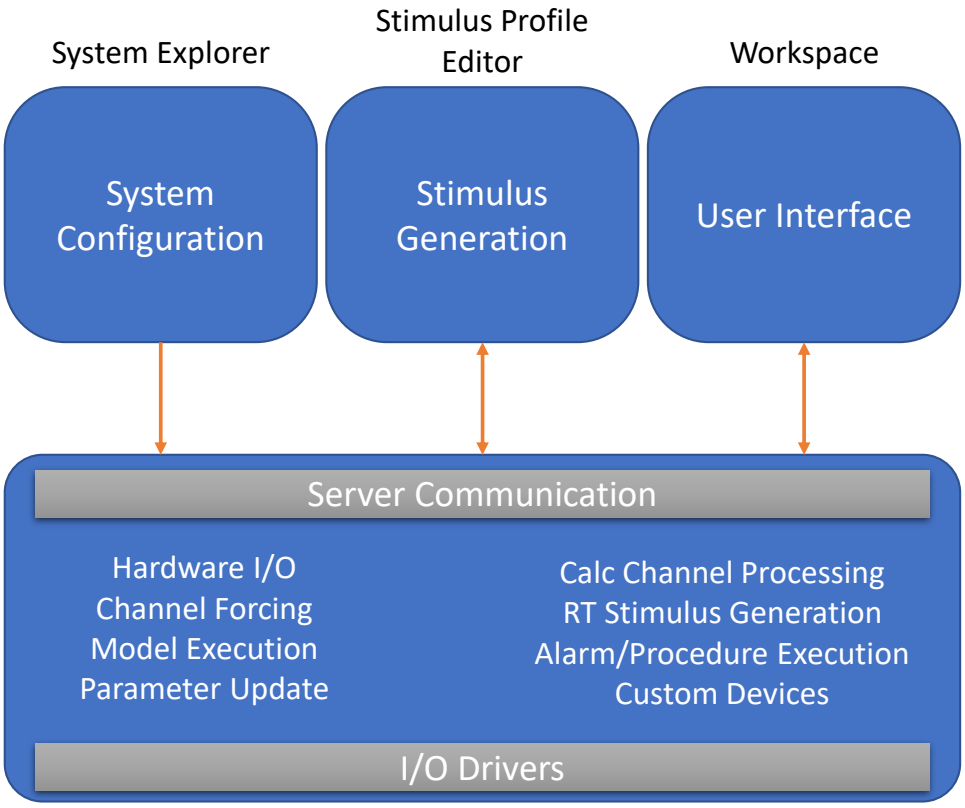


FIGURE 3.3: Component of a test application using NI VeriStand.

Using the NI VeriStand System Explorer window, the NI VeriStand Real-Time Engine is configured to run on a real-time execution target, such as a real-time NI CompactRIO system. The NI VeriStand Workspace window offers a run-time interface to the engine once I have deployed this configuration to the NI VeriStand Real-Time Engine and contains several resources that we can use to track and communicate with the real-time test application.

3.4 NI CompactRIO

CompactRIO is a robust, lightweight hardware design suitable for most harsh environments and for laboratory settings that require a small physical footprint. CompactRIO offers high-performance processing and heterogeneous computing elements, including ARM-based Xilinx Zynq SoCs, and quad-core Intel Atom processors and Xilinx Kintex-7 FPGAs, although it usually provides lower

computational performance than PXI. CompactRIO also provides signals, built-in isolation, and industrial I/O with measurement-specific signal conditioning.

For bench-top HIL systems, CompactRIO is an ideal match. It does not have as many I/O options as PXI, but it is a brilliant choice of unit testing of ECUs that require integrated processing and FPGA performance, such as ECU testing, because of the compact form factor and lower price.

I will build this project using CompactRIO 9082, so understanding its structure and characteristics will be essential. The Figure 3.4 will help us to do it.

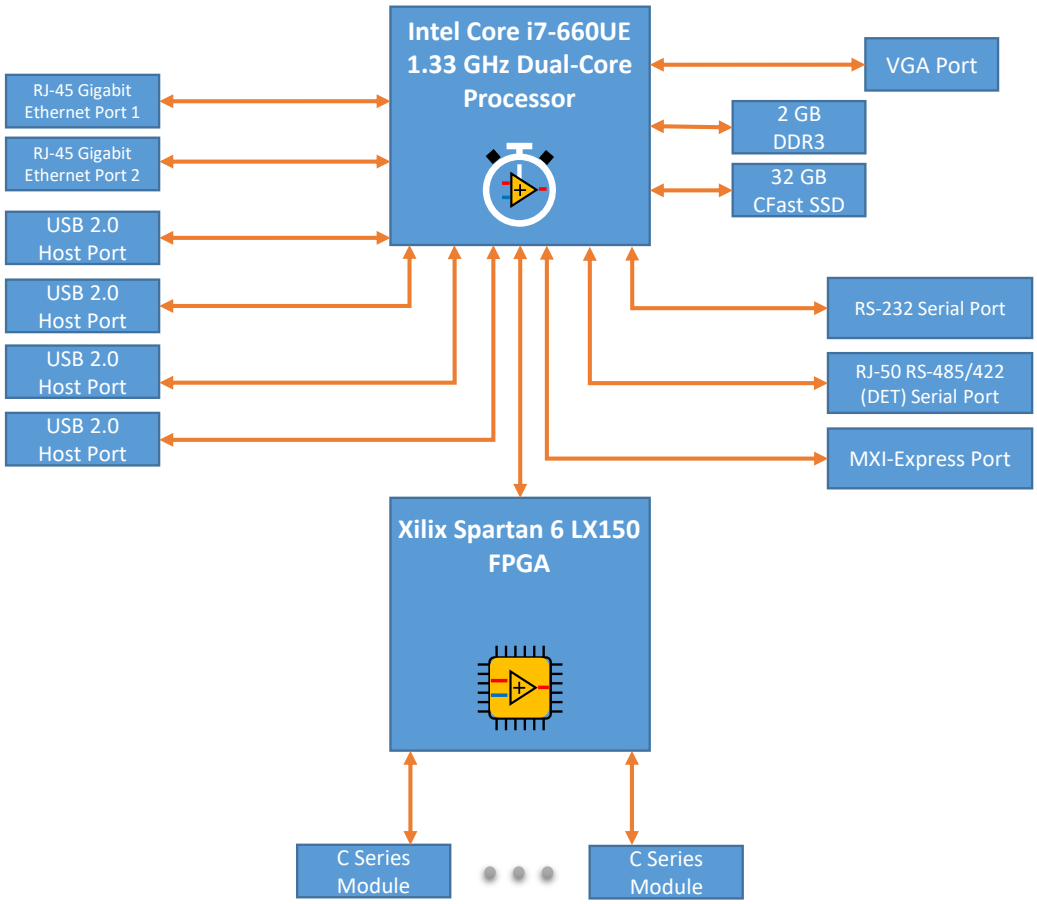


FIGURE 3.4: CompactRIO NI 9082 Structure.

I will list some more detailed characteristics below.

About processor:
The processor is the logic circuitry that response to and processes the basic instructions that drive a cRIO. It interprets most of the commands. CPUs will perform most basic arithmetic, logic and I/O operations, and allocate commands for the FPGA and components running in the cRIO.

Processor	
CPU	Intel Core i7-660UE
Number of cores	2
CPU frequency	1.33 GHz (base); 2.4 GHz (single-core Turbo mode)
On-die L2 cache	256 KB x2 (256 KB/core)
On-die L3 cache	4 MB (shared)

TABLE 3.1: Processor Features.

About FPGA:
An FPGA is a hardware circuit a user can program which to perform one or more logical operations. FPGAs are integrated circuits, or ICs, that are sets of circuits on a chip which is the "collection" portion. Those circuits, or arrays, are groups of gates, memory, or other elements of programmable logic.

Reconfigurable FPGA	
FPGA type	Xilinx Spartan-6 LX150
Number of flip-flops	184,304
Number of 6-input LUTs	92,152
Number of DSP slices	180
Available block RAM	4,824 kbit
Number of DMA channels	3
Number of logical interrupts	32

TABLE 3.2: FPGA Features.

About Ethernet Port: Ethernet is the typical wired local area network (LAN) or wide area network (WAN) system communication technology that enables the cRIO to communicate with each other through a protocol — a collection of rules or a common language of the network.

Network/Ethernet Port	
Number of ports	Xilinx Spartan-6 LX150
Network interface	1000Base-T Ethernet
Compatibility	IEEE 802.3
Communication rates	1000 Mbit/s
Maximum cabling distance	100 m/segment

TABLE 3.3: Ethernet Port Features.

Once the capabilities of the device have been summarized, we propose the following architecture:

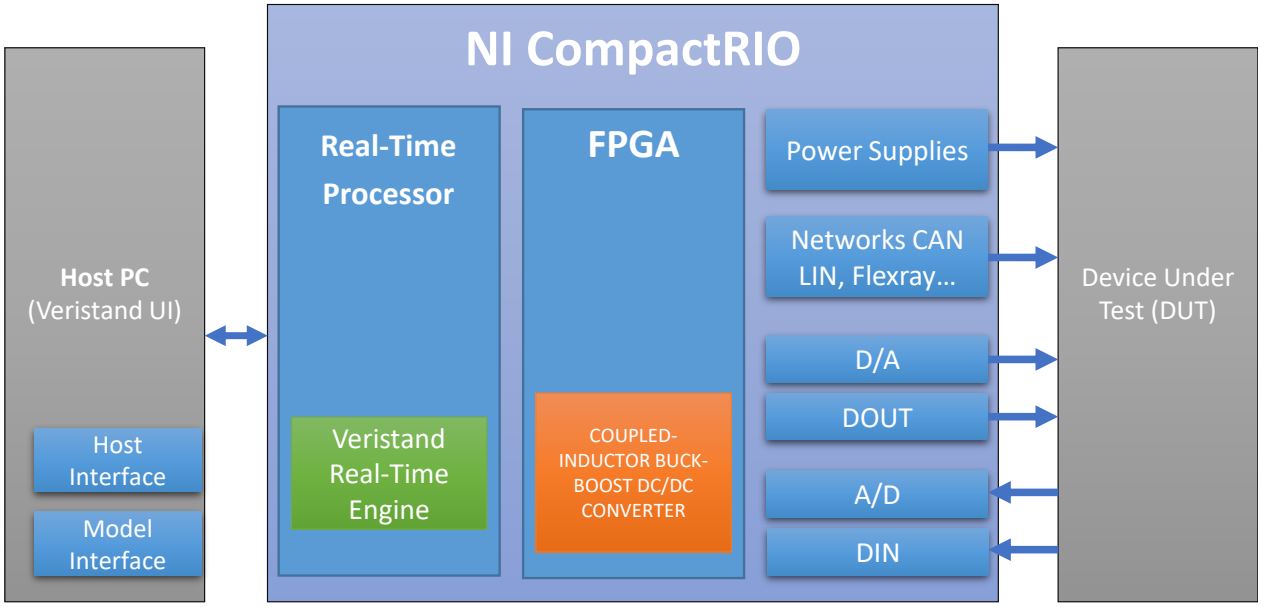


FIGURE 3.5: Hardware in the Loop Architecturewith CompactRIO.

3.5 Laboratory Virtual Instrument Engineering Workbench

The edit-time and run-time capabilities of the NI VeriStand framework can be customized using LabVIEW. Figure 3.6 displays the elements of the application's features. The blue components are configured using the NI VeriStand environment, as you may remember. You can use LabVIEW to build the white component and add it as a native component of the NI VeriStand application that works seamlessly with the environment.

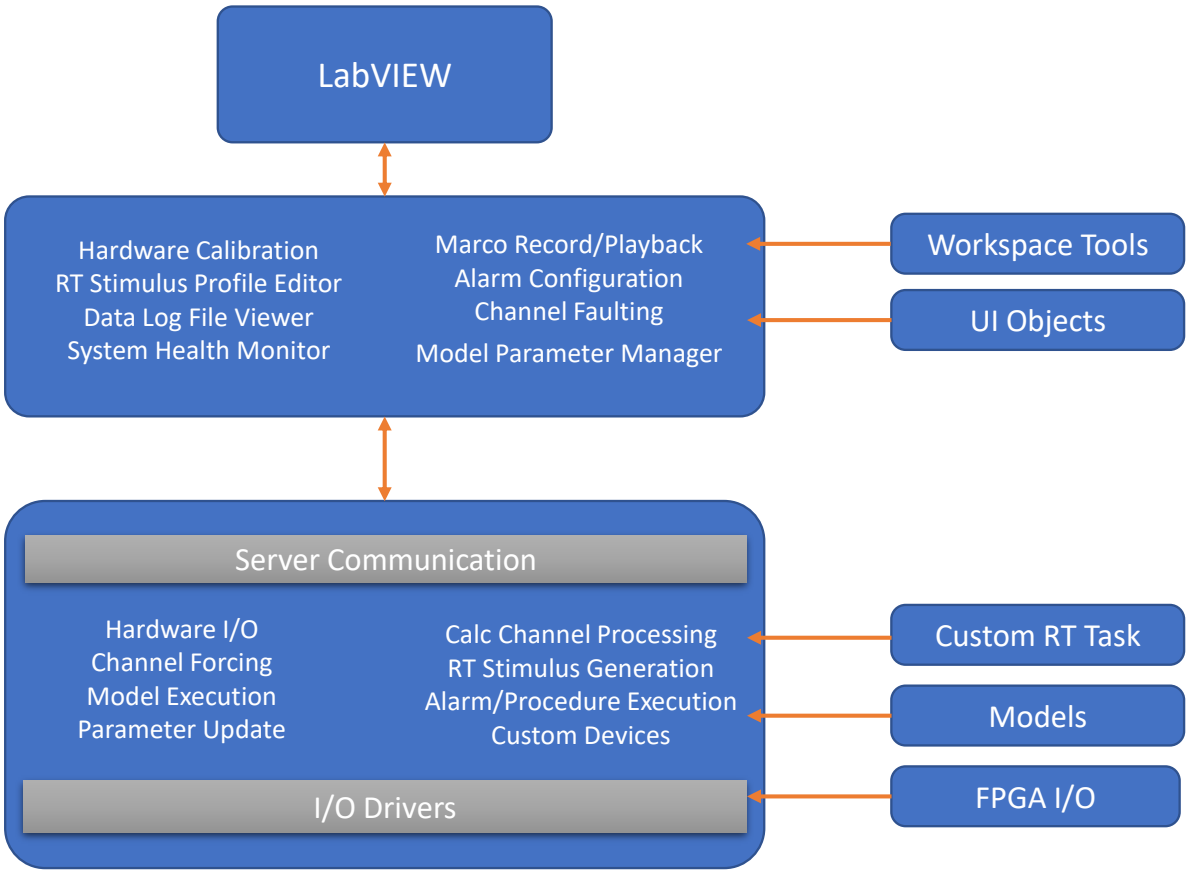


FIGURE 3.6: VeriStand Framework of configurable functionality created using LabVIEW.

3.5.1 Model

The most common way to use other environments to add functionality to NI VeriStand is to import compiled models into the real-time application of NI VeriStand. NI VeriStand can import compiled code from features or templates that you build from LabVIEW, MathWorks, Inc. Simulink ® applications, SimulationX from ITI, Maplesoft's MapleSim, and several other environments for modeling and programming.You can add closed-loop power, device emulation, signal processing, and signal generation to NI VeriStand applications in real-time with this capability. In Device Explorer, these components have a standard edit-time interface, making it easy to use compiled models in the same application from a variety of environments or move between compiled models from different environments.

Usually, these compiled versions of the models are created using a single configuration dialog from the C-code generation utility of the environment.

It latches the inputs; the code is executed, and it modifies the outputs every time the NI VeriStand real-time application calls the compiled model. Compared to each execution iteration, model parameters, or variables, are modified on demand within the model. The simplest technique to add custom functionality to your NI VeriStand real-time application is given by NI VeriStand models. Since we are working with discrete systems, we must use discretization techniques with the model we found in the previous chapter.

Here, we will need to convert the equations of state in continuous time to equations of state in discrete time to implement the model in the embedded system and later design a digital control system that allows achieving the desired behavior of the plant. I will show the procedure we used to achieve this below:

We can express the state equations for the continuous system as follows:

$$\begin{aligned} \dot{x}(t) &= A_{M_c} x(t) + B_{M_c} u(t) \\ y(t) &= C_{M_c} x(t) + D_{M_c} u(t) \end{aligned} \tag{3.1}$$

The solution of the above equation is:

$$x(t) = \phi_c(t - t_0)\, x(t_0) + \int_{t_0}^{t} \phi_c(t - \tau)\, B_{M_c} u(\tau) \cdot d\tau \tag{3.2}$$

Where t_0 is the initial time and we can express $\phi_c(t)$ as:

$$\phi_c(t) = e^{At} = \mathcal{L}^{-1}\left[\left(sI - A_{M_c}\right)^{-1}\right] \tag{3.3}$$

To get the discrete model of the system, we evaluate the previous equation in $t = T + kT$ with $kT = t_0$ and $u(t) = m(kT)$, which corresponds to the input of the system in the interval $kT \leq t \leq kT + T$. We must remember the procedure is only valid if $u(t)$ is the output of a zero order hold (ZOH). So:

$$x(t + kT) = \phi_c(T)\, x(kT) + m(kT) \int_{kT}^{T+kT} \phi_c(T + kT - \tau)\, B_{M_c} \cdot d\tau \tag{3.4}$$

We can express the state equations in discrete time as:

$$\begin{aligned} x(k+1) &= A_{M_d} x(k) + B_{M_d} u(k) \\ y(k) &= C_{M_d} x(k) + D_{M_d} u(k) \end{aligned} \tag{3.5}$$

For this reason we can conclude:

$$x_d(k) = x_c(kT) \tag{3.6}$$

While the state matrix is:

$$A_{M_d} = \phi_c(T) \tag{3.7}$$

The input matrix is:

$$A_{M_d} = \int_{kT}^{T+kT} \phi_c \left(T + kT - \tau\right) B_{M_c} \cdot d\tau \tag{3.8}$$

The matrices C and D of the system in discrete time are equal to the matrices C and D in continuous time. We can evaluate matrices A and B from the PhiC calculation using the procedure described above. For the extraction of the model, we calculate the values of PhiC function using the following expression:

$$\phi_c \left(T\right) = e^{A_{M_c} T} = I + A_{M_c} + \frac{A^2{}_{M_c} T}{2!} + \frac{A^3{}_{M_c} T}{3!} + \cdots \tag{3.9}$$

We reach a convergent series that can be truncated once for the above equation. Then, I will show an algorithm in MATLAB in which arrays of continuous-time system based on the model of the previous chapter are declared.

```
clc
clearvars
close all
% Continuous Space-State Model
% ----- State Matrix [Amc] -----
Amc = [ 0,  0,  0,  (D1*L - Lm2 - L + D1*Lm2 + D2*M)/(- M^2 + L*Lm1 + ...
    Lm1*Lm2), -(M - D1*M + D1*D2*M)/(- M^2 + L*Lm1 + Lm1*Lm2); ...
         0,  0,  0,  (D2*Lm1 - M + D1*M)/(- M^2 + L*Lm1 + Lm1*Lm2), -(Lm1 - ...
             D1*Lm1 + D1*D2*Lm1)/(- M^2 + L*Lm1 + Lm1*Lm2); ...
         0,  0,  -(D1*D2 - D1 + 1)/(Cd*Rd), (D1*D2 - D1 + 1)/(Cd*Rd), 0; ...
         -(D1 - 1)/Ci), -D2/Ci, (D1*D2 - D1 + 1)/(Ci*Rd), -(D1*D2 - D1 + ...
             1)/(Ci*Rd), 0; ...
         0,  (D1*D2 - D1 + 1)/Co, 0, 0, -(D1*D2 - D1 + 1)/(Co*RL)];
Amc = simplify(Amc);

% ----- Input Matrix [Bmc] -----
Bmc = [ (L + Lm2 - D1*L - D1*Lm2 + D1*D2*L + D1*D2*Lm2)/(- M^2 + L*Lm1 + ...
    Lm1*Lm2); ...
         (M - D1*M + D1*D2*M)/(- M^2 + L*Lm1 + Lm1*Lm2); ...
         0; ...
         0; ...
         0];
Bmc = simplify(Bmc);

% ----- Output Matrix -----
Cmc = eye(5);

% ----- Feedforward Matrix -----
Dmc = zeros(5,1);
```

ALGORITHM 3.1: Declaration of the continuous time system Matrix in MATLAB.

$$A_{M_c} = \begin{bmatrix} 0 & 0 & 0 & \frac{D_2M-\left(L+L_{m_2}\right)(1-D_1)}{L_{m_1}\left(L+L_{m_2}\right)-M^2} & -\frac{M(D_1D_2-D_1+1)}{L_{m_1}\left(L+L_{m_2}\right)-M^2} \\ 0 & 0 & 0 & \frac{D_1M+D_2Lm_1-M}{L_{m_1}\left(L+L_{m_2}\right)-M^2} & -\frac{L_{m_1}(D_1D_2-D_1+1)}{L_{m_1}\left(L+L_{m_2}\right)-M^2} \\ 0 & 0 & -\frac{D_1D_2-D_1+1}{C_dR_d} & \frac{D_1D_2-D_1+1}{C_dR_d} & 0 \\ \frac{1-D_1}{C_i} & -\frac{D_2}{C_i} & \frac{D_1D_2-D_1+1}{C_iR_d} & -\frac{D_1D_2-D_1+1}{C_iR_d} & 0 \\ 0 & \frac{D_1D_2-D_1+1}{C_o} & 0 & 0 & -\frac{D_1D_2-D_1+1}{C_oR_L} \end{bmatrix} \tag{3.10}$$

$$B_{M_c} = \begin{bmatrix} \frac{\left(L+L_{m_2}\right)(D_1D_2-D_1+1)}{L_{m_1}\left(L+L_{m_2}\right)-M^2} \\ \frac{M(D_1D_2-D_1+1)}{L_{m_1}\left(L+L_{m_2}\right)-M^2} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.11}$$

```
1   Phi_c = simplify(ilaplace(inv(s*eye(2) - Amc)));
2
3   Amd = subs(Phi_c,t,Ts);
4   Bmd = int(Phi_c,t,0,Ts)*Bmc;
5
6   Amd = simplify(Amd);
7   Bmd = simplify(Bmd);
8
9   pretty(Amd)
10  pretty(Bmd)
```

ALGORITHM 3.2: Declaration of the discrete time system Matrix in MATLAB.

Now, with the discrete time model of the converter, it will be programmed into the FPGA. We will see details of the process in the next section.

3.6 LabVIEW Model

Simulation models and control algorithms need to be integrated into LabVIEW and the LabVIEW Real-Time Module. The model loaded on the FPGA had to be implemented using Look At Tables since the parametric model implies having changes in the transfer functions of the system.

In LabVIEW, the Block Diagram objects include terminals, subVIs, functions, constants, structures, and wires, which transfer data along with other objects on the block diagram. The Block Diagram of the model will be shown in the Figure 3.7.

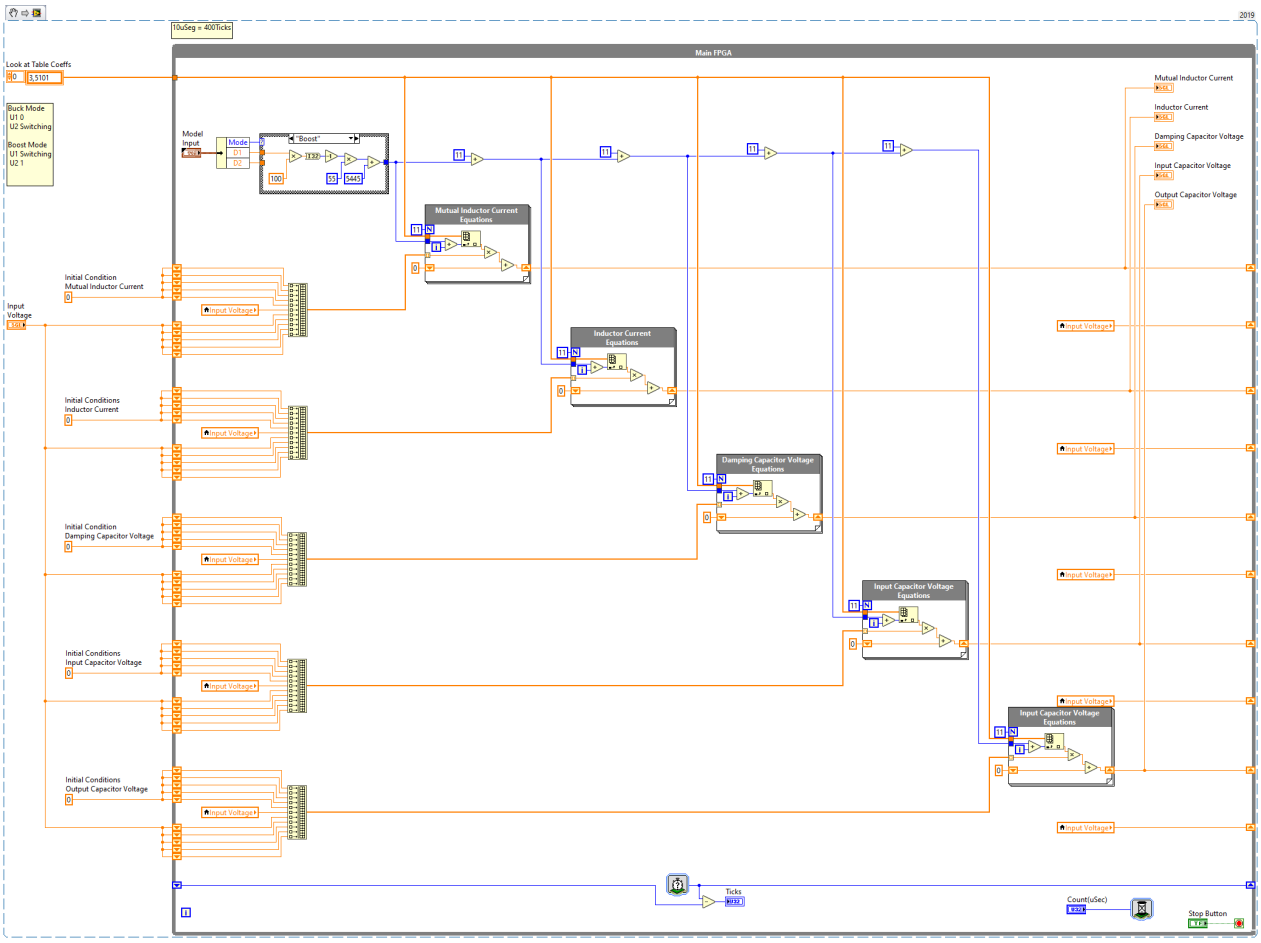


FIGURE 3.7: Converter Model on a FPGA.

On the other hand, the front panel window, which is normally the user interface for the VI, will be shown below, in Figure 3.8.

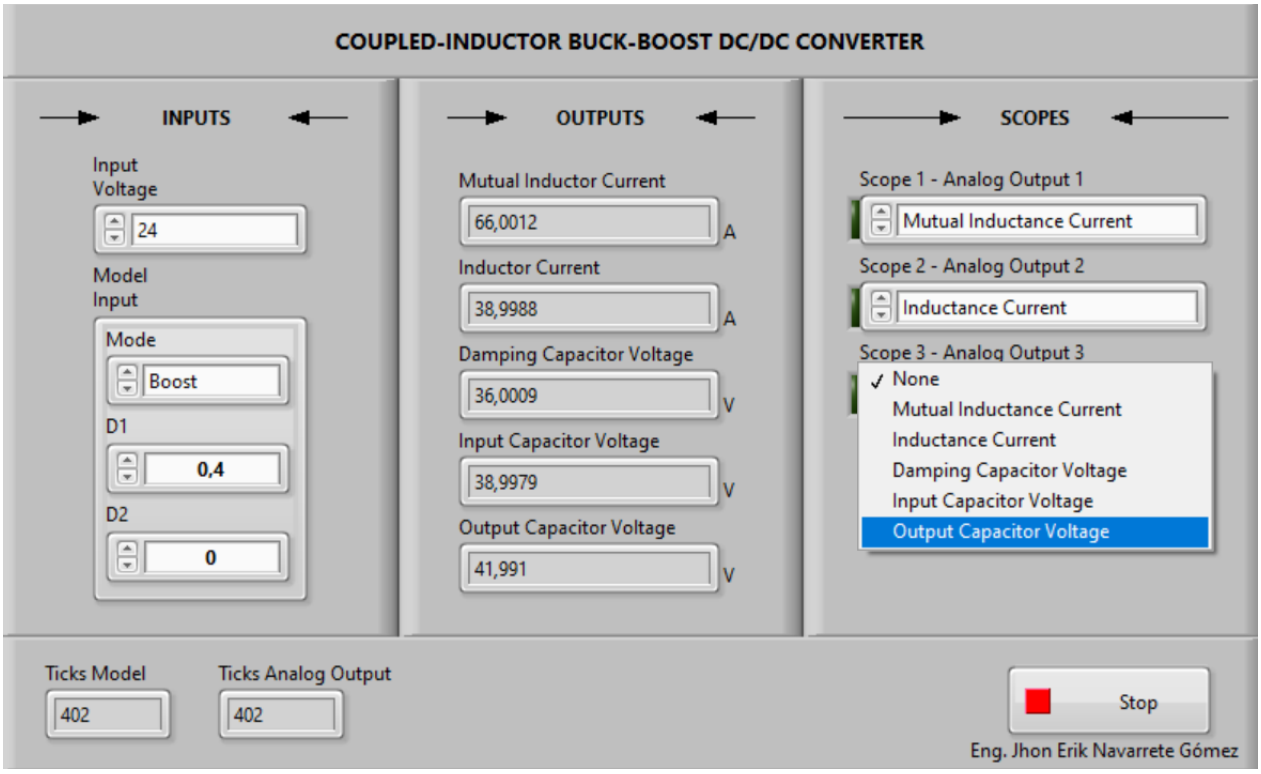


FIGURE 3.8: Front Panel of the FPGA VI.

At the time of loading the model, we obtained the following results:

```
1   Compilation completed successfully.
2
3   Device Utilization
4   ---------------------------
5   Total Slices: 20,9% (4809 out of 23038)
6   Slice Registers: 6,5% (11971 out of 184304)
7   Slice LUTs: 15,5% (14313 out of 92152)
8   Block RAMs: 7,8% (21 out of 268)
9   DSP48s: 5,6% (10 out of 180)
10
11  Timing
12  ---------------------------
13  MiteClk (Used by non-diagram components): 33,00 MHz (91,47 MHz maximum)
14  40 MHz Onboard Clock: 40,00 MHz (62,32 MHz maximum)
15  80MHz (Used by non-diagram components): 80,00 MHz (149,41 MHz maximum)
16
17  Compilation Time
18  ---------------------------
19  Date submitted: 18/10/2020 22:42
20  Date results were retrieved: 18/10/2020 22:56
21  Time waiting in queue: 00:09
22  Time compiling: 13:43
```

```
23      - Generate Xilinx IP: 00:00
24      - Estimate Resources - PlanAhead: 00:17
25      - Synthesize - XST: 04:30
26      - Translate: 01:10
27      - Map: 03:55
28      - Place & Route: 02:57
29      - Generate Programming File: 00:43
```

ALGORITHM 3.3: Build report on the FPGA.

The cRIO analog output module only has 3 outputs available. Taking this into account, we implemented an analog multiplexing to visualize each state variable.
This implies an advantage regarding the implementation in the FPGA since no more computation time is required if we implement it in parallel.
How we implemented analog multiplexing in the FPGA will be shown below:

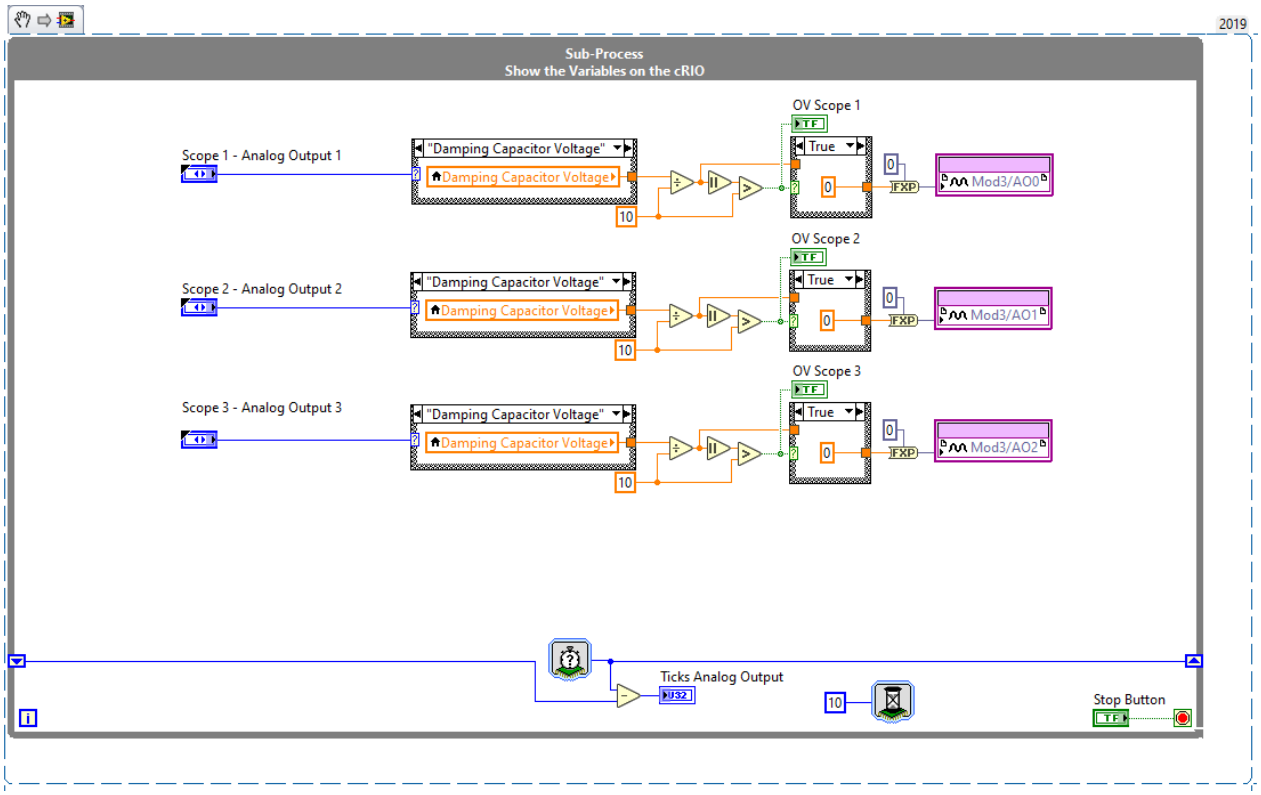


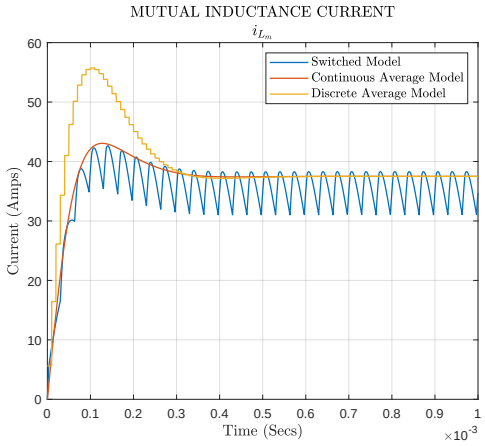
FIGURE 3.9: Analog Multiplexing Schematic.

3.7 Experimental Results

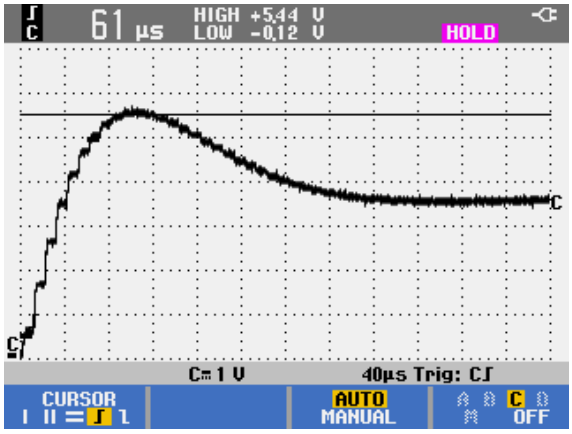
This section will show the comparison between the simulated state variables and those implemented as Hardware in the Loop. We will start with the mutual Inductance Current:

3.7.1 Open-Loop Experiments

In this section you will see the results of the model of the plant loaded in the FPGA:



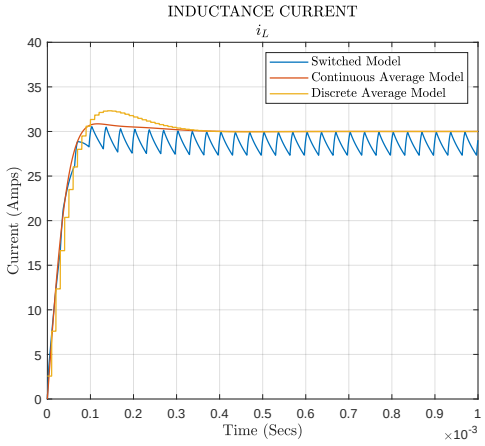
(A) Mutual Inductance Current on the Models.



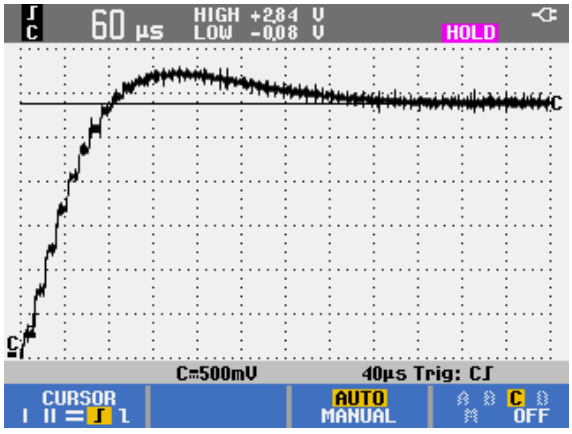
(B) Mutual Inductance Current on the Scope - cRIO Analog Output(x10).

FIGURE 3.10: Mutual Inductor Current Comparison.

For the Inductance Current:



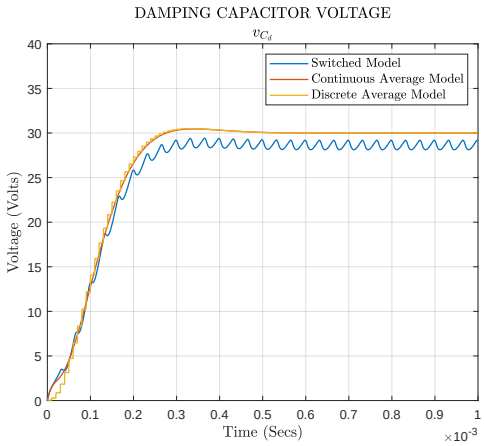
(A) Inductance Current on the Models.



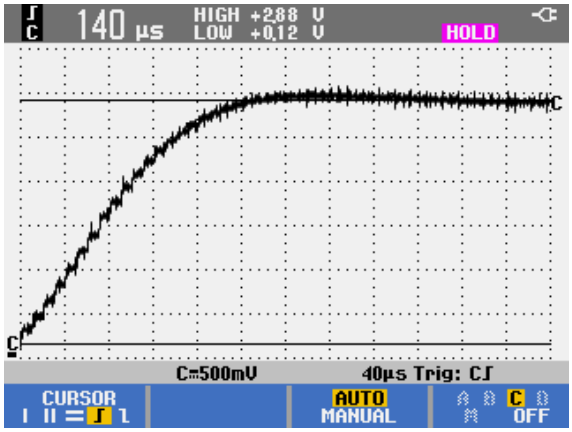
(B) Inductance Current on the Scope - cRIO Analog Output(x10).

FIGURE 3.11: Inductor Currents Comparison.

For the Damping Capacitor Voltage:



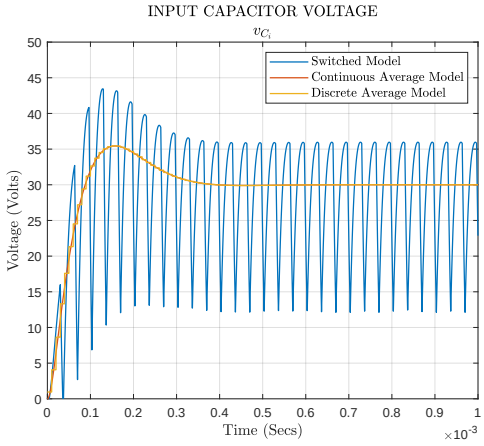
(A) Damping Capacitor Voltage on the Models.



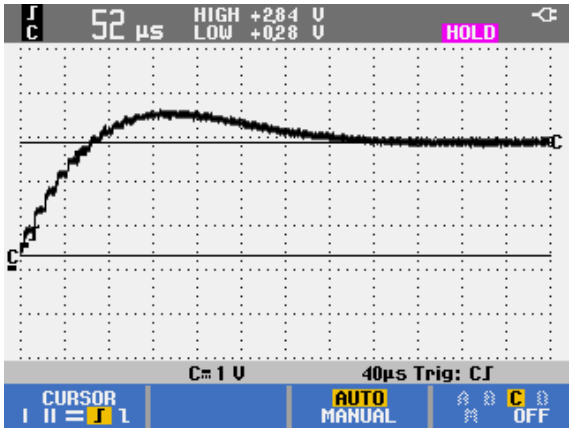
(B) Damping Capacitor Voltage on the Scope - cRIO Analog Output(x10).

FIGURE 3.12: Damping Capacitor Voltages Comparison.

For the Input Capacitor Voltage:



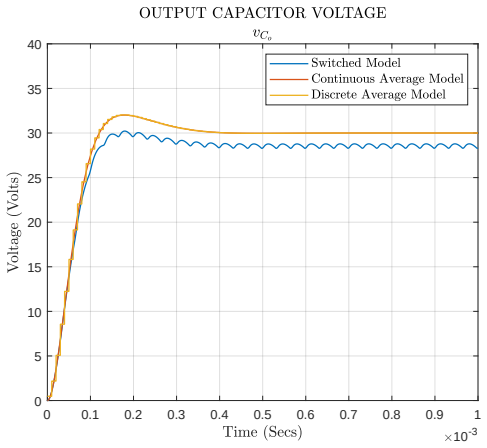
(A) Input Capacitor Voltage on the Models.



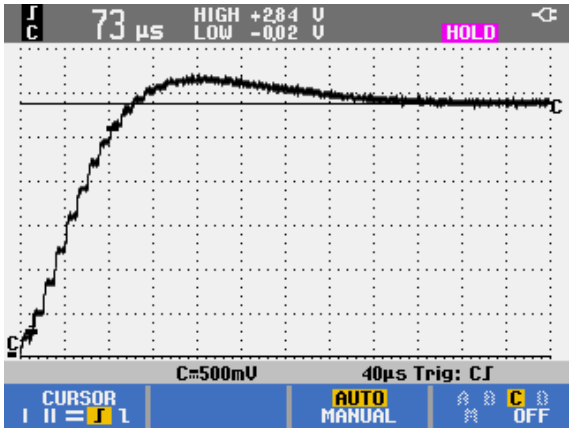
(B) Input Capacitor Voltage on the Scope - cRIO Analog Output(x10).

FIGURE 3.13: Input Capacitor Voltages Comparison.

For the Output Capacitor Voltage:



(A) Output Capacitor Voltages on the Models.



(B) Output Capacitor Voltages on the Scope - cRIO Analog Output(x10).

FIGURE 3.14: Output Capacitor Voltages Comparison.

3.7.2 Close-Loop Experiments

Our intention is to test the plant with a closed loop control system with the goal of regulating a process variable to a specific set point without human interaction. The variable that we will regulate in the output voltage the voltage of the output capacitor.

The implemented controller is of the Proportional-Integrative type. I will give design details below:

$$u(t) = k_p \left[e(t) + \frac{1}{\tau_i} \int e(t)\, dt \right] + M_o \tag{3.12}$$

Where:

$u(t)$	is the input signal of the plant
$e(t)$	is the error signal Regarding the reference
k_p	is the Proportional Gain Setting Parameter
τ_i	is the Integral Time Setting Parameter
M_o	is the Controller output for null error

For small sampling times T_s, it is possible to take the previous equation to an equation for differences in discretization, replacing the integral part by an accumulator, or what is the same, a sum. Continuous integration can be expressed using rectangular or trapezoidal integration as it suits the criteria. For this particular case, the trapezoidal integration will be used.
With particular integration we can express the total area under the curve as the sum of the areas of all trapezoids into which the area can be subdivided, that is:

$$\int e(t)\, dt = \sum T_s \left[\frac{e(k) + e(k+1)}{2} \right] \tag{3.13}$$

Replacing equation 3.13 in equation (3.12) and taking the values of $u(t)$ and $e(t)$ at the sampling instant k we can obtain:

$$\int e(t)\,dt = \sum T_s \left[\frac{e(k) + e(k+1)}{2}\right] \tag{3.14}$$

Thus:

$$u(k) = q_0 e(k) + q_1 e(k-1) + m(k-1) \tag{3.15}$$

Where:

$$q_0 = k_c\left[1 + \frac{T_s}{2\tau_i}\right] \tag{3.16}$$

$$q_1 = -k_c\left[1 - \frac{T_s}{2\tau_i}\right] \tag{3.17}$$

Taking the Z transform of the function gives:

$$C(Z) = \frac{u(Z)}{e(Z)} = \frac{q_0 + q_1 Z^{-1}}{1 - Z^{-1}} = \frac{q_0 Z + q_1}{Z - 1} \tag{3.18}$$

With this procedure we have found the way in which the controller will be implemented. The code shown below will find, based on the model from the previous chapter, find the small-signal model that will allow us to design the controller.

```
1   Amc = [ 0,  0,  0,  (D1*L - Lm2 - L + D1*Lm2 + D2*M)/(- M^2 + L*Lm1 + ...
        Lm1*Lm2), -(M*(D1*D2 - D1 + 1))/(- M^2 + L*Lm1 + Lm1*Lm2); ...
2           0,  0,  0,  (D2*Lm1 - M + D1*M)/(- M^2 + L*Lm1 + Lm1*Lm2), ...
                -(Lm1*(D1*D2 - D1 + 1))/(- M^2 + L*Lm1 + Lm1*Lm2); ...
3           0,  0,  -(D1*D2 - D1 + 1)/(Cd*Rd),  (D1*D2 - D1 + 1)/(Cd*Rd), 0; ...
4           -(D1 - 1)/Ci,  -D2/Ci,  (D1*D2 - D1 + 1)/(Ci*Rd),  -(D1*D2 - D1 + ...
                1)/(Ci*Rd), 0; ...
5           0,  (D1*D2 - D1 + 1)/Co,  0,  0, -(D1*D2 - D1 + 1)/(Co*RL)];
6
7   Bmc = [ (L*VCI - L*Vs + Lm2*VCI - Lm2*Vs + M*VCO + D2*L*Vs + D2*Lm2*Vs - ...
        D2*M*VCO)/(- M^2 + L*Lm1 + Lm1*Lm2), (M*VCI + D1*L*Vs + D1*Lm2*Vs - ...
        D1*M*VCO)/(- M^2 + L*Lm1 + Lm1*Lm2); ...
8           (Lm1*VCO + M*VCI - M*Vs - D2*Lm1*VCO + D2*M*Vs)/(- M^2 + L*Lm1 + ...
                Lm1*Lm2), (Lm1*VCI - D1*Lm1*VCO + D1*M*Vs)/(- M^2 + L*Lm1 + ...
                Lm1*Lm2); ...
9           -((VCD - VCI)*(D2 - 1))/(Cd*Rd), -(D1*(VCD - VCI))/(Cd*Rd); ...
10          -(VCD - VCI - D2*VCD + D2*VCI + ILM1*Rd)/(Ci*Rd), -(D1*VCI - D1*VCD ...
                + IL*Rd)/(Ci*Rd); ...
11          -((D2 - 1)*(VCO - IL*RL))/(Co*RL), -(D1*(VCO - IL*RL))/(Co*RL)];
12
13  format short
14  Amc = subs(Amc, ILM1, (D2^2*Vs)/(RL*(D1 - 1)^2));
15  Amc = subs(Amc, IL, -(D2*Vs)/(RL*(D1 - 1)));
16  Amc = subs(Amc, VCD, -(Vs - D1*Vs + D1*D2*Vs)/(D1 - 1));
17  Amc = subs(Amc, VCI, -(Vs - D1*Vs + D1*D2*Vs)/(D1 - 1));
18  Amc = subs(Amc, VCO, -(D2*Vs)/(D1 - 1));
19  Amc = simplify(Amc);
20
21  Bmc = subs(Bmc, ILM1, (D2^2*Vs)/(RL*(D1 - 1)^2));
22  Bmc = subs(Bmc, IL, -(D2*Vs)/(RL*(D1 - 1)));
23  Bmc = subs(Bmc, VCD, -(Vs - D1*Vs + D1*D2*Vs)/(D1 - 1));
```

```
Bmc = subs(Bmc, VCI, -(Vs - D1*Vs + D1*D2*Vs)/(D1 - 1));
Bmc = subs(Bmc, VCO, -(D2*Vs)/(D1 - 1));
Bmc = simplify(Bmc);

% Replace Elements Values ---------------------------------
Amc = subs(Amc, Vs, 24);
    % Source Voltage
Amc = subs(Amc, D1, 0.4286);
    % Duty Cycle 1
Amc = subs(Amc, D2, 1);
    % Duty Cycle 2
% Circuit Elements
Amc = subs(Amc, Lm1, 35e-6);
Amc = subs(Amc, Lm2, 35e-6);
Amc = subs(Amc, L, 35e-6);
Amc = subs(Amc, M, 0.9*sqrt(35e-6*35e-6));
Amc = subs(Amc, Rd, 1.5);
Amc = subs(Amc, Cd, 66e-6);
Amc = subs(Amc, Ci, 7e-6);
Amc = subs(Amc, Co, 22e-6);
Amc = subs(Amc, RL, 1);
Amc = double(Amc)

Bmc = subs(Bmc, Vs, 24);
    % Source Voltage
Bmc = subs(Bmc, D1, 0.4286);
    % Duty Cycle 1
Bmc = subs(Bmc, D2, 1);
    % Duty Cycle 2
% Circuit Elements
Bmc = subs(Bmc, Lm1, 35e-6);
Bmc = subs(Bmc, Lm2, 35e-6);
Bmc = subs(Bmc, L, 35e-6);
Bmc = subs(Bmc, M, 0.9*sqrt(35e-6*35e-6));
Bmc = subs(Bmc, Rd, 1.5);
Bmc = subs(Bmc, Cd, 66e-6);
Bmc = subs(Bmc, Ci, 7e-6);
Bmc = subs(Bmc, Co, 22e-6);
Bmc = subs(Bmc, RL, 1);
Bmc = double(Bmc)

% ----- Output Matrix -----
Cmc = [0 0 0 0 1];

% ----- Feedforward Matrix -----
Dmc = 0;

[NewNum,NewDen] = ss2tf(Amc,Bmc(:,2),Cmc,Dmc);
H = tf(NewNum,NewDen);
Ts = 10e-6;
H = c2d(H,Ts,'tustin')
```

ALGORITHM 3.4: Algorithm to find the discrete average model.

Now, with the small-signal model, we find the discrete-time transfer function of the output voltage and calculate the controller with the following characteristics:

- Steady-state error equal to zero.

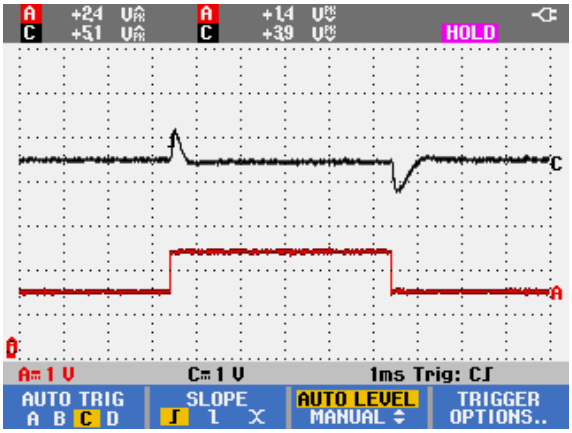
- Maximum overshoot less than 10%.
- Settling time less than 1mSec.

The compensator found of the proportional integrative type is of the form:

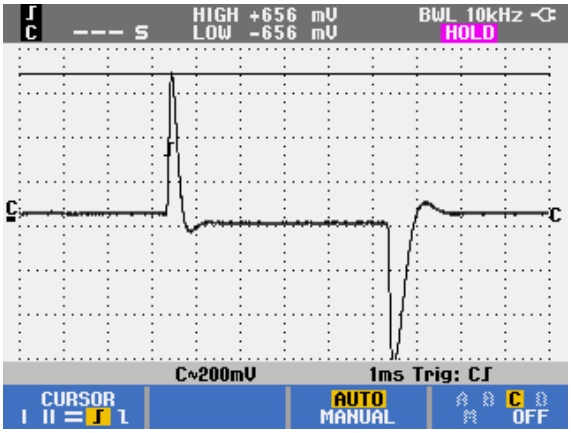
$$Comp\,(Z) = \frac{0.0027z - 0.0018}{z - 1} \tag{3.19}$$

The compensator has been designed by conventional methods; I expect it for future work to use advanced and robust control techniques that provide better performance.

I will show the results of establishing a control loop for the plant and testing it against different disturbances below:



(A) Source Disturbance, Reference: 42v, Input Source from: 12v to 25v.



(B) Load Disturbance from 20% to 100%. Maximum Power: 2kW.

FIGURE 3.15: Testing the control law against different disturbances.

The answer in MATLAB is quite similar and will be shown below:

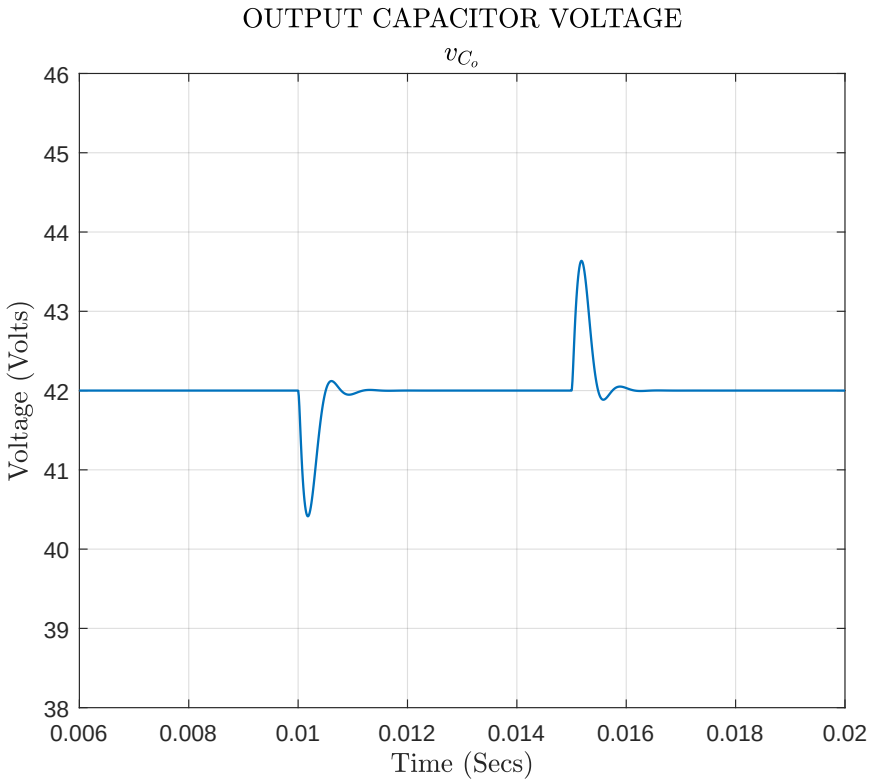


FIGURE 3.16: System response in Simulink/MATLAB with Input Source Perturbance.

This concludes the project and we will enter the conclusions.

Chapter 4

Conclusions

4.1 Power Electronics

An input current control for an 1000W coupled-inductor buck-boost dc-dc switching converter has been designed and simulated in this work. This control technique would make the converter also suitable for large voltage conversion ratio applications like unregulated sources such as battery charging and discharging, power factor correction, fuel cell regulation, maximum power tracking of solar panels, among others. Additionally, the zero crossing saturation problem in the low-level analog current controllers could be addressed by means of this state- feedback controller. The theoretical analyses have been simulated and validated by means of MATLAB and LABVIEW.

Throughout this thesis, I confirm the principle of RTS. The coherence between the signals shows that, for control and maintenance purposes, the virtual power system operates just like the real one and can therefore be replaced by it. However, Compact-RIO in Off-line was used to generating the simulation results shown above.

The outcomes got were coherent. This strategy enables us not only to improve the maintenance and operation of the system but also to optimize both management and control processes, reduce the cost and time of any operation, while also reducing emissions because of the unnecessary use of materials needed in the implementation process.

4.2 Identification and Control

In this article, we built a digital power controlled with hardware-in-the-loop for the Coupled Inductor Buck-Boost DC / DC and validated the design. The RT-HIL research method was first reviewed. With the device models designed for an orbital period on the real-time HIL platform, the control algorithm implemented on a dedicated micro-controller was tested in real-time.

The results of RT-HIL testing show that the built system configuration and parameters meet the system specifications and that the power converters and power flow on the system are controlled by the established control algorithm. The prototype is a powerful platform for automated power controls for real-time hardware-in-the-loop research.

With an efficient method of hardware-in-the-loop (HIL) simulation, we can test embedded control systems more effectively. The availability or cost considerations will make it impractical to carry out all the required tests by conventional methods, for future projects, it will be possible to simulate the parts of the device that pose these challenges using HIL simulation. It will also be possible to

keep reliability and time-to-market specifications in a cost-effective manner, even as the systems being tested become more complex.

This degree work contains all the functionality required to help develop firmware dedicated to the testing of ECUs and controllers found in power electronics applications. A comprehensive library of LabVIEW functions and project templates were created with the purpose that all configurations perfectly suited to NI CompactRIO and PXI hardware platforms.

Bibliography

[1] D. Biel, F. Guinjoan, E. Fossas, and J. Chavarria, "Sliding-mode control design of a boost-buck switching converter for AC signal generation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 8, pp. 1539–1551, Aug. 2004, ISSN: 10577122. DOI: 10.1109/TCSI.2004.832803.

[2] J. Chen, D. Maksimović, and R. W. Erickson, "Analysis and design of a low-stress buck-boost converter in universal-input PFC applications," *IEEE Transactions on Power Electronics*, vol. 21, no. 2, pp. 320–329, Mar. 2006, ISSN: 08858993. DOI: 10.1109/TPEL.2005.869744.

[3] M. Gaboriault and A. Notman, "A high efficiency, non-inverting, buck-boost DC-DC converter," in *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, vol. 3, 2004, pp. 1411–1415. DOI: 10.1109/apec.2004.1296049.

[4] P. C. Huang, W. Q. Wu, H. H. Ho, and K. H. Chen, "Hybrid buckboost feedforward and reduced average inductor current techniques in fast line transient and high-efficiency buck-boost converter," *IEEE Transactions on Power Electronics*, vol. 25, no. 3, pp. 719–730, 2010, ISSN: 08858993. DOI: 10.1109/TPEL.2009.2031803.

[5] B. Sahu and G. A. Rincon-Mora, "A low voltage, dynamic, noninverting, synchronous buck-boost converter for portable applications," *IEEE Transactions on Power Electronics*, vol. 19, no. 2, pp. 443–452, Mar. 2004, ISSN: 08858993. DOI: 10.1109/TPEL.2003.823196.

[6] Y. J. Lee, A. Khaligh, and A. Emadi, "A compensation technique for smooth transitions in a noninverting buck-boost converter," *IEEE Transactions on Power Electronics*, vol. 24, no. 4, pp. 1002–1015, 2009, ISSN: 08858993. DOI: 10.1109/TPEL.2008.2010044.

[7] Y. J. Lee, A. Khaligh, A. Chakraborty, and A. Emadi, "Digital Combination of Buck and Boost Converters to Control a Positive Buck-Boost Converter and Improve the Output Transients," *IEEE Transactions on Power Electronics*, vol. 24, no. 5, pp. 1267–1279, 2009, ISSN: 19410107. DOI: 10.1109/TPEL.2009.2014066.

[8] E. Schaltz, P. O. Rasmussen, and A. Khaligh, "Non-inverting buck-boost converter for fuel cell applications," in *IECON Proceedings (Industrial Electronics Conference)*, IEEE Computer Society, 2008, pp. 855–860, ISBN: 9781424417667. DOI: 10.1109/IECON.2008.4758065.

[9] J. K. Shiau, C. J. Cheng, and C. E. Tseng, "Stability analysis of a non-inverting synchronous buck-boost power converter for a solar power management system," in *2008 IEEE International Conference on Sustainable Energy Technologies, ICSET 2008*, 2008, pp. 263–268, ISBN: 9781424418886. DOI: 10.1109/ICSET.2008.4747014.

[10] O. Mourra, A. Fernandez, and F. Tonicello, "Buck boost regulator (B2R) for spacecraft solar array power conversion," in *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, 2010, pp. 1313–1319, ISBN: 9781424447824. DOI: 10.1109/APEC.2010.5433399.

[11] S. Waffler and J. W. Kolar, "A Novel Low-Loss Modulation Strategy for High-Power Bidirectional Buck + Boost Converters," *IEEE Transactions on Power Electronics*, vol. 24, no. 6, pp. 1589–1599, 2009, ISSN: 19410107. DOI: 10.1109/TPEL.2009.2015881.

[12] X. Ren, X. Ruan, H. Qian, M. Li, and Q. Chen, "Three-mode dual-frequency two-edge modulation scheme for four-switch buck-boost converter," *IEEE Transactions on Power Electronics*, vol. 24, no. 2, pp. 499–509, 2009, ISSN: 08858993. DOI: 10.1109/TPEL.2008.2005578.

[13] D. M. Sable, B. H. Cho, and R. B. Ridley, "Elimination of the positive zero in fixed frequency boost and flyback converters," in *Fifth Annual Proceedings on Applied Power Electronics Conference and Exposition*, 1990, pp. 205–211. DOI: 10.1109/APEC.1990.66413.

[14] Wei-Chung Wu, R. M. Bass, and J. R. Yeargan, "Eliminating the effects of the right-half plane zero in fixed frequency boost converters," in *PESC 98 Record. 29th Annual IEEE Power Electronics Specialists Conference (Cat. No.98CH36196)*, vol. 1, May 1998, 362–366 vol.1. DOI: 10.1109/PESC.1998.701924.

[15] S.-K. E., F. A., M. E., E. J.B., E. V., C. J., and G. A., "Bidirectional High-Power High-Efficiency non-isolated step-up DC-DC Converter," 2003.

[16] J. Calvente, L. Martinez-Salamero, H. Valderrama, and E. Vidal-Idiarte, "Using magnetic coupling to eliminate right half-plane zeros in boost converters," *IEEE Power Electronics Letters*, vol. 2, no. 2, pp. 58–62, 2004, ISSN: 15407985. DOI: 10.1109/LPEL.2004.834615.

[17] B. Poorali and E. Adib, "Right-Half-Plane Zero Elimination of Boost," vol. 66, no. 11, pp. 8454–8462, 2019.

[18] C. Restrepo, J. Calvente, A. Cid-Pastor, A. El Aroudi, and R. Giral, "A noninverting buck-boost dc-dc switching converter with high efficiency and wide bandwidth," *IEEE Transactions on Power Electronics*, vol. 26, no. 9, pp. 2490–2503, 2011, ISSN: 08858993. DOI: 10.1109/TPEL.2011.2108668.

[19] C. Restrepo, S. Member, J. Calvente, A. Romero, E. Vidal-idiarte, R. Giral, and S. Member, "Current-Mode Control of a Coupled-Inductor Buck – Boost DC – DC Switching Converter," vol. 27, no. 5, pp. 2536–2549, 2012.

[20] J. Calvente, L. Martínez-Salamero, P. Garcés, and A. Romero, "Zero Dynamics-Based Design of Damping Networks for Switching Converters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1292–1303, 2003, ISSN: 00189251. DOI: 10.1109/TAES.2003.1261129.

[21] C. Restrepo, T. Konjedic, J. Calvente, M. Milanovič, and R. Giral, "Fast transitions between current control loops of the coupled-inductor buck-boost DC-DC switching converter," *IEEE Transactions on Power Electronics*, vol. 28, no. 8, pp. 3648–3652, 2013, ISSN: 08858993. DOI: 10.1109/TPEL.2012.2231882.

[22] M. N. Alexander, Charles K. and Sadiku, *Fundamentals of Electric Circuits*, Fifth Edit. Cleveland, Ohio: McGraw-Hill, 2013, p. 905, ISBN: 978-0-07-338057-5.

[23] Z. Zhang, "Coupled-Inductor Magnetics," Ph.D. dissertation, California Institute of Technology, 1987, p. 243.

[24] M. A. Mannah, A. Haddad, and H. Bazzi, "Hardware in the loop simulation for optimal management of electrical power converters," *2014 International Conference on Renewable Energies for Developing Countries, REDEC 2014*, pp. 43–48, 2014. DOI: 10.1109/REDEC.2014.7038529.

[25] W. Ren, M. Steurer, and S. Woodruff, "Progress and challenges in real time hardware-in-the loop simulations of integrated ship power systems," *2005 IEEE Power Engineering Society General Meeting*, vol. 1, pp. 534–537, 2005. DOI: 10.1109/pes.2005.1489721.

[26] A. Monti, H. Figueroa, S. Lentijo, X. Wu, and R. Dougal, "Interface issues in hardware-in-the-loop simulation," *2005 IEEE Electric Ship Technologies Symposium*, vol. 2005, pp. 39–44, 2005. DOI: 10.1109/ESTS.2005.1524650.

[27] Z. Jiang, R. A. Dougal, R. Leonard, H. Figueroa, and A. Monti, "Hardware-in-the-loop testing of digital power controllers," *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, vol. 2006, pp. 901–906, 2006. DOI: 10.1109/apec.2006.1620645.

[28] B. Lu, X. Wu, H. Figueroa, and A. Monti, "A low-cost real-time hardware-in-the-loop testing approach of power electronics controls," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 919–931, 2007, ISSN: 02780046. DOI: 10.1109/TIE.2007.892253.

[29] J. Wu, Y. Cheng, A. K. Srivastava, N. N. Schulz, and H. L. Ginn, "Hardware in the Loop test for power system modeling and simulation," *2006 IEEE PES Power Systems Conference and Exposition, PSCE 2006 - Proceedings*, pp. 1892–1897, 2006. DOI: 10.1109/PSCE.2006.296201.

[30] M. Steurer, F. Bogdan, W. Ren, M. Sloderbeck, and S. Woodruff, "Controller and power hardware-in-loop methods for accelerating renewable energy integration," *2007 IEEE Power Engineering Society General Meeting, PES*, pp. 44–47, 2007. DOI: 10.1109/PES.2007.386022.

[31] L. A. Gregoire, K. Al-Haddad, and G. Nanjundaiah, "Hardware-in-the-Loop (HIL) to reduce the development cost of power electronic converters," *India International Conference on Power Electronics, IICPE 2010*, 2011. DOI: 10.1109/IICPE.2011.5728153.

[32] M. Lemaire, P. Sicard, and J. Belanger, "Prototyping and Testing Power Electronics Systems Using Controller Hardware-In-the-Loop (HIL) and Power Hardware-In-the-Loop (PHIL) Simulations," *2015 IEEE Vehicle Power and Propulsion Conference, VPPC 2015 - Proceedings*, pp. 2–7, 2015. DOI: 10.1109/VPPC.2015.7353000.