Application Note

Implementation of Single-Phase Off-Grid Inverter With Digital Control Using PLECS® Simulation

Yishan Chen

ABSTRACT

Simulation is an effective method for studying the feasibility and performance of systems, including converter and control algorithms. Using code to realize digital control in simulation tools can be more flexible and similar to using C2000™ control. This application note introduces how to implement a single-phase, off-grid inverter with all digital control in a simulation tool and provides a verification method for off-grid control in the PMP23338 TI reference design. Voltage and current loops with a PI compensator are used in the control algorithm. A true RMS calculation block is configured in the voltage loop as the input sample signal. Totem-pole modulation is used to increase the efficiency of the GaN-based inverter or in power factor correction (PFC) circuits like PMP23338. A detailed implementation and code for each block are included in this application note, and the simulation results verify the correctness of the code and simulation model.

Table of Contents

1 Introduction....2
2 Overview of Inverter Model....3
2.1 Power Stage....3
2.2 Control and Modulation Method....4
3 Realization of Digital Control Using C-Script Block....6
3.1 Overview of Project Structure....6
3.2 AC Voltage Sampling and True RMS Value Calculation....8
3.3 Multiple Loop Control....8
4 Simulation Results....12
5 Summary....14
6 References....14

List of Figures

Figure 2-1. Inverter Model Overview....3
Figure 2-2. Control Method Block Diagram....4
Figure 2-3. Totem-Pole Modulation....4
Figure 2-4. Operation Mode in Each State....5
Figure 3-1. Project Structure Image....6
Figure 3-2. C-Script Setup....7
Figure 3-3. Input Signal Definitions....7
Figure 3-4. Variables Initialized in Start Function Code....7
Figure 3-5. AC Voltage Sampling Code in Update Function....7
Figure 3-6. Declaration in Output Function....7
Figure 3-7. AC Voltage Sampling and RMS Value Calculation Code....8
Figure 3-8. Initialization of Notch Filter....8
Figure 3-9. Voltage Loop With Notch Filter....9
Figure 3-10. Current Loop Code....9
Figure 3-11. Sinusoidal Wave Generator....10
Figure 3-12. Sawtooth Wave Generator....10
Figure 3-13. Background Dead-Time Counter in Update Function....11
Figure 3-14. Totem-Pole Control Code in Positive Cycle....11
Figure 4-1. Start-Up Process....12
Figure 4-2. Load Transient (100% to 50% to 10%)....13

Figure 4-3. Load Transient (50% to 100% to 50%)....13
Figure 4-4. True RMS Calculation in Load Transient....13

List of Tables

Table 2-1. Power Stage Specifications....3
Table 2-2. Switching Mode....5
Table 4-1. THD% and Settle Time....13

Trademarks

C2000™ is a trademark of Texas Instruments.
PLECS® is a registered trademark of Plexim GmbH.
All trademarks are the property of their respective owners.

1 Introduction

Inverters are widely used in various applications. For example, solar inverters, uninterruptible power supplies (UPS), and onboard chargers (OBC) utilize an inverter to convert DC power to AC. In these systems, the inverter performance has a significant impact on the performance of the entire system. So, many topologies and control algorithms are proposed and investigated to improve the performance of the inverter. One effective way to preliminarily verify the feasibility and performance of the converter is simulation. Besides, digital control is widely used to control the power stage in realistic situations and code is more flexible than a control block in simulation software. Hence, the purpose of this application note is to introduce the implementation of a single-phase off-grid inverter with digital control, and another purpose is to verify the performance of totem-pole modulation with multiple loop control. In conventional control methods, the input signal of the voltage loop comes directly from the sampling AC voltage in output. But in this application note, the RMS value of the output voltage is used as the reference value in the voltage loop. The benefit of this method is discussed in this note.

The PMP23338 is TI's 3.6kW, single-phase totem-pole bridgeless power factor correction (PFC) circuits reference design with e-meter function. The totem-pole bridgeless structure allows this reference design to support the inverter function. Two active half bridges form this topology: one half bridge works at a low frequency and the other one works at a high frequency.

The high-frequency half bridge in the PMP23338 consists of two LMG3522R030 devices. LMG3522R030 is TI's 650V GaN FET with an integrated driver and protection. Compared with Si FETs, the GaN FETs switching loss is much lower due to the lower C_{OSS}. In addition, the unique structure of the GaN FETS brings zero reverse recovery charge, which further improves the performance of GaN FETs at high frequencies.

2 Overview of Inverter Model

2.1 Power Stage

Figure 2-1 shows an overview of the inverter model, where the *C-Script* represents the digital controller that is used to sample data, calculate control parameters, and output the PWM signal. Power stage parameter is the same as the PMP23338 reference design, and the specifications are listed in Table 2-1. Conventional H-bridge topology with LC output filter is used. TI GaN devices LMG352xR030 are used in the high-frequency bridge and common Si FETs are used as the low-frequency bridge (50/60Hz).

Table 2-1. Power Stage Specifications

Parameter	Specifications	Units
Input Voltage	380	V
Output Voltage	220	V_{RMS}
Output Voltage Frequency	50	Hz
Output Power	3.6	kW
Output Current	16	A_{RMS}
PWM Frequency (GaN Bridge)	100	kHz
PWM Frequency (SI Bridge)	50	Hz

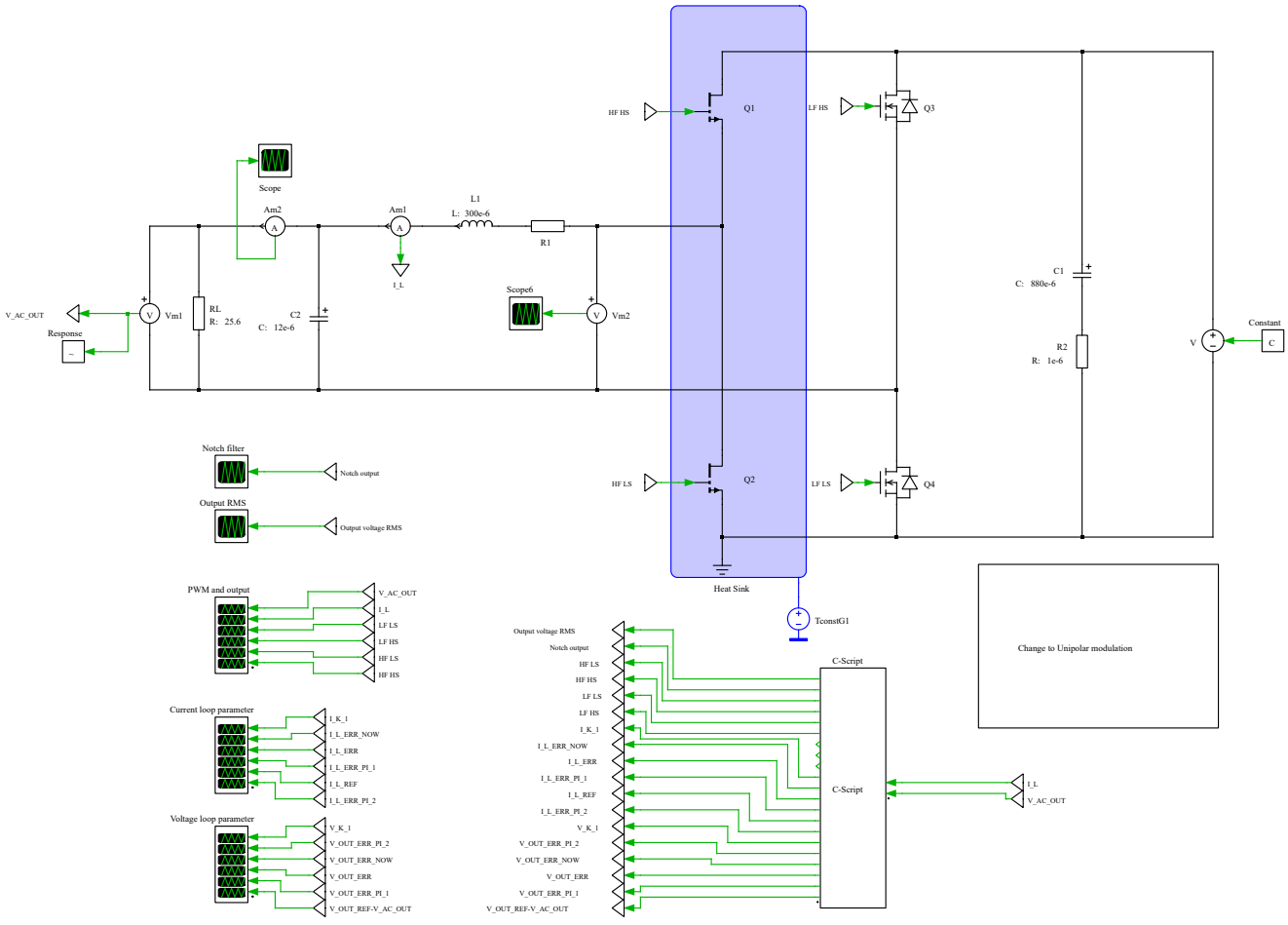


Figure 2-1. Inverter Model Overview

This model also provides an interface to the switch modulation method, simply change to unipolar modulation by copying the output PWM signal into the unipolar modulation block.

2.2 Control and Modulation Method

A common control method for off-grid inverters is multiple-loop control with a PI compensator. The output of the voltage loop is the reference value for the current loop. In this model, the common control method is utilized except that the voltage reference and sampling signal is the RMS value of output voltage. Besides, an additional notch filter is used to attenuate an amplitude of 100Hz ripple of output voltage. Conventional voltage loop directly uses sampling AC voltage to produce the error signal. In this application note, using the RMS value can quickly switch the voltage reference to the DC bus voltage when the inverter needs to connect to the grid and can also reduce the issue caused by sampling noise. Figure 2-2 shows a block diagram of the control system.

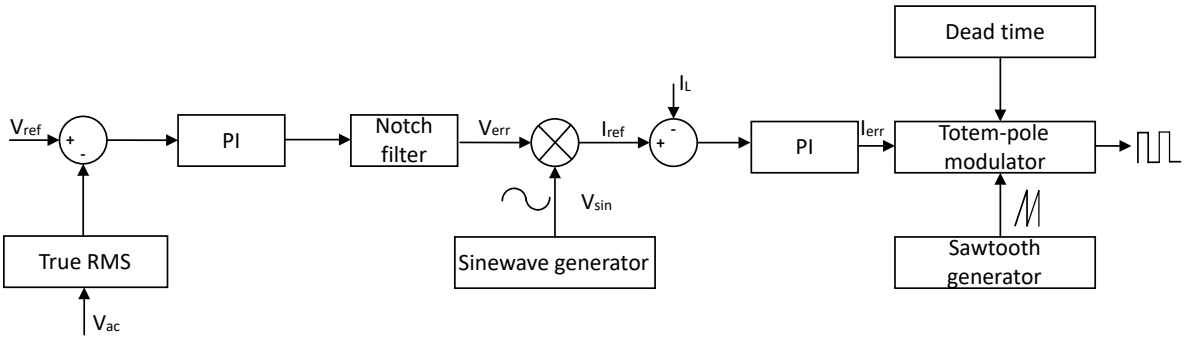


Figure 2-2. Control Method Block Diagram

Several modulation methods have been proposed over the past few decades. Bipolar, unipolar, and unipolar modulation with multiple frequencies are widely used in current applications. Different modulation methods have different influences on current total harmonic distortion (THD) and the common voltage which is the voltage change between the mid-point of the bridge and neutral point. However, in this application note, totem-pole modulation – also known as 'modified' unipolar modulation – is used to decrease switching loss to optimize the efficiency. The totem-pole bridgeless PFC structure is now widely used in PFC circuits and inverter circuits, such as the PMP23338. In unipolar or bipolar modulation, all four switches operate at high frequency and the switching loss in Si FETs is much higher than GaN FETs. Therefore, if totem-pole modulation is used to let Si FETS operate with low frequency and GaN FET operate with high frequency, totem-pole modulation can fully utilize the characteristic of GaN thus reducing the switching loss.

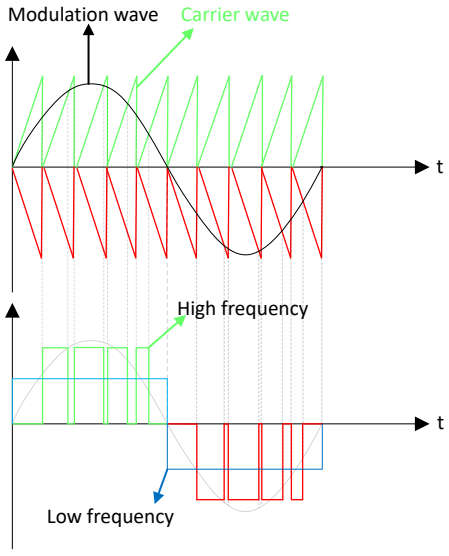


Figure 2-3. Totem-Pole Modulation

For the low-frequency bridge, the switching point is at the zero of modulation signal and for the high-frequency bridge, the switching point is the intersection point with the sawtooth wave. Table 2-2 shows the switching states using the totem-pole modulation.

Table 2-2. Switching Mode

Parameter	Q1	Q2	Q3	Q4	MODE
Positive half cycle	ON	OFF	OFF	ON	1
	OFF	ON	OFF	ON	2
Negative half cycle	OFF	ON	ON	OFF	3
	ON	OFF	ON	OFF	4

Figure 2-4 shows the basic operation situation of each mode.

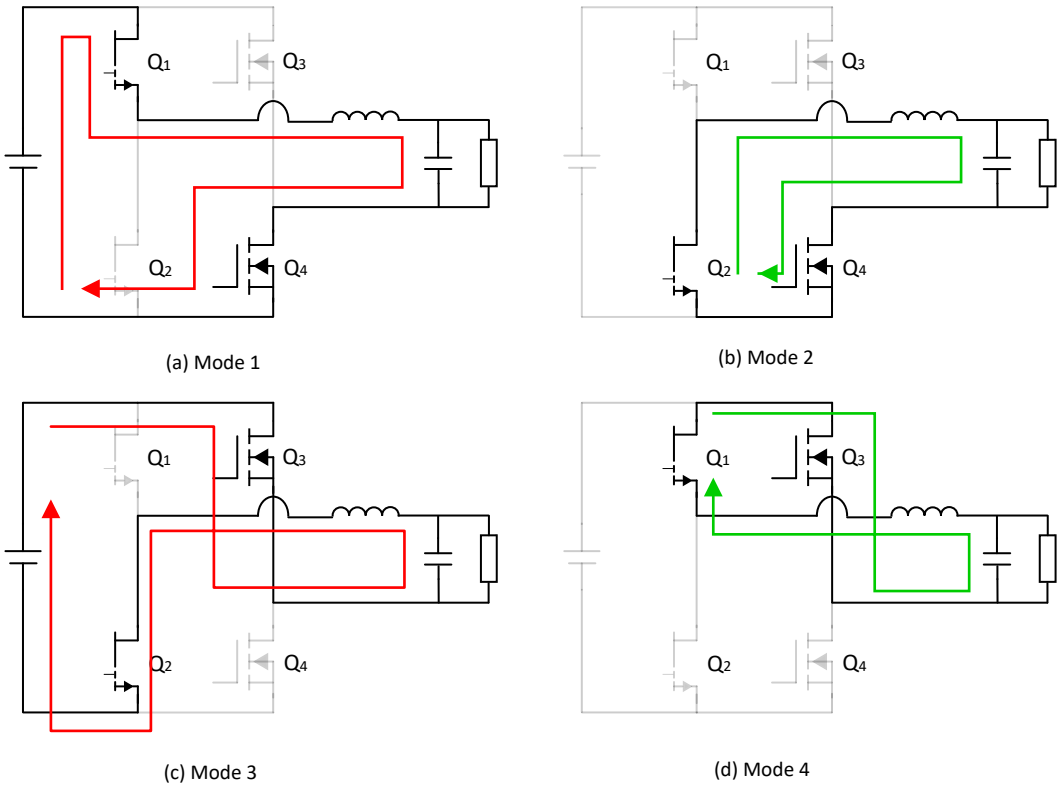


Figure 2-4. Operation Mode in Each State

3 Realization of Digital Control Using C-Script Block

This section introduces the detailed process of using the C-Script block to implement digital control. The impact of sampling is not included, which means that all sampling current and voltages are the original values in the power stage. This section describes the use of simulations to realize digital control and provides ideas for porting these codes to C2000.

3.1 Overview of Project Structure

For a better understanding of C-Script processing, Figure 3-1 provides an overall introduction of project structure and the C-Script function. Code declaration, start function, output function, and update function are used in block. For a detailed explanation of built-in functions and update logic, see the PLECS® C-Script user manual.

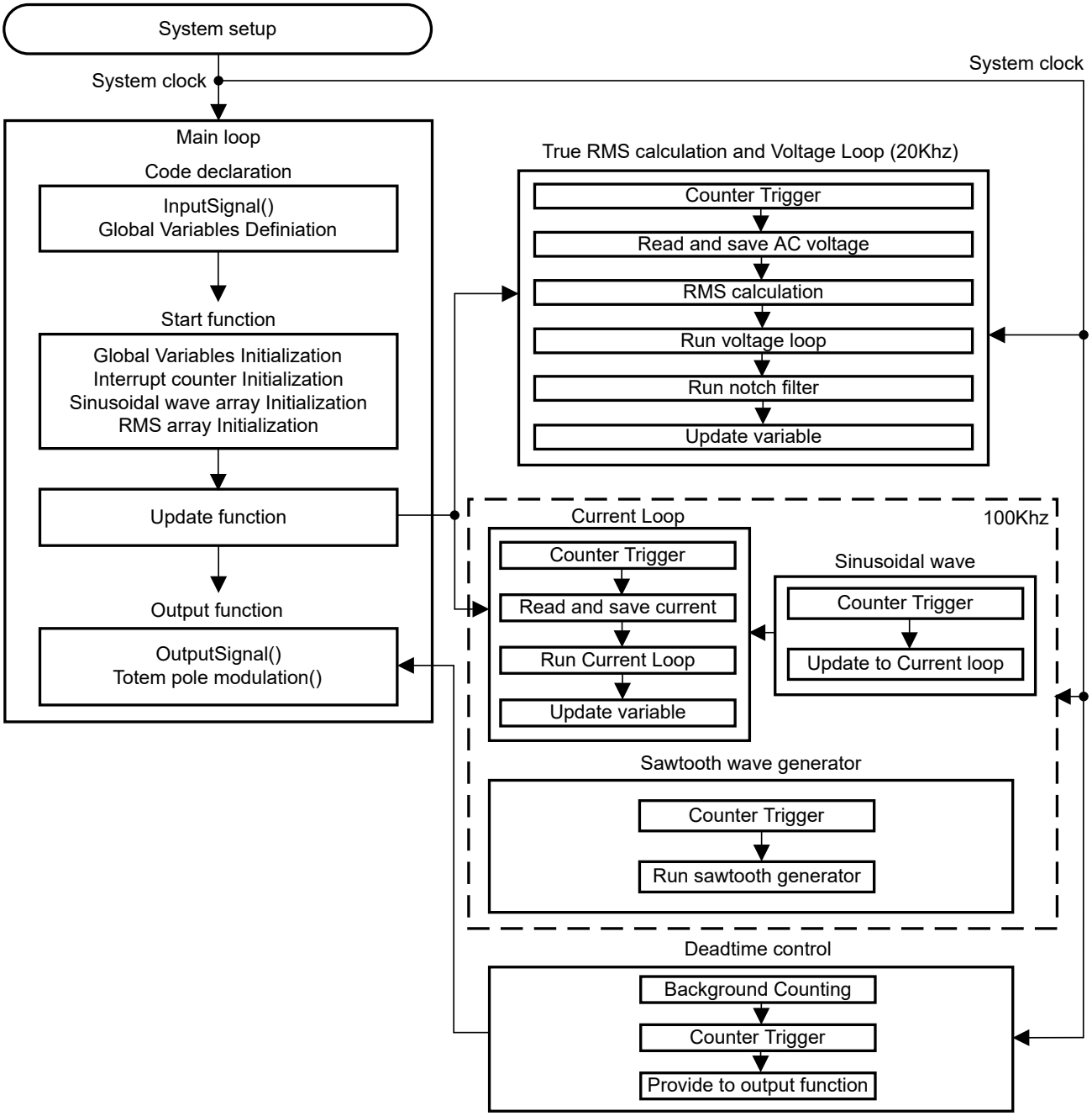


Figure 3-1. Project Structure Image

Some C-Script setups also need to be made in the C-Script block. Figure 3-2 shows an overview of the setup. The number of inputs represents the input signals from the power stage or other signals generated by other blocks, the number of outputs represents the number of output signals. An important parameter is the sample time, 83ns is set which means the frequency is 120MHz, same as 280039. This also means that the C-Script is simulated in continuous mode, similar to real MCU processing.

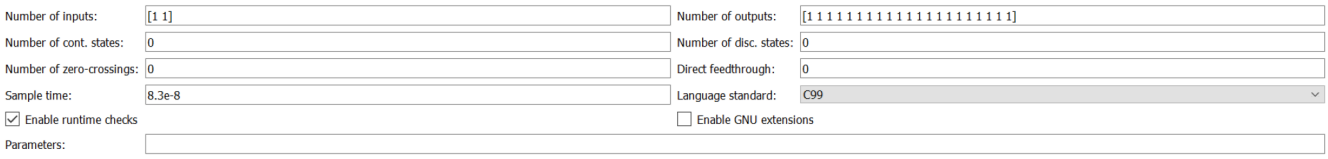


Figure 3-2. C-Script Setup

Next, a brief introduction of the code function is provided, in which global declarations and definitions are made. For example, in control loop, the inductor current and output voltage are sampled and need to be input to the C-Script block, so a built-in macro is used to define the input signal here. All parameters are initialized in the function.

```
//Used for control algorithm
#define V_AC_OUT InputSignal(0, 0)// Define the first input port is the AC Voltage Output
#define I_L InputSignal(1, 0)// Define the second input port is the inductor current
```

Figure 3-3. Input Signal Definitions

Global variables defined in code declarations are initialized in the start function.

```
// Constant value
V_OUT_REF=220;// The RMS value of Output voltage
I_L_REF=0;//Initialize the current reference
```

Figure 3-4. Variables Initialized in Start Function Code

For example, the voltage loop and current loop reference is initialized as in Figure 3-4.

The update function is the main processing part of the C-Script. Control loop calculation, output voltage sampling, sawtooth, and sinusoidal wave generator, and dead-time control are all accomplished in the update function. For example, AC voltage is sampled at 20kHz.

```
//AC voltage output sample
V_sample_Counter++;
if(V_sample_Counter==600)//20Khz
{
V_OUT_Temp=V_AC_OUT;// Save the value

V_RMS_Array[V_RMS_Tempcounter]=V_OUT_Temp;//update this value into the array

V_RMS_Tempcounter++;
```

Figure 3-5. AC Voltage Sampling Code in Update Function

The desired signal and port outputs are set in the output function. Also, the PWM signal is defined. For example, Figure 3-6 shows how the twentieth port output is defined as the RMS value of sample AC voltage. `OutputSignal` is the built-in C-Script macro.

```
OutputSignal(20,0)=V_RMS_Out;
```

Figure 3-6. Declaration in Output Function

Section 3.2 discusses implementation of different blocks.

3.2 AC Voltage Sampling and True RMS Value Calculation

Sample rate is set to 20kHz, an addition counter operating at 120Mhz is utilized to trigger the sampling process. The sample is updated when the counter reaches 600. An array is used to save the sample voltage and calculate the RMS value of the output voltage, as Figure 3-7 shows. Code declaration and initialization were accomplished previously. The length of the array number is a trade-off between accuracy and update speed. A larger array number indicates better accuracy and lower speed. Figure 3-8 shows the simulation results for this module. To determine the exact length of the array, assume the window width is four periods of 50Hz sinusoidal wave and the sample rate is 20kHz, then an array length of 1600 is obtained.

```
//AC voltage output sample
V_sample_Counter++;
if(V_sample_Counter==600)//20Khz
{
V_OUT_Temp=V_AC_OUT;// Save the value

V_RMS_Array[V_RMS_Tempcounter]=V_OUT_Temp;//update this value into the array

V_RMS_Tempcounter++;

//RMS calculation

for(i=0;i<1601;i++)
{
Temp1=V_RMS_Array[i]*V_RMS_Array[i]+Temp2;//Accumulation
Temp2=Temp1;
}
V_RMS_Out=sqrt(Temp1/1600);//Samplepoint=12,but now 11 may because the operation step
Temp1=Temp2=0;


if(V_RMS_Tempcounter==V_OUT_Samplepoint)
{
V_RMS_Tempcounter=0;
}

V_sample_Counter=0;
}
```

Figure 3-7. AC Voltage Sampling and RMS Value Calculation Code

3.3 Multiple Loop Control

As previously mentioned, the multiple-loop control method is widely used for inverter control. This section introduces how to realize the multiple-loop control method. See Figure 2-2 for the control block diagram.

3.3.1 Voltage Loop With Notch Filter

After the RMS value of the AC voltage is obtained, the value is utilized in the voltage loop first. The reference voltage is first initialized in the start function which is 220. The loop calculation rate is the same as the sample rate, 20kHz. First, the error value between the sampled RMS value and the reference voltage is obtained, using the PI compensator to adjust gain and voltage loop phase.

To further reduce the 100Hz voltage ripple in the sampled signal, a notch filter is employed after the voltage PI compensator. The central frequency of the notch filter is set to 100Hz or 628rad/s, with a bandwidth of 5Hz. Figure 3-8 illustrates the parameter initialization of the notch filter in the start function, while Figure 3-9 demonstrates the code of the voltage loop with the notch filter implemented.

```
//Norch filter parameter
a0=4+W_C*W_C*Tn*Tn;
a1=2*Tn*Tn*W_C*W_C-8;
a2=4+W_C*W_C*Tn*Tn;
b0=4+W_C*W_C*Tn*Tn+2*W_Bw*Tn;
b1=2*Tn*Tn*W_C*W_C-8;
b2=4+Tn*Tn*W_C*W_C-2*W_Bw*Tn;
```

Figure 3-8. Initialization of Notch Filter

```
//Voltage loop control
ISR_SLOW_COUNT++;
if(ISR_SLOW_COUNT==600) // Interrupt of Voltage loop
{
 ISR_SLOW_COUNT=0;
 V_OUT_ERR_PI_1=((V_OUT_REF-V_RMS_Out)*Gv_Kp);//this time error for P
 V_OUT_ERR=V_OUT_ERR_PI_2+Gv_Ki*Ts*250*((V_K_1)+(V_OUT_REF-V_RMS_Out));//this time minus last time *Ki for I,
 V_OUT_ERR_NOW=V_OUT_ERR_PI_1+V_OUT_ERR;//output of voltage loop
 V_OUT_ERR_PI_2=V_OUT_ERR;//Save this time I error
 V_K_1=V_OUT_REF-V_RMS_Out;//Save this time P error
 //Notch filter
 r_k=V_OUT_ERR_NOW;
 y_k=(a0/b0)*r_k+(a1/b0)*r_k_1+(a2/b0)*r_k_2-(b1/b0)*y_k_1-(b2/b0)*y_k_2;
 y_k_2=y_k_1;
 y_k_1=y_k;
 r_k_2=r_k_1;
 r_k_1=r_k;

}
```

Figure 3-9. Voltage Loop With Notch Filter

3.3.2 Current Loop With PI Compensator Anti-Windup

Current loop is connected after the voltage loop. The updated rate is set to 100kHz, so the sample rate of the inductor current is also 100kHz. The output signal of the notch filter multiplied by the sinusoidal wave created by the sinusoidal module creates the reference current signal. Similar to the voltage loop, current error signal is processed by the PI compensator. To avoid oversaturation of integral, an anti-windup is used in the PI compensator. Figure 3-10 shows the code for the current loop.

```
//Current loop control
ISR_FAST_COUNT++;//Current loop counter
if(ISR_FAST_COUNT==120)//120=100Khz
{
ISR_FAST_COUNT=0;// reset counter
//I_L_REF=0.7*SinwaveOutput;// Open loop setting
I_L_REF=y_k*SinwaveOutput;
I_L_ERR_PI_1=(I_L_REF-I_L)*Gi_Kp;//Error of Proportion this time
I_L_ERR=I_L_ERR_PI_2+(Gi_Ki*Ts*50*(I_K_1+(I_L_REF-I_L)));// Error of intergation this time
// Anti-windup
if(I_L_ERR>0.95&&(I_L_REF-I_L)>0)
{
I_L_ERR=0.95;
}
if(I_L_ERR<-0.95&&(I_L_REF-I_L)<0)
{
I_L_ERR=-0.95;
}
//I_L_ERR_NOW=I_L_REF;//open loop setting
I_L_ERR_NOW=I_L_ERR+I_L_ERR_PI_1;
I_L_ERR_PI_2=I_L_ERR; //I_L_ERR_PI_2 Error of integration k-1
I_K_1=(I_L_REF-I_L);//e(k-1)
}
```

Figure 3-10. Current Loop Code

3.3.3 Sinusoidal and Sawtooth Wave Generator

In PLECS, the update and output function are processed for each sample step. Therefore, the designer can consider the sample time as the system clock, using a counter to generate these signals with proper frequency. For the sinusoidal generator, first create an array and initialize this array with the number that changes sinusoidally. This array is set to 2000 to correspond with the current loop operating frequency.

```
//Update sinwave number
step_counter++;
if(step_counter==120)
{
if(Steptemp<2000)
{
SinwaveOutput=Sinwave[Steptemp];
Steptemp++;
}
else
{
Steptemp=0;
}

step_counter=0;
}
```

Figure 3-11. Sinusoidal Wave Generator

Logic is the same as in the sawtooth wave generator. One important aspect of totem-pole modulation is that two sawtooth signals need to be created; one for the positive cycle and the other for the negative cycle to generate a proper PWM signal, shown in Figure 2-3. The amplitude of these two carrier signals is from 0 to 1 and 0 to –1, respectively. It is worth mentioning that 2 signals are needed because of the case where one sawtooth signal with an amplitude from –1 to 1 to generate PWM signal is used. When the error signal is near zero, the lowest duty cycle is 50%, and the correct duty cycle is supposed to be near zero. Besides, the upper limit is set to 12 and divided by 120 to set the right frequency under the system clock. Figure 3-12 shows the sawtooth generator code.

```
//update sawtooth number
CurrentNum=CurrentNum+1;
CurrentNum2=CurrentNum2-1;//for upperlimit=1,1/120 one step
if(CurrentNum==120)
{
CurrentNum=0;
}
if(CurrentNum2==-120)
{
CurrentNum2=0;
}
```

Figure 3-12. Sawtooth Wave Generator

3.3.4 Totem-Pole Modulation and Dead-Time Control

To realize totem-pole modulation, compare the output error signal of current loop with the generated sawtooth wave. See Table 2-2 for the switching state. The positive and negative values of the error signal determine the switching state of the low-frequency bridge, and the intersection point between the sawtooth and error signals determines the switching state of the high-frequency bridge. To avoid the PWM signal glitch near the zero point, set the threshold value.

Additionally, to avoid short circuits in bridge, the dead time is set. Since a continuous sample rate is set, all these functions are executed with a certain sequence; therefore, a delay function cannot be added in the transient time. Instead, a background timer is set in the update function and accumulates every sample time. Therefore, once the counter reaches the desired value, the timer turns off or turns on another FET in the bridge. Accordingly, the turn-on or turn-off time can be extended, but cannot be decreased to achieve dead-time control. Figure 3-13 and Figure 3-14 show the codes.

```
//deadtime
DB_Counter++;
if(DB_Counter==11)
DB_Counter=0;
```

Figure 3-13. Background Dead-Time Counter in Update Function

Although the counter number is set to 11, the effective number is 10 which means the dead time is 83.3ns.

```
//Positive cycle
if(I_L_ERR_NOW>0.003)// Sin>0,low frequency lower mosfet on
{
LF_HS=0;
if(DB_Counter==10)//Dead time
{
LF_LS=1;
}

if(I_L_ERR_NOW>Sawtooth)
{
HF_LS=0;
if(DB_Counter==10)
{
HF_HS=1;
}
}
else
{
HF_HS=0;
if(DB_Counter==10)
{
HF_LS=1;
}
}
}
```

Figure 3-14. Totem-Pole Control Code in Positive Cycle

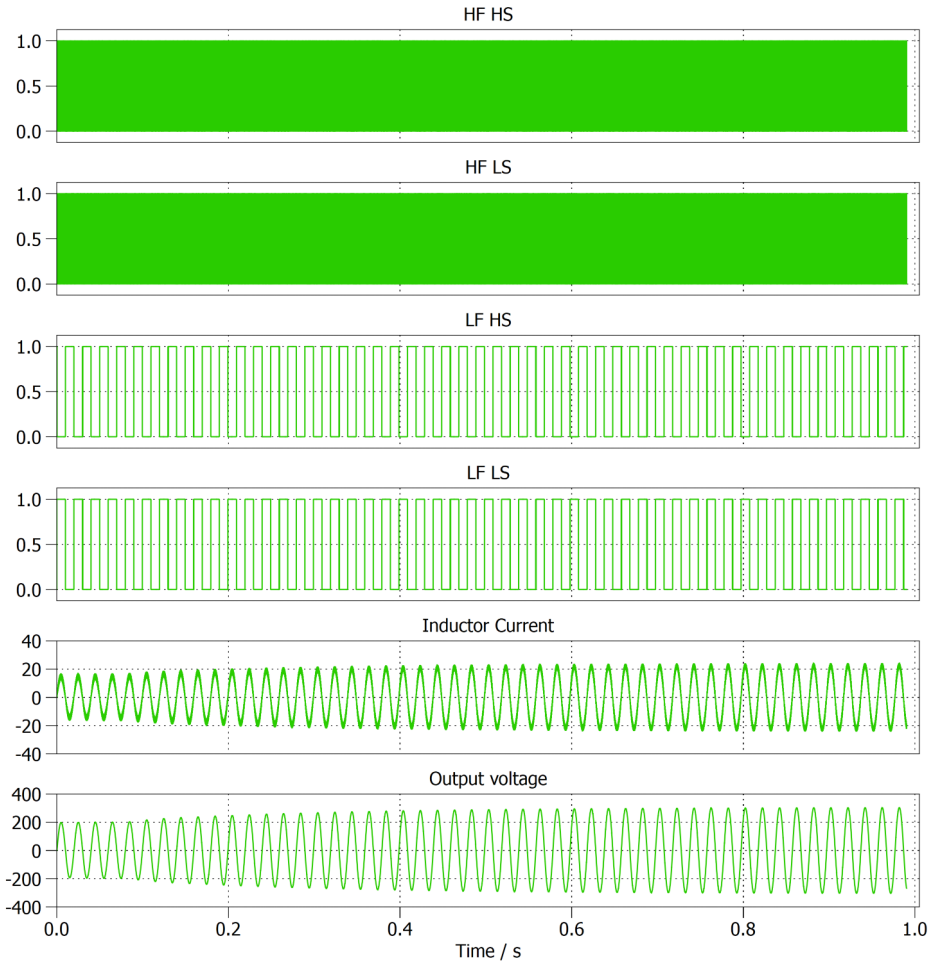
4 Simulation Results

This section presents simulation results to verify the theory and code implementation. See Table 2-1 for the power stage parameters.

During the start-up process, to avoid a voltage surge, the initial value of the RMS array is set to 70. From start-up to 95% of the final value is defined as the settle time. Figure 4-1 shows low-frequency bridge switches at 50Hz, and high-frequency bridge switches at 100kHz. In the zero crossing, some glitches happened because of the error signal, this can be undermined by increasing the threshold.

From start-up to 95% of the final value, 660ms is required. To better evaluate the performance of control parameters, a series of transient tests are set. As illustrated during the full-load and half-load situation, inductor current is a relatively standard sine wave. However, under light load, the inductor current exhibits obvious distortion. One effective way to decrease the distortion is to increase the inductance.

From Table 4-1, THD% and setting time indicate that the control system can regulate the inductor current and output voltage quickly and effectively.



HF HS: High frequency and High Side PWM signals
HF LS: High frequency and Low Side PWM signals
LF HS: Low frequency and High Side PWM signals
LF LS: Low frequency and Low Side PWM signals

Figure 4-1. Start-Up Process

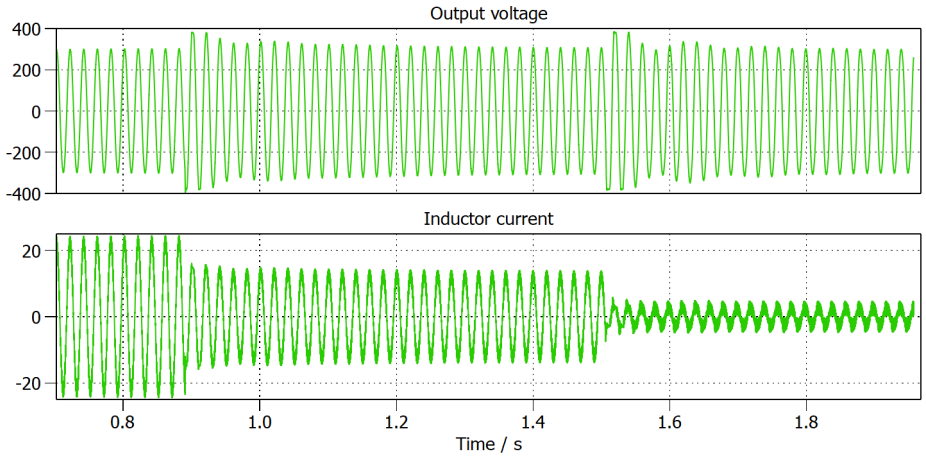


Figure 4-2. Load Transient (100% to 50% to 10%)

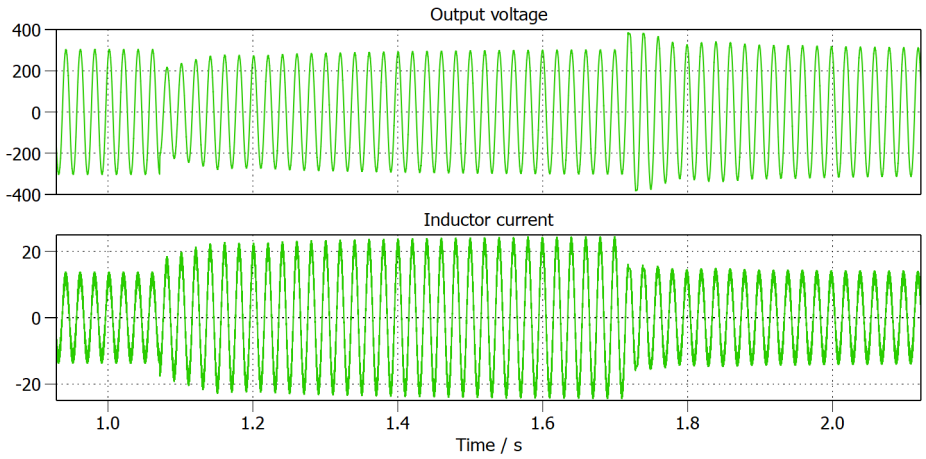


Figure 4-3. Load Transient (50% to 100% to 50%)

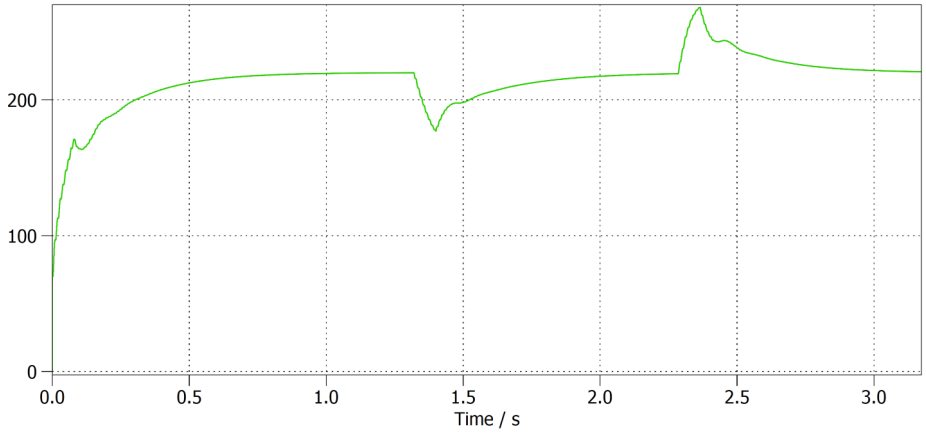


Figure 4-4. True RMS Calculation in Load Transient

Table 4-1. THD% and Settle Time

Condition	THD%	Settle Time
Full load start-up	2.7%	0.660s
Full load to half-load	2.8%	0.281s
Half-load to 10%-load	2.6%	0.259s

Figure 4-4 shows the output of true RMS calculation block. Note the trade-off in this block. A larger array number slows the calculation and adds additional delay to the system. This causes the output voltage to exceed the reference value because the RMS calculation lags behind the actual value. However, a short array length brings a 50Hz frequency ripple into the RMS value, which causes oscillation in the control. After many tests, a window width of 4 was found to be a good value in this model.

5 Summary

This application note introduces the implementation of single phase off-grid inverter with digital control in PLECS. All function blocks are realized using a C-Script block with code. This model matches the TI reference design PMP23338 or other TI totem-pole PFCs or inverter reference designs. By utilizing this model, the parameters of the power stage or control block can be easily verified. TI GaN LMG352XR030 is also implemented in this model, and the performance of this device can be verified by using this model.

In addition, the description of the multi-loop control loop with the true RMS calculation can be used as a design reference for a single-phase off-grid inverter. The complex control method and topology can be extended by using this model.

Furthermore, through a detailed description of the project software structure and code, this application note provides ideas for porting code to TI's C2000 MCU.

6 References

1. Texas Instruments, *PMP23338 3.6kW, single-phase totem-pole bridgeless PFC reference design with e-meter functionality*
2. Texas Instruments, *TIDM-02008 Bidirectional high density GaN CCM totem pole PFC using C2000™ MCU reference design*
3. Texas Instruments, *TIDM-HV-1PH-DCAC Single-Phase Inverter Reference Design With Voltage Source and Grid Connected Modes*
4. Texas Instruments, *LMG3522R30 650-V 30-mΩ GaN FET With Integrated Driver, Protection, and Temperature Reporting Data Sheet*
5. PLECS User Manual
6. IEEE, Design and analysis of single phase voltage source inverter using Unipolar and Bipolar pulse width modulation techniques, *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, Putrajaya, Malaysia, 2016, pp. 277-282.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265