Oversampling of PWM converters in SimScape Electrical

Kyle Diller

v0.1

Introduction

In real-time simulators, the oversampling technique for high frequency power converter/inverters is very useful to obtain an accurate simulation of such converters with minimum implementation efforts. Let's define some terms here. So these converters are driven at a PWM frequency of Fpwm (Hz) while being simulated at a fixed time step of Ts seconds. The switch are normally modeled as binary switches, that is either in low or high impedance state; IGBT/GTO/MOSFET can be simulated this way, with the possibility of adding some forward voltage drop. It then incurs a sampling error of the IGBTs switching actions that is proportional to Ts*Fpwm, the greater, the less accurate, *if this done in a dumb manner.*

In [1], the authors first demonstrated how to use effectively oversampling in a train traction real-time simulator. The idea is clever. Instead of trying to interpolate all switching actions as binary switches, we consider the 2-level inverter as a kind of voltage router with the following equation in continuous time:

$$V_{out} = (g_{up} == 1) * V_{dc} + (g_{up} == 0)\ \&(g_{lo} == 0)\ \&(I_{load} < 0) * V_{dc} \qquad (1)$$

V_{out} is the inverter voltage output, V_{dc} is the inverter input DC voltage, g_{up} and g_{lo} are the upper and lower IGBT gate logic signals respectively and I_{load} the load current, that is the current going out of the inverter.

Interpretation. The first term of the equation is obvious: the load voltage is V_{dc} when the upper IGBT is ON. The second term is for the dead time: when both IGBT are OFF and the current is negative, that is entering the inverter, the upper protection diode turns ON instantenously, which set $V_{out} = V_{dc}$ during that time. A similar equation can be build for the inverter input current drawn from the DC bus: for all the time that $V_{out} = V_{dc}$ the input current equals I_{load}.

Oversampling now. V_{out} must be computed for each time step to compute the complete system equations. Voltage inverters are most of the time connected to some kind of inductive device, such as permanent magnet motor or simple RL loads. Computing these load currents involves integrating the applied voltage at each time step in the following manner:

$$I_{device} = \int_{t}^{t+h} V_{out}\, dt \qquad (2)$$

The key is to obtain accurate timings of the gate switching actions to compute Equation 2 accurately. In [1], the train traction simulator runs at 30 microseconds, but the GTO gates are sampled much faster by FPGA board implementing 40 MHz counters. This enables the very accurate simulation of the applied motor voltages at each time step and thus the train motor current is very accurately computed.

So, for each time step, the averaged inverter output voltages are computed from the 40 MHz board, which capture the switching times with 25ns resolution. (Today, counters with 100 MHz or 200 MHz are more common). The following figure shows the transformation of the high-frequency sampling to time-step

averaged inverter voltage. The key to understand this technique is that the integral of the 'Real inverter voltage' (that is at a very high resolution) is actually equal to the integral of the 'Time-step Averaged voltage'.

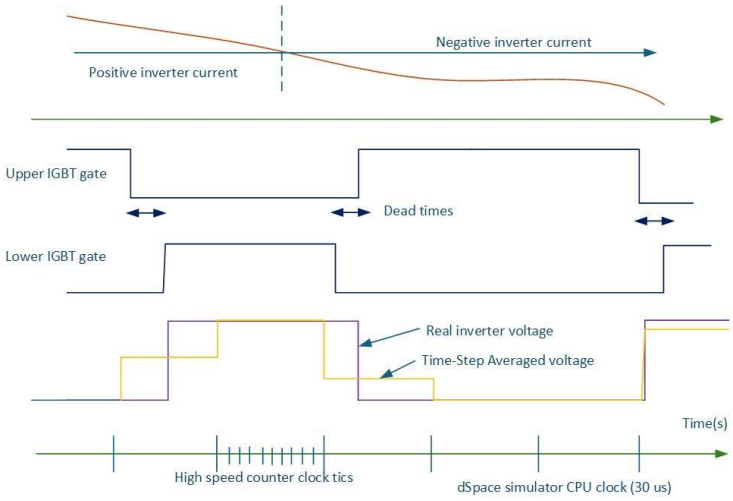

Figure 1: Computation of Time-Step Averaged voltage

A converter model with oversampling (without FPGA counters)

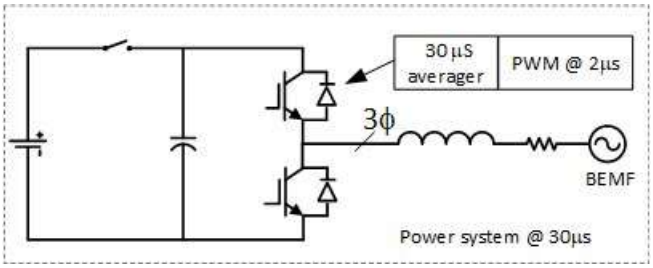

Figure 2: Two-level inverter with RL load with back-EMF voltage

Figure 2 describes the Simulink model named **'MATE_2level_DriveOversampling.slx'**. It is comprised of a capacitor that is connected to a battery via a switch, an Oversampling 2-level inverter, a 3-phase RL load with a back-EMF.

In this multi-rate model, the PWM pulses are computed at 2 microseconds while the inverter drive is computed at 30 microseconds, in a multi-rate simulation. The PWM resolution is therefore 15 times greater than the circuit sampling resolution. To keep the 2 microseconds resolution into the 30 microseconds simulation, 'time-step duty cycle averager' is used in this manner:

$$duty\_avg_{30us} = \frac{1}{15}\sum_{1}^{15} duty_{2u} \qquad (3)$$

This is like using an FPGA board running at 500 kHz to capture the IGBT pulses. Think about it.

Interpretation: for each large time step of 30 micro-seconds, there are 15 steps of 2 micro-seconds, at which resolution, the logical value can either be 0 or 1. At 30 microseconds, we obtain the averaged value of the 15 sample of gate at 2us, which integration accuracy is kept intact in Equation 2.

Special mode of operation of power inverters.

The previous section discussed the standard or active operation mode of inverters, that is when the drive is controlled by the IGBT pulses. But other modes exist, for which the previous techniques need some adaptation. The inverter will definitively act sometimes as a natural rectifier (to charge the DC bus during power-up sequence) or simply shutdown (resulting in no load current). For these cases, Simscape switches are added to the model.

One could ask why not use real SimScape Electrical switches from the start? The reason is the Oversampling method cannot be implemented simply and reliably using time-domain interpolation-extrapolation techniques like [2]. The problem of the dead-time is crucial also because typically much smaller than the large simulation time-step.

Coming back to our model **'MATE_2level_DriveOversampling.slx'**, the simulation was run to validate various operating modes:

Time = 0 seconds: Simulation starts with the charging of the DC link capacitor, inverter is shutdown.

Time = 0.08 seconds: the back-EMF voltage is raised causing some natural rectification.

Time = 0.1 seconds: the inverter is activated. Currents are smooth because of the oversampling.

Time = 0.4 seconds: the inverter is shutdown. The back-EMF is clipped at Vdc, the battery is charging.

Time = 0.6 seconds: the capacitors are disconnected from the battery, and their voltage starts to rise.

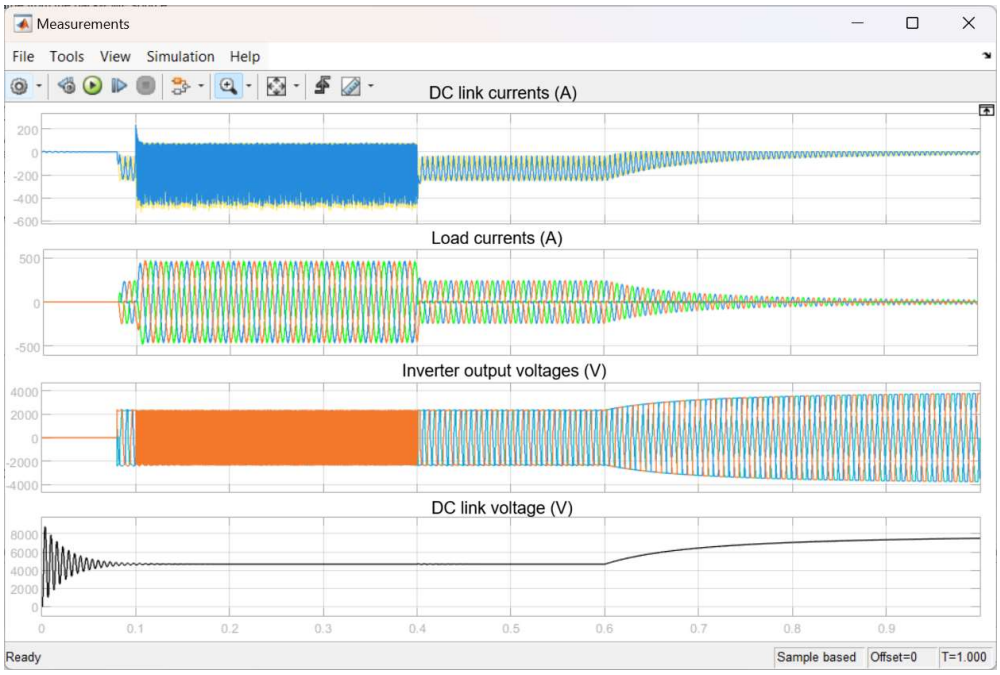

Figure 3: Simulation results for the Oversampled inverter model in various operational modes

References

[1] P. Terwiesch, T. Keller, E. Scheiben, “Rail Vehicle Control System Integration Testing Using Digital Hardware-in-the-Loop Simulation”, IEEE Trans. On Control Systems Technology, Vol. 7, No. 3, May 1999.

[2] M. O. Faruque, V. Dinavahi, W. Xu, "Algorithms for the Accounting of Multiple Switching Events in Digital Simulation of Power Electronic Systems", IEEE Trans. on Power Delivery, Vol. 20, No. 2, April 2005, pp.1157-1167.