The Impact of Timer Resolution in the Efficiency Optimization of Synchronous Buck Converters

Pedro Amaral\textsuperscript{1}, Cândido Duarte\textsuperscript{2}, Pedro Costa\textsuperscript{3}

Abstract—This dissertation aims at analyzing the influence of one of the most important characteristics of a digital controller, the timer resolution, in the issue of dead time optimization for synchronous buck converters. A sensorless duty cycle minimizing algorithm has been designed and implemented using state-of-the-art hardware from Infineon Technologies to prove the concept. The results demonstrate a significant power efficiency improvement, not only over a fixed dead time approach, but also with the increment of timer resolution.

Index Terms—Synchronous buck converter, dead time, sensorless, DC/DC converter, microcontroller, timer resolution

I. INTRODUCTION

POWER efficiency is now, more than ever, a major topic for systems, households, providers and governments. Due to an imminent energy crisis, the industry is working hard on reducing power losses. Additionally the market of non-digital power supplies is steadily losing ground to the digitalized counterparts. This creates new possibilities to tackle old issues with unprecedented approaches.

One of these issues has been the existence of a dead time between turning one of the MOSFETs off and the other on, in synchronous buck converters. This short time, that occurs twice every switching cycle, is necessary to prevent short circuit between the input voltage to the ground. However this has the side effect of forward biasing the body diode integrated in the synchronous switch and causing high conduction losses, which can be described as

\[
\text{\(P_{\text{loss}} = V_D I_{\text{out}} \frac{t_{d,r} + t_{d,f}}{T_s}\)}
\]

where \(V_D\) is the diode voltage drop, \(I_{\text{out}}\) is the output voltage, \(T_s\) is the switching period and \(t_{d,r}\), \(t_{d,f}\) are the rising and falling edge dead times.

The objective of this dissertation is to design and implement a dead time optimization algorithm, that is simple, computationally light and needs no additional hardware, seamlessly taking advantage of the existing controller. Furthermore, the impact of the controller resources, namely the timer resolution, on the efficiency improvement is to be studied.

Many authors have already worked on the issue of dead time. The simplest approach is to set a fixed dead time for the entire operation time of the converter. Most gate driving ICs and microcontrollers already have complementary PWM modes with fixed dead time.

![Synchronous buck converter and PWM signals with dead time.](image)

Another class of methods is based on measuring the current across the transistor half-bridge [1], [2]. However the most popular approach depends on voltage measurements rather than current. These divide into adaptive methods, which emulate a diode and optimize only in the current switching cycle [3], [4] and predictive methods, which use the information of a switching cycle to optimize the next ones [5], [6].

Recently a new kind of optimization methods arose: these do not depend on any additional hardware and use the existing controller. These approaches use variables that are directly related to the efficiency like input current, duty cycle and temperature [7], [8]. This is the group of methods that best fits the objectives of this dissertation. By using a duty cycle minimizing algorithm [8], the influence of timer resolution can be further explored, which fits the second objective of this work.

II. ANALYSIS OF DEAD TIME OPTIMIZATION

By analyzing the circuit, it is possible to model the output voltage as function of the dead time \(t_d\) and duty cycle \(t_{on}\).

\[
\text{\(V_{\text{out}} = V_{\text{in}} \frac{t_{on}}{T_s} - V_D \frac{t_{d,r} + t_{d,f}}{T_s}\)}
\]

A generic duty cycle minimizing optimization algorithm can be divided in three sequential steps:

1) Dead time is varied, which perturbs the output voltage:

\[
\Delta t_{on} = 0 : \quad \Delta V_{\text{out}} = V_D \frac{\Delta t_d}{T_s}
\]
2) The controller detects the output voltage variation and compensates it by changing the duty cycle:
\[
\Delta t_d = 0 : \Delta t_{on} = \frac{T_s \Delta V_{out}}{V_{in}} \quad (4)
\]
3) The output voltage returns to its regular value, while the converter now operates with a different duty cycle and dead time:
\[
\Delta V_{out} = 0 : \Delta t_d = \frac{V_{in}}{V_{D}} \Delta t_{on} \quad (5)
\]
These steps define an algorithm operation point which is part of the set of all possible operation points, defined by the aforementioned three equations:
\[
r = (\Delta t_{on}, \Delta t_d, \Delta V_{out}) = \Delta t_{on} \cdot \left(1, \frac{V_{in}}{V_{D}}, \frac{V_{in}}{T_s}\right) \in \mathbb{R}^3 \quad (6)
\]
Since the algorithm runs in a controller with finite resources, namely the ADC and timer, not all operation points are possible for a given number of bits \(N_{timer}\) and \(N_{ADC}\). The maximum algorithm resolution is guaranteed by the minimum operation point. This point is given by
\[
\Delta t_{d,\text{min}} = T_s \frac{V_{in}}{V_D} \max \left\{2^{-N_{timer}}, \frac{V_{ref} V_{in}}{V_{in}}, 2^{-N_{ADC}}\right\} \quad (7)
\]
The total initial losses \(P_{loss,i}\), caused by the existence of a dead time \(t_{d,i}\), can be eliminated in \(\gamma\) steps of \(\Delta P_{loss,min}\), which is defined by the minimum dead time variation \(\Delta t_{d,\text{min}}\) calculated before. A factor \(\Psi = P_{loss,\text{elim}} / P_{loss,i}\) can express the amount of losses that are eliminated in the body diode in terms of the controller resources.
\[
\Psi = 1 - \frac{T_s \frac{V_{in}}{V_D} \max \left\{2^{-N_{timer}}, \frac{V_{ref} V_{in}}{V_{in}}, 2^{-N_{ADC}}\right\}}{2t_{d,i}} \quad (8)
\]
### III. IMPLEMENTATION
In order to prove the theory, a simple, effective, computationally light, a dead time optimization algorithm, without additional hardware, has been developed in a XMC4200 microcontroller unit and tested on a prototype board. This powertrain is a 12-to-1.8 V, 5 A, 9 W converter.

The optimization algorithm depends exclusively on the duty cycle value and aims at minimizing it, in order to maximize the efficiency. By computing the average duty cycle in every output voltage control cycle, the algorithm can detect a load variation and trigger the optimization, which is run for rising and falling edge dead time.

The algorithm starts from a fixed dead time of 200 ns and decreases it with a certain step \(\Delta t_d\). In each iteration, the duty cycle and dead time gradients are calculated and used to determine in which region the algorithm is.
\[
\begin{align*}
\text{sign}(\Delta DC) = \text{sign}(\Delta t_d) & \implies t_d < t_{d,\text{opt}} \\
\text{sign}(\Delta DC) \neq \text{sign}(\Delta t_d) & \implies t_d > t_{d,\text{opt}}
\end{align*} \quad (9)
\]
This allows that, after the border between regions is crossed, a finer search with a smaller dead time step in the opposite direction is started. This is repeated until the stopping condition \(\Delta DC < \varepsilon\) is fulfilled. \(\varepsilon\) can be changed to assure more precision or more convergence speed. Furthermore, there is a user defined minimum dead time value to ensure the integrity of the hardware.

The algorithm was tested using both low and high resolution PWM, in order to compare the efficiency improvement with different timer resolutions. The results can be seen in Table I.

**Table I**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Low Resolution</th>
<th>High Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N_{ADC})</td>
<td>12 bits</td>
<td>12 bits</td>
</tr>
<tr>
<td>(N_{timer})</td>
<td>8 bits</td>
<td>14.3 bits</td>
</tr>
<tr>
<td>(t_{d,r,\text{opt}})</td>
<td>25 ns</td>
<td>27.5 ns</td>
</tr>
<tr>
<td>(t_{d,\text{f,opt}})</td>
<td>25 ns</td>
<td>31.25 ns</td>
</tr>
<tr>
<td>(P_{loss,\text{elim}})</td>
<td>72%</td>
<td>98.6%</td>
</tr>
<tr>
<td>(\Delta \mu_{\text{system}})</td>
<td>2.5%</td>
<td>3.6%</td>
</tr>
<tr>
<td>(\Delta \mu_{\text{converter}})</td>
<td>3.6%</td>
<td>4.9%</td>
</tr>
<tr>
<td>(\mu_{\text{converter}})</td>
<td>95.1%</td>
<td>96.1%</td>
</tr>
</tbody>
</table>

### IV. CONCLUSION
In the reviewed literature, dead time optimization methods are implemented in DSPs, FPGAs or ICs. In this work a microcontroller unit was used, which was an innovation. Furthermore, such an extensive analysis of the impact of timer resolution in the dead time optimization has never been done. This work also makes some contributions in this topic.

The designed algorithm not only fulfilled the requirements but also produced good results: \(\sim 5\%\) efficiency improvement over a fixed dead time and \(1\%\) using high resolution PWM over low resolution. These results are significant and may influence the decision of choosing a controller with this kind of resources for the design of a power supply.

**REFERENCES**