

Internal Resistance Calculation for Small Vertical Axis Wind Turbine

Author: Ian Eykamp, **Date:** 6/16/2023

Purpose

To document certain calculations for the design of the small wind turbine buck converter, including

- Maximum output current
- Input capacitor size
- Slew rate of current waveforms for spec'ing diff amps
- Indicator LED resistances

Import libraries

`plecs_helper.py` is a file I created which defines some helper functions for nice plots and getting data off the oscilloscope and out of PLECS.

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
from UliEngineering.EngineerIO import format_value
from si_prefix import si_format
import plecs_helper as helper
%matplotlib
%matplotlib inline

# Imports and setup
from pint import UnitRegistry, Quantity
from scipy.signal import find_peaks
from scipy.optimize import fsolve

# pandas display using scientific notation
# pd.set_option('display.float_format', lambda x: f'{x:.3e}')

# use pint
units = UnitRegistry()
units.default_format = "~P.2f"

def to_decibels(arr, dc_gain = 0):
    return np.log10(np.abs(arr)) * 20 + dc_gain
```

Using matplotlib backend: TkAgg

Calculating maximum current

The buck converter will draw a certain average current from the rectifier; in doing so, it produces a known output voltage (the battery voltage, which may be between 12V and 14.6V) and a known output current. The buck converter acts like an ideal transformer in that it preserves the product of voltage and amperage from the input to the output,

$P_{in} = V_{in} \cdot I_{in} = V_{out} \cdot I_{out} = P_{out}$. The input voltage is the voltage between the terminals of the rectifier. Using this, we can calculate $I_{in} = V_{out} \cdot I_{out} / V_{in}$. Once we have I_{in} , we can calculate the voltage drop across the internal resistance of the generator by $V = IR$. I will call this $\Delta V_{internal}$. The voltage now available for the buck converter to use is $V_{in} = V_{gen} - \Delta V_{internal}$. So, depending on how much output current we want to draw into the battery, it will affect the voltage available at the buck converter input.

We need to charge a 12V battery (really, when it's done charging, it will be at 14.6V) using our buck converter. Because of the chip we are using, our buck converter can have no more than a 50% duty cycle, which means our input voltage must be at least $2 \cdot 14.6V = 29.2V$ at a bare minimum. Round up to $V_{in} = 30V$ at a minimum.

To spin the generator and get power out of it, I will place it in front of the 4th floor wind tunnel in the Rocketry lab. Without getting to a point where I felt it was dangerous, I can get it to 30V with a 100 Ω load resistance. I am confident I can spin it faster if we need a higher voltage; in fact, I made my design with 50V or higher in mind.

We know our voltage divider has to give us at least 30V from the 50V available. That means there can be no more than a 20V drop across the 14.5 Ω internal resistance, which corresponds with a 1.3A current through the rectifier.

We have $I_{in} = 1.3A$, $V_{in} = 30V$, $V_{out} = 14.6V$, and $V_{in} \cdot I_{in} = V_{out} \cdot I_{out}$. Thus, the maximum current we can safely draw for a 50V generator voltage is 2.8A. Likely, the available current to the battery would be much smaller because the power converter is not 100% efficient (in fact 70% or 80% efficiency is a fairly good design) and because there are many auxiliary IC's and the microcontroller that will eat up a lot of the excess current.

Well, I guess I'll just keep our current down between 0 and 0.5A to be safe. (Just kidding, I already designed the thing with a maximum of 0.5A in mind, and I don't want to redo all the analysis after realizing that 1A or more would be fine).

```
In [ ]: Vout = 14.6 * units.volt # battery voltage
        Rinternal = 14.5 * units.ohm
        Vgen = 50 * units.volt
        Vin = 30 * units.volt
        Vinternal = Vgen - Vin
        Iin = Vinternal / Rinternal
        Iout = Iin * Vin / Vout
        print(f"Assuming Vgen = 50V, we have Iout = {Iout.to_compact(units.amp)}")
```

Assuming $V_{gen} = 50V$, we have $I_{out} = 2.83 A$

At 0.5A of output current, it would take only 35V at the generator, or actually probably closer to 40V because of the inefficiencies.

```
In [ ]: Vout = 14.6 * units.volt # battery voltage
Rinternal = 14.5 * units.ohm
Vin = 30 * units.volt
Iout = 0.5 * units.amp + 0.5 * units.amp
Vgen = Vin + Iout * Vout / Vin * Rinternal
print(f"Assuming Iout = 0.5A, with another 0.5A for auxiliary components, we need V
```

Assuming $I_{out} = 0.5A$, with another 0.5A for auxiliary components, we need $V_{gen} = 37.06 V$

Calculating Input Capacitance

This averaged transformer analysis has all been made possible by assuming the input voltage does not change in response to the switching

We have been calculating using the average input and output currents. For the buck converter to work properly, it needs to satisfy the input voltage even at the moment of peak current draw. In boundary conduction mode (BCM) where we are nominally operating, the peak current is two times the average current. For higher currents, we will go into continuous conduction mode (CCM), where the peak current is slightly less than twice the average current; for lower currents, we will go into discontinuous conduction mode (DCM) and require more than twice the average current at the peak.

Fortunately, if we have a big enough input capacitor, then we can smooth out the current waveform and almost exclusively care about the average current. By $I = C \frac{dV}{dt}$, there is always some voltage drop at the V_{in} node when the extra current is being sourced by the capacitor, but by choosing a large enough capacitance value, we can keep the voltage change to within $\pm 1V$ of the average $\Delta V_{internal}$.

The source current follows a triangular waveform where it rises linearly from its starting current I_{min} to I_{peak} and then falls to zero. If we integrate this current over time, we get the total charge that is transferred during the upwards ramp as

$\int_{t=0}^{D \cdot T_s} I_g dt = \frac{1}{2} T_s \cdot D \cdot (I_{peak} - I_{min}) + T_s \cdot D \cdot I_{min}$. If we are okay with rounding up by a factor of two, we can pretend the current is always I_{peak} , so that the charge becomes $Q = T_s \cdot D \cdot I_{peak}$. During this time, a current of $I_{avg} = \frac{I_{peak}}{2}$ is always being sourced by the generator, which over the switching cycle is $Q = T_s \cdot D \cdot \frac{I_{peak}}{2}$. So our capacitor only needs to source the other $T_s \cdot D \cdot \frac{I_{peak}}{2}$ of the charge, all while depleting only 1V of its voltage.

In reality, the charge required is about half of that value for CCM and BCM and has a limit approaching that value as we go deep into DCM.

We have $C = \frac{Q}{V}$, so we have $C_{input} = \frac{T_s \cdot D \cdot \frac{I_{peak}}{2}}{1V}$

```
In [ ]: Fs = 50 * units.kilohertz
Ts = 1 / Fs
D = 0.36
Ipeak = 1 * units.amp
Vripple = 1 * units.volt

Cinput = Ts * D * Ipeak / 2 / Vripple
print(f"Cinput = {Cinput.to_compact(units.farad)}")
```

Cinput = 3.60 µF

So yeah, 10uF should be about enough.

Calculating Diff Amp Slew Rate Across Shunt Resistors

(To ensure our diff amps are fast enough to measure the current waveforms)

The slope of the current waveform over time is linear and determined by the inductance value we chose for the inductor, $V_L = L \cdot \frac{dI}{dt}$, where $V_L = V_g - V_{bat}$, so $\frac{dI}{dt} = \frac{(V_g - V_{bat})}{L}$. We chose the up-going slope because it is necessarily steeper than the down-going slope because we are limited to a <50% duty cycle. This tells us the rate of current rise across the first shunt resistor. Due to the low-pass filtering of the capacitor, the rate of rise of the second shunt resistor will be smaller. For a given rise in current ΔI , the voltage across the shunt resistor will rise by $\Delta V = \Delta I \cdot R_{shunt}$. Therefore, the slew rate of the voltage rise across the shunt resistor will be $\frac{dV_{shunt}}{dt} = \frac{dI}{dt} \cdot R_{shunt} = \frac{(V_g - V_{bat})}{L} \cdot R_{shunt}$.

```
In [ ]: Rshunt = 50 * units.milliohm
Vg = 40 * units.volt
Vbat = 12 * units.volt
L = 345 * units.microhenry
dIdt = (Vg - Vbat) / L * Rshunt
print(f"Slew rate = {dIdt.to(units.volt / units.microsecond):.2e}")
```

Slew rate = 4.06e-03 volt / microsecond

Shunt resistor power rating

```
In [ ]: Ipeak = 2 * units.amp
Rshunt = 10 * units.milliohm
shunt_resistor_power_rating = Ipeak ** 2 * Rshunt
print(f"Shunt Resistor Power = {shunt_resistor_power_rating.to_compact(units.watt)}")
```

Shunt Resistor Power = 40.00 milliwatt

LED resistor calculations

3 voltage levels: 15V, 5V, and 3.3V 2V forward voltage across LEDs 30mA maximum DC current, aiming for 10mA

```
In [ ]: V1 = 50 * units.volt
        V2 = 5 * units.volt
        V3 = 3.3 * units.volt
        VF = 2 * units.volt
        Idesired = 10 * units.milliamp

        R1 = (V1 - VF) / Idesired
        R2 = (V2 - VF) / Idesired
        R3 = (V3 - VF) / Idesired
        print(f"R1 = {R1.to_compact(units.ohm):.2f}")
        print(f"R2 = {R2.to_compact(units.ohm):.2f}")
        print(f"R3 = {R3.to_compact(units.ohm):.2f}")

R1 = 4.80 kilohm
R2 = 300.00 ohm
R3 = 130.00 ohm
```