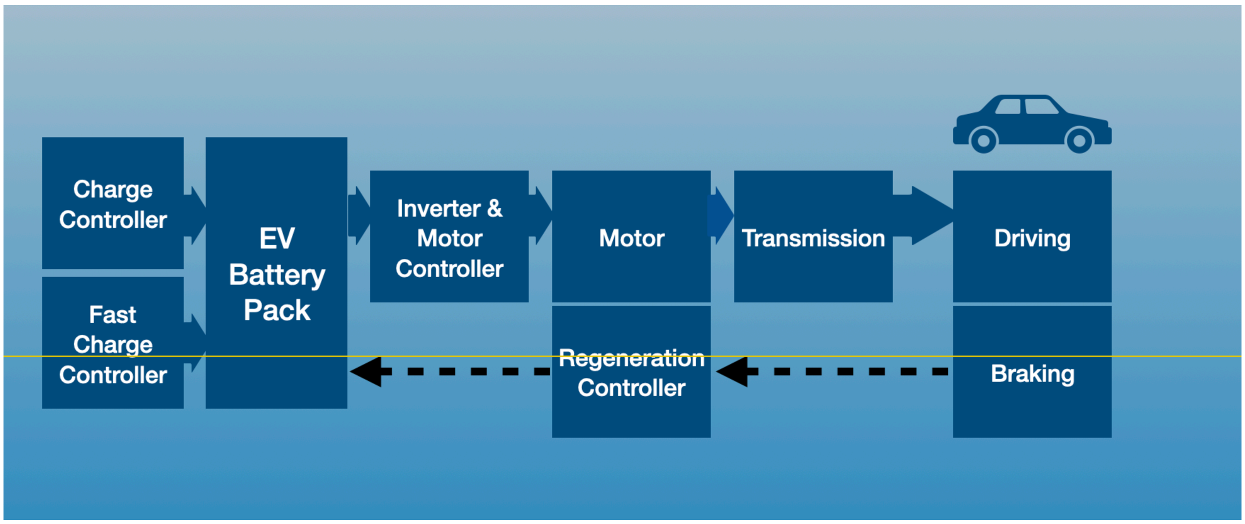
Power Train Efficiency Study.

Current EV Drivetrain design.



Block diagram of current EV Drivetrain design

Overall EV Drive Train design characteristics.

- Battery pack is used to generate a deterministic DC value called V-Bus. Typical V-bus in EV is around 400-800V.
- Battery pack includes BMS to monitor cells and allow active/passive cell balancing. This is a key to make sure the battery pack cells SOC/SOH stay close to each other, as divergence will cause imbalance stress levels between cells.
- Different Charge controllers are needed to charge Battery pack from different sources, AC 110/220 V, DC fast charge and so on.
- Motor controller is needed to convert a fix V-BUS into 3-phase AC waveform that generates the right phase/amplitude on the phases of BLDC motor based on rotor position(FOC). (AC induction motors have similar requirements)
- Motor controller uses high power switches with control loop and rotor position informations to drive the exact current phase/amplitude on each of the three motor phases.

- Braking controller is needed to allow brake regeneration to charge the battery pack. This controller converts 3-phase AC input (based on motor RPM/position) into a DC output that should be very close to V-BUS. Protection path to diverts the extra energy to resistance to avoid overcharging of the battery pack.
- Since V-BUS is fixed for most configurations, finding an optional V-BUS value can be challenging as it needs to support different motor RPM (slow start vs Highway speed) as well as different charge controllers.
- In some cases a multi-level transmissions is required to reduce the supported range for RPM/Torque.
- Motor controller/regeneration controller need to support the peak current/voltage for all of the operating conditions. This cause scalability issue when power train performance needs to increase, since support for larger amount of current/voltage increases the cost and complexity as well as heat.
- The Motor controller have to use relatively low frequency switching rate (typical 10KHz) due to component limitation at high power and noise generated by fast switching high current.

Simulation highlight.

To fully understand the limitations of using current motor controller and the challenges described above, we developed a simulation using LTSPICE and Python. The goal of such simulation is to further understand the losses happening in motor controller due to switching frequency, Control loop convergence speed, switching losses, harmonic losses, and so on; it also helps to understand the controller behavior/system performance outside the of the preferred operating range (ie high rpm, low rpm, high torque low torque conditions) and the effect of such conditions on overall losses and efficiency.

Simulation circuit.

In this simulation, we represent BLDC motor with there voltage source generators (V EMF) each with 120 degree phase shift between each two:

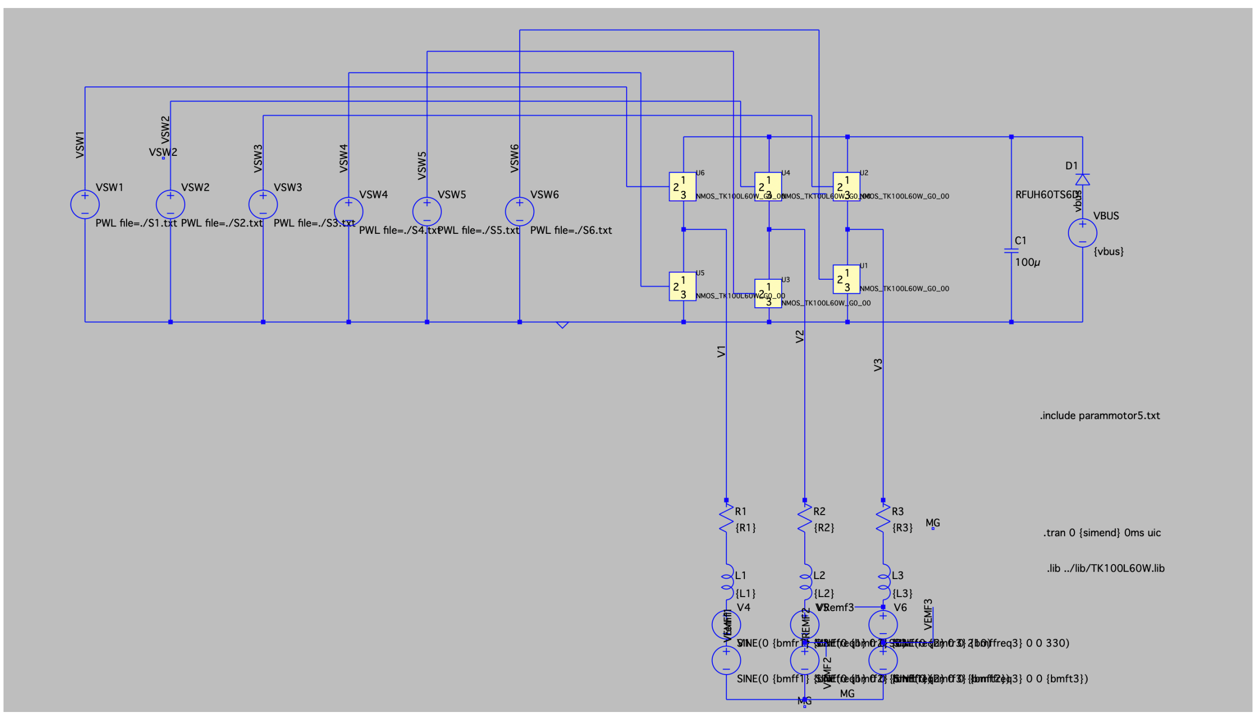


Figure 1: Simulation Circuit Schematics

The frequency of Back EMF (bemf) is a function of rpm:

bmffreq=args.rpm*4/60

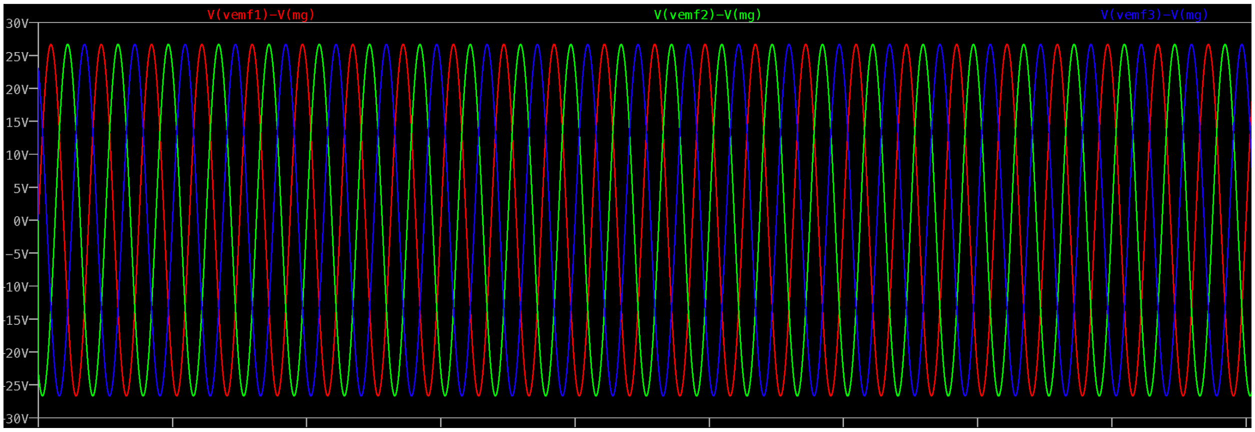


Figure 2: Three Phases Motor Back-EMF

Typical BLDC motor have 4 poles, meaning each mechanical rotation is equivalent to 4 electrical ones;

Back EMF voltage is a function of rpm as well:

bmff=bmffreq/emfK

emfK can be configured in simulation with a value of 1.0 used in simulation.

Besides the Back EMF voltage source at each phase, motor can be represented by inductor/ resistor at each phase, the value of the inductor and resistor can be set in summation:
In this simulation 200 m-ohm used for Resistor per phase, and 5 mH for Inductor.

In this simulation the battery pack are simulated as a voltage source of V-BUS=550V.
The Battery pack is protected by capacitor C(100uF) and Diode D1 to prevent the transient large current to go back to battery pack and cause harm.

The core controller use 6 high efficient Toshiba NMOS switches with very low Ron resister(TK100L60W)

The 6 NMOS switches are driven by a controllable voltage sources (SW1..SW6) with enough voltage to turn on/off this switches at right time.

The goal of such control loop is to generate three phases current that each are in sync with Back EMF, any angle different between phase current and Back EMFwill cause losses of efficiency.

Controller overview.

This picture below show how three phase investor works.
Basically the 6 Switches can generate combined 6 possible Vectors plus a Null vector:

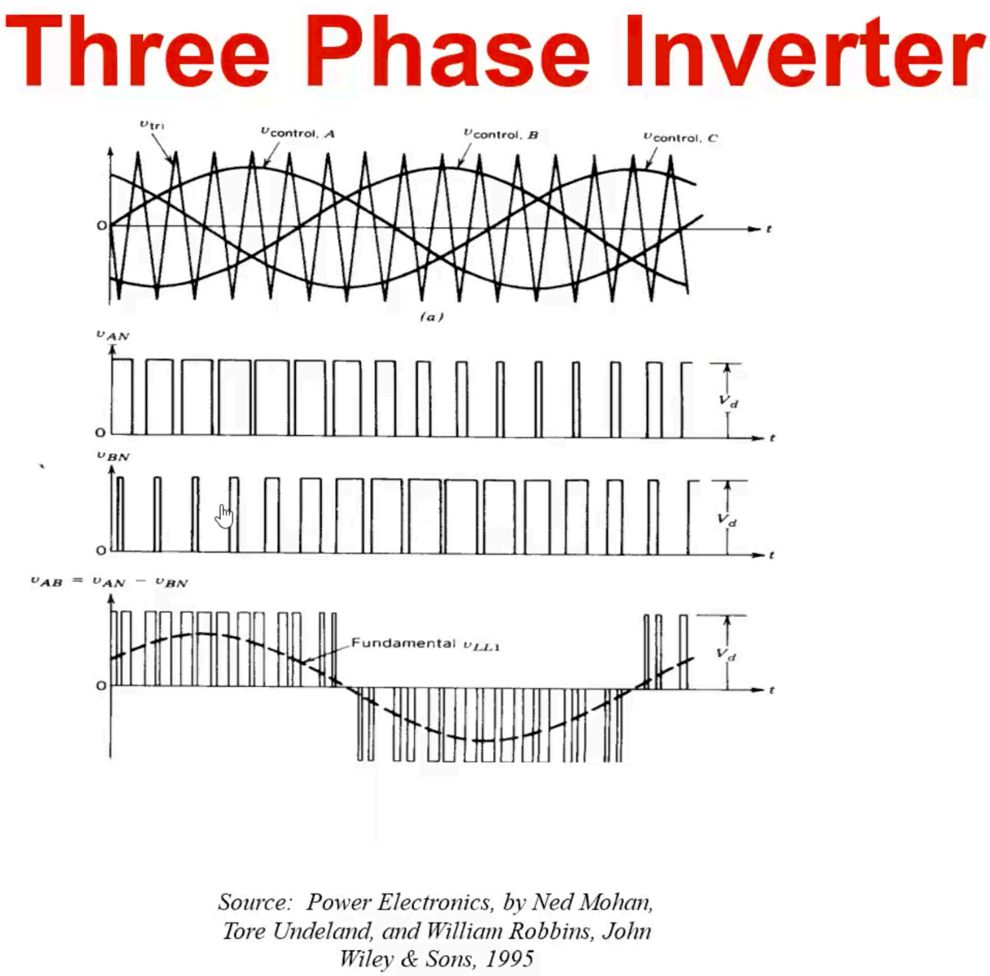


Figure 3: Three Phase Inverter

FOC relies on providing optimal total force vector perpendicular to motor position to provide max flux and use all power to rotate. This is key to reduce heat losses and increase efficiency. It is also key to reduce vibrations. Below is an example of implementing FOC.

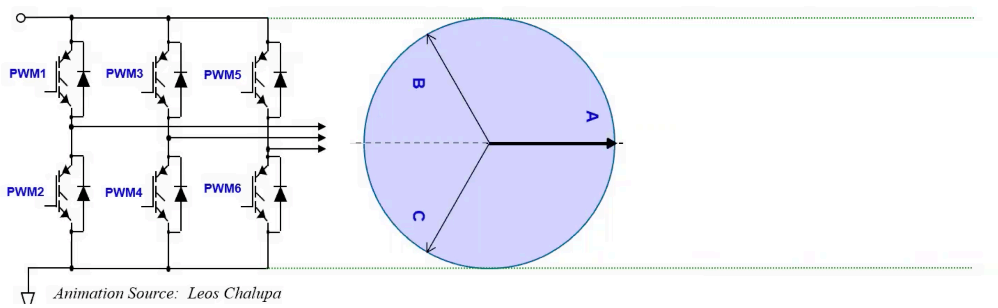


Figure 4: Output Current Vector Generated by 6-switch topology

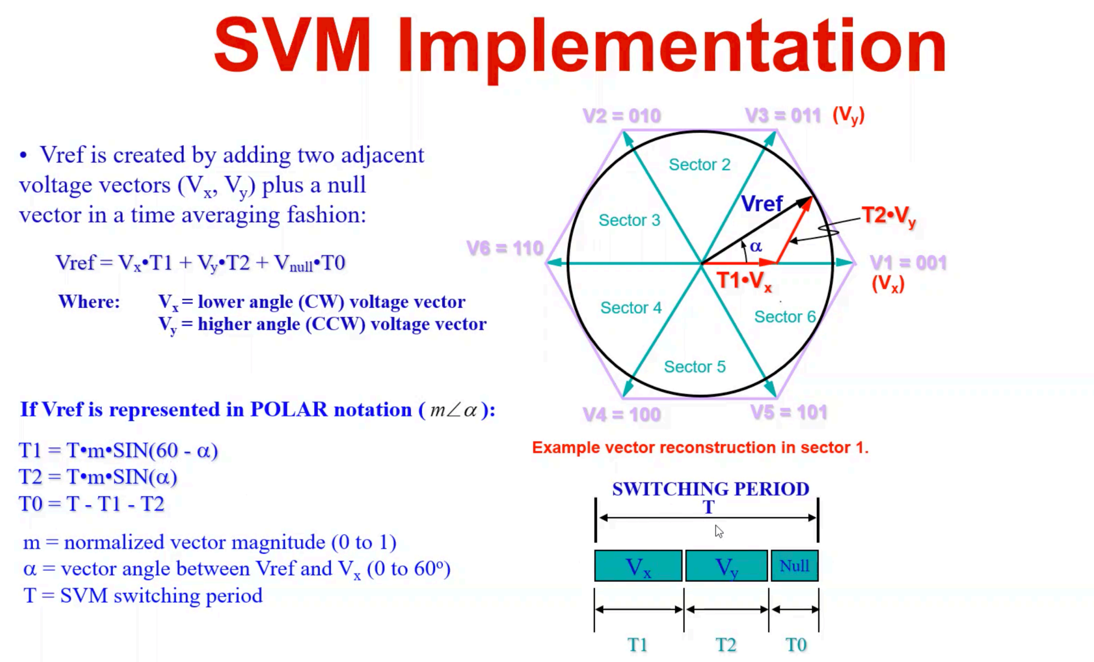


Figure 5: FOC Implementation (source: Magna Motors LLC)

Basically, for each rotor position, a duty cycle need to be generated for Vx/Vy/Null where Vx/Vy ratio relate to how close Vref to Vx vs Vy. While Null ratio allow scaling the power to increase/decrease torque.

Python overview.

Python code receives all simulation parameters, e.g. motor rpm/torque VBUS and motor resister/inductor and so on.
Python then generates the 6 SW files that drive VSW1..VSW6 based on motor RPM and torque requirements. After generating the input files, Python runs LTSpice, collects all the results (analog waveforms of the pertinent signals) and post processes them later.

The table below describes the all input parameters.

Parameter	Description
--st	Starting theta
--et	Ending theta
--ts	theta steps
--uc	number of switch period before updating the angle (convergence time)'
--te	Max time Error in usec
--swfreq	Switch frequency in Hz
--vbus	VBUS level (Volts)
--rpm	RPM mechanical rotation with x4 electrical
--swonv	Switch ON voltage
--swoffv	Switch OFF voltage
--swot	Switch dead time (in ns) when all the switches are off;
--scale	Percent of ON time, i.e. torque
--simdur	Duration of simulation in ms
--emfK	RPM (in Hz) to Back EMF voltage scaler
--R	Motor per phase resistance (in mOhm)
--L	Motor per phase inductance (in mH)
--harmonic	Calculate harmonic losses
--fn	Output file name

Here is an example of the input parameters:

python3 gen6s_4.py --swonv 560 --swfreq 10000 --swot 400 --swoffv 0 --rpm 2000 --emfK 1 --RemfK 100 --scale 60 --R 200 --st 300 --et 400 --ts 5 --uc 10 --harmonic 1 --pn motor5 --vbus 550 --simdur 200

In this example we specify:
--swonv 560 (voltage to turn on the NMOS) in V
-swoffv 0 (voltage to turn off the NMOS) in V
--swfreq 10000 (the switching frequency of the NMOS) in HZ
--swot 400 (how many nanosecond we keep all switches off between changing switches state "off time") in nanosecond
--emfK 1 (ratio between RPM/Back EMF)
--scale 60 (40% in NULL and 60% driving the motor) in %
—rpm 2000 (rotation per minute) in rotation
--R 200 (motor resister per phase) in mOhm.
--st 300 (start angle between phase 0 and Back EMF) in degree
--et 400 (end angle between phase 0 and Back EMF) in degree
--ts 5 (theta step between start to end) in degree
--uc 10 (how many cycles to keep the control loop at same target) in cycles.
--vbus 550 VBUS Voltage in V
--simdur 200 (Simulation duration) in msec

Explanation about UC parameter.

In real life, motor controller use a control loop to drive the switches so that the three phases current vector are in sync with rotor position (FOC). To achieve this using the 6 SW each position represent a one/six vector states. A desired results by switching duty cycle between two adjacent vector states. Since the control loop needs time to converge we assume the target need to stay still for multiple switching cycles, otherwise the control loop might not converge. UC=1 represent an idealistic control loop that allows to change target each switching cycle (~10Khz in this case) while UC=10 means that for 10 switching cycles we keep the desired output the same, then change it only after 10 cycles and so on. A typical control system needs 5 switching cycles to converge and 5 to keep it stable. So UC=10 should be the typical case. While UC=20 represent very slow convergence of 10 switching cycle.

Explanation of Theta.

Since the circuit have active components (Z load) it is hard to predict the best angle that will cause the current to be in sync with Back Emf, since LTSpice/Python does not fully simulate the control loop, we scan all possible values of theta to find the best angle that will get the desired current phase alignment with respect to Back Emf. Once the optimal value of theta found, it will be used to collect the simulation data for this specific configurations (rpm/torque/switching freq/resister/inductor value and so on). This can be achieved by first running all possible theta then pick the winner for each configuration and run it for larger simulation durations (800 msec)

Results analysis.

The LTSpice simulation generate all current/voltage at any point of the circuit which allow python to post process it and generate all losses/efficiencies and other metrics.

pn=motor5,**simdur**=200,**rpm**=1000,**swfreq**=10000,**R**=200,**L**=5,**scale**=70,**uc**=15,**emfK**=1.0,\
RemfK=100.0,**swot**=400,**theta**=300,**phase1**=-58.05936000000003,\
phase2=-53.29151999999999,**phase3**=-65.54424,**phase12**=114.58224000000001,\
phase23=133.11504,**phase31**=112.30272000000001,**IPT**=-5187.464234033038,\
IPN=-5237.272899783411,**IPP**=49.80866575097231,**IPRMS**=19759.848,\
SWL1=68.40208396802177,**SWL2**=68.78818865761497,**SWL3**=73.54495788779742,\
SWL4=167.1644711968488,**SWL5**=169.71355542069435,**SWL6**=159.8511842497563,\
SWLT=707.4644413807337,**RL1**=522.7017128032982,**RL2**=538.2407136877831,\
RL3=532.2459634018544,**OP1T**=926.0724594176103,**OP1N**=-356.77815829089974,\
OP1P=1282.8506177085133,**OP1RMS**=1975.0673,**OP2T**=999.2553095713725,\
OP2N=-343.1765869150584,**OP2P**=1342.4318964864583,**OP2RMS**=1985.3706,\
OP3T=1011.5625230331156,**OP3N**=-343.5407104568908,**OP3P**=1355.1032334900078,\
OP2RMS=1968.2606,**OP1F**=727.1709,**OP2F**=753.12427,**OP3F**=626.5302,\
OP1FRMS=1859.8208,**OP2FRMS**=1894.6075,**OP3FRMS**=1834.4551,\
GENR=0.262159378967165,**EFF**=56.615142958562394,**EFF2**=40.61378246487023,\
SWLOSSES=0.13637962778409546,**RLOSSES**=0.3071227709755789,

Key results details.

IPT: Input power total, is the vector multiplication of V-BUS with I-BUS, this represent the total power that was pulled from the battery pack.

OP1T/OP2T/OP3T: per phase actual output power that was spent motoring. This is the product vector of the Back-EMF and the current of that phase.

OP1N/OP2N/OP3N: per phase negative power generated from the motor, this happens when the motor was not motoring but generating and is felt as undesired braking during motoring. This can cause unpleasant driving experience and create losses since part of the extra energy generated will get lost in heat and inefficiency. The goal is always to minimize this power during motoring, and maximize it during active braking.

This is an example for OPT1 where the majority of power are positive, but small portion is negative (due to out-of-sync between current and back-emf)

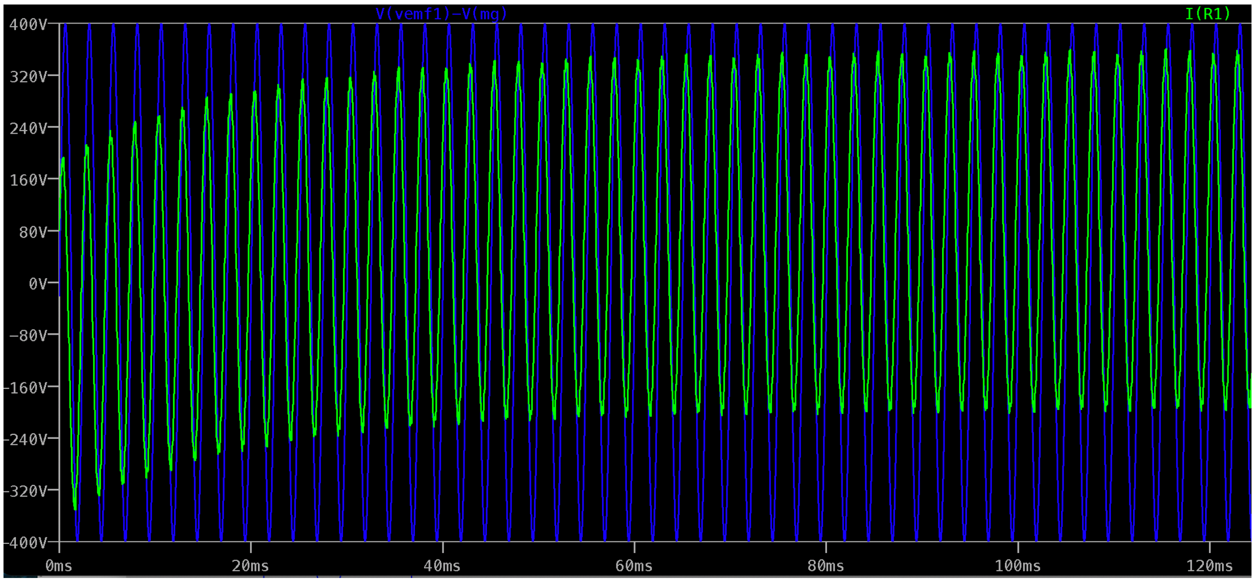


Figure 6: Phase Current Aligned with Back EMF Voltage Waveform

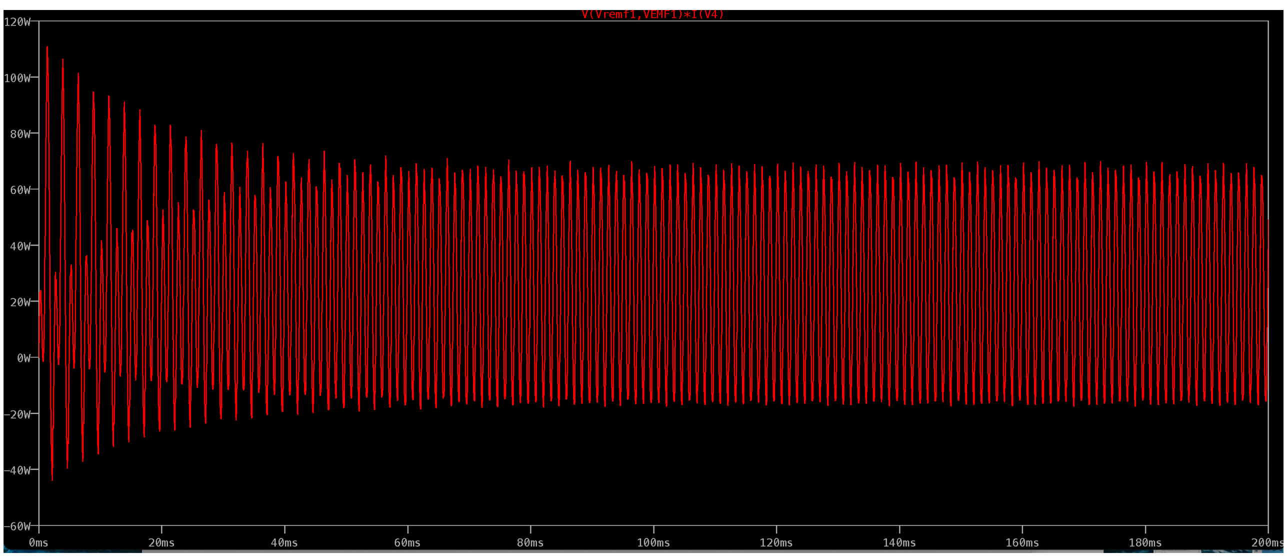


Figure 7: Phase Output Power

GENR: is the generation ratio from total power per phase, this get calculated by having the ratio of total negative power and total output power. This should be very close to 0 in normal operations.

SWL1..SWL6: are the losses on each of the 6 Switches calculated by adding the gate current multiply with gate voltage drop, and source/drain current multiply by source/drain voltage drop:

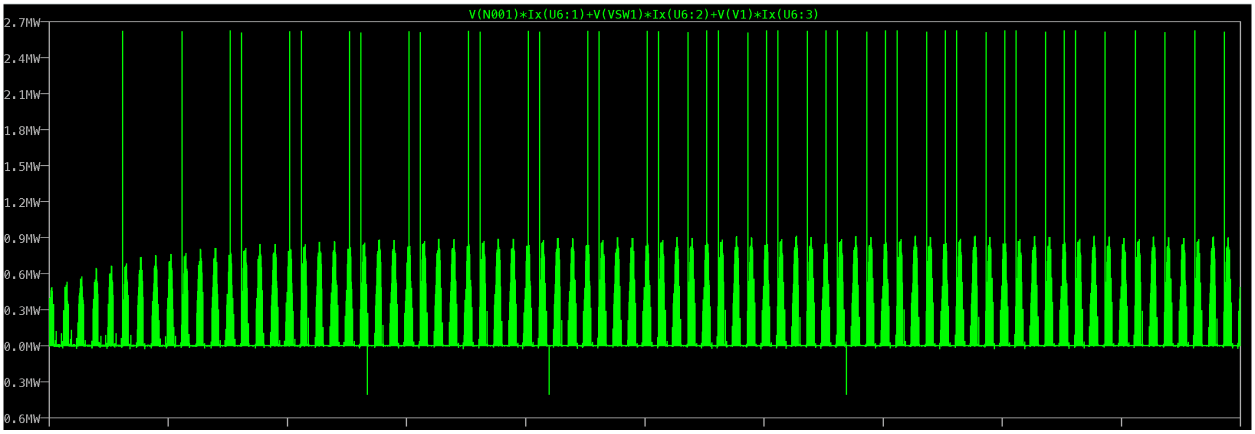


Figure 8: Switching Losses

SWLOSSES: is the sum of all losses in all the switches.

RLOSSES: are all losses on resisters (multiplying current by voltage drop on each phase resister and add the results)

HLOSSES: Harmonic losses cause by harmonic frequency that are outside the desired motor rpm. It is calculated by using FFT of the output power and filter only the desired frequency that match the rpm.

GENLOSSES: energy lost from motor generation during motoring, it is calculated by calculating of 25% of generation energy

EFF: Controller/Motor Efficiency calculated by the ratio of all output power and input power.

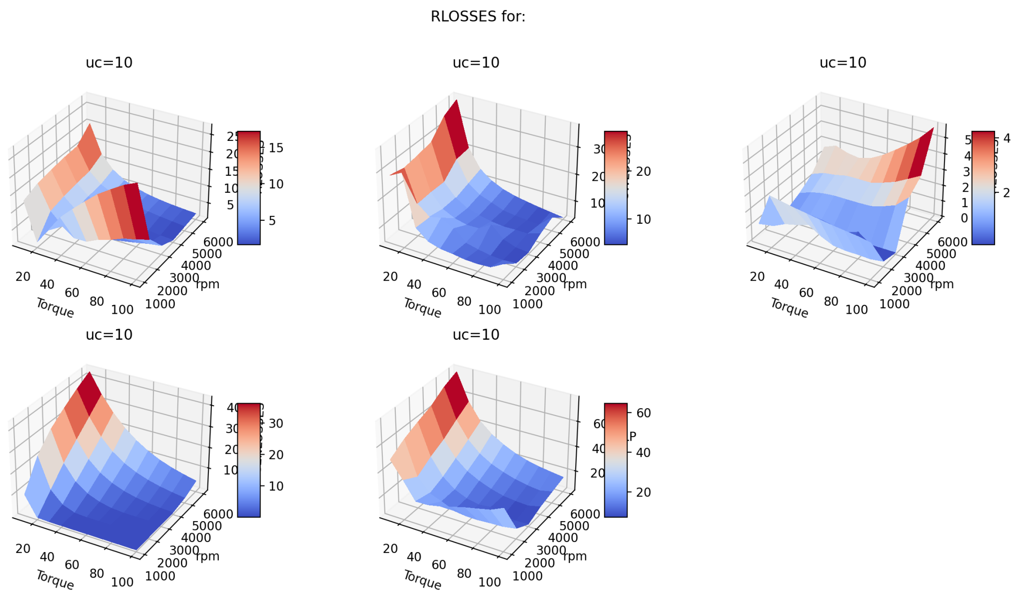
EFF2: Only use first harmonic of the current waveform to calculate the output power, as higher frequency harmonic will cause undesired forces and should be considered as losses.

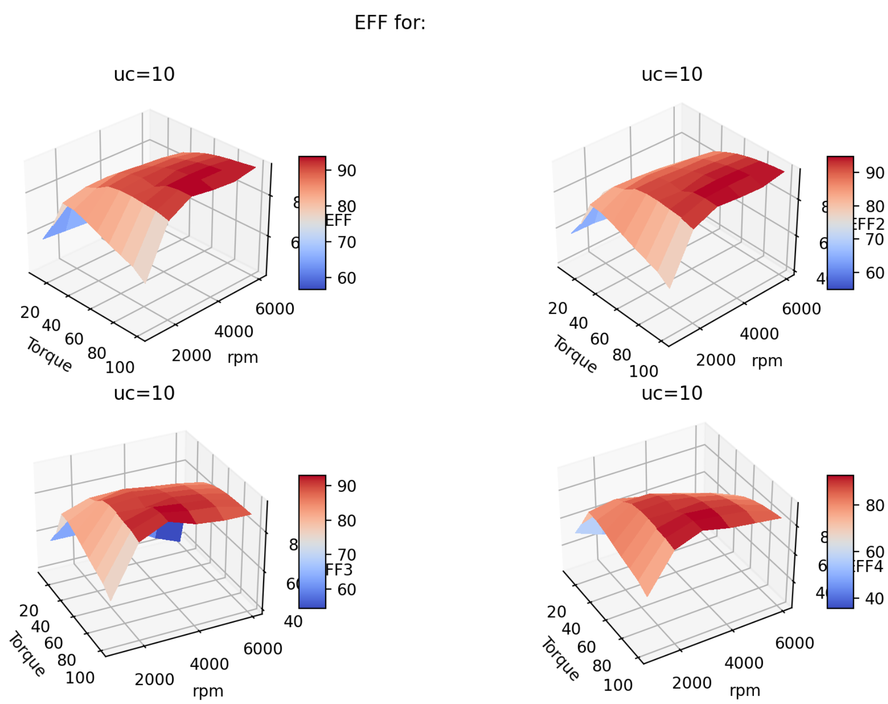
EFF3: take into account the RMS ratio between Input power and Output power.

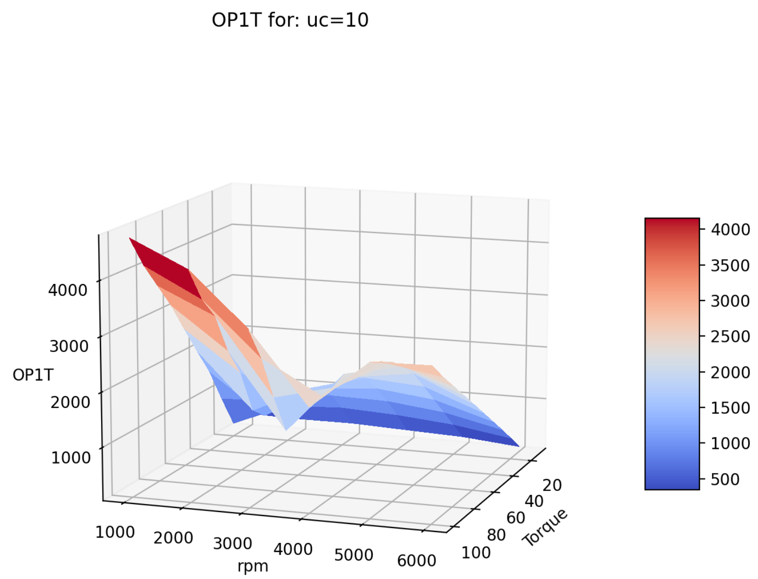
EFF4: Include GENLOSSES (penalize generation power by 25% since in reality generation power need to go through the system losses and do not fully add to total power in the circuit losses such as generation efficiency and heat losses)

Results analysis.

These graphs show all losses as function of RPM and Torque:







Observations.

- Motor/controller efficiency in the steady state and @ sweet spot of rpm/torque could reach 80-90%;
- Very Low Torque causes high losses since V-BUS can not be **decreased**, a high V-BUS will need very high NULL time which will not allow the current to be in sync with Back-EMF.
- At high RPM, Back EMF is very high, since V-BUS can not be **increased** this will not allow the current to be in sync with Back-EMF.
- Also 10KHz switching frequency will cause a challenge at high RPM to create a good waveform, increasing the harmonic losses and causing motor generation.
- High Torque creates more resistor losses since it requires high current.
- Higher switching frequency can cause significant increase in switching losses and create a challenge to meet EMI requirements, but it can reduce the generation losses and harmonic losses.
- Higher rpm can not be achieved efficiently without gear shift;
- Lower torque can not be used beyond certain level (30% duty cycle in this example) which will limit the range of operation.
- Variable V-BUS with higher switching frequency can provide a better alternative.
- Adding transmission won't solve all issues, as it is only tradeoff between rpm/torque but it doesn’t cover all rpm/torque regions where efficiency is very low.
- In some applications two separate motors are used one for starting and one for higher speed, with current/voltage tradeoff that suite each use cases.
- This study only simulate steady state where rpm/torque are fix for long duration of time (over many rotations). In case of transients, when motor is accelerating or decelerating or adding/

reducing torque (like in rough terrain or on change in road incline, change of speed and so on) the control loop will take longer time to converge creating more phase between current and Back-EMF, thus decreasing efficiency rapidly.

- Overall average motor/controller efficiency is around 65% in range of the acceptable EV efficiency studies.

Design tradeoffs.

- Picking the right V-BUS is the key that dictates the rpm/torque range that can still generate decent efficiency and acceptable or no regenerations.
- Picking the switching frequency is the key but can be dictated by EMI and switch limitations.