Bevindingen i.v.m. VESC 6:

* BLDC-detectie is zeker niet verbeterd, eventueel zelfs moeilijker geworden om parameters te detecteren in vergelijking met VESC 4. (Voor maxonmotoren: met koppeling)
* Verschillende malen dezelfde detectie uitvoeren levert niet altijd dezelfde reactie van de motor, evenals de detectieresultaten: soms niet anders wel.
* Niet duidelijk wat de aansturing met de pijltjes doet, ook soms anders. Dit is afhankelijk van instellingen die ik nog niet gevonden heb (dus niet van FOC <-> BLDC-aansturing). De pijltjes links en rechts lijken de Duty op -10 resp. 18 te zetten, maar boven/onder levert soms andere resultaten op. Soms meteen +-95 duty, anders een zachte oploop (vermoedelijk current control). Ik denk niet dat ik hiervoor parameters veranderd heb. (Gevonden in code mainwindow.cpp line 209 e.v.: pijltjes boven/onder is current control met als current de huidige waarde rechtsonder in de UI, links en rechts is duty control, maar met een ingebouwde gradient + een extra verschil als beide pijltjes ingedrukt zijn. Zie mainwindow.cpp line 343 e.v.)
* Een extra functietoets pagedown is in de code gevonden. Dit is de break (met breakcurrent)
* FOC – detectie véél beter. Waar het bij vorige versies nog verkeerd liep – o.a. motor stuk, en nooit detectie-, liep de detectie nu vlotjes.
* FOC – detectie volgens mij niet altijd correct. Waarden als 1000 voor Kp lijken met gewoon de grens die hij toelaat. Toch liep current control ok. (Gevonden in code detectfoc.ui line 393: dit is inderdaad gewoon een maximum. Misschien beter een error geven als zo’n waarde gevonden wordt.)
* FOC met hall-sensoren werkte in mijn geval slechter dan sensorless. Dit lag misschien ook aan de kwaliteit van de sensoren.
* De documentatie en uitleg omtrent wat verschillende parameters zijn en doen is veel beter geworden in vergelijking met de vorige versie, maar kan nog altijd beter. (vb: bij omega van detectie van BLDC)
* In sommige aanduidingen i.v.m. snelheid staat RPM, andere keren staat er ERPM. Ik vermoed dat allen ERPM moeten zijn, maar ben niet zeker.
* Detectie van BLDC (en in mindere maten FOC) bleek afhankelijk van een parameter die niet in de wizard voorkwam: de spanning die werd aangelegd aan de VESC. Specifieker: de battery voltage cutoff start en battery voltage cutoff end.
* Position control lijkt goed te werken... met encoder.
* Hall/encoder detectie zeer goed. Een enkele keer lukte het niet om een encoder te detecteren. Dit was omdat de encoder stuk was.
* Extra uitleg over de hall tabel. Nu is het gokwerk wat deze betekent. (Gevonden in de code: mc\_pwm\_foc.cpp line 1423 e.v. Dit is de hoek van de hall sensor in graden maal 200/360 en dit afgerond naar een int. Verder 255 indien hall sensor niet gebruikt. Er worden altijd 2 sensoren verwacht op 255 te staan, anders wordt de hall sensor niet herkend. De waarden zijn een gemiddelde van een aantal meetingen.)
* Kleinere motoren zijn niet volledig ondersteund. Bijvoorbeeld: bij de detectie wordt een waarde 1568 voor BEMF Coupling gevonden. Deze wordt teruggebracht tot 900 standaard. Waarom? Verschillende parameters worden afgekapt (clamped) op specifieke waarden. (Zie FOC-detectie)

(Gevonden in de code: detectbldc.cpp line 134 e.v.)

* Er is een commando om de Kv te berekenen. Waarom geen knop?
* Als de index van de encoder niet gevonden wordt, loopt de FOC in openloop. (mcpwm\_foc line 1754)
* Current in de motor in FOC is: ingewikkeld. Zie control\_current in mcpwm\_foc.c
* Deze worden berekend op basis van de gemeten stromen en de Clarke tranformatie.
* In BLDC is de ‘current in’ van de motor gewoon de stroom van de huidige commutatie.
* Tijd tussen versturen van boodschap en verkrijgen = 0.1s (GET\_VALUES commando) Waarde verkregen met timing met python.
* Minimum duty = 0.06 % in FOC voor het dubbele wiel (in zonder last)
* Tijdens real time data is de updateperiode +- 0.3 s. Deze waarde is verkregen door met een listener naar de packets te kijken die verstuurd worden.