# Propeller Electrical Power-train Design

## Model, control strategy and hardware dimensioning

**Davide Bagnara**

December 17, 2023

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Preliminary Electro-thermal Simulations

The following chapter concerns the preliminary dimensioning of an electrical drive for permanent magnet synchronous motor (PMSM) for drone application. The chapter in question consists of the following parts:

- parameters identification of the selected PMSM;
- preliminary dimensioning of the hardware components, DClink, MOSFET, Heat-Sink, etc.;
- preliminary electro-thermal performance evaluation via Simulink/Simscape modelization;
- thermal analysis.

## 1.1 Motor Parameter Identification

In this section the motor data are here summarized and the main motor parameters, necessary for the Simulink model implementation as well as for the preliminary electrical dimensioning, are here extrapolated.

**The motor in question is here reported as follows**:

- MAD M50C35 EEE 9KV.

The M50C35 EEE 9KV PMSM synchronous motor is supplied by *MAD Components*, and the main characteristics, in term of voltage/current for nominal torque rating are referred to the DC-Battery stage, as Table 1.1 reproduces.

| **Quantity** | **Nominal value** |
|---|---|
| Motor Model | M50C35 EEE 9KV |
| Nominal DC Voltage (battery side) | 100S (lipo) – (360 V - 400 V) – Max 126 °C |
| RPM/V | 9 |
| Internal Battery Resistance | 59 mΩ |
| Maximum DC Current | 49.1 A |
| Maximum Torque | 64 N m |
| Maximum Power | 19.5 kW |
| Nominal RPM | 2291 $\mathrm{min}^{-1}$ |
| Number of poles | 62 |
| Recommended ESC | MAD AMPX 60A (100S) |
| Motor Weight | 4.2 kg |

Table 1.1: Motor Parameters from MAD Components Specification.

| **Battery Voltage** | **Battery Current** | **Input Power** | **Shaft Power** | **Torque** | **RPM** |
|---|---|---|---|---|---|
| 398 V | 15.4 A | 6.1 kW | 5 kW | 29.8 N m | 1600 $\mathrm{min}^{-1}$ |

Table 1.2: Continuos Operative Data (from MAD Components).

| **Battery Voltage** | **Battery Current** | **Input Power** | **Shaft Power** | **Torque** | **RPM** |
|---|---|---|---|---|---|
| 397 V | 49.1 A | 19.5 kW | 15.3 kW | 63.9 N m | 2291 $\mathrm{min}^{-1}$ |

Table 1.3: Short Term (30 s) Operative Data (from MAD Components).

By the data available from Table 1.1, Table 1.2 and Table 1.3 a set of motor parameters has been calculated/estimated as reported in Table 1.4.

| **BEMF Voltage** (2291 $\mathrm{min}^{-1}$) | **Nominal Current** | **Synch. Induct.** | **Flux Link**. |
|---|---|---|---|
| $u_{bez} = 160\,\mathrm{V}$ (peak) | $i_{bez} = 64\,\mathrm{A}$ (peak) | $L_s = 120 \div 200\,\mu\mathrm{H}$ | $\psi^m = 0.0215\,\mathrm{V\,s}$ |

Table 1.4: M50C35 EEE 9KV Estimated Motor Parameters (for control system).

Where $u_{bez}$ and $i_{bez}$ values represent the peak of the phase (phasor diagram). The motor is supposed isotropic ($L_s = L_d = L_q$).

Additional motor parameters representation, for thermal and power evaluation, are in Table 1.5 reported.

| **Load Voltage** $(2291\,\mathrm{min}^{-1})$ | **Nominal Current** | **Power Factor** | **Phase Resistance** |
|---|---|---|---|
| $U_m = 247 \div 262\,\mathrm{V}$ (ph-ph RMS) | $I_m = 45\,\mathrm{A}$ (RMS) | $\cos(\phi) = 0.9$ | $R_s = 0.5\,\Omega$ |

Table 1.5: M50C35 EEE 9KV Estimated Nominal Motor Parameters.

Where $U_m = 247\,\mathrm{V}$ is related to the synchronous inductance of $L_s = 120\,\mu\mathrm{H}$ and $U_m = 262\,\mathrm{V}$ is related to the synchronous inductance of $L_s = 200\,\mu\mathrm{H}$.

## 1.2 Basic Electro-thermal Inverter Layout

In this section the electrical layout of the inverter used as driver for the motor is shown, see Figure 1.1.

The design of the electro-thermal equipments consist of the the following parts:

- power components: MOSFET modules;
- MOSFET-driver for the power components module;
- isolated power supplies for the MOSFET-drivers;
- current and voltage sensors;
- dc-link capacitor;
- heat-sink for the MOSFET-modules.

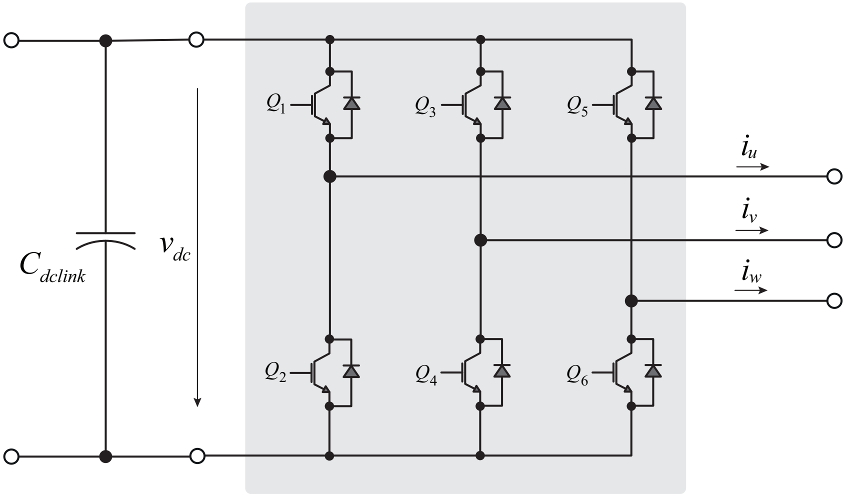


Figure 1.1: Inverter electrical layout principle.

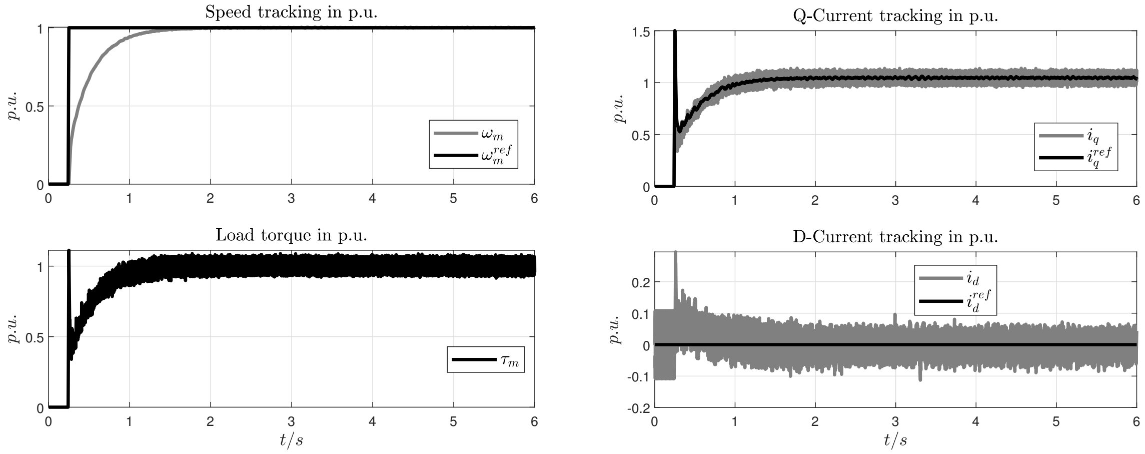
## 1.3 Preliminary Simulation Results

This section shows some preliminary simulation results which have been performed in order to determine the main physical quantities used to design and sizing the hardware components of the inverter and to proof the feasibility of the control system architecture in term of sampling time and switching frequency.

Simulations have been implemented using the following data:

- nominal speed : $n = 2291\,\mathrm{min}^{-1}$, which results in a fundamental frequency of $f_e = 1184\,\mathrm{Hz}$;
- nominal load current : $i_{load}^{nom} = 64\,\mathrm{A}$;
- nominal load torque : $\tau_e = 64\,\mathrm{N\,m}$;
- rotor position observer based on Back-EMF if the PMSM;
- modulation strategy based on space vector at fixed switching frequency of $f_{sw} = 25\,\mathrm{kHz}$;
- DC-link capacitance of $C_{dclink} = 90\,\mu\mathrm{F}$.

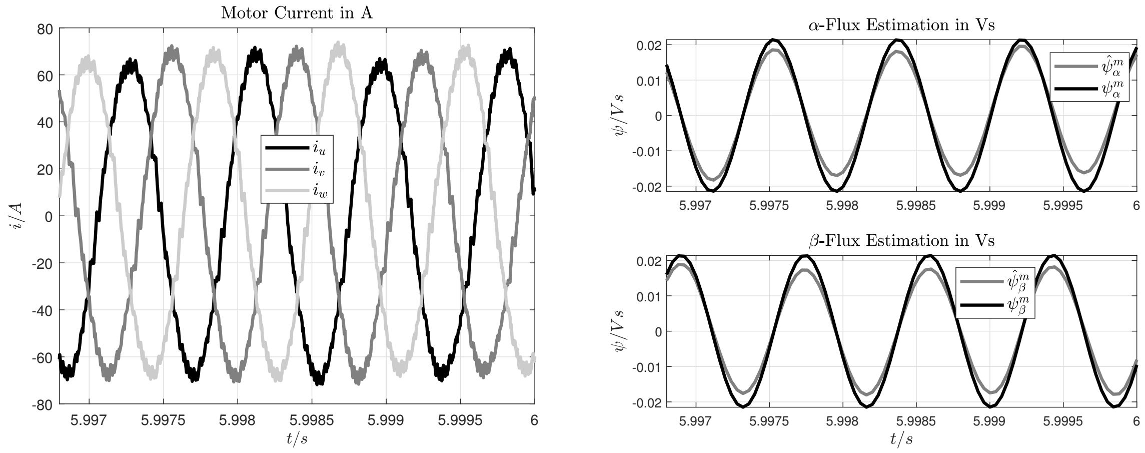
Figure 1.2 to Figure 1.4 show the main motor and inverter quantities involved into the preliminary design of the inverter hardware components.



(a) Speed tracking performance in per unit.

(b) Quadrature and direct currents tracking performance in per unit..

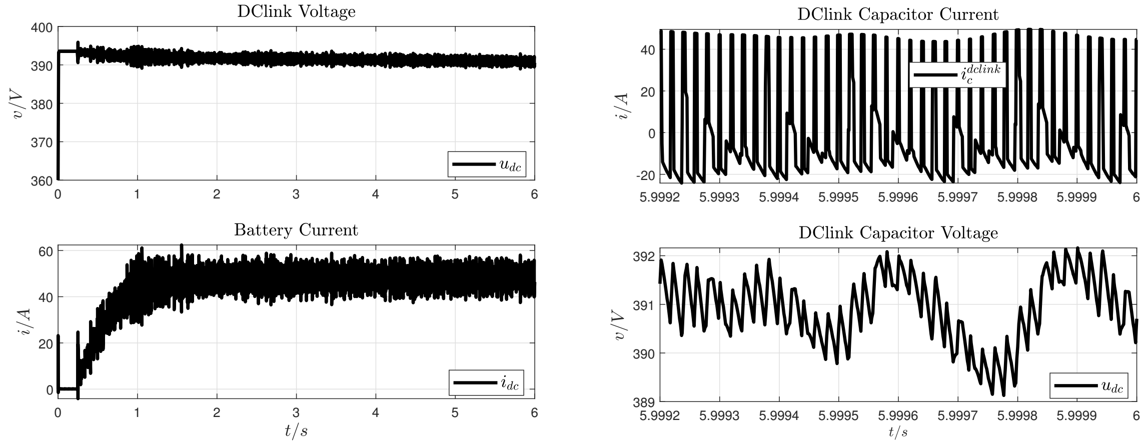
Figure 1.2: Tracking performance



(a) Inverter three phase currents in SI.

(b) $\alpha$ and $\beta$ fluxes from PMSM and from Back EMF observer in SI.

Figure 1.3: Inverter currents and fluxes observer performance



(a) DC-link voltage and battery current in SI.

(b) DC-link capacitor voltage and current (24 A RMS)in SI.

Figure 1.4: Battery voltage and current in SI. DC-link capacitor voltage and current in SI

## 1.4 Preliminary Components Selection

As result of an extensive market search for product availability and thermal performance evaluation the selected component is here illustrated. According to Table 1.4 the component **CAB016M12FM3** has been selected, and thermal performance will be presented as well, see Table 1.7.

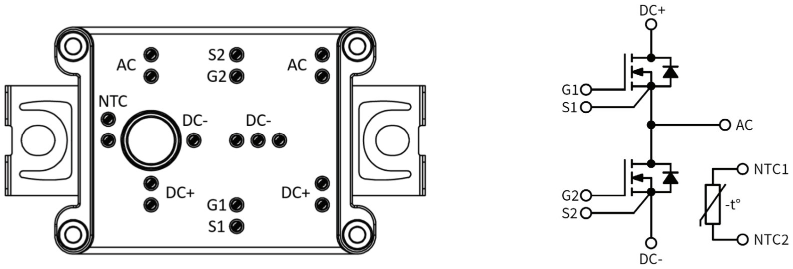


Figure 1.5: CAB016 module description.

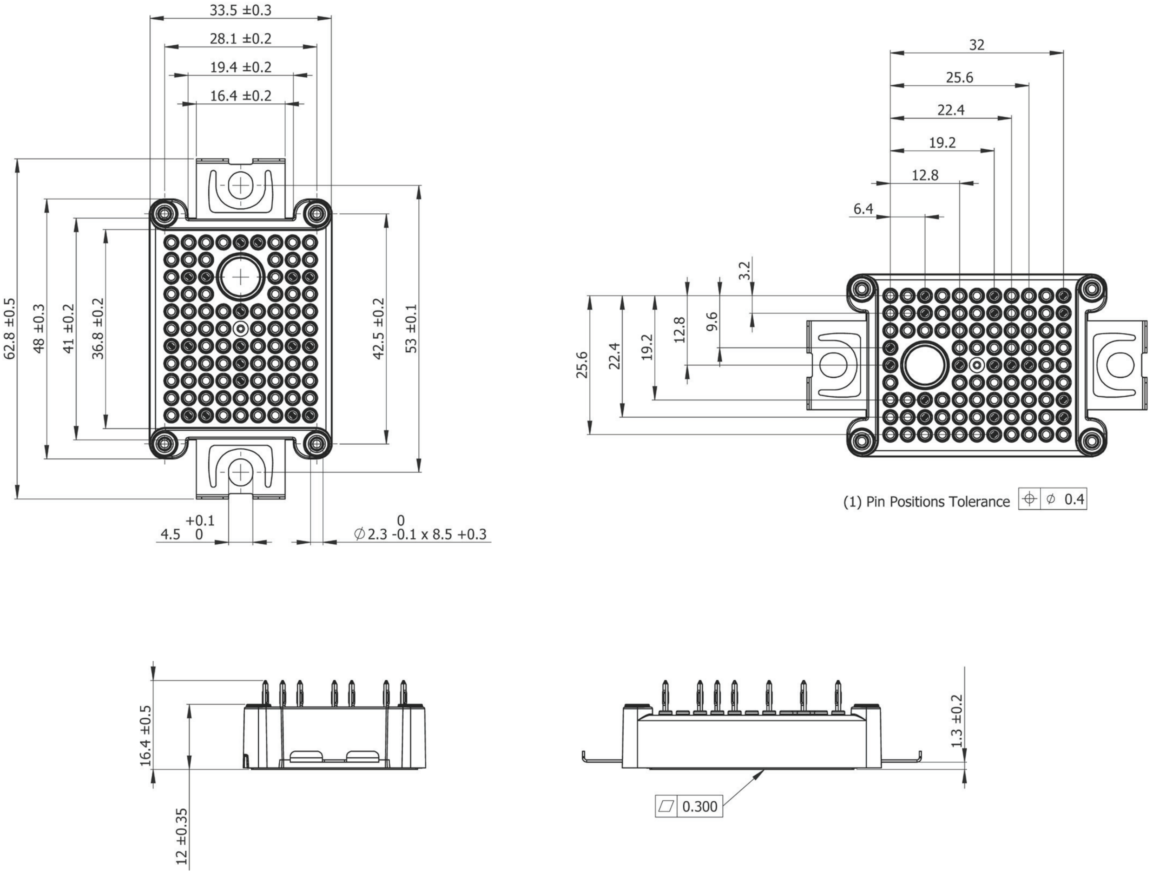


Figure 1.6: CAB016 module size description.

| Quantity | Nominal value |
|---|---|
| Drain-Source Voltage | $V_{DSmax} = 1200\,\mathrm{V}$ |
| Continuos DC (RMS) Drain Current | $I_D = 80\,\mathrm{A}$ |
| Maximum Junction Temp. Switching Conditions | $T_{jop} = 150\,°\mathrm{C}$ |
| Drain-Source On-State Resistance ($T_j = 125\,°\mathrm{C}$) | $R_{DSon} = 21.6\,\mathrm{m\Omega}$ |
| Terminal Resistance ($T_j = 125\,°\mathrm{C}$) | $R_{term} = 2.06\,\mathrm{m\Omega}$ |
| Turn-On Switching Energy ($R_g^{on} = 4\,\Omega$, $T_j = 125\,°\mathrm{C}$, $u_{dc} = 600\,\mathrm{V}$, $i_d = 80\,\mathrm{A}$) | $E_{on} = 1.16\,\mathrm{mJ}$ |
| Turn-Off Switching Energy ($R_g^{off} = 4\,\Omega$, $T_j = 125\,°\mathrm{C}$, $u_{dc} = 600\,\mathrm{V}$, $i_d = 80\,\mathrm{A}$) | $E_{off} = 0.54\,\mathrm{mJ}$ |
| Reserve Recovery Energy ($T_j = 125\,°\mathrm{C}$, $u_{dc} = 600\,\mathrm{V}$, $i_d = 80\,\mathrm{A}$) | $E_{rr} = 0.075\,\mathrm{mJ}$ |
| MOSFET Thermal Resistance, Junction to Heatsink | $R_{jh} = 0.642\,°\mathrm{C\,W^{-1}}$ |

Table 1.6: CAB016M12FM3 Main Datasheet Parameters.

The estimated power losses, for the selected component, are reported in Tables 1.7 and they have been evaluated at the maximum operative working condition in order to extrapolate the maximum junction temperature.

| Model | $P_{sw}^{switch}$ | $P_{cond}^{switch}$ | $P_{loss}^{switch}$ | $P_{loss}^{tot}$ | $T_J^{max}$ |
|---|---|---|---|---|---|
| **CAB016M12FM3** | 26.0 W | 10 W | 36.0 W | 216 W | 110 °C |

Table 1.7: Preliminary power losses evaluation and maximum reached junction temperature.

### 1.4.1 DC-link Capacitor

The initial selection of the DC-link capacitor value starts from the consideration to have a switching voltage ripple around $\Delta v_{dc} \approx 1\,\mathrm{V}$ peak to peak. The preliminary capacitor is here selected using the component **C4AQLBU4560A1WK** from KEMET.

The component **C4AQLBU4560A1WK** has the following characteristics:

- Capacitance: $C = 15 \times 5.6\,\mathrm{\mu F}$;
- Nominal current: $I_c^{nom} = 15 \times 6.8\,\mathrm{A}$ (RMS) at 75 °C;
  - expected current: $i_c = 24\,\mathrm{A}$ (RMS)
- Nominal voltage: $V_C^{nom} = 500\,\mathrm{V} - DC$;
- Dimensions: $L = 32\,\mathrm{mm}$, $T = 11\,\mathrm{mm}$, $H = 20\,\mathrm{mm}$;
- Weight: $Wg = 15 \times 8\,\mathrm{g} = 120\,\mathrm{g}$.

For preliminary design, it is assumed that the maximum junction temperature at nominal condition should not exceed the value of $T_J = 125\,°\mathrm{C}$ supposing a constant heat sink temperature of $T_H = 84\,°\mathrm{C}$.

### 1.4.2 Electro-thermal Simulation Results

The power losses will be evaluated with the following rating condition:

| **Quantity** | **value** |
|---|---|
| Load current | $I_m = 45\,\mathrm{A}$ (RMS) |
| Load voltage | $U_m = 262\,\mathrm{A}$ (ph-ph RMS) |
| Load power factor | $\cos(\phi) = 0.9$ |
| Load fundamental frequency | $f_{load} = 1250\,\mathrm{Hz}$ |
| Inverter switching frequency | $f_{sw} = 25\,\mathrm{kHz}$ |
| DC-link voltage | $v_{dc} = 400\,\mathrm{V}$ |
| External gate ON resistance | $R_g^{on} = 6\,\Omega$ |
| External gate OFF resistance | $R_g^{off} = 6\,\Omega$ |
| Turn-On Switching Energy ($R_g^{on} = 6\,\Omega$, $T_j = 125\,°\mathrm{C}$, $u_{dc} = 600\,\mathrm{V}$, $i_d = 80\,\mathrm{A}$) | $E_{on} = 1.56\,\mathrm{mJ}$ |
| Turn-Off Switching Energy ($R_g^{off} = 6\,\Omega$, $T_j = 125\,°\mathrm{C}$, $u_{dc} = 600\,\mathrm{V}$, $i_d = 80\,\mathrm{A}$) | $E_{off} = 0.84\,\mathrm{mJ}$ |
| Reserve Recovery Energy ($T_j = 125\,°\mathrm{C}$, $u_{dc} = 600\,\mathrm{V}$, $i_d = 80\,\mathrm{A}$) | $E_{rr} = 0.075\,\mathrm{mJ}$ |
| MOSFET Thermal Resistance, Junction to Heatsink (per MOSFET) | $R_{jh} = 0.642\,°\mathrm{C\,W^{-1}}$ |
| Thermal Interface Material (per MOSFET) | $R_{TIM} = 0.5\,°\mathrm{C\,W^{-1}}$ |
| MOSFET Thermal Resistance, Heatsink to Ambient (per MOSFET) | $R_{ha} = 1.88\,°\mathrm{C\,W^{-1}}$ |

Table 1.8: List of condition for the electro-thermal simulation.

Simulations have been carried out neglecting the thermal capacity of the heat-sink in order to evaluate the steady state temperature for different load conditions.

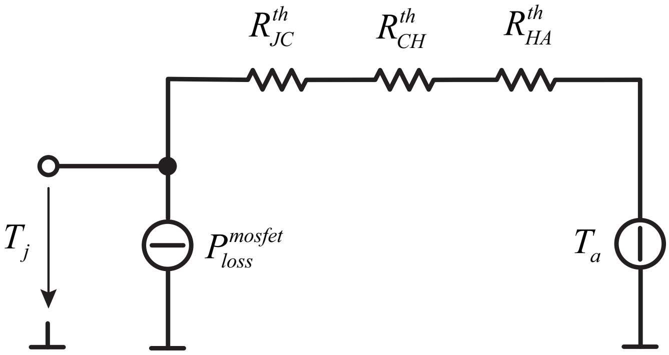


Figure 1.7: Steady state thermal model of the single MOSFET component.

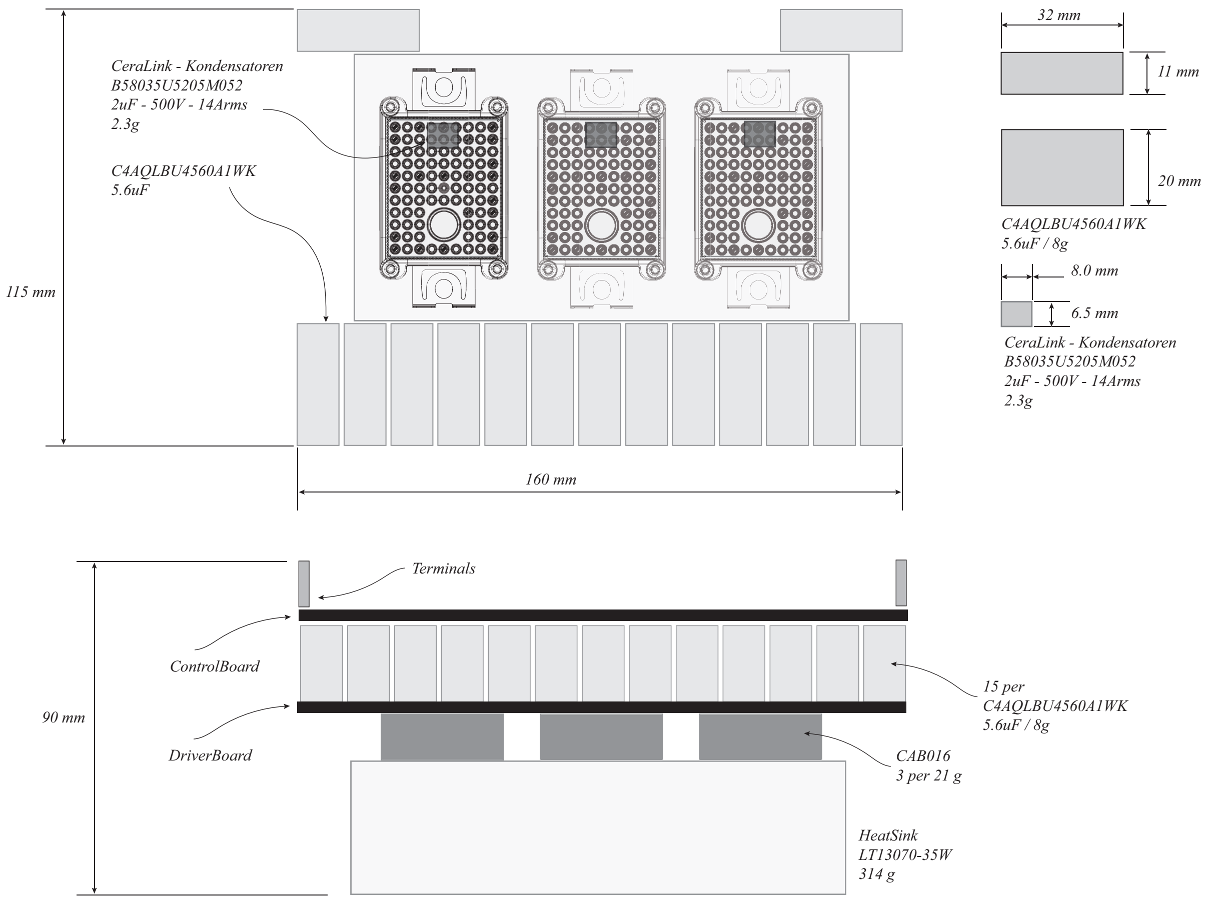
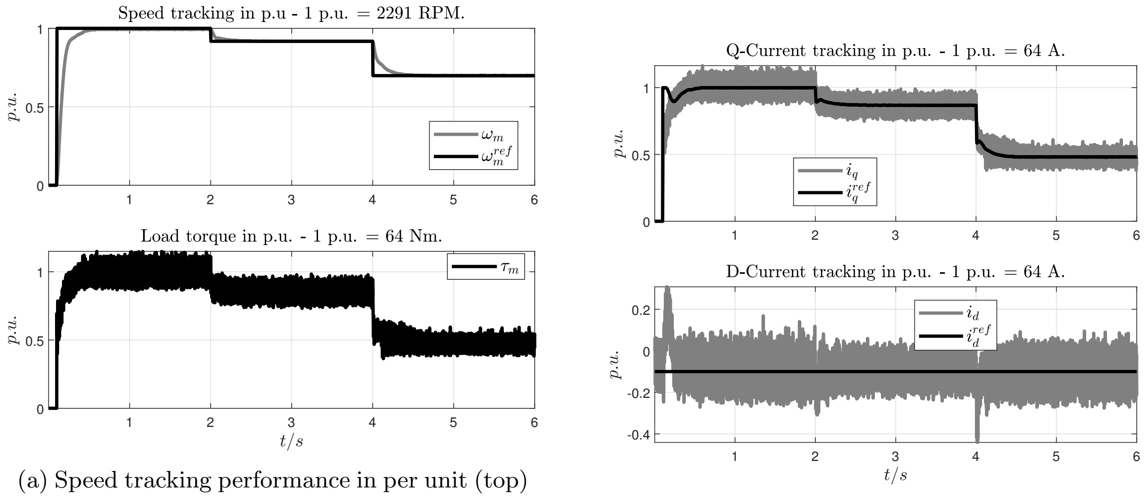


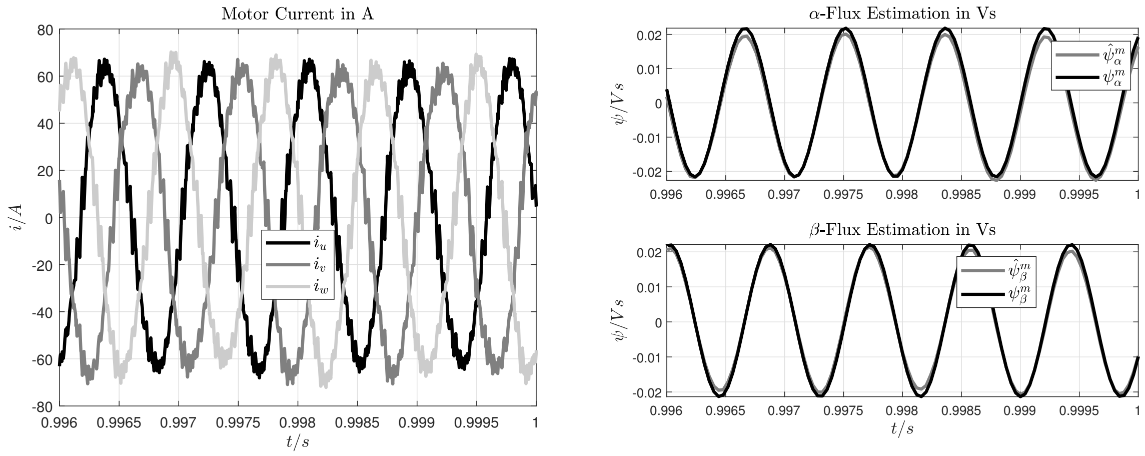
Figure 1.8: Preliminary design for electro/thermal simulation. In scale assembling layout: module+heatsink+dclink with weights.



(a) Speed tracking performance in per unit (top) - Electromagnetic torque in per unit (bottom). At time $t = 2\,\mathrm{s}$ the speed reference change from $2291\,\mathrm{min}^{-1}$ to $2102\,\mathrm{min}^{-1}$ and at time $t = 4\,\mathrm{s}$ the speed reference change from $2102\,\mathrm{min}^{-1}$ to $1600\,\mathrm{min}^{-1}$.

(b) Quadrature (top) and direct (bottom) currents tracking performance in per unit.

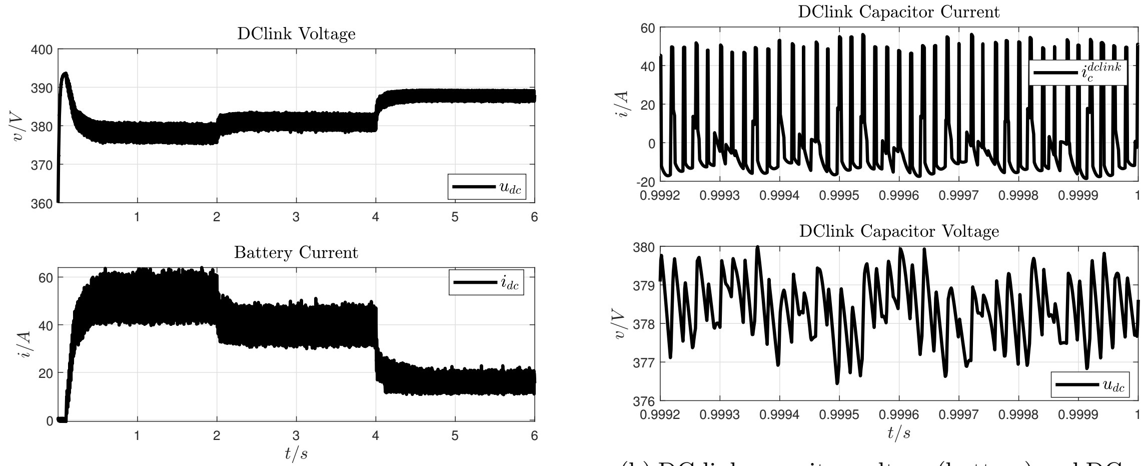
Figure 1.9: Electro-thermal simulation results at different loads.



(a) Inverter three phase currents in SI, at nominal torque.

(b) $\alpha$ and $\beta$ fluxes from PMSM and from Back EMF observer in SI, at nominal torque.

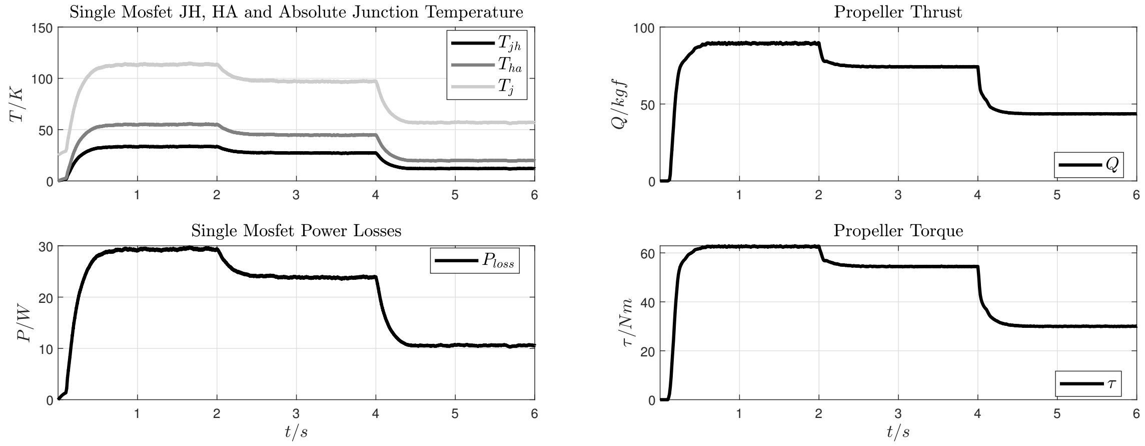
Figure 1.10: Electro-thermal simulation results at different loads.



(a) DC-link voltage (top) and battery current in SI (bottom).

(b) DC-link capacitor voltage (bottom) and DC-link capacitor current (top) - internal capacitor current is $I_c = 22.0\,\mathrm{A}$ RMS in SI, at nominal torque.

Figure 1.11: Electro-thermal simulation results at different loads.



(a) MOSFET power losses (bottom), MOSFET junction temperature (top-light grey).

(b) Propeller thrust (top) and torque (bottom) with FLUXER PRO 63x22.

Figure 1.12: Electro-thermal simulation results at different loads.

The CAB016 Module coupled with the Heatsink ALPHA-LT13070-35W can drive full load torque in continuos mode, as shown in the following thermal simulation (no thermal capacities are taken into account). The maximum MOSFET Junction temperature is always below than 125 °C under an ambient temperature of 25 °C:

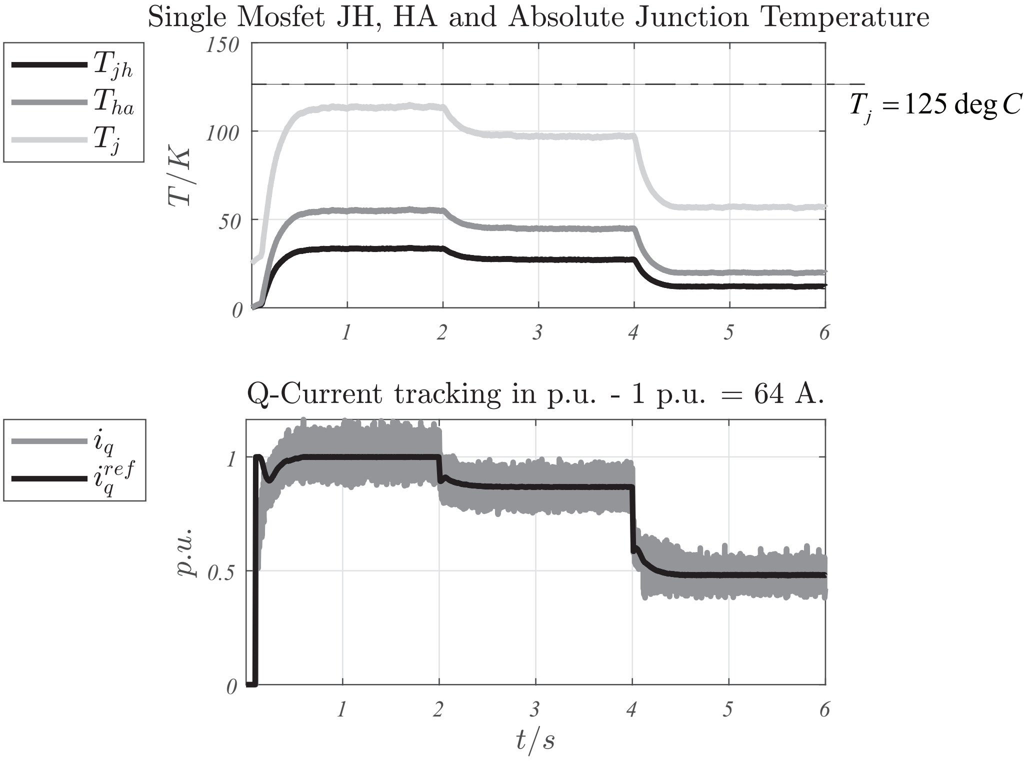


Figure 1.13: Simulation results: Motor current in per unit (bottom), MOSFET junction temperature (top-light grey).

The maximum continuative motor current which the CAB016 module is able to supply will be defined as the maximum motor current in which the Junction temperature of CAB016 module will rise up to $T_j^{max} = 125\,°\mathrm{C}$ when $T_{ambient} = 25\,°\mathrm{C}$. The evaluation will be done exploring by simulation three propeller working points as shown in the following table.

| **Rotor Speed** | **Torque** | **Thrust** |
|---|---|---|
| $2291\,\mathrm{min}^{-1}$ | 64 N m | 93 kg |
| $2102\,\mathrm{min}^{-1}$ | 54.5 N m | 75 kg |
| $1600\,\mathrm{min}^{-1}$ | 29.8 N m | 43 kg |

Table 1.9: Performance evaluation of the with CAB016 Module with M50C35-EEE-9KV Motor, and FLUXER-PRO-63x22 Propeller.

## 1.5 Dynamic thermal analysis

In the following the thermal capacity of the ALPHA-LT13070-35W heatsink will be taken into account. The following figure shows the equivalent model of the “sandwich” CAB016M12FM3 module and ALPHA-LT13070-35W heatsink. The equivalent MOSFET surface where power is generated is in some around half of the heatsink surface. This effect is modelized splitting in two identical parts the **equivalent** thermal capacity and thermal resistance.

Figure 1.14 shows the equivalent thermal circuit seen by every single MOSFET is analysed,

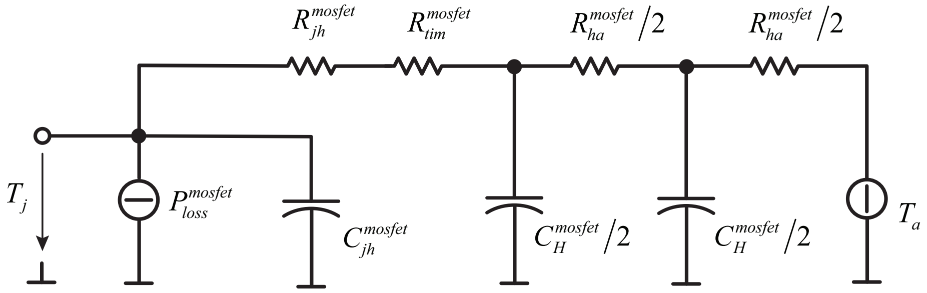


Figure 1.14: Equivalent thermal circuit (seen by a single MOSFET) where the thermal capacity of the heatsink has been included.

Considering Figure 1.15 as reference, the thermal impedances of the equivalent thermal network reported in Figure 1.14 can be derived.

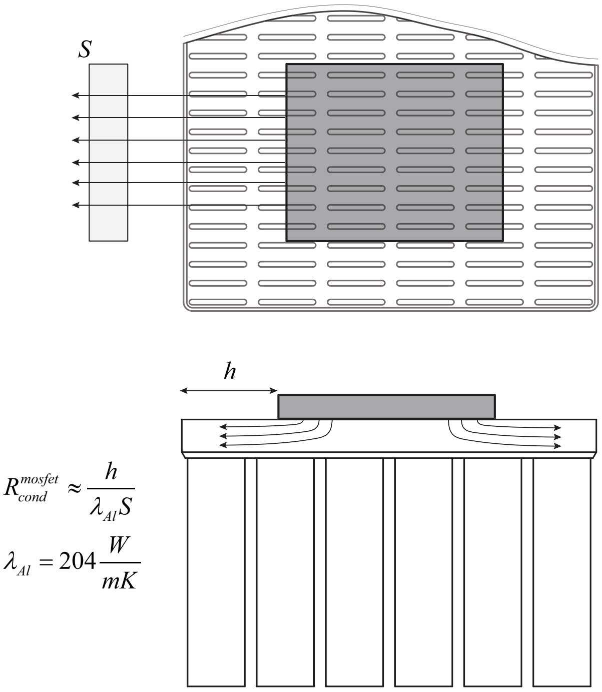

S
h
$R_{cond}^{mosfet} \approx \frac{h}{\lambda_{Al} S}$
$\lambda_{Al} = 204 \frac{W}{mK}$

Figure 1.15: CAB016M12FM3 case and ALPHA-LT13070-35W heatsink thermal exchange analysis.

Let

$$R_{ha} \approx 0.25\,^{\circ}\mathrm{C\,W^{-1}} \tag{1.5.1}$$

be the whole nominal (from data-sheet) thermal impedance of the heat-sink. The equivalent thermal impedance seen by one MOSFET can be derived assuming a reduced equivalent surface, seen by the MOSFET, as follows

$$\begin{aligned} R_{cond}^{mosfet} &\approx 0.35\,^{\circ}\mathrm{C\,W^{-1}} \\ R_{ha}^{mosfet} &\approx 6 \cdot R_{ha} + R_{cond}^{mosfet} \approx 2\,^{\circ}\mathrm{C\,W^{-1}} \end{aligned} \tag{1.5.2}$$

The total equivalent thermal impedance seen by an MOSFET from junction to ambient can be represented as follows

$$R_{ja}^{mosfet} \approx R_{jh}^{mosfet} + R_{tim}^{mosfet} + R_{ha}^{mosfet} \approx 3.3\,^{\circ}\mathrm{C\,W^{-1}} \tag{1.5.3}$$

The equivalent heat capacities of the mosfet and of the heat-sink are as follows

$$\begin{aligned} C_{H}^{mosfet} &= 47.8\,\mathrm{J\,K^{-1}} \\ C_{jh}^{mosfet} &= 0.025\,\mathrm{J\,K^{-1}} \end{aligned} \tag{1.5.4}$$

Dynamic thermal simulation of the CAB016M12FM3 module assuming $R_{JA}^{th} = 3.3\,\mathrm{K\,W^{-1}}$ (thermal resistance seen by the single individual MOSFET) and $T_a = 30\,^{\circ}\mathrm{C}$.

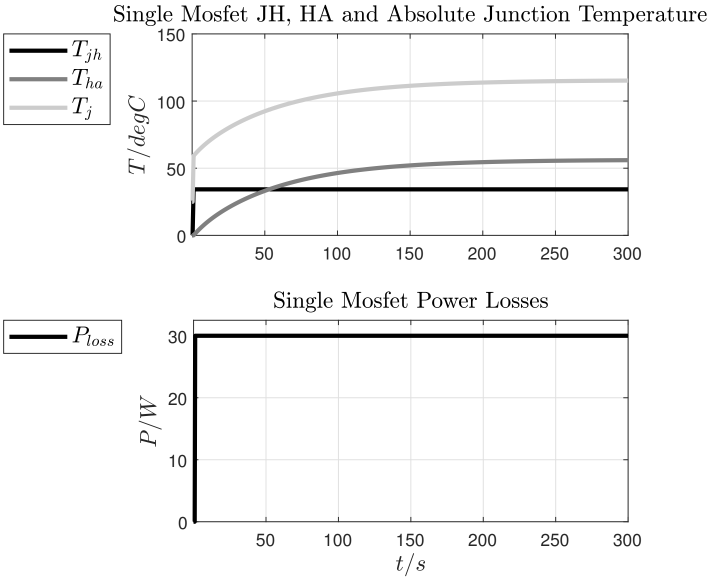


Figure 1.16: Thermal response of the ALPHA-LT13070-35W heatsink seen by a single MOSFET for the case of $P_{loss}^{mosfet} = 30\,\mathrm{W}$ - case for maximum continuative current of 45 A (RMS). Top Figures shows the Junction temperature (light grey) of a single MOSFET during time response.

# Chapter 2

# PMSM Control Architectures

## 2.1 Introduction

In the following sections different control approaches will be investigated. The traditional vector control based on rotor flux estimation is used as first approach in order to obtain a design layout for components selection and minimum required microcontroller performance. Additional control strategies like model predictive control will be investigated.

For the purpose of modelization of the *plant* as well as of the *control system* in *Simulink/Simscape* ambient, two different representations are adopted:

– system representation in continuous time domain with parameters in SI unit for plant models implementation;

– system representation in discrete time domain with parameters in per unit for control systems (controllers, state observers, model based controllers, etc.) implementation.

### 2.1.1 Nomenclature

Here, a list of used symbols:

– $s.r.f.$: stationary reference frame;

– $r.r.f.$: rotor reference frame;

– $r.r.f. - \omega_0$: rotating reference frame at $\omega_0$;

– $p$: number of pole pairs;

– $\omega_m$: mechanical rotor speed $\left[\mathrm{rad\,s^{-1}}\right]$;

– $\omega$ : electrical rotor speed $\left[\mathrm{rad\,s^{-1}}\right]$;

– $\vartheta$ : electrical rotor position $\left[\mathrm{rad}\right]$;

– $\vartheta_{emf}$ : electrical rotor position derived from rotor fluxes $\left[\mathrm{rad}\right]$;

– $\tau_m$ : electromagnetic torque $\left[\mathrm{N\,m}\right]$;

– $i_\alpha$ : direct current in $s.r.f.$ $\left[\mathrm{A}\right]$;

- $i_\beta$ : quadrature current in *s.r.f.* $\Big[\text{A}\Big]$;
- $i_d$ : direct current in *r.r.f.* $\Big[\text{A}\Big]$;
- $i_q$ : quadrature current in *r.r.f.* $\Big[\text{A}\Big]$;
- $\psi_d^s = \psi_d^r + i_d L_d$ : direct flux $\Big[\text{Wb}\Big]$;
- $\psi_q^s = \psi_q^r + i_q L_q$ : quadrature flux $\Big[\text{Wb}\Big]$;
- $\psi_\alpha^s = \psi_\alpha^r + i_\alpha L_s$ : $\Big[\text{Wb}\Big]$;
- $\psi_\beta^s = \psi_\beta^r + i_\beta L_s$ : $\Big[\text{Wb}\Big]$;
- $\psi^m$ : permanent magnet linkage flux $\Big[\text{Wb}\Big]$;
- $u_d$ : direct motor terminal voltage in *r.r.f.* $\Big[\text{V}\Big]$;
- $u_q$ : quadrature motor terminal voltage in *r.r.f.* $\Big[\text{V}\Big]$;
- $u_\alpha$ : direct motor terminal voltage in *s.r.f.* $\Big[\text{V}\Big]$;
- $u_\beta$ : quadrature motor terminal voltage in *s.r.f.* $\Big[\text{V}\Big]$;

### 2.1.2 Normalization and vector current control implementation.

The per unit representation of a dynamic system passes through the *normalization* process, which can be described as follows

- Reference quantities
  - $u_{bez}$ : peak phase voltage (at no-load, at nominal rotor speed $\omega_m^{nom}$) of the motor in V;
  - $i_{bez}$ : peak phase nominal current of the motor in A and can be derived from parameter $\tau_m^{nom}$ (Nominal Torque) as follows

$$i_{bez} = \frac{2}{3}\frac{\tau_m^{nom}}{p\,\psi_{bez}} \qquad \text{we consider } i_d = 0 \text{ control;} \tag{2.1.1}$$

  - $\omega_m^{nom}$ : nominal mechanical rotor speed of the motor in $\text{rad}\,\text{s}^{-1}$;
  - $\omega_{bez} = p\,\omega_m^{nom}$ : nominal electrical speed of the motor in $\text{rad}\,\text{s}^{-1}$;
  - $X_{bez} = u_{bez}/i_{bez}$ : reference electrical impedance of the motor in $\Omega$;
  - $L_{bez} = X_{bez}/\omega_{bez}$ : reference inductance of the motor in H;
  - $\psi_{bez} = u_{bez}/\omega_{bez}$ : reference flux of the motor in Wb.
- Per unit quantities
  - $R_s^{norm} = R_s/X_{bez}$ : per unit of the phase resistance;

- $L_s^{norm} = L_s/L_{bez}$ : per unit of $L_s$ inductance;
- $\omega^{norm} = p\omega_m/\omega_{bez}$ : per unit of the electrical speed;
- $i^{norm} = i/i_{bez}$ : per unit of $i$ current;
- $u^{norm} = u/u_{bez}$ : per unit of $u$ voltage;
- $\psi_m^{norm} = \psi^m/\psi_{bez} = 1$ : per unit of the linkage permanent magnet flux.

## 2.2 Vector Control based on Flux Observers

A preliminary control architecture, based on direct and quadrature current control obtained via Clarke and Park transformation is assumed. The rotor position, needed for Park transformation is estimated by Luenberger observer. The equations of the back EMF model are used to estimated the stator linkage fluxes respect to the stationary reference frame (obtaining the components $\hat{\psi}_\alpha(t)$ and $\hat{\psi}_\beta(t)$). From the flux components $\hat{\psi}_\alpha(t)$ and $\hat{\psi}_\beta(t)$ the rotor position is calculated via atan2() function, obtaining the estimated rotor phase $\hat{\vartheta}_{bemf}(t)$. The rotor phase $\hat{\vartheta}_{bemf}(t)$ is additionally filtered and time derivate using the double integrator observer. In the following control schematic layouts will be presented.

The current vector control can be written as follows:

$$\begin{cases} i_q^{ref} = k_p^w\Big(\omega^{ref} - \omega\Big) + i_q^i \\ \dfrac{d}{dt}i_q^i = k_i^w\Big(\omega^{ref} - \omega\Big) \\ u_d^{ctrl} = k_p^i\Big(i_d^{ref} - i_d\Big) + u_d^i - \omega\hat{\psi}_q - \omega L_q i_q \\ \dfrac{d}{dt}u_d^i = k_i^i\Big(i_d^{ref} - i_d\Big) \\ u_q^{ctrl} = k_p^i\Big(i_q^{ref} - i_q\Big) + u_d^i + \omega\hat{\psi}_d - \omega L_d i_d \\ \dfrac{d}{dt}u_q^i = k_i^i\Big(i_q^{ref} - i_q\Big) \end{cases} \tag{2.2.1}$$

where for the sake of clarity equations are kept in continuous time domain and the superscript $\Big(\Big)^{norm}$ is omitted. Figure 2.1 shows the equivalent diagram representation comprehensive of the vector saturation and corresponding integrals anti-windup.

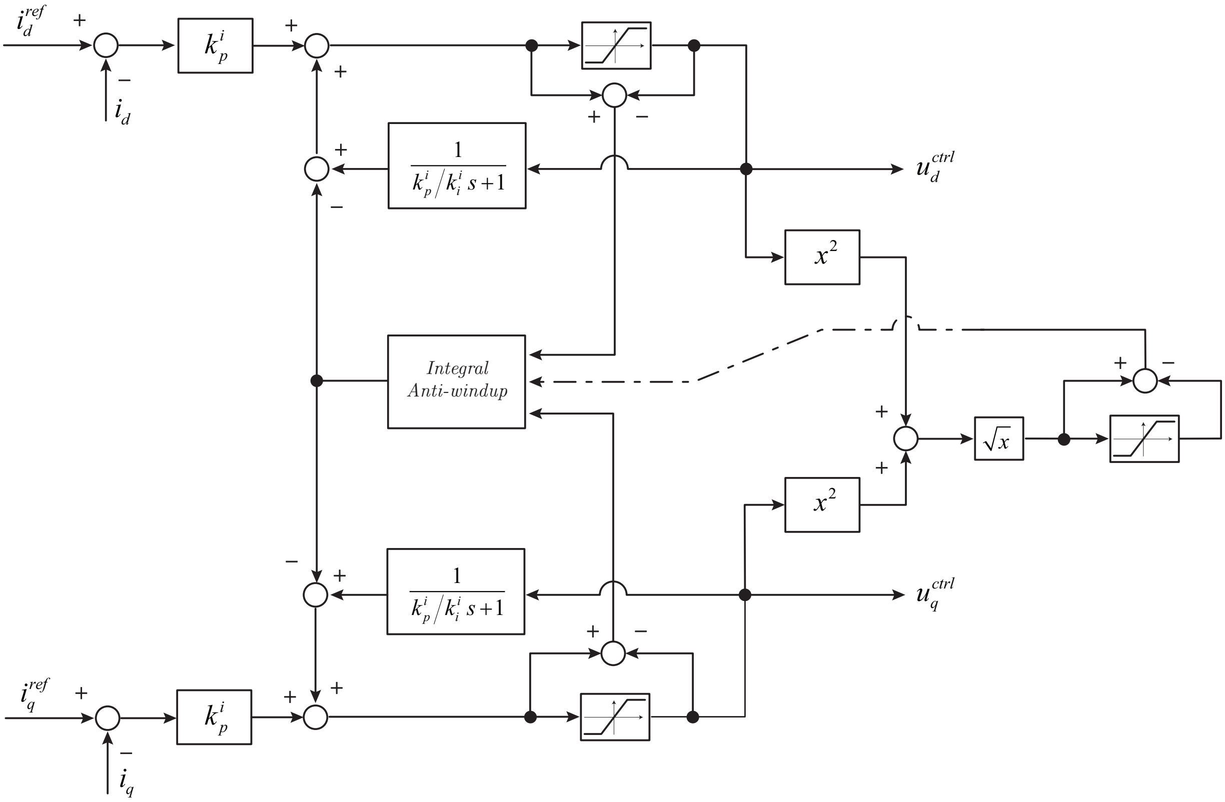


Figure 2.1: Vector PI with components and magnitude saturation with anti-windup diagram.

### 2.2.1 Back EMF based Observer

Define

$$|\psi^m| = 1 \qquad \text{(in per unit parameter)} \tag{2.2.2}$$

$$\left(\psi_\alpha^r\right)^* = |\psi^m| \cos\vartheta = \psi^m \cos\vartheta \tag{2.2.3}$$

$$\left(\psi_\beta^r\right)^* = |\psi^m| \sin\vartheta = \psi^m \sin\vartheta \tag{2.2.4}$$

The flux state observer can be written as follows

$$\frac{d\hat{\psi}_\alpha^s}{dt} = -R_s i_\alpha + u_\alpha + k_\psi\Big(\psi^m \cos\hat{\vartheta} - \hat{\psi}_\alpha^r\Big) \tag{2.2.5}$$

$$\frac{d\hat{\psi}_\beta^s}{dt} = -R_s i_\beta + u_\beta + k_\psi\Big(\psi^m \sin\hat{\vartheta} - \hat{\psi}_\beta^r\Big) \tag{2.2.6}$$

where

$$\hat{\psi}_\alpha^r = \hat{\psi}_\alpha^s - L_s i_\alpha \tag{2.2.7}$$

$$\hat{\psi}_\beta^r = \hat{\psi}_\beta^s - L_s i_\beta \tag{2.2.8}$$

The mechanical state observer[*] can be written as follows

$$\frac{d\hat{\vartheta}}{dt} = \hat{\omega} + k_{\vartheta}\Big(\hat{\vartheta}_{emf} - \hat{\vartheta}\Big) \tag{2.2.9}$$

$$\frac{d\hat{\omega}}{dt} = k_{\omega}\Big(\hat{\vartheta}_{emf} - \hat{\vartheta}\Big) \tag{2.2.10}$$

where

$$\hat{\vartheta}_{emf} = \text{atan2}\left(\hat{\psi}^r_\beta,\, \hat{\psi}^r_\alpha\right) \tag{2.2.11}$$

See also the control layout diagram depicted in Figure 2.2.

For practical implementation the equations

$$\frac{d\hat{\psi}^s_\alpha}{dt} = -R_s i_\alpha + u_\alpha + k_\psi\Big(\psi^m \cos\hat{\vartheta} - \hat{\psi}^r_\alpha\Big) \tag{2.2.12}$$

$$\frac{d\hat{\psi}^s_\beta}{dt} = -R_s i_\beta + u_\beta + k_\psi\Big(\psi^m \sin\hat{\vartheta} - \hat{\psi}^r_\beta\Big) \tag{2.2.13}$$

are implemented as follows

$$\frac{d\hat{\psi}^s_\alpha}{dt} = -R_s i_\alpha + u_\alpha + k_\psi\Big(\psi^m \cos\hat{\vartheta} - \hat{\psi}^r_\alpha\Big) - k_d\,\hat{\psi}^s_\alpha \tag{2.2.14}$$

$$\frac{d\hat{\psi}^s_\beta}{dt} = -R_s i_\beta + u_\beta + k_\psi\Big(\psi^m \sin\hat{\vartheta} - \hat{\psi}^r_\beta\Big) - k_d\,\hat{\psi}^s_\beta \tag{2.2.15}$$

which converter the pure integration of Eqs. (2.2.12) - (2.2.13) into a pt1 low pass filter of Eqs. (2.2.14) - (2.2.15)

[*] see § 4.1

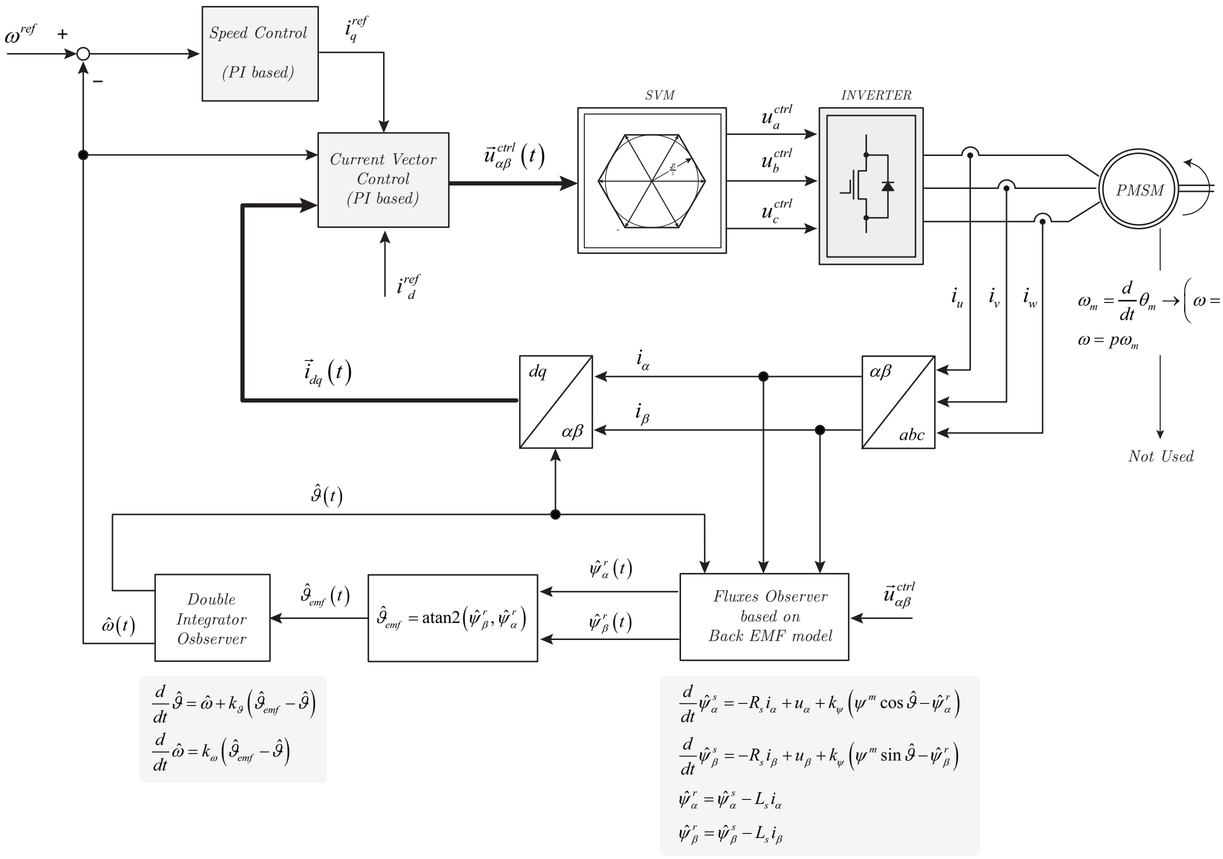

$\omega^{ref}$
+
−
Speed Control
(PI based)
$i_q^{ref}$
Current Vector Control (PI based)
$\vec{u}_{\alpha\beta}^{ctrl}(t)$
$i_d^{ref}$
SVM
INVERTER
$u_a^{ctrl}$
$u_b^{ctrl}$
$u_c^{ctrl}$
PMSM
$i_u$
$i_v$
$i_w$
$\omega_m = \frac{d}{dt}\theta_m \rightarrow \left(\omega = \frac{d}{dt}\theta\right)$
$\omega = p\omega_m$
Not Used
$\vec{i}_{dq}(t)$
dq
$\alpha\beta$
$i_\alpha$
$i_\beta$
$\alpha\beta$
abc
$\hat{\vartheta}(t)$
$\hat{\psi}_\alpha^r(t)$
$\hat{\psi}_\beta^r(t)$
$\hat{\vartheta}_{emf}(t)$
$\hat{\omega}(t)$
Double Integrator Osbserver
$\hat{\vartheta}_{emf} = \text{atan2}\left(\hat{\psi}_\beta^r, \hat{\psi}_\alpha^r\right)$
Fluxes Observer based on Back EMF model
$\vec{u}_{\alpha\beta}^{ctrl}$
$\frac{d}{dt}\hat{\vartheta} = \hat{\omega} + k_\vartheta\left(\hat{\vartheta}_{emf} - \hat{\vartheta}\right)$
$\frac{d}{dt}\hat{\omega} = k_\omega\left(\hat{\vartheta}_{emf} - \hat{\vartheta}\right)$
$\frac{d}{dt}\hat{\psi}_\alpha^s = -R_s i_\alpha + u_\alpha + k_\psi\left(\psi^m \cos\hat{\vartheta} - \hat{\psi}_\alpha^r\right)$
$\frac{d}{dt}\hat{\psi}_\beta^s = -R_s i_\beta + u_\beta + k_\psi\left(\psi^m \sin\hat{\vartheta} - \hat{\psi}_\beta^r\right)$
$\hat{\psi}_\alpha^r = \hat{\psi}_\alpha^s - L_s i_\alpha$
$\hat{\psi}_\beta^r = \hat{\psi}_\beta^s - L_s i_\beta$

Figure 2.2: State observer and vector control diagram.

# Chapter 3

# Firmware Implementation

## 3.1 Introduction

In the following chapter the layout of the firmware implementation will be shown. The concept of the firmware implementation can be exploited as follows:

- device and peripherals configuration;
    - timer for the main task configuration (`TIM2`) and synchronization with the timer for the three-phase PWM;
    - timer for the three-phase PWM with embedded dead-time configuration (`TIM1`) and synchronization with the ADCs (`TRGO(2)`);
    - configuration of digital I/O (`GPIO`) for *mosfet* drivers enabling and over-current interrupts;
    - configuration of CANBUS communication;
- global software architecture which can be exploited as follows;
    - a main state machine process (`global_state_machine_process()`) which coordinates the following sub-functions;
        - an automatic *cmd* word generation (`global_cmd_word_generator_process()`) which is triggered by the status of the *can-bus* inputs and by the status of the *DC-link* voltage power supply;
        - a sub-state-machine structure which properly selects the way of operation of the inverter control and triggers anomalous operating condition (`inverter_state_machine_process()`);
        - the inverter inner controls are managed by a specific function which enables the inner vector current loops, state observers and operates the conversion from the machine state variables ($uvw$) into the quadrature representation of the motor ($\alpha\beta$) and ($dq$) (`inverter_ctrl_process()`);
- additional ancillary and initialization functions have been implement as follows;
    - `peripherals_init()`, where DMA-ADC, IWD - independent watch-dog and DMA-PWM have been initialized;
    - `adc_current_calibration()`;
    - `global_state_machine_init()`;

  - `global_cmd_word_generator_init();`
  - `inverter_ctrl_init()`.

## 3.2 Device Configuration

In this section the peripherals configuration of the STM32F746 is described. Conceptually a main task timer (`TIM2`) is selected to generated a global interrupt which will be used for

- start the main software process (`global_state_machine_process()`);
- start and synchronize the PWM timer (`TIM1`).

Analog acquisition are automatically activated by the signal (`TRGO(2)`) regularly launched by the (`TIM1`) at every vertex of the PWM carry triangle waveform (this is activated by selecting `Repetition Counter = 0`). The overall timing of the (`TIM2`) and (`TIM1`) can be seen as depicted in Figure 3.2.

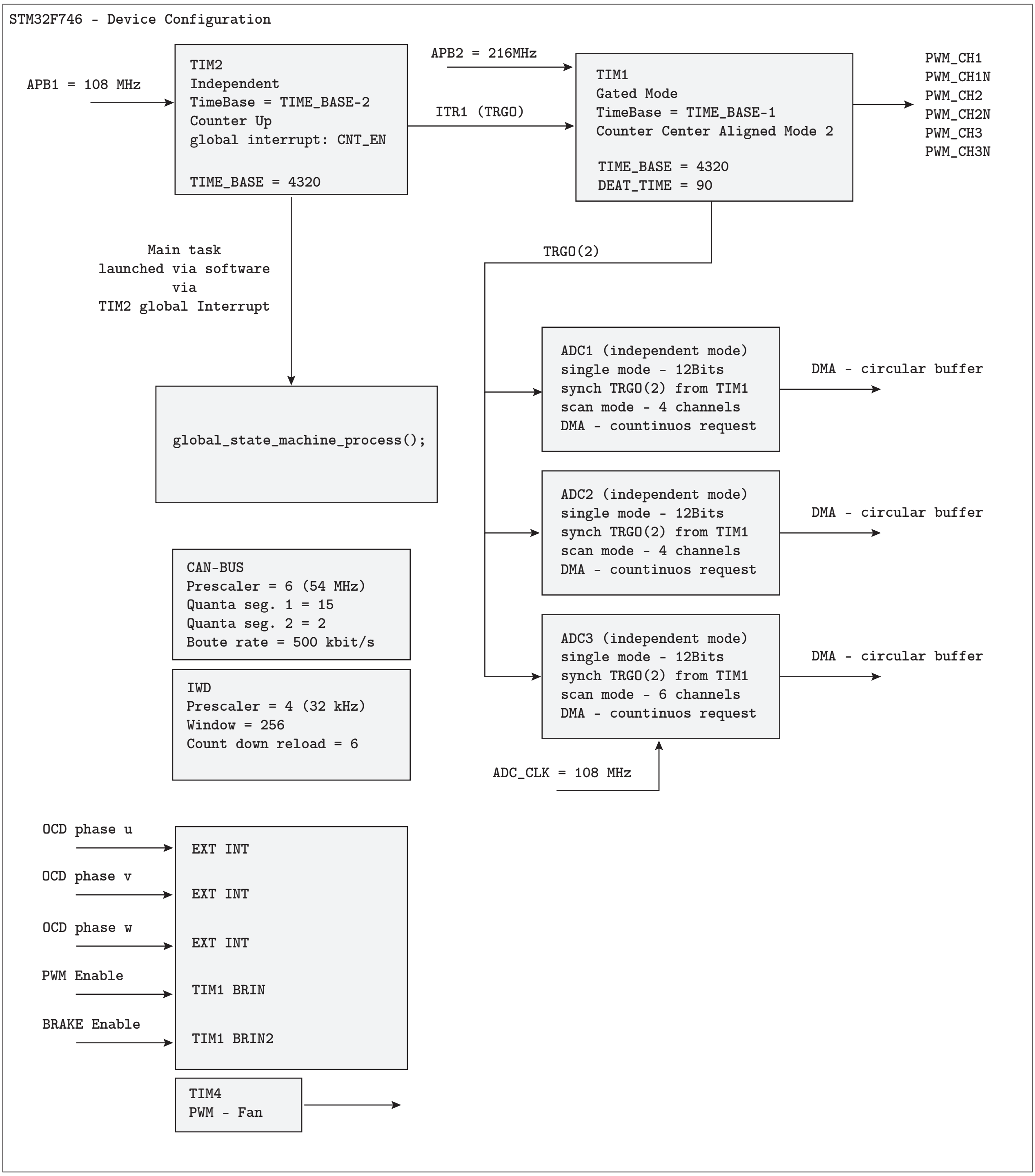


Figure 3.1: Device configuration overview.

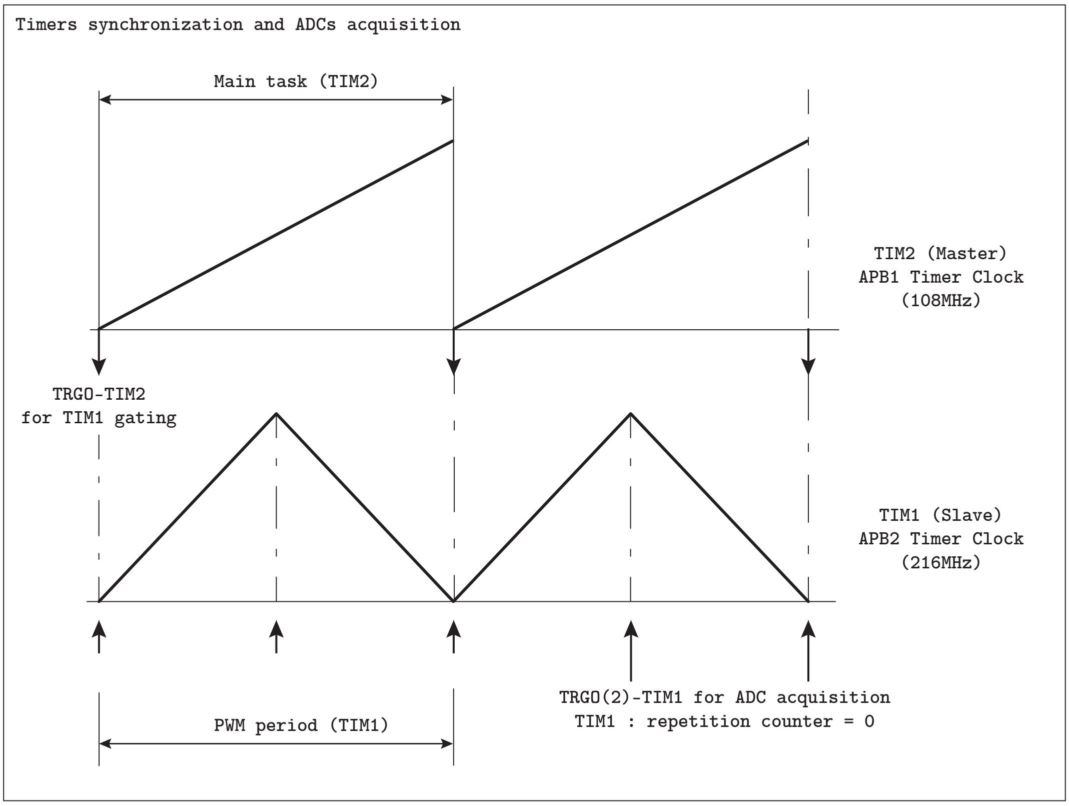


Figure 3.2: ADCs and PWM synchronization overview.

## 3.3 State Machines and control architecture

The global control application is based on two main state machine control

– `global_state_machine_process();`

– `inverter_state_machine_process();`

these states machines coordinates the status of a *global system* with the status of a *inner system*. The *inner system* manages the `inverter_control_process()`, see Figure 3.3;

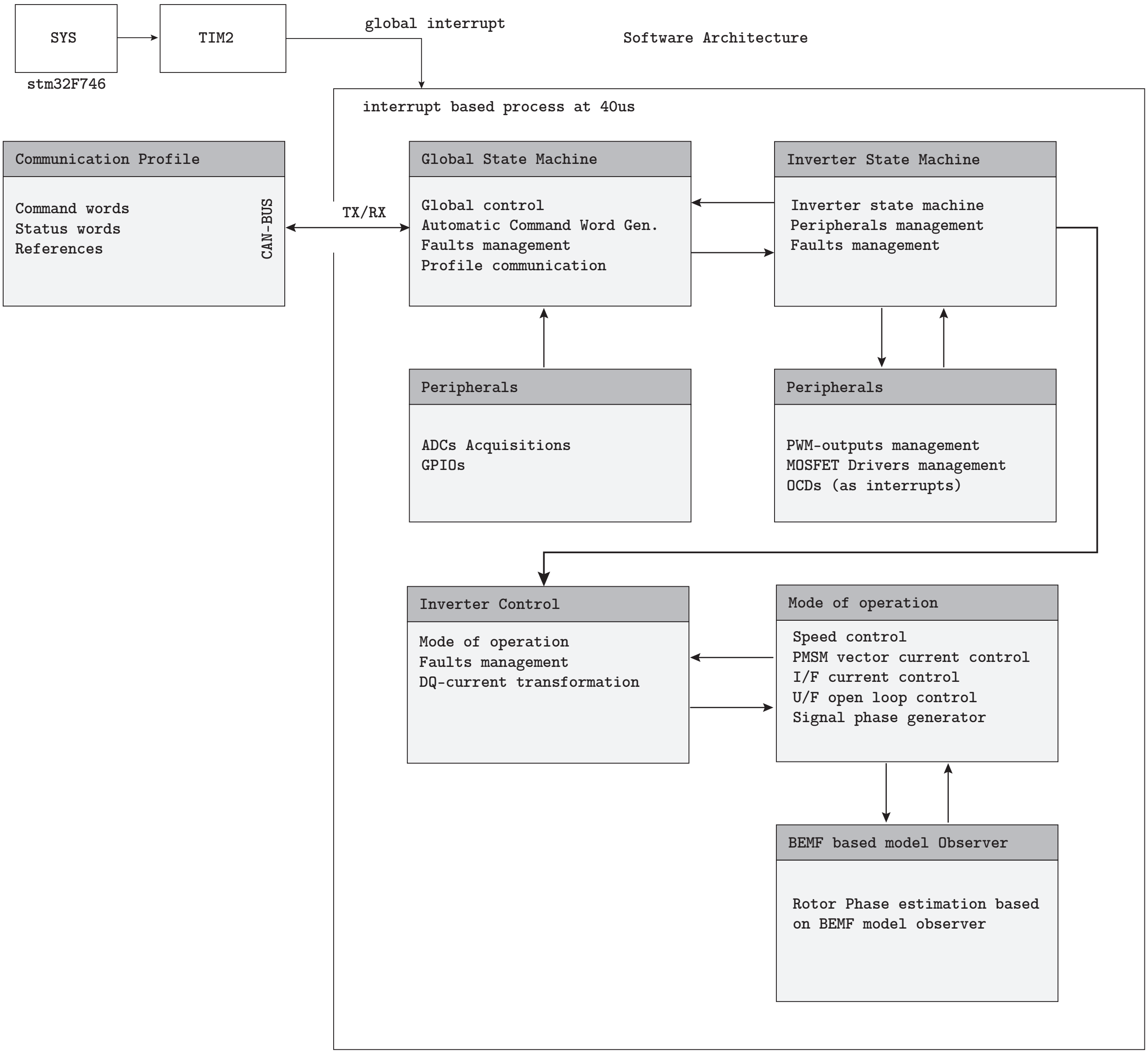
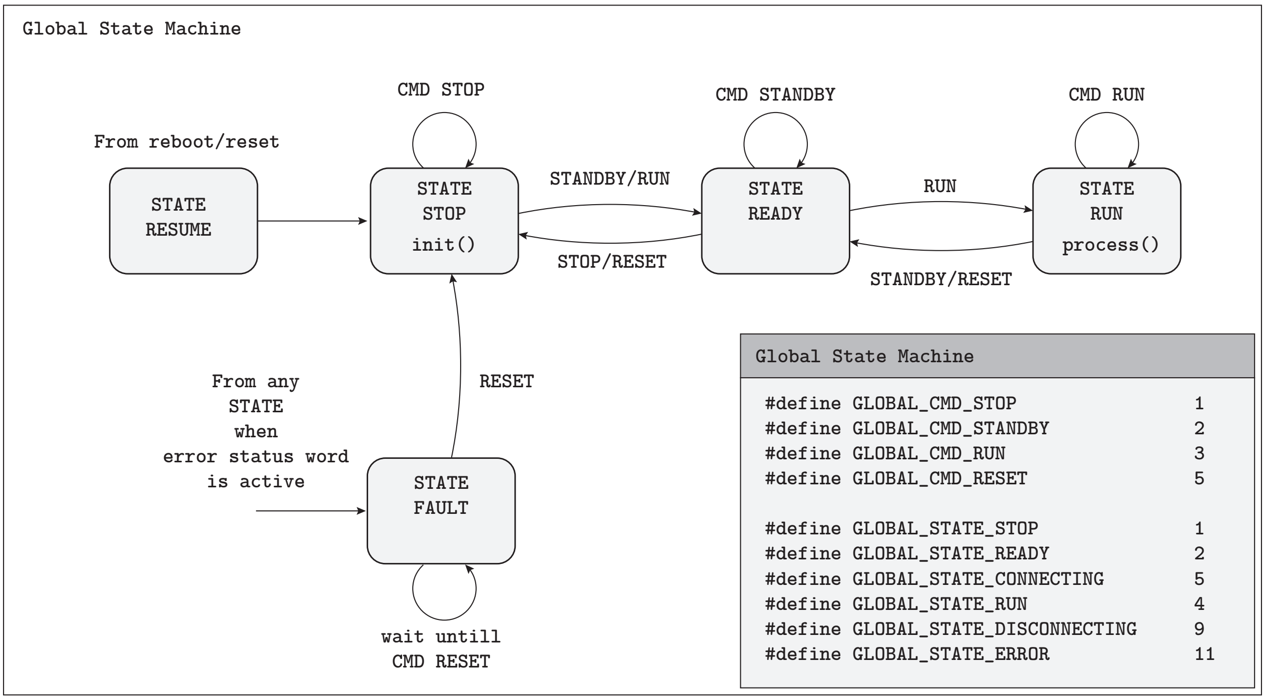


Figure 3.3: Global state machine and auxiliary functions architecture overview.

Figure 3.4 shows a qualitative representation the state machine flow of the `global_state_machine_process()`.

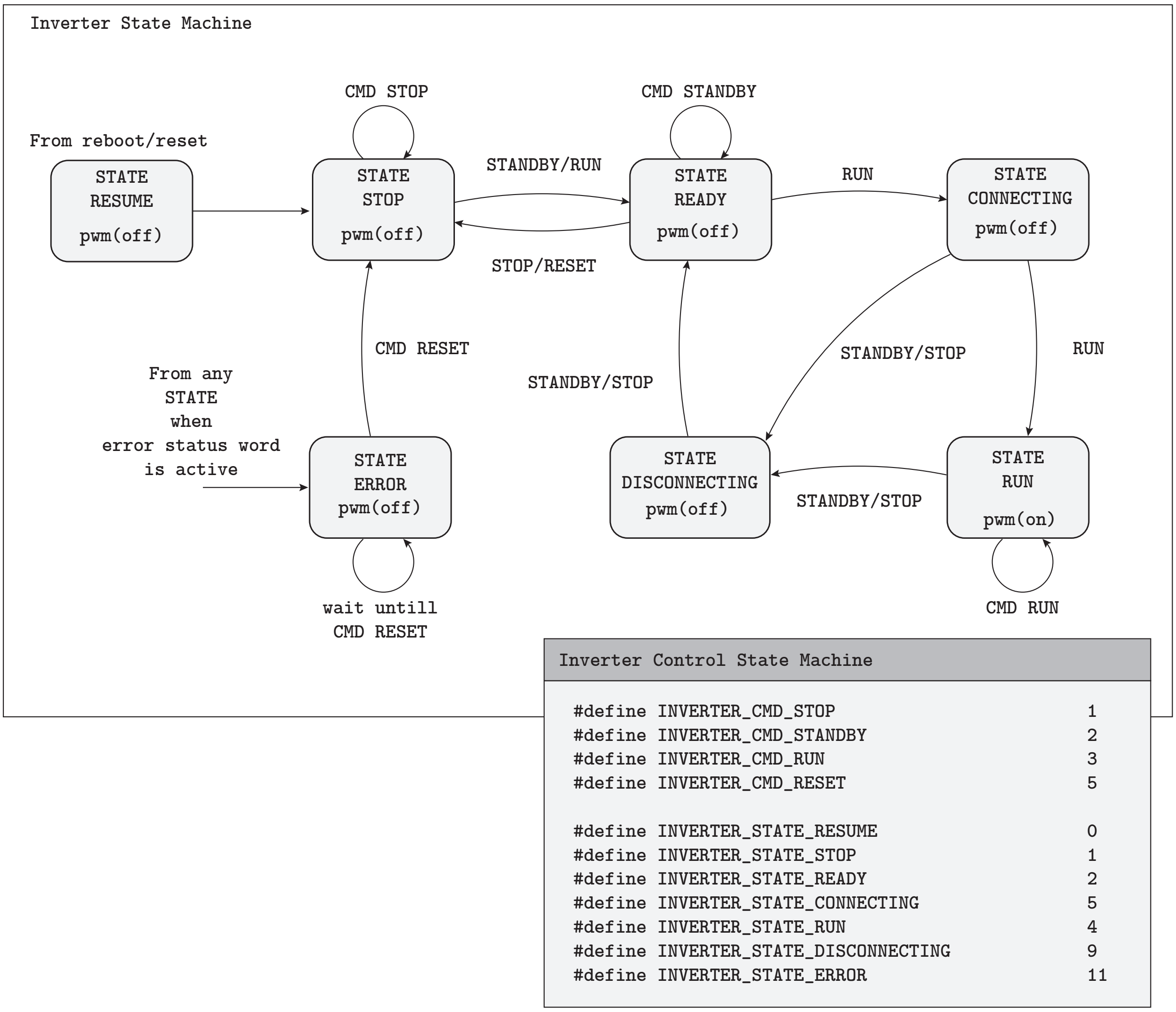


```
#define GLOBAL_CMD_STOP                1
#define GLOBAL_CMD_STANDBY             2
#define GLOBAL_CMD_RUN                 3
#define GLOBAL_CMD_RESET               5

#define GLOBAL_STATE_STOP              1
#define GLOBAL_STATE_READY             2
#define GLOBAL_STATE_CONNECTING        5
#define GLOBAL_STATE_RUN               4
#define GLOBAL_STATE_DISCONNECTING     9
#define GLOBAL_STATE_ERROR             11
```

Figure 3.4: Global state machine overview.

Figure 3.5 shows a qualitative representation the state machine flow of the `inverter_state_machine_process()`.



```
#define INVERTER_CMD_STOP                               1
#define INVERTER_CMD_STANDBY                            2
#define INVERTER_CMD_RUN                                3
#define INVERTER_CMD_RESET                              5

#define INVERTER_STATE_RESUME                           0
#define INVERTER_STATE_STOP                             1
#define INVERTER_STATE_READY                            2
#define INVERTER_STATE_CONNECTING                       5
#define INVERTER_STATE_RUN                              4
#define INVERTER_STATE_DISCONNECTING                    9
#define INVERTER_STATE_ERROR                            11
```

Figure 3.5: Inverter state machine overview.

An additional software control block concerns the automatic generation of the command word for the
`global_state_machine_process()`. Figure 3.6 shows the implemented solution, which can be summarized as follows:

- The DC-link voltage measure is filtered and sent to a comparator for unconstrained `STANDBY/STOP` internal commands;

- The duty reference word received from can bus is filtered and sent to a comparator for constrained `RUN/STANDBY` internal commands;

Figure 3.7 shows the internal state machine of the `global_state_machine_process()`.

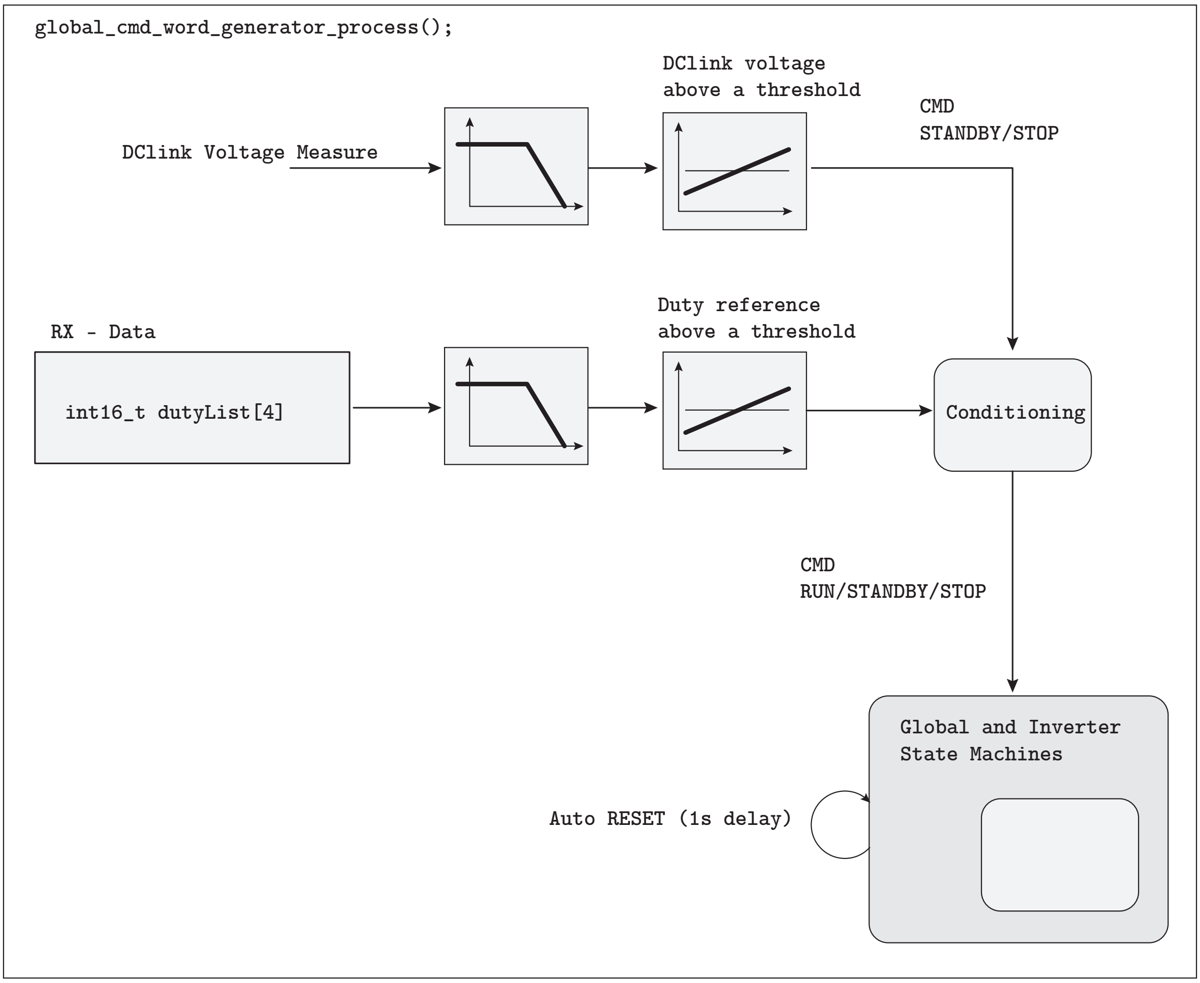
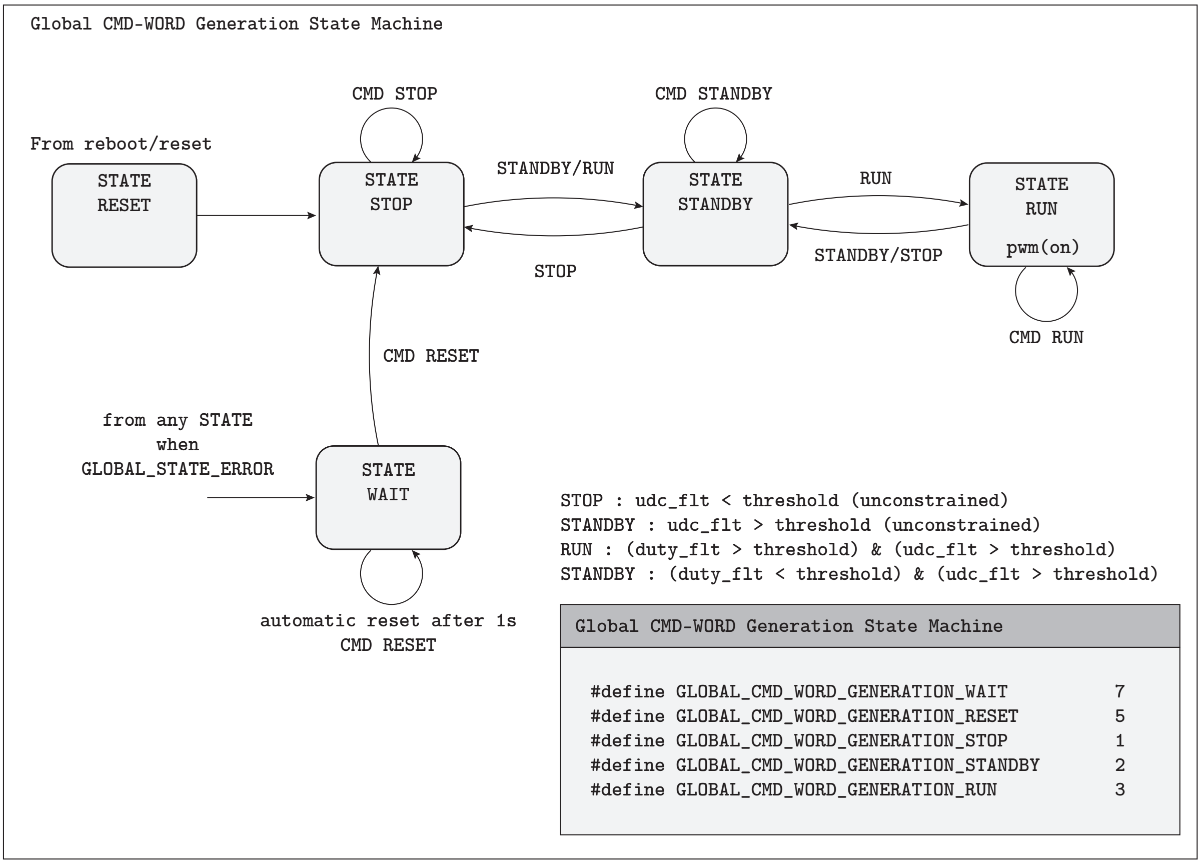


Figure 3.6: Automatic command word generation overview.



```
STOP : udc_flt < threshold (unconstrained)
STANDBY : udc_flt > threshold (unconstrained)
RUN : (duty_flt > threshold) & (udc_flt > threshold)
STANDBY : (duty_flt < threshold) & (udc_flt > threshold)
```

Global CMD-WORD Generation State Machine

```
#define GLOBAL_CMD_WORD_GENERATION_WAIT          7
#define GLOBAL_CMD_WORD_GENERATION_RESET         5
#define GLOBAL_CMD_WORD_GENERATION_STOP          1
#define GLOBAL_CMD_WORD_GENERATION_STANDBY       2
#define GLOBAL_CMD_WORD_GENERATION_RUN           3
```

Figure 3.7: Automatic command word generation state machine.

# Chapter 4

# Appendix A - Control Engineering

## 4.1 State Observer

In the pole placement approach the design of a control system assumes that all state variable are available for feedback. In practice, however, not all state variables are available for feedback. That, the need to estimate unavailable state variables using the theory of the *state observer* or *Luenberger observer.*

**Luenberger observer**. A state observer estimates the state variables based on the measurements of the output and control variables. *State observers can be designed if and only if the observability condition is satisfied.* Considering the plant defined by

$$\begin{aligned} \dot{\vec{x}} &= \tilde{\mathbf{A}}\vec{x} + \tilde{\mathbf{B}}u \\ y &= \mathbf{C}\vec{x} \end{aligned} \tag{4.1.1}$$

The observer is a subsystem to reconstruct the state vector of the plant. The mathematical model of the observer is basically the same as that of the plant, except that we include an additional term that includes the estimation error to compensate for inaccuracies in matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ and the lack of the initial error.The estimation error or the observation error is the difference between the measured output and the estimated output. The initial error is the difference between the initial state and the initial estimated state. Thus, we define the mathematical model of the observer to be

$$\begin{aligned} \dot{\hat{\vec{x}}} &= \tilde{\mathbf{A}}\hat{\vec{x}} + \tilde{\mathbf{B}}\vec{u} + \mathbf{L}(y - \mathbf{C}\hat{\vec{x}}) \\ &= (\tilde{\mathbf{A}} - \mathbf{LC})\hat{\vec{x}} + \tilde{\mathbf{B}}u + \mathbf{L}y \end{aligned} \tag{4.1.2}$$

where $\hat{\vec{x}}$ is the estimated state and $\mathbf{C}\hat{\vec{x}}$ is the estimated output. *The inputs of the observer are the output $\vec{y}$ and the control input $\vec{u}$.* Matrix $\mathbf{L}$ which is called the observer gain matrix, is a weighting matrix to the correction term involving the difference between the measured output $\vec{y}$ and the estimated output $\mathbf{C}\hat{\vec{x}}$. this term continuously corrects the model output and improves the performance of the observer. Figure 4.1 shows the block diagram of the system and the full-order state observer. The order of the state observer that will be discussed here is the same as that of the plant.

To obtain the observer error equation, let us subtract Eq. (4.1.2) from Eq. (4.1.1):

$$\begin{aligned} \dot{\vec{x}} - \dot{\hat{\vec{x}}} &= \tilde{\mathbf{A}}\vec{x} - \tilde{\mathbf{A}}\hat{\vec{x}} - \mathbf{L}(\mathbf{C}\vec{x} - \mathbf{C}\hat{\vec{x}}) \\ &= (\mathbf{A} - \mathbf{LC})(\vec{x} - \hat{\vec{x}}) \end{aligned} \tag{4.1.3}$$

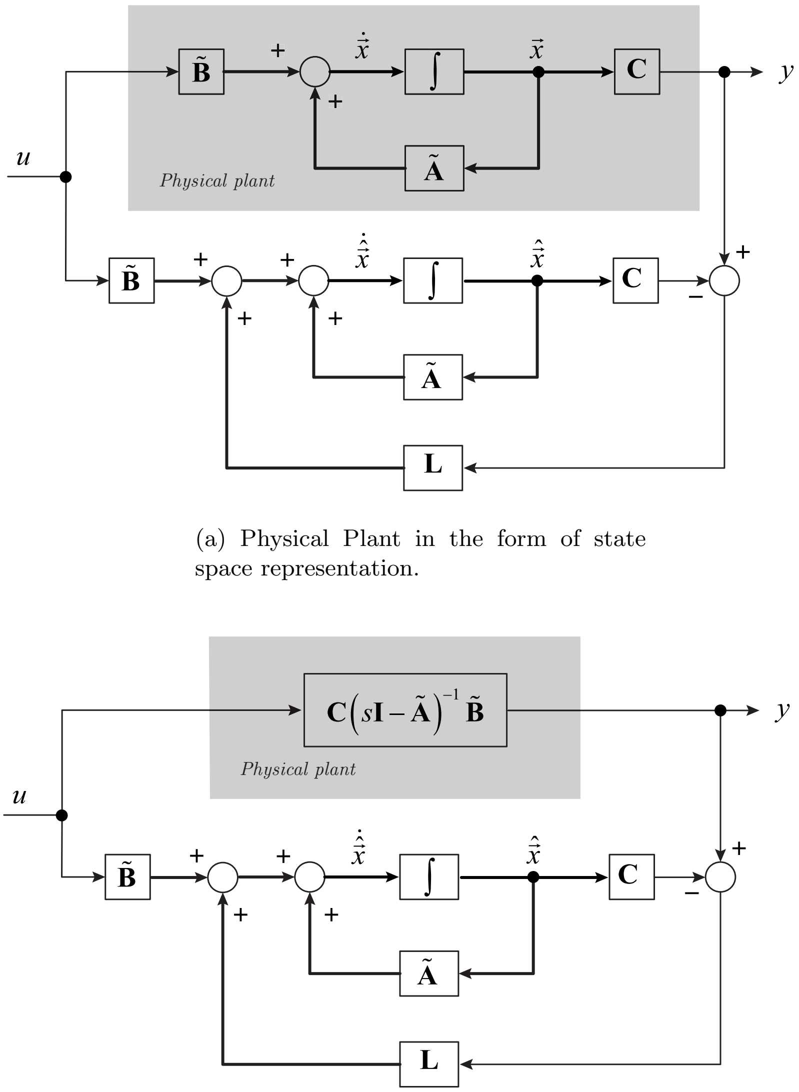
Define the difference between $\vec{x}$ and $\hat{\vec{x}}$ as the error vector $\vec{e}$, or

$$\vec{e} = \vec{x} - \hat{\vec{x}} \tag{4.1.4}$$

the equation (4.1.3) becomes

$$\dot{\vec{e}} = (\tilde{\mathbf{A}} - \mathbf{LC})\vec{e} \tag{4.1.5}$$

From Eq. (4.1.5), we see that the dynamic behaviour of the error vector is determined by the eigenvalues of matrix $(\tilde{\mathbf{A}} - \mathbf{LC})$. If matrix $(\tilde{\mathbf{A}} - \mathbf{LC})$ is a stable matrix, the error vector will converge to zero for any initial error vector $\vec{e}(0)$. That is, $\hat{\vec{x}}(t)$ will converge to $\vec{x}(t)$ regardless of the value of $\vec{x}(0)$ and $\hat{\vec{x}}(0)$. If the eigenvalues of the matrix $(\tilde{\mathbf{A}} - \mathbf{LC})$ are chosen in such a way that the dynamic behaviour of the error vector is asymptotically stable and is adequately fast, then any error vector will tend to zero (the origin) with an adequate speed. If the plant is completely observable, then it can be proved that it is possible to choose matrix $\mathbf{L}$ such that $(\tilde{\mathbf{A}} - \mathbf{LC})$ has arbitrarily desired eigenvalues. That is, the observer gain matrix $\mathbf{L}$ can be determined to yield the desired matrix $(\tilde{\mathbf{A}} - \mathbf{LC})$.



(a) Physical Plant in the form of state space representation.

(b) Physical Plant in the form of transfer function.

Figure 4.1: Full-order state observer.

Chapter 5

# Appendix B - User-Manual

## 5.1 Introduction

In the following a short introduction to the *how to use* the firmware developed for the flying basket project. This document can be useful in case of testing, debugging and firmware code improvement.

## 5.2 Firmware configuration

The firmware source code is basically fully allocated into the folder:

– `Application\`

where the following configuration file is available.

```
Application\Inc\fb_application_config.h

#define CLOCK_APB2  216000000

/* begin - Configuration application */

/* MCI-EPL Inverter */
#define MAD_MOTOR
#define MCI_EPL_INVERTER
#define PMSM_LOAD

/* end - Configuration devices and application */

/* begin - Configuration switching frequency */

/* PWM Base counter - set according to APB2 clock frequency
*  25kHz switching frequency */
#define PWM_TB  4320  /* PWM switching frequency and main task */
#define DEAD_TIME  170  /* dead time */

/* end - Configuration switching frequency */
```

In `fb_application_config.h` is possible to define the switching frequency of the driver as well as the dead-time, by a proper setting of `TB_PWM` and `DEAD_TIME`. Additional configuration regarding the application are available.

**Example 5.2.1.** In case we want to set a switching frequency of $f_{pwm} = 25\,\mathrm{kHz}$ the term `TB_PWM` shall be set as $f_{clock}/f_{pwm}/2$ where $f_{clock} = 216\,\mathrm{MHz}$.

## 5.3 Debug Mode

The inverter control mode can be preset as follows

- **U/f mode control** - inverter is controlled in open loop. The speed observer is still enable for estimating the phase and speed of the load for debugging. The involved parameters are as follows

  - `param.signal_generator_freq_bez;`
  - `param.signal_generator_u_ref_out;`

  an example of setting is

  - `param.signal_generator_freq_bez = 50;`
  - `param.signal_generator_u_ref_out = 0.5;`

- **I/f mode control** - inverter is controlled in current closed loop via a vector current control. The speed observer is still enable for estimating the phase and speed of the load for debugging. The involved parameters are as follows

  - `param.signal_generator_freq_bez;`
  - `param.id_ext_ref;`
  - `param.iq_ext_ref;`

  an example of setting is

  - `param.signal_generator_freq_bez = 50;`
  - `param.id_ext_ref = 0.1;`
  - `param.iq_ext_ref = 0.0;`

- **Speed mode control** - the load can be an RL as well as a PMSM and the load is controlled in speed mode, ny the use of a current vector control and speed observer. The involved parameters are as follows

  - `param.omega_ref_pu;`

  the limit of the applied torque can be adjusted by the following parameters

  - `param.torque_lim_top;`
  - `param.torque_lim_bottom;`

- **Double Pulse Mode** - in double pulse mode the control generate a modified PWM output for n cycles. The involved parameters are as follows

  - `param.double_pulse_m_duty;` - counter reference channels 1 or 2 or 3 as per leg identification parameter.
  - `param.double_pulse_n_duty;` - counter reference channel 5.
  - `param.double_pulse_leg;` - leg identification.
  - `param.double_pulse_test_counter;` - number of active cycles, one as per default.

  example of setting

  - param.double_pulse_m_duty = 0.85;
  - param.double_pulse_n_duty = 0.35;
  - param.double_pulse_leg = 1;
  - param.double_pulse_test_counter = 1;

An exhaustive list of parameters useful for debugging is as follows

- param.global_state;
- param.global_cmd;
- param.inverter_ctrl_mode;
- param.inverter_ctrl_state;
- param.inverter_current_u;
- param.inverter_current_v;
- param.inverter_current_w;
- param.dclink_voltage;
- param.omega_ref_pu;
- param.omega_hat_pu;
- param.inverter_current_d;
- param.inverter_current_q;
- param.id_ref_fb;
- param.iq_ref_fb;
- param.inverter_overcurrent_fault;
- param.inverter_t_pwm;

Outside of the debug mode the inverter control is set in torque mode.

## 5.4 Power-up

Here an example of power-up using e.g. the double pulse test.

In debug mode -¿ Live Expression select the following parameters:

- param.global_state;
- param.global_cmd;
- param.inverter_ctrl_mode;
- param.inverter_ctrl_state;
- param.dclink_voltage;

- `param.double_pulse_m_duty = 0.85;`
- `param.double_pulse_n_duty = 0.35;`
- `param.double_pulse_leg = 1;`
- `param.double_pulse_test_counter = 1;`

The test can be performed simply applying the `param.global_state;` from 5 (reset) to 3 (run).

**Example 5.4.1** (Double Pulse Test)**.** Here some results of the double pulse test, according to the configuration of 5.1

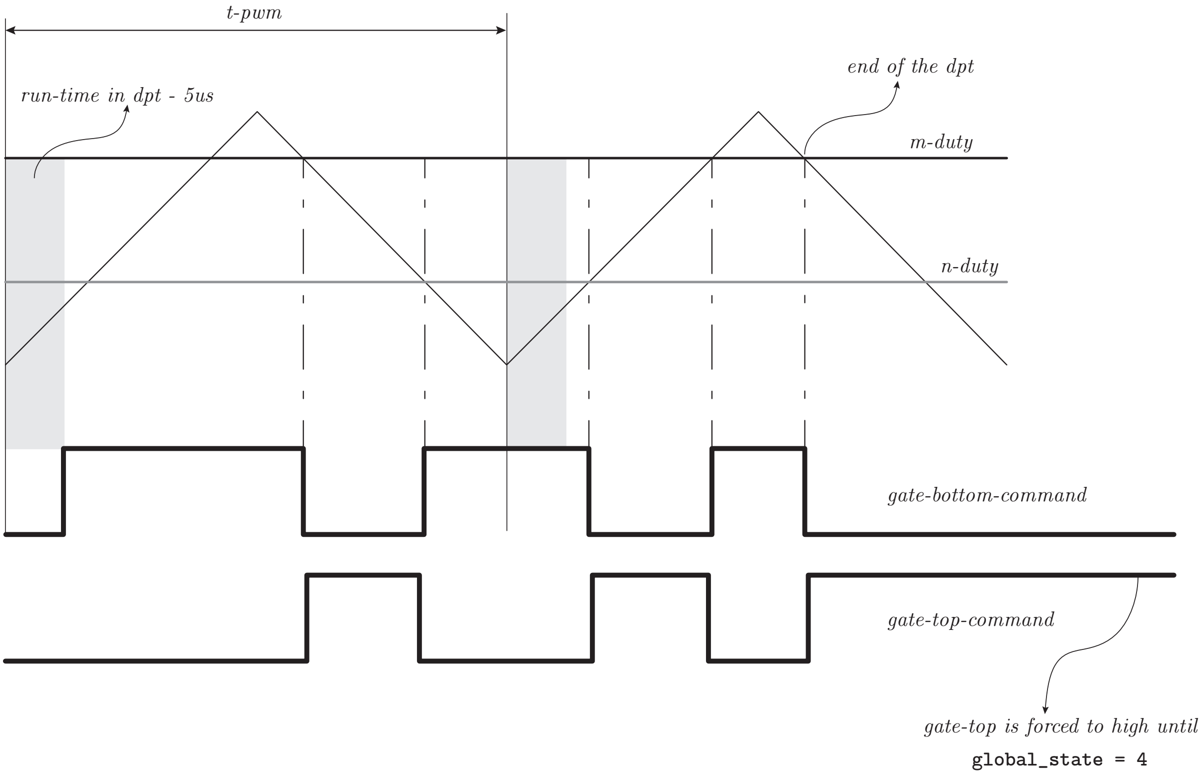
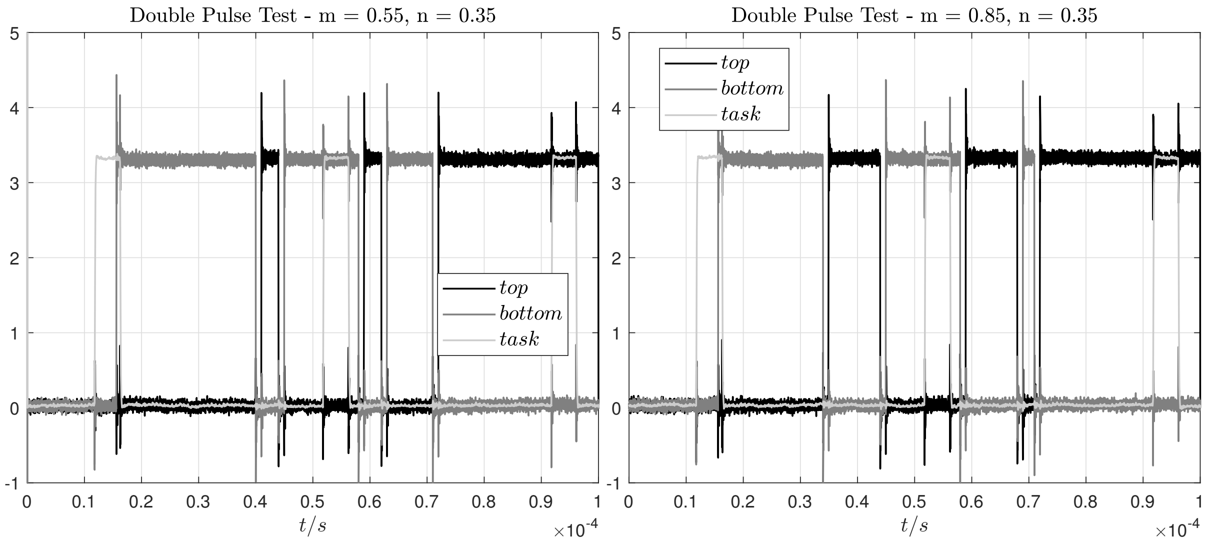


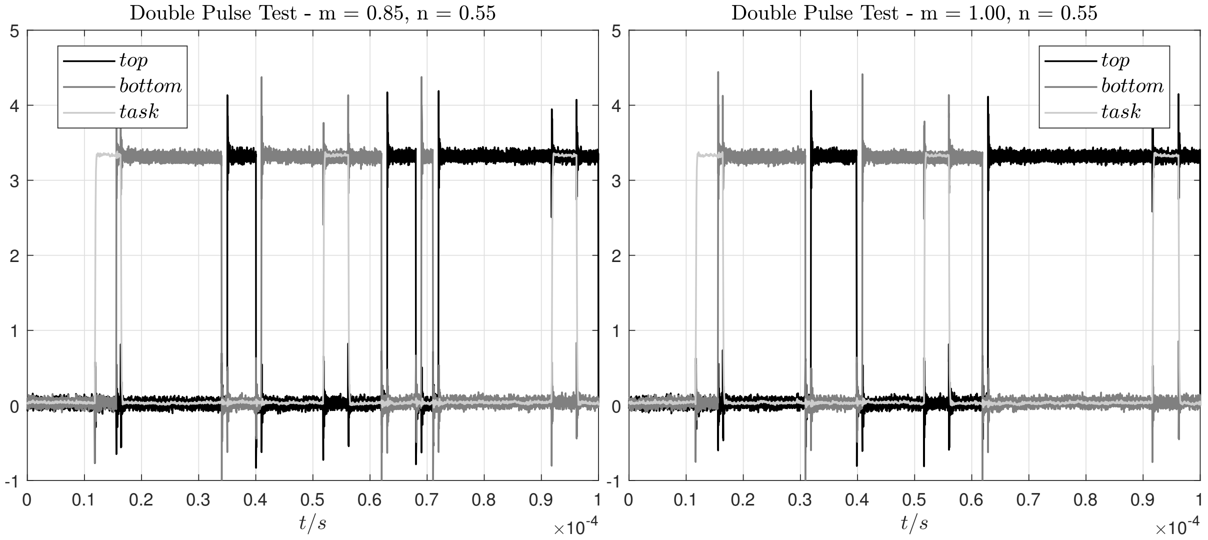
Figure 5.1: Description how the double pulse is built.



(a) Double pulse test for the case $m = 0.55$ and $n = 0.35$.

(b) Double pulse test for the case $m = 0.85$ and $n = 0.35$.

Figure 5.2: Double pulse tests gate commands measure.



(a) Double pulse test for the case $m = 0.85$ and $n = 0.55$.

(b) Double pulse test for the case $m = 1.00$ and $n = 0.55$.

Figure 5.3: Double pulse tests gate commands measure.

## 5.5 List of Parameters

| `param.inverter_current_max` |
|---|
| This parameter defines the maximum (readable) current of the inverter. It correlates the end-scale voltage of the ADC channel $u_{adc}^{max} = 3.3\,\mathrm{V}$ with the end scale of the corresponding sensor in SI $i_{inverter}^{max} = [\mathrm{A}]$, which in this case will be end-scale of the current measure in Ampere. |

| `param.inverter_current_nom` |
|---|
| This parameter defines the nominal current of the inverter. The nominal value is the value which is used for normalization. When the parameter (`param.inverter_current_u`) results as 1, it corresponds to the nominal value. |

| `param.inverter_current_scale` |
|---|
| This parameter defines the conversion factor between the ADC channel measure (in `uint32_t`) into the per unit float normalized current representation. This quantity is automatically calculated during initialization process. |

| `param.inverter_current_u_offset` |
|---|
| This parameter identifies the offset of the ADC channel (for bipolar signals). This variable is automatically generated during calibration process and used for the representation of the normalized current. |

| `param.inverter_current_v_offset` |
|---|
| This parameter identifies the offset of the ADC channel (for bipolar signals). This variable is automatically generated during calibration process and used for the representation of the normalized current. |

| `param.inverter_current_w_offset` |
|---|
| This parameter identifies the offset of the ADC channel (for bipolar signals). This variable is automatically generated during calibration process and used for the representation of the normalized current. |

# Bibliography

[1] M. Schroedl, M. Hofer, W. Staffler, *Combining INFORM method, voltage model and mechanical observer for sensorless control of PM synchronous motors in the whole speed range including standstill.* Springer 2006.

[2] J. Rodriguez, P. Cortes, *Predictive Control of Power Converters and Electrical Drives.* Wiley 2012.

[3] P. Krause, O. Wasynczuk, S. Sudhoff, S. Pekarek, *Analysis of Electric Machinery and Drive Systems.* Third Edition Wiley 2013.

[4] N. Bianchi, T. Jahns, *Design, analysis, and control of interior PM synchronous machines.* CLEUP 2004.

[5] G. L. Plett, *Battery Modeling. Volume I.* Artech House 2015.

[6] G. L. Plett, *Battery Modeling. Volume II.* Artech House 2015.

[7] G. L. Plett, *Extended Kalman filtering for battery management systems of LiPB/based HEV battery packs. Part 1. Background.* Journal of Power Sources 134 (2004).

[8] G. L. Plett, *Extended Kalman filtering for battery management systems of LiPB/based HEV battery packs. Part 2. Modeling and identification.* Journal of Power Sources 134 (2004).

[9] G. L. Plett, *Extended Kalman filtering for battery management systems of LiPB/based HEV battery packs. Part 3. State and parameter estimation.* Journal of Power Sources 134 (2004)