

metawells

Motor Tuning Reference Guide

Version 1.3 |  2025/08/13 |  Aaron Huang (PMD Team)

◆ Basic Theory

- Permanent Magnet Synchronous Motor Coordinate System
- Multi-Loop Control Architecture

◆ Motor Tuning GUI

- Motor tuning process
- FOC work flow
- Parameters

◆ EVB

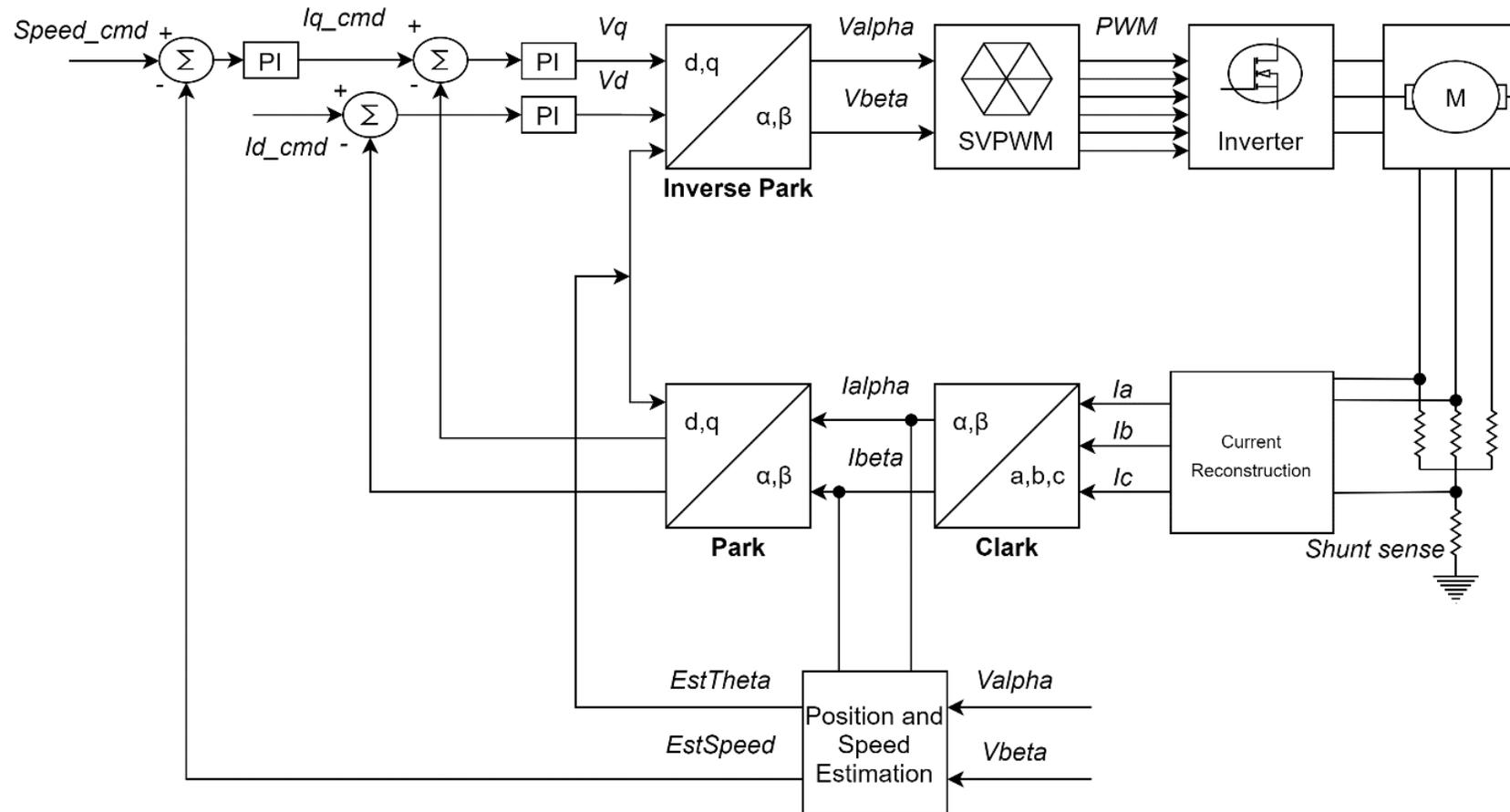
- Board
- SCH

◆ Monitoring

- J-Scope

Permanent Magnet Synchronous Motor Speed Control Block Diagram

The speed control loop of the permanent magnet synchronous motor is shown in the diagram below. In this system, the permanent magnet synchronous motor uses vector control (indirect field-oriented control, also known as FOC control), where the d-q axis current, specifically i_d^* , is set to a constant value. This allows the coupled nonlinear behavior of the permanent magnet synchronous motor to be decoupled, enabling the control of the q-axis current to control the motor torque (T_e).



- ◆ By using coordinate transformation techniques, the original a-b-c coordinate system can be converted into the d-q coordinate system, which is easier to control. The dynamic equations of the permanent magnet synchronous motor can be expressed as follows:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \end{bmatrix} = \begin{bmatrix} r_s + sL_{qs} & \omega_e L_{ds} \\ -\omega_e L_{qs} & r_s + sL_{ds} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} + \begin{bmatrix} \omega_e \lambda_f \\ 0 \end{bmatrix} \quad (1)$$

Where v_{qs} and v_{ds} are the d-q axis voltages, i_{qs} and i_{ds} are the d-q axis currents, r_s is the phase resistance, ω_e is the electrical angular speed, L_{ds} and L_{qs} are the d-q axis inductances, and λ_f is the d-q rotor field flux.

The motor torque generated by the direct axis flux and quadrature axis current is given by the following equation:

$$T_e = \frac{3}{2} \frac{P}{2} (\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) \quad (2)$$

Where T_e is the torque, P is the number of poles, and λ_{ds} and λ_{qs} are the d-q axis stator fluxes.

If the i_{ds} control in equation (2) is set to zero, then the following can be obtained:

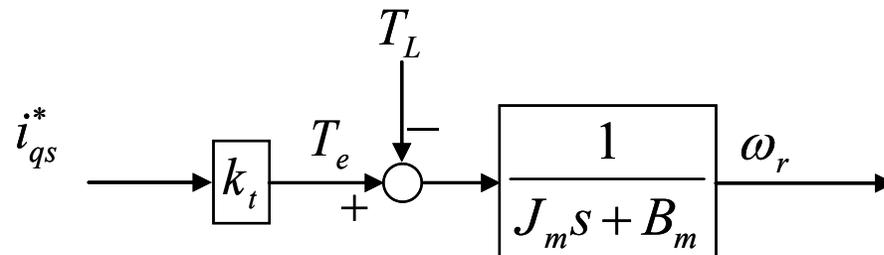
$$T_e = \frac{3P}{4} \lambda_{ds} i_{qs} = K_t i_{qs} \quad (3)$$

The mechanical equation after adding the load is as follows:

$$T_e = J_m \frac{d\omega_r}{dt} + B_m \omega_r + T_L \quad (4)$$

Where J_m is the moment of inertia, B_m is the viscous coefficient, and ω_r is the rotor speed.

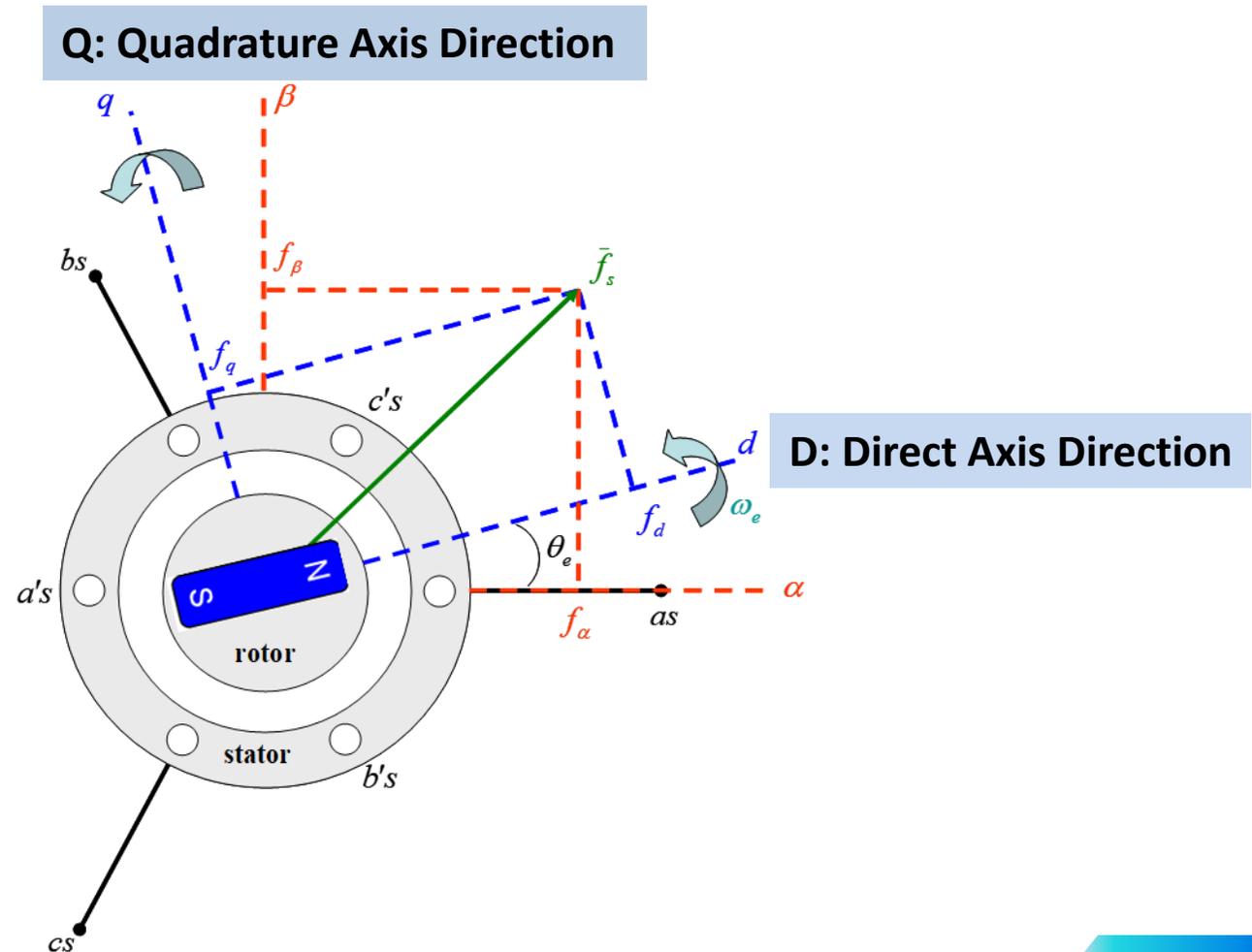
If the magnetization produced by the motor's magnets lies on the d-axis and i_{ds} is controlled to be zero, then the stator current will completely lie on the q-axis. In this case, the electromagnetic torque will be directly proportional to i_{qs} , and the AC motor can be decoupled, allowing for control similar to that of a DC motor.



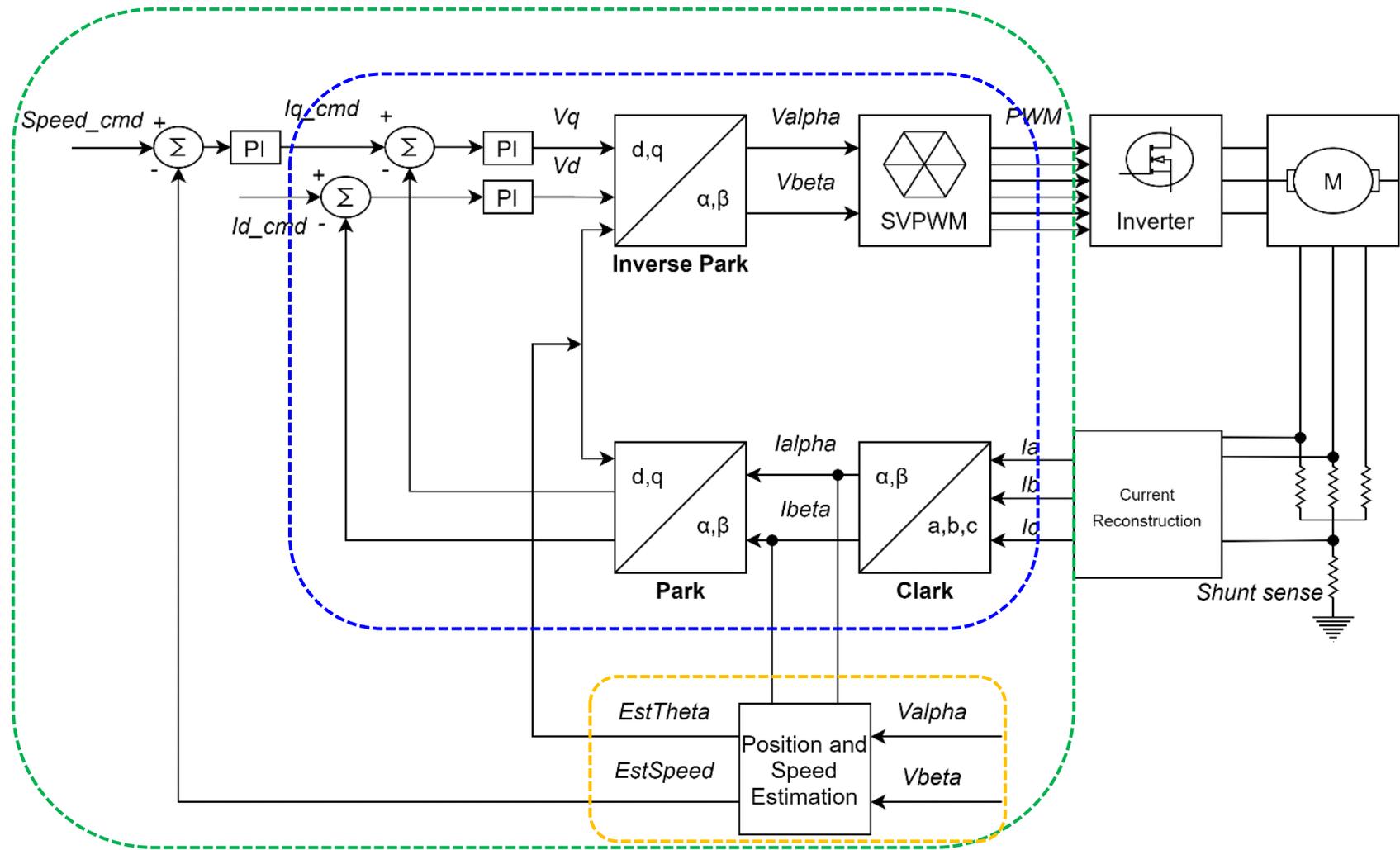
Permanent Magnet Synchronous Motor Coordinate System

- ◆ Coordinate transformation is the conversion between the three-phase coordinate system and the two-axis perpendicular (stationary and synchronous) coordinate system, as shown in the following diagram.

a-b-c: Stationary three-phase coordinate system
 α - β : Stationary two-axis coordinate system
d-q: Synchronous rotating axis



Multi-Loop Control Architecture

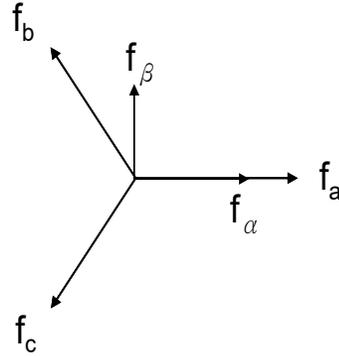


- Blue: Current loop
- Orange: Observer
- Green: Speed loop

Coordinate Transformation

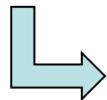
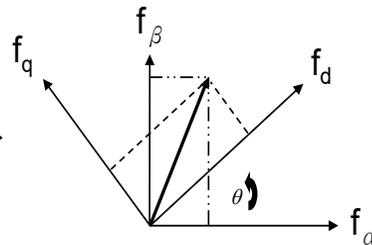
CLARK transform

$$\begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix}$$



PARK transform

$$\begin{bmatrix} f_d \\ f_q \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix}$$



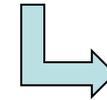
Motor Current Torque

CLARK⁻¹ transform

$$\begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix}$$

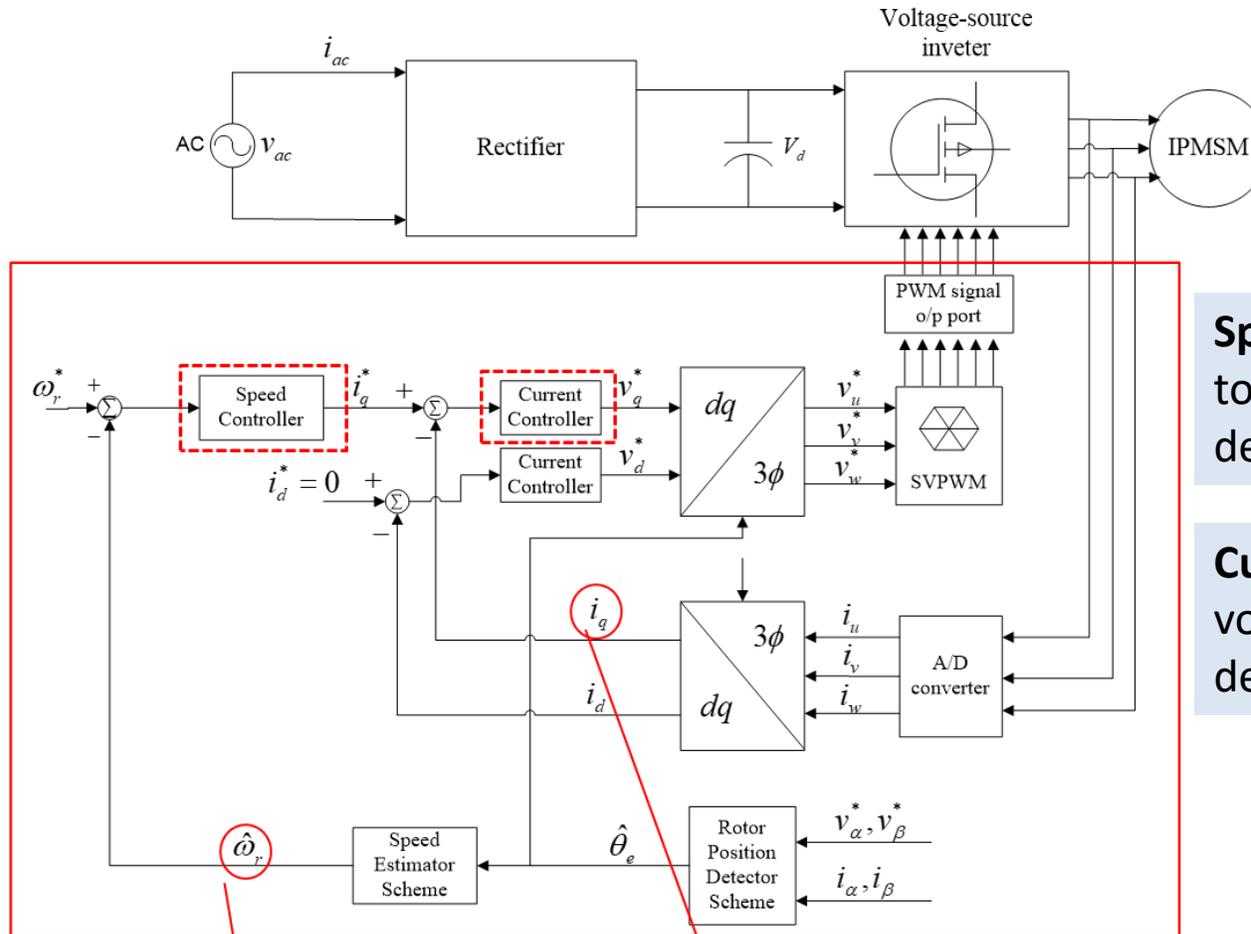
PARK⁻¹ transform

$$\begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} f_d \\ f_q \end{bmatrix}$$



Output Torque Command

Speed and Current Loop



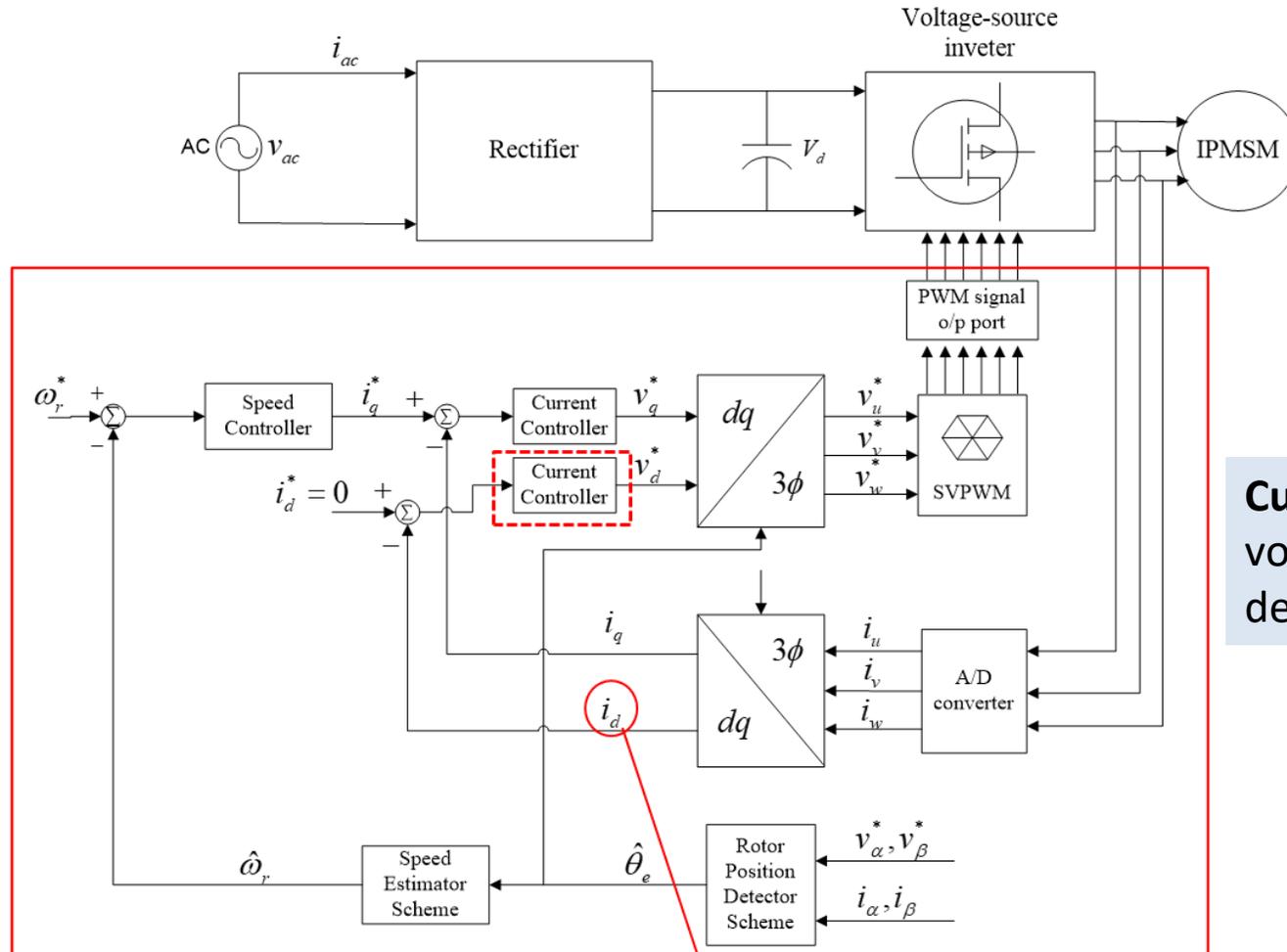
Speed Controller: Adjusts the torque command to achieve the desired speed.

Current Controller: Adjusts the voltage command to achieve the desired torque

Achieving Desired Speed?

Achieving Desired Torque?

Speed and Current Loop

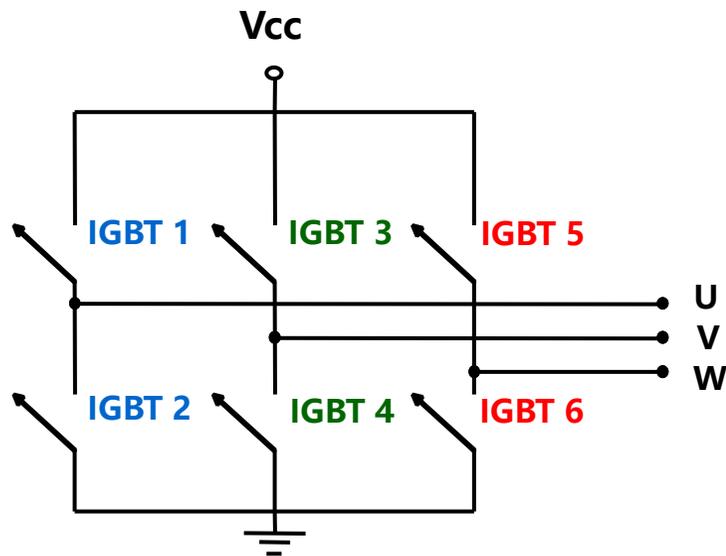


Current Controller: Adjusts the voltage command to achieve decoupling control.

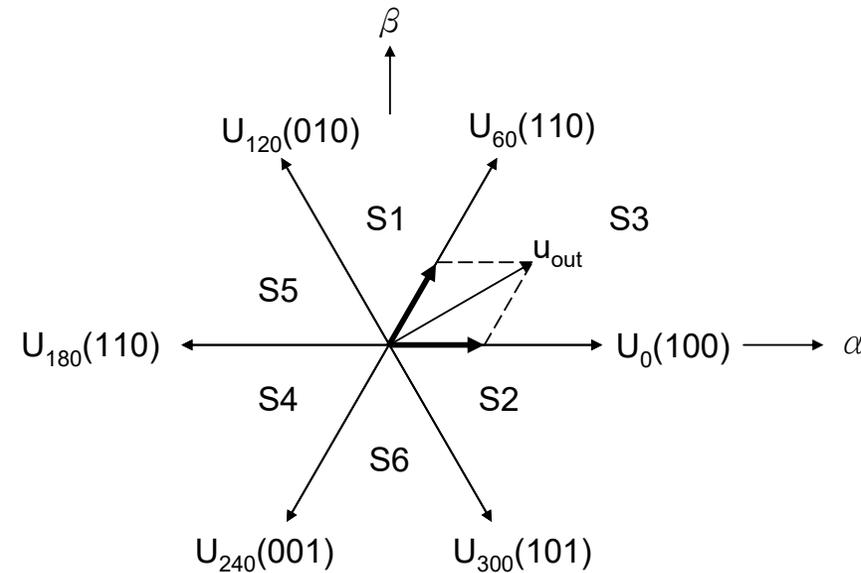
MCU

**Is Decoupling Achieved?
Or used for weak magnetic control?**

- ◆ The Space Vector Pulse Width Modulation (SVPWM) method uses a combination of 8 switching patterns from six power transistors to approximate the motor voltage vector U_{out} . The motor voltage vector U_{out} will fall within one of the six sectors ($S_3, S_1, S_5, S_4, S_6, S_2$) as shown in Figure 7, and its value is synthesized from two of the six basic voltage coordinates.



Three-Phase Inverter Architecture



Voltage Basic Vector Space and Switching Patterns

◆ Basic Theory

- Permanent Magnet Synchronous Motor Coordinate System
- Multi-Loop Control Architecture

◆ Motor Tuning GUI

- Motor tuning process
- FOC work flow
- Parameters

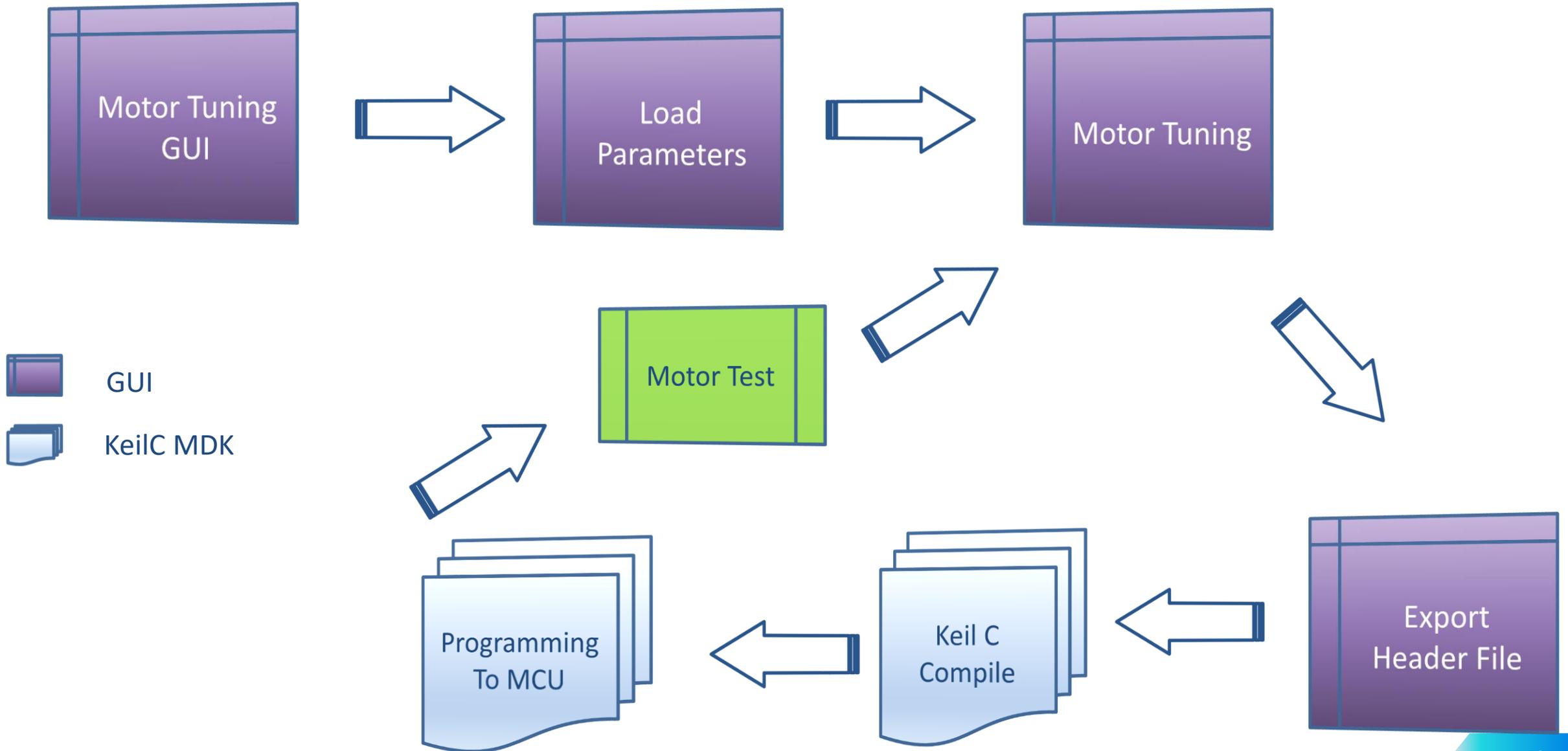
◆ EVB

- Board
- SCH

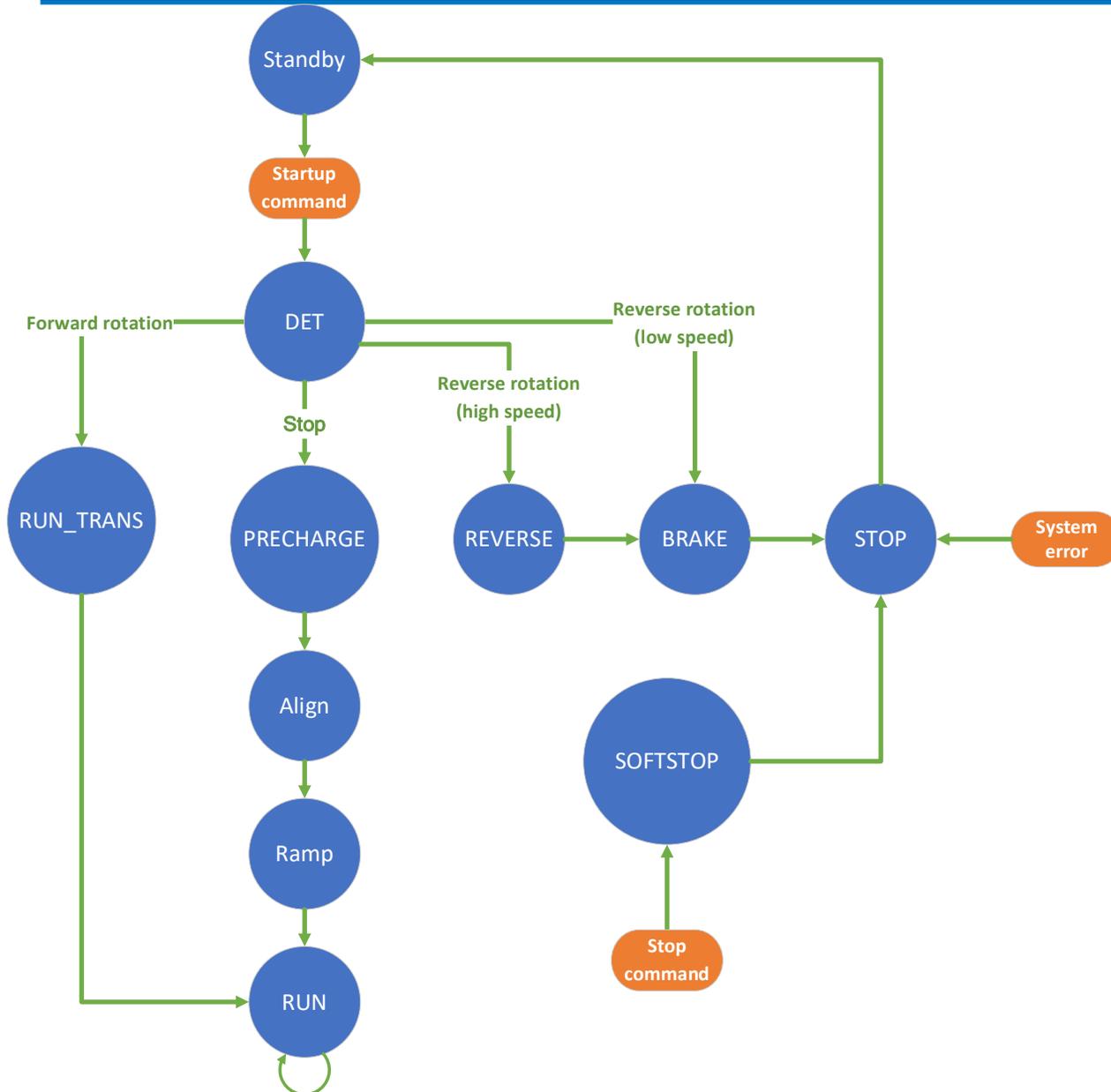
◆ Monitoring

- J-Scope

Motor tuning process

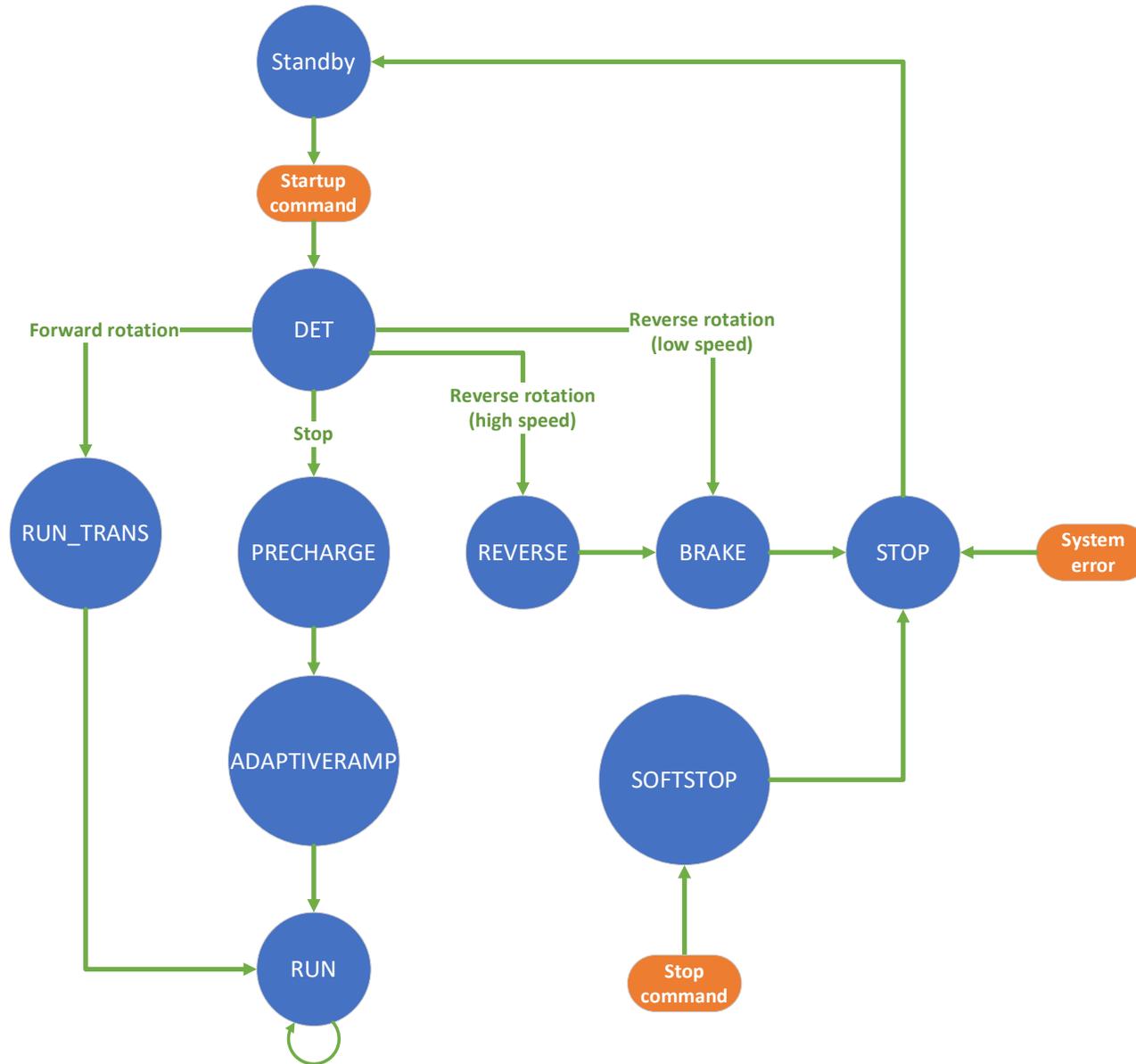


Open-Loop Ramp-Up work flow



State name	Description
Standby	Wait for startup command
DET	Up-Down wind detection
PRECHARGE	Bootstrap Cap. charging
ALIGN	Alignment
RAMP	Ramp-up
RUN_TRANS	Down-Wind speed connect
RUN	Close loop running
REVERSE	Up-Wind brake
BRAKE	Brake
SOFTSTOP	Soft stop
STOP	Stop

Adaptive Ramp-Up work flow



State name	Description
Standby	Wait for startup command
DET	Up-Down wind detection
PRECHARGE	Bootstrap Cap. charging
ADAPTIVERAMP	Adaptive Ramp-up
RUN_TRANS	Down-Wind speed connect
RUN	Close loop running
REVERSE	Up-Wind brake
BRAKE	Brake
SOFTSTOP	Soft stop
STOP	Stop

Motor Tuning GUI

C:\Users\aaaron\Documents\MetaWells\user_parameters.xlsx

Load Parameters Save as Parameters Export Header File Save as Header File About

1.Function	2.Motor	3.HW	4.User	5.Alarm	6.PreCharge	7.Alignment	8.AdaptiveRamp	9.Ramp	10.RunTrans	12.Brake	13.Current Loop	14.Speed Loop	16.BEMFOB	17.Softstop	18.BEMFDet	19.DeadTimeComp	20.ReverseRamp	Note
Ramp_Mode:	<input type="radio"/> SYNC_RAMP <input checked="" type="radio"/> ADAPTIVE_RAMP		<pre>#define RAMP_MODE #define OBMODE #define SVM_MODE_PARA ADAPTIVE_RAMP E_SVM7 BEMFOB</pre>															
OBMODE:	<input checked="" type="radio"/> BEMFOB																	
SVM_MODE_PARA	<input type="radio"/> E_SVM5 <input checked="" type="radio"/> E_SVM7																	

Adjust Parameters Area

PARA	Unit	Value
DISP_MAX_CURR_PARA	A	6.60
DISP_MAX_VBUS_PARA	V	36.26
DISP_MAX_SPD_PARA	RPM	4200
DISP_MAX_DUTY_PARA	%	84.00
DISP_BEMFDet_TIMEOUT(sec)		0.32
PU_CURR_PARA		4964
PU_SPD_PARA		7.80
PU_VOLT_PARA		2730
PU_VBUS_PARA		112

Information Area

Motor Tuning GUI – Simplified vs. Complete

- ◆ Simplified (28 parameters)
 - Maximally streamlined, retaining only the basic parameter page, suitable for beginners to quickly get the motor running.
- ◆ Complete (93 parameters)
 - Suitable for advanced users, with all parameter interfaces enabled.

2.Motor		3.HW		4.User		5.Alarm		8.AdaptiveRamp		13.Current Loop		14.Speed Loop		16.BEMFOB		Note
Num	Title	Unit	Value	Min	Typical	Max										
1	Rshunt	mohm	50				Simplified									
2	ADC_VREF	V	3.3	3.3	5											
3	PWM_FREQ	Khz	16		16	25										
4	DeadTime	ns	1000													

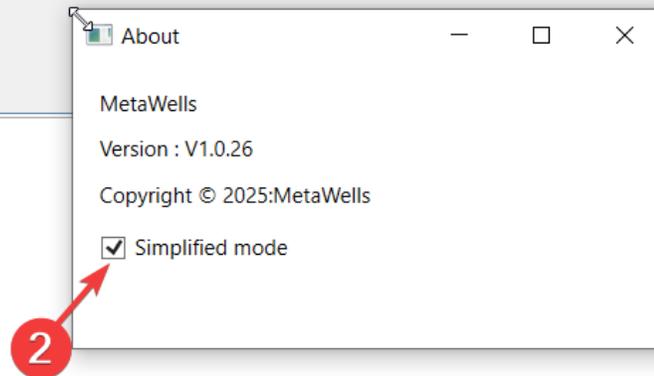
1.Function		2.Motor		3.HW		4.User		5.Alarm		6.PreCharge		7.Alignment		8.AdaptiveRamp		9.Ramp		10.RunTrans		12.Brake		13.Current Loop		14.Speed Loop		16.BEMFOB		17.Softstop		18.BEMFDet		19.DeadTimeComp		20.ReverseRamp		Note
Num	Title	Unit	Value	Min	Typical	Max																														
1	Rshunt	mohm	50				Complete																													
2	Rshunt_Gain		5																																	
3	VBUS_Gain		0.091																																	
4	BEMF_Gain		0.175																																	
5	ADC_VREF	V	3.3	3.3	5																															
6	ADC_Dig_Bit	bits	12		12																															
7	MCU_MCLK	MHz	48																																	
8	PWM_FREQ	Khz	16		16	25																														
9	DeadTime	ns	1000																																	
10	ADC_SH_TR	ns	4000																																	
11	TS	ns	0																																	
12	Loop_Div		2	1	2																															

Motor Tuning GUI – Simplified vs. Complete

Load Parameters Save as Parameters Export Header File Save as Header File About

2.Motor 3.HW 4.User 5.Alarm 8.AdaptiveRamp 13.Current Loop 14.Speed Loop 16.BEMFOB Note

Num	Title	Unit	Value	Min	Typical	Max
1	Rshunt	mohm	50			
2	ADC_VREF	V	3.3	3.3	5	
3	PWM_FREQ	Khz	16		16	25
4	DeadTime	ns	1000			



Tips:
When checked, the software runs in Simplified mode. When unchecked, the complete version with all parameter interfaces is available.

PARA	Unit	Value
DISP_MAX_CURR_PARA	A	3.30
DISP_MAX_VBUS_PARA	V	36.26
DISP_MAX_SPD_PARA	RPM	4200
DISP_MAX_DUTY_PARA	%	84.00
DISP_BEMFDet_TIMEOUT(sec)		0.64
PU_CURR_PARA		9929
PU_SPD_PARA		7.80
PU_VOLT_PARA		2730
PU_VBUS_PARA		112

Motor Tuning GUI – Load Parameters

Load Parameters Save Parameters Export Header File Save as Header File About

Function Motor HW User Alarm PreCharge Alignment AdaptiveRamp Ramp RunTrans Brake Current Loop Speed Loop BEMFOB Softstop BEMFDet DeadTimeCo

Ramp_Mode: **1** SYNC_RAMP ADAPTIVE_RAMP #define RAMP_MODE
 OBMODE: BEMFOB #define OBMODE
 SVM_MODE_PARA E_SVM5 E_SVM7 #define SVM_MODE_PARA

Open

« Meta... > PM30003A_2R_Sam... Search PM30003A_2R_Sample...

Organize New folder

Name	Date modified	Typ
BSP_DRV	5/7/2025 2:30 PM	File
KEIL_PRJ	5/7/2025 3:43 PM	File
PM30003A	5/7/2025 2:24 PM	File
RTT	5/7/2025 2:30 PM	File
SYSTEM	5/7/2025 2:24 PM	File
USER	5/7/2025 2:31 PM	File
user para	5/7/2025 11:28 AM	Microsoft Excel File

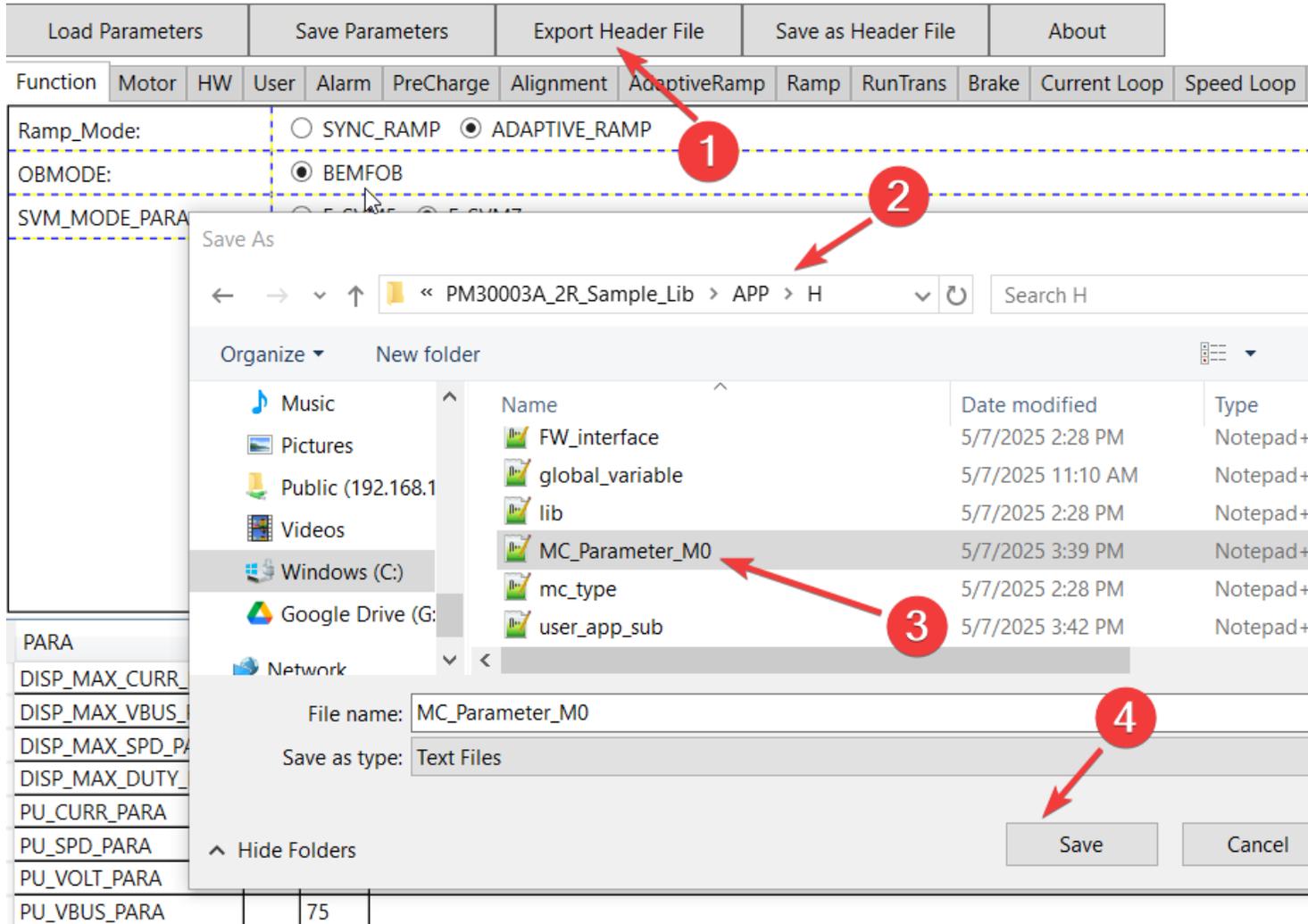
File name: user para Excel Files **3** Open Cancel

PARA	Unit	Value
DISP_MAX_CURR_PARA	A	5.00
DISP_MAX_VBUS_PARA	V	54.95
DISP_MAX_SPD_PARA	RPM	3000
DISP_MAX_DUTY_PARA	%	84.00
PU_CURR_PARA		6553
PU_SPD_PARA		10.92
PU_VOLT_PARA		1365
PU_VBUS_PARA		75

Load Parameters: *.xls

Function	PARA	UNIT	VALUE	MIN	TYP	MAX
3	Ramp_Mo	SYNC_RAMP/A ADAPTIVE_RAMP				
4	OBMODE	SYNC_RAMP/A BEMFOB				
5	SVM_MOE	E_SVM5/E_SVI E_SVM7				
8	Motor	DIR	E_CW/E_CCW			
9	MaxSpd	RPM	2000			
10	Pole	pole	10		2	
11	R(Phase)	ohm	0.7			
12	Ld(Phase)	mH	1.14			
13	Lq(Phase)	mH	1.29			
14	Rated V	V	24			
18	HW	Rshunt	mohm	50		
19	Rshunt_Ge		10			
20	VBUS_Gai		0.091			
21	BEMF_Gai		0.175			
22	ADC_VREF	V	5			
23	ADC_Dig_f	bits	12	3.3	5	
24	MCU_MCL	MHz	48		12	
25	PWM_FRE	Khz	16			
26	DeadTime	ns	1000		16	25
27	ADC_SH_T	ns	4000			
28	TS	ns	0			
29	Loop_Div		2		1	2

Motor Tuning GUI – Export Header File



The screenshot shows the Motor Tuning GUI with the 'Export Header File' menu item highlighted. A 'Save As' dialog box is open, showing the file name 'MC_Parameter_M0' and the save type 'Text Files'. The dialog box is located in the folder 'PM30003A_2R_Sample_Lib > APP > H'. The file 'MC_Parameter_M0' is selected in the file list. The 'Save' button is highlighted.

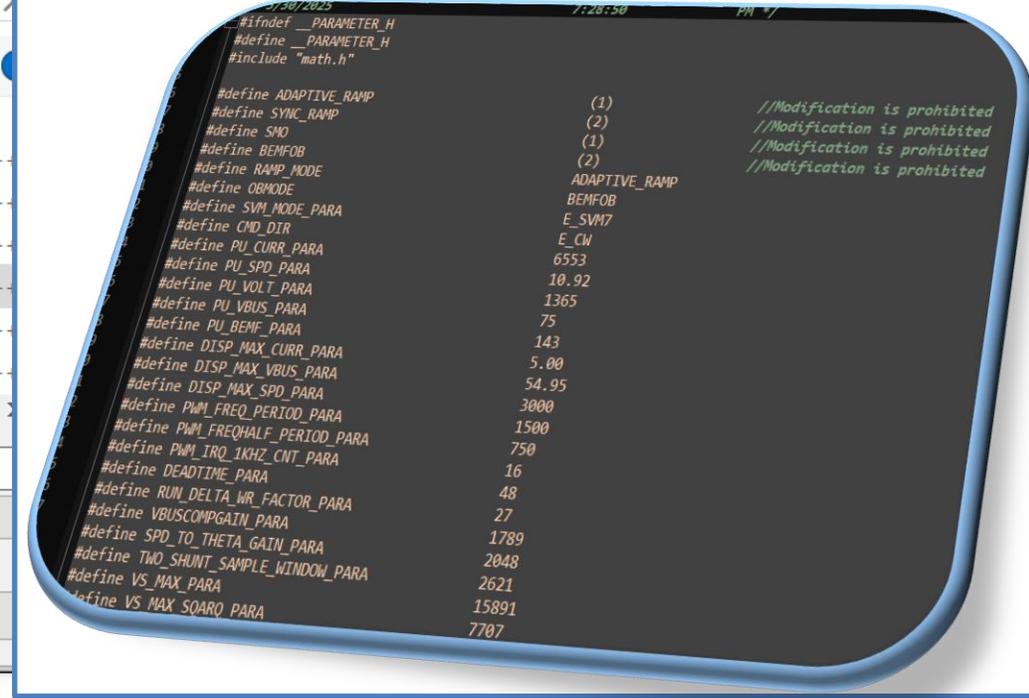
1. Click on 'Export Header File' in the menu.

2. Click on the 'Save As' dialog box.

3. Click on the file 'MC_Parameter_M0' in the file list.

4. Click on the 'Save' button.

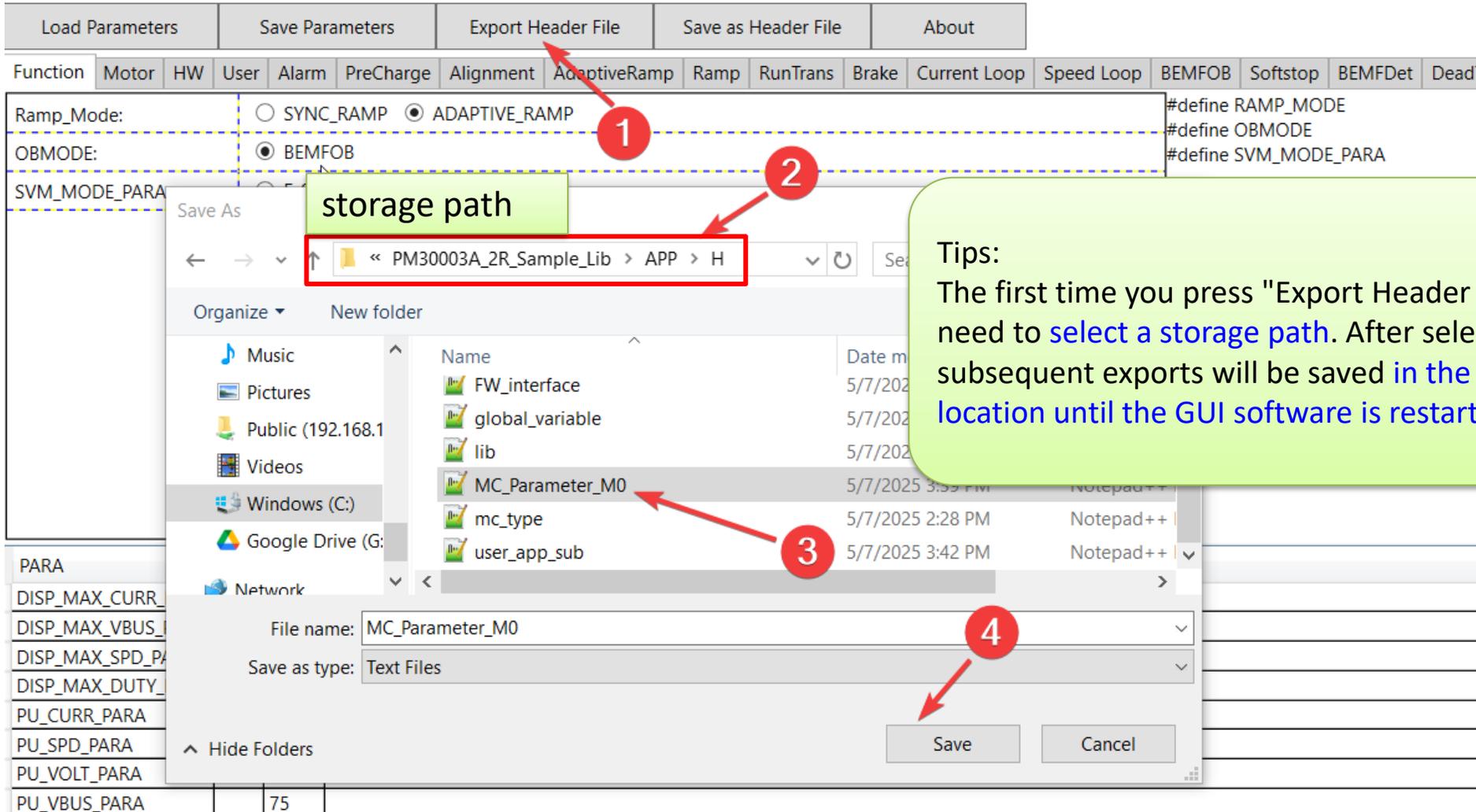
Export Header File:
MC_Parameter_M0.h



```
#ifndef __PARAMETER_H
#define __PARAMETER_H
#include "math.h"

#define ADAPTIVE_RAMP (1) //Modification is prohibited
#define SYNC_RAMP (2) //Modification is prohibited
#define SMO (1) //Modification is prohibited
#define BEMFOB (2) //Modification is prohibited
#define RAMP_MODE ADAPTIVE_RAMP
#define OBMODE BEMFOB
#define SVM_MODE_PARA SVM_MODE
#define CMD_DIR E_SVM7
#define PU_CURR_PARA E_CW
#define PU_SPD_PARA 6553
#define PU_VOLT_PARA 10.92
#define PU_VBUS_PARA 1365
#define PU_BEMF_PARA 75
#define DISP_MAX_CURR_PARA 143
#define DISP_MAX_VBUS_PARA 5.00
#define DISP_MAX_SPD_PARA 54.95
#define PWM_FREQ_PERIOD_PARA 3000
#define PWM_FREQ_HALF_PERIOD_PARA 1500
#define PWM_IRQ_1KHZ_CNT_PARA 750
#define DEADTIME_PARA 16
#define RUN_DELTA_WR_FACTOR_PARA 48
#define VBUSCOMP_GAIN_PARA 27
#define SPD_TO_THETA_GAIN_PARA 1789
#define TMO_SHUNT_SAMPLE_WINDOW_PARA 2048
#define VS_MAX_PARA 2621
#define VS_MAX_SOARO_PARA 15891
#define VS_MAX_SOARO_PARA 7707
```

Motor Tuning GUI – Export Header File



Load Parameters | Save Parameters | **Export Header File** | Save as Header File | About

Function: Motor | HW | User | Alarm | PreCharge | Alignment | **AdaptiveRamp** | Ramp | RunTrans | Brake | Current Loop | Speed Loop | BEMFOB | Softstop | BEMFDet | DeadT

Ramp_Mode: SYNC_RAMP ADAPTIVE_RAMP

OBTMODE: BEMFOB

SVM_MODE_PARA

Save As: storage path

File name: MC_Parameter_M0

Save as type: Text Files

Save | Cancel

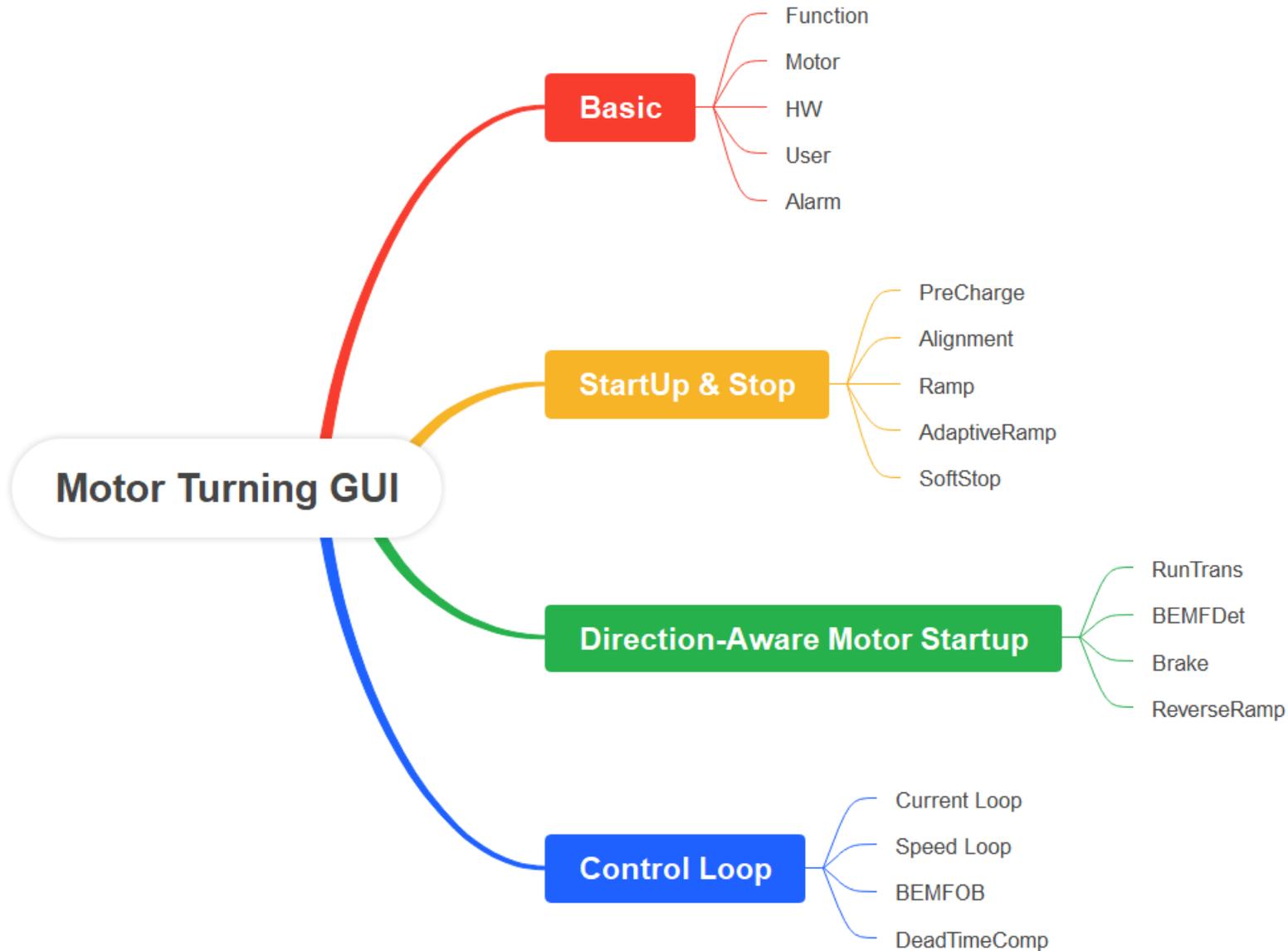
PARA

DISP_MAX_CURR_		
DISP_MAX_VBUS_		
DISP_MAX_SPD_P		
DISP_MAX_DUTY_		
PU_CURR_PARA		
PU_SPD_PARA		
PU_VOLT_PARA		
PU_VBUS_PARA	75	

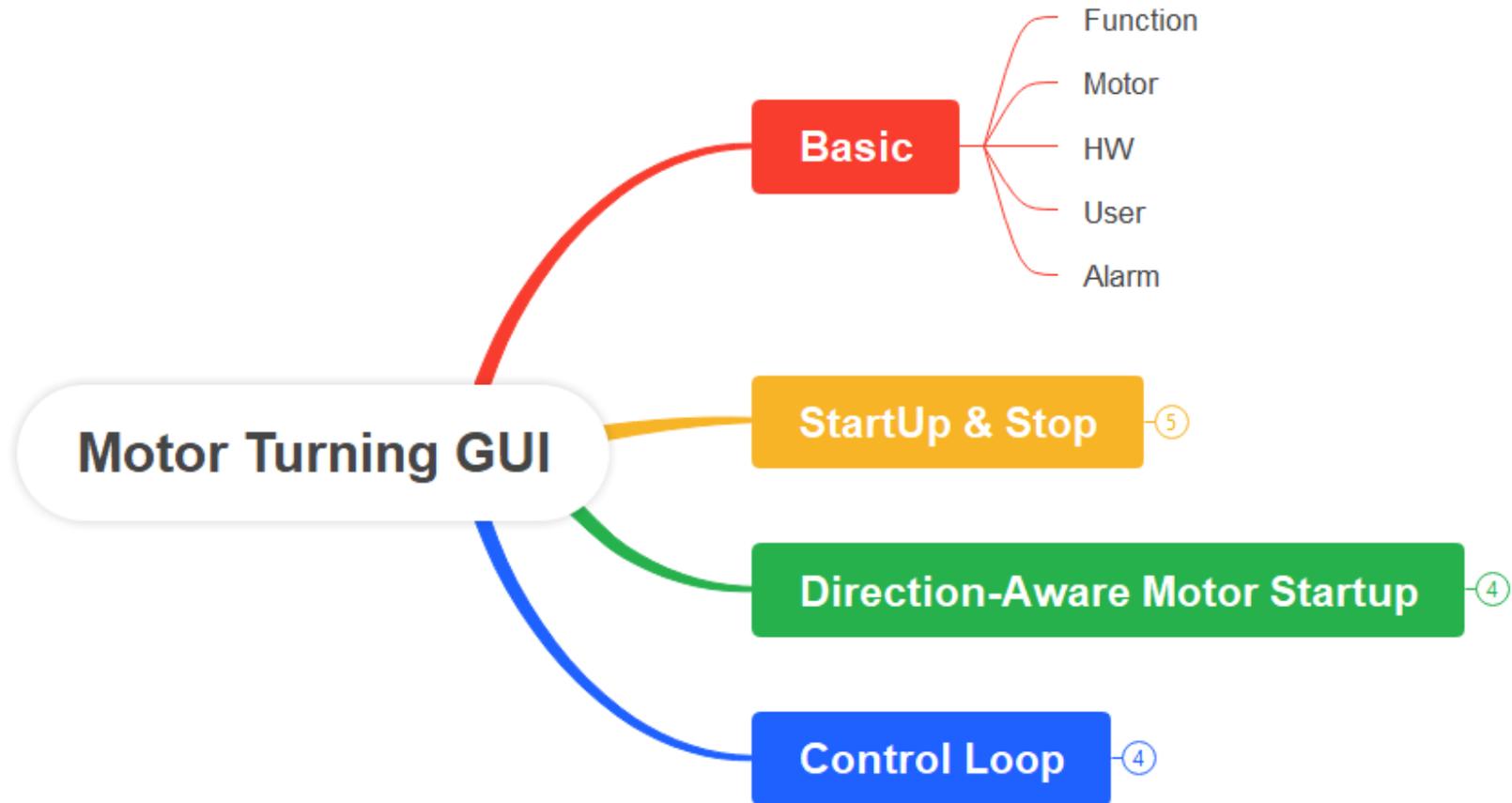
Tips:

The first time you press "Export Header File," you need to **select a storage path**. After selecting it, all subsequent exports will be saved **in the same location until the GUI software is restarted**.

Motor Tuning GUI



Motor Tuning GUI - Basic



Basic - Function

Function Motor HW User Alarm PreCharge Alignment AdaptiveRamp Ramp RunTrans Brake Current Loop Speed Loop E

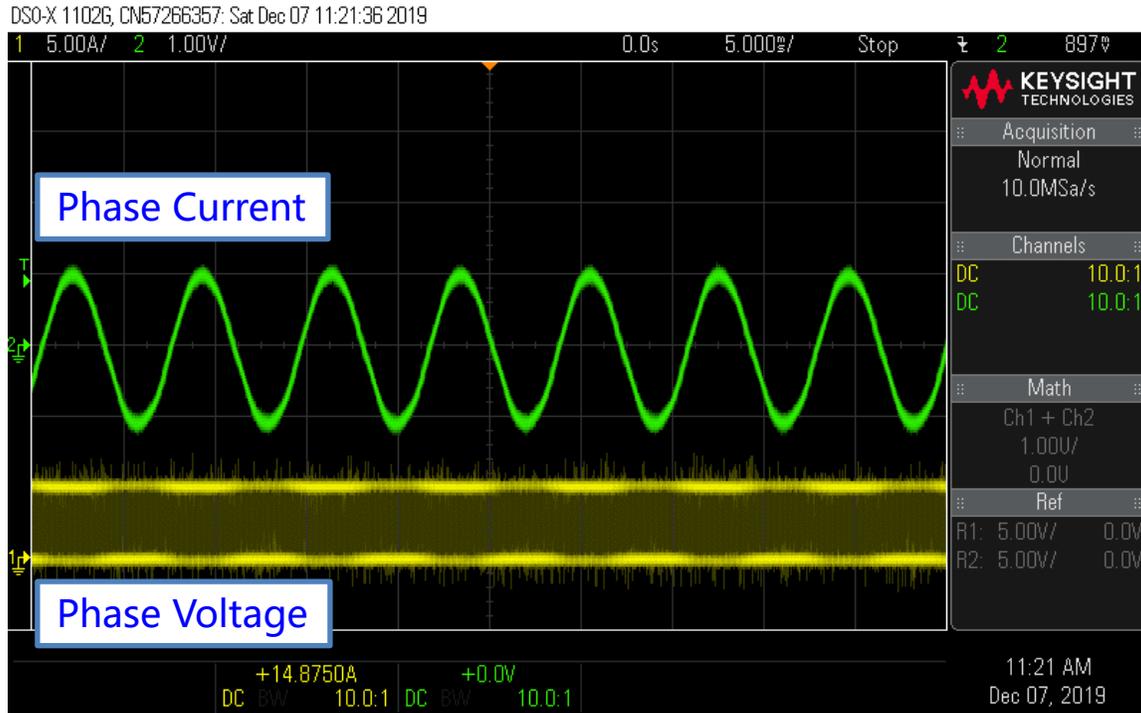
Ramp_Mode:	<input type="radio"/> SYNC_RAMP <input checked="" type="radio"/> ADAPTIVE_RAMP
OBMODE:	<input checked="" type="radio"/> BEMFOB
SVM_MODE_PARA	<input type="radio"/> E_SVM5 <input checked="" type="radio"/> E_SVM7

Parameters	Description
Ramp_Mode	SYNC_RAMP: Open-loop startup ADAPTIVE_RAMP: Adaptive startup
OBMODE	Observer
SVM_MODE_Para	E_SVM5: Five-segment SVM E_SVM7: Seven-segment SVM

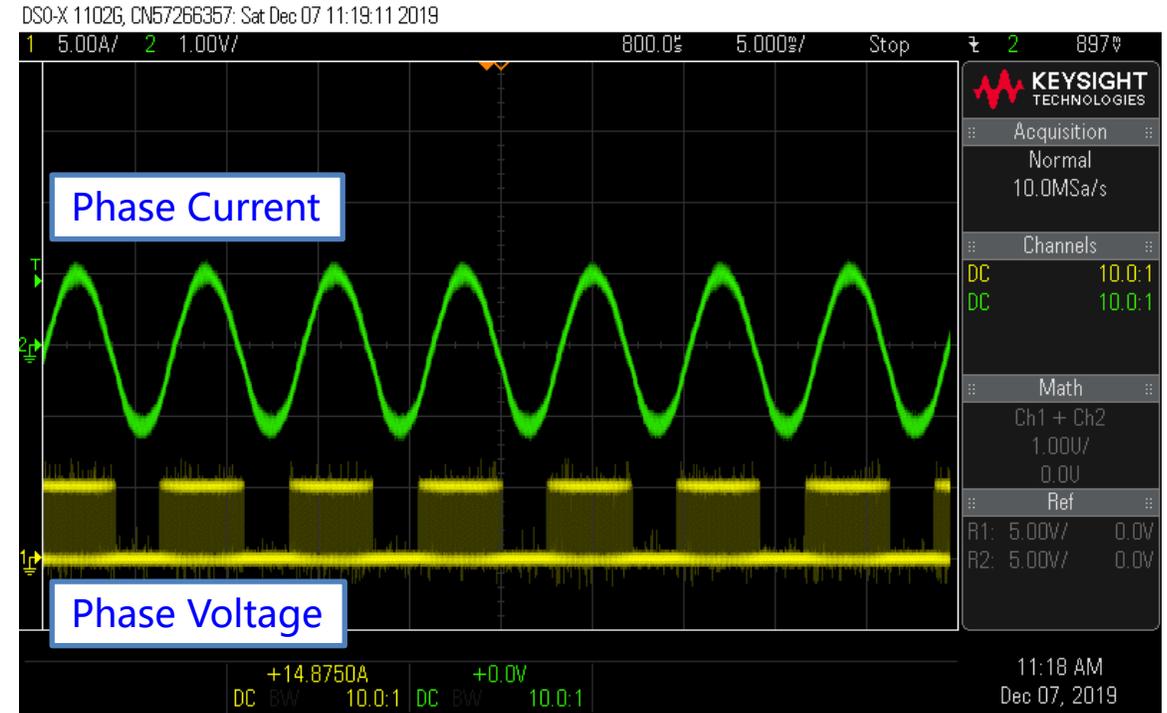
Comparison between SVPWM and SPWM

Comparison Item	5-Segment SVPWM	7-Segment SVPWM	SPWM (Sinusoidal Pulse Width Modulation)
Control Segments	5	7	No fixed segments, generated based on sine wave comparison results
DC Voltage Utilization	About 86.6%	About 86.6%	About 78.5%
Switching Loss	Lower, fewer switch transitions	Higher, high switching frequency, greater switching loss	Higher, high switching frequency, greater switching loss
Harmonic Distortion (THD)	Moderate, lower than SPWM, slightly higher than 7-Segment SVPWM	Lowest, minimal THD	Higher, greater total harmonic distortion

SVPWM waveform



7-Segment SVPWM



5-Segment SVPWM

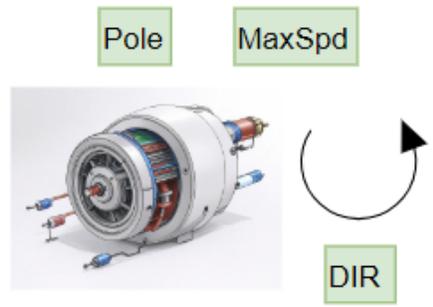
Basic - Motor

Function **Motor** HW User Alarm PreCharge Alignment

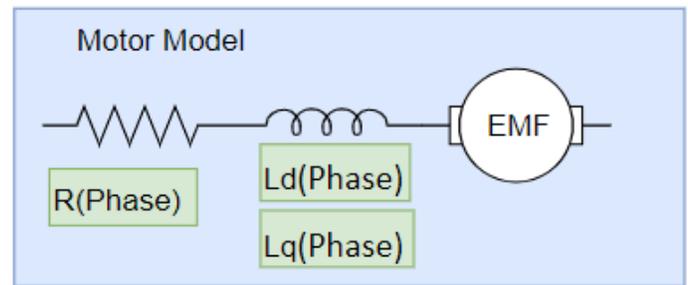
DIR: E_CW E_CCW

Num	Title	Unit	Value	Min	Typical	Max
1	MaxSpd	RPM	2000			
2	Pole	pole	10	2		
3	R(Phase)	ohm	0.7			
4	Ld(Phase)	mH	1.14			
5	Lq(Phase)	mH	1.29			
6	Rated V	V	24			

Parameters	Description
DIR	Rotation direction
MaxSpd	Motor's maximum rated speed
Pole	Motor pole pairs
R(Phase)	Motor phase resistance
Ld(Phase)	Motor D-axis inductance
Lq(Phase)	Motor Q-axis inductance
Rated V	Rated input voltage



Tips:
 $L_d \neq L_q$: IPMSM has a saliency structure
 $L_d = L_q$: SPMSM



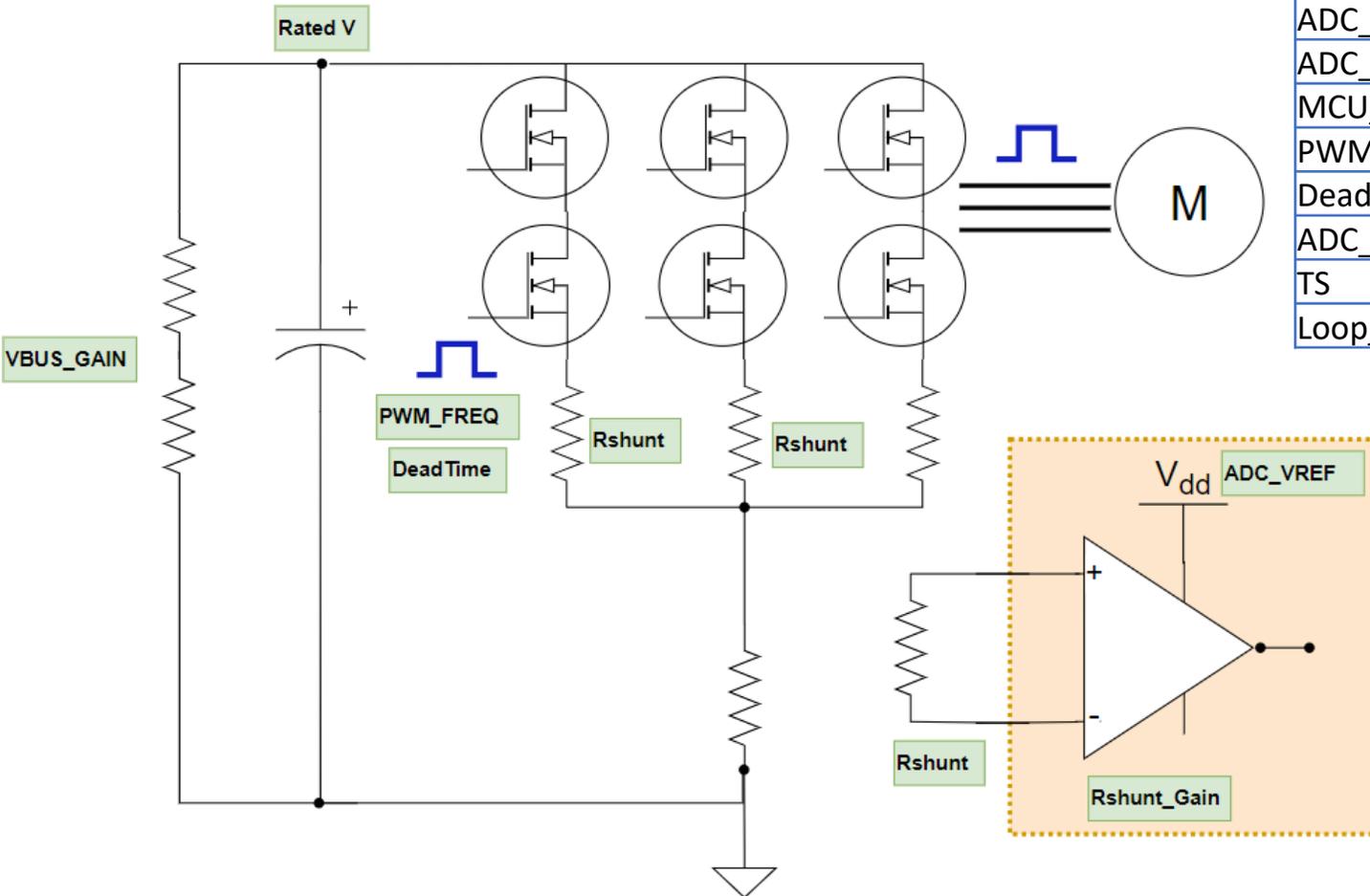
Basic – HW (Hardware)

Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRat
Num	Title	Unit	Value	Min	Typical	Max	
1	Rshunt	mohm	50				
2	Rshunt_Gain		10				
3	VBUS_Gain		0.091				
4	BEMF_Gain		0.175				
5	ADC_VREF	V	5	3.3	5		
6	ADC_Dig_Bit	bits	12		12		
7	MCU_MCLK	MHz	48				
8	PWM_FREQ	Khz	16		16	25	
9	DeadTime	ns	1000				
10	ADC_SH_TR	ns	4000				
11	TS	ns	0				
12	Loop_Div		2	1	2		

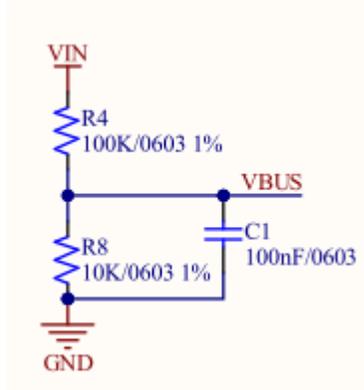
Parameters	Description
Rshunt	Phase current sensing resistor
Rshunt_Gain	Phase current amplification ratio
VBUS_Gain	Bus voltage sensing ratio
BEMF_Gain	Back-EMF sensing ratio
ADC_VREF	ADC reference voltage
ADC_Dig_Bit	ADC resolution (bits)
MCU_MCLK	MCU main clock frequency
PWM_FREQ	PWM frequency
DeadTime	Dead time
ADC_SH_TR	ADC sampling and conversion time
TS	Advanced parameters (do not modify)
Loop_Div	Phase current sensing resistor

Basic - HW (Hardware)

Parameters	Description
Rshunt	Phase current sensing resistor
Rshunt_Gain	Phase current amplification ratio
VBUS_Gain	Bus voltage sensing ratio
BEMF_Gain	Back-EMF sensing ratio
ADC_VREF	ADC reference voltage
ADC_Dig_Bit	ADC resolution (bits)
MCU_MCLK	MCU main clock frequency
PWM_FREQ	PWM frequency
DeadTime	Dead time
ADC_SH_TR	ADC sampling and conversion time
TS	Advanced parameters (do not modify)
Loop_Div	Phase current sensing resistor

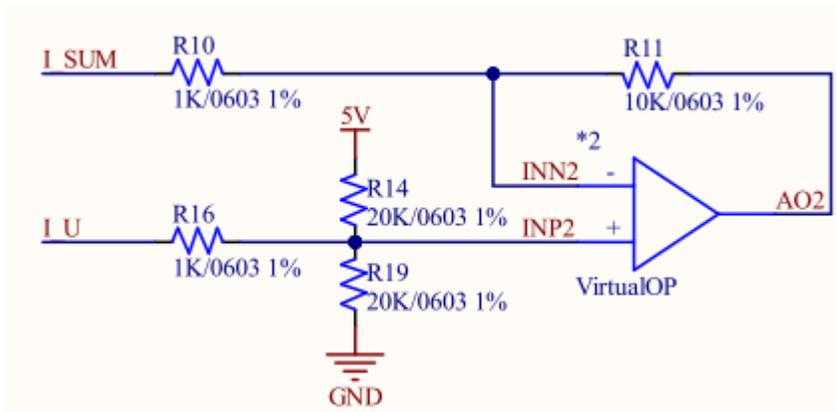


HW - Circuit calculation



$$V_{bus_Gain} = \frac{R_8}{R_8 + R_4}$$

$$V_{bus_Gain} = \frac{10k}{10k + 100k} = 0.0909 \cong 0.091$$



Differential amplifier

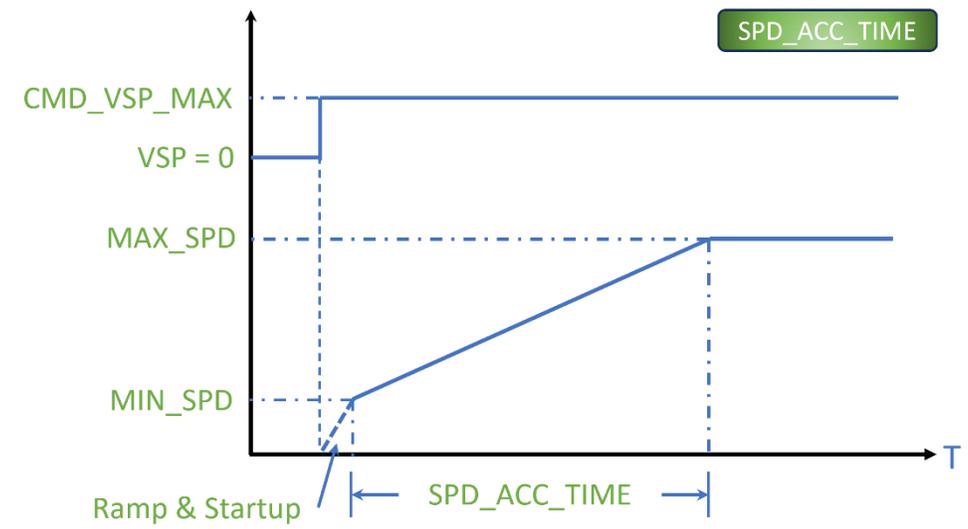
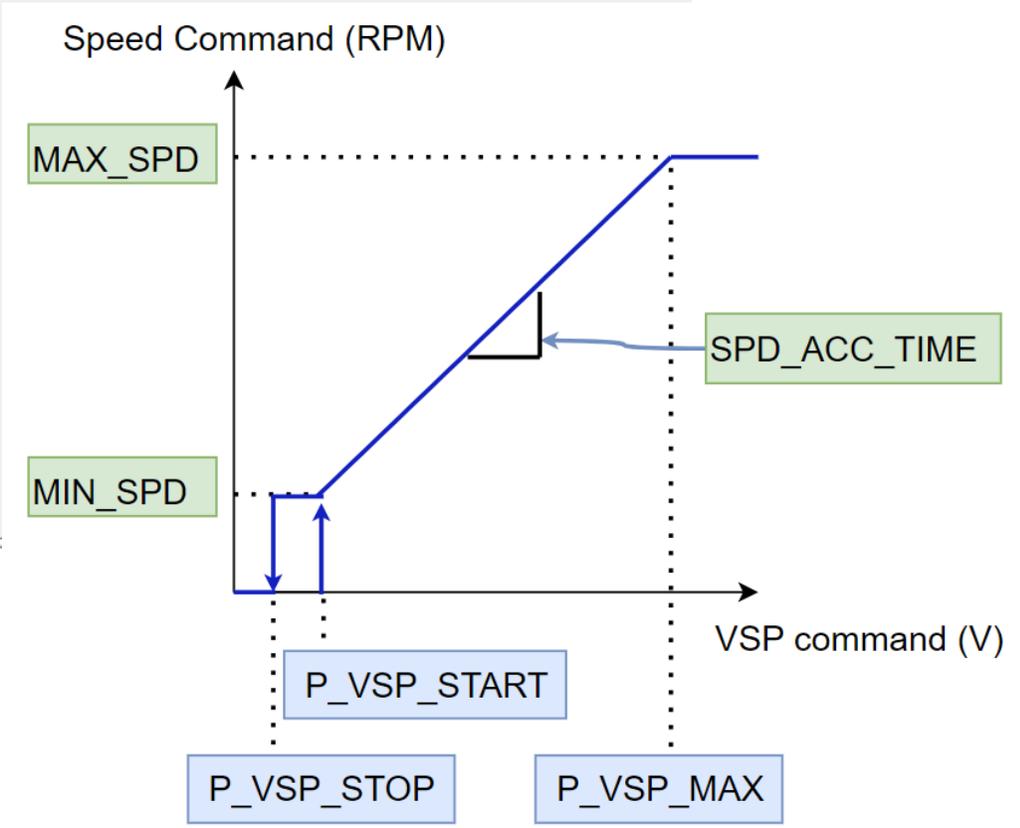
$$R_{shunt\ Gain} = \left(\frac{R_{11}}{R_{10}} \right)$$

$$R_{shunt\ Gain} = \left(\frac{10k}{1k} \right) = 10$$

Basic - User

Function	Motor	HW	User	Alarm	PreCharge	Alignment
Num	Title	Unit	Value	Min	Typical	Max
1	MAX_SPD	RPM	1200			
2	MIN_SPD	RPM	200			
3	SPD_ACC_TIME	msec	4000			

Parameters	Description
MAX_SPD	Maximum user interface speed
MIN_SPD	Minimum user interface speed
SPD_ACC_TIME	Acceleration time from minimum to maximum speed command



Protections

Error Code	Error Name	Description
1	E_OCP	FW phase over current
2	E_OFFSET_FAULT	Current sense offset voltage
4	E_OVP	Over voltage (DC Bus)
5	E_UVP	Under voltage (DC Bus)
8	E_LOSTSYNC	Observer loss of synchronization
12	E_OVERSPEED	Observer speed estimation error
13	E_RAMP_OVERTIME	Ramp-up Timeout

Tips:
Contact your FAE with the error code for faster support.

Basic - Alarm

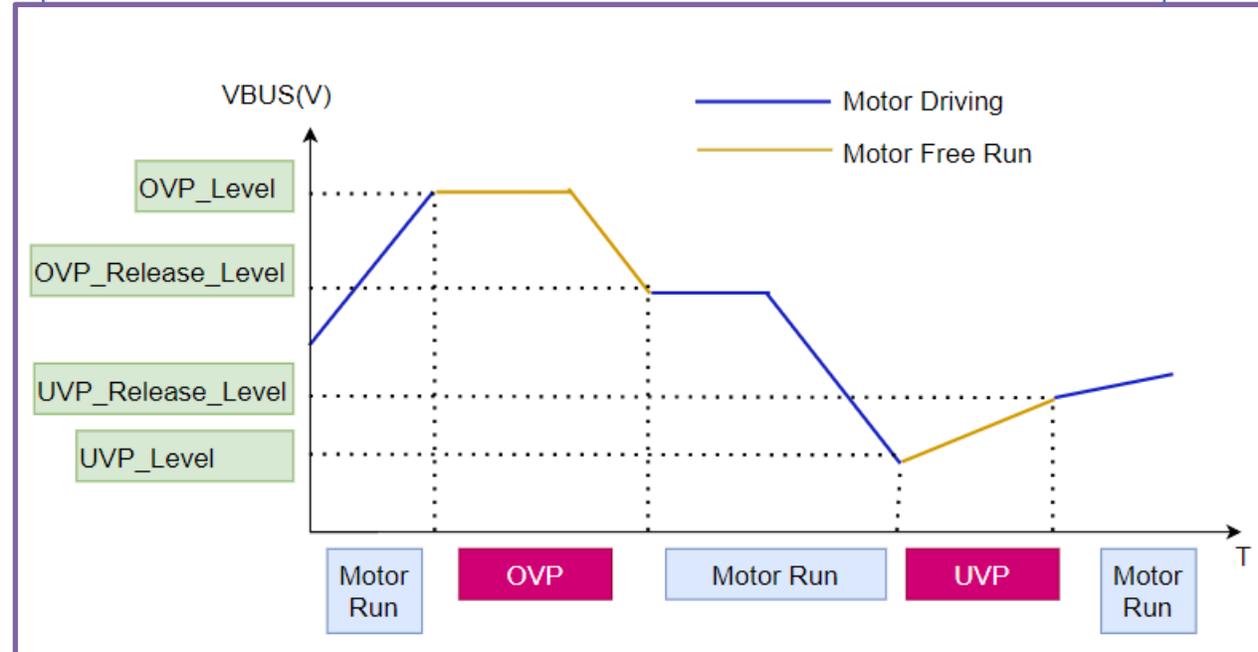
Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp
Num	Title	Unit	Value	Min	Typical	Max	
1	OVP_Level	V	32.5		36.00		
2	OVP_Release_Level	V	24		31.20		
3	UVP_Level	V	8		16.80		
4	UVP_Release_Level	V	11.5		19.20		
5	OCP_Level	A	2.5			5.00	
6	OCP_Times	times	5	1	5	30	
7	OverSpd_Time	msec	100		100		
8	LostSYNC_Time	msec	2000		2000		
9	LostSYNC_SpdFactor	times	0.5		0.5		
10	Ramp_TimeOut	msec	5000		5000		
11	LostPhase_Times	times	50		50	255	

Parameters	Description
OVP_Level	Overvoltage protection level
OVP_Release_Level	Overvoltage recovery level
UVP_Level	Undervoltage protection level
UVP_Release_Level	Undervoltage recovery level
OCP_Level	Phase current overcurrent protection level
OCP_Times	Phase current overcurrent protection count
OverSpd_Time	Overspeed condition duration before protection activation
LostSYNC_Time	Observer failure protection time
LostSYNC_SpdFactor	Advanced parameters (do not modify)
Ramp_TimeOut	Startup timeout
LostPhase_Times	Time duration before triggering phase loss protection

Basic – Alarm (OVP & UVP)

Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp
Num	Title	Unit	Value	Min	Typical	Max	
1	OVP_Level	V	32.5		36.00		
2	OVP_Release_Level	V	24		31.20		
3	UVP_Level	V	8		16.80		
4	UVP_Release_Level	V	11.5		19.20		
5	OCP_Level	A	2.5			5.00	
6	OCP_Times	times	5	1	5	30	
7	OverSpd_Time	msec	100		100		
8	LostSYNC_Time	msec	2000		2000		
9	LostSYNC_SpdFactor	times	0.5		0.5		
10	Ramp_TimeOut	msec	5000		5000		
11	LostPhase_Times	times	50		50	255	

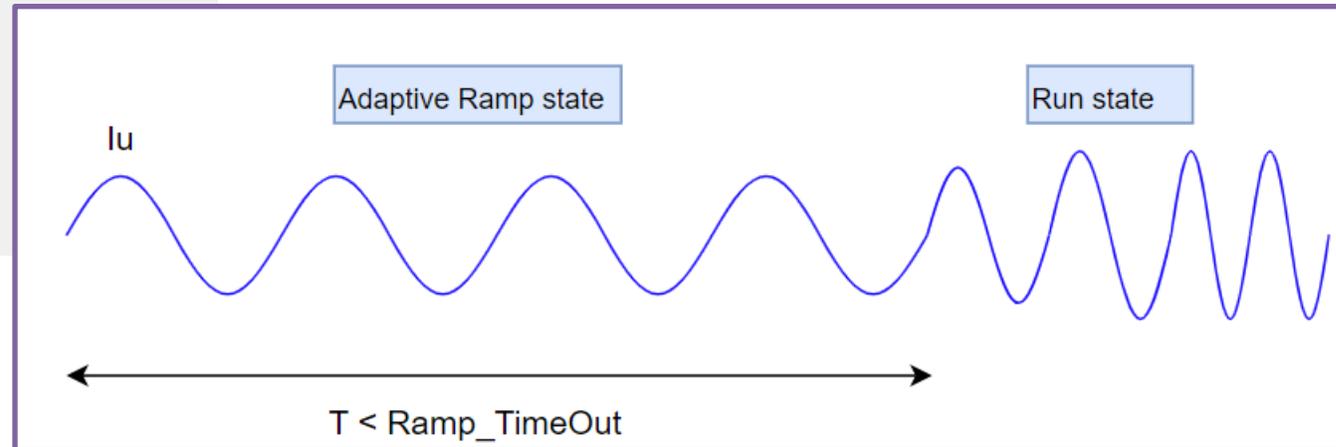
Parameters	Description
OVP_Level	Overvoltage protection level
OVP_Release_Level	Overvoltage recovery level
UVP_Level	Undervoltage protection level
UVP_Release_Level	Undervoltage recovery level



Basic – Alarm (Ramp_TimeOut)

Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp
Num	Title	Unit	Value	Min	Typical	Max	
1	OVP_Level	V	32.5		36.00		
2	OVP_Release_Level	V	24		31.20		
3	UVP_Level	V	8		16.80		
4	UVP_Release_Level	V	11.5		19.20		
5	OCP_Level	A	2.5			5.00	
6	OCP_Times	times	5	1	5	30	
7	OverSpd_Time	msec	100		100		
8	LostSYNC_Time	msec	2000		2000		
9	LostSYNC_SpdFactor	times	0.5		0.5		
10	Ramp_TimeOut	msec	5000		5000		
11	LostPhase_Times	times	50		50	255	

Parameters	Description
Ramp_TimeOut	Startup timeout



Basic – Alarm (OverSpd)

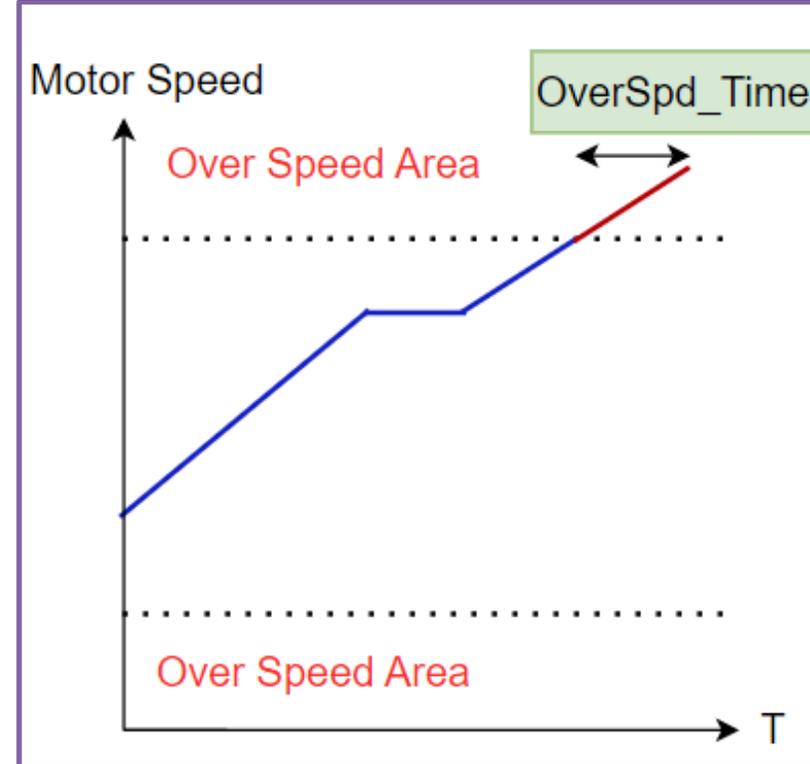
Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp
Num	Title	Unit	Value	Min	Typical	Max	
1	OVP_Level	V	32.5		36.00		
2	OVP_Release_Level	V	24		31.20		
3	UVP_Level	V	8		16.80		
4	UVP_Release_Level	V	11.5		19.20		
5	OCP_Level	A	2.5			5.00	
6	OCP_Times	times	5	1	5	30	
7	OverSpd_Time	msec	100		100		
8	LostSYNC_Time	msec	2000		2000		
9	LostSYNC_SpdFactor	times	0.5		0.5		
10	Ramp_TimeOut	msec	5000		5000		
11	LostPhase_Times	times	50		50	255	

Parameters

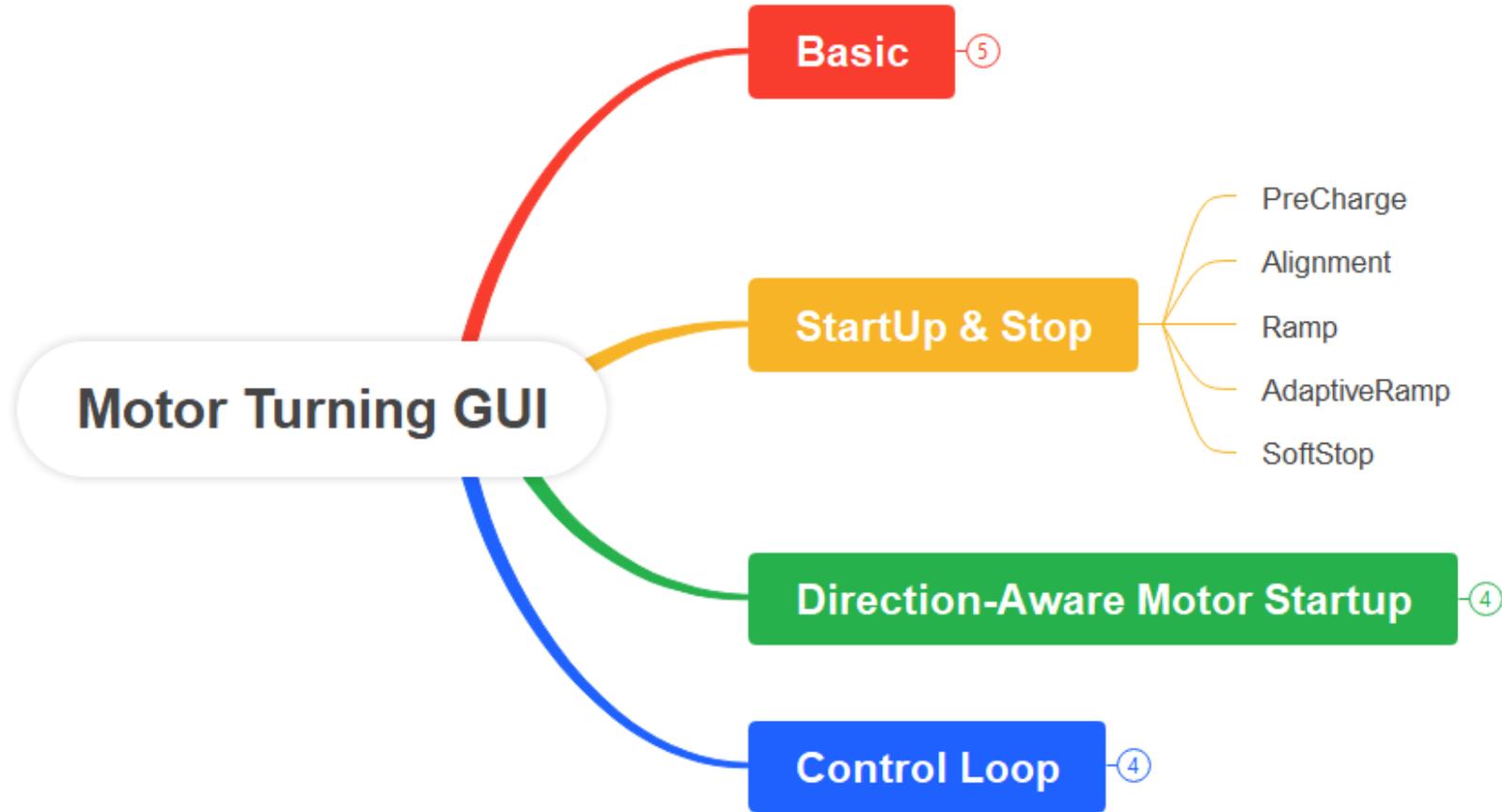
OverSpd_Time

Description

Overspeed condition duration before protection activation

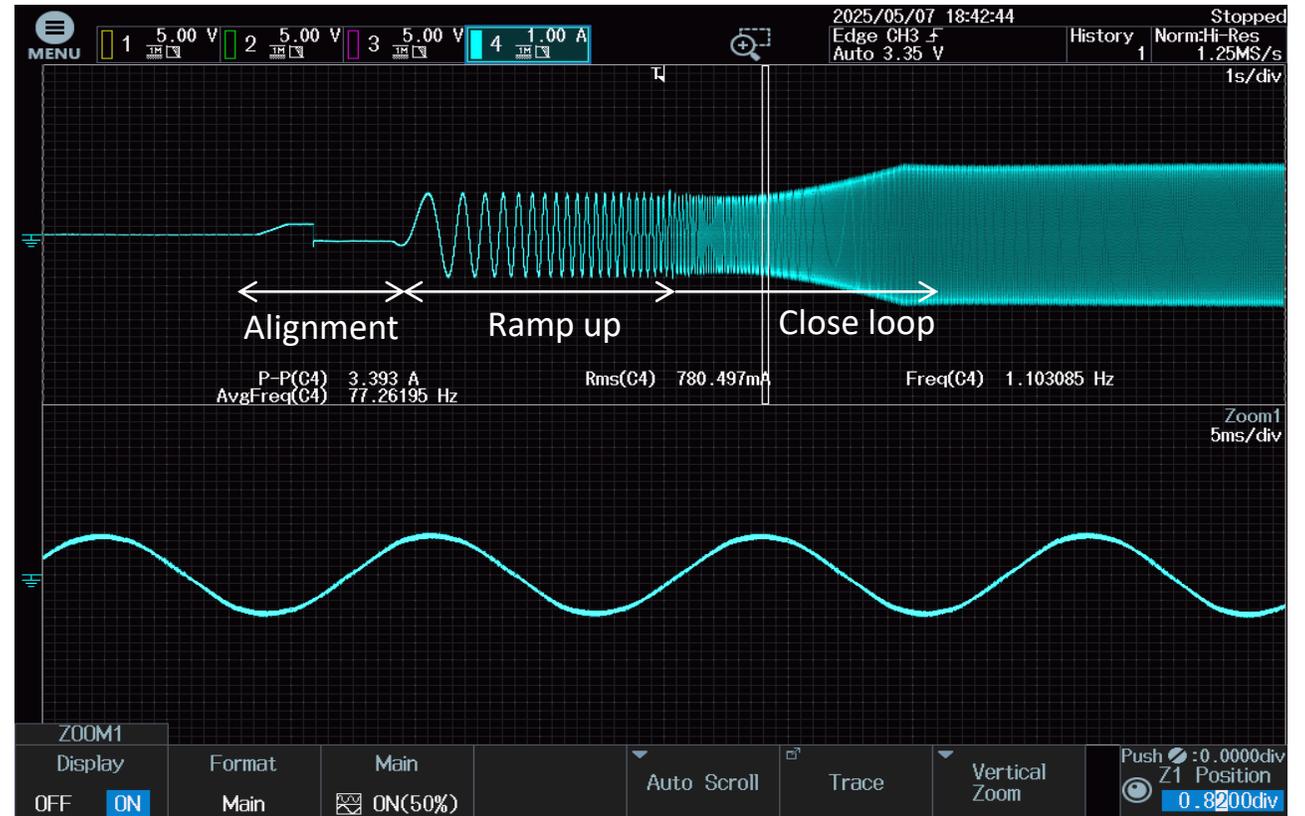


Motor Tuning GUI - StartUp and Stop

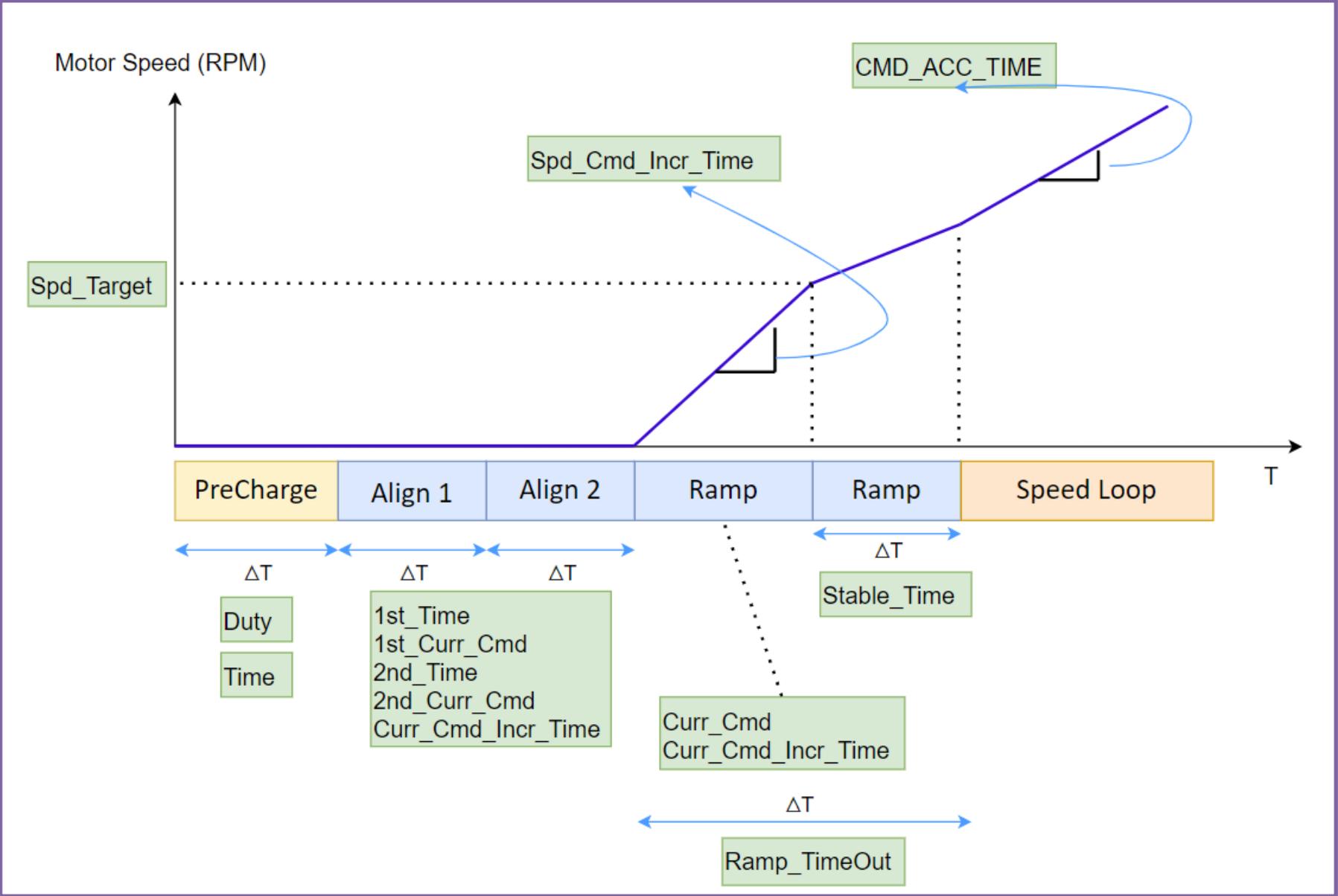


Startup method 1 - SYNC_RAMP

- ◆ Open loop synchronous ramp up
 - Step 0: PreCharge
 - Step 1: Two steps alignment
 - Step 2: Synchronous ramp up
 - Step 3: Close loop

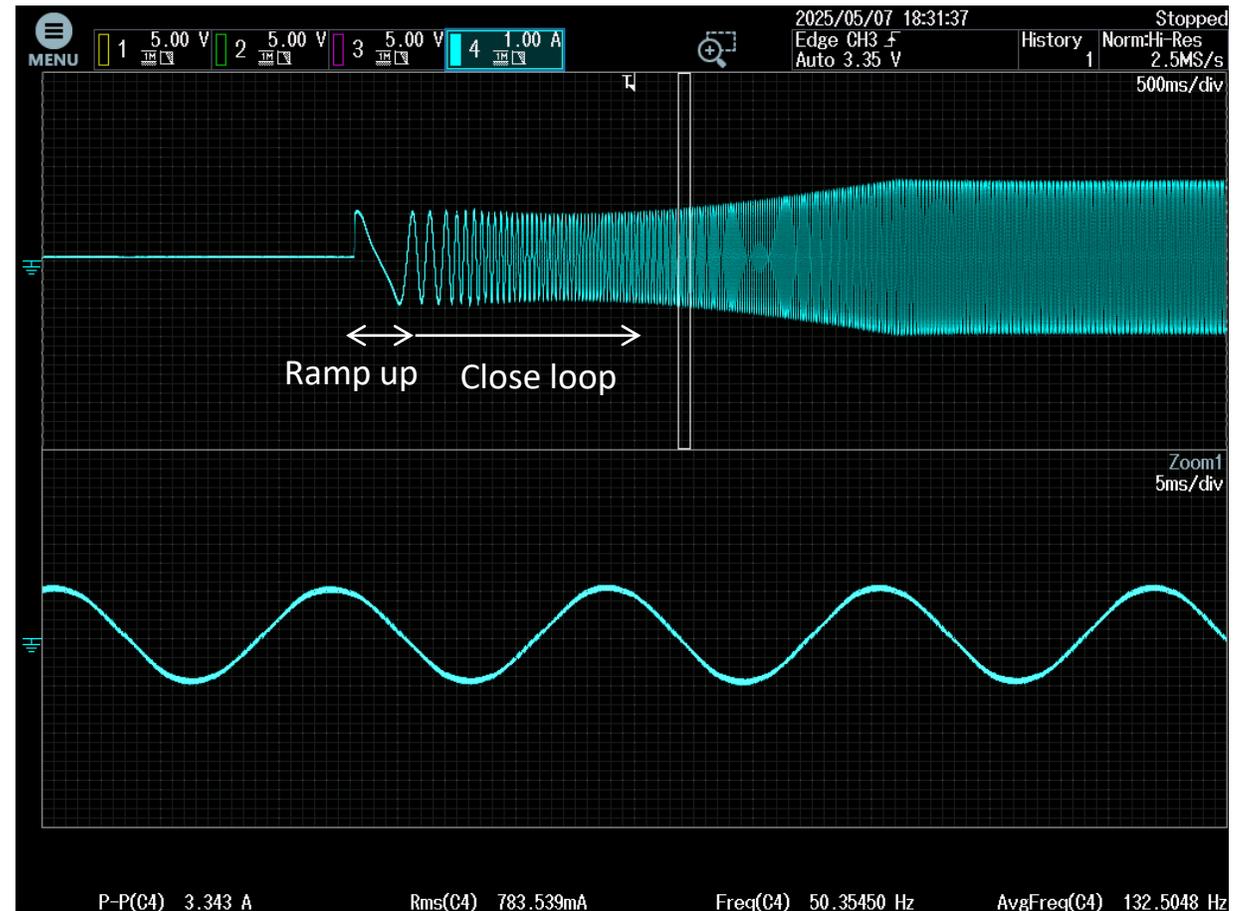


Startup method 1 - SYNC_RAMP

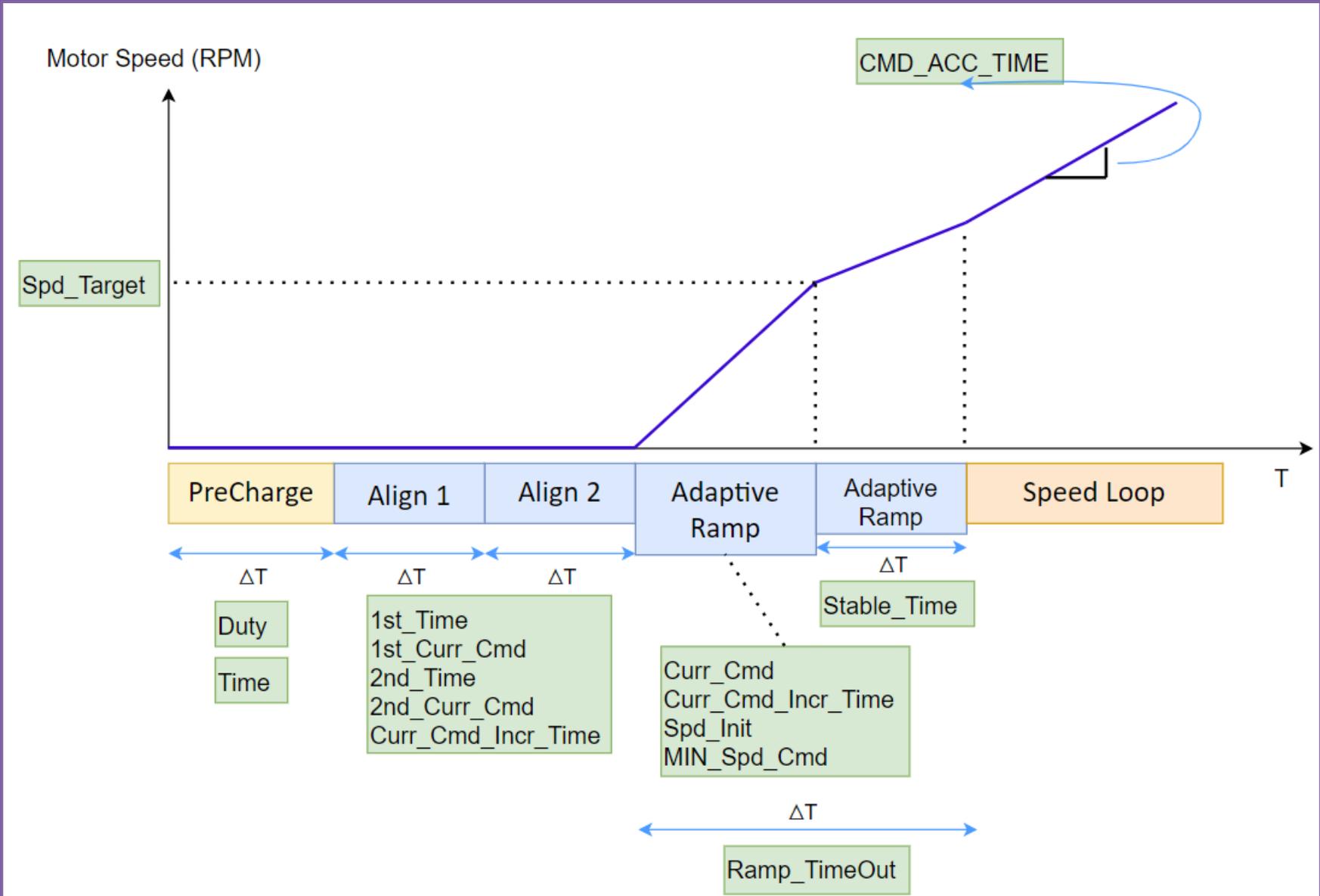


Startup method 2 - ADAPTIVE_RAMP

- ◆ Adaptive ramp up
 - Step 1 : Adaptive ramp up
 - Step 2 : Close loop



Startup method 2 - ADAPTIVE_RAMP



StartUp and Stop - Alignment

Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp	F
Num	Title	Unit	Value	Min	Typical	Max		
1	1st_Time	msec	200					
2	2nd_Time	msec	400					
3	Curr_Cmd_Incr_Time	msec	250					
4	1st_Curr_Cmd	A	0.3			5		
5	2nd_Curr_Cmd	A	0.3			5		

Parameters	Description
1st_Time	First pre-alignment time
2nd_Time	Second pre-alignment time
Curr_Cmd_Incr_Time	Current ramp-up time
1st_Curr_Cmd	First pre-alignment current command
2nd_Curr_Cmd	Second pre-alignment current command

StartUp and Stop - Ramp

Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp	Ramp
----------	-------	----	------	-------	-----------	-----------	--------------	------

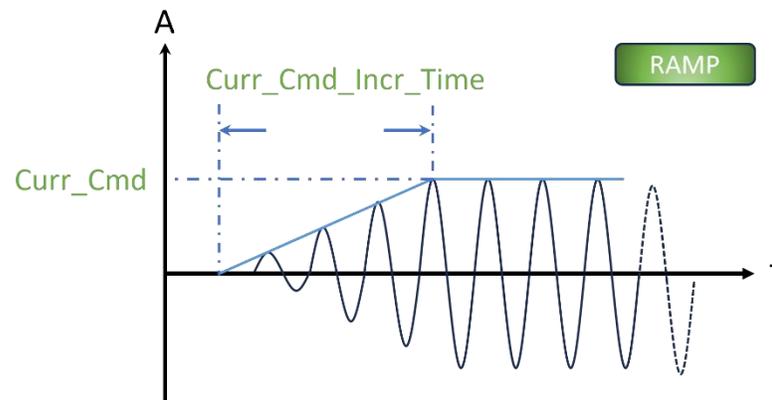
Num	Title	Unit	Value	Min	Typical	Max
1	Curr_Cmd	A	1			5
2	Curr_Cmd_Incr_Time	msec	200			
3	Stable_Time	msec	100			
4	Spd_Target	RPM	200		100	
5	Spd_Cmd_Incr_Time	msec	2000			

Parameters	Description
Curr_Cmd	Startup current command
Curr_Cmd_Incr_Time	Current ramp-up time
Stable_Time	Stabilization time
Spd_Target	Speed to transition into closed-loop
Spd_Cmd_Incr_Time	Time to accelerate from 0 RPM to Spd_Target

StartUp and Stop - AdaptiveRamp

Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp	R
Num	Title	Unit	Value	Min	Typical	Max		
1	Curr_Cmd	A	1			5.00		
2	Curr_Cmd_Incr_Time	msec	1					
3	Stable_Time	msec	20					
4	Spd_Target	RPM	300	40	100			
5	Spd_Init	RPM	50		10	5		
6	MIN_Spd_Cmd	RPM	20		10			

Parameters	Description
Curr_Cmd	Startup current command
Curr_Cmd_Incr_Time	Current ramp-up time
Stable_Time	Stabilization time
Spd_Target	Speed to transition into closed-loop
Spd_Init	Initial speed value
MIN_Spd_Cmd	Startup speed

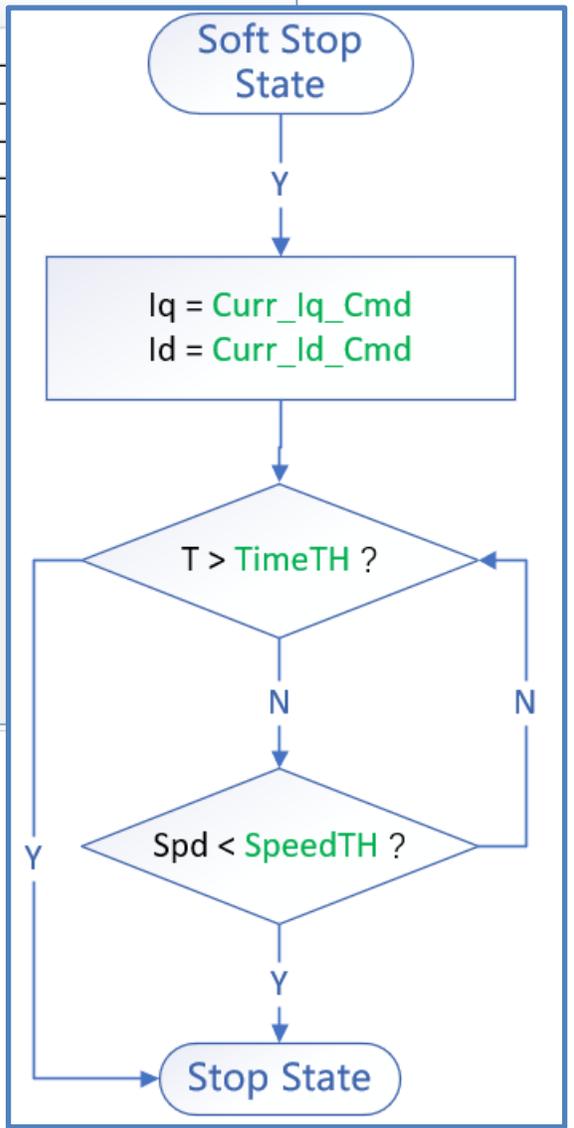


StartUp and Stop - Softstop

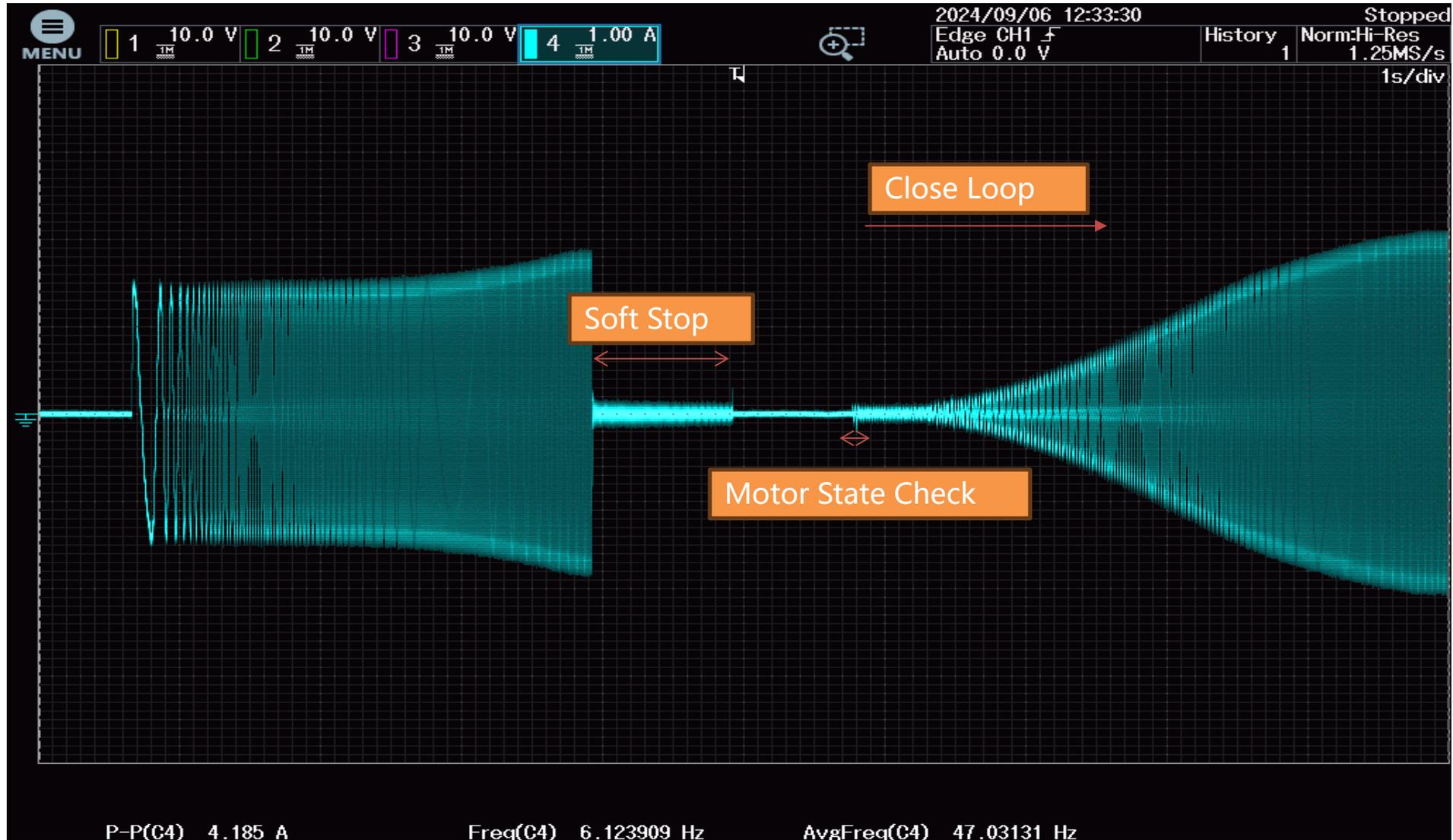
Function Motor HW User Alarm PreCharge Alignment AdaptiveRamp Ramp RunTrans Brake Current Loop Speed Loop BEMFOB Softstop BEMFD

Num	Title	Unit	Value	Min	Typical	Max
1	TimeTH	msec	100		100	
2	SpeedTH	RPM	300			
3	Curr_Id_Cmd	A	0		0	
4	Curr_Iq_Cmd	A	0		0	

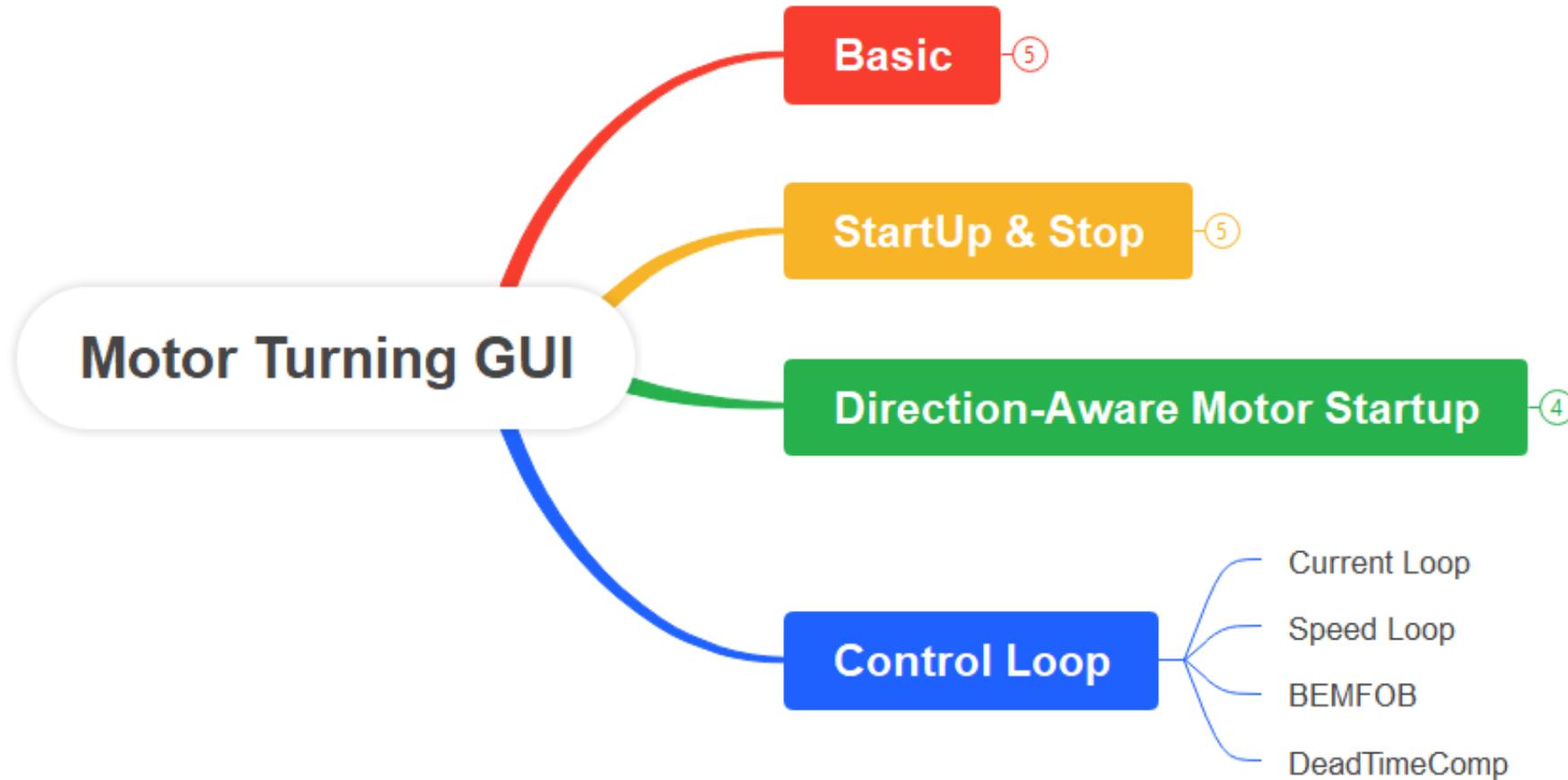
Parameters	Description
TimeTH	Soft stop maximum time
SpeedTH	Soft stop minimum speed
Curr_Id_Cmd	D-axis current command
Curr_Iq_Cmd	Q-axis current command



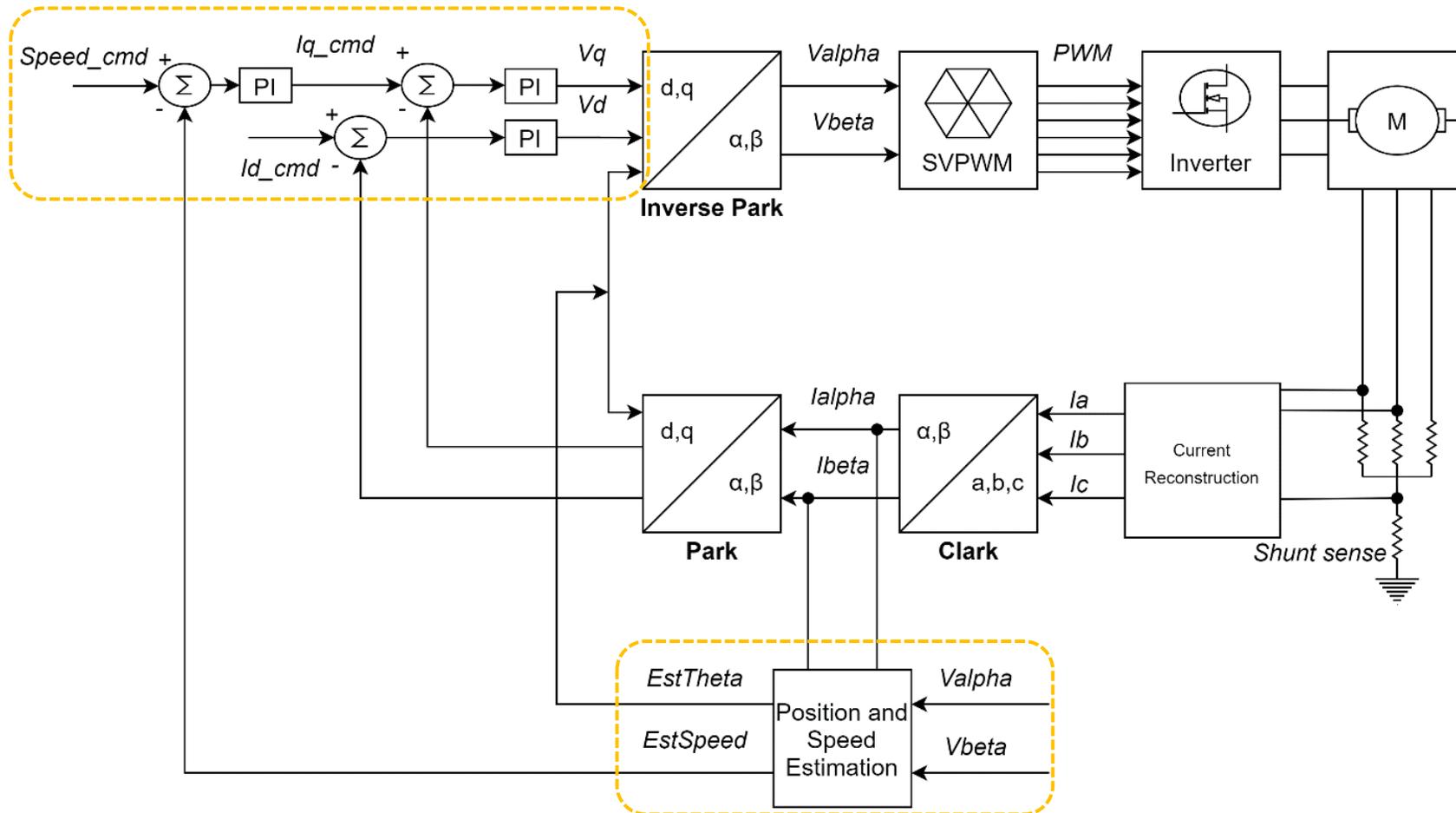
Tailwind start



Motor Tuning GUI - Control Loop



Multi-Loop Control Architecture



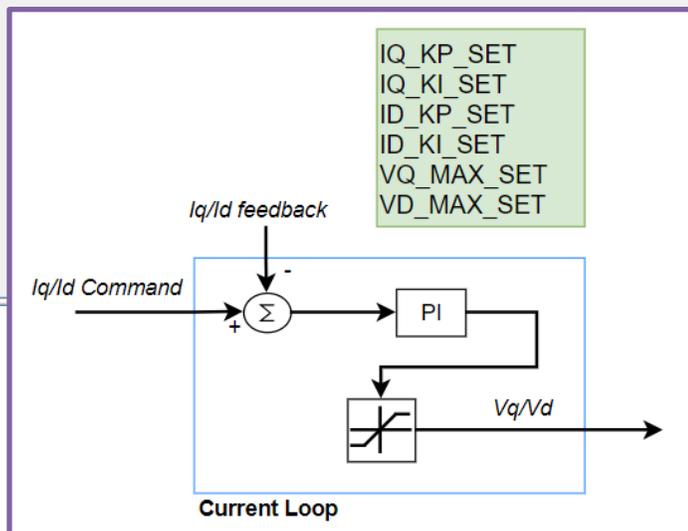
Control Loop – Current loop

Function Motor HW User Alarm PreCharge Alignment AdaptiveRamp Ramp RunTrans Brake **Current Lo**

AutoFunction: enable disable

Num	Title	Unit	Value	Min	Typical	Max
1	wc	freq	2	0.1	3	10
2	IQ_KP_SET	0~1	0.5628858616814307		0.5628858616814307	
3	IQ_KI_SET	0~1	0.03818024255591099		0.03818024255591099	
4	ID_KP_SET	0~1	0.4974340172998689		0.4974340172998689	
5	ID_KI_SET	0~1	0.03818024255591099		0.03818024255591099	
6	VQMAX_SET	%	100			100
7	VQMIN_SET	%	-100			-100
8	VDMAX_SET	%	100			100
9	VDMIN_SET	%	-100			-100

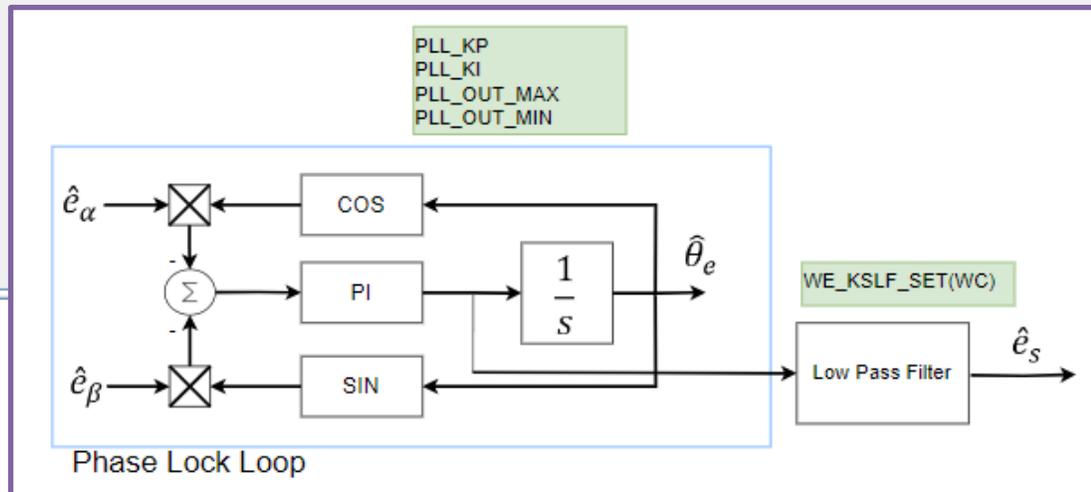
Parameters	Description
AutoFunction	Enable/disable auto-tuning
wc	Cutoff frequency ratio
IQ_KP_SET	Q-axis current loop KP
IQ_KI_SET	Q-axis current loop KI
ID_KP_SET	D-axis current loop KP
ID_KI_SET	D-axis current loop KI
VQMAX_SET	Advanced parameters (do not modify)
VQMIN_SET	Advanced parameters (do not modify)
VDMAX_SET	Advanced parameters (do not modify)
VDMIN_SET	Advanced parameters (do not modify)



Control Loop - EMFOB

Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp	Ramp	RunTrans	Brake	Current Loop	Speed Loop	BEMFOE
Num	Title	Unit	Value	Min	Typical	Max							
1	BEMFOB_GAIN_R		0.14583778394929858										
2	BEMFOB_GAIN_L		0.00026875820184942174										
3	BEMFOB_GAIN_W		1.533980787885641										
4	WE_KSLF_SET(WC)		1	0.5	3								
5	PLL_KP		0.005			1							
6	PLL_KI		0.0001			1							
7	PLL_OUT_MAX	%	80		80								
8	PLL_OUT_MIN	%	-80		-80								
9	BEMFOB_GAIN_L_CAILk		1.5	0.5	1	2							

Parameters	Description
BEMFOB_GAIN_R	Observer R gain
BEMFOB_GAIN_L	Observer L gain
BEMFOB_GAIN_W	Observer W gain
WE_KSLF_SET(WC)	Speed estimation low-pass filter cutoff frequency
PLL_KP	Phase-locked loop KP
PLL_KI	Phase-locked loop KI
PLL_OUT_MAX	Phase-locked loop maximum output
PLL_OUT_MIN	Phase-locked loop minimum output
BEMFOB_GAIN_L_CAILk	---



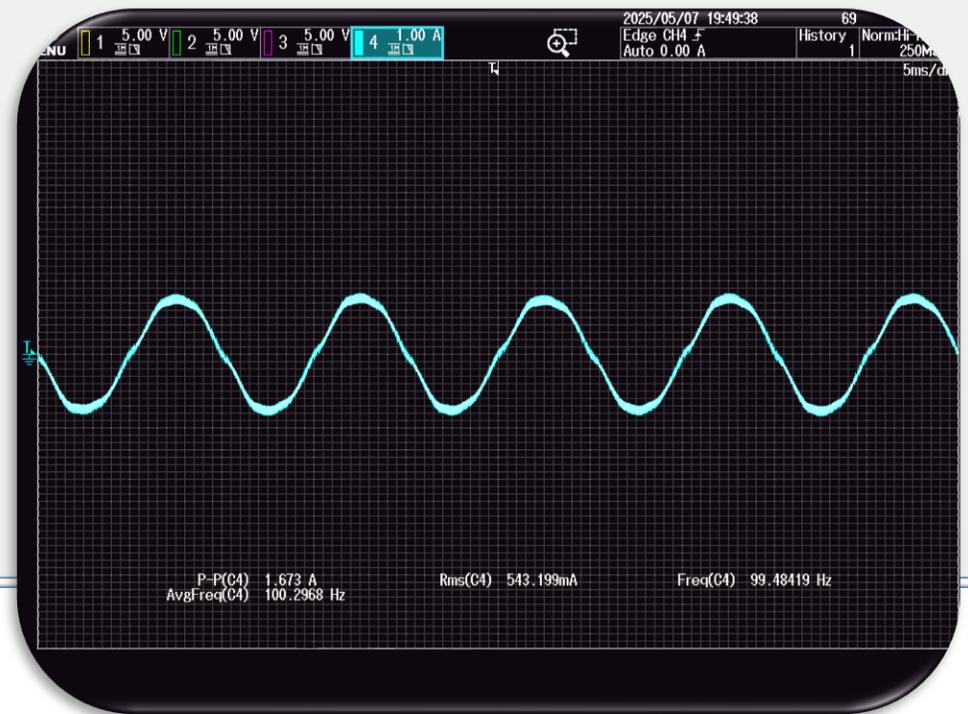
Control Loop - DeadTimeComp

Function Motor HW User Alarm PreCharge Alignment AdaptiveRamp Ramp RunTrans Brake Current Loop Speed Loop BEMFOB Softstop BEMFDet DeadTimeComp

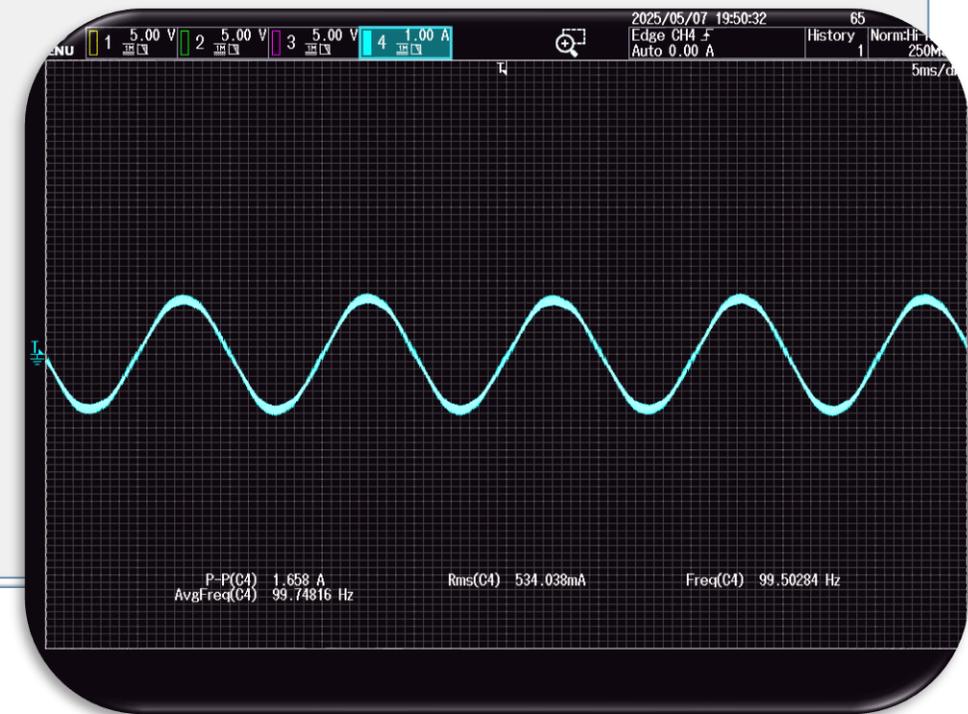
Num	Title	Unit	Value	Min	Typical	Max	
1	COMP_SET	%	100		100		

Parameters	Description
COMP_SET	Dead time compensation setting

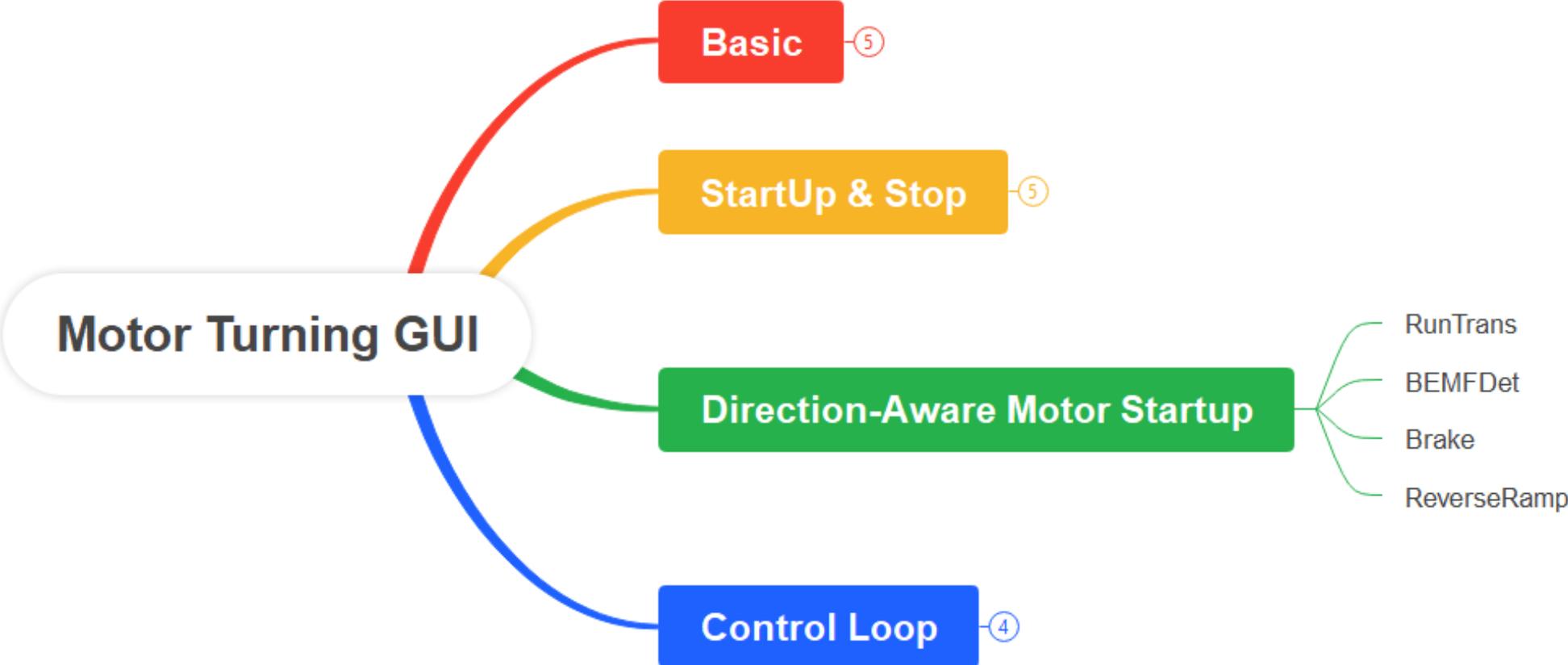
#define DEADTIME_



COMP_SET = 0%



COMP_SET = 100%

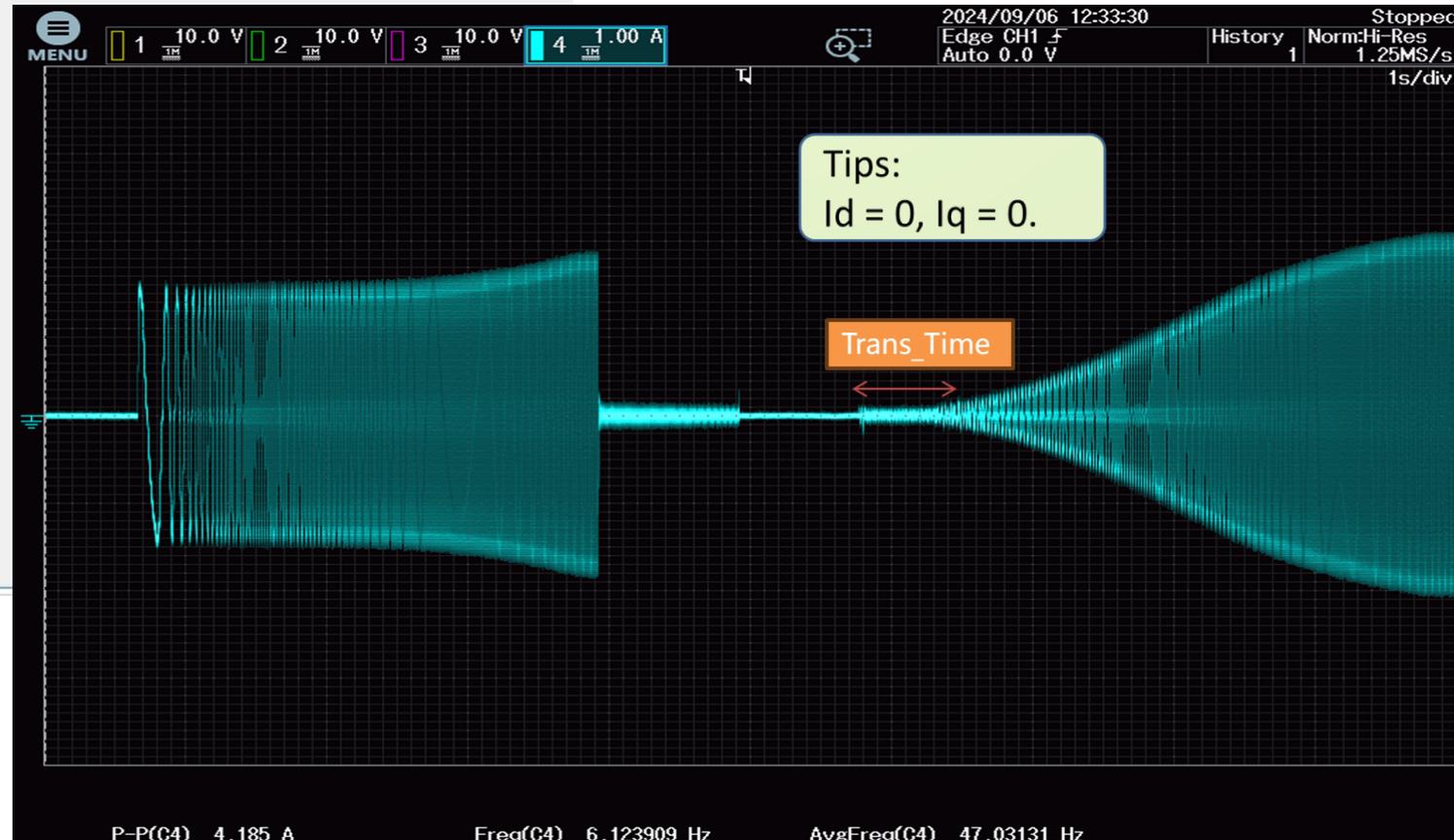


StartUp and Stop - RunTrans

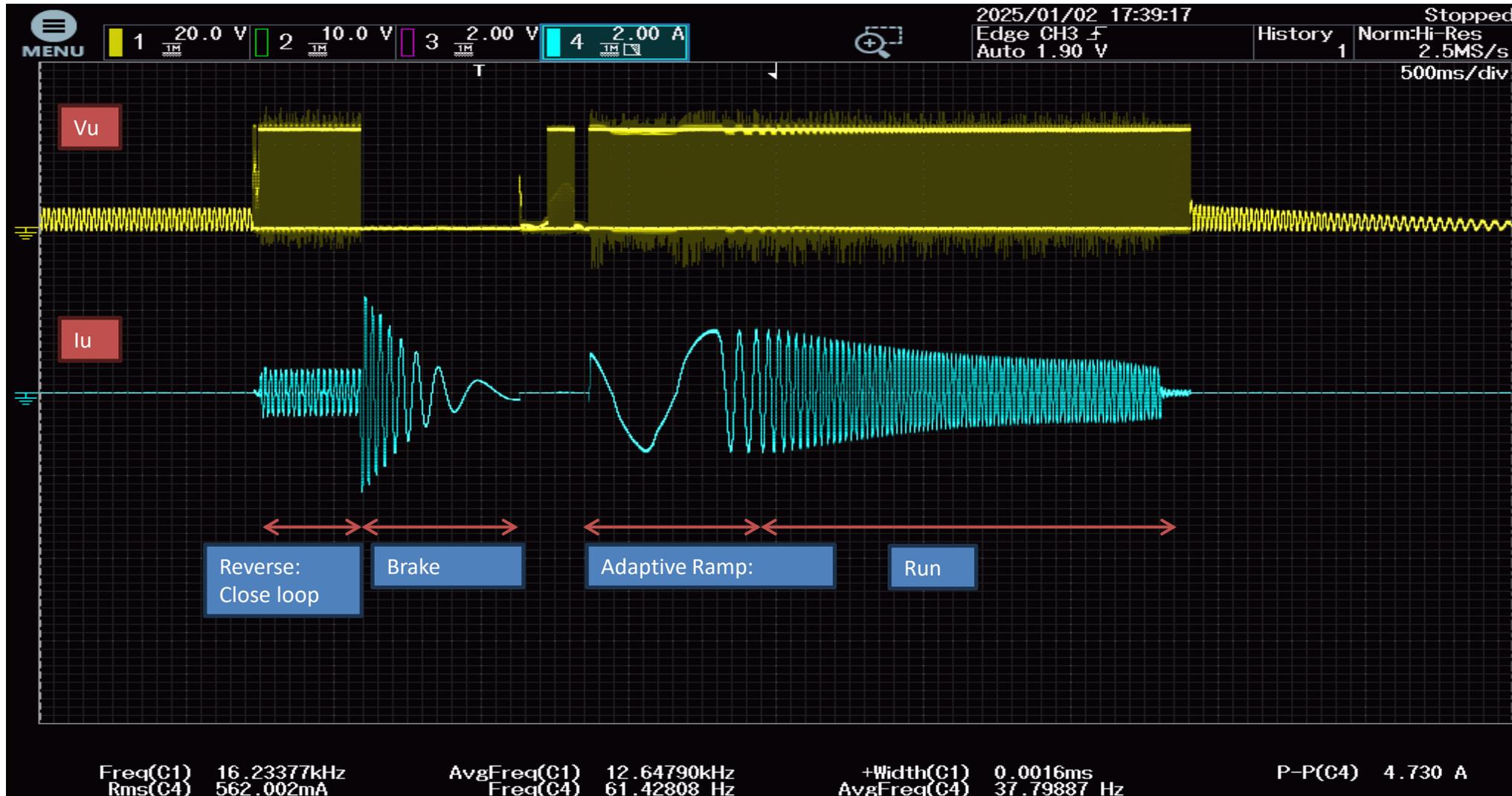
Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp	Ramp	RunTrans
----------	-------	----	------	-------	-----------	-----------	--------------	------	----------

Num	Title	Unit	Value	Min	Typical	Max
1	Trans_Time	msec	100			

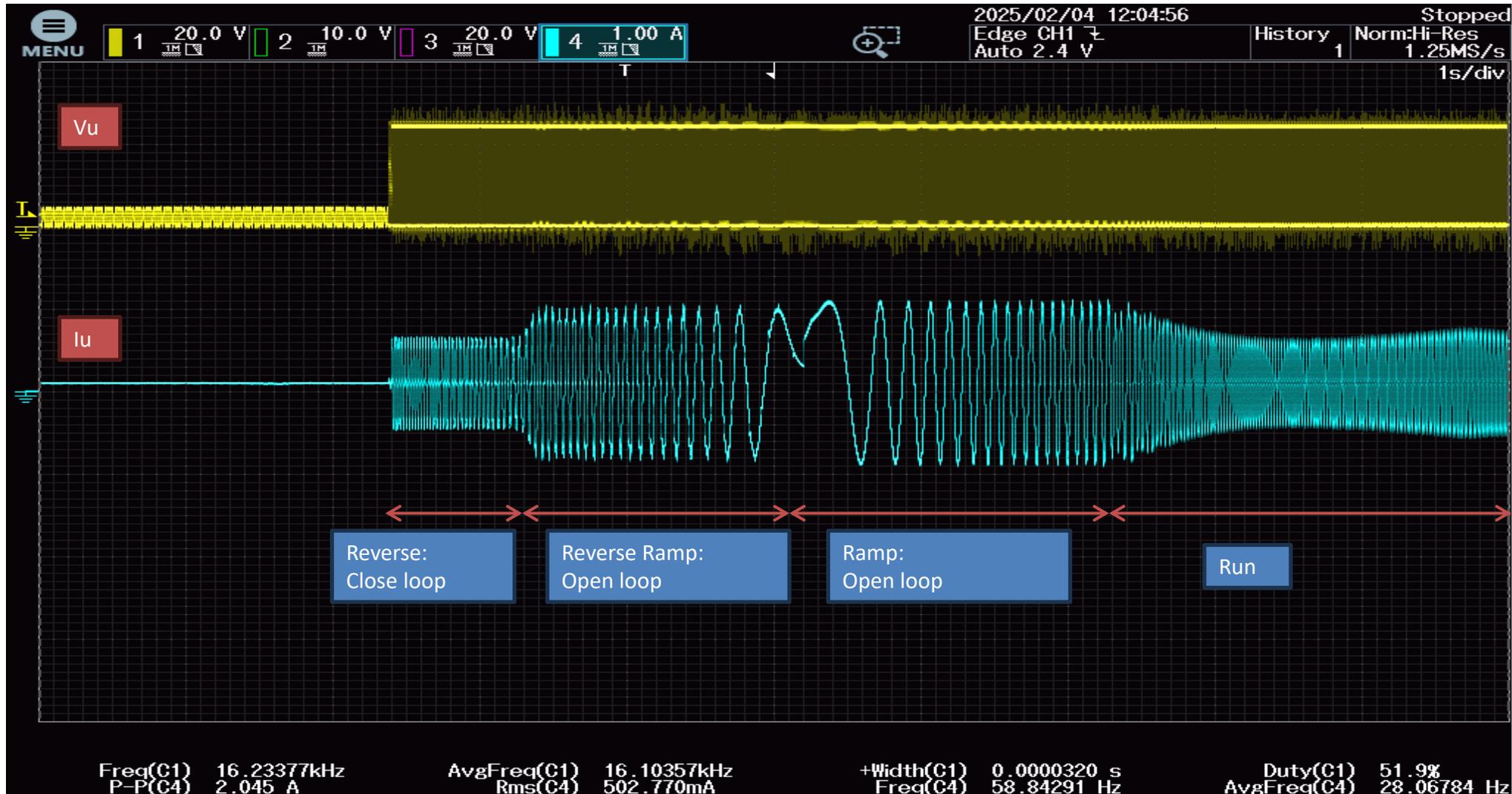
Parameters	Description
Trans_Time	Transition instantaneous stabilization time



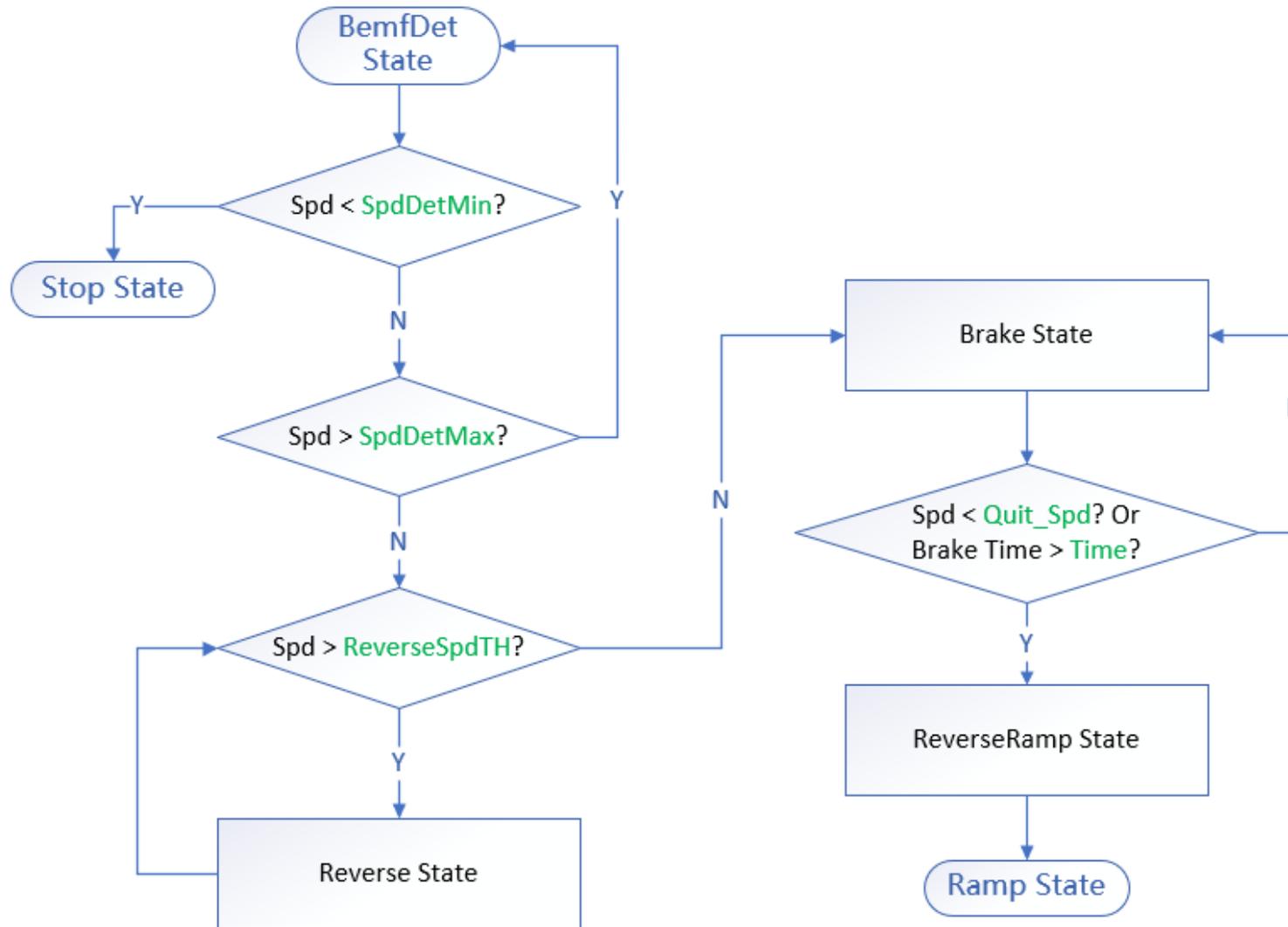
Headwind Startup - 1



Headwind Startup - 2



Headwind Startup – 2 , Flowchart



StartUp and Stop – BEMFDet & Reverse

Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp	Ramp	RunTrans	Brake	Current Loop	Speed Loop	BEMFOB	Softstop	BEMFDet
----------	-------	----	------	-------	-----------	-----------	--------------	------	----------	-------	--------------	------------	--------	----------	---------

Num	Title	Unit	Value	Min	Typical	Max	
1	SpdDetMin	%(Motor_MaxSpd)	10		5	20	#d
2	SpdDetMin_Margain	Times	3		3		#d
3	SpdDetMax	%(Motor_MaxSpd)	80	50	75	100	#d
4	InitDuty	%(MaxSpd to duty)	70		70		
5	ReverseSpdTH	%(Motor_MaxSpd)	15		10		
6	ReverseCurr_Id_Cmd	A	0.5				
7	ReverseCurr_Iq_Cmd	A	0.25				

Parameters	Description
SpdDetMin	Below this speed is considered stopped
SpdDetMin_Margain	Minimum detection speed cycle , this value is timeout duration
SpdDetMax	Above this speed, re-detect
InitDuty	Do not modify
ReverseSpdTH	Above this speed enters closed-loop braking, below this speed enters short-circuit braking
ReverseCurr_Id_Cmd	D-axis reverse current command
ReverseCurr_Iq_Cmd	Q-axis reverse current command

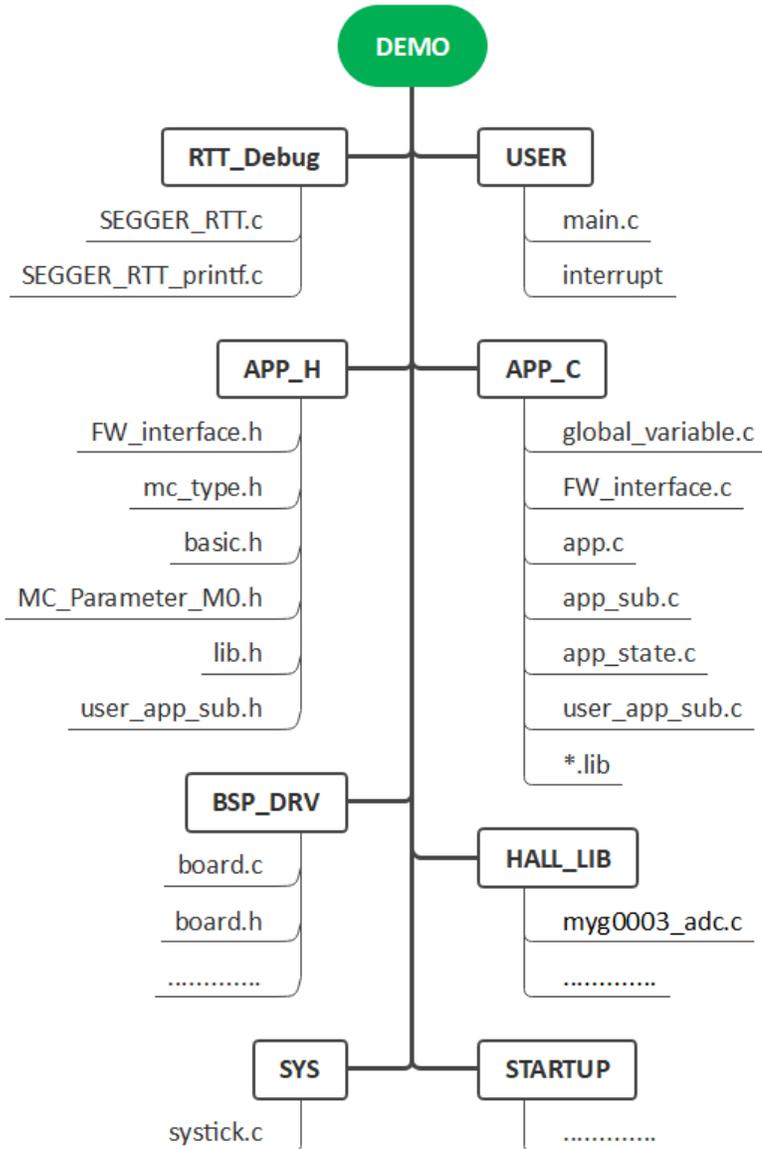
Tips:
Reverse state is close loop control.

Note - example

Load Parameters		Save as Parameters		Export Header File		Save as Header File		About										
Function	Motor	HW	User	Alarm	PreCharge	Alignment	AdaptiveRamp	Ramp	RunTrans	Brake	Current Loop	Speed Loop	BEMFOB	Softstop	BEMFDet	DeadTimeComp	ReverseRamp	Note

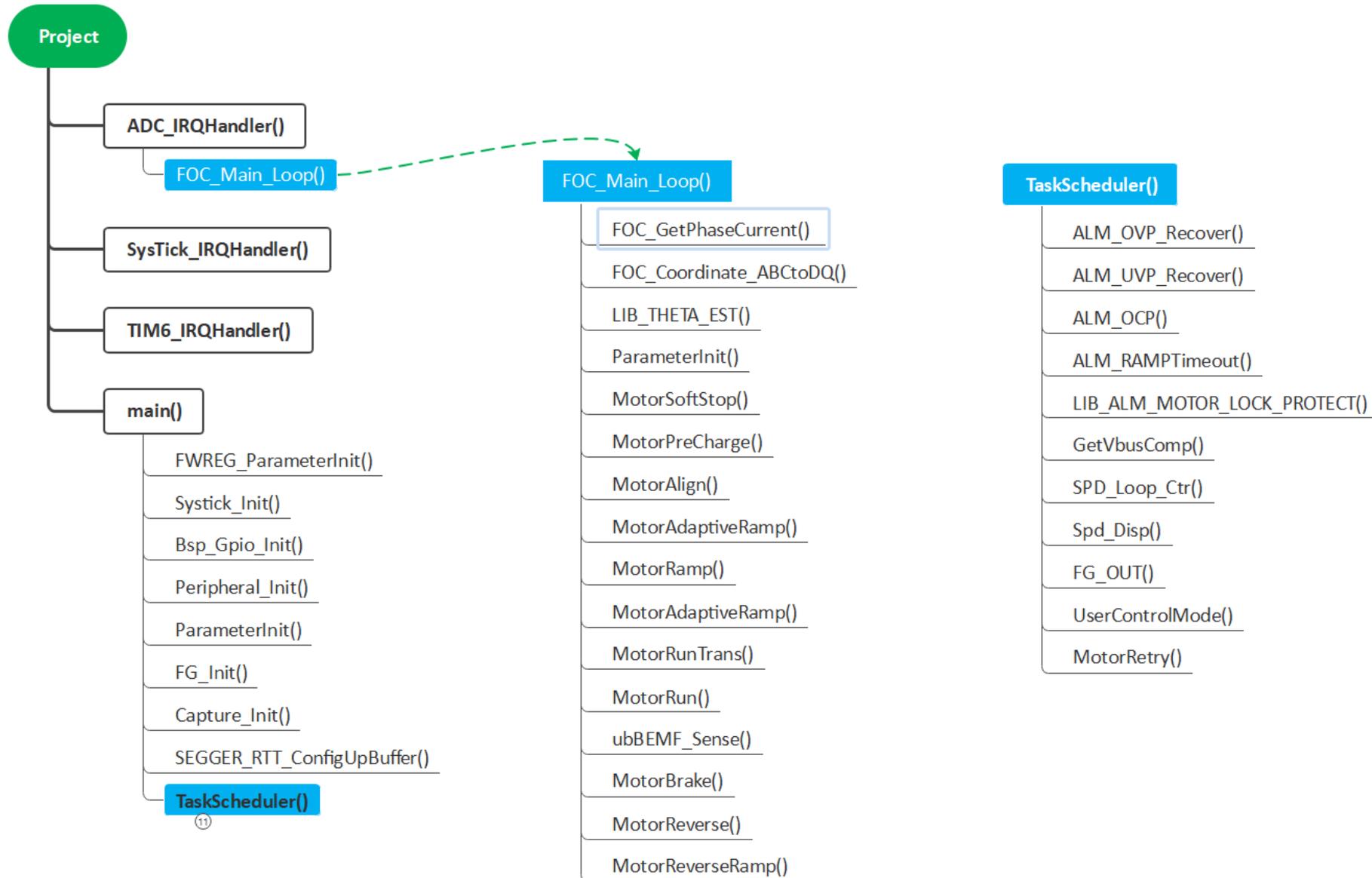
1. Date:
2. Customer Name:
3. Project / Product Name:
4. Motor Model / Part Number:
5. Engineer Name:

Project Folder Structure



Folder	File	Description
RTT_Debug	SEGGER_RTT.c	Jscope debug
	SEGGER_RTT_printf.c	Jscope debug
USER	main.c	main source file
	interrupt	Interrupt handling
APP_C	global_variable.c	Global variable definitions
	FW_interface.c	Firmware interface declarations
	app.c	Main app logic
	app_sub.c	App sub-functions
	app_state.c	motor control state machine
	user_app_sub.c	User-defined app sub-functions
	*.lib	
APP_H	FW_interface.h	Firmware interface functions
	mc_type.h	Motor control types(do not modify)
	basic.h	Common macros & defs
	MC_Parameter_M0.h	GUI parameters
	lib.h	
	user_app_sub.h	User-defined app sub-functions
BSP_DRV	board.c	Board-level init code
	board.h	Board-level defs
	
HALL_LIB	
SYS	systick.c	System tick timer
STARTUP	

Project Structure



- ◆ Motor Retry
- ◆ Interface setting: VSP/PWM
- ◆ Control mode: Speed Loop/ Power Loop
- ◆ FG function

```
user_app_sub.h
1 #include "mc_type.h"
2 #include "global_variable.h"
3 #include "mc_parameter_M0.h"
4 #include "FW_interface.h"
5 #include "myg0003_conf.h"
6
7 #define P_DEBUG_RTT_EN          (1) //Debug RTT Mode, 1:Enable, 0:Disable
8 #define P_VREF                  (5) //V
9
10 //Retry
11 #define RETRY_TIMES             (0xFF) //0xff is unlimited retry
12 #define RETRY_DELAY_TIME       (10) //unit:500ms
13
14 //Interface setting
15 #define P_CONTROL_DEBUG        (0) //Modification is prohibited
16 #define P_CONTROL_VSP         (1) //Modification is prohibited
17 #define P_CONTROL_DUTY        (2) //Modification is prohibited
18 #define P_CONTROL_MODE        P_CONTROL_VSP
19
20 //Control mode
21 #define P_LOOP_SPD             (0) //Modification is prohibited
22 #define P_LOOP_POWER          (1) //Modification is prohibited
23 #define P_LOOP_MODE           P_LOOP_SPD
24
25 #define P_FG_FUNC_EN           (1) //FG function enable
26 #define P_FG_GAIN              ((float)(5.0))
27 #define P_FG_Clock              ((float)(1000000)) //Modification is prohibited
28 #define P_FG_GAIN_DEC_PARA     (float)(P_FG_Clock/(P_FG_GAIN*1/60.0))//Modification is prohibited
29
```

user_app_sub.h – Motor Retry

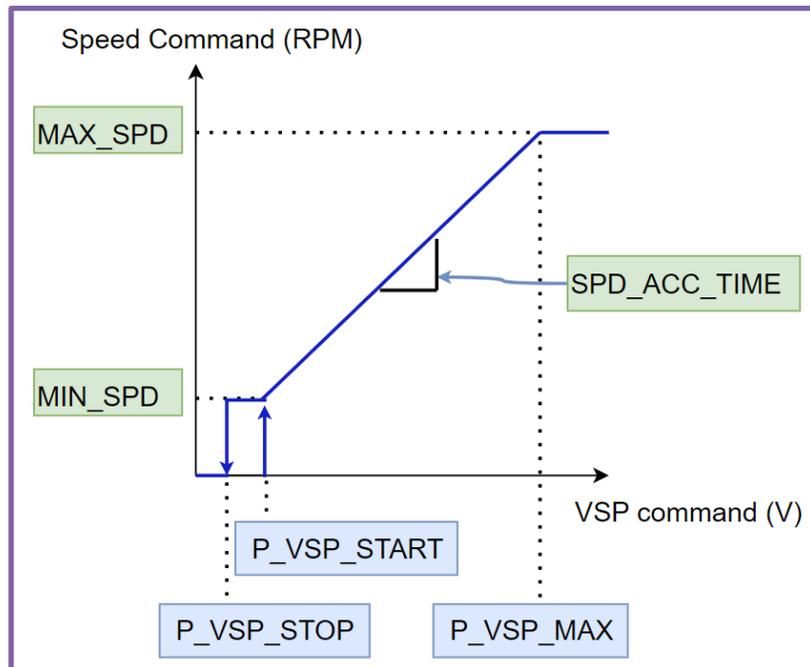
Parameters	Value	Unit	Description
RETRY_TIMES	0~255	Times	Restart count after an error
RETRY_DELAY_TIME	0~65535	500ms	The delay between each retry attempt

Tips:

RETRY_TIMES = 255 → Retry indefinitely

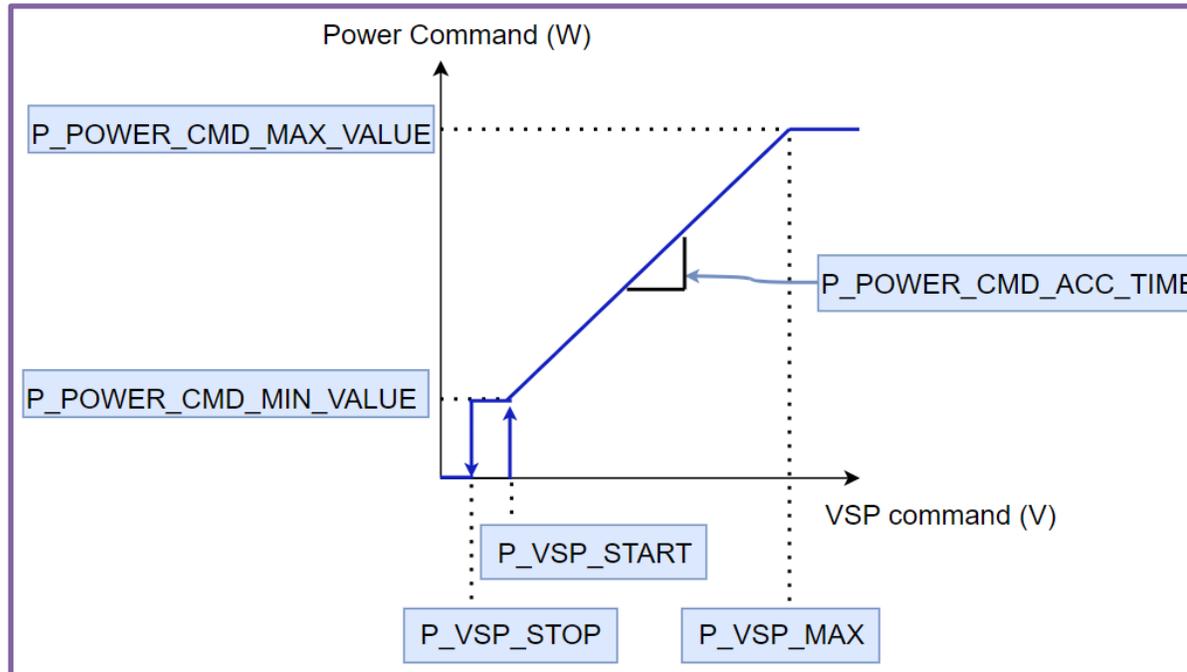
user_app_sub.h – Interface setting

Parameters	Value	Unit	Description
			For VSP, the input voltage level is used as the control command.
P_CONTROL_MODE	P_CONTROL_VSP, P_CONTROL_PWM		For PWM, the input PWM duty cycle is used as the control command.
P_VSP_STOP	0~VDD	V	Stop voltage
P_VSP_START	0~VDD	V	Startup threshold
P_VSP_MAX	0~VDD	V	Maximum voltage
P_PWM_STOP	0~1000	0.1%	Stop duty cycle
P_PWM_START	0~1000	0.1%	Startup threshold
P_PWM_MAX	0~1000	0.1%	Maximum duty cycle



user_app_sub.h – Control mode - 1

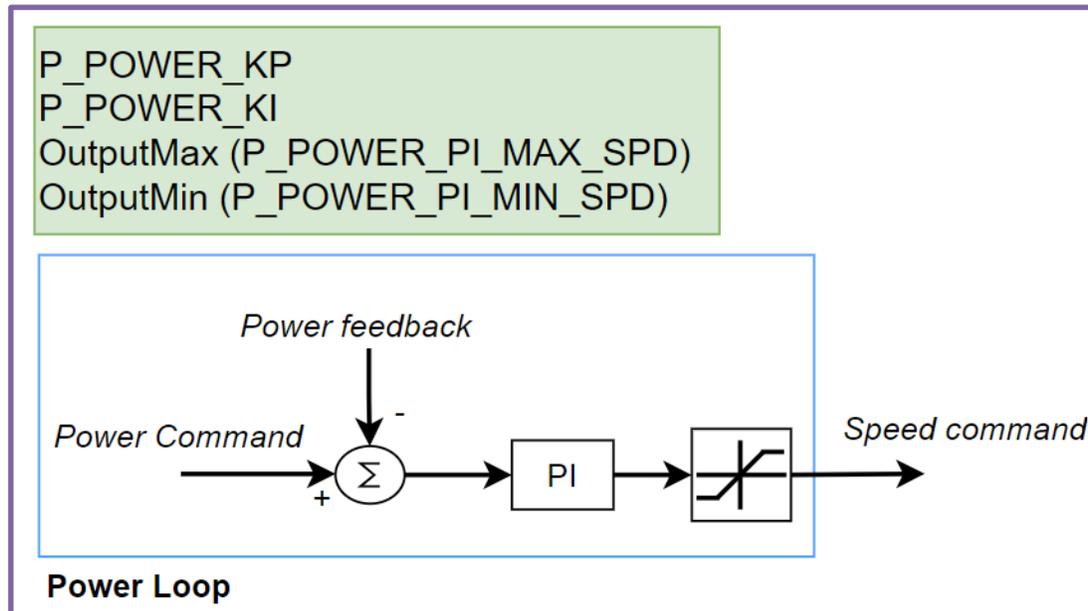
Parameters	Value	Unit	Description
P_LOOP_MODE	P_LOOP_SPD, P_LOOP_POWER		P_LOOP_SPD : Speed loop control. P_LOOP_POWER: Bus power control
P_POWER_IBUS_PU			Calculated as $VDD / (R_{shunt} \times I_{GAIN})$
P_POWER_VBUS_PU			Calculated as VDD / V_{GAIN}
P_POWER_POWER_PU_GAIN	2		Modification is prohibited
P_POWER_CMD_MAX_VALUE		W	Maximum power command
P_POWER_CMD_MIN_VALUE		W	Minimum power command
P_POWER_CMD_ACC_TIME		sec	Ramp time from minimum to maximum power



Tips:
The power control feature is currently not in use and has not been enabled yet.

user_app_sub.h – Control mode - 2

Parameters	Value	Unit	Description
P_POWER_KP	0~1		Proportional (KP) parameters
P_POWER_KI	0~1		Integral (KI) parameters
P_POWER_PI_MAX_SPD	<MaxSpd	RPM	Maximum output of the PI controller
P_POWER_PI_MIN_SPD	>MinSpd	RPM	Minimum output of the PI controller



Tips:

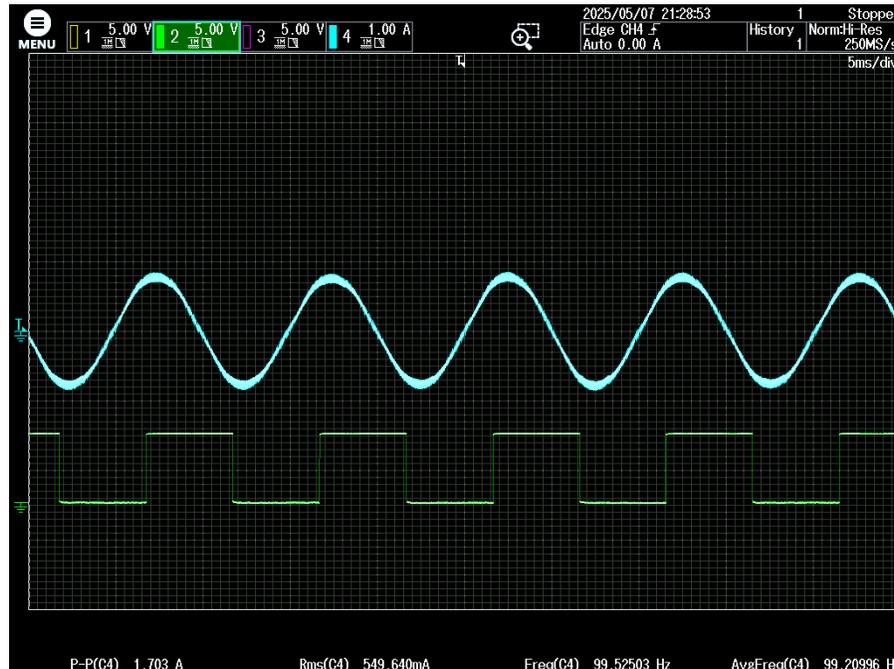
The power control feature is currently not in use and has not been enabled yet.

user_app_sub.h – FG function

Parameters	Value	Unit	Description
P_FG_FUNC_EN	0,1		0: Disable 1: Enable
P_FG_GAIN		Pole pairs	Number of square waves per electrical cycle: calculated as (P_FG_GAIN) / (pole pairs)

Tips:
The motor has 5 pole pairs.

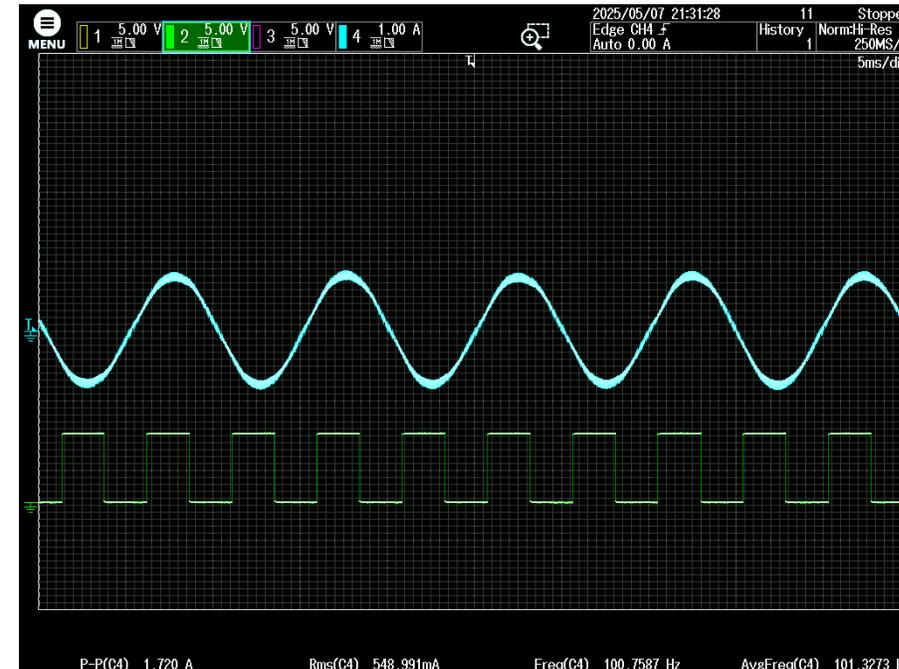
Tips:
FG (Frequency Generator).



P_FG_GAIN = 5

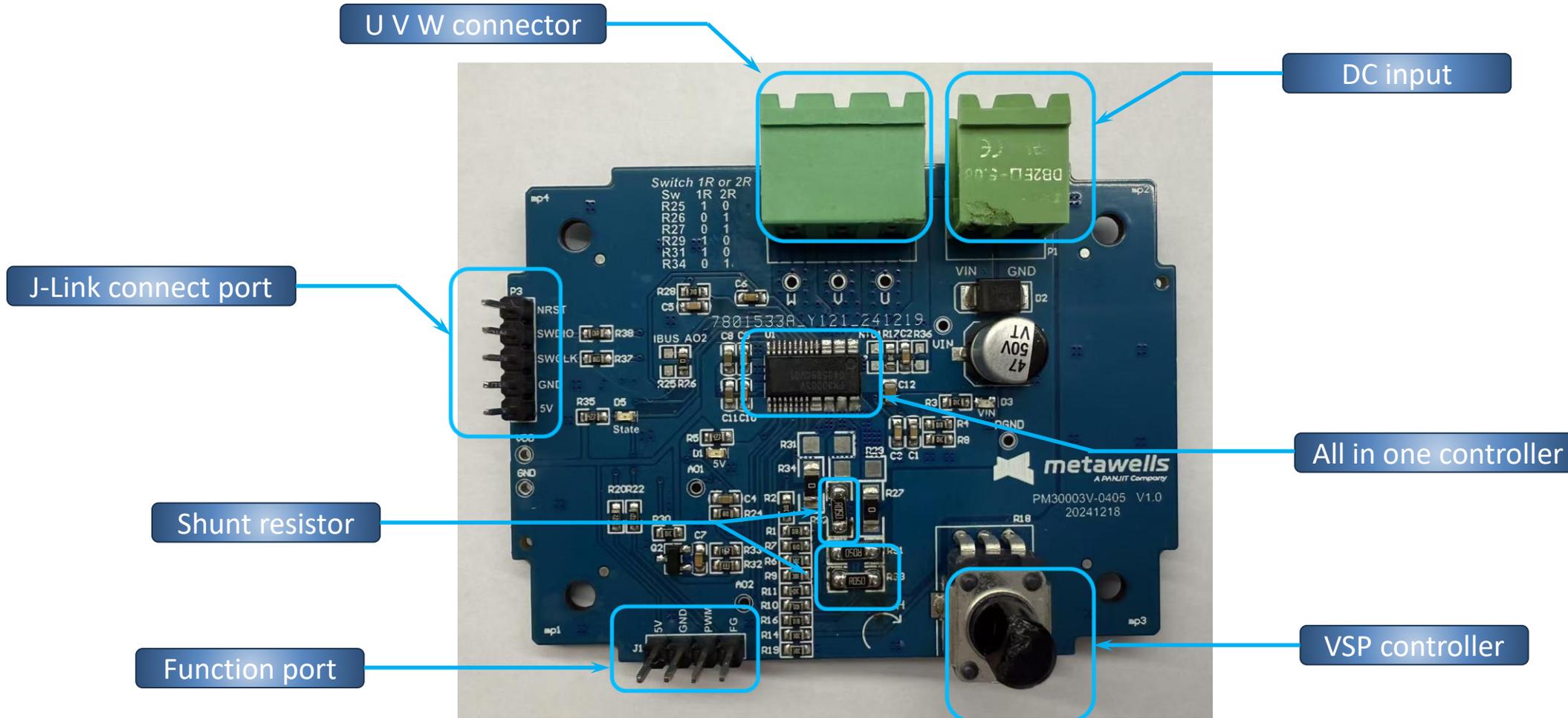
Iu

FG



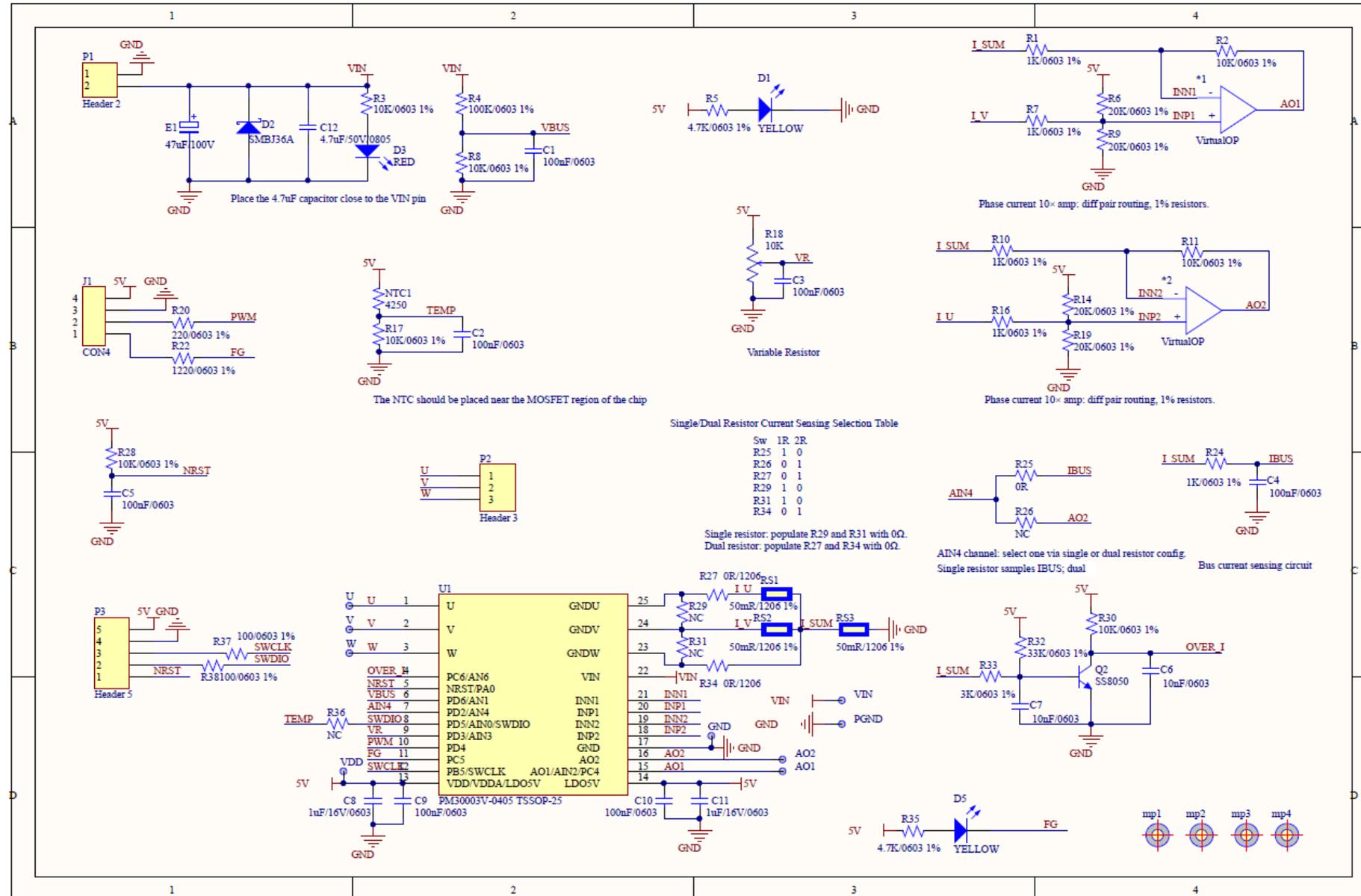
P_FG_GAIN = 10

EVB – PM30003V-0405



Tips:
Standard interfaces: PWM and FG.

Schematic



Monitoring values online

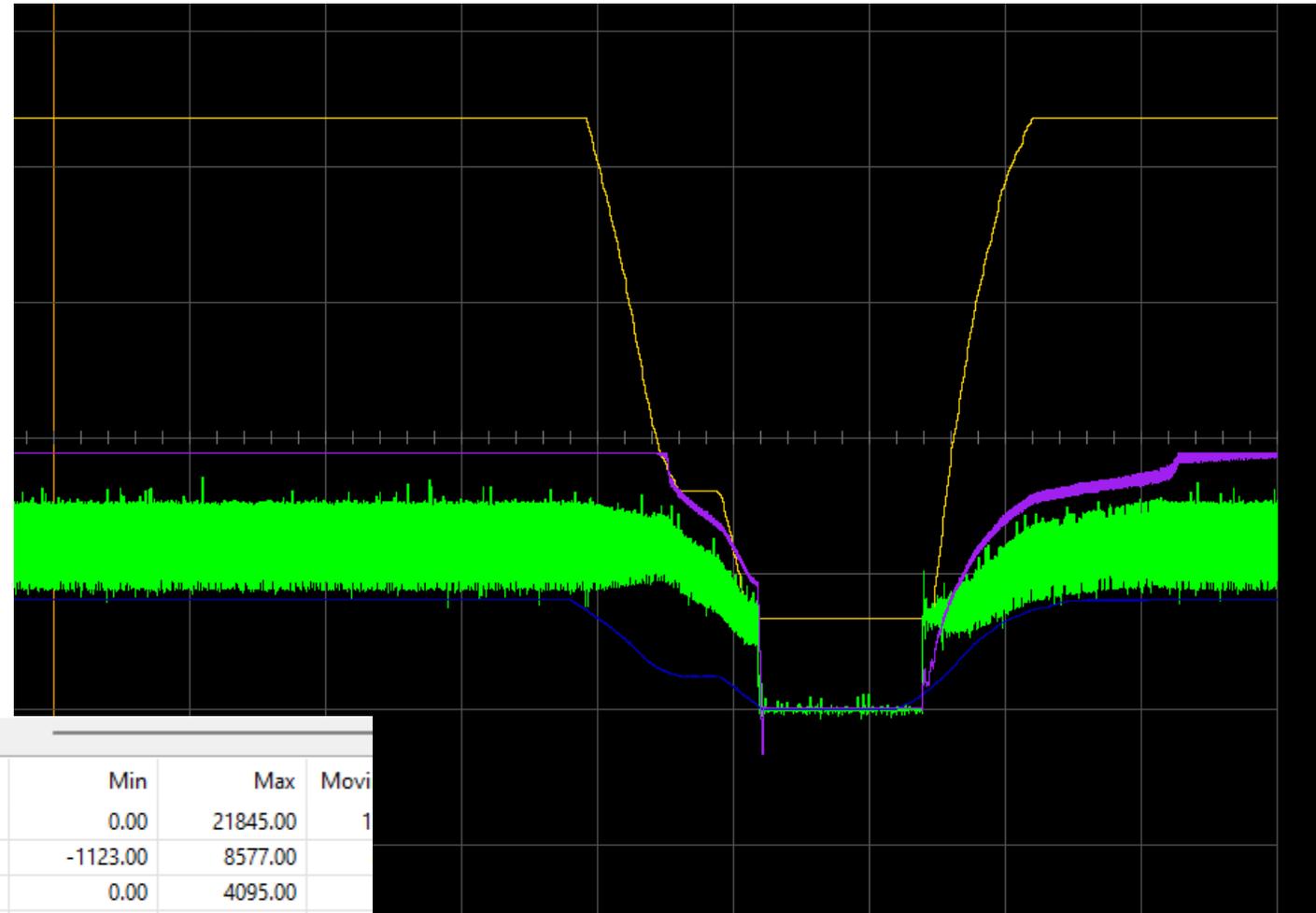
Tips:
Use J-Scope to monitoring values from program for helping analyze problems.



Start/Resume

Stop

Pause



Name	Address	Size	Type	Value	Color	Min	Max	Movi
✓ Symbol ...	0x00000000	2 Bytes	2 Byte unsi...	5696.00	#FFD700	0.00	21845.00	1
✓ Symbol ...	0x00000000	2 Bytes	2 Byte sign...	3595.00	#00FF00	-1123.00	8577.00	
✓ Symbol ...	0x00000000	2 Bytes	2 Byte sign...	823.00	#0000B0	0.00	4095.00	
✓ Symbol ...	0x00000000	2 Bytes	2 Byte sign...	7795.00	#A020F0	-3369.00	18918.00	1



Your Best Partner Our Profession