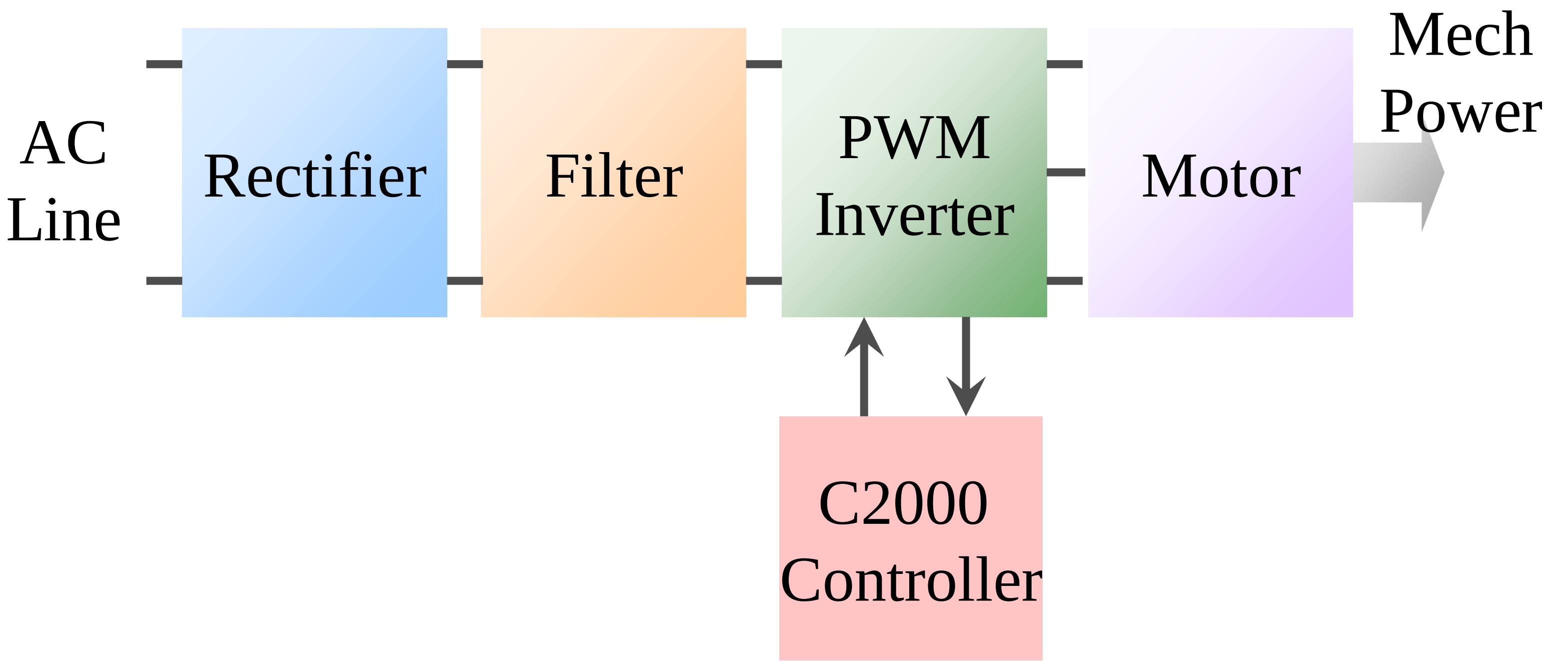
Implementing Digital Motor Control with C2000

Texas Instruments

Power Flow Block Diagram



- Power is transferred from the AC line to the motor through the rectifier, and inverter.
- The C2000 controls the flow of power from the AC line to the motor.

Content

- Motor Control Methods Overview
- Inverter / Hardware considerations
- Software / Algorithm Considerations
- DMC Development Kits
- System Incremental Build
- Demo, Q&A

Content

- Motor Control Methods Overview
- Inverter / Hardware considerations
- Software / Algorithm Considerations
- DMC Development Kits
- System Incremental Build
- Demo, Q&A

The "Ideal" Motor Control

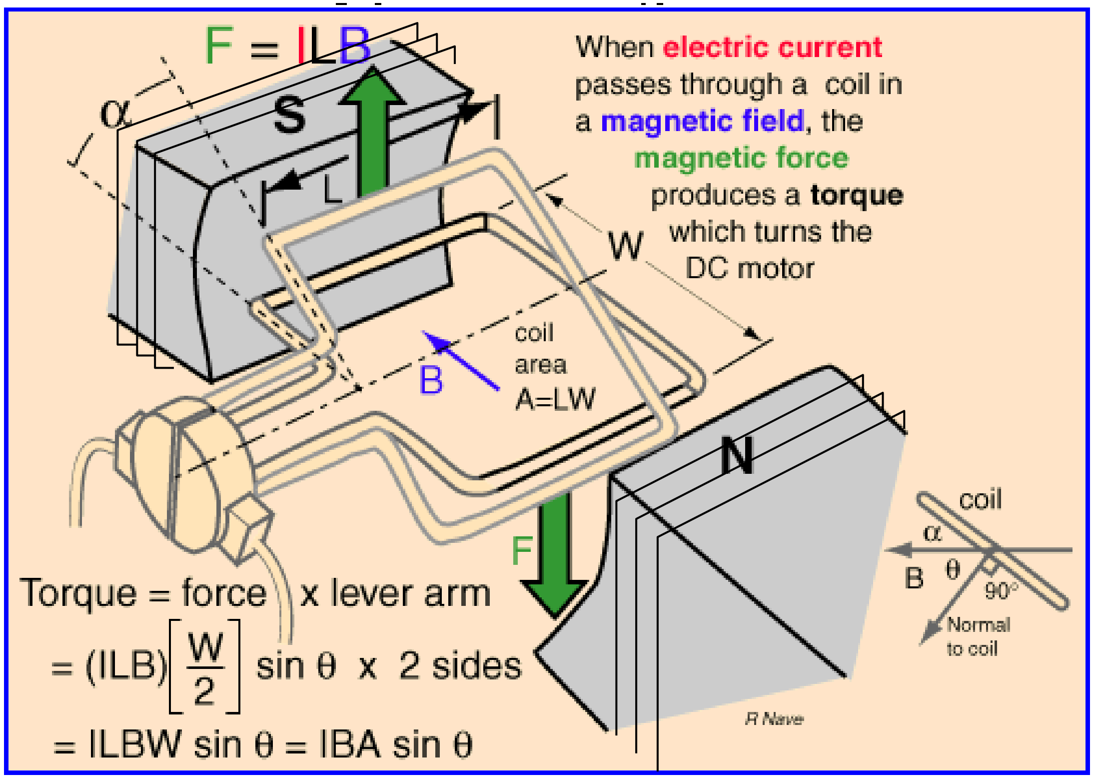
- Achieve maximum torque at every speed
- Good transient control (currents, speed)
- Efficient control

- Low EMI
- Low electrical network pollution (harmonics)
- No reactive power (power factor correction)
- Low acoustical noise level

- Board cost
- Development time

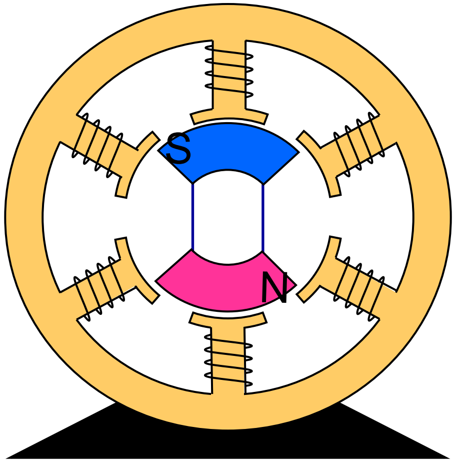
Basic Principles of DC Motors

Torque in DC Motor



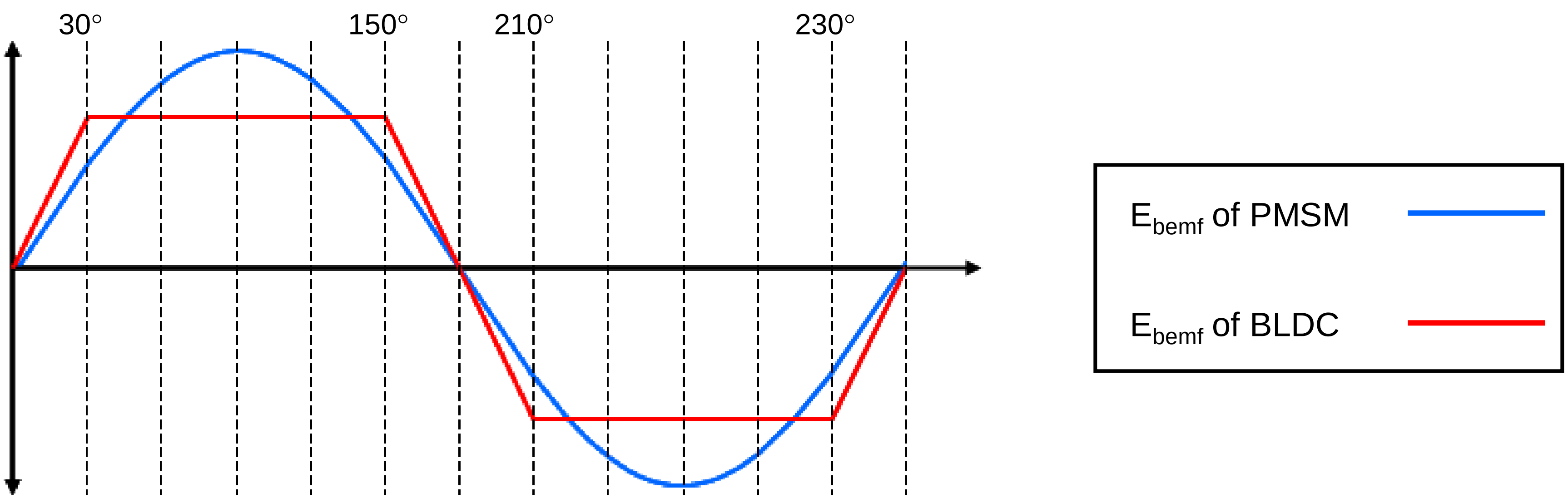
Synchronous Motors: PMSM & BLDC

- From the previous slides we know that if we excite the three stator coils with three-phase voltages we will get a rotating magnetic field at the centre
- All we now have to do to invent our PMSM (or BLDC) is to pivot a permanent magnet at the centre



- The rotor will always rotate at exactly the same speed as the stator, which is why this type of machine is called "synchronous"

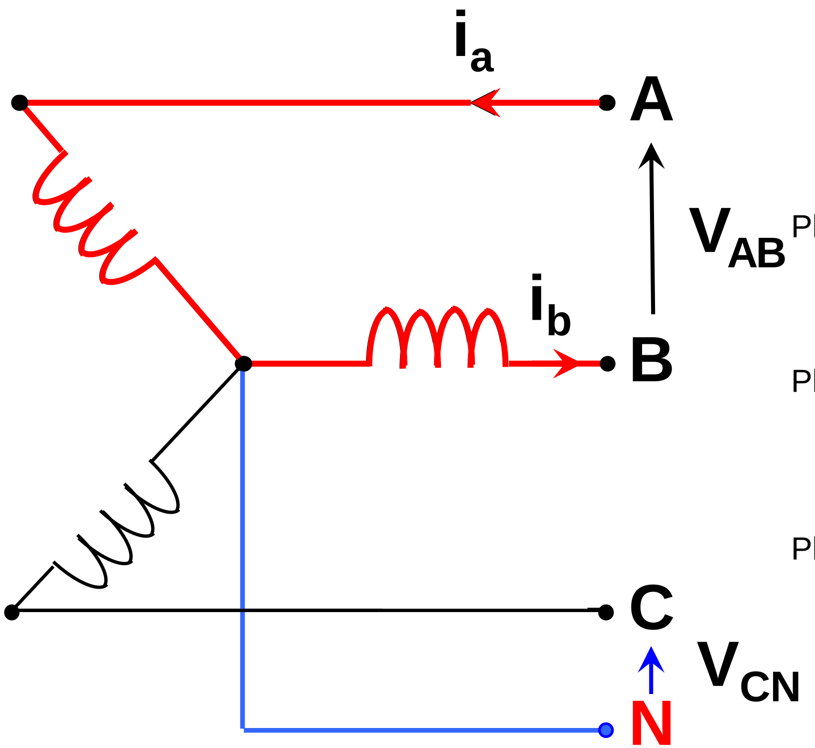
Difference Between BLDC & PMSM



- **BLDC Control**
  - Fed with direct current
  - Stator Flux commutation each 60º
  - Two phases ON at the same time
  - High torque ripple
  - Commuation at high speed difficult
  - High noise
- **PMSM control**
  - Fed with sinusoidal current
  - Continuous stator flux angle change
  - All phases ON at the same time
  - Low torque ripple
  - Higher max achievable speed
  - Low noise

BLDC control strategy

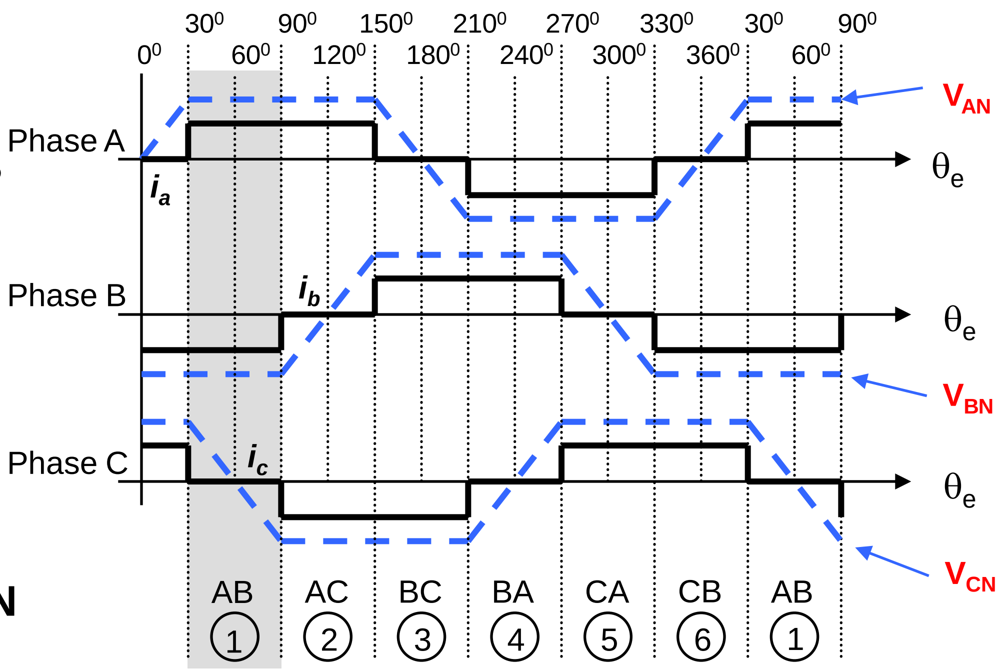
3 phases BLDC
star connection
with central point N



Two of the three phases are always energized, while the third phase is turned off.

Switching instant are linked to rotor position

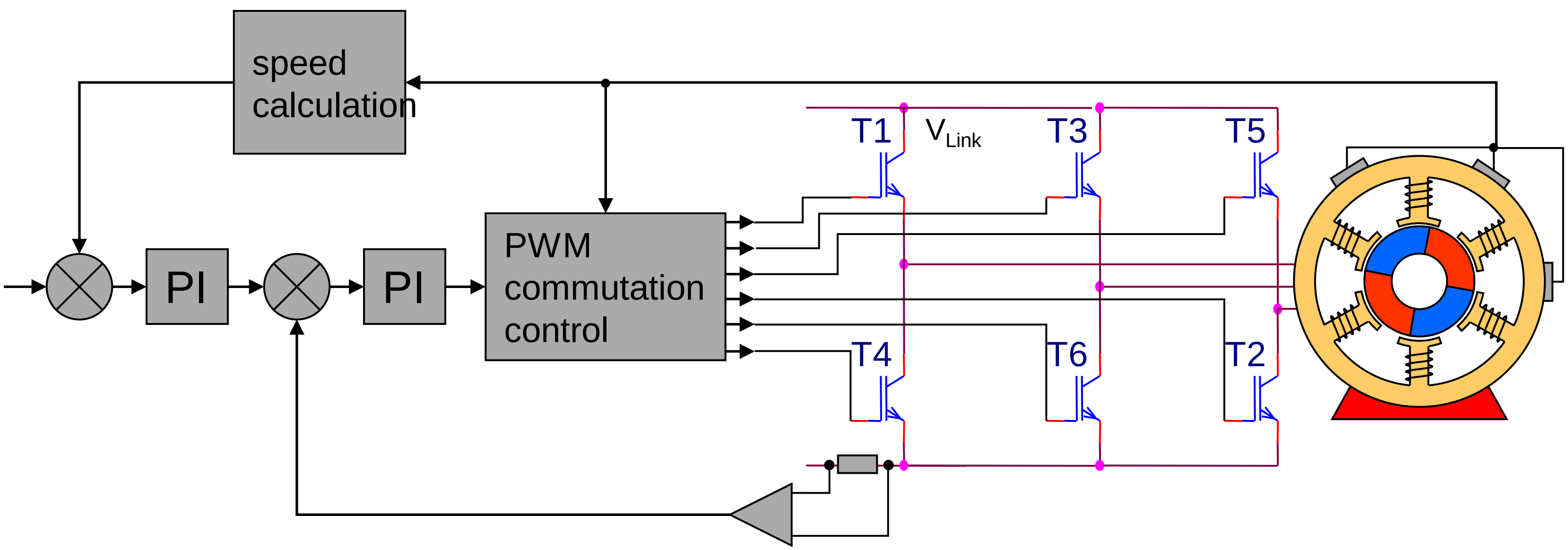
→ need for precise position sensing/evaluation of 30º, 90º, 150º, 210º, … positions



Hall Effect Control of BLDC Motors

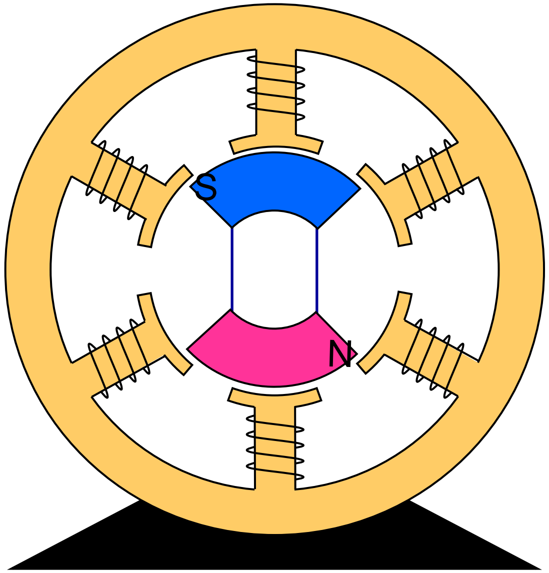
For better performance we use closed loop control

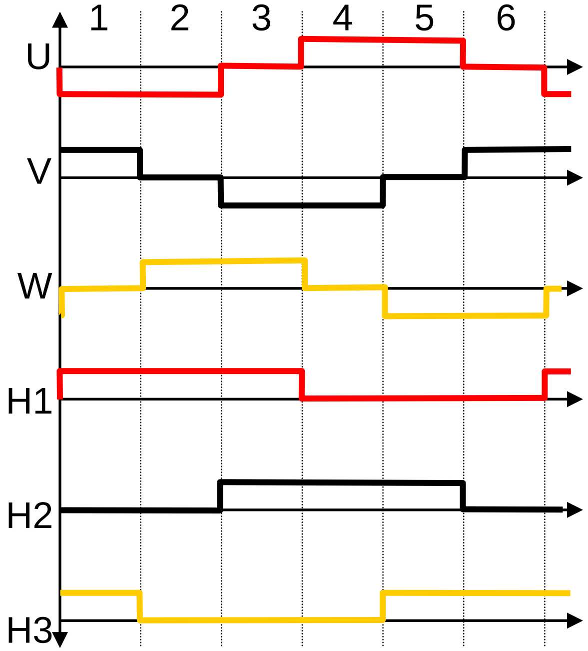
- **The Hall sensors and associated electronics will generate the signals necessary for correct commutation**
- **Information from the Hall sensors are also used to calculate and feed back the velocity**
- **Only two phases conduct at any one time; for current feedback dc-link current is measured and fed back**



Hall Effect Control of BLDC Motors

- Only two out of the three phases are energised at any one time
- The phases are energised in a 6 step manner
- The Hall sensors and the associated electronics will generate the signals necessary for correct commutation
- Current is then injected when the E_{bemf} of each phase as reached its flat portion. This will ensure constant torque

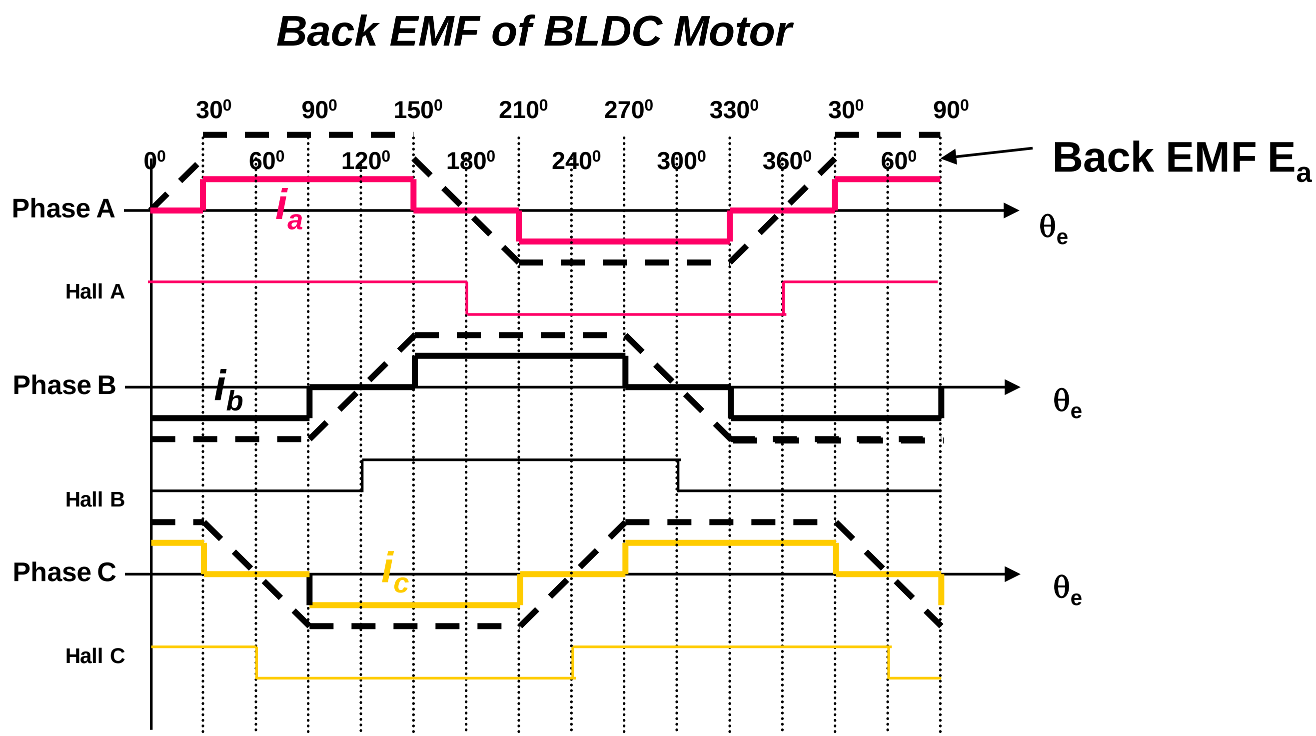




Hall Effect Control of BLDC Motors

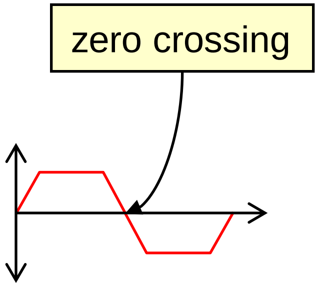
In the Previous slide, we stated that we use the signal from the hall effect to inject a DC current when the Bemf reaches it flat region

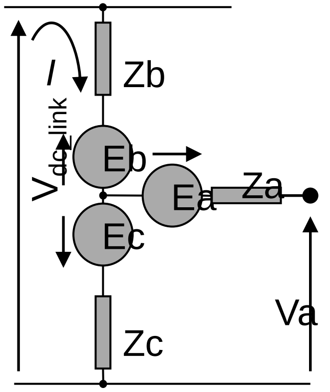
– Flat current & Flat Bemf → Flat torque (i.e. constant torque)
– No need for complex PWM (i.e. Slow switching)



Sensorless Control of BLDC Machines

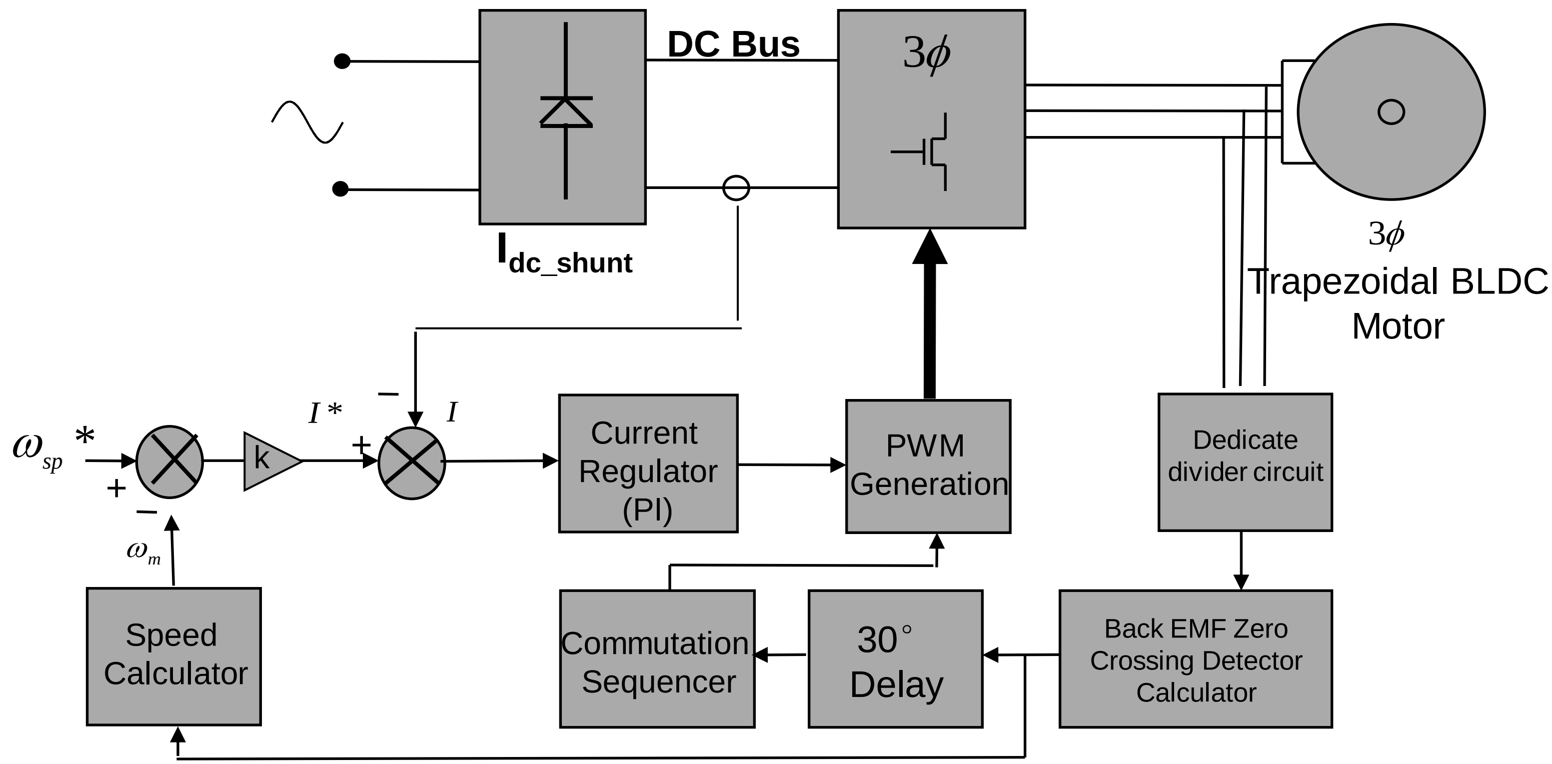
- The back-emf E_{bemf} waveform is directly related to the position of the rotor. If we could detect the zero crossing of the back-emf waveform we could deduce the position of the rotor
- We will then need to wait for 30° for the E_{bemf} to reach its constant region and then turn on the current in that phase
- The waiting time needed is dependant on the motor speed and can be deduced by continuously measuring the previous E_{bemf} zero crossings.
- Usually operated open loop at low speeds and when E_{bemf} becomes large enough to estimate accurately the loop is closed
- In a BLDC system only two coils are "on" at any moment in time.
- It has been shown* that E_{bemf} at the unconnected phase is crossing its zero point when the terminal voltage at that phase is equal to V_{dc_link} /2
- In other words if we measure V_a, when $V_a = V_{dc\_link}/2$ the $E_{bemf} = 0$





* "Microcomputer Control of Sensorless Brushless Motor", K. Iizuka et.al, IEEE Transactions on Industry Applications, Vol IA-21, No4, May/June 1985, pp. 595 - 601

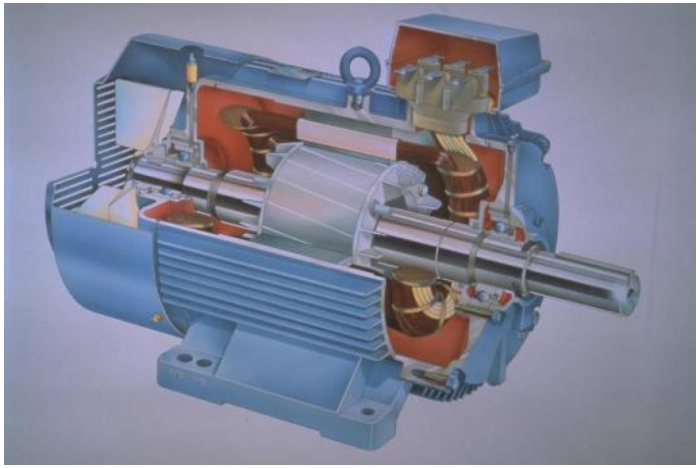
Block Diagram of Sensorless BLDC Control

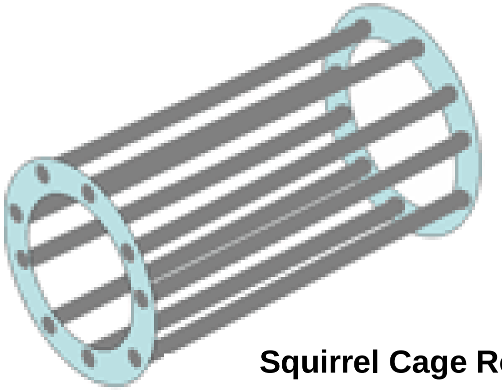


- Hall sensors are removed. Resistor dividers and on-chip ADC are used to sense phase voltages.
- Phase voltages are used to detect zero crossing of back EMF and trigger commutation
- See TI Application reports (SPRA498 and BPRA072) for more details

Speed Closed-Loop Control with Current Control

AC Induction Motors



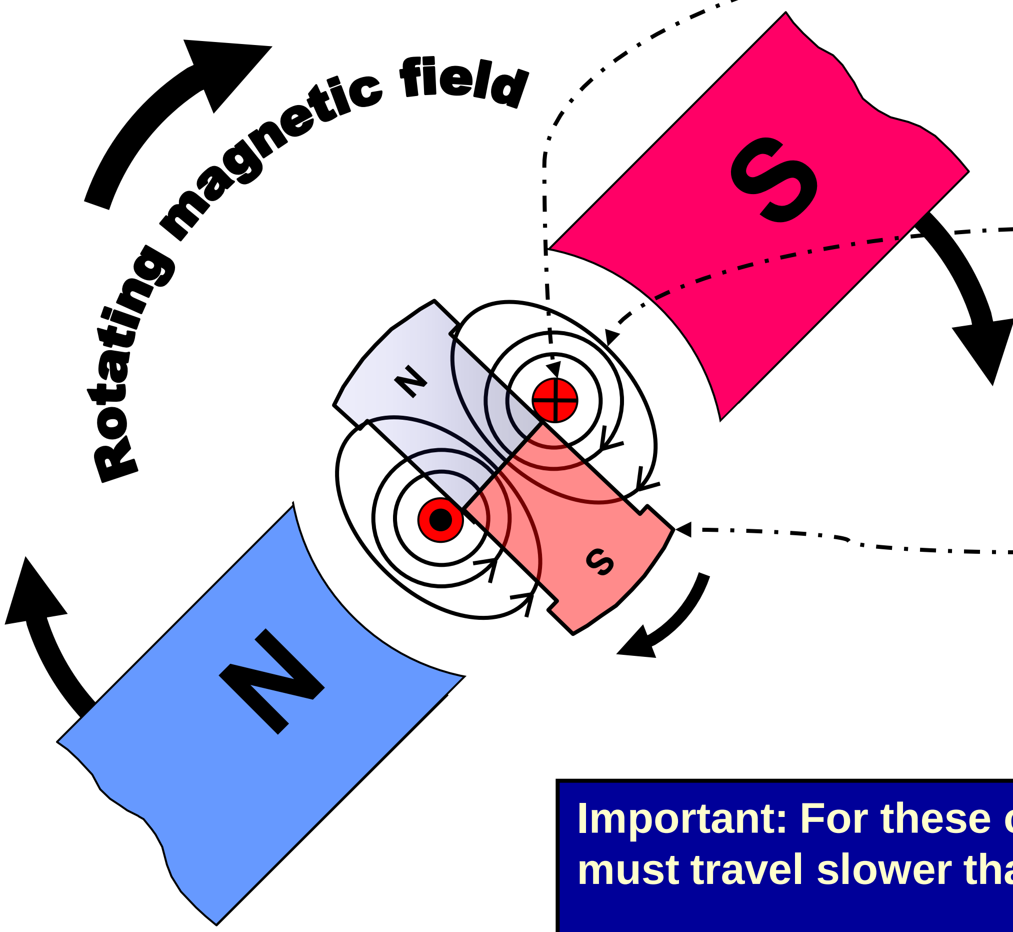


Squirrel Cage Rotor

- Invented in 1888 by Nikola Tesla
- Reliable construction: no brushes
- Simple, low cost design
- Good efficiency at fixed speeds.
- Asynchronous
- Speed and control position are expensive
- Poor performance at low speed operation
- Requires complex control to be competitive

Typical applications: industrial drives, white goods, fans, high speed applications

Induction Motor Operation

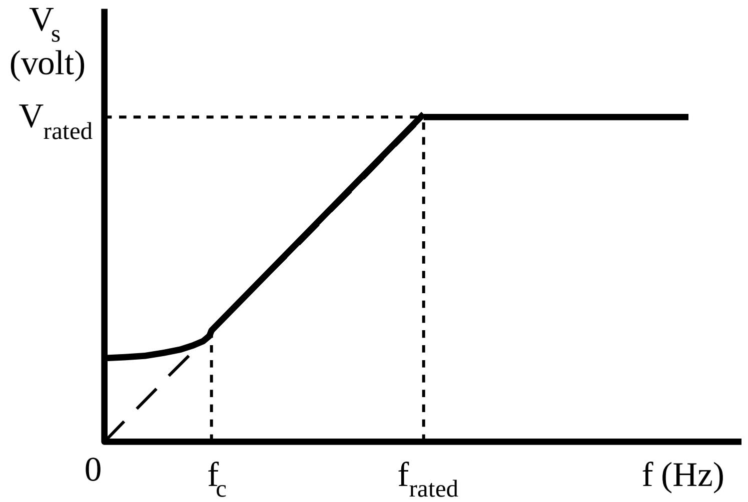


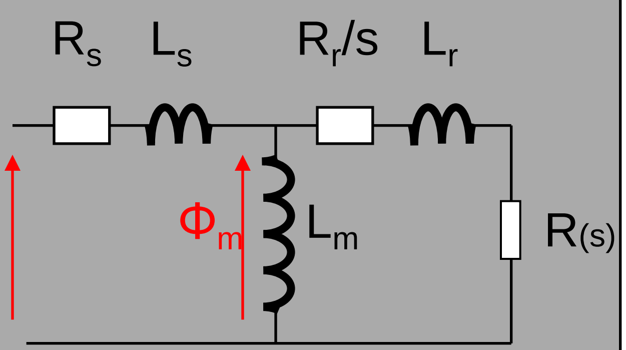
1. The rotating magnetic field in the stator, induces a current in the rotor
2. This current will have a magnetic field associated with it
3. This magnetic field makes the rotor behave like a magnet which will then follow the stator's rotating magnetic field

Important: For these currents to be induced the rotor must travel slower than the stator this is called slip:

$$s = \frac{\omega_e - \omega_r}{\omega_e}$$

Scalar Control (V/f) - Limitations



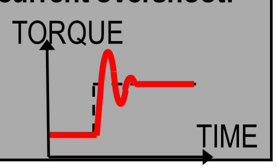


$$I_m = (V_m / 2\pi f\ L_m)$$

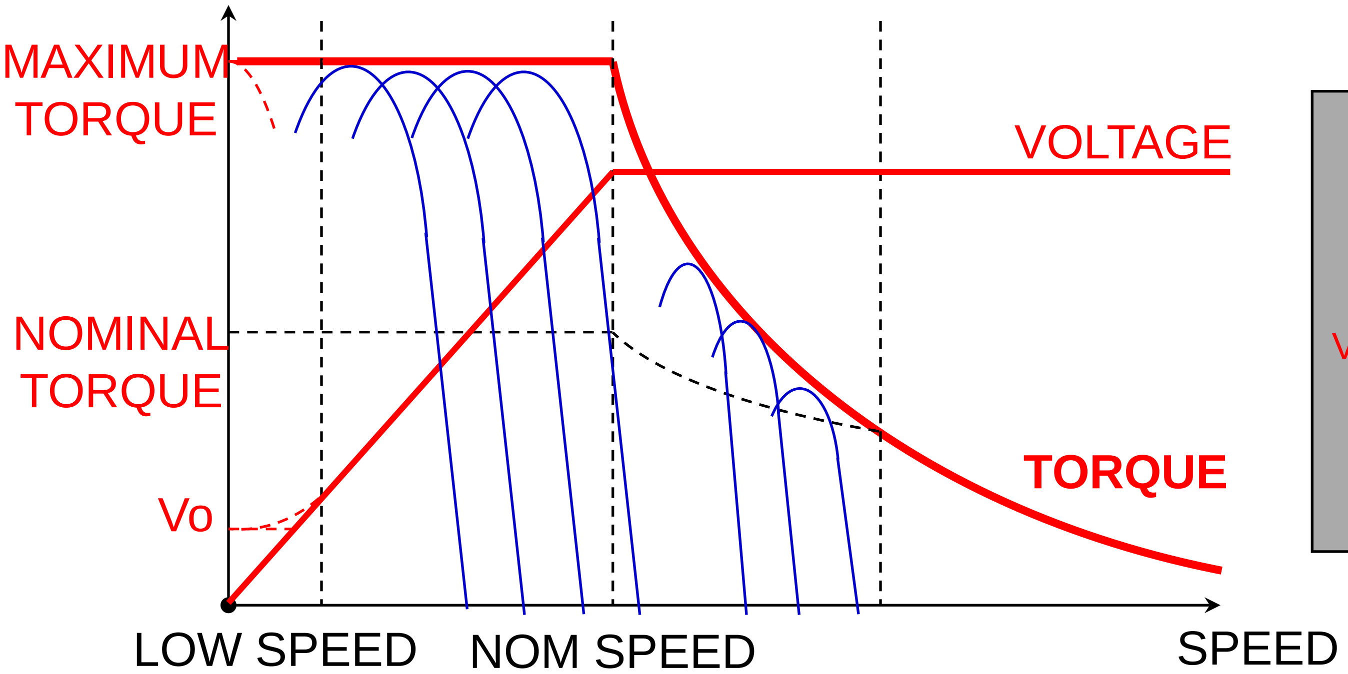
$$\Phi_m = (V_m / 2\pi f)$$

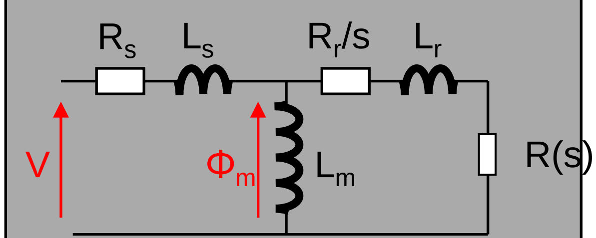
- **+ Simple to implement: All you need is three sine waves feeding the motor**
- **+ Position information not required (optional).**
- **– Doesn't deliver good dynamic performance.**
- **– Torque delivery not optimized for all speeds**

Torque oscillation generates uncontrolled current overshoot:



Scalar Control (V/f) Limitations





$$I_m = (V_m / 2\pi f L_m)$$

$$\Phi_m = (V_m / 2\pi f)$$

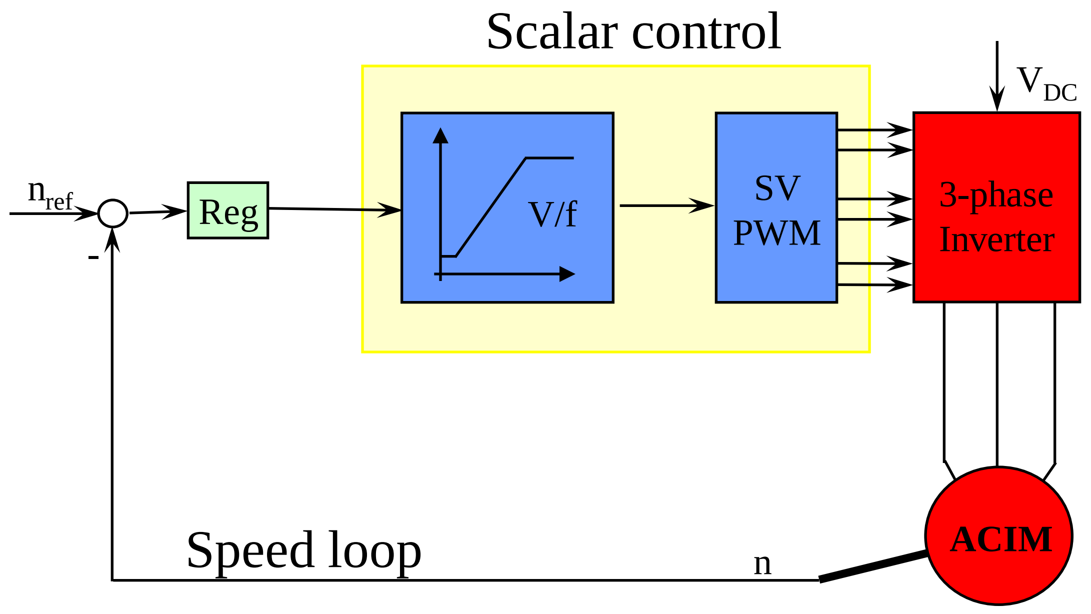
At or near nominal speed:

Stator voltage drop negligible: (Vm = V),

At low speed: Rs is no longer negligible: Vm < V

A large portion of energy is now wasted.

V/F control definition

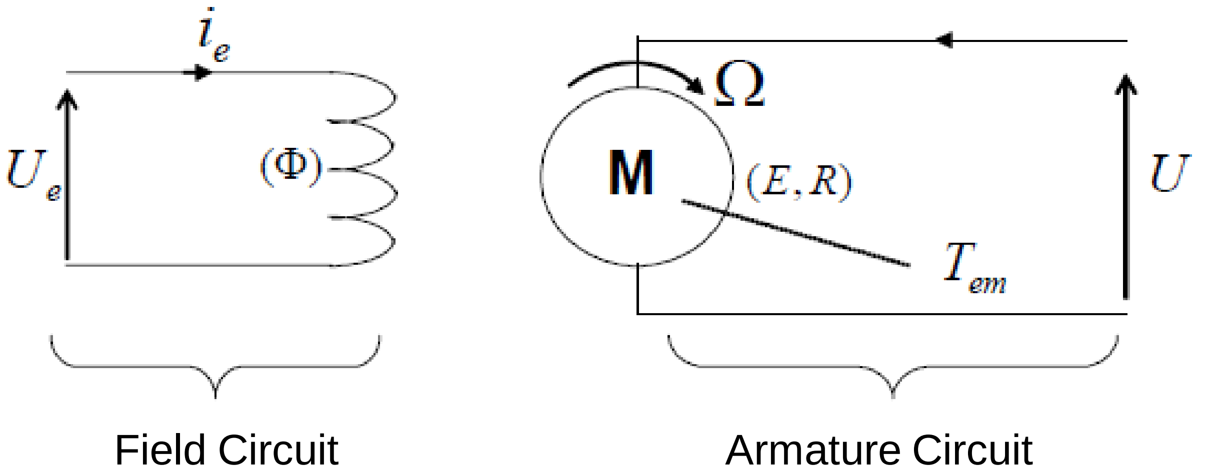


+ Simple to implement: All you need is three sine waves feeding the ACI
+ No position information needed.

– Doesn’t deliver good dynamic performance.
– Not so good at low speed.

Vector Control

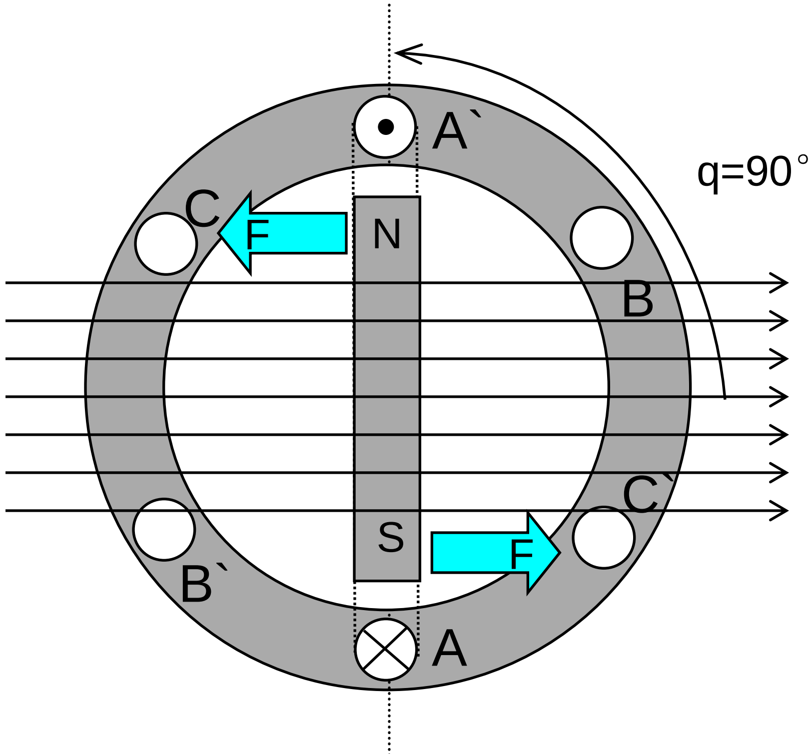
Field Oriented Control (FOC)

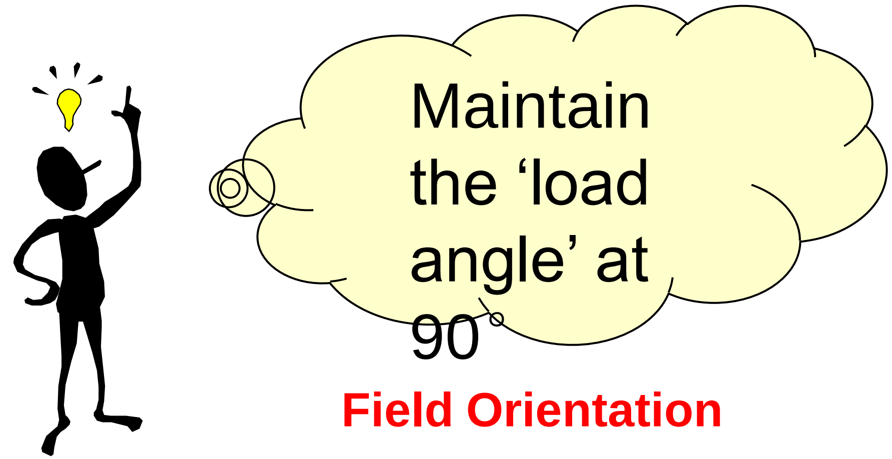


$$T_{em} = K.\Phi.I$$
$$E = K.\Phi.\Omega$$
$$\Phi = f(I_e)$$

- **Separated excitation DC motor model**
- **Flux and torque are independently controlled**
- **The current through the rotor windings determines how much torque is produced.**

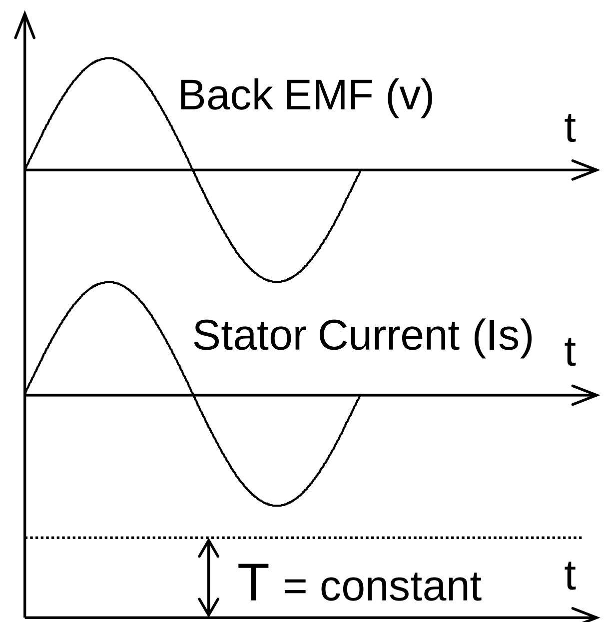
Vector Control Concepts





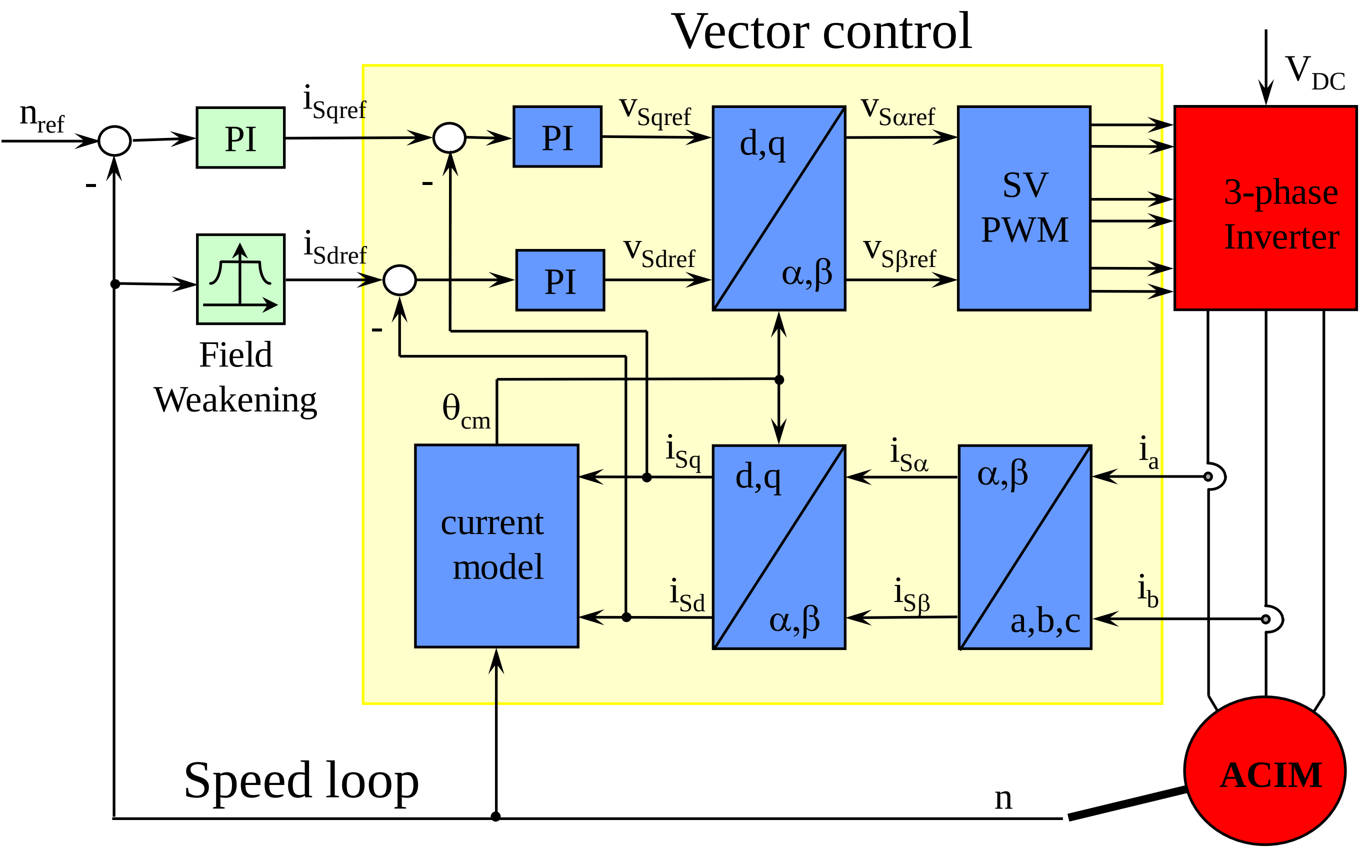
Field Orientation

$$T_{em} = \vec{B}_{stator} \times \vec{B}_{rotor}$$



- **+ Reduced torque ripple**
- **+ Better dynamic response**
- **+ Good performance at lower speeds**
- **- Need rotor position info**

FOC control Overview



Require MIPS

Require current information (sensing)

Vector control:

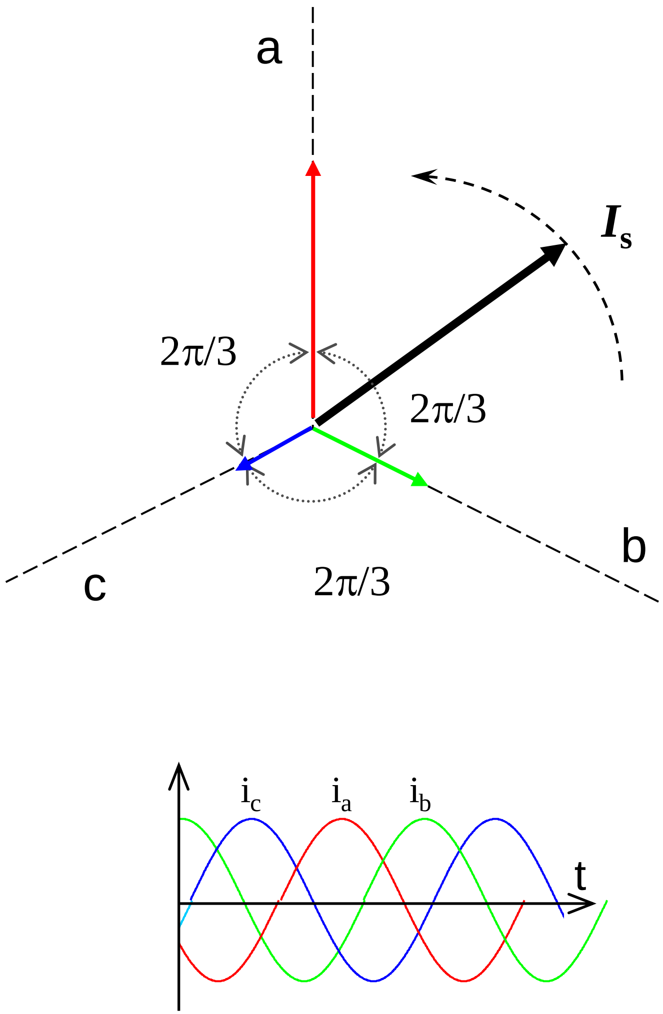
This method is based on the dynamic model of the motor.

The **Flux (Id)** and The **Torque (Iq)** are controlled **separately.**

Flux and Torque are controlled **in real time.**

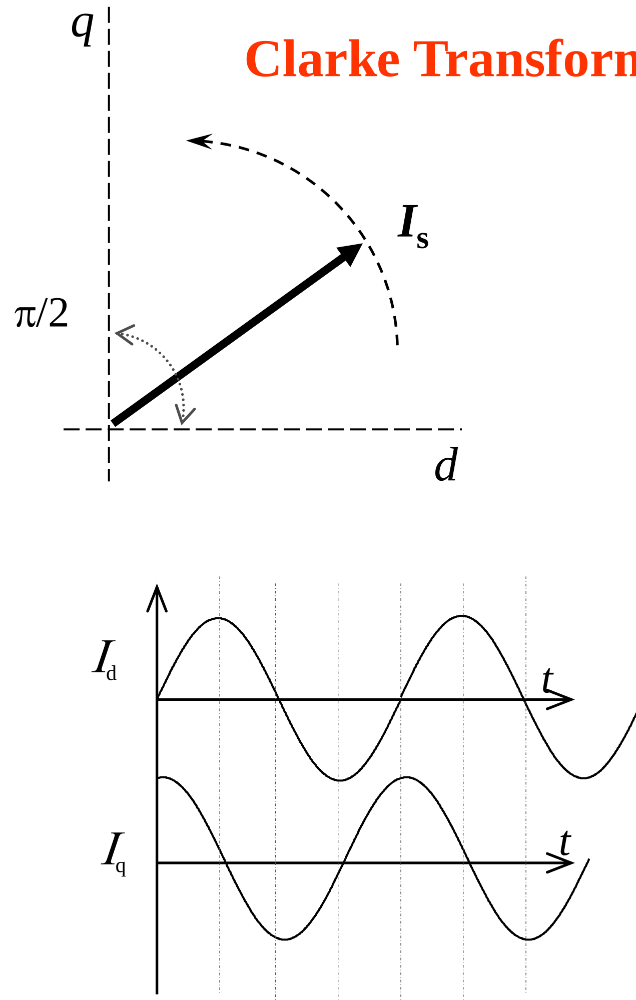
Some key mathematical components are required!

Stationary Reference Frame



Three phase reference frame

Clarke Transform



Two phase orthogonal reference frame

Stationary to Rotating Reference Frame

Again the transformation equations can simply be derived by resolving f_s, along the desired axis:

dq^s to dq^e transform:

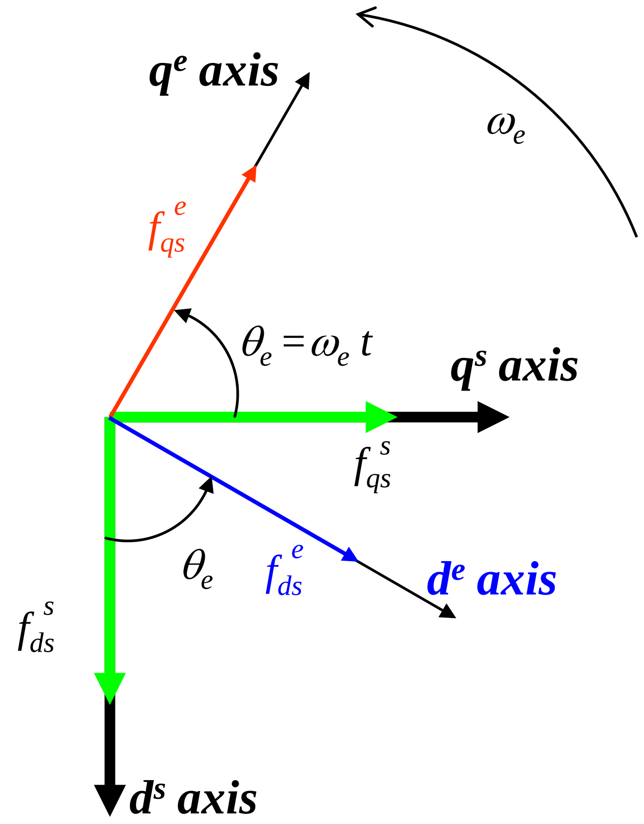
$$f_{qs}^{e} = f_{qs}^{s}\,Cos(\omega_e t) - f_{ds}^{s}\,Sin(\omega_e t)$$

$$f_{ds}^{e} = f_{qs}^{s}\,Sin(\omega_e t) + f_{ds}^{s}\,Cos(\omega_e t)$$

Inverse dq^s to dq^e transform:

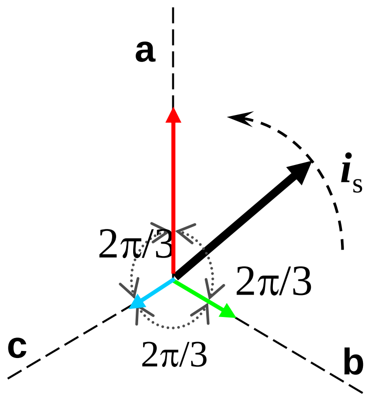
$$f_{qs}^{s} = f_{qs}^{e}\,Cos(\omega_e t) + f_{ds}^{e}\,Sin(\omega_e t)$$

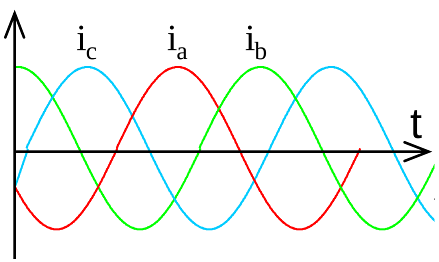
$$f_{ds}^{s} = -f_{qs}^{e}\,Sin(\omega_e t) + f_{ds}^{e}\,Cos(\omega_e t)$$



Determination of Torque & Flux from Stator Currents

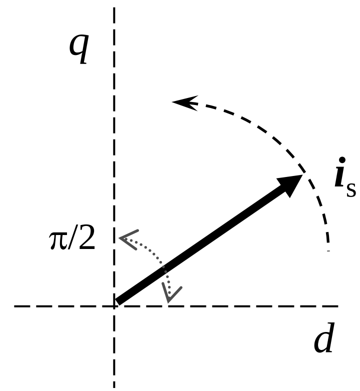
3-phase Currents

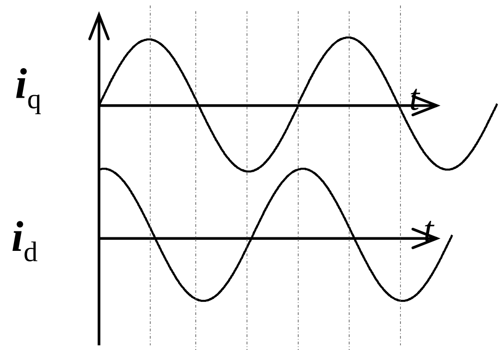




Three phase stationary reference frame

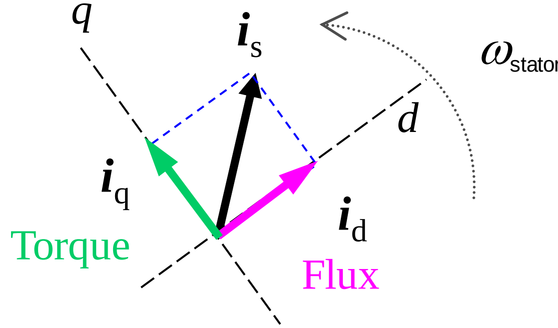
Clarke Transformation

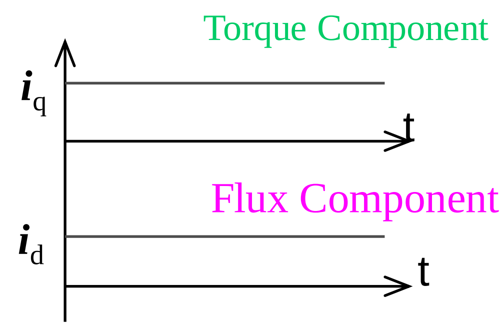




Two phase orthogonal stationary reference frame

Park Transformation



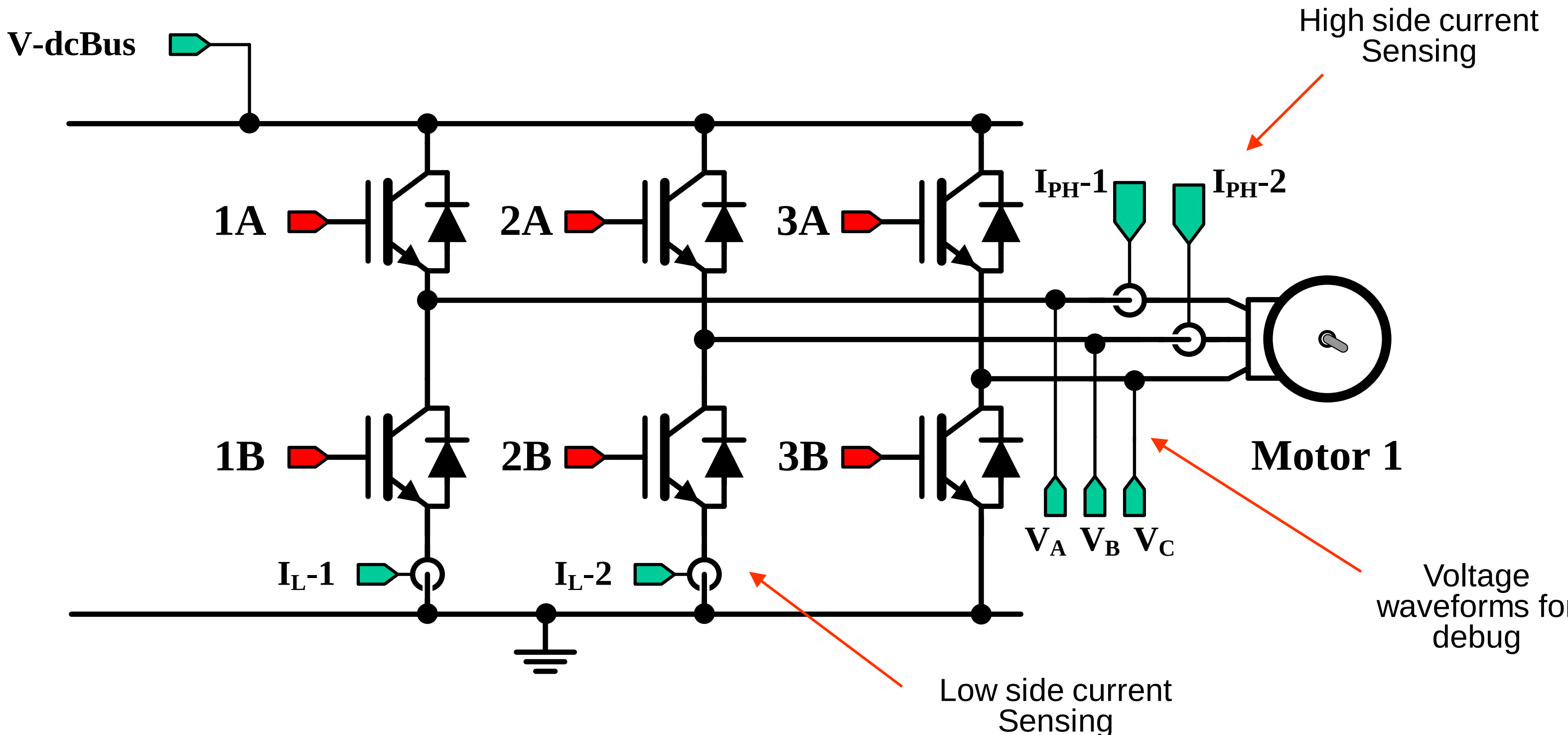


Rotating reference frame

Content

- Motor Control Methods Overview
- Inverter / Hardware considerations
- Software / Algorithm Considerations
- DMC Development Kits
- System Incremental Build
- Demo, Q&A

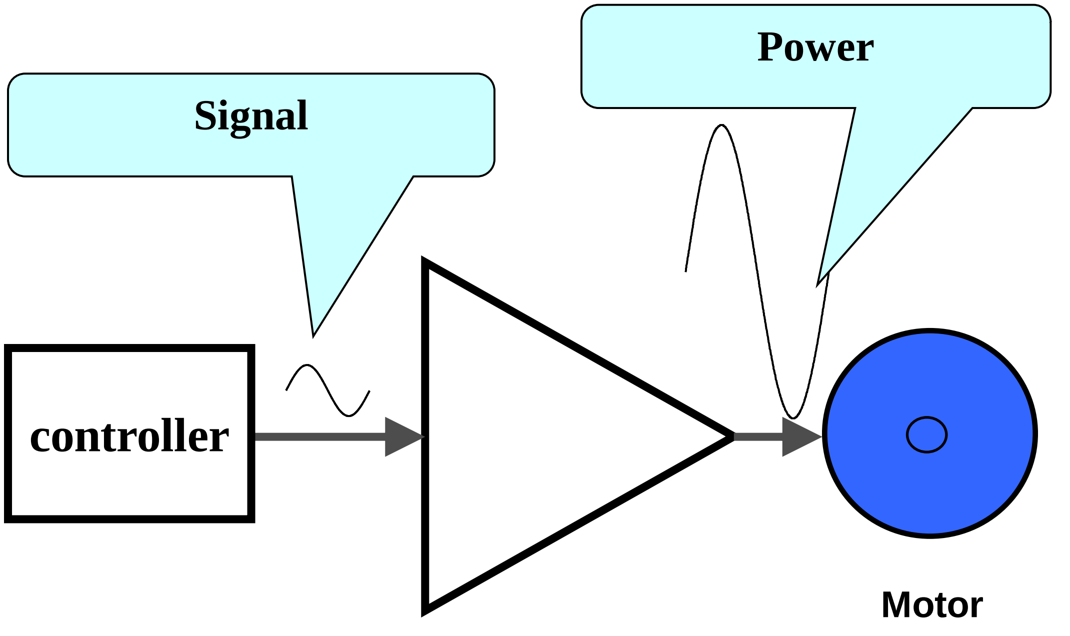
3-phase Inverter Control



Decisions that need to be made:

- **How to drive complementary PWM?**
  - In the Motor Control and PFC Developer's Kit we chose to use a Integrated Power Module (IPM) to generate the complementary PWM. We could have used the C2000's deadband submodule within each PWM module instead.
- **In sensorless, how will we sense current?**
  - In the Motor Control and PFC Developer's Kit we chose to use low-side sensing.
  - Low-side current sensing is more difficult, but more scalable. In high-side current sensing, amplifiers that allow high common-mode voltages are expensive.

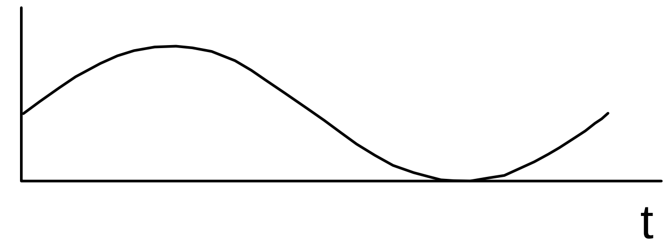
Why Pulse Width Modulation?



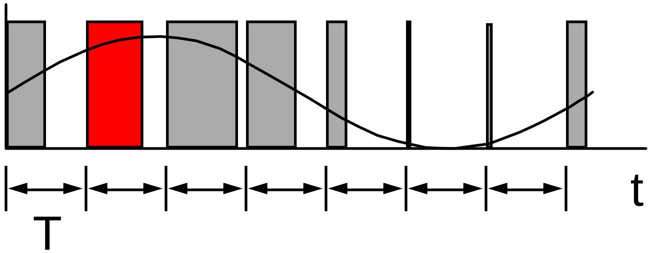
A linear power amplifier will:

- waste too much power
- cost money
- heat dissipation
- bad for environment

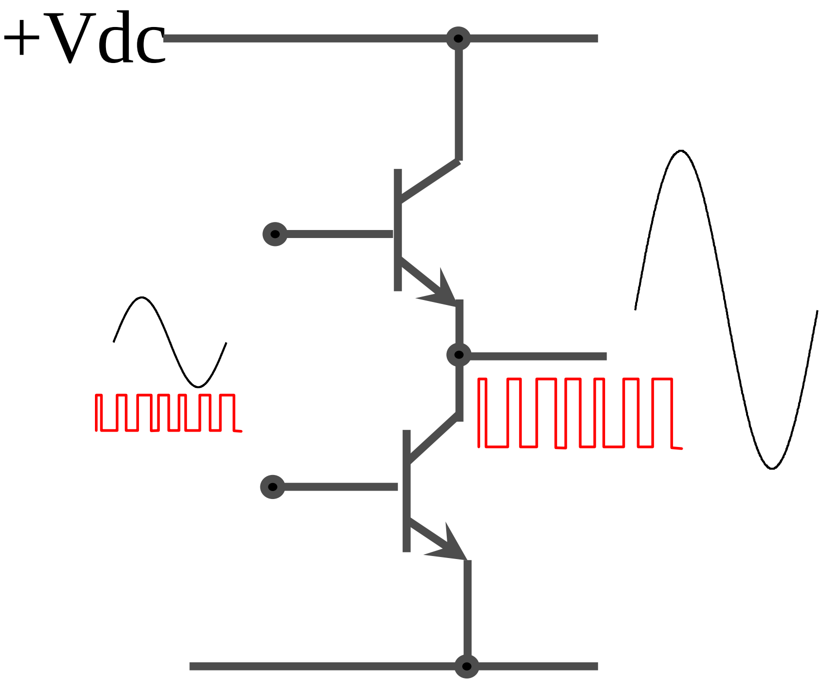
Why Pulse Width Modulation ?



Original Signal

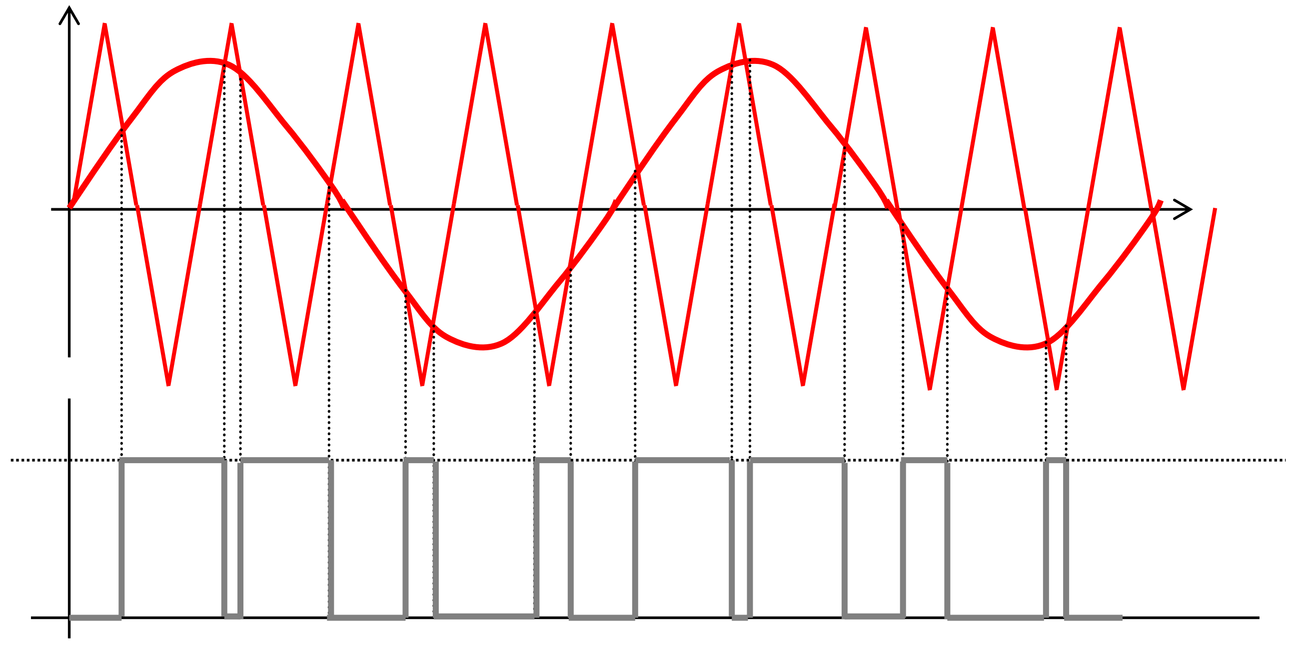


PWM representation



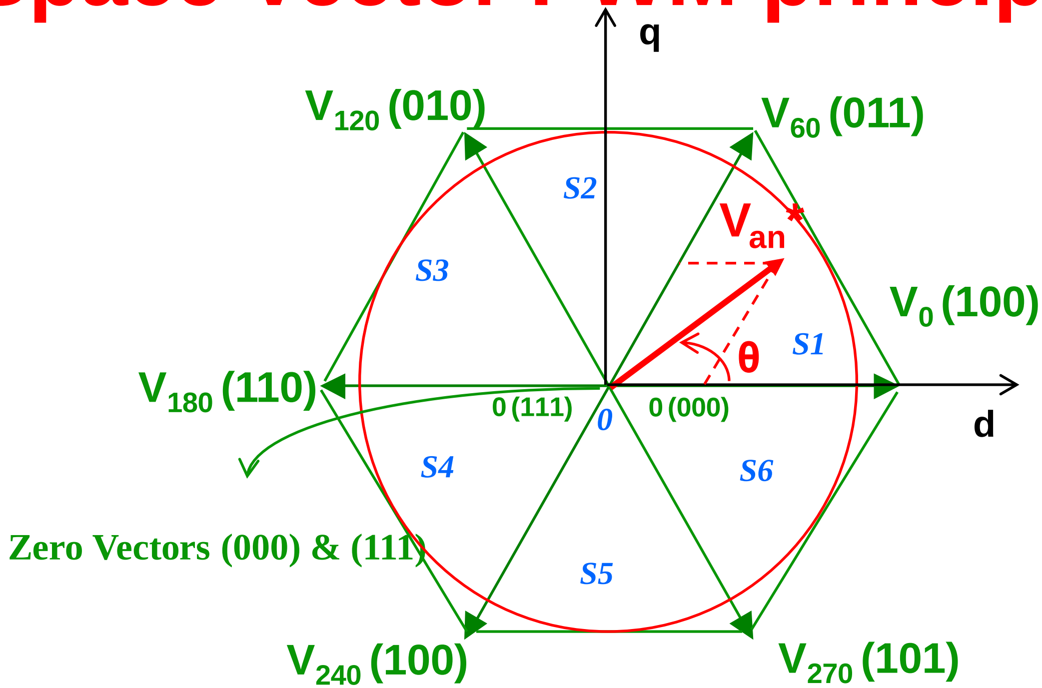
Digital signal ⇒ easy to output
High efficiency switching

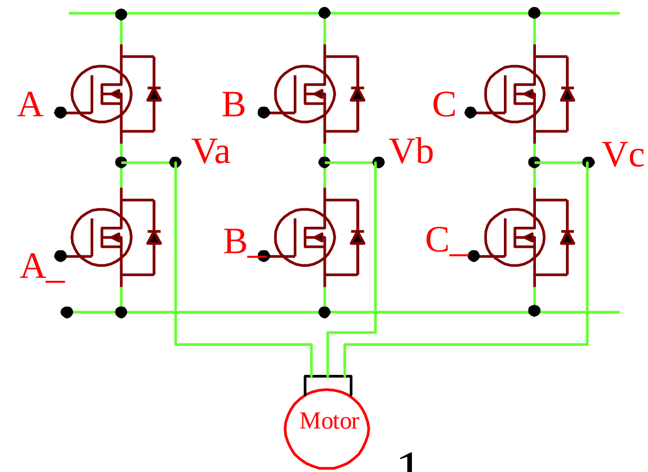
PWM Signal Generation



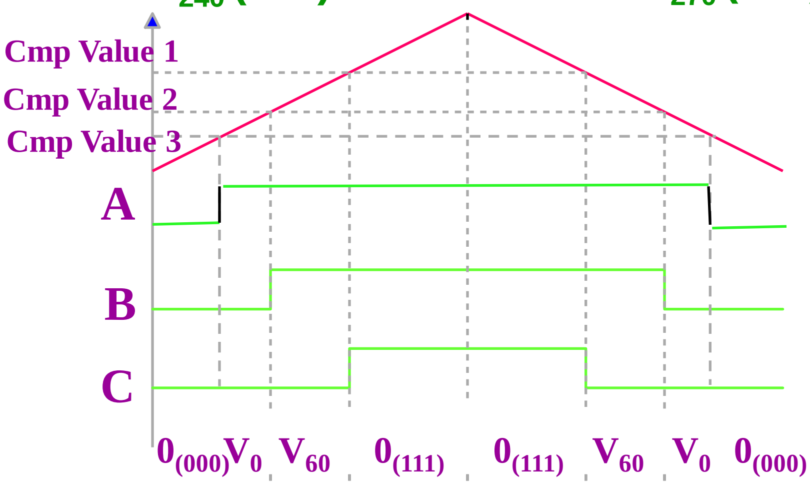
- Traditional way: comparing three-phase sinusoidal waveforms with a triangular carrier
- V_{an} = V.sin(ωt) (Van phase-neutral voltage)
- PWM consideration:
  - Time Step
  - Bit width
  - Update conditions
  - Dead time
  - ....

Space Vector PWM principle





$$V_{an}^{*} = V.\sin(\omega t) + \frac{1}{6}.\sin(3\omega t)$$

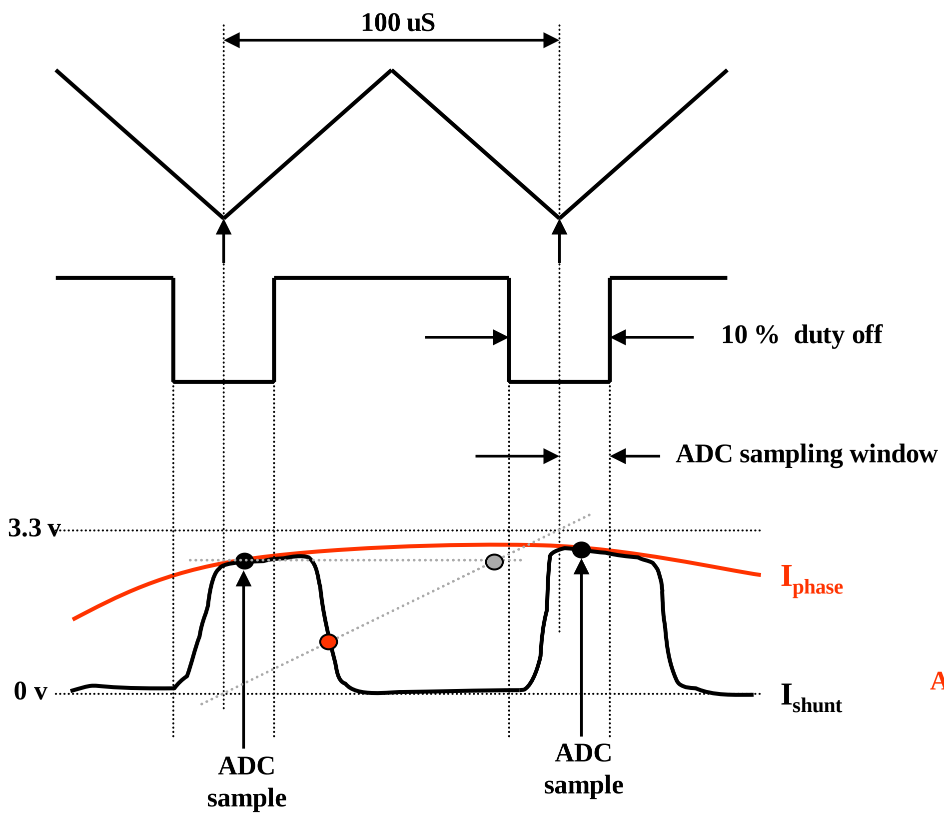


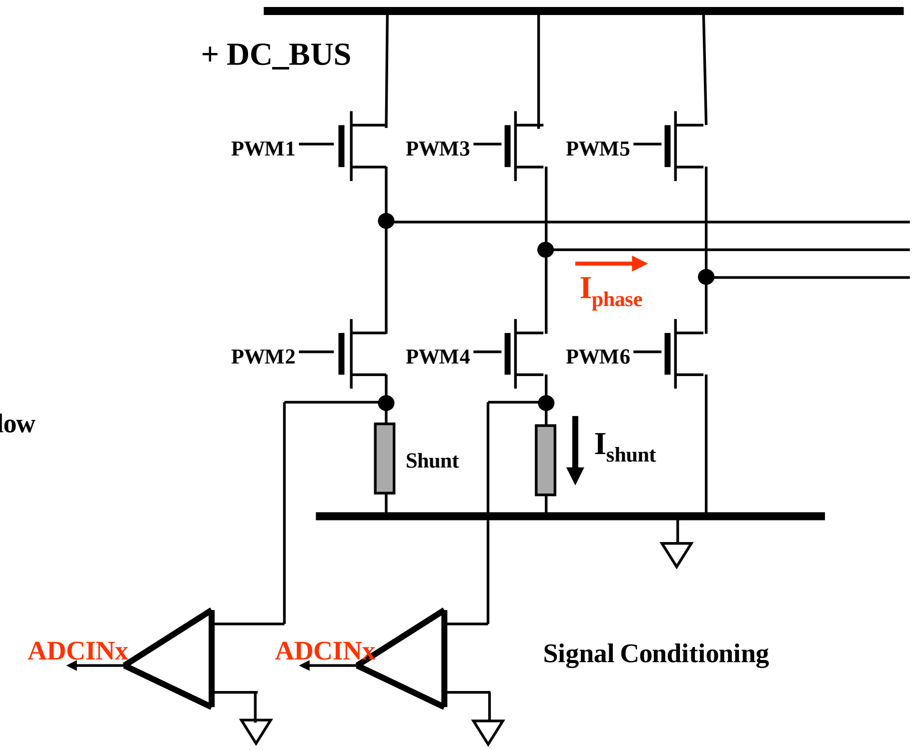
- Third harmonic injection
- Line to line voltage still sinusoidal
- PWM technique
- DSP hardware implemented
- Increase the maximum inverter output voltage of 15%
- Reduce transistor commutations

- **We build the required voltage vector as a combination of one of the six basic switches configuration**

Texas Instruments

Current Sensing

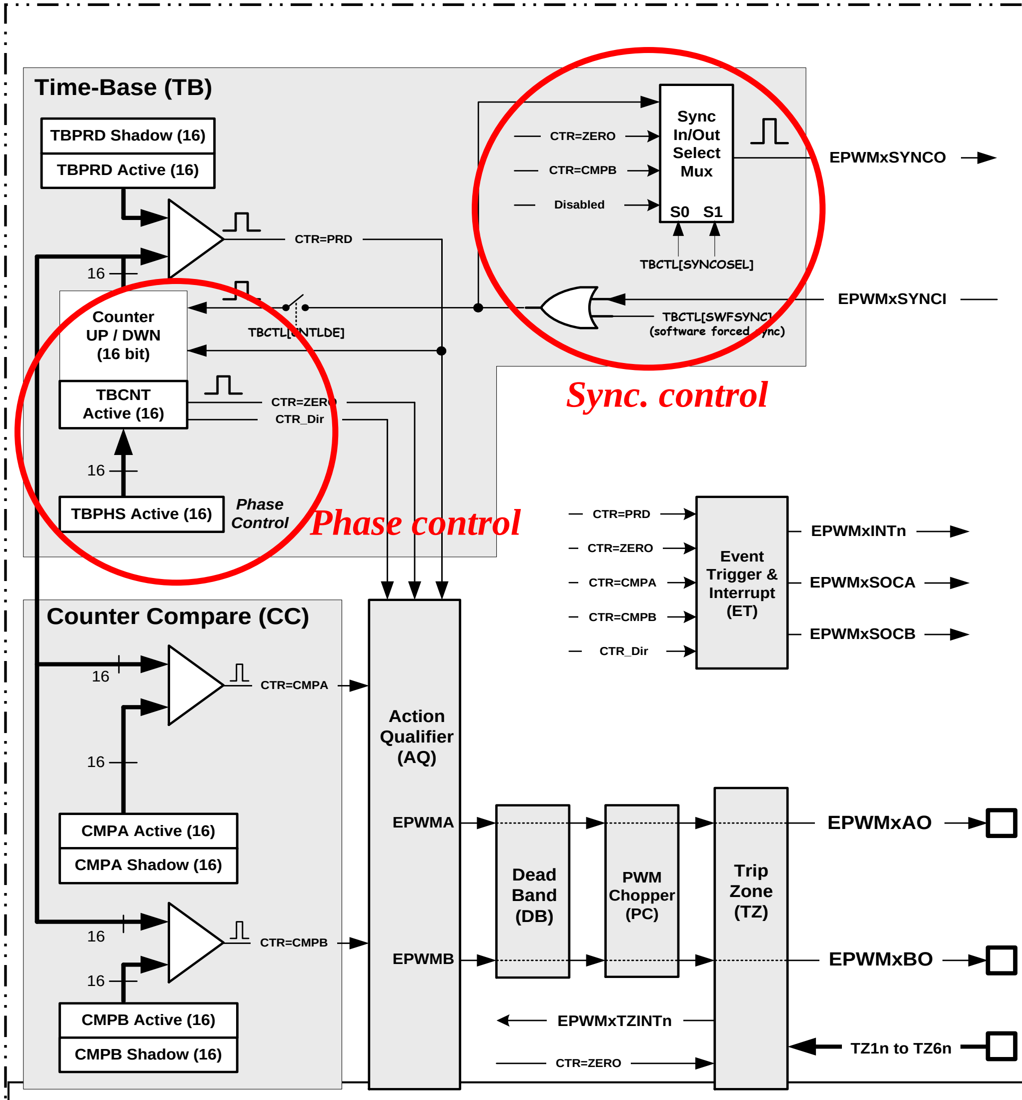




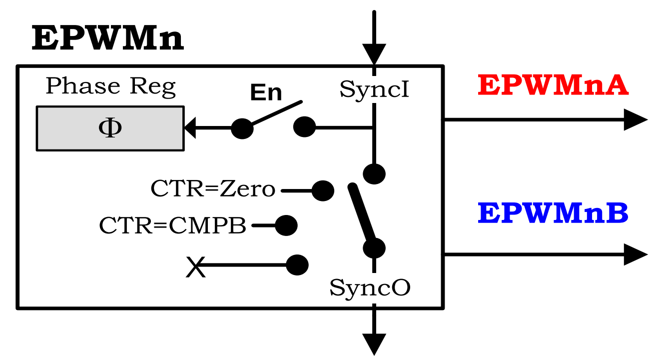
- Requires flexibility of triggering sample and conversion in middle of PWM pulse
- Fast ADC S/H is required
- CPU operation promptly
- 3 phase current are available under all load conditions

Flexible PWM module

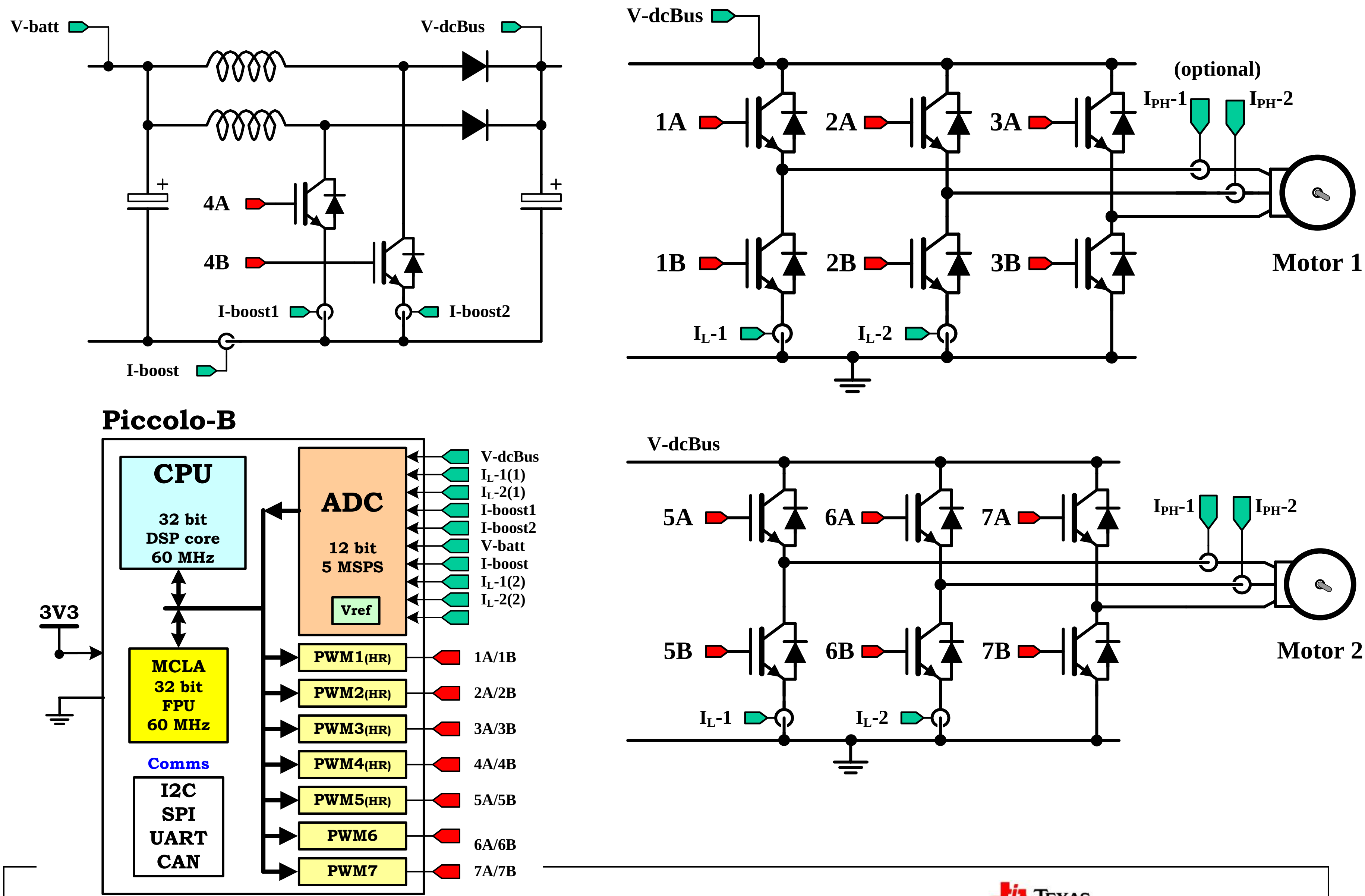
Single EPWM module (in detail)



Simple view of EPWM

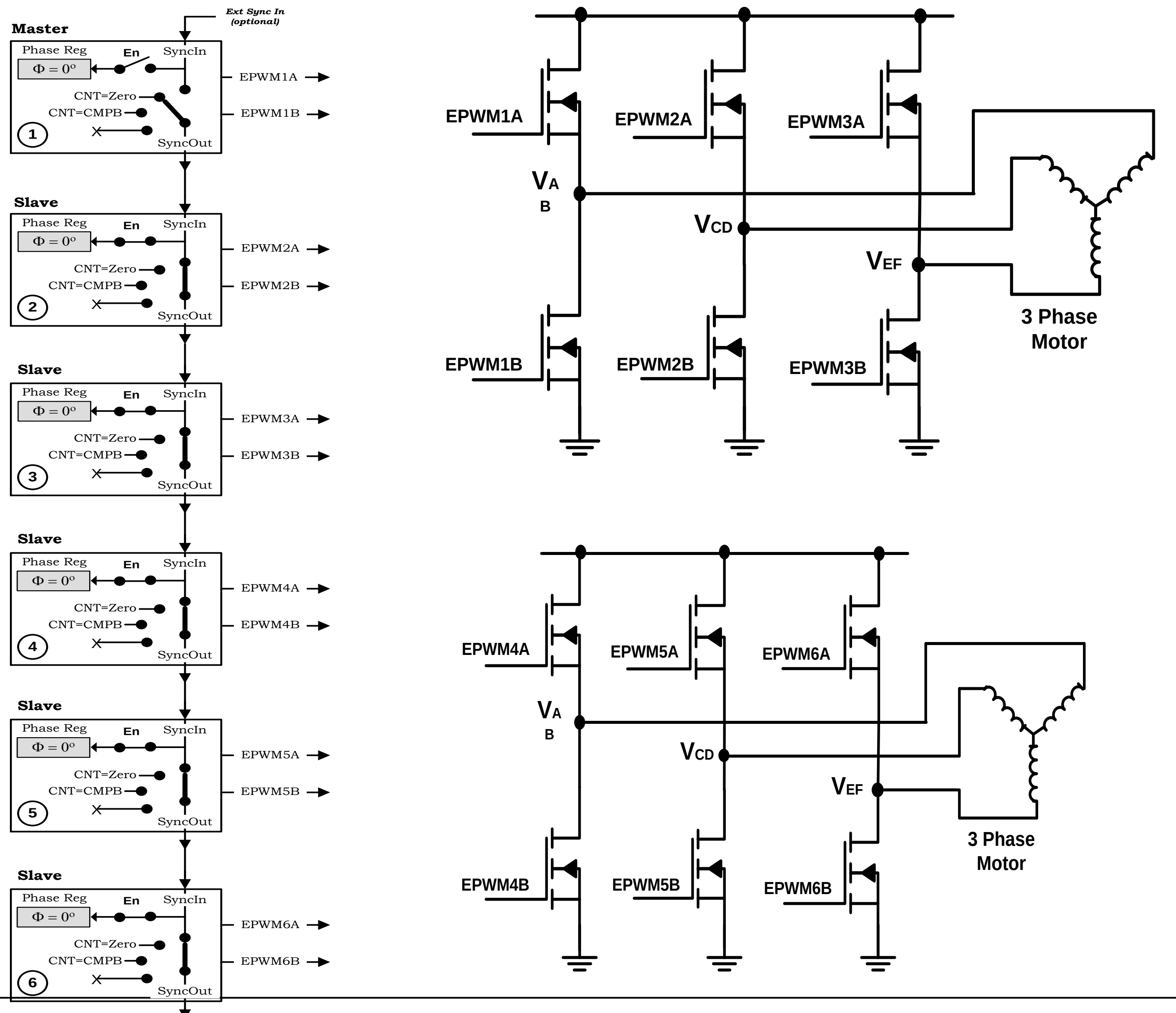


Dual Inverter + Boost



Texas Instruments

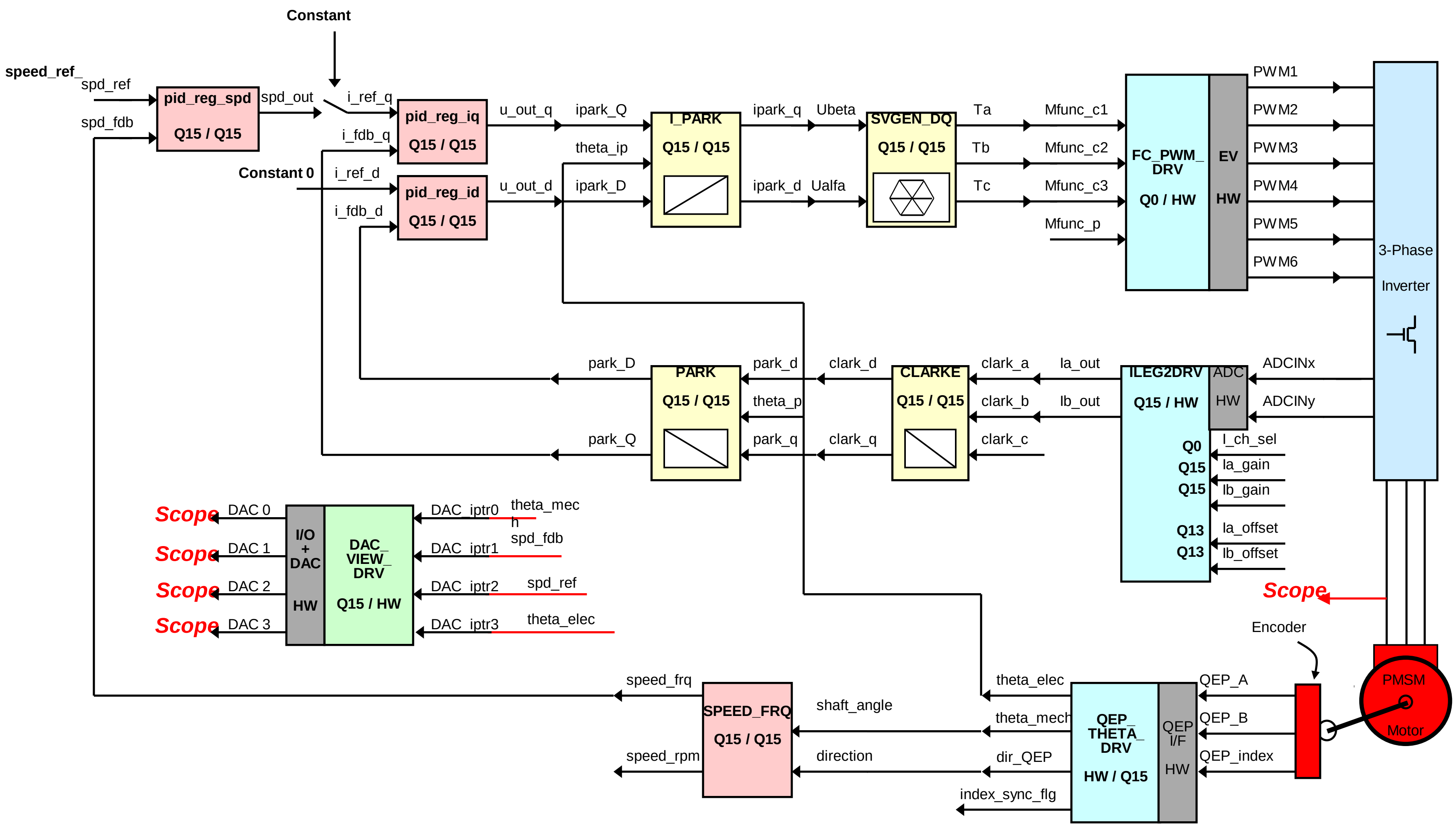
Dual - 3 Phase Inv. for Motor Drives



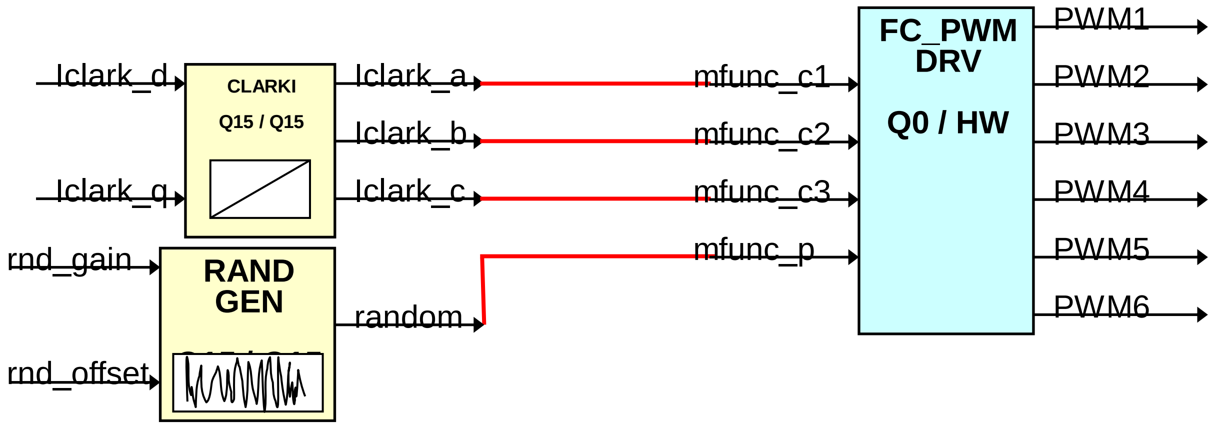
Content

- Motor Control Methods Overview
- Inverter / Hardware considerations
- Software / Algorithm Considerations
- DMC Development Kits
- System Incremental Build
- Demo, Q&A

TI DMC Software Library



Interconnecting Modules



At the "C" level:

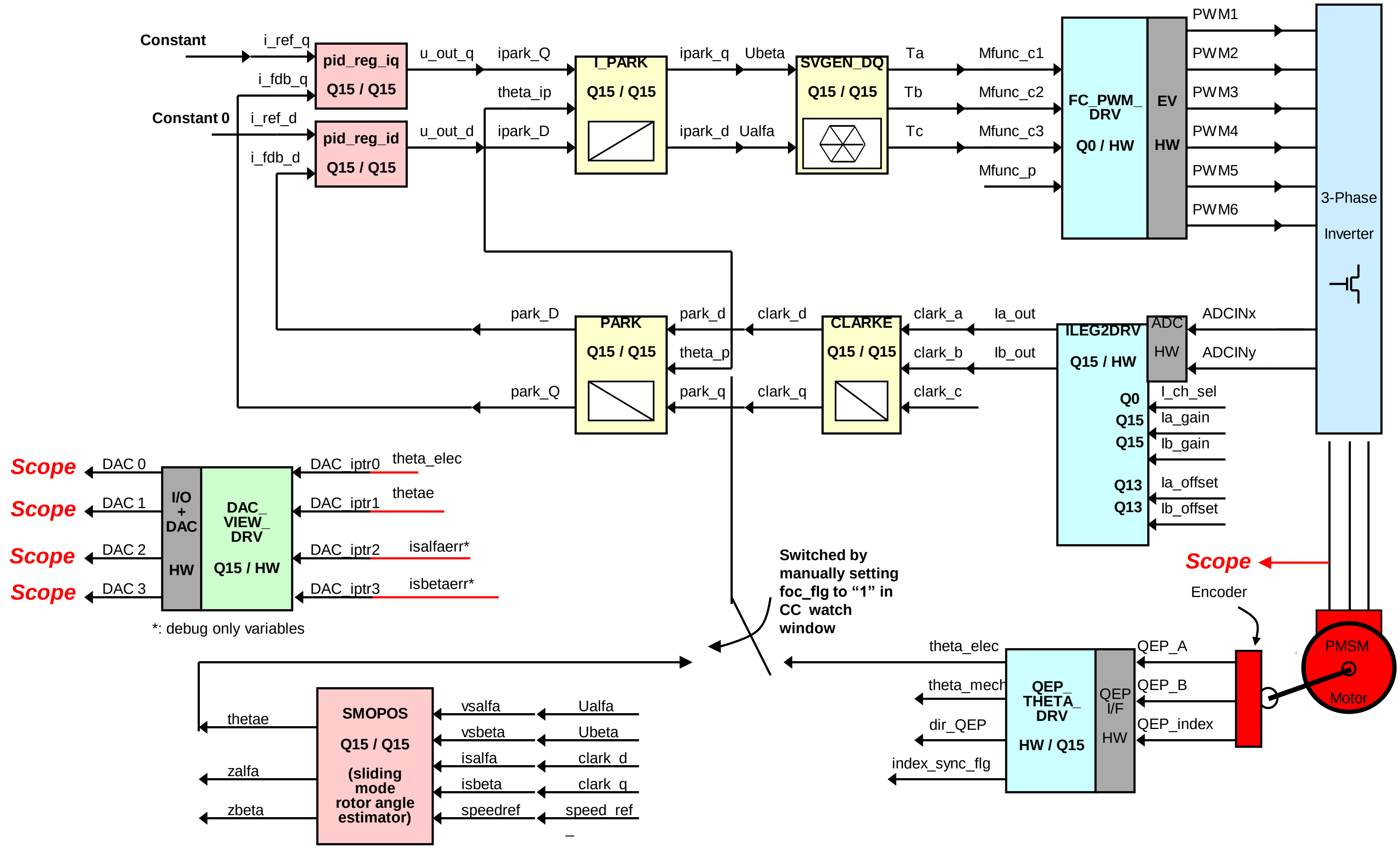
```
clarkInv(&dqBuffer, &fcPwm.InputBuffer)
fcPwmInputBuffer.ditherIn = randomGen1.calc(&randomGen1)
fcPwm.calc(&fcPwm);
```

Digital Motor Control Library (DMC-Lib)

The DMC-Lib contains:

- PID regulators,
- Clarke transforms,
- Park (& Inverse) transforms,
- Ramp generators,
- Sine generators,
- Space Vector generators,
- Speed / Position meas. / estimators
- and more…

PMSM FOC Sensorless with SMO



Content

- Motor Control Methods Overview
- Inverter / Hardware considerations
- Software / Algorithm Considerations
- DMC Development Kits
- System Incremental Build
- Demo, Q&A

Platform for Motor Control

controlCARD

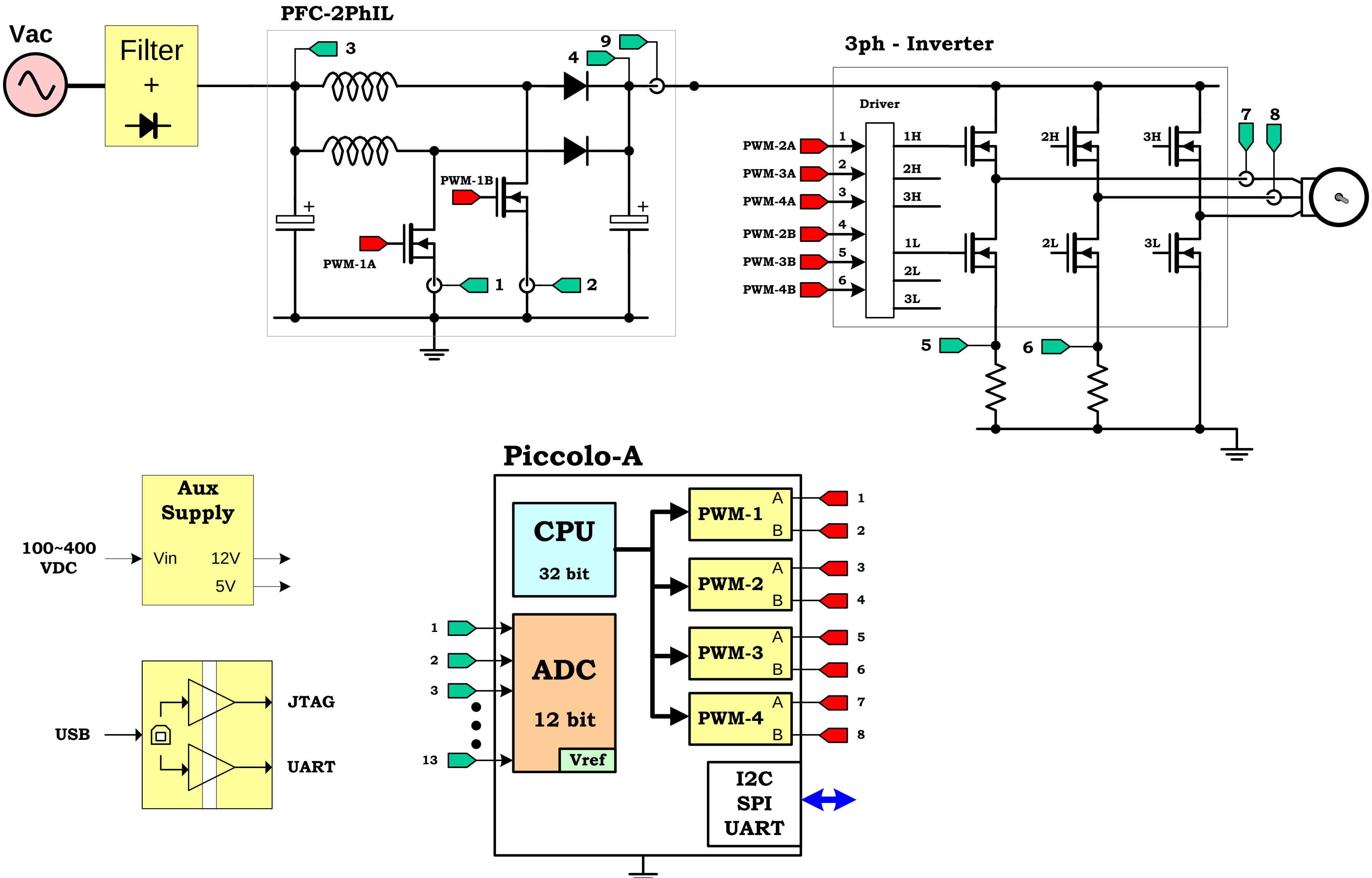
	High Voltage + PFC $599 TMDSHVMTRPFCKIT	Low Voltage + PFC $369/$399 TMDS1MTRPFCKIT / 2MTRPFCKIT
Motor Types	ACI, BLDC, PMSM	BLDC
Control	ACI & PMSM FOC BLDC Commutation	FOC (BLDC Commutation in Q3 HW Rev)
Sensor Interface	Sensorless or Hall/QEP	Sensorless or Hall/QEP
Communications	Isolated USB/UART to JTAG Isolated CAN	Isolated USB/UART to JTAG (Isolated CAN in Q3 HW Rev)
Three Phase Inverter	350V, 1.5 Kw, IPM, Fault Protection	2 x TI DRV8402 IPM, 24-36V, 40W
Current Sense	Low-side current sense OPA2350	Low-side current sense OPA2350
Optional Power Factor Correction	Digital 750W 2PHIL, 200KHz, UCC27234 IN: 85-132V, 170-250V OUT:400V DC	Digital 100W 2PHIL, 100KHz IN: 13-16V OUT:24V DC

Platform for Motor Control

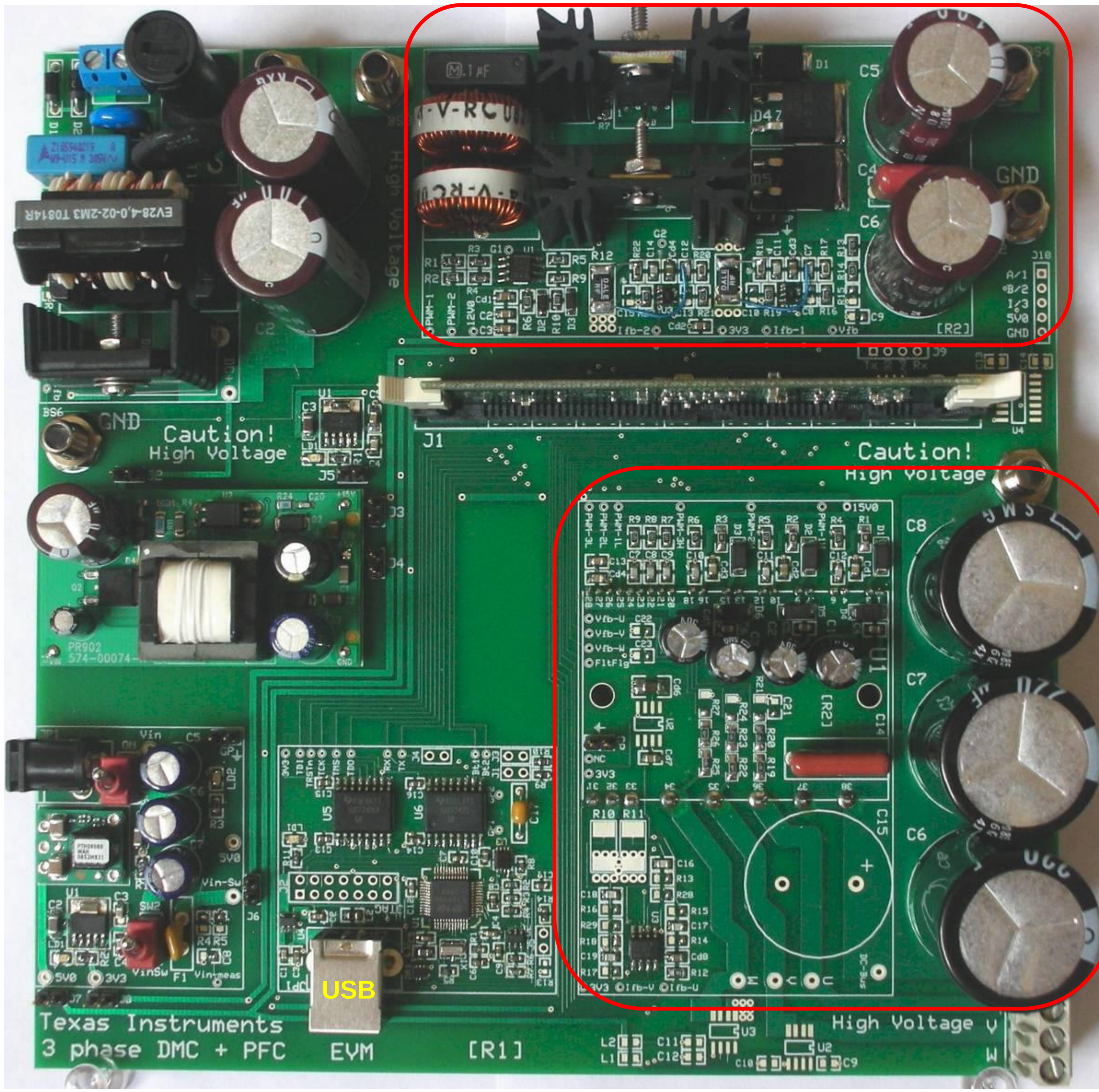
controlCARD

	High Voltage + PFC $599 TMDSHVMTRPFCKIT	Low Voltage + PFC $369/$399 TMDS1MTRPFCKIT / 2MTRPFCKIT	
Software	controlSUITE	controlSUITE	SPRC922
IDE	CCSv4.x	CCSv4.x	CCSv3.3
DMCLibrary	NEW & OPTIMIZED	NEW	OLD
GUI	YES	Q210	NO
PFC SW	Q310 w/ new PowerLIb	Q310	Yes
Projects	ACI FOC, PMSM FOC, BLDC ; All Sensored/-less	2xPMSM FOC	1/2xPMSM, +PFC
Family Support	Piccolo F2803x Delfino F2833x Q2	Piccolo F2803x (minor mods for F2802x)	Piccolo F2803x (minor mods for F2802x)

DMC – Single Axis + PFC HV



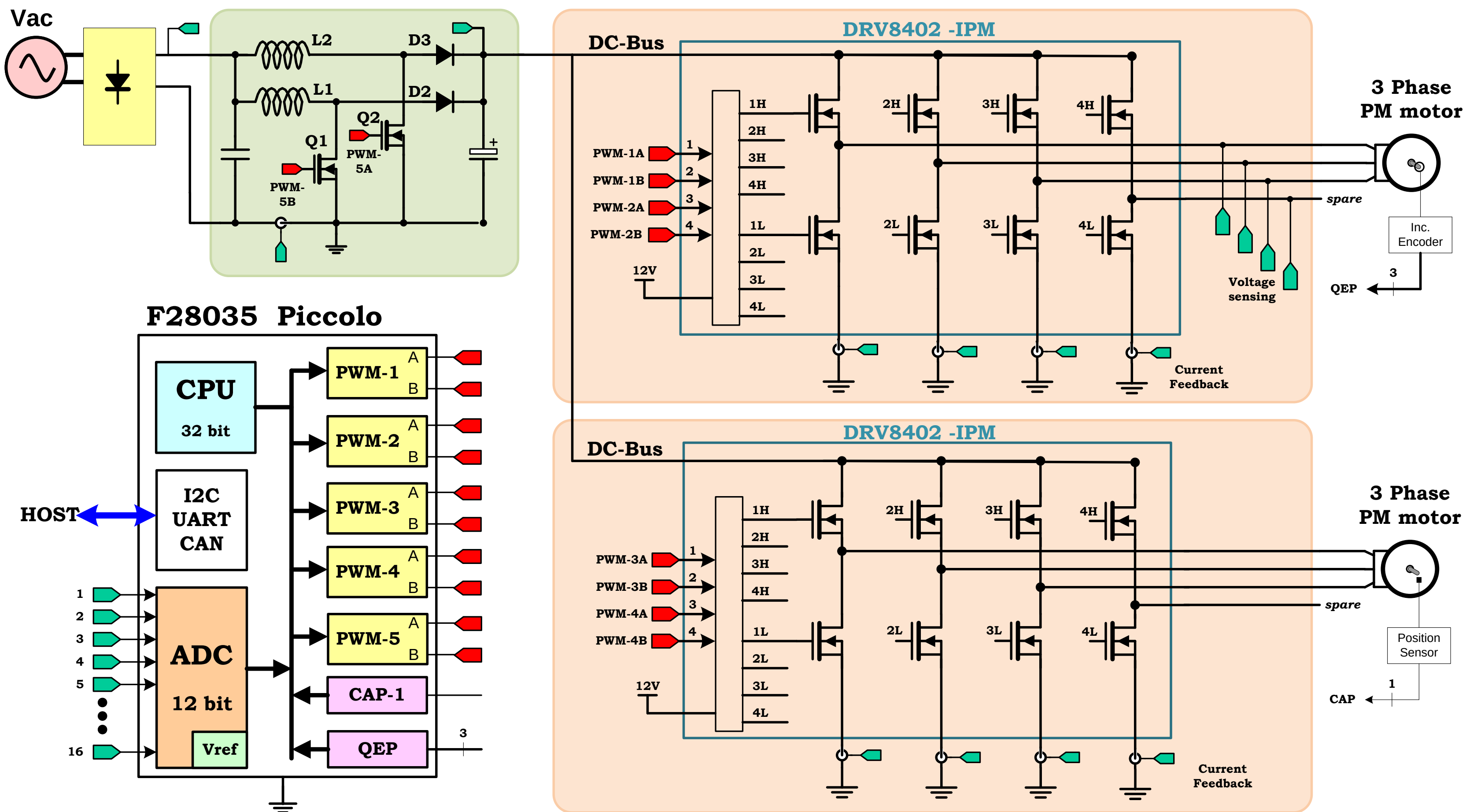
DMC – Single Axis + PFC HV



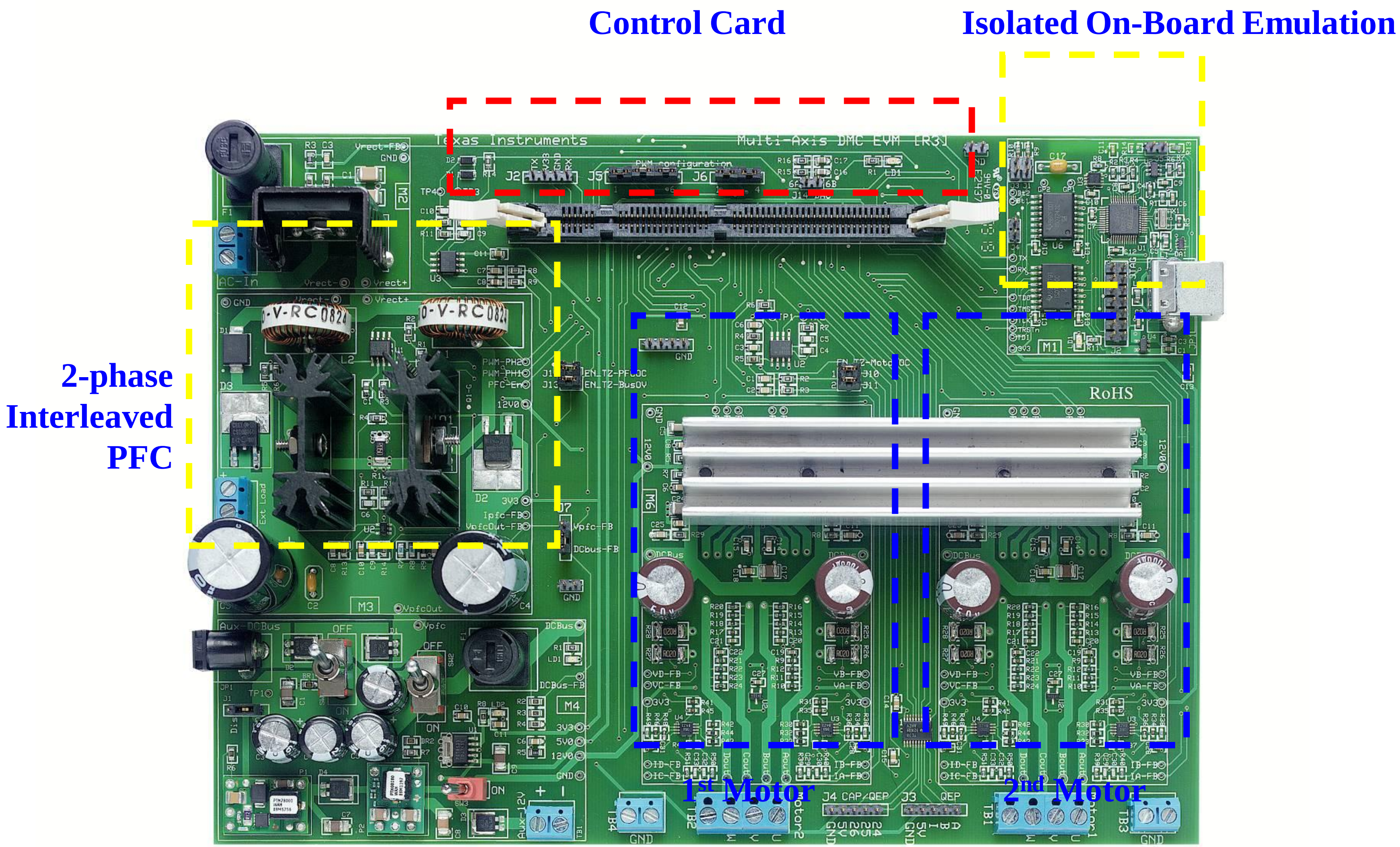
PFC – 2PhIL

DC-AC
3 Ph
Inverter

Dual Motor Control and PFC Developer's Kit



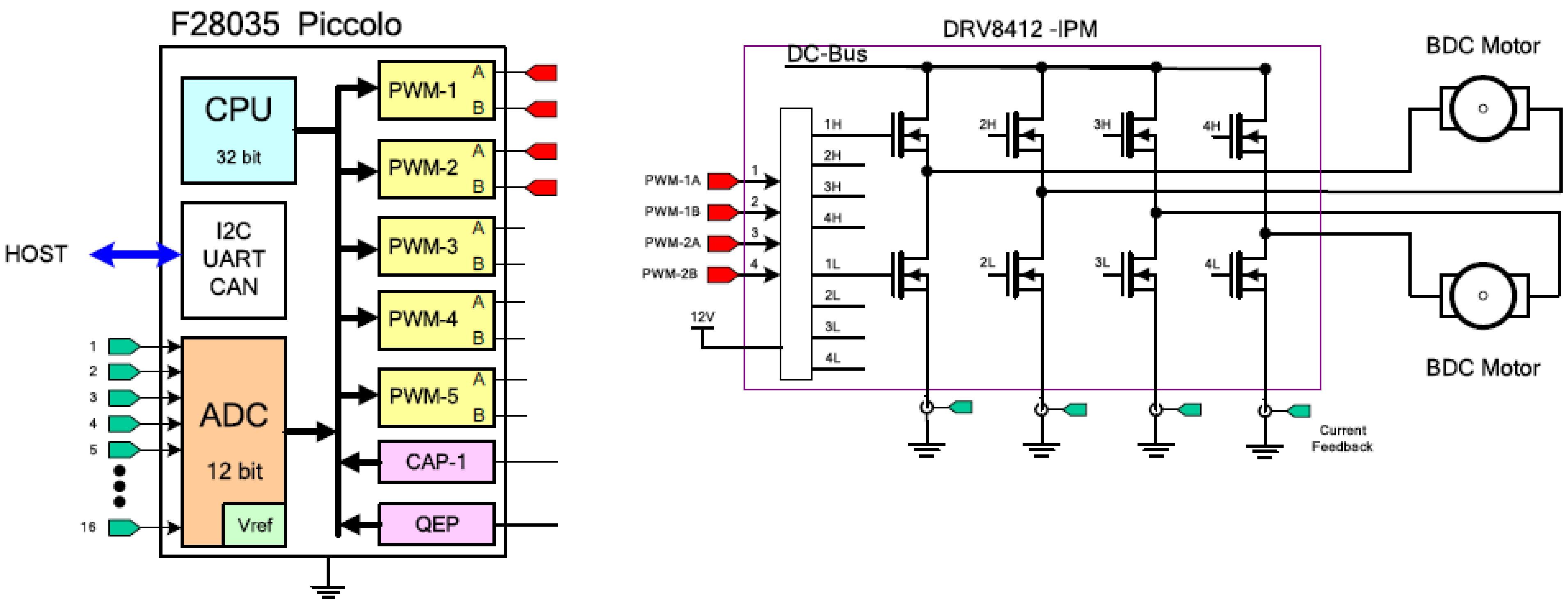
Dual Axis Motor Control Board



Controls two motors and performs PFC

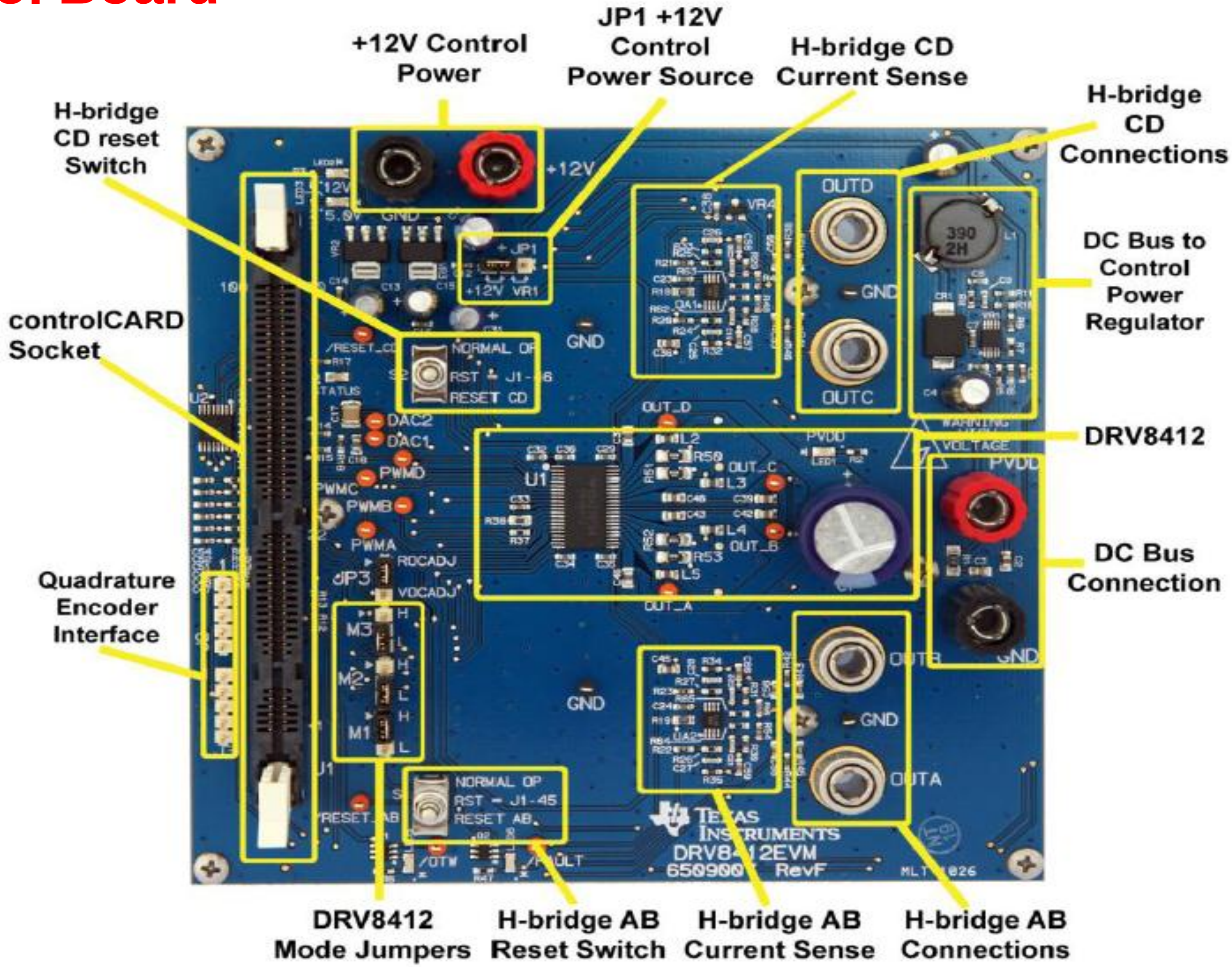
--Available--

DRV8412-C2-KIT - Motor Driver for Brushed and Stepper Motors with Piccolo F28035 controlCARD

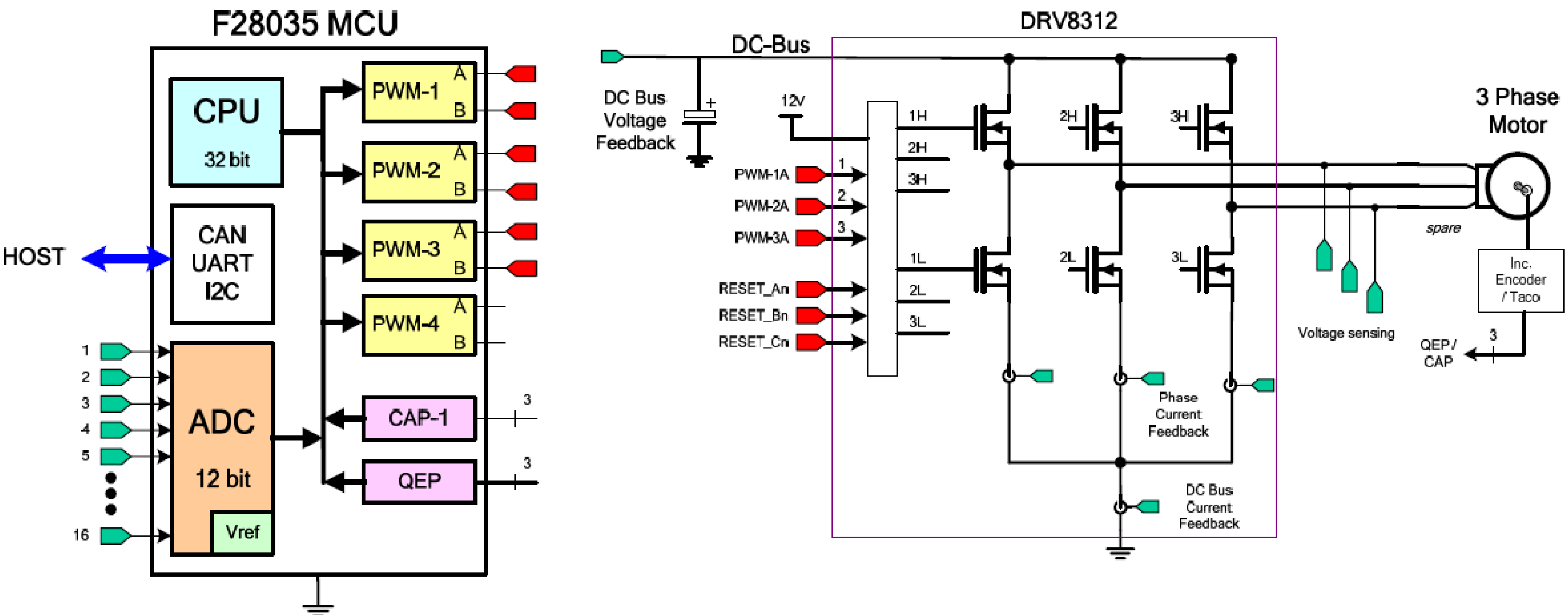


Texas Instruments

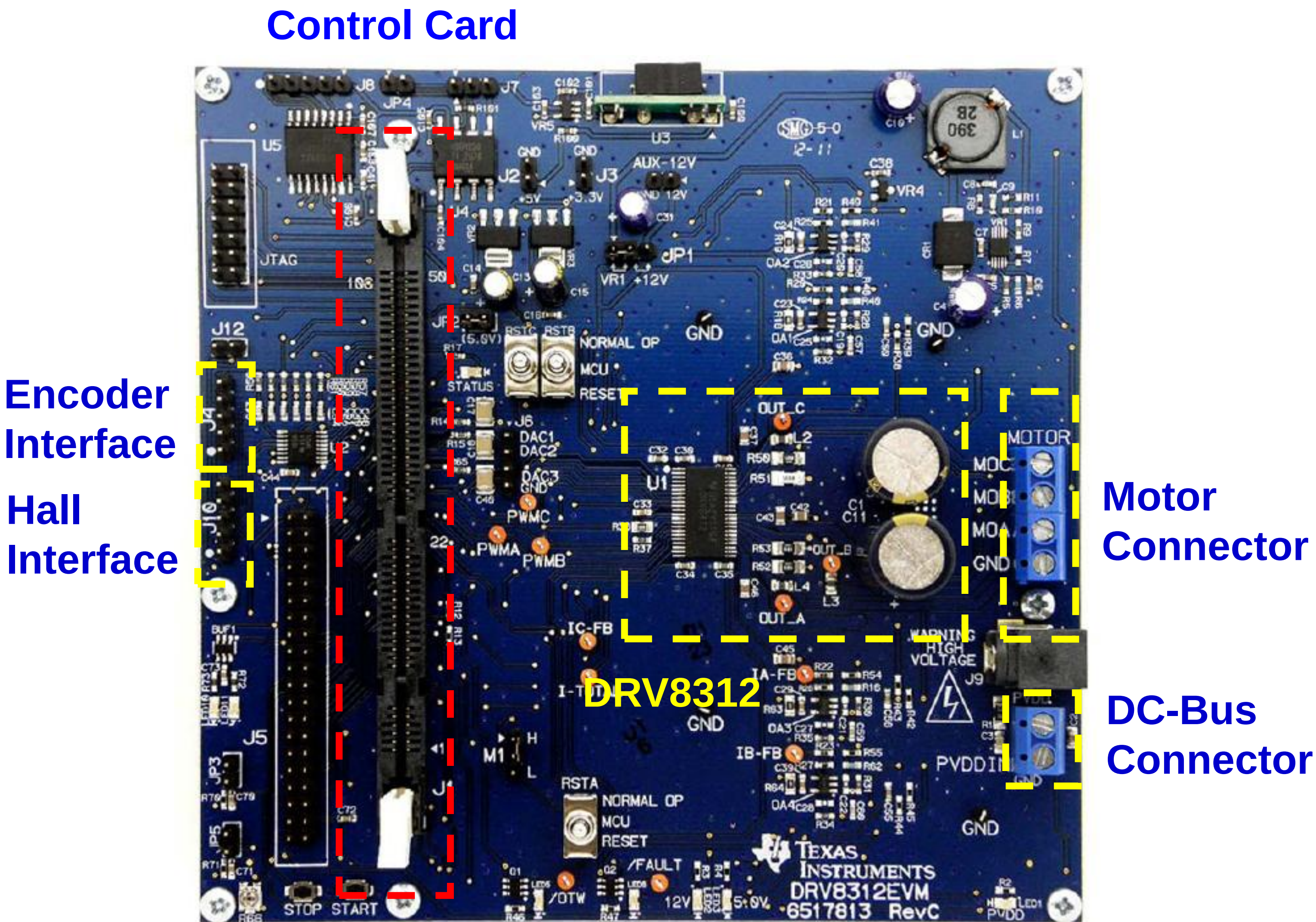
DRV8412-F28035 Brushed and Stepper Motors Control Board



DRV8312-C2-KIT - Three Phase BLDC Motor Kit with DRV8312 and F28035



DRV8312-F28035 Three Phase BLDC Motor Control Board



Getting Started

www.ti.com/c2000tools

	High Voltage PFC and Motor Control Developer's Kit	TMDSHVMTRPFCKIT This kit does NOT require an external JTAG emulator. Digital Motor Control Accessories	1.5KW digital motor control combined with 700W power factor correction using a single Piccolo MCU	$599.00 Order Now	Control SUITE

* Please install the baseline software before installing the board specific software.

Digital Motor Control Accessories

Compatible with the High Voltage PFC and Motor Control Developer's Kit

	Kit	Part Number	Description	Price	Software
	AC Induction Motor	HVACIMTR	AC Induction motor with encoder	$379.00 Order Now	Control SUITE
	Permanent Magnet Synchronous Motor	HVPMSMMTR	PMSM motor with encoder	$299.00 Order Now	Control SUITE
	Brushless DC Motor	HVBLDCMTR	BLDC motor with hall effect sensors	$199.00 Order Now	Control SUITE

Microcontrollers (MCU)

▪ MCU Products ▪ MCU Design Support ▪ Getting Started ▪ MCU Selection Tool ▪ Training

MCU Product Tree

Stellaris® ARM® Cortex™-M3-based MCUs (172)
- 1000 Series (40)
- 2000 Series (26)
- 3000 Series (10)
- 5000 Series (26)
- 6000 Series (19)
- 8000 Series (12)
- 9000 Series (9)
- X00 Series (30)

MSP430™16-bit Ultra-Low Power MCUs (231)
- 1xx 8MHz Series (36)
- 2xx 16MHz Series (39)
- G2xx Value Series (9)
- 3xx Legacy 8MHz Series (16)
- 4xx 8/16MHz LCD Series (87)
- 5xx 25MHz Series (36)
- CC430 RF SoC Series (8)

C2000™ 32-bit Real-time MCUs (71)
- 28x Delfino™ Floating-point Series (9)
- 28x Piccolo™ Series (26)
- 28x Fixed-point Series (23)
- 24x 16-bit Series (13)

ARM® Cortex™-R4F-based MCUs (12)
- TMS570LS Series (12)

16-bit performance for an 8-bit price

New Ultra-Low Power MSP430™ microcontroller Value Line starts at 25 cents (in 100K volumes) to give 8-bit developers up to 10X performance and 10X battery

C2000™ 32-bit Real-time MCUs

Overview | Product Search | Tech Docs | Tools | **controlSUITE** | Training | Partners

controlSUITE™

A comprehensive software platform for all your needs. Begin with device-specific peripheral configuration packs, which contain the baseline software and examples for you to hit the ground running. When you're ready to harness C2000's abilities in a real application, simply add the application libraries or system examples.

Run controlSUITE™ Installer

Run our smart installer and pick the packages you would like to use. All dependencies will be automatically selected, downloaded and installed.

- View the controlSUITE Getting Started Guide.
- View the controlSUITE brochure

Visit the pre-controlSUITE software page for CCSv3.3 software.

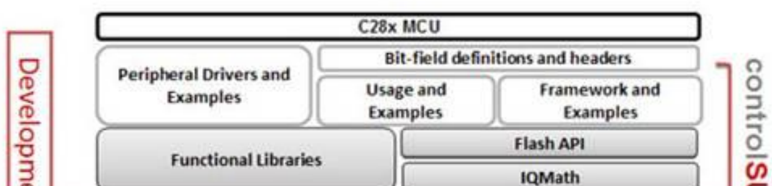
Learn more about

Device Support | Libraries | Application Kit Packages | Debug and Software Tools

Looking deeper into controlSUITE's contents, you can see how it's a comprehensive top-down set of software and software tools to guide development from beginning to end

Device Support

Hardware abstraction to get you started quickly regardless of application. These packages are family-specific.



controlSUITE:

Content + Content Management

- **C2000 Device**
  - Piccolo F2802x
  - Piccolo F2803x
  - Delfino C2834x
  - Delfino F2833x
  - F2823x
- **Device Support**
  - Bit Fields
  - API Drivers
  - Examples
- **Library Repository**
  - Math Library
  - IQMath
  - DSP Library
  - Application Library
  - Utilities
- **Development Kits**
  - Hardware Package
  - Software Examples
  - System Framework
  - Graphical User Interfaces
- **Debug and Software Tools**
  - IDE
  - RTOS
  - Real-time Debug
  - 3rd Party Tools

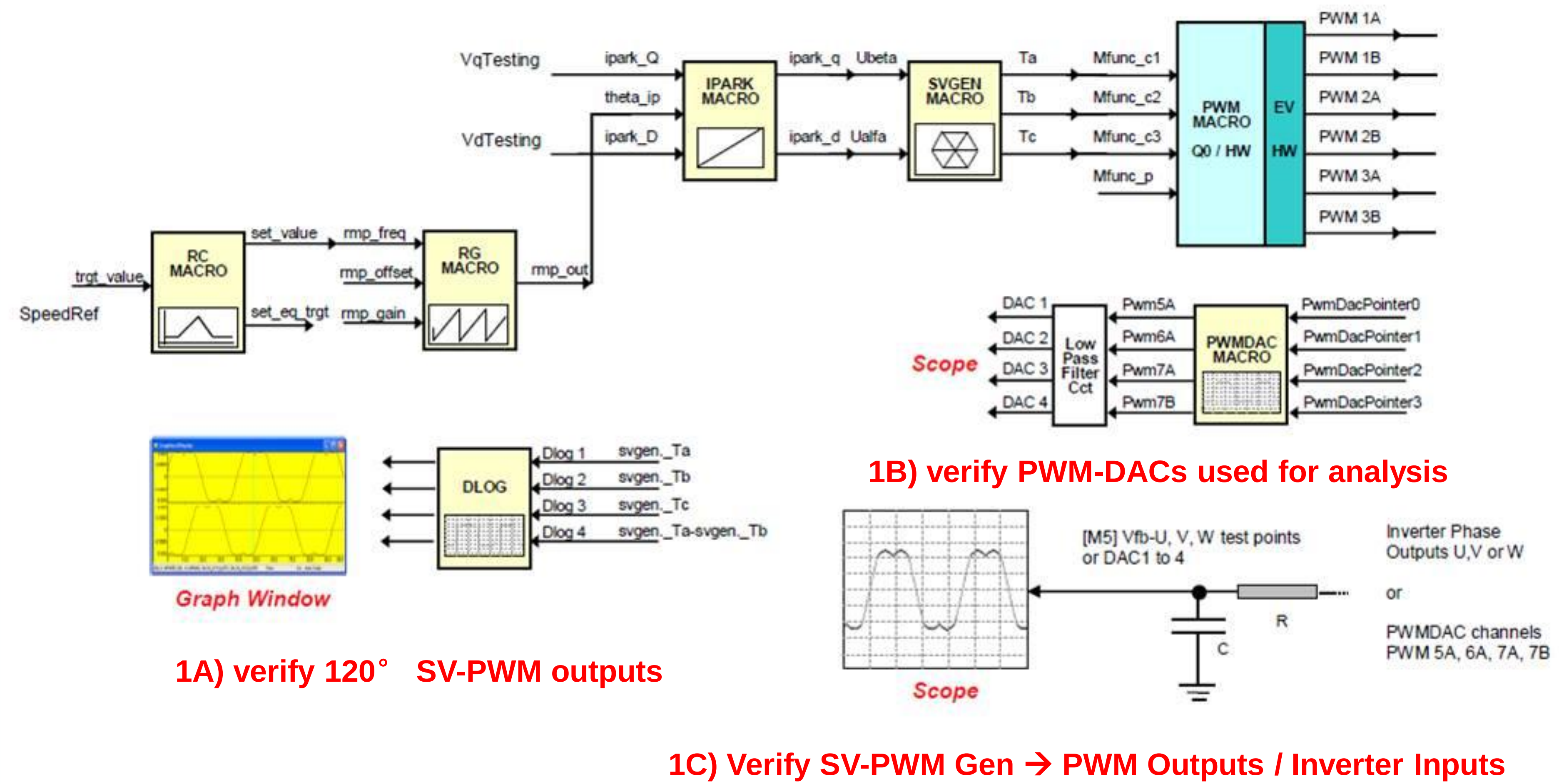
Please see the Tools Presentation for Details

Content

- Motor Control Methods Overview
- Inverter / Hardware considerations
- Software / Algorithm Considerations
- DMC Development Kits
- System Incremental Build
- Demo, Q&A

Incremental Build

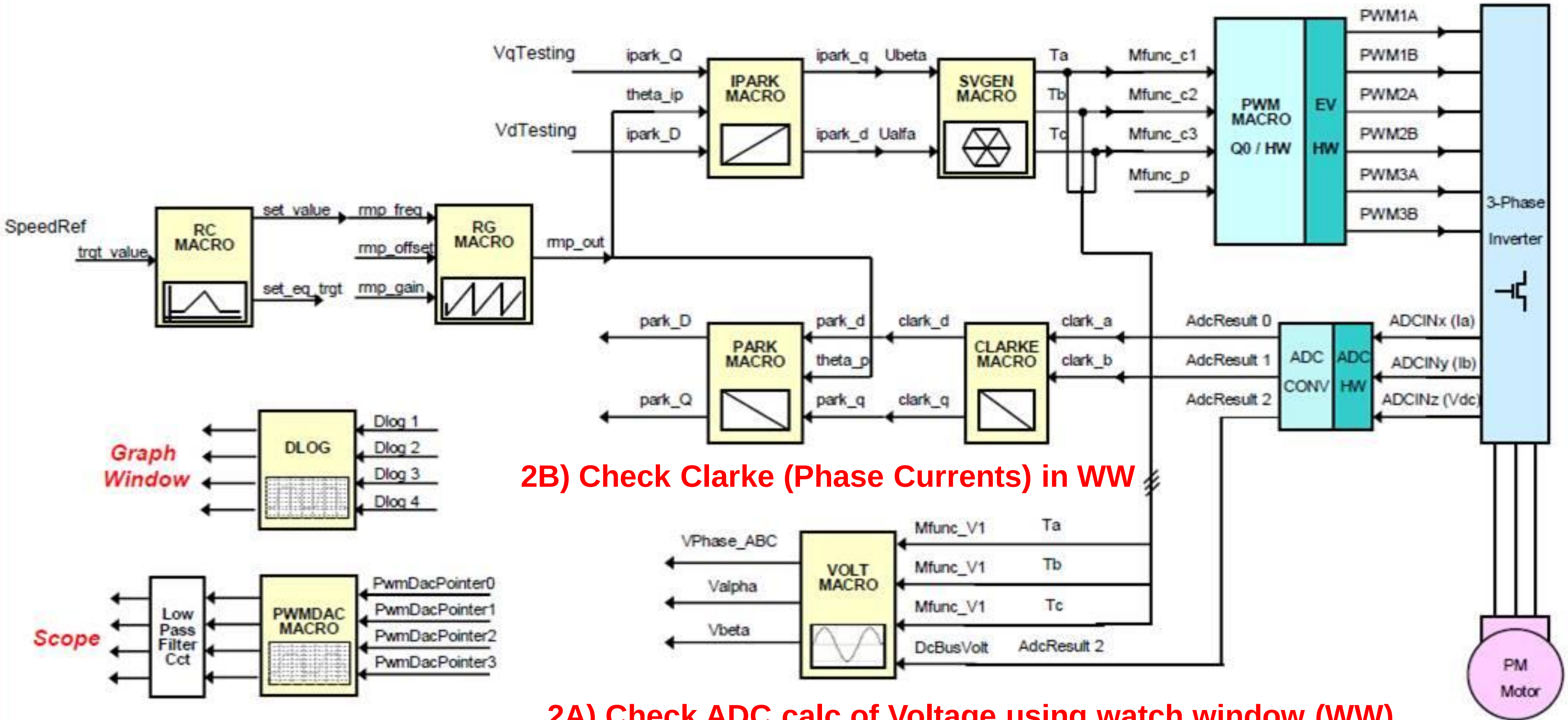
Level 1 - Incremental System Build Block Diagram



Level 1 verifies the target independent modules, duty cycles and PWM update. The motor is disconnected at this level.

Incremental Build

Level 2 - Incremental System Build Block Diagram



2B) Check Clarke (Phase Currents) in WW

2A) Check ADC calc of Voltage using watch window (WW)

2C) Calibrate phase current off-set to enable low load sensorless

Level 2 verifies the analog-to-digital conversion, offset compensation, clarke / park transformations, phase voltage calculations

Real-Time Debug

Traditional debugging (Stop Mode)

- stops all threads and prevents interrupts from being handled
- makes debugging real-time systems extremely difficult

C2000 Real-time Mode:

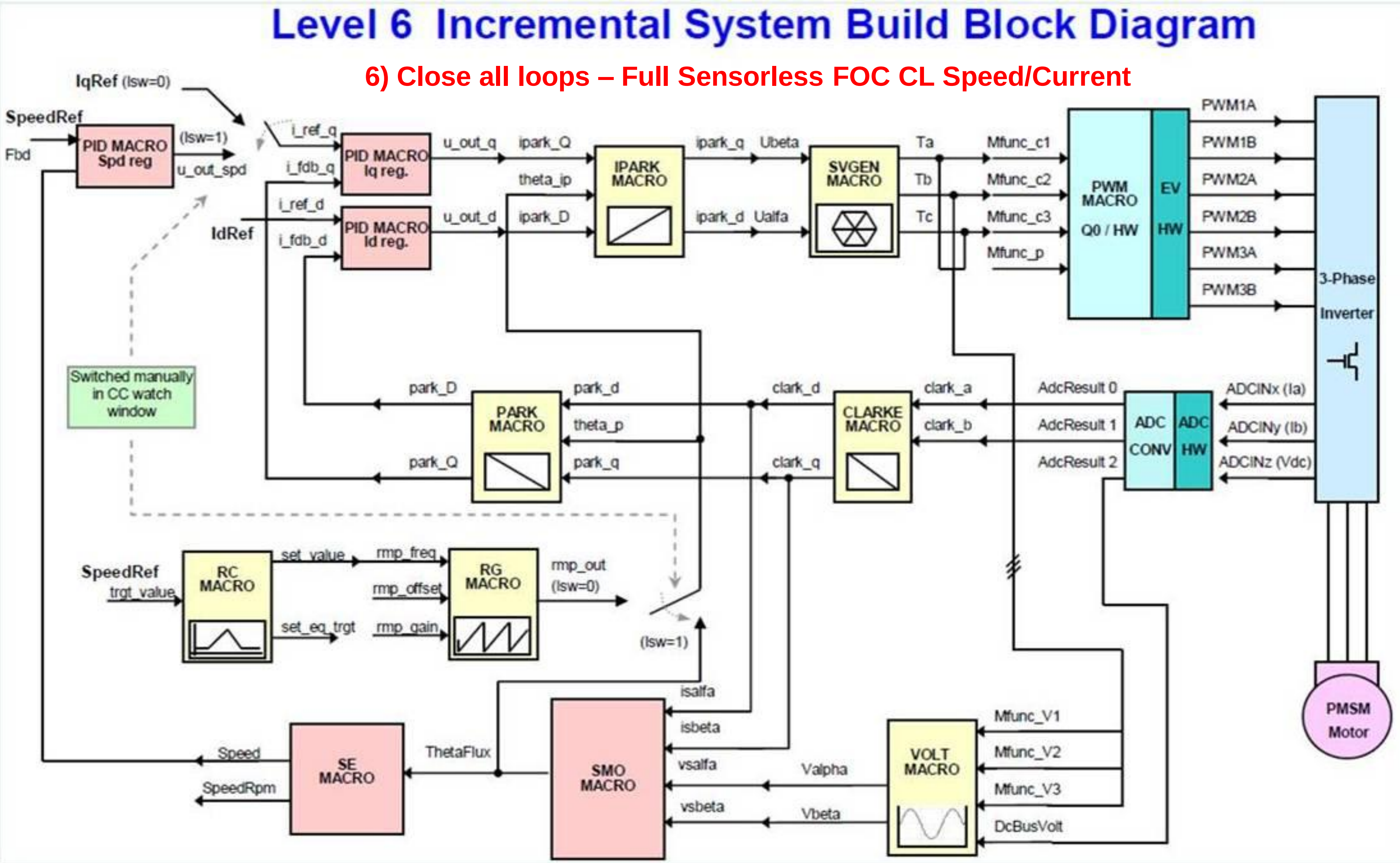
- real-time, non-intrusive, continuous
- Does not require use of target memory, special interrupts, or SW intrusiveness
- Allows time critical interrupts to be marked for special treatment (high priority)
- Allows time-critical interrupts to be serviced while background program execution is suspended
- Included on all C2000 devices and integrated with Code Composer Studio

Resources:

Real-Time Mode on wiki

Chapter 7.4 in the C28x CPU Reference Guide

Incremental Build



Demo, Q&A

Thanks!