Win with InstaSPIN™

EDERC2014

September 11-12 2014, Milan, Italy

Univ.-Prof. Dr. ir. D.W.J. Pulle RWTH-ISEA Germany
CEO EMsynergy, Australia

Member Texas Instruments InstaSPIN Development Team

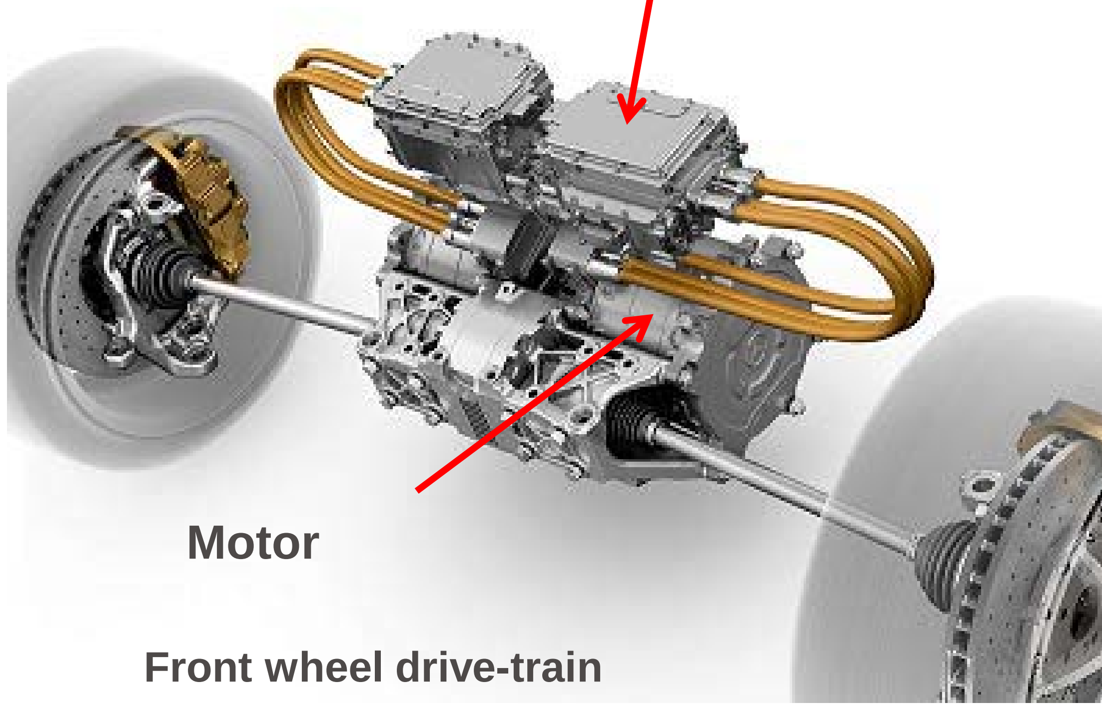
Content

- Introduction
  - what is sensorless control?
- Challenges to realise sensorless control
  - Ability to track the magnetic field in the machine
  - Need to identify and track critical motor parameters
  - Measurement of the induced voltage
- Solutions
  - use of TI InstaSPIN algorithm
  - Use of VisSim embedded control tools
  - PLECS processor in the loop (PIL) technology
- Application examples
- Questions ?

Introduction: typical electrical drive

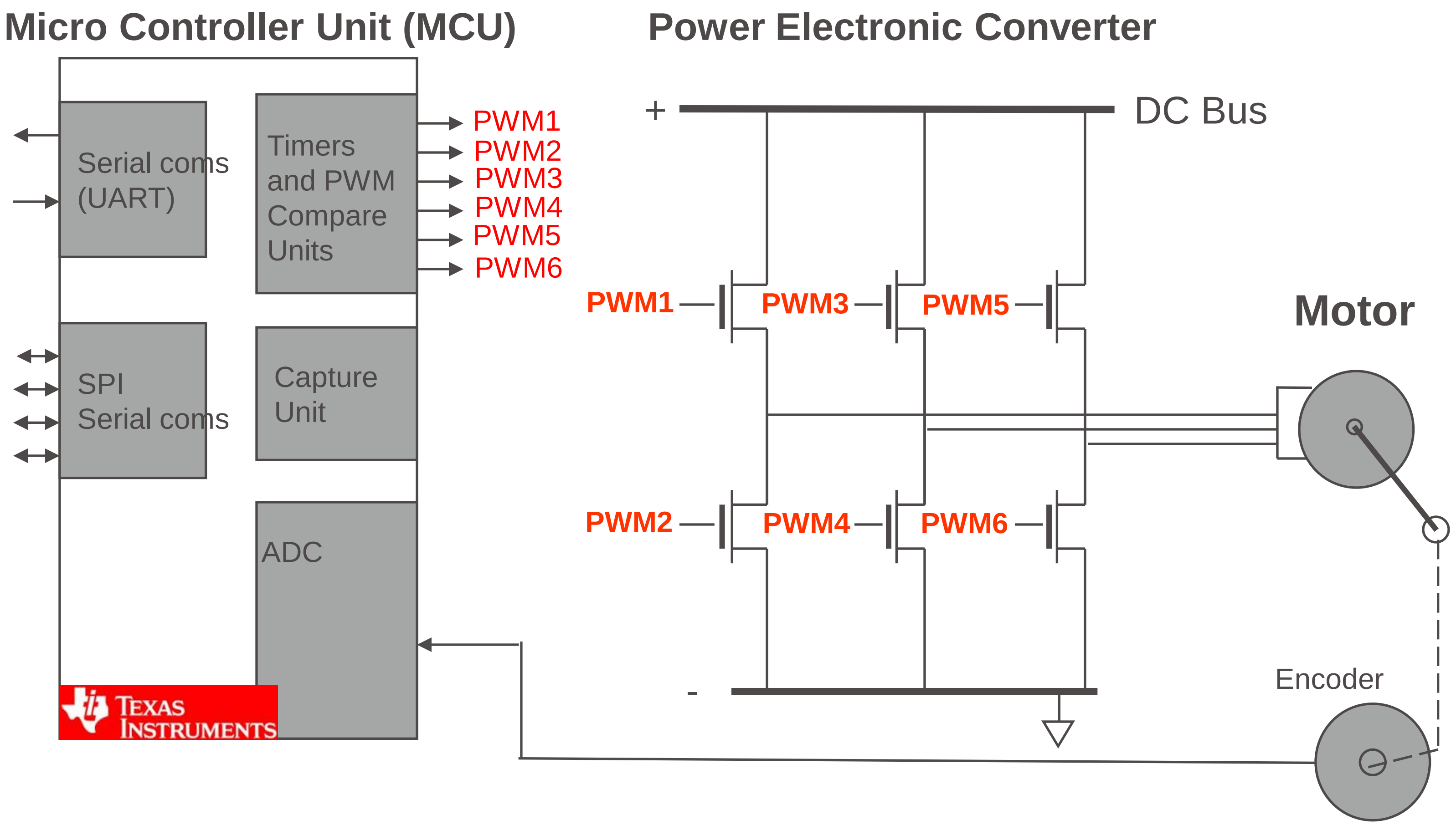
Power Electronic Converter with controller and micro-processing unit (MCU)

Mercedes SLS
Electric vehicle



➔ Schematic representation electrical drive ?

Introduction: typical electrical drive



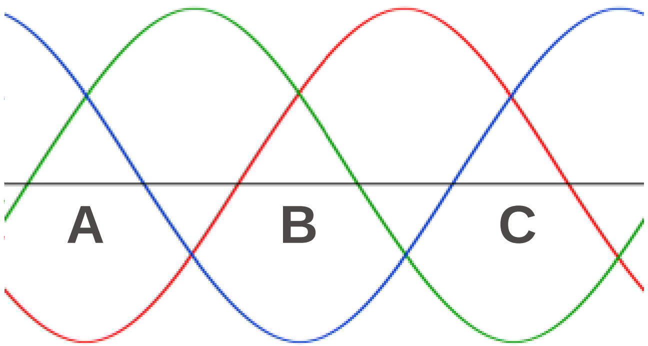
➔ What is 'Field Oriented Control' (FOC) ?

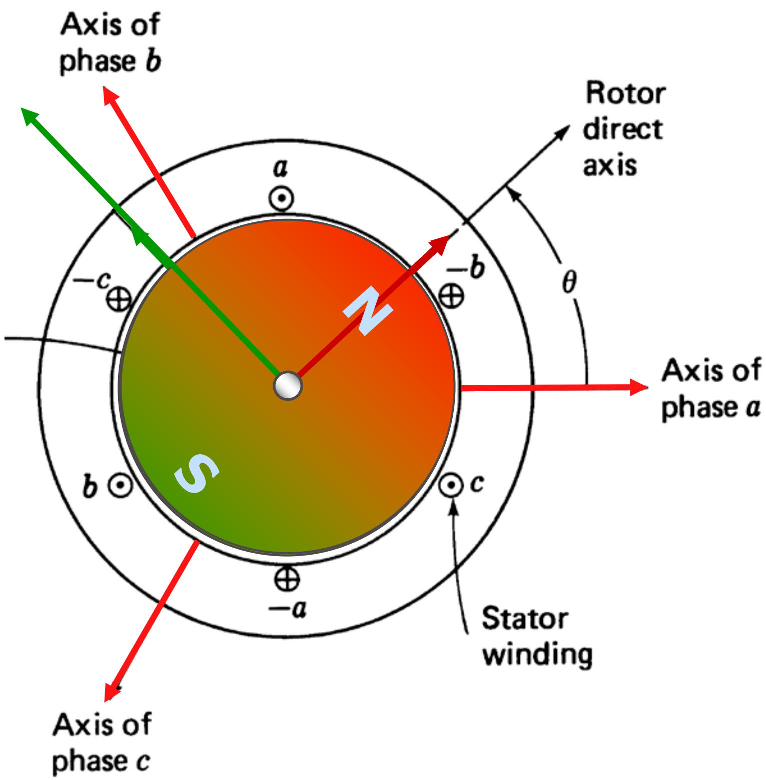
Institute for Power Electronics and Electrical Drives
RWTH AACHEN

Low Torque

Medium Torque

High Torque

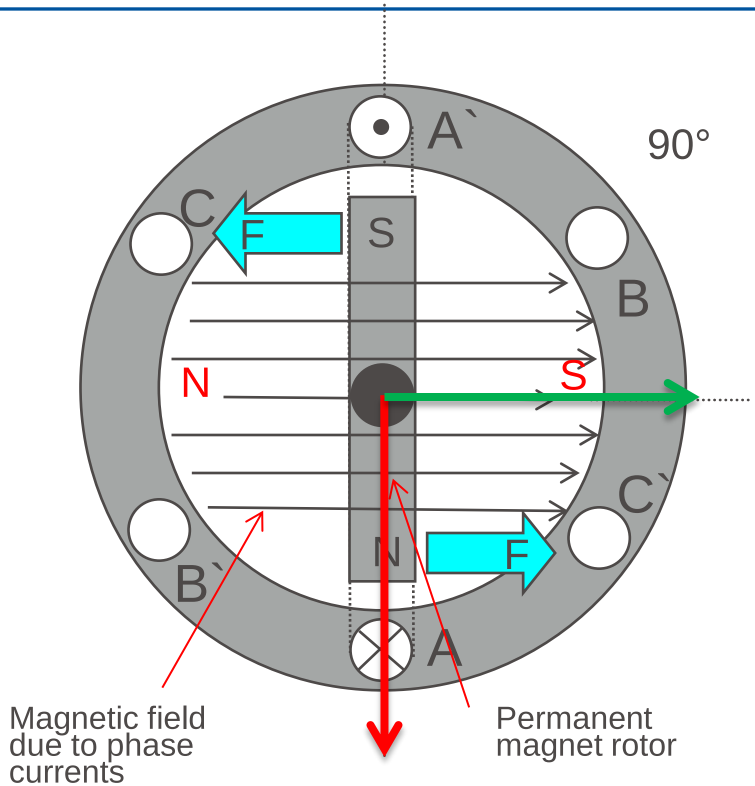


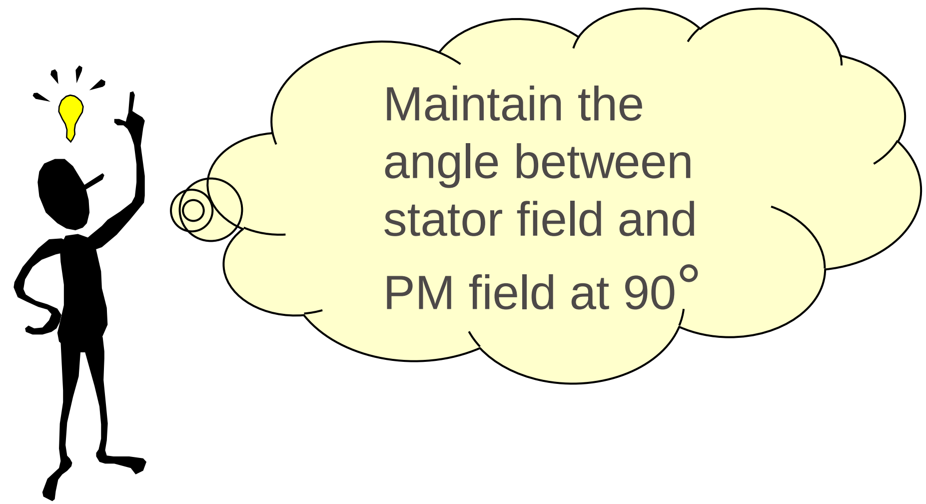


$$Torque \propto \psi_{PM} I_q$$

Adjustable

Introduction: What is 'Field Oriented control' ?

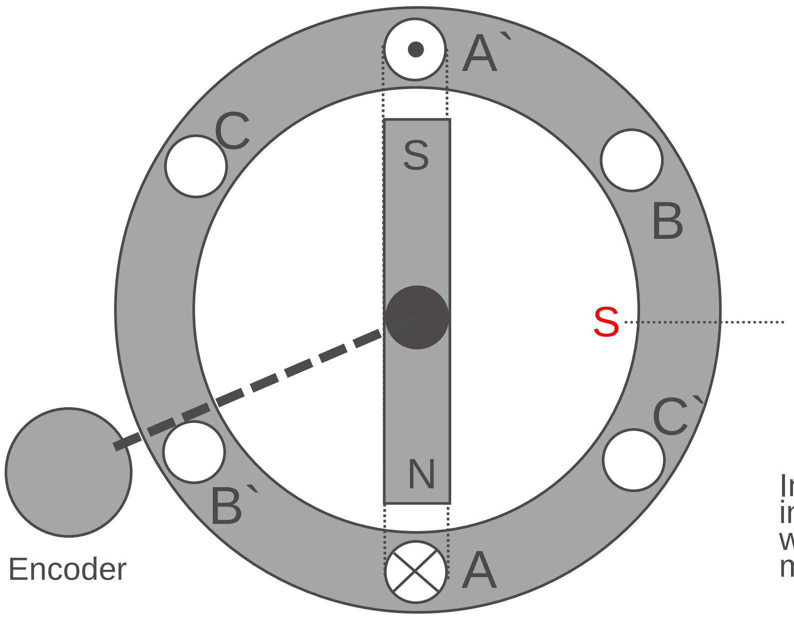




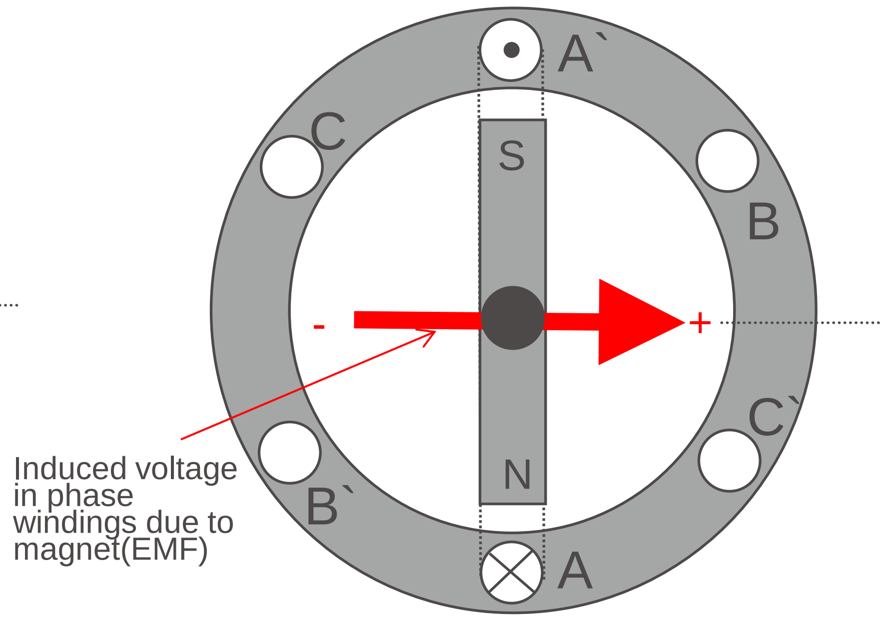
Principle of 'Field Oriented Control' (FOC) for PM [1], [2]

- Gives highest torque for the lowest current
- High dynamic response fully equivalent to DC motor
- Requires control of the currents in the stator by using the converter
- Requires knowledge of the PM magnetic field orientation

Institute for Power Electronics and Electrical Drives
RWTH AACHEN



Use of a shaft encoder

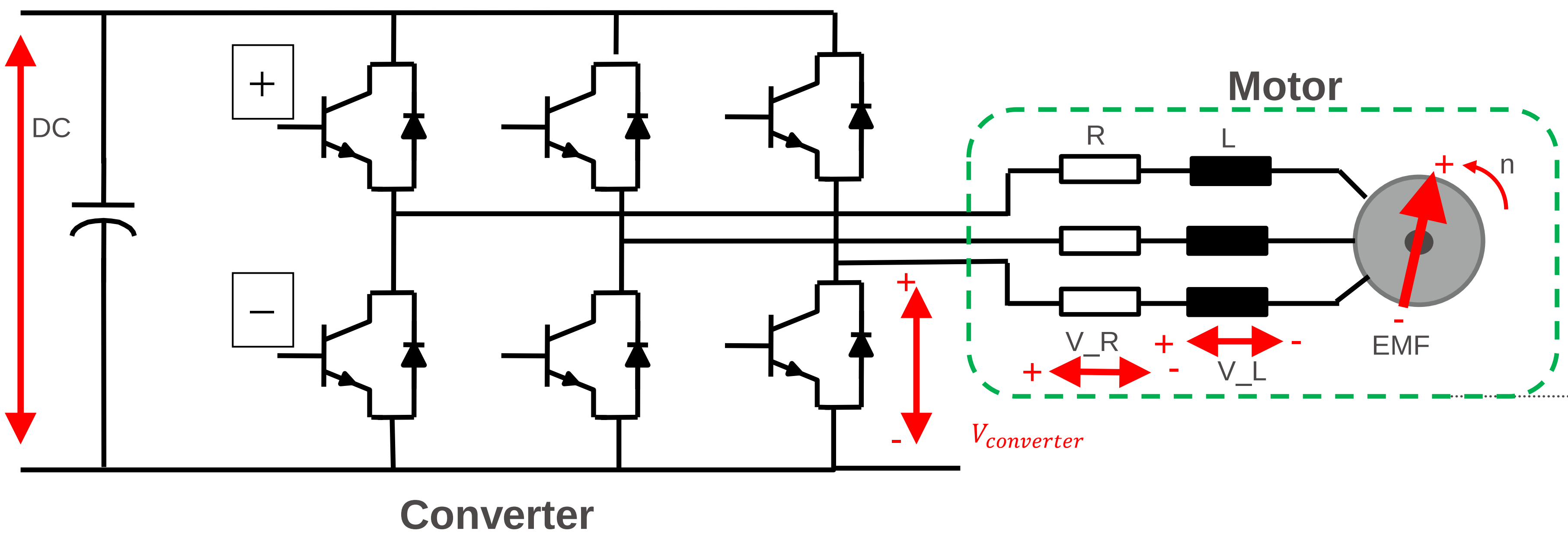


Use EMF induced by rotor magnets

- Shaft encoder: reliable position information, but expensive , increases drive complexity and requires an additional cable to MCU
- Encoder: requires access to a motor shaft end and increases motor volume
- EMF sensing required sohisticated algorithm that works at near zero speed

Challenges to achieving sensorless control

- Need to find EMF 'vector' in amplitude and orientation using
  - $EMF = V_{motor} - V_R - V_L$
  - Approach requires knowledge of the motor parameters R, L and currents



➔ Complication: motor voltage (V_{motor}) must be reconstructed from converter voltage ($V_{converter}$) which has a value V_DC or 0 V

Challenges to achieving sensorless control

- Estimate the rotor flux in terms of amplitude AND orientation
  - Undertake this at near zero speed and remain STABLE at zero speed
  - Estimate the motor parameters in order to 'reconstruct the EMF of the motor
  - Be used universally for all three-phase machines (PM and INDUCTION)
  - Accurate measurement of the motor voltages and currents

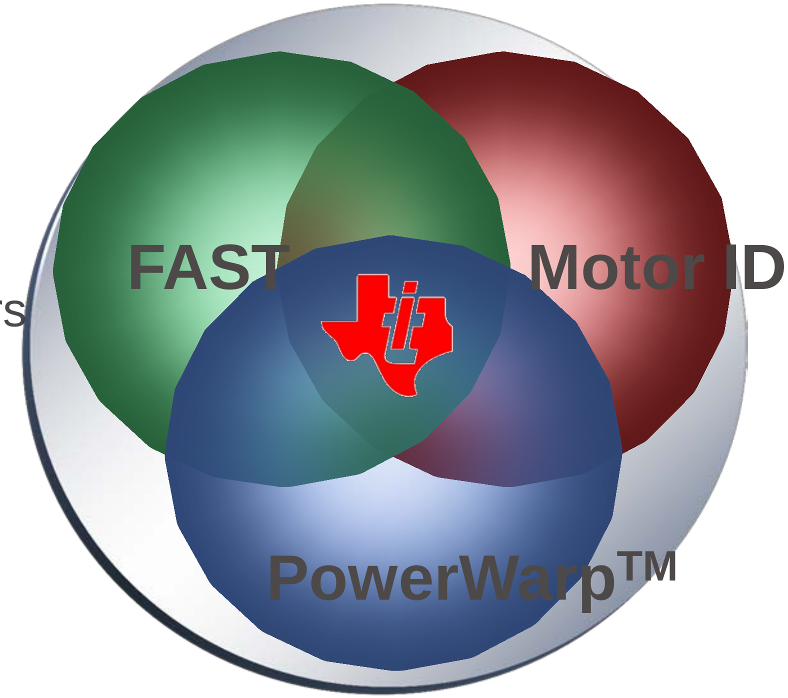
- Need for a software package than can communicate with the sensorless algorithm at a 'high level' so

  that inexperienced users can use this technology: VisSim
- Need for a software package that can evaluate sensorless operation with a model of the machine and converter: PLECS PIL

➔ Solution??

Solution to realizing sensorless control

- Use of Texas Instruments InstaSPIN™-FOC [3]
  - A new advanced field oriented control technique for sensorless control of permanent magnet (salient and non-salient) and Induction motors
  - Comprehensive self- commissioning capabilities
  - On-line estimation of key variables
  - Relatively easy to use by inexperienced users
- What are the components ?
  - FAST algorithm that provides:
    - Flux amplitude , Angle and Speed of the flux vector and machine Torque
  - Motor ID:
    - Identification of motor parameters
  - PowerWarp:
    - Energy efficient induction machine operation under partial load

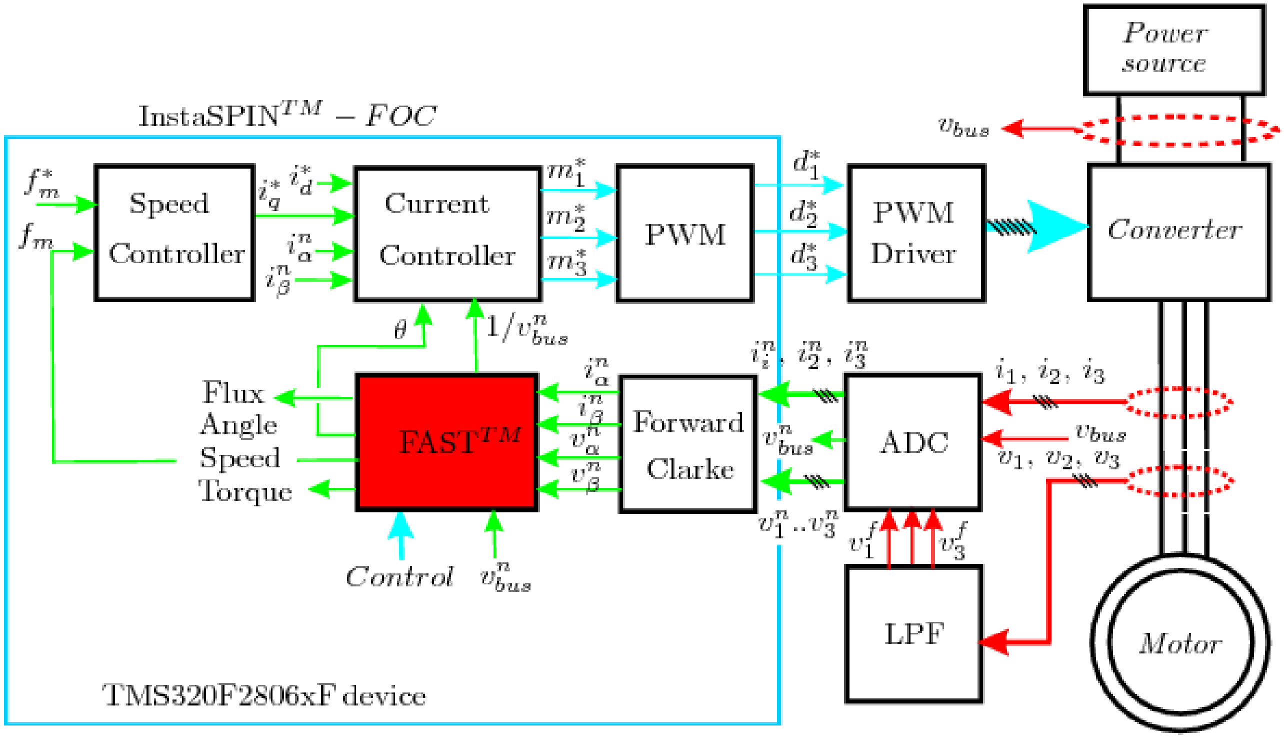


➔ System architecture using InstaSPIN-FOC?

Solution to realizing sensorless control

Institute for Power Electronics and Electrical Drives
RWTH AACHEN

- System architecture of InstaSPIN-FOC located in MCU: TMS320F280xF

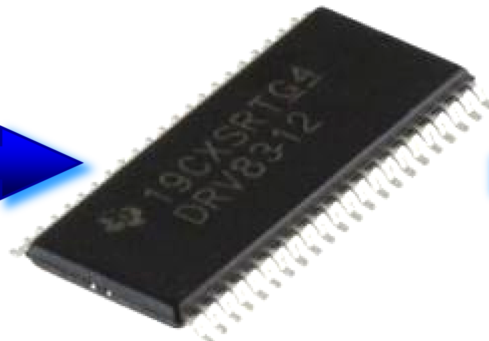


- FAST module: provides flux amplitude/angle/speed and torque
- required speed and current control algorithms to achieve FOC

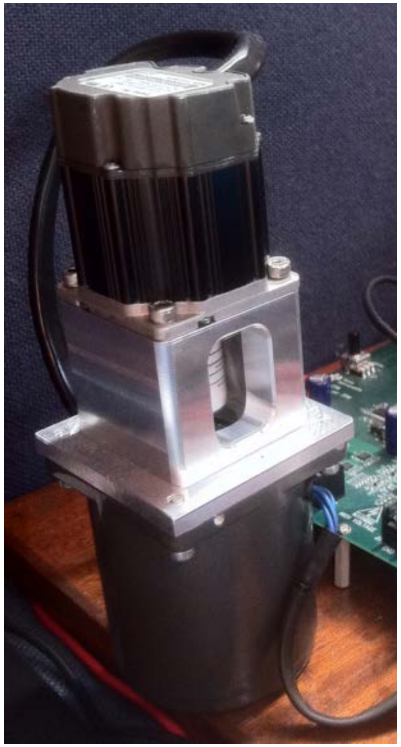
➔ Tool set ?

Solution to realizing sensorless control

027F-069M MCU



DRV8312 Kits
DRV8301 Kits



High Voltage Motor Control + PFC Kit

C2000 Launchpad (027 and 069) and Booster Pack

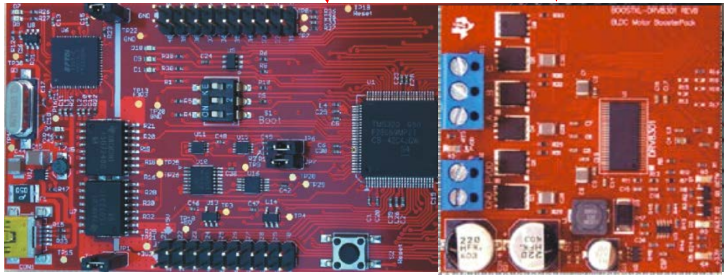
TI PM/IM Motors

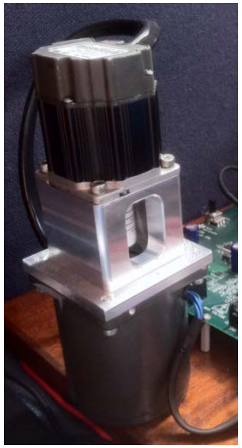
InstaSPIN-FOC

- BLDC FOC
- ACIM FOC (3 and 1 phase)
- PMSM FOC
- IPM FOC
- Stepper FOC (Pending release)

Solution to realizing sensorless control

- Development solutions:
- System components:
  - DRV 8312-069M -kit (24V/3.5A)
  - DRV 8301-069M -kit (60V/40A)
  - HV+PFC kit (350V/10A)
  - LaunchPad (027 and 069M) with Booster pack (24V/10A)





➔ What development tools are available to apply InstaSPIN??

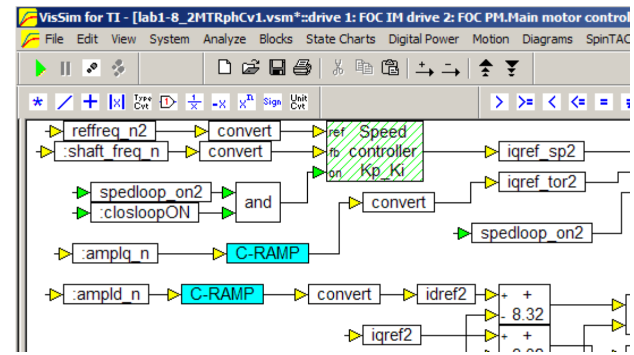
InstaSPIN™ development tools

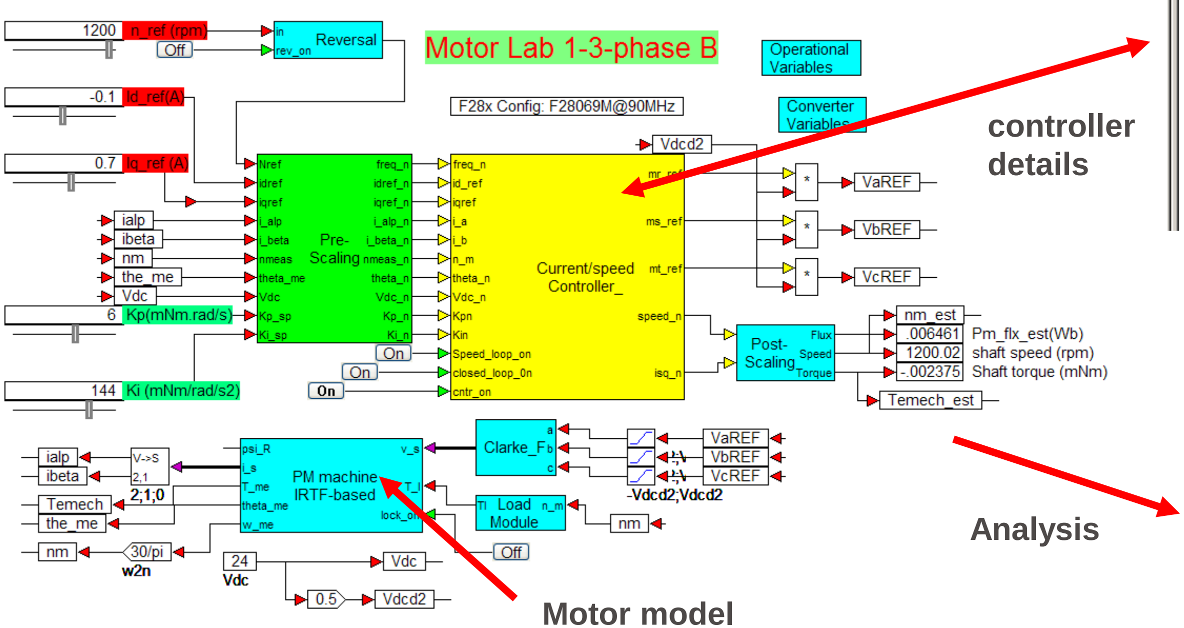
- VisSim™
  - Embedded control software
  - Use of custom designed InstaSPIN™ modules
- MotorWare™
  - Texas Instrument development tools
  - Includes set of MotorWare™ laboratories which use InstaSPIN™
- PLECS ™
  - Software simulation environment with electrical and control block set
  - PLECS processor in the loop (PIL) technology, works together with (PIL prepared) MotorWare™ laboratories that use InstaSPIN™

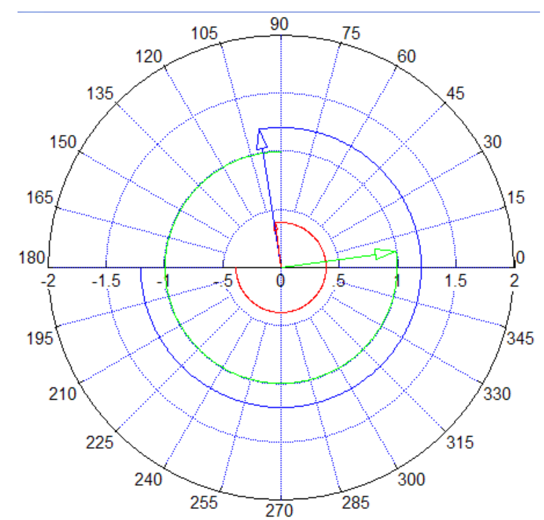
➔ Closer look at development tools

InstaSPIN™ development tools

- Example of VisSim [4] 'embedded control' development software
  - Ability to develop and veritfy a fixed point control algorithm for the MCU and test this with a simulated motor model first



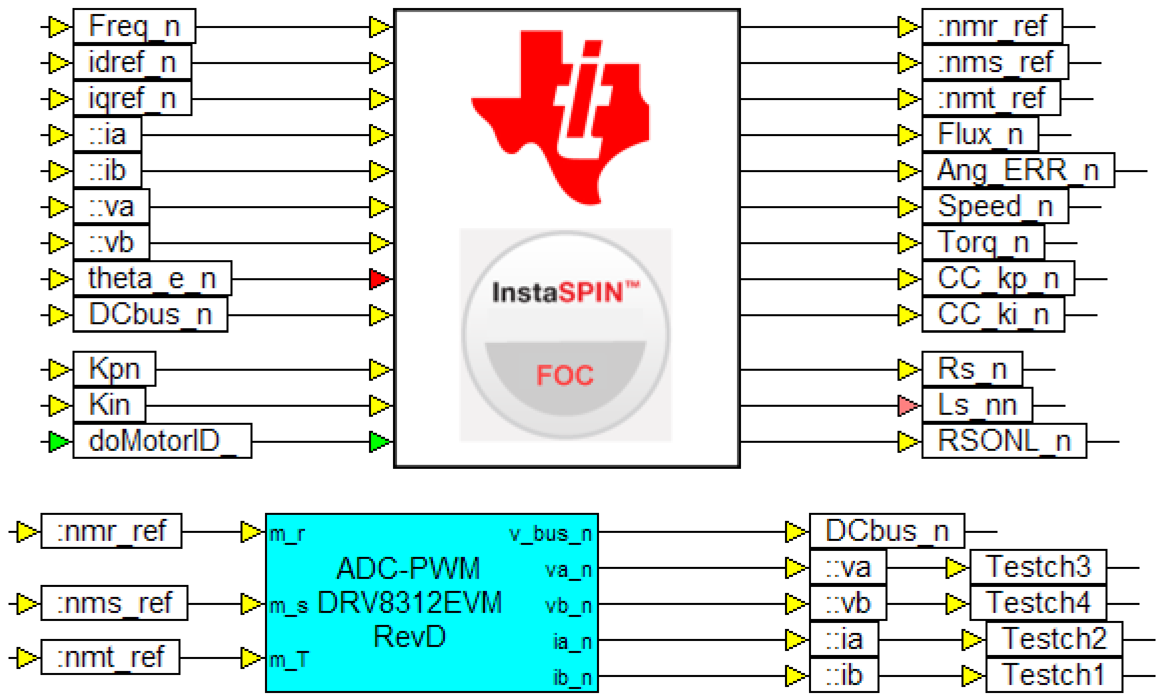




➔ VisSim for sensorless control using InstaSPIN?

InstaSPIN™ development tools

- Use of VisSim to develop complete sensorless embedded controller structure
- Modules present:
  - InstaSPIN-FOC : specially designed VisSim module [5] which executes the FAST algorithm and all control tasks
  - ADC-PWM: module used to obtain the measured voltage/currents and controls the power electronic converter

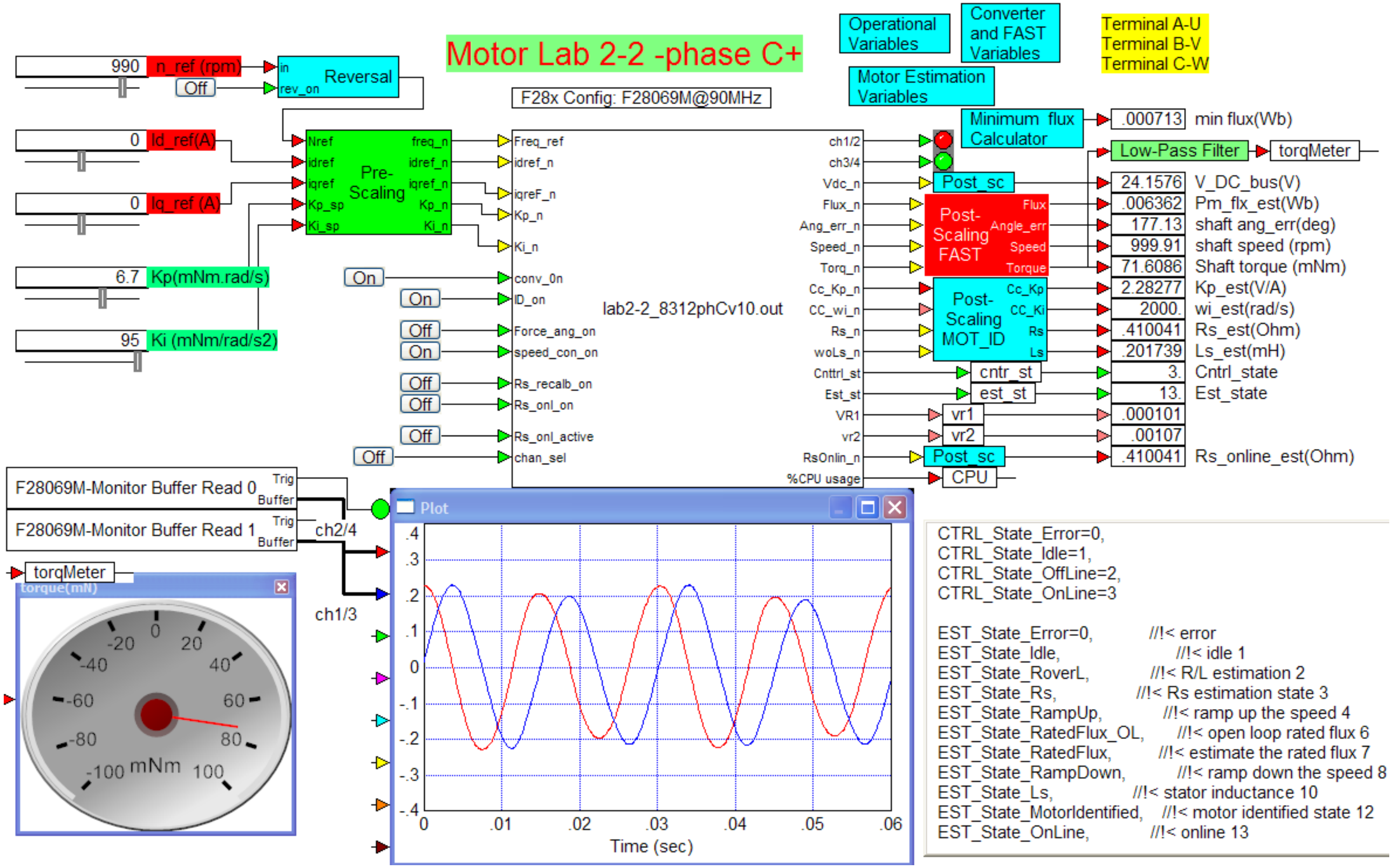


➔ Compilation of this module to C-code generates an 'outfile' that runs the drive

InstaSPIN™ development tools

Institute for Power Electronics and Electrical Drives
RWTH AACHEN

- Example of a VisSim based controller which operates a sensorless drive



➔ Example from Texas Instruments InstaSPIN- VisSim workshop program [5]

InstaSPIN™ development tools

Institute for Power Electronics and Electrical Drives
RWTH AACHEN

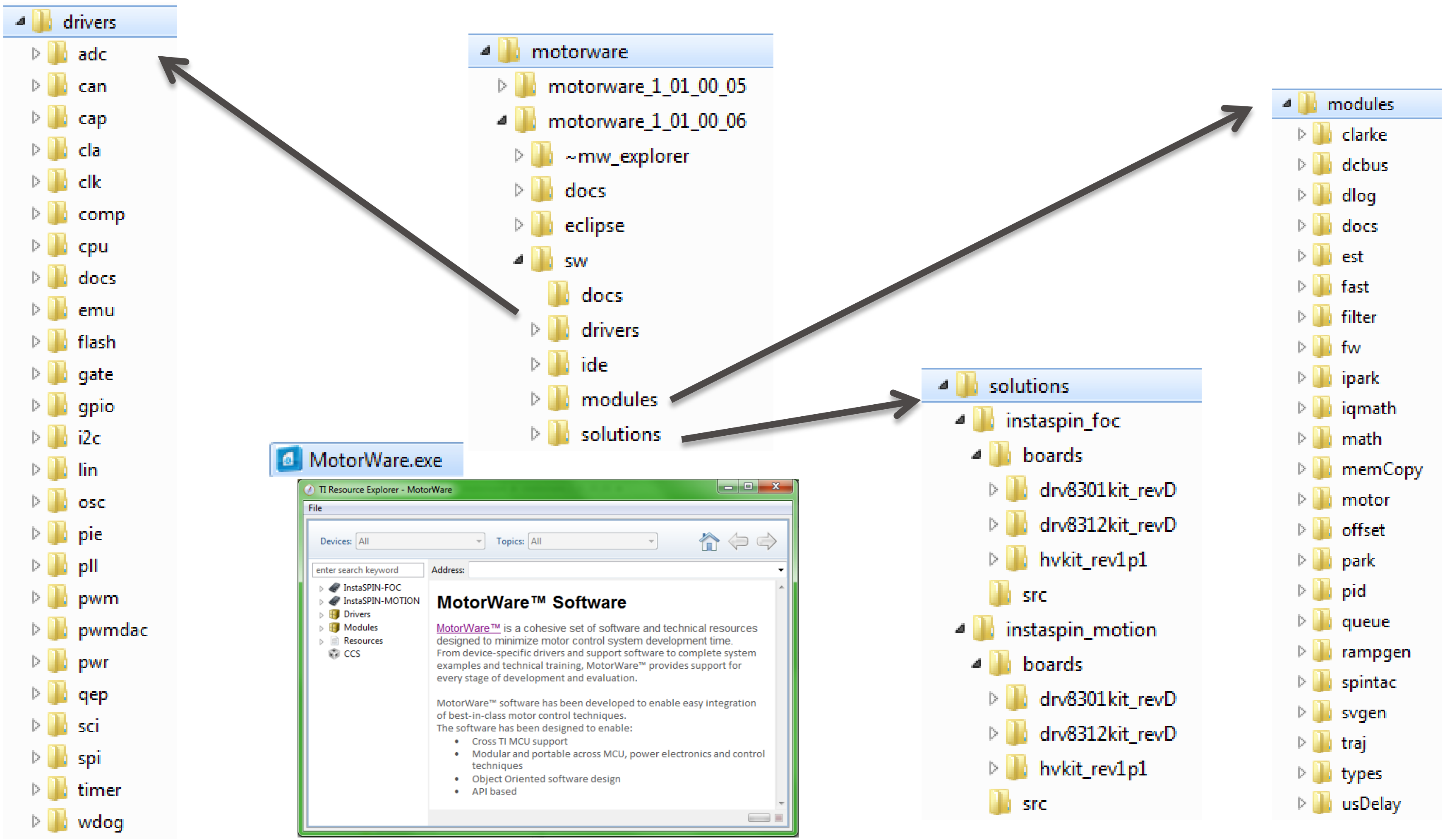
- Introduction to Texas Instruments MotorWare Software
- Content:

  - Motor Ware software structure
  - InstaSPIN implementation
  - Use of MotorWare to run VisSim lab 2-2 (previous slide) but implementation using CCS instead of VisSim

InstaSPIN™ development tools

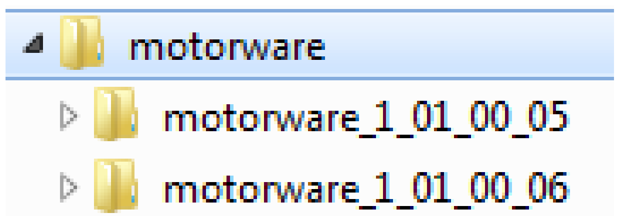
Institute for Power Electronics and Electrical Drives
RWTH AACHEN

- MotorWare™ is a repository to build solutions:



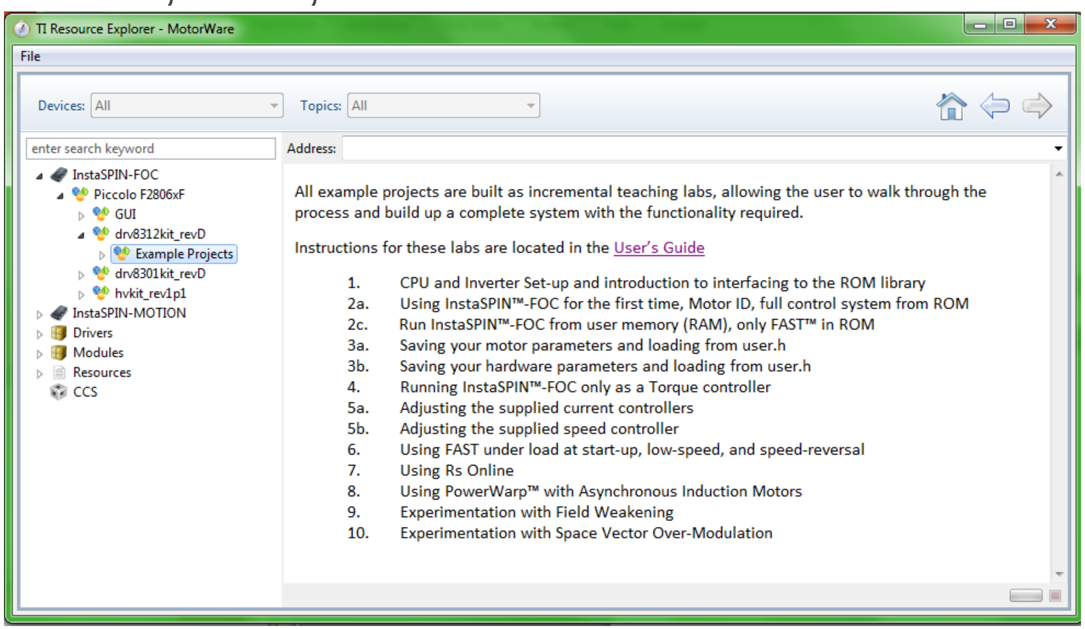
InstaSPIN™ development tools

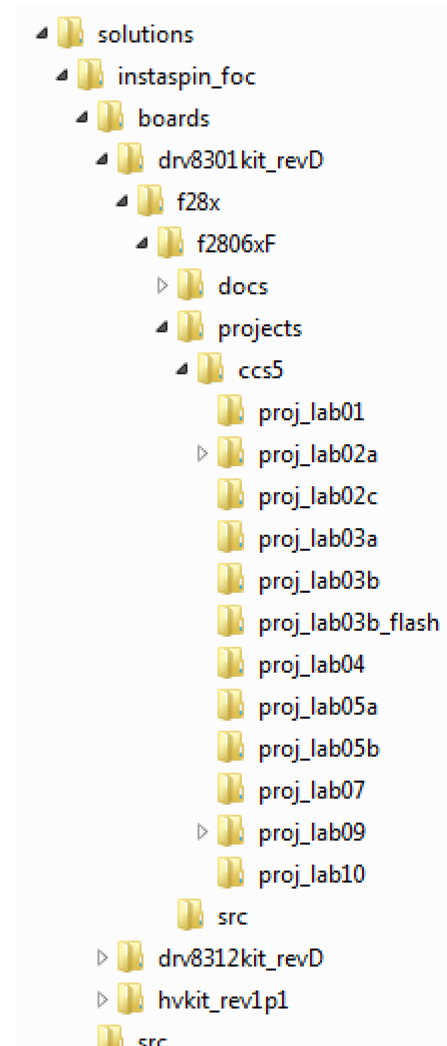
- MotorWare™ contains 99% source code
  - ➔ Source: Drivers, Modules, most code used in any solution
  - ➔ Closed source is only some key IP
    - Ex: FAST software sensor (ROM), initial position detection, etc.
- MotorWare™ is under revision and collaboration control using GIT inside TI
- Entire repository is versioned for public release



Institute for Power Electronics and Electrical Drives
RWTH AACHEN

- MotorWare™ contains multiple example projects for each solution
  - ➔ The main proj_lab##.c and ALL other files are *common* through the instaspin_foc solution *across* all boards and MCUs
  - ➔ To port to a new board only three files change: drv.c, drv.h, user.h





- MotorWare™ contains code that produces html documentation using Doxygen

```
//! \brief     Gets the angle value from the estimator in per unit (pu), IQ24.
//! \details   This function returns a per units value of the rotor flux angle. This value wraps around
//!            at 1.0, so the return value is between 0x00000000 or _IQ(0.0) to 0x00FFFFFF or _IQ(1.0).
//!            An example of using this angle is shown:
//! \code
//! _iq Rotor_Flux_Angle_pu = EST_getAngle_pu(handle);
//! \endcode
//! \param[in] handle  The estimator (EST) handle
//! \return    The angle value, pu, in IQ24.
extern _iq EST_getAngle_pu(EST_Handle handle);
```

_iq EST_getAngle_pu (EST_Handle handle)

Gets the angle value from the estimator.

This function returns a per units value of the rotor flux angle. This value wraps around at 1.0, so the return value is between 0x00000000 or _IQ(0.0) to 0x00FFFFFF or _IQ(1.0). An example of using this angle is shown:

```
_iq Rotor_Flux_Angle_pu = EST_getAngle_pu(obj->estHandle);
```

Gets the angle value from the estimator.

Parameters:

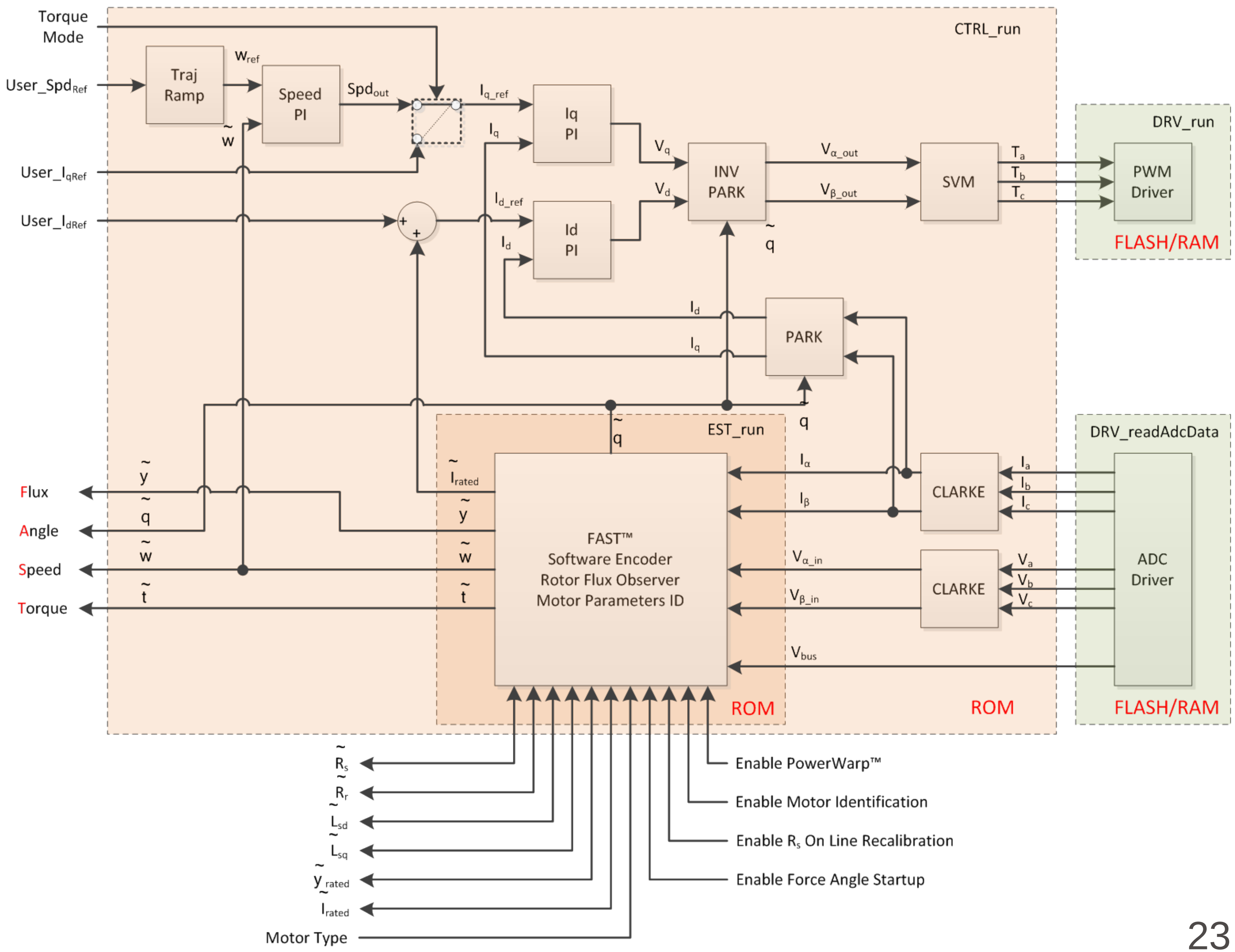
[in] *handle* The estimator (EST) handle

Returns:

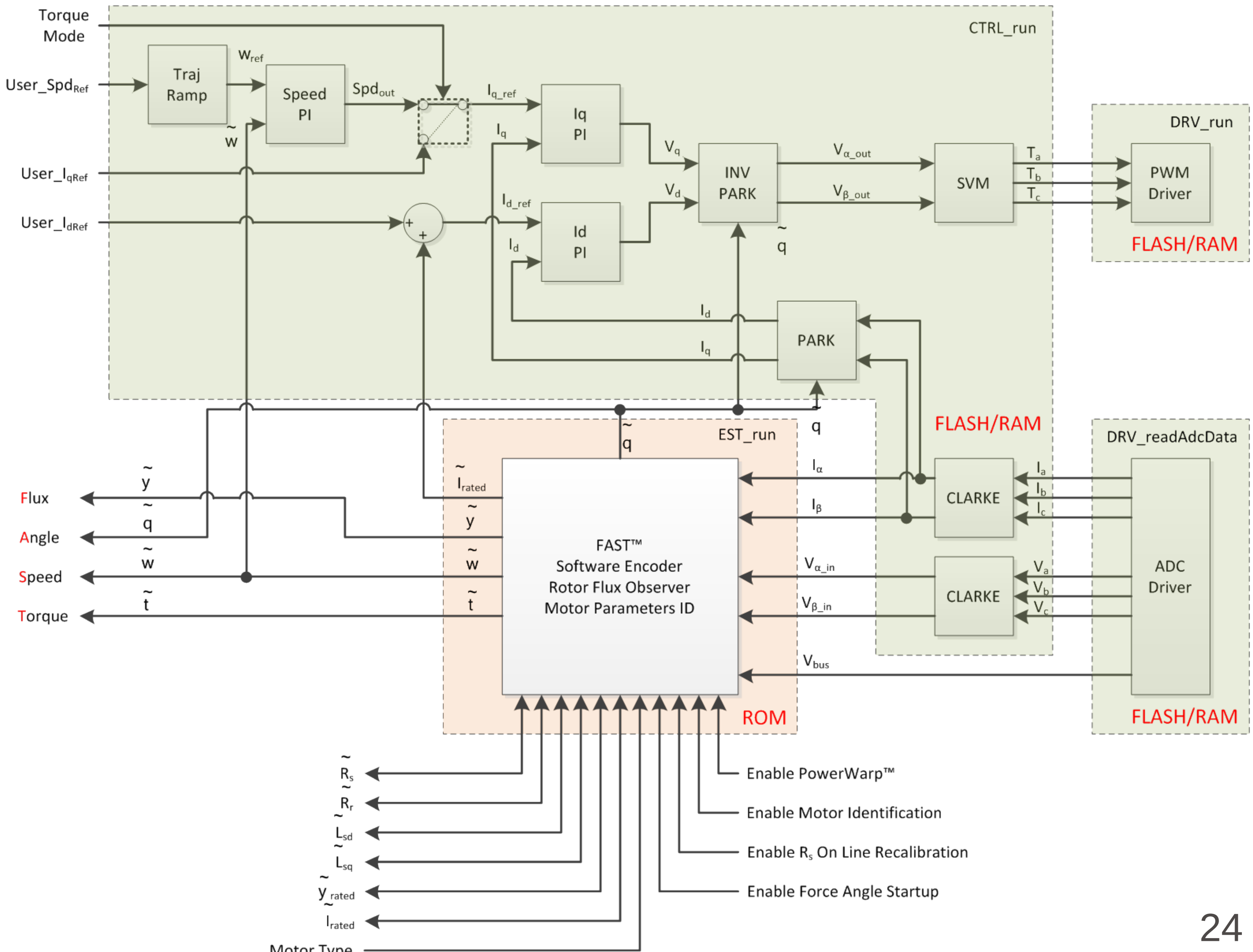
The angle value, pu, in IQ24.

Institute for Power Electronics and Electrical Drives
RWTHAACHEN

All FOC in ROM



Estimator Only in ROM



InstaSPIN™ development tools

SW Structure

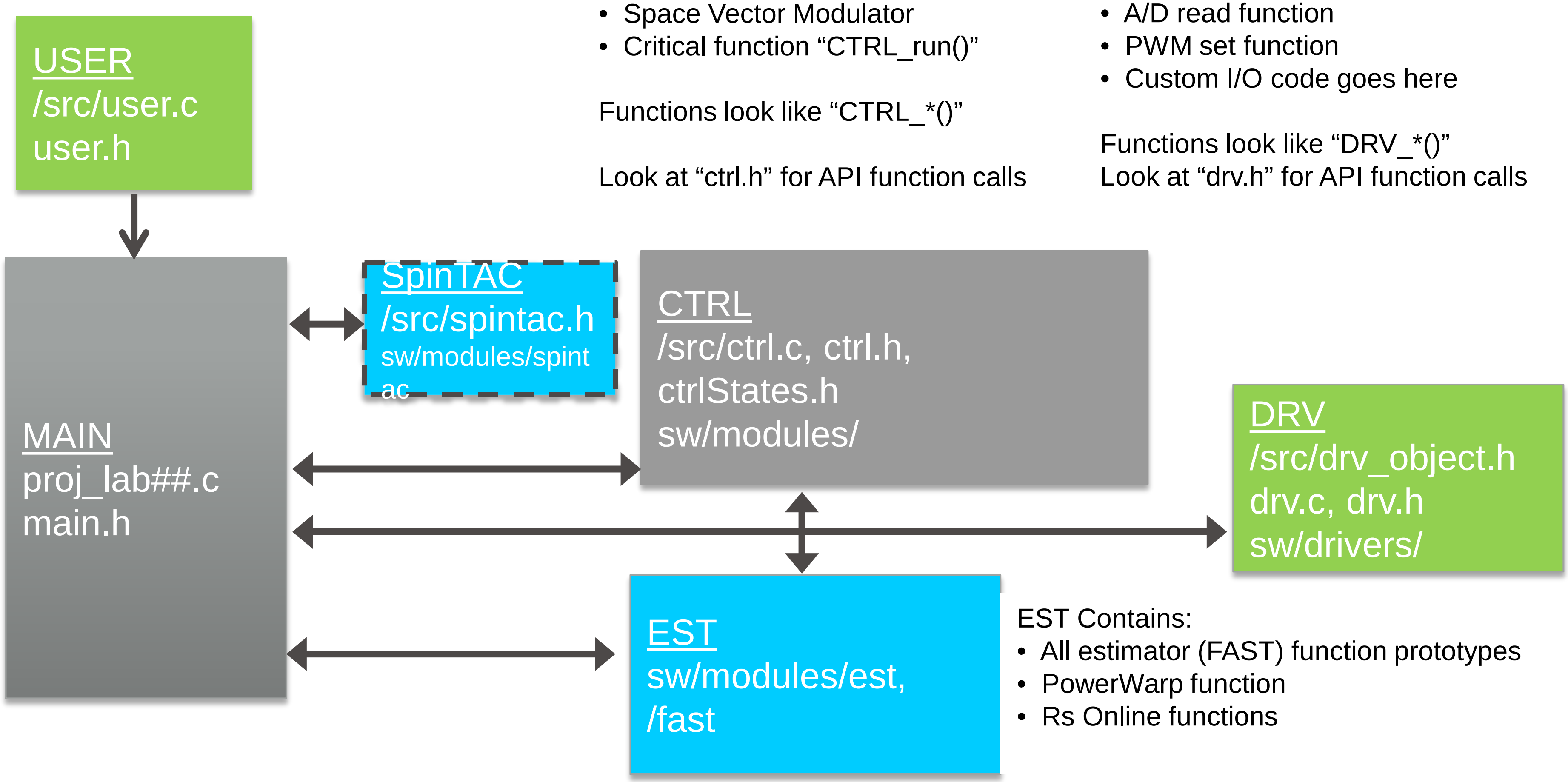
CTRL Contains:
- All PI controllers
- Clarke, Park, Inverse Park
- Space Vector Modulator
- Critical function “CTRL_run()”

Functions look like “CTRL_*()”

Look at “ctrl.h” for API function calls

DRV Contains:
- Inverter/MCU setup/control functions
- A/D read function
- PWM set function
- Custom I/O code goes here

Functions look like “DRV_*()”
Look at “drv.h” for API function calls



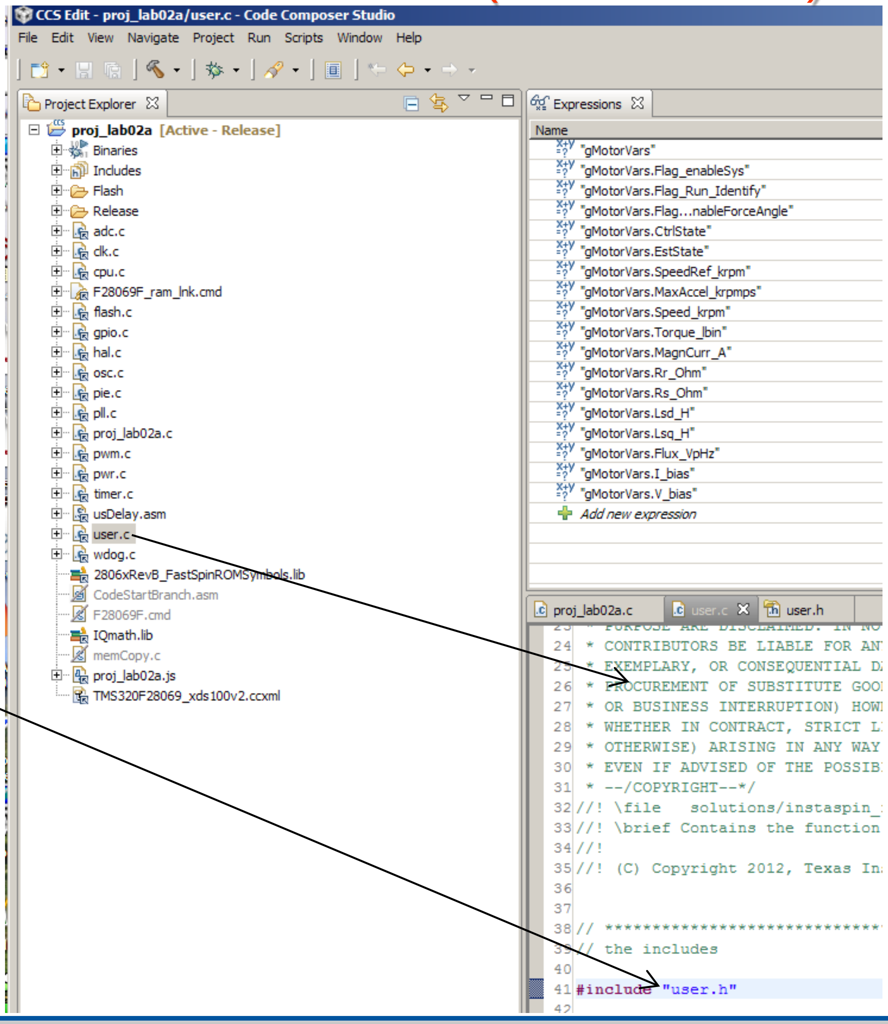
EST Contains:
- All estimator (FAST) function prototypes
- PowerWarp function
- Rs Online functions

Institute for Power Electronics and Electrical Drives
ISEA RWTH AACHEN

Motor Identification and sensorless operation : Lab 2 MotorWare (lab 2-2 VisSim)

Steps required:

1. Open TI motorware 12
2. Project> Import 'Existing CCS Eclipse project' : need to work down the software tree to find 'lab02a' for DRV8312 board
3. 'right-click' project name and 'rebuild project': result as shown
4. Select (left click) user.c
5. Within user.c select (left click with ctrl down) user.h



Next: configure user.h file according to dialog box data VisSim lab 2-2

Institute for Power Electronics and Electrical Drives

ISEA RWTH AACHEN

Configure user.h file on the basis of dialog box entries VisSim lab 2-2

User.h MotorWare

user.c | user.h

```
// *****************************************************
// the defines


//! \brief CURRENTS AND VOLTAGES
// *****************************************************
//! \brief Defines the full scale frequency for IQ va
//! \brief All frequencies are converted into (pu) ba
//! \brief this value MUST be larger than the maximum
//#define USER_IQ_FULL_SCALE_FREQ_Hz          (800.0)
#define USER_IQ_FULL_SCALE_FREQ_Hz          (400.0)   /
//! \brief Defines full scale value for the IQ30 vari
//! \brief All voltages are converted into (pu) based
//! \brief WARNING: this value MUST meet the followin
//! \brief WARNING: otherwise the value can saturate
//! \brief WARNING: this value is OFTEN greater than
//! \brief WARNING: if you know the value of your Ben
//! \brief It is recommended to start with a value ~3
//! \brief This value is also used to calculate the m
//#define USER_IQ_FULL_SCALE_VOLTAGE_V        (24.0)
#define USER_IQ_FULL_SCALE_VOLTAGE_V        (48.0)    //
//! \brief Defines the maximum voltage at the input t
//! \brief The value that will be represented by the
//! \brief Hardware dependent, this should be based o
#define USER_ADC_FULL_SCALE_VOLTAGE_V       (66.32)

//! \brief Defines the voltage scale factor for the s
//! \brief Compile time calculation for scale factor
#define USER_VOLTAGE_SF                 ((float_t)((USE

//! \brief Defines the full scale current for the IQ
//! \brief All currents are converted into (pu) based
//! \brief WARNING: this value MUST be larger than th
#define USER_IQ_FULL_SCALE_CURRENT_A            (10.0)
```

lab3-2_8312phCv8.c - Notepad

File Edit Format View Help

User.h compiled in phase C with VisSim

```
/*** VisSim Automatic C Code Generator Version 9.0PR
/*  Output for lab3-2_8312phCv8.vsm at Tue Jul 08 15

#include "math.h"
#include "cgen.h"
#include "cgendll.h"
#include "c2000.h"
#include "MotorWare.h"
#include "DMC32.h"
int MHZ=90;
int curCtlStateGb;
int stateChangeld;
int junk;
DEF_MONITORBUF(_sigBuffer0,901,int,15,3,16);
DEF_MONITORBUF(_sigBuffer1,901,int,15,3,16);
static DRV_PwmData_t gPwmData_0;
static DRV_Obj drvState_0;
static DRV_Handle drvHandle_0;
static DRV_AdcData_t gAdcData_0;
static CTRL_Obj IS_controller_obj0_var;
static CTRL_Obj *IS_controller_obj0=&IS_controller_o
static USER_Params gUserParams_0={10,   /* iqFullSca
48,     /* iqFullScaleVoltage_V */
400,    /* iqFullScaleFreq_Hz */
1,      /* numIsrTicksPerCtrlTick */
1,      /* numCtrlTicksPerCurrentTick */
2,      /* numCtrlTicksPerEstTick */
15,     /* numCtrlTicksPerSpeedTick */
15,     /* numCtrlTicksPerTrajTick */
3,      /* numCurrentSensors */
3,      /* numVoltageSensors */
6.2831853,      /* offsetPole_rps */
628.31853,      /* fluxPole_rps */
0.00125,        /* zeroSpeedLimit */
1,      /* forceAngleFreq_Hz */
4000,   /* maxAccel_Hzps */
20,     /* maxAccel_est_Hzps */
37.699112,      /* directionPole_rps */
314.15927,      /* speedPole_rps */
125.66371,      /* dcBusPole_rps */
1,      /* fluxFraction */
1,      /* indEst_speedMaxFraction */
1,      /* eplGain */
90,     /* systemFreq_MHz */
66.666667,      /* pwmPeriod_usec */
1.3816667,      /* voltage_sf */
1.7277, /* current_sf */
4487.074,       /* voltageFilterPole_rps */
0.92376043,     /* maxVsMag_pu */
1.5,    /* estKappa */
0,      /* motor_type */
```

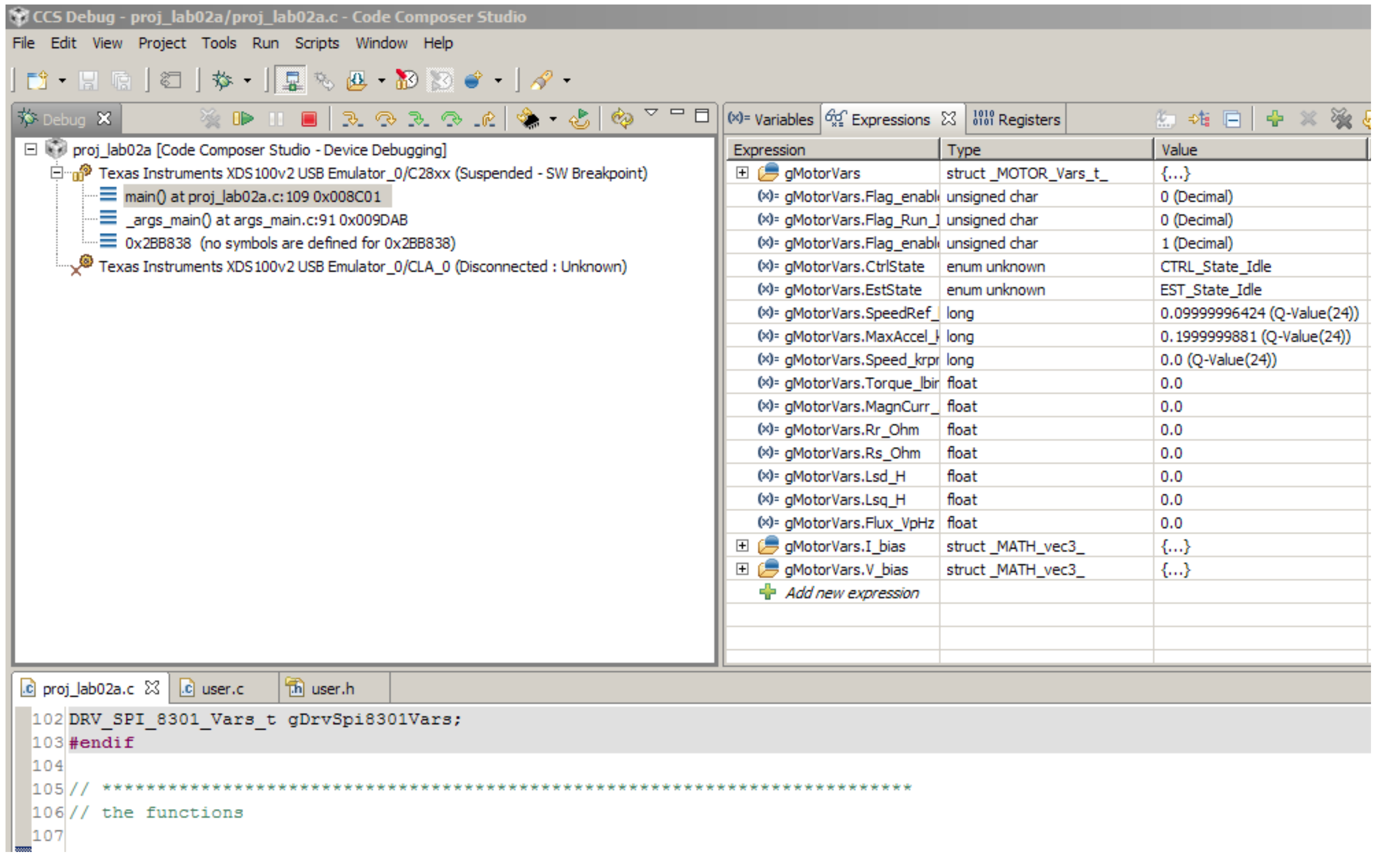
Institute for Power Electronics and Electrical Drives
RWTH AACHEN

Move to 'debug' mode of CCS to run lab 02a

Green arrow starts debug process:

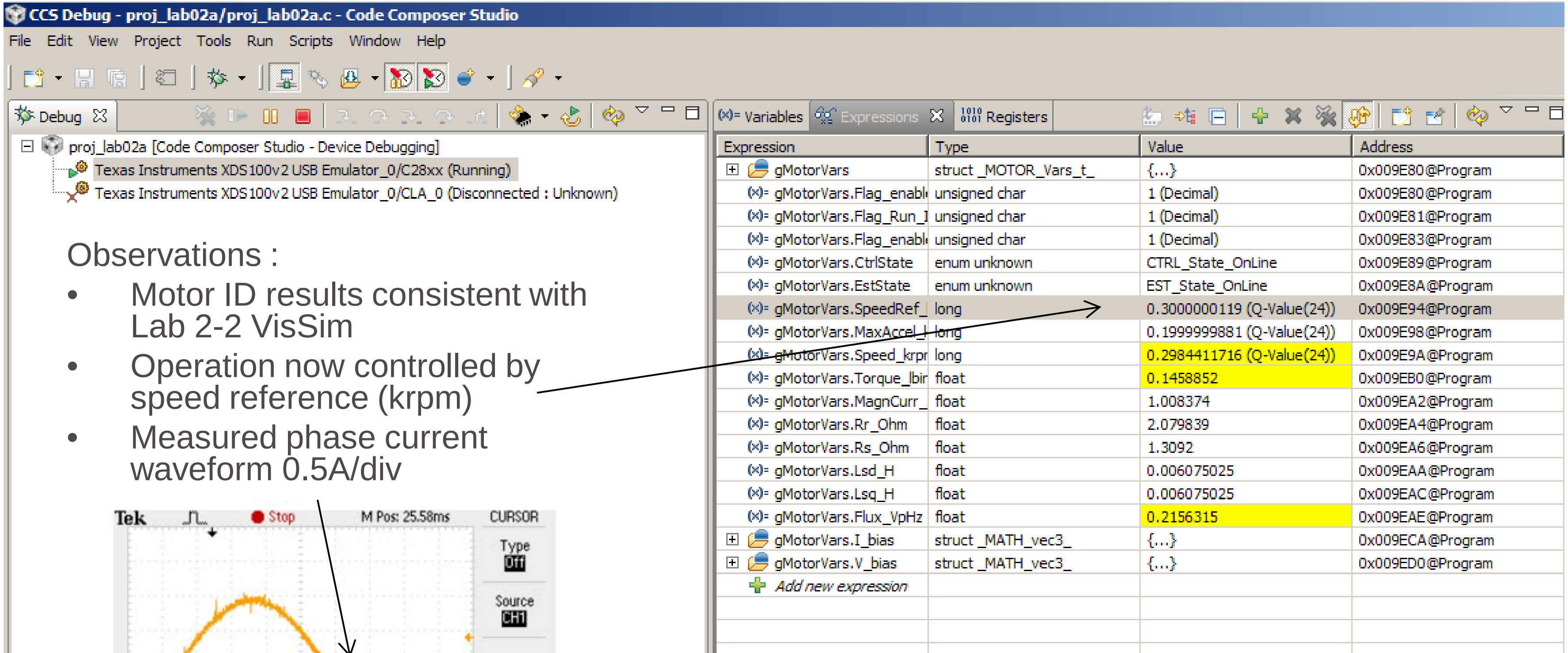
Flags/variables can be set in Watch window

Output variables shown in Watch (Expression) window during operation. (see next slide)



Institute for Power Electronics and Electrical Drives
RWTH AACHEN

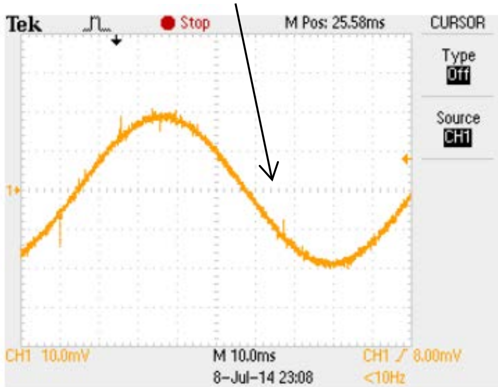
'Debug' mode during operation:



Expression	Type	Value	Address
gMotorVars	struct _MOTOR_Vars_t_	{...}	0x009E80@Program
gMotorVars.Flag_enabl	unsigned char	1 (Decimal)	0x009E80@Program
gMotorVars.Flag_Run_I	unsigned char	1 (Decimal)	0x009E81@Program
gMotorVars.Flag_enabl	unsigned char	1 (Decimal)	0x009E83@Program
gMotorVars.CtrlState	enum unknown	CTRL_State_OnLine	0x009E89@Program
gMotorVars.EstState	enum unknown	EST_State_OnLine	0x009E8A@Program
gMotorVars.SpeedRef_	long	0.3000000119 (Q-Value(24))	0x009E94@Program
gMotorVars.MaxAccel_	long	0.1999999881 (Q-Value(24))	0x009E98@Program
gMotorVars.Speed_krpr	long	0.2984411716 (Q-Value(24))	0x009E9A@Program
gMotorVars.Torque_lbir	float	0.1458852	0x009EB0@Program
gMotorVars.MagnCurr_	float	1.008374	0x009EA2@Program
gMotorVars.Rr_Ohm	float	2.079839	0x009EA4@Program
gMotorVars.Rs_Ohm	float	1.3092	0x009EA6@Program
gMotorVars.Lsd_H	float	0.006075025	0x009EAA@Program
gMotorVars.Lsq_H	float	0.006075025	0x009EAC@Program
gMotorVars.Flux_VpHz	float	0.2156315	0x009EAE@Program
gMotorVars.I_bias	struct _MATH_vec3_	{...}	0x009ECA@Program
gMotorVars.V_bias	struct _MATH_vec3_	{...}	0x009ED0@Program
Add new expression			

Observations :

- Motor ID results consistent with Lab 2-2 VisSim
- Operation now controlled by speed reference (krpm)
- Measured phase current waveform 0.5A/div



TPS 2014B - 17:12:47 7/8/2014

Institute for Power Electronics and Electrical Drives
RWTH AACHEN

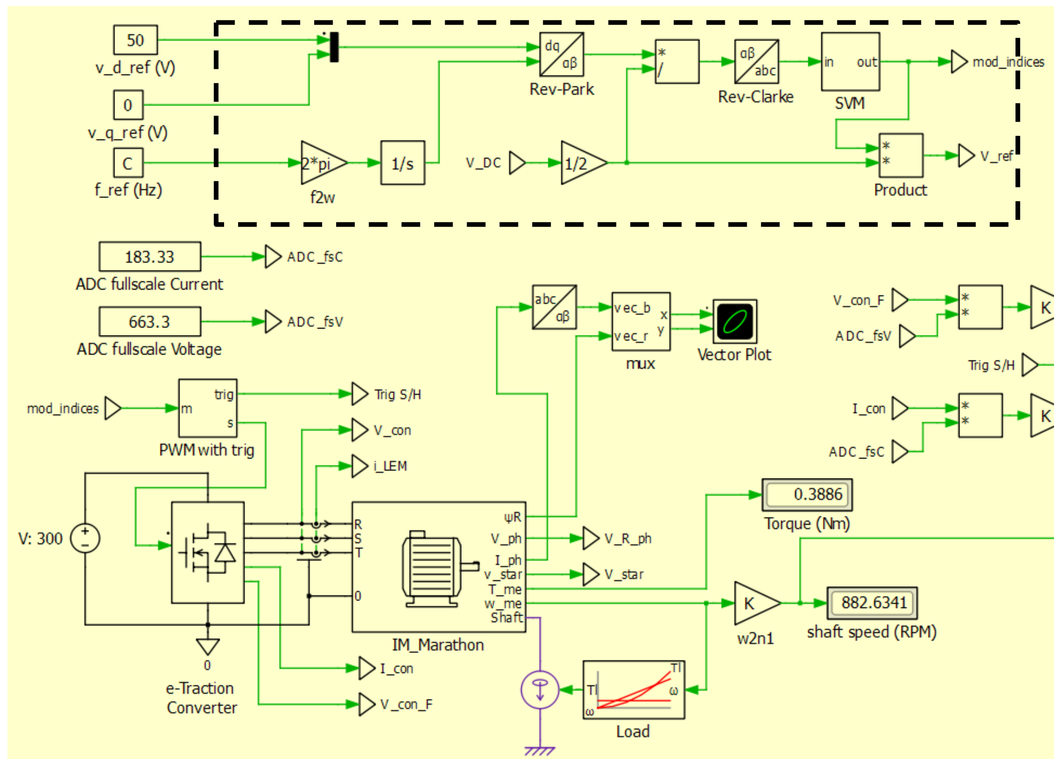
PLECS simulation software:

What is PLECS [6]? :

- Comprehensive software development platform which is available as 'stand-alone' or with MATLAB/Simulink
- Ability to use Electrical and control blocks together
- Example PLECS: Case3av2 V/f controller with e-Traction converter and IM motor

Example:

- V/f controller
- SVM modulator
- PWM
- ADC with Sampling
- Converter with ideal or non ideal power electronic devices
- Thermal modelling converter (heat sink)
- Motor model
- Mechanical model interfaces to load
- Comprehensive analysis tools
- Fast computation times

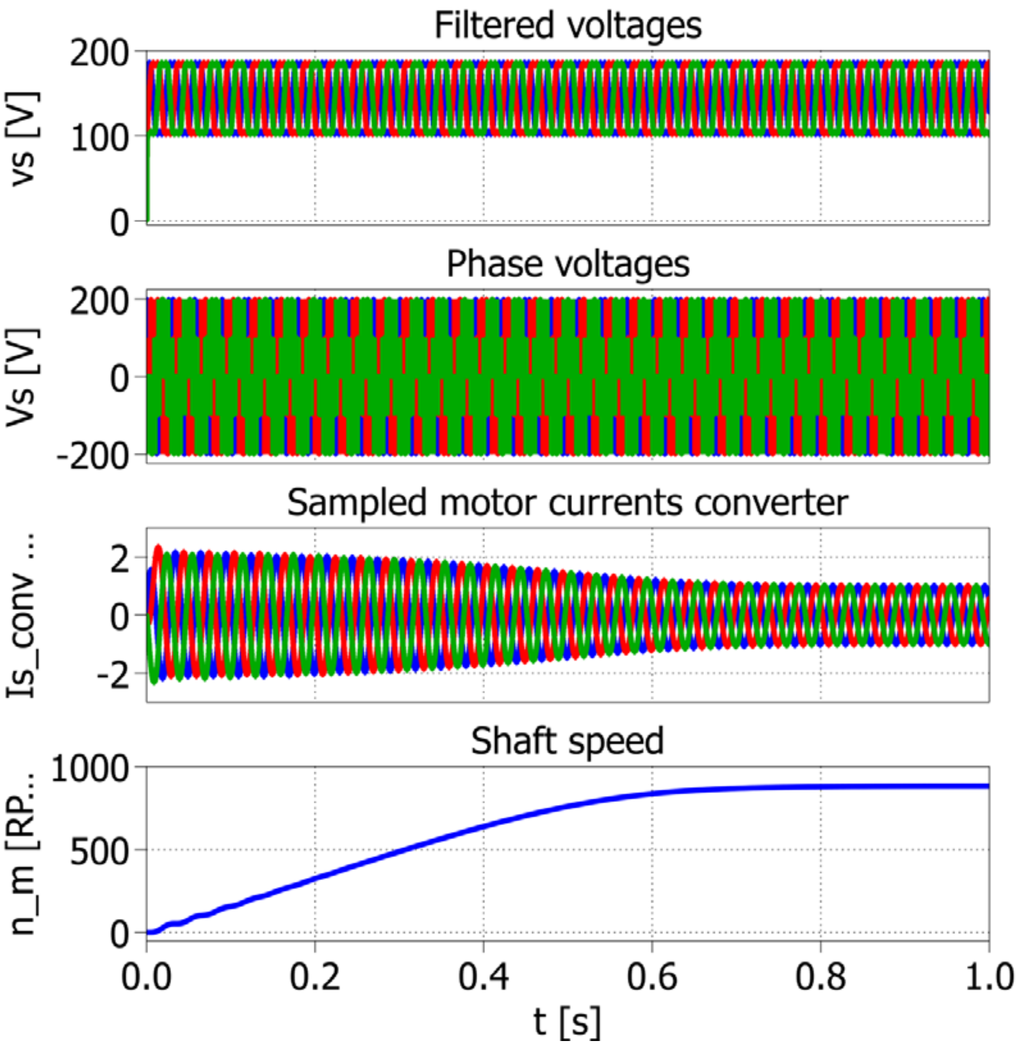


PLECS model

Institute for Power Electronics and Electrical Drives
RWTH AACHEN

PLECS simulation software:

- V/f controller
- Typical results :



User has the ability to fully tailor drive design to meet performance requirement

PLECS software

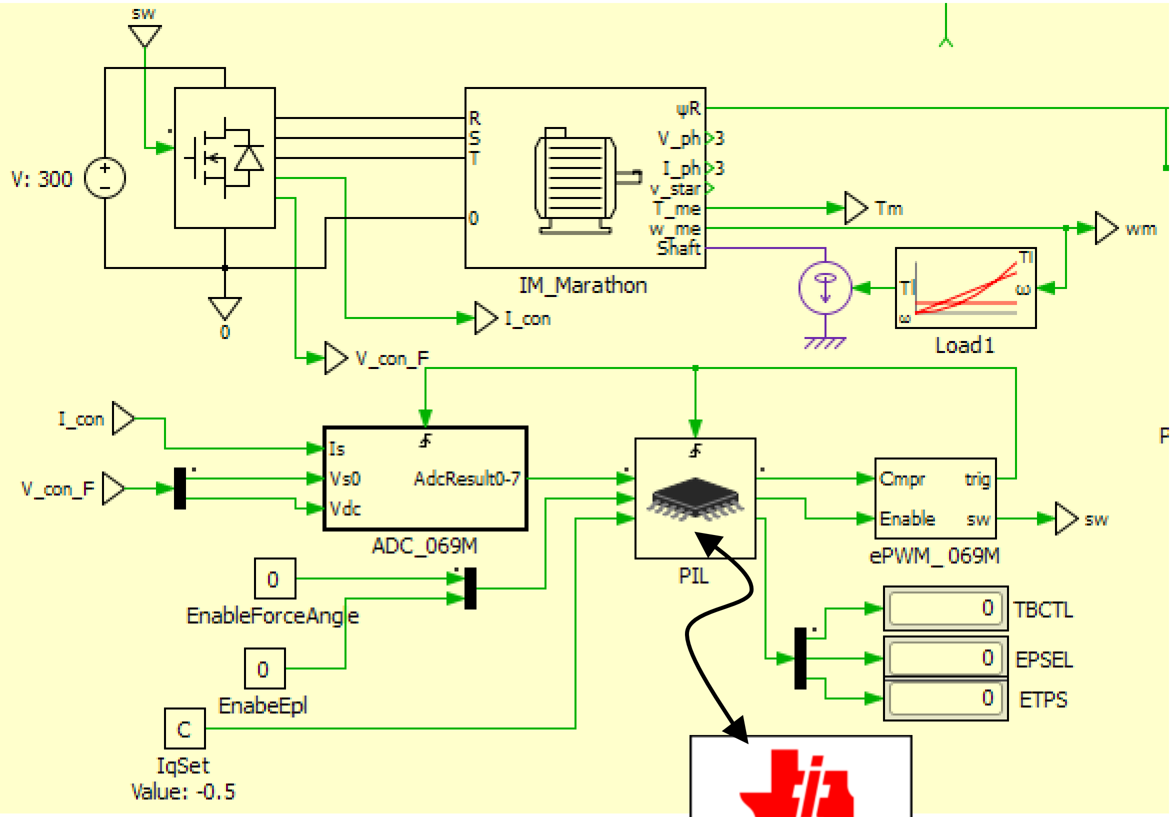
- Example PLECS case study
- e-Traction 3 kW/400V converter and Marathon IM motor
- SETUP in use with:
  - **TMS32028069M control card with USB JTAG interface located on e-Traction board**
  - **400V/2 A DC power power supply**
  - **24V auxiliary power supply for logic circuit**
  - **CAN BUS interface (not used here)**
- We want to evaluate sensorless operation WITHOUT running the hardware
- Answer: PLECS processor in the loop technology (PIL)

Institute for Power Electronics and Electrical Drives
RWTH AACHEN

PIL: 'Processor in the loop'

What is PIL ?

- TMS320F8069M processor in the 'loop'
- ADC inputs from hardware 'overridden' by inputs from 'ADC-069M' PLECS module
- PWM outputs 'sw' by MCU now generated by 'ePWM_069M' PLECS module, which controls PLECS converter
- MCU 'trig' output used to trigger internal ADC now triggers PLECS 'ADC-069M'
- Additional inputs/outputs to PIL can be generated to suit user requirements.
- MCU runs the TI MotorWare labs, provided they are 'PIL prepared'
- In this example MotorWare laboratory 4e is used to control the drive: FAST as encoder, with known parameters and control of 'IqSet' (reference Iq)



Next: how to set up PIL operation

Institute for Power Electronics and Electrical Drives
RWTH AACHEN

How to set up PIL operation

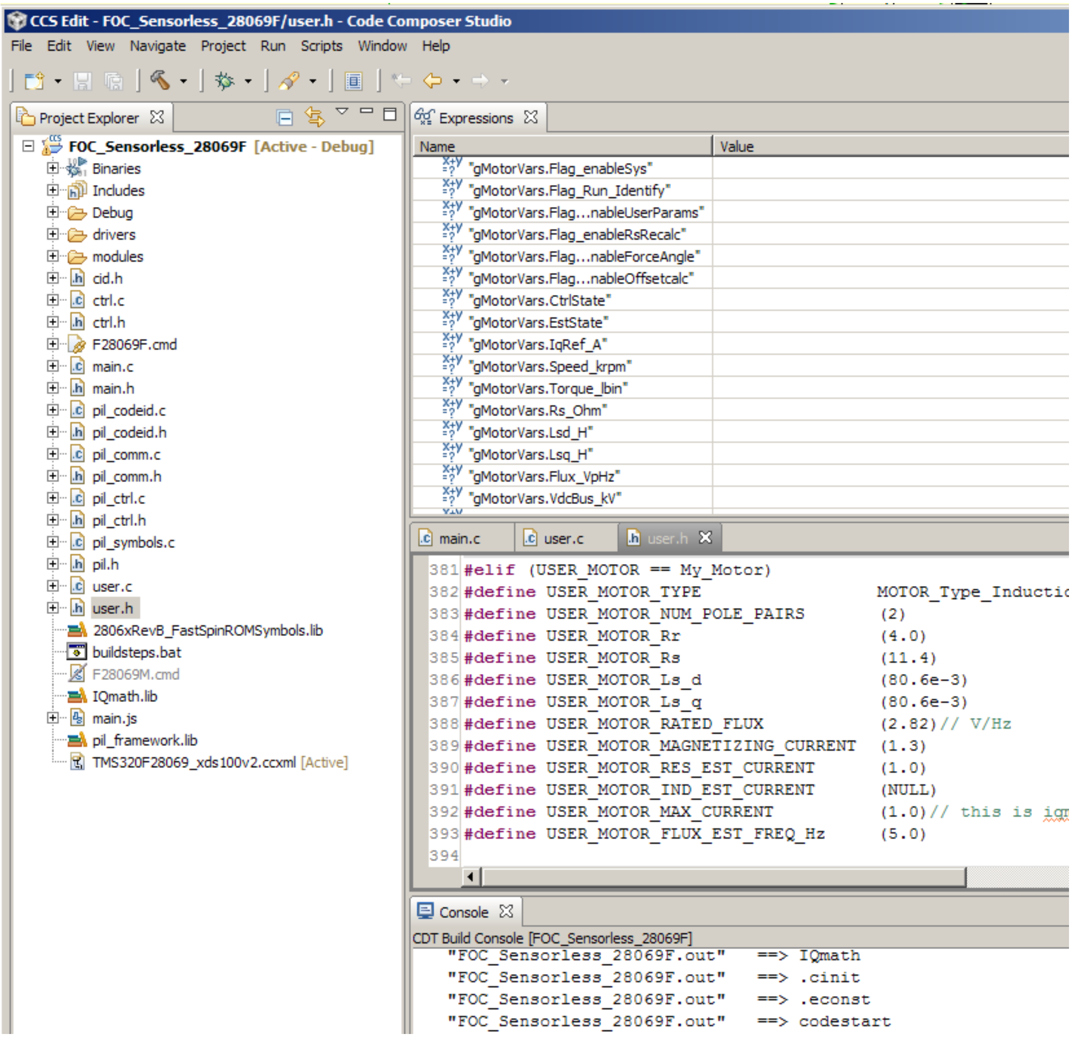
Actions to take:

- Import the PLECS PIL prepared MotorWare lab 4e
- Edit the user.h file to match the motor and drive data used in PLECS model.
- Edit drv.c if the MCU clock freq. need to change as is the case here: set to 60 MHz
- Rebuild the project, which generates a xxx. out file
- Keep the MotorWare and PLECS file in the same folder
- Move to 'Debug' mode and which downloads

  xxx.out file

- Start debug : LED flashing on control board

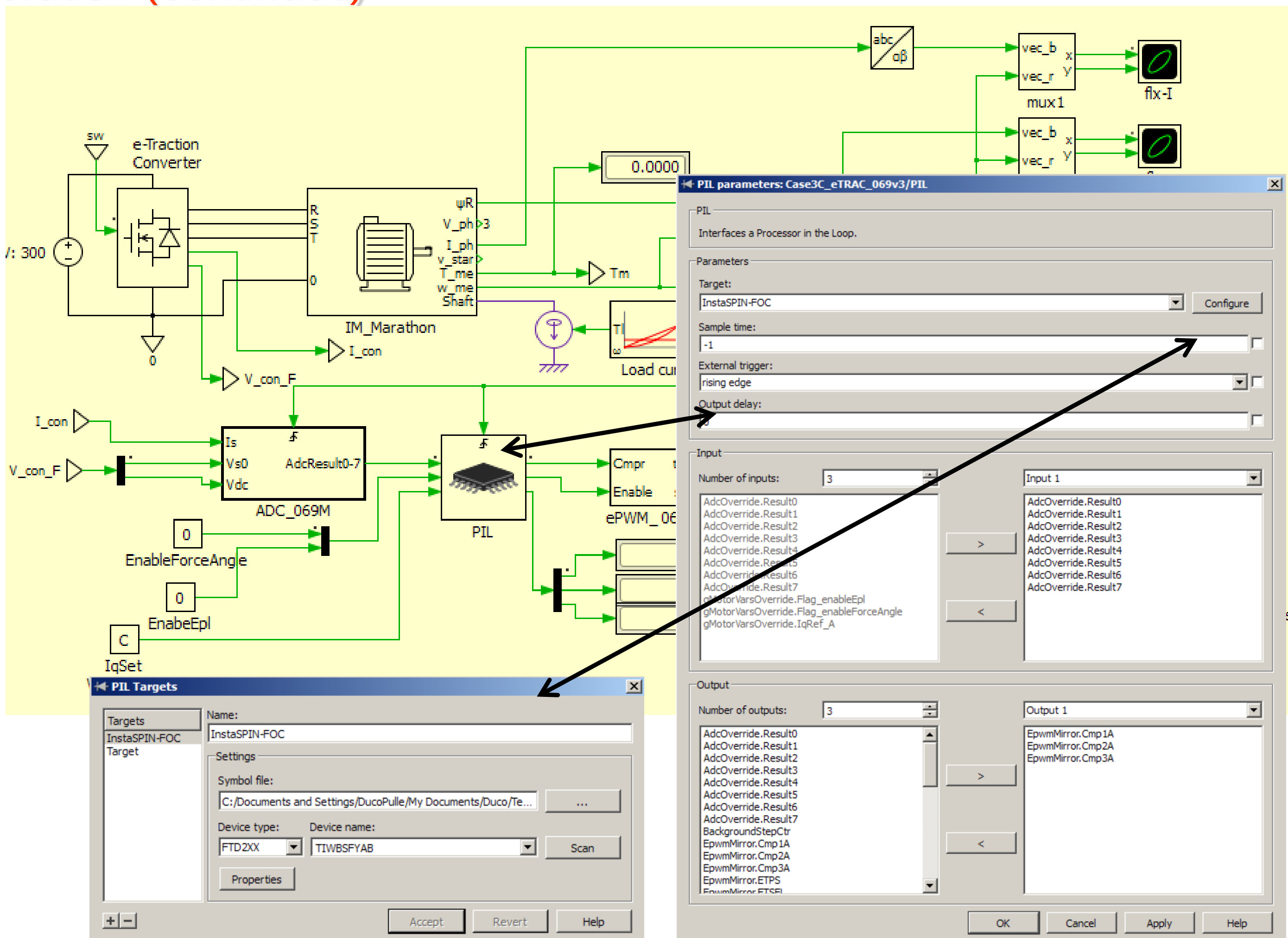
Next: move to the PLECS model



How to set up PIL operation (continued)

Actions to take:

- Double left click on PLECS PIL
- Set target to InstaSPIN-FOC
- Select 'configure'
- Enter the location of for the xxx. out in the 'symbol' entry
- Entries for the ePWM-069M must be made, but requires one time PIL operation so that info is on displays.
- In PLECS Simulation parameters> Initialization box: set System CLK and PWM frequency

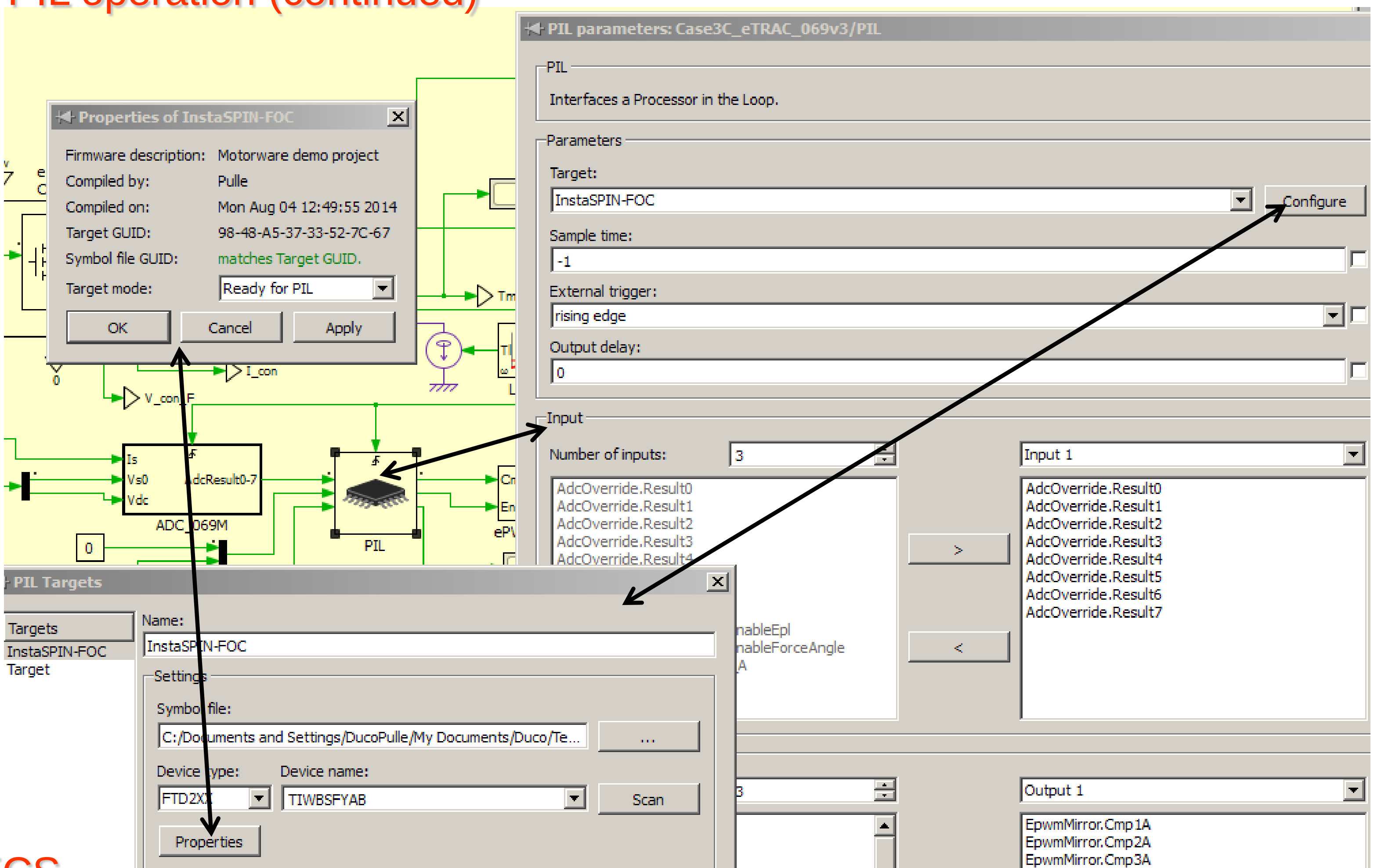


Institute for Power Electronics and Electrical Drives
RWTH AACHEN

How to set up PIL operation (continued)

Actions to take:

- Click on PLL
- Select 'Configure'
- Select 'Properties'
- Confirm that symbol file GUID states 'matches Target GUID'
- Check that 'Target mode' set to 'ready for PIL'



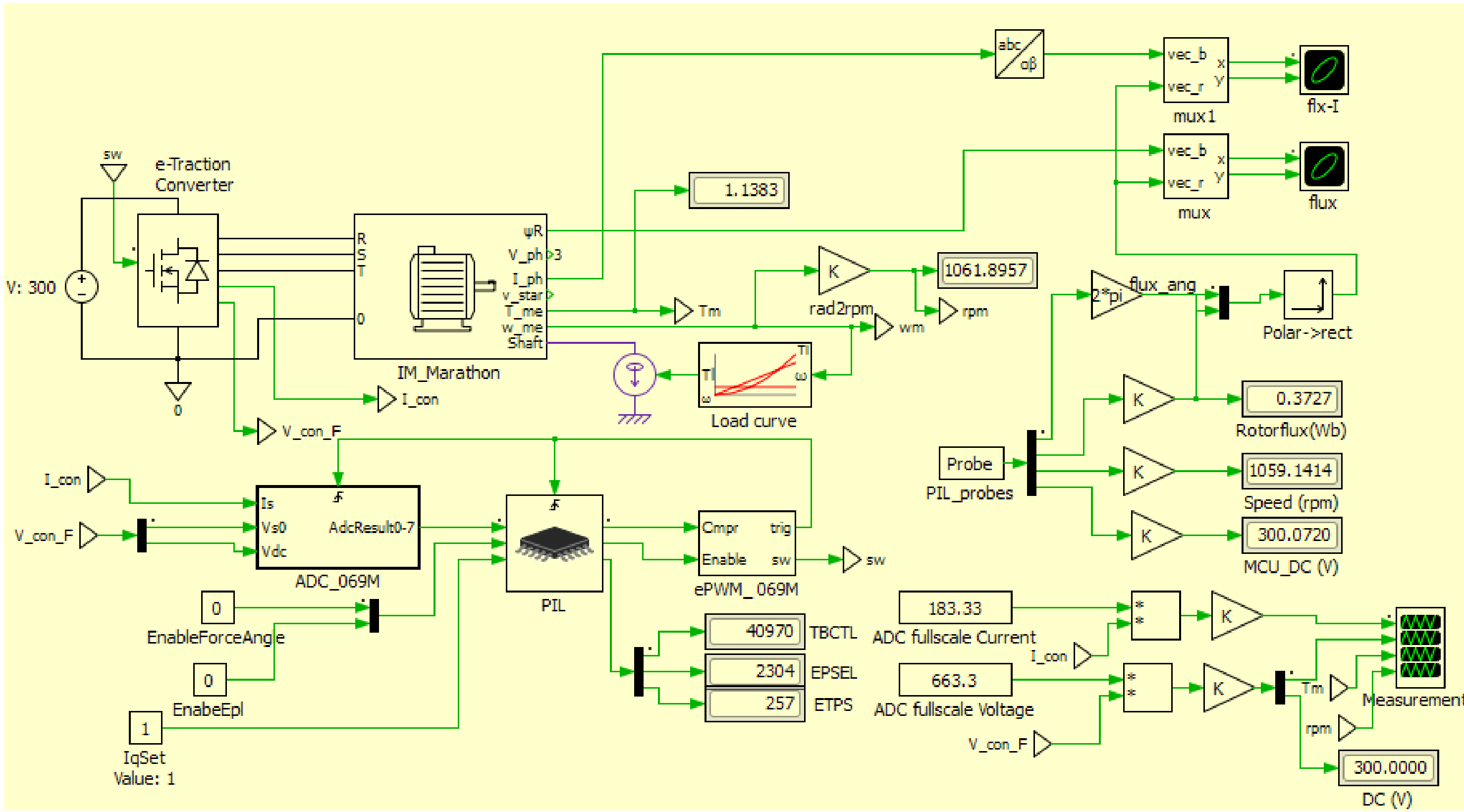
Next: RUN PLECS

Institute for
Power Electronics and
Electrical Drives
RWTHAACHEN

Running PIL

Actions to take:

- Set 'enable force angle 1'
- set iqSet to 1 A
- We have quadratic load curve 1 Nm at 1000 rpm
- Start PLECS
- Observe:
  - MCU DC is 300V
  - check TBCL, EPSEL and ETPS match results in ePWM mod
- Selected magnetizing current Id=1.3A corresponds to rotor flux of 0.37Wb

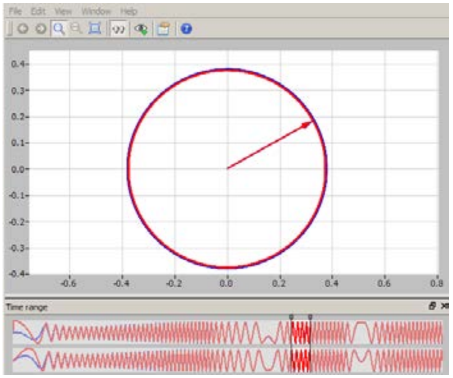


Next: Results achieved with PIL

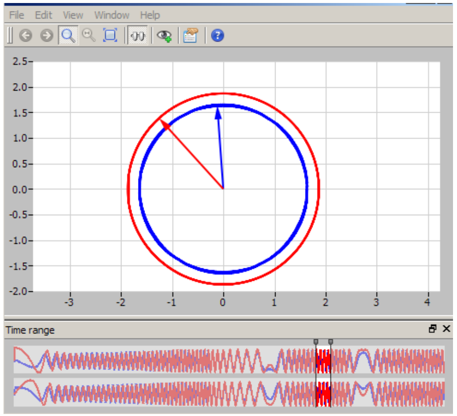
Results when Running PIL with PLECS

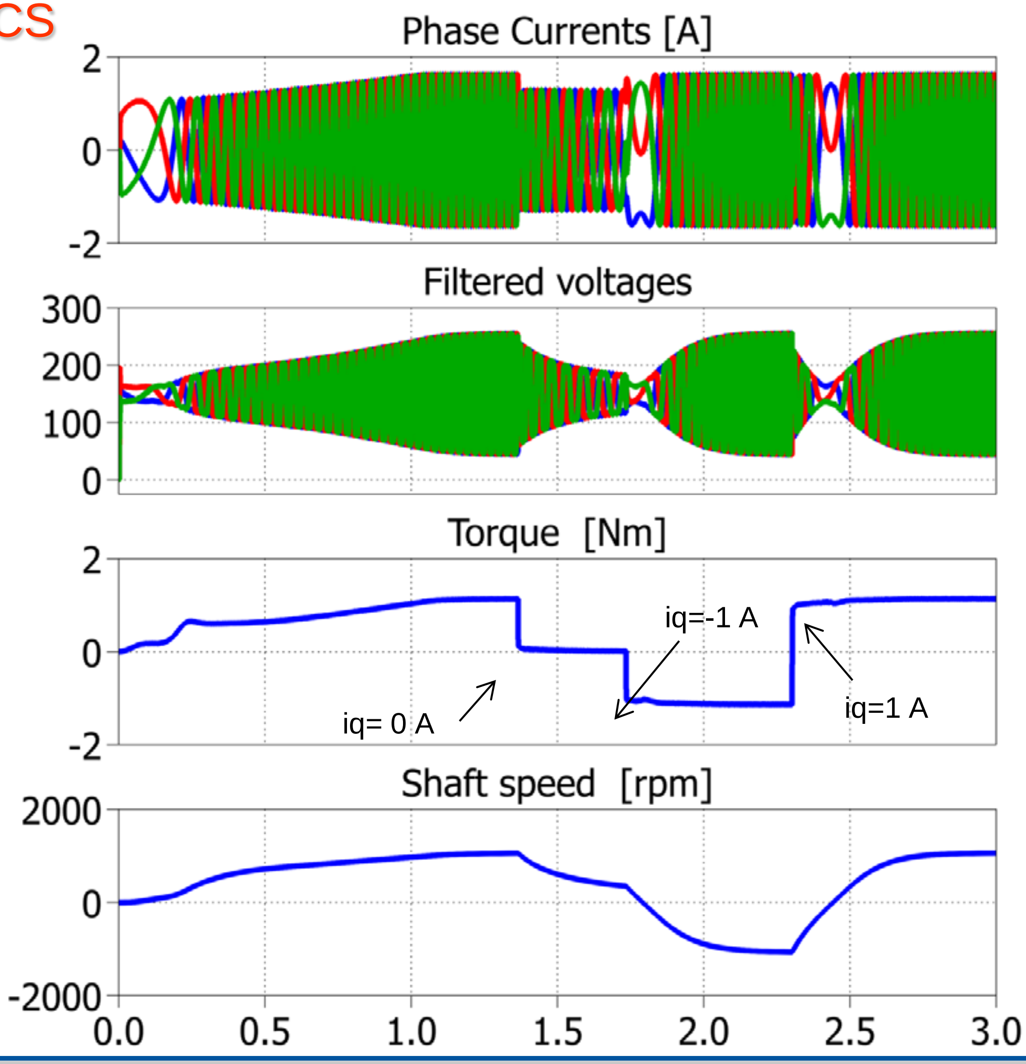
Observations:

- Machine fully fluxed at t=1.2s (force angle turned off)
- Actual and FAST (red) flux vectors aligned



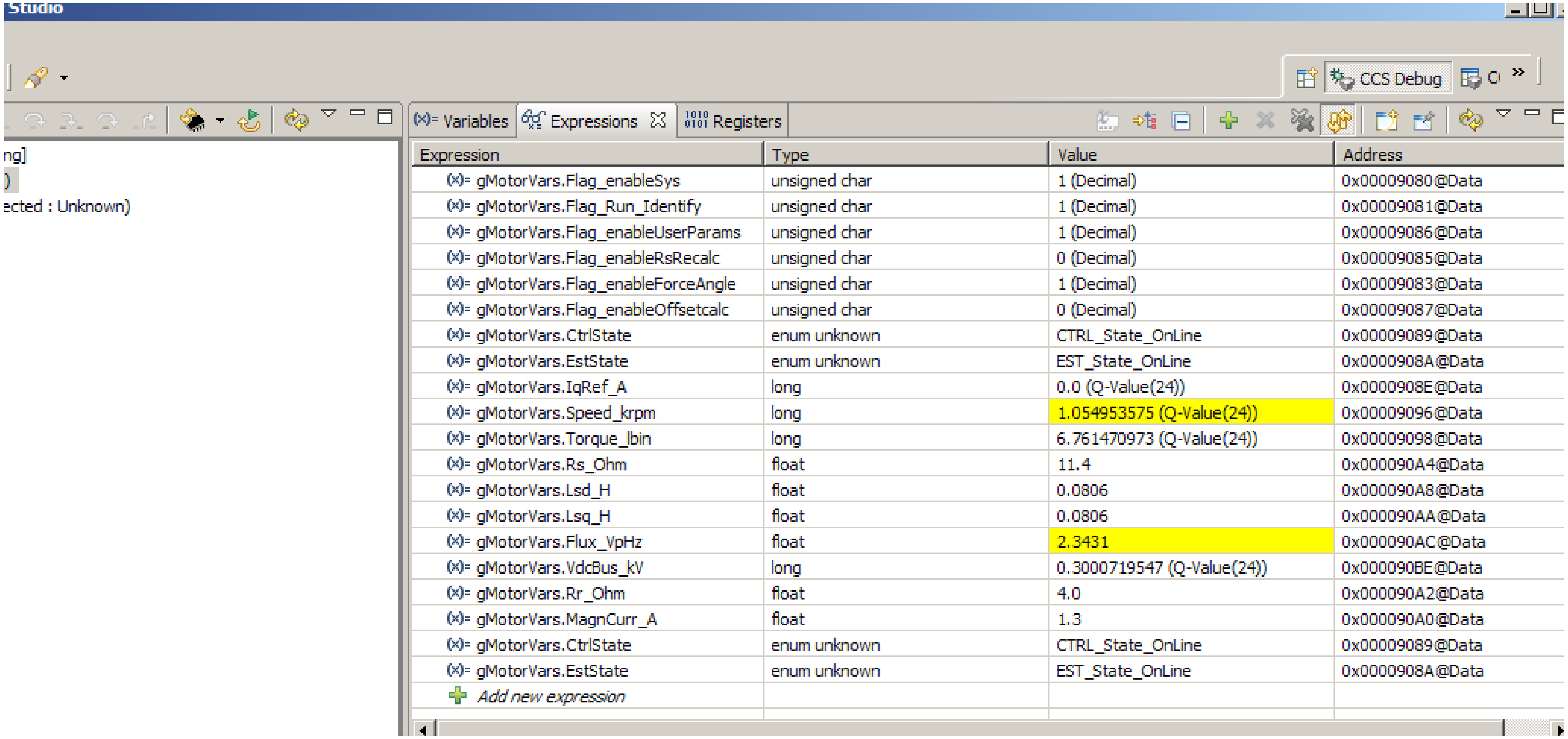
- FOC torque control example :
  - flux times 5 (red) and current vector





Institute for Power Electronics and Electrical Drives
RWTH AACHEN

Result when Running PIL with CCS



Expression	Type	Value	Address
gMotorVars.Flag_enableSys	unsigned char	1 (Decimal)	0x00009080@Data
gMotorVars.Flag_Run_Identify	unsigned char	1 (Decimal)	0x00009081@Data
gMotorVars.Flag_enableUserParams	unsigned char	1 (Decimal)	0x00009086@Data
gMotorVars.Flag_enableRsRecalc	unsigned char	0 (Decimal)	0x00009085@Data
gMotorVars.Flag_enableForceAngle	unsigned char	1 (Decimal)	0x00009083@Data
gMotorVars.Flag_enableOffsetcalc	unsigned char	0 (Decimal)	0x00009087@Data
gMotorVars.CtrlState	enum unknown	CTRL_State_OnLine	0x00009089@Data
gMotorVars.EstState	enum unknown	EST_State_OnLine	0x0000908A@Data
gMotorVars.IqRef_A	long	0.0 (Q-Value(24))	0x0000908E@Data
gMotorVars.Speed_krpm	long	1.054953575 (Q-Value(24))	0x00009096@Data
gMotorVars.Torque_lbin	long	6.761470973 (Q-Value(24))	0x00009098@Data
gMotorVars.Rs_Ohm	float	11.4	0x000090A4@Data
gMotorVars.Lsd_H	float	0.0806	0x000090A8@Data
gMotorVars.Lsq_H	float	0.0806	0x000090AA@Data
gMotorVars.Flux_VpHz	float	2.3431	0x000090AC@Data
gMotorVars.VdcBus_kV	long	0.3000719547 (Q-Value(24))	0x000090BE@Data
gMotorVars.Rr_Ohm	float	4.0	0x000090A2@Data
gMotorVars.MagnCurr_A	float	1.3	0x000090A0@Data
gMotorVars.CtrlState	enum unknown	CTRL_State_OnLine	0x00009089@Data
gMotorVars.EstState	enum unknown	EST_State_OnLine	0x0000908A@Data
Add new expression			

Observations from Watch Window :

- Shows the machine parameters in use and magnetizing current selected
- Shows rotor flux (V/Hz) of the estimator, rotor speed (krpm) and supply voltage

Control of drive may also be realized directly from CCS watch window by setting PIL to 'normal operation'

Applications of sensorless control

- Why customers change to Sensorless control using InstaSPIN?
  - To reduce product cost : removal of the encoder leads to significant savings
  - To reduce product development time
  - To increase reliability: to avoid encoder alignment and breakdown problems
  - To enhance their product: having access to instantaneous shaft torque and speed   without requiring additional (to the power leads) is attractive
  - Ability to   measure and track key motor parameters: measurement of, for example, resistance gives the ability to monitor temperature of the motor
  - Need for sophisticated motion control: use of SpinTAC [7] control suite
  - To provide back up to encoder based drives: provides the ability to keep the drive in operation if an encoder related problem occurs
  - To improve energy efficiency: for drive which use an induction machine, which allows field weakening during partial load operation

➔ Application examples

Applications of sensorless control

- Application examples: currently in use or being developed

PUMPS

Automotive	Industrial/Consumer
•Transmission •Brake/Boost •Oil •Turbo •Fuel/Water	•Constant pressure •Water/Waste/Chemical •Spa/pool pump •Geothermal pump •Dishwashers

COMPRESSORS

Automotive	Industrial/Consumer
•Refrigeration	•Air/Con •Refrigeration

LAUNDRY

•Washers
•Dryers

BLOWERS/FANS

Automotive	Industrial/Consumer
•Air/Con Blowers •Cooling Fan	•Respiratory •Vacuum •Fans •Air/Con Blowers •Exhaust

HIGH TORQUE

Transit	Conveyors
•Traction •eBike/Moped/Scooter •Off-highway Vehicles •Carts, Transport •Fork lifts •Wheel chairs	•Escalators •Elevators •Treadmill •Tools •AC Drive / Inverter •Assembly Line

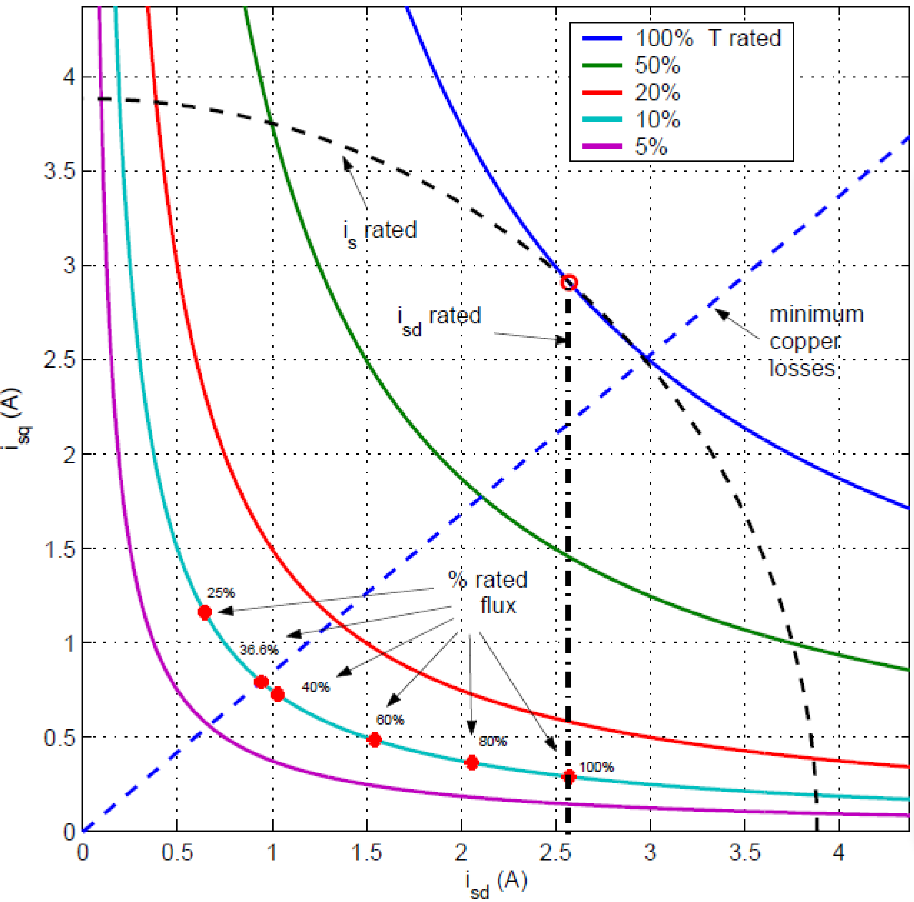
Applications of sensorless control

- Specific application examples: Traction drive for Electric Vehicle

  - Use of InstaSPIN-FOC to realise a highly responsive & efficient torque machine like this high performance 'Tesla'

Applications of sensorless control

- Specific application examples: Induction machine drives using PowerWarp



Adaptively reduce magnetizing current to only induce the field required for the torque required!

Real World Field Trial

Induction Motors used for Agriculture Air & Humidity Control

PowerWarp Savings

80% of energy vs. Traditional Triac

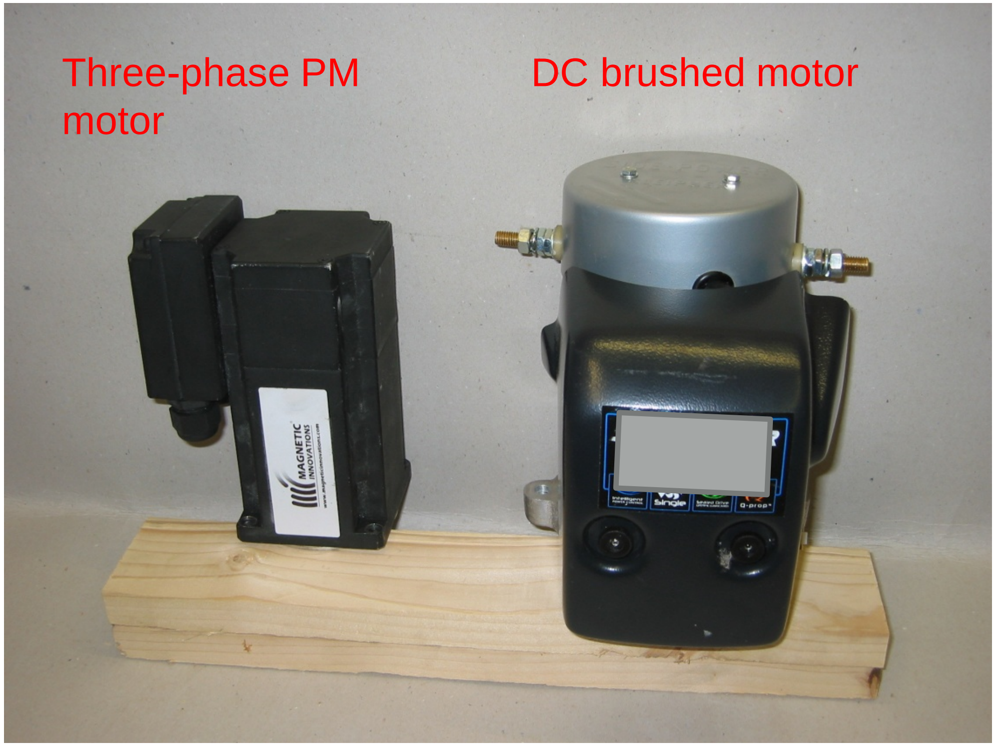
45% of energy vs. IS-FOC

Algorithm is based on reducing motor copper losses in the stator AND the rotor!

www.ti.com/powerwarp

Applications of sensorless control

- Specific application examples: Bow thruster for yacht



Max power: 3500W
Speed: 6000 rpm
Max current: 15 A
Efficiency : 90%
Weight: 3.5 kg

Max power: 2200W
Speed: 4100 rpm
Max current: 280 A
Efficiency : 75%
Weight: 8.9 kg

Summary/comments

- Introduction on sensorless electrical drive technology:
- Challenges faced when trying to implement a sensorless drive
- Introduction on the InstaSPIN Sensorless solution
- Introduction of development tools that can be used to speed up and simplify the development of MCU based control in general and InstaSPIN in particular
- Overview of InstaSPIN based applications currently in placed and those being developed world wide
- Book 'Embedded control of Electrical Drives using VisSim and PLECS' to appear later this year (publisher Springer)
- Demo table at conference which shows InstaSPIN in action!

Thank you for your attention !

References

1. Fundamentals of Electrical Drives, Veltman, A. , Pulle, D.W,J. and De Doncker R. , Springer 2007
2. Advanced Electrical Drives,, De Doncker R. , Pulle, D.W,J. and Veltman, A. , Springer 2010
3. Texas Instruments InstaSPIN: www.ti.com/instaspin
4. VisSim: www.vissim.com
5. C2000 based 2-Day Hands-On Motor Control Workshops
6. PLEXIM: www..plexim.com
7. SpinTAC motion control suite: part of InstaSPIN-Motion control

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265