Simplifying Dual Motor Control in Energy-Efficient Appliances

When permanent magnet brushless motor drives were introduced to the market more than twenty years ago, the control algorithms were implemented using a combination of analog amplifiers and logic components. Today, highly integrated mixed signal controllers enable the implementation of complex control algorithms that maximise the efficiency of permanent magnet AC drives in various applications.

Aengus Murray, International Rectifier, El Segundo, USA

Advances in power and digital control silicon technology over the past few decades have enabled a continuous improvement in motor drive technology. Japanese air conditioning manufacturers recently started switching from sensorless control based on back EMF (electro-magnetic force) sensing to sensorless field oriented control based on current feedback. Field oriented control (FOC) with current angle phase advance maximises the efficiency of interior permanent magnet motors, providing an efficiency gain of almost 5%. The switch from trapezoidal to sinusoidal control also minimises torque ripple to reduce acoustic noise in the fan motor.

A recently introduced mixed signal control IC can simultaneously run both the compressor and fan motors in an air conditioning system. A combination of highly optimised hardware control blocks and a configurable control sequencer enables rapid execution of complex motor control algorithms. The IC is one element in an appliance design platform that includes all the power and control silicon needed to drive the fan and compressor motor in an air conditioning system. The same IC can also be applied to the latest energy efficient laundry systems that use an energy saving heat pump in the drying cycle.

Air conditioning design platform

The circuit schematic in Figure 1 includes the major components in the outdoor unit controller for an air conditioning system. The mixed signal motor control IC drives the compressor motor, the fan motor and the power factor correction circuit. The IC integrates three major functions: the Motion Control Engine (MCE), the Analog Signal Engine (ASE) and an 8bit microcontroller. The MCE executes the motor control algorithms, while an independent 8bit microcontroller core implements the application functions.

The sensorless field oriented control algorithm derives all the required motor information from the currents flowing in the DC link shunts. This avoids the need for position sensors on the motor shaft and isolated current transducers in the power inverter circuit. The Analog Signal Engine includes the fast A/D converter, multiple sampling circuits and differential amplifiers needed to extract the motor winding current from the DC link current signal.

The Motion Control Engine executes the sensorless FOC algorithm described by the control schematic in Figure 2. The algorithm includes a reverse rotation function that transforms the measured stator currents into a reference frame synchronised with the angle of the rotor magnet flux. The transformed currents have two quasi DC components: a direct axis current is aligned with the rotor flux and a quadrature axis component that generates motor torque due interaction with the rotor magnet.

The d and q axis current loop compensators calculate the stator voltages to force the currents to track the set point values. The forward rotation function transforms these voltages to sinusoidal AC voltages in the stator reference frame. The space vector PWM generator uses these signals to derive transistor switching signals for the three-phase inverter. When driving a classical permanent magnet synchronous motor with surface mounted rotor magnets,

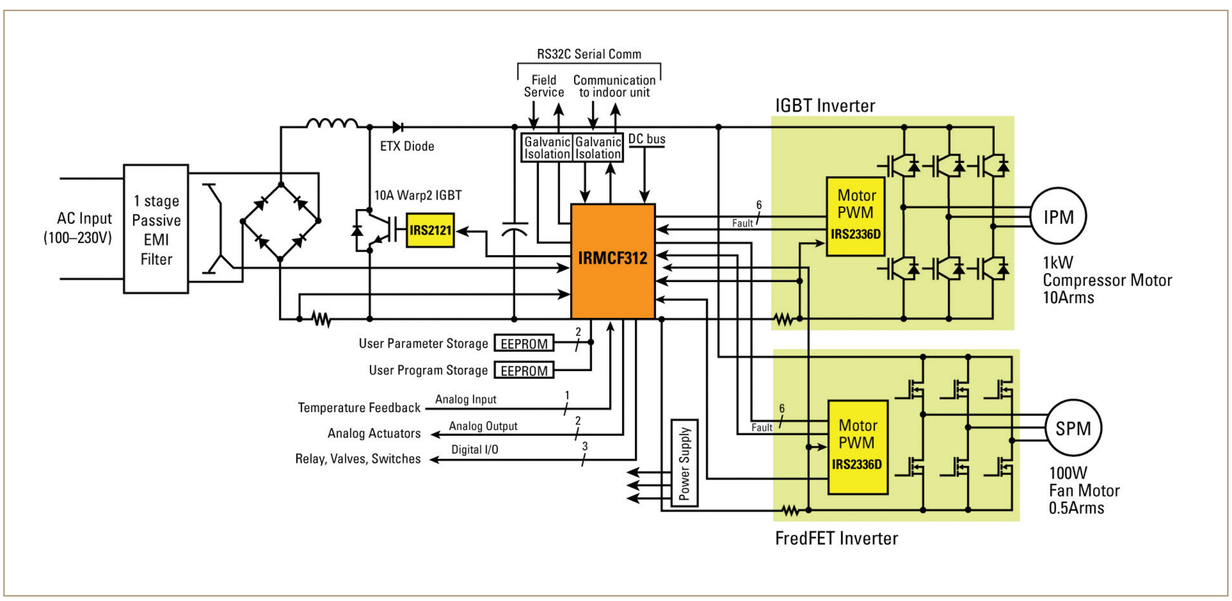


Figure 1: Air conditioning design platform

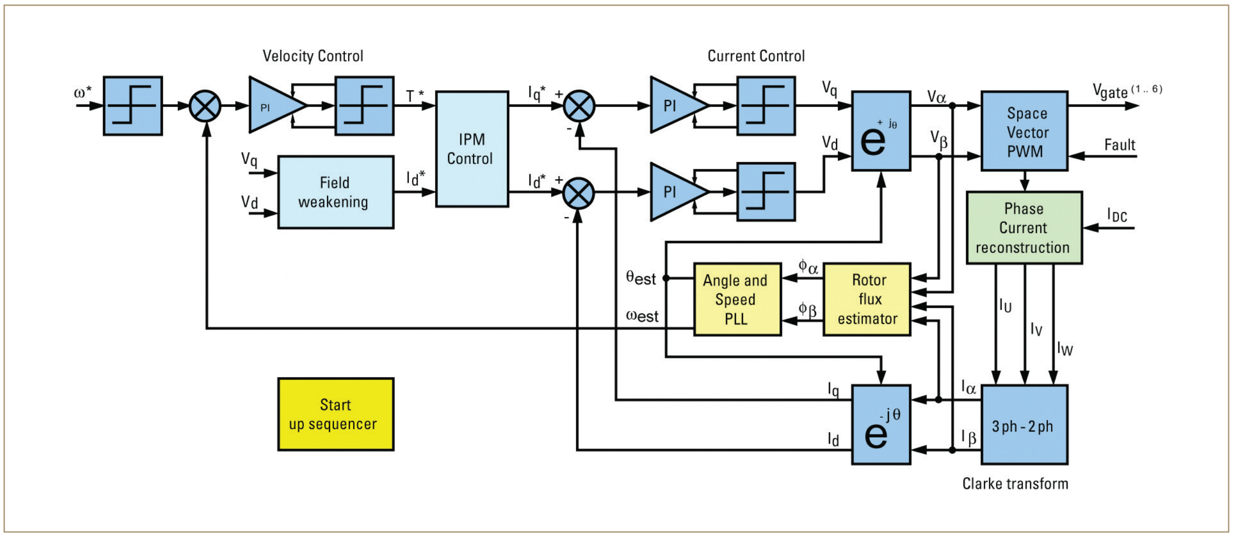


Figure 2: **Sensorless field oriented control algorithm**

the d axis reference current is set to zero to maximise the torque per amp. However, when driving an interior permanent magnet motor, the d axis current will generate a reluctance torque component to augment the torque produced by the rotor magnets. The IPM control function is the key control element that enables the operation at a higher efficiency when driving the IPM motor.

The other key feature of the sensorless FOC algorithm is that it does not require high resolution rotor angle sensors typically found in industrial drive systems. In appliance drives, where low speed performance is not important, the rotor angle can be derived from the winding back EMF signal. In brushless DC drives with a six-step commutation sequence, the back EMF is available directly by sampling the voltage on the unconnected winding. However, when driving the motor with sinusoidal currents the back EMF has to be calculated indirectly from the motor circuit model. The equations 1

$$v_\alpha = R_S.i_\alpha + L_S.\tfrac{di_\alpha}{dt} + \tfrac{d}{dt}\left(-\Psi_r.\cos(\theta_r)\right)$$

and 2

$$v_\beta = R_S.i_\beta + L_S.\tfrac{di_\beta}{dt} + \tfrac{d}{dt}\left(-\Psi_r.\sin(\theta_r)\right)$$

describe the two-phase equivalent circuit for the permanent magnet synchronous motor. The two-phase currents are derived using the Clarke transform that calculates the currents in two quadrature windings that will produce the same field as the currents in the three phase windings. The two-phase winding currents are the outputs of the forward rotation function that drive the space vector PWM generator.

The important aspect of the two-phase circuit model is that the back EMF terms are time derivatives of cosine and sine flux functions, and so they can be determined through integration. The detailed control schematic for angle and speed estimator shown in Figure 3 has two major subsystems. During the first stage, the flux estimator derives the rotor cosine and sine flux functions. The flux integrators include low frequency gain compensation to avoid DC saturation. At the second stage, the rotor angle phase locked loop (PLL) forces the error between the rotor angle and the estimated angle to zero. The error is calculated using a vector rotation function whose quadrature output will be zero when the rotation angle input matches the angle of the cosine and sine flux functions. The second order feedback loop in the PLL generates both angle and velocity signals.

A further feature of the PLL is the start up sequencer that is required at low speeds when the winding back EMF signal is swamped by circuit noise. The first part of the start up sequence is a parking function that drives DC current into the stator windings to align the rotor at a

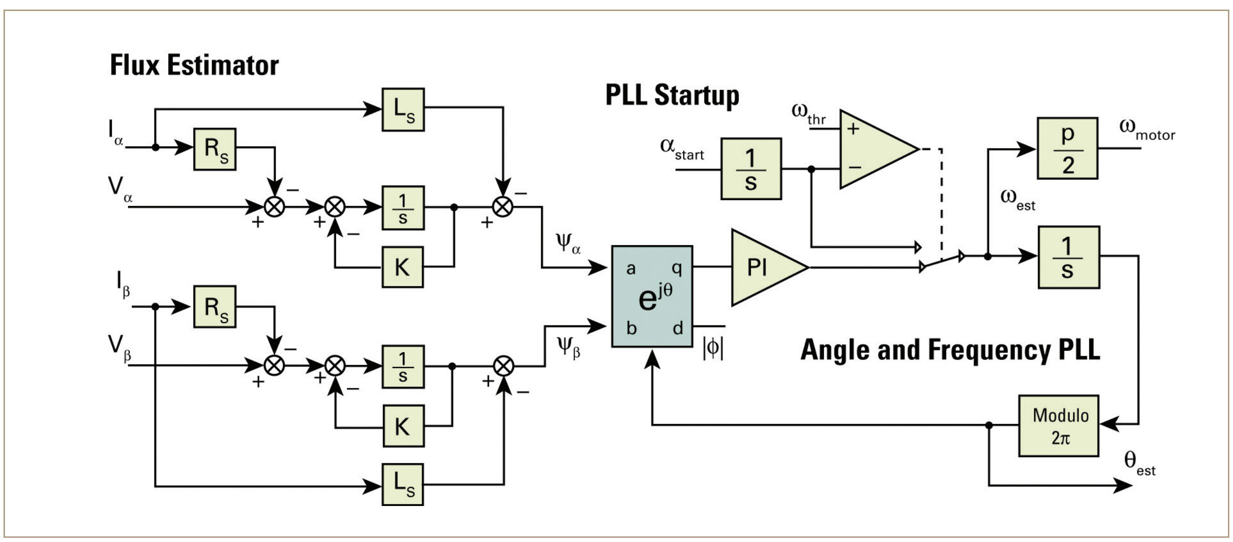
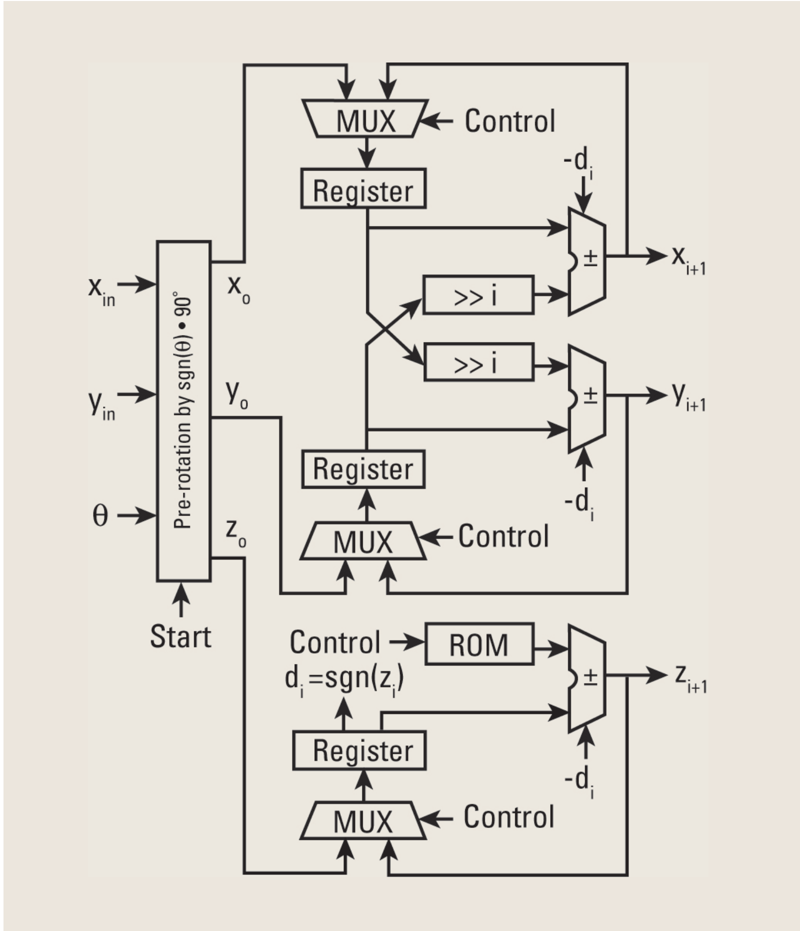


Figure 3: **Rotor angle and speed estimator**

Figure 4: **Vector rotation hardware**



known angle. Then the motor is driven with a constant current to generate a constant torque. The PLL speed integrator is fed by a motor mechanical model that estimates the motor speed from the accelerating torque and system inertia. Once the motor reaches a certain minimum speed, the PLL switches to a closed loop mode and tracks the rotor flux angle.

The motion control engine

When using a traditional DSP or RISC processor, the motor control schematic is first translated into state equations that are written in C code before software development tools can generate machine code for the processor. The motion control engine (MCE) supports a unique approach to motor control algorithm development. The MCE graphical compiler is an algorithm development tool that transforms the control schematic directly to MCE sequencer code avoiding all intermediary steps. This allows the developer to modify the reference algorithm directly, making use of a library of optimised control blocks such as PI compensators and vector rotations.

The reference algorithm includes the basic AC motor control functions, including the FOC current loop with sensorless rotor position estimation and an outer velocity loop. The inner current loop is a well-established FOC algorithm used for PMSM control, and typically does not require modification, but the optimal algorithm for the outer control loops will vary with application. Fan or pump controllers, for example, may need to regulate torque to maintain pressure, while compressor controllers may just regulate speed. In compressor control, a simple velocity loop may not properly regulate at low speeds because of the load torque ripple. A feed forward algorithm can compensate for the load torque to eliminate mechanical vibration at low speeds.

In washing machines, the controller can detect the wash load imbalance by analysing the ripple signature in the motor speed and torque before entering the spin cycle. The appliance designer edits the control algorithm schematic using the Matlab Simulink graphical user interface and can add control blocks such as comparators, summing junctions, switches and integrators. The digital control IC executes the algorithm on the IC using the matching hardware blocks from the MCE control library based on the MCE sequencer code generated by the graphical compiler.

Optimised control blocks enable a significant reduction in execution time relative to software implementations. One example of an optimised control block is the vector rotator shown in Figure 4. The CORDIC vector rotation has been developed specifically for ASIC implementation that relies on a series add, subtract and shift functions that yield 12bit accuracy in only 13 cycles [1]. This calculation is 10 times faster than the calculation using Taylor expansion on a 32bit RISC processor.

The digital timing circuits that generate the inverter PWM signals also generate the sample timing signals that allow the ASE to extract the motor winding current from the inverter DC link. This optimised combination of analog and digital signal processing circuits can simultaneously control two permanent magnet synchronous motors. There is additional signal conditioning and MCE processing capacity to support the execution of a power factor control (PFC) algorithm. Thus, the air conditioning control IC can control the input power factor, the fan and compressor motor, while traditional RISC processor based systems require separate fan and PFC control ICs.

Literature

[1] Ray Andraka, 'A survey of CORDIC algorithms for FPGA based computers', Proc. of ACM/SIGDA Sixth International Symposium on FPGAs, 1998, Monterey, CA, pp. 191-200.