FPGA-based High Performance AC Servo Motor Drive
– Accelerator™ configurable servo drive design platform

Toshio Takahashi, International Rectifier

Abstract:
In recent years, a Digital Signal Processor (DSP) or Microcontroller has been widely used for implementing the digital motor control algorithm in the motor drive industry. Application spans widely from a low performance appliance AC inverter drive to a high performance servo drive. Today’s digital AC drive consists of the implementation of torque control by means of regulating the motor current which requires very high speed computation in the range of tens of microseconds, and speed control which requires relatively moderate computation intensity in the range of hundreds of microseconds. All these functions have been implemented in one DSP or one microcontroller with/without a separate motion ASIC which performs much faster tasks such as the PWM (Pulse Width Modulated waveform generator), encoder signal interface, etc. This popularity has been due partly to availability and flexibility of desired algorithm implementation. However, as the performance envelope is further expanded and application is further diversified, new issues arise to satisfy the need for new performance challenges.

1) Traditional DSP and microcontroller have the fixed motion peripheral hardware logic and communication ports. Constraint on these fixed hardware logics inside a DSP and microcontroller imposes limit on the adaptation of different types of feedback devices and network connectivity, and unique customization of motion peripheral hardware logics.

2) Inadequate performance due to a lack of computation power for high-end servo torque control results in a multiple DSP or microcontroller implementation mixed with specific motion peripheral ASIC(s); One for torque control and the other for other controls. This creates additional complexity associated with functional partitioning and interconnectivity among split modules/functions.

3) Impeded process and challenge to meet faster time-to-market development. Exponential effort in programming is already inevitable in today’s drive development process. A Specific coding technique is often used to meet demanding challenges, and to satisfy new computational speed requirements. Code maintenance ability is further deteriorated if a cryptic assembly language is used as opposed to a high level language, such as C, in order to meet the speed requirement. In reality, the torque control algorithm is particularly written in the dedicated assembly language, which adversely affects programming productivity.

In order to solve the arising problems stemmed from the traditional DSP and microcontroller approach, a new method of hardware control has been developed. International Rectifier recently introduced the Accelerator™ design platform for a flexible high performance servo drive system. The system is based on the latest generation of the FPGA which recently became affordable for implementing complicated closed loop current control and velocity control with desired motion peripheral and communication ports.

Review of the today’s servo drive - digital control architecture and functional partitioning
Today, most servo motor drive systems are implemented by digital closed loop control
instead of analog control. This has been due primarily to the rapid advancement of a Digital Signal Processor (DSP) and microcontroller applied to motor control applications. In a typical servo control system, several functions are divided into tasks which run at a different update rate depending on the required bandwidth and nature of processing priority need – real-time operation versus delayed batch process, scanned task versus one-time event driven task. Each task is controlled by a multitask operating system closely coupled with a DSP or microcontroller interrupt structure. Figure 1 shows a typical structure of a servo drive system in terms of function. Functional element, which deals with much closer machine control, requires fast processing, fast update rate and real-time processing. They are closely tied with a specific motion peripheral hardware and it sometimes requires specific coding unique to the peripheral hardware and interrupt structure inside the DSP or microcontroller.

On the contrary, tasks that are far apart from the machine side and close to the host communication or man-machine interface side, require less frequent updating and slow processing. However, it requires more memory-intensive calculation since reference command generation over the controlling parameter is more complicated than those, which are close to the machine side. For example, position reference command is much more complex as sophisticated motion profile generation advances. However, torque command is produced in a simple step function.

The fact is that torque is the fastest machine parameter, and needs to be controlled much more quickly than the speed of the motor shaft. Integral of torque is speed. Integral of speed is position. Integral of power results in motor temperature rise. Because of this chain of physical motor parameters, each parameter requires different speed of processing. It is typical that a real-time multi-tasking operating system is used to satisfy each required processing power.

However, today’s DSP or microcontroller is facing a more difficult and fundamental challenge to deal with all these tasks requiring different processing power.

Today’s servo drive requires different effort and enhancement in two diversified directions. Controlling torque requires a much faster computation update rate to sustain the closed loop bandwidth and the increasing demand of dynamics of the machine. For example, in order to achieve 4kHz torque bandwidth, the torque control loop computation update rate requires 25 microseconds range.

The role of the power management function increases. More advanced deadtime compensation, close monitoring of temperature rise of power devices, accurate motor phase voltage feedback sensing, current sensing, and motor machine parameter estimates are emerging functions which require more specific programming, fast computation, unique digital hardware logic and analog circuit. Processing speed dealing with these logic reaches nanoseconds order and is usually implemented by the dedicated ASIC.

On the other hand, the task of dealing with position control and host communication requires more memories and programming effort. More sophisticated motion profiling requires more memories in both instruction and data with complex programming and a decent computation update rate.
As shown in Figure 1, computation requirement is basically divided into two parts. A hardware rich environment requires fast computation and more intensified dedicated hardware logic in power management peripherals. A software intensive environment requires a lot of memories with more programming effort.

It is impossible to achieve both sides of the function altogether in one DSP or microcontroller. Today’s mostly available motion-DSP does not adequately satisfy this need. Lack of computation power, exponentially increasing demand of memory for programming and data, and new hardware logic in motion peripheral are difficult to implement without a compromise in performance.

If today’s motion-DSP tries to satisfy the need without modifying the fundamental structure of the DSP, the chip becomes impractical and less economical.

It also becomes difficult to implement all the motion functions in ASIC plus DSP in the framework of traditional functional partitioning, where all of the closed control including torque control, speed control, and positioning control are implemented in one DSP or microcontroller, while the motion peripheral function is implemented in the dedicated ASIC. This is due largely to the fact that torque control demands a much faster computation update rate thus it becomes difficult to perform the torque control function along with other control functions within a DSP or microcontroller software computation environment.

The traditional approach is quickly approaching the limit of not being able to satisfy performance demands, faster development time, and easy and low cost software maintenance.

**Accelerator™ servo drive architecture**
International Rectifier has recently introduced the Accelerator™ system. It is an unique servo drive development system that simplifies the development process of the digital servo drive system while achieving high bandwidth in torque control response and flexible customization of the motion peripheral and communication port. The heart of the flexibility is achieved by a configurable FPGA combined with several pre-provided
object code sets. Figure 2 shows a basic block diagram of the Accelerator™ system. All necessary control algorithms are implemented in the FPGA including communication protocol logics. The FPGA can be directly interfaced with high-voltage ICs which control and sense electrical power to the motor.

![Accelerator™ Chip Set](image.png)

**Figure 2. Basic block diagram of Accelerator system**

**Performance advantage – hardware torque loop.**

Time critical control execution is implemented by hardware rather than by software. No instruction execution is required to perform this function. It rather directly executes functions by the dedicated hardware logics. Latency and execution time does not vary. It is inherently fast and deterministic. Computation speed of the whole current control is executed well below 5 microseconds enabling a high PWM carrier frequency update. The result of torque control loop response reaches 5kHz at –3dB point. This high bandwidth torque control allows interfacing with recent low-inductance servomotors such as linear motors in order to achieve less harmonic current ripple. All control algorithms are implemented in the Xilinx Spartan2-300 FPGA.

**Flexibility – customizable motion peripheral logic**

The Servo application requires many different variations of the interface circuit of the sensor and feedback devices. The motor position sensor can be either an encoder or resolver device, depending on application needs. For example, there are many different types of encoders: incremental type or absolute type or sine/cosine type or serial data encoder. Each position feedback device requires a unique hardware logic structure to interface within a controller. The Accelerator system provides the ability to configure different types of feedback devices such as resolver interface and incremental encoder interface without changing hardware pieces. All are configurable with the on-board FPGA.

**Accelerator™ hardware construction**

Figure 3 shows the Accelerator™ servo drive design platform. The system can handle up to...
1.5kW output with 300%/1 second overload yielding 230V/7Arms continuous power rating. The position feedback is an incremental encoder interface with an index marker pulse. Input voltage can be AC115V or AC230V and either single phase or three-phase configuration. The system is equipped with a variety of protection circuitry ranging from over-current/short circuit protection circuit, over-temperature protection circuit, and over-voltage protection circuit.

The system employs the latest chip set consisting of the IR2137 and the IR2175 from International Rectifier. The IR2137 is a monolithic 600V high-voltage gate driver IC and has built-in over-current protection. The IR2175 is a monolithic 600V current-sensing IC in the small SO-8 package. These high-voltage ICs, together with the Accelerator™ FPGA, will simplify the complicated design task of analog and power electronics circuits such as gate drive, protection, and current sensing functions. The IGBT module, rated at 600V/30A, is an industry standard ECONO-PIM2 module and has all the necessary IGBTs, diodes and a thermister temp sensors. The system comes with complete schematics, layout Gerber file and Bill of Materials so that the user can modify/create his/her own version of the PCB.

Accelerator™ system Design Flow and configuration

The Accelerator™ system comes with the pre-loaded object code, which contains all the necessary elements of a servo control. There are three types of object codes available today for licensing: IRACO201(encoder based servo amplifier), IRACO202(resolver based servo amplifier), and IRACO203(sensorless control algorithm). With each object code, the user can configure the control structure, tune the regulator control loop, and choose the communication protocol without going through the restless effort of programming. Configuration details are depicted in Figure 4 where configurable switches are located at each control section. For example, the induction machine can be configured by enabling the slip-gain block and adjusting the ID and IQ current-feedback scaling gains.
In order to assist configuration, the PC based configurator program is provided. This Window based configuration tool allows the user to configure the servo amplifier and motor, tune the closed loop control regulators, and monitor any signals within the control. Typical configuration does not require even a day to complete a desired servo drive system. When compared against traditional programming approach based on a DSP or a microcontroller, the Accelerator™ approach saves significant amount of design time by eliminating programming, coding, debugging, and code maintenance.

Figure 5 shows a design process comparison between the Accelerator system and a traditional DSP based system. If a user needs to change some mechanical structure of the hardware (i.e. different type of connectors), the user can easily create his/her own PCB based on the provided layout design.

If a situation arises to change the object code, International Rectifier also licenses the source code in addition to the object code. Each datasheet is also provided. If needed, each associated test-bench file is also available as an option. All source codes are written in the high level Verilog code and highly modularized for easy expansion.

International Rectifier also has a plan to enhance functionality for future release. One of them is the network connectivity in order to satisfy the need for multi-axis motion control application.
Figure 5. Servo Drive system design process comparison