A faster current loop pays off in servo motor control

Texas Instruments

Brian Fortman
DesignDRIVE marketing manager
C2000™ MCU

Texas Instruments

Introduction

Improving the performance or bandwidth of industrial servo-drive applications like robotic assembly systems, factory automation machinery and material handling systems has several direct effects on a manufacturer's bottom line. Less waste in the manufacturing process, increased production throughput, and greater product consistency and quality will reduce manufacturing costs and in turn increase a manufacturer's revenues and profitability.

At the heart of the matter of improving servo drive performance is the system's current-loop performance or torque response. The current loop is the linchpin upon which the performance of the rest of the system depends. Even though several alternative architectures can provide a pathway to a greater current-loop bandwidth, some are more effective and cost-efficient than others. Only one solution in particular can optimize or eliminate the design trade-offs involving long, drawn-out and complex development programs, expensive bill of materials (BOM) costs, excessive power consumption, cumbersome thermal-management strategies and others. This solution, which is based on TI's commercial-off-the-shelf (COTS) C2000™ real-time microcontrollers (MCUs) and the enhanced DesignDRIVE development platform, increases the bandwidth of a servo motor drive's current control loop.

Loops within loops

The architecture of real-time industrial servo drive applications involves several control loops that monitor the motor's critical parameters. These control loops provide feedback for the purpose of maintaining an established setpoint as operating conditions change or when responding to a new setpoint, thus controlling parameters such as current, torque, speed and position. The control system uses feedback loops to ensure that its actions achieve the intended results. If the feedback indicates that a parameter has strayed beyond its target for that cycle, the system must respond quickly and adjust its actions. Responding quickly to sudden disturbances reduces the risk of faulty or inconsistent servo motor operations that will affect the output of the industrial process.

Servo applications typically employ at least three nested control loops for current, speed and position. The current-loop bandwidth is the most critical because it dictates the maximum performance of the speed and position loops. The system's control loops are interrelated and dependent upon each other, but the current loop is foundational. In fact, the bandwidth of the current loop will limit the maximum performance of the entire servo system. In most systems, the current-loop bandwidth will exceed the bandwidth of the speed and position loops by nearly 10 times or even higher. As a result, a faster current loop is essential to higher system performance.

Current loop close-up

The current loop controls the torque in a servo motor by manipulating the pulse-width modulator (PWM) outputs that drive an inverter. The motor currents are monitored and fed back to the current-loop controller and the controller updates the PWM outputs if necessary. The current-loop feedback path quantifies the analog output of the motor current sensor with a high-precision analog-to-digital converter (ADC), and then feeds the result to the current-loop controller. Several different modules of field-oriented control (FOC) algorithms process this sample before the controller makes any adjustments to the PWM's outputs.

Many conventional proportional integral (PI) current-loop controllers limit the current-loop bandwidth to approximately 10 percent of the PWM carrier frequency, which is typically in the 10-kHz range. This would yield a current-loop bandwidth of 1 kHz. To increase the bandwidth, some drive systems increase the carrier frequency to 30 kHz or higher, which can increase the current loop's bandwidth but will also increase the inverter switching losses and demand higher processing performance from the digital control system. A higher switching frequency requires increased gate-drive power, which then increases the size and complexity of the system's power supply. Plus, the higher switching losses will mean additional heat dissipation, necessitating more extensive thermal-management strategies such as larger, heavier and more expensive heat sinks, fans or coolants.

Clearly, the historical solutions for increasing current-loop bandwidth come with strings attached. The cleanest, most effective architecture would simultaneously address the need for improved processing capabilities to accommodate shorter feedback cycles and retain a lower carrier frequency to avoid the undesirable power and thermal trade-offs. To augment the system's processing resources, some designs have employed current-loop processing data paths in a field programmable gate array (FPGA) in addition to the MCU, but this only addresses the processing half of the equation and comes with its own set of strings.

FPGA architectures

Adding an FPGA does not eliminate the need for an MCU in a servo motor drive design because the MCU controls the speed and position feedback loops, and in some cases the system's user interface and connectivity. Inserting an FPGA into the current control loop has a ripple effect throughout the rest of the design. See **Figure 1** on the following page. To accommodate the entire current control loop, the FPGA will require custom design work to incorporate PWM generation, inverter protection circuitry and logic to control the analog sampling. This design work will complicate system development, drive up the gate count and cost of the FPGA, and increase design and support risks.

In addition, most FPGAs do not have embedded ADCs capable of sufficient performance and resolution for servo motor-control applications, thus necessitating the design of discrete ADCs and analog comparator circuitry into the system. Also, the FPGA will likely require parallel input/output (I/O) interfaces to minimize data-transfer latencies from the ADC and enable fast communications with the MCU. Again, the FPGA's pin count will go up, along with cost and board-space requirements. And lastly, an FPGA design will typically demand a more complex power subsystem, possibly requiring additional power rails and higher currents than the MCU and other components would typically need.

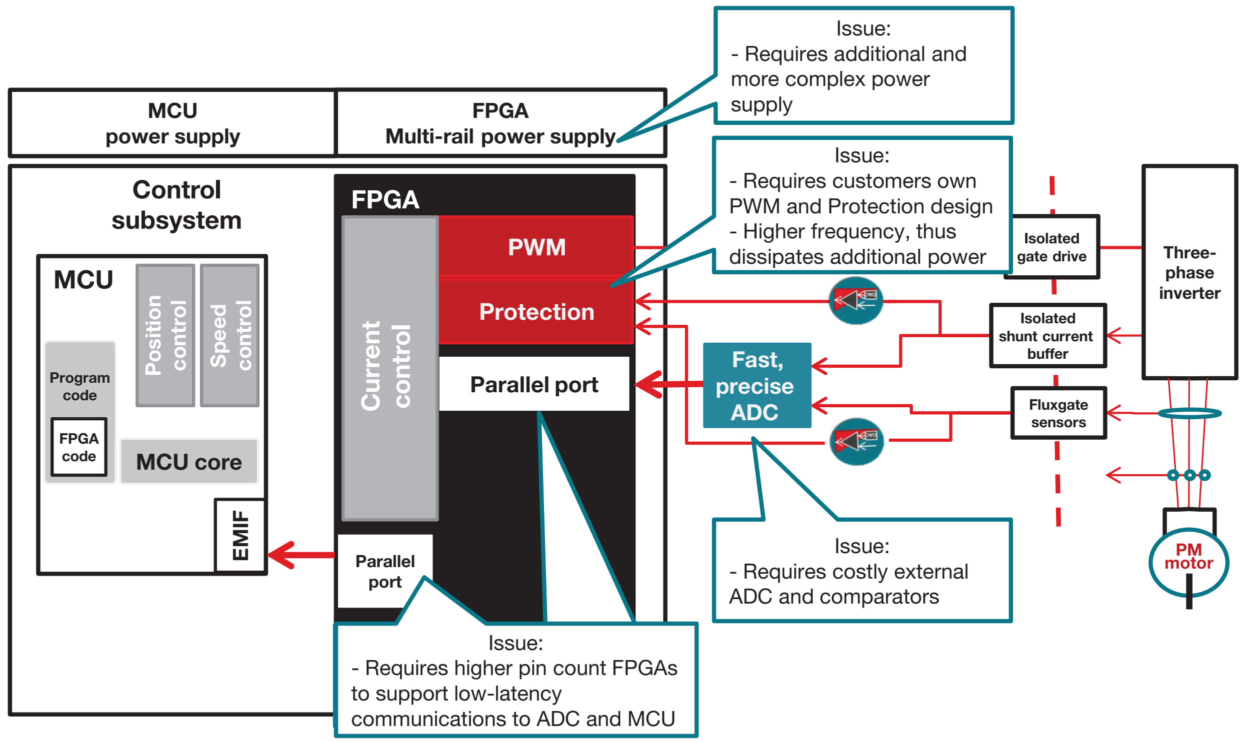

***Figure 1**. System challenges associated with FPGA-based current-loop control.*

Of course, once an FPGA is present in a design, it invariably becomes the home for functionality that may have been previously implemented in discrete devices or not included in the system at all. In a servo motor-control application, this could mean integrating delta-sigma ADC filters, encoder interfaces, pulse-train output generation and other capabilities. Once this trend begins, the logical end is a very large, high-pin-count, expensive FPGA.

Fast Current Loop software

An effective alternative that avoids many of the challenges that arise with the addition of an expensive FPGA to a servo-control application would be to implement these functions in the MCU. Outside of TI's C2000 family of real-time MCUs, few if any such capable MCUs exist. TI harnessed two decades of real-time industrial control system design experience when applying C2000 MCUs to the development of the new Fast Current Loop (FCL) software. The DesignDRIVE development platform with FCL software vastly simplifies the design process while overcoming the less desirable trade-offs.

FCL software, quite simply, breaks many of the long-held assumptions in current-loop design. For instance, designers of servo motor control applications have for years placed limitations on the current-loop bandwidth because they assumed that the controller could update the PWM only once per cycle of the control loop. Now, with an integrated high-performance successive approximation register (SAR) ADC, ADC post-processing hardware, trigonometric math accelerator and other cycle-scavenging resources, C2000 MCUs can sample the motor currents, convert them to digital data, process the data and update the PWM generator *in less than 1 μs*. The field-oriented control (FOC) processing and PWM update takes *less than 500 ns*. Because of FCL software and C2000 MCU ePWM immediate mode capabilities, subcycle PWM updates occur as soon as possible, instead of waiting for an entire control loop cycle to transpire.

Designers also assumed that achieving a higher current-loop bandwidth would require higher carrier frequencies. So, for example, increasing the bandwidth by a factor of 3 from 1 kHz to 3 kHz would require increasing the carrier frequency from 10 kHz to 30 kHz.

But the C2000 MCU's ability to perform subcycle PWM updates in less than a microsecond improves the current-loop bandwidth, making it possible to reduce or optimize the switching frequency for a given bandwidth requirement. Tests have shown that FCL can increase the current-loop bandwidth by a factor of 3 from 1 kHz to approximately 3.3 kHz while maintaining a carrier frequency of 10 kHz. Therefore, using FCL software avoids the trade-offs of increased power consumption, greater heat dissipation, and more complex and expensive thermal-management strategies.

The TMS320F28379 MCUs are highly integrated for industrial drive systems. On-chip features such as Sigma-Delta filters, analog comparators and PWM protection circuits save the system-level costs when compared to adding these external components. The Position Manager solutions for absolute encoder interfaces as well as Pulse Train Output generation reduce the need to add these logic gates as well. With the release of the Fast Current Loop software, the system-level functions that have traditionally been developed for FPGAs are now all available in the DesignDRIVE solutions library. See **Figure 2**. Therefore, we can refer to the TMS320F2837x family of MCUs as drive-control systems-on-chip.

Another challenge of high-performance current-loop design has been the stability of the current-loop controller at high speeds. Digital or transport delays progressively reduce the phase margin of the current control loop at higher speeds, eventually leading to a loss of control. Traditional controllers in the current loop do not model these digital delays well.

However, FCL software includes an efficient algorithm that assumes a more appropriate system

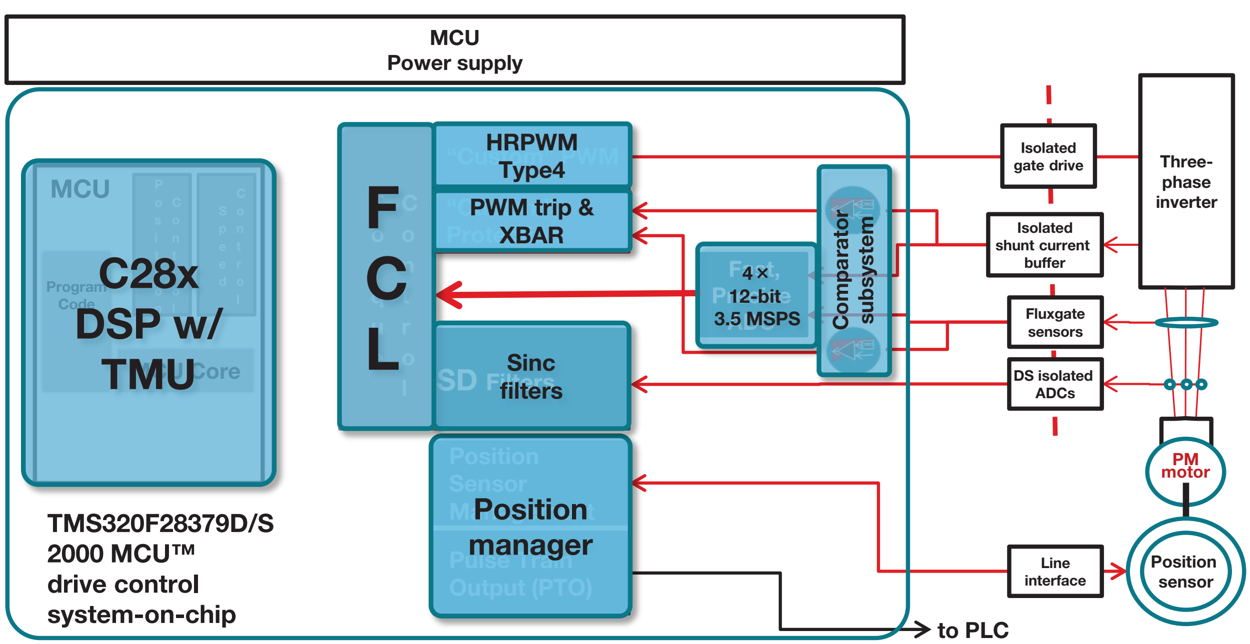

Figure 2. *C2000 system integration for industrial drives.*

model, compensating for the inherent transport delay of the system. This is embodied in the DesignDRIVE complex controller (CC) and results in perfect pole-zero cancellation at all times, ensuring stability at higher speeds.

The hardware and technology breakthroughs that contribute to FCL software have emerged from more than 20 years of experience in embedded processing and motor-control systems by the C2000 MCU DesignDRIVE engineers. Certainly, the MCUs themselves have been endowed with powerful, high-performance processing units, coprocessors and specialized processing units, but the C2000 MCU's architecture is just as essential because it enables a high degree of real-time parallel processing through very low latencies and high processing determinism. Ultimately, this makes a C2000 MCU in a servo-drive application a *cycle scavenger*, saving processor cycles over the entire loop and transforming what may have been longer sequential steps into shorter simultaneous processes.

For example, 200-MHz C28x processing cores are, of course, critical for breaking apart the workload to optimize parallel processing, but so is the real-time deterministic architecture that surrounds them. Because the architecture keeps the system's processing as jitter-free as possible, cycles are not wasted recovering from timing issues or running error-correction routines. There are no data-transfer latencies between the ADC and the C28x cores, or between the cores and the PWM. Transferring a sample from the ADC to the C28x cores or updating the PWM takes only a single cycle. See **Figure 3**.

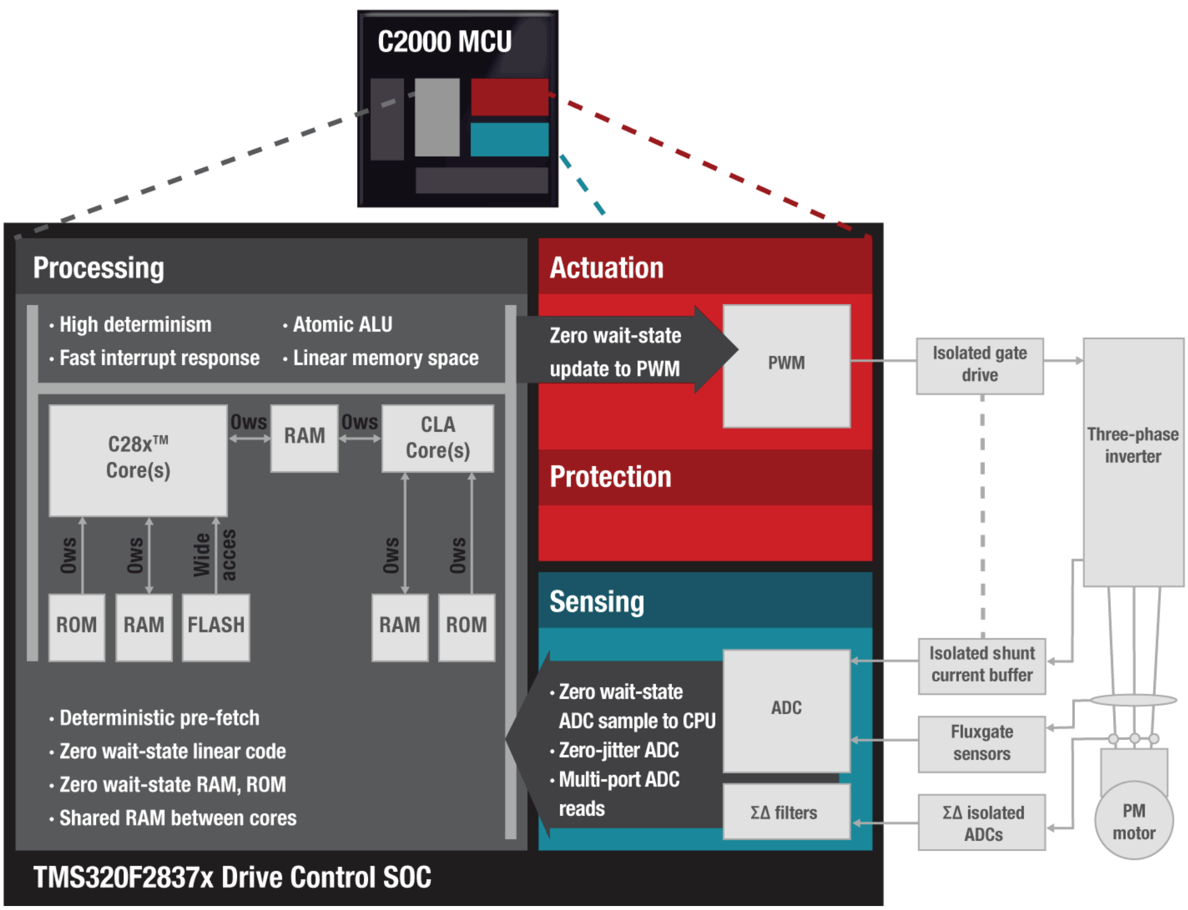

Figure 3. C2000 MCUs architecture for real time control.

Several C2000 MCUs, such as the Delfino™ TMS320F28379D processor, include a powerful ADC post-processing block capable of performing certain sample-preparation routines in dedicated hardware that the C28x cores would normally have executed, saving cycles on the main processors and freeing them to perform other tasks in parallel. The cores themselves have been equipped with coprocessors and specialized processing units that prove useful in servo-control applications. For instance, the main processing cores can offload encoder-feedback processing to the control law accelerator (CLA) coprocessor, which excels at high-level mathematical processing. Moreover, both the FOC processing and the CC use a specialized processing unit, the trigonometric math unit (TMU). With the aid of the TMU, you can improve Park transform cycle performance by a factor of 8 over comparable MCUs in the industry. In addition, the C28x core's 32-bit floating-point processing accelerates the execution of several different types of mathematically intense calculations used by the FCL algorithms, such as minimum/maximum, compare and square root.

Conclusion

During the development of a servo motor-drive application, designers face a diverse set of subtle trade-offs, many times placing high performance, cost effectiveness, power efficiency and other factors in direct opposition with one another. Simple yet effective solutions may be hard to come by, but fortunately, TI's experience and expertise in real-time industrial drive control systems has led to many enhancements to the C2000 family of real-time MCUs.

The DesignDRIVE developers kit features the latest enhancement, FCL software. Taking advantage of a real-time cycle-scavenging architecture, high-performance processing resources and the fast data throughput of C2000 MCUs to significantly increase the current control-loop bandwidth, FCL software achieves subcycle updates of the PWM in less than 1 µs and without the assistance of external processing elements like an FPGA or ADC. Compared to FPGA-based systems, a C2000 drive control system-on-chip with FCL software delivers similar performance, while simplifying servo drive development and reducing system costs, power complexities and board space. Compared to traditional MCU-based systems, FCL software can potentially triple a drive system's torque response and double its maximum speed without increasing the carrier frequency.

For more information

For additional information about Fast Current Loop software, visit our website.

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

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