

Accelerated Real-Time Control Processing: From Tried-and-True to Next-Generation MCUs



ABSTRACT

This paper examines the significance of specialized hardware accelerators for real-time control systems. The document emphasizes the role of the processing phase of a real-time control loop and explains the role of different hardware accelerators integrated into high-performance microcontrollers (MCUs) that assist in reducing the overall latency of the system. Enhanced floating-point capabilities, integrated trigonometric and control-law accelerators, and various techniques for optimizing communication are discussed.

The paper concludes by presenting two MCU product series tailored for advanced real-time control applications: the C2000™ real-time MCU portfolio for dedicated control system designs and the Arm®-based AM26x MCU portfolio for expansive processing abilities within Arm-centric ecosystems.

Table of Contents

1 Introduction	2
2 Floating-Point Unit	2
3 Trigonometric Math Unit	3
4 Control Law Accelerator	3
5 Viterbi, Complex Math, and CRC Unit	3
6 Industrial Communication Subsystem	3
7 Discover the Best MCU for Real-Time Control Needs	4

List of Figures

Figure 1-1. Primary Functional Blocks of a Real-Time Control Loop.....	2
--	---

1 Introduction

Real-time control systems serve as the nerve centers for numerous critical applications. These systems orchestrate a delicate balance of sensing, processing, and actuating, and their significance cannot be overstated.

Consider the traction inverter in an electric vehicle. The primary function of the inverter is to convert the DC power of the battery into the three-phase AC power on which motors rely. In this context, every millisecond is crucial. The inverter consistently adjusts to variables like road conditions, load demands, and driver inputs, providing a smooth and energy-efficient drive. Similarly, an industrial robot is programmed for precision in repetitive tasks. Whether it is the angle of rotation or the force applied, speed and accuracy are paramount. Even a minor deviation can lead to defective products. Thanks to real-time control systems, motors in both robots and vehicles operate safely and seamlessly around the clock.

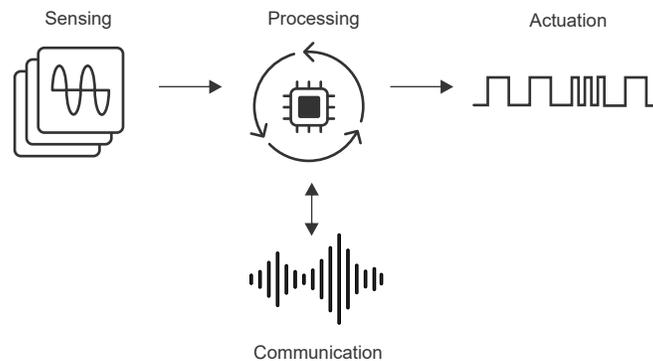


Figure 1-1. Primary Functional Blocks of a Real-Time Control Loop

Every real-time control system encompasses:

1. **Sensing:** Devices consistently capture real-time data, from temperature sensors in EV batteries to position sensors in robotic arms.
2. **Processing:** Sophisticated algorithms interpret this raw data. Speed and accuracy in this phase are crucial, given the real-time demands.
3. **Actuation:** The system receives commands, resulting in tangible actions based on the processed data.

In designing real-time control systems—whether for multi-axis drives, industrial robots, electric vehicles, or solar inverters—the primary objective is to minimize latency to optimize performance. High-performance microcontrollers, equipped with specialized hardware accelerators, make this possible. These accelerators are specifically designed for the efficient processing of complex control algorithms. The following sections focus on the processing phase of a real-time control loop, discussing algorithm development and the role of hardware accelerators in minimizing system latency. The paper concludes with a review of the standout MCU product series TI has created for these applications.

2 Floating-Point Unit

Many real-time control system designs start with simulation tools, where control algorithms are developed with floating-point math. Floating-point math provides a large dynamic range, thereby making it easier to develop code than using fixed-point math. For example, scaling and saturation are no longer an issue. A control algorithm developed with floating-point math is also more robust because the values do not wrap around the number line on an overflow or underflow.

A high-performance MCU for real-time control has an extremely fast CPU with hardware support for 32-bit fixed-point and 32-bit (or 64-bit) floating-point mathematical operations. This makes it easy to port the control algorithm to the device and allows for versatile concurrent processing.

3 Trigonometric Math Unit

Within the control algorithm, certain mathematical operations emerge as especially critical. Operations such as Park and inverse Park transforms, space-vector generation, direct-quadrature-zero and inverse direct-quadrature-zero transforms, FFT magnitude, and phase calculations are fundamental. Why? Because these operations play pivotal roles in signal modulation, data conversion, and system analysis, helping in translating the raw sensor data into actionable insights. Furthermore, other foundational operations like sine, cosine, arctangent, division, and square root form the bedrock of these mathematical manipulations. However, being cycle-intensive, these operations can potentially slow down system responses when run on the CPU of the microcontroller.

To make sure that these operations do not become a bottleneck, using enhanced floating-point capabilities is paramount. One effective approach is to optimize these functions using an extended instruction set. A well-designed compiler is also essential, capable of generating trigonometric and arithmetic instructions automatically. This makes sure that the compiler leans on these optimized instructions rather than relying solely on standard run-time support library calls, leading to faster computations.

4 Control Law Accelerator

Many complex real-time control applications require an even higher level of processing performance from an integrated control-law accelerator. For example, a typical power digital controller consists of an analog-to-digital converter (ADC) to capture the input data, a math engine to compute the control law algorithms (for example, PID, two-pole or two-zero, and three-pole or three-zero compensators), and a PWM channel to output the calculated waveform. A high-performance MCU integrates these functions within a single device to minimize latency (and reduce system complexity and cost), yielding the absolute minimum sample to output delay.

Support for 32-bit floating-point math-intensive computations is standard, so low-level control loops can be managed with more performance and efficiency than the CPU. Additionally, direct access to memory and control peripherals, such as the A/D convertor and PWM modules, further helps to minimize latency. An integrated control law accelerator of this kind can respond to peripheral triggers without CPU intervention. The accelerator does not use interrupts for hardware synchronization, nor must the accelerator do any context switch, increasing efficiency. This approach eliminates jitter and the execution time becomes deterministic. For detailed benchmarks and insights on these accelerators, see the [Enhancing the Computational Performance of the C2000™ Microcontroller Family](#) application note.

5 Viterbi, Complex Math, and CRC Unit

As the complexity of a system grows, so too does the importance of reliable and rapid communication between devices and control units. This is not merely about data exchange but providing timely, precise responses to changing conditions. From electric vehicles syncing with charging stations to industrial robots relaying data, the quality of communication protocols directly impacts system performance.

A high-performance MCU with a tightly-coupled fixed-point accelerator for Viterbi decoding, complex FFT, complex filters, and cyclical redundancy check (CRC) greatly improves communication performance. This accelerator can also be used for general-purpose signal processing, such as filtering and spectral analysis, expanding the function and value in the system.

6 Industrial Communication Subsystem

Another option is to use a programmable real-time unit with a specialized industrial communication subsystem. This option provides versatility and facilitates real-time processing and industrial protocol integration, notably in systems that require deterministic communication. A dual-core architecture and dedicated RAM facilitate swift and concurrent data processing, providing minimal lag. Independent cores can operate simultaneously, managing different tasks, thus optimizing communication flows and making sure that high-priority tasks are executed without delay. This subsystem, by design, can interface seamlessly with a wide range of industrial communication protocols, providing flexibility and broad application potential.

7 Discover the Best MCU for Real-Time Control Needs

As real-time control systems evolve in complexity and their requirements broaden, selecting the right microcontroller is more crucial than ever. With over 25 years of experience in developing real-time control technology, TI proudly offers two exceptional microcontroller product series tailored to meet these advanced demands.

For those engaged in rigorous real-time control applications, the [TI C2000 real-time MCU portfolio](#), based on proprietary architecture, provides our most dedicated control-system designs. These devices are tailored specifically for demanding domains such as motor drives and renewable energy systems. These products offer advanced on-chip components, including high-resolution PWMs, ADCs, and many of the distinct hardware accelerators discussed in this article. With all of these tailored features, C2000 MCUs are our premier option for high-performance control projects that demand precision.

Conversely, when projects necessitate both robust real-time control and the ability to handle diverse workloads, our *Arm[®]-based AM26x MCU portfolio* performs extremely well. Based on the popular Arm Cortex-R architecture, this portfolio does not just excel in control (that leverage integrated C2000 control features) but is also adept at managing tasks ranging from data analytics to intricate communication protocols. The comprehensive set of features, inclusive of enhanced security modules and specialized communication interfaces, as well as up to [SIL 3](#) or [ASIL-D](#) functional safety marks our AM26x MCUs as the top pick for professionals seeking expansive processing abilities and seamless integration within Arm-centric ecosystems.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2023, Texas Instruments Incorporated