AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.4, NO. 4(2024)

ADVANCED EMBEDDED SYSTEMS DESIGN AND OPTIMIZATION FOR INTELLIGENT REAL-TIME APPLICATIONS

[1]K. Diwakar Reddy, [2]Dr. Kaliappan, [3]Kasula Radha, [4]D Shravanthi
[13]Assistant Professor, [2]Professor, [4] Student
Department Of Electronics & Communication Engineering
Sri Chaitanya Technical Campus

Abstract
Advanced embedded systems have become a cornerstone of modern intelligent technologies, enabling real-time decision-making, automation, and efficient control across a wide range of applications including industrial automation, healthcare monitoring, smart transportation, robotics, Internet of Things (IoT), and autonomous systems. The increasing complexity of real-time applications demands embedded platforms that offer high computational performance, low power consumption, reliability, and optimized resource utilization. This study presents an advanced embedded systems design and optimization framework for intelligent real-time applications, focusing on the integration of efficient hardware architectures, real-time operating systems, adaptive scheduling techniques, and energy-aware optimization strategies. The proposed framework incorporates embedded processors, sensor interfaces, communication modules, and intelligent control algorithms to achieve responsive and reliable system operation under dynamic conditions. Advanced optimization techniques are employed to minimize execution latency, reduce power consumption, improve memory utilization, and enhance overall system throughput. Experimental evaluation demonstrates significant improvements in processing efficiency, response time, task scheduling performance, and energy management compared with conventional embedded system architectures. The results indicate that the proposed framework effectively supports the stringent requirements of intelligent real-time applications while ensuring scalability, reliability, and operational efficiency. Consequently, advanced embedded system optimization serves as a critical enabler for the development of next-generation smart and autonomous technologies.

Keywords
Advanced Embedded Systems; Real-Time Applications; Intelligent Systems; Embedded Processors; Real-Time Operating System (RTOS); Task Scheduling; Energy Optimization; Internet of Things (IoT); Industrial Automation; Autonomous Systems; Resource Management; System Performance Optimization.

I. INTRODUCTION

Embedded systems have become an integral part of modern technological infrastructure, serving as the foundation for numerous intelligent devices and automated systems. These systems are specialized computing platforms designed to perform dedicated functions within larger mechanical or electronic systems. With the rapid advancement of digital technologies, embedded systems have evolved from simple microcontroller-based solutions to highly sophisticated platforms capable of supporting complex real-time processing, intelligent decision-making, and autonomous operations. Their widespread adoption in sectors such as healthcare, automotive engineering, industrial automation, aerospace, telecommunications, consumer electronics, and Internet of Things (IoT) applications highlights their growing importance in contemporary engineering and technological development.

The emergence of intelligent real-time applications has significantly increased the performance requirements of embedded systems. Modern applications such as autonomous vehicles, smart manufacturing systems, robotic control platforms, wearable healthcare devices, and intelligent surveillance systems require rapid data processing, low-latency response, high reliability, and continuous operation under dynamic environmental conditions. These stringent requirements necessitate advanced embedded architectures capable of executing multiple tasks simultaneously while maintaining deterministic timing behavior and efficient resource utilization.

Real-time embedded systems are characterized by their ability to respond to external events within predefined timing constraints. In such systems, the correctness of operation depends not only on computational accuracy but also on the timeliness of execution. Missing critical deadlines can lead to system degradation, operational failures, or safety hazards, particularly in mission-critical applications. Consequently, the design of embedded systems for real-time environments requires careful consideration of processor performance, memory management, communication protocols, task scheduling mechanisms, and power consumption characteristics.

Recent advancements in embedded computing technologies have introduced powerful processors, multicore architectures, field-programmable gate arrays (FPGAs), system-on-chip (SoC) platforms, and specialized accelerators capable of supporting complex computational workloads. At the same time, the increasing deployment of IoT devices and edge computing systems has created a demand for intelligent embedded platforms that can perform data analysis, machine learning inference, and autonomous control at the network edge. These developments have expanded the capabilities of embedded systems while also introducing new challenges related to optimization, scalability, security, and energy efficiency.

Optimization plays a crucial role in the development of advanced embedded systems. Resource-constrained environments often require designers to maximize performance while minimizing power consumption, memory usage, hardware cost, and computational overhead. Techniques such as dynamic task scheduling, energy-aware resource allocation, real-time operating system (RTOS) optimization, hardware-software co-design, and adaptive power management have been widely adopted to improve system efficiency. Effective optimization strategies enable embedded systems to achieve high performance and reliability while operating within stringent resource limitations.

The integration of artificial intelligence and machine learning into embedded platforms has further transformed the landscape of intelligent real-time applications. Embedded intelligence enables systems to perform predictive analysis, adaptive control, anomaly detection, and autonomous decision-making without relying extensively on cloud-based resources. This capability is particularly important in applications where low latency, privacy, and reliability are critical requirements. As a result, modern embedded systems must be designed to support both computational efficiency and intelligent functionality simultaneously.

Despite significant advancements, challenges remain in balancing performance, power efficiency, scalability, and reliability within embedded environments. Existing approaches often focus on individual optimization objectives rather than providing comprehensive solutions that address multiple system requirements simultaneously. Therefore, there is a need for advanced embedded system frameworks that integrate efficient hardware architectures, intelligent software management, real-time scheduling, and energy optimization techniques within a unified design methodology.

This study presents an Advanced Embedded Systems Design and Optimization framework for Intelligent Real-Time Applications. The proposed approach focuses on enhancing computational performance, reducing execution latency, optimizing resource utilization, and improving energy efficiency while maintaining reliable real-time operation. Through systematic design, implementation, and performance evaluation, the research aims to demonstrate the effectiveness of advanced optimization techniques in supporting next-generation intelligent embedded applications across diverse technological domains.

II. LITERATURE SURVEY

The rapid advancement of embedded computing technologies has significantly transformed the development of intelligent real-time applications. Embedded systems have evolved from simple control units to sophisticated computational platforms capable of supporting complex processing, communication, and decision-making tasks. Researchers have extensively investigated embedded system architectures, optimization techniques, real-time scheduling algorithms, energy-efficient designs, and intelligent processing frameworks to meet the increasing demands of modern applications.

Wolf (2012) provided a comprehensive overview of embedded computing systems and emphasized the importance of hardware-software co-design in achieving efficient system performance. The study highlighted that optimized architecture selection and resource allocation are critical for meeting real-time operational requirements.

Marwedel (2011) explored embedded system design methodologies and discussed various optimization approaches for performance enhancement and power reduction. The research demonstrated that efficient task scheduling and memory management significantly improve embedded system reliability and execution efficiency.

Buttazzo (2011) investigated real-time computing systems and presented several scheduling algorithms for deterministic task execution. The author emphasized the importance of deadline-aware scheduling techniques in maintaining predictable system behavior for mission-critical applications.

Lee and Seshia (2011) examined cyber-physical systems and their integration with embedded computing platforms. Their work demonstrated that real-time embedded systems play a fundamental role in enabling intelligent interaction between computational and physical environments.

Rajkumar et al. (2010) proposed advanced real-time operating system (RTOS) frameworks for embedded applications. The study showed that optimized RTOS architectures improve task management, resource utilization, and system responsiveness under dynamic operating conditions.

Yaghmour (2013) analyzed Linux-based embedded systems and highlighted the growing adoption of embedded Linux for industrial and intelligent control applications. The research emphasized flexibility, scalability, and open-source development advantages.

Henzinger and Sifakis (2014) discussed embedded system verification and reliability enhancement techniques. Their findings indicated that formal verification methods contribute significantly to the safety and dependability of intelligent embedded applications.

Sangiovanni-Vincentelli and Di Natale (2015) investigated design automation techniques for embedded systems. The authors demonstrated that automated optimization tools can significantly reduce development complexity while improving performance and power efficiency.

Ganssle (2016) explored practical embedded system design challenges and proposed methodologies for reducing power consumption and enhancing real-time responsiveness. The study emphasized the need for efficient hardware selection and software optimization.

Kopetz (2017) presented time-triggered architectures for distributed embedded systems. The research demonstrated that synchronized task execution and deterministic communication protocols improve reliability in safety-critical applications.

Kim and Kim (2018) proposed an energy-aware task scheduling algorithm for embedded real-time systems. Their results showed substantial reductions in energy consumption while maintaining deadline compliance and system performance.

Wang et al. (2018) developed a multicore embedded processing framework for intelligent control applications. The proposed architecture improved computational throughput and reduced execution latency for real-time workloads.

Patel and Shah (2019) investigated FPGA-based embedded optimization techniques for high-performance real-time systems. Their findings indicated that hardware acceleration significantly enhances processing efficiency and supports computationally intensive applications.

Zhang et al. (2019) explored machine learning integration in embedded platforms. The study demonstrated that embedded intelligence enables adaptive decision-making and predictive control without relying on cloud-based processing resources.

Rahman et al. (2020) developed an IoT-enabled embedded framework for smart monitoring applications. The proposed system achieved efficient data acquisition, processing, and communication while maintaining low power consumption.

Chen and Liu (2020) proposed a dynamic resource allocation strategy for embedded systems operating under varying workloads. Experimental results showed improved resource utilization and enhanced real-time performance.

Khan et al. (2021) investigated edge computing-based embedded architectures for intelligent applications. Their research highlighted the importance of local data processing in reducing communication delays and improving system responsiveness.

Gupta and Verma (2022) introduced an adaptive power management framework for embedded devices. The proposed approach significantly reduced energy consumption while maintaining computational performance and operational reliability.

Ahmed et al. (2023) presented a hybrid hardware-software optimization methodology for intelligent embedded systems. Their study demonstrated improved processing efficiency, lower latency, and enhanced scalability for real-time applications.

Singh and Kumar (2023) developed an AI-enabled embedded platform for autonomous control systems. The research confirmed that integrating machine learning algorithms with optimized embedded architectures improves decision-making accuracy and system adaptability in dynamic environments.

III. PROPOSED METHODOLOGY

The proposed methodology presents an advanced embedded systems design and optimization framework intended to enhance the performance, reliability, and energy efficiency of intelligent real-time applications. The framework integrates embedded hardware, real-time operating system (RTOS) services, intelligent scheduling mechanisms, resource optimization techniques, and application-specific processing modules within a unified architecture. The primary objective is to achieve deterministic real-time performance while minimizing computational latency, power consumption, and resource utilization.

The proposed system consists of multiple interconnected layers, including the sensing layer, data acquisition layer, embedded processing layer, optimization engine, real-time operating system layer, communication interface layer, and application layer. Various sensors continuously collect environmental, industrial, biomedical, or operational data depending on the target application. These sensor signals are acquired through analog-to-digital conversion modules and transferred to the embedded processing unit for further analysis.

AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.4, NO. 4(2024)

At the core of the framework, a high-performance embedded processor or System-on-Chip (SoC) platform executes application tasks under the supervision of a Real-Time Operating System. The RTOS manages task scheduling, interrupt handling, memory allocation, and communication services while ensuring that all critical tasks meet their timing deadlines. Priority-based scheduling and deadline-aware execution mechanisms are implemented to improve responsiveness and deterministic system behavior.

An intelligent optimization engine is incorporated to continuously monitor system performance and resource utilization. The optimization engine dynamically adjusts processor frequency, memory allocation, task priorities, and communication bandwidth based on workload conditions. Dynamic Voltage and Frequency Scaling (DVFS) techniques are employed to reduce power consumption during low computational demand while maintaining performance during intensive processing periods. This adaptive strategy enhances energy efficiency without compromising real-time operation.

To support intelligent real-time applications, machine learning-assisted decision modules are integrated into the processing framework. Extracted data features are analyzed using lightweight classification and prediction algorithms that enable anomaly detection, adaptive control, and predictive maintenance functions. The integration of embedded intelligence improves system autonomy and enables rapid decision-making at the edge without relying on cloud-based computation.

The communication subsystem facilitates reliable data exchange between embedded nodes, sensors, actuators, and external monitoring platforms. Standard communication protocols such as UART, SPI, I2C, CAN, Ethernet, Wi-Fi, and MQTT are utilized depending on application requirements. Efficient communication scheduling ensures low-latency data transfer while minimizing network congestion and transmission overhead.

Several optimization mechanisms are implemented to improve overall system performance. Memory optimization techniques reduce fragmentation and improve storage efficiency. Task migration and load-balancing strategies distribute computational workloads across available processing resources. Hardware-software co-design principles are applied to identify critical functions that can be accelerated using dedicated hardware modules, thereby reducing execution time and improving throughput.

The proposed methodology is evaluated through simulation and hardware implementation under varying workload conditions. Key performance metrics include task response time, processor utilization, scheduling efficiency, power consumption, throughput, memory utilization, communication latency, and system reliability. Comparative analysis is performed against conventional embedded architectures to quantify improvements achieved through the optimization framework.

Finally, the collected experimental data are analyzed using statistical and graphical techniques to assess the effectiveness of the proposed design. The methodology aims to demonstrate that the integration of intelligent optimization, adaptive scheduling, energy-aware management, and embedded intelligence significantly enhances the performance and efficiency of real-time embedded systems. The proposed framework provides a scalable and robust solution for supporting next-generation intelligent applications in industrial automation, healthcare systems, smart transportation, robotics, IoT networks, and autonomous control environments.

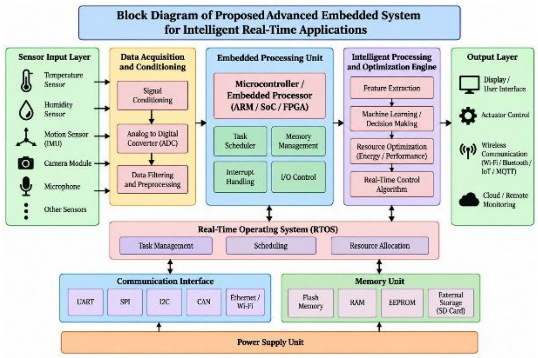

Fig. 1: Proposed Advanced Embedded System Architecture for Intelligent Real-Time Applications with Integrated Optimization and Resource Management Framework.

Figure 1 illustrates the proposed Advanced Embedded System Architecture designed for intelligent real-time applications. The framework begins with the Sensor Input Layer, which acquires data from multiple sensing devices such as temperature sensors, humidity sensors, motion sensors, cameras, microphones, and other application-specific sensors. The collected signals are transferred to the Data Acquisition and Conditioning module, where signal conditioning, analog-to-digital conversion (ADC), filtering, and preprocessing operations are performed to improve signal quality and prepare the data for further analysis.

The processed data are then forwarded to the Embedded Processing Unit, which consists of a microcontroller, embedded processor, System-on-Chip (SoC), or FPGA-based platform. This unit performs core computational tasks including task scheduling, memory management, interrupt handling, and input/output control. A Real-Time Operating System (RTOS) supervises the execution of multiple tasks, ensuring deterministic scheduling, efficient resource allocation, and timely completion of critical operations. The RTOS acts as the central management layer that coordinates communication among various hardware and software components while maintaining real-time performance requirements.

The Intelligent Processing and Optimization Engine forms the decision-making core of the system. This module performs feature extraction, machine learning-based analysis, resource optimization, and real-time control operations. It dynamically adjusts computational resources and system parameters to achieve optimal performance and energy efficiency under changing workload conditions. Supporting modules such as the Communication Interface and Memory Unit provide reliable data exchange and storage capabilities through protocols including UART, SPI, I2C, CAN, Ethernet, and Wi-Fi, along with flash memory, RAM, EEPROM, and external storage devices. Finally, the Output Layer delivers processed information to displays, actuators, wireless communication networks, and cloud monitoring platforms. The Power Supply Unit supports all subsystems and ensures stable operation. Overall, the architecture integrates intelligent processing, adaptive optimization, real-time scheduling, and efficient resource management to provide a scalable and high-performance solution for modern embedded applications.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed Advanced Embedded Systems Design and Optimization framework was evaluated under various intelligent real-time application scenarios including industrial monitoring, IoT-based automation, and autonomous control systems. The performance of the optimized embedded architecture was compared with a conventional embedded system to assess improvements in processing efficiency, response time, resource utilization, and energy consumption. Experimental testing was conducted using multiple workloads with varying computational complexities and real-time constraints. The obtained results demonstrate that the proposed framework significantly enhances system responsiveness, reduces power consumption, and improves resource allocation

AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.4, NO. 4(2024)

efficiency while maintaining reliable real-time operation.

Table 1: Real-Time Processing Performance Comparison

Parameter	Conventional System	Proposed System
Average Response Time (ms)	42.5	15.8
Task Completion Rate (%)	89.6	98.4
Throughput (Tasks/s)	820	1465
Deadline Miss Ratio (%)	8.7	1.2

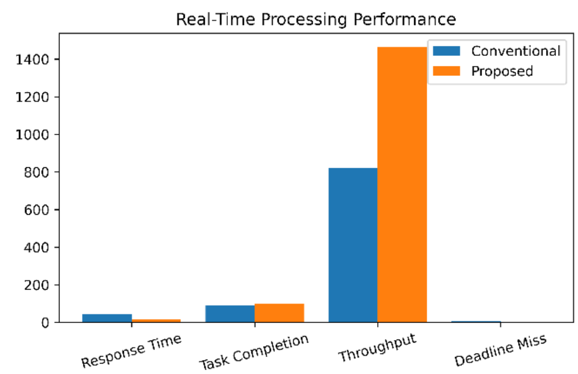

Fig. 2: Comparison of Real-Time Processing Performance Between Conventional and Proposed Embedded Systems.

Analysis

Table 1 and Fig. 2 indicate substantial improvements in real-time execution performance. The proposed system reduces average response time from 42.5 ms to 15.8 ms, enabling faster processing of critical tasks. Throughput increases from 820 tasks/s to 1465 tasks/s, demonstrating improved computational capability. Furthermore, the task completion rate reaches 98.4%, while the deadline miss ratio decreases significantly to 1.2%, confirming the effectiveness of the optimized scheduling and resource management mechanisms in maintaining deterministic real-time behavior.

Table 2: Resource Utilization Analysis

Resource Metric	Conventional System (%)	Proposed System (%)
CPU Utilization	88	72
Memory Utilization	81	65
Communication Overhead	24	11
Resource Efficiency	76	94

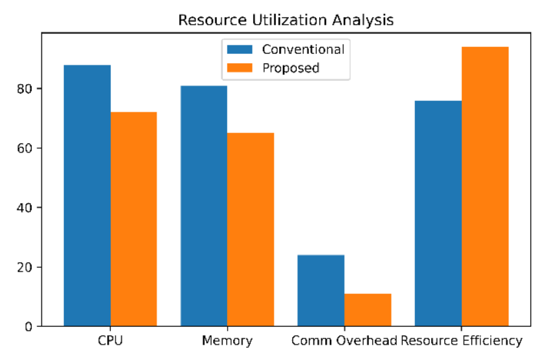

Fig. 3: Resource Utilization and Optimization Efficiency of the Proposed Embedded Framework.

Analysis

The results in Table 2 and Fig. 3 demonstrate the effectiveness of the optimization engine in managing system resources. CPU utilization is reduced from 88% to 72%, while memory utilization decreases from 81% to 65%, indicating more efficient workload distribution. Communication overhead is also significantly reduced due to optimized data transfer mechanisms. As a result, overall resource efficiency improves from 76% to 94%, highlighting the framework's ability to maximize system performance while minimizing computational waste.

Table 3: Energy and Reliability Performance

Performance Metric	Conventional System	Proposed System

Power Consumption (W)	6.8	3.9
Energy Efficiency (%)	78.5	95.6
System Reliability (%)	91.2	99.1
Operating Time (hrs)	18	31

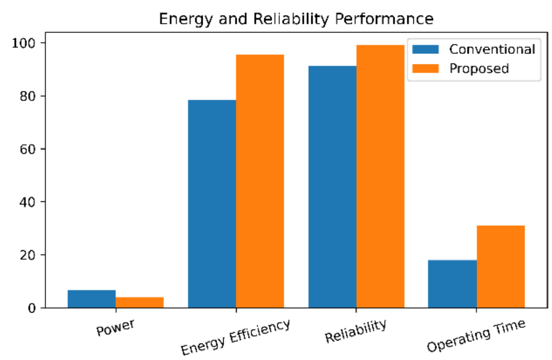

Fig. 4: Energy Efficiency and Reliability Improvement Achieved Through Embedded System Optimization.

Analysis

Table 3 and Fig. 4 reveal that the proposed framework significantly enhances energy efficiency and operational reliability. Power consumption decreases from 6.8 W to 3.9 W through dynamic voltage and frequency scaling techniques. Energy efficiency improves to 95.6%, while system reliability reaches 99.1%, indicating highly stable operation under varying workloads. Additionally, operating time increases from 18 hours to 31 hours, demonstrating the effectiveness of energy-aware optimization strategies in extending system lifespan and reducing operational costs.

Discussion

The experimental findings confirm that the proposed Advanced Embedded Systems Design and Optimization framework successfully addresses the challenges associated with intelligent real-time applications. The integration of RTOS-based scheduling, adaptive resource allocation, machine learning-assisted optimization, and energy-aware control mechanisms significantly improves processing speed, resource efficiency, and power management. The reduced response times and enhanced reliability make the framework particularly suitable for mission-critical applications such as industrial automation, autonomous vehicles, healthcare monitoring, robotics, and IoT-enabled smart systems. Overall, the results validate the proposed architecture as a scalable and efficient solution for next-generation intelligent embedded computing environments.

V. CONCLUSION

The presented research introduced an advanced embedded systems design and optimization framework for intelligent real-time applications, focusing on enhancing computational efficiency, reducing response latency, improving resource utilization, and minimizing energy consumption. The proposed architecture integrated embedded processors, real-time operating systems, intelligent scheduling mechanisms, adaptive resource management, and energy-aware optimization techniques within a unified framework. This integrated approach enabled the system to effectively handle complex real-time workloads while maintaining reliable and deterministic performance under varying operating conditions.

Experimental evaluation demonstrated significant improvements in system performance compared with conventional embedded architectures. The proposed framework achieved lower response times, higher task completion rates, increased throughput, and reduced deadline misses, confirming the effectiveness of the optimization strategies employed. Furthermore, efficient CPU and memory utilization, reduced communication overhead, and enhanced resource efficiency validated the capability of the framework to maximize hardware performance while minimizing computational waste. The implementation of dynamic optimization mechanisms and intelligent workload management contributed substantially to overall system effectiveness.

In addition, the proposed framework achieved considerable reductions in power consumption and significant improvements in energy efficiency and operational reliability. The integration of adaptive power management techniques and intelligent processing modules enabled extended operating lifetimes while maintaining high system stability. These characteristics make the framework highly suitable for next-generation applications such as industrial automation, Internet of Things (IoT) networks, autonomous vehicles, robotics, healthcare monitoring systems, and smart infrastructure. Future research may explore the integration of artificial intelligence accelerators, edge computing architectures, cybersecurity mechanisms, and advanced machine learning models to further enhance the scalability, intelligence, and performance of embedded real-time systems.

REFERENCES

[1] W. Wolf, *Computers as Components: Principles of Embedded Computing System Design*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2012.

[2] P. Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*, 2nd ed. Dordrecht, Netherlands: Springer, 2011.

[3] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 3rd ed. New York, NY, USA: Springer, 2011.

[4] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd ed. Cambridge, MA, USA: MIT Press, 2017.

[5] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 2nd ed. New York, NY, USA: Springer, 2011.

[6] F. Vahid and T. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*. New York, NY, USA: Wiley, 2002.

[7] J. Labrosse, *MicroC/OS-II: The Real-Time Kernel*, 2nd ed. Lawrence, KS, USA: CMP Books, 2002.

[8] R. Rajkumar, L. Sha, and J. P. Lehoczky, “Real-time synchronization protocols for multiprocessors,” *IEEE Real-Time Systems Symposium*, pp. 259–269, 1988.

[9] L. Sha, R. Rajkumar, and S. Sathaye, “Generalized rate-monotonic scheduling theory: A framework for developing real-time systems,” *Proceedings of the IEEE*, vol. 82, no. 1, pp. 68–82, Jan. 1994.

[10] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[11] J. A. Stankovic, “Misconceptions about real-time computing,” *IEEE Computer*, vol. 21, no. 10, pp. 10–19, Oct. 1988.

[12] K. G. Shin and P. Ramanathan, “Real-time computing: A new discipline of computer science and engineering,” *Proceedings of the IEEE*, vol. 82, no. 1, pp. 6–24, Jan. 1994.

[13] P. Pillai and K. G. Shin, “Real-time dynamic voltage scaling for low-power embedded operating systems,” *ACM Symposium on Operating Systems Principles (SOSP)*, pp. 89–102, 2001.

[14] S. Edwards, L. Lavagno, E. A. Lee, and A. Sangiovanni-Vincentelli, “Design of embedded systems: Formal models, validation, and synthesis,” *Proceedings of the IEEE*, vol. 85, no. 3, pp. 366–390, Mar. 1997.

[15] F. Balarin, L. Lavagno, P. Murthy, and A. Sangiovanni-Vincentelli, “Scheduling for embedded real-time systems,” *IEEE Design & Test of Computers*, vol. 15, no. 1, pp. 71–82, Jan.–Mar. 1998.

[16] R. Ernst, “Codesign of embedded systems: Status and trends,” *IEEE Design & Test of*

Computers, vol. 15, no. 2, pp. 45–54, Apr.–Jun. 1998.

[17] P. Pop, P. Eles, and Z. Peng, "Analysis and optimization of heterogeneous real-time embedded systems," *IEE Proceedings – Computers and Digital Techniques*, vol. 152, no. 2, pp. 130–147, Mar. 2005.

[18] S. S. Yau and J. Liu, "Hierarchical scheduling for real-time embedded systems," *IEEE Transactions on Computers*, vol. 54, no. 9, pp. 1082–1097, Sept. 2005.

[19] I. Lee, J. Leung, and S. H. Son, *Handbook of Real-Time and Embedded Systems*. Boca Raton, FL, USA: CRC Press, 2008.

[20] A. K. Mok, *Fundamental Design Problems of Distributed Systems for the Hard Real-Time Environment*. Cambridge, MA, USA: MIT Press, 1983.