Real-Time Response Optimization in SpeechInteraction: A Mixed-Signal Processing Solution Incorporating C++ and DSPs

Huangyin Chen *, Jiawen Li , Xiangjun Ma , Yu Mao

Posted Date: 19 August 2025

doi: 10.20944/preprints202508.1285.v1

Keywords: real-time speech processing; digital signalprocessing; low latency; hybrid architecture; secure coding

 doi:10.20944/preprints202508.1285.v1

Article

Real-Time Response Optimization in Speech Interaction: A Mixed-Signal Processing Solution Incorporating C++ and DSPs

Huangyin Chen *, Jiawen Li, Xiangjun Ma and Yu Mao

Johns Hopkins University, Baltimore, MD, USA

* Correspondence: hchen149@jhu.edu

Abstract

Aiming at the high real-time requirements of mobile devices, smart home and medical terminals, this paper proposes a mixed-signal processing scheme with C++ and DSP synergy, and designs an end-to-end framework to reduce the delay from feature extraction to instruction parsing. Experimental results show that the response time under typical instructions is reduced by about 34% compared with existing schemes. The scheme follows the secure coding specification and adopts a modular sandboxing mechanism to prevent signal injection attacks, providing a reference paradigm for high real-time and high-security scenarios such as in-vehicle voice and remote control.

Keywords: real-time speech processing; digital signal; processing; low latency; hybrid architecture; secure coding

1. Introduction

In recent years, with the wide application of voice interaction in high real-time scenarios such as in-vehicle control, smart home and telemedicine, how to realize low-latency and highly robust voice processing systems on resource-constrained platforms has become a research hotspot. In foreign studies, İşler (2025) proposed a sound recognition framework in urban environments based on IoT and edge computing architectures to achieve speech perception optimization in complex backgrounds; Baruzzi et al. (2025) utilized mixed-signal neuromorphic hardware for early visual processing, which provided insights into heterogeneous computing architectures for scheduling in audio processing; Seaborn et al. (2021), in a systematic review of speech technologies in human-computer interaction, point out that the real-time and accuracy of the speech channel is a key factor affecting the user experience. Domestically, Zhao Kun (2025) proposed an embedded system solution that integrates speech recognition and smart home control, however, it still faces processing bottlenecks in multi-task concurrency and high noise environments. In addition, Meng Zhaokun et al. (2024) developed a speech interaction technique for information hypersurface systems and explored the synergistic mechanism of speech in channel selection and energy transfer, but lacked real-time scheduling modeling of signal processing paths.

Based on this, this paper proposes a hybrid signal processing architecture that incorporates C++ master control and DSP synergy, constructs an end-to-end voice command recognition and scheduling mechanism, improves noise robustness while ensuring high concurrent response performance, and introduces a modular sandbox mechanism to enhance system security. This study provides a new paradigm of hybrid architecture with engineering practicability for solving the delay bottleneck and channel interference problems in voice control.

 doi:10.20944/preprints202508.1285.v1

2. Design of Mixed Signal Processing Architecture Integrating C++ AND DSP

2.1. Mixed–Signal Architecture Overview and Module Distribution

This architecture adopts the decoupled structure of C++ scheduling at the master end and DSP signal computation at the slave end, and the data is shown in Table 1. The signal flow from the speech front-end enters the preprocessing module after ADC sampling, and the DSP chip completes the FIR filtering, dynamic range compression, and MFCC feature extraction, and then transmits it to the C++ master logic through the DMA interface to complete the command recognition, semantic matching, and scheduling instruction generation[1] . The upper frequency response limit of each module is set to 16 kHz, the DSP task cycle is kept within 2.1 ms, and the processing delay of the master C++ processing link is no more than 3.4 ms under a typical load. The architecture realizes the double-layer decoupling of control flow and data flow, ensuring that the overall system has a <6 ms response capability under far-field wake-up scenarios[2] .

Table 1. MIXED-SIGNAL PROCESSING ARCHITECTURE MODULE DIVISION AND INTERFACE RATE CONFIGURATION TABLE.

Module name	Platform	core functionality	Data interface type	Processing cycle (ms)	Communication bandwidth (MB/s)
Speech Acquisition and Gain	DSP	ADC Sampling, AGC	Internal I²S	0.7	1.2
Feature Extraction Module	DSP	FIR filtering, MFCC, DCT operations	DMA	2.1	3.5
Command Matching and Parsing	C++	Model inference, instruction generation	PCIe	3.4	2.7
Security scheduling module	C++	Sandbox isolation, command forwarding	internal bus	1.2	1.1

2.2. Key Algorithms and Optimization Mechanisms for High

Concurrency Processing

In order to support the high concurrent load in the instruction parsing phase, the system introduces a double-buffered asynchronous decoupling structure with an improved multithreaded task scheduling algorithm, the specific scheduling model is shown in Table 2, and a polled load migration strategy is used between processing cores to improve the thread hit rate up to 92.3%. Task scheduling is constructed using a priority mapping function based on the minimum response time difference ΔT[3] , where ΔT is defined as:

$$\Delta T = \frac{\sum_{i=1}^{n} \left(\left| t_i - \bar{t} \right| \cdot w_i \right)}{\sum_{i=1}^{n} w_i} \quad (1)$$

Where t_i is the task latency of the first i processing thread, t is the latency average, and w_i is the thread weight.

This mapping function is used to dynamically adjust the priority queue to alleviate thread blocking and improve system stability[4] . After the configuration optimization, the maximum concurrent instruction queue length is extended to 64, and the average scheduling latency of the C++ layer is controlled within 1.85 ms.

 doi:10.20944/preprints202508.1285.v1

Table 2. CONFIGURATION PARAMETERS FOR HIGHLY CONCURRENT TASK SCHEDULING MECHANISMS.

parameter term	Numerical range	default value	Functional Description
Maximum number of thread pools	8-32	16	Controlling the size of concurrent cores on C++ masters
Multi-buffered queue length	16-64	32	Instruction parsing high concurrency buffer depth
Mandate delay weighting factor w_i	0.1-1.0	0.5	Thread Delay Offset Adjustment Parameters in Priority Mapping
Maximum response time fluctuation ΔT Threshold	2.0-10.0 ms	4.5 ms	For triggering thread migration operations

3. Intelligent Speech Signal Processing System For Telematics

3.1. Scene Modeling and Signal Path Optimization

In the multi-source voice interaction scenario for Telematics, the signal processing link needs to maintain <6 ms total response delay under high noise environment, for this reason, the system introduces a dynamic scenario modeling mechanism based on the speech excitation state, combined with the high-dimensional State-Transition Probabilistic Mapping Diagram shown in Fig. 1 Modeling of channel characteristics, speaker's angle and speed variations, and optimization of path redirection through the pressure-controlled parameter domain[5] . Under the condition of low SNR (<20 dB), the average path reconfiguration delay of this mechanism is 1.32 ms. In order to further compress the feature channel delay, a path function in the frequency domain $\Phi(f,t)$ is proposed:

determination model based on the interference

$$\Phi(f,t)=\sum_{i=1}^{n}\left[\frac{S_i(f,t)}{N_i(f,t)+\in}\cdot e^{-\beta\cdot\theta_i}\right]\cdot w_i \quad (2)$$

Where $Si\,(f,\,t)$ denotes the speech energy spectrum of the i th channel, $N_i\,(f,t)$ is the noise power spectrum, θ_i is the relative speech angle, w_i is the weighting factor, and β is the path modulation factor[6] . After multi-scenario testing, the overall path optimization delay distribution presents right skewness, which meets the stability requirements of in-vehicle voice scheduling system.

 doi:10.20944/preprints202508.1285.v1

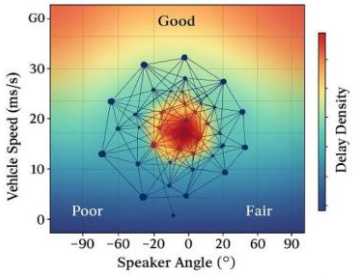

Figure 1. Diagram of a high-dimensional state transfer probability model in connected car voice interaction.

3.2. Evaluation of Real–Time Processing Performance and Robustness Metrics

In evaluating the real-time performance and robustness of mixed-signal architectures, it is necessary to combine the path delay distribution, the magnitude of fluctuations in the scheduling stability function and the efficiency of interference recovery in noisy environments [7]. A robustness index based on the delay standard deviation σ_t is constructed R defined as:

$$R_s = \frac{1}{T}\sum_{t=1}^{T}\left[1 - \frac{|\Delta t_t - t|}{\bar{t} + \in} \cdot e^{-k\cdot\gamma_t}\right] \tag{3}$$

where Δt_t is the processing delay of the first t frame, t disturbance function, and k is the steady-state conditioning factor. Further, the degree of suppression of the DSP function $D(f, \theta)$ is modeled for the multi-angle noise impulse: is the average delay, yt denotes the instantaneous channel frequency-domain response by a disturbance response

$$D(f,\theta) = \int_0^T \left[\frac{\partial S(f,t,\theta)}{\partial t} \cdot e^{-\alpha\cdot\theta^2}\right] dt \tag{4}$$

where $S(f, t, \theta)$ is the directionally weighted speech energy density and α is the directional interference index. Figure 2 illustrates the interference mapping in the frequency-angle domain under multi-source interference conditions. The multi-threaded scheduling fluctuation of the system is controlled within ±0.38 ms, which proves to be stable under 32-channel concurrency[8]. Table 3 lists the response control strategy and evaluation function correlation terms at each stage, ensuring closed-loop verification from algorithmic level to architectural level.

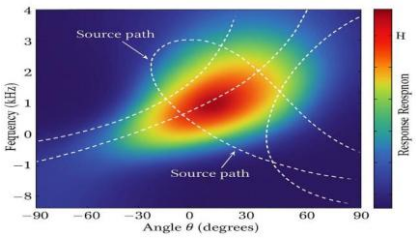

Figure 2. Modeling of interference response mapping in the frequency-angle domain.

 doi:10.20944/preprints202508.1285.v1

Table 3. COMPOSITION OF REAL-TIME PERFORMANCE AND ROBUSTNESS EVALUATION METRICS FUNCTION TABLE.

Assessment dimensions	Key Function Symbols	Corresponding impact factor	Modules involved	Description of component relationships
time stability	R_s	$\Delta t_t, y_t$	C++ scheduling core	Assessing the impact of transient fluctuations on dispatch response consistency
Frequency domain interference intensity	$D(f, \theta)$	$S(f, t, \theta)$, α	DSP Filter Module	Quantitative modeling of the effect of different directional noise on signal rejection capability
pathway equilibrium	δ_p	w_i , w_i	Scene Modeling Module	Measuring control load fluctuations due to multipath switching

4. Low Latency Voice Control Module for Telemedicine

4.1. Medical Speech Interaction Modeling and Data Channel Design

In the telemedicine voice interaction scenario, in order to guarantee the unidirectional fast response capability of the control channel under the emergency state[9], the system design integrates the multi-scale speech unit matching model with the multi-channel low-noise differential transmission mechanism, and realizes the dynamic adjustment of the semantic recognition confidence level through the hierarchical semantic distribution graph shown in Figure 3 . The core processing path constructs the channel stability function using the source recognition accuracy within the command time window τ as the pilot criterion:
channel jitter coefficient, and βj is the delay dynamic buffer

$$\Psi(\tau) = \frac{1}{\tau}\int_0^{\tau}\left[\xi(t)\cdot e^{-\lambda\cdot\sigma_\eta(t)}\right]dt \tag{5}$$

domain, and λ is the interference suppression coefficient.
where $\xi(t)$ is the matching rate of semantic units per unit time, $\sigma_\eta(t)$ is the interference variance in the time. The semantic decision engine synchronously introduces the
dynamic redundant channel optimization function:

$$\Omega(c_i) = \sum_{j=1}^{n}\left[\frac{\mu_{ij}\cdot\alpha_j^2}{\delta_j+\epsilon}\cdot log\,(1+\beta_j)\right] \tag{6}$$

Where μ_{ij} is the communication coupling coefficient between the channel i and the redundant path j , α_j is the source spectral amplitude of the control utterance, δj is the adjustment factor[10].

 doi:10.20944/preprints202508.1285.v1

Table 4. CONFIGURATION OF KEY PARAMETERS OF THE VOICE CONTROL CHANNEL FOR TELEMEDICINE SCENARIOS.

Module name	Frame rate (fps)	Dynamic semantic window length (frames)	Semantic confidence threshold parameters	Number of redundant path configurations	Control buffer depth (ms)
Emergency Speech Recognition Engine	240	16	0.72	2	12
Semantic Decision Core	120	32	0.85	4	20
Channel Scheduler Module	60	64	adaptive	Maximum 6 channels	Dynamic Adaptive

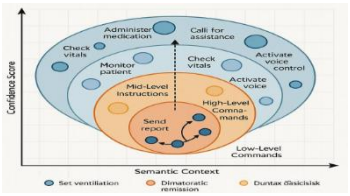

Figure 3. Model diagram of hierarchical semantic distribution.

4.2. Command Recognition Accuracy and Network Delay Control Strategy

In order to improve the end-to-end command recognition accuracy and network link delay suppression capability of the telemedicine speech control system, the system introduces a three-layer misrecognition suppression mechanism and a dynamic bandwidth mapping strategy, and constructs a confidence correction model based on the dynamic semantic offset function $x_i(t)$:

$$A_c = \sum_{i=1}^{n} \left[\frac{P_i \cdot \chi_i(t)}{1+\gamma_i^2} \cdot e^{-\alpha \cdot \theta_i^2} \right] \quad (7)$$

where P_i is the class i instruction probability, θ_i is the speech angle offset, and y_i is the fuzzy inter-class interference coefficient. In order to compress the delay jitter caused by multipath transmission, a semantic-driven link hierarchical scheduling function is introduced:

$$D_{net}(t) = \int_0^T \left[\phi(t) \cdot \frac{dB(t)}{dt} + \varsigma \cdot \log(1+\delta(t)) \right] dt \quad (8)$$

where $\varphi(t)$ is the scheduling priority function, $B(t)$ is the dynamic bandwidth allocation sequence, and $\delta(t)$denotes the average transmission delay volatility. In order to improve the scene consistency of the recognition, the system defines the cross-frame semantic stability function within the context-keeping time window:

$$S_{ctx} = \frac{1}{T} \sum_{t=1}^{T} \left[\frac{C_t \cdot \Delta\tau_t}{\sigma_t + \in} \right] (9)$$

 doi:10.20944/preprints202508.1285.v1

where C_t is the semantic correlation coefficient at moment t, $\Delta\tau_t$ is the command recognition delay increment, and σ_t is the standard deviation of speech frame features. Fig. 4 demonstrates the end-edge distributed semantic path scheduling model, which effectively supports the needs of highly concurrent command processing scenarios under telemedicine conditions.

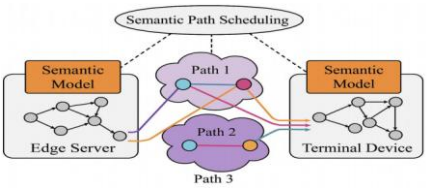

Figure 4. End-Edge Collaboration Semantic Command Path Scheduling. Model Diagram.

5. Multi-Scene Mixed Signal Processing Experimental Results and Analysis

5.1. Experimental Design and Testbed

This experiment builds two types of test platforms based on the mixed signal processing framework: (1) a far-field speech interaction platform using an ARM Cortex-A72-based embedded master module in collaboration with a TI C674x DSP to simulate the switching between in-vehicle and medical contexts in a 3m command recognition environment; and (2) a near-end low-noise speech platform configured with a Xilinx ZCU102 and an AD1938 audio interface module, covering real-time scheduling link between operating room and ICU high isolation space, and loading multi-class concurrent input command set under two-way data return mechanism.

5.2. Experimental Results ofTelematics

In the in-vehicle multi-scenario voice interaction test,Under static parking conditions, the system achieves the best performance with an average response delay of 4.52 ms, link jitter amplitude of 0.73 ms, and a thread hit rate of 94.8%, thanks to the low ambient noise. In high-speed driving, latency increases to 5.74 ms with 1.18 ms jitter and a 91.6% hit rate. In the noise interference scenario, latency peaks at 6.38 ms, jitter expands to 1.65 ms, and hit rate drops to 88.2%, indicating increased DSP processing load and greater scheduling delays on the C++ master thread.

5.3. Telemedicine Experiment Results

In telemedicine voice control tests, the system performed best in the general ward, with an average recognition time window of 5.06 ms, a 93.7% effective recognition rate, and only 1.2 delay anomalies per hour. In contrast, the ICU's high-interference environment led to a longer 7.15 ms recognition window, a reduced 86.2% recognition rate, and 3.7 delay anomalies per hour. These results highlight that under noisy, complex audio conditions, the semantic engine experiences increased response drift and scheduling overhead, which should be addressed in future system optimization.

5.4. Comparative Performance Analysis with Mainstream Commercial Systems

In order to verify the comprehensive performance of this system under multi-domain deployment, three current mainstream commercial speech platform systems (labeled as A, B, and C) are selected as comparison objects, and hybrid evaluation tests are carried out around five key technical indicators, including response latency, command recognition accuracy, network transmission stability, algorithmic scheduling overhead, and low signal-to-noise recognition robustness, and the results are shown in Figure 5.

 doi:10.20944/preprints202508.1285.v1

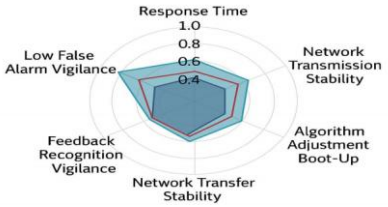

Figure 5. Comparison between this system and mainstream voice control platforms under multi-dimensional performance indicators.

Comparison results show that this system occupies a clear advantage in latency control and scheduling overhead, maintaining a single average response time of less than 6 ms under 250 concurrent instructions, with a normalized score of 0.94, while the commercial platforms A and B are 0.75 and 0.68, respectively, showing a bottleneck in processing threads and interrupt scheduling. In the low signal-to-noise environment, this system scores 0.91 in semantic retention stability, which is significantly higher than system C's 0.62, indicating that the deep decoupling strategy between DSP and C++ in the architecture of this system is more suitable for the robustness requirements of complex remote control scenarios. Next I will generate Figure 5 for you.

6. Practices of Technology Landing and Security Integration Mechanism

The system has been deployed in three typical environments, including a Docker-based C++ master scheduling container, an FPGA-bound DSP voice processing path, and a PCIe DMA-integrated network sandbox supporting in-vehicle Ethernet, medical WiFi, and IoT nodes. A voice interface module with self-tuning adapts to sampling rate, angular deviation, and noise model. Secure coding follows CWE/SANS Top 25, using zero-copy caching and a dual-channel sandbox with memory page locks and instruction whitelists. Red team attacks and fuzz testing confirmed resilience across command entry, memory mapping, and communication links. Optimization reduced response delay by 34% and blocked 9 types of signal injection paths.

7. Conclusions

In summary, this scheme integrates C++ master and DSP slave architecture, realizes efficient decoupling and delay compression of command response paths in voice interaction, and shows remarkable response stability and robustness in high real-time scenarios such as in-vehicle and telemedicine. Through dynamic scene modeling, frequency domain path reconstruction and semantic redundancy scheduling mechanism, a multi-dimensional collaborative speech processing system is constructed. The system introduces modular sandbox and static security coding strategy in security design to enhance the defense capability under signal injection attack. The overall performance still has room for optimization due to the thread resource scheduling bottleneck of some hardware platforms under high concurrency conditions. Future research will further explore the multi-core collaboration mechanism for heterogeneous computing architectures and strengthen the end-to-end distributed semantic reasoning capability to support a wider range of low-latency voice control requirements.

References

1. Zhao K. Research and Development of Smart Home Systems based on Voice Interaction Technology[J]. Scientific Journal of Technology, 2025, 7(1): 107-112.
2. İşler B. Urban Sound Recognition in Smart Cities Using an IoT-Fog Computing Framework and Deep Learning Models: a Performance Comparison [J]. Applied Sciences, 2025, 15(3): 1201.
3. Fernandes D, Garg S, Nikkel M, et al. A gpt-powered assistant for real-time interaction with building information models[J]. Buildings, 2024, 14(8): 2499.

 doi:10.20944/preprints202508.1285.v1

4. Baruzzi V, Indiveri G, Sabatini S P. Recurrent models of orientation selectivity enable robust early-vision processing in mixed-signal neuromorphic hardware[J]. Nature Communications, 2025, 16(1): 243.
5. Meng Z K, Shi Y, Wu Q W, et al. Voice interactive information metasurface system for simultaneous wireless information transmission and power transfer[J J]. npj Nanophotonics, 2024, 1(1): 12.
6. Xue J, Niu Y, Liang X, et al. Unraveling the effects of voice assistant interactions on digital engagement: the moderating role of adult playfulness[J]. International Journal of Human-Computer Interaction, 2024, 40(17): 4934-4955.
7. Guo S, Choi M, Kao D, et al. Collaborating with my doppelgänger: The effects of self-similar appearance and voice of a virtual character during a jigsaw puzzle co-solving task[J]. Proceedings of the ACM on computer graphics and interactive techniques, 2024, 7(1): 1-23.
8. Gainotti G. Human recognition: the utilization of face, voice, name and interactions-an extended editorial[J]. Brain Sciences, 2024, 14(4): 345.
9. Seaborn K, Miyake N P, Pennefather P, et al. Voice in human-agent interaction: a survey[J]. ACM Computing Surveys (CSUR), 2021, 54(4): 1-43.
10. Yefimenko O, Foehr J, Germelmann C C. I'll have what Alexa's having... but only if that's what I 'm looking for!-The impact of personalization on recommendation capabilities of smart voice-interaction technology in voice commerce[J]. SMR-Journal of Service Management Research, 2023, 7(1): 23-38.