IJHSR
ijhighschoolresearch.org
RESEARCH ARTICLE

A Lightweight Decision Model Design for FPGA-based High-Concurrency Spacecraft Testing Systems

Yuqian Han

Jinling High School Hexi Campus, Nanjing, 210019, China; hanyuqian_2024@qq.com

ABSTRACT: To address the high-concurrency challenges faced by spacecraft testing systems, a widely adopted and effective solution is to migrate the core task module of the testing system, the real-time decision subsystem, from the CPU to a high-concurrency FPGA. However, due to the limited chip area and the need to support programmability, FPGA hardware resources are inherently constrained. To address these resource limitations, we propose a decision model construction method that integrates hierarchical clustering and envelope interval computation. This method is further enhanced by iterative optimization through independent classification and dynamic confidence level adjustment, effectively minimizing errors introduced by the regression model. Additionally, we introduce a regression analysis approach to approximate envelope interval judgment rules with functional expressions, replacing discrete rule sets to achieve significant model storage reduction while maintaining decision accuracy and efficiency. Validation experiments comprehensively verify the accuracy and reliability of the method, demonstrating a substantial reduction in storage footprint (from 500MB to 1.47MB), effectively alleviating the restrictions imposed by FPGA chip storage resources.

KEYWORDS: Systems Software, Algorithms, Spacecraft Testing Systems, Decision Model, High-concurrency FPGA.

Introduction

The spacecraft testing, launch, and control system constitutes the fundamental cornerstone of terrestrial infrastructure for space missions, performing critical operational functions that include spacecraft status monitoring, collaborative multi-system debugging, and launch process coordination.[1] Within this framework, the testing subsystem operates as a critical module designed to perform comprehensive parameter detection across multiple domains. This process entails the acquisition and analytical processing of a diverse array of signal types, including digital signals, analog signals (voltage/current), frequency signals, and pulse signals.[2,3] Furthermore, the spacecraft testing system is responsible for implementing a series of critical testing protocols to validate the integrity of the spacecraft. These protocols include evaluations of power supply characteristics, including measurements of grounding resistance and insulation impedance, assessments of static operating points, analyses of dynamic conditions, and verifications of environmental adaptability, such as electromagnetic compatibility tests and vibration testing procedures.[4,5] Such rigorous testing is essential to ensure the reliability and optimal performance of the spacecraft prior to its launch.

However, driven by the exponential growth in the complexity of electronic systems (as articulated by Moore's law),[6] the demands for real-time testing of contemporary spacecraft have increased beyond 2000 electronic channels, with data throughput achieving gigabit-per-second (Gb/s) levels. This presents a formidable challenge to the high-concurrency processing capacity of the testing system, which requires its ability to handle parallel data stream processing in more than 2000 channels.[7,8]

Currently, spacecraft testing systems are primarily reliant on CPU architectures. However, due to inherent architectural constraints, their performance is progressively insufficient to address the requirements mentioned above. CPUs are structured around the Single Instruction Stream, single data stream (SISD) paradigm, which demonstrates limited instruction-level parallelism (ILP), thus significantly impeding their capacity for multitasking parallel processing.[4] Furthermore, the memory wall problem in CPUs is particularly severe.[9] The shared bus architecture of the CPU results in memory access latency accounting for over 60% of the total processing time,[10] thus significantly impairing overall system efficiency. Additionally, CPUs are subject to notable real-time limitations. Under a polling scheduling mechanism,[11] the response time for 2000 channels can extend to several seconds, thus failing to meet the high-speed real-time data processing requirements of the testing subsystem. To address the challenges encountered by spacecraft testing systems, a widely implemented and efficient approach is to transition the core task module, namely the real-time decision-making subsystem, from the CPU to a high-concurrency FPGA.[12] Using its superior parallel processing capabilities, the FPGA can perform multiple independent operations concurrently, effectively meeting the high-concurrency requirements of spacecraft testing systems. However, due to the limitations imposed by the area of the chip, its storage memory is severely restricted (typically less than 5MB),[13] making it challenging for the FPGA chip to store the data of the decision subsystem. This results in extensive off-chip communication, thereby degrading overall performance.

To address this limitation, we have devised a decision model construction methodology that combines hierarchical clustering with envelope interval computation, further refined through iterative optimization via independent classification and dynamic confidence level adjustment, and incorporates a

DOI: 10.36838/IJHSR84.88

regression analysis approach to reduce the size of the decision model, thus effectively minimizing regression model errors and reducing storage requirements. The main contributions of this paper are as follows.

(1) We proposed a decision model construction method to provide a lightweight decision model for FPGA-based spacecraft testing systems, which integrates hierarchical clustering and envelope interval computation, further enhanced through iterative optimization by independent classification and dynamic confidence level adjustment to minimize regression model errors.

(2) We introduced a regression analysis approach to approximate envelope interval judgment rules with functional expressions, replacing discrete rule sets. This achieves a further reduction in the storage of the lightweight decision model while preserving the accuracy and efficiency of the decision.

(3) We designed validation experiments to evaluate the proposed method. The experiments confirmed that the proposed method significantly reduces the storage requirements of the decision model from 500MB to 1.47MB without compromising accuracy, effectively addressing FPGA on-chip storage limitations.

Methods

Motivation:

A comprehensive spacecraft testing system is typically composed of core modules, including data acquisition, signal processing, data analysis and decision making, and communication interfaces, as shown in **Figure 1**. Within this architectural framework, the front-end subsystem is primarily tasked with data acquisition, whereas the real-time decision subsystem assumes the critical roles of processing input signals, parsing interface protocols, and performing data analysis. Subsequently, the data decision module performs a target judgment based on the analytical results and generates the corresponding decision outcomes.

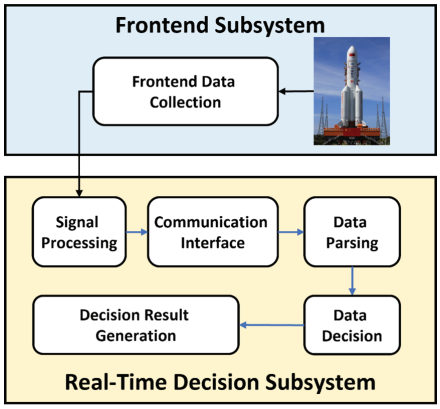

Figure 1: Spacecraft Testing System Architecture outlines the workflow for data acquisition, processing, and decision-making in aircraft testing. The front-end subsystem handles data acquisition, while the real-time decision subsystem performs signal processing, protocol handling, and data parsing, followed by evaluation and result generation by the data judgment module. This architecture is extensively utilized in spacecraft testing and is of considerable research significance.

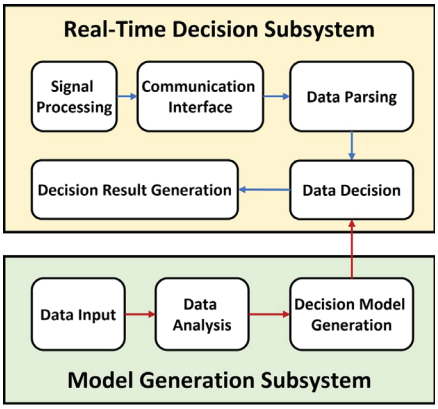

Figure 2: Model Generation and Decision Subsystems illustrates the interaction between the decision model and the real-time decision subsystem after integrating the model generation module. Built upon the existing testing system, the model generation subsystem focuses on modeling the decision module and generating decision models, thereby separating model generation from real-time decision-making processes.

The real-time decision subsystem, which serves as the central component of the testing system, also represents the main bottleneck of the system's concurrent processing capabilities. Currently, the data decision process within this subsystem is based primarily on an empirical threshold-based methodology. This approach involves establishing predefined threshold ranges for the transmission values of each electronic and electrical component at specific temporal points. By comparing the input signals with these threshold ranges, the system evaluates the normality of the data and evaluates the health status of the electronic and electrical components. The current data decision mechanism of the real-time decision subsystem relies on empirical threshold-based methods, which are not only inefficient but also require the collection of extensive data.

Historically, CPUs have served as the main hardware execution platforms for spacecraft testing systems. However, they have encountered substantial limitations in terms of concurrency capabilities, which have hindered their effectiveness in completing spacecraft testing tasks efficiently.[14,15] To address the performance bottlenecks encountered by CPUs in executing the complete testing system, the industry is actively exploring solutions to migrate the testing system to FPGA platforms. FPGAs, leveraging their ability to execute multiple independent hardware digital circuits in parallel, can effectively meet the demands of high-concurrency computing.[15-19] However, while enabling high-concurrency task processing, FPGAs also face substantial limitations in storage resources. Due to chip area constraints, independent digital circuits occupy most FPGA resources, leaving limited space for on-chip storage deployment. For example, the Xilinx Virtex high-end chip XC7VX690T.20 provides 6.615 MB of embedded RAM (52920 kbit) and 1.361 MB of distributed RAM (10888 kbit), totaling 7.976 MB of on-chip storage. These resources must support multiple functions, including data caching, register files, and First In First Out (FIFO) buffers. After accounting for system overheads such as data stream caching, data parsing, and cache buffers, less than 5 MB remains available for storing decision thresholds. However, the current spacecraft real-time

decision subsystem requires approximately 500MB for threshold storage, which significantly exceeds the available storage capacity of the FPGA. Consequently, the decision module must undergo model-based processing and simplification optimization to meet the deployment requirements on FPGA platforms.

Generation of the Decision Model:

To enhance the efficiency and accuracy of data decisions in the real-time decision subsystem, we introduce a model generation subsystem into the existing test system architecture to model the data decision method, a decision model. The collaborative relationship between the model generation subsystem and the real-time decision subsystem is illustrated in **Figure 2**. Considering the significant periodicity of spacecraft testing and control systems, where the testing process for specific spacecraft models is highly repetitive, we propose generating decision models by collecting sample data. These models can effectively guide the testing processes of similar spacecraft. Specifically, we first collect sample data, then use the model generation subsystem to construct the decision model, and finally deploy this model to the data judgment module of the real-time decision subsystem. This approach enables the real-time decision subsystem to perform an efficient and accurate real-time decision of input signals.

Table 1: Normal Data Sampling Segment presents a typical data segment after the cleaning process. In this data structure, the 'millisecond' field precisely records the sampling timestamp, while fields such as 'data1' and 'data2' correspond to the values of measured parameters of different test objects. The data cleansing process eliminated a substantial amount of redundant data.

Table I: Normal data sampling segment

Millisecond value	Data1	Data2
1.72283	5.01793	29.4375
1.72283	5.01793	29.4375
1.72283	5.01854	29.4375
1.72283	5.01854	29.4375
1.72283	5.01793	29.4375
1.72283	5.01793	29.4375
1.72283	5.01793	29.4375
1.72283	5.01793	29.4375
1.72283	5.01793	29.4375
1.72283	5.01793	29.4375
1.72283	5.01793	29.4375
1.72283	5.01732	29.46875
1.72283	5.01732	29.46875
1.72283	5.01793	29.46875
1.72283	5.01793	29.46875
1.72283	5.01854	29.46875

The decision model developed in this study is constructed based on a large-scale measured dataset. To ensure the effectiveness of the model, we conducted systematic data collection on approximately 1100 electronic and electrical objects. Depending on the characteristics of each object, sample data was collected between 1200 and 34000 times per object during the entire testing cycle. It should be noted that all sampled data constitute a noise-free dataset, indicating that all electronic and electrical objects were in standard operating conditions during data acquisition. The total size of the raw dataset is approximately 390 MB, which was reduced to 196 MB after a rigorous data cleaning process, including the removal of invalid data and the handling of outliers.

Based on the pre-processed normal dataset (shown in **Table 1**), the generation of the decision model follows a systematic process:

(1) Data Object Separation: Extract the complete data sequence of each independent sampled object from the overall dataset.

(2) Hierarchical Clustering Analysis: Apply hierarchical clustering based on temporal and numerical features to the time-series data of individual objects.

(3) Envelope Interval Calculation: Compute the envelope limits of the confidence interval 95\% using an improved statistical method based on the results of the clustering.

(4) Model Integration and Construction: Systematically integrate the envelope interval results of all objects to form the final decision model.

During this generation process, it is essential to perform continuous clustering on the data, which entails grouping a set of data points that are temporally continuous and fall within a predefined deviation threshold into the same category. This methodology is driven by the observation that spacecraft exhibit distinct electronic and electrical values in different operational states. From a data perspective, a state is characterized by fluctuations confined to a limited range over a continuous time interval. As a result, the results of continuous clustering correspond to the effective datasets of the spacecraft under various states in practical scenarios.

In the model generation process, clustering serves as the core step, and its quality directly determines the performance of the final model. Currently, the main clustering methods include k-means,[21] hierarchical clustering, and DBSCAN.[22,23] Among these, k-means complete clustering by iteratively adjusting cluster centers based on a predefined number of clusters; hierarchical clustering, as an unsupervised learning method, starts from each data point and groups points within a user-defined difference range into a cluster; while DBSCAN is more suitable for handling datasets containing noise. Given that this study is based on a noise-free dataset and does not require predefining the number of clusters, hierarchical clustering, which achieves clustering by controlling the difference boundary, becomes the most suitable choice. The implementation process of our proposed hierarchical clustering is as follows:

(1) Calculate the Euclidean distance between each pair of data points in the dataset based on the sampled data of each object; subsequently, define the clustering boundary α. The initial value of the clustering boundary α is set to $\mu + 1.5\sigma$ based on the mean (μ) and standard deviation (σ) of the Euclidean distances, and it is iteratively adjusted: α is increased if it splits data of the same state, and decreased if it merges data of different states.

(2) Initialize clusters, starting with the first point as the initial cluster.

(3) Apply the complete linkage method, where the distance between two clusters is defined as the farthest distance be-

DOI: 10.36838/IJHSR84.88

tween any two points in the clusters. If this distance is less than α, merge the clusters.

(4) Update the distance matrix and repeat the above cluster merging and distance matrix updating process.

(5) After clustering is completed, assign a cluster label to each data point.

Table 2: Data Segment for Envelopes presents the numerical results of the upper and lower envelopes corresponding to the time values; the "num" field denotes the number of sampled data points within each single envelope cluster. The envelope results can serve as the basis for judgment.

Table II. Data segment for envelopes

Data	Time	Lower_confidence_bound	Upper_confidence_bound	num
data1	1722829052481-1722830226888	4.993	4.994	8744
data2	1722829052481-1722829058477	28.327	28.469	50
data2	1722829058727-1722829068478	28.531	28.688	80
data2	1722829068728-1722829084480	28.781	28.969	128
data2	1722829084730-1722829087980	29.000	29.062	28
data2	1722829167879-1722829181881	30.250	30.438	114
data2	1722829182131-1722829197878	30.500	30.625	128
data2	1722829198128-1722829215880	30.719	30.875	144
data2	1722829216130-1722829233882	30.906	31.094	144
data2	1722829234132-1722829245878	31.094	31.219	96
data2	1722829246128-1722829259880	31.250	31.406	112
data2	1722829260130-1722829275881	31.438	31.625	128
data2	1722829276131-1722829289878	31.594	31.750	112
data2	1722829290128-1722829307880	31.781	31.938	144
data2	1722829308130-1722829333878	32.000	32.188	208
data2	1722829334128-1722829351880	32.219	32.375	144
data2	1722829352130-1722829371882	32.375	32.562	160
data2	1722829372132-1722829401881	32.594	32.781	240
data2	1722829402131-1722829423883	32.844	33.031	176
data2	1722829424133-1722829449881	33.031	33.219	208
data2	1722829450131-1722829465883	33.188	33.344	128
data2	1722829466133-1722829485880	33.344	33.531	160
data2	1722829486130-1722829503882	33.469	33.625	144
data2	1722829504132-1722829525880	33.656	33.812	176
data2	1722829526130-1722829547882	33.781	33.969	176
data2	1722829548132-1722829557883	33.938	34.094	80
data2	1722829558133-1722829581885	34.031	34.219	176

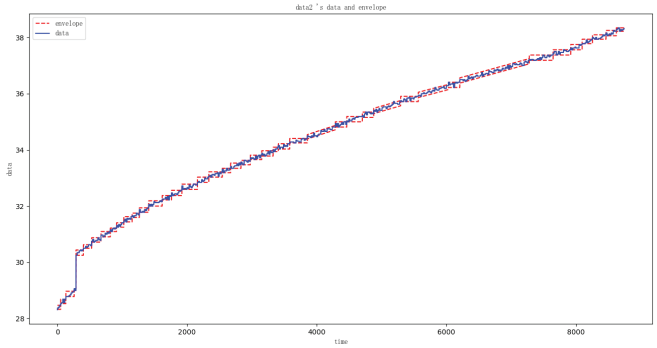

Figure 3: Envelope of Object 2 displays the upper and lower bounds as red lines, while the original sampled data is shown in blue. This envelope plot intuitively reflects the excellent stability of data distribution by presenting the 95% confidence intervals of clustering results. These results validate the favorable robustness of the clustering algorithm proposed in this paper and the rationality of parameter selection.

After clustering, the interval of the confidence envelope at 95% is calculated based on the clustering results. The specific calculation method is as follows:

(1) Extract a clustered dataset.

(2) Remove the top and bottom 2.5% of the data.

(3) Take the maximum value of the remaining data as the upper envelope boundary and the minimum value as the lower envelope boundary.

(4) Repeat the above process to generate the complete envelope intervals.

Through the envelope calculation process derived from **Table 2**, the envelope result of object 2 can be obtained, as depicted in **Figure 3**. The generated envelope model has a size of approximately 3.5 MB. Since the 1100 electronic and electrical objects in the original dataset represent only half of all such objects, the expected size of the complete rule model is approximately several tens of megabytes, which exceeds the storage constraints of FPGAs. Therefore, it is necessary to simplify the judgment model to meet the practical requirements of deployment.

Optimization of the Efficiency of the Decision Model:

To meet the storage constraints of FPGAs, which typically have available storage < 5 MB, we set a clear goal of model simplification: compressing the rule table of the decision model to less than 2 MB. In this paper, we employ regression analysis to approximate multiple envelope interval judgment rules with functional expressions, thereby replacing the original discrete rule sets with concise regression functions to significantly reduce storage requirements. The proposed regression analysis implementation process includes the following key steps:

(1) Envelope Boundary Preprocessing: First, hierarchical clustering is applied to the original envelope boundary data to identify rule sets with similar characteristics.

(2) Linear Regression Modeling: For each clustering result, a linear regression model is established to approximate the boundaries of the discrete envelope with continuous functions.

(3) Rule Reconstruction: Based on the results of the regression analysis, new functional expression decision rules are generated to replace the original interval-based decision rules.

Hierarchical clustering is applied to the envelope boundary rules, primarily based on the following two key considerations: First, clustering enables a more compact data distribution, thus effectively constraining the approximation errors that may arise during regression analysis. Second, given the unique properties of envelope interval data, hierarchical clustering preserves the hierarchical relationships within the data, which aligns well with the gradual nature of spacecraft state changes. Leveraging these advantages, we employ a hierarchical clustering algorithm to process the upper and lower boundary data of the envelope intervals separately, ensuring that the clustering results accurately reflect the boundary characteristics of the system behavior.

In regression analysis based on the clustering of envelope boundary intervals, we systematically compare the main regression methods: linear regression is used to establish a linear mapping between independent and dependent variables; polynomial regression captures nonlinear features through higher-order terms; logistic regression is suitable for classification tasks involving probability prediction; ridge regression addresses multicollinearity using L_2 regularization; and lasso regression achieves feature selection by L_1 regularization. Through an in-depth analysis of data characteristics, we identify the independent variables as channel numbers (discrete identifiers) and timestamps (continuous variables), with the dependent variable being the envelope boundary values (confidence intervals).

Furthermore, our research reveals that channel numbers, as categorical identifiers, do not have a significant impact on boundary value variations. Consequently, we reduce the dimensionality of the independent variables by retaining only the time dimension as the regression variable, thereby simplifying the regression problem to establish a functional relationship between envelope boundaries and time. In method selection, we adopt a progressive strategy: initially attempting a linear regression model; if the fitting error of the linear model exceeds the acceptable threshold, we upgrade to polynomial regression. This stepwise approach achieves an optimal balance between model complexity and prediction accuracy.

The linear regression modeling process adopted in this study consists of three key steps: First, the envelope boundary interval data is extracted from the clustered dataset; subsequently, a dual-boundary linear regression model is established, where the lower boundary model is given by $y = k_1 \times t + b_1$ and the upper boundary model by $y = k_2 \times t + b_2$ (where t represents the time variable, and k and b are the regression coefficients to be determined); Finally, the least squares method is used for regression calculation, whose mathematical essence is to optimize the objective function.

$$S = \sum_i (y_i - kt_i - b)^2$$

By taking partial derivatives of the system of equations and setting them to zero:

$$\partial S/\partial K = -2 \sum_i t_i (y_i - kt_i - b) = 0$$

$$\partial S/\partial b = -2 \sum_i (y_i - kt_i - b) = 0$$

Solving this system of equations yields the optimal regression coefficients k and b. Geometrically, this method essentially seeks the best-fitting line that minimizes the sum of the squared vertical distances of all data points to the line, thereby obtaining a precise linear model describing the envelope boundary variation over time. This process not only ensures the mathematical rigor of the model but also meets the computational efficiency requirements for engineering applications, laying a theoretical foundation for subsequent FPGA implementation. To evaluate the fitting performance of the linear regression model, two error metrics are employed: Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). RMSE is utilized to quantify the overall deviation between predicted values and actual values, and its formula is defined as:

$$RMSE = \sqrt{1/n \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

where y_i represents the true envelope boundary value, $\hat{y}_i$ denotes the predicted value, and n is the number of samples. MAPE is adopted to measure the percentage deviation of predicted values relative to the true values, and its formula is expressed as:

$$MAPE = 1/n \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%$$

It is important to note that when yi =0, absolute error is used instead of percentage error to avoid division by zero. The combined use of these two metrics provides a comprehensive assessment of the model's fitting performance and offers reliable data support for subsequent rule reconstruction.

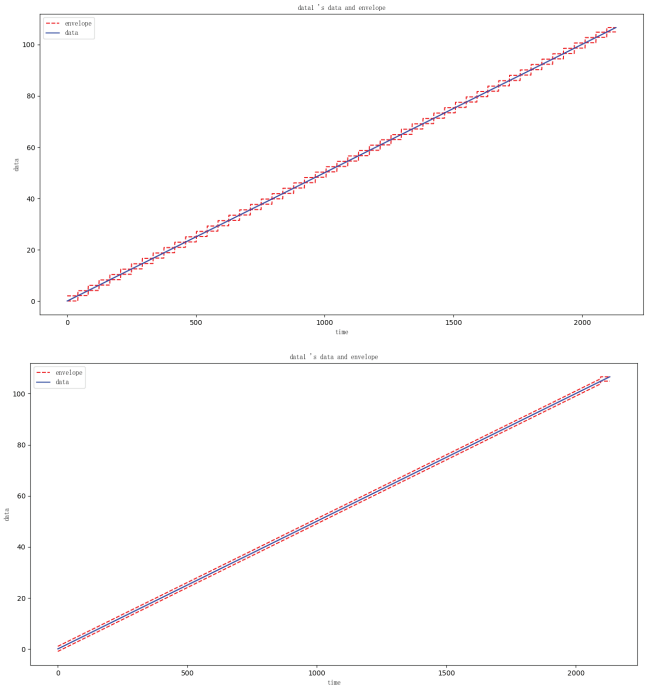

Figure 4: Comparison Between Original and Regression Envelopes. The left panel shows the original envelope results, contrasting with the regression-processed results in the right panel. Red curves represent the envelope boundaries, while blue traces depict the original data. Our evaluation demonstrates that the regression envelope encloses the original data with linear lines, and its parameter recording can replace the original segment-wise intervals, thereby effectively saving decision model space.

As illustrated in **Figure 4**, the regression envelope effectively approximates the boundaries of the original envelope through linear fitting. By storing only linear equations, it replaces the traditional method of storing envelope interval data segment by segment, significantly reducing the storage space requirements of the decision model. This parametric representation not only maintains the accuracy of the envelope boundaries but also greatly enhances the compression efficiency of the model.

Table 3: Actual Sample Dataset displays the envelope data derived from actual measurements using linear regression, with parameters K_1/b_1 characterizing the lower envelope and k_2/b_2 defining the upper envelope.

Table III. Actual sample dataset

Data	Time	k1	b1	k2	b2	num
Data1	1722830068038-1722830108604	0	0	0	0	16228
Data1	1722830108609-1722830110099	-0.00083	-0.15976	-0.00083	-0.06045	598
Data1	1722830110104-1722830111614	0	-0.703	0	-0.613	606
Data1	1722830111619-1722830115164	0.00087	-0.65121	0.00087	-0.55135	1420
Data1	1722830115169-1722830116604	0	0.613	0	0.703	576
Data1	1722830116609-1722830118094	-0.00083	0.54208	-0.00083	0.64236	596
Data1	1722830118099-1722830123654	0	0	0	0.073	2224
Data1	1722830123659-1722830125274	0.00077	0.05805	0.00077	0.15474	648
Data1	1722830125279-1722830126694	0	0.581	0	0.674	568
Data1	1722830126699-1722830130134	-0.00083	0.54551	-0.00083	0.64403	1376
Data1	1722830130139-1722830131129	0	-0.667	0	-0.574	398
Data1	1722830131134-1722830131694	0	-0.674	0	-0.576	226
Data1	1722830131699-1722830133044	0.00071	-0.61345	0.00072	-0.51834	540
Data1	1722830133049-1722830133639	0	-0.153	0	-0.057	238

DOI: 10.36838/IJHSR84.88

Data1	1722830133644-1722830170719	0	-0.018	0	0	14832
data2	1722830068038-1722830108604	0	0	0	0	16228
data2	1722830108609-1722830110099	0.00083	0.06045	0.00083	0.15976	598
data2	1722830110104-1722830111614	0	0.613	0	0.703	606
data2	1722830111619-1722830115164	-0.00087	0.55135	-0.00087	0.65121	1420
data2	1722830115169-1722830116604	0	-0.703	0	-0.613	576
data2	1722830116609-1722830118094	0.00083	-0.64236	0.00083	-0.54208	596
data2	1722830118099-1722830123364	0	-0.039	0	0	2108
data2	1722830123369-1722830125024	0.00075	-0.02648	0.00075	0.07033	664
data2	1722830125029-1722830125549	0	0.539	0	0.638	210
data2	1722830125554-1722830126694	0	0.581	0	0.674	458
data2	1722830126699-1722830130134	-0.00083	0.54551	-0.00083	0.64403	1376
data2	1722830130139-1722830131129	0	-0.667	0	-0.574	398
data2	1722830131134-1722830131694	0	-0.674	0	0.576	226
data2	1722830131699-1722830133044	0.00071	-0.61345	0.00072	-0.51834	540
data2	1722830133049-1722830133639	0	-0.153	0	-0.057	238
data2	1722830133644-1722830170719	0	-0.018	0	0	14832

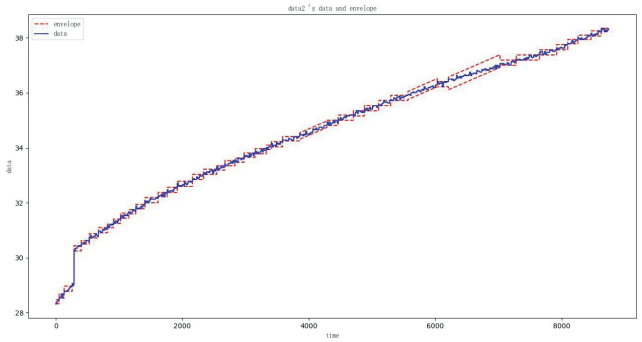

Figure 5: Envelope after Linear Regression on the Actual Sample Set illustrates the correspondence between the envelopes from **Table 3** and the original dataset, where the envelope boundaries are depicted in red curves and the raw data traces are shown in blue. This figure illustrates that clustered adjacent envelope intervals form linear regression lines, constituting the regression envelope that reduces the model space.

The experimental results shown in **Figure 5** demonstrate that this envelope simplification method based on regression analysis preserves the statistical characteristics of the original data while reducing the model storage space to an acceptable level. The regression envelope formation process fully considers the temporal characteristics of the spacecraft test data, ensuring the reliability of the model in practical engineering applications.

Experimental statistics indicate that the total size of the envelope interval data after linear regression is 1.47M, which meets our spatial requirements. Should the data size exceed the spatial threshold in future systems, the number of classifications can be reduced by appropriately expanding the clustering boundaries, thereby decreasing the overall data size.

Result and Discussion

Model Performance Evaluation:

We designed an error-statistical method to evaluate the performance of the proposed decision model. Since both the envelope interval calculation and the regression analysis process may introduce errors, it is necessary to estimate the overall error of the model to verify its reliability. The specific evaluation procedure includes: (1) selecting a normal data sample set; (2) applying the decision model to calculate the number of correctly classified samples; and (3) computing the confidence rate (number of correct samples / total number of samples).

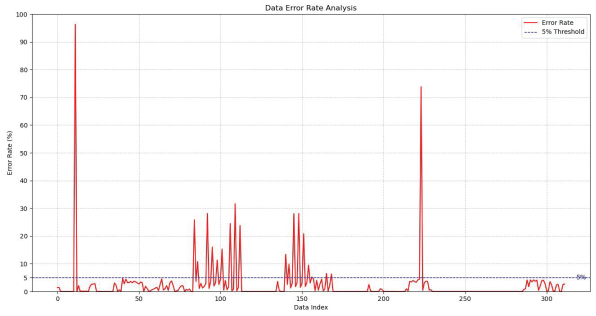

Figure 6: Evaluation Results of the Model presents the evaluation results for approximately 317 test subjects. The red curve represents the actual error rate, while the blue line indicates the 5% error threshold. Error evaluation indicates that approximately 92% of all samples have a judgment error below 5%, while around 8% of the samples exceed the judgment error threshold.

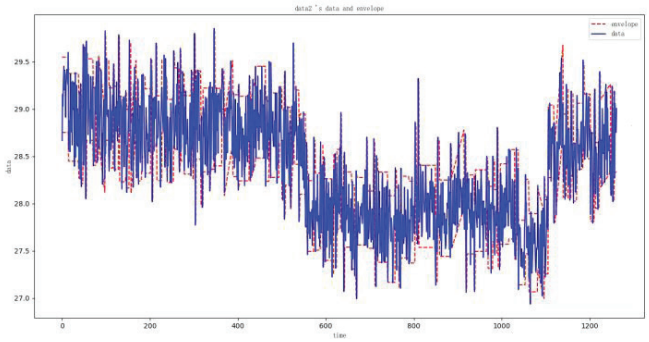

Figure 7: Error Out-of-Bounds Sample Set. In the plot of out-of-bounds samples, the red line represents the envelope curve, while the blue line depicts the sample curve. This analysis demonstrates that when the original data values in the sample sets are extremely close, the error introduced by linear regression exceeds the threshold range.

Based on the evaluation results of 317 test subjects in **Figure 6**, approximately 92% of the samples (292 samples) exhibit errors controlled within 5%, meeting the expected requirements, while approximately 8% of the samples (25 samples) exceed the error threshold. Analysis reveals that the primary cause of excessive errors is that when the range of the original data values is too narrow, the computed envelope interval becomes excessively tight. This results in the linear regression coefficient k being zero even when retaining five decimal places of precision, leading to significant errors (as illustrated in the case of **Figure 7**, where the error rate reaches 25.832%).

Summarize the results and the methods used with data from experiments. Ensure that all tables, figures, and schemes are cited in the text in numerical order. Trade names should have an initial capital letter, and trademark protection should be acknowledged in the standard fashion, using the superscripted characters for trademarks and registered trademarks, respectively. All measurements and data should be given in SI units where possible. Abbreviations should be used consistently throughout the text, and all nonstandard abbreviations should be defined on first usage. Authors are requested to draw attention to hazardous materials or procedures by adding the word CAUTION, followed by a brief descriptive phrase and literature references if appropriate. The experimental information should be as concise as possible, while containing all the information necessary to guarantee reproducibility.

Model Adaptive Adjustment:

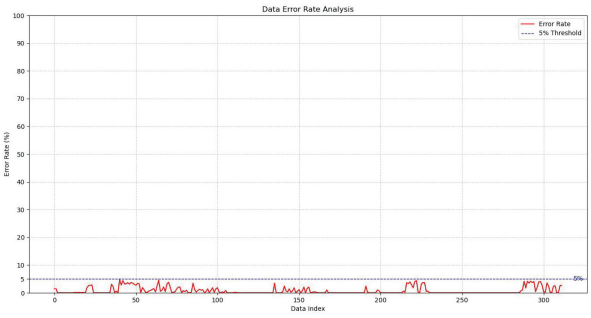

Figure 8: Post-Adjustment Evaluation shows error rates after model optimization. The red curve represents actual errors, while the blue line marks the 5% threshold. These results confirm that the adjusted regression envelope method ensures all test sample errors remain below 5%, meeting real-time precision requirements.

We conducted experiments to validate the effectiveness of the proposed systematic model optimization scheme in addressing excessive errors. Following the steps outlined in our optimization framework, we first classified the error-out-of-bounds samples into an independent dataset. Subsequently, we increased the confidence level of the envelope interval to 96%-99%, relaxing the boundary constraints to reserve error space for subsequent regression analysis. Next, we re-performed linear regression modeling and validated the results. Finally, we iteratively optimized the model until the error requirements were met. The core of this strategy lies in the elevation of the confidence level, which moderately expands the envelope interval to offset the approximation errors introduced during the regression process. The adjusted envelope is illustrated in **Figure 8**. The results demonstrate that the optimized model significantly improves decision accuracy while adhering to the 5MB storage limitation of an FPGA, thereby achieving the dual standards of performance and resource utilization required by spacecraft real-time testing systems. These findings validate the practical value of the confidence-level-based iterative optimization mechanism in handling boundary-sensitive data.

To validate the rationality of the proposed lightweight decision-making model, we conducted comparative experiments with three mainstream lightweight methods (decision trees, neural network compression, and quantization). The results demonstrate that the proposed method outperforms the others in storage efficiency (1.47 MB vs. ~6-18 MB), real-time performance (less than 100 ns per channel vs. ~200-500 ns), and balancing accuracy (less than 5%) while adapting to the periodicity of spacecraft testing and FPGA storage constraints. This method avoids the high update costs of neural networks, the storage redundancy of quantization, and the poor parallelism of decision trees, fully meeting the system's requirements for low storage, high real-time performance, and high reliability.

■ Conclusion

This study introduces a lightweight decision model construction method for Spacecraft Testing Systems using innovative envelope analysis and regression modeling. The constructed model ensures decision accuracy while compressing storage to under 2MB, fitting FPGA storage limits, and supporting a high-performance CPU + FPGA hybrid architecture for a real-time decision system. Experiments confirm its effectiveness in overcoming the storage and computational inefficiencies of traditional methods. Future work will focus on: (1) developing an FPGA-based online real-time detection system, and (2) establishing a CPU-FPGA collaborative online learning mechanism to enable adaptive model evolution through dynamic updates and parameter optimization, enhancing system robustness and intelligence in complex conditions.

■ Acknowledgments

I am deeply grateful to my high school for its support in experimental design, laboratory setup, testing, and other aspects of my research.

■ References

1. S. Y. Zhou, "Research on an automatic testing technology of incremental angular encoder based on LabVIEW," in Annual Meeting of the China Aviation Industry Technical Equipment Engineering Association, Qingdao, China, 2021, pp. 147–151.
2. Q. Li, D. H. Wu, D. Z. Yang, M. H. Fei, Y. Song, and Y. Liu, "Design and realization of fully automatic pump performance test system," International Journal of Engineering Research in Africa, vol. 52, pp. 69–75, 2024.
3. Y. Yi, H. Qu, S. Wang, Y. Jian, H. Wang, and N. Qu, "Design of batch automatic testing system based on electric cylinder," Journal of Physics: Conference Series, vol. 1234, no. 1, p. 012345, 2023.
4. J. L. Hennessy and D. A. Patterson, Computer Architecture, Fifth Edition: A Quantitative Approach, 5th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.
5. E. Mehmood and T. Anees, "Challenges and solutions for processing real-time big data stream: A systematic literature review," IEEE Access, vol. 8, pp. 119123–119143, 2020.
6. G. Moore, "Cramming more components onto integrated circuits," Proceedings of the IEEE, vol. 86, no. 1, pp. 82–85, Jan. 1998.
7. S. Borkar and A. A. Chien, "The future of microprocessors," Communications of the ACM, vol. 54, no. 5, pp. 67–77, May 2011. [Online]. Available: https://doi.org/10.1145/1941487.1941507
8. A. Sunil, N. R. G. S., A. Sri Pulijala, and S. S. C. V. R., "Spacecraft payload data processing with in-memory concurrent threads and data buffers," in 2024 IEEE Space, Aerospace and Defence Conference (SPACE), 2024, pp. 144–147.
9. S. A. McKee, "Reflections on the memory wall," in CF '04: Proceedings of the 1st Conference on Computing Frontiers, 2004, pp. 162–167.
10. V. Cuppu, B. Jacob, B. Davis, and T. Mudge, "High-performance DRAMs in workstation environments," IEEE Transactions on Computers, vol. 50, no. 11, pp. 1133–1153, Nov. 2001.
11. C. 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.
12. K. Compton and S. Hauck, "Reconfigurable computing: A survey of systems and software," ACM Computing Surveys, vol. 34, no. 2, pp. 171–210, Jun. 2002.
13. J. Su, F. Yang, X. Zeng, D. Zhou, and J. Chen, "Efficient memory partitioning for parallel data access in FPGA via data reuse," IEEE

DOI: 10.36838/IJHSR84.88

Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 10, pp. 1674–1687, Oct. 2017.
14. Y.-K. Choi, J. Cong, Z. Fang, Y. Hao, G. Reinman, and P. Wei, "A quantitative analysis on microarchitectures of modern CPU-FPGA platforms," in 2016 53rd ACM/EDAC/IEEE Design Automation Conference (DAC), 2016, pp. 1–6.
15. R. Shu, P. Cheng, G. Chen, Z. Guo, L. Qu, Y. Xiong, D. Chiou, and T. Moscibroda, "Direct universal access: Making data center resources available to FPGA," in 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19), Boston, MA, USA, Feb. 2019, pp. 127–140. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/shu
16. W. Yanwei, L. Rengang, X. Ran, and L. Junkai, "Data center heterogeneous acceleration software-hardware system-level platform based on reconfigurable architecture," Journal of Computer Research and Development, vol. 62, no. 4, pp. 963–977, Apr. 2025. [Online]. Available: https://crad.ict.ac.cn/en/article/doi/10.7544/issn1000-1239.202440041
17. Microchip USA. (2023). "A complete overview of FPGAs for space applications". Retrieved September 8, 2025, from https://www.microchipusa.com/industry-news/a-complete-overview-of-fpgas-for-space-applications
18. Sarah Azimi, Eleonora Vacca, Corrado De Sio, Luca Sterpone, and Andrea Portaluri. 2025. Mitigating Cross-Domain SEU Corruption in FPGA-Based AI Accelerators for Space Applications: Invited Paper. In Proceedings of the 22nd ACM International Conference on Computing Frontiers: Workshops and Special Sessions (CF'25 Companion). Association for Computing Machinery, New York, NY, USA, 193–198. https://doi.org/10.1145/3706594.3727947
19. Z. He, Y. Sun, B. Wang, S. Li, and B. Zhang, "CPU–GPU Heterogeneous Computation Offloading and Resource Allocation Scheme for Industrial Internet of Things," in IEEE Internet of Things Journal, vol. 11, no. 6, pp. 11152-11164, 15 March 15 2024, doi: 10.1109/JIOT.2023.3332748.
20. AMD, "XC7VX690T FPGA," 2023. [Online]. Available: https://www.amd.com/en/products/ adaptive-socs-and-fpgas/fpga/virtex-7.html
21. S. Lloyd, "Least squares quantization in PCM," IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129–137, Mar. 1982.
22. C. Weber, S. Hahmann, and H. Hagen, "Methods for feature detection in point clouds," in Proceedings of the 2010 International Conference on Computer Graphics and Vision, 2010, pp. 1–10.
23. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, OR, USA, 1996, pp. 226–231.

Authors

My name is Yuqian Han, and I am currently a 10th-grade student at Jinling High School, Hexi Branch in Nanjing, China. I have a strong interest in electronic engineering and computer science. In my spare time, I enjoy participating in scientific experiments to solve practical problems.