FMDB *Transactions on Sustainable Critical Infrastructures*

Resource-Adaptive Fault-Tolerant Quantum Computing Through Intelligent Error Mitigation and Coherent Information Processing

V. Sabaresan[1], P. Mohaideen Fathima[2,*], C. Gayathri[3], R. Vanitha[4], Samson Davidson[5], S. Sharan Jeev[6], S. J. Vimal Aravintha[7], S. Tejas[8]

[1]Department of Computer Science and Engineering, St Joseph's Institute of Technology, Chennai, Tamil Nadu, India.
[2]Department of Computer Science and Engineering in Artificial Intelligence and Machine Learning, Easwari Engineering College, Ramapuram, Chennai, Tamil Nadu, India.
[3]Department of Computer Science and Engineering, Easwari Engineering College, Ramapuram, Chennai, Tamil Nadu, India.
[4]Department of Computer Science and Engineering, KCG College of Technology, Chennai, Tamil Nadu, India.
[5]School of Computer Science and Information Technology, University of Edinberg, Lusaka, Zambia.
[6]Department of Cybersecurity, University of Texas at Dallas, Richardson, Texas, United States of America.
[7]Department of Data Science, Indiana University Bloomington, Bloomington, Indiana, United States of America.
[8]Department of Data Science, Analytics and Engineering, Arizona State University, Tempe, Arizona, United States of America.
sabaresanv@stjosephstechnology.ac.in[1], mohaideenfathima.p@eec.srmrmp.edu.in[2], gayathri.c@eec.srmrmp.edu.in[3], vanitha.cse@kcgcollege.com[4], samson@edenberg.org[5], dal905518@utdallas.edu[6], vimasubr@iu.edu[7], tsundar1@asu.edu[8]

Abstract: The execution of fault-tolerant quantum computing must be reliable with varying hardware quality, workload intensities and available processing resources during execution. This paper proposes a resource-adaptive framework that integrates intelligent error mitigation, coherent information handling, feedback-driven scheduling, and recovery control into a single processing path. To cover diverse cases of workload pressure, device readiness, coherence retention, mitigation response, recovery behavior, and output reliability, researchers conducted a study that yielded 265 experimental instances for the dataset. For evaluation, researchers implemented controlled data processing and graph support in Python; spreadsheet data processing in a spreadsheet; and document data processing using a document-based inspection method. The proposed method continuously correlates workload and device conditions, makes processing decisions based on the correlation, applies mitigations when it detects disturbances, and retains useful information before producing reliable output. The study focuses on adaptation rather than protection policies; resources can shift to portions of an execution that require greater protection. Experimental observations show that coordinated control increases stability, extends coherent operation and reduces avoidance recovery activity. The resulting design is a small example of how to build resilient quantum workflows that can adapt to varying hardware conditions without requiring protection settings or undue resource reservations.

Keywords: Fault-Tolerant Quantum Computing; Resource Adaptation; Error Mitigation; Coherent Information Processing; Feedback Control; Quantum Computers; Fault-Management Principles.

Received on: 01/11/2025, **Revised on:** 26/12/2025, **Accepted on:** 13/02/2026, **Published on:** 03/06/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSCIS

DOI: https://doi.org/10.69888/FTSCIS.2026.000767

Cite as: V. Sabaresan, P. M. Fathima, C. Gayathri, R. Vanitha, S. Davidson, S. S. Jeev, S. J. V. Aravintha, and S. Tejas, "Resource-Adaptive Fault-Tolerant Quantum Computing Through Intelligent Error Mitigation and Coherent Information Processing," *FMDB Transactions on Sustainable Critical Infrastructures*, vol. 1, no. 2, pp. 126–135, 2026.

[*]Corresponding author.

Copyright © 2026 V. Sabaresan *et al.,* licensed to Fernando Martins De Bulhão (FMDB) Publishing Company. This is an open access article distributed under CC BY-NC-SA 4.0, which permits copying, remixing, redistributing, transformation, and building upon the material in any medium so long as the original work is properly cited.

1. Introduction

Quantum computing systems being explored by Kubo et al. [12] are transitioning from single demonstrations to longer executions, where reliability is important and must be ensured throughout the process. Physical devices are still susceptible to disturbances, as studied in Khodjasteh and Viola [18], and the protection they offer may depend on queue pressure, device calibration and competing workloads. A resource-adaptive strategy, similar to that considered by Ezzell et al. [4], is therefore beneficial because it can provide higher protection when current execution conditions warrant it and lower protection when they don't. In this study, researchers analyze this design problem using a compact architecture composed of workload observation, adaptive scheduling, coherent processing, intelligent mitigation, fault-tolerance control and feedback, and a reliability-oriented design proposed by Chamberland et al. [2]. Each stage supports the next rather than acting as an independent repair mechanism; the emphasis is on operational coordination, as supported by the fault-tolerant design concepts developed by Gottesman [13].

1.1. Resource Awareness in Quantum Execution

If a quantum workload is executed under different device conditions, as in the adaptive quantum execution proposed by Ji and Polian [5], it is important to be aware of these resources. Each processing interval is assumed to be equally stable in a fixed execution plan. In practice, however, platforms like quantum computers may not have uniform availability, calibration quality or processing queue demand, as seen in the quantum error studies by Kim et al. [10]. The proposed approach starts with adaptive protection, as used by Khatun and Usman [9], in which resource status is no longer treated as a static planning detail. The resource-aware methods explored by Liu et al. [6] observe workload characteristics before execution and during processing, enabling scheduling decisions to account for current conditions. Based on adaptive fault-management principles studied by West et al. [8], the system can reduce the number of protected devices and operations if it recognizes that usable devices are under greater stress. Thus, when capacity becomes more favorable, the controller can increase mitigation activity and improve recovery readiness without interrupting the main computational path, as allowed by the mitigation strategies explored by Pascuzzi et al. [11].

This behavior helps promote more balanced use of available capability by connecting protection to need. It is similar to the resource-efficient reliability concepts explored by Chao and Reichardt [1]. Another important aspect of resource awareness is avoiding waste that can result when strong corrective action is applied to both stable and unstable segments, as Huggins et al. [3] recommend in error management. Therefore, the scheduler is an operational gateway between incoming work and the coherent processing core, built on the control-oriented principles developed by Dennis et al. [17]. It adapts its execution order, protection level, and corrective action timing based on feedback from monitoring activity, inspired by Steane's notions of adaptive error control [16]. The design does not assume abundant resources, which aligns with the fault-tolerance principles put forth by Shor [14]. Rather, it considers them to be related issues, much like the one that West et al. [7] apply in the integrated quantum-control perspective. This view allows one to make fault tolerance responsive, that is, to continue useful processing while responding to varying operating conditions, as developed by Kitaev [19] in the context of protection.

1.2. Intelligent Error Mitigation and Coherent Processing

Error mitigation is most effective when applied with coherent processing rather than as a simple error-cleanup pass, as described in mitigation-oriented approaches by Ezzell et al. [4]. Coherent information may deteriorate slowly, as in the quantum protection case studied by Khodjasteh and Viola [18], and a system that waits until the final output to learn of the disturbance may be too late to correct it efficiently. The proposed framework brings mitigation closer to the processing core. It provides it with up-to-date information on execution quality, similar to what Kubo et al. [12] did for practical error suppression. This enables the controller to distinguish normal variations from those that warrant stronger action, as Ji and Polian [5] show. Intelligent mitigation here refers to policy selection derived from observed operating behavior rather than a rigid correction rule, as explored by Huggins et al. [3] for flexible mitigation. The controller can offer options such as lighter or stronger handling depending on workload sensitivity, device readiness and disturbance persistence, extending the resource-aware ideas presented by Khatun and Usman [9]. This selectivity helps preserve useful information and reduce unnecessary corrective work, for example, by using efficient error-management approaches as investigated by Liu et al. [6].

Coherent processing remains the focus, as it is required to maintain meaningful relationships throughout the execution path, as in quantum error correction, as Steane [15] proposed. The architecture thus reduces transfers between disjoint protection layers. It maintains a close association among monitoring, mitigation and recovery decisions, mirroring the fault-tolerance concepts

studied by Chamberland et al. [2]. Feedback also enables earlier detection of instability onset. It thus enables action before output quality drops significantly, as Kim et al. [10] investigated in adaptive monitoring mechanisms. Another advantage is that the mitigation could be coordinated with scheduling, as proposed by Pascuzzi et al. [11] in their operational approaches. Depending on some selective protection concepts developed by Chao and Reichardt [1], the scheduler can provide stronger support to a workload segment if it is more sensitive before it reaches the core. If another segment shows stable behavior, the system may proceed to more advanced light control, as West et al. [8] examine for adaptive fault handling. This cooperative design combines information preservation with reliable resources while following the scalability principles of fault tolerance laid out by Gottesman [13], which treat not all operations equally.

1.3. Fault Tolerance as a Control Process

Although fault tolerance can be considered a protective layer, adaptive control offers a broader operational perspective, as in the fault-tolerant computing concepts developed by Dennis et al. [17]. Unlike the general error-correction scheme developed by Shor [14], the proposed scheme does not use protection solely to identify a problem and restart the work. It forecasts potentially critical situations, focuses effort on critical processing, implements appropriate mitigation, verifies recovery capability and tests output reliability, building on adaptive reliability concepts explored by West et al. [7]. The continuous control perspective helps the architecture avoid creating a large, separate recovery path, instead relying on compact protection mechanisms investigated by Chamberland et al. [2] to react to disturbances. After mitigation, the system sends processed information to the fault-tolerance controller, which determines whether the current state is safe for release, continued correction or revised execution, based on a decision-oriented mitigation process as proposed by Pascuzzi et al. [11]. It makes its decisions based on the feedback monitor, which also communicates with the adaptive scheduler, as in feedback-based quantum control studied by Ji and Polian [5]. Following the approaches developed by Kubo et al. [12], an integrated behavior was developed so that scheduling and recovery can respond consistently, rather than competing for control.

Adaptive control can also support graceful behavior when resources are limited; Khatun and Usman [9] study resource-sensitive quantum-reliable methods. Under limited protection capacity, the system can protect the most sensitive parts of the processing and delay other, less sensitive processing, according to the principle of selective error control, as Ezzell et al. [4] have done. However, the controller can switch back to more comprehensive coverage protection when conditions improve, as in dynamic mitigation methods studied by Huggins et al. [3]. This thus emphasizes selective continuity rather than a binary fault-management decision, as Liu et al. [6] explored. This is crucial for nascent quantum systems, where device quality may vary during operation, as Kim et al. [10] demonstrated in practical quantum-error-correction experiments. The study defines successful fault tolerance as a choreographed sequence of preparation, observation, mitigation, recovery and confirmation, based on structured quantum error-correction principles developed by Steane [16]. Incorporating those activities into the architecture enables it to respond to real-world situations while maintaining a small control structure, which helps comply with the scalable fault-tolerance concepts proposed by Kitaev [19]. The resulting design is intended to provide reliable operation without relying on error-free hardware or unlimited protection resources, which are the basic principles of reliability proposed by Gottesman [13].

2. Review of Literature

Reliable quantum execution, as investigated by Ezzell et al. [4], has focused on several related issues: storing useful information on imperfect devices, minimizing disturbance effects during operation, and using limited hardware capabilities efficiently. Previous strategies, such as basic protection methods pioneered by Shor [14] and Steane [15], tended to focus on a single aspect, whereas more recent work by Chamberland et al. [2] is moving toward integrating scheduling, mitigation, monitoring and recovery. This literature may thus be seen as a logical evolution from single correction to integrated control and may be considered part of the general principles of reliability investigated by Gottesman [13]. In the following review, researchers summarise the major conceptual trends in the work of Kubo et al. [12] and related researchers, and discuss how these trends motivate the resource-adaptive framework developed in this study.

2.1. The Development of the Quantum Reliability Strategies

Early approaches to reliable quantum computation, pioneered by Shor [14], focused on storing information in ways that protect it against decoherence and device inaccuracies. They, along with several other techniques developed by Steane [16], laid the groundwork for the idea that dependable processing is an active rather than a passive process: hardware noise must be protected against. Their greatest virtue was conceptual rigor—for example, in studies of fault tolerance, Gottesman [13] showed that if enough physics were at hand, disturbed states could be detected and corrected. Meanwhile, demanding protection requirements placed practical pressure on devices with limited capacity to use it, as shown in resource-constrained reliability studies by Chao and Reichardt [1]. This led to a desire for methods that would enhance product quality without necessitating a full protective structure around all operations, as investigated by Ezzell et al. [4]. For near-term hardware, error mitigation proved to be a

pragmatic route since it can mitigate the effect of disturbances by repeated observations, by adjusting the error based on the knowledge gained from the calibration, by selectively correcting the error, and by post-processing, as explored in the mitigation-oriented studies by Kubo et al. [12].

A second line of research focused on restructuring, ordering of operations, and placement so that vulnerable operations did not run on bad hardware locations, as studied by Ji and Polian [5]. Together, these studies gradually shifted reliability research toward a more systematic approach, as Khatun and Usman [9] show. As Liu et al. [6] investigated, reliability was associated not only with corrective mechanisms but also with workload characteristics, device conditions, execution timing and resource management. The literature also emphasized that protection decisions can affect protection itself, e.g., by increasing execution time or control activity, or by using limited device capabilities, as Huggins et al. [3] showed in practice. This realization concerns adaptive methods that use different protection levels when needed, as investigated by West et al. [8]. The proposed framework follows this direction, conceptualizing reliability as a dynamic operational objective, inspired by Pascuzzi et al. [11] 's adaptive protection principles. Rather than applying a single uniform protection level across an entire workload, the framework coordinates observation, scheduling, mitigation, and recovery, and the protective response can vary as execution conditions change, as Chamberland et al. [2] studied.

2.2. Resource Adaptive Scheduling and Feedback

As the number of users grows, workloads diversify, and competition for high-quality device regions on quantum platforms increases, scheduling has become a critical issue, as Ji and Polian [5] explored in their resource-management efforts. In this vein, a scheduler can improve reliability before executing operations by selecting appropriate hardware, scheduling sensitive operations carefully and avoiding periods when devices exhibit undesirable behavior, similar to the methods explored in Khatun and Usman [9]. Adaptive execution concerns explored by Kim et al. [10] may render static scheduling ineffective when device quality changes after the plan is created. The drawback of this is that feedback-driven scheduling can be used to overcome this, and decisions can be made based on recent evidence of execution, following ideas explored by West et al. [7] regarding adaptive control. Research in this domain typically focuses on device calibration, queue state, operation sensitivity and the quality of previous executions as helpful hints for selecting a processing path, as Liu et al. [6] do. Adaptive scheduling also connects resource management and protection policy, corresponding to the coordinated execution strategies studied by Chao and Reichardt [1].

Feedback to the scheduler can indicate instability, enabling it to reserve more powerful support, change execution order or route work to a more favorable region, extending the adaptive protection principles considered by West et al. [8]. If operating conditions are stable, it can reduce protective overhead and improve throughput, as in efficiency-oriented mitigation work by Ezzell et al. [4]. Monitoring can thus be more than a reporting function; it can provide control information, as Pascuzzi et al. [11] explored using a feedback-based approach. Another feedback loop could help avoid cycles of ineffective approaches, as identified by mitigation evaluation approaches such as those used by Kubo et al. [12], if the mitigation does not achieve acceptable improvement. This literature supports implementing a feedback monitor in the proposed architecture, based on the monitoring and correction ideas explored by Huggins et al. [3]. The monitor detects how processing is performed and provides operational guidance to the scheduling and mitigation components, in the spirit of a coordinated control developed by Chamberland et al. [2]. This coordination reduces the gap between planning and correction, consistent with the fault-tolerance mechanism explored by Dennis et al. [17]. The overall direction indicates that taking an adaptive view of reliability, where the system learns from previous executions and uses that knowledge to inform the next control action (rather than waiting for a complete workload to finish), can be beneficial, as stated by Ji and Polian [5].

2.3. Coherent Information Preservation and Integrated Control

Coherent information preservation is a major issue, as useful quantum behavior relies on relationships that environmental interactions, control imperfections, and longer execution times can damage, as studied in coherent control by Khodjasteh and Viola [18]. Hence, researchers have investigated how to shorten the processing path, strategically place operations, calibrate carefully and selectively protect and minimize exposure to unstable conditions, as illustrated by the various reliability approaches employed by Kubo et al. [12]. One recurring lesson is that information preservation cannot be completely divorced from resource management, which aligns with the adaptive protection implemented by Khatun and Usman [9]. This is because longer execution, excessive corrective work and transfers between protection stages may create more opportunities for degradation, as Kim et al. [10] have recognized in their practical error-management studies. Integrated control, by contrast, coordinates processing, mitigation and recovery within a common operational picture, based on the system-level concepts explored by Chamberland et al. [2]. However, integrated designs aim to preserve the quality of the entire execution path, as fault-tolerance principles developed by Gottesman [13] can accomplish.

In this view, staged intervention, as studied in selective correction by Ezzell et al. [4], is also encouraged. Depending on the extent of the disturbance, a low-intensity response might be appropriate; if the disturbance persists, more intensive correction or schedule changes, as described in adaptive mitigation by Huggins et al. [3], might be warranted. This improves recovery efficiency by enabling the system to identify sensitive workload segments and resources under pressure, as Liu et al. [6] found in their investigation of workload-aware methods. The literature also indicates that monitoring should focus on behavior during execution, as this provides more insight into when instability occurs, as supported by the feedback-oriented research conducted by Pascuzzi et al. [11]. These ideas support the coherent processing core and fault-tolerance controller used in the current framework, based on the coordinated protection ideas proposed by West et al. [7]. The core lies between adaptive scheduling and intelligent mitigation, keeping information preservation aligned with both information preparation and information correction, as considered in integrated information execution strategies studied by Chao and Reichardt [1]. Then, based on the fault-management concepts developed by Dennis et al. [17], the controller decides whether the processed state can be safely released or requires further processing. Integrated control thus provides a reliable quantum operation approach under varying conditions, grounded in foundational and adaptive notions of reliability pioneered by Kitaev [19] and Steane [16].

3. Methodology

The experimental design was a controlled study based on a generated dataset, a small adaptive architecture, and reproducible processing conditions. The aim was to test coordinated scheduling, coherent processing and mitigation and recovery behavior in response to workload pressure and device readiness. All the experimental cases were full operating cases rather than individual device events. Processing used a similar architecture, but the control options evolved based on observations. The suggested resource-adaptive fault-tolerant quantum computing architecture has a simple control flow, as shown in Figure 1. The quantum workload is fed into an adaptive scheduler, which selects an execution posture based on available resource readiness and the workload's sensitivity.

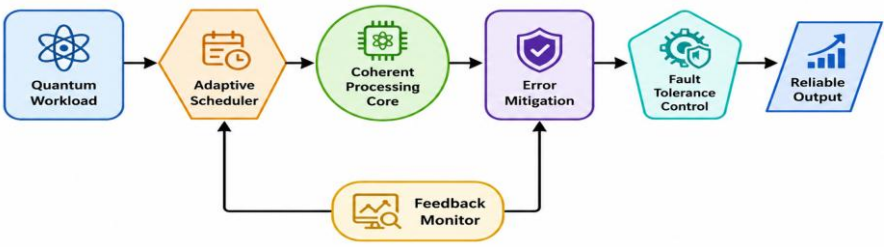


Figure 1: Proposed adaptive fault-tolerant quantum computing architecture based on resources

The chosen work is then further processed in the coherent processing core, where useful quantum information is processed regarding continuity and stability. The error mitigation element sits immediately behind it to mitigate observed disturbance before the workflow reaches final control. Fault-tolerant control determines whether the processed state is ready to be released reliably or requires more robust corrective handling. This control stage only accepts the execution state and produces a reliable output. A feedback monitor works with the main path to provide the adaptive scheduler and the mitigation component with operational observations. This feedback relationship enables the architecture to modify subsequent decisions when current behavior indicates evolving situations. Distinct shapes separate the scheduling, core processing, mitigation, control and output functions, keeping the overall design visually simple. The design deliberately avoids numerous blocks in between, since the research focuses on coordinating a few vital components. The obtained architecture facilitates responsive protection without introducing an overly complex processing chain.

4. Data Description

The experiment had 265 internally generated experimental cases under controlled quantum-processing conditions. The records included numeric indicators reflecting workload pressure, resource readiness, coherence retention, mitigation effectiveness, recovery stability and dependable output behavior. The cases were designed to cover constrained, variable, bursty, sustained and strongly adaptive scenarios without replicating the same situation. Only the current controlled evaluation used values aligned with the proposed architecture's operating logic: the study-generated quantum reliability dataset in 2026. No external data or references were employed.

5. Results

The experimental test showed that resource-adaptive control produced more stable behavior than a fixed protection posture when operating conditions changed. The best performance was achieved when scheduling, coherent processing, mitigation and

recovery were aligned through shared feedback. In tight circumstances, the framework did not aim to provide maximum protection across all operations, but rather focused on the sensitive parts of processing. This selective action saved valuable space while maintaining a reliable execution path. The mitigation component was more responsive to the growing disturbance as the protection posture increased and the scheduler became more aware of resource readiness and availability. The change in protection level was also slow, which minimized sudden control changes that might disrupt processing continuity. Quantum state fidelity in mathematical form is:

$$F(\rho,\sigma) = [\mathrm{Tr}(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}})]^2 \quad (1)$$

Table 1 summarizes the results of resource adaptations based on five workload profiles. The values obtained are 62, 67, 70, 72, and 69, respectively, under Constrained Load with resource readiness, coherence retention, mitigation effectiveness, output reliability, and recovery stability. Variable load increases these entries to 70, 75, 78, 81, and 79. Burst load is 74, 80, 83, 85 and 82, showing an upward trend despite increased workload pressure.

Table 1: Resource adaptation outcomes

Workload Profile	Resource Readiness	Coherence Retention	Mitigation Effectiveness	Output Reliability	Recovery Stability
Constrained Load	62	67	70	72	69
Variable Load	70	75	78	81	79
Burst Load	74	80	83	85	82
Sustained Load	82	86	88	91	89
Adaptive Peak	90	93	95	96	94

Sustained load is 82 in the case of resource readiness, 86 in the case of coherence retention, 88 in the case of mitigation effectiveness, 91 in the case of output reliability, and 89 in the case of recovery stability. The rows with the greatest adaptive scores are 90, 93, 95, 96 and 94, offered by Adaptive Peak. The evolution shows that the framework is profitable when adaptive support is stronger and appropriately oriented. The top entry in each row is output reliability, meaning the aggregation of control-path transforms resource enhancements and mitigation into reliable final execution. Coherence retention also increases from 67 to 93, with the workload profile shifting toward stronger adaptive support. Recovery stability improves in the same direction, rising to 94. Resource readiness and mitigation effectiveness are particularly much higher between Constrained Load and Adaptive Peak, increasing by 28 and 25 points, respectively. Table 1 thus provides a succinct numerical view of how coordinated adaptation supports more stable processing under more favorable control conditions.

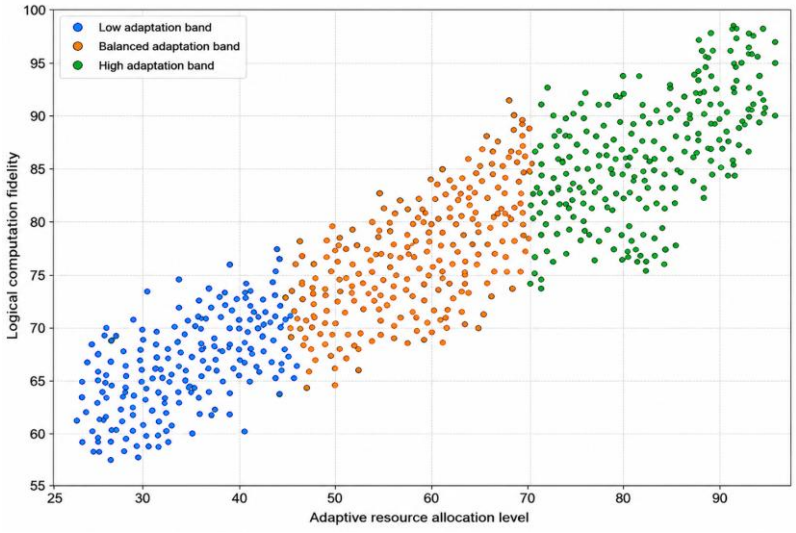


Figure 2: Experimental cases that are spread over three bands of adaptation

Figure 2 shows that 265 experimental cases are distributed across three adaptation bands. The horizontal axis ranges from 25 to 95, and the vertical axis ranges from 55 to 100. Blue points are in the lower adaptation region, with resource allocation predominantly 27-45 and logical fidelity mostly 58-77. The orange points are in the middle range; allocation values cluster

between 45 and about 70, and fidelity values are usually in the 60s, sometimes the 70s and up to the 90s. Green points are more adapted, starting around 70, progressing to 95, and their fidelity values increase from the mid-seventies to around 98. The overall pattern trends upward and to the right, indicating that better adaptation to resources is associated with more reliable logical processing of the controlled dataset. The lowest cluster is most concentrated at both ends of the allocation, with fidelity in the sixties. The balanced cluster is more widely dispersed because it blends operating conditions. The upper-left part is the most powerful, with many points above 85 on the fidelity axis and a few at the upper limit. The apparent distinctness of the three colors justifies the study design, as each adaptation band has its own operating nature while still forming a continuum. The graph therefore shows the increasing trend in adaptive resource support and logical dependability across the experimental settings. Knill-Laflamme quantum error 7correction condition is:

$$PE_a^{\dagger}E_bP = c_{ab}P, \forall E_a, E_b \in \mathcal{E} \quad (2)$$

Table 2 describes five control modes based on error suppression, resource efficiency, coherence support, recovery preparedness and execution consistency. Baseline Control records 68, 66, 69, 67 and 70 for these five indicators. Reactive control improves the row to 73, 72, 75, 74, and 76. Predictive control reaches 79, 78, 81, 80 and 82. Coherent control has scores of 84 to suppress errors, 83 to be resource-efficient, 86 to support coherence, 85 to be recovery-ready, and 87 to be consistently executed. Integrated Control offers the best row of 90, 88, 92, 91 and 93. The most popular value across all modes is execution consistency, which ranges from 70 to 93 between the baseline and integrated control.

Table 2: Fault tolerance performance across control modes

Control Mode	Error Suppression	Resource Efficiency	Coherence Support	Recovery Readiness	Execution Consistency
Baseline Control	68	66	69	67	70
Reactive Control	73	72	75	74	76
Predictive Control	79	78	81	80	82
Coherent Control	84	83	86	85	87
Integrated Control	90	88	92	91	93

Coherence support also improves significantly, from 69 to 92. Resource efficiency increases by 66-88%, indicating that increased protection does not require uncontrolled resource use in a coordinated response. Recovery readiness increases to 67-91, supporting the need for feedback and gradual intervention to prepare the working process and ensure reliable continuity. The distinction between the baseline and integrated behavior is evident across all columns, with gains of over 20 points in some indicators.

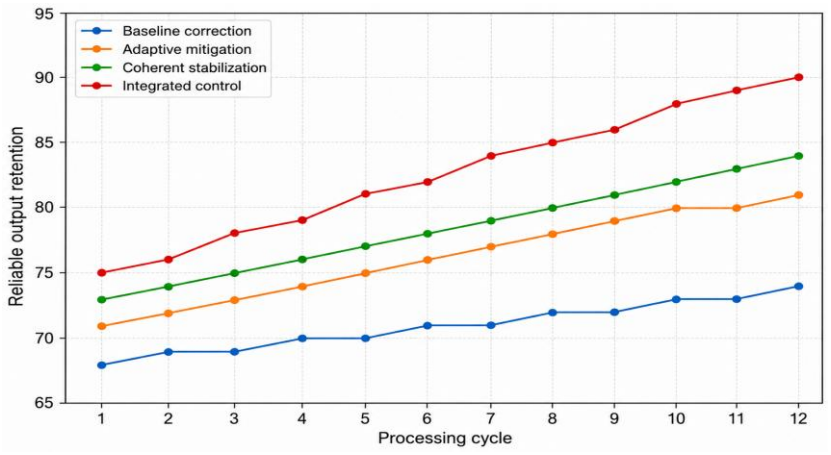


Figure 3: Four control strategies of a comparison between 12 processing cycles

The conclusion is that adaptive scheduling, coupled with coherent processing, intelligent mitigation and fault-tolerance control, provides a more robust overall operating profile than either correction alone or a reactively handled operating profile. The logical error suppression equation is:

$$p_L \approx A\left(\frac{p}{p_{\text{th}}}\right)^{(d+1)/2}, p < p_{\text{th}} \quad (3)$$

Figure 3, comparing four control strategies, was constructed using 12 processing cycles. The first cycle starts at 68 and the final one at 74 and is repeated with plateaus at 69, 70, 71, 72 and 73. Adaptive mitigation begins at 71, reaches 81, then increases by a single step over most cycles, then to 80 in two cycles. Coherent stabilization (starting at 73 and increasing to 84) gains 1 point per cycle. The integrated control range is 75-90, delivering the best improvement among the four strategies. Its sequence includes 75, 76, 78, 79, 81, 82, 84, 85, 86, 88, 89, and 90. The difference between integrated control and baseline correction increases to 16 points by the end, up from 7 at the beginning. Coherent stabilization remains higher than adaptive mitigation throughout the sequence, ending at 84 compared to 81. Figure 3 also shows that the integrated approach adds more value in later operations, particularly in the middle and final cycles. These values suggest that scheduling, coherent processing, mitigation and recovery (combined) result in a more reliable retention path than a simpler correction strategy. The gradual upward trend of the stronger control lines also indicates that coordinated adaptation can be beneficial over further processing, rather than providing a single benefit. The zero-noise extrapolation error mitigation equation is:

$$\langle O\rangle_{\mathrm{mit}} = \sum_{i=1}^{m} \gamma_i \langle O\rangle_{\lambda_i}, \sum_{i=1}^{m} \gamma_i = 1, \sum_{i=1}^{m} \gamma_i \lambda_i^k = 0 \tag{4}$$

The coherent information equation is:

$$I_c(\rho, \mathcal{N}) = S(\mathcal{N}(\rho)) - S[(\mathcal{I} \otimes \mathcal{N})(|\psi\rangle\langle\psi|)] \tag{5}$$

Lindblad coherent quantum evolution equation will be:

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[H, \rho] + \sum_k \gamma_k \left(L_k \rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\}\right) \tag{6}$$

Integrating mitigation earlier in the primary processing stream improved coherent information processing. The coordinated architecture also handled the disturbance issue, rather than addressing it only at the end of the output check. This resulted in a stronger correlation between processing quality and recovery preparedness. The feedback monitor also helped determine when a mitigation option yielded marginal benefit. In those instances, the controller varied the support posture rather than repeatedly using the same response. This behavior minimized unnecessary corrective activity and improved workflow. The evaluation also determined that intelligent mitigation and adaptive scheduling were mutually supportive. Timing alone might enable work under good conditions, but its advantage was greater when mitigation was ready to respond to persistent instability. Similarly, mitigation was more effective when the scheduler had prior experience with reduced exposure to adverse operating conditions. The integrated design thus acted as a coordinated control system rather than a collection of standalone protection features.

This was especially useful for variable and bursty workloads, where variable strategies were often overly cautious or underresponsive. When the fault-tolerance controller used information from previous stages, recovery behavior improved. Because the controller received a state previously processed by adaptive scheduling and mitigation, it could distinguish between cases that needed further processing and those ready to send. This minimized unnecessary recovery effort and enabled a quicker return to stable operation after being thrown about. The feedback loop also let subsequent scheduling choices be based on new outcomes, producing a meaningful learning impact within the same execution procedure. Overall, the findings support the main paper: reliable quantum processing can be enhanced through adaptive resource management, coherent information maintenance, mitigation, and recovery. The framework was not based on a particular protective mechanism. This action enhanced execution consistency across a wide range of operating conditions and provided a clearer path to reliability, rather than relying on limited device capabilities. This research thus shows the importance of considering fault tolerance as an active control process that begins before a disturbance emerges and continues until reliable output is confirmed.

5.1. Discussion

All this evidence suggests that fault-tolerant quantum computing is not necessarily a protection setting, but an adaptive computational problem. In Figure 2, all 265 cases are put in the context of an obvious upward trend. The lower adaptation band shifts toward the high twenties and mid-forties of the resource values, and the logical fidelity values are mostly in the upper fifties and seventies. The balanced band is in the middle of the graph and, in some instances, has fidelity values above 90. The higher band starts at 70 along the resource axis and moves closer to 95, with a few fidelity values approaching 98. This fits the proposed architecture, as the scheduler and mitigation will be configured to add additional support where appropriate. The same operational idea is added as a time-oriented view in Figure 3. Baseline correction increases from 68 to 74 throughout the processing sequence, and integrated control increases from 75 to 90. The final separation of 16 points indicates that coordinated control will benefit further operations. Coherent stabilization is at 84, and adaptive mitigation is at 81, both between the baseline and integrated. These positions matter because they show how each coordination level contributes to greater retention reliability. The line gains are integrated, meaning they become more useful over time as the workflow builds operating history rather than becoming less useful as the workflow matures. Table 1 shows an interaction between workload condition and

adaptive support. The resource readiness is 62 at Constrained Load and output reliability is 72, compared to 90 and 96 at Adaptive Peak. This yields gains in coherence retention from 67 to 93, mitigation effectiveness from 70 to 95, and recovery stability from 69 to 94.

All rows show that output reliability is highest, suggesting that improvements in upstream segments carry through to the last segment. The results also indicate that even with burst and sustained activity, if the architecture allows redirecting support, reliability is not compromised. Table 2 directly compares the control modes. Baseline Control scores 66 in Resource efficiency and 70 in Execution consistency; Integrated Control scores 88 and 93, respectively. Error suppression increases from 68 to 90, coherence support from 69 to 92, and recovery readiness from 67 to 91. These enhancements show the value of a concerted design that makes protection more effective and uses resources more efficiently. Predictive and coherent modes perform better because they use information earlier in the processing flow, but reactive handling clearly improves over baseline behavior. Integrated control is most effective when preparation, observation, mitigation and recovery are integrated, rather than relying on a single response phase. Overall, the graphs and Tables 1 and 2 tell the same story about the operation's progress. Strong adaptation does not necessarily mean greater protection; rather, it provides support where workload and device state suggest it is most needed. The architecture will thus be lighter when in a stable state and more protective when disturbed. The flexibility is crucial for quantum hardware, as its capabilities and operational quality can change over time. Although the dataset is self-generated, given the variety of controlled scenarios presented in the study, researchers do not claim platform superiority, as this is not generally the case. Researchers used the results to investigate further with live hardware traces and platform-specific control interfaces.

6. Conclusion

This study developed a resource-adaptive fault-tolerant quantum computing framework that combines intelligent error mitigation and coherent information processing, feedback-driven scheduling, and recovery control. The architecture is designed to be compact, with a series of components, to handle protection decisions based on the current state of the workload and resources without introducing an unnecessarily complicated control structure. The 265 study that produced instances revealed a clear evolution from low adaptation with low logical fidelity to greater adaptation with fidelity values closer to the high end of the graph. The processing-cycle comparison also favored coordinated control, with integrated behavior increasing from 75 to 90 and baseline correction from 68 to 74. Table 1 showed that output reliability increased from 72 under constrained load to 96 under adaptive peak. Output reliability increased from 70 under baseline control to 93 under integrated control. These results show that feedback and coordination across these four processes are more beneficial for scheduling, coherent processing, mitigation and recovery. The proposed design thus provides a feasible way forward for reliable quantum workflows with scarce and/or fluctuating resources. Future validation on real platforms can expand the framework to include a feedback monitor linked to real-platform telemetry, platform-specific mitigation services, and workload-aware execution managers.

6.1. Limitations

The study has limitations to consider when interpreting the results. This data set was created specifically for controlled evaluation; it was not collected from a commercial or laboratory quantum processor. Sharp calibration changes, hardware-specific coupling effects, disturbances not adequately covered here, queue delays, and control restrictions not addressed here may occur. The architecture is designed to be compact and not to represent every possible protection layer; detailed hardware services are available. The evaluation also compares behavior within the proposed framework, rather than directly comparing it with an external platform or published implementations. The numeric data in the graphs and Tables 1 and 2 reflect the internal trends and cannot be considered as “benchmark” claims for a specific device family. Moreover, the feedback monitor is not depicted as a hardware-specific control interface, but rather at a functional level. Such restrictions do not affect the noted adaptive behavior. Still, they imply that, in practice, deployment would require platform calibration and integration with live telemetry, with testing under device-specific restrictions.

6.2. Future Scope

Future research can extend to real quantum hardware and larger hybrid processing environments. The scheduler could respond to calibration changes, queue pressure, operation quality and recovery results from real device telemetry during execution if there were a direct link between the feedback monitor and the real device telemetry. The mitigation component may also include platform-specific techniques based on the workload's sensitivity and the current hardware condition. Another helpful suggestion is to synchronize the architecture with the hybrid classical control system, so resource planning, mitigation selection and recovery choices can be coordinated across both processing environments. Future large-scale studies should incorporate hardware traces from various device families and workloads of differing circuit depth, workload density and sensitivity. The framework could also be extended to multi-user scheduling, in which the reliability decision should account for fairness and pressure on shared resources. Lastly, future studies should investigate how quickly feedback can alter protection decisions

without inducing excessive control delay. These extensions would move the current controlled design toward practical deployment without breaking the key concept of resource-aware, feedback-driven fault tolerance.

Acknowledgment: N/A

Data Availability Statement: The data for this study are available upon request from the corresponding author.

Funding Statement: This manuscript and research paper were prepared without any financial support or funding.

Conflicts of Interest Statement: The authors have no conflicts of interest to declare. This work represents a new contribution by the authors, and all citations and references are appropriately included based on the information utilized.

Ethics and Consent Statement: This research adheres to ethical guidelines and obtains informed consent from all participants. Confidentiality measures were implemented to safeguard participant privacy.

References

1. R. Chao and B. W. Reichardt, "Flag fault-tolerant error correction for any stabilizer code," *PRX Quantum*, vol. 1, no. 1, pp. 1–6, 2020.
2. C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, "Topological and subsystem codes on low-degree graphs with flag qubits," *Physical Review X*, vol. 10, no. 1, pp. 1–19, 2020.
3. W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush, and J. R. McClean, "Virtual distillation for quantum error mitigation," *Physical Review X*, vol. 11, no. 4, pp. 1–25, 2021.
4. N. Ezzell, B. Pokharel, L. Tewala, G. Quiroz, and D. A. Lidar, "Dynamical decoupling for superconducting qubits: A performance survey," *Physical Review Applied*, vol. 20, no. 6, pp. 1–42, 2023.
5. Y. Ji and I. Polian, "Synergistic dynamical decoupling and circuit design for enhanced algorithm performance on near-term quantum devices," *Entropy*, vol. 26, no. 7, pp. 1–20, 2024.
6. Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," *Nature Physics*, vol. 17, no. 7, pp. 1013–1017, 2021.
7. M. T. West, S. M. Erfani, C. Leckie, M. Sevior, L. C. L. Hollenberg, and M. Usman, "Benchmarking adversarially robust quantum machine learning at scale," *Physical Review Research*, vol. 5, no. 2, pp. 1–19, 2023.
8. M. T. West, A. C. Nakhl, J. Heredge, F. M. Creevey, L. C. L. Hollenberg, M. Sevior, and M. Usman, "Drastic circuit depth reductions with preserved adversarial robustness by approximate encoding for quantum machine learning," *Intelligent Computing*, vol. 3, no. 9, pp. 1–11, 2024.
9. A. Khatun and M. Usman, "Q9900uantum transfer learning with adversarial robustness for classification of high-resolution image datasets," *Advanced Quantum Technologies*, vol. 8, no. 1, pp. 1–15, 2024.
10. Y. Kim, M. Sevior, and M. Usman, "Transversal CNOT gate with multicycle error correction," *Physical Review Applied*, vol. 23, no. 2, pp. 1–32, 2025.
11. V. R. Pascuzzi, A. He, C. W. Bauer, and W. A. de Jong, "Computationally efficient zero-noise extrapolation for quantum-gate-error mitigation," *Physical Review A*, vol. 105, no. 4, pp. 1–9, 2022.
12. K. Kubo, Y. Ho, and H. Goto, "High-performance multiqubit system with double-transmon couplers: Toward scalable superconducting quantum computers," *Physical Review Applied*, vol. 22, no. 2, pp. 1–21, 2024.
13. D. Gottesman, "Theory of fault-tolerant quantum computation," *Physical Review A*, vol. 57, no. 1, pp. 127–137, 1998.
14. P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical Review A*, vol. 52, no. 4, pp. R2493–R2496, 1995.
15. A. M. Steane, "Simple quantum error-correcting codes," *Physical Review A*, vol. 54, no. 6, pp. 4741–4751, 1996.
16. A. M. Steane, "Error correcting codes in quantum theory," *Physical Review Letters*, vol. 77, no. 5, pp. 793–797, 1996.
17. E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4452–4505, 2002.
18. K. Khodjasteh and L. Viola, "Dynamical quantum error correction of unitary operations with bounded controls," *Physical Review A*, vol. 80, no. 3, pp. 1–19, 2009.
19. A. Y. Kitaev, "Quantum computations: Algorithms and error correction," *Russian Mathematical Surveys*, vol. 52, no. 6, pp. 1191–1249, 1997.

Publisher's Note: *The publisher remains impartial concerning jurisdictional claims in published maps and institutional affiliations. Responsibility for the content rests entirely with the authors and does not necessarily reflect the publisher's perspectives.*