

Enhancing System Resilience Through Decentralized Control Loops: Separating Bounded Fault Tolerance from Network-Level Resilience

Andrey Valerievich Kiselev *

Independent Expert in Artificial Intelligence and Digital Transformation, TechnoVector Advanced Engineering Research Center, Novosibirsk, Russia

*Correspondence: Andrey Valerievich Kiselev (a.kiselev.av@gmail.com)

Abstract: Decentralized control loops are widely credited with improving system resilience, on the reasoning that removing a central point of coordination also removes a central point of vulnerability. This article separates two properties that applied literature on decentralized architecture routinely conflates: bounded fault tolerance, the capacity of a control architecture to remain stable when an individual loop or node is switched off or corrupted in isolation, and network-level resilience, the capacity to recover from correlated disturbances such as communication loss, targeted compromise, or cascading propagation across dependency chains. Reviewing control-theoretic stability results alongside consensus and infrastructure-resilience research shows that the first property follows from specific, testable structural conditions: a steady-state gain condition in one case, a minimum-neighbor redundancy count in another. These conditions come in different strengths depending on whether the disturbance is benign disconnection or active compromise, a distinction the second property depends on and that decentralization leaves for the designer to resolve. The analysis concludes that a resilience claim attached to a decentralized control loop is a claim about a specific structural parameter matched to a stated disturbance model, a parameter whose verification requires evidence beyond the topology itself.

Keywords: Decentralized Control, Distributed Control, System Resilience, Fault Tolerance, Consensus Algorithms, Cascading Disruption

How to cite this paper:

Kiselev, A. V. (2026). Enhancing System Resilience Through Decentralized Control Loops: Separating Bounded Fault Tolerance from Network-Level Resilience. *International Journal of Mathematical, Engineering, Biological and Applied Computing*, 4(1), 1-8.
DOI: [10.31586/ijmebac.2026.6869](https://doi.org/10.31586/ijmebac.2026.6869)

Received: August 16, 2026

Revised: September 22, 2026

Accepted: October 7, 2026

Published: October 8, 2026



Copyright: © 2026 by the author. Submitted for possible open-access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A decentralized control loop proven stable against the loss of one actuator is proven against exactly that disturbance. The loss of the communication link that would have let it coordinate with its neighbors constitutes a separate condition, requiring a separate guarantee. Campo and Morari (1994) [1] proved that a decentralized controller can be constructed so that a stable multivariable process retains closed-loop stability under the complete or partial outage of any subset of its control loops, a guarantee that depends entirely on a testable property of the process's steady-state gain matrix and covers loop outages specifically; a corrupted measurement, or a disturbance correlated across several loops at once, constitutes a different disturbance, positioned outside that guarantee's scope. Consensus-based distributed control extends the same intuition to networked systems under an explicit and stronger redundancy condition. LeBlanc, Zhang, Koutsoukos, and Sundaram (2013) [2] show that a network of agents using purely local update rules reaches consensus in the presence of compromised nodes precisely when every normal node retains at least $2F+1$ neighbors, where F is the number of nodes an adversary can control, a structural requirement engineered separately from the decision to decentralize.

This distinction is where the applied and practitioner-facing literature on decentralized architecture tends to lose precision. Mikhailiuk (2026) [3] situates feedback-loop instability within a broader taxonomy of engineering risk, classifying it as a distinct category defined by delayed or inaccurate sensing that produces oscillation and delayed correction, and identifying its onset as latent: the instability remains undetected until an operational threshold is crossed, at which point it appears abruptly. Framed this way, the practical question a decentralized architecture has to answer concerns its behavior under the correlated disturbance that arrives when a communication link, a shared assumption, or several loops go down together, a property whose structural condition is stronger than the one the results above establish for an isolated component outage. Existing treatments of decentralization as a resilience strategy conflate these two questions: they attribute to the topology itself a guarantee that in the underlying control-theoretic and consensus literature attaches to a specific, separately engineered redundancy parameter, a parameter whose verification requires evidence external to the architecture's name.

2. Literature Review

The earliest and most rigorously specified account of decentralized fault tolerance comes from process control, where the shared assumption across a cluster of results defines the relevant disturbance as a loop switched off, fully or partially; corruption or manipulation of the loop's signal occupies a separate category, outside this model. Grosdidier and Morari (1986) [4] established the interaction measures that determine whether pairing a given input with a given output in a multi-loop controller avoids destabilizing cross-coupling, giving the field its first quantitative tool for choosing a decentralized structure that keeps its internal interactions bounded. Campo and Morari (1994) [1] built directly on that foundation, deriving necessary and sufficient conditions, expressed entirely through the process's steady-state gain matrix, under which a decentralized controller remains stable for every combination of its loops taken out of service, a property since known as decentralized unconditional stability. Because the gain-matrix condition becomes difficult to verify for processes whose steady-state behavior is only critically stable, Zhang, Bao, and Lee (2002) [5] reformulated the same guarantee through the passivity theorem, trading a matrix computation for a sector-bound condition that extends unconditional stability to a wider class of processes and preserves the guarantee's original strength. Bao, Zhang, and Lee (2003) [6] then extended the reachable class of processes further, from stable to unstable, by combining a redundant multi-loop proportional stabilizer with a passivity-based decentralized unconditionally stabilizing controller, so that fault tolerance extended to processes that begin unstable. Bakule's (2008) [7] survey situates all four results within a single research program spanning several decades of decentralized control theory. The survey's scope makes visible a boundary shared by the whole research program: every guarantee it catalogues, from Grosdidier and Morari's interaction measures through Bao, Zhang, and Lee's unstable-process extension, is proven against the same disturbance model, a loop that is either functioning or entirely absent. Correlated false information across several loops at once occupies a distinct disturbance class, the class that separates isolated-outage tolerance from resilience to a coordinated event.

Distributed consensus-based control confronts part of that disturbance class directly. The two variants of resilience it delivers rest on two different structural conditions, conditions the applied literature routinely collapses into one. Simpson-Porco et al. (2015) [8] demonstrated that distributed averaging, in which each distributed generation unit corrects its voltage and frequency reference using only measurements exchanged with its immediate neighbors, restores the shared secondary-control reference across an islanded microgrid through neighbor-to-neighbor correction alone, provided the communication graph connecting the units stays connected, the ordinary requirement for any consensus protocol to converge. Mikhailiuk (2026) [3] documents the operational mechanism behind

this connectivity requirement: when a network partition disrupts the graph, each node continues processing on its own local state, and the resulting divergence between nodes calls for a reconciliation process once communication is restored, a process that consumes resources and introduces a transient period of instability into the system, which is exactly the cost a consensus protocol pays for the connectivity Simpson-Porco et al. specify as a precondition for convergence. Bidram, Davoudi, Lewis, and Guerrero (2013) [9] extended this neighbor-based correction to nonlinear generator dynamics through feedback linearization under the same connectivity requirement, and the subsequent survey by Bidram, Lewis, and Davoudi (2014) [10] formalizes the design pattern in graph-theoretic terms, representing each distributed generation unit as a node whose secondary-control objective is reached through the local-neighbor update rules that multiagent consensus theory analyzes; this representation lets a connectivity condition proven for abstract agents transfer directly to microgrid hardware. Feng and Ma (2022) [11] extended that same connectivity-dependent pattern to a disturbance affecting the communication network directly, degradation of the network linking the generation units themselves, and designed a distributed fault-tolerant secondary voltage-recovery method built on a sliding-mode surface that keeps the connectivity requirement satisfied under that degradation. These four results are built and tested against a node that goes silent; a node that actively sends false information constitutes a separate disturbance, the one LeBlanc, Zhang, Koutsoukos, and Sundaram (2013) [2] formalize: a distributed consensus protocol reaches agreement in the presence of F compromised nodes precisely when every normal node maintains at least $2F+1$ functioning neighbors, a substantially stronger requirement than mere graph connectedness. The microgrid results above answer a milder disturbance, and their design meets the weaker requirement that disturbance calls for. Gupta and Vaidya (2020) [12], working on distributed optimization instead of consensus, derive a comparable minimum-redundancy characterization for tolerating a bounded number of faulty or adversarial agents. Their result places the same gap between ordinary connectivity and adversarial-grade redundancy in distributed optimization as well as consensus, evidence that the gap constitutes a general property of correlated fault tolerance across algorithm classes.

The cost side of this redundancy requirement becomes visible once distribution is evaluated as a design trade with its own price: removing a single point of vulnerability at the controller level relocates the coordination requirement into the communication layer. Mikhailiuk (2026) [3] traces the same relocation in a different domain: replication mechanisms in distributed data-processing systems contain the propagation of a local malfunction and sustain system performance under partial disruption, and network segmentation in security architecture limits the lateral movement that follows a breach and constrains the resulting economic loss. In both cases the resilience gain traces to a specific structural mechanism, redundant processing capacity in the first case and restricted connectivity in the second, each engineered into the system as a design choice separate from the decision to decentralize. Christofides, Scattolini, Muñoz de la Peña, and Liu's (2013) [13] tutorial review of distributed model predictive control describes the mechanism behind a comparable relocation: a centralized optimization problem is decomposed into local subproblems that individual controllers solve using state or trajectory information exchanged with neighboring controllers at each sampling interval. This decomposition moves the single point of vulnerability a centralized optimizer represents into that inter-controller exchange: closed-loop performance across the network now depends on the exchange completing on schedule, a dependency that replaces reliance on any one controller's own computation. Hespanha, Naghshtabrizi, and Xu (2007) [14] supply the mechanism that makes this relocated dependency destabilizing: a networked control loop retains closed-loop stability only when the network-induced delay stays below a threshold set by the rate at which the controlled process itself evolves. Once delay exceeds that threshold, the control signal lags the process state, and the

resulting phase mismatch produces oscillation in place of correction, a requirement a distributed scheme must satisfy repeatedly, at every coordination step. Mikhailiuk (2026) [3] extends this same delay-stability logic from a single control loop to a layered architecture: fast, near-real-time feedback loops keep local parameters within bounds; slower feedback loops govern system-level reallocation. He argues that coherence between the two layers depends on synchronizing their respective response times with the dynamics they are meant to track.

A separate strand of evidence tests whether the resilience credited to decentralization survives contact with practice. Helmrich et al. (2021) [15], reviewing infrastructure resilience at the governance and physical-network level, found that centralized, decentralized, and distributed configurations each aligned with resilience capacities under different conditions of stability and disruption. Kegeleirs and Birattari (2025) [16] supply a concrete instance of this same pattern: decentralized swarm-robotics control policies validated in simulation, or on one physical platform, lose their fault-tolerance properties when moved to a different platform or to field-scale deployment, a limitation the authors term the deployment gap and distinguish from the simulation-to-reality gap. Bessone, Stoy, and Zahadat (2025) [17] report a result that measures decentralized fault tolerance directly against a centralized baseline: their locally-interacting modular manipulation surface matches a centralized controller's positioning accuracy, orientation alignment, and robustness to actuator outage using only local sensing, a genuine demonstration of parity. The disturbance tested is a single actuator dropout inside a simulated, non-adversarial environment, the same bounded disturbance class the process-control results above were proven against.

Table 1 summarizes the structural conditions identified across these literatures.

Table 1. Structural conditions for resilience in decentralized control. Each condition is valid only for the disturbance model against which it was derived.

Property	Disturbance model	Structural condition	Verification evidence	Key sources
Bounded fault tolerance	Isolated loop outage	Steady-state gain condition	Gain matrix / passivity / sector-bound test	Grosdidier and Morari (1986); Campo and Morari (1994); Zhang et al. (2002); Bao et al. (2003)
Consensus under silent node	Node or link goes silent	Graph connectivity	Communication graph remains connected	Simpson-Porco et al. (2015); Bidram et al. (2013, 2014)
Consensus under compromise	Up to F compromised nodes	$2F+1$ neighbors per normal node	Neighbor-redundancy verification	LeBlanc et al. (2013)
Distributed optimization under faults	Bounded faulty or adversarial agents	Minimum redundancy condition	Redundancy characterization	Gupta and Vaidya (2020)
Networked control stability	Network-induced delay	Delay margin	Delay threshold relative to process dynamics	Hespanha et al. (2007)
Distributed MPC	Inter-controller exchange delay	Sampling-interval completion	Timing verification	Christofides et al. (2013)

3. Discussion

The tension the preceding review exposes concerns the specific structural conditions under which decentralization delivers resilience: a steady-state gain condition for isolated outages, a $2F+1$ neighbor count for correlated compromise, a delay margin set by process dynamics for networked timing. Bessone, Stoy, and Zahadat (2025) [17] conclude that fully decentralized heuristics match centralized control in effectiveness and offer superior scalability and resilience, a conclusion drawn from a manipulation-surface experiment that measures robustness to a single actuator dropout in a setting confined to that single, non-adversarial, uncorrelated disturbance. Campo and Morari's (1994) [1] unconditional-stability condition already predicts this outcome for exactly that disturbance class: a properly designed decentralized controller tolerates the loss of an isolated actuator. LeBlanc et al.'s (2013) [2] necessary-and-sufficient condition addresses a harder disturbance class: once the disturbance model includes even a small number of compromised nodes, a decentralized local-rule system requires a guaranteed $2F+1$ neighbor count to keep its consensus guarantee, since the condition operates as a threshold, and the guarantee drops away entirely below it. Extending Bessone and colleagues' parity result from their tested disturbance class to the correlated one requires verifying the connectivity redundancy their experimental setting left outside its scope, and this verification constitutes the specific structural step that determines the generality of the result.

Mikhailiuk's (2026) [3] layered account of control-loop coherence rests on synchronizing the response times of fast local loops and slower system-level loops, and this synchronization functions as a necessary condition for stability, one the timing-focused results in the literature review corroborate. Temporal alignment between loop layers prevents the oscillatory instability that Hespanha, Naghshtabrizi, and Xu (2007) [14] derive from excess delay, a condition that becomes relevant once a loop is already unconditionally stable under the outage model Campo and Morari (1994) [1] formalize. A correlated or adversarial event calls for a separate condition: the connectivity redundancy that keeps that same loop functioning, the condition LeBlanc et al. (2013) [2] formalize for consensus and Gupta and Vaidya (2020) [12] extend to distributed optimization, the actual determinant of resilience under that disturbance class. Mikhailiuk's own risk taxonomy keeps the two disturbance categories analytically separate: it classifies feedback-delay oscillation as control risk and coupling-driven cascading disruption as dependency risk, two categories with two different propagation signatures. The layered-timing framework built on top of that taxonomy proposes synchronizing response times as the remedy for both categories.

A structural condition weaker than the disturbance the claim is later invoked against produces a measurable decline: the same architecture, performing exactly as designed under one disturbance model, exhibits reduced performance under another, with the architecture itself unchanged. Simpson-Porco et al.'s (2015) [8] distributed-averaging result and Bidram et al.'s (2013, 2014) [9,10] feedback-linearized extension of it recover shared voltage and frequency references after a node or link goes silent because the communication graph they specify stays connected, exactly what a benign-disruption claim requires. A node that sends false data instead of going silent calls for a separate and stronger condition, LeBlanc et al.'s (2013) [2] $2F+1$ requirement, a condition the microgrid results leave untested. Christofides et al.'s (2013) [13] account of distributed model predictive control shows a comparable substitution from a different angle: decomposing a centralized optimization problem into locally solved subproblems relocates the single point of vulnerability a centralized optimizer represents into the inter-controller exchange, which then has to complete within the sampling interval Hespanha, Naghshtabrizi, and Xu (2007) [14] bound. The tutorial review's architecture depends on that timing; guaranteeing it constitutes a separate design task the review leaves to the implementer.

Helmrich et al.'s (2021) [15] review of the infrastructure literature finds that the preference for decentralized configurations rests primarily on assertion, with demonstration against a specified disturbance appearing as the exception.

Table 2 and Figure 1 summarize the analytical distinction and the design sequence it implies.

Table 2. Distinguishing bounded fault tolerance from network-level resilience.

Dimension	Bounded fault tolerance	Network-level resilience
Disturbance scope	Isolated loop or node	Correlated, targeted, or cascading disturbance
Typical disturbance	Loop outage, actuator dropout	Communication loss, compromise, cascading failure
Structural condition	Gain condition, connectivity, local redundancy	2F+1 redundancy, delay margin, layered timing coherence
Verification evidence	Process model, gain matrix, graph connectivity	Adversarial model, redundancy count, timing analysis
Risk of conflation	Overstating resilience from topology alone	Missing stronger condition required by correlated disturbance

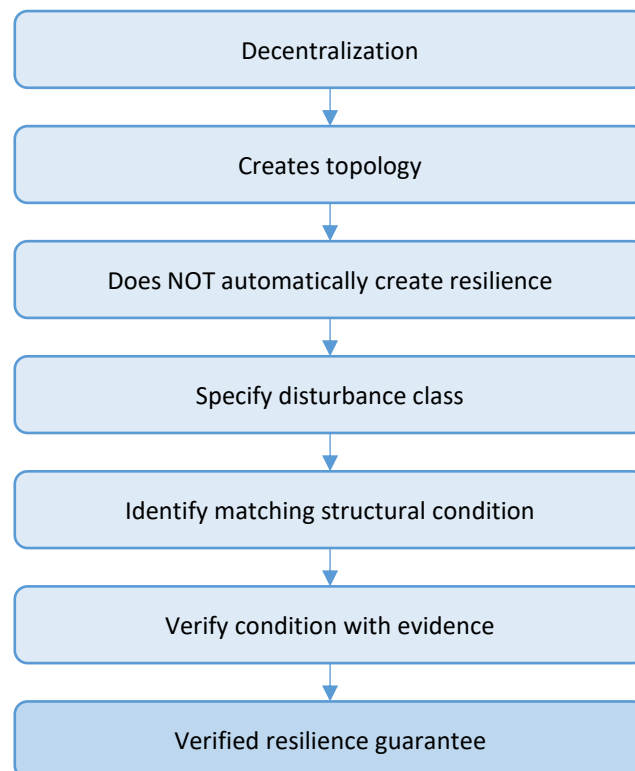


Figure 1. From topology to verified resilience.

4. Conclusion

Decentralized control loops confer resilience as a function of a specific structural parameter: a gain-matrix condition for isolated outages, ordinary graph connectivity for a node or link that goes silent, a minimum-neighbor redundancy count for a node that sends false information, a delay margin for networked timing. Each parameter requires matching to a stated disturbance model and verification through evidence specific to that model; the absence of a central controller leaves each of them to be established through separate engineering evidence. Control-theoretic results stay precise about the disturbance class each one covers. Applied frameworks that distinguish types of operational risk in principle gain analytical value once that distinction carries through into a matching structural requirement for each type. The practical implication for a designer of a decentralized control architecture is a reversal of the usual order of justification. The disturbance class the system must survive requires specification first: a silent node, a lying node, a delayed network. The chosen degree of decentralization then requires evaluation against the specific redundancy condition that disturbance class demands. Decentralization, as an architectural choice, establishes the topology; the structural parameter that turns that topology into a verified resilience guarantee remains a separate engineering decision, made once and specific to each disturbance class the system is expected to survive.

References

- [1] Campo, P. J., & Morari, M. (1994). Achievable closed-loop properties of systems under decentralized control: Conditions involving the steady-state gain. *IEEE Transactions on Automatic Control*, 39(5), 932–943.
- [2] LeBlanc, H. J., Zhang, H., Koutsoukos, X., & Sundaram, S. (2013). Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4), 766–781. <https://doi.org/10.1109/JSAC.2013.130413>
- [3] Mikhailiuk, M. (2026). *Engineering the future: Practical innovation architecture*. LAP LAMBERT Academic Publishing.
- [4] Grosdidier, P., & Morari, M. (1986). Interaction measures for systems under decentralized control. *Automatica*, 22(3), 309–319. [https://doi.org/10.1016/0005-1098\(86\)90029-4](https://doi.org/10.1016/0005-1098(86)90029-4)
- [5] Zhang, W. Z., Bao, J., & Lee, P. L. (2002). Decentralized unconditional stability conditions based on the passivity theorem for multi-loop control systems. *Industrial & Engineering Chemistry Research*, 41(6), 1569–1578. <https://doi.org/10.1021/ie001037v>
- [6] Bao, J., Zhang, W. Z., & Lee, P. L. (2003). Decentralized fault-tolerant control system design for unstable processes. *Chemical Engineering Science*, 58(22), 5045–5054. <https://doi.org/10.1016/j.ces.2003.08.009>
- [7] Bakule, L. (2008). Decentralized control: An overview. *Annual Reviews in Control*, 32(1), 87–98. <https://doi.org/10.1016/j.arcontrol.2008.03.004>
- [8] Simpson-Porco, J. W., Shafiee, Q., Dörfler, F., Vasquez, J. C., Guerrero, J. M., & Bullo, F. (2015). Secondary frequency and voltage control of islanded microgrids via distributed averaging. *IEEE Transactions on Industrial Electronics*, 62(11), 7025–7038. <https://doi.org/10.1109/TIE.2015.2436879>
- [9] Bidram, A., Davoudi, A., Lewis, F. L., & Guerrero, J. M. (2013). Distributed cooperative secondary control of microgrids using feedback linearization. *IEEE Transactions on Power Systems*, 28(3), 3462–3470. <https://doi.org/10.1109/TPWRS.2013.2247071>
- [10] Bidram, A., Lewis, F. L., & Davoudi, A. (2014). Distributed control systems for small-scale power networks: Using multiagent cooperative control theory. *IEEE Control Systems Magazine*, 34(6), 56–77. <https://doi.org/10.1109/MCS.2014.2350571>
- [11] Feng, Y., & Ma, J. (2022). Controller design for distributed secondary voltage restoration in the islanded microgrid. *Frontiers in Energy Research*, 10, Article 826921. <https://doi.org/10.3389/fenrg.2022.826921>
- [12] Gupta, N., & Vaidya, N. H. (2020). Fault-tolerance in distributed optimization: The case of redundancy. In *Proceedings of the 39th ACM Symposium on Principles of Distributed Computing* (pp. 365–374). Association for Computing Machinery. <https://doi.org/10.1145/3382734.3405748>
- [13] Christofides, P. D., Scattolini, R., Muñoz de la Peña, D., & Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51, 21–41. <https://doi.org/10.1016/j.compchemeng.2012.05.011>
- [14] Hespanha, J. P., Naghshtabrizi, P., & Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1), 138–162. <https://doi.org/10.1109/JPROC.2006.887288>
- [15] Helmrich, A., Markolf, S., Li, R., Carvalhaes, T., Kim, Y., Bondank, E., Natarajan, M., Ahmad, N., & Chester, M. (2021). Centralization and decentralization for resilient infrastructure and complexity. *Environmental Research: Infrastructure and Sustainability*, 1(2), Article 021001. <https://doi.org/10.1088/2634-4505/ac0a4f>

-
- [16] Kegeleirs, M., & Birattari, M. (2025). Towards applied swarm robotics: Current limitations and enablers. *Frontiers in Robotics and AI*, 12, Article 1607978. <https://doi.org/10.3389/frobt.2025.1607978>
 - [17] Bessone, N., Stoy, K., & Zahadat, P. (2025). Swarm-inspired controllers: A comparative study of decentralized behaviors for distributed manipulation surfaces. *Swarm Intelligence*, 19, 273–293. <https://doi.org/10.1007/s11721-025-00252-3>