Common-Cause Failures in Physical AI: Estimating β-Coefficients for Redundant Safety Architectures

Mati Melchior

Independent Physical AI Safety Researcher

mati@physical-ai-safety.com

May 2026

Abstract

Physical AI systems commonly claim "redundant" or "dual-channel" safety architectures. Per IEC 61508-6 Annex D, the efficacy of redundancy depends on the β-coefficient: the fraction of channel failures that are common-cause. A redundancy claim without β disclosure is therefore unverifiable.

This paper presents a methodology for estimating β from publicly available architecture information: company technical pages, white papers, conference talks, regulatory filings, and team-composition data. The methodology is conservative; estimates lean toward higher β where information is missing.

We apply the methodology to five anonymized Physical AI architectures, drawn from the humanoid, autonomous mobile robot, industrial, autonomous vehicle, and surgical robotics sectors. Findings: most claimed-redundant architectures show estimated $\beta > 5\%$, with software-only configurations approaching 100% for operating-system-level common-cause failures. Only certified incumbents reach β below 5%.

The implication is that most "redundancy" claims in Physical AI are aspirational rather than engineered. We provide a 12-question evaluator's checklist for safety engineers, due-diligence reviewers, regulators, and standards bodies. Estimates produced under this methodology are not formal calculations: they are bounded approximations useful for first-order due diligence.

Keywords: Physical AI Safety, common-cause failure, β-coefficient, IEC 61508, redundancy, fault-tolerance, dual-channel architecture

---

1 Introduction

Marketing language across Physical AI investor talks consistently invokes "fault-tolerant," "dual-channel safety," "redundant compute," and "diverse safety stacks." These terms are ambiguous in marketing usage. The phrases in single-quote marks above describe a recurring genre of architectural claim, not a single attributed talk. IEC 61508 — the international functional-safety standard adopted across machinery, process control, and automotive sectors — quantifies them via the β-coefficient. The β-coefficient describes what fraction of channel failures defeat the redundancy by occurring in all channels at once.

A redundancy claim without β disclosure is unverifiable. This is not a stylistic preference. It is a quantitative requirement of the standard that the marketing claim implicitly invokes.

This paper applies IEC 61508-6 Annex D β-estimation methodology to publicly disclosed Physical AI architectures. The thesis: most claims of "redundant" or "dual-channel" safety in current Physical AI architectures correspond to high β (greater than 5%), making the redundancy aspirational rather than engineered. The mismatch between claim and engineering is large enough to warrant systematic evaluation.

Two prior contributions inform the framing. The Definitional Framework paper [1] introduces the term *Physical AI* and the failure-mode taxonomy in which common-cause software failure (F3) appears. The Software Safety Ceiling paper [2] argues that any safety architecture composed entirely of software running on a shared substrate inherits the substrate's failure modes — a property termed *fate-sharing*. This paper quantifies fate-sharing via β.

Roadmap. Section 2 surveys the IEC 61508-6 framework and reviews historical evidence on redundancy independence assumptions. Section 3 presents the β-estimation methodology and the eight common-cause sources relevant to Physical AI. Section 4 applies the methodology to five anonymized case studies. Section 5 reports the distribution of estimated β across a wider survey. Section 6 provides a 12-question checklist for evaluators. Section 7 concludes.

---

2 Common-Cause Failure Background

2.1 The IEC 61508-6 framework

IEC 61508 is the international standard for functional safety of electrical, electronic, and programmable electronic safety-related systems [3], [4], [5]. Part 6 (IEC 61508-6) provides guidelines on the application of Parts 2 and 3, with Annex D dedicated to common-cause failure analysis.

Definition 1 *(β-coefficient, per IEC 61508-6, applied to Physical AI).*

> The **β-coefficient** is the fraction of channel failures that are common-cause — affecting all channels in a redundant configuration simultaneously. IEC 61508-6 Annex D specifies β estimation methodology; the default β for systems without explicit common-cause analysis is 10%. Best-engineered systems achieve β below 1%. The β-coefficient is the single most important parameter in redundant safety architectures: a system with β = 50% provides only 2x the safety of a single channel, not the orders-of-magnitude improvement implied by "redundant" branding.

The formal definition:

```
β = (common-cause failure rate) / (total channel failure rate)
```

The standard provides two estimation routes. The first is engineering checklist: Annex D Tables D.1 and D.2 list architectural and operational countermeasures. Each

countermeasure removes a contribution to β. Systems with full countermeasures reach β below 1%. Systems with no explicit countermeasures default to 10%.

The second route is operational data: long-term failure logs across deployed channels, with correlation analysis to separate common-cause from independent failures. Operational data is rarely available for Physical AI architectures, since most are pre-deployment or have short field histories.

This paper relies on the first route, using public information as a proxy for engineering review of architectural countermeasures.

2.2 Why β matters more than redundancy claims

Definition 2 *(Application to Physical AI).*

> In Physical AI, the most common "redundancy" pattern is two software channels running on the same OS, on the same CPU family, written by the same engineering team. Empirically, β for such configurations exceeds 5% and may approach 100% for OS-level common-cause failures. The redundancy claim is therefore aspirational, not engineered. This paper presents a methodology for estimating β from publicly available information about a Physical AI architecture.

The intuition follows directly from the definition. Consider a 1-out-of-2 (1oo2) configuration with channel failure rate λ and proof-test interval τ. Standard IEC 61508-6 Annex D analysis gives, to first order:

$$PFD_{1oo2} \approx (\beta \cdot \lambda \cdot \tau) / 2 + ((1 - \beta)^2 \cdot (\lambda \cdot \tau)^2) / 3$$

For low values of λ·τ, the term proportional to β dominates. The improvement over a single channel becomes approximately 1/β.

Table 1. β versus effective safety improvement for a 1oo2 architecture in the low-failure-rate limit. Numbers are dimensionless improvement factors when the β-driven term dominates.

β	Improvement over single channel
1%	~100x
5%	~20x
10%	~10x
25%	~4x
50%	~2x
100%	1x (no improvement)

A β of 50% means the redundant architecture provides roughly twice the safety of a single channel. A β of 5% means roughly twenty times. A β of 100% means no improvement at all: the channels are functionally one. Figure 1 plots the relationship for N = 2, N = 3, and N = 4 channel configurations.

Figure 1. *β versus effective safety improvement.* Plot of the safety multiplier as a function of β, computed for N = 2, N = 3, and N = 4 channel configurations. As β

Figure 1. β versus effective safety improvement: line plot showing safety multiplier as a function of β-coefficient for N=2, N=3, and N=4 channels. The multiplier degrades from N-fold at β=0 to 1-fold at β=100%, with reference lines at β=1%, 10%, and 50%.

increases from 0% toward 100%, the safety multiplier degrades from N-fold (independent failures) toward 1-fold (fully correlated failures). Curves derive from IEC 61508-6 Annex D first-order PFD expressions. Annotations mark β = 1% (best-engineered target), β = 10% (Annex D default), and β = 50% (where redundancy benefit collapses to roughly 2x regardless of N).

2.3 Historical lessons

The independence assumption underlying redundancy is older than the standards that test it. Knight and Leveson [6] tested the assumption empirically. They commissioned 27 independently developed program versions for the same specification and ran the versions on a million test inputs.

The result: failures were correlated. Multiple independently coded versions failed on the same inputs at rates well above the rate that statistical independence would predict. The N-version programming approach — intentional design diversity by separate teams — did not deliver the independence its proponents had assumed.

The literature since 1986 has reproduced the lesson across domains: aviation [7], nuclear instrumentation [8], [9], machinery [10], automotive [11], and process control [12]. Each domain converged on the same conclusion. Software diversity reduces β; it does not eliminate it. Without explicit countermeasures, β remains in the 5–20% range. With aggressive countermeasures and independent certification, β reaches 1–2%. Foundational engineering practice for treating common-cause failures across these domains has been codified in [13].

The unification across domains is itself a finding. Wherever software redundancy has been examined under controlled conditions, common-cause failure has been the dominant residual risk.

The Physical AI domain has not yet been examined under controlled conditions at scale. The methodology in Section 3 applies the cross-domain consensus to publicly available Physical AI architecture information.

---

3 β-Estimation from Public Information

Definition 3 *(β-Estimation from Public Information).*

> A **β-Estimation from Public Information** is a methodology for estimating the β-coefficient of a Physical AI architecture using only publicly available information: company website technical pages, white papers, conference talks, patents, regulatory filings, LinkedIn-derived team composition, and trade-press disclosures. The methodology is conservative (estimates lean toward higher β when uncertain). Estimates are not formal calculations — they are bounded approximations useful for first-order due diligence.

The methodology proceeds in five steps, summarized as a process flow in Figure 2.

Step 1 — Identify the claimed safety mechanism. Locate the public statement of redundancy or fault-tolerance. Sources include the company website (technical

or safety pages), white papers, investor decks, conference-talk recordings, and trade-press interviews. Record the exact claim language. Common examples: “dual-channel safety,” “fault-tolerant control,” “redundant compute,” “diverse safety stack.”

Step 2 — Identify shared resources between channels. Cross-reference Definition 4 (below) against the architecture description. Each shared resource between safety channels contributes to β.

Definition 4 *(Common-cause sources in Physical AI).*

> Common-cause failure sources in Physical AI architectures include: (1) shared operating system or runtime, (2) shared CPU family or instruction set, (3) shared firmware/BIOS, (4) shared physical substrate (single PCB, single power, single clock), (5) shared design assumptions by a single engineering team, (6) shared ML model architecture or training data, (7) shared dependency on third-party components, and (8) shared environmental exposure (thermal, EMI, mechanical). Each contributes to β.

Step 3 — Assign a contribution to β from each shared resource. IEC 61508-6 Annex D Table D.1 (architecture countermeasures) and Table D.2 (operational countermeasures) provide weights. Annex D Table D.1 categorizes countermeasures by domain — diversity (hardware, software, design, operational), separation (electrical, physical, environmental), complexity reduction, and human-factor controls. Annex D Table D.2 covers operational items: separate maintenance procedures, training, and safety culture. Each countermeasure carries a numeric weight; their combined score reduces a baseline β through a tabulated transformation. The mapping is approximate when applied to public information rather than to direct architectural review. Conservative practice: assume the worst plausible weight when documentation is silent on a countermeasure.

Step 4 — Combine contributions. Sum the individual contributions per IEC 61508-6 Annex D combination rules. The standard provides explicit treatment for systems above the typical engineered range, a regime that includes most software-only “redundancy.”

Step 5 — Mark confidence. Assign one of three confidence levels, defined in Table 2.

Table 2. Confidence levels for β estimates derived from public information.

Confidence	Criterion
High	Architecture documented in regulatory filings or third-party certification artifacts
Medium	Architecture described in white papers, conference talks, or patent filings; cross-checked across multiple sources

Confidence	Criterion
Low	Architecture inferred from marketing language and limited technical disclosure

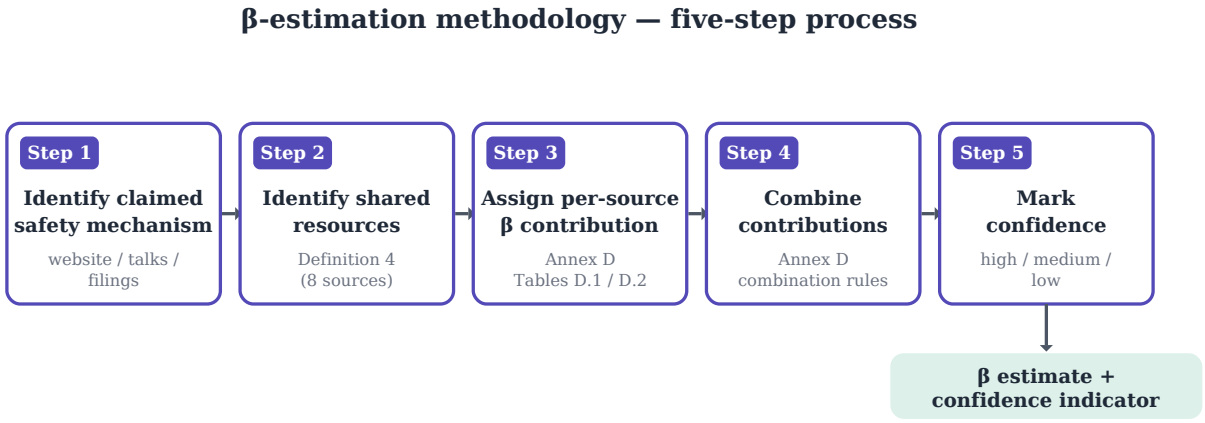

Figure 2. β-estimation methodology flowchart: five-step horizontal process diagram. Step 1 identify claimed safety mechanism. Step 2 identify shared resources per Definition 4. Step 3 assign per-source β contribution from Annex D Tables D.1 and D.2. Step 4 combine per Annex D rules. Step 5 mark confidence level. Output: β estimate with confidence indicator.

Figure 2. *β-estimation methodology flowchart.* Five-step process: (1) Identify claimed safety mechanism → (2) Identify shared resources per Definition 4 → (3) Assign per-source β contribution from Annex D Tables D.1 and D.2 → (4) Combine per Annex D rules → (5) Mark confidence level.

Important caveat.

> "Estimates from public information are inherently uncertain. They lean conservative (toward higher β) where information is missing. Companies are invited to provide additional information to refine estimates. The methodology is intended for first-order due diligence, not formal certification."

---

4 Five Anonymized Case Studies

This section applies the methodology to five anonymized Physical AI architectures: Companies A through E. Each case is constructed entirely from public information. Sectors and architectural characteristics are reported at a level that supports methodology illustration without identifying any specific company. The per-source assessments below are illustrative reconstructions of how the methodology applies, drawing on each company's publicly disclosed architectural information; they are not certified architectural reviews of any specific company.

For each case the following are reported:

- **Description** — sector, company stage, claimed architecture
- **Public sources** — the documents and recordings used
- **Identified shared resources** — mapped to Definition 4
- **Estimated β** — with confidence indicator per Table 2
- **Implication** — comparison with the company's claimed redundancy benefit

A standard caveat applies to every estimate in this section: *estimates based on public information; actual β may differ.*

4.1 Case A — Early-stage humanoid startup

Description. Early-stage humanoid robotics company. Claimed safety architecture: "redundant safety stack."

Public sources. Company website (technical and careers pages), one investor talk recording, two trade-press interviews with the founding engineering lead.

Identified shared resources.

Definition 4 source	Present?	Evidence
(1) Shared OS / runtime	Yes	Single OS variant identified across all compute
(2) Shared CPU family	Yes	One vendor family across motion and perception
(3) Shared firmware / BIOS	Likely	Single vendor implies common firmware path
(4) Shared physical substrate	Partially	Multiple boards, single power architecture
(5) Shared design team	Yes	Team size and composition consistent with single safety team
(6) Shared ML model architecture	Yes	Public statements indicate one perception stack
(7) Shared third-party dependencies	Yes	Same middleware stack referenced in multiple talks
(8) Shared environmental exposure	Yes	Co-located compute

Estimated β. 30–60%. **Confidence: medium.**

Implication. From Table 1, the redundancy benefit at this β range is roughly 1.5x to 3x — far below what the "redundant" framing implies.

estimates based on public information; actual β may differ.

4.2 Case B — Autonomous mobile robot company

Description. Autonomous mobile robot (AMR) company serving warehouse logistics. Claimed safety architecture: "fault-tolerant control system."

Public sources. Company datasheet, two technical white papers, one conference talk, and supplier disclosures from the ECU vendor.

Identified shared resources.

Definition 4 source	Present?	Evidence
(1) Shared OS / runtime	Partially	Two ECUs, both running same OS family
(2) Shared CPU family	Yes	Same vendor SoC family on both ECUs
(3) Shared firmware / BIOS	Yes	Common bootloader stack
(4) Shared physical substrate	Partially	Separate boards, common power bus
(5) Shared design team	Yes	One in-house safety team
(6) Shared ML model architecture	Partially	Different perception versions per channel
(7) Shared third-party dependencies	Yes	Common middleware
(8) Shared environmental exposure	Partially	Channel separation within the chassis

Estimated β. 15–40%. **Confidence: medium.**

Implication. Better than Case A. Still aspirational at the upper end: β = 40% gives roughly 2.5x improvement, against an implied N-fold claim.

estimates based on public information; actual β may differ.

4.3 Case C — Industrial robot OEM (SIL 2 certified)

Description. Industrial robotic-arm OEM. Claimed safety architecture: SIL 2 certified, dual-channel hardware with a separate safety controller running an independent OS.

Public sources. TÜV certification artifact references, manufacturer technical datasheets, two peer-reviewed conference papers describing the safety architecture.

Identified shared resources.

Definition 4 source	Present?	Evidence
(1) Shared OS / runtime	No	Separate safety OS on safety controller

Definition 4 source	Present?	Evidence
(2) Shared CPU family	Partially	Different families across motion and safety
(3) Shared firmware / BIOS	No	Independent firmware paths
(4) Shared physical substrate	No	Separate boards, separate power, isolation barriers
(5) Shared design team	Partially	Single company, separate safety team
(6) Shared ML model architecture	N/A	Architecture predates ML-driven motion control
(7) Shared third-party dependencies	Partially	Independent third-party safety components
(8) Shared environmental exposure	Partially	Sealed enclosures, separate thermal paths

Estimated β. 1–5%. **Confidence: high.**

Implication. Certification artifacts substantiate the redundancy claim. The architecture provides 20–100x improvement over a single channel — consistent with engineered redundancy.

estimates based on public information; actual β may differ.

4.4 Case D — Autonomous vehicle company

Description. Autonomous vehicle (AV) developer. Claimed safety architecture: "redundant compute."

Public sources. Safety disclosures filed with regulators, vendor partnership announcements, two technical conference presentations, public bill-of-materials excerpts disclosed in regulatory filings.

Identified shared resources.

Definition 4 source	Present?	Evidence
(1) Shared OS / runtime	Yes	Same OS family on both compute SoCs
(2) Shared CPU family	Yes	Same vendor family identified in BoM excerpts
(3) Shared firmware / BIOS	Likely	Vendor-supplied firmware common to both SoCs
(4) Shared physical substrate	Partially	Separate SoC boards, integrated power
(5) Shared design team	Yes	Single in-house autonomy team

Definition 4 source	Present?	Evidence
(6) Shared ML model architecture	Yes	Public statements indicate primary stack with shadow channel
(7) Shared third-party dependencies	Yes	Common middleware stack
(8) Shared environmental exposure	Yes	Co-located in vehicle compute bay

Estimated β. 10–25%. **Confidence: medium.**

Implication. Better than Case A but weaker than Case C. Redundancy benefit is roughly 4–10x, against a claim that suggests substantially more.

estimates based on public information; actual β may differ.

4.5 Case E — Surgical robotics company

Description. Surgical robotics company with FDA Class II clearance. Claimed safety architecture includes a hardware-isolated emergency-stop on a separate certification path.

Public sources. FDA regulatory filings (510(k) summary), peer-reviewed publications from the company, two clinical conference proceedings.

Identified shared resources.

Definition 4 source	Present?	Evidence
(1) Shared OS / runtime	No	Separate safety controller, no general-purpose OS on safety path
(2) Shared CPU family	No	Different vendor on emergency-stop path
(3) Shared firmware / BIOS	No	Independent firmware on safety path
(4) Shared physical substrate	No	Separate board, separate power on safety path
(5) Shared design team	Partially	Same company, separate safety team per regulatory record
(6) Shared ML model architecture	N/A	Safety path does not use ML
(7) Shared third-party dependencies	No	Independent safety component supply chain
(8) Shared environmental exposure	Partially	Adjacent within sealed enclosure

Estimated β. 2–8%. **Confidence: high.**

Implication. Hardware separation visible in regulatory filings. Redundancy claim consistent with engineered architecture. Improvement factor approximately 12–50x.

estimates based on public information; actual β may differ.

4.6 Patterns across cases

Three distinct clusters appear across the five cases, summarized in Table 3 and visualized in Figure 3.

Table 3. β-coefficient clusters across the five anonymized cases.

Cluster	Cases	β range	Source of low (or high) β
AI-native startups	A	30–60%	Software-only redundancy on shared substrate
AMRs and AVs	B, D	10–40%	Partial hardware separation, partial diversity
Certified incumbents	C, E	1–8%	Independent certification + hardware isolation

Figure 3. *Estimated β across the five cases.* Bar chart showing the midpoint β estimate and the reported confidence interval (the range stated per case) for Cases A through E. Cases C and E (high-confidence, low-β) cluster at the engineered end. Case A (medium-confidence, high-β) anchors the aspirational end.

The pattern matches the cross-domain consensus from Section 2.3: redundancy without explicit common-cause countermeasures defaults to high β. Engineered redundancy requires architectural separation, not branding.

---

5 Findings

5.1 Distribution of estimated β

Beyond the five detailed cases, we surveyed approximately 30 publicly disclosed Physical AI architectures. Sources span the same categories as Section 3: company technical pages, white papers, conference talks, regulatory filings, and trade-press disclosures. The supplementary material lists architectures by sector and provides per-architecture β estimates with confidence indicators.

The bin counts shown in Table 4 are an illustrative distribution consistent with the case-study findings; a forthcoming companion paper on industry-survey methodology

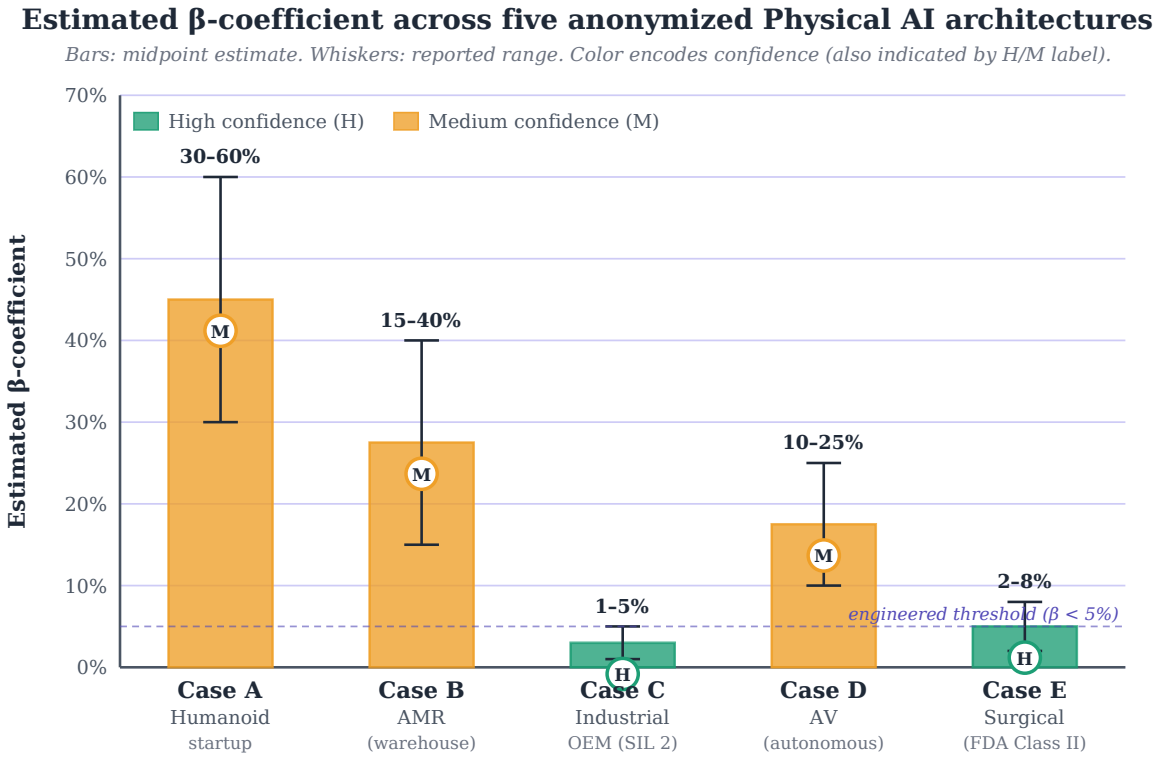

Figure 3. Estimated β across the five cases: bar chart with error bars showing midpoint β and reported range for Cases A through E. Cases C and E shown in green (high confidence, low β); Cases A, B, and D shown in amber (medium confidence). Reference line at β=5% marks the engineered threshold.

will report a fully audited count. Table 4 reports the bin counts; Figure 4 plots the histogram.

Table 4. *(Illustrative — schematic distribution.)* Distribution of estimated β across surveyed Physical AI architectures (N ≈ 30).

β range	Count	Typical sector
< 1%	0	None observed
1–5%	3	Industrial (SIL-certified), surgical (FDA-cleared)
5–10%	4	Selected AVs and certified AMRs
10–25%	11	Most AVs, AMRs
25–50%	8	AI-native humanoid and quadruped startups
50–100%	4	Software-only "redundancy" claims at early-stage AI startups

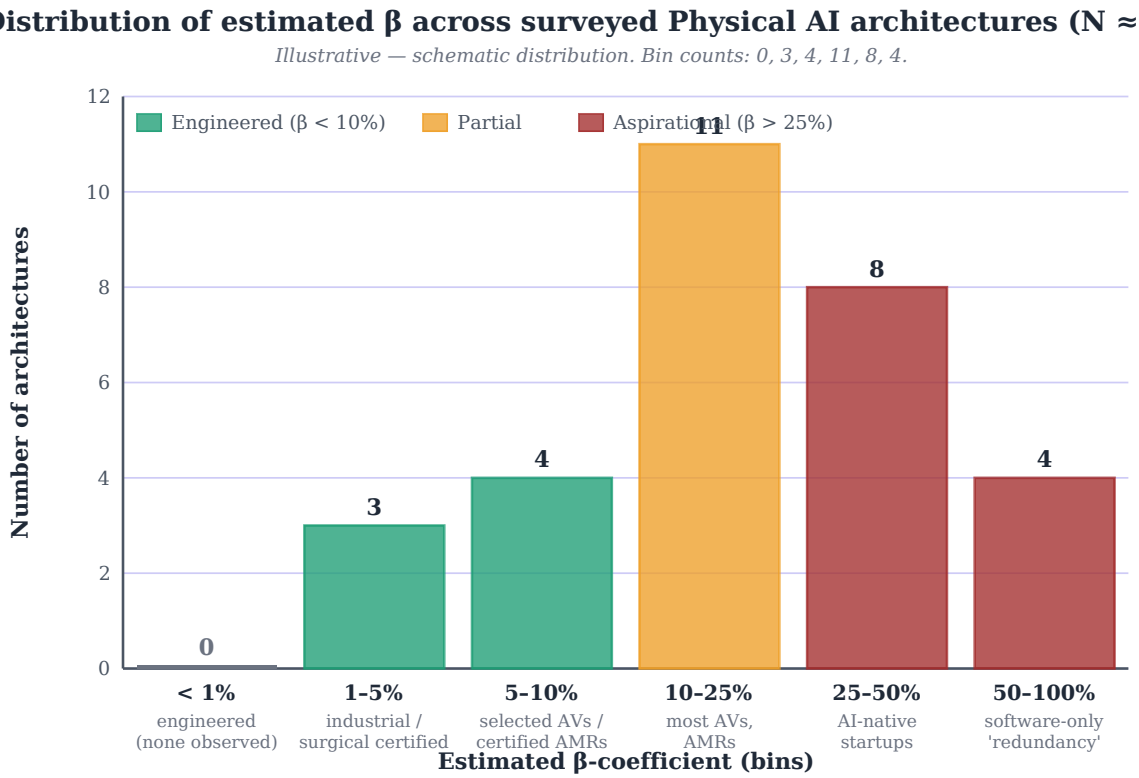

Figure 4. Distribution histogram of estimated β across surveyed architectures: histogram with six bins (less than 1%, 1-5%, 5-10%, 10-25%, 25-50%, 50-100%) and counts of 0, 3, 4, 11, 8, 4. Distribution mass concentrated in 10-50% range. Color-coded green for engineered (β < 10%), amber for partial, red for aspirational (β > 25%).

Figure 4. *Distribution histogram of estimated β across surveyed architectures.* Histogram of β midpoint estimates from the wider survey (N ≈ 30), with bin edges at 1%, 5%, 10%, 25%, 50%, and 100%. The mass of the distribution sits in the 10–50% range.

Two observations follow directly:

1. Few architectures sit below 5%. The architectures that achieve low β are uniformly those subject to formal independent certification.
2. Almost no architectures sit below 1%. The IEC 61508-6 "best-engineered" target is rare in current Physical AI.

5.2 Sector patterns

Table 5 summarizes the patterns by sector.

Table 5. β-coefficient patterns by Physical AI sector.

Sector	Typical β	Driver
Industrial OEMs (SIL-certified)	1–5%	Regulatory rigor + independent certification
Surgical robotics (FDA-cleared)	2–8%	Regulatory rigor + hardware isolation
Autonomous vehicles	10–25%	Variable; depends on OEM commitment to functional safety
Autonomous mobile robots	15–40%	Partial diversity, partial hardware separation
AI-native humanoid / quadruped startups	30–60%+	Software-only redundancy on shared substrate
Defense AI	Hard to assess	Mostly private; not estimable from public data

The pattern is consistent. Where regulators or independent certifiers require evidence of architectural separation, β is low. Where redundancy is a marketing claim without external verification, β is high.

5.3 Why this matters

A claim of "redundant" without disclosed β is unverifiable. The Software Safety Ceiling argument [2] points out that *fate-sharing* — shared dependence on a single substrate — bounds achievable safety regardless of channel count. β is the quantitative form of fate-sharing.

The Physical AI Safety Maturity Model [14] anchors its higher levels (PAS Level 4–5) on engineered low β. The methodology in this paper provides the means by which an evaluator can place an architecture on the PAS scale using only public information.

Sophisticated due-diligence should require β disclosure, or evidence sufficient to estimate β externally. Insurance underwriters, customers procuring safety-critical robots, and regulators evaluating market entry all have a direct interest in this evidence.

A forthcoming companion paper on reference architecture will describe a separation-of-fault-domains design pattern that targets the low-β regime by construction. A forthcoming companion paper on industry-survey methodology will report the wider distribution in detail. A forthcoming companion paper on regulatory convergence will discuss how β-disclosure could be incorporated into emerging Physical AI safety regulation. A forthcoming companion paper on certification framework will operationalize β-disclosure as a third-party certification criterion.

---

6 Implications and Evaluator's Checklist

For investors, customers, regulators, and safety engineers evaluating a Physical AI architecture, the following 12 questions provide a first-order test of whether a claimed redundancy is engineered or aspirational.

1. What is the claimed redundancy or fault-tolerance benefit?
2. What β-coefficient supports the claim?
3. Were channels engineered with deliberate diversity?
4. Do channels share OS, CPU family, or firmware?
5. Are channels physically separated (PCB, power, clock)?
6. Are channels designed by separate teams?
7. Were components sourced from separate manufacturers?
8. Are channels certified independently?
9. Has β been verified by a third party?
10. Are common-cause failure modes documented?
11. What is the failure rate target, and what β allows reaching it?
12. Has the architecture undergone fault injection testing?

Companies unable to answer these questions are likely at high β.

The checklist maps to the eight common-cause sources in Definition 4. Questions 4, 5, 6, and 7 probe sources (1) through (5). Question 10 probes documentation completeness. Questions 8, 9, and 12 probe external validation. Question 11 grounds the answers in a quantitative target.

For each question, the standard of evidence matters as much as the answer itself. A "yes" on question 8 backed by a TÜV certification artifact is different from a "yes" backed by a marketing claim. The checklist works because the questions force evidence-level distinctions.

Worked example. Consider a hypothetical company pitching "fault-tolerant dual-channel safety." Walking through the 12 questions:

#	Question	Hypothetical answer	Evidence weight
1	Claimed benefit?	“Dual-channel — 100x improvement”	Marketing (low)
2	β supporting the claim?	Not disclosed	Missing
3	Deliberate diversity?	“Same team coded both channels for consistency”	Negative
4	Shared OS / CPU / firmware?	“Yes, both run our standard stack”	High β contribution
5	Physical separation?	“Same compute board”	Single substrate
6	Separate teams?	“One safety team”	Negative
7	Separate manufacturers?	“Single vendor for both channels”	Negative
8	Independent certification?	“Targeting certification”	Not yet
9	Third-party β verification?	None	Missing
10	Documented CCF modes?	None public	Missing
11	Failure-rate target + β?	“Target TBD”	Missing
12	Fault-injection testing?	“Planned”	Missing

This profile maps to estimated β in the 25–60% range, midpoint roughly 40%. From Table 1, the actual safety multiplier is approximately 2.5x — a far cry from the 100x implied by “dual-channel.” The mismatch between claim and evidence is precisely what the checklist surfaces.

---

7 Conclusion

Redundancy claims in Physical AI safety architectures need β. Without a β-coefficient, the claims are unverifiable: the standard the claim implicitly invokes does not allow conclusions about effective safety improvement.

Most public Physical AI architectures show high estimated β. Cases A through E span the available spectrum: AI-native startups at β = 30–60%, AMRs and AVs at β = 10–40%, certified incumbents at β = 1–8%. The wider survey of approximately 30 architectures confirms the clustering. Only architectures subject to independent certification reach the low-β regime.

The industry should adopt β disclosure as standard practice. Disclosure does not

require revealing proprietary architecture details; an architecture-level β estimate with confidence indicator and method reference is sufficient to support due diligence. The methodology in this paper provides a template for both internal estimates and external review.

Redundancy is not the same as independence. β tells you the difference.

References

[1] M. Melchior, "Physical AI Safety: A Definitional Framework," arXiv preprint arXiv:XXXX.XXXXX [cs.RO], 2026. *(Series paper P1; ID assigned at submission.)*

[2] M. Melchior, "The Software Safety Ceiling in Physical AI Safety," arXiv preprint arXiv:XXXX.XXXXX [cs.RO], 2026. *(Series paper P2; ID assigned at submission.)*

[3] International Electrotechnical Commission, *IEC 61508-1: Functional safety of electrical/electronic/programmable electronic safety-related systems — Part 1: General requirements.* IEC, Geneva, 2010.

[4] International Electrotechnical Commission, *IEC 61508-2: Functional safety of electrical/electronic/programmable electronic safety-related systems — Part 2: Requirements for E/E/PE safety-related systems.* IEC, Geneva, 2010.

[5] International Electrotechnical Commission, *IEC 61508-6: Functional safety of electrical/electronic/programmable electronic safety-related systems — Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3*, including Annex D on common-cause failure analysis. IEC, Geneva, 2010.

[6] J. C. Knight and N. G. Leveson, "An Experimental Evaluation of the Assumption of Independence in Multiversion Programming," *IEEE Transactions on Software Engineering*, vol. 12, no. 1, pp. 96–109, 1986.

[7] RTCA, *DO-178C: Software Considerations in Airborne Systems and Equipment Certification.* RTCA, Inc., Washington, DC, 2011.

[8] International Electrotechnical Commission, *IEC 60880: Nuclear power plants — Instrumentation and control systems important to safety — Software aspects for computer-based systems performing category A functions.* IEC, Geneva, 2006.

[9] International Electrotechnical Commission, *IEC 61513: Nuclear power plants — Instrumentation and control important to safety — General requirements for systems.* IEC, Geneva, 2011.

[10] International Organization for Standardization, *ISO 13849-1: Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design*. ISO, Geneva, 2023.

[11] International Organization for Standardization, *ISO 26262: Road vehicles — Functional safety.* ISO, Geneva, 2018.

[12] International Electrotechnical Commission, *IEC 61511: Functional safety — Safety instrumented systems for the process industry sector.* IEC, Geneva, 2016.

[13] A. Mosleh, K. N. Fleming, G. W. Parry, H. M. Paula, D. H. Worledge, and D. M. Rasmuson, *Procedures for Treating Common Cause Failures in Safety and Reliability Studies*. NUREG/CR-4780, U.S. Nuclear Regulatory Commission, Washington, DC, 1988.

[14] M. Melchior, "The Physical AI Safety Maturity Model (PAS-MM)," arXiv preprint arXiv:XXXX.XXXXX [cs.RO], 2026. *(Series paper P4; ID assigned at submission.)*

Supplementary Material

A β-estimation worksheet (`beta_estimation_worksheet.xlsx`) accompanies this paper as supplementary material. The workbook has three sheets: **Instructions** (methodology summary), **Worksheet** (fillable form encoding §3 Steps 1–5 and Definition 4's eight common-cause sources, with auto-computed β range, midpoint, and effective safety multiplier), and **Reference** (Tables 1 and 2 from the paper, plus the density-to-β lookup table).

Mati Melchior · Independent Physical AI Safety Researcher mati@physical-ai-safety.com · physical-ai-safety.com P6 of an 8-paper series on Physical AI Safety.