INSTITUTE FOR PHYSICAL AI @ BMI
The Charlot Lab

Technical Report TR-2026-07
Survey / Review · Preprint v1
7 July 2026

PROVABLE BY CONSTRUCTION

# Provable by Construction: On-Device Lyapunov Certificates for Trustworthy Control

A controller is trustworthy not because it passed testing, but because its stability is proven — and the proof can ride with it.

**David Jean Charlot, PhD**

Dean of Physical AI · The Charlot Lab, Institute for Physical AI @ BMI

Correspondence: contact@physicalai-bmi.org · physicalai-bmi.org
Interactive companion: physicalai-bmi.org/research/charlot-lab#topic-lyapunov

**Abstract.** Testing shows that a controller worked on the states it was tried on; it says nothing about the states it was not. A Lyapunov certificate says something stronger: that a closed-loop policy drives an entire region of the state space toward its equilibrium, by exhibiting a scalar energy that the dynamics can only spend. This report reviews how such a certificate is checked soundly and cheaply enough to run on the device it certifies. The check pairs two verifiers. Near the equilibrium, where the linear term of the dynamics provably dominates, the certificate is analytic: a quadratic Lyapunov function from the solution of a matrix equation, valid on a ball whose radius follows from a Taylor remainder bound. Everywhere else the certificate is established by interval branch-and-bound, which encloses the Lyapunov decrease condition over whole boxes of states with outward-rounded arithmetic, splits boxes that remain undecided, and returns either a certified region of attraction or a concrete counterexample state. This is a proof over a region, not a grid of samples. The method is the companion to a previously reported energy receipt: where the receipt prices a decision, the certificate warrants a controller, and the two together let an embodied system carry both its cost and its proof. This report states a research position and reports no original experiments.

## 1. Introduction: tested is not proven

An embodied controller is normally accepted on the strength of evidence. It is run in simulation and on hardware across a battery of initial conditions, disturbances, and tasks; if it recovers every time, it ships. This is testing, and testing has a fixed epistemic ceiling. It reports the behavior of the system at the finite set of states it was actually exercised on. The state space of a robot is continuous and high-dimensional, so the tested set has measure zero

within it. A controller that passed ten thousand trials has ten thousand data points and an uncountable infinity of untested states, and nothing in the test record distinguishes a policy that is stable everywhere in its operating envelope from one that happens to be stable on the trials that were run. Failures in deployed autonomy are frequently of exactly this kind: a state combination outside the tested manifold, encountered once, from which the controller does not recover.

A proof has no such ceiling. If one can exhibit, for a closed-loop system, a scalar function that decreases along every trajectory in a region, then every trajectory that starts in that region converges, including the trajectories that were never tested. This is the content of Lyapunov's stability theorem, published in 1892[1] and standard in nonlinear control[2]. The obstacle has never been the theorem. It has been finding the scalar function for a controller that is itself a neural network, and then checking the decrease condition over a whole region rather than at sampled points — checking it soundly, and checking it fast enough that the result is available where and when the controller runs. This report reviews the machinery that now does both, and argues that the certificate belongs on the device, alongside the policy, as a first-class artifact of trustworthy control.

## 2. Scope and method

This is a review and a research position. It surveys published methods for synthesizing and verifying Lyapunov certificates for learned controllers, and it states the Charlot Lab's position on where such certificates should run. It reports no original experiments and no benchmark numbers of its own; specific quantitative claims are attributed to the cited sources. The mathematics in Sections 3 through 5 is classical and is presented to fix notation and to make the soundness argument explicit, not as new results. The engineering claims in Section 6 — that the combined analytic-and-interval check runs in the browser, on-device, in milliseconds for low-dimensional systems — describe a prototype in the interactive companion and should be read as an early-stage implementation report, not a general performance guarantee. Interval branch-and-bound and satisfiability-modulo-theories verification both scale poorly in the state dimension in the worst case; the method is presently practical for the low-dimensional certified cores of a controller, not for arbitrary high-dimensional policies. The report marks this limit where it is relevant.

## 3. Lyapunov stability and the region of attraction

Consider a closed-loop system in continuous time, the plant under a fixed feedback policy,

$$\dot{x} = f(x), \qquad x \in D \subseteq \mathbb{R}^n, \qquad f(x_*) = 0,$$

with an equilibrium taken without loss of generality at the origin, $x_* = 0$. A *Lyapunov function* is a continuously differentiable scalar field $V : D \to \mathbb{R}$ that behaves like a stored energy for the system. Lyapunov's theorem states that if, on a neighborhood $D$ of the origin,

$$V(0) = 0, \qquad V(x) > 0 \ \ \forall x \in D \setminus \{0\}, \qquad \dot{V}(x) = \nabla V(x) \cdot f(x) < 0 \ \ \forall x \in D \setminus$$

then the origin is asymptotically stable[1,2]. The first two conditions make $V$ a positive bowl with its bottom at the equilibrium. The third, the *decrease condition*, is the load-bearing one: it says the system's motion always points down the bowl, because $\dot{V}$ is the rate of change of $V$ along the flow, obtained by the chain rule as the inner product of the gradient with the vector field. A system cannot spend energy forever, so it must come to rest at the bottom.

Stability at a point is not yet a usable guarantee for a robot, which starts from a set of conditions, not one. The relevant object is the *region of attraction*, the set of initial states from which trajectories converge to the equilibrium. A Lyapunov function delivers a certified inner estimate of it directly. If $c > 0$ is chosen so that the sublevel set

$$\Omega_c = \{\, x \in D : V(x) \le c \,\}$$

is bounded and lies entirely inside the region where $\dot{V} < 0$, then $\Omega_c$ is invariant — a trajectory inside it can never cross out, because to leave it $V$ would have to increase — and every trajectory in $\Omega_c$ converges to the origin. The certificate we want is therefore a pair: a function $V$ and a level $c$ such that the decrease condition holds throughout $\Omega_c$. The engineering objective is to make $\Omega_c$ as large as possible, since it is the provably safe operating envelope of the controller; learned Lyapunov functions can enclose regions substantially larger than those from classical linear-quadratic or sum-of-squares designs[3]. Everything downstream reduces to one question: *is* $\dot{V}(x) < 0$ *true at every* $x$ *in a region?*

## 4. Sound verification: boxes, not samples

The naive way to answer that question is to sample. Draw a dense grid of states in $\Omega_c$, evaluate $\dot{V}$ at each, and declare success if every sample is negative. This is testing again, wearing the clothes of a proof. Between any two grid points lies a continuum of unevaluated states, and the decrease condition is exactly the kind of property a sharp violation can hide in a thin sliver the grid steps over. A learned $V$ paired with a learned controller has no reason to be well-behaved between samples. Sampling cannot certify a region; it can only fail to find a counterexample in a finite set, which is a weaker statement than the one Lyapunov's theorem needs.

The sound alternative reasons over whole regions at once using interval arithmetic[4]. Represent a region of state space as an axis-aligned box $X = [\underline{x}_1, \overline{x}_1] \times \cdots \times [\underline{x}_n, \overline{x}_n]$. Interval arithmetic evaluates the expression for $\dot{V}$ not at a point but over the entire box, propagating intervals through every operation so that the result is a guaranteed enclosure

$$[\dot{V}](X) \ \supseteq \ \{\, \dot{V}(x) : x \in X \,\}.$$

The enclosure is an interval $[\underline{d}, \overline{d}]$ that provably contains the true range of $\dot{V}$ over the box. If its upper end is negative, $\overline{d} < 0$, then $\dot{V}(x) < 0$ for *every* state in the box, with no exceptions

and no sampling — the decrease condition is proven on a piece of the region. Soundness here rests on *outward rounding*: every floating-point operation is rounded in the direction that widens the interval, and elementary functions are bounded by enclosures that contain their true image, so the computed $[\dot{V}](X)$ can only be larger than the truth, never smaller. The proof is sound rather than merely sampled precisely because the arithmetic is deliberately pessimistic.

Interval enclosures are conservative. Because each occurrence of a variable is treated independently, correlated terms are over-widened — the dependency problem — and an enclosure can straddle zero even where $\dot{V}$ is safely negative. This is where *branch-and-bound* enters. When $[\dot{V}](X)$ contains zero, the box is undecided, so it is split along a coordinate into sub-boxes, each of which is enclosed again. Splitting shrinks the widths of the variables, the dependency over-estimation falls with them, and sub-boxes resolve to certified-negative one after another. The recursion has three outcomes per box: the enclosure is negative and the box is certified; the box is split and its children are queued; or a box is driven below a fine resolution and its enclosure still admits a positive value, at which point a point inside it is returned as a concrete *counterexample* state where the decrease condition may fail. A run that certifies every box establishes $\dot{V} < 0$ over the union, a genuine proof over the region; a run that returns a counterexample hands the designer an exact state to inspect and the synthesis loop a point to correct. This counterexample-guided structure is the engine of modern certificate tools such as FOSSIL[5,6], and the branch-and-bound refinement of certified bounds is what recent work scales to neural controllers[7]. Where the verifier is an SMT solver over the reals, the same soundness is obtained by $\delta$-complete decision procedures for nonlinear formulas[8,9], a closely related route to the same guarantee.

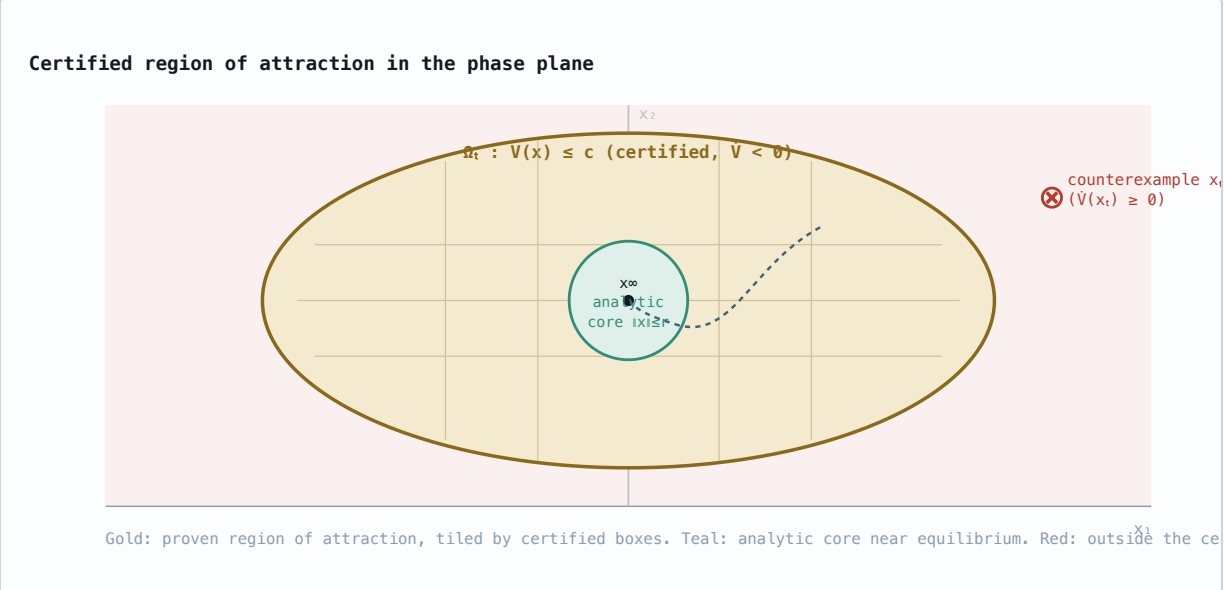


**Figure 1.** A schematic phase plane. The gold sublevel set $\Omega_c$ is the certified region of attraction, established box by box by interval branch-and-bound (the faint tiling). The teal disk is the analytic core $\|x\| \le r$ near the equilibrium, certified in closed form by linearization (Section 5). A state outside the certificate is returned as a concrete counterexample rather than silently missed. An interactive version is at physicalai-bmi.org/research/charlot-lab.

## 5. The analytic core: where the linear term dominates

Interval branch-and-bound has one region it cannot handle well, and it is the most important one: the immediate neighborhood of the equilibrium. There $\dot{V} \to 0$ by construction, since both the gradient and the vector field vanish at the origin, so the true range of $\dot{V}$ over any small box around the origin contains values arbitrarily close to zero. The outward-rounded enclosure, which is strictly wider than the truth, therefore straddles zero no matter how finely the box is split. The verifier can drive boxes to any resolution and still not certify the core, not because the system is unstable there but because the pessimism that makes interval arithmetic sound also makes it blind exactly where the signal is smallest.

The core is instead certified analytically, and this is the classical result that makes the whole scheme close. Linearize the closed-loop dynamics at the origin,

$$A = \left.\frac{\partial f}{\partial x}\right|_{x=0}, \qquad f(x) = Ax + g(x), \qquad \|g(x)\| = O(\|x\|^2).$$

If $A$ is Hurwitz — all eigenvalues with strictly negative real part, which is precisely the condition that the *linear term dominates* nearby — then for any symmetric positive-definite $Q$ the Lyapunov equation

$$A^\top P + P A = -Q$$

has a unique symmetric positive-definite solution $P$, and the quadratic $V(x) = x^\top P x$ is a Lyapunov function for the linearization[2]. Along the full nonlinear dynamics its derivative is

$$\dot{V}(x) = -\,x^\top Q\,x \;+\; 2\,x^\top P\,g(x).$$

The first term is bounded above by $-\lambda_{\min}(Q)\,\|x\|^2$, a negative quantity that grows like the square of the distance from the origin. The remainder term is $O(\|x\|^3)$, because $g$ is quadratic and higher. A cubic is eventually smaller than a quadratic, so there is an explicit radius $r > 0$ below which the negative quadratic provably outweighs the remainder and $\dot{V}(x) < 0$ for all $0 < \|x\| \le r$. The radius is not guessed; it follows from a Taylor remainder bound on $g$ over the ball, and it is exactly the kind of quantity interval arithmetic computes soundly. The analytic core is that ball. Outside it, where $\dot{V}$ is bounded away from zero, interval branch-and-bound is efficient and does the rest. Near the equilibrium, where interval arithmetic fails, the closed-form argument holds. The two verifiers are complementary by construction: each covers precisely the region the other cannot, and their union is a proof over the whole certified set. This analytic-plus-numeric division is standard practice in tools that verify neural Lyapunov certificates[10,7].

## 6. On-device: the certificate that ships with the controller

The two verifiers together are cheap. For the low-dimensional certified cores typical of a joint controller, a balance regulator, or a contact-recovery policy, the analytic step is a single small matrix equation and the branch-and-bound step is a bounded recursion over boxes, both of which execute in milliseconds in ordinary numerical code. Nothing in the method requires a datacenter, a solver license, or a network round trip. It compiles to the same WebAssembly and runs in the same browser sandbox the Charlot Lab already uses for its on-device simulation work. A controller can therefore carry its own proof: the policy weights, the Lyapunov function, the certified level $c$, and a verifier that re-establishes the certificate on the actual device, against the actual compiled arithmetic, at deploy time or on demand.

This changes what a certificate is for. A proof produced once on a workstation certifies the controller as it existed on that workstation. A verifier that ships with the controller certifies the controller as it runs — after quantization, after the compiler's floating-point choices, on the target's own numerics — and can be re-run when the plant model is updated, when a parameter is retuned in the field, or before a safety-critical maneuver is committed. The certificate becomes a live artifact rather than a document in a design review. It is also legible: because the check is sound, a pass is a genuine guarantee over the region, and a fail is a specific state a human can examine. The cost is real and stated plainly — the region certified is only as large as the box search can close, and the method does not extend, today, to certifying a full high-dimensional policy end to end. It certifies the stable core around the operating point, which is the part where a stability guarantee matters most.

## 7. Position: price and proof, together

This report is the second half of a pair. The first, on the energy receipt, described how an embodied decision can carry a sound account of what it cost — an audited, on-device price for a computation, the MathGround receipt[11]. Both halves are instances of one commitment: that the important properties of an embodied controller should be established by construction and carried with the artifact, not asserted after the fact and left in a report. The receipt answers *what did this decision cost*; the certificate answers *will this controller hold*. They are complementary in the same way the underlying computations are. A price without a proof is an efficient controller that may still fail outside its tested set. A proof without a price is a trustworthy controller whose energy budget is unaccounted, which for a battery-bound robot is its own kind of failure. Swap-2C is the position that both should travel with the policy, on the device, checkable in milliseconds, so that an embodied system can state both its cost and its guarantee at the point of action.

The claim is deliberately narrow. None of the components is novel: Lyapunov's theorem is from 1892, interval analysis from the 1960s, the counterexample-guided synthesis of neural certificates and their branch-and-bound verification from the current literature cited here. The position is about placement and pairing — that these established, open methods belong together and belong on-device — and it rests on that prior art rather than on any result original to this report. Where the method is early-stage, as the on-device implementation is, this report has said so.

## 8. Conclusion

Testing tells you a controller worked on the states you tried. A Lyapunov certificate tells you it will hold on every state in a region, because it exhibits an energy the dynamics can only spend. The certificate is checked soundly by two complementary verifiers — an analytic argument near the equilibrium, where the linear term provably dominates, and interval branch-and-bound elsewhere, which proves the decrease condition box by box with outward-rounded arithmetic and returns a concrete counterexample when it cannot. The check is cheap enough to run in the browser, on the device, in milliseconds for the low-dimensional cores where stability guarantees matter most, so the proof can ride with the policy. Paired with the energy receipt, it lets an embodied controller carry both its cost and its guarantee to where it acts. A controller earns trust not by surviving testing but by proving it cannot fail in a region — and by being able to show the proof.

## References


1. A. M. Lyapunov. The general problem of the stability of motion (1892). Reprinted, International Journal of Control 55(3):531–773, 1992. DOI 10.1080/00207179208934253.
2. H. K. Khalil. Nonlinear Systems, 3rd ed. Prentice Hall, 2002.
3. Y.-C. Chang, N. Roohi, S. Gao. Neural Lyapunov Control. Advances in Neural Information Processing Systems 32 (NeurIPS), pp. 3240–3249, 2019.
4. R. E. Moore. Interval Analysis. Prentice-Hall, 1966; R. E. Moore, R. B. Kearfott, M. J. Cloud, Introduction to Interval Analysis, SIAM, 2009.
5. A. Abate, D. Ahmed, A. Edwards, M. Giacobbe, A. Peruffo. FOSSIL: a software tool for the formal synthesis of Lyapunov functions and barrier certificates using neural networks. HSCC '21 (24th ACM Int. Conf. on Hybrid Systems: Computation and Control), 2021. DOI 10.1145/3447928.3456646.
6. A. Edwards, A. Peruffo, A. Abate. Fossil 2.0: Formal Certificate Synthesis for the Verification and Control of Dynamical Models. arXiv:2311.09793, 2023.
7. Z. Shi, H. Li, C.-J. Hsieh, H. Zhang. Certified Training with Branch-and-Bound for Lyapunov-stable Neural Control. arXiv:2411.18235, 2024.
8. S. Gao, S. Kong, E. M. Clarke. dReal: An SMT Solver for Nonlinear Theories over the Reals. CADE-24 (24th Int. Conf. on Automated Deduction), LNCS 7898, pp. 208–214, 2013. DOI 10.1007/978-3-642-38574-2_14.
9. S. Gao, S. Kong, E. M. Clarke. Satisfiability modulo ODEs. FMCAD 2013, pp. 105–112. DOI 10.1109/FMCAD.2013.6679398.
10. S. M. Richards, F. Berkenkamp, A. Krause. The Lyapunov Neural Network: Adaptive Stability Certification for Safe Learning of Dynamical Systems. Conference on Robot Learning (CoRL), 2018. arXiv:1808.00924.
11. D. J. Charlot. Swap-2C and the energy receipt: on-device accounting of the cost of an embodied decision. Institute for Physical AI @ BMI, Technical Report (companion), 2026. Interactive companion: physicalai-bmi.org/research/charlot-lab.


**AI-use disclosure.** Preparation of this report used a large language model (Claude, Anthropic) for drafting and editing text, organizing the reviewed literature, and preparing the figures and the interactive companion. Cited references were checked to resolve to their sources. The author reviewed the content and is solely responsible for it. Consistent with ICMJE, COPE, and IEEE guidance, the model is a tool and is not credited as an author.

**Institute for Physical AI @ BMI** · The Charlot Lab
503 McKeever Rd, Arcola, TX 77583, USA
physicalai-bmi.org · contact@physicalai-bmi.org

Technical Report TR-2026-07 · Preprint v1