Engineered Vitality Systems (EVS) in ΔE-CAS-T and D^3A Architectures

Introduction

Engineered Vitality Systems (EVS) is a concept for designing adaptive systems aimed at "viability" and the long-term stability of their behavior. Unlike classical controllers, which stabilize signals but do not account for the meaning of actions, EVS relies on semantic coherence - the internal alignment of perception, reaction, and context. Traditional approaches (PID controllers, Kalman filters, etc.) can maintain statistical stability while still allowing fragmented or incoherent system behavior. Modern learning algorithms react to errors, but often lose contextual correspondence - achieving accuracy at the cost of losing the meaning of the situation. A new approach is needed, in which the system itself understands what equilibrium is in the context of its goals and environment.

The EVS idea arose from an attempt to create a "synthetic conscience" (Synthetic Conscience) - a special layer in the algorithm responsible for the internal consistency of behavior with its consequences. Research has shown that for intelligent action, a machine needs not just a goal or command, but the ability to maintain semantic coherence between its inputs, reactions, and surroundings. The Petronus project, within which this architecture is being developed, proceeds from a philosophical principle: technology must enhance awareness and mutual understanding between living beings and AI. The EVS architecture embodies this principle in engineering form - it does not fight chaos, but learns to live with it, turning randomness into harmony.

This work examines the architectural foundations of EVS based on ΔE-CAS-T (Coherence Adaptive System with Thermostat) and the D^3A extension (Directional–Deformation–Dissipation Architecture). We focus on the conceptual core of EVS: structural invariants (unchanging characteristics of the system), mechanisms for ensuring coherence, drift regulation (accumulated deviations), the introduction of the concept of internal time, adaptive operating regimes, and the role of the observer as the center of the system's semantic stabilization. It is shown how the principles of coherent control underlying EVS are not reducible to classical controllers or a set of heuristics, but form a new logic of "viability-oriented" control.

Conceptual Foundations of Coherent Control

Coherence in EVS acts as a key invariant - a kind of "internal truth" of the system. For living systems, coherence manifests as the consistency of internal states (a sense of the rightness or wrongness of an action, conscience). The engineering system ΔE-CAS-T translates this concept into technical terms: instead of simple error minimization, it strives for semantic equilibrium - a state in which actions are adequate to the context and goals. This approach gives rise to a new type of regulator - a coherent controller that governs not only the dynamics of the signal, but also the harmony between parts of the system. The criterion of effectiveness here is not the minimum error, but the maximum semantic correspondence of behavior to the current situation. In other words, the system is evaluated by how meaningfully and holistically it responds to the world, and not only by the accuracy of following target values.

It is important to emphasize that coherence is understood as a structural invariant associated with the identity and integrity of the system. In Petronus research, coherence is contrasted with drift: drift is regarded as a loss of internal consistency (structural self-unity) of the system, whereas coherence is the preservation of this integrity. Thus, the task of EVS is to preserve the structural identity of the system in the process of adaptation. Even as learning proceeds or conditions change, the system must remain faithful to some core of meaning and goals - this is served by the mechanisms of structural constraints and observation described below.

A fundamental difference between the EVS architecture and classical control schemes is the presence of an internal meaning loop (synthetic conscience). This additional regulatory loop ensures a link between technical parameters and semantic evaluations. For example, ΔE-CAS-T introduces the concept of coherent entropy - a measure that connects noise disturbances, stability, and semantic coherence of behavior. In this way, the program does not merely keep balance, but truly "understands" equilibrium as applied to its own state. Through the lens of coherence, the system is able to distinguish whether its response is stable precisely in the context of the task, and not abstractly. If classical algorithms optimize purely quantitative metrics, EVS introduces a qualitative aspect - correspondence of action to context and intention. This is the attempt to "embed conscience" into an algorithm: to give the machine a formal, measurable equivalent of a sense of the appropriateness of its actions.

Thus, the EVS concept rests on several interconnected foundations:

Semantic coherence as the control goal: the system strives to minimize not so much physical error as the mismatch between its action and the semantic context. Coherence becomes an invariant that the system maintains in the way a thermostat maintains temperature.

Structural invariants and the system's "$Will$": constraints are introduced on possible changes of the system, ensuring the preservation of its "core" - mission, values, target patterns. These constraints act as guides for the evolution of the state, analogous to a volitional selection of permissible development paths.

Coherent control instead of reactive control: thanks to an internal observer, the system evaluates not only the magnitude of deviation, but the meaning of deviation. This brings the algorithm's operation closer to meaningful behavior - the system, as it were, asks whether the current reaction is consistent with its internal model of the world and its goal.

Long-term adaptation without losing itself: EVS is aimed at long-term viability - the ability to continuously learn and adapt without structurally falling apart. For this, special measures are introduced against the accumulation of drift - the gradual divergence of the internal model from reality, loss of former skills or goal. The system tracks its own state in a long-term perspective and takes action in time if it notices a threat of coherence degradation.

Next, we will consider how these principles are implemented at the architectural level—first in the basic three-loop ΔE-CAS-T scheme, and then in its D^3A extension, which adds a layer of directional constraints and drift regulation.

ΔE-CAS-T Architecture: Three Loops of Coherent Control

The ΔE-CAS-T architecture is a three-level closed loop combining several types of feedback at once to maintain coherence. This abbreviation means E – Coherence Adaptive System with Thermostat, that is, an adaptive system controlled by coherence error and equipped with a thermostat. It includes three main modules (regulation loops):

E-Core (behavioral core): a fast internal loop responsible for the system's immediate reaction. It measures the mismatch E between current perception and response (as well as the deviation of the response from some internal equilibrium center) and corrects the output to reduce this mismatch. Put simply, E-Core smooths and centers behavior: it adaptively adjusts inertia, center of gravity, and damping of the output so that the reaction is smooth and stable, while still following changes in the environment. Formally, parameters are introduced, for example, (adaptive inertia), $_t$ (centering to an internal reference), and D_t (damping of jerks), which are adjusted step by step. As a result, ΔE-Core acts as a nonlinear smoothing controller, minimizing the instantaneous coherence error ΔE_t and providing basic trajectory stability.

CCI-Observer (contextual coherence observer): a medium-speed loop performing the role of the system's "inner eye" or conscience. It does not affect the output directly, but evaluates the quality of behavior -computing the coherence index (CCI) based on deviation ΔE_t and additional factors reflecting context. For example, CCI may increase if the system demonstrates stable correspondence to context (the formula uses parameters trust and stability - trust and stability characterizing the reliability and temporal consistency of the context). The CCI value is normalized: CCI 1 means harmonious alignment, CCI 0 means desynchronization and semantic dissonance. The observer acts as the center of semantic stabilization: it sets a reference relative to which the system "recognizes" deviations not only quantitatively, but also qualitatively. In essence, CCI-Observer provides the system with self-reflection - an internal assessment of how its current state and action fit into the overall picture. This is the "personal center of coherence" - a dynamic axis around which the system's reactions and predictions are built.

Thermostat (entropy homeostat): the slowest loop regulating the system's internal entropy. Its task is to maintain an optimal level of variability (stochasticity) in behavior, linking the degree of randomness to the level of coherence. If the system begins to lose coherence (low CCI), the thermostat can reduce entropy - that is, reduce variability and make behavior more conservative and stable. Conversely, at high CCI the thermostat can allow more experimentation (more entropy) so that the system does not stiffen and can adapt. In the ΔE-CAS-T architecture, the thermostat computes the current entropy H_t of the system and compares it with the target level H^*; the difference $e_t = H^* - \widehat{H}_t$ is used to adaptively correct the entropy coefficient. The important point is that entropy is tied to coherence: noise is not generated arbitrarily, but only to the degree compatible with preserving meaning. This achieves a balance between accuracy and behavioral plasticity: the thermostat does not allow the system to fall either into chaos from excessive randomness or into inertia from excessive rigidity.

These three loops, working together, form a closed coherence–entropy control cycle in which each subsystem supports the others. ΔE-Core directly shapes behavior and minimizes local error; CCI-Observer gives this error a semantic evaluation and sends a signal to the thermostat; the thermostat regulates the "temperature" of the system - the degree of allowable deviations - and thereby influences the parameters of ΔE-Core (${}_t$, ${}_t$, etc.) back. As a result, a dynamic equilibrium is achieved between accuracy and flexibility. The system constantly self-calibrates: unnecessary oscillations and jerks are eliminated, while adaptivity and readiness to respond to new conditions are preserved. Engineering-wise, this is implemented as optimization of a composite criterion - a functional that penalizes both coherence error and excessive sharpness of changes (even introducing the concept of "jerk-penalty" - a penalty for the second derivative of acceleration), ensuring smooth, meaningful trajectories.

It can be said that ΔE-CAS-T acts as a "coherent master controller", coordinating several aspects of behavior at once. For the first time, not only the output signal but also the very criteria of its evaluation are subjected to control—the system has an embedded sense of how well it is coping with the task in a contextual sense. Such a controller does not merely execute commands, but understands the dynamics of the task around which it stabilizes. Thanks to this, ΔE-CAS-T is capable of operating autonomously and softly, anticipating changes in the environment and maintaining the integrity of operation without external prompts. Practical tests and analysis show that such an architecture is universal: it can be applied from robotics (for smooth and meaningful robot motion) to cognitive agents and biotechnical systems. Everywhere it brings a new principle - Coherence-Based Adaptive Intelligence, adaptive intelligence based on coherence, where meaning serves as the main measure of effectiveness.

D^3A: Structural Invariants and Drift Regulation

Although the basic ΔE-CAS-T architecture already provides short-term coherence and stability, this is not sufficient for the long-term viability of the system. In the real world, accumulated small deviations can over time lead to structural drift - a gradual change in behavior or internal parameters that pulls the system away from its originally specified goals and principles. For a system to be truly "viable", it must constrain unacceptable changes in its structure and control the accumulation of hidden deviations. This function is performed by the architectural superstructure denoted as D^3A (Directional–Deformation–Dissipation Architecture).

D^3A extends ΔE-CAS-T by introducing yet another level of regulation related to the direction of changes and their geometric distortions in the system. The three key innovations of D^3A can be summarized as follows:

Directional Constraint Operator (operator of directional constraints): this module introduces into the system the concept of structural invariants and permissible directions of state evolution. Put simply, it determines which changes the system can permit and which it must forbid, based on preserving its integrity and safety. . It embeds constraints of the "*Will*" type - whether safety rules, risk limits, preservation of key variables (energy, resources), or even semantic invariants (inadmissibility of actions contradicting embedded values or the goal). Formally, W_DC (will-derived constraint) can be implemented as a set of permissible trajectories $_t$ or control actions U_t, from which the system chooses the closest to the desired one, filtering out dangerous or "meaningless" development options. Thanks to this component, system behavior gains directionality: even in a disturbed environment, the system moves within a channel that does not violate its internal laws and goals. It can be said that Directional Constraint acts as a formal analogue of a volitional component - it brings "intentionality" into the system without metaphysics, through strict constraints and trajectory priorities.

Operational Spin Estimator (operational spin estimation): introducing constraints on direction does not fully eliminate all consequences of disturbances. When the system deviates in a directed way (for example, follows a new course imposed by a volitional operator), local asymmetric distortions can arise in its dynamics. This phenomenon in D^3A is described by the term operational spin - by analogy with vortices in a continuous medium. Operational Spin is a measure of local "*twistedness*" of the field of changes arising due to a directed impulse. For example, if you suddenly change the course of an autopilot, the system experiences not only linear deviation, but also internal redistribution of stresses, a certain "vortex" of transient processes. Spin can be formally defined as an antisymmetric component of the gradient of the rate of state change. Its appearance means that the system temporarily exited isotropic equilibrium - hidden deformations appeared that are not visible at the level of scalar error, but can accumulate as structural changes. D^3A includes a spin estimator extracting these components from observed gradients of control actions or response. Intuitively, it answers the question: do "*twisted*" stresses appear in the system that can remain after the error has been corrected? Detecting spin makes it possible not to miss subtle inconsistencies that would otherwise accumulate as drift.

Adaptive Dissipative Response (adaptive dissipative response): this block is, in fact, the familiar ΔE loop, but applied specifically to spin regulation. Its task is to "dissipate" emerging deformations by smoothing antisymmetric components, but not destroying dynamics completely. ADR acts similarly to damping of oscillations: it adaptively reduces residual spin (denote it $_t$) at each step, aiming to return the system to smooth, unbiased flow of processes. At the same time, ADR must preserve the system's ability to respond- one cannot simply rigidly suppress all deviations, otherwise the system will lose adaptivity. The ΔE algorithm is well suited for the role of such a regulator - dissipator because it was originally developed as a controller minimizing mismatch with minimal extra energy and jerks. In the D^3A architecture, ΔE is embedded as a layer of adaptive dissipation acting over directional constraints: it absorbs the of a directed impulse, smooths "throws", and thus prevents these deformations from turning into long-term drift. Formally, the spin update through ADR can be expressed, for example, by the equation: $W_{t+1} = (1 - {}_t) \cdot W_t + {}_t \cdot {}_t$, where $_t$ is an adaptive coefficient that increases under large disturbances, and $_t$ is some target "zero" spin level (ideally zero or a small value). This formula means that under strong shocks ADR quickly damps spin ($_t$ close to 1), and under weak ones leaves the system more free (small). As a result, controlled dissipation of the energy of directed disturbances is achieved - the system weakens harmful vortices but does not suppress useful dynamics.

Drift Accumulation Monitor (drift accumulation monitor): the final element of D^3A is an observer that sums residual distortions over time. Even with ADR present, some part of spin may remain ($\tilde{\Omega}_{\text{res}}$) - for example, small structural changes that cannot be eliminated immediately. The drift monitor integrates this residual quantity: $D_t = \Sigma_k \ W_k$, i.e., it estimates accumulated drift D_t at step t. This is, in essence, the memory of the system about how much it has "shifted aside" from its initial state over a long period. If D_t grows and exceeds some threshold D_max, this is a signal: the system has begun to shift noticeably, losing its previous coherence. In that case, drift feedback is activated - the system automatically revises its constraints or increases dissipation. For example, when the drift threshold is exceeded, one can tighten Directional Constraint (narrow the permissible directions of change) and/or strengthen ADR (make it more aggressive in damping deviations). In other words, the system transitions into a self-protection mode, preventing further slipping. This mechanism guarantees that long-term stability will not be lost: as soon as accumulated changes threaten to take the system beyond stability bounds, EVS corrects its course, "pulling the reins".

Taken together, the expanded D^3A + ΔE-CAS-T architecture forms a multi-level control system in which semantic invariants, coherence, and adaptation are intertwined. The upper level (Directional Constraint) sets a volitional frame - invariants and goals that cannot be violated; the middle levels (ΔE-Core + ADR + Observer) ensure continuous behavioral coherence, comparing actions with context and dissipating deviations; the lower level (Thermostat) and the drift monitor are responsible for bringing variability to a safe optimum and controlling long-term changes. Such modularity makes it possible to design and analyze the system at different levels: constraints, dynamics, and environmental effects are separable and observable independently. Moreover, the introduction of explicit metrics - coherence C_t and drift D_t - provides a tool for quantitative evaluation of system viability. For example, coherence can be used as a signal indicator: if C_t falls below a specified level, the system knows that the operating regime must be changed (for example, switched to a safe mode). Similarly, the growth of drift D_t serves as an objective criterion for the onset of a regime transition - a change in adaptation strategy or the need for external intervention.

It should be emphasized that D^3A principles make the EVS architecture is robust not only to one-time disturbances disturbances, but also to repeated loads and long-term operation. In ordinary systems, control is tuned for average conditions; as changes accumulate, they either require manual retuning or the system degrades. EVS, however, effectively includes a mechanism for continuous self-recalibration : it tracks its own state in the internal state space and maintains structural integrity almost the same way living organisms maintain homeostasis. This approach brings a technological system closer to the biological paradigm of viability, where what matters is not optimal instantaneous behavior, but the ability to survive and preserve functions in the long term.

Internal Time and Adaptive Operating Regimes

One characteristic feature of EVS is the presence in the system of something akin to internal time - a set of time scales and memory mechanisms allowing it to distinguish fast changes from slow ones and to account for the history of its own state. In classical controllers, long-term memory is often absent: they respond only to current errors. In EVS, time becomes an explicit adaptation parameter.

First, the three-loop structure of ΔE-CAS-T itself implies different update rates: the behavioral loop ΔE-Core triggers instantly (at every control step), the CCI observer integrates information over a medium interval, and the thermostat regulates entropy over even longer time spans. This separation introduces an effect similar to the presence of "fast" and "slow" variables in the system, analogous to how an organism has fast reflexes and slow hormonal regulation. For example, ΔE-Core instantly damps a fluctuation, but if fluctuations occur frequently , the observer captures the tendency (through a reduction in the trust/stability indicator s) and the thermostat gradually tunes parameters, changing the system's operating regime. Something like an automatic shift of the adaptation time window occurs: with a stable context, the system can slow down changes (low entropy), while with fast changes it can accelerate response (increase entropy and sensitivity).

Second, EVS integrates memory of the past to predict the future. The architecture includes elements that extrapolate trends and prevent lag. Thus, ΔE-CAS-T includes an external observer (*ExternalObserver*) that performs a prediction of the equilibrium center c_t - essentially predicting where the optimal point will shift in the near future so the system can adjust in advance. This mechanism reduces phase lag and prevents overcontrol: the system behaves proactively, "looks forward". From the standpoint of internal time, the system gains a forecasting vector by which it partially goes beyond the current moment. Figuratively speaking, EVS can "feel its future state", turning adaptation into a kind of anticipation. This is extremely important for preventing hidden problems: the system can respond in advance if it sees that under continuation of the current tendency, coherence will worsen or drift will exceed the norm.

Third, the drift loop described above itself gives the system a long memory. Accumulated drift D_t is the aggregate memory of all past disturbances, a "trace of history" influencing present control. Through parameter D_t, EVS "remembers" how strongly and in what direction it has already adapted, and can make decisions based not only on the current state, but on integral experience. For example, if the system sees that it has already moved far from the starting point (large D_t), it tends to act more cautiously so as not to lose stability completely. In biological terms, this is an analogue of accumulated fatigue or wear, to which the organism responds by rest or behavior change. EVS, of course, does not tire, but it takes into account its "age" or "path traveled", tuning algorithms to long-term load.

All these elements allow the system to operate in different adaptive regimes. A regime here can be understood as a set of internal parameters and strategies characteristic of certain conditions. EVS smoothly switches between regimes as the environment changes, and switching occurs not discretely by a rigid command, but self-regulated - by internal criteria of coherence and drift.

For example, in a normal situation with high CCI and low drift, the system operates in an expansive adaptation regime: entropy is sufficient for exploring new variants, constraints are relaxed within reasonable limits (the system can learn and try new things relatively freely). If the trust coefficient CCI falls - a sign that the current model is not coping well - the system can shift to a reconnaissance or recalibration regime: the thermostat temporarily reduces variability (stabilizing behavior), and the observer may initiate, for example, a revision of internal hypotheses (interaction with a learning module, if present).If an exceedance of the drift threshold $D_{\max}$ is detected, EVS enters a protection regime: Directional Constraint is tightened (not allowing risky maneuvers), and ADR max damps any arising oscillations. This regime is analogous to a physiological state of resource mobilization - when an organism or system throws all its forces into maintaining vital functions, sacrificing nonessential activity.

The advantage is that all these transitions occur smoothly and meaningfully, without panic jerks. A coherent architecture can predict phase shifts and restore stability in advance rather than reacting post factum. If environmental changes are gradual, EVS will gradually shift its parameters, avoiding sharp boundaries between regimes. If a sharp jump happens, then thanks to embedded forecasting and protection mechanisms, the system quickly finds a new balance. Stability boundaries are explicitly defined: in terms of coherence and drift. Once abstract notions like "failure" or "overload" receive a quantitative expression - for example, CCI $<$ threshold or $D > D_{\max}$ signal approach to a critical point. This means the system knows when it is close to losing viability and takes measures instead of suddenly breaking.

As a result, EVS functions not as a one-dimensional regulator, but rather as a directed adaptive organism. It has an internal observer equivalent to primitive self-awareness that keeps the system near an "axial" state -the core of meaning. It has a "heart" of adaptation - the ΔE-Core + Thermostat loop, pumping energy in the right rhythm, and it also has a "mind" / "will" - a block of directional constraints specifying where to strive and what to avoid. Finally, it has "memory and time" - drift accumulation, trust, forecasting, which make behavior not merely reactive, but infused with experience.

Conclusion

The Engineered Vitality Systems architecture demonstrates a new approach to building adaptive systems - an approach in which viability, integrity, and semantic stability of the system throughout its entire operation are placed at the center. At the implementation level, EVS integrates several layers of control: from physical reactions to semantic observation and structural supervision. At the conceptual level, it connect the philosophy of semantic alignment with strict technical mechanics of adaptation.

It can be said that the EVS architecture forms the basis for a new generation of "thinking" control programs - coherent master controllers capable of acting autonomously, softly, and meaningfully. For the first time, an engineering system not only reacts to external stimuli but strives for an internal truth of its behavior. Thus, a qualitatively different inventive step is realized: the transfer of the notions of understanding, trust, and semantic stability into the space of technical systems.

The consequence of this step is that system stability ceases to be a purely technical notion - it acquires an ethical dimension. When we speak about maintaining coherence, in essence we are speaking about an engineering equivalent of ethics: the system seeks such regimes where its survival is aligned with the integrity of the goal and, if we broaden the context, with the good of those around it. EVS binds together energy and awareness, survival and meaning, showing that adaptation can be based not only on computation, but also on resonance of meanings. Thus, Engineered Vitality Systems open a path to creating systems that do not merely learn - they understand and preserve the continuity of their "being". In the long-term perspective, this brings us closer to engineering systems with elements of consciousness, where coherence becomes a new fundamental principle alongside energy and information. The presented ΔE-CAS-T architecture with the D^3A superstructure is only the first step in this direction, but already now it forms a practical framework for building empathic, self-organizing, and ethically calibrated technologies of the future.

MxBv, Poznan 2025.

References

Barziankou M. Philosophical Foundations of the ΔE-CAS-T Architecture // Zenodo (2024) – philosophical context and the concept of coherent control in ΔE-CAS-T.

Barziankou M. Synthetic Conscience in Project Petronus: Towards a Technology of Empathy v2 // Zenodo (2025) – description of the "synthetic conscience" concept and the implementation of empathy in Project Petronus algorithms.

Barziankou M. Directional–Deformation–Dissipation Architecture (D^3A) // Zenodo (2025) – technical description of the D^3A architecture: directional constraint operator, operational spin, adaptive dissipation, and drift monitoring.

Barziankou M. Modeling Coupling Frameworks, Architectures, and Emergent Phenomena // Zenodo (2025) – study of links between multi-loop control architectures and the emergence of emergent effects (intuition, synthetic empathy, etc.).

Barziankou M. A Structural Framework for Regime Transitions and Coherence in Adaptive Systems // Zenodo (2025) – a structural approach to describing regime transitions in complex systems and maintaining coherence in them; introduction of semantic stability metrics.