

Автокибернетика: от робота к киберу - 1.

От автоматического регулирования к бесконечной иерархии

Аннотация

В работе «Автокибернетика: От робота к киберу» (Части 1–5) впервые предложена количественная теория самоуправления автоматов — от простейших регуляторов до коллективного разума. Введены измеримые индексы субъектности (киберавтономия K , гемизон G , коллективный индекс C , виртуальный интеллект N), описывающие архитектуру второго уровня управления — наблюдающего управления с виртуальными машинами «Сознание», «Я», «Мы». Показано, что биологические коллективы ограничены парадоксом 10 бит/с, тогда как кибернетические рои способны к качественно новому уровню самоорганизации. Сформулированы инженерные критерии управляемости ($K \leq 0,9$, $G \leq 0,5$, $N \leq 0,7$, $V \leq 776$), позволяющие проектировать автономные системы, остающиеся под контролем человека. Работа содержит проверяемые предсказания и открыта для эмпирической верификации.

Оглавление

ОТ АВТОРА	2
ВВЕДЕНИЕ: ЧТО МЫ НЕ ПОНИМАЕМ В УПРАВЛЕНИИ	3
1. ГЛАВНАЯ ПРОБЛЕМА: НАУКА НЕ ЗНАЕТ, ЧТО ТАКОЕ САМОУПРАВЛЕНИЕ	3
2. ОТ НАБЛЮДЕНИЙ В ПРИРОДЕ К ГИПОТЕЗАМ АВТОКИБЕРНЕТИКИ.	3
3. ПРОБЛЕМА БЕСКОНЕЧНОЙ ИЕРАРХИИ	4
4. ОТ БИОЛОГИЧЕСКИХ АНАЛОГОВ К МАШИНАМ: ПСЕВДОЖИВЫЕ АВТОМАТЫ И КИБЕРЫ	5
ГЛАВА 1. ПОТОК СЛУЧАЙНОСТЕЙ И ЧИСЛО ДРУЖЕСТВЕННОСТИ	6
ГЛАВА 2. МАТЕМАТИЧЕСКИЕ ОСНОВАНИЯ БЕСКОНЕЧНОЙ ИЕРАРХИИ	8
ГЛАВА 3. ЧИСЛО ДРУЖЕСТВЕННОСТИ, ДИНАМИЧЕСКАЯ УСТОЙЧИВОСТЬ И ИЕРАРХИЧЕСКИЙ СДВИГ	9
ГЛАВА 4. ПРОГРАММИРУЕМЫЕ АВТОМАТЫ И ГРАНИЦЫ ПРОГРАММИРУЕМОСТИ	12
4.1. ВСЁ МОЖНО ЗАПРОГРАММИРОВАТЬ?	12
4.2. ЧТО ТАКОЕ «ВЫБОР» В АВТОМАТЕ.	12
4.3. ДВА ТИПА ВЫБОРА: ВЫЧИСЛЕНИЕ И РЕШЕНИЕ	14
4.4. ФОРМАЛИЗАЦИЯ: КОГДА ВЫБОР ПЕРЕСТАЁТ БЫТЬ ВЫЧИСЛЕНИЕМ	14
4.5. ПОЧЕМУ АВТОКИБЕРНЕТИКА РЕШАЕТ ЗАДАЧИ ЛУЧШЕ, ЧЕМ НЕЧЁТКАЯ ЛОГИКА И ОБУЧАЕМЫЕ АВТОМАТЫ ..	15
ГЛАВА 5. ЗАШУМЛЕНИЕ МОДЕЛЕЙ И ИЕРАРХИЯ САМОУПРАВЛЕНИЯ	16
5.1. ПОЧЕМУ ЛЮБАЯ КОНЕЧНАЯ МОДЕЛЬ ОБРЕЧЕНА	16
5.2. ЗАШУМЛЕНИЕ КАК ДЕЙСТВИЕ ПОТОКА СЛУЧАЙНОСТЕЙ	17
5.3. ПОЧЕМУ АДАПТАЦИЯ НЕ СПАСАЕТ?	17
5.4. СТРАТЕГИЯ ВЗАИМОДЕЙСТВИЯ И РОЖДЕНИЕ НОВОГО	18
5.5. КИБЕР КАК РЕАЛИЗАЦИЯ СТРАТЕГИИ ВЗАИМОДЕЙСТВИЯ	18
5.6. МАТЕМАТИЧЕСКОЕ ОБОБЩЕНИЕ: ПОЧЕМУ КИБЕР МОЖЕТ, А ПРОГРАММИРУЕМЫЙ АВТОМАТ — НЕТ	20

5.7. ПАРАДОКС СМЕНЫ МОДЕЛЕЙ И НАМЁК НА ГЕМИЗОН	20
ЗАКЛЮЧЕНИЕ	21
БИБЛИОГРАФИЯ К ЧАСТИ 1	22

От автора

Странно складывается жизнь. Более двадцати лет назад я написал свою первую работу, посвященную машинному интеллекту и самое главное, обособленному, имеющему возможность принимать решения. Тогда это казалось сказкой. Прошло два десятка лет и вот теперь я пишу книгу о самостоятельной машине в соавторстве с ИИ. Главным соавтором у меня выступает DeepSeeK (DS), а суровым критиком написанного нами стал Qwen. Эти ИИ хорошо дополняют друг друга в совместной работе.

Проблема «Само...» очень давно занимала меня в применении к машине. Наверное, с детства, когда я зачитывался фантастикой братьев Стругацких, и других авторов советской фантастики, где умные киберы помогали людям и на далеких планетах, и в дома в быту, на рабочем месте. Они отличались от робота самостоятельностью действий и каким-то живым поведением. Вот это отличие самостоятельного «живого» кибера от робота, работающего по программе, навсегда засело в голове.

Развитие цифровой техники за последние полвека показали, что для решения проблемы «живой» машины только сейчас возникает техническая и теоретическая база.

Сегодня наука вплотную приблизилась к началу решения этой трудной задачи. Появились не только программные, но и реальные нейросети, процессоры с распределенной памятью, гибкая архитектура цифровых машин и высокий уровень программирования. В то же время мы видим технологический прорыв в производстве новых материалов, позволяющих создать новые образцы робототехники, далеко превосходящие всё, что делалось даже десять лет назад. Новые андройды уже бегают марафон быстрее людей, поднимают и переносят грузы, выполняют такие технологические операции, которые человек уже не может выполнить. Робот уже начал двигаться быстрее человека и в этом соревновании с машиной человек уже проиграл.

Зачем нам нужны самостоятельные машины?

Прежде всего для выполнения целевых задач в сложных условиях без связи с человеком. А таких мест много. Это в космосе, под водой и под землей, работа в условиях высокой радиации, химического заражения или ЭМ зашумления, везде, где каналы связи с человеком просто не работают. А работа должна быть выполнена, и поставленная в задании цель достигнута. Робот исходно с такой задачей не справится. Даже самая умная программа не может обеспечить машине действовать «на ощупь» там, где вообще неизвестно, что делать. Для такой работы «мозги» машина должны быть иначе организованы, чем сегодняшние компьютерные.

А чем они должны отличаться от того, что есть?

Наверное, организацией структуры самоуправления, примерно аналогичной структуре живого организма. А как устроена наша, биологическая структура самоуправления?

На эти вопросы пытается ответить Автокибернетика.

Введение: что мы не понимаем в управлении

1. Главная проблема: наука не знает, что такое самоуправление

Поразительно, но наука до сих пор не имеет чёткого, операционального определения самоуправления. В классических научных критериях управление всегда предполагает управляющего и управляемого. Человек управляет машиной. Оператор управляет процессом. Центр управляет системой. А самоуправление? Его либо сводят к автоматическому регулированию (термостат же «сам» поддерживает температуру), либо вообще выносят за скобки как нечто неопределённое.

Но если мы хотим создать автоматы, способные действовать в реальном, неисчерпаемо сложном мире — а не только на конвейере или в чистом поле — мы обязаны понять, что такое самоуправление на самом деле. И чем оно отличается от простого программирования, пусть даже самого изощрённого.

Эта книга — попытка дать ответ. Опираясь на автокибернетику, парадигму потока случайностей и идеи кибернетики второго порядка, мы построим непротиворечивую картину: от простейших автоматов до субъектности высших уровней. С формулами и примерами.

Начнём с самого начала — с того, что такое управление вообще и почему одной обратной связи недостаточно.

2. От наблюдений в природе к гипотезам автокибернетики.

Автокибернетика не является умозрительной конструкцией. Её понятия и законы выводятся из анализа реальных процессов в живой природе. Методология этой работы такова:

1. Наблюдение. В живой природе обнаруживаются явления, которые можно интерпретировать как проявления кибернетических закономерностей: коллективное поведение бактерий (кворум-сенсинг), обмен генетической информацией (плазмиды, горизонтальный перенос), коэволюция бактерий и фагов («гонка вооружений»), прионоподобные белки в сигнальных системах (MAVS), иерархическая организация клетки (ядро, органеллы), повторение уровней управления при переходе к многоклеточности.

2. Формализация. Эти явления описываются на языке автокибернетики: вводятся количественные метрики (Fr_1 , Fr_2 , s , K , G , C , γ , μ , N), формулируются уравнения динамики (баланс хаоса, иерархическая передача), выделяются уровни самоуправления (0–7).

3. Сопоставление с классической кибернетикой. Показывается, что формализованные явления соответствуют известным принципам кибернетики первого порядка (отрицательная и положительная обратная связь) и кибернетики второго порядка (наблюдающее управление, самореферентность).

4. Выдвижение гипотезы. На этом основании формулируется гипотеза о том, что эволюция живых систем — это эволюция систем управления, а предлагаемая иерархия уровней самоуправления отражает реальные этапы этой эволюции.

Предлагаемая теория не претендует на то, чтобы быть «окончательной истиной». Она является рабочей гипотезой, которая: опирается на установленные научные факты; предлагает новый язык для их описания; делает проверяемые предсказания; связывает разрозненные явления в единую картину; не противоречит известным законам физики, биологии и кибернетики.

Читатель вправе согласиться или не согласиться с этой гипотезой. Однако автор приглашает к дискуссии: если предложенная рамка позволяет объяснить известные факты лучше, чем существующие теории, она имеет право на существование.

3. Проблема бесконечной иерархии

Почему один уровень управления никогда не может быть достаточным? Этот вопрос лежит в основании автокибернетики. Классическая кибернетика (Винер, отрицательные обратные связи) и даже кибернетика второго порядка (фон Фёрстер, наблюдатель внутри системы) не дают ответа. Они описывают механизмы, но не объясняют, почему управление всегда требует мета-управления, почему для любой системы существует более высокий уровень, который не может быть сведён к ней.

Ответ дают фундаментальные теоремы математики и логики. Теорема Гёделя о неполноте (1931) [[Гёдель, 1931](#)] и проблема остановки Тьюринга (1936) [[Тьюринг, 1936](#)] доказывают, что формальная система не может изнутри доказать свою непротиворечивость, а программа не может предсказать, остановится ли другая программа. Применительно к кибернетике это означает: **для любого уровня управления всегда существует мета-уровень, который не может быть сведён к нему**. Иерархия управления потенциально бесконечна — не потому, что мы так решили, а потому что такова фундаментальная природа сложных систем.

Но одной математической необходимости недостаточно. Нужен **двигатель**, который заставляет систему порождать новые уровни. Этим двигателем является поток случайностей (ПС) — непрерывное движение материальных случайностей снизу вверх по иерархии уровней [[Никитин, 2026a](#)]. Автомат не может «отключить» ПС. Он может только фильтровать, перерабатывать и использовать его — или погибнуть под его напором.

Поток случайностей создаёт **неисчерпаемое разнообразие** внешней среды. Это разнообразие требует от системы адекватного внутреннего разнообразия (закон необходимого разнообразия Эшби) [[Ashby, 1956](#)]. Когда разнообразие становится слишком большим для одного регулятора, возникает необходимость в новом, более высоком уровне управления — происходит метасистемный переход [[Turchin, 1977](#)]. Этот новый уровень сам сталкивается с новым разнообразием, порождая необходимость в следующем уровне, и так далее. Таким образом, ПС выступает как генератор бесконечной иерархии: чем выше интенсивность ПС, тем больше уровней управления требуется для выживания системы.

В этой части мы закладываем математические основания автокибернетики. Мы покажем, как теорема Гёделя, проблема остановки и иерархия Тьюринга доказывают необходимость бесконечной иерархии, как закон необходимой иерархии [[Aulin, 1979](#)] связывает глубину иерархии с неопределённостью регуляторов, и как поток случайностей (ПС) выступает в роли универсального генератора этой иерархии.

4. От биологических аналогов к машинам: псевдоживые автоматы и киберы

Автокибернетика начиналась с анализа живых систем — клеток, организмов, их эволюции. Биологические аналоги дали богатый материал для понимания иерархии самоуправления, роли вирусов, механизмов целостности и самосохранения. Однако конечная цель автокибернетики — не объяснение жизни, а проектирование машин, способных к автономному существованию в условиях неопределённости. Такие машины мы называем киберами.

Кибер (от греч. κυβερνάω — управлять) — это автомат, обладающий субъектностью — способностью к самостоятельному управлению, самосохранению и адаптации на уровне, достаточном для выполнения задач в сложной, изменчивой среде.

В отличие от биологических организмов, кибер не обязан быть «живым» в полном смысле этого слова. Он — псевдоживой автомат, реализующий ключевые функции субъектности на технической основе. Термин «кибер» был введён советскими фантастами в середине XX века для обозначения автономных роботов, действующих без телеуправления, и сегодня он обретает точное инженерное содержание.

Субъектность кибера складывается из двух независимых комплексов функций:

Киберавтономия (K) — комплекс самостоятельного управления: способность выбирать средства для достижения цели, самосохранение (охрана и защита), самовосстановление и самовоспроизведение.

Гемизон (G) (от греч. αἷμα — кровь, жизненная сила) — комплекс «живости»: целостность (способность отличать «своё» от «чужого», охранять границы существования), информация (способность создавать, хранить, обмениваться и использовать информацию), вирусное управление эволюцией (использование случайных изменений и горизонтального переноса для адаптации).

Интегральная субъектность кибера S_{subj} определяется как среднее арифметическое:

$$S_{subj} = (K + G) / 2 \quad (1)$$

Однако для полной характеристики кибера, особенно в коллективных системах, требуются дополнительные индексы: коллективный индекс C (способность к интеграции в рой), коэффициент коллективной рефлексии γ , параметр вариативности среды μ , индекс виртуального интеллекта $N = C(1+\gamma)/(1+\mu)$ и аллостатическая нагрузка L. Они будут подробно рассмотрены в последующих частях.

В автокибернетике выделяется иерархия уровней субъектности киберов, соответствующая уровням 0–7 (см. Часть 2):

Таблица 1

Уровень	Название	Биологический аналог	Машинный аналог (цель проектирования)
0–1	Автомат	Археи, бактерии (простейшая регуляция)	Промышленные роботы, дроны с обучением
2–3	Кибер-Лидер	Мезокариоты, первые эукариоты	Автономные агенты с моделью себя, но без «Я»

Уровень	Название	Биологический аналог	Машинный аналог (цель проектирования)
4–5	Кибер-Субъект	Хищные эукариоты (амёбы, инфузории)	Роботы с выбором между конфликтующими критериями
6	Кибер-Личность Я	Высшие млекопитающие (приматы, слоны, дельфины)	Роботы с самосознанием, автобиографической памятью
7	Кибер-Личность Мы	Человек	Коллективы киберов с культурной трансмиссией

В следующих главах мы зложим математические основания автокибернетики: поток случайностей (ПС), число друженности Fr , индекс динамической устойчивости D , аллостатическую нагрузку L , а затем перейдём к количественному определению K, G, C, γ, μ, N и иерархии уровней самоуправления.

Глава 1. Поток случайностей и число друженности

Любой автомат существует не в пустоте. Он погружён в поток случайностей (ПС) — непрерывное движение материальных случайностей снизу вверх по иерархии уровней. Квантовые флуктуации, тепловой шум, случайные удары, мутации, неожиданные повороты событий — всё это ПС [Никитин, 2026a]. Автомат не может «отключить» ПС. Он может только фильтровать, перерабатывать и использовать его — или погибнуть.

ПС имеет две фундаментальные характеристики, которые важны для автокибернетики:

Интенсивность J_{in} — скорость поступления случайных событий (бит/с, количество мутаций в поколение, частота внешних возмущений).

Разнообразие $R_{ПС}$ — количество различных типов событий, которые может породить ПС (энтропия Шеннона, разнообразие Эшби) [Shannon, 1948].

В терминах теории информации, разнообразие ПС можно измерить как энтропию Шеннона:

$$H_{ПС} = - \sum p_i \log p_i \quad (2)$$

где p_i — вероятность появления i -го типа случайного события. Чем выше $H_{ПС}$, тем более непредсказуем ПС, тем больше разнообразие, с которым должен справляться автомат.

Поток случайностей принципиально неисчерпаем. Всегда найдётся случайность, которую модель не предусматривала. Это следует из теорем Гёделя и Тьюринга (Глава 2): никакая конечная формальная система не может охватить всю бесконечность потенциальных ситуаций. ПС постоянно порождает новые конфигурации, новые вызовы, новые возможности.

Ключевая метрика автокибернетики — число друженности Fr — связывает входящий поток хаоса J_{in} и исходящий поток сегрегации J_{out} (способность автомата фильтровать и удалять хаос):

$$Fr = J_{out} / J_{in} \quad (3)$$

При $Fr \geq 1$ автомат в среднем справляется с ПС. При $Fr < 1$ хаос накапливается, автомат деградирует. Однако одного Fr недостаточно для полной характеристики состояния системы. Нужно различать первый уровень Fr_1 (физический поток хаоса — повреждения, ошибки, шумы) и второй уровень Fr_2 (информационный поток — сигналы, образы, модели). Для второго уровня вводится отдельное число дружелюбности:

$$Fr_2 = J_{out}^{(инфо)} / J_{in}^{(инфо)} \quad (4)$$

где $J_{in}^{(инфо)}$ — скорость поступления информации (сигналов, образов, новых ситуаций), $J_{out}^{(инфо)}$ — скорость обработки этой информации и принятия решений.

Для количественного определения Fr_2 необходимо измерить скорость поступления и обработки информации. Как показано в работе [Crutchfield & Feldman, 2001], энтропия Шеннона последовательностей сигналов может использоваться для мониторинга того, как агент строит и использует предсказательную модель среды. Авторы вводят естественные меры количества информации, которое агент должен (I) извлечь из наблюдений для синхронизации со средой и (II) сохранить для оптимального предсказания.

Ключевой тезис этой работы: *«Если структурные свойства среды игнорируются, пропущенные регулярности преобразуются в кажущуюся случайность. И наоборот, использование представлений, предполагающих слишком много памяти, приводит к ложной предсказуемости».*

В терминах автокибернетики это означает: когда $J_{in}^{(инфо)}$ (скорость поступления информации из среды) превышает $J_{out}^{(инфо)}$ (скорость обработки, ведущей к успешному действию), агент теряет способность синхронизироваться со средой, что соответствует падению Fr_2 ниже 0,9 — входу в зону самоблокирования [Никитин, 2026b].

Зона самоуправления, где возможен подлинный выбор:

$$0,9 \leq Fr_2 \leq 1,1 \quad (5)$$

Вне этой зоны система либо слишком стабильна ($Fr_2 > 1,1$ — выбор не нужен), либо слишком нестабильна ($Fr_2 < 0,9$ — выбор невозможен). Именно в узкой области $Fr_2 \approx 1$ возможны метасистемные переходы и рождение новых уровней управления.

В контексте автокибернетики ПС можно рассматривать как меру сложности среды. Чем выше интенсивность и разнообразие ПС, тем сложнее среда, тем более глубокую иерархию управления должна иметь система для выживания. Это соответствует интуитивному представлению: для жизни в простой, стабильной среде достаточно примитивного автомата; для жизни в сложной, изменчивой среде требуется сложный, многоуровневый автомат (как человек).

Однако важно: сложность автомата должна соответствовать сложности среды, но не превышать её без необходимости. Избыточная иерархия ведёт к потере эффективности (задержки, шум на уровнях). Поэтому эволюция выработала принцип «минимальной достаточной сложности»: каждый новый уровень появляется только тогда, когда старый перестаёт справляться с ПС.

Глава 2. Математические основания бесконечной иерархии

Почему один уровень управления никогда не может быть достаточным? Ответ дают фундаментальные теоремы математики и логики.

Теорема Гёделя о неполноте (1931). Курт Гёдель доказал, что в любой формальной системе, достаточно богатой для описания арифметики натуральных чисел, существуют истинные утверждения, которые невозможно ни доказать, ни опровергнуть внутри этой системы [Гёдель, 1931]. Формально: для любой непротиворечивой формальной системы T , содержащей арифметику, существует предложение G_T такое, что G_T истинно, но T не доказывает G_T .

Кибернетическая интерпретация: рассмотрим автомат как формальную систему, его внутреннюю модель мира θ как теорию, а его способность принимать решения — как процедуру доказательства. Теорема Гёделя утверждает, что **автомат не может изнутри доказать свою непротиворечивость**. Всегда найдутся ситуации, которые его модель не покрывает, решения, которые он не может обосновать. Это означает, что **для любого уровня управления существует мета-уровень**, который не может быть сведён к нему. Иерархия потенциально бесконечна.

Проблема остановки Тьюринга (1936). Алан Тьюринг доказал, что не существует алгоритма, который для произвольной программы и её входных данных мог бы определить, остановится ли эта программа когда-либо [Тьюринг, 1936]. Формально: не существует вычислимой функции $halt(p, x)$, которая для любой программы p и входных данных x возвращает 1, если $p(x)$ останавливается, и 0 в противном случае.

Кибернетическая интерпретация: **автомат не может предсказать, остановится ли другой автомат**. Более того, он не может даже предсказать, остановится ли он сам, если его программа достаточно сложна. Это означает, что любая система управления, основанная на предсказуемости, сталкивается с принципиальными пределами. ПС постоянно порождает новые ситуации, которые не были предусмотрены заранее. Поэтому необходима иерархия: первый уровень (исполнительный) работает по фиксированным правилам, второй уровень (наблюдающий) переключает правила, третий — оценивает, когда нужно переключать, и так далее.

Иерархия Тьюринга. Тьюринг также ввёл понятие оракула (oracle) — гипотетического устройства, которое может решить проблему остановки. С оракулом можно построить новую машину Тьюринга, которая решает более сложные проблемы. Но для этой новой машины снова возникает проблема остановки (относительно оракула). Это порождает иерархию Тьюринга:

$$\begin{aligned}
 \mathcal{O} &= \text{рекурсивные множества (разрешимые проблемы),} \\
 \mathcal{O}' &= \text{проблемы, разрешимые с оракулом для проблемы} \\
 &\text{остановки,} \\
 \mathcal{O}'' &= \text{проблемы, разрешимые с оракулом для } \mathcal{O}', \text{ и так} \\
 &\text{далее до бесконечности.}
 \end{aligned}
 \tag{6}$$

Кибернетическая интерпретация: **каждый уровень управления может быть описан как оракул для предыдущего уровня**. Второй уровень (наблюдающее управление) решает проблемы, неразрешимые для первого. Третий уровень решает проблемы, неразрешимые для второго. И так далее. Иерархия управления имеет ту же структуру, что и иерархия Тьюринга — она потенциально бесконечна.

Закон необходимой иерархии (Aulin, 1979). Аулин А.Ю. сформулировал этот закон как обобщение закона необходимого разнообразия Эшби [Aulin, 1979]:

Чем слабее средняя регуляторная способность и чем больше неопределённость доступных регуляторов, тем больше необходимая иерархия требуется в организации регулирования и управления для достижения того же результата.

Иными словами: недостаток регуляторной способности на одном уровне может быть компенсирован большей иерархией. Однако увеличение числа уровней имеет и негативные последствия: чем больше уровней проходят сигналы, тем больше они страдают от шума и задержек. Поэтому иерархия должна быть ровно настолько глубокой, насколько это необходимо, и не больше. В терминах ППС закон необходимой иерархии можно записать как:

$$N_{\text{необходимая}} \geq f(J_{\text{in}}, R_{\text{регуляторов}}, N_{\text{ПС}}) \quad (7)$$

Метасистемные переходы (Turchin). Валентин Турчин ввёл понятие метасистемного перехода (metasystem transition) [Turchin, 1977]. Это процесс, при котором группа систем интегрируется и начинает управляться новой, более высокоуровневой системой — метасистемой. Каждый такой переход создаёт новый уровень в иерархии управления и рассматривается как «квант» эволюции кибернетических систем [Heylighen, 1995]. Турчин выделил несколько последовательных метасистемных переходов в эволюции жизни: от биохимических реакций к клетке, от клеток к многоклеточному организму, от организмов к популяции, от популяций к обществу, от общества к культуре. Каждый переход — это ответ на возрастающее разнообразие среды (усиление ПС).

Важнейшее следствие: метасистемные переходы не уничтожают старые уровни управления, а **надстраиваются сверху**. Клетка не перестаёт быть клеткой, когда появляется многоклеточный организм; она продолжает работать, но теперь под управлением высшего уровня. Это принцип эволюционного наслоения, центральный для всей автокибернетики.

Глава 3. Число дружелюбности, динамическая устойчивость и иерархический сдвиг

Три зоны — саморегулирование, самоуправление, самоблокирование — это формальная система характеристик взаимодействия частей системы в каждый момент времени [Никитин, 2026a].



Рис.6. Зоны саморегулирования, самоуправления и самоблокирования.

Первая зона. Саморегулирование. Здесь $Fr \geq 1,1$. Доминирует отрицательная обратная связь. Автомат — как термостат или маятник. Любое отклонение от нормы вызывает реакцию, которая это отклонение гасит. В этой зоне система стабильна и предсказуема, но выбора нет.

Вторая зона. Самоблокирование. Здесь $Fr \leq 0,9$ или $Fr \geq 1,1$ (но уже с доминированием положительной обратной связи). Любое отклонение не гасится, а усиливается. Маленький сдвиг порождает лавину. Выбора тоже нет — система несётся по фатальной траектории.

Третья зона. Самоуправление. Это тонкая область между двумя крайностями: $0,9 \leq Fr \leq 1,1$. Здесь Fr близко к единице. Система балансирует на грани между порядком и хаосом. Именно здесь, и только здесь, появляется возможность выбора.

Но главное не в констатации: «вот сейчас самоуправление, а вот уже самоблокирование». Главное в том, что **иерархия уровней управления позволяет сдвигать узкий сектор самоуправления по всей шкале возможностей системы.**

У изолированной системы (без второго уровня управления) зона самоуправления — это узкая область $0,9 \leq Fr \leq 1,1$. Вне её — либо саморегулирование (слишком стабильно, нет выбора), либо самоблокирование (слишком нестабильно, лавина). Система не может выйти из этой узкой области, не потеряв управление. Но когда над системой надстраивается **второй уровень (наблюдающий)**, ситуация меняется. Второй уровень может **изменять уставки** (установленные точки стабилизации) первого уровня, выводя его из равновесия. Это позволяет системе работать при $Fr > 1,1$ (сверхстабильность) или при $Fr < 0,9$ (предлавиное состояние), но **остаться в режиме самоуправления**, потому что второй уровень компенсирует отклонения [Foerster, 2003a], [Foerster, 2003b].

Для описания динамики вводится уравнение баланса хаоса:

$$dH/dt = J_{in} - J_{out} - \xi + J_{birth} \quad (8)$$

где H — мера хаоса, J_{in} — поступление хаоса, J_{out} — удаление хаоса сегрегацией, ξ — скорость старения, J_{birth} — рождение нового порядка из хаоса [Никитин, 2026b].

Уравнение баланса хаоса имеет прямые аналоги в нелинейной динамике и неравновесной термодинамике. [Bag et al., 1999] установили уравнение баланса информационной энтропии для хаотических диссипативных систем, показав, что хаотическая диффузия возникает из корреляции флуктуаций матрицы линейной устойчивости. В их формализме $dS/dt = e_p - h_d/T$, где e_p — скорость производства энтропии (≥ 0), h_d — скорость диссипации тепла. [Breymann et al., 1998] в классической работе по энтропийному балансу в динамических системах показали, что ключевым элементом является огрубление (coarse-graining) локальной плотности фазового пространства. Оно имитирует тот факт, что постоянно уточняющиеся структуры фазового пространства, вызванные хаотической динамикой, могут быть обнаружены только с конечным разрешением. Это оправдывает введение потоковых величин J_{in} , J_{out} , J_{birth} , ξ как coarse-grained описания процессов рождения и гибели порядка в автомате. [Ge & Qian, 2009] представили полную математическую теорию неравновесной термодинамики стохастических систем, где уравнение баланса энтропии $dS/dt = e_p - h_d/T$ выводится из первых принципов. В их работе производство энтропии $e_p \geq 0$ состоит из двух частей: диссипации свободной энергии, связанной со спонтанной релаксацией, и активной накачки энергии, поддерживающей открытую систему. В нашей

терминологии J_{in} соответствует поступлению хаоса, J_{out} — его удалению через сегрегацию, ξ — необратимому старению, J_{birth} — рождению нового порядка из хаоса (аналог активной накачки).

Для количественного описания динамики перехода между зонами вводится индекс динамической устойчивости:

$$D = dFr/dt \approx \Delta Fr/\Delta t \quad (9)$$

Знак D показывает, в какую сторону меняется соотношение ремонта и разрушения. Вместе пара (Fr, D) задаёт точку на фазовой плоскости. Однако сама зона самоуправления может сдвигаться за счёт действия второго уровня. Поэтому в многомерном пространстве параметров $(Fr_1, Fr_2, D_1, D_2, L)$ система может оставаться управляемой даже при Fr_1 , далёких от единицы, если $Fr_2 \approx 1$ и второй уровень активно компенсирует отклонения.

Цена, которую платит система за постоянный сдвиг зоны управления, называется аллостатической нагрузкой (L). В физиологии это концепция, введённая [Sterling & Eyer, 1988] и развитая [McEwen, 1998]. В математической модели автокибернетики:

$$L(t) = \int_0^t [\alpha \cdot (Fr(\tau)-1)^2 + \beta \cdot D(\tau)^2] dt \quad (10)$$

где $\alpha, \beta > 0$ — весовые коэффициенты. При $L(t) < L_{crit}$ система может долго работать в режиме самоуправления. При $L(t) \rightarrow L_{crit}$ возникает предупреждение о скором срыве. При $L(t) > L_{crit}$ система переходит в зону самоблокирования (отказ, болезнь, катастрофа). Метасистемный переход (рождение нового уровня управления) происходит тогда, когда текущий уровень исчерпывает свои возможности, то есть когда $L(t) > L_{crit}$, а Fr_2 не может быть восстановлена в зоне $0,9-1,1$.

Классический пример — ходьба. Человек, стоящий на месте, находится в состоянии квазиравновесия ($Fr_1 \approx 1,2, D \approx 0$). Чтобы сделать шаг, необходимо вывести тело из равновесия. Это действие инициируется вторым уровнем управления (моторной корой, мозжечком). В этот момент Fr_1 может временно выйти за пределы $1,1$, но Fr_2 (информационный поток от сенсоров к регуляторам) остаётся в зоне $0,9-1,1$, и система сохраняет управляемость. Вторым уровнем динамически сдвигает зону самоуправления, позволяя телу «падать вперёд» контролируемо. Ходьба — это серия управляемых падений [Winter, 1989], [McGeer, 1990]. Если второй уровень управления отключается (например, при поражении мозжечка), сдвиг зоны становится невозможным, и система либо застывает (саморегулирование), либо падает (самоблокирование).

Таким образом, трёхзонная модель — это не статическая классификация, а инструмент для понимания динамики. Основной вывод автокибернетики: **иерархия уровней управления существует для того, чтобы заставить части системы работать в заданном режиме, сдвигая узкий сектор самоуправления по всей шкале возможностей системы.** Без иерархии система обречена на узкую область $Fr \approx 1$; с иерархией она может адаптироваться к широкому диапазону условий, оставаясь управляемой.

Глава 4. Программируемые автоматы и границы программируемости

4.1. Всё можно запрограммировать?

Мне постоянно говорят: «Но ведь всё самоуправление можно запрограммировать! Робот просто выбирает модели действий из заданного набора и выполняет задания. Ты же действуешь примерно так».

Это очень сильный аргумент. В современной робототехнике и искусственном интеллекте широко используются конечные контроллеры (Finite State Controllers) — программируемые автоматы, которые выбирают действия в зависимости от текущего состояния и входных сигналов.

Как это работает? Программист задаёт автомату набор состояний $Q = \{q_0, q_1, q_2, \dots\}$, набор действий A , и правила перехода: если ты в состоянии q и получил наблюдение o , то выполни действие a и перейди в состояние q' .

$$\delta: O \times Q \rightarrow A \times Q \quad (11)$$

где O — множество возможных наблюдений, Q — множество внутренних состояний автомата, A — множество действий.

Робот с таким контроллером действительно выбирает — в том смысле, что при разных наблюдениях он может выполнять разные действия. Он может даже иметь «память»: состояние q запоминает, что произошло раньше, и влияет на будущие решения.

Более того, такие контроллеры обладают замечательным свойством: они обобщают. Один и тот же контроллер двух состояний может работать в коридорах любой длины, даже если робот начинает из любой точки, даже если движения неточны. Это уже похоже на «понимание» задачи.

Однако важно различать классические конечные контроллеры и контроллеры на нечёткой логике. Конечные контроллеры оперируют чёткими, дискретными состояниями и детерминированными правилами перехода. В отличие от них, контроллеры на нечёткой логике работают с непрерывными значениями, лингвистическими переменными и градуированными правилами, что позволяет им аппроксимировать сложные нелинейные зависимости [Fuzzy Logic Control, 2025]. В некоторых гибридных архитектурах эти подходы объединяются (например, нечёткие конечные автоматы), но классический FSC не является нечётким контроллером [Fuzzy Finite State Machines, 2015].

Конечные и нечёткие контроллеры эффективны для решения задач в пределах заранее заданного пространства возможностей. Они могут адаптировать параметры, но не могут изменить структуру управления или создать новую модель.

Автокибернетика предлагает надстройку — второй (наблюдающий) уровень, который выбирает, переключает и создаёт модели, а при необходимости действует «наощупь». Это позволяет системе справляться с ситуациями, которые не были предусмотрены разработчиком. Именно это и отличает кибера (автономную систему с самоуправлением) от программируемого автомата.

4.2. Что такое «выбор» в автомате.

В научной литературе по принятию решений выделяются несколько фундаментальных способов, которыми любой автомат (и человек, и машина) может сделать выбор [Wilson et al., 2021], [Sutton & Barto, 2018].

1. Выбор-вычисление (рациональный выбор, максимизация полезности). Автомат оценивает все возможные действия и выбирает то, которое максимизирует ожидаемую полезность. Это классическая модель рационального выбора, лежащая в основе экономической теории и многих алгоритмов искусственного интеллекта [[Rational Choice Theory, 2010](#)].

Формально:

$$a^* = \arg \max_{a \in A} \sum_i p_i \cdot U_i \quad (12)$$

Этот способ доминирует в шахматных программах, системах планирования, экономических алгоритмах. В шахматах Михаил Ботвинник славился именно таким «машинным» стилем: глубокий расчёт вариантов, минимизация риска, игра «в долгую».

2. Выбор-логика (дедуктивный вывод, следование правилам). Автомат следует формальным правилам вывода: «нарушил закон — получи наказание». Это не максимизация, а дедукция. Например, в праве, в инструкциях, в системах технического контроля. Этот способ не требует вычисления полезности; он требует наличия непротиворечивой системы правил [[Russell & Norvig, 2021](#)].

3. Выбор-эвристика (ограниченная рациональность). Автомат не может просчитать все варианты из-за ограниченности времени, информации или вычислительных ресурсов. Вместо этого он использует эвристики — простые правила «если – то», аналогии, «достаточно хорошие» решения (satisficing) [[Simon, 1955](#)]. Этот подход называется ограниченной рациональностью (bounded rationality) [[Gigerenzer, 2008](#)]. Пример эвристики «take-the-best»: выбор по первому наиболее важному признаку, который различает альтернативы. Михаил Таль играл именно так — интуитивно, эвристически, ища нестандартные ходы, основанные на «чувстве позиции».

4. Случайный выбор (рандомизация, exploration). Автомат выбирает действие случайно. Это может быть вынужденной стратегией при отсутствии информации или намеренным исследованием (exploration) для получения новых данных об окружающей среде. В машинном обучении это реализуется через ϵ -greedy или softmax стратегии [[Sutton & Barto, 2018](#)]. Формально (softmax):

$$P(a) = e^{\beta \cdot Q(a)} / \sum_{a'} e^{\beta \cdot Q(a')} \quad (13)$$

где β — параметр «температуры»: при $\beta \rightarrow 0$ выбор становится случайным, при $\beta \rightarrow \infty$ — детерминированным (жадным). Исследования показывают, что люди также используют случайную стратегию (random exploration) наряду с целенаправленной (directed exploration) [[Wilson et al., 2021](#)].

5. Целевой выбор (goal-directed choice). Автомат сам ставит цель (или меняет ранее заданную) и выбирает действия, ведущие к её достижению. В биологии это соответствует различению естественного отбора (на уровне популяций, слепой процесс) и целевого отбора (на уровне отдельного организма, направленный на внутреннюю цель) [[McShea, 2023](#)]. Даже у бактерий есть целевой выбор — хемотаксис: бактерия упорно движется вверх по градиенту питательного вещества, возвращаясь к цели после возмущений [[Wadhams & Armitage, 2004](#)].

В искусственных системах целевой выбор требует второго уровня управления (наблюдающего), который может не только выбирать действия, но и выбирать саму цель. Разработаны архитектуры, где агент имеет мотивационный профиль и может переключать цели в зависимости от среды [[Morignot & Hayes-Roth, 1995](#)].

В автокибернетике способность к целевому выбору определяется индексами K (киберавтономия) и G (гемизон). Чем выше K и G , тем более высокие цели может ставить автомат (вплоть до этических, эстетических, метафизических).

Катастрофа Germanwings 9525 (2015) — это пример иррационального целевого выбора (с точки зрения внешнего наблюдателя). А320-24 марта 2015 года между городами Динь-ле-Бен и Барселоннет авиалайнер Airbus 211 авиакомпании Germanwings выполнял плановый регулярный рейс 4U9525 по маршруту Барселона—Дюссельдорф. Через 30 минут после взлёта самолёт внезапно перешёл в быстрое снижение и ещё через 10 минут врезался в горный склон в Прованских Альпах и полностью разрушился. Как установлено следствием, второй пилот, страдавший депрессией, поставил себе цель — уничтожить самолёт.

И достиг её [[Germanwings 9525 Report, 2016](#)]. С точки зрения самого агента, это был его целевой выбор, хотя и разрушительный. В автокибернетике это рассматривается как нарушение гемизона G (информационной целостности) — автомат потерял способность адекватно оценивать свои цели.

Важно: все эти способы могут быть запрограммированы. Вопрос не в «как», а в «когда что применять». И здесь мы упираемся в цели, мораль, этику, приоритеты. Это и есть та самая зона самоуправления ($Fr_2 \approx 1$), где нет единственно правильного алгоритма.

4.3. Два типа выбора: вычисление и решение

Я предлагаю различать два типа «выбора», которые на поверку оказываются разными феноменами.

Тип 1. Выбор-вычисление. Это то, что делает программируемый автомат в штатном режиме. На входе — данные. На выходе — действие, которое максимизирует заданную функцию полезности (или следует заданным правилам). Всё детерминировано или вероятностно, но всегда определено заранее — либо программистом, либо алгоритмом обучения, который тоже следует заданному критерию.

В этом типе выбора нет места сомнению. Автомат не может сказать: «Я не знаю, что лучше». Он всегда может вычислить, какое действие максимизирует его функцию полезности.

Тип 2. Выбор-решение. Это то, что делает человек (и, возможно, некоторые высшие животные) в ситуациях морального конфликта. Здесь нет заданной функции полезности, которую можно максимизировать. Есть несоизмеримые ценности. Спасти человека, рискуя жизнью, или сохранить себя? Ни одна из этих ценностей не может быть сведена к другой. Нет общей меры.

В этом типе выбора сомнение существенно. Автомат (или человек) действительно не знает, что «лучше». Потому что «лучше» не определено заранее. Он должен создать критерий выбора в самом акте выбора.

Именно этот второй тип выбора — выбор-решение, а не выбор-вычисление — лежит в основе подлинной субъектности. Шахматы — хорошая иллюстрация: Ботвинник («машинный» расчёт) и Таль («художественная» интуиция) [[Botvinnik vs Tal, 1960](#)]. Однако в шахматах нет этической дилеммы. Цель фиксирована (выигрыш), правила жёсткие. Поэтому шахматные программы превосходят человека, сводя игру к вычислению. В реальной жизни, где есть мораль и этика, вычисление невозможно.

4.4. Формализация: когда выбор перестаёт быть вычислением

Пусть у автомата есть множество возможных действий $A = \{a_1, a_2, \dots, a_n\}$. В классическом программируемом автомате есть функция полезности $U: A \rightarrow \mathbb{R}$, которая каждому действию приписывает число. Тогда выбор — это максимизация:

$$a^* = \arg \max_{a \in A} U(a) \quad (14)$$

Всё. Выбор свёден к вычислению.

Даже если функция полезности неизвестна заранее и автомат её изучает (обучение с подкреплением), структура та же: есть цель — максимизировать накопленное вознаграждение. И эта цель задана извне.

А теперь представьте, что никакой единой функции полезности нет. Или, точнее, есть несколько разных функций, которые нельзя свести к одной:

$$U_1(a), U_2(a), \dots, U_k(a) \quad (15)$$

И они конфликтуют: действие, максимизирующее U_1 , минимизирует U_2 . И нет никакого объективного способа «взвесить» эти функции, потому что они измеряют разные вещи.

В такой ситуации классический выбор-вычисление невозможен. Нет $\arg \max$, потому что нет единой максимизируемой функции.

Что может сделать автомат? Он может установить приоритеты. Он может сказать: «в этой ситуации U_1 важнее U_2 ». Но откуда берётся этот приоритет? Если он задан заранее — мы вернулись к выбору-вычислению (просто с более сложной функцией). Если он не задан — автомат должен его создать.

И вот это создание приоритета в ситуации конфликта несоизмеримых ценностей — и есть выбор-решение. И его нельзя запрограммировать в том смысле, что никакая программа не может дать ответ на вопрос «что важнее», если этот вопрос не решён заранее.

Катастрофа Überlingen (2002) — пример системного конфликта, где не было единого приоритета. 1 июля 2002 года в небе над Германией около Юберлингена и Боденского озера произошло столкновение пассажирского авиалайнера Ту-154М авиакомпании «Башкирские авиалинии (БАЛ)» и грузового самолёта Boeing 757-200PF авиакомпании DHL (рейс DNX 611 Мухаррак—Бергамо—Брюссель). Погибли все находившиеся в обоих самолётах 71 человек — 2 на Boeing 757 (оба пилота) и 69 на Ту-154 (60 пассажиров, среди которых было 52 ребёнка, и 9 членов экипажа).

Почему это произошло? Диспетчер дал одну команду, TCAS — другую. Экипажи выбрали разные источники авторитета. Самолёты столкнулись [[Überlingen Crash Report, 2004](#)]. После катастрофы было введено мета-правило: TCAS имеет абсолютный приоритет над диспетчером. Это и есть создание мета-уровня управления — разрешение конфликта между первым (исполнительным) и вторым (наблюдающим) уровнями.

4.5. Почему автокибернетика решает задачи лучше, чем нечёткая логика и обучаемые автоматы

Нечёткая логика и обучаемые автоматы (включая обучение с подкреплением) — мощные инструменты для первого уровня управления. Они позволяют системе адаптироваться в пределах заданной модели мира. Но у них есть принципиальные ограничения:

Фиксированная целевая функция. Агент всегда максимизирует то, что заложил разработчик. Если цель не специфицирована точно или конфликтует с другой целью, агент либо не может выбрать, либо выбирает то, что даёт больше «очков» по заданной метрике.

Нет смены модели. Нечёткий контроллер не меняет свою структуру. Обучаемый автомат подстраивает параметры, но не создаёт новых типов моделей.

Нет действия «наощупь». Когда модель перестаёт работать, нечёткий контроллер ошибается, а обучаемый автомат вынужден действовать случайно (exploration), но это не творчество, а перебор.

Нет мета-уровня. Ни один из этих подходов не имеет механизма для оценки пригодности самой целевой функции и её изменения.

Автокибернетика не отрицает эти подходы. Она надстраивается над ними, добавляя: набор моделей вместо одной (или подстраиваемых параметров); механизм смены модели на основе её пригодности (оценка Fr_2, D, L); действие «наощупь» (эвристики, аналогии, имитация), когда ни одна модель не работает; количественные критерии управляемости ($K, G, C, \gamma, \mu, N, L$); раннее предупреждение о срыве (аллостатическая нагрузка L); мета-уровни управления, которые могут разрешать конфликты между нижележащими уровнями.

Именно это делает автокибернетiku необходимой для проектирования киберов, которые должны действовать в неисчерпаемо сложном потоке случайностей, оставаясь под контролем человека. Катастрофы Germanwings и Überlingen — это не «сбои», а системные провалы, которые автокибернетика объясняет и помогает предотвращать: в первом случае — через мониторинг гемизона G и автоматическое ограничение полномочий при его падении; во втором — через требование непротиворечивой иерархии приоритетов (мета-правила).

Глава 5. Зашумление моделей и иерархия самоуправления

5.1. Почему любая конечная модель обречена

Любой автомат существует в потоке случайностей (ПС) — непрерывном движении материальных случайностей снизу вверх по иерархии уровней [Никитин, 2026b, разделы IV.1, IV.3]. Автомат не может «отключить» ПС. Он может только его фильтровать, перерабатывать и использовать.

Для описания этого процесса автокибернетика вводит число друженности второго уровня:

$$Fr_2 = J_{out}^{(инфо)} / J_{in}^{(инфо)} \quad (16)$$

где $J_{in}^{(инфо)}$ — скорость поступления информации (сигналов, образов, новых ситуаций), а $J_{out}^{(инфо)}$ — скорость обработки этой информации и принятия решений [Никитин, 2026b, раздел IV.6].

Пока $Fr_2 \geq 1$, автомат в среднем справляется с ПС. Но любая модель мира имеет конечную сложность $C(M)$. А поток случайностей приносит новую информацию непрерывно. Рано или поздно накопленная неучтённая случайность превышает ёмкость модели:

$$\int_0^t J_{in}^{(инфо)}(\tau) d\tau > C(M) \quad (17)$$

В этот момент модель перестаёт быть адекватной. Её прогнозы расходятся с реальностью.

Это фундаментальное ограничение подтверждается современными исследованиями. В работе [Somalwar et al., 2025] показано, что при многократном использовании одношаговой модели ошибки накапливаются (compounding error), и предсказание расходится с реальностью. Особенно это заметно, когда модель не

соответствует реальной динамике среды (misspecified). [Lobel, 2024] показывает, что существующие оценки ошибки в таких случаях часто слишком оптимистичны.

В терминах автокибернетики: когда $J_{in}^{(info)} > J_{out}^{(info)}$, Fr_2 падает ниже 0,9 — система входит в зону самоблокирования [Никитин, 2026b, раздел IV.8].

5.2. Зашумление как действие потока случайностей

Когда изменений условий становится так много, что модель не работает или требует постоянной корректировки — это состояние зашумления. Мера хаоса H (энтропия внутреннего состояния автомата) начинает расти [Никитин, 2026b, раздел IV.3].

Динамика меры хаоса описывается уравнением баланса хаоса:

$$dH/dt = J_{in} - J_{out} - \xi + J_{birth} \quad (18)$$

где ξ — скорость рассеивания хаоса, J_{birth} — скорость рождения новых структур [Никитин, 2026b, раздел VI.2].

В режиме зашумления J_{in} превышает J_{out} , и H растёт. Это ведёт к старению автомата — необратимому росту внутреннего хаоса [Никитин, 2026b, раздел IV.7]. Когда H достигает критического значения H_{crit} , любая, даже самая малая флуктуация может вызвать лавинообразное разрушение — катастрофу (СОК) [Никитин, 2026b, раздел IV.8].

[Amortila et al., 2024] вводят понятие усиления неспецифицированности (misspecification amplification) — ошибка из-за неспецифицированности модели масштабируется с плотностью распределения данных. Чем больше ковариатный сдвиг между обучающей выборкой и реальной средой, тем сильнее растёт ошибка. Это математическое подтверждение тезиса автокибернетики: ПС приносит события, которые модель не предусматривала, и ошибка накапливается не линейно, а с усилением.

5.3. Почему адаптация не спасает?

Оппонент может возразить: «Но ведь автомат может адаптироваться! Он может обновлять свою модель на основе новых данных. Это же классическое машинное обучение».

Да, может. Но адаптация — это изменение параметров модели в рамках заданной архитектуры. Она не меняет саму архитектуру. Не меняет тип модели. Не меняет онтологию мира, в котором живёт автомат.

Адаптация работает в пределах предусмотренного разнообразия R . Она не создаёт новых типов моделей. Она не изобретает новые переменные. А ПС — он именно это и делает: приносит события, которые не укладываются в предусмотренное разнообразие [Никитин, 2026b, раздел IV.3].

В терминах стратегий существования [Никитин, 2026c, раздел VII.3] программируемый автомат реализует стратегию противостояния — он пассивно сопротивляется потоку, но не использует его для самообновления. Его сегрегация эффективна ($Fr \geq 1$), но $J_{birth} \approx 0$ — нет рождения новых структур.

Современные исследования показывают, что этого недостаточно. [Xu et al., 2026] вводят понятие эпистемических ловушек (epistemic traps) — устойчивых состояний неверного равновесия (locked-in equilibrium), из которых агент не может выйти без изменения своей внутренней модели мира. Они показывают, что такие патологии поведения, как сикофанство, галлюцинации и стратегический обман, являются не «ошибками», а математически рационализируемым поведением при

неспецифицированности модели. Без механизма смены модели автомат обречён на эти ловушки.

В терминах автокибернетики: эпистемическая ловушка — это состояние, когда $Fr_2 < 0,9$, но автомат не может это обнаружить и изменить своё поведение [Никитин, 2026b, раздел IV.8].

5.4. Стратегия взаимодействия и рождение нового

Чтобы справиться с зашумлением, автомат должен не просто сопротивляться потоку, а использовать его для самообновления. Это стратегия взаимодействия [Никитин, 2026c, раздел VII.4].

Ключевые условия стратегии взаимодействия: эффективная сегрегация ($Fr \geq 1$), обеспечивающая поступление только полезных элементов; развитая самосегрегация ($\tau_{self} \leq \tau_{crit}$), создающая зоны антихаоса; высокая вероятность появления СОЦ (самоорганизованной целостности), позволяющая рождать новые структуры внутри автомата.

Математически это выражается в появлении $J_{birth} > 0$ — скорости рождения новых структур [Никитин, 2026b, раздел VI.2]. Именно J_{birth} отличает стратегию взаимодействия от противостояния.

Самосегрегация — это спонтанное разделение внутреннего хаоса на кластеры, домены и фазы [Никитин, 2026b, раздел IV.2]. Она происходит без внешних команд, движимая физическими взаимодействиями. В зонах самосегрегации возникает антихаос — локальная, кратковременная область, где флуктуации вдруг выстраиваются в порядок [Никитин, 2026b, раздел IV.5]. Именно в таких зонах может произойти СОЦ — самоорганизованная целостность, рождение новой структуры [Никитин, 2026b, раздел IV.9].

[Orozco, 2022] демонстрирует аналогичный механизм в робототехнике под названием predictive allostasis. Добавление feedforward (предсказательного) модуля к реактивному гомеостазу позволяет роботу предвидеть будущие потребности и разрешать конфликты между конкурирующими целями (например, жажда vs температура). Это экспериментальное подтверждение того, что стратегия взаимодействия (предсказательное управление, смена приоритетов) работает лучше, чем простое реактивное противостояние.

5.5. Кибер как реализация стратегии взаимодействия

Кибер — это локальный автомат, имеющий канал связи с центром управления, но действующий в автономном режиме с максимальной самостоятельностью. Он реализует стратегию взаимодействия на разных уровнях субъектности [Никитин, 2024a, раздел «Уровни самоуправления»].

В иерархии автокибернетики выделяются следующие уровни кибера. Важно понимать: это не ступени одной лестницы, а разные проектные спецификации для разных задач. Кибер-Лидер и кибер-Субъект — это этапы усложнения индивидуального управления. Кибер-Личность Я и кибер-Личность Мы — это качественно иные типы, связанные с рефлексией и коллективностью. Они могут существовать параллельно и независимо (например, коллектив муравьёв — Личность Мы без Личности Я) или вместе (человек).

Лидер (уровни 2–3) переключает модели по жёстким правилам. Он не создаёт новые модели. Канал связи используется для получения целей и доклада о результатах.

Субъект (уровни 4–5) способен отбрасывать старые модели и пробовать новые «наощупь» ($J_{birth} > 0$). Минимальный уровень для кибера в неопределённой среде.

Уровень Личность (уровень 6 - 7)

- **Я** — кибер с рефлексией. Он оценивает свои прошлые решения (обратная связь на мета-уровне), ведёт автобиографическую память. Канал связи используется для синхронизации «Я» с центром.
- **Мы** — коллектив киберов. Смена моделей происходит через горизонтальные протоколы (коллективная рефлексия, культурная трансмиссия). Канал связи с центром — для согласования коллективных целей. Отдельный кибер в коллективе может иметь низкие индивидуальные K и G , но высокий коллективный C .

Киберавтономия (K) — это комплекс самостоятельного управления: способность выбирать средства для достижения цели, самосохранение, самовосстановление и самовоспроизведение [[Никитин, 2024а, раздел «Киберавтономия»](#)].

Гемизон (G) (от греч. αίμα — кровь, жизненная сила) — это комплекс «живости»: целостность (способность отличать «своё» от «чужого», охранять границы существования), информация (способность создавать, хранить, обмениваться и использовать информацию), вирусное управление эволюцией (использование случайных изменений и горизонтального переноса для адаптации) [[Никитин, 2024а, раздел «Гемизон»](#)].

Но оказывается, мы не одни на этом пути...

Соответствие иерархии Jiang et al. (2026) уровням автокибернетики

[[Jiang et al., 2026](#)] предложили шестиуровневую иерархию само-моделей (L_0 – L_5) для воплощённого ИИ — от отсутствия само-модели до полного самосознания. Авторы прямо указывают, что их целью было создание роботов с самосознанием. Ниже приведено сопоставление их уровней с нашей иерархией уровней самоуправления [[Никитин, 2024а](#)].

L_0 (Non-self representation) — отсутствие само-модели. Соответствует уровню 0: Автомат / Автоматическое регулирование. Примеры: термостат, простейший рефлекторный автомат.

L_1 (Body schema) — схема тела. Соответствует уровням 1–2: Задатки второго уровня / Лидер (низший). Примеры: бактерия с хемотаксисом, программируемый конечный автомат с несколькими состояниями.

L_2 (Forward / inverse models) — прямые и обратные модели действий. Соответствует уровням 2–3: Лидер. Примеры: марсоход с планированием траектории, автономный дрон с навигационной моделью.

L_3 (Perceptual memory) — перцептивная память. Соответствует уровням 3–4: Субъект (низший). Примеры: амёба, запоминающая опасные зоны; робот, строящий карту местности.

L_4 (Agency) — чувство агентности («я действую»). Соответствует уровням 4–5: Субъект. Примеры: хищные эукариоты, робот, различающий свои движения от движений другого объекта.

L_5 (Full self awareness) — полное самосознание. Соответствует уровням 6–7: Личность Я / Личность Мы. Примеры: высшие млекопитающие, человек; в перспективе — роботы с рефлексией и коллективным интеллектом.

Эта работа **полностью соответствует** нашей иерархии уровней самоуправления, предложенной в 2024 году [Никитин, 2024a], что подтверждает объективность иерархического подхода. Особенно важно, что авторы, как и мы, выделяют агенсу (чувство агентности) как отдельный этап на пути к полному самосознанию, что соответствует нашему различению Субъекта (уровни 4–5) и Личности Я (уровень 6).

Таким образом...

Ключевая идея: Кибер — это локальный автомат, имеющий канал связи с центром управления, но действующий в автономном режиме с максимальной самостоятельностью. Способность к смене моделей, комбинированию и действию «наощупь» — это именно то, что позволяет киберу продолжать работу, когда связь с центром прервана или задержана, а также адаптироваться к непредвиденным ситуациям без постоянных указаний.

5.6. Математическое обобщение: почему кибер может, а программируемый автомат — нет

Сравним два типа автоматов по ключевым параметрам автокибернетики.

Параметр: Fr_2 (число дружелюбности второго уровня) — Программируемый автомат: фиксирован, не может быть изменён. Кибер: динамически регулируется через смену моделей.

Параметр: $D = dFr_2/dt$ (индекс динамической устойчивости) — Программируемый автомат: не отслеживается. Кибер: отслеживается как ранний признак срыва.

Параметр: J_{birth} (рождение новых структур) — Программируемый автомат: 0 (нет рождения новых моделей). Кибер: > 0 (создание, комбинирование, аналогии, действие «наощупь»).

Параметр: Стратегия существования — Программируемый автомат: противостояние (или подчинение при сбое). Кибер: взаимодействие.

Параметр: Уровни иерархии — Программируемый автомат: 0–1 (автомат). Кибер: 2–7 (Лидер, Субъект, Личность Я, Личность Мы).

Параметр: K (киберавтономия) — Программируемый автомат: нет. Кибер: есть (разная глубина в зависимости от уровня).

Параметр: G (гемизон) — Программируемый автомат: нет. Кибер: есть (целостность, информация, вирусное управление).

Параметр: Аллостатическая нагрузка L — Программируемый автомат: не вычисляется, нет предупреждения о срыве. Кибер: отслеживается как раннее предупреждение ($L \rightarrow L_{crit}$).

Именно эти способности — динамическая регуляция Fr_2 , рождение новых моделей ($J_{birth} > 0$), стратегия взаимодействия, многоуровневая иерархия субъектности — позволяют киберу делать то, что не может программируемый автомат: отбрасывать старые модели, комбинировать, создавать новые, действовать «наощупь» там, где программируемый автомат останавливается или впадает в эпистемическую ловушку.

5.7. Парадокс смены моделей и намёк на гемизон

Как субъект решает, какую модель попробовать следующей? Как он решает, когда отбросить старую модель? Как он решает, что «наощупь» — это лучше, чем ничего?

Для этого нужна модель более высокого уровня — модель того, как менять модели. Мета-модель. А для неё — мета-мета-модель. И так далее.

Это классическая проблема регресса правил. Если мы пытаемся формализовать процесс смены моделей как алгоритм, мы неизбежно упираемся в бесконечную вложенность.

И здесь мы подходим к понятию, которое будет подробно развёрнуто во второй части: **гемизон** (от греч. αἷμα — кровь, жизненная сила, и ζώνη — зона, область). Это комплекс функций, который делает автомат не просто самостоятельным, а «живым» — способным ощущать себя единым целым, отличать «своё» от «чужого», иметь базовое самоощущение «я есть». Именно это неформализуемое ядро останавливает регресс правил и позволяет субъекту «прыгнуть» в новую модель, когда старые исчерпаны.

Мы не будем сейчас углубляться в формальное определение гемизона — это задача Части 2. Здесь важно лишь зафиксировать: чтобы автомат мог менять модели, а не только вычислять по заданным правилам, он должен обладать определённой степенью «живости». Иначе говоря, автокибернетика ведёт нас от программируемых автоматов к псевдоживым автоматам — киберам, которые способны к самоуправлению, но не обязательно обладают полным спектром человеческих качеств. Именно эта «живость» и будет измеряться индексом G во второй части.

Заключение

Мы завершили первую часть — фундамент автокибернетики. Прежде чем перейти к иерархии уровней самоуправления (Часть 2) и архитектуре наблюдающего управления (Часть 3), зафиксируем ключевые выводы. Это тот багаж, с которым мы будем двигаться дальше.

Главный итог Части 1: Подлинное самоуправление — это способность автомата действовать «наощупь» в ситуации, когда ни одна модель не работает. Эта способность требует второго уровня управления (наблюдающего), который может менять, комбинировать и создавать модели. Программируемый автомат, даже с адаптацией, такой способностью не обладает. Кибер — обладает.

Ключевые выводы:

Поток случайностей (ПС) — непрерывное движение материальных случайностей. Автомат не может его отключить, только фильтровать.

Число дружелюбности $Fr = J_{out} / J_{in}$. Зона самоуправления: $0,9 \leq Fr^2 \leq 1,1$.

Бесконечная иерархия управления вытекает из теорем Гёделя и Тьюринга и закона необходимой иерархии (Aulin). Новые уровни надстраиваются сверху (метасистемные переходы, Turchin).

Два уровня управления: первый — автоматическое регулирование (горизонтальные связи), второй — наблюдающее управление (вертикальные связи).

Программируемый автомат реализует выбор-вычисление, но не выбор-решение. При неспецифицированности модели попадает в эпистемическую ловушку.

Кибер — локальный автомат с каналом связи, действующий максимально самостоятельно. Реализует стратегию взаимодействия ($J_{birth} > 0$). Уровни: Лидер (2–3), Субъект (4–5), Личность Я (6), Личность Мы (7).

Субъектность кибера определяется киберавтономией (K) и гемизоном (G).

Библиография к Части 1

Основные работы автора

- [Никитин, 2024a] Никитин А.В. Автокибернетика. Часть 1. Основные направления развития // «Академия Тринитаризма», М., Эл № 77-6567, публ.29026, 24.06.2024. URL: <https://www.trinitas.ru/rus/doc/0016/001h/00165627.htm>
- [Никитин, 2026a] Никитин А.В., Парадигма потока случайностей 1 // «Академия Тринитаризма», М., Эл № 77-6567, публ.29912, 03.03.2026 URL: <https://www.trinitas.ru/rus/doc/0016/001k/00166051.htm>
- [Никитин, 2026b] Никитин А.В., Парадигма потока случайностей 2 // «Академия Тринитаризма», М., Эл № 77-6567, публ.29920, 09.03.2026 URL: <https://www.trinitas.ru/rus/doc/0016/001k/00166056.htm>
- [Никитин, 2026c] Никитин А.В., Парадигма потока случайностей 3 // «Академия Тринитаризма», М., Эл № 77-6567, публ.29932, 17.03.2026. URL: <https://www.trinitas.ru/rus/doc/0016/001k/00166061.htm>

Фундаментальные теоремы и классические работы

- [Гёдель, 1931] Gödel K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme // Monatshefte für Mathematik und Physik, 1931. Vol. 38, 173–198.
- [Тьюринг, 1936] Turing A.M. On Computable Numbers, with an Application to the Entscheidungsproblem // Proceedings of the London Mathematical Society, 1936. Vol. 42, 230–265.
- [Ashby, 1956] Ashby W.R. An Introduction to Cybernetics. — Chapman & Hall, 1956.
- [Shannon, 1948] Shannon C.E. A Mathematical Theory of Communication // Bell System Technical Journal, 1948. Vol. 27, 379–423, 623–656.
- [Aulin, 1979] Aulin A.Y. The Law of Requisite Hierarchy // Kybernetes, 1979. Vol. 8, № 1, 25–30.
- [Turchin, 1977] Turchin V.F. The Concept of a Metasystem Transition // The Cybernetics of Complex Systems. — Intersystems Publications, 1977.
- [Foerster, 2003a] Foerster H. von. Cybernetics of Cybernetics // Understanding Understanding. — Springer, 2003.
- [Foerster, 2003b] Foerster H. von. On Constructing a Reality // Understanding Understanding. — Springer, 2003.
- [Heylighen, 1995] Heylighen F. The Evolution of Complexity: The Law of Requisite Hierarchy Revisited // World Futures, 1995. Vol. 45, 165–182.

Физическое обоснование уравнения баланса хаоса

- [Bag et al., 1999] Bag B.C., Chaudhuri J.R., Ray D.S. Chaos and information entropy production // Journal of Physics A: Mathematical and General, 1999. Vol. 33, № 46, 8331–8350. DOI: 10.1088/0305-4470/33/46/304.
- [Breyman et al., 1998] Breyman W., Tél T., Vollmer J. Entropy balance, time reversibility, and mass transport in dynamical systems // Chaos, 1998. Vol. 8, № 2, 396–408. DOI: 10.1063/1.166322.
- [Ge & Qian, 2009] Ge H., Qian H. The physical origins of entropy production, free energy dissipation and their mathematical representations // Physical Review E, 2009. Vol. 81, № 5, 051133. DOI: 10.1103/PhysRevE.81.051133.

Измерение информационного потока и channel capacity

- [Crutchfield & Feldman, 2001] Crutchfield J.P., Feldman D.P. Synchronizing to the Environment: Information Theoretic Constraints on Agent Learning // *Advances in Complex Systems*, 2001. Vol. 4, № 2–3, 251–264. DOI: 10.1142/S021952590100016X.
- [Lawless et al., 2022] Lawless W.F., Mittu R., Sofge D.A., Shortell T., McDermott T. *Systems Engineering and Artificial Intelligence*. — Springer, 2022. DOI: 10.1007/978-3-030-97283-5.

Аллостатическая нагрузка и управляемая нестабильность

- [Sterling & Eyer, 1988] Sterling P., Eyer J. Allostasis: A new paradigm to explain arousal pathology // *Handbook of Life Stress, Cognition and Health*. — John Wiley & Sons, 1988.
- [McEwen, 1998] McEwen B.S. Stress, Adaptation, and Disease: Allostasis and Allostatic Load // *Annals of the New York Academy of Sciences*, 1998. Vol. 840, № 1, 33–44.
- [Winter, 1989] Winter D.A. Biomechanics of normal and pathological gait: implications for understanding human locomotor control. — *Journal of Motor Behavior*, 1989.
- [McGeer, 1990] McGeer T. Passive dynamic walking // *The International Journal of Robotics Research*, 1990. Vol. 9, № 2, 62–82.
- [Orozco, 2022] Orozco V. Predictive Allostasis: Overcoming Allostatic Challenges Through Predictive Robot Regulatory Behavior // *GitHub*, 2022.

Программируемые автоматы и выбор

- [Wilson et al., 2021] Wilson R.C., Bonawitz E., Collins A.G.E., et al. Ten simple rules for the computational modeling of behavioral data // *eLife*, 2021. Vol. 10, e71147.
- [Sutton & Barto, 2018] Sutton R.S., Barto A.G. *Reinforcement Learning: An Introduction*. — 2nd ed. — MIT Press, 2018.
- [Simon, 1955] Simon H.A. A Behavioral Model of Rational Choice // *Quarterly Journal of Economics*, 1955. Vol. 69, № 1, 99–118.
- [Gigerenzer, 2008] Gigerenzer G. *Rationality for Mortals*. — Oxford University Press, 2008.
- [Russell & Norvig, 2021] Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. — 4th ed. — Pearson, 2021.
- [Rational Choice Theory, 2010] *Stanford Encyclopedia of Philosophy*. Rational Choice Theory. — 2010.
- [McShea, 2023] McShea D.W. Evolutionary trends and goal directedness // *Biological Theory*, 2023. Vol. 18, 89–104.
- [Wadhams & Armitage, 2004] Wadhams G.H., Armitage J.P. Making sense of it all: bacterial chemotaxis // *Nature Reviews Molecular Cell Biology*, 2004. Vol. 5, 1024–1037.
- [Morignot & Hayes-Roth, 1995] Morignot P., Hayes-Roth B. Goal selection in adaptive agents // *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 1995. Vol. 1, 753–759.
- [Fuzzy Logic Control, 2025] *Fuzzy Logic Control: Principles and Applications*. — ResearchGate, 2025.
- [Fuzzy Finite State Machines, 2015] *Fuzzy Finite State Machines: A Review*. — *International Journal of Fuzzy Systems*, 2015.

Катастрофы как системные провалы управления

- [Germanwings 9525 Report, 2016] *Germanwings Flight 9525 Crash Report*. — BEA, 2016.

[Überlingen Crash Report, 2004] Überlingen Mid-Air Collision Investigation Report. — BFU, 2004.

[Botvinnik vs Tal, 1960] Botvinnik M., Tal M. World Chess Championship Matches. — 1960.

Современные исследования по неспецифицированности моделей и эпистемическим ловушкам

[Somalwar et al., 2025] Somalwar A., Lee B.D., Pappas G.J., Matni N. Learning with Imperfect Models: When Multi-step Prediction Mitigates Compounding Error // arXiv, 2025.

[Lobel, 2024] Lobel S. An Optimal Tightness Bound for the Simulation Lemma // arXiv, 2024.

[Amortila et al., 2024] Amortila P., Cao T., Krishnamurthy A. Mitigating Covariate Shift in Misspecified Regression with Applications to Reinforcement Learning // arXiv, 2024.

[Xu et al., 2026] Xu X., Qu J., Zhang Q., Lu C., Yang Y., Zou N., Hu X. Epistemic Traps: Rational Misalignment Driven by Model Misspecification // arXiv, 2026.

Иерархии само-моделей и экспериментальные прототипы

[Jiang et al., 2026] Jiang S.-Q., Zhang S.-X., Tao S.-D., Zhu X.-H., Qi T.-L., Song X.-H. Self Model for Embodied Artificial Intelligence // Journal of Computer Science and Technology, 2026. DOI: 10.1007/s11390-026-6289-3.