КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ АЛЬ-ФАРАБИ

ЛЕКЦИЯ № 1

Введение. Постановка задачи и технология интеллектуального управления нейрофизическим IoT-устройством в быту и для промышленных технических объектов

Бельгибаев Бауржан Абдрахимович
Д.т.н., доцент · Кафедра «Искусственный интеллект и big data»
«Технологии в интеллектуальных системах управления в IoT» · 4 курс, бакалавриат

1 Цель и задачи лекции

Основа темы: календарный план силлабуса, неделя 1. Последующее теоретическое раскрытие, формулы и примеры являются учебно-методическим расширением темы силлабуса.

Цель — сформировать системное представление об интеллектуальном управлении физическими техническими объектами, подключёнными к Интернету вещей (IoT), и научить формализовать задачу проектирования нейрофизического устройства с обратной связью, локальным вычислением и возможностью удалённого мониторинга.

- Раскрыть содержание понятий «технический объект», «киберфизическая система», «IoT-устройство» и «интеллектуальная система управления».
- Показать функциональную структуру интеллектуального контура управления и информационные потоки между датчиками, вычислителем, исполнительными механизмами и облаком.
- Сопоставить классические алгоритмы ПИД-регулирования с нечётким, адаптивным и нейросетевым управлением.
- Рассмотреть математическую постановку задачи управления с ограничениями и критериями качества.
- Разобрать бытовой и промышленный примеры, включая риски задержек связи и требования безопасности.

2 План лекции

1. Предпосылки появления интеллектуальных IoT-систем управления
2. Понятие и состав нейрофизического IoT-устройства
3. Архитектура измерения, обработки информации и управления
4. Математическая постановка задачи управления
5. ПИД-, нечёткое и нейросетевое управление: сравнительный анализ
6. Протоколы связи, периферийные вычисления и кибербезопасность
7. Практические примеры и методика оценки эффективности

3 Основные понятия и технологические предпосылки

Интернет вещей (Internet of Things, IoT) — концепция объединения физических устройств, снабжённых датчиками, вычислительными средствами и сетевыми интерфейсами, для обмена данными и предоставления цифровых сервисов. Для задачи управления принципиально важно отличать передачу измерений от замкнутого контура регулирования: наличие Wi-Fi или MQTT само по себе не делает устройство интеллектуальным.

Киберфизическая система (Cyber-Physical System, CPS) объединяет физический процесс, вычислительную модель, коммуникационную инфраструктуру и алгоритмы воздействия на объект. Интеллектуальная система управления (ИСУ) использует методы представления знаний, обучения, оптимизации, прогнозирования или адаптации, чтобы улучшать управление при неопределённости, изменении параметров и сложных внешних воздействиях.

3 Основные понятия и технологические предпосылки (продолжение)

Термин «нейрофизическое IoT-устройство» в данной лекции используется как учебное обозначение физического устройства, в контур которого встроена нейросетевая обработка сенсорных данных либо нейросетевая часть алгоритма управления. Это не отдельный общепринятый класс оборудования. Нейронная сеть может выполнять классификацию состояния, оценку неизвестных параметров, прогноз или формирование управляющего воздействия.

3.1. Отличие автоматизации от интеллектуализации

Традиционная автоматизация обеспечивает воспроизводимое выполнение заранее заданного алгоритма. Интеллектуализация добавляет способность учитывать сложные зависимости, обобщать данные и адаптировать решение к состоянию объекта. Однако применение машинного обучения не отменяет требований к устойчивости, ограничению сигналов, диагностике датчиков и независимой аварийной защите.

3 Основные понятия и технологические предпосылки (продолжение)

Признак	Традиционная автоматизация	Интеллектуальное управление
Основной метод	Логика, ПИД, модельные регуляторы	Прогноз, адаптация, ML, нечёткие правила
Работа с неопределённостью	Заложенные допуски и настройки	Оценка состояния и адаптация в допустимых границах
Требования к данным	Измерения для обратной связи	Дополнительно данные для обучения и проверки
Риски	Ошибки модели и настройки	Также смещение данных и выход за область обучения

4 Функциональная архитектура интеллектуального IoT-устройства

Типовая система состоит из объекта управления, сенсорного слоя, локального контроллера, исполнительных механизмов, сетевого шлюза и верхнего уровня аналитики. Для непрерывного физического процесса быстрый и критически важный контур управления предпочтительно замыкать локально: зависимость от внешнего сервера увеличивает задержку и создаёт дополнительную точку отказа.

Структурная последовательность: физический объект → датчики → фильтрация и оценка состояния → алгоритм управления (ПИД/нечёткая логика/нейросеть) → драйвер исполнительного механизма → физический объект. Параллельная ветвь: локальный контроллер ↔ сетевой шлюз ↔ сервер/панель мониторинга. Аварийная защита действует независимо от удалённого интерфейса.

4.1. Сенсорный уровень

В бытовой автоматизации применяются датчики температуры, влажности, освещённости, движения и качества воздуха; в промышленности — датчики давления, расхода, уровня, вибрации, тока и положения. Измерение описывают как $y_m(k) = h(x(k)) + v(k)$, где x — истинное состояние, h — функция измерения, v — шум. Практически необходимы калибровка, проверка диапазона, временные метки и обработка пропусков.

Функциональная архитектура интеллектуального IoT-устройства (продолжение)

4.2. Вычислительный уровень

Микроконтроллеры подходят для детерминированных циклов измерения и управления при ограниченных ресурсах. Одноплатные компьютеры (например, Raspberry Pi) обеспечивают более сложные вычисления, компьютерное зрение и сетевые сервисы, но обычная ОС не гарантирует жёстких временных ограничений без специальных решений. Для критических промышленных функций необходимы соответствующие контроллеры, архитектура защиты и верифицированные механизмы отказобезопасности.

4.3. Исполнительный уровень

Управляющее воздействие реализуется через реле, полупроводниковые ключи, частотные преобразователи, сервоприводы, клапаны или нагревательные элементы. Следует учитывать физические ограничения: насыщение, скорость изменения команды, минимальную длительность переключения и безопасное состояние при обрыве связи.

4 Контур интеллектуального управления

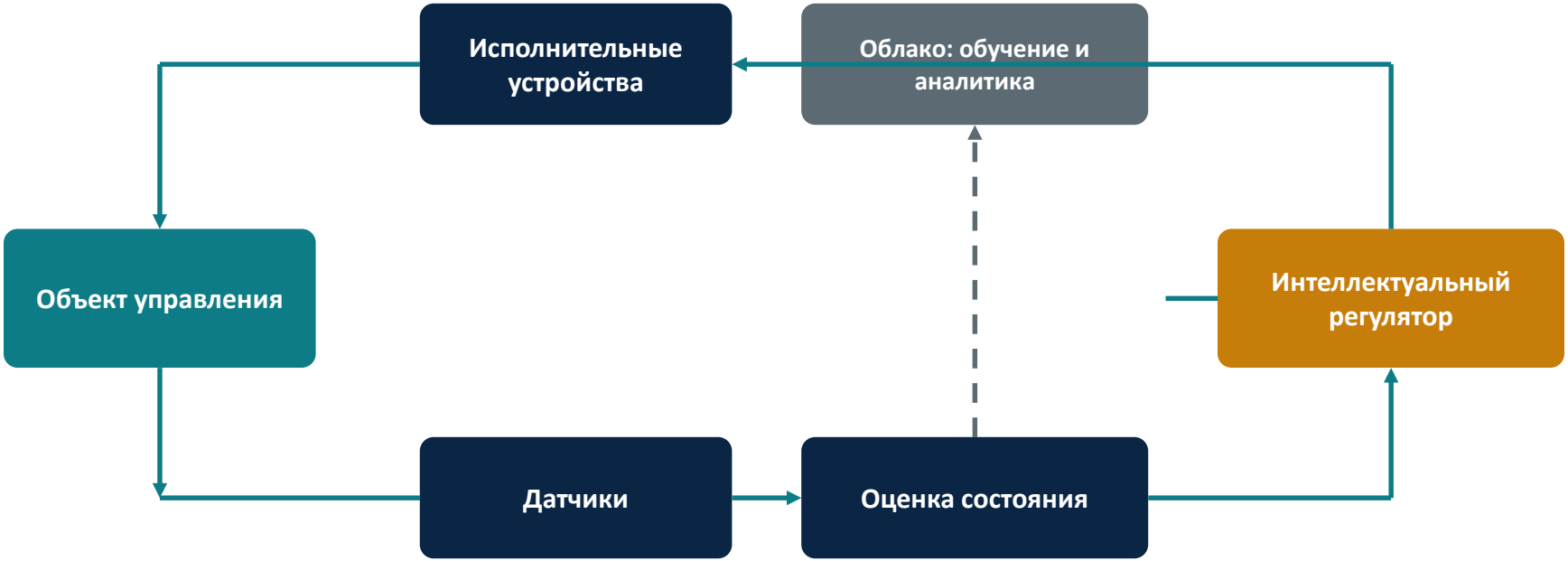

Рис. 1. Замкнутый контур: измерение → оценка → решение → воздействие; штрих — обмен с облаком.

5 Математическая постановка задачи интеллектуального управления

Рассмотрим дискретную модель технического объекта:

$$x(k+1) = f(x(k), u(k), d(k), \theta) + w(k), \quad y(k) = h(x(k)) + v(k),$$

где x(k) — вектор состояния, u(k) — управляющее воздействие, d(k) — измеряемые возмущения, θ — неизвестные или изменяющиеся параметры объекта, w(k) и v(k) — процессный и измерительный шумы, y(k) — выход системы.

Необходимо найти закон u(k)=π(x̂(k), r(k), z(k)), который обеспечивает слежение за заданием r(k), устойчивую работу при возмущениях и соблюдение ограничений; x̂ — оценка состояния, z — дополнительные признаки, включая выход нейросетевого блока.

В качестве показателя качества на горизонте N можно использовать функционал:

$$J = \Sigma[k=0\ldots N-1]\ \{\ (y(k)-r(k))^{T} Q (y(k)-r(k)) + u(k)^{T} R u(k) + \Delta u(k)^{T} S \Delta u(k)\ \},$$

5 Математическая постановка задачи интеллектуального управления (продолжение)

где Q, R и S — положительно полуопределённые весовые матрицы, $\Delta u(k)=u(k)-u(k-1)$. Минимизация J проводится при ограничениях $u_{min} \le u(k) \le u_{max}$, $x_{min} \le x(k) \le x_{max}$ и при требованиях безопасности.

При проектировании следует отдельно определить период дискретизации T_s, максимальную допустимую задержку τ, требуемую точность и доступные вычислительные ресурсы. Если суммарная задержка измерения, расчёта и передачи сопоставима с характерным временем объекта, качество регулирования может существенно ухудшиться.

6 Классический ПИД-регулятор и интеллектуальные методы

6.1. ПИД-регулирование

Ошибка управления $e(t)=r(t)-y(t)$. Непрерывный ПИД-закон записывается как $u(t)=K_p e(t)+K_i \int e(t)dt+K_d de(t)/dt$. Пропорциональная часть реагирует на текущую ошибку, интегральная компенсирует установившееся отклонение, дифференциальная учитывает скорость изменения ошибки.

Для цифровой реализации применяют дискретизацию, фильтрацию производной и защиту интегратора от накопления при насыщении (anti-windup). ПИД остаётся предпочтительным базовым решением для многих объектов с известной динамикой и строгими требованиями к интерпретируемости.

6.2. Нечёткое управление

Нечёткий регулятор преобразует измеренные величины в лингвистические переменные («ошибка мала», «температура высокая») и применяет базу правил вида: ЕСЛИ ошибка положительная большая И скорость ошибки положительная, ТО уменьшить управляющее воздействие. Этапы: фаззификация, логический вывод, дефаззификация. Сильная сторона — включение экспертных знаний; слабая — необходимость проверки полноты правил и устойчивости.

Классический ПИД-регулятор и интеллектуальные методы (продолжение)

6.3. Нейросетевое управление

Нейронная сеть аппроксимирует нелинейное отображение z=Fϕ(s), где s — входной вектор признаков, ϕ — обучаемые параметры. В ИСУ она может: (1) оценивать состояние; (2) прогнозировать выход y(k+1); (3) подбирать параметры ПИД; (4) формировать рекомендованную команду управления. Прямая выдача команд нейросетью требует особенно тщательной валидации и ограничения безопасного диапазона.

Гибридная архитектура сохраняет проверенный регулятор в основном контуре, а нейросеть использует как вспомогательный прогнозирующий блок. Например, u(k)=sat(u_PID(k)+αu_NN(k)), где α задаёт допустимый вклад нейросети, а sat ограничивает команду. Такая формула описывает возможную учебную схему, но сама по себе не доказывает устойчивость.

6 Классический ПИД-регулятор и интеллектуальные методы (продолжение)

Критерий	ПИД	Нечёткий регулятор	Нейросетевой блок
Интерпретируемость	Высокая	Средняя–высокая	Зависит от модели
Необходимость обучающих данных	Не обязательна	Не обязательна	Обычно существенна
Нелинейности	Ограниченные возможности	Правила для нелинейных режимов	Аппроксимация сложных зависимостей
Верификация	Развитые инженерные методы	Проверка правил и режимов	Дополнительно проверка данных и обобщения

6 Сравнение подходов к управлению

Метод	Основа	Интерпретируемость	Адаптивность	Вычислительные затраты
ПИД-регулятор	Ошибка и коэффициенты	Высокая	Низкая	Низкие
Нечёткое управление	Экспертные правила	Высокая	Средняя	Низкие – средние
Нейросетевое управление	Данные и обучение	Низкая	Высокая	Высокие

Таблица. Качественное сравнение (обобщённая оценка, зависит от реализации).

7 Связь и распределение вычислений в IoT

Сетевой обмен в IoT может строиться на MQTT, HTTP(S), CoAP и промышленных протоколах. MQTT поддерживает модель публикации и подписки через брокер; он удобен для телеметрии и диспетчеризации, но не заменяет детерминированный контур управления. HTTP(S) широко применяется для конфигурации и API; выбор транспорта определяется требованиями к надёжности, задержке, энергии и безопасности.

Архитектура edge–cloud разделяет задачи. На периферии выполняются сбор данных, локальная оценка состояния, защита и оперативное управление. На сервере — долговременное хранение, агрегированная аналитика, переобучение моделей и визуализация. При потере Интернета локальный объект должен переходить в заранее определённый режим, а не продолжать принимать устаревшие команды.

7.1. Основные угрозы

Угрозы включают подмену измерений, несанкционированное изменение уставок, повтор старых команд, отказ в обслуживании и компрометацию прошивки. Базовые меры: аутентификация устройств, шифрование канала, контроль прав, журналирование, обновления ПО, проверка целостности, сегментация сети и физические ограничения команд. Требования функциональной безопасности и кибербезопасности следует рассматривать раздельно, хотя они взаимосвязаны.

8

Практический пример 1. Умная система поддержания температуры

Объект: бытовая тепловая камера с нагревателем, датчиком температуры и локальным IoT-контроллером. Задание r=35 °C; диапазон допустимой рабочей температуры 33–37 °C (условные учебные параметры). Контроллер считывает температуру раз в секунду, фильтрует измерение, вычисляет ошибку и управляет мощностью нагревателя через исполнительный модуль.

Упрощённая тепловая модель: $C \cdot dT/dt = P \cdot \eta - K(T - T_a) + q_d$, где C — тепловая ёмкость, P — подаваемая мощность, η — коэффициент преобразования, K — коэффициент теплопотерь, T_a — температура среды, q_d — внешнее тепловое возмущение. Для нейросетевого прогноза могут использоваться последние значения T, P, T_a и скорость изменения температуры.

Порядок исследования: снять экспериментальную переходную характеристику; настроить базовый ПИД; собрать данные при нескольких возмущениях; обучить модель прогноза; сравнить ошибку слежения, перерегулирование, энергопотребление и число срабатываний защиты. Аварийное отключение нагрева при превышении аппаратного порога должно оставаться независимым от нейросетевой модели.

Практический пример 2. Промышленный узел мониторинга электропривода

Объект: электродвигатель конвейера с датчиками тока, температуры и вибрации, локальным контроллером и удалённой панелью мониторинга. Классический контур поддерживает скорость привода, а интеллектуальный модуль анализирует временные окна измерений и оценивает вероятность аномального режима.

Входные признаки могут включать среднеквадратичное значение вибрации, спектральные компоненты, ток, температуру и нагрузку. Выход модели — оценка состояния или диагностический класс. Система выдаёт предупреждение оператору и рекомендует осмотр; автоматическое отключение допускается только при заранее проверенной логике и требованиях безопасности оборудования.

В отличие от бытового примера здесь особенно важны электромагнитная совместимость, устойчивость к отказам, синхронизация времени, защита промышленной сети и разделение диагностического канала и контура управления приводом.

10 Методика проектирования и экспериментальной оценки

1. Определить объект, технологические ограничения и перечень опасных состояний.
2. Составить функциональную схему и таблицу входов/выходов контроллера.
3. Задать критерии качества, допустимые задержки и период дискретизации.
4. Выбрать локальный регулятор и определить его базовые настройки.
5. Организовать сбор репрезентативных данных, включая переходные и нештатные режимы.
6. Разделить данные на обучающую, валидационную и независимую тестовую выборки без утечки по времени или объектам.
7. Сопоставить базовый и интеллектуальный методы на одинаковых сценариях.
8. Проверить поведение при обрыве связи, ошибках датчика, насыщении привода и выходе за область обучения.
9. Подготовить воспроизводимый отчёт с параметрами оборудования, схемой, кодом, графиками и ограничениями исследования.

10 Методика проектирования и экспериментальной оценки (продолжение)

Для количественного сравнения применяют $MAE=(1/N)\Sigma|r(k)-y(k)|$, $RMSE=\sqrt{[(1/N)\Sigma(r(k)-y(k))^2]}$, максимальное перерегулирование, время установления, энергопотребление и долю нарушений ограничений. При диагностике дополнительно оценивают precision, recall, F1 и матрицу ошибок. Выбор метрик должен соответствовать реальной задаче, а не только точности нейросетевой модели.

11 Выводы

Интеллектуальная IoT-система управления объединяет физический объект, измерения, локальные алгоритмы, коммуникационную инфраструктуру и средства аналитики. Её инженерная ценность определяется не наличием нейросети, а измеримым улучшением качества, адаптивности или диагностики при сохранении безопасности. Для быстрых и критических контуров локальная автономность является базовым принципом. Сравнение с классическим регулятором, корректная постановка эксперимента и проверка ограничений обязательны.

12 Контрольные вопросы

1. Чем IoT-устройство отличается от киберфизической системы управления?
2. Какие признаки позволяют отнести систему к интеллектуальным?
3. Почему наличие нейросети не гарантирует улучшение качества управления?
4. Назовите основные элементы замкнутого контура управления.
5. Какие задачи целесообразно выполнять локально, а какие — в облаке?
6. Как интерпретируются переменные дискретной модели состояния?
7. Для чего вводятся ограничения на $u(k)$ и $\Delta u(k)$?
8. Какие преимущества и ограничения имеет ПИД-регулятор?
9. Как формируется база правил нечёткого регулятора?
10. Какие функции нейронная сеть может выполнять без прямого управления приводом?
11. Почему важно разделять обучающую и независимую тестовую выборки?

12 Контрольные вопросы (продолжение)

12. Как сетевые задержки и потери пакетов влияют на управление?
13. Какие требования предъявляются к аварийному отключению исполнительного механизма?
14. Какие метрики подходят для сравнения качества регулирования?
15. Предложите архитектуру интеллектуального IoT-мониторинга промышленного электродвигателя.

13

Рекомендуемые источники

Ли П. Архитектура интернета вещей. М.: ДМК Пресс, 2018. (Источник из силлабуса.)

Матюшин А. О. Программирование микроконтроллеров: стратегия и тактика. М.: ДМК, 2016. (Источник из силлабуса.)

Arduino Documentation: https://docs.arduino.cc/learn/ (ресурс из силлабуса).

ThingSpeak Documentation: https://www.mathworks.com/help/thingspeak/ (ресурс из силлабуса).

Siemens Industry Online Support: https://support.industry.siemens.com/ (ресурс из силлабуса).

Дополнительное учебное расширение: базовые учебники по теории автоматического управления, киберфизическим системам и методам машинного обучения; конкретные издания следует выбирать по доступности библиотеки.