ЛЕКЦИЯ № 15 · Проект MATLAB IoT

Перспективы расширения практического применения ThingSpeak в технологии IoT и MATLAB-аналитики

Бельгибаев Бауржан Абдрахимович
Д.т.н., доцент
Кафедра «Искусственный интеллект и big data»
ОП 6B07113 «Интеллектуальные системы управления»

Цель лекции

Систематизировать знания о проектировании IoT-систем на базе ThingSpeak и MATLAB, рассмотреть направления масштабирования, интеллектуального анализа, предиктивного обслуживания и цифровых двойников, а также сформировать критерии выбора архитектуры и оценки инженерного результата.

План лекции

1. Эволюция IoT: от телеметрии к интеллектуальным системам
2. Архитектуры расширения ThingSpeak и MATLAB
3. Масштабирование каналов, устройств и потоков данных
4. Предобработка, качество и синхронизация измерений
5. Прогнозирование временных рядов и обнаружение аномалий
6. Предиктивное обслуживание технологического оборудования
7. Цифровые двойники и модели технологических процессов
8. Граничные вычисления и распределённая аналитика
9. Интеграция с промышленными протоколами и SCADA
10. Кибербезопасность, надёжность и управление данными
11. Оценка эффективности и ограничения внедрения
12. Итоговый проект: интеллектуальный мониторинг оборудования

1 Эволюция IoT: от телеметрии к интеллектуальным системам

На начальном этапе внедрения IoT датчики передают измерения в удалённое хранилище, а пользователь наблюдает графики. Следующий этап — автоматизированная обработка, при которой система вычисляет статистические показатели, сравнивает их с допустимыми диапазонами и формирует уведомления. Более развитые системы оценивают состояние оборудования и прогнозируют будущие события.

ThingSpeak удобен как учебная и прототипная платформа сбора телеметрии и запуска MATLAB-аналитики. При этом необходимо различать возможности прототипа и промышленной системы: доступность, пропускная способность, требования к безопасности и стоимость сопровождения могут существенно отличаться.

Переход к интеллектуальной аналитике требует не только алгоритма, но и качественных исходных данных. Если показания датчика содержат систематическую ошибку, модель машинного обучения может научиться воспроизводить эту ошибку. Поэтому жизненный цикл IoT-проекта включает измерение, проверку, хранение, моделирование, валидацию и мониторинг качества модели.

1 Эволюция IoT: от телеметрии к интеллектуальным системам (продолжение)

Перспективные области применения: мониторинг микроклимата, энергетический учёт, контроль сельскохозяйственных объектов, диагностика насосов и электродвигателей, учебные мобильные роботы, производственные линии и логистические комплексы. Для каждой области выбираются собственные параметры и показатели качества.

2 Архитектуры расширения ThingSpeak и MATLAB

Базовая архитектура состоит из датчика, локального устройства Raspberry Pi или Arduino, сети, канала ThingSpeak и MATLAB-скрипта. Расширенная архитектура может включать шлюз, локальную очередь сообщений, базу исторических данных, сервис уведомлений и панель оператора.

Для небольшого учебного стенда допустима прямая передача нескольких показателей. При росте числа датчиков удобнее вводить слой шлюза, который проверяет форматы, преобразует единицы, назначает идентификаторы устройств и управляет повторными отправками.

В проекте следует разделять три контура: сбор телеметрии, аналитика и управление. Сбор должен сохранять измерения при кратковременной потере сети; аналитика может выполняться периодически; управление с критическими ограничениями времени и безопасности должно оставаться локальным.

При выборе архитектуры оценивают задержку, доступность интернета, объём данных, число устройств, необходимость исторического хранения, конфиденциальность и возможность миграции на другую платформу. ThingSpeak не следует считать универсальной заменой SCADA или промышленного контроллера.

3 Масштабирование каналов, устройств и потоков данных

Масштабирование означает увеличение количества датчиков, частоты измерений и числа аналитических задач без потери управляемости системы. Простейшая оценка объёма данных: $V = N \cdot f \cdot b \cdot t$, где N — число устройств, f — частота записей в секунду, b — средний объём записи в байтах, t — время наблюдения.

Например, при 20 устройствах, частоте 1 запись в минуту и условном размере записи 100 байт за сутки формируется $20 \cdot 1440 \cdot 100 = 2\,880\,000$ байт, без учёта сетевых заголовков и служебных данных. Это расчётный пример, а не измерение реального трафика ThingSpeak.

Условия использования ThingSpeak, включая частоту обновления, квоты и доступные функции, зависят от актуального плана и лицензии. Перед развёртыванием следует проверить официальную документацию и выполнить нагрузочное тестирование на целевом тарифе.

Для уменьшения объёма передачи применяют агрегирование, передачу только изменений и локальное хранение высокочастотных данных. Однако агрегирование может скрыть короткие опасные события, поэтому для критичных сигналов необходимо сохранять достаточную детализацию.

3

Масштабирование каналов, устройств и потоков данных (продолжение)

Нужно определить единый формат сообщений: идентификатор устройства, время UTC, номер последовательности, измеряемая величина, единица, признак качества. Это упрощает последующее объединение данных из разных источников.

4 Предобработка, качество и синхронизация измерений

Реальные временные ряды могут содержать пропуски, выбросы, дубликаты, неравномерные интервалы и ошибки датчиков. До построения модели выполняют проверку диапазона, анализ временных меток, фильтрацию и документирование отброшенных записей.

Показатели качества включают долю валидных измерений Q = Nvalid/Ntotal и долю потерь L = Nmissing/Nexpected. Значения Nmissing можно оценивать по последовательным номерам сообщений или по ожидаемой частоте дискретизации с учётом допустимого джиттера.

Для медленно меняющейся температуры скользящее среднее уменьшает случайный шум. Для сигналов вибрации такое сглаживание может удалить диагностически важные пики. Метод обработки должен соответствовать физике измеряемого процесса.

Если несколько устройств передают данные с разными часами, сравнение временных рядов без синхронизации может приводить к ложным выводам. Следует использовать согласованный часовой пояс, желательно UTC, и регистрировать время измерения отдельно от времени приёма облаком.

При заполнении пропусков интерполяцией необходимо сохранять отметку, какие точки были восстановлены. Интерполированное значение — оценка, а не новое физическое измерение.

4

Предобработка, качество и синхронизация измерений (продолжение)

```
% MATLAB: проверка и обработка учебного ряда
T = [25.0 25.3 NaN 25.9 26.1 50.0 26.5];
valid = isfinite(T) & T >= -20 & T <= 45;
quality = sum(valid)/numel(T);
fprintf('Доля корректных данных: %.1f %%\n',100*quality);
Tclean = T; Tclean(~valid) = NaN;
Tfilled = fillmissing(Tclean,'linear');
plot(T,'o-'); hold on; plot(Tfilled,'-'); grid on;
legend('Исходные','Восстановленные');
```

5 Прогнозирование временных рядов и обнаружение аномалий

Прогнозирование оценивает будущее значение технологического параметра на основании предыдущих наблюдений. Простейшая модель линейного тренда имеет вид $y(t)=a\cdot t+b$. Коэффициенты можно оценить методом наименьших квадратов, а затем вычислить прогноз для заданного горизонта.

В MATLAB функции polyfit и polyval позволяют построить простую регрессионную модель. Однако линейный тренд плохо описывает циклические процессы и режимные переключения. Для сложных временных рядов могут применяться авторегрессионные модели, деревья решений и нейронные сети при наличии соответствующих инструментов и данных.

Аномалией называют наблюдение, которое существенно отличается от ожидаемого поведения. Один из простых подходов — сравнение с диапазоном $\mu\pm k\sigma$. Он требует осторожности: при нестационарном процессе распределение меняется со временем, а реальные технологические переходы могут выглядеть как выбросы.

Для оценки прогнозов используют $MAE = (1/n)\Sigma|y_i-\hat{y}_i|$ и $RMSE = \mathrm{sqrt}[(1/n)\Sigma(y_i-\hat{y}_i)^2]$. Модель необходимо проверять на данных, не использованных при обучении, соблюдая временной порядок: будущие наблюдения нельзя включать в обучение при оценке прошлого прогноза.

5

Прогнозирование временных рядов и обнаружение аномалий (продолжение)

Результат алгоритма обнаружения аномалий следует рассматривать как сигнал к проверке, а не как доказательство неисправности. Важны доля ложных тревог, пропусков опасных событий и время обнаружения.

```
% MATLAB: прогноз на основе линейного тренда
x = (1:8)';
y = [24.1;24.3;24.6;24.7;25.0;25.3;25.5;25.7];
coef = polyfit(x,y,1);
xFuture = (9:11)';
yFuture = polyval(coef,xFuture);
plot(x,y,'o',xFuture,yFuture,'--o'); grid on;
xlabel('Номер отсчёта'); ylabel('Температура, °C');
legend('Измерения','Учебный прогноз');
```

6 Предиктивное обслуживание оборудования

Предиктивное обслуживание направлено на выявление признаков ухудшения состояния оборудования до отказа. Для электродвигателя могут анализироваться температура корпуса, вибрация, ток и частота вращения; для насоса — давление, расход, ток и температура подшипников.

В ThingSpeak можно сохранять агрегированные диагностические показатели, а MATLAB использовать для оценки трендов и формирования предупреждений. Высокочастотные сигналы вибрации часто целесообразно обрабатывать локально, передавая в облако RMS, спектральные признаки и другие компактные характеристики.

Среднеквадратичное значение вибрации вычисляется как $RMS = \sqrt{[(1/n)\Sigma x_i^2]}$. Рост RMS может указывать на изменение состояния, но сам по себе не определяет причину. Для диагностики требуются базовый режим, сравнимые условия нагрузки и валидация.

Эффективность системы оценивают не только точностью алгоритма, но и полезностью предупреждений: насколько заранее обнаруживается отклонение, сколько ложных срабатываний возникает и какие действия может предпринять персонал.

В учебной работе допускается моделировать постепенный рост вибрации, но в отчёте необходимо явно обозначать искусственный характер данных и не выдавать их за экспериментальные результаты.

7 Цифровые двойники технологических процессов

Цифровой двойник связывает модель физического объекта с актуальными данными и позволяет сравнивать наблюдаемое и ожидаемое поведение. ThingSpeak может выступать источником телеметрии, а MATLAB — средой расчёта модели, оценки параметров и визуализации отклонений.

Для упрощённого теплового объекта можно использовать модель $C \cdot dT/dt = P - k(T - Tamb)$, где C — эффективная теплоёмкость, P — тепловая мощность, k — коэффициент потерь, Tamb — температура окружающей среды. Параметры модели оцениваются по измерениям.

Разность $e(t) = Tmeasured(t) - Tmodel(t)$ называется невязкой. Устойчивый рост $|e(t)|$ может указывать на изменение характеристик объекта, ошибку модели или неисправность датчика. Поэтому интерпретация требует дополнительных проверок.

Цифровой двойник отличается от простого графика: он содержит математическое или вычислительное представление объекта, механизм обновления параметров и проверяемую связь с измерениями. Наличие трёхмерной визуализации не является обязательным признаком цифрового двойника.

При внедрении важно оценивать точность модели на независимых данных, задержку обновления, устойчивость к пропускам и диапазон режимов, в котором модель остаётся применимой.

8 Граничные вычисления и распределённая аналитика

Граничные вычисления (edge computing) предполагают обработку данных непосредственно на Raspberry Pi или другом локальном устройстве. Облако получает уже подготовленные показатели, события и результаты локального анализа.

Преимущества такого подхода — снижение объёма передачи, меньшая зависимость от интернет-соединения и возможность быстрой локальной реакции. Недостатки — ограниченные вычислительные ресурсы, необходимость обновления программ и более сложное управление версиями моделей.

Гибридная архитектура распределяет задачи: локально выполняются сбор, базовая фильтрация и безопасные действия; в ThingSpeak хранится история; в MATLAB анализируются тренды, строятся прогнозы и сравниваются модели.

При передаче только агрегатов важно сохранять возможность расследования событий. Например, при обнаружении аномалии Raspberry Pi может временно записывать более подробные данные в локальный файл для последующего анализа.

Для алгоритмов компьютерного зрения желательно передавать в облако числовые показатели, например количество транспортных средств за интервал, а не непрерывное видео без необходимости. Это снижает сетевую нагрузку и риски конфиденциальности.

9 Интеграция с промышленными протоколами и SCADA

В промышленных системах датчики и контроллеры могут использовать Modbus RTU/TCP, OPC UA и другие интерфейсы. ThingSpeak обычно не заменяет эти протоколы, а подключается через программный шлюз, который считывает данные и публикует их в согласованном формате.

При интеграции следует сохранять инженерные единицы, имена тегов, временные метки и признаки качества. Например, температура может храниться в десятых долях градуса в регистре контроллера; шлюз должен правильно преобразовать значение в °C.

SCADA ориентирована на оперативное наблюдение и управление объектом, тогда как ThingSpeak может использоваться для учебного анализа, облачной визуализации и прототипирования. Возможности, требования к доступности и нормативные ограничения этих решений различаются.

Особое внимание уделяют разделению сетей: прямое подключение критически важных промышленных контроллеров к публичному интернету недопустимо без специально спроектированной защиты. Предпочтительны сегментация, шлюзы и доступ только к необходимым данным.

В отчёте по интеграции следует указать путь данных от регистра или датчика до облачного поля, преобразования единиц, обработку ошибок и правила доступа.

10

Кибербезопасность, надёжность и управление данными

IoT-проект должен защищать конфиденциальность, целостность и доступность данных. Практические меры включают HTTPS, ограничение прав API-ключей, безопасное хранение секретов, обновления устройств и контроль журналов.

Потеря связи не должна приводить к потере критических измерений или к небезопасному состоянию оборудования. Для этого предусматривают локальный буфер, ограниченные повторные отправки, контроль переполнения и автономные механизмы защиты.

Качество данных связано с их происхождением: важно знать, каким устройством, когда и в каких условиях получено значение. Для воспроизводимых исследований сохраняют конфигурацию датчиков, версии кода, коэффициенты калибровки и описание обработки.

Для видеоданных, геолокации и персональной информации могут действовать дополнительные требования к правомерности сбора и срокам хранения. Поэтому следует минимизировать передаваемые сведения и заранее определять цели обработки.

Надёжность оценивают по доле доступного времени, потерянным записям, задержкам передачи и времени восстановления после отказа. Эти показатели необходимо измерять, а не выводить только из наличия облачного сервиса.

11 Оценка эффективности и ограничения внедрения

Инженерная оценка проекта включает технические, экономические и организационные показатели. Технические: точность измерений, полнота данных, задержка, доступность и доля ложных тревог. Экономические: стоимость датчиков, вычислений, связи, сопровождения и обучения персонала.

Для оценки задержки можно использовать $\tau = t_{receive} - t_{measure}$. Средняя задержка полезна, но не отражает редкие длительные сбои; поэтому дополнительно рассматривают медиану, 95-й процентиль и максимальное значение на заданном интервале.

Условная экономическая эффективность определяется сравнением затрат до и после внедрения. Например, сокращение внеплановых простоев может быть преимуществом, но его нельзя утверждать без наблюдений и учёта внешних факторов.

При росте требований к масштабу, хранению и непрерывной работе может потребоваться переход к другой архитектуре. Такой переход следует планировать через открытые форматы, документированные API и независимые от конкретной платформы алгоритмы обработки.

Главный критерий успешности учебного проекта — не количество подключённых технологий, а воспроизводимое решение измеримой задачи с корректной проверкой результатов.

12

Итоговый проект: интеллектуальный мониторинг оборудования

Задача: спроектировать систему мониторинга условного электродвигателя по температуре и вибрации. Raspberry Pi собирает данные, проверяет корректность, сохраняет локальную копию и публикует агрегированные показатели в ThingSpeak. MATLAB вычисляет тренды, пороговые признаки и простую оценку риска.

В учебном наборе используются искусственно сформированные данные: температура постепенно увеличивается от 45 до 56 °C, а показатель вибрации — от 1,2 до 2,4 условных единиц. Для демонстрации выбраны условные пороги 55 °C и 2,2; они не являются эксплуатационными нормативами.

В MATLAB рассчитываются среднее значение, максимумы и логические признаки превышения. Далее строятся два графика и формируется краткий отчёт о числе предупреждений. После этого предлагается добавить модель линейного тренда и оценить прогноз на два шага вперёд.

Проверка включает подстановку NaN, пропуск одного отсчёта и имитацию потери сети. Следует убедиться, что ошибочные значения не трактуются как нормальные, а исходные данные не теряются без регистрации.

12

Итоговый проект: интеллектуальный мониторинг оборудования (продолжение)

Итоговая документация должна включать структурную схему, перечень оборудования, формат сообщений, алгоритм, MATLAB-код, результаты на тестовом наборе, оценку ограничений и предложения по развитию прототипа.

```
% MATLAB: автономный итоговый пример
clear; clc; close all;
time_min = (0:5:35)';
temp_C = [45;46;47;49;51;53;55;56];
vib = [1.2;1.3;1.4;1.5;1.7;1.9;2.2;2.4];
valid = isfinite(temp_C) & isfinite(vib);
alarm = valid & (temp_C>=55 | vib>=2.2);
result = table(time_min,temp_C,vib,valid,alarm);
disp(result);
fprintf('Предупреждений: %d\n',sum(alarm));
figure; plot(time_min,temp_C,'-o'); hold on;
yline(55,'--','Учебный порог'); grid on;
xlabel('Время, мин'); ylabel('Температура, °C');
```

12

Итоговый проект: интеллектуальный мониторинг оборудования (продолжение)

```
figure; plot(time_min,vib,'-s'); hold on;
yline(2.2,'--','Учебный порог'); grid on;
xlabel('Время, мин'); ylabel('Вибрация, усл. ед.');
coef = polyfit(time_min,temp_C,1);
forecast = polyval(coef,[40;45]);
disp(table([40;45],forecast, ...
 'VariableNames',{'Time_min','PredictedTemp_C'}));
```

3 Заключение

В заключительной лекции систематизированы возможности ThingSpeak и MATLAB для построения интеллектуальных IoT-систем: масштабирование телеметрии, предобработка и прогнозирование данных, диагностика оборудования, цифровые двойники, граничные вычисления и интеграция с промышленными системами. Показано, что практическая ценность определяется качеством измерений, надёжностью архитектуры, проверкой моделей и безопасностью эксплуатации. Полученные знания могут служить основой для курсовых проектов и дальнейших инженерных исследований.

Контрольные вопросы (1/2)

1. Какие этапы развития проходит IoT-система от телеметрии к аналитике?

2. В чём отличие прототипа ThingSpeak от промышленной системы?

3. Какие уровни включает расширенная IoT-архитектура?

4. Почему нужно разделять сбор данных, аналитику и управление?

5. Как оценить суточный объём телеметрии?

6. Почему частоту обновления ThingSpeak нужно проверять по текущей лицензии?

7. Какие поля должен содержать пакет измерения?

8. Что показывает доля валидных измерений?

9. Как выявить дубликаты и пропуски?

10. Почему временные метки следует синхронизировать?

11. Какие ограничения имеет линейный прогноз?

12. Что такое MAE и RMSE?

13. Почему обнаружение аномалии не доказывает неисправность?

14. Какие показатели применяются в предиктивном обслуживании?

15. Как вычисляется RMS сигнала?

Контрольные вопросы (2/2)

16. Что отличает цифровой двойник от визуализации?

17. Как используется невязка модели?

18. Что такое edge computing?

19. Какие задачи лучше выполнять локально?

20. Как интегрировать данные Modbus или OPC UA с облаком?

21. Чем SCADA отличается от ThingSpeak?

22. Какие меры необходимы для защиты API-ключей?

23. Как оценивать доступность и задержку системы?

24. Какие критерии применяются для оценки эффективности внедрения?

25. Какие материалы должен содержать итоговый проект?

Самостоятельная работа

1. Разработайте архитектуру мониторинга пяти датчиков с локальным буфером и облачной аналитикой.

2. Рассчитайте объём данных для 30 устройств с периодом измерения 20 секунд за 30 суток; отдельно укажите допущения.

3. Сформируйте MATLAB-скрипт обнаружения пропусков, выбросов и вычисления доли корректных измерений.

4. Постройте прогноз температуры по учебному временному ряду и рассчитайте MAE на отложенных последних отсчётах.

5. Предложите цифровой двойник нагреваемой ёмкости: входы, выходы, уравнение и метод проверки модели.

6. Подготовьте концепцию итогового IoT-проекта с KPI, рисками, схемой защиты и планом испытаний.

Рекомендуемая литература и ресурсы

- Силлабус дисциплины «Проект MATLAB IoT», тема лекции №15.
- MathWorks. ThingSpeak Documentation: https://www.mathworks.com/help/thingspeak/
- MathWorks. MATLAB Documentation: https://www.mathworks.com/help/matlab/
- Raspberry Pi Documentation: https://www.raspberrypi.com/documentation/
- MathWorks. Predictive Maintenance Toolbox Documentation: https://www.mathworks.com/help/predmaint/