Динамични системи - линейни и нелинейни, IoT

Практикум по приложение на графични програмни среди

Владимир Димитров

Катедра Силова Електроника
Факултет по Електронна Техника и Технологии

10 декември 2018 г.

Съдържание

Видове динамични системи

Динамична система

Дефиниция

Система за която взаимовръзката между входните и изходни величини се описва чрез функция, зависеща от времето и нейните производни.

Динамична система

Дефиниция

Система за която взаимовръзката между входните и изходни величини се описва чрез функция, зависеща от времето и нейните производни.

- Методи за описание

Динамична система

Дефиниция

Система за която взаимовръзката между входните и изходни величини се описва чрез функция, зависеща от времето и нейните производни.

- Методи за описание
  - Комбинация от алгебрични системи
  - Диференциално уравнение (непрекъснати), може и комбинация с алгебрични

Динамична система

Дефиниция

Система за която взаимовръзката между входните и изходни величини се описва чрез функция, зависеща от времето и нейните производни.

- Методи за описание
  - Комбинация от алгебрични системи
  - Диференциално уравнение (непрекъснати), може и комбинация с алгебрични
  - Диференчно уравнение (дискретни)

Динамична система

Дефиниция

Система за която взаимовръзката между входните и изходни величини се описва чрез функция, зависеща от времето и нейните производни.

- Методи за описание
  - Комбинация от алгебрични системи
  - Диференциално уравнение (непрекъснати), може и комбинация с алгебрични
  - Диференчно уравнение (дискретни)
  - Комбинация от диференциално и диференчни уравнения - хибридни системи
- Методи за анализ

Динамична система

Дефиниция

Система за която взаимовръзката между входните и изходни величини се описва чрез функция, зависеща от времето и нейните производни.

- Методи за описание
  - Комбинация от алгебрични системи
  - Диференциално уравнение (непрекъснати), може и комбинация с алгебрични
  - Диференчно уравнение (дискретни)
  - Комбинация от диференциално и диференчни уравнения - хибридни системи
- Методи за анализ
  - Аналитично - решаване на уравненията със символи

Динамична система

Дефиниция

Система за която взаимовръзката между входните и изходни величини се описва чрез функция, зависеща от времето и нейните производни.

- Методи за описание
  - Комбинация от алгебрични системи
  - Диференциално уравнение (непрекъснати), може и комбинация с алгебрични
  - Диференчно уравнение (дискретни)
  - Комбинация от диференциално и диференчни уравнения - хибридни системи
- Методи за анализ
  - Аналитично - решаване на уравненията със символи
  - Числено - решаване за конкретни стойности

Динамична система

	Предимства	Недостатъци
Символно	▶ Аналитично решение ▶ Оценка на влиянието на всеки от параметрите	▶ Не винаги има решение ▶ Може получените изрази да са прекалено сложни
Числено	▶ Винаги има решение ▶ Точността може да бъде настройване до желаната ▶ По-лесни	▶ Определяне на стъпка за решаване ▶ Не е подходящо за изводи относно развитието на процесите ▶ Трудно за много нелинейни уравнения

Видове динамични системи

Символно решаване

Символни променливи

- Дефиниране на символни променливи чрез syms x.

Символни променливи

- Дефиниране на символни променливи чрез syms x.
- Основни действия
  - assume(x,'real') - улеснява изчисленията

Символни променливи

- Дефиниране на символни променливи чрез syms x.
- Основни действия
  - assume(x,'real') - улеснява изчисленията
  - expand((x-2)*(x-4)) - разлага

Символни променливи

- Дефиниране на символни променливи чрез syms x.
- Основни действия
  - assume(x,'real') - улеснява изчисленията
  - expand((x-2)*(x-4)) - разлага
  - factor(x^2+2x+1) - общ множител

Символни променливи

- Дефиниране на символни променливи чрез syms x.
- Основни действия
  - assume(x,'real') - улеснява изчисленията
  - expand((x-2)*(x-4)) - разлага
  - factor(x^2+2x+1) - общ множител
  - pretty($x^2 + 6$)

Символни променливи

- Дефиниране на символни променливи чрез syms x.
- Основни действия
  - assume(x,'real') - улеснява изчисленията
  - expand((x-2)*(x-4)) - разлага
  - factor(x^2+2x+1) - общ множител
  - pretty($x^2 + 6$)
  - collect(5x*(x+3)) - подрежда по степен

Символни променливи

- Дефиниране на символни променливи чрез syms x.
- Основни действия
  - assume(x,'real') - улеснява изчисленията
  - expand((x-2)*(x-4)) - разлага
  - factor(x^2+2x+1) - общ множител
  - pretty($x^2 + 6$)
  - collect(5x*(x+3)) - подрежда по степен
  - simplify($cos(x)^2 + sin^2(x)$) - опростява израза използвайки алгебрични правила и единства

Символни променливи

- Дефиниране на символни променливи чрез syms x.
- Основни действия
  - assume(x,'real') - улеснява изчисленията
  - expand((x-2)*(x-4)) - разлага
  - factor(x^2+2x+1) - общ множител
  - pretty($x^2 + 6$)
  - collect(5x*(x+3)) - подрежда по степен
  - simplify($cos(x)^2 + sin^2(x)$) - опростява израза използвайки алгебрични правила и единства
  - subs(x,p,5) - замества в символната функция x p=5. **Много забавя изчислението**

Символни променливи, мерни единици

- Към всяка символна променлива може да се добави мерна единица, която се дефинира $u = \mathrm{symunit}$.

Символни променливи, мерни единици

- Към всяка символна променлива може да се добави мерна единица, която се дефинира u = symunit.
- Символна променлива с мерни единица d = 5*u.m

Символни променливи, мерни единици

- Към всяка символна променлива може да се добави мерна единица, която се дефинира u = symunit.
- Символна променлива с мерни единица d = 5*u.m
- Основни функции

Символни променливи, мерни единици

- Към всяка символна променлива може да се добави мерна единица, която се дефинира u = symunit.
- Символна променлива с мерни единица d = 5*u.m
- Основни функции
  - unitConvert(d,u.cm) - преобразува d в см

Символни променливи, мерни единици

- Към всяка символна променлива може да се добави мерна единица, която се дефинира u = symunit.
- Символна променлива с мерни единица d = 5*u.m
- Основни функции
  - unitConvert(d,u.cm) - преобразува d в см
  - c =newUnit('speedOfLight',3e8*u.m/u.s) - добавяне на нова мерна единица

Символни променливи, мерни единици

- Към всяка символна променлива може да се добави мерна единица, която се дефинира u = symunit.
- Символна променлива с мерни единица d = 5*u.m
- Основни функции
  - unitConvert(d,u.cm) - преобразува d в см
  - c =newUnit('speedOfLight',3e8*u.m/u.s) - добавяне на нова мерна единица
  - checkUnits(eqn,'Compatible') - Сравнява мерните единици от двете страни на уравнение, връща 0, ако са еднакви

Решаване на системи от уравнения

Функция Y = solve(eqns,vars), решава уравненията eqns за променливите vars

Пример

Да се реши системата от уравнения $\frac{x}{4} + y - 1 = 0,\ y - 4x + 3 = 0$

limit(x,t,z) x=x(t) е символна функция, t е независимата променлива, която има гранична стойност z

Пример

Да се намери границата

$$\lim_{a\to\infty} \frac{2a+b}{3a-4}$$

Символно интегриране

Функция int(x,t,c,d), където x=x(t) е символна функция, t е независимата променлива, c и d са съответно долната и горната граница на интеграла

Пример

Да се реши интеграла $\int_0^1 x \log(1+x)\,dx$

Символно диференциране

diff(x,t,n), x=x(t) е символна функция, t е независимата променлива, n е степента на търсената производна

Пример

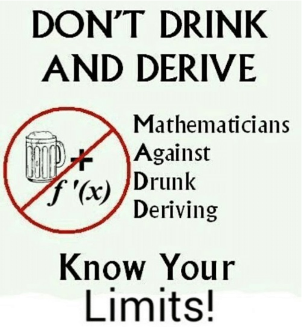
Да се диференцира $\frac{de^{2t}}{dt}$

Символно диференциране

diff(x,t,n), x=x(t) е символна функция, t е независимата променлива, n е степента на търсената производна

Пример

Да се диференцира $\frac{de^{2t}}{dt}$



Know your limits ⚠️

- Динамични системи от първи ред:

$$\frac{dx}{dt} = f(x)$$

- Динамични системи от първи ред:

$$\frac{dx}{dt} = f(x)$$

- Ако $f(x) = Ax$ е линейна функция има аналитично решение

- Динамични системи от първи ред:

$$\frac{dx}{dt} = f(x)$$

- Ако $f(x) = Ax$ е линейна функция има аналитично решение
- Може да се добави независима променлива u и алгебрично уравнение

- Динамични системи от първи ред:

$$\frac{dx}{dt} = f(x)$$

- Ако $f(x) = Ax$ е линейна функция има аналитично решение
- Може да се добави независима променлива u и алгебрично уравнение

$$\frac{dx}{dt} = Ax + Bu$$
$$z = Cx + Du$$

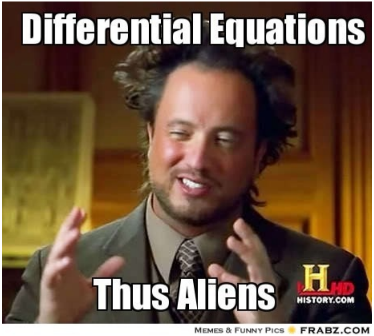
- Динамични системи от първи ред:

$$\frac{dx}{dt} = f(x)$$

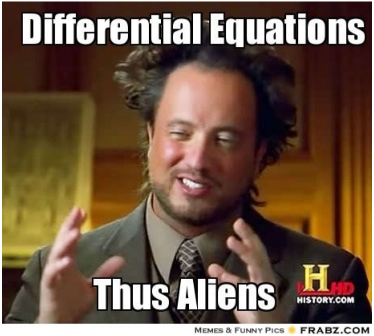
- Ако $f(x) = Ax$ е линейна функция има аналитично решение
- Може да се добави независима променлива u и алгебрично уравнение

$$\frac{dx}{dt} = Ax + Bu$$
$$z = Cx + Du$$

- x се нарича променливи на състоянието и в електрониката обикновено е ток през индуктивност и напрежение на кондензатор

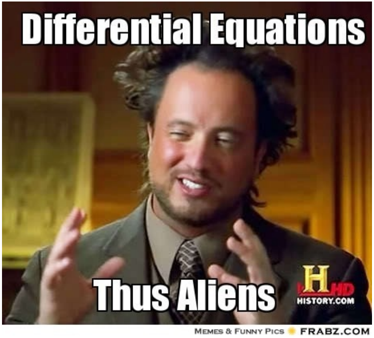

Differential Equations
Thus Aliens
HISTORY.COM
MEMES & FUNNY PICS FRABZ.COM

- Стандартно:



- Стандартно:

$$y = f(x, \frac{dx}{dt}, \frac{d^2x}{dt^2})$$

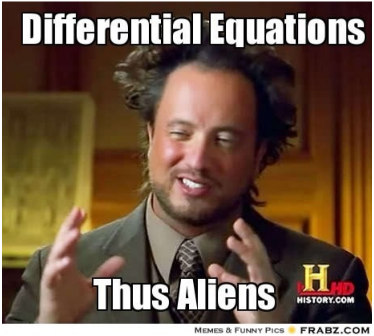


- Стандартно:

$$y = f(x, \frac{dx}{dt}, \frac{d^2x}{dt^2})$$

- Matlab го иска като система от уравнения от първи ред, ако е линейно има вида:

Динамични системи от втори ред

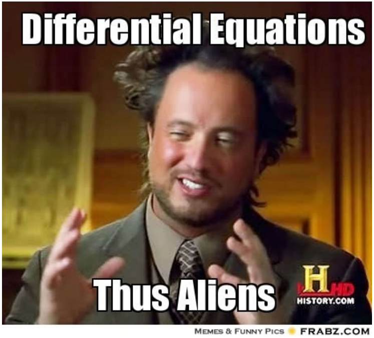


- Стандартно:

$$y = f(x, \frac{dx}{dt}, \frac{d^2x}{dt^2})$$

- Matlab го иска като система от уравнения от първи ред, ако е линейно има вида:

$$\frac{dx}{dt} = Ax + Bu$$
$$z = Cx + Du$$



- Стандартно:

$$y = f(x, \frac{dx}{dt}, \frac{d^2x}{dt^2})$$

- Matlab го иска като система от уравнения от първи ред, ако е линейно има вида:

$$\frac{dx}{dt} = Ax + Bu$$
$$z = Cx + Du$$

- Аналитично решение има за показаната система, както и за линейни системи до четвърти ред

Примери за линейни динамични системи

Пример

Да се преобразува в стандартна форма диференциалното уравнение за напрежението на кондензатора на RLC верига при постоянно входно напрежение.

Пример по желание

Да се състави модел в стандартна форма на понижаващ ключов преобразувател. Полупроводниковите елементи да се приемат за идеални, но да се моделира еквивалентното серийно съпротивление на изходния кондензатор.

- dsolve(eqn,con)- решава уравнението eqn (може да бъде и матрица ред от уравнения), където con е вектор с началните условия

Пример

Да се реши диференциалното уравнение $\frac{dx}{dt} + x = 0$, x(0)=25

Символно решаване на диференциални уравнения

- dsolve(eqn,con)- решава уравнението eqn (може да бъде и матрица ред от уравнения), където con е вектор с началните условия

Пример

Да се реши диференциалното уравнение $\frac{dx}{dt} + x = 0$, x(0)=25

- Ако това ще ви е основното занимание, обмислете алтернативи като Mathematica и Maple, който са по-добри ilaplace(F,s,t) Лапласовия образ на функцията f(t) е F(s), където s е Лапласовия оператор.

Пример

Да се намери оригинала на Лапласовата функция $F(s) = \frac{1}{s+1}$

Видове динамични системи

Числено решаване

- Функция diff(x,n)- намира диференциалната разлика от n-та степен на матрицата x

Пример

Да се диференцира X = [1 1 2 3 5 8 13 21] и y=sin(x) в интервала [-π π] със стъпка h=0.001

Функции в Matlab за числено интегриране

- q=integral('myFun',a,b), myFun е име на функцията, където е записана подинтегралната функция, която ще се интегрира в интервала [a,b].

Функции в Matlab за числено интегриране

- q=integral('myFun',a,b), myFun е име на функцията, където е записана подинтегралната функция, която ще се интегрира в интервала [a,b].
  - Функцията може да бъде анонимна, например myFun = @(x)sin((1:5)*x) или в отделен файл

Функции в Matlab за числено интегриране

- q=integral('myFun',a,b), myFun е име на функцията, където е записана подинтегралната функция, която ще се интегрира в интервала [a,b].
  - Функцията може да бъде анонимна, например myFun = @(x)sin((1:5)*x) или в отделен файл
- Q = trapz(X,Y), намиране на интеграл по метод на Newton-Cotes от 1 ред

Пример

Да се интегрира числено $y = sin^2 x$ за интервала x=[0 2π]

Нелинейни динамични системи

- Общ вид на уравненията

$$\frac{dx}{dt} = f(x, u)$$

- Общ вид на уравненията

$$\frac{dx}{dt} = f(x, u)$$

- f е нелинейна функция

- Общ вид на уравненията

$$\frac{dx}{dt} = f(x, u)$$

- f е нелинейна функция
- Ако, уравнението е не е от първи ред, то уравненията са векторни

- Общ вид на уравненията

$$\frac{dx}{dt} = f(x, u)$$

- f е нелинейна функция
- Ако, уравнението е не е от първи ред, то уравненията са векторни
- Методи за оценка на поведението

- Общ вид на уравненията

$$\frac{dx}{dt} = f(x, u)$$

- f е нелинейна функция
- Ако, уравнението е не е от първи ред, то уравненията са векторни
- Методи за оценка на поведението
  - За системи от първи, втори ред и трети ред - фазов портрет

Нелинейни динамични системи

- Общ вид на уравненията

$$\frac{dx}{dt} = f(x, u)$$

- f е нелинейна функция
- Ако, уравнението е не е от първи ред, то уравненията са векторни
- Методи за оценка на поведението
  - За системи от първи, втори ред и трети ред - фазов портрет
  - За всички видове - числено решение

Демонстрация

Да се реши числено нелинейното диференциално уравнение от първи ред:

$$\frac{dx}{dt} = rx\left(1 - \frac{x}{K}\right)$$

За същото уравнение да се начертае фазовият портрет.

Системи до трети ред - фазов портрет

- Общ вид на система от втори ред

$$\frac{dx_1}{dt} = f_1(x_1, x_2)$$
$$\frac{dx_2}{dt} = f_2(x_1, x_2)$$

Системи до трети ред - фазов портрет

- Общ вид на система от втори ред

$$\frac{dx_1}{dt} = f_1(x_1, x_2)$$
$$\frac{dx_2}{dt} = f_2(x_1, x_2)$$

- Фазовият портрет се използва за оценка на поведението, но без конкретно количествено описание на процесите.

Системи до трети ред - фазов портрет

- Общ вид на система от втори ред

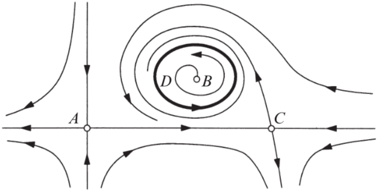
$$\frac{dx_1}{dt} = f_1(x_1, x_2)$$
$$\frac{dx_2}{dt} = f_2(x_1, x_2)$$

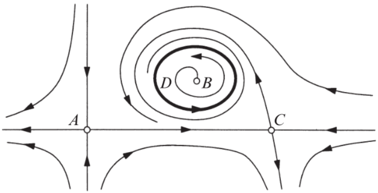
- Фазовият портрет се използва за оценка на поведението, но без конкретно количествено описание на процесите.
- Векторната диаграма чрез quiver(x,y,u,v). За масив от точки (x,y) се рисуват вектори, който имат компоненти (u,v).

- Общ вид на система от втори ред

$$\frac{dx_1}{dt} = f_1(x_1, x_2)$$
$$\frac{dx_2}{dt} = f_2(x_1, x_2)$$

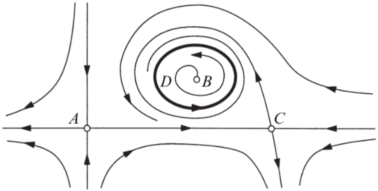
- Фазовият портрет се използва за оценка на поведението, но без конкретно количествено описание на процесите.
- Векторната диаграма чрез quiver(x,y,u,v). За масив от точки (x,y) се рисуват вектори, който имат компоненти (u,v).



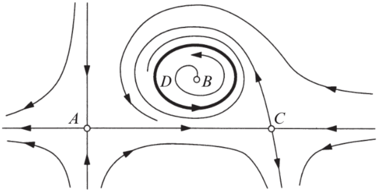


- Точки A,B,C се наричат фиксирани, решават уравнението $f(x^*) = 0$ - установена стойност

Системи до втори ред - фазов портрет



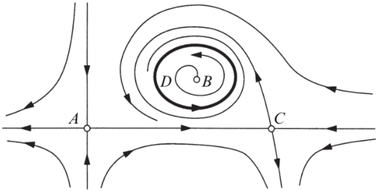
- Точки A,B,C се наричат фиксирани, решават уравнението $f(x^*) = 0$ - установена стойност
- D е затворена орбита - отговарят на периодични решения $x(t+T) = x(t)$



- Точки A,B,C се наричат фиксирани, решават уравнението $f(x^*) = 0$ - установена стойност
- D е затворена орбита - отговарят на периодични решения $x(t + T) = x(t)$

Пример

Да се начертае фазовият портрет на нелинейното диференциално уравнение от втори ред $\frac{d\theta^2}{dt^2} + \frac{g}{L} sin(\theta) = 0$



- Точки A,B,C се наричат фиксирани, решават уравнението $f(x^*) = 0$ - установена стойност
- D е затворена орбита - отговарят на периодични решения $x(t+T) = x(t)$

Пример

Да се начертае фазовият портрет на нелинейното диференциално уравнение от втори ред $\frac{d\theta^2}{dt^2} + \frac{g}{L} sin(\theta) = 0$

- Професионален софтуер за фазови диаграми **XPPAUT** или **PPLANE**, повече и на упражнение 5

Задача

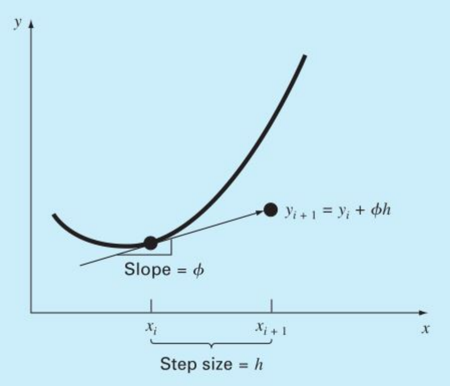
Да се състави система описваща динамиката на взаимоотношенията (чувствата) между Ромео и Жулиета. Динамиката да зависи от собствените чувства и тези на другия.

- Да се състави динамична система в матричен вид, която да описва техните взаимоотношения.
- Да се добавят коефициенти определящи динамиката на техните взаимоотношения в зависимост от собствените чувства и чувствата на другия (какви варианти съществуват?)
- Да се начертае фазовият портрет на системата за различни начални условия.

Дефиниция

Да се реши числено диференциалното уравнение $\frac{dy}{dt} = f(y, t)$

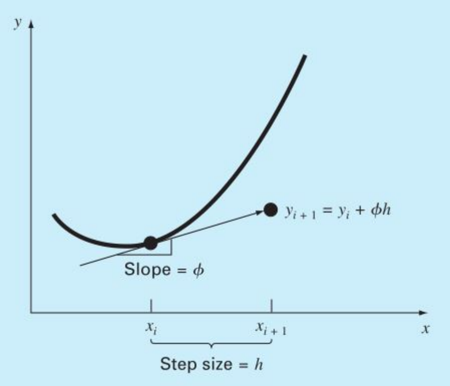
- Идея: Нова стойност = предишна стойност + наклон * стъпка



Дефиниция

Да се реши числено диференциалното уравнение $\frac{dy}{dt} = f(y, t)$

- Идея: Нова стойност = предишна стойност + наклон * стъпка



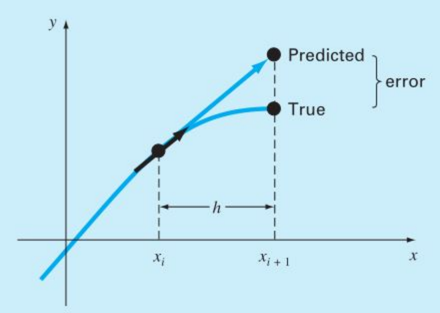
- Как да намерим наклона ϕ?

Методи на Runge-Kutta

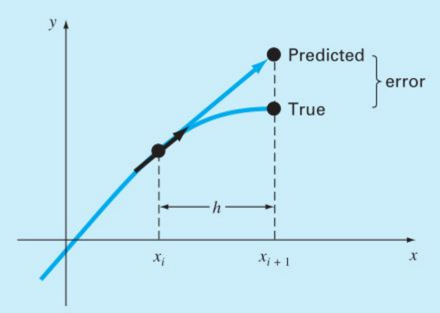
- Метод на Ойлер: $\phi = f\left(y_i, t_i\right)$

Методи на Runge-Kutta

- Метод на Ойлер: $\phi = f\left(y_i, t_i\right)$

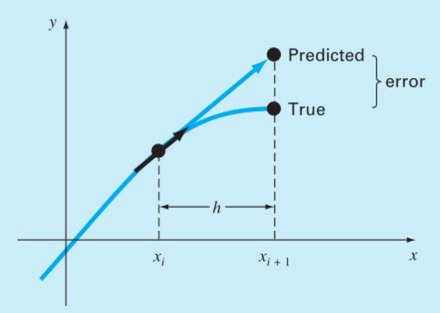


- Метод на Ойлер: $\phi = f\left(y_i, t_i\right)$



- Решението за всяка стъпка се определя съгласно $y_{i+1} = y_i + f\left(y_i, t_i\right) h$

- Метод на Ойлер: $\phi = f(y_i, t_i)$



- Решението за всяка стъпка се определя съгласно $y_{i+1} = y_i + f(y_i, t_i)\, h$

Пример

Да се напише функция за реализиране на метода на Ойлер и да се изпробва върху диференциалното уравнение $\frac{dy}{dt} = y$

Методи на Runge-Kutta

- Метод на Heun: На две стъпки:

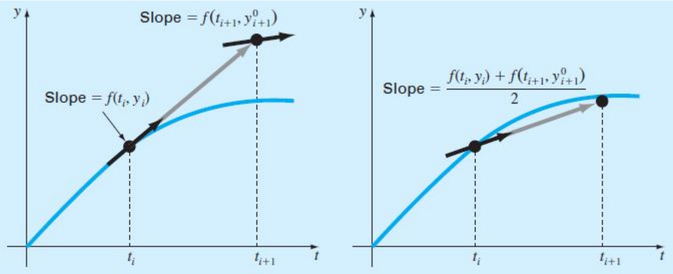
Методи на Runge-Kutta

- Метод на Heun: На две стъпки:
  - Предсказваме с метода на Ойлер $y_p = y_i + f\left(y_i, t_i\right) h$

Методи на Runge-Kutta

- Метод на Heun: На две стъпки:
  - Предсказваме с метода на Ойлер $y_p = y_i + f\left(y_i, t_i\right) h$
  - Усредняваме двете производни $y_{i+1} = y_i + \left(f\left(y_i, t_i\right) + f\left(y_p, t_i\right)\right) \frac{h}{2}$

- Метод на Heun: На две стъпки:
  - Предсказваме с метода на Ойлер $y_p = y_i + f\left(y_i, t_i\right) h$
  - Усредняваме двете производни $y_{i+1} = y_i + \left(f\left(y_i, t_i\right) + f\left(y_p, t_i\right)\right) \frac{h}{2}$



Пример

Да се допълни написаната функция с метода на Heun

Методи на Runge-Kutta

- Метод на Runge-Kutta от четвърти ред $y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)\,\frac{h}{6}$:

- Метод на Runge-Kutta от четвърти ред $y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)\,\frac{h}{6}$:
  - $k_1 = f\,(y_i, t_i)$
  - $k_2 = f\left(y_i + \frac{1}{2}k_1 h, t_i + \frac{h}{2}\right)$
  - $k_3 = f\left(y_i + \frac{1}{2}k_2 h, t_i + \frac{h}{2}\right)$
  - $k_4 = f\,(y_i + k_3 h, t_i + h)$

- Метод на Runge-Kutta от четвърти ред $y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)\,\frac{h}{6}$:
  - $k_1 = f\left(y_i, t_i\right)$
  - $k_2 = f\left(y_i + \frac{1}{2}k_1 h, t_i + \frac{h}{2}\right)$
  - $k_3 = f\left(y_i + \frac{1}{2}k_2 h, t_i + \frac{h}{2}\right)$
  - $k_4 = f\left(y_i + k_3 h, t_i + h\right)$

Пример

Да се допълни написаната функция с метода на RK4. Да се реши диференциалното уравнение $\frac{dy}{dt} = y^2 - y^3, y(0) = \delta$, за интервала $[0, \frac{2}{\delta}]$

- Метод на Runge-Kutta от четвърти ред $y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)\frac{h}{6}$:
  - $k_1 = f(y_i, t_i)$
  - $k_2 = f\left(y_i + \frac{1}{2}k_1 h, t_i + \frac{h}{2}\right)$
  - $k_3 = f\left(y_i + \frac{1}{2}k_2 h, t_i + \frac{h}{2}\right)$
  - $k_4 = f(y_i + k_3 h, t_i + h)$

Пример

Да се допълни написаната функция с метода на RK4. Да се реши диференциалното уравнение $\frac{dy}{dt} = y^2 - y^3, y(0) = \delta$, за интервала $[0, \frac{2}{\delta}]$

- Диференциални уравнения с различни порядъци на времеконстантите се наричат твърди (stiff).

- Метод на Runge-Kutta от четвърти ред $y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)\,\frac{h}{6}$:
  - $k_1 = f\left(y_i, t_i\right)$
  - $k_2 = f\left(y_i + \frac{1}{2}k_1 h, t_i + \frac{h}{2}\right)$
  - $k_3 = f\left(y_i + \frac{1}{2}k_2 h, t_i + \frac{h}{2}\right)$
  - $k_4 = f\left(y_i + k_3 h, t_i + h\right)$

Пример

Да се допълни написаната функция с метода на RK4. Да се реши диференциалното уравнение $\frac{dy}{dt} = y^2 - y^3, y(0) = \delta$, за интервала $[0, \frac{2}{\delta}]$

- Диференциални уравнения с различни порядъци на времеконстантите се наричат твърди (stiff).
- Необходими са специални методи за числено решаване, които да си адаптират стъпката на интегриране в процеса на работа.

Грешки при числени методи

- Основни видове грешки:

Грешки при числени методи

- Основни видове грешки:
  - От дискретизацията

Грешки при числени методи

- Основни видове грешки:
  - От дискретизацията
    - локално на всяка стъпка

- Основни видове грешки:
  - От дискретизацията
    - локално на всяка стъпка
    - глобално в края на изчислението

Грешки при числени методи

- Основни видове грешки:
  - От дискретизацията
    - локално на всяка стъпка
    - глобално в края на изчислението
  - От закръгляването при всяко изчисление

Грешки при числени методи

- Основни видове грешки:
  - От дискретизацията
    - локално на всяка стъпка
    - глобално в края на изчислението
  - От закръгляването при всяко изчисление
- Оценка на локална грешка от дискретизация - ред на Тейлър на $y' = f\left(t, y\right)$

Грешки при числени методи

- Основни видове грешки:
  - От дискретизацията
    - локално на всяка стъпка
    - глобално в края на изчислението
  - От закръгляването при всяко изчисление
- Оценка на локална грешка от дискретизация - ред на Тейлър на $y' = f(t, y)$

$$y_{i+1} = y_i + y_i' h + \frac{y_i''}{2!} h^2 + \ldots + \frac{y_i^n}{n!} h^n h^n + R_n$$

Грешки при числени методи

- Основни видове грешки:
  - От дискретизацията
    - локално на всяка стъпка
    - глобално в края на изчислението
  - От закръгляването при всяко изчисление
- Оценка на локална грешка от дискретизация - ред на Тейлър на $y' = f(t, y)$

$$y_{i+1} = y_i + y_i' h + \frac{y_i''}{2!} h^2 + \ldots + \frac{y_i^n}{n!} h^n h^n + R_n$$

- При метода на Ойлер се пренебрегва всичко над първата производна (Runge-Kutta от първи ред). Грешката има вида

$$E_t = \frac{f'(t_i, y_i)}{2!} h^2 + \mathcal{O}\left(h^3\right)$$

Функции в Matlab за числено решаване

- ode23 - Апроксимация от нисък ред. Използвайте при интегриране за малки интервали или когато точността е по-маловажна от скоростта
- ode45 - Най-често използвано.
- ode15s - Когато диференциалното уравнение има много времеконстанти от различен порядък.
- bvp4c - за решаване на обикновени диференциални уравнения с гранични условия
- dde23 - за решаване на обикновени диференциални уравнения със закъснение $\frac{dy}{dt} = f(t, y(t), y(t-\tau_1), y(t-\tau_2), ..)$
- pdepe - за частни диференциални уравнения

- Общ Синтаксис
  [t,y]=ode45('myODE',[t0,tf],[a1,a2...,an],options,p1,p2)

Функции в Matlab за диференциални уравнения

- Общ Синтаксис
  [t,y]=ode45('myODE',[t0,tf],[a1,a2...,an],options,p1,p2)
- Първият аргумент е име на файл с функцията (или анонимна), която ще взима за аргументи t и y (може допълнително и p1,p2) и връща диференциалното уравнение dy/dt.

- Общ Синтаксис
  [t,y]=ode45('myODE',[t0,tf],[a1,a2...,an],options,p1,p2)
- Първият аргумент е име на файл с функцията (или анонимна), която ще взима за аргументи t и y (може допълнително и p1,p2) и връща диференциалното уравнение dy/dt.
  - Пример: function yprime = myODE(t,y,p1,p2, . . .)

Функции в Matlab за диференциални уравнения

- Общ Синтаксис
  [t,y]=ode45('myODE',[t0,tf],[a1,a2...,an],options,p1,p2)
- Първият аргумент е име на файл с функцията (или анонимна), която ще взима за аргументи t и y (може допълнително и p1,p2) и връща диференциалното уравнение dy/dt.
  - Пример: function yprime = myODE(t,y,p1,p2, . . .)
- Вторият аргумент е интервала за който ще се интегрира.
- Третият аргумент е вектор с началните стойности.

- Общ Синтаксис
  [t,y]=ode45('myODE',[t0,tf],[a1,a2...,an],options,p1,p2)
- Първият аргумент е име на файл с функцията (или анонимна), която ще взима за аргументи t и y (може допълнително и p1,p2) и връща диференциалното уравнение dy/dt.
  - Пример: function yprime = myODE(t,y,p1,p2, . . .)
- Вторият аргумент е интервала за който ще се интегрира.
- Третият аргумент е вектор с началните стойности.
- Функцията връща вектор t за времената и изчислената стойност y за тях.

Допълнителни възможности - options

- Краят на решението на диференциалното уравнение може да бъде по време или при настъпване на определено условие - event

- Краят на решението на диференциалното уравнение може да бъде по време или при настъпване на определено условие - event
- За целта е необходима функция, която да го описва function [gstop,isterminal,direction] = g(t,y)

Допълнителни възможности - options

- Краят на решението на диференциалното уравнение може да бъде по време или при настъпване на определено условие - event
- За целта е необходима функция, която да го описва function [gstop,isterminal,direction] = g(t,y)
  - Условието се случва, когато определен параметър (gstop) стигне 0 (условие isterminal=1).

Допълнителни възможности - options

- Краят на решението на диференциалното уравнение може да бъде по време или при настъпване на определено условие - event
- За целта е необходима функция, която да го описва function [gstop,isterminal,direction] = g(t,y)
  - Условието се случва, когато определен параметър (gstop) стигне 0 (условие isterminal=1).
  - Direction определя допълниетелно дали условието е изпълнено при при спадане на функцията (-1), покачване(1) или без значение ([]).

Допълнителни възможности - options

- Краят на решението на диференциалното уравнение може да бъде по време или при настъпване на определено условие - event
- За целта е необходима функция, която да го описва function [gstop,isterminal,direction] = g(t,y)
    - Условието се случва, когато определен параметър (gstop) стигне 0 (условие isterminal=1).
    - Direction определя допълниетелно дали условието е изпълнено при при спадане на функцията (-1), покачване(1) или без значение ([]).
- Тя се подава като опция на метода използван за числено решение opts = odeset('events',@g)

Допълнителни възможности - options

- Краят на решението на диференциалното уравнение може да бъде по време или при настъпване на определено условие - event
- За целта е необходима функция, която да го описва function [gstop,isterminal,direction] = g(t,y)
  - Условието се случва, когато определен параметър (gstop) стигне 0 (условие isterminal=1).
  - Direction определя допълниетелно дали условието е изпълнено при при спадане на функцията (-1), покачване(1) или без значение ([]).
- Тя се подава като опция на метода използван за числено решение opts = odeset('events',@g)

Пример

Да се намери времето за което диференциалното уравнение $\frac{dy^2}{dt^2} = -1 + y^2, y(0) = 1, \frac{y(0)}{dt} = 0$ достига нула.

Системи от висок ред

- Нелинейните диференциални уравнения до втори ред могат да имат затворени цикли или фиксирани точки - упражнение 5

Системи от висок ред

- Нелинейните диференциални уравнения до втори ред могат да имат затворени цикли или фиксирани точки - упражнение 5
- Тези от трети ред нагоре - могат да имат много по-впечатляващо поведение, което се нарича хаотично

Системи от висок ред

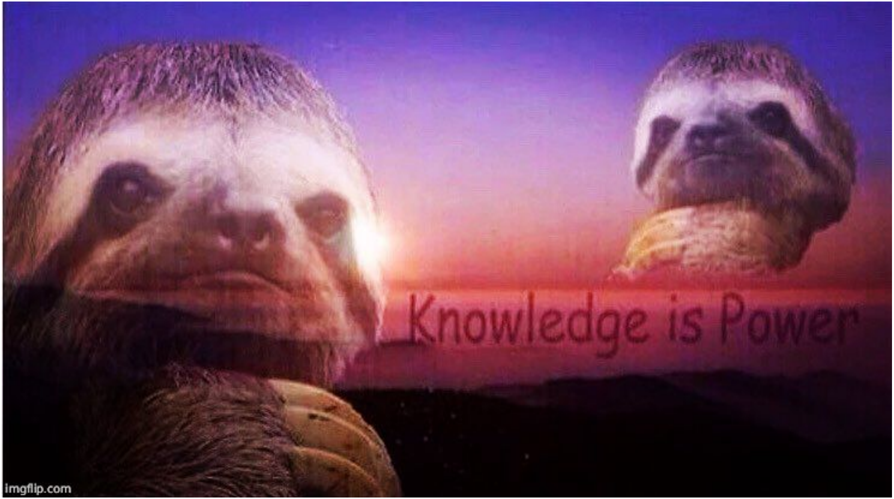
- Нелинейните диференциални уравнения до втори ред могат да имат затворени цикли или фиксирани точки - упражнение 5
- Тези от трети ред нагоре - могат да имат много по-впечатляващо поведение, което се нарича хаотично

Пример

Да се реши Лоренцовата система диференциални уравнения

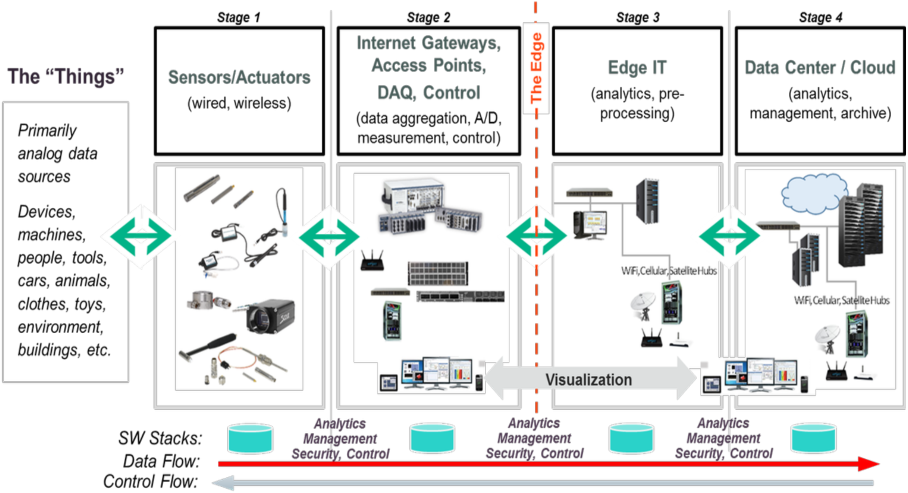
$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = rx - y - xz$$
$$\frac{dz}{dt} = xy - bz$$

$b = \frac{8}{3}$, r=10, $\sigma = 28$


Knowledge is Power
imgflip.com

Internet of Things (IoT)

Internet of Things - Архитектура



Internet of Things - Архитектура

Пет основни компонента за всяка IoT архитектура:

- Сензорите и актуаторите

Пет основни компонента за всяка IoT архитектура:

- Сензорите и актуаторите
- Gateway- връзката между сензорите и крайната им обработка

Internet of Things - Архитектура

Пет основни компонента за всяка IoT архитектура:

- Сензорите и актуаторите
- Gateway- връзката между сензорите и крайната им обработка
- Backend services - процеси за крайна обработка, съхранение и визуализация на данните

Internet of Things - Архитектура

Пет основни компонента за всяка IoT архитектура:

- Сензорите и актуаторите
- Gateway- връзката между сензорите и крайната им обработка
- Backend services - процеси за крайна обработка, съхранение и визуализация на данните
- Протокол за пренос на данни Сензор - Gateway

Internet of Things - Архитектура

Пет основни компонента за всяка IoT архитектура:

- Сензорите и актуаторите
- Gateway- връзката между сензорите и крайната им обработка
- Backend services - процеси за крайна обработка, съхранение и визуализация на данните
- Протокол за пренос на данни Сензор - Gateway
- Протокол за пренос на данни Gateway - Backend

Internet of Things (IoT)

IoT - протоколи Сензор - Gateway

Протоколи Сензор/Gateway - с електрическа връзка

- RS232, RS485, RS422
- I2C
- SPI
- CAN,LIN
- X10
- 1-Wire

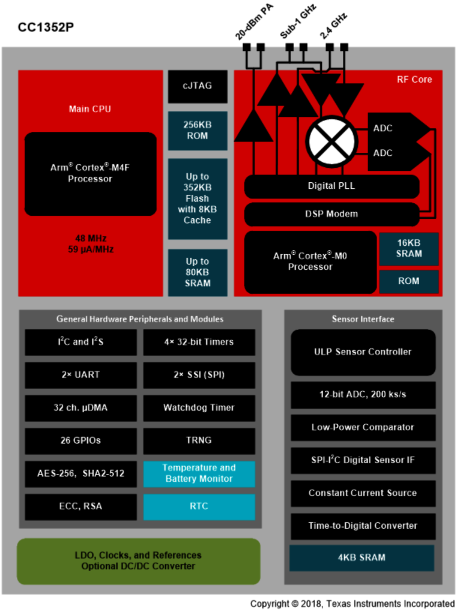
Протоколи Сензор/Gateway - безжични

- 6LowPAN - 250 kbit/s до 100м, до 100 устройства http://www.ti.com/lsds/ti/wireless-connectivity/6lowpan/overview.page
- Bluetooth Low Energy - 5 Mbit/s (v5) до 100м
- Z-Wave - 30 до 100m http://www.z-wave.com/
- ZigBee - 250kbits/s, 100m http://www.zigbee.org/
- INSTEON - 30cm, 40kbit/s, http://www.insteon.com/
- Infrared - 1 Mbit/s до 45 метра
- MiWi http://www.microchip.com/design-centers/wireless-connectivity/embedded-wireless/802-15-4/software/miwi-protoclxo
- Xbee https://www.digi.com/xbee
- Sigfox https://www.sigfox.com/en
- LoRa http://www.microchip.com/design-centers/wireless-connectivity/embedded-wireless/lora-technology
- Weighless http://www.weightless.org/
- NWave https://www.nwave.io/

Internet of Things (IoT)

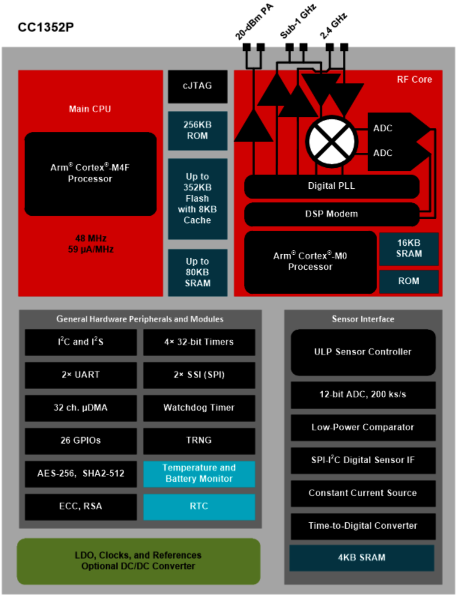
IoT - Gateway реализация

Gateway to hell



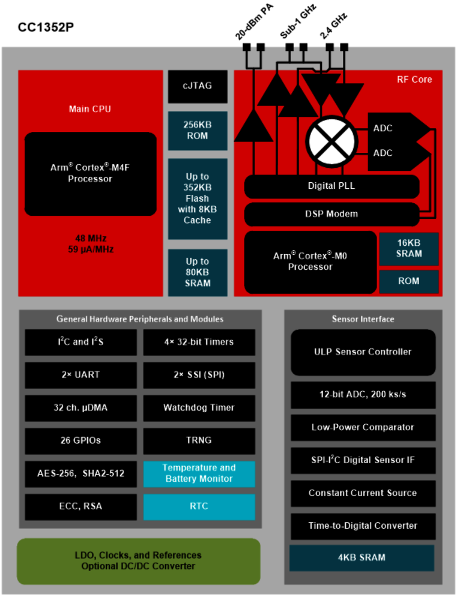
- Микроконтролер

Gateway to hell



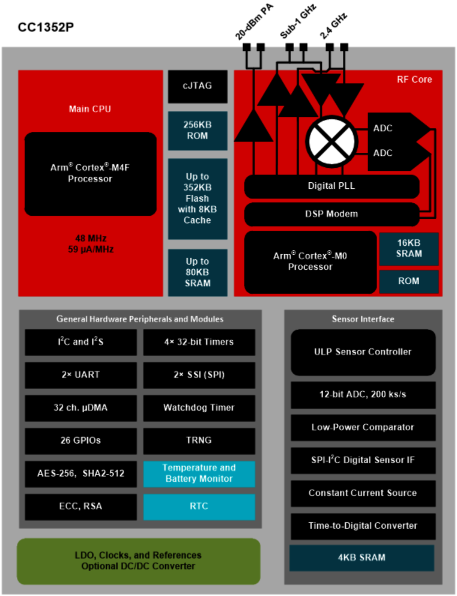
- Микроконтролер
  - Серия SimpleLink на TI

Gateway to hell



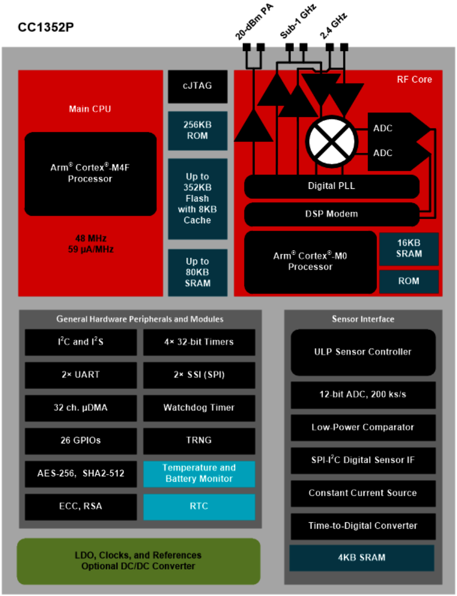
- Микроконтролер
  - Серия SimpleLink на TI
- SoC - Едноплаткови компютри

Gateway to hell

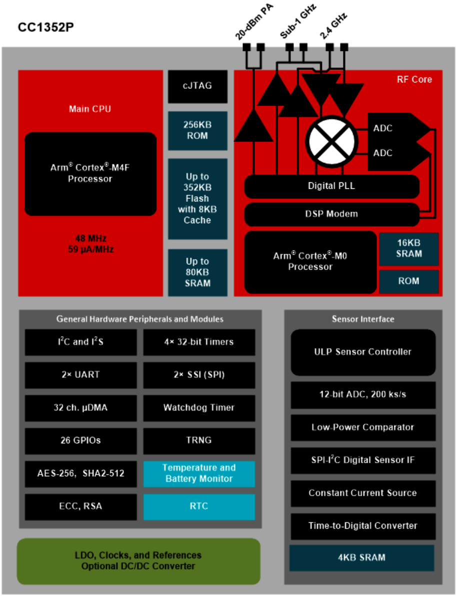


- Микроконтролер
  - Серия SimpleLink на TI
- SoC - Едноплаткови компютри
  - Raspberry Pi

Gateway to hell

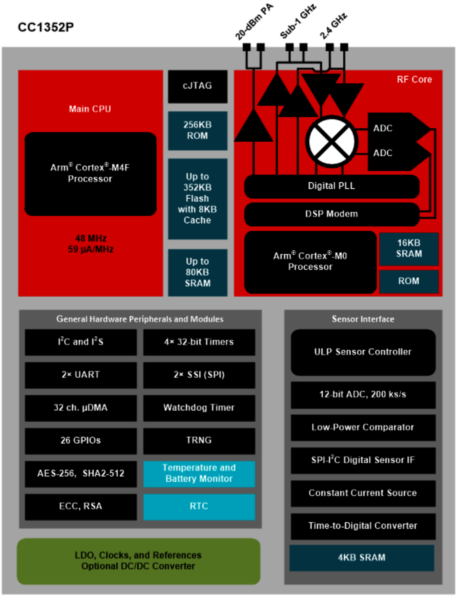


- Микроконтролер
  - Серия SimpleLink на TI
- SoC - Едноплаткови компютри
  - Raspberry Pi
  - Beaglebone



- Микроконтролер
  - Серия SimpleLink на TI
- SoC - Едноплаткови компютри
  - Raspberry Pi
  - Beaglebone
  - BananaPi

Gateway to hell

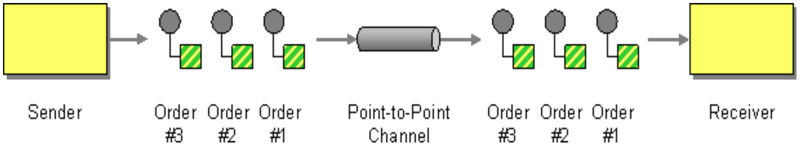


- Микроконтролер
  - Серия SimpleLink на TI
- SoC - Едноплаткови компютри
  - Raspberry Pi
  - Beaglebone
  - BananaPi
  - Odroid XU4

Internet of Things (IoT)

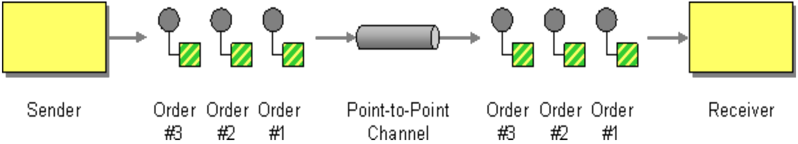
IoT - протоколи Сензор - Gateway

- Според връзката между клиент и сървър
  - Point to Point протоколи - CoAP - http://coap.technology/.



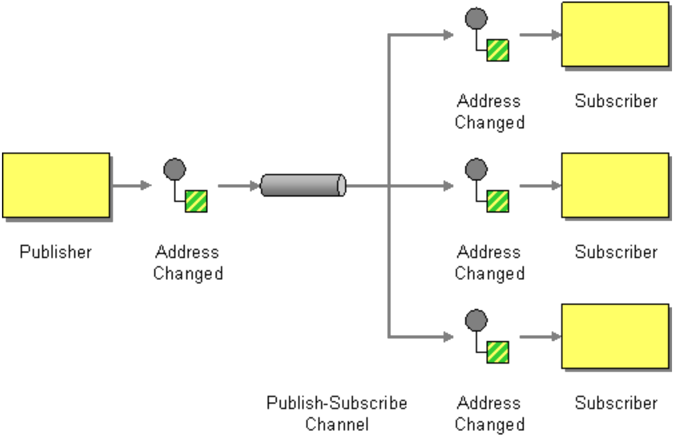
Gateway/Backend Протоколи

- Според връзката между клиент и сървър
  - Point to Point протоколи - CoAP - http://coap.technology/.



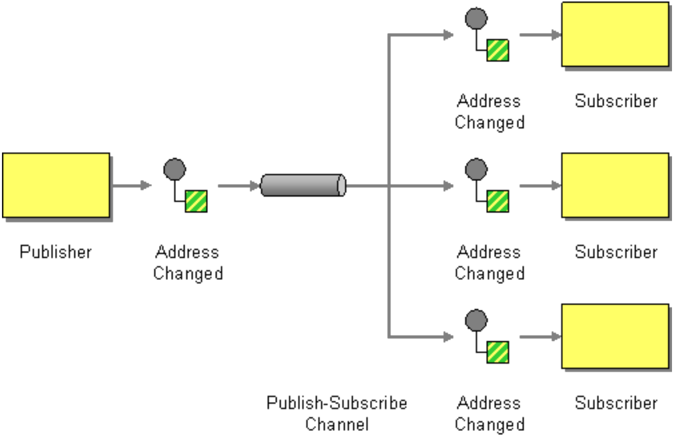
  - Publish/Subsribe протоколи

Publish/Subscibe протоколи



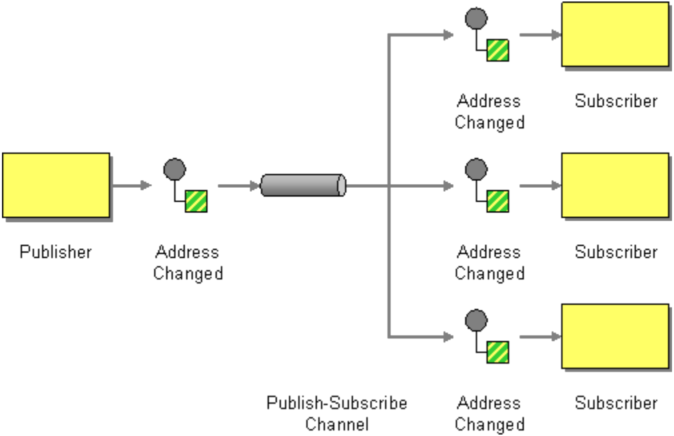
- MQTT - https://www.hivemq.com/mqtt-essentials

Publish/Subscibe протоколи



- MQTT - https://www.hivemq.com/mqtt-essentials

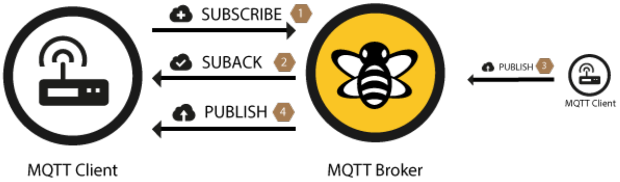
Publish/Subscibe протоколи



- MQTT - https://www.hivemq.com/mqtt-essentials
- AMQP - https://www.rabbitmq.com/tutorials

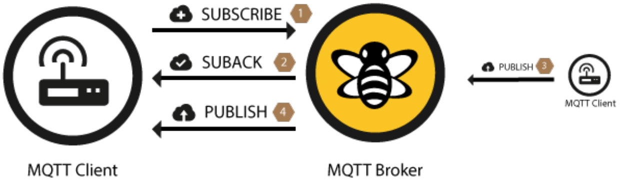
MQTT

- Всеки клиент се записва за съобщения от брокера, които го интересуват и получава само тях

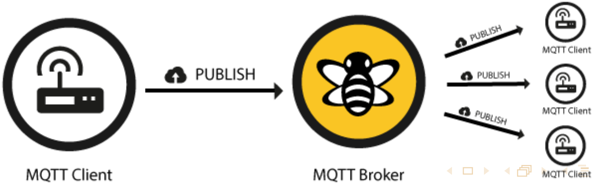


MQTT

- Всеки клиент се записва за съобщения от брокера, които го интересуват и получава само тях



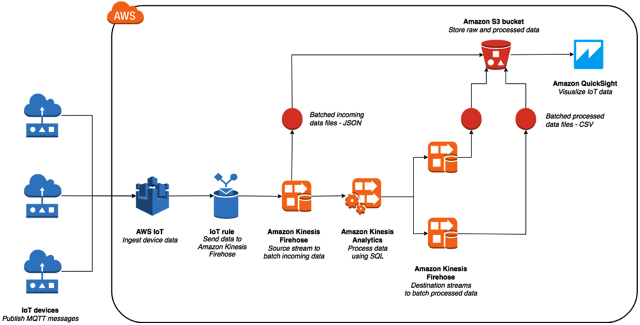
- При зпращане на данни от клиент към брокер, той ги препраща към всички записани за него



Internet of Things (IoT)

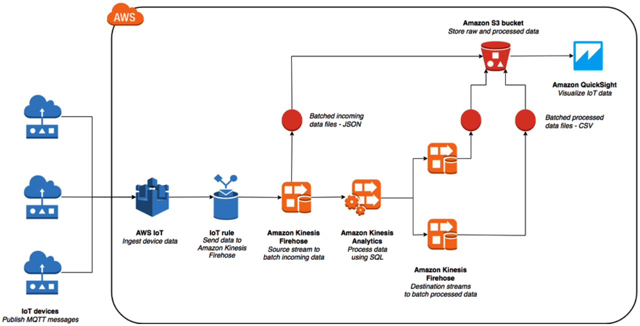
IoT - Backend services

Backend services - PaaS



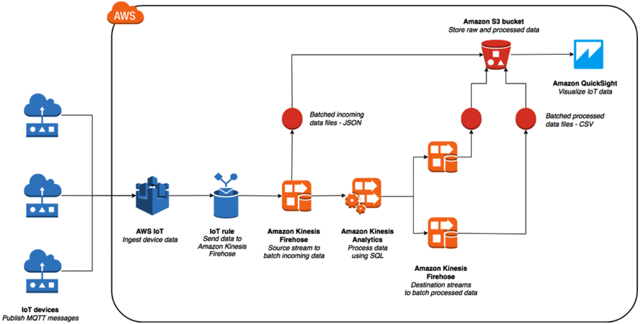
- Съществуват множество платформи за цялостно събиране и обработка на данните
  - Amazon IoT

Backend services - PaaS

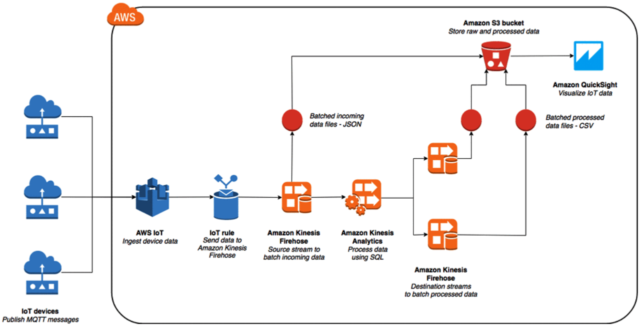


- Съществуват множество платформи за цялостно събиране и обработка на данните
  - Amazon IoT
  - Thingspeak

Backend services - PaaS



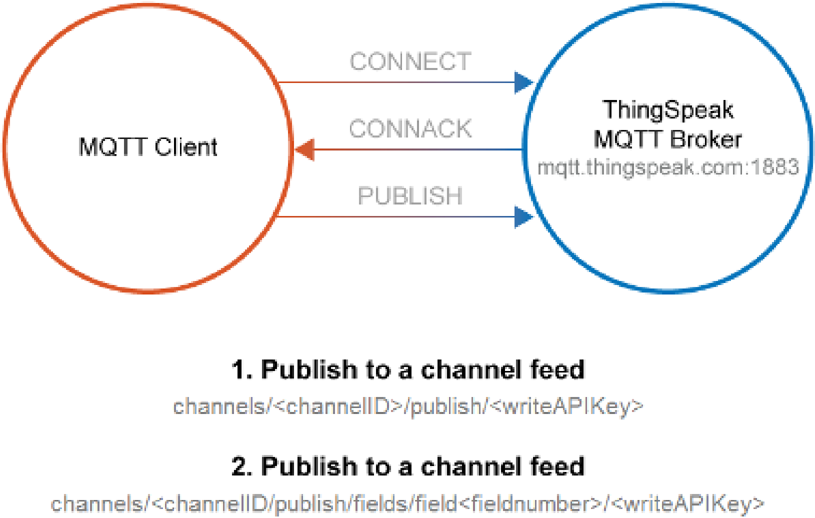
- Съществуват множество платформи за цялостно събиране и обработка на данните
  - Amazon IoT
  - Thingspeak
  - Microsoft Azure



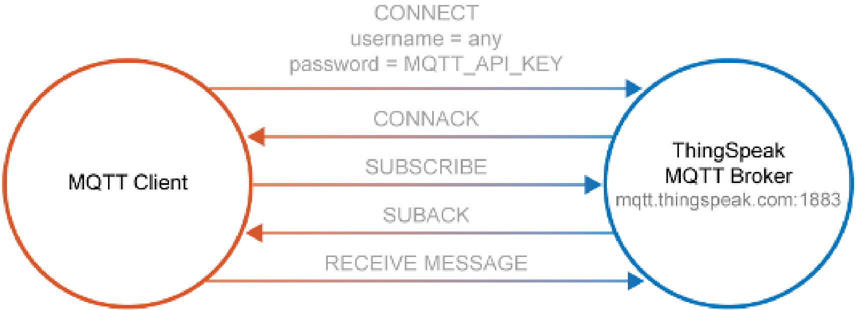
- Съществуват множество платформи за цялостно събиране и обработка на данните
  - Amazon IoT
  - Thingspeak
  - Microsoft Azure
  - IBM Bluemix

Thingspeak- идея за публикуване

https://www.mathworks.com/help/thingspeak/mqtt-basics.html



Thingspeak- идея за теглене на данни



1. Subscribe to a channel feed

channels/<channelID>/subscribe/<format>/<api_key>

2. Subscribe to a private channel feed

channels/<channelID>/subscribe/fields/field<fieldNumber>/<apiKey>

3. Subscribe to all fields of a channel

channels/<channelID>/subscribe/fields/+/<apiKey>

Thingspeak- MQTT интеграция

Пример

Да се демонстира качване на данни от Matlab към Thingspeak (Ако има интернет)

На когото му се чете вкъщи . . .

- R.P. Canale and D. Steven C. Chapra.
  Numerical Methods for Engineers.
  McGraw-Hill Education, 2014.
- E.B. Magrab, S. Azarm, B. Balachandran, J. Duncan, K. Herold, and G. Walsh.
  An Engineers Guide to MATLAB.
  Pearson Education, 2011.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery.
  Numerical Recipes 3rd Edition: The Art of Scientific Computing.
  Cambridge University Press, 2007.
- S.H. Strogatz.
  Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering.
  Studies in Nonlinearity. Avalon Publishing, 2014.