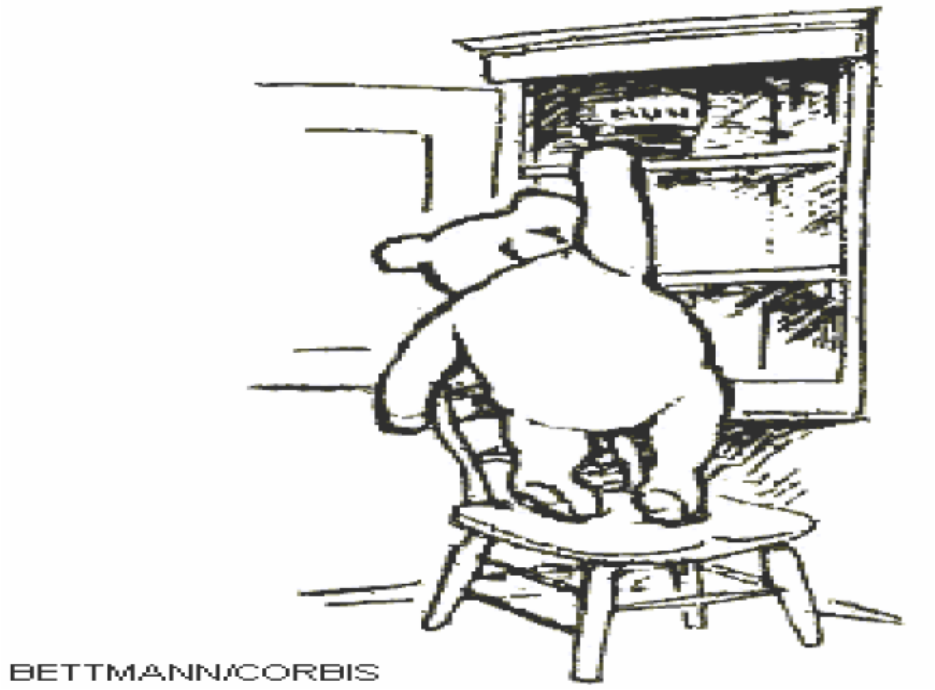
Embedded Real time systems – an Introduction

- What's a "Real time system"? The concept of ***control***.
- Embedded real time control – some examples
- Some terminology
- "Scheduling" – determination of actions
- "Execution time analysis" – a prerequisite for scheduling

In Control ?

In Control ?



In Control ?

Swedish fighter JAS 39 GRIPEN

In Control ?

Swedish fighter JAS 39 GRIPEN

In Control ?

Swedish fighter JAS 39 GRIPEN

In Control ?

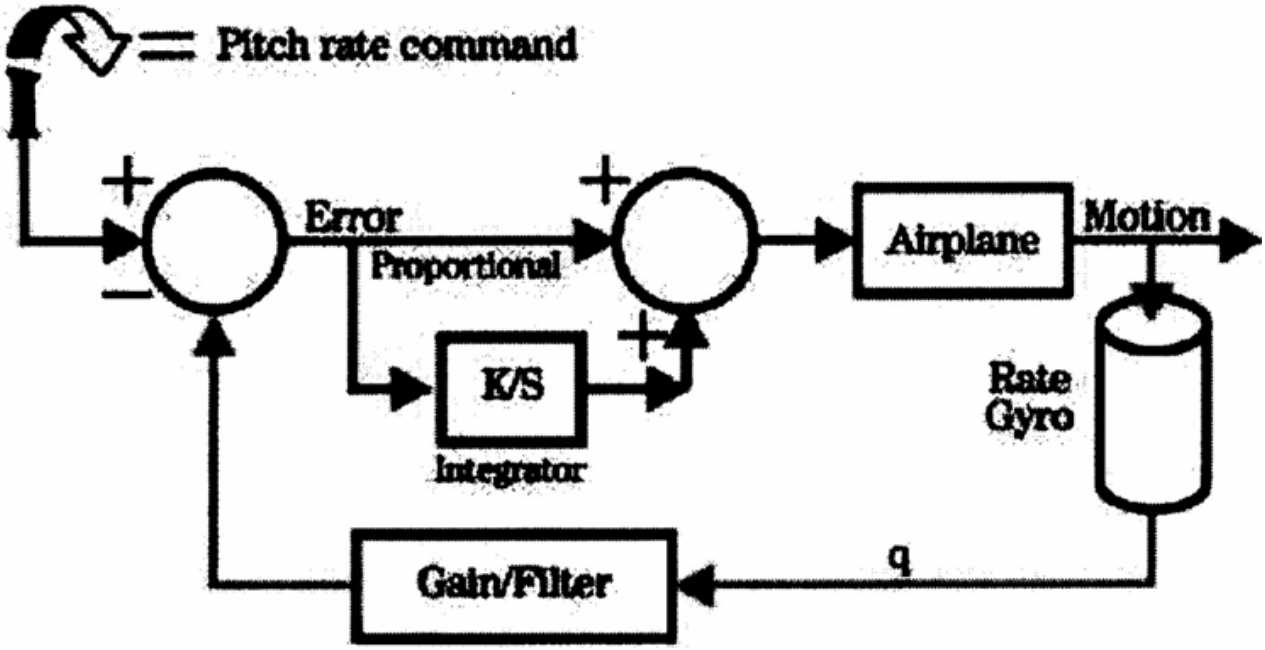
but even big planes crash ...

In Control ?

and helicopters ...

In Control ?

Control requires feedback



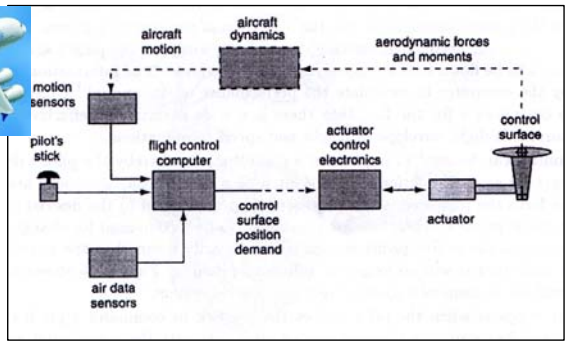
In Control ?

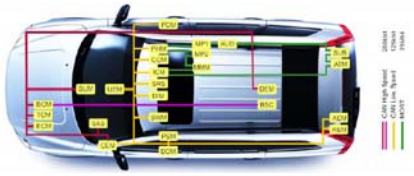
A control system is a feedback system

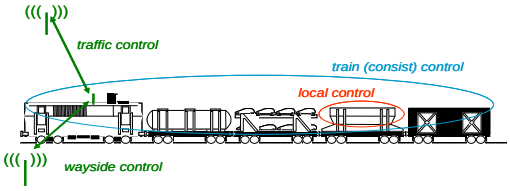
Time properties (temporal aspects) is fundamental for a well designed control system.

A computer designed for control has to handle REAL-TIME as desired by the controlled object.

Real time control applications

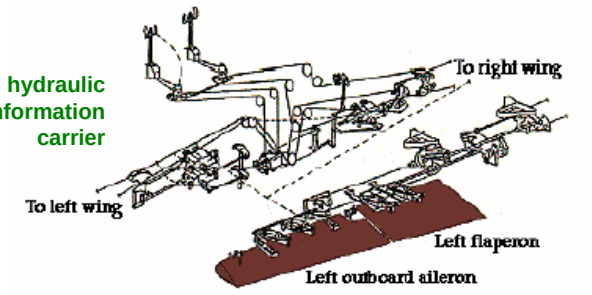




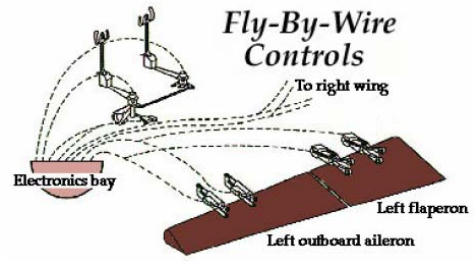


steer ”by-wire”

hydraulic information carrier



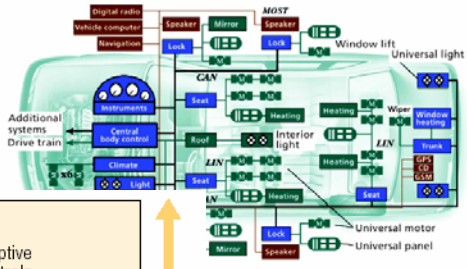
electronic information carrier

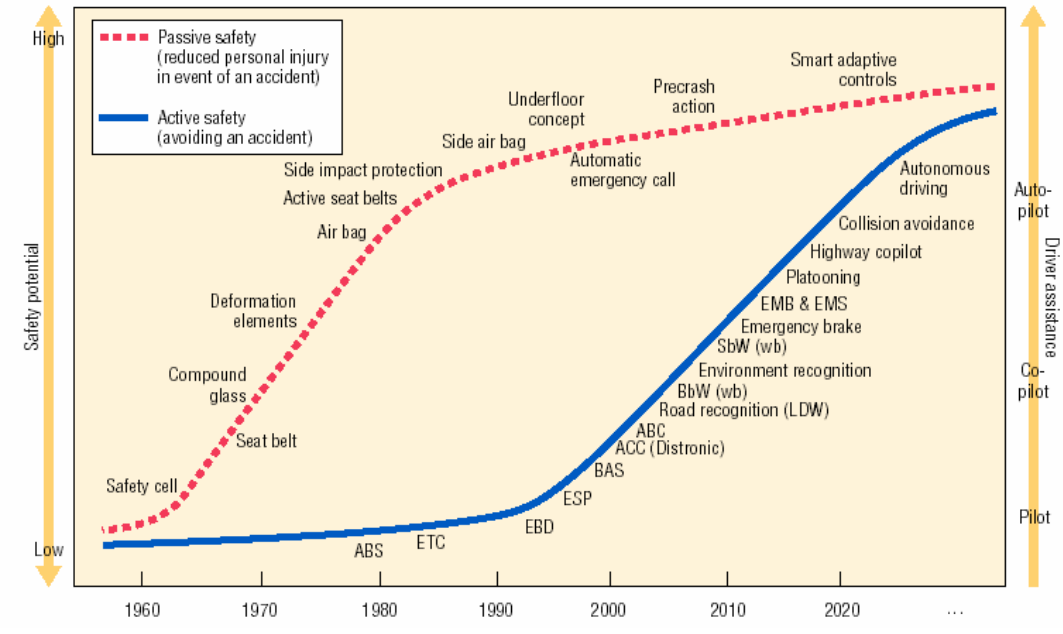


The “F-8 Digital Fly-By-Wire (DFBW)” validated fundamental concepts still used today.
The first “by-wire” flight was in the 25’th of May in 1972. The project lasted for 13 years.

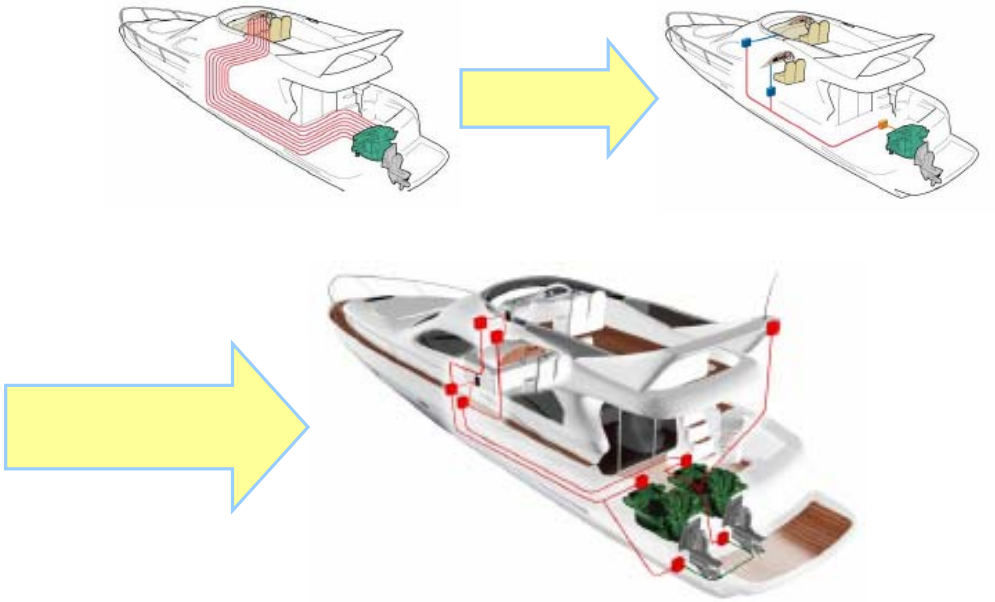
Computers in space…

..development of Automotive electronics





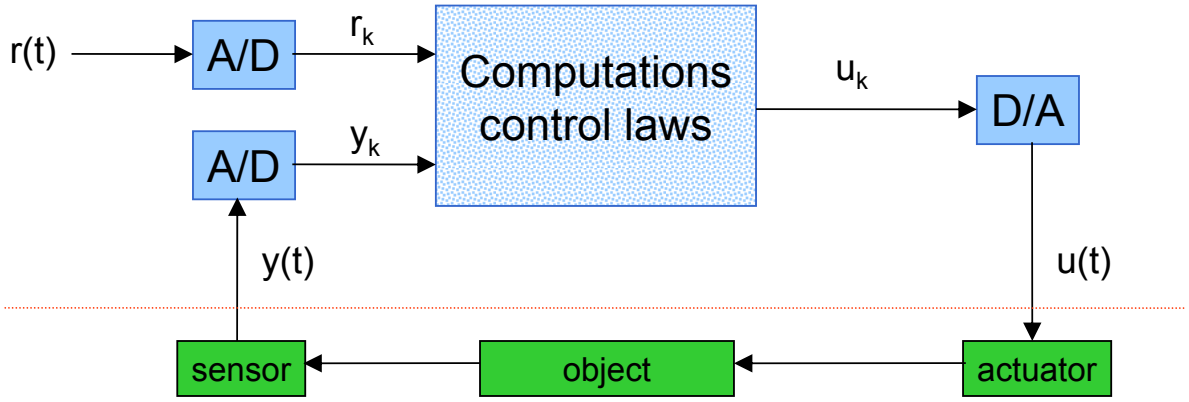
..and within marine applications.



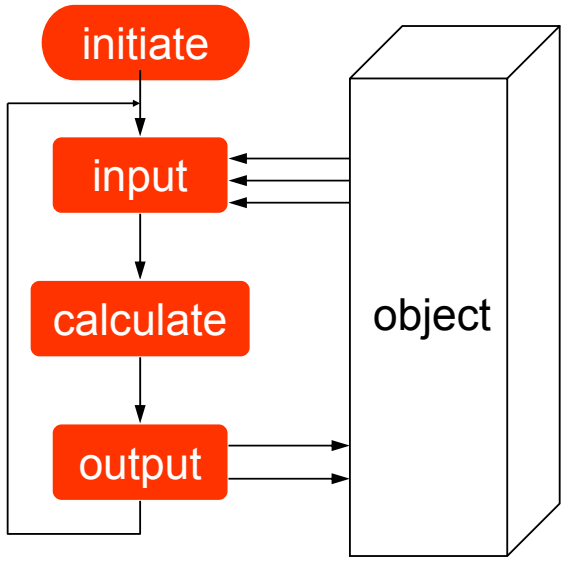
control characteristics in a car

- simple reactive (ON/OFF) …
  climate, windows …
- reactive, autonomous …
  engine- and gear- control, anti-brake-lock, anti-spin, etc.
- safety critical
  airbag ...
- others..
  navigation, audio-video ...

A control system



Time triggered implementations



- Pros

  Easy to understand, easy to implement.

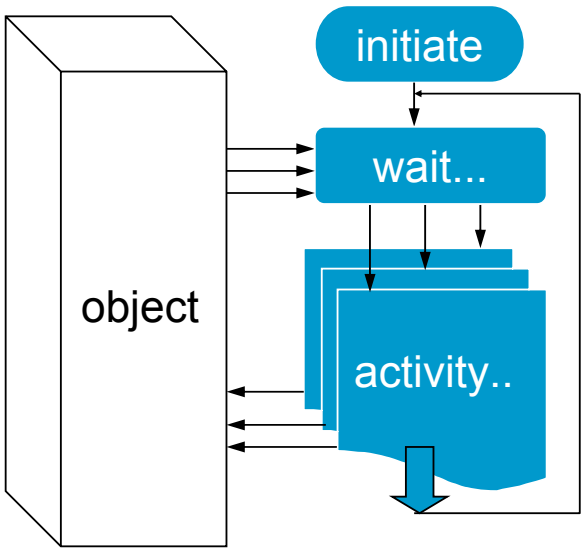
- Cons

  Time synchronization is tied to underlying hardware.

  Difficult to anticipate and handle unordered events.

  Software complexity depends on object .

Event triggered implementations



- Pros

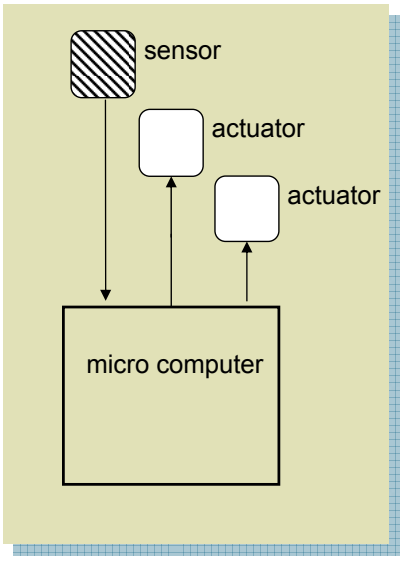
  Easy to understand.

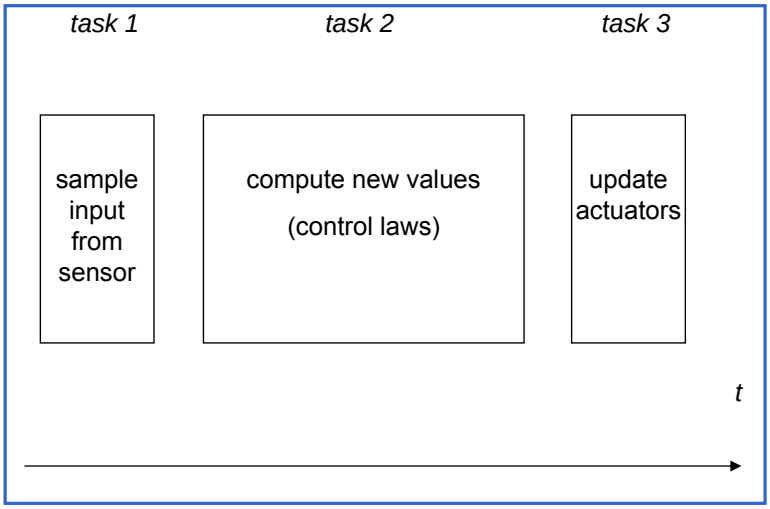
  Short response times (good feedback performance)

- Cons

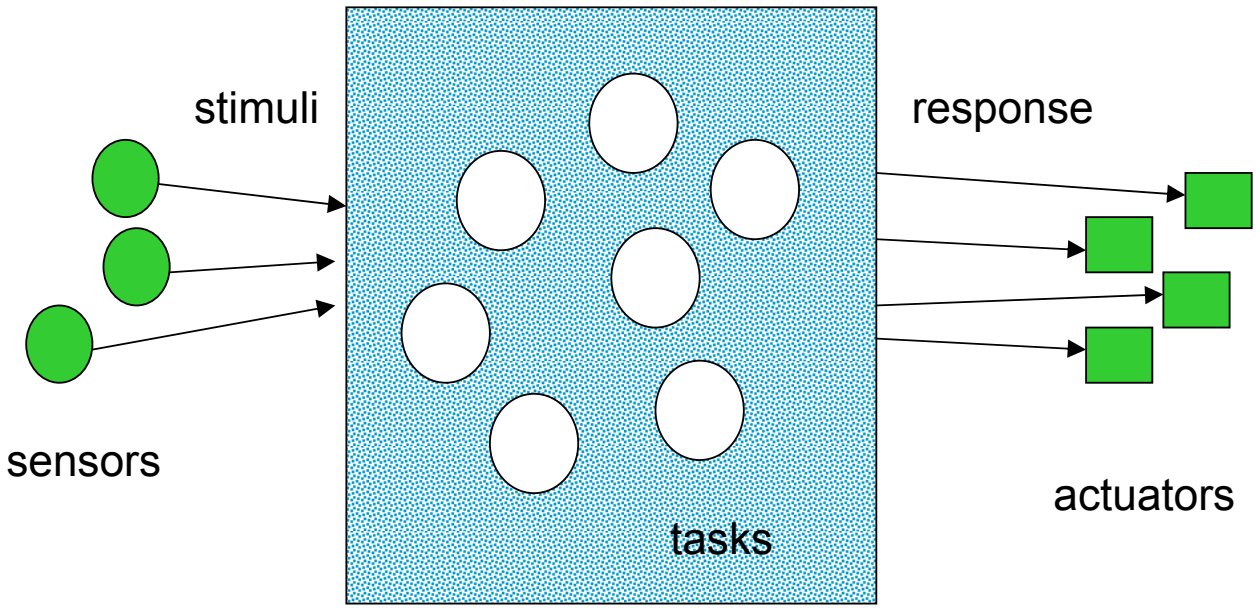
  Exteremely complex software when object becomes complex.

A software *task*

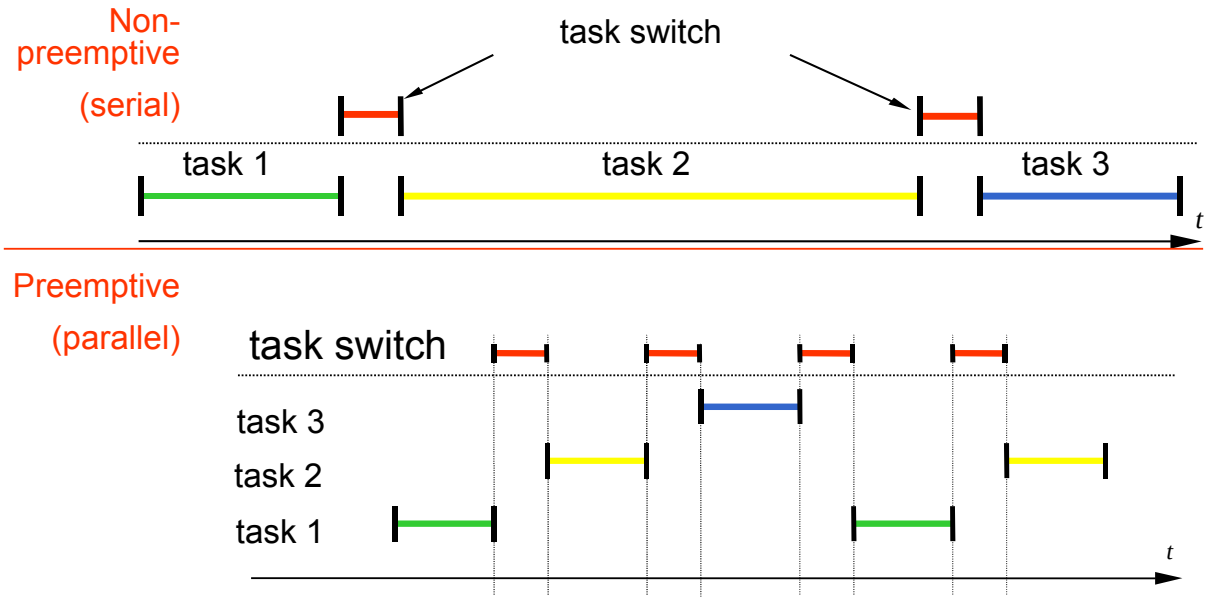




Computational model



Scheduling strategies



Scheduling methods

Pre run-time scheduling

- static scheduling
- fixed priority scheduling
- ….

Run-time scheduling

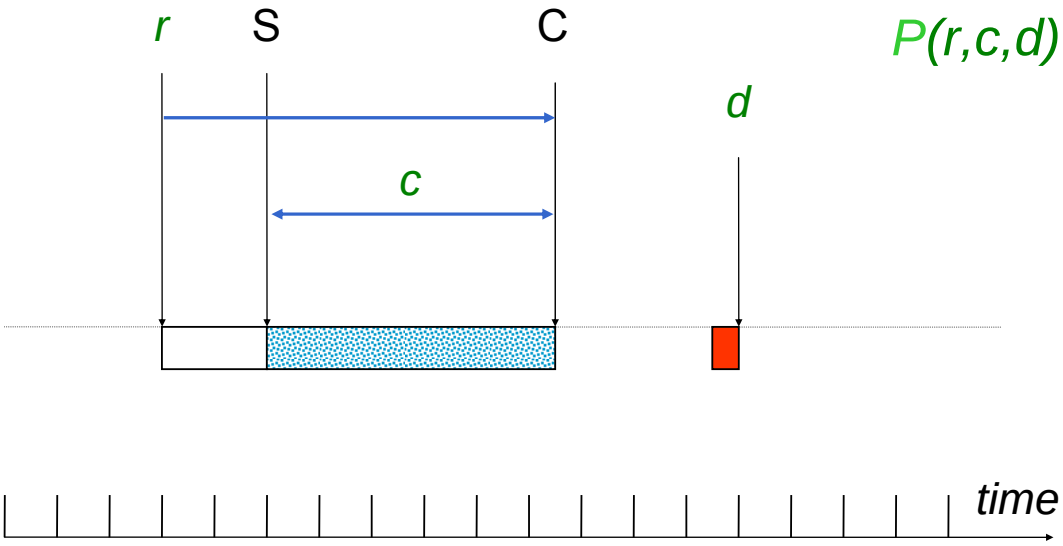
- EDF (earliest deadline first)
- FCFS (first comes first served)
- ….

Task characteristics

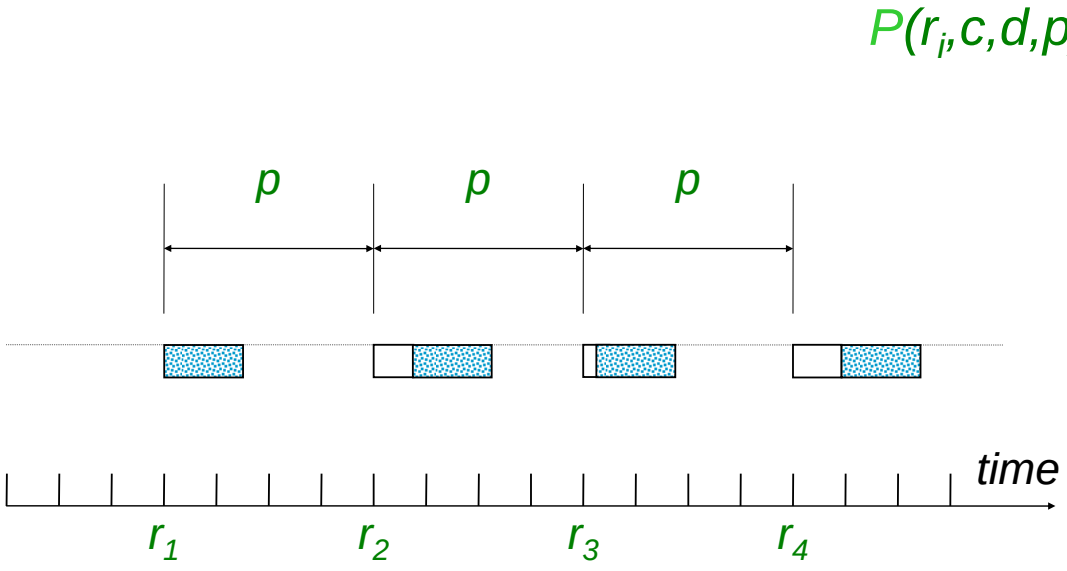
```
while(1){
   A = read_sensor(…);
   nA = compute(A);
   write_sensor(nA);
   delay( 10 );
}
```

$P(r_i,c,d,p_{min})$

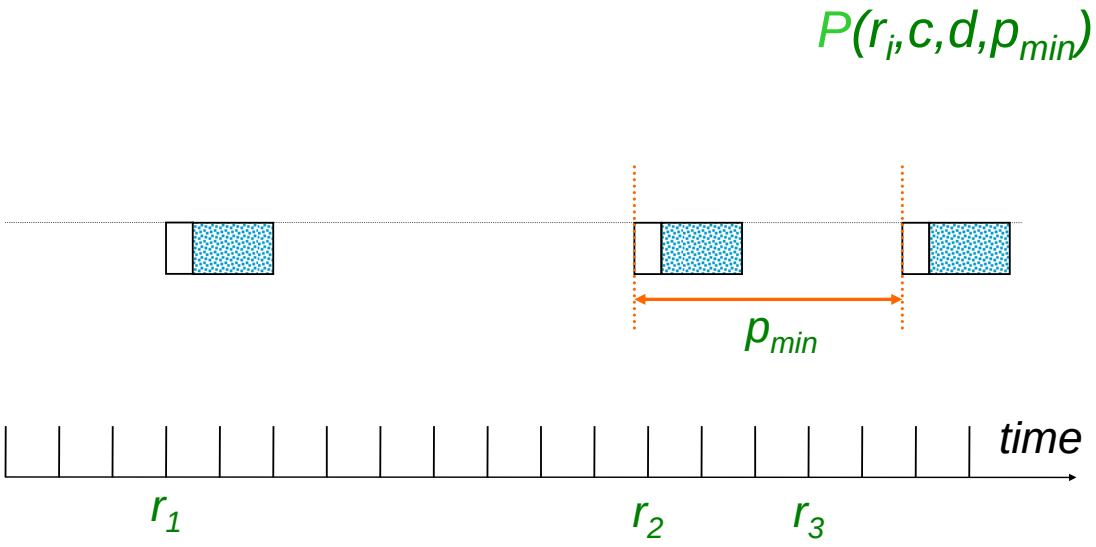
Task characteristics



Periodic tasks



Aperiodic tasks with multiple start times



Task Execution time analysis

Measurements, program under test is executed a number of times. A *Worst Case Execution Time is estimated*.

Code analysis, determine the most time consuming path through the program and determine execution time for this path.

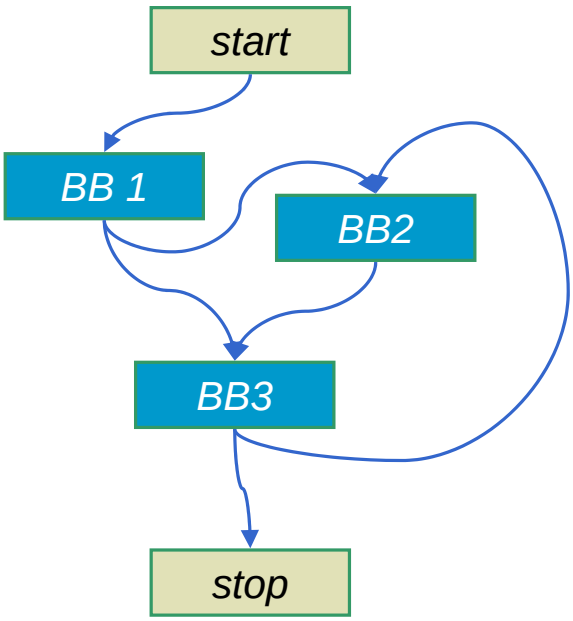
Execution time measure

```
// Example

t_start = C(t);
P();
t_finish = C(t);

t_exec = t_finish – t_start;
```


Execution time analysis, basic blocks



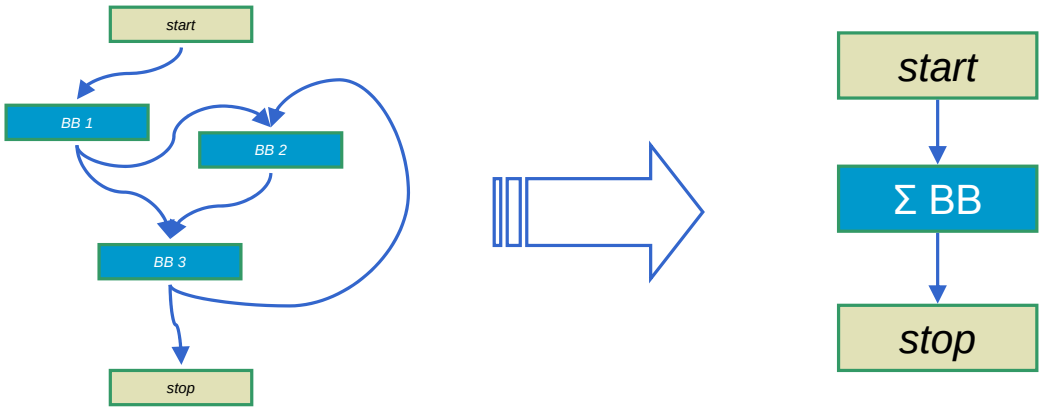

```
label:
        ..instr 1
        ..instr 2
        ..instr 3

        ...
        ..instr n

        bra   ...
```

Execution time analysis, Control Flow Graphs

A Control Flow Graph represents all possible execution paths in a program. Contributions from all basic blocks are summed.



Conclusions

we have got a brief introduction to

- The concept of ***control***.
- Embedded real time control
- Real Time Systems terminology
- Task Scheduling
- Execution time analysis

which finishes today's lecture ...