

Úvod do mobilní robotiky – NAIL028

Architektury řídicích systémů, Filtrování dat

Jiří Iša

Matematicko-fyzikální fakulta Univerzity Karlovy v Praze

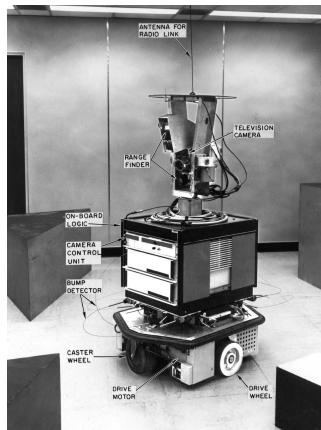
30. října 2008

Obsah

- 1 Architektury řídicích systémů
- 2 Filtrování dat – úvod

Sense-plan-act architektura (SPA)

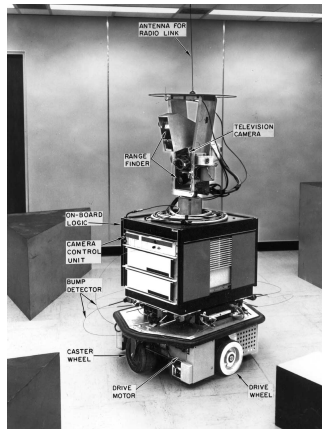
- 1 Pozorovat prostředí.



Shakey the Robot, 1966 – 1972

Sense-plan-act architektura (SPA)

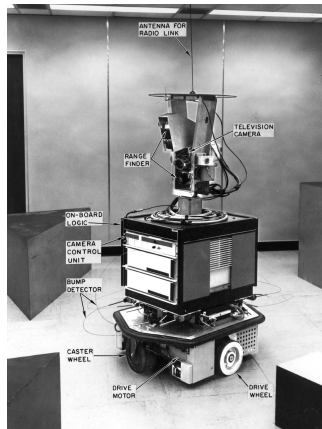
- 1 Pozorovat prostředí.
- 2 Naplánovat akci/akce.



Shakey the Robot, 1966 – 1972

Sense-plan-act architektura (SPA)

- 1 Pozorovat prostředí.
- 2 Naplánovat akci/akce.
- 3 Provést naplánovanou činnost.



Shakey the Robot, 1966 – 1972

Architektury řídicích systémů
Filtrování dat – úvod

Sense-plan-act architektura (SPA)
Reaktivní systémy
Subsumption architektura
Stavové automaty
Petriho sítě
Třívrstvá architektura
Plánování se zásobníkem
Jak to vypadá

Svět kostek – ze života

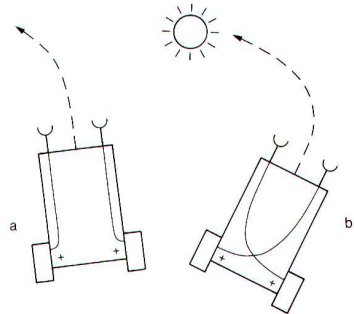
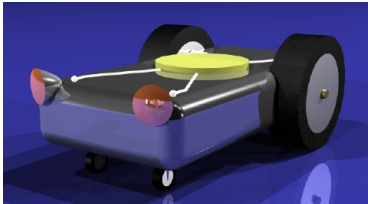


Reaktivní systémy

Rodney Brooks:

Svět je sám sobě nejlepším modelem.

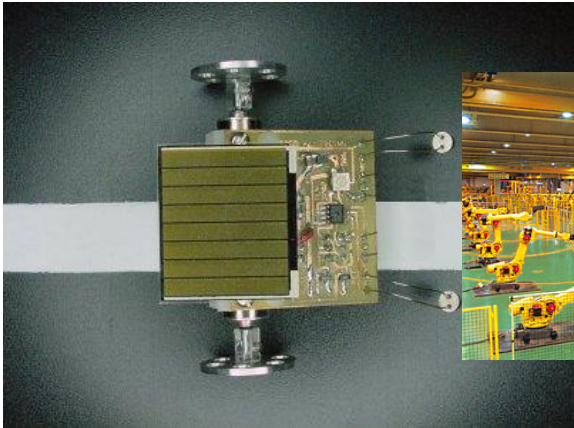
Braitenbergova autíčka



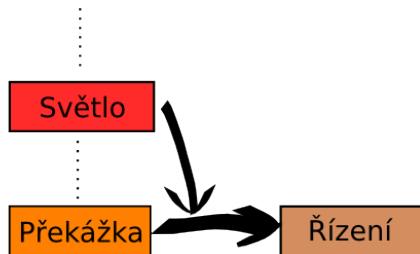
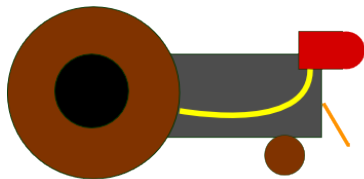
Architektury řídicích systémů
Filtrování dat – úvod

Sense-plan-act architektura (SPA)
Reaktivní systémy
Subsumption architektura
Stavové automaty
Petriho sítě
Třívrstvá architektura
Plánování se zásobníkem
Jak to vypadá

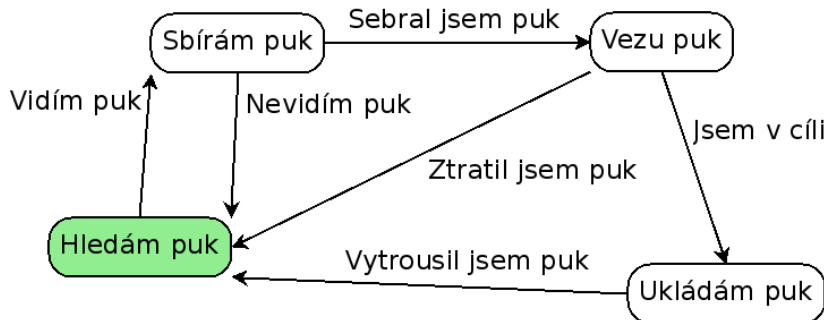
Reaktivní systémy – ze života



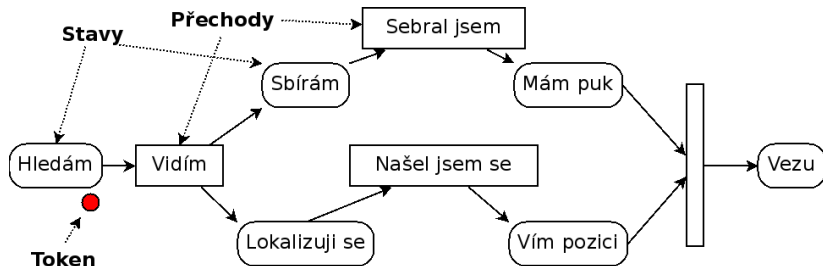
Subsumption architektura



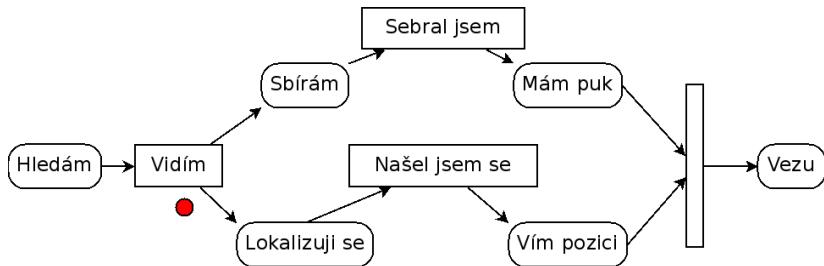
Stavové automaty



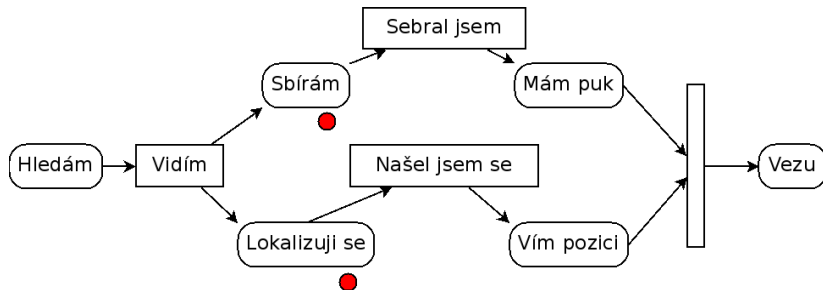
Petriho síť



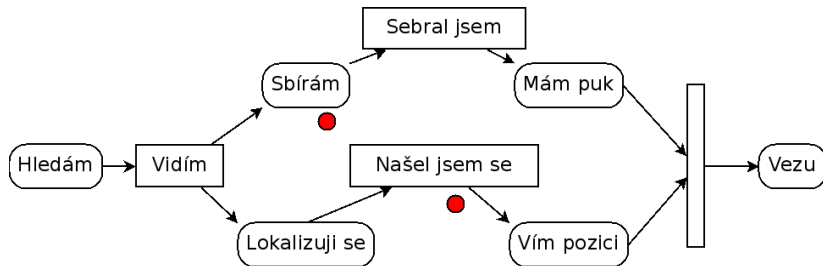
Petriho síť



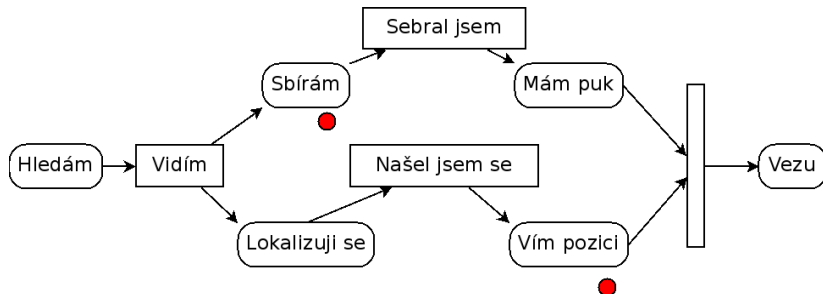
Petriho síť



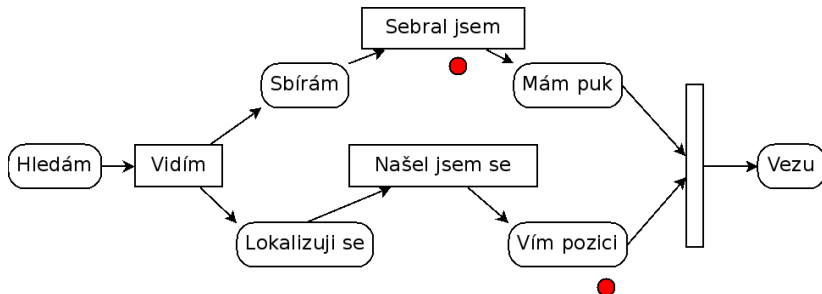
Petriho síť



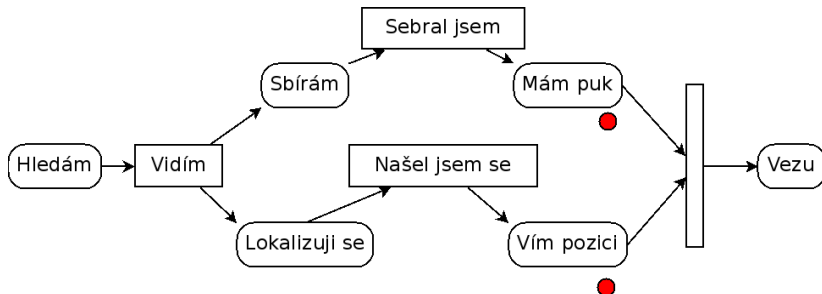
Petriho síť



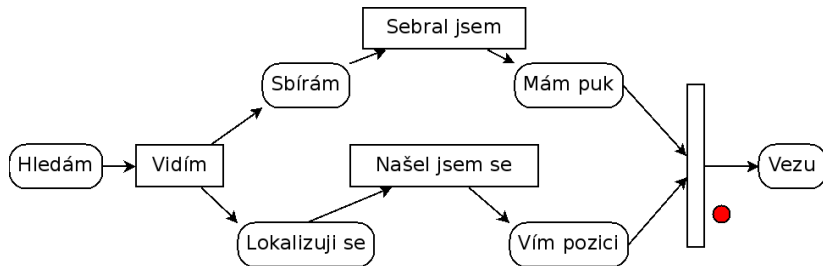
Petriho síť



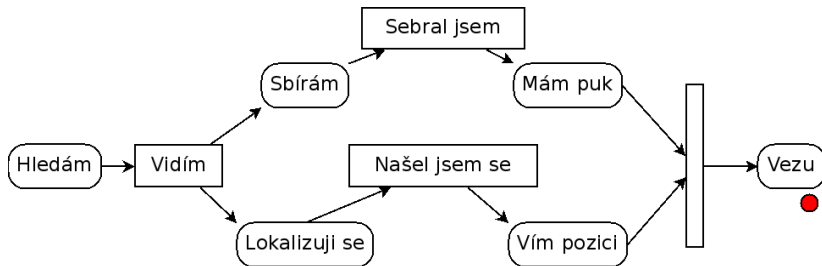
Petriho síť



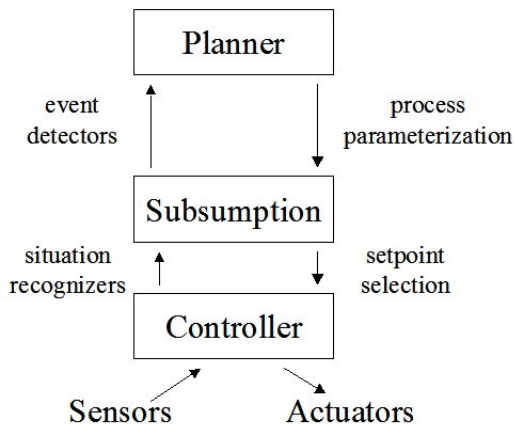
Petriho síť



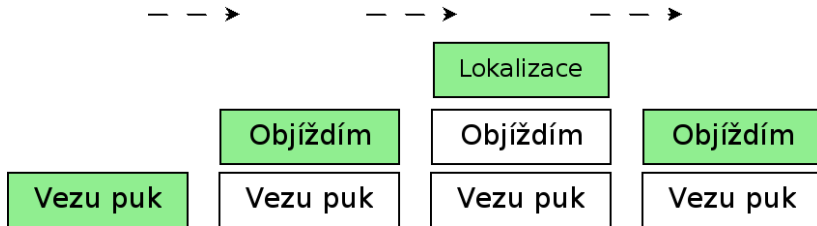
Petriho síť



Třívrstvá architektura



Plánování se zásobníkem



Jak to vypadá

```
while(true)
{
    sync(state, command);
    switch(state)
    {
        case SEARCHING:
            if(found())
            {
                state = PICKING_UP;
                command.stop();
            }
            break;
        case PICKING_UP:
            ...
    }
}
```

```
while(true)
{
    search();
    pick_up();
    transfer();
}
void search(void)
{
    if(opponent_ahead())
    {
        avoid_opponent();
    }
}
...
}
```

Tip dne: Synchronize()

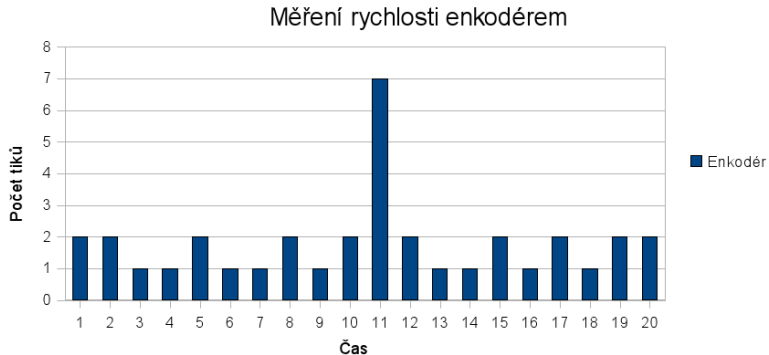
Synchronizace s pevnou časovou periodou

- `synchronize(COMMAND *, STATUS *, TIME);`
- `synchronize(COMMAND *, STATUS *);`

```
struct HWWrite
{
    uint8_t W_executeAt;
    uint8_t W_watchDog;
    uint8_t W_servo[4];
    uint8_t W_digitalOutputs;
};
```

```
struct HWRead
{
    uint8_t R_timer;
    uint8_t R_digitalInputs;
    uint8_t R_analog[6];
};
```


Tiky enkodéru

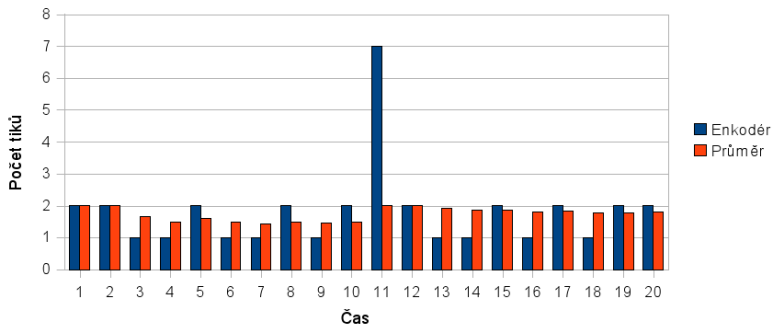


Průměr

$$\bar{x}_n = \frac{1}{n} \sum_{i=1}^n x_i$$

Sonar – průměr

Měření rychlosti enkodérem



Inkrementální počítání průměru

- Posloupnost částečných součtů:

$$s_0 = 0$$

$$s_{n+1} = s_n + x_{n+1}$$

$$\bar{x}_n = \frac{s_n}{n}$$

Inkrementální počítání průměru

- Posloupnost částečných součtů:

$$s_0 = 0$$

$$s_{n+1} = s_n + x_{n+1}$$

$$\bar{x}_n = \frac{s_n}{n}$$

- Inkrementální počítání:

$$\bar{x}_n = (\bar{x}_{n-1} * (n - 1) + x_n) / n$$

$$= \bar{x}_{n-1} + \frac{1}{n}(x_n - \bar{x}_{n-1})$$

$$= \left(1 - \frac{1}{n}\right) * \bar{x}_{n-1} + \frac{1}{n} * x_n$$

Plovoucí průměr

- Počítání s okénkem:

$$\bar{x}_n = \frac{1}{k} \sum_{i=n-k+1}^n x_i$$

Plovoucí průměr

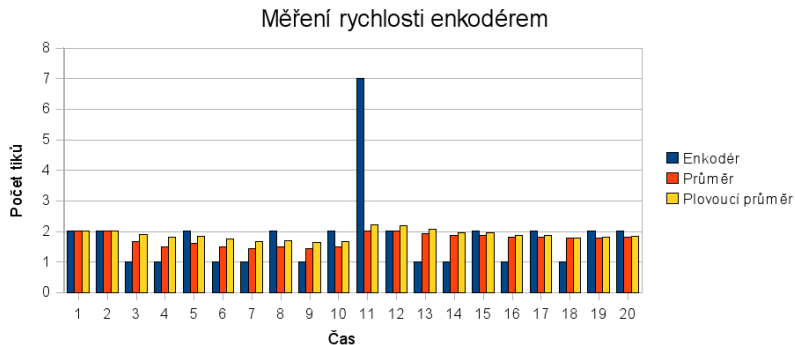
- Počítání s okénkem:

$$\bar{x}_n = \frac{1}{k} \sum_{i=n-k+1}^n x_i$$

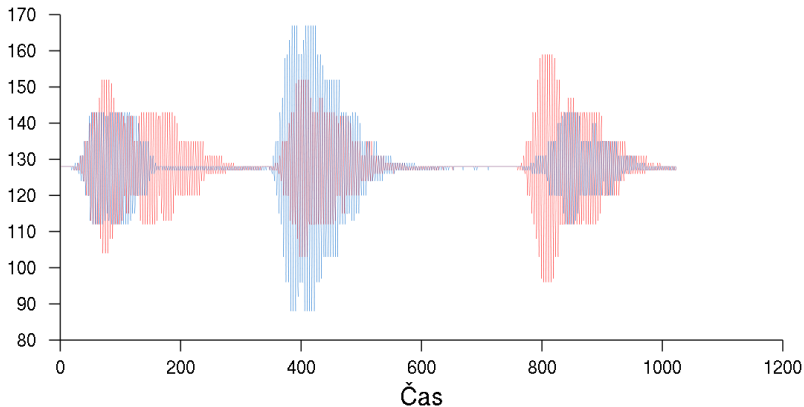
- Inkrementální počítání:

$$\begin{aligned}\bar{x}_n &= \left(1 - \frac{1}{k}\right) * \bar{x}_{n-1} + \frac{1}{k} * x_n \\ &= (1 - \alpha) * \bar{x}_{n-1} + \alpha * x_n\end{aligned}$$

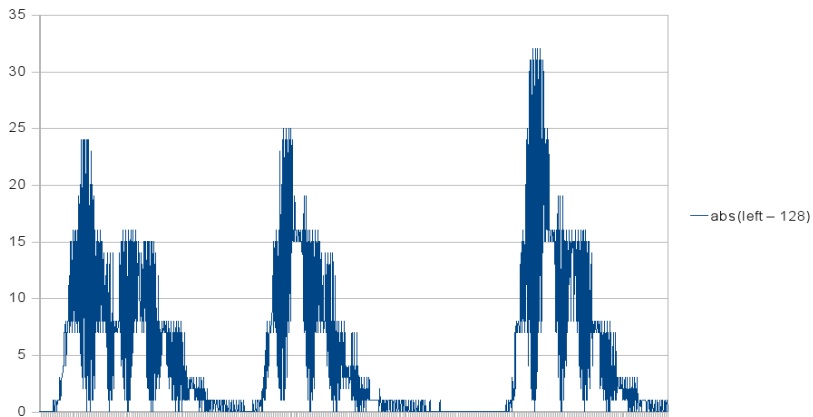
Sonar – plovoucí průměr



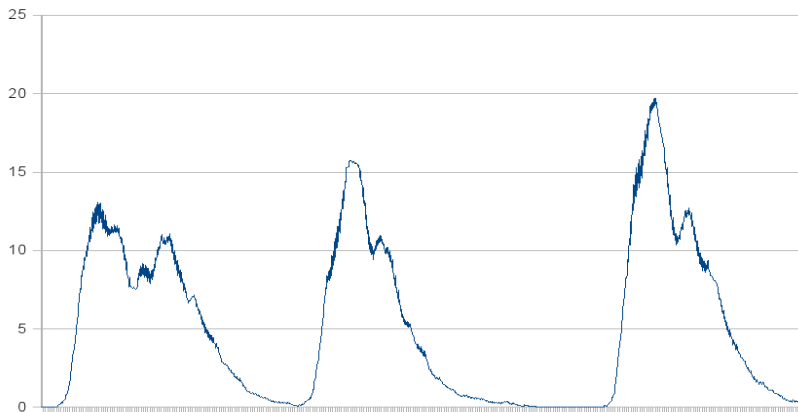
Dvoukanálový sonar



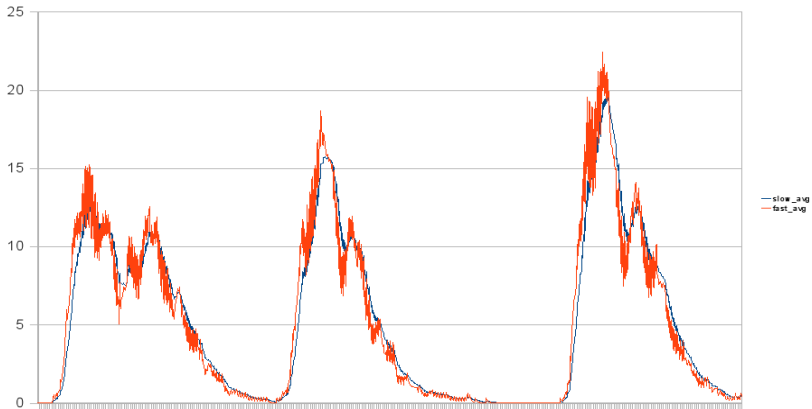
Dvoukanálový sonar – $\text{abs}(\text{value} - \text{offset})$



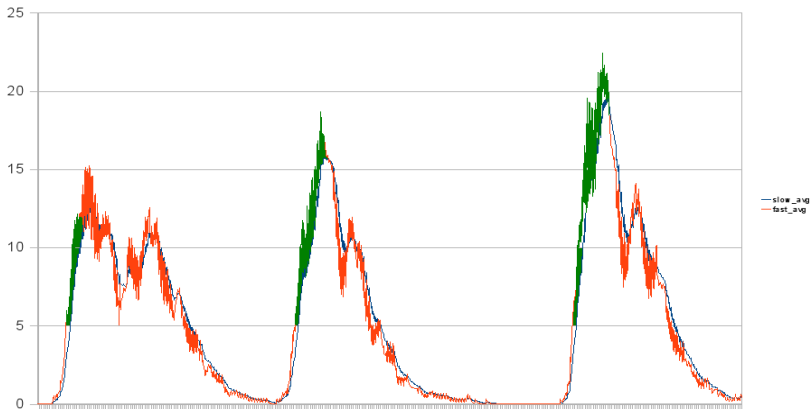
Dvoukanálový sonar – plovoucí průměr



Dvoukanálový sonar – dva plovoucí průměry



Dvoukanálový sonar – dva plovoucí průměry



Dvoukanálový sonar – detekované body

