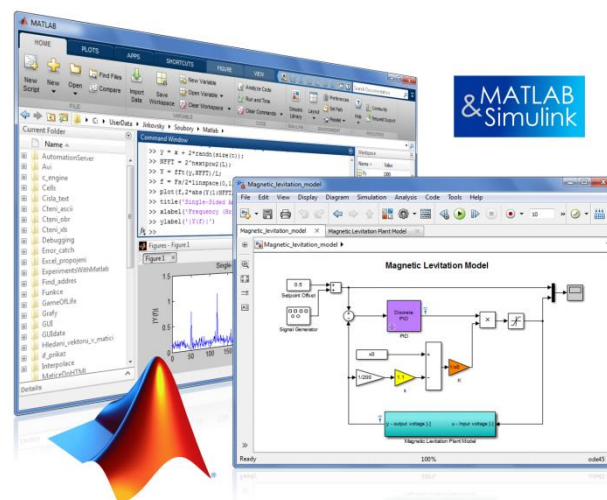


10.9.2015 Brno

Analýza a zpracování signálu v MATLABu Implementace algoritmů na cílovou platformu



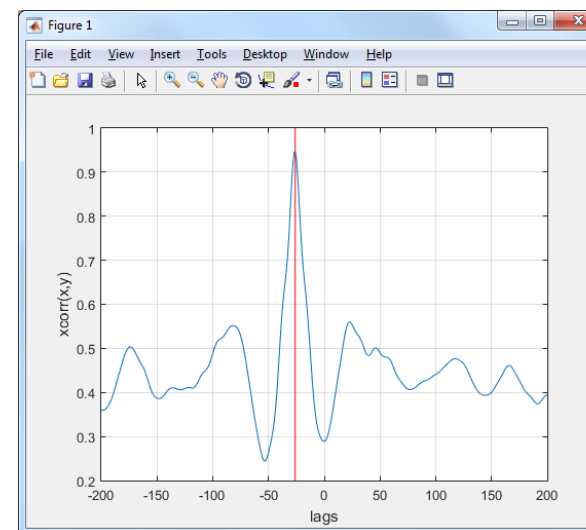
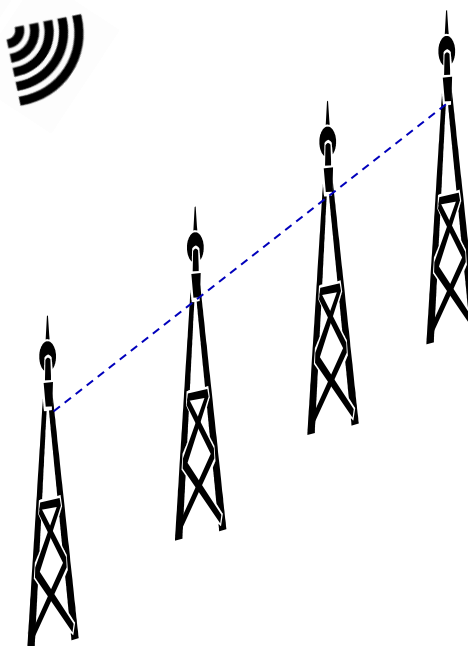
Jaroslav Jirkovský
jirkovsky@humusoft.cz

www.humusoft.cz
info@humusoft.cz

www.mathworks.com

Návrh algoritmu

- Úkol
 - navrhnout algoritmus pro odhad směru příchodu signálu
- Řešení
 - využijeme připravené funkce nástroje Signal Processing Toolboxu a navrhne algoritmus
 - algoritmus doplníme o vizualizaci výsledku



Začlenění algoritmů do simulace systémů

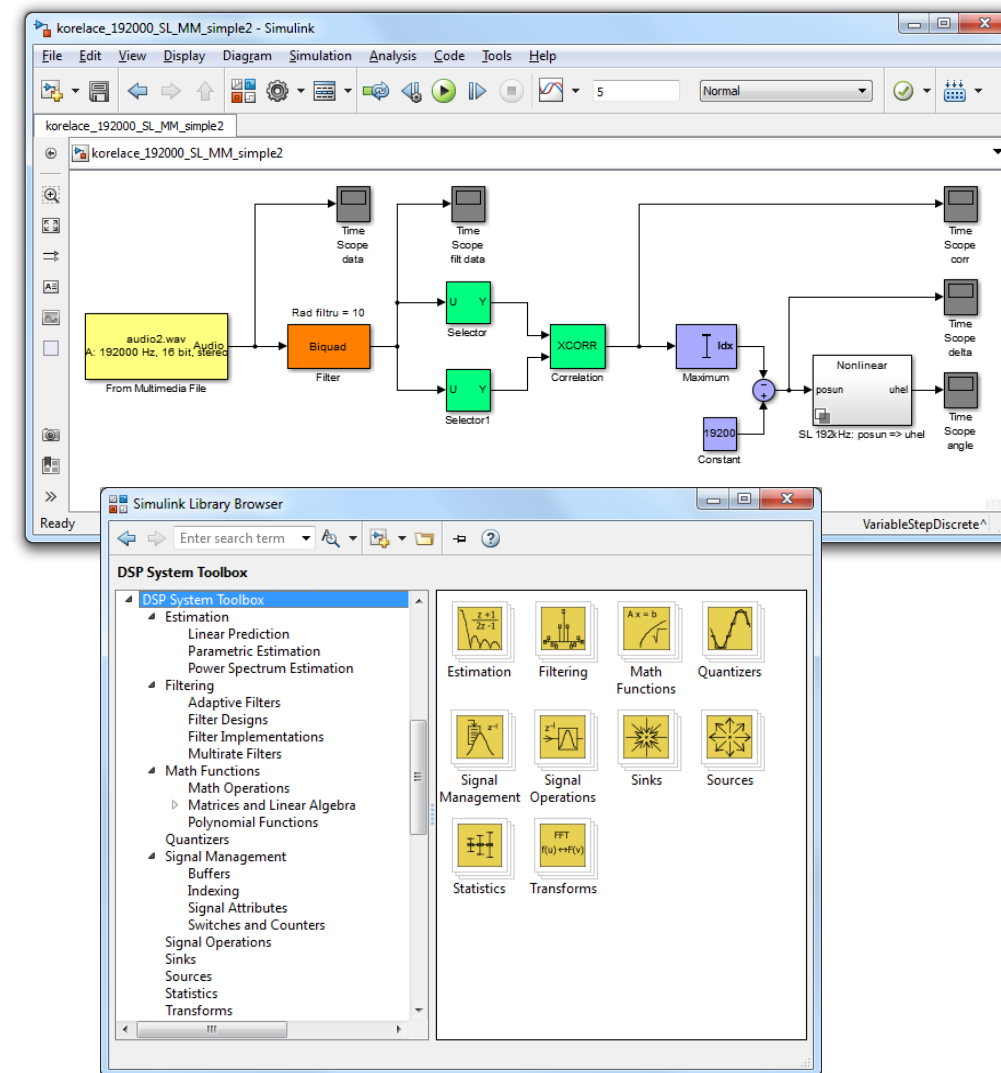
- Úkol

- vytvořit model algoritmu v Simulinku

- Řešení

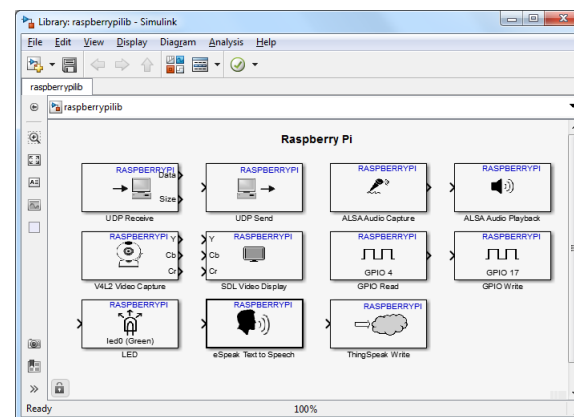
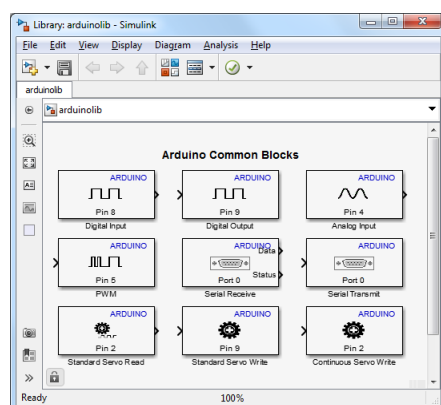
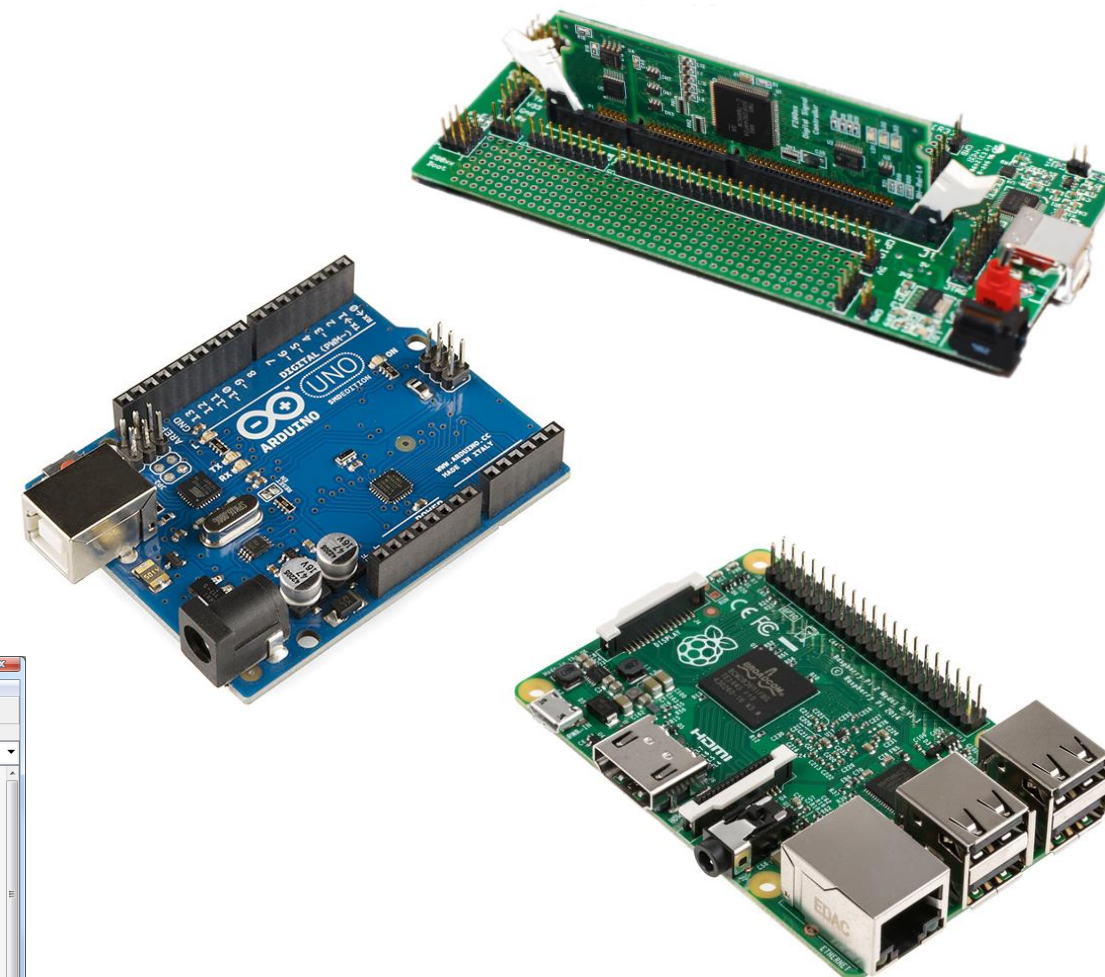
- využít knihovnu bloků DSP System Toolbox

- použít možnost automatické generování simulinkového bloku z navrženého filtru



Implementace algoritmů na cílovou platformu

- Úkol
 - připravit algoritmus pro implementaci na cílovou platformu
- Řešení
 - přidat bloky pro periferie Raspberry Pi
 - exportovat model



Generování kódu pro cílové platformy

Modely systémů v Simulinku

Diagrams illustrating system models in Simulink, including:

- Anti-Windup PID Control with Feedforward Control
- Adaptive Filtering
- Stateflow diagrams
- 3D surface plot

Algoritmy v MATLABu

Algorithms implemented in MATLAB, showing code for a Kalman filter and associated plots:

```

function y = kalmanfilter(s)
%codegen
% Initialize state transition matrix
A = [1 0 0 0 0 0; ...
     0 1 0 0 0 0; ...
     0 0 1 0 0 0; ...
     0 0 0 1 0 0; ...
     0 0 0 0 1 0; ...
     0 0 0 0 0 1];
B = [1 0 0 0 0 0];
Q = eye(6);
R = 1000 * eye(2);
persistent x_est p_est
if isempty(x_est)
    x_est = zeros(6, 1);
    p_est = zeros(6, 6);
end
% Predicted state and covariance
x_pred = A * x_est;
p_pred = A * p_est * A' + Q;
% Estimation
% Compute the estimated measurements
y = B * x_pred;
end
    
```

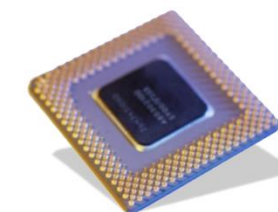
Generovaný C nebo HDL kód

Generated C and HDL code, showing the transition from MATLAB to hardware:

```

function y = kalmanfilter(s)
%codegen
% Initialize state transition matrix
A = [1 0 0 0 0 0; ...
     0 1 0 0 0 0; ...
     0 0 1 0 0 0; ...
     0 0 0 1 0 0; ...
     0 0 0 0 1 0; ...
     0 0 0 0 0 1];
B = [1 0 0 0 0 0];
Q = eye(6);
R = 1000 * eye(2);
persistent x_est p_est
if isempty(x_est)
    x_est = zeros(6, 1);
    p_est = zeros(6, 6);
end
% Predicted state and covariance
x_pred = A * x_est;
p_pred = A * p_est * A' + Q;
% Estimation
% Compute the estimated measurements
y = B * x_pred;
end
    
```

Implementace



Real-Time simulace



Vývoj metodou Model-Based Design

