
VAF calculation function for the VAR model

```
function [VAF] = VAF_VAR(G,H, sigma_e, A, K, phi_sim)
% VAF calculation
% IN
% G      : measurement matrix
% H      : influence matrix mapping the wavefront on the mirror
% sigma_e : measurement noise parameter for determining its covariance
% A : A matrix from the state space model
% K : Kalman gain value
% phi_sim : simulation data for the wavefront
% OUT
% VAF : VAF value

% dimension lifted wavefront
n_H = size(H,1);

% dimension lifted sensor slopes
n_G = size(G,1);

% Number of sample points for phi_sim
T = length(phi_sim);

u = zeros(n_H,T);

% epsilon matrix
eps_k = zeros(size(phi_sim,1),T);

% An assumption is that H is full rank.

%  $s(k) = G*eps(k) + e(k)$ 
% Hence  $s(k) = G*phi(k) - G*H*u(k-1) + e(k)$ 

%  $u(k) = 0$  for  $k < 1$ 

%  $eps(k+1|k) = Ks(k) + A*H*u(k-1) - H*u(k) + (A - KG)eps(k|k-1)$ 

%  $eps(k+1|k) = K*G*phi(k) - K*G*H*u(k-1) + K*e(k) + A*H*u(k-1) - H*u(k) + (A - KG)eps(k|k-1)$ 

% First slope value, no input applied before
s_k = G*phi_sim(:,1) + sigma_e*randn(n_G,1);

% Initial input
u(:,1) = H\((K*s_k);

%  $e(1|0) = 0$ ;
eps_kplus1k(:,1) = K*G*phi_sim(:,1) + K*sigma_e*randn(n_G,1) - H*u(:,1);

deviation_phi_norm_sum = 0;
actual_phi_norm_sum = 0;
```

```

for k = 2:T-1
    % Slope value
    s_k = G*phi_sim(:,k) - G*H*u(:,k-1) + sigma_e*randn(n_G,1);

    % Optimum input value
    u(:,k) = H\((K*s_k + (A - K*G)*eps_kplus1k(:,k-1) + A*H*u(:,k-1)));

    % Next eps(k+1|k) value
    eps_kplus1k(:,k) = K*s_k + A*H*u(:,k-1) - H*u(:,k) + (A -
    K*G)*eps_kplus1k(:,k-1);

    predicted_phi = eps_kplus1k(:,k) + H*u(:,k);
    predicted_phi = predicted_phi - mean(predicted_phi);
    phi_sim_mean_removed = phi_sim(:,k+1) - mean(phi_sim(:,k+1));
    current_norm = (norm(phi_sim_mean_removed - predicted_phi))^2;
    deviation_phi_norm_sum = deviation_phi_norm_sum + current_norm;
    actual_phi_norm_sum = actual_phi_norm_sum +
    (norm(phi_sim_mean_removed))^2;

end

mean_deviation_norm = deviation_phi_norm_sum/(T-2);
mean_actual_norm = actual_phi_norm_sum/(T-2);
VAF = max(0,100*(1 - mean_deviation_norm/mean_actual_norm));

end

```

Published with MATLAB® R2018a