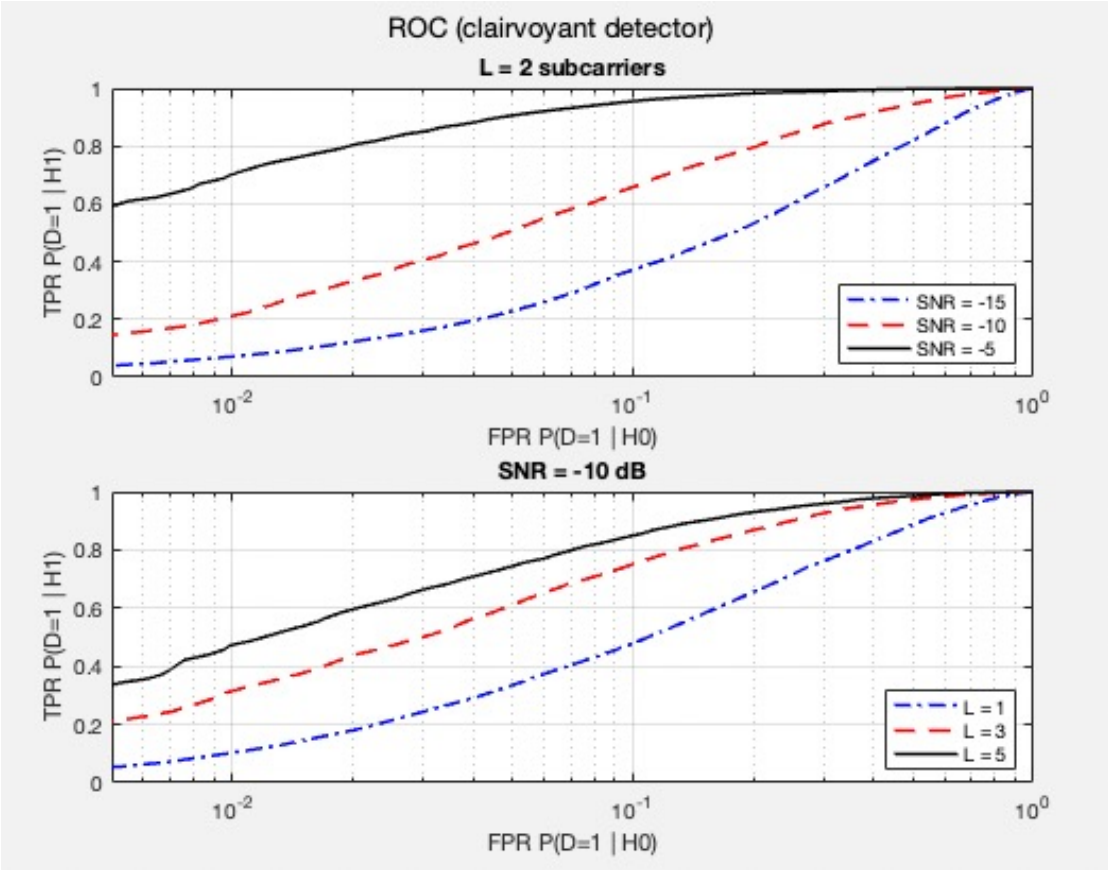
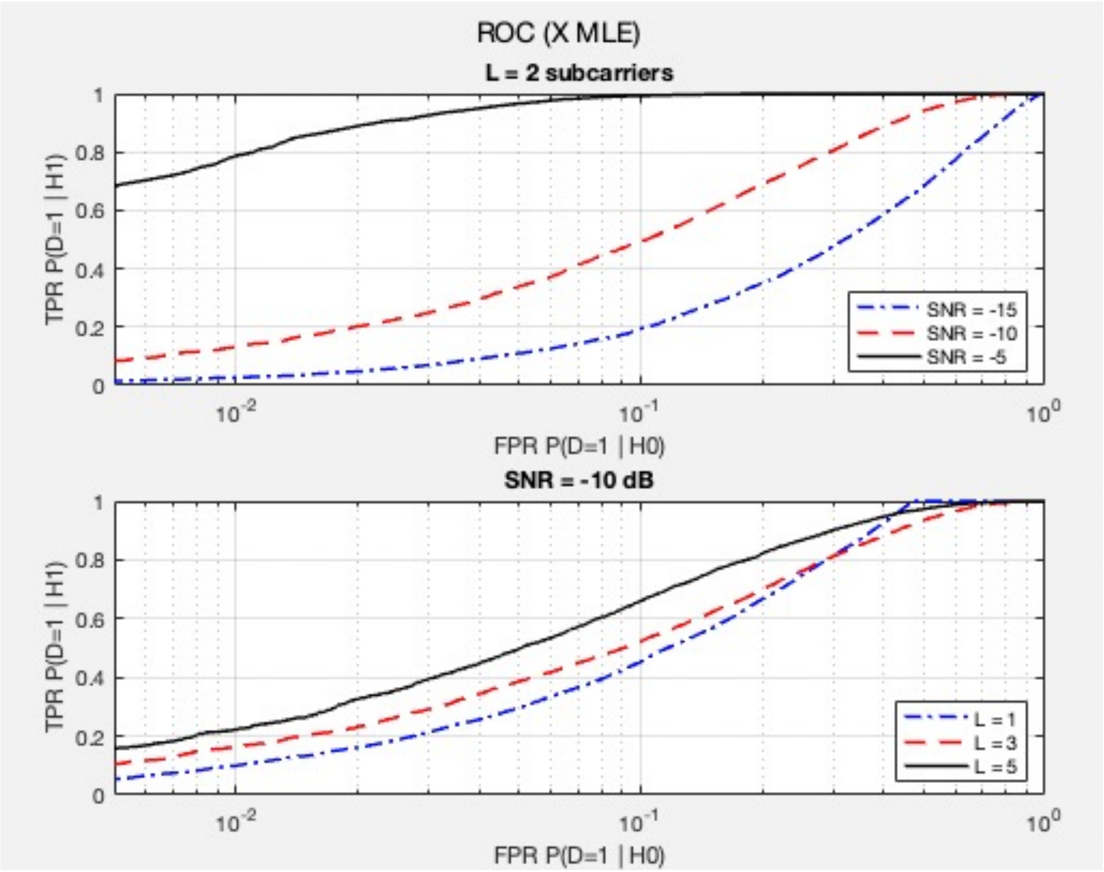
Cognitive IRR

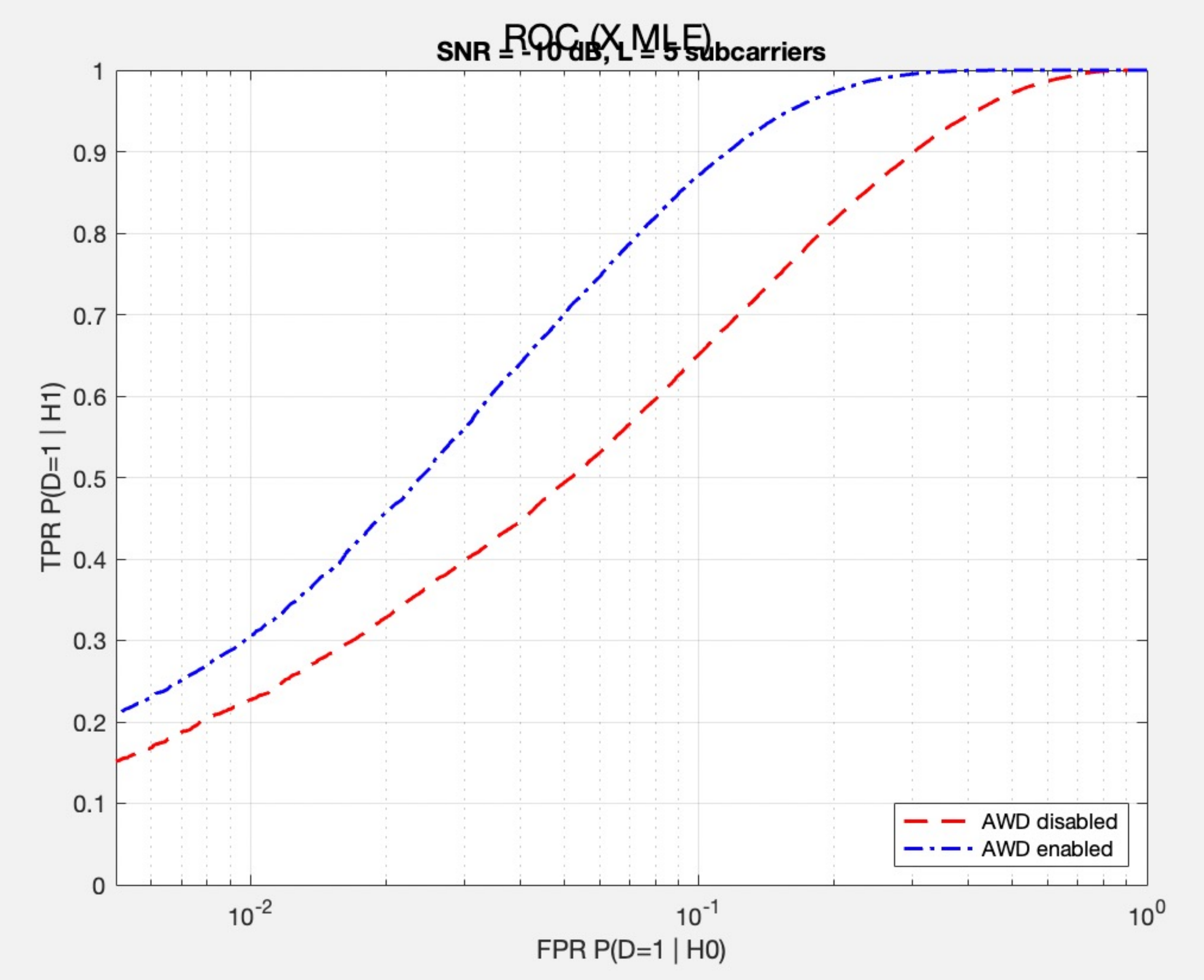
Update Tue 15th Aug

Detection results





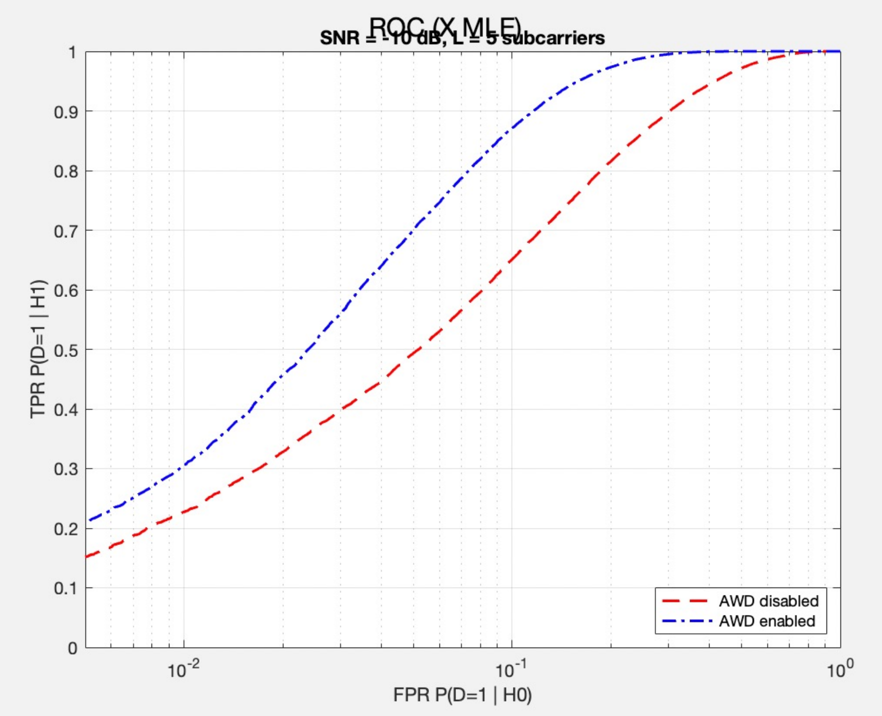
Waveform optimization (AWD module)



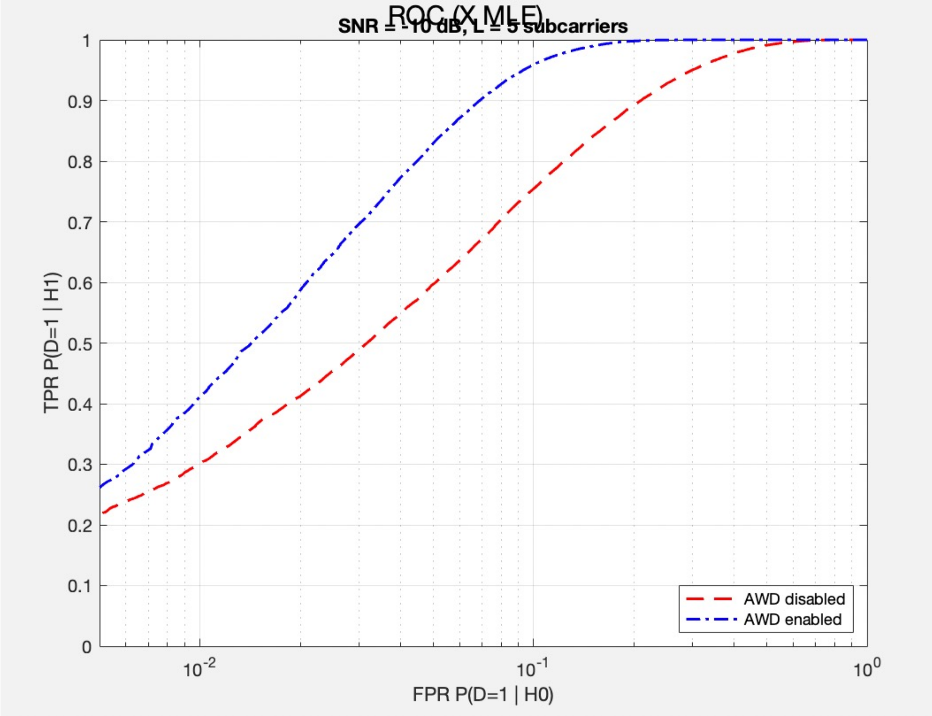

```
Command Window

A_nonopt =

     4.472135954999579e-01                         0                         0                         0                         0
                         0     4.472135954999579e-01                         0                         0                         0
                         0                         0     4.472135954999579e-01                         0                         0
                         0                         0                         0     4.472135954999579e-01                         0
                         0                         0                         0                         0     4.472135954999579e-01

K>> A_opt

A_opt =

     4.094307129291800e-01                         0                         0                         0                         0
                         0     5.246537681258305e-01                         0                         0                         0
                         0                         0     6.057606806448812e-01                         0                         0
                         0                         0                         0     8.284657941164966e-01                         0
                         0                         0                         0                         0     6.942260073469741e-01

K>>
```

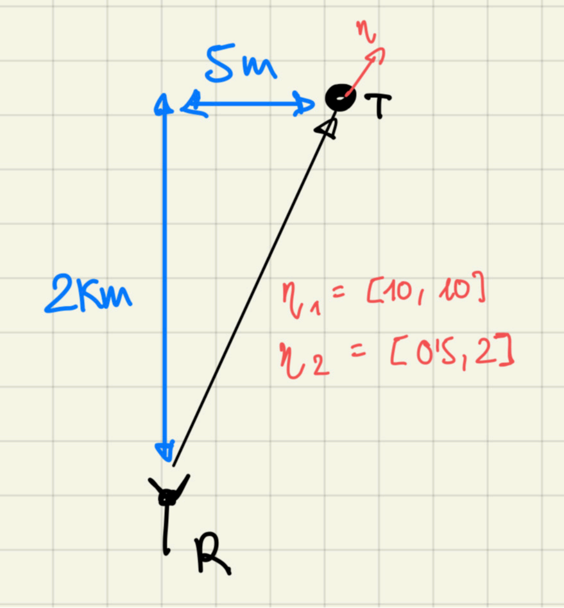
Reducing angle between target velocity vector and radar LOS

```
config.target_velocity = [10; 10];
```




```
config.target_velocity = [0.5; 2];
```

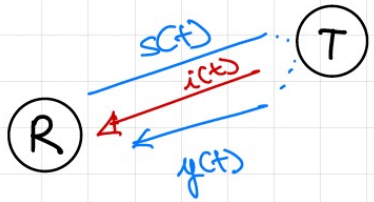




Interference module (1)

Helena August 2023

Interference Module



Type 1 : Delay Interference → T sends $i_1(t)$ to lie about its POSITION

Description: uniform distribution around correct delay value.

Type 2 : Doppler Interference → T sends $i_2(t)$ to lie about its VELOCITY

Description: uniform distribution around the correct doppler freq.
Adversarial effect on $\Phi(\eta_T)$ matrix, where $\eta_T = [v_x, v_y]^T$ is the target velocity

Interference module (2)

Description: uniform distribution around the correct doppler freq.

Adversarial effect on $\Phi(\eta_T)$ matrix, where $\eta_T = [v_x, v_y]^T$ is the target velocity

$\Phi = [\phi(t_1,\eta) \cdots \phi(t_2,\eta)]$ $L \times N$

$\phi(t,\eta) = \left[e^{j\omega_{1,D} \cdot t} \cdots e^{j\omega_{L,D} \cdot t} \right]^T$ $L \times 1$

$$\boxed{\begin{array}{l} \omega_{l,D} = 2\pi\beta f_l \ , \ \beta = 2 \cdot \langle \vec{\eta}, \vec{u} \rangle / c \\ f_l = f_c + l\,\Delta f \end{array}}$$

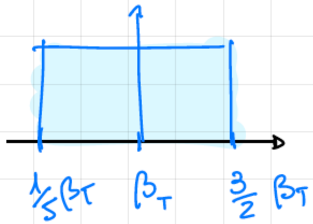
- Example Φ for $L = 2$ subcarriers and $N = 3$:

$t_n = (\tau_0 + n \cdot T_p) \ \forall n = 0, \dots, N-1$

$$\Phi = \begin{bmatrix} e^{j2\pi\beta f_0 t_1} & e^{j2\pi\beta f_0 t_2} & e^{j2\pi\beta f_0 t_3} \\ e^{j2\pi\beta f_1 t_1} & e^{j2\pi\beta f_1 t_2} & e^{j2\pi\beta f_1 t_3} \\ e^{j2\pi\beta f_2 t_1} & e^{j2\pi\beta f_2 t_2} & e^{j2\pi\beta f_2 t_3} \end{bmatrix} \longrightarrow \text{same subcarrier freq}$$

Same time instant

β is the RELATIVE DOPPLER SHIFT along the signal path



$\beta_i = \beta_T \left[1 + \alpha \cdot \mathcal{U}(-1, 1) \right]$

$\beta_T = 2/c \ \langle \vec{\eta}, \vec{v} \rangle$

Interference module (3)

For each time instant t, we have $\underbrace{\phi(t, \eta_T)}_{\text{TRUE}}$ and $\underbrace{\phi(t, \eta_i)}_{\text{interfered}}$

(we have knowledge of it)

```matlab
function varargout = get_ROC_from_config(config, LOS_vector, T_rangecell)

    % Given the input configuration, obtain the OFDM model parameters
    [A_nonopt, X_H1, Phi_t, Sigma_c] = get_OFDM_model(config, LOS_vector, T_rangecell);

    % Initialize GLRT metrics
    GLRT_H0 = zeros(config.N_mc, 1);
    GLRT_H1 = zeros(config.N_mc, 1);

    % Initialize GLRT metrics for AWD, if enabled
    if config.enable_awd
        GLRT_H0_opt = zeros(config.N_mc, 1);
        GLRT_H1_opt = zeros(config.N_mc, 1);
    end

    % For each Monte Carlo realization, generate OFDM measurements
    finalA = zeros(size(A_nonopt,1), size(A_nonopt,2), config.N_mc);
    for idx_mc = 1:config.N_mc

        [Y_H0, Y_H1] = build_OFDM_meas(config, Sigma_c, A_nonopt, X_H1, Phi_t);

        % MLE of target coefficients X under hypothesis H1
        if config.clairvoyant
            X_hat = X_H1;
        else
            X_hat = inv(A_nonopt) * (Y_H1*Phi_t') * pinv(Phi_t*Phi_t'); % expression after eq 8 paper 0
%             X_hat = inv(A_nonopt)*Y_H1*Phi_t'*inv(Phi_t*Phi_t');
%             X_hat = diag(diag(X_hat));
        end

        % GLRT (non-optimal)
        GLRT_H0(idx_mc, 1)= det(Y_H0*Y_H0')/det((Y_H0-A_nonopt*X_hat*Phi_t)*(Y_H0-A_nonopt*X_hat*Phi_t)');
        GLRT_H1(idx_mc, 1)= det(Y_H1*Y_H1')/det((Y_H1-A_nonopt*X_hat*Phi_t)*(Y_H1-A_nonopt*X_hat*Phi_t)');
```

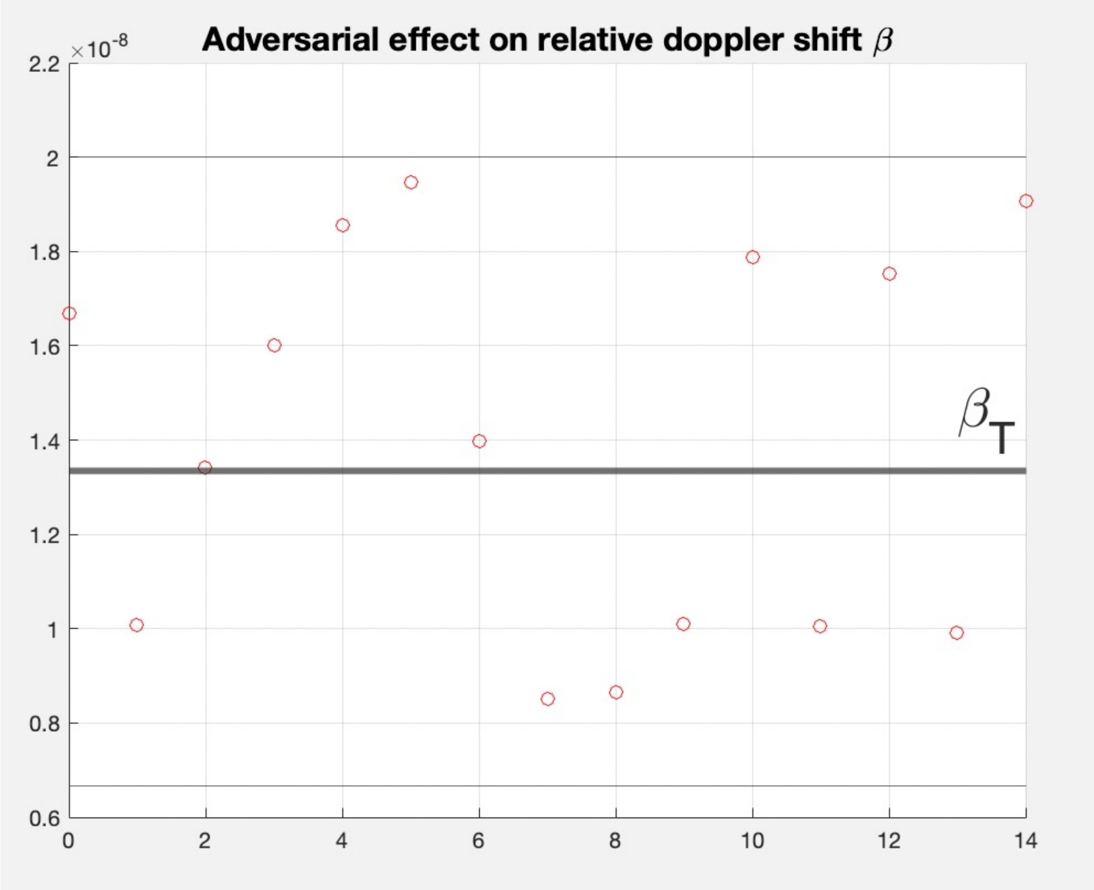
```matlab
        % Adaptive Waveform Design (AWD) module:
        if config.enable_awd
            A_ini = ones(1,config.L);
%             A_ini = diag(A_nonopt);
            sgm_sqr = 1; % set to 1 (best!) or sqrt(Sigma_c(1,1)) for good AWD performance
            [A_opt, ~] = fminsearch(@(A) opt_waveform(A, config.L, X_H1, Phi_t, chol(Sigma_c), sgm_sqr), A_ini);
            A_opt = diag(A_opt);
            finalA(:,:,idx_mc) = A_opt;
```

this damages the AWD module!

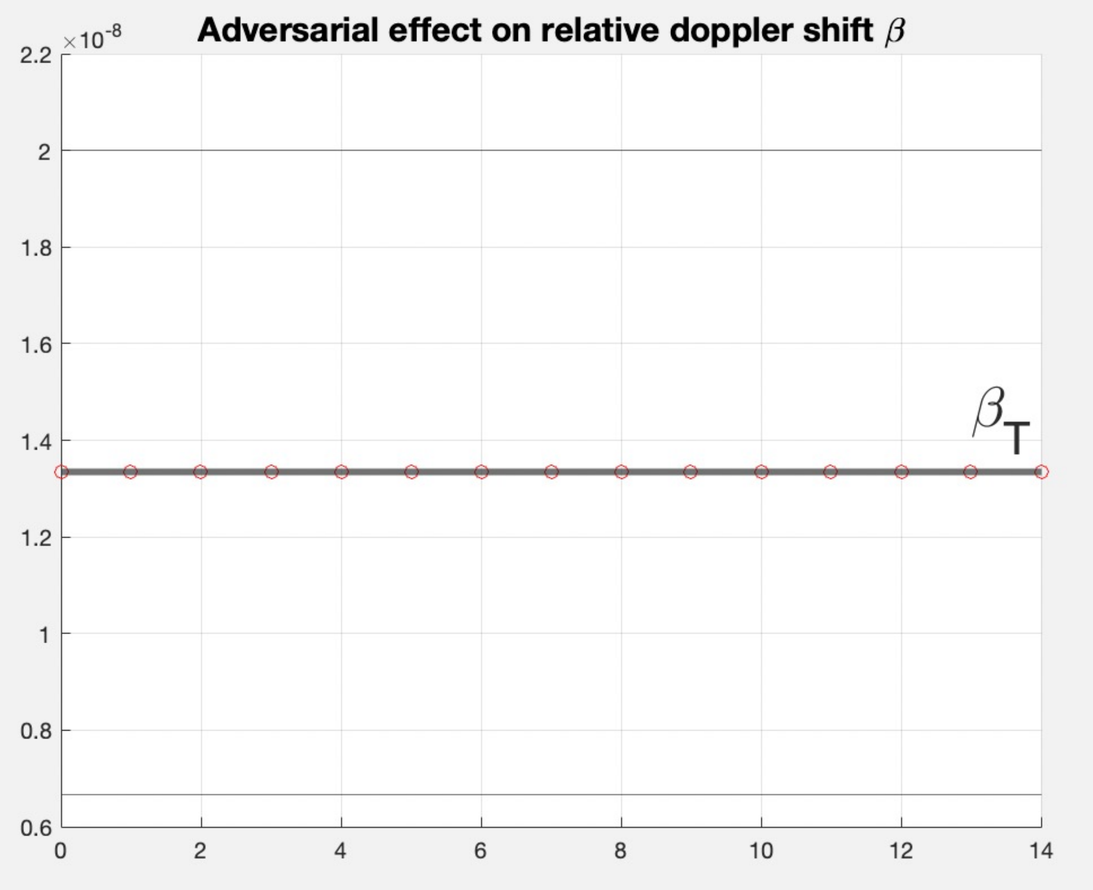
```matlab

            % GLRT (optimal)
            X_hat_opt = X_hat;
%             X_hat_opt = inv(A_opt)*Y_H1*Phi_t'*inv(Phi_t*Phi_t');
            GLRT_H0_opt(idx_mc, 1)= det(Y_H0*Y_H0')/det((Y_H0-A_opt*X_hat_opt*Phi_t)*(Y_H0-A_opt*X_hat_opt*Phi_t)');
            GLRT_H1_opt(idx_mc, 1)= det(Y_H1*Y_H1')/det((Y_H1-A_opt*X_hat_opt*Phi_t)*(Y_H1-A_opt*X_hat_opt*Phi_t)');
        end

        if mod(idx_mc, 50) == 0
            disp(['Monte carlo it ', num2str(idx_mc)])
        end
    end

%     opt_A = mean(finalA); % we average A matrices from all MC realizations

    [Ppn, Ppp] = get_ROC_from_GLRTs(config, GLRT_H0, GLRT_H1);
    varargout{1} = Ppn;
    varargout{2} = Ppp;

    if config.enable_awd
        [Ppn_opt, Ppp_opt] = get_ROC_from_GLRTs(config, GLRT_H0_opt, GLRT_H1_opt);
        varargout{3} = Ppn_opt;
        varargout{4} = Ppp_opt;
    end
```

Adversarial effect on rel. doppler shift (1)

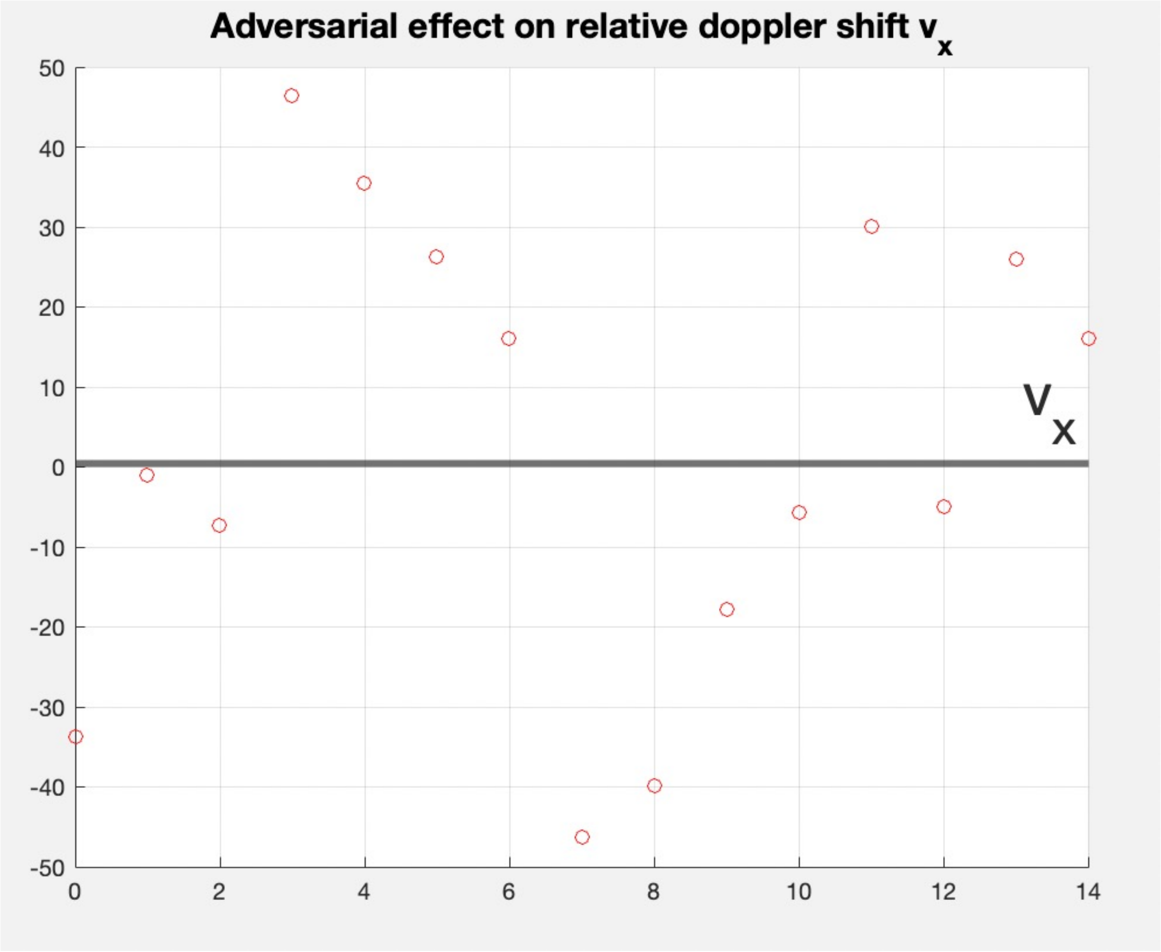
\beta_i = \beta_i + U(-1, 1)*alpha*\beta_t

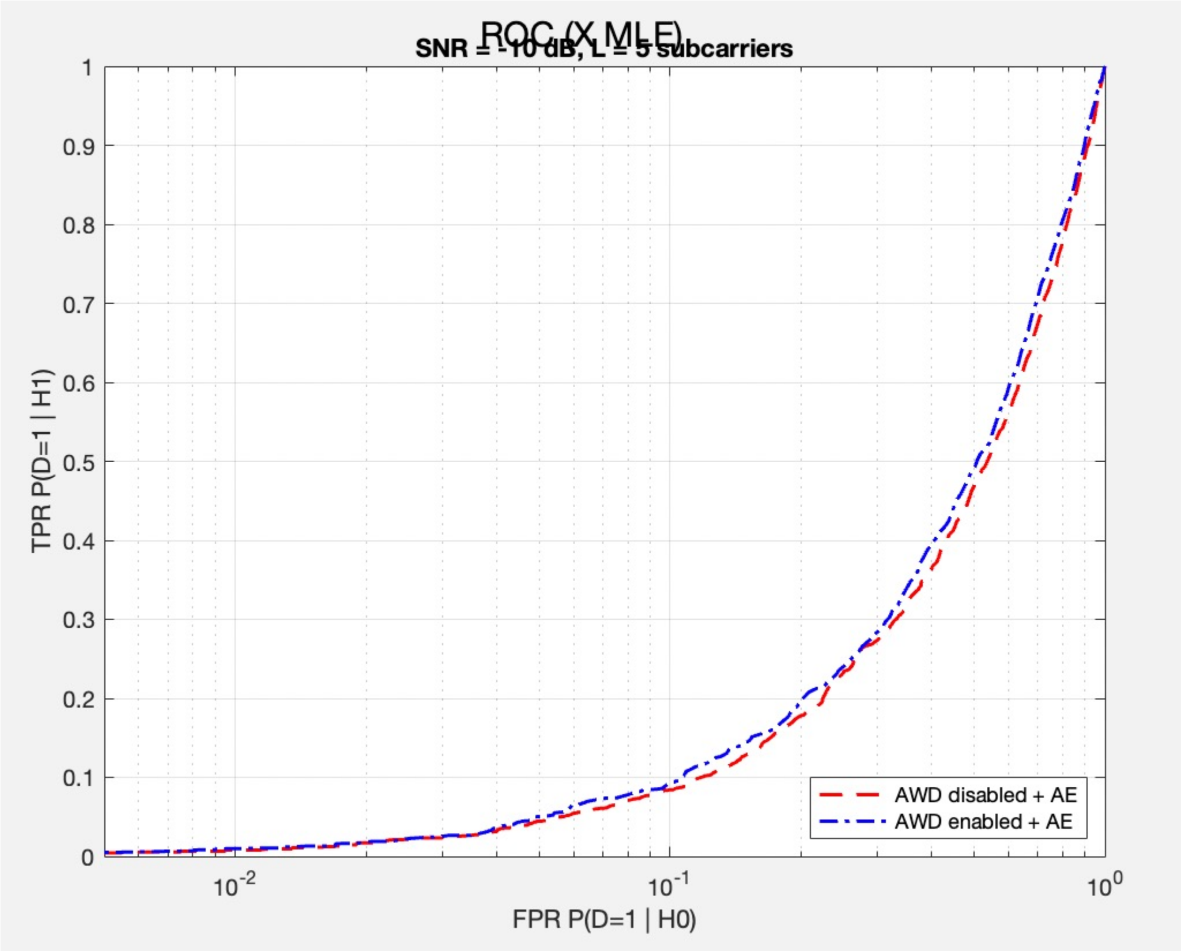


non-interference case

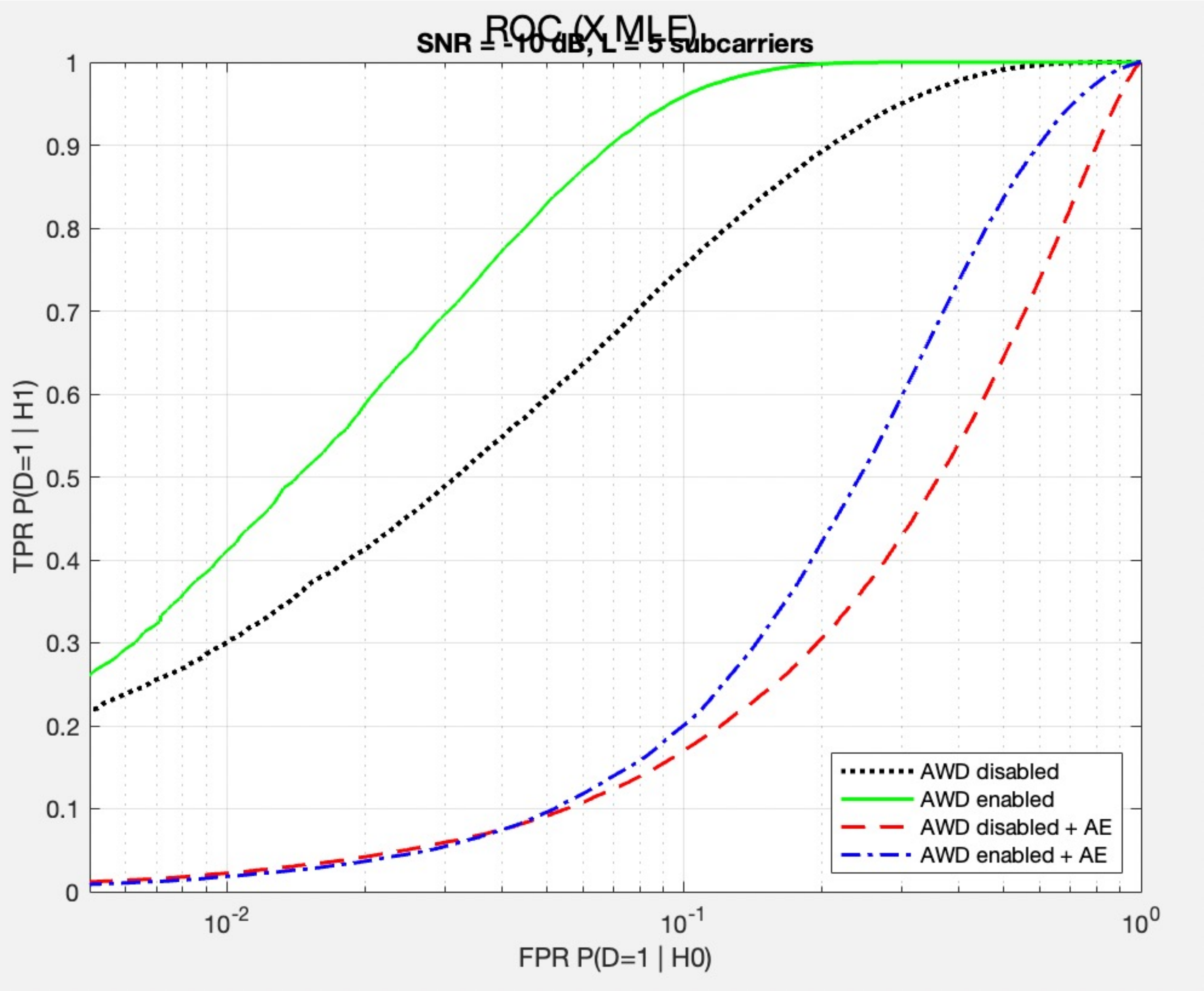


Changing target velocity





Adversarial effect on rel. doppler shift (2)

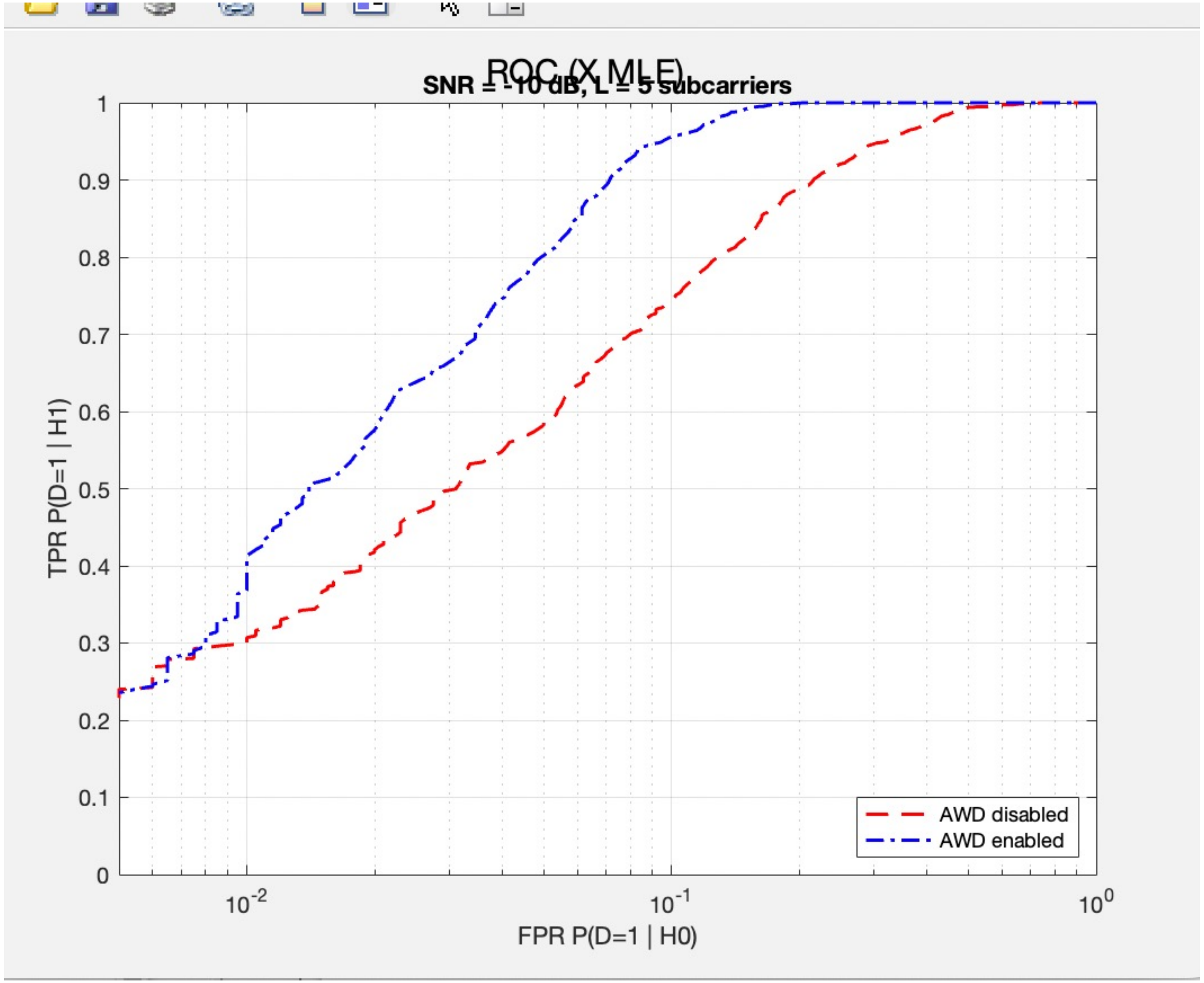


When adversarial effects are added, the AWD helps. Nevertheless, the performance is clearly disturbed even when enabling the AWD.

Our objective is to propose a method that can do that.

Conclusion/how to proceed:

- I need to make the parameter alpha too large to observe a strong adversarial effect. How does this translate in terms of velocity? Asking this to see if alpha=1000 is realistic.
- To check: AWD should not work because all subcarriers are equally disturbed.
- To think about: AWD would not work even if few subcarriers were disturbed because the AWD module uses Phi_TRUE to optimize A, while the actual Phi is Phi_INTERFERENCE.



ROC (X MLE)
SNR = -10 dB, L = 5 subcarriers
TPR P(D=1 | H1)
FPR P(D=1 | H0)
AWD disabled
AWD enabled
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
10^{-2}
10^{-1}
10^{0}