Exercise – Function – Objective

This function defines the objective used in indirect inference to evaluate how well a candidate structural parameter value, `theta_candidate`, replicates the behavior of observed data. It operates by simulating `N_sim` synthetic time series using `theta_candidate` in an MA(1) process. For each simulated series, an auxiliary AR(1) model is fitted to estimate the autoregressive coefficient, denoted `beta_hat_sim`. These estimates are averaged to produce `beta_tilde`, a summary statistic representing the simulated data. The function then compares `beta_tilde` to `beta_hat`, the auxiliary estimate obtained from the observed data, using a quadratic form weighted by matrix `W`. The resulting value `Q` measures the discrepancy between simulated and observed behavior. Lower values of `Q` indicate closer alignment, guiding the optimization process toward the best-fitting structural parameter theta.

```matlab
% Exercise - Function - Objective

% This function computes the objective value for indirect inference.
% It compares the auxiliary parameter (AR(1) estimate) from observed
% data with the average auxiliary estimates from simulated MA(1) data.
function Q = objective(theta_candidate,N_sim,T,W,beta_hat)
    %% Preallocate vector to store AR(1) estimates from simulations
    beta_hat_sim = NaN(N_sim,1);
    %% Simulate MA(1) data and estimate AR(1) parameters
    for i = 1:N_sim
        %% Simulate error terms (white noise)
        epsilon_sim = random("Normal",0,1,[T+1,1]);
        %% Preallocate vector to store simulated MA(1) outcomes
        y_sim = zeros(T,1);
        %% Generate simulated data from MA(1) using theta_candidate
        for t = 2:T+1
            y_sim(t-1) = epsilon_sim(t)+theta_candidate ...
                *epsilon_sim(t-1);
        end
        %% Estimate AR(1) coefficient from simulated data
        beta_hat_sim(i) = auxiliary(y_sim);
    end
    %% Compute average AR(1) estimate across simulations
    beta_tilde = mean(beta_hat_sim);
    %% Calculate the distance between observed and simulated estimates
    Q = (beta_hat-beta_tilde)'*W*(beta_hat-beta_tilde);
end
```