

Apéndice 4: Algoritmo Orthogonal Matching Pursuit (OMP)

```
%-----%
%IMPLEMENTACIÓN DEL ALGORITMO ORTHOGONAL MATCHING PURSUIT PARA LA
%RECONSTRUCCIÓN DE SEÑALES SPARSE
%-----%

function OMP

n = 1024;           % Longitud de la señal x
m = 300;           % Número de proyecciones aleatorias
k = 20;            % Nivel de escasez de la señal

% Generación de la señal sparse

x = zeros (n,1);   % Generar señal aleatoria
x(1:k) = randn(k,1);

%Generación del diccionario

psi = eye(n,n);    % Diccionario(matriz identidad)

% Proyecciones aleatorias de la señal x

phi = 2*randint(m,n)-1; % Matriz de proyecciones aleatorias
y = phi*x;         % Proyecciones aleatorias de la señal

% ALGORITMO OMP

% Inicialización

r = zeros(m,2);   % Residuo en las iteraciones t y t-1
r(:,1) = y;       % Residuo igual a las proyecciones
V = phi*psi;      % Diccionario holográfico
t = 0;           % Contador de iteraciones igual a 1
A = zeros(m,k);  % Matriz que almacena los átomos
                  % seleccionados en cada iteración

epsilon = 10e-16;
xest = zeros(n,1); % Iniciar a cero el estimado de la
                  % señal
 exitos = 0;      % Contador de reconstrucciones
                  % exitosas
 error = 0;       % Acumulador de error

% Norma de las columnas de V = phi*psi

Vn = sqrt(sum(V.^2));

while(t < k)      % Decisión
    t = t+1;     % Incremento del contador de
                  % iteraciones
    nt = abs (V'*r(:,1)./Vn'); % Proyecciones del residuo sobre
                              % los átomos del diccionario
                              % holográfico
```

```

[val pos] = max(nt);           % Valor y posición de la máxima
                                % proyección
ind(1,t) = pos;               % Agregar el índice al conjunto de
                                % índices
A(:,t) = V(:,pos);           % Obtener la nueva matriz de medición
xt = pinv(A)*y;               % Obtener el estimado de la señal
at = A*xt;                    % Calcular la nueva aproximación
r(:,2) = r(:,1)- A*pinv(A)*r(:,1); % Calculo del nuevo
                                % residuo
r(:,1) = r(:,2);             % Actualizar el residuo

end

% Obtengo el estimado de la señal

for i=1:1:length(ind)

    xest(ind,:) = xt;

end

```