```
function [x_ukf, P_ukf] = ukf_propagation(x0, P0, tprop)
%--------------------------------------------------------------------------
% UKF_PROPAGATION Propagate orbit uncertainty using Unscented Transform
%--------------------------------------------------------------------------
%
% Inputs:
%   x0    : 6x1 initial state [r0; v0]
%   P0    : 6x6 initial covariance
%   tprop : propagation time [s]
%   mu    : gravitational parameter [m^3/s^2]
%
% Outputs:
%   x_ukf : propagated mean state
%   P_ukf : propagated covariance

    n = 6;              % state dimension
    alpha = 1e-3;
    beta = 2;
    kappa = 0;

    lambda = alpha^2*(n + kappa) - n;

    % Sigma point weights
    Wm = [lambda/(n+lambda); 0.5/(n+lambda)*ones(2*n,1)];
    Wc = Wm;
    Wc(1) = Wm(1) + (1 - alpha^2 + beta);

    % Sigma points
    % P0 = (P0 + P0')/2;  % force symmetry
    % [eigvec, eigval] = eig(P0);
    % eigval(eigval < 0) = 0;  % set negative eigenvalues to zero
    % P_psd = eigvec * eigval * eigvec';
    %
    P0 = (P0 + P0') / 2;
    % Eigen-decomposition
    [V, D] = eig(P0);
    D(D < 0) = 0;           % zero negative eigenvalues
    P_clean = V * D * V';
    % Optional small diagonal for strict PD
    P_clean = P_clean + 1e-12 * eye(size(P_clean));



    S = chol((n+lambda)*P_clean, 'lower');
    Xi = [x0, x0+S, x0-S];    % 6 x (2n+1)

    % Propagate sigma points
    Xi_prop = zeros(size(Xi));
    for i = 1:(2*n+1)
        [r_temp,v_temp, ~, ~, ~] = Propagate(tprop, Xi(1:3,i), Xi(4:6,i), 0);
        Xi_prop(:,i) = [r_temp; v_temp];
    end

    % Reconstruct mean
    x_ukf = Xi_prop * Wm;

    % Reconstruct covariance
    P_ukf = zeros(n);
    for i = 1:(2*n+1)
        dx = Xi_prop(:,i) - x_ukf;
        P_ukf = P_ukf + Wc(i) * (dx*dx');
    end
end
```