```
function [phi, theta, psi] = xyz_to_zxz(alpha, beta, gamma)
    % Input:
    % alpha - rotation about x-axis (in radians)
    % beta  - rotation about y-axis (in radians)
    % gamma - rotation about z-axis (in radians)
    %
    % Output:
    % phi   - first rotation about z-axis in z-x-z convention (in radians)
    % theta - rotation about x-axis in z-x-z convention (in radians)
    % psi   - second rotation about z-axis in z-x-z convention (in radians)

    % Step 1: Create the rotation matrix for the x-y-z rotation sequence
    R_x = [1, 0, 0;
           0, cosd(alpha), -sind(alpha);
           0, sind(alpha), cosd(alpha)];

    R_y = [cosd(beta), 0, sind(beta);
           0, 1, 0;
           -sind(beta), 0, cosd(beta)];

    R_z = [cosd(gamma), -sind(gamma), 0;
           sind(gamma), cosd(gamma), 0;
           0, 0, 1];

    % Combined rotation matrix for x-y-z convention
    R_xyz = R_z * R_y * R_x;

    % Step 2: Extract the Euler angles in z-x-z convention from the rotation matrix
    theta = acosd(R_xyz(3,3)); % theta is the angle between z-axes

    if abs(sind(theta)) > 1e-6  % Check if sin(theta) is not too close to 0 (singularity)
        psi = atan2d(R_xyz(3,2), R_xyz(3,1)); % psi (final rotation about z)
        phi = atan2d(R_xyz(2,3), -R_xyz(1,3)); % phi (initial rotation about z)
    else
        % Gimbal lock condition: theta = 0 or pi
        phi = 0;  % Arbitrary value, because the first rotation is not defined
        psi = atan2d(R_xyz(1,2), R_xyz(1,1)); % Combine rotations around z-axis
    end
end
```