```
function radius = turn_radius(aircraft,h,v)
% TURN_RADIUS Finds the minimum turning radius given aircraft parameters
%             as well as the cruise conditions.
%   Inputs are:
%   aircraft   :a struct aircraft data in SI
%   h          :a numeric array of Mx1 altitude in m
%   v          :a numeric array of Mx1 cruise speed in m/s
%
%   Output is:
%   radius     :a numeric array of Mx1 minimum turn radius in m

    arguments
        aircraft {mustBeA(aircraft,"struct")}
        h (:,1) {mustBeNumeric, mustBeReal}
        v (:,1) {mustBeNumeric, mustBeReal}
    end

    W = aircraft.W;
    S = aircraft.S;
    Cd_0 = aircraft.Cd_0;
    Cl_max = aircraft.Cl_max;
    K = aircraft.K;
    Tsl = aircraft.Tsl;
    nmax_s = aircraft.nmax_s;

    if length(h) == 1 && length(v) > 1
        h = h.*ones(length(v));
    elseif length(v) == 1 && length(h) > 1
        v = v.*ones(length(v));
    elseif length(h) ~= length(v)
        error('Incompatible h and v array sizes')
    end

    [~,~,rho] = stdatm(h); % atmospheric density at altitude (kg/m^3)
    Q = 0.5.*rho.*v.^2; % dynamic pressure (N/m^2)
    T = Tsl.*(rho./1.225); % thrust at altitude (N)

    nmax_s = nmax_s.*ones(1,length(h));
    nmax_t = sqrt((Q./(K.*W./S)).*((T./W)-((Q.*Cd_0)./(W./S))));
    nmax_alpha = (Q.*Cl_max)./(W./S);

    radius = zeros(1,length(h));
    for i = 1:length(h)
        n = [nmax_s(i);nmax_t(i);nmax_alpha(i)];
        radius(i) = (v(i).^2)./(9.81.*sqrt((min(n).^2)-1));
    end
end
```