```
function  [RC,gamma,maxRC,vmaxRC,gammamaxRC] = steady_climb(aircraft,h,v)
% STEADY_CLIMB Steady climb rate and maximum RC with velocity
%   Inputs are:
%   aircraft   :a struct aircraft data in SI
%   h          :a numeric array of 1xN aircraft altitude in m
%   v          :an optional numeric array of Mx1 aircraft velocity in m/s
%
%   Outputs are:
%   RC         :a numeric array of 1xN steady climb rate in m/s
%   gammamaxRC :a numeric array of 1xN climb angle in radians
%   maxRC      :a numeric array of 1xN maximum steady climb rate in m/s
%   vmaxRC     :a numeric array of 1xN velocity at maximum steady climb
%               rate in m/s
%   gammamaxRC :a numeric array of 1xN climb angle at maximum steady climb
%               rate in radians
%
%   When no v is passed to the function RC and gamma are set to NaN, with
%   the presumption that the arguments will be ignored in the output call.

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

    W = aircraft.W;
    S_w = aircraft.S_w;
    Cd_0 = aircraft.Cd_0;
    K = aircraft.K;
    Tsl = aircraft.Tsl;

    [~,~,rho] = stdatm(h); % atmospheric density at altitude (kg/m^3)
    T = Tsl.*(rho./1.225); % thrust at altitude (N)
    TtoW = T./W; % thrust to weight ratio

    switch nargin
        case 2
            RC = NaN;
            gamma = NaN;
        case 3
            Q = 0.5.*rho*v.^2; % dynamic pressure (N/m^2)
            RC = v.*(TtoW-((Q.*Cd_0)./(W./S_w))-(W./S_w).*(K./Q));
            gamma = asin(RC./v);
    end

    [~,maxLtoD] = LtoD_ratio(aircraft,h); % maximum L/D

    Z = 1+sqrt(1+(3./((maxLtoD.^2).*(TtoW.^2))));
    maxRC = sqrt(((W./S_w).*Z)/(3.*rho.*Cd_0)).*(TtoW.^(3./2)).*(1-(Z./6)-(3./(2.*(TtoW.↙
^2).*(maxLtoD.^2).*Z)));
    vmaxRC = sqrt((TtoW.*(W./S_w).*Z)./(3.*rho.*Cd_0));
    gammamaxRC = asin(maxRC./vmaxRC);
end
```