```
function [tFull, xFull, uFull, cmdFull] = UAVFlyWaypointSequence(x0_orig, wpSet, data, Rmin, hDotMax)
    % Trevor Burgoyne
    % 9 Dec 2021
    %
    % Usage: Simulates a UAV flight that starts at the initial state and
    % flies to a sequence of waypoints.
    % Function Call: [tFull, xFull, uFull, cmdFull] = UAVFlyWaypointSequence(x0_orig, wpSet, p)
    %
    % INPUTS:
    %
    % x0_orig     (1,7)   x = [V;gamma;psi;x;y;h;Tbar]
    %
    %     V     true airspeed (m/s)
    %     gamma  air relative flight path angle (rad)
    %     psi   air relative flight heading angle (rad)
    %     x     East position (m)
    %     y     North position (m)
    %     h     altitude (m)
    %     Tbar  normalized excess thrust
    %
    % wpSet       (3,N)     matrix of N waypoints, in order
    % Rmin        (1,1)     minimun turn radius of UAV
    % hDotMax     (1,1)     maximum altitude rate of change
    %
    % OUTPUTS:
    % tFull       (1,M)     time vector
    % xFull       (7,M)     states across time, in form x = [V;gamma;psi;x;y;h;Tbar]
    % uFull       (3,M)     controls across time
    % cmdFull     (5,M)     commands across time

    x0 = x0_orig; % For first waypoint, we start at x0_orig
    n = size(wpSet,2); % number of waypoints
    tFull = []; xFull = []; uFull = []; cmdFull = []; % Initialize arrays to store all the flight data

    % Loop through the waypoints and navigate from point to point
    for i=1:n

        % Get current waypoint
        wp = wpSet(:,i); % column vector with [x; y; h;]

        % set the flight parameters for this segment
        p = struct();
        p.wp = wp;
        p.Rmin = Rmin;
        p.hDotMax = hDotMax;
        p.dT = .001; % sec
        p.duration = 120; % sec

        % Stopping function
        stop = @(t,x) stopSim(t,x,wp,p.duration);
        p.stopSim = stop;

        % disp(i)
        % disp('delta_h')
        % disp(wp(3)-x0(6))

        % Navigate to waypoint from current x0
        [tSeg, xSeg, uSeg, cmdSeg] = UAVFlyToWaypoint(x0, data, p);
```

```
        % Append path details to the total flight data arrays
        xFull = [xFull, xSeg];
        uFull = [uFull, uSeg];
        cmdFull = [cmdFull, cmdSeg];

        % Time vector need to be offset based on last waypoint's final timestamp
        if (i > 1)
            tSeg = tSeg + tFull(end);
        end
        tFull = [tFull, tSeg];

        % Update x0 to be the final state just calculated (last column of xSeg)
        x0 = xSeg(:, end);

    end

end
```

Published with MATLAB® R2020a