

```

function [kep_est_final, rv_cov_final] = batch_gps_multiarc(times, measurements, ↵
R_obs, arc_length, Epoch)
%-----
% Multi-arc batch LS where the last observation of each arc becomes the first ↵
observation of the next arc
%-----
%
% Inputs:
%   times      [Nx1] time stamps
%   measurements [Nx6] r,v measurements
%   R_obs      [6x6] measurement covariance
%   arc_length  arc duration in seconds
%
% Outputs:
%   kep_est_final  final arc's Kepler estimate
%   rv_cov_final   final arc's covariance
%-----

N = length(times);
arc_solutions = struct([]);

% -----
% Build arc index ranges
% -----
arc_starts = [];
arc_ends = [];

i1 = 1; % start index

while true
    t_start = times(i1);
    t_end = t_start + arc_length;

    % find max index within arc length
    i2 = find(times <= t_end, 1, 'last');

    if isempty(i2) || i2 <= i1
        break;
    end

    arc_starts(end+1) = i1;
    arc_ends(end+1) = i2;

    % Next arc starts at the LAST index of current arc
    i1 = i2;

    if i1 >= N
        break;
    end
end

n_arcs = length(arc_starts);

% -----

```

```
% Process each arc
%
for j = 1:n_arcs
    idx1 = arc_starts(j);
    idx2 = arc_ends(j);

    t_arc = times(idx1:idx2);
    measArc = measurements(idx1:idx2,:);

    % Perform batch LS on the arc
    [kep_est, rv_cov] = batch_gps(t_arc, measArc, R_obs, Epoch, measurements(1,:));

    arc_solutions(j).kep = kep_est;
    arc_solutions(j).P = rv_cov;
    arc_solutions(j).range = [idx1 idx2];

    % Propagate final state to next arc's first timestamp
    if j < n_arcs
        % Convert to Cartesian
        [r0, v0] = keplerian2ijk(kep_est(1), kep_est(2), kep_est(3), kep_est(4), ↵
        kep_est(5), kep_est(6));

        % Propagate to next arc start time
        dt = times(arc_starts(j+1)) - t_arc(1);

        [r_next, v_next] = propagateOrbit([Epoch, Epoch+seconds(dt)], r0, v0);

        % Overwrite the first measurement of next arc's start
        measurements(arc_starts(j+1),1:3) = r_next(:, end)';
        measurements(arc_starts(j+1),4:6) = v_next(:, end)';
    end
end

% Final outputs = last arc
kep_est_final = arc_solutions(end).kep;
rv_cov_final = arc_solutions(end).P;

end
```