

```

function H = compute_H_analytic(r, v)
%-----
% Computes analytically the H matrix for a batch filter taking in RA, DEC,
% and their rates and converts them to r and v
%-----
% Inputs: r and v [m] and [m/s] of reference trajectory to find the H matrix for
rx = r(1); ry = r(2); rz = r(3);
vx = v(1); vy = v(2); vz = v(3);

rnorm2 = rx^2 + ry^2 + rz^2;
rhor2 = rx^2 + ry^2;
rho_norm = sqrt(rnorm2);
rhor = sqrt(rhor2);

% RA
dRA_dr = [-ry/rhor2, rx/rhor2, 0];
dRA_dv = [0,0,0];

% DEC
dDEC_dr = [-rx*rz/(rhor2*rho_norm), -ry*rz/(rhor2*rho_norm), rhor/rnorm2];
dDEC_dv = [0,0,0];

% RA_dot
dRAdot_dr = [(ry*(rx*vx+ry*vy)/rhor2 - vy)/rhor2, (-rx*(rx*vx+ry*vy)/rhor2 + vx) /
/rhor2, 0];
dRAdot_dv = [-ry/rhor2, rx/rhor2, 0];

% DEC_dot
dot_rv = dot(r,v);
dDECdot_dr = [-(vx*rhor2 - rx*(dot_rv*rz/rho_norm^2))/(rho_norm*rhor), ...
              -(vy*rhor2 - ry*(dot_rv*rz/rho_norm^2))/(rho_norm*rhor), ...
              -(vz*rhor2 - rz*(dot_rv*rz/rho_norm^2) + dot_rv)/(rho_norm*rhor)];
dDECdot_dv = [-rx*rz/(rho_norm*rhor), -ry*rz/(rho_norm*rhor), rhor/rho_norm];

H = [dRA_dr, dRA_dv;
      dDEC_dr, dDEC_dv;
      dRAdot_dr, dRAdot_dv;
      dDECdot_dr, dDECdot_dv];
end

```