

---

```

function state_dot = two_body_state_dot(t, state, opts)
%two_body_state_dot    Return state_dot given state. Used for numerical
%integration
% The first 6 elements are assumed to be r and v. If a state transition
% matrix (STM) is to be calculated as well, opts.OD needs to be set up.
% Currently assumes r/v are the only state elements that have non-zero
% derivatives.
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app

state_dot = zeros(length(state),1);
state_dot(1:3) = state(4:6);
mu = 3.986e5; % km3/s2
if isfield(opts, 'mu')
    mu = opts.mu;
end

r_vec = (state(1:3));
r = norm(r_vec);
state_dot(4:6) = -mu * r_vec/(r*r*r); % Simple 2-body

% Use J2
if isfield(opts, 'J2') && isfield(opts.J2, 'use')
    if opts.J2.use == 1
        if isfield(opts.J2, 'params')
            state_dot(4:6) = state_dot(4:6) + J2_accel(state(1:3), opts.J2.params)
        else
            state_dot(4:6) = state_dot(4:6) + J2_accel( state(1:3) );
        end
    end
end

% Use J3
if isfield(opts, 'J3') && isfield(opts.J3, 'use')
    if opts.J3.use == 1
        if isfield(opts.J3, 'params')
            state_dot(4:6) = state_dot(4:6) + J3_accel(state(1:3), opts.J3.params)
        else
            state_dot(4:6) = state_dot(4:6) + J3_accel( state(1:3) );
        end
    end
end

% Use drag
if isfield(opts, 'drag') && isfield(opts.drag, 'use')
    if opts.drag.use == 1
        state_dot(4:6) = state_dot(4:6) + drag_accel( state, opts.drag );
    end
end

% For orbit determination application
if isfield(opts, 'OD')
    if opts.OD.use == 1

```

---

---

```

    opts.OD.state_len;
    % The OD.state_len is the length of the estimation state. The rest
    % is the STM, numerically propagated with the A-Matrix
    A = opts.OD.A_mat_handle(state(1:opts.OD.state_len),opts.OD.A_params);

    % Block matrix multiplication
    long_dim = 9;
    STM = zeros(long_dim);
    STM = reshape(state(opts.OD.state_len+1:end),...
        opts.OD.A_params.important_block(1),...
        opts.OD.A_params.important_block(2));
    STM_dot = ...
        A(1:opts.OD.A_params.important_block(1),1:opts.OD.A_params.important_b
        *STM;
    % Pack up the important stuff
    state_dot(opts.OD.state_len+1:end) = reshape(STM_dot,...
        opts.OD.A_params.important_block(1)*opts.OD.A_params.important_block(2
end
end

```

*Published with MATLAB® R2013b*