
Table of Contents

..... 1

```
function V_soln = gen_3d_periodic_orbit_single_shooting(V0, system_params,
plot_input)
    % Script to compute a general three-dimensional periodic orbit via single
shooting
    % Inputs
    % V0 - initial guess for a free variable vector
    % statef_V0 - final state when V0 is used as initial guess using CR3BP
    % EOMs
    % system_params - system parameters
    %
    % Output
    % V_soln - free variable vector corresponding to a solution

    % Get mass ratio of system
    mu = system_params(1);

    % Set tolerance for numerical integrator and constraint vector
    TOL = 5e-14;

    % Set options for ode113
    options = odeset('RelTol', TOL, 'AbsTol', TOL);

    % Propagate V0 non-linear CR3BP EOMs
    [tout, xout] = ode113(@(t, state)CR3BP(state, mu), [0 V0(end)], V0(1:6),
options);

    % Final final variables using V0
    statef_V0 = xout(end,:);

    % Period is a free variable
    T = V0(end);

    % Initialize constraint vector norm
    F_norm(1) = norm(F(V0(:,1), statef_V0));

    % Matrix of all free variable vectors
    V(:,1) = V0;

    % While loop params
    counter = 1;
    counter_max = 50;

    % While loop to reduce F_norm
    while ((F_norm(counter) > TOL) && (counter < counter_max))
        phi0 = reshape(eye(6), [36, 1]); % Initial phi is identity
        state0 = [V(1:6,counter); phi0];
```

```

        % Propagate full state and STM
        [t_out, state_out] = ode113(@(t,state)CR3BP_full(state, mu), [0,
V(7,counter)], state0, options);
        statef = state_out(end, :);
        state_dot = CR3BP_full(statef, mu);
        F_i = F(state0, statef);
        DF_i = DF_mat(statef, state_dot);

        % Find V_i+1
        V(:,counter+1) = V(:,counter) - DF_i' * inv(DF_i * DF_i') * F_i;

        % Calculate F_norm and update counter
        F_norm(counter+1) = norm(F_i);
        counter = counter + 1;
    end

    if plot_input
        figure()
        plot([1:counter], F_norm, '-o', 'LineWidth', 2)
        yscale log
        grid on
        xlabel("Iterations")
        ylabel("F Norm")
        title("Constraint Vector Norm for each Iteration")
        hold on
        tol_yline = ones([counter,1])*TOL;

        plot([1:counter], tol_yline, 'red', 'LineWidth', 2)
        hold off
        legend("Norm", "Threshold")
    end

    print_out = sprintf('Difference in y_dot_0^2 and y_dot_f^2 - %d', V0(5)^2
- V(5,end)^2);
    disp(print_out)

    V_soln = V(:,end);
end

```

Not enough input arguments.

Error in gen_3d_periodic_orbit_single_shooting (line 13)
mu = system_params(1);

Published with MATLAB® R2024a