```
function I = ODEsim(Y0,tau,X)
% X(1) - r_2 value - m
% X(2) - \alpha_1 value - rad
% X(3) - \omega value - rad/s

[~,YP] = ode45(@(t,Y)chaos_subfn(t,Y,X),tau,Y0);

% YP(:,1) - holds \phi values
% YP(:,2) - holds d\phi/d\tau values

% d\phi/dt = ( d\phi/d\tau ) * ( d\tau/dt)
% \tau = 3 * X(3) * t
% d\tau/dt = 3* X(3)

dphi_dt = YP(:,2) * 3 * X(3);

% T - range of time (T = tau / ( 3 * X(3) ))
T = tau./ (3 * X(3));

% compute mean of dphi_dt - (1/T_end) * integral ( dphi_dt dt)
mean_dpdt = (1/T(end))*trapz(T,dphi_dt);

% squared difference calculation
sq_diff = (dphi_dt-mean_dpdt).^2;

% Final Integral calculation
I = sqrt((1/T(end))*trapz(T,sq_diff));


    function Ydot = chaos_subfn(t,Y,X)
        Q0 = 20;
        r1 = 4.3;
        g = 9.81;
        alpha0 = 0.036;

        omega = X(3);
        alpha1 = X(2);
        r2 = X(1);

        e = r1/(9*r2);
        Gamma = (1/(3*omega))*sqrt(g/r2);
        alpha = alpha0-alpha1*cos(t);
        beta = 3*alpha1*sin(t);

        Ydot(1) = Y(2);
        Ydot(2) = -(e-Gamma^2*alpha)*sin(Y(1)) - ...
            Gamma^2*beta*cos(Y(1))-(Gamma/Q0)*Y(2);
        Ydot = Ydot';
    end
end
```

Published with MATLAB® R2021a