Supplementary Material for “Higher-order Lie bracket approximation and averaging of control-affine systems with application to extremum seeking”

Sameer Pokhrel and Sameh A. Eisa

I. Introduction

This document serves to supplement the paper titled as ”Higher-order Lie bracket approximation and averaging of control-affine systems with application to extremum seeking”. Particulary, we provide MATLAB code and explain the procedure for Example 4. Furthermore, codes for Example 3 and 4 are provided in the github repository `https://github.com/MDCL-UC/Higher-Order-Lie-Brackets`.

II. Higher order Lie bracket system

From [1, Theorem 4], for a control-affine system given by

$$\dot{\boldsymbol{x}} = \boldsymbol{b_0}(\boldsymbol{x}) + \sum_{i=1}^{m} \omega^{p_i} \boldsymbol{b_i}(\boldsymbol{x}) u_i(k_i \omega t), \tag{1}$$

$(r-1)^{th}$ order Lie bracket system is given by r^{th} order averaging, expression for which is provided in (2)

$$\dot{\boldsymbol{z}} = \boldsymbol{b_0}(\boldsymbol{z}) + \sum_{i=1}^{r} \boldsymbol{L}_i(\boldsymbol{z}), \tag{2}$$

with

$$\boldsymbol{L}_1 = 0, \tag{3}$$

$$\boldsymbol{L}_2 = \sum_{j_1=1}^{m} \sum_{j_2=j_1+1}^{m} \nu_{j_1 j_2} [\boldsymbol{b}_{j_1}, \boldsymbol{b}_{j_2}], \tag{4}$$

$$\boldsymbol{L}_3 = \sum_{j_1=1}^{m} \sum_{j_2=j_1+1}^{m} \sum_{j_3=1}^{m} \nu_{j_1 j_2 j_3} [\boldsymbol{b}_{j_3}, [\boldsymbol{b}_{j_1}, \boldsymbol{b}_{j_2}]], \tag{5}$$

$$\begin{aligned}\boldsymbol{L}_4 = &\sum_{j_1=1}^{m} \sum_{j_2=j_1+1}^{m} \sum_{j_3=1}^{m} \sum_{j_4=j_3+1}^{m} \beta_{1_{j_1 j_2 j_3 j_4}} \Big[[\boldsymbol{b}_{j_1}, \boldsymbol{b}_{j_2}], [\boldsymbol{b}_{j_3}, \boldsymbol{b}_{j_4}]\Big] \\ &+ \sum_{j_1=1}^{m} \sum_{j_2=j_1+1}^{m} \sum_{j_3=1}^{m} \sum_{j_4=1}^{m} \beta_{2_{j_1 j_2 j_3 j_4}} [[[\boldsymbol{b}_{j_1}, \boldsymbol{b}_{j_2}], \boldsymbol{b}_{j_3}], \boldsymbol{b}_{j_4}]\end{aligned} \tag{6}$$

where

$$\begin{aligned}\nu_{j_1 j_2} = \frac{\omega^{p_{j_1}+p_{j_2}-1}}{2T} \int_0^{T'} \Big(u_{j_2}(k_{j_2}\tau) \int_0^{\tau} u_{j_1}(k_{j_1}p) \\ - u_{j_1}(k_{j_1}\tau) \int_0^{\tau} u_{j_2}(k_{j_2}p) \Big),\end{aligned} \tag{7}$$

$$\begin{aligned}\nu_{j_1 j_2 j_3} = \frac{\omega^{p_{j_1}+p_{j_2}+p_{j_3}-2}}{3T} \int_0^{T'} \int_0^{\tau} u_{j_3}(k_{j_3}s) ds \Big(u_{j_2}(k_{j_2}\tau) \\ \int_0^{\tau} u_{j_1}(k_{j_1}p) dp - u_{j_1}(k_{j_1}\tau) \int_0^{\tau} u_{j_2}(k_{j_2}p) dp \Big) d\tau,\end{aligned} \tag{8}$$

and

$$\beta_{1_{j_1j_2j_3j_4}} = \omega^{p_{j_1}+p_{j_2}+p_{j_3}+p_{j_4}-3}\frac{1}{12T}\int_0^{T'} \alpha_5(\tau)_{j_1j_2j_3j_4}$$
$$\beta_{2_{j_1j_2j_3j_4}} = \omega^{p_{j_1}+p_{j_2}+p_{j_3}+p_{j_4}-3}\frac{1}{12T}\int_0^{T'} \big(\alpha_8(\tau)_{j_1j_2j_3j_4} - \alpha_{10}(\tau)_{j_1j_2j_3j_4}\big),$$

with

$$\alpha_5(\tau)_{j_1j_2j_3j_4} = \int_0^{\tau}\bigg(\Big(u_{j_4}(\tau)u_{j_3}(p) - u_{j_3}(\tau)u_{j_4}(p)\Big)\Big(\int_0^{p}(u_{j_2}(q) \int_0^{q} u_{j_1}(k_{j_1}r) - u_{j_1}(q)\int_0^{q} u_{j_2}(k_{j_2}r))dq\Big)\bigg), \tag{9}$$

$$\alpha_8(\tau)_{j_1j_2j_3j_4} = \int_0^{\tau}\bigg(\int_0^{p}\Big(u_{j_2}(q)\int_0^{q} u_{j_1}(k_{j_1}r) - u_{j_1}(q)\int_0^{q} u_{j_2}(k_{j_2}r)\Big)dq u_{j_3}(p)\bigg)dp\; u_{j_4}(\tau), \tag{10}$$

$$\alpha_{10}(\tau)_{j_1j_2j_3j_4} = \int_0^{\tau}\bigg(\Big(u_{j_2}(p)\int_0^{p} u_{j_1}(k_{j_1}q) - u_{j_1}(p)\int_0^{p} u_{j_2}(k_{j_2}q)\Big) u_{j_3}(\tau)\int_0^{p} u_{j_4}(k_{j_4}q)\bigg)dp; \tag{11}$$

where p, q, r, τ are the variables of integration, and $T = (2\pi/\omega)LCM(k_1^{-1}, k_2^{-1}, ..., k_m^{-1})$. Readers are encouraged to refer to [1] for more information on symbols. Now, to find the higher order Lie bracket system, one needs to find $\boldsymbol{L}_2, \boldsymbol{L}_3, \boldsymbol{L}_4$. This involves calculating Lie brackets $[\boldsymbol{b}_{j_1}, \boldsymbol{b}_{j_2}], [\boldsymbol{b}_{j_3}, [\boldsymbol{b}_{j_1}, \boldsymbol{b}_{j_2}]], [[[\boldsymbol{b}_{j_1}, \boldsymbol{b}_{j_2}], \boldsymbol{b}_{j_3}], \boldsymbol{b}_{j_4}]$ and the parameters $\nu_{j_1j_2}, \nu_{j_1j_2j_3}, \beta_{1_{j_1j_2j_3j_4}}, \beta_{2_{j_1j_2j_3j_4}}$. In the next subsection, we provide methods and codes to calculate the Lie brackets and the parameters.

A. Calculation of Lie brackets

In this subsection, we explain the procedure and provide the codes to get Lie brackets corresponding to Example 4. One can follow a similar procedure for other examples also.

Example 4: The dynamics of the ESC system is given by

$$\dot{x} = \sqrt{\frac{1-e^{-J(x)}}{1+e^{J(x)}}}\Big(\omega^{p_1}sin(\psi)u_1 + \omega^{p_2}cos(\psi)u_2 + \omega^{p_3}sin(\psi)u_3 + \omega^{p_4}cos(\psi)u_4\Big), \tag{12}$$

for $J(x) \neq 0$ and $\dot{x} = 0$ for $J(x) = 0$ with $\psi = e^{J(x)} + 2ln(e^{J(x)} - 1)$ with $u_1 = cos(k_1\omega t)$, $u_2 = sin(k_2\omega t)$, $u_3 = cos(k_3\omega t)$, $u_4 = sin(k_4\omega t)$. We take $J = H(x-1)^4$. We choose $p_1 = p_3 = 0.99$, $p_2 = p_4 = 0.01$, $k_1 = k_2 = 1, k_3 = k_4 = 0.25$, $\omega = 100$, $H = 1/3$ and $x_0 = 2$. We use (2) to obtain a third-order LBS for the proposed ESC (12).

It is clear that for the ESC in (12) that $b_1 = b_3$ and $b_2 = b_4$ with $b_1 = \sqrt{\phi^*}\sin\psi, b_2 = \sqrt{\phi^*}\cos\psi$, where $\phi^* = (1-e^{-J(x)})/(1+e^{J(x)})$. Now, for L_2 corresponding to first-order Lie brackets (see (4)), we have $[b_1, b_3] = [b_2, b_4] = 0$ (since $b_1 = b_3$ and $b_2 = b_4$). So, $[b_1, b_2], [b_1, b_4], [b_2, b_3], [b_3, b_4]$ are the only relevant non-zero Lie brackets. Given the equality $b_1 = b_3$ and $b_2 = b_4$, all of these Lie brackets depend on $[b_1, b_2]$. Thus it is sufficient to calculate the Lie bracket $[b_1, b_2]$ and is provided in the attached code.

Now, for L_3 corresponding to second-order Lie brackets (see (5)), Lie brackets having $[b_1, b_3]$ or $[b_2, b_4]$ in them will be zero. Given the equality $b_1 = b_3$ and $b_2 = b_4$, the only relevant independent Lie brackets are $[b_1, [b_1, b_2]]$ and $[b_2, [b_1, b_2]]$. These Lie brackets are calculated using the attached code.

For L_4 corresponding to third-order Lie brackets (see (6)), due to the relation $b_1 = b_3$ and $b_2 = b_4$, one will find that the Lie brackets of the form $[[b_{j_1}, b_{j_2}], [b_{j_3}, b_{j_4}]]$ are zeros and the non-zero relevant independent Lie brackets are $[[[b_1, b_2], b_1], b_1]$, $[[[b_1, b_2], b_1], b_2]$, $[[[b_1, b_2], b_2], b_1]$ and $[[[b_1, b_2], b_2], b_2]$. The attached code provides these Lie brackets.

```
clc
clear
close all
syms H x real

% State
```

```
X = [x];

% Cost function and vector fields
J = H*(x-1)^4;
phi = (1-exp(-(J)))/(1+exp(J));
psi = exp((J))+2*log(exp(J)-1);
f1 = sqrt(phi)*sin(psi);
f2 = sqrt(phi)*cos(psi);
f3 = sqrt(phi)*sin(psi);
f4 = sqrt(phi)*cos(psi);


%___________ Generate first order Lie brackets
f1_f2 = generateLieBracket(f1,f2,X)


%___________ Generate second order Lie brackets
% [[f1,f2],f1]
f1f2_f1 = generateLieBracket(f1_f2,f1,X)

% [[f1,f2],f2]
f1f2_f2 = generateLieBracket(f1_f2,f2,X)

%_____________Third order Lie brackets

% 1211
% In the form [[f1,f2],[f1,f1]] is zero
% In the form [[[f1,f2],f1],f1]
f1f2_f1_f1 = generateLieBracket(f1f2_f1,f1,X)

% 1212
% In the form [[f1,f2],[f1,f2]] is zero
% In the form [[[f1,f2],f1],f2]
f1f2_f1_f2 = generateLieBracket(f1f2_f1,f2,X)

% 1221
% In the form [[f1,f2],[f2,f1]] is zero
% In the form [[[f1,f2],f2],f1]
f1f2_f2_f1 = generateLieBracket(f1f2_f2,f1,X)

% 1222
% In the form [[f1,f2],[f2,f2]] is zero
% In the form [[[f1,f2],f2],f2]
f1f2_f2_f2 = generateLieBracket(f1f2_f2,f2,X)


function lieBracket = generateLieBracket(b1, b2, X)
    % Inputs:
    %   - b1, b2: Vector fields (symbolic expressions) representing the dynamics.
    %   - X: Symbolic vector representing the state variables (e.g., [x1; x2; x3]).

    % Calculate the Lie derivatives of b1 and b2 with respect to each state variable
    diff_b1 = jacobian(b1, X);
    diff_b2 = jacobian(b2, X);

    % Compute the Lie bracket: b1b2 = diff_b2 * b1 - diff_b1 * b2
    lieBracket = simplify(diff_b2 * b1 - diff_b1 * b2);
end
```

B. Calculation of the parameters

In this subsection, we explain the procedure and provide the codes to get the parameters $\nu_{j_1j_2}, \nu_{j_1j_2j_3}, \beta 1_{j_1j_2j_3j_4}, \beta 2_{j_1j_2j_3j_4}$ corresponding to Example 4. One can follow a similar procedure for other examples also.

For L_2, only the parameters $\nu_{12}, \nu_{14}, \nu_{23}, \nu_{34}$ corresponding to the non-zero Lie brackets are relevant. Furthermore, it was found that only ν_{12} and ν_{34} have non-zero values given by $\nu_{12} = 0.5$ and $\nu_{34} = 2$, both of which have $p_{j_1} + p_{j_2} = 1$ yielding L_2 bounded as $\omega \to \infty$.

For L_3, There are 16 relevant $\nu_{j_1j_2j_3}$ (corresponding to non-zero Lie brackets) given by

$$\begin{array}{cccccccc} \boldsymbol{\nu_{121}}, & \nu_{122}, & \boldsymbol{\nu_{123}}, & \nu_{124}, & \boldsymbol{\nu_{141}}, & \nu_{142}, & \boldsymbol{\nu_{143}}, & \nu_{144}, \\ \boldsymbol{\nu_{231}}, & \nu_{232}, & \boldsymbol{\nu_{233}}, & \nu_{234}, & \boldsymbol{\nu_{341}}, & \nu_{342}, & \boldsymbol{\nu_{343}}, & \nu_{344}, \end{array} \tag{13}$$

where bold $\nu_{j_1j_2j_3}$ correspond to those with $p_{j_1} + p_{j_2} + p_{j_3} \approx 2$. However, the iterated integrals for said $\nu_{j_1j_2j_3}$ were found to be zero. Thus, only $\nu_{j_1j_2j_3}$ with $p_{j_1} + p_{j_2} + p_{j_3} \approx 1$ (i.e. with power of ω approximately -1 in (8)) are the only ones with non-zero values, thus making L_3 bounded as $\omega \to \infty$, but less significant on the higher-order LBS (i.e., L_3 almost vanish for large ω).

For L_4, since Lie brackets multiplied with $\beta 1_{j_1j_2j_3j_4}$ (see (6)) are all zeros, $\beta 1_{j_1j_2j_3j_4}$ don't need to be computed and there are only 64 relevant $\beta 2_{j_1j_2j_3j_4}$ that need to be computed. These are

$$\begin{array}{cccccccc} \beta 2_{1211} & \beta 2_{1212} & \beta 2_{1213} & \beta 2_{1214} & \beta 2_{1221} & \beta 2_{1222} & \beta 2_{1223} & \beta 2_{1224} \\ \beta 2_{1231} & \beta 2_{1232} & \beta 2_{1233} & \beta 2_{1234} & \beta 2_{1241} & \beta 2_{1242} & \beta 2_{1243} & \beta 2_{1244} \\ \beta 2_{1411} & \beta 2_{1412} & \beta 2_{1413} & \beta 2_{1414} & \beta 2_{1421} & \beta 2_{1422} & \beta 2_{1423} & \beta 2_{1424} \\ \beta 2_{1431} & \beta 2_{1432} & \beta 2_{1433} & \beta 2_{1434} & \beta 2_{1441} & \beta 2_{1442} & \beta 2_{1443} & \beta 2_{1444} \\ \beta 2_{2311} & \beta 2_{2312} & \beta 2_{2313} & \beta 2_{2314} & \beta 2_{2321} & \beta 2_{2322} & \beta 2_{2323} & \beta 2_{2324} \\ \beta 2_{2331} & \beta 2_{2332} & \beta 2_{2333} & \beta 2_{2334} & \beta 2_{2341} & \beta 2_{2342} & \beta 2_{2343} & \beta 2_{2344} \\ \beta 2_{3411} & \beta 2_{3412} & \beta 2_{3413} & \beta 2_{3414} & \beta 2_{3421} & \beta 2_{3422} & \beta 2_{3423} & \beta 2_{3424} \\ \beta 2_{3431} & \beta 2_{3432} & \beta 2_{3433} & \beta 2_{3434} & \beta 2_{3441} & \beta 2_{3442} & \beta 2_{3443} & \beta 2_{3444} \end{array} \tag{14}$$

It is found that $\beta 2_{j_1j_2j_3j_4}$ corresponding to those with $p_{j_1} + p_{j_2} + p_{j_3} + p_{j_4} \approx 3$ do not all have zero iterated integrals, thus making them significantly important in the higher order Lie bracket system. The code attached below provides functions to calculate the parameters.

Code containing all parameters is ”param.m”.

```
k1 = 1;
k2 = 1;
k3 = 1/4;
k4 = 1/4;
k = [k1,k2,k3,k4];


p1 = 0.99 ;
p2 = 0.01 ;
p3 = 0.99 ;
p4 = 0.01;
p_val = [p1,p2,p3,p4]; % Write in order

w_val = 100;

u1 = @(x) cos(x);
u2 = @(x) sin(x);
u3 = @(x) cos(x);
u4 = @(x) sin(x);

% Indices for nu_ij
index_all2 = [12,14,23,34];

% Indices for nu_ijk
index_all3 = [121,122,123,124,...
```

```
    141,142,143,144,...
    231,232,233,234,...
    341,342,343,344];
```

Code to find $\nu_{j_1 j_2}$ and $\nu_{j_1 j_2 j_3}$. This provides output $nu\_val2.mat$ and $nu\_val3.mat$.

```
function Find_nu


syms p s w tau
params


u1_p = u1(k1 * p);
u2_p = u2(k2 * p);
u3_p = u3(k3 * p);
u4_p = u4(k4 * p);
u_p = [u1_p, u2_p, u3_p, u4_p];

u1_tau = u1(k1 * tau);
u2_tau = u2(k2 * tau);
u3_tau = u3(k3 * tau);
u4_tau = u4(k4 * tau);
u_tau = [u1_tau, u2_tau, u3_tau, u4_tau];


% Find the integral of u_p
U1 = int(u1_p, p, 0, tau);
U2 = int(u2_p, p, 0, tau);
U3 = int(u3_p, p, 0, tau);
U4 = int(u4_p, p, 0, tau);
U = [U1, U2, U3, U4];



nu2 = dictionary();
for i=1:length(index_all2)
    index = index_all2(i);
    nu2(index) = find_nu_val(index,k,p_val,u_tau,U,tau,w,w_val);
end
% nu_val2 = double(subs(values(nu2),w,w_val));
nu_val2 = nu2;
save("nu_val2","nu_val2")


nu3 = dictionary();
for i=1:length(index_all3)
    index = index_all3(i);
    nu3(index) = find_nu_val(index,k,p_val,u_tau,U,tau,w,w_val);
end
% nu_val3 = double(subs(values(nu3),w,w_val));
nu_val3 = nu3;
save("nu_val3","nu_val3")

end



function nu = find_nu_val(index,k,p_val,u_tau,U,tau,w,w_val)
```

```
switch 1
    case index>10 & index<100 % 2 digit
        k2_pos = rem(index,10);
        k1_pos = (index-k2_pos)/10;

        k1 = k(k1_pos);
        k2 = k(k2_pos);
        U1 = U(k1_pos);
        U2 = U(k2_pos);
        u1_tau = u_tau(k1_pos);
        u2_tau = u_tau(k2_pos);
        int_val = calculate_int_second_order(k1, k2, U1, U2, u1_tau, u2_tau, tau);
        nu = w_val^calculate_sump(p_val,[k1_pos,k2_pos])*int_val;

    case index>100 && index<1000 % 3 digit
        k3_pos = rem(index,10);
        first2terms = (index-k3_pos)/10;
        k2_pos = rem(first2terms,10);
        k1_pos = (first2terms-k2_pos)/10;

        k1 = k(k1_pos);
        k2 = k(k2_pos);
        k3 = k(k3_pos);
        U1 = U(k1_pos);
        U2 = U(k2_pos);
        U3 = U(k3_pos);
        u1_tau = u_tau(k1_pos);
        u2_tau = u_tau(k2_pos);
        u3_tau = u_tau(k3_pos);
        int_val = calculate_int_third_order(k1, k2, k3, U1, U2, U3, u1_tau, u2_tau,
           u3_tau, tau);
        nu = w_val^calculate_sump(p_val,[k1_pos,k2_pos,k3_pos])*int_val;
end
end



%%  Necessary functions
function integral = calculate_int_second_order(k1, k2, U1, U2, u1_tau, u2_tau, tau)
    if k1 <= 1 && k2 <= 1
        m = lcm(1 / k1, 1 / k2);
    else
        m = lcm(ceil(1 / k1), ceil(1 / k2));
    end
    T = 2 * pi * m;
    beta = (u2_tau * U1) - (u1_tau * U2);
    integral = eval(1 / (2 * T) * int(beta, tau, 0, T));
end

function integral = calculate_int_third_order(k1, k2,k3, U1, U2, U3, u1_tau, u2_tau,
   u3_tau,tau)
    if k1<=1 && k2<=1 && k3<=1
        m = lcm(1/k1,lcm(1/k2,1/k3));
    else
        m=lcm(ceil(1/k1),lcm(ceil(1/k2),ceil(1/k3)));
    end
    T = 2 * pi * m;
    beta = (u2_tau * U1) - (u1_tau * U2);
```

```
    integral = eval(1 / (3 * T) * int(beta*U3, tau, 0, T));
end


function sump = calculate_sump(p,m)
n = length(m); % Number of elements in p

switch n
    case 2
        i = m(1);
        j = m(2);
        sump = p(i) + p(j) - 1;
    case 3
        i = m(1);
        j = m(2);
        k = m(3);
        sump = p(i) + p(j) + p(k) - 2;

     case 4
        i = m(1);
        j = m(2);
        k = m(3);
        l = m(4);
        sump = p(i) + p(j) + p(k) +p(l) - 3;
    otherwise
        error('Unsupported number of elements in p or incorrect number of indices.');
end
end
```

Code to find parameters $\beta_{2j_1j_2j_3j_4}$ are attached below

```
clc
clear

params

% Betas
beta2 = beta2_fun(k,p_val, w_val);

beta3 = beta3_fun(k,p_val, w_val);

beta_val = beta_fun(beta2,beta3);

function out =beta_fun(beta2,beta3)

%% Betas
beta = dictionary();
beta(1211) = beta2(1211) - beta3(1211);
beta(1212) = beta2(1212) - beta3(1212);
beta(1213) = beta2(1213) - beta3(1213);
beta(1214) =  beta2(1214) - beta3(1214);

beta(1221) = beta2(1221) - beta3(1221);
beta(1222) = beta2(1222) - beta3(1222);
beta(1223) = beta2(1223) - beta3(1223);
beta(1224) = beta2(1224) - beta3(1224);

beta(1231) = beta2(1231) - beta3(1231);
beta(1232) = beta2(1232) - beta3(1232);
```

```
beta(1233) = beta2(1233) - beta3(1233);
beta(1234) = beta2(1234) - beta3(1234);

beta(1241) = beta2(1241) - beta3(1241);
beta(1242) = beta2(1242) - beta3(1242);
beta(1243) = beta2(1243) - beta3(1243);
beta(1244) = beta2(1244) - beta3(1244);

beta(1411) = beta2(1411) - beta3(1411);
beta(1412) = beta2(1412) - beta3(1412);
beta(1413) = beta2(1413) - beta3(1413);
beta(1414) = beta2(1414) - beta3(1414);

beta(1421) = beta2(1421) - beta3(1421);
beta(1422) = beta2(1422) - beta3(1422);
beta(1423) = beta2(1423) - beta3(1423);
beta(1424) = beta2(1424) - beta3(1424);

beta(1431) = beta2(1431) - beta3(1431);
beta(1432) = beta2(1432) - beta3(1432);
beta(1433) = beta2(1433) - beta3(1433);
beta(1434) = beta2(1434) - beta3(1434);

beta(1441) = beta2(1441) - beta3(1441);
beta(1442) = beta2(1442) - beta3(1442);
beta(1443) = beta2(1443) - beta3(1443);
beta(1444) = beta2(1444) - beta3(1444);

%
beta(2311) = beta2(2311) - beta3(2311);
beta(2312) = beta2(2312) - beta3(2312);
beta(2313) = beta2(2313) - beta3(2313);
beta(2314) = beta2(2314) - beta3(2314);

beta(2321) = beta2(2321) - beta3(2321);
beta(2322) = beta2(2322) - beta3(2322);
beta(2323) = beta2(2323) - beta3(2323);
beta(2324) = beta2(2324) - beta3(2324);

beta(2331) = beta2(2331) - beta3(2331);
beta(2332) = beta2(2332) - beta3(2332);
beta(2333) = beta2(2333) - beta3(2333);
beta(2334) = beta2(2334) - beta3(2334);

beta(2341) = beta2(2341) - beta3(2341);
beta(2342) = beta2(2342) - beta3(2342);
beta(2343) = beta2(2343) - beta3(2343);
beta(2344) = beta2(2344) - beta3(2344);
%

beta(3411) = beta2(3411) - beta3(3411);
beta(3412) = beta2(3412) - beta3(3412);
beta(3413) = beta2(3413) - beta3(3413);
beta(3414) = beta2(3414) - beta3(3414);

beta(3421) = beta2(3421) - beta3(3421);
beta(3422) = beta2(3422) - beta3(3422);
beta(3423) = beta2(3423) - beta3(3423);
```

```
beta(3424) = beta2(3424) - beta3(3424);

beta(3431) = beta2(3431) - beta3(3431);
beta(3432) = beta2(3432) - beta3(3432);
beta(3433) = beta2(3433) - beta3(3433);
beta(3434) = beta2(3434) - beta3(3434);

beta(3441) = beta2(3441) - beta3(3441);
beta(3442) = beta2(3442) - beta3(3442);
beta(3443) = beta2(3443) - beta3(3443);
beta(3444) = beta2(3444) - beta3(3444);

%% Output
out = beta;
beta_val = out;
save("beta_val","beta_val")

end


function out = beta2_fun(k,p_val, w)

%% Parameters
k1 = k(1);
k2 = k(2);
k3 = k(3);
k4 = k(4);

%% Control inputs
syms p q r tau

% u
u1_r = cos(k1*r);
u2_r = sin(k2*r);
u3_r = cos(k3*r);
u4_r = sin(k4*r);

u1_q = cos(k1*q);
u2_q = sin(k2*q);
u3_q = cos(k3*q);
u4_q = sin(k4*q);

u1_p = cos(k1*p);
u2_p = sin(k2*p);
u3_p = cos(k3*p);
u4_p = sin(k4*p);

u1_tau = cos(k1*tau);
u2_tau = sin(k2*tau);
u3_tau = cos(k3*tau);
u4_tau = sin(k4*tau);


% U
U1_q = int(u1_r,0,q);
U2_q = int(u2_r,0,q);
U3_q = int(u3_r,0,q);
U4_q = int(u4_r,0,q);
```

```
%% Alphas
% Alpha1
alpha1_12 =  u2_q*U1_q-u1_q*U2_q;
alpha1_13 =  u3_q*U1_q-u1_q*U3_q;
alpha1_14 =  u4_q*U1_q-u1_q*U4_q;
alpha1_23 =  u3_q*U2_q-u2_q*U3_q;
alpha1_24 =  u4_q*U2_q-u2_q*U4_q;
alpha1_34 =  u4_q*U3_q-u3_q*U4_q;

% Alpha2
alpha2_12 = int(alpha1_12,q,0,p);
alpha2_13 = int(alpha1_13,q,0,p);
alpha2_14 = int(alpha1_14,q,0,p);
alpha2_23 = int(alpha1_23,q,0,p);
alpha2_24 = int(alpha1_24,q,0,p);
alpha2_34 = int(alpha1_34,q,0,p);

% Alpha 6
alpha6_121 = alpha2_12*u1_p;
alpha6_122 = alpha2_12*u2_p;
alpha6_123 = alpha2_12*u3_p;
alpha6_124 = alpha2_12*u4_p;

alpha6_131 = alpha2_13*u1_p;
alpha6_132 = alpha2_13*u2_p;
alpha6_133 = alpha2_13*u3_p;
alpha6_134 = alpha2_13*u4_p;

alpha6_141 = alpha2_14*u1_p;
alpha6_142 = alpha2_14*u2_p;
alpha6_143 = alpha2_14*u3_p;
alpha6_144 = alpha2_14*u4_p;

alpha6_231 = alpha2_23*u1_p;
alpha6_232 = alpha2_23*u2_p;
alpha6_233 = alpha2_23*u3_p;
alpha6_234 = alpha2_23*u4_p;

alpha6_341 = alpha2_34*u1_p;
alpha6_342 = alpha2_34*u2_p;
alpha6_343 = alpha2_34*u3_p;
alpha6_344 = alpha2_34*u4_p;

% Alpha 7
alpha7_121 = int(alpha6_121,p,0,tau);
alpha7_122 = int(alpha6_122,p,0,tau);
alpha7_123 = int(alpha6_123,p,0,tau);
alpha7_124 = int(alpha6_124,p,0,tau);

alpha7_131 = int(alpha6_131,p,0,tau);
alpha7_132 = int(alpha6_132,p,0,tau);
alpha7_133 = int(alpha6_133,p,0,tau);
alpha7_134 = int(alpha6_134,p,0,tau);

alpha7_141 = int(alpha6_141,p,0,tau);
```

```
alpha7_142 = int(alpha6_142,p,0,tau);
alpha7_143 = int(alpha6_143,p,0,tau);
alpha7_144 = int(alpha6_144,p,0,tau);

alpha7_231 = int(alpha6_231,p,0,tau);
alpha7_232 = int(alpha6_232,p,0,tau);
alpha7_233 = int(alpha6_233,p,0,tau);
alpha7_234 = int(alpha6_234,p,0,tau);

alpha7_341 = int(alpha6_341,p,0,tau);
alpha7_342 = int(alpha6_342,p,0,tau);
alpha7_343 = int(alpha6_343,p,0,tau);
alpha7_344 = int(alpha6_344,p,0,tau);

%% Alpha 8
% Alpha 8
% 12
alpha8_1211 = alpha7_121*u1_tau;
alpha8_1212 = alpha7_121*u2_tau;
alpha8_1213 = alpha7_121*u3_tau;
alpha8_1214 = alpha7_121*u4_tau;

alpha8_1221 = alpha7_122*u1_tau;
alpha8_1222 = alpha7_122*u2_tau;
alpha8_1223 = alpha7_122*u3_tau;
alpha8_1224 = alpha7_122*u4_tau;

alpha8_1231 = alpha7_123*u1_tau;
alpha8_1232 = alpha7_123*u2_tau;
alpha8_1233 = alpha7_123*u3_tau;
alpha8_1234 = alpha7_123*u4_tau;

alpha8_1241 = alpha7_124*u1_tau;
alpha8_1242 = alpha7_124*u2_tau;
alpha8_1243 = alpha7_124*u3_tau;
alpha8_1244 = alpha7_124*u4_tau;

% 13
alpha8_1311 = alpha7_131*u1_tau;
alpha8_1312 = alpha7_131*u2_tau;
alpha8_1313 = alpha7_131*u3_tau;
alpha8_1314 = alpha7_131*u4_tau;


alpha8_1321 = alpha7_132*u1_tau;
alpha8_1322 = alpha7_132*u2_tau;
alpha8_1323 = alpha7_132*u3_tau;
alpha8_1324 = alpha7_132*u4_tau;


alpha8_1331 = alpha7_133*u1_tau;
alpha8_1332 = alpha7_133*u2_tau;
alpha8_1333 = alpha7_133*u3_tau;
alpha8_1334 = alpha7_133*u4_tau;

alpha8_1341 = alpha7_134*u1_tau;
alpha8_1342 = alpha7_134*u2_tau;
alpha8_1343 = alpha7_134*u3_tau;
```

```
alpha8_1344 = alpha7_134*u4_tau;

% 14
alpha8_1411 = alpha7_141*u1_tau;
alpha8_1412 = alpha7_141*u2_tau;
alpha8_1413 = alpha7_141*u3_tau;
alpha8_1414 = alpha7_141*u4_tau;


alpha8_1421 = alpha7_142*u1_tau;
alpha8_1422 = alpha7_142*u2_tau;
alpha8_1423 = alpha7_142*u3_tau;
alpha8_1424 = alpha7_142*u4_tau;


alpha8_1431 = alpha7_143*u1_tau;
alpha8_1432 = alpha7_143*u2_tau;
alpha8_1433 = alpha7_143*u3_tau;
alpha8_1434 = alpha7_143*u4_tau;

alpha8_1441 = alpha7_144*u1_tau;
alpha8_1442 = alpha7_144*u2_tau;
alpha8_1443 = alpha7_144*u3_tau;
alpha8_1444 = alpha7_144*u4_tau;


% 23
alpha8_2311 = alpha7_231*u1_tau;
alpha8_2312 = alpha7_231*u2_tau;
alpha8_2313 = alpha7_231*u3_tau;
alpha8_2314 = alpha7_231*u4_tau;


alpha8_2321 = alpha7_232*u1_tau;
alpha8_2322 = alpha7_232*u2_tau;
alpha8_2323 = alpha7_232*u3_tau;
alpha8_2324 = alpha7_232*u4_tau;


alpha8_2331 = alpha7_233*u1_tau;
alpha8_2332 = alpha7_233*u2_tau;
alpha8_2333 = alpha7_233*u3_tau;
alpha8_2334 = alpha7_233*u4_tau;

alpha8_2341 = alpha7_234*u1_tau;
alpha8_2342 = alpha7_234*u2_tau;
alpha8_2343 = alpha7_234*u3_tau;
alpha8_2344 = alpha7_234*u4_tau;

% 34
alpha8_3411 = alpha7_341*u1_tau;
alpha8_3412 = alpha7_341*u2_tau;
alpha8_3413 = alpha7_341*u3_tau;
alpha8_3414 = alpha7_341*u4_tau;


alpha8_3421 = alpha7_342*u1_tau;
alpha8_3422 = alpha7_342*u2_tau;
```

```
alpha8_3423 = alpha7_342*u3_tau;
alpha8_3424 = alpha7_342*u4_tau;


alpha8_3431 = alpha7_343*u1_tau;
alpha8_3432 = alpha7_343*u2_tau;
alpha8_3433 = alpha7_343*u3_tau;
alpha8_3434 = alpha7_343*u4_tau;

alpha8_3441 = alpha7_344*u1_tau;
alpha8_3442 = alpha7_344*u2_tau;
alpha8_3443 = alpha7_344*u3_tau;
alpha8_3444 = alpha7_344*u4_tau;


%% Betas
%12
beta2 = dictionary();
beta2(1211) = calculate_beta(w,k,p_val,[1,2,1,1],alpha8_1211);
beta2(1212) = calculate_beta(w,k,p_val,[1,2,1,2],alpha8_1212);
beta2(1213) = calculate_beta(w,k,p_val,[1,2,1,3],alpha8_1213);
beta2(1214) = calculate_beta(w,k,p_val,[1,2,1,4],alpha8_1214);

beta2(1221) = calculate_beta(w,k,p_val,[1,2,2,1],alpha8_1221);
beta2(1222) = calculate_beta(w,k,p_val,[1,2,2,2],alpha8_1222);
beta2(1223) = calculate_beta(w,k,p_val,[1,2,2,3],alpha8_1223);
beta2(1224) = calculate_beta(w,k,p_val,[1,2,2,4],alpha8_1224);

beta2(1231) = calculate_beta(w,k,p_val,[1,2,3,1],alpha8_1231);
beta2(1232) = calculate_beta(w,k,p_val,[1,2,3,2],alpha8_1232);
beta2(1233) = calculate_beta(w,k,p_val,[1,2,3,3],alpha8_1233);
beta2(1234) = calculate_beta(w,k,p_val,[1,2,3,4],alpha8_1234);

beta2(1241) = calculate_beta(w,k,p_val,[1,2,4,1],alpha8_1241);
beta2(1242) = calculate_beta(w,k,p_val,[1,2,4,2],alpha8_1242);
beta2(1243) = calculate_beta(w,k,p_val,[1,2,4,3],alpha8_1243);
beta2(1244) = calculate_beta(w,k,p_val,[1,2,4,4],alpha8_1244);


%14
beta2(1411) = calculate_beta(w,k,p_val,[1,4,1,1],alpha8_1411);
beta2(1412) = calculate_beta(w,k,p_val,[1,4,1,2],alpha8_1412);
beta2(1413) = calculate_beta(w,k,p_val,[1,4,1,3],alpha8_1413);
beta2(1414) = calculate_beta(w,k,p_val,[1,4,1,4],alpha8_1414);

beta2(1421) = calculate_beta(w,k,p_val,[1,4,2,1],alpha8_1421);
beta2(1422) = calculate_beta(w,k,p_val,[1,4,2,2],alpha8_1422);
beta2(1423) = calculate_beta(w,k,p_val,[1,4,2,3],alpha8_1423);
beta2(1424) = calculate_beta(w,k,p_val,[1,4,2,4],alpha8_1424);

beta2(1431) = calculate_beta(w,k,p_val,[1,4,3,1],alpha8_1431);
beta2(1432) = calculate_beta(w,k,p_val,[1,4,3,2],alpha8_1432);
beta2(1433) = calculate_beta(w,k,p_val,[1,4,3,3],alpha8_1433);
beta2(1434) = calculate_beta(w,k,p_val,[1,4,3,4],alpha8_1434);

beta2(1441) = calculate_beta(w,k,p_val,[1,4,4,1],alpha8_1441);
beta2(1442) = calculate_beta(w,k,p_val,[1,4,4,2],alpha8_1442);
beta2(1443) = calculate_beta(w,k,p_val,[1,4,4,3],alpha8_1443);
```

```
beta2(1444) = calculate_beta(w,k,p_val,[1,4,4,4],alpha8_1444);

%23
beta2(2311) = calculate_beta(w,k,p_val,[2,3,1,1],alpha8_2311);
beta2(2312) = calculate_beta(w,k,p_val,[2,3,1,2],alpha8_2312);
beta2(2313) = calculate_beta(w,k,p_val,[2,3,1,3],alpha8_2313);
beta2(2314) = calculate_beta(w,k,p_val,[2,3,1,4],alpha8_2314);

beta2(2321) = calculate_beta(w,k,p_val,[2,3,2,1],alpha8_2321);
beta2(2322) = calculate_beta(w,k,p_val,[2,3,2,2],alpha8_2322);
beta2(2323) = calculate_beta(w,k,p_val,[2,3,2,3],alpha8_2323);
beta2(2324) = calculate_beta(w,k,p_val,[2,3,2,4],alpha8_2324);

beta2(2331) = calculate_beta(w,k,p_val,[2,3,3,1],alpha8_2331);
beta2(2332) = calculate_beta(w,k,p_val,[2,3,3,2],alpha8_2332);
beta2(2333) = calculate_beta(w,k,p_val,[2,3,3,3],alpha8_2333);
beta2(2334) = calculate_beta(w,k,p_val,[2,3,3,4],alpha8_2334);

beta2(2341) = calculate_beta(w,k,p_val,[2,3,4,1],alpha8_2341);
beta2(2342) = calculate_beta(w,k,p_val,[2,3,4,2],alpha8_2342);
beta2(2343) = calculate_beta(w,k,p_val,[2,3,4,3],alpha8_2343);
beta2(2344) = calculate_beta(w,k,p_val,[2,3,4,4],alpha8_2344);

%34
beta2(3411) = calculate_beta(w,k,p_val,[3,4,1,1],alpha8_3411);
beta2(3412) = calculate_beta(w,k,p_val,[3,4,1,2],alpha8_3412);
beta2(3413) = calculate_beta(w,k,p_val,[3,4,1,3],alpha8_3413);
beta2(3414) = calculate_beta(w,k,p_val,[3,4,1,4],alpha8_3414);

beta2(3421) = calculate_beta(w,k,p_val,[3,4,2,1],alpha8_3421);
beta2(3422) = calculate_beta(w,k,p_val,[3,4,2,2],alpha8_3422);
beta2(3423) = calculate_beta(w,k,p_val,[3,4,2,3],alpha8_3423);
beta2(3424) = calculate_beta(w,k,p_val,[3,4,2,4],alpha8_3424);

beta2(3431) = calculate_beta(w,k,p_val,[3,4,3,1],alpha8_3431);
beta2(3432) = calculate_beta(w,k,p_val,[3,4,3,2],alpha8_3432);
beta2(3433) = calculate_beta(w,k,p_val,[3,4,3,3],alpha8_3433);
beta2(3434) = calculate_beta(w,k,p_val,[3,4,3,4],alpha8_3434);

beta2(3441) = calculate_beta(w,k,p_val,[3,4,4,1],alpha8_3441);
beta2(3442) = calculate_beta(w,k,p_val,[3,4,4,2],alpha8_3442);
beta2(3443) = calculate_beta(w,k,p_val,[3,4,4,3],alpha8_3443);
beta2(3444) = calculate_beta(w,k,p_val,[3,4,4,4],alpha8_3444);

%% Output
out = beta2;
save("beta2","beta2");

end


function out = beta3_fun(k,p_val, w)

k1 = k(1);
k2 = k(2);
k3 = k(3);
k4 = k(4);
```

```
%% Control inputs
syms p q r tau

u1_r = cos(k1*r);
u2_r = sin(k2*r);
u3_r = cos(k3*r);
u4_r = sin(k4*r);

u1_q = cos(k1*q);
u2_q = sin(k2*q);
u3_q = cos(k3*q);
u4_q = sin(k4*q);

u1_p = cos(k1*p);
u2_p = sin(k2*p);
u3_p = cos(k3*p);
u4_p = sin(k4*p);

u1_tau = cos(k1*tau);
u2_tau = sin(k2*tau);
u3_tau = cos(k3*tau);
u4_tau = sin(k4*tau);

% U
U1_p = int(u1_q,0,p);
U2_p = int(u2_q,0,p);
U3_p = int(u3_q,0,p);
U4_p = int(u4_q,0,p);
%% Alphas
% Alpha 1
alpha1_12 =  u2_p*U1_p-u1_p*U2_p;
alpha1_13 =  u3_p*U1_p-u1_p*U3_p;
alpha1_14 =  u4_p*U1_p-u1_p*U4_p;
alpha1_23 =  u3_p*U2_p-u2_p*U3_p;
alpha1_24 =  u4_p*U2_p-u2_p*U4_p;
alpha1_34 =  u4_p*U3_p-u3_p*U4_p;


% Alpha 9
alpha9_121 = alpha1_12*u1_tau;
alpha9_122 = alpha1_12*u2_tau;
alpha9_123 = alpha1_12*u3_tau;
alpha9_124 = alpha1_12*u4_tau;

alpha9_131 = alpha1_13*u1_tau;
alpha9_132 = alpha1_13*u2_tau;
alpha9_133 = alpha1_13*u3_tau;
alpha9_134 = alpha1_13*u4_tau;

alpha9_141 = alpha1_14*u1_tau;
alpha9_142 = alpha1_14*u2_tau;
alpha9_143 = alpha1_14*u3_tau;
alpha9_144 = alpha1_14*u4_tau;

alpha9_231 = alpha1_23*u1_tau;
alpha9_232 = alpha1_23*u2_tau;
alpha9_233 = alpha1_23*u3_tau;
alpha9_234 = alpha1_23*u4_tau;
```

```
alpha9_341 = alpha1_34*u1_tau;
alpha9_342 = alpha1_34*u2_tau;
alpha9_343 = alpha1_34*u3_tau;
alpha9_344 = alpha1_34*u4_tau;

% Alpha 10
alpha10_1211 = int(alpha9_121*U1_p,p,0,tau);
alpha10_1212 = int(alpha9_121*U2_p,p,0,tau);
alpha10_1213 = int(alpha9_121*U3_p,p,0,tau);
alpha10_1214 = int(alpha9_121*U4_p,p,0,tau);

alpha10_1221 = int(alpha9_122*U1_p,p,0,tau);
alpha10_1222 = int(alpha9_122*U2_p,p,0,tau);
alpha10_1223 = int(alpha9_122*U3_p,p,0,tau);
alpha10_1224 = int(alpha9_122*U4_p,p,0,tau);

alpha10_1231 = int(alpha9_123*U1_p,p,0,tau);
alpha10_1232 = int(alpha9_123*U2_p,p,0,tau);
alpha10_1233 = int(alpha9_123*U3_p,p,0,tau);
alpha10_1234 = int(alpha9_123*U4_p,p,0,tau);

alpha10_1241 = int(alpha9_124*U1_p,p,0,tau);
alpha10_1242 = int(alpha9_124*U2_p,p,0,tau);
alpha10_1243 = int(alpha9_124*U3_p,p,0,tau);
alpha10_1244 = int(alpha9_124*U4_p,p,0,tau);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alpha10_1311 = int(alpha9_131*U1_p,p,0,tau);
alpha10_1312 = int(alpha9_131*U2_p,p,0,tau);
alpha10_1313 = int(alpha9_131*U3_p,p,0,tau);
alpha10_1314 = int(alpha9_131*U4_p,p,0,tau);

alpha10_1321 = int(alpha9_132*U1_p,p,0,tau);
alpha10_1322 = int(alpha9_132*U2_p,p,0,tau);
alpha10_1323 = int(alpha9_132*U3_p,p,0,tau);
alpha10_1324 = int(alpha9_132*U4_p,p,0,tau);

alpha10_1331 = int(alpha9_133*U1_p,p,0,tau);
alpha10_1332 = int(alpha9_133*U2_p,p,0,tau);
alpha10_1333 = int(alpha9_133*U3_p,p,0,tau);
alpha10_1334 = int(alpha9_133*U4_p,p,0,tau);

alpha10_1341 = int(alpha9_134*U1_p,p,0,tau);
alpha10_1342 = int(alpha9_134*U2_p,p,0,tau);
alpha10_1343 = int(alpha9_134*U3_p,p,0,tau);
alpha10_1344 = int(alpha9_134*U4_p,p,0,tau);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alpha10_1411 = int(alpha9_141*U1_p,p,0,tau);
alpha10_1412 = int(alpha9_141*U2_p,p,0,tau);
alpha10_1413 = int(alpha9_141*U3_p,p,0,tau);
alpha10_1414 = int(alpha9_141*U4_p,p,0,tau);

alpha10_1421 = int(alpha9_142*U1_p,p,0,tau);
alpha10_1422 = int(alpha9_142*U2_p,p,0,tau);
alpha10_1423 = int(alpha9_142*U3_p,p,0,tau);
alpha10_1424 = int(alpha9_142*U4_p,p,0,tau);
```

```
alpha10_1431 = int(alpha9_143*U1_p,p,0,tau);
alpha10_1432 = int(alpha9_143*U2_p,p,0,tau);
alpha10_1433 = int(alpha9_143*U3_p,p,0,tau);
alpha10_1434 = int(alpha9_143*U4_p,p,0,tau);

alpha10_1441 = int(alpha9_144*U1_p,p,0,tau);
alpha10_1442 = int(alpha9_144*U2_p,p,0,tau);
alpha10_1443 = int(alpha9_144*U3_p,p,0,tau);
alpha10_1444 = int(alpha9_144*U4_p,p,0,tau);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alpha10_2311 = int(alpha9_231*U1_p,p,0,tau);
alpha10_2312 = int(alpha9_231*U2_p,p,0,tau);
alpha10_2313 = int(alpha9_231*U3_p,p,0,tau);
alpha10_2314 = int(alpha9_231*U4_p,p,0,tau);

alpha10_2321 = int(alpha9_232*U1_p,p,0,tau);
alpha10_2322 = int(alpha9_232*U2_p,p,0,tau);
alpha10_2323 = int(alpha9_232*U3_p,p,0,tau);
alpha10_2324 = int(alpha9_232*U4_p,p,0,tau);

alpha10_2331 = int(alpha9_233*U1_p,p,0,tau);
alpha10_2332 = int(alpha9_233*U2_p,p,0,tau);
alpha10_2333 = int(alpha9_233*U3_p,p,0,tau);
alpha10_2334 = int(alpha9_233*U4_p,p,0,tau);

alpha10_2341 = int(alpha9_234*U1_p,p,0,tau);
alpha10_2342 = int(alpha9_234*U2_p,p,0,tau);
alpha10_2343 = int(alpha9_234*U3_p,p,0,tau);
alpha10_2344 = int(alpha9_234*U4_p,p,0,tau);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alpha10_3411 = int(alpha9_341*U1_p,p,0,tau);
alpha10_3412 = int(alpha9_341*U2_p,p,0,tau);
alpha10_3413 = int(alpha9_341*U3_p,p,0,tau);
alpha10_3414 = int(alpha9_341*U4_p,p,0,tau);

alpha10_3421 = int(alpha9_342*U1_p,p,0,tau);
alpha10_3422 = int(alpha9_342*U2_p,p,0,tau);
alpha10_3423 = int(alpha9_342*U3_p,p,0,tau);
alpha10_3424 = int(alpha9_342*U4_p,p,0,tau);

alpha10_3431 = int(alpha9_343*U1_p,p,0,tau);
alpha10_3432 = int(alpha9_343*U2_p,p,0,tau);
alpha10_3433 = int(alpha9_343*U3_p,p,0,tau);
alpha10_3434 = int(alpha9_343*U4_p,p,0,tau);

alpha10_3441 = int(alpha9_344*U1_p,p,0,tau);
alpha10_3442 = int(alpha9_344*U2_p,p,0,tau);
alpha10_3443 = int(alpha9_344*U3_p,p,0,tau);
alpha10_3444 = int(alpha9_344*U4_p,p,0,tau);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Betas
%12
beta3 = dictionary();
```

```
beta3(1211) = calculate_beta(w,k,p_val,[1,2,1,1],alpha10_1211);
beta3(1212) = calculate_beta(w,k,p_val,[1,2,1,2],alpha10_1212);
beta3(1213) = calculate_beta(w,k,p_val,[1,2,1,3],alpha10_1213);
beta3(1214) = calculate_beta(w,k,p_val,[1,2,1,4],alpha10_1214);

beta3(1221) = calculate_beta(w,k,p_val,[1,2,2,1],alpha10_1221);
beta3(1222) = calculate_beta(w,k,p_val,[1,2,2,2],alpha10_1222);
beta3(1223) = calculate_beta(w,k,p_val,[1,2,2,3],alpha10_1223);
beta3(1224) = calculate_beta(w,k,p_val,[1,2,2,4],alpha10_1224);

beta3(1231) = calculate_beta(w,k,p_val,[1,2,3,1],alpha10_1231);
beta3(1232) = calculate_beta(w,k,p_val,[1,2,3,2],alpha10_1232);
beta3(1233) = calculate_beta(w,k,p_val,[1,2,3,3],alpha10_1233);
beta3(1234) = calculate_beta(w,k,p_val,[1,2,3,4],alpha10_1234);

beta3(1241) = calculate_beta(w,k,p_val,[1,2,4,1],alpha10_1241);
beta3(1242) = calculate_beta(w,k,p_val,[1,2,4,2],alpha10_1242);
beta3(1243) = calculate_beta(w,k,p_val,[1,2,4,3],alpha10_1243);
beta3(1244) = calculate_beta(w,k,p_val,[1,2,4,4],alpha10_1244);


%14
beta3(1411) = calculate_beta(w,k,p_val,[1,4,1,1],alpha10_1411);
beta3(1412) = calculate_beta(w,k,p_val,[1,4,1,2],alpha10_1412);
beta3(1413) = calculate_beta(w,k,p_val,[1,4,1,3],alpha10_1413);
beta3(1414) = calculate_beta(w,k,p_val,[1,4,1,4],alpha10_1414);

beta3(1421) = calculate_beta(w,k,p_val,[1,4,2,1],alpha10_1421);
beta3(1422) = calculate_beta(w,k,p_val,[1,4,2,2],alpha10_1422);
beta3(1423) = calculate_beta(w,k,p_val,[1,4,2,3],alpha10_1423);
beta3(1424) = calculate_beta(w,k,p_val,[1,4,2,4],alpha10_1424);

beta3(1431) = calculate_beta(w,k,p_val,[1,4,3,1],alpha10_1431);
beta3(1432) = calculate_beta(w,k,p_val,[1,4,3,2],alpha10_1432);
beta3(1433) = calculate_beta(w,k,p_val,[1,4,3,3],alpha10_1433);
beta3(1434) = calculate_beta(w,k,p_val,[1,4,3,4],alpha10_1434);

beta3(1441) = calculate_beta(w,k,p_val,[1,4,4,1],alpha10_1441);
beta3(1442) = calculate_beta(w,k,p_val,[1,4,4,2],alpha10_1442);
beta3(1443) = calculate_beta(w,k,p_val,[1,4,4,3],alpha10_1443);
beta3(1444) = calculate_beta(w,k,p_val,[1,4,4,4],alpha10_1444);

%23
beta3(2311) = calculate_beta(w,k,p_val,[2,3,1,1],alpha10_2311);
beta3(2312) = calculate_beta(w,k,p_val,[2,3,1,2],alpha10_2312);
beta3(2313) = calculate_beta(w,k,p_val,[2,3,1,3],alpha10_2313);
beta3(2314) = calculate_beta(w,k,p_val,[2,3,1,4],alpha10_2314);

beta3(2321) = calculate_beta(w,k,p_val,[2,3,2,1],alpha10_2321);
beta3(2322) = calculate_beta(w,k,p_val,[2,3,2,2],alpha10_2322);
beta3(2323) = calculate_beta(w,k,p_val,[2,3,2,3],alpha10_2323);
beta3(2324) = calculate_beta(w,k,p_val,[2,3,2,4],alpha10_2324);

beta3(2331) = calculate_beta(w,k,p_val,[2,3,3,1],alpha10_2331);
beta3(2332) = calculate_beta(w,k,p_val,[2,3,3,2],alpha10_2332);
beta3(2333) = calculate_beta(w,k,p_val,[2,3,3,3],alpha10_2333);
beta3(2334) = calculate_beta(w,k,p_val,[2,3,3,4],alpha10_2334);
```

```
beta3(2341) = calculate_beta(w,k,p_val,[2,3,4,1],alpha10_2341);
beta3(2342) = calculate_beta(w,k,p_val,[2,3,4,2],alpha10_2342);
beta3(2343) = calculate_beta(w,k,p_val,[2,3,4,3],alpha10_2343);
beta3(2344) = calculate_beta(w,k,p_val,[2,3,4,4],alpha10_2344);

%34
beta3(3411) = calculate_beta(w,k,p_val,[3,4,1,1],alpha10_3411);
beta3(3412) = calculate_beta(w,k,p_val,[3,4,1,2],alpha10_3412);
beta3(3413) = calculate_beta(w,k,p_val,[3,4,1,3],alpha10_3413);
beta3(3414) = calculate_beta(w,k,p_val,[3,4,1,4],alpha10_3414);

beta3(3421) = calculate_beta(w,k,p_val,[3,4,2,1],alpha10_3421);
beta3(3422) = calculate_beta(w,k,p_val,[3,4,2,2],alpha10_3422);
beta3(3423) = calculate_beta(w,k,p_val,[3,4,2,3],alpha10_3423);
beta3(3424) = calculate_beta(w,k,p_val,[3,4,2,4],alpha10_3424);

beta3(3431) = calculate_beta(w,k,p_val,[3,4,3,1],alpha10_3431);
beta3(3432) = calculate_beta(w,k,p_val,[3,4,3,2],alpha10_3432);
beta3(3433) = calculate_beta(w,k,p_val,[3,4,3,3],alpha10_3433);
beta3(3434) = calculate_beta(w,k,p_val,[3,4,3,4],alpha10_3434);

beta3(3441) = calculate_beta(w,k,p_val,[3,4,4,1],alpha10_3441);
beta3(3442) = calculate_beta(w,k,p_val,[3,4,4,2],alpha10_3442);
beta3(3443) = calculate_beta(w,k,p_val,[3,4,4,3],alpha10_3443);
beta3(3444) = calculate_beta(w,k,p_val,[3,4,4,4],alpha10_3444);

%% Output
out = beta3;
save("beta3","beta3")

end



function beta_out = calculate_beta(w,k,p,order,alpha)
syms tau
m = calculate_m(k,order);
sum_p = calculate_sump(p,order);
T = 2*pi*m;
beta_out = w^sum_p/(12*T)*int(alpha,tau,0,T);
end

function m = calculate_m(k,order)
n = length(k);
switch n
    case 2
       k1 = k(order(1));
       k2 = k(order(2));
       term1 = ceil(1/k1);
       term2 = ceil(1/k2);
       m= lcm(term1,term2);

    case 3
       k1 = k(order(1));
       k2 = k(order(2));
       k3 = k(order(3));
       term1 = ceil(1/k1);
       term2 = ceil(1/k2);
```

```
        term3 = ceil(1/k3);
        lcm12 = lcm(term1,term2);
        m = lcm(lcm12,term3);

    case 4
        k1 = k(order(1));
        k2 = k(order(2));
        k3 = k(order(3));
        k4 = k(order(4));
        term1 = ceil(1/k1);
        term2 = ceil(1/k2);
        term3 = ceil(1/k3);
        term4 = ceil(1/k4);
        lcm12 = lcm(term1,term2);
        lcm123  = lcm(lcm12,term3);
        m = lcm(lcm123,term4);
end
end

function sump = calculate_sump(p,order)
n = length(order); % Number of elements in p

switch n
    case 2
         i = order(1);
         j = order(2);
         sump = p(i) + p(j) - 1;
    case 3
         i = order(1);
         j = order(2);
         k = order(3);
         sump = p(i) + p(j) + p(k) - 2;

    case 4
         i = order(1);
         j = order(2);
         k = order(3);
         l = order(4);
         sump = p(i) + p(j) + p(k) +p(l) - 3;
    otherwise
         error('Unsupported number of elements in p or incorrect number of indices.');
end
end
```

Finally, the code to run the simulation and generate plot is attached below

```
    clc
params
P.a = 1;
P.H = 1/3;

% Parameters for integral
P.k1 = k1;
P.k2 = k2;
P.k3 = k3;
P.k4 = k4;


% Parameters for nu
```

```
P.p1 = p1;
P.p2 = p2;
P.p3 = p3;
P.p4 = p4;
P.w  = w_val;

% Simulation
x0=[2];
P.simruntime = 20;

options=odeset('RelTol',1e-10,'Stats','on');
% ESC
[t_esc2grush,x_esc2grush]= ode15s(@(t,x) dynamicsEg4_ESC2Grush(t,x,P), [0, P.
    simruntime], x0,options);

% ESC
[t_esc2,x_esc2]= ode15s(@(t,x) dynamicsEg4_ESC2(t,x,P), [0, P.simruntime], x0,options
    );

% ESC
[t_esc4,x_esc4]= ode15s(@(t,x) dynamicsEg4_ESC4(t,x,P), [0, P.simruntime], x0,
    options);

% First order LBS Grush
load nu_val2.mat
[t_lbsgrush,x_lbsgrush]= ode15s(@(t,x) dynamicsEg4_LBS_Grush(x,P), [0, P.simruntime],
     x0,options);

% First order LBS corresponding to 4 control inputs
load nu_val2.mat
P.nu12 = nu_val2(12);
P.nu34 = nu_val2(34);
[t_lbs,x_lbs]= ode15s(@(t,x) dynamicsEg4_LBS(x,P), [0, P.simruntime], x0,options);

% First order LBS corresponding to 2 control inputs
[t_lbs_2control,x_lbs_2control]= ode15s(@(t,x) dynamicsEg4_LBS_2control(x,P), [0, P.
    simruntime], x0,options);

% Second order LBS
load nu_val3.mat
P.nu_val3 = double(values(nu_val3));
[t_lbs_2nd,x_lbs_2nd]=ode15s(@(t,x) dynamicsEg4_LBS2ndOrder(x,P), [0, P.simruntime],
    x0, options);


% Third order LBS
load beta_val.mat
all_beta = double(values(beta_val));
[t_lbs_3rd,x_lbs_3rd]=ode23(@(t,x) dynamicsEg4_LBS3rdOrder(x,P,all_beta), [0, P.
    simruntime], x0);

%% Plots
% Plots
figure(1)
subplot(2,1,1)
plot(t_esc2grush,x_esc2grush,'m',"LineWidth",2)
hold on
plot(t_lbsgrush,x_lbsgrush,'b--',"LineWidth",2)
```

```
plot(t_esc4,x_esc4,'g',"LineWidth",2)
plot(t_lbs,x_lbs,"color",[0.3010 0.7450 0.9330],"LineWidth",2)
plot(t_lbs_2nd,x_lbs_2nd,'r--',"LineWidth",2)
plot(t_lbs_3rd,x_lbs_3rd,'k--',"LineWidth",2)
grid on
legend("ESC in [22] (m=2)","LBS in [22] (r =2)", "Proposed ESC (m=4)", "LBS of
   Proposed ESC (r=2)", "LBS of Proposed ESC (r=3)", "LBS of Proposed ESC (r=4)")
xlabel("Time")
ylabel("States")
title("State vs time")


subplot(2,1,2)
control_effort
fontsize(10,"points")


%%
function out=dynamicsEg4_ESC2Grush(t,x,P)

k1 = P.k1;
k2 = P.k2;

p1 = P.p1;
p2 = P.p2;


H = P.H;
w = P.w;

J = H*(x - 1)^4;

phi = (1-exp(-(J)))/(1+exp(J));
psi = exp((J))+2*log(exp(J)-1);
u = sqrt(phi)*w^0.5*sin(psi)*cos(k1*w*t)+...
    sqrt(phi)*w^0.5*cos(psi)*sin(k2*w*t);

xdot = u;

out = xdot;

end

function out=dynamicsEg4_ESC2(t,x,P)

k1 = P.k1;
k2 = P.k2;

p1 = P.p1;
p2 = P.p2;


H = P.H;
w = P.w;

J = H*(x - 1)^4;

phi = (1-exp(-(J)))/(1+exp(J));
```

```
psi = exp((J))+2*log(exp(J)-1);
u = sqrt(phi)*w^p1*sin(psi)*cos(k1*w*t)+...
    sqrt(phi)*w^p2*cos(psi)*sin(k2*w*t);

xdot = u;

out = xdot;

end

function out=dynamicsEg4_ESC4(t,x,P)

k1 = P.k1;
k2 = P.k2;
k3 = P.k3;
k4 = P.k4;

p1 = P.p1;
p2 = P.p2;
p3 = P.p3;
p4 = P.p4;

H = P.H;
w = P.w;

J = H*(x - 1)^4;

phi = (1-exp(-(J)))/(1+exp(J));
psi = exp((J))+2*log(exp(J)-1);
u = sqrt(phi)*w^p1*sin(psi)*cos(k1*w*t)+...
    sqrt(phi)*w^p2*cos(psi)*sin(k2*w*t)+...
    sqrt(phi)*w^p3*sin(psi)*cos(k3*w*t)+...
    sqrt(phi)*w^p4*cos(psi)*sin(k4*w*t);

xdot = u;

out = xdot;

end

function out=dynamicsEg4_LBS_Grush(x,P)

% Parameters
H = P.H;

% Nu
nu12 = 0.5;

% LBs
f1f2 = -4*H*(x - 1)^3;

% Dynamics
xdot = nu12*f1f2;

% Outputs
out = xdot;
end
```

```
function out=dynamicsEg4_LBS_2control(x,P)
% Parameters
H = P.H;

% Nu
nu12 = double(P.nu12);



% LBs
f1f2 = -4*H*(x - 1)^3;

% Dynamics
xdot = nu12*f1f2;

% Outputs
out = xdot;
end

function out=dynamicsEg4_LBS(x,P)
% Parameters
H = P.H;

% Nu
nu12 = double(P.nu12);
nu34 = double(P.nu34);



% LBs
f1f2 = -4*H*(x - 1)^3;
f3f4 = f1f2;

% Dynamics
xdot = nu12*f1f2+nu34*f3f4;

% Outputs
out = xdot;
end

function out=dynamicsEg4_LBS2ndOrder(x,P)

x = x(1);
H = P.H;


% all_nu = double(values(nu));
all_nu = P.nu_val3;

% Vector field
f1f2_f1 = (12*H*exp((-0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^0.5 - 4*H*(x -
     1)^3*((4*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)
    ^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 - (2*H*exp
    ((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H
    *(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x - 1)^4) + 1)^1.5 + (2*H*exp((-(0.5*H*(x - 1)
    ^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x -
    1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5));
```

```
f1f2_f2 = 4*H*(x - 1)^3*((4*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1)
    + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^3)/(exp(H*(x - 1)^4) - 1)
    ^0.5 + (2*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)
    ^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x - 1)^4) + 1)^1 - (2*H*exp
    ((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)) + (12*H*exp((-(0.5*H
    *(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x -
    1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^0.5;

f = dictionary();
f(121) = f1f2_f1;
f(122) = f1f2_f2;
f(123) = f1f2_f1;
f(124) = f1f2_f2;

f(141) = f1f2_f1;
f(142) = f1f2_f2;
f(143) = f1f2_f1;
f(144) = f1f2_f2;

f(231)= f1f2_f1;
f(232) = f1f2_f2;
f(233) = f1f2_f1;
f(234) = f1f2_f2;

f(341) = f1f2_f1;
f(342) = f1f2_f2;
f(343)= f1f2_f1;
f(344) = f1f2_f2;


% The negative sign is present since we need f1_f1f2 instead of f1f2_f1.
all_f = -f.values();


% Dynamics
xdot1 = dynamicsEg4_LBS(x,P);
xdot2 = sum(all_nu.*all_f);
xdot = xdot1+xdot2;

% Output
out = xdot;
end

function out = dynamicsEg4_LBS3rdOrder(x,P,all_beta)
H = P.H;




f1f2f1f1 = - (4*H*(x - 1)^3*((4*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) -
     1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^3)/(exp(H*(x - 1)^4) -
    1)^0.5 - (2*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x - 1)^4) + 1)^1.5 + (2*H*exp
    ((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)) - (12*H*exp((-(0.5*H
    *(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x -
    1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^0.5)*((4*H*exp((0.5*H*(x - 1)^4))*cos
```

```
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x -
    1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 - (2*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x
    - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x -
     1)^4) + 1)^1.5 + (2*H*exp((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) +
     exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)
    ^0.5)) - (exp((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(4*H*(x - 1)^3*((4*H^2*exp(1.5*H*(x - 1)^4)*sin
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) -
    1)^0.5*(exp(H*(x - 1)^4) + 1)^1.5) - (12*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H
    *(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^2)/(exp(H
    *(x - 1)^4) - 1)^0.5 + (16*H^2*exp(1.5*H*(x - 1)^4)*sin(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*(exp(H*(x - 1)^4) + 1)^1.5*(x - 1)^6)/(exp(H*(x - 1)^4) -
    1)^1.5 + (6*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^1.5 - (6*H*exp
    ((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^2)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5) - (8*H^2*exp((0.5*H*(x
     - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)
    ^4) + 1)*(x - 1)^6)/(exp(H*(x - 1)^4) - 1)^0.5 - (8*H^2*exp((0.5*H*(x - 1)^4))*cos
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x -
    1)^6)/(exp(H*(x - 1)^4) - 1)^1.5 + (4*H^2*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(
    x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^6)/(exp(H*(x
     - 1)^4) + 1)^1.5 - (12*H^2*exp(1.5*H*(x - 1)^4)*sin(2*log(exp(H*(x - 1)^4) - 1) +
     exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^6)/(exp(H*(x - 1)^4) + 1)
    ^2.5 + (8*H^2*exp(1.5*H*(x - 1)^4)*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)
    ^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^6)/(exp(H*(x - 1)^4) - 1)^1.5 + (4*H^2*exp
    ((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^6)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5) + (4*H^2*exp((0.5*H*(x
     - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(
    x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^1.5) + (4*H^2*exp((0.5*H*(x - 1)^4))*sin
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) -
    1)^1.5*(exp(H*(x - 1)^4) + 1)^0.5)) - 12*H*(x - 1)^2*((4*H*exp((0.5*H*(x - 1)^4))*
    cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x
    - 1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 - (2*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(
    x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x
     - 1)^4) + 1)^1.5 + (2*H*exp((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1)
     + exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)
    ^0.5)) + (12*H*exp((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(
    x - 1)^4))*(2*x - 2)*sqrt(exp(H*(x - 1)^4) - 1))/(exp(H*(x - 1)^4) + 1)^0.5 + (48*
    H^2*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*
    sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^5)/(exp(H*(x - 1)^4) - 1)^0.5 - (24*H^2*exp
    ((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(
    exp(H*(x - 1)^4) - 1)*(x - 1)^5)/(exp(H*(x - 1)^4) + 1)^0.5 - (24*H^2*exp((0.5*H*(
    x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)
    ^4) - 1)*(x - 1)^5)/(exp(H*(x - 1)^4) + 1)^1.5 + (24*H^2*exp((0.5*H*(x - 1)^4))*
    sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^5)/((exp(H*(x - 1)^4)
    - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)))/(exp(H*(x - 1)^4) + 1)^0.5;
f1f2f1f2 = (4*H*(x - 1)^3*((4*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^3)/(exp(H*(x - 1)^4) -
    1)^0.5 - (2*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x - 1)^4) + 1)^1.5 + (2*H*exp
    ((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)) - (12*H*exp((-(0.5*H
    *(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x -
    1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^0.5)*((4*H*exp((0.5*H*(x - 1)^4))*sin
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x -
    1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 + (2*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x
    - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x -
     1)^4) + 1)^1.5 - (2*H*exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) +
```

```
    exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)
    ^0.5)) - (exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(4*H*(x - 1)^3*((4*H^2*exp(1.5*H*(x - 1)^4)*sin
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) -
    1)^0.5*(exp(H*(x - 1)^4) + 1)^1.5) - (12*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H
    *(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^2)/(exp(H
    *(x - 1)^4) - 1)^0.5 + (16*H^2*exp(1.5*H*(x - 1)^4)*sin(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*(exp(H*(x - 1)^4) + 1)^1.5*(x - 1)^6)/(exp(H*(x - 1)^4) -
    1)^1.5 + (6*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^1.5 - (6*H*exp
    ((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^2)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5) - (8*H^2*exp((0.5*H*(x
     - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)
    ^4) + 1)*(x - 1)^6)/(exp(H*(x - 1)^4) - 1)^0.5 - (8*H^2*exp((0.5*H*(x - 1)^4))*cos
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x -
    1)^6)/(exp(H*(x - 1)^4) - 1)^1.5 + (4*H^2*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(
    x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^6)/(exp(H*(x
     - 1)^4) + 1)^1.5 - (12*H^2*exp(1.5*H*(x - 1)^4)*sin(2*log(exp(H*(x - 1)^4) - 1) +
     exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^6)/(exp(H*(x - 1)^4) + 1)
    ^2.5 + (8*H^2*exp(1.5*H*(x - 1)^4)*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)
    ^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^6)/(exp(H*(x - 1)^4) - 1)^1.5 + (4*H^2*exp
    ((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^6)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5) + (4*H^2*exp((0.5*H*(x
     - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(
    x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^1.5) + (4*H^2*exp((0.5*H*(x - 1)^4))*sin
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) -
    1)^1.5*(exp(H*(x - 1)^4) + 1)^0.5)) - 12*H*(x - 1)^2*((4*H*exp((0.5*H*(x - 1)^4))*
    cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x
    - 1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 - (2*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(
    x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x
     - 1)^4) + 1)^1.5 + (2*H*exp((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1)
     + exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)
    ^0.5)) + (12*H*exp((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(
    x - 1)^4))*(2*x - 2)*sqrt(exp(H*(x - 1)^4) - 1))/(exp(H*(x - 1)^4) + 1)^0.5 + (48*
    H^2*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*
    sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^5)/(exp(H*(x - 1)^4) - 1)^0.5 - (24*H^2*exp
    ((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(
    exp(H*(x - 1)^4) - 1)*(x - 1)^5)/(exp(H*(x - 1)^4) + 1)^0.5 - (24*H^2*exp((0.5*H*(
    x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)
    ^4) - 1)*(x - 1)^5)/(exp(H*(x - 1)^4) + 1)^1.5 + (24*H^2*exp((0.5*H*(x - 1)^4))*
    sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^5)/((exp(H*(x - 1)^4)
    - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)))/(exp(H*(x - 1)^4) + 1)^0.5;
f1f2f2f1 = (4*H*(x - 1)^3*((4*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^3)/(exp(H*(x - 1)^4) -
    1)^0.5 + (2*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x - 1)^4) + 1)^1.5 - (2*H*exp
    ((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)) + (12*H*exp((-(0.5*H
    *(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x -
    1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^0.5)*((4*H*exp((0.5*H*(x - 1)^4))*cos
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x -
    1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 - (2*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x
    - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x -
     1)^4) + 1)^1.5 + (2*H*exp((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) +
     exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)
    ^0.5)) - (exp((-(0.5*H*(x - 1)^4)))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(12*H*(x - 1)^2*((4*H*exp((0.5*H*(x - 1)^4))*sin
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x -
```

```
    1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 + (2*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x
    - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x -
    1)^4) + 1)^1.5 - (2*H*exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) +
    exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)
    ^0.5)) + 4*H*(x - 1)^3*((16*H^2*exp(1.5*H*(x - 1)^4)*cos(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*(exp(H*(x - 1)^4) + 1)^1.5*(x - 1)^6)/(exp(H*(x - 1)^4) -
    1)^1.5 + (12*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^2)/(exp(H*(x - 1)^4) - 1)^0.5 + (6*H*
    exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(
    exp(H*(x - 1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^1.5 - (6*H*exp((-(0.5*H*(x
    - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^2)/((exp(H
    *(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5) + (8*H^2*exp((0.5*H*(x - 1)^4))*
    sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x
    - 1)^6)/(exp(H*(x - 1)^4) - 1)^0.5 + (4*H^2*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H
    *(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^6)/(exp(H
    *(x - 1)^4) + 1)^1.5 + (8*H^2*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^6)/(exp(H*(x - 1)^4) -
    1)^1.5 - (12*H^2*exp(1.5*H*(x - 1)^4)*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^6)/(exp(H*(x - 1)^4) + 1)^2.5 + (4*H^2*
    exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x -
    1)^6)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5) - (8*H^2*exp(1.5*H
    *(x - 1)^4)*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)
    ^4) + 1)*(x - 1)^6)/(exp(H*(x - 1)^4) - 1)^1.5 + (4*H^2*exp((0.5*H*(x - 1)^4))*cos
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) -
    1)^0.5*(exp(H*(x - 1)^4) + 1)^1.5) + (4*H^2*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H
    *(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) - 1)^1.5*(exp(H
    *(x - 1)^4) + 1)^0.5) + (4*H^2*exp(1.5*H*(x - 1)^4)*cos(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) +
    1)^1.5)) + (12*H*exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H
    *(x - 1)^4))*(2*x - 2)*sqrt(exp(H*(x - 1)^4) - 1))/(exp(H*(x - 1)^4) + 1)^0.5 -
    (24*H^2*exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)
    ^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^5)/(exp(H*(x - 1)^4) + 1)^0.5 - (48*H^2*
    exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(
    exp(H*(x - 1)^4) + 1)*(x - 1)^5)/(exp(H*(x - 1)^4) - 1)^0.5 - (24*H^2*exp((0.5*H*(
    x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)
    ^4) - 1)*(x - 1)^5)/(exp(H*(x - 1)^4) + 1)^1.5 + (24*H^2*exp((0.5*H*(x - 1)^4))*
    cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^5)/((exp(H*(x - 1)^4)
    - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)))/(exp(H*(x - 1)^4) + 1)^0.5;
f1f2f2f2 = - (4*H*(x - 1)^3*((4*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^3)/(exp(H*(x - 1)^4) -
    1)^0.5 + (2*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x - 1)^4) + 1)^1.5 - (2*H*exp
    ((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)
    ^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)) + (12*H*exp((-(0.5*H
    *(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x -
    1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^0.5)*((4*H*exp((0.5*H*(x - 1)^4))*sin
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x -
    1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 + (2*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x
    - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x -
    1)^4) + 1)^1.5 - (2*H*exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) +
    exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)
    ^0.5)) - (exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
    1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(12*H*(x - 1)^2*((4*H*exp((0.5*H*(x - 1)^4))*sin
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x -
    1)^3)/(exp(H*(x - 1)^4) - 1)^0.5 + (2*H*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x
    - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^3)/(exp(H*(x -
    1)^4) + 1)^1.5 - (2*H*exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) +
    exp(H*(x - 1)^4))*(x - 1)^3)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)
```

```
    ^0.5)) + 4*H*(x - 1)^3*((16*H^2*exp(1.5*H*(x - 1)^4)*cos(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*(exp(H*(x - 1)^4) + 1)^1.5*(x - 1)^6)/(exp(H*(x - 1)^4) -
    1)^1.5 + (12*H*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
     1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^2)/(exp(H*(x - 1)^4) - 1)^0.5 + (6*H*
    exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(
    exp(H*(x - 1)^4) - 1)*(x - 1)^2)/(exp(H*(x - 1)^4) + 1)^1.5 - (6*H*exp((-(0.5*H*(x
     - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^2)/((exp(H
    *(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5) + (8*H^2*exp((0.5*H*(x - 1)^4))*
    sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x
    - 1)^6)/(exp(H*(x - 1)^4) - 1)^0.5 + (4*H^2*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H
    *(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^6)/(exp(H
    *(x - 1)^4) + 1)^1.5 + (8*H^2*exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)^4) + 1)*(x - 1)^6)/(exp(H*(x - 1)^4) -
    1)^1.5 - (12*H^2*exp(1.5*H*(x - 1)^4)*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x -
     1)^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^6)/(exp(H*(x - 1)^4) + 1)^2.5 + (4*H^2*
    exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x -
     1)^6)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5) - (8*H^2*exp(1.5*H
    *(x - 1)^4)*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)
    ^4) + 1)*(x - 1)^6)/(exp(H*(x - 1)^4) - 1)^1.5 + (4*H^2*exp((0.5*H*(x - 1)^4))*cos
    (2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) -
    1)^0.5*(exp(H*(x - 1)^4) + 1)^1.5) + (4*H^2*exp((0.5*H*(x - 1)^4))*cos(2*log(exp(H
    *(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) - 1)^1.5*(exp(H
    *(x - 1)^4) + 1)^0.5) + (4*H^2*exp(1.5*H*(x - 1)^4)*cos(2*log(exp(H*(x - 1)^4) -
    1) + exp(H*(x - 1)^4))*(x - 1)^6)/((exp(H*(x - 1)^4) - 1)^0.5*(exp(H*(x - 1)^4) +
    1)^1.5)) + (12*H*exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H
    *(x - 1)^4))*(2*x - 2)*sqrt(exp(H*(x - 1)^4) - 1))/(exp(H*(x - 1)^4) + 1)^0.5 -
    (24*H^2*exp((-(0.5*H*(x - 1)^4)))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)
    ^4))*sqrt(exp(H*(x - 1)^4) - 1)*(x - 1)^5)/(exp(H*(x - 1)^4) + 1)^0.5 - (48*H^2*
    exp((0.5*H*(x - 1)^4))*sin(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(
    exp(H*(x - 1)^4) + 1)*(x - 1)^5)/(exp(H*(x - 1)^4) - 1)^0.5 - (24*H^2*exp((0.5*H*(
    x - 1)^4))*cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*sqrt(exp(H*(x - 1)
    ^4) - 1)*(x - 1)^5)/(exp(H*(x - 1)^4) + 1)^1.5 + (24*H^2*exp((0.5*H*(x - 1)^4))*
    cos(2*log(exp(H*(x - 1)^4) - 1) + exp(H*(x - 1)^4))*(x - 1)^5)/((exp(H*(x - 1)^4)
    - 1)^0.5*(exp(H*(x - 1)^4) + 1)^0.5)))/(exp(H*(x - 1)^4) + 1)^0.5;
f2f1f1f1 = - f1f2f1f1;
f2f1f1f2 = - f1f2f1f2;
f2f1f2f1 = - f1f2f2f1;
f2f1f2f2 = - f1f2f2f2;

f1f2f1f3 = f1f2f1f1;
f1f2f1f4 = f1f2f1f2;

% f1f2f2f1 =
% f1f2f2f2 =
f1f2f2f3 = f1f2f2f1;
f1f2f2f4 = f1f2f2f2;

f1f2f3f1 = f1f2f1f1;
f1f2f3f2 = f1f2f1f2;
f1f2f3f3 = f1f2f1f2;
f1f2f3f4 = f1f2f1f2;

f1f2f4f1 = f1f2f2f1;
f1f2f4f2 = f1f2f2f2;
f1f2f4f3 = f1f2f2f1;
f1f2f4f4 = f1f2f2f2;
```

```
% 23
f2f3f1f1 = f2f1f1f1;
f2f3f1f2 = f2f1f1f2;
f2f3f1f3 = f2f1f1f1;
f2f3f1f4 = f2f1f1f2;

f2f3f2f1 = f2f1f2f1;
f2f3f2f2 = f2f1f2f2;
f2f3f2f3 = f2f1f2f1;
f2f3f2f4 = f2f1f2f2;

f2f3f3f1 = f2f1f1f1;
f2f3f3f2 = f2f1f1f2;
f2f3f3f3 = f2f1f1f1;
f2f3f3f4 = f2f1f1f2;

f2f3f4f1 = f2f1f2f1;
f2f3f4f2 = f2f1f2f2;
f2f3f4f3 = f2f1f2f1;
f2f3f4f4 = f2f1f2f2;

%14
f1f4f1f1 = f1f2f1f1;
f1f4f1f2 = f1f2f1f2;
f1f4f1f3 = f1f2f1f1;
f1f4f1f4 = f1f2f1f2;

f1f4f2f1 = f1f2f2f1;
f1f4f2f2 = f1f2f2f2;
f1f4f2f3 = f1f2f2f1;
f1f4f2f4 = f1f2f2f2;

f1f4f3f1 = f1f2f1f1;
f1f4f3f2 = f1f2f1f2;
f1f4f3f3 = f1f2f1f1;
f1f4f3f4 = f1f2f1f2;

f1f4f4f1 = f1f2f2f1;
f1f4f4f2 = f1f2f2f2;
f1f4f4f3 = f1f2f2f1;
f1f4f4f4 = f1f2f2f2;

%34
f3f4f1f1 = f1f2f1f1;
f3f4f1f2 = f1f2f1f2;
f3f4f1f3 = f1f2f1f1;
f3f4f1f4 = f1f2f1f2;

f3f4f2f1 = f1f2f2f1;
f3f4f2f2 = f1f2f2f2;
f3f4f2f3 = f1f2f2f1;
f3f4f2f4 = f1f2f2f2;

f3f4f3f1 = f1f2f1f1;
f3f4f3f2 = f1f2f1f2;
f3f4f3f3 = f1f2f1f1;
f3f4f3f4 = f1f2f1f2;
```

```
f3f4f4f1 = f1f2f2f1;
f3f4f4f2 = f1f2f2f2;
f3f4f4f3 = f1f2f2f1;
f3f4f4f4 = f1f2f2f2;


all_vec_field = [f1f2f1f1 ;f1f2f1f2 ;f1f2f1f3 ;f1f2f1f4 ;
f1f2f2f1 ;f1f2f2f2;f1f2f2f3;f1f2f2f4 ;
f1f2f3f1 ;f1f2f3f2 ;f1f2f3f3 ;f1f2f3f4 ;
f1f2f4f1 ;f1f2f4f2 ;f1f2f4f3 ;f1f2f4f4 ;
%
f1f4f1f1 ;f1f4f1f2 ;f1f4f1f3 ;f1f4f1f4 ;
f1f4f2f1 ;f1f4f2f2 ;f1f4f2f3 ;f1f4f2f4 ;
f1f4f3f1 ;f1f4f3f2 ;f1f4f3f3 ;f1f4f3f4 ;
f1f4f4f1 ;f1f4f4f2 ;f1f4f4f3 ;f1f4f4f4 ;
%
f2f3f1f1 ;f2f3f1f2 ;f2f3f1f3 ;f2f3f1f4 ;
f2f3f2f1 ;f2f3f2f2 ;f2f3f2f3 ;f2f3f2f4 ;
f2f3f3f1 ;f2f3f3f2 ;f2f3f3f3 ;f2f3f3f4 ;
f2f3f4f1 ;f2f3f4f2 ;f2f3f4f3 ;f2f3f4f4 ;
%
f3f4f1f1 ;f3f4f1f2 ;f3f4f1f3 ;f3f4f1f4 ;
f3f4f2f1 ;f3f4f2f2 ;f3f4f2f3 ;f3f4f2f4 ;
f3f4f3f1 ;f3f4f3f2 ;f3f4f3f3 ;f3f4f3f4 ;
f3f4f4f1 ;f3f4f4f2 ;f3f4f4f3 ;f3f4f4f4 ];

xdot1 = dynamicsEg4_LBS(x,P);
% xdot1and2 = dynamicsEg4_LBS2ndOrder(x,P);
xdot3 =  sum(all_beta.*all_vec_field);


x_dot = xdot1+xdot3;

out = x_dot;
end
```

with the $control\_effort.m$ as

```
    figure(1)
%% For States
yyaxis left

% For 4 input system
sum_val_4i = cumtrapz(t_esc4,abs(x_esc4).^2);
plot(t_esc4,sum_val_4i,"color",[0.3010 0.7450 0.9330],"LineWidth",2)
hold on
sum_state_4i = sum_val_4i(end)

% For Grushkovskaya
sum_val = cumtrapz(t_esc2grush,abs(x_esc2grush).^2);
plot(t_esc2grush,sum_val,"color",[0.3010 0.7450 0.9330],"LineStyle","--","LineWidth
   ",2)
hold on
ylabel("State effort")
sum_state_grush = sum_val(end)


%% For control inputs
% For 4 control inputs
```

```
yyaxis right

t = 0:0.1:P.simruntime;
u1 = P.w^P.p1*cos(P.k1*P.w*t);
u2 = P.w^P.p2*sin(P.k2*P.w*t);
u3 = P.w^P.p3*cos(P.k3*P.w*t);
u4 = P.w^P.p4*sin(P.k4*P.w*t);

sum_val1 = cumtrapz(t,abs(u1).^2);
sum_val2 = cumtrapz(t,abs(u2).^2);
sum_val3 = cumtrapz(t,abs(u3).^2);
sum_val4 = cumtrapz(t,abs(u4).^2);

sum_val = sum_val1 + sum_val2+ sum_val3+ sum_val4;
plot(t,sum_val,'r',"LineWidth",2)
sum_u_4i = sum_val(end)

% For Grushkovskaya
t = 0:0.1:P.simruntime;
u1 = P.w^0.5*cos(P.k1*P.w*t);
u2 = P.w^0.5*sin(P.k2*P.w*t);

sum_val1 = cumtrapz(t,abs(u1).^2);
sum_val2 = cumtrapz(t,abs(u2).^2);

sum_val = sum_val1 + sum_val2;
plot(t,sum_val,'r--',"LineWidth",2)
sum_u_grush = sum_val(end)

ylabel("Control effort")
legend("State effort for proposed ESC","State effort for ESC in [22]",...
    "Control effort for proposed ESC","Control effort for ESC in [22]")
grid on
xlabel("Time")
title("Effort vs time")
```

REFERENCES

[1] S. Pokhrel and S. A. Eisa, "Higher order lie bracket approximation and averaging of control-affine systems with application to extremum seeking," *arXiv preprint arXiv:2310.07092*, 2023.