b1 = [359, 527, 649, 819, 1405 ,1695];%, 2069, 2323, 2601, 2803, 4563, 5592, 5861, 6076, 6368];

b2 = [1735, 4692, 5755, 8809, 11525, 11807];%, 12143, 12400, 12516, 12922, 14692, 15687, 15944, 16133, 16607];

tn = 10000; % maximum microstates numbers

l = max(length(b1), length(b2));

fig = 0;

if length(b1) > length(b2)

dx = (b2(length(b2)) - b2(1))/length(b2);

b2 = [b2, b2+dx:dx:b2+(l - length(b2))\*dx];

else

dx = (b1(length(b1)) - b1(1))/length(b1);

b1 = [b1, b1+dx:dx:b1+(l - length(b1))\*dx];

end

Size\_t = {};

ActionRatio = zeros(1, length(b1));

Action = zeros(1, length(b1));

Res = zeros(1, length(b1));

TransitionRatio = zeros(length(b1), 4);

Tau = zeros(1, length(b1));

Cost = zeros(1, length(b1));

TX0 = {};

TY0 = {};

TXc = {};

TYc = {};

TX0Hat = {};

TY0Hat = {};

TXcHat = {};

TYcHat = {};

XHat = {};

YHat = {};

HX = {};

HY = {};

HXhat = {};

HYhat = {};

HxXhat = {};%zeros(length(b1), b2(length(b1)));

HxYhat = {};%zeros(length(b1), b2(length(b1)));

CV = {};

ConvertRate = {};

HthetaXhat = {};%zeros(length(b1), ll);

HthetaYhat = {};%zeros(length(b1), ll);

ActionRatioL = zeros(1, length(b1));

ActionL = zeros(1, length(b1));

ResL = zeros(1, length(b1));

TransitionRatioL = zeros(length(b1), 4);

TauL = zeros(1, length(b1));

CostL = zeros(1, length(b1));

TX0L = {};

TY0L = {};

TXcL = {};

TYcL = {};

TX0HatL = {};

TY0HatL = {};

TXcHatL = {};

TYcHatL = {};

HXL = {};

HYL = {};

HXhatL = {};

HYhatL = {};

XHatL = {};

YHatL = {};

HxXhatL = {};%zeros(length(b1), b2(length(b1)));

HxYhatL = {};%zeros(length(b1), b2(length(b1)));

CVL = {};

ConvertRateL = {};

HthetaXhatL = {};%zeros(length(b1), ll);

HthetaYhatL = {};%zeros(length(b1), ll);

C1 = b1(2:length(b1))-b1(1:(length(b1)-1));

C2 = b2(2:length(b2))-b2(1:(length(b2)-1));

gamma = C2./C1;

%i = 1;

%while i <= length(b1)-1

for i = 1:length(b1)-1

thetax = i;

thetay = i;

a1 = b1(i):b1(i+1)-1;

a2 = b2(i):b2(i+1)-1;

% ll = max(length(a1), length(a2));

% if length(a1) > length(a2)

% da = (a2(length(a2)) - a2(1))/(length(a1)-1);

% a2 = a2(1):da:a2(length(a2));

% else

% da = (a1(length(a1)) - a1(1))/(length(a2)-1);

% a1 = a1(1):da:a1(length(a1));

% end

dt =1;

%psudotime

tol = 100;

T = length(a1);

fail = 0;

steps = 50;

if exist('t')== 0

t = 0;

elseif t > tn | steps == 0| T<= 0 | T>= max(b1(i)/K1, b2(i)/K2) | Action(i) ~= 0

i = i+1;

% break;

end

K1 = 1;

K2 = 1;

B1 = 1;

B2 = 1;

mu1 = 1;

mu2 = 1;

r = 1; %set unit time as one unit change per unit time

% a = a1 + a2;

m = 8;%proteins a per polymer

n = 2;%proteins b per polymer

% if t == 0

% t= t+1;

% end

% if exist('t')== 0

% t = 1;

% end

% if exist('T')== 0

% T = length(a1);;

% end

% if exist('steps')== 0

% steps = length(a1);

% end

if exist('Size\_t(i,:)')== 0

Size\_t(i,:) = {[T, t, steps]};

elseif isempty(Size\_t(i,:))

Size\_t(i,:) = {[T, t, steps]};

end

while steps > 0 & T> 0 & T< max(b1(i)/K1, b2(i)/K2)

%x as normal, y as persistor

if exist("Xhat")==0

dl = (a1(length(a1))-a1(1))/(T-1);

X = a1(1):dl:a1(length(a1));

Y = spline(b1,b2,X);

if isempty(X)==0 & isempty(Y)==0

[HthetaX, HthetaY, HxX, HxY, hx, hy, hxhat, hyhat, Xhat, Yhat, sX, sY, Px, Py, Pxhat, Pyhat, actionratio, delta, gamma, deltahat, gammahat, cr, C1, C2, crhat, C1hat, C2hat, tol, Tx0, Ty0, Txc, Tyc, Tx0hat, Ty0hat, Txchat, Tychat] = dynamics1(X, Y, i, m, n, K1, K2, B1, B2, mu1, mu2, l, dt, '+');

end

% sX

t = t+dt;

else

% Hamilton Markov (Hierarchical)

if Xhat(length(Xhat))< a1(length(a1)) & Xhat(1)>= a1(1)

X0 = Xhat;

Y0 = spline(b1,b2,X0);

if isempty(X0)==0 & isempty(Y0)==0

[HthetaX, HthetaY, HxX, HxY, hx, hy, hxhat, hyhat, Xhat, Yhat, sX, sY, Px, Py, Pxhat, Pyhat, actionratio, delta, gamma, deltahat, gammahat, cr, C1, C2, crhat, C1hat, C2hat, tol, Tx0, Ty0, Txc, Tyc, Tx0hat, Ty0hat, Txchat, Tychat] = dynamics1(X0, Y0, i, m, n, K1, K2, B1, B2, mu1, mu2, l, dt, '+');

end

t = t + dt;

% sX

elseif Xhat(length(Xhat))< a1(length(a1)) & Xhat(1) < a1(1)

X1 = Xhat(Xhat < a1(1));

Y1 = spline(b1,b2,X1);

if isempty(X1)==0 & isempty(Y1)==0

[HthetaX1, HthetaY1, HxX1, HxY1, hx1, hy1, hxhat1, hyhat1, Xhat1, Yhat1, sX1, sY1, Px1, Py1, Pxhat1, Pyhat1, actionratio1, delta1, gamma1, deltahat1, gammahat1, cr1, C11, C21, crhat1, C1hat1, C2hat1, tol1, Tx01, Ty01, Txc1, Tyc1, Tx0hat1, Ty0hat1, Txchat1, Tychat1] = dynamics1(X1, Y1, i-1, m, n, K1, K2, B1, B2, mu1, mu2, l, dt, '+');

t = t + dt;

end

% sX1

X0 = Xhat(Xhat >= a1(1));

Y0 = spline(b1,b2,X0);

if isempty(X0)==0 & isempty(Y0)==0

[HthetaX0, HthetaY0, HxX0, HxY0, hx0, hy0, hxhat0, hyhat0, Xhat0, Yhat0, sX0, sY0, Px0, Py0, Pxhat0, Pyhat0, actionratio0, delta0, gamma0, deltahat0, gammahat0, cr0, C10, C20, crhat0, C1hat0, C2hat0, tol0, Tx00, Ty00, Txc0, Tyc0, Tx0hat0, Ty0hat0, Txchat0, Tychat0] = dynamics1(X0, Y0, i, m, n, K1, K2, B1, B2, mu1, mu2, l, dt, '+');

end

% sX0

%IF CONSIDER -

% if Xhat(1) < b1(i-1)

% X2 = Xhat(Xhat < b1(i-1));

% Y2 = spline(b1,b2,X2);

% [HthetaX2, HthetaY2, HxX2, HxY2, Xhat2, Yhat2, sX2, sY2, Px2, Py2, Pxhat2, Pyhat2, actionratio2, delta2, gamma2, deltahat2, gammahat2, cr2, C12, C22, crhat2, C1hat0, C2hat0, tol0, Tx00, Ty00, Txc0, Tyc0, Tx0hat0, Ty0hat0, Txchat0, Tychat0] = dynamics1(X0, Y0, i, m, n, K1, K2, B1, B2, mu1, mu2, l, dt, '+');

%end

end

end

if exist("X1") ~= 0 & exist("X0")

X = sort([X1, X0]);

Y = sort([Y1, Y0]);

Xhat = sort([Xhat1, Xhat0]);

Yhat = sort([Yhat1, Yhat0]);

HthetaX = sort([HthetaX0, HthetaX1]);

HthetaY = sort([HthetaY0, HthetaY1]);

HxX = sort([HxX0, HxX1]);

HxY = sort([HxY0, HxY1]);

hx = sort([hx0, hx1]);

hy = sort([hy0, hy1]);

hxhat = sort([hxhat0, hxhat1]);

hyhat = sort([hyhat0, hyhat1]);

sX = sort([sX0, sX1]);

sY = sort([sY0, sY1]);

Px = sort([Px0, Px1]);

Py = sort([Py0, Py1]);

Pxhat = sort([Pxhat0, Pxhat1]);

Pyhat = sort([Pyhat0, Pyhat1]);

actionratio = mean([actionratio0, actionratio1]);

delta = mean([delta0, delta1]);

gamma = mean([gamma0, gamma1]);

deltahat = mean([deltahat0, deltahat1]);

gammahat = mean([gammahat0, gammahat1]);

cr = mean([cr0, cr1]);

C1 = mean([C10, C11]);

C2 = mean([C20, C21]);

crhat = mean([crhat0, crhat1]);

C1hat = mean([C1hat0, C1hat1]);

C2hat = mean([C2hat0, C2hat1]);

tol = mean([tol0, tol1]);

Tx0= sort([Tx00, Tx01]);

Ty0 = sort([Ty00, Ty01]);

Txc = sort([Txc0, Txc1]);

Tyc = sort([Tyc0, Tyc1]);

Tx0hat = sort([Tx0hat0, Tx0hat1]);

Ty0hat = sort([Ty0hat0, Ty0hat1]);

Txchat = sort([Txchat0, Txchat1]);

Tychat = sort([Txchat0, Tychat1]);

clear X0 X1

elseif exist("X0")

% X = X1;

% Y = Y1;

% Xhat = X1hat;

% Yhat = Y1hat;

% HthetaX = HthetaX1;

% HthetaY = HthetaY1;

% HxX = HxX1;

% HxY = HxY1;

% HXhat = Hxhat1;

% HYhat = Hyhat1;

% sX = sX1;

% sY = sY1;

% Px = Px1;

% Py = Py1;

% Pxhat = Pxhat1;

% Pyhat = Pyhat1;

% actionratio = actionratio1;

% delta = delta1;

% gamma = gamma1;

% deltahat = deltahat1;

% gammahat = gammahat1;

% cr = cr1;

% C1 = C11;

% C2 = C21;

% crhat = crhat1;

% C1hat = C1hat1;

% C2hat = C2hat1;

% tol = tol1;

% Tx0= Tx01;

% Ty0 = Ty01;

% Txc = Txc1;

% Tyc = Tyc1;

% Tx0hat = Tx0hat1;

% Ty0hat = Ty0hat1;

% Txchat = Txchat1;

% Tychat = Txchat1;

clear X0

end

XL = floor(min(X)):ceil(max(X));

YL = floor(min(Y)):ceil(max(Y));

l = max(length(XL), length(YL));

XX = spline(YL(1):(l-1)/(length(XL)-1):YL(l), XL, YL(1):YL(1)+l-1);

if isempty(XX)==0 & isempty(YL)==0

[HthetaXL, HthetaYL, HxXL, HxYL, hxL, hyL, hxhatL, hyhatL, XhatL, YhatL, sXL, sYL, PxL, PyL, PxhatL, PyhatL, actionratioL, deltaL, gammaL, deltahatL, gammahatL, crL, C1L, C2L, crhatL, C1hatL, C2hatL, tolL, Tx0L, Ty0L, TxcL, TycL, Tx0hatL, Ty0hatL, TxchatL TychatL] = dynamics1(XX, YL, i, m, n, K1, K2, B1, B2, mu1, mu2, l, dt, '+');

end

if Action(i) ==0

Action(i) =(mean(sX) + mean(sY))/2;

end

if ActionL(i) ==0

ActionL(i) =(mean(sXL) + mean(sYL))/2;

end

if (abs(ActionL(i)) < 0.15\*abs((mean(sXL) + mean(sYL))/2) | (abs((mean(sXL) + mean(sYL))/2) >= 1000 & abs(ActionL(i)) < abs((mean(sXL) + mean(sYL))/2) - 80) | (abs((mean(sXL) + mean(sYL))/2) >= 100 & abs(ActionL(i)) < abs((mean(sXL) + mean(sYL))/2) - 10)) & (abs(Action(i)) < 0.15\*abs((mean(sX) + mean(sY))/2) | (abs((mean(sX) + mean(sY))/2) >= 1000 & abs(Action(i)) < abs((mean(sX) + mean(sY))/2) - 80) | (abs((mean(sX) + mean(sY))/2) >= 100 & abs(Action(i)) < abs((mean(sX) + mean(sY))/2) - 10))

T = T + 5;

Size\_t(i, :) = {[Size\_t{i,: }, [T, t, steps]]};

fail = fail +1;

end

if abs(ActionL(i)) >= 0.15\*abs((mean(sXL) + mean(sYL))/2) | (abs((mean(sXL) + mean(sYL))/2) < 1000 & abs(ActionL(i)) >= abs((mean(sXL) + mean(sYL))/2) + 80) | (abs((mean(sXL) + mean(sYL))/2) < 100 & abs(ActionL(i)) >= abs((mean(sXL) + mean(sYL))/2) + 10)

Size\_t(i, :) = {[Size\_t{i,: }, [T, t, steps]]};

d = min([length(XhatL), length(YhatL), length(XL), length(PxL), length(XhatL), length(PxhatL), length(PxL), length(PyL), length(PxhatL), length(PyhatL), length(YL), length(PyL), length(PxhatL), length(PyhatL)])

figure,

subplot(2,2,1)

plot(XX(1:d), YL(1:d))

hold on

plot(XhatL(1:d), YhatL(1:d), 'r--')

xlabel('X normals')

ylabel('Y persisters')

subplot(2,2,2)

plot(XX(1:d), PxL(1:d))

hold on

plot(XhatL(1:d), PxhatL(1:d), 'r--')

xlabel('X normals')

ylabel('Px')

subplot(2,2,3)

plot(PxL(1:d), PyL(1:d))

hold on

plot(PxhatL(1:d), PyhatL(1:d), 'r--')

ylabel('Py')

xlabel('Px')

subplot(2,2,4)

plot(YL(1:d), PyL(1:d));

hold on

plot(YhatL(1:d), PyhatL(1:d), 'r--');

ylabel('Py' )

xlabel('Y persisters')

title( ['at time',num2str(t),' with size' ,num2str(T),' from interval',num2str(i)])

saveas(gca, [num2str(fig),'pqt.png']);

fig = fig+1;

d = min([length(TxcL), length(Txchat), length(TycL), length(TychatL), length(Tx0L), length(Tx0hatL), length(Ty0L), length(Ty0hatL)])

figure,

subplot(2,2,1)

plot(0:1/(length(TxcL(1:d))-1):1, TxcL(1:d))

hold on

plot(0:1/(length(TxchatL(1:d))-1):1,TxchatL(1:d), 'r--')

xlabel('Nx/Nxc')

ylabel('Txc(Large scale)')

subplot(2,2,2)

plot(0:1/(length(TycL(1:d))-1):1, TycL(1:d))

hold on

plot(0:1/(length(TycL(1:d))-1):1, TychatL(1:d), 'r--')

xlabel('Ny/Nyc')

ylabel('Tyc(Large scale)')

subplot(2,2,3)

plot(0:1/(length(Tx0L(1:d))-1):1, Tx0L(1:d))

hold on

plot(0:1/(length(Tx0hatL(1:d))-1):1, Tx0hatL(1:d), 'r--')

xlabel('Nx/Ny0')

ylabel('Tx0(Large scale)')

subplot(2,2,4)

plot(0:1/(length(Ty0L(1:d))-1):1, Ty0L(1:d))

hold on

plot(0:1/(length(Ty0hatL(1:d))-1):1, Ty0hatL(1:d), 'r--')

xlabel('Ny/Nx0')

ylabel('Ty0(Large scale)')

title( ['at time',num2str(t),' with size' ,num2str(T),' from interval',num2str(i)])

saveas(gca, [num2str(fig),'pqt.png']);

fig = fig+1;

d = min([length(hxL), length(hyL), length(hxhatL), length(hyhatL), length(HthetaXL), length(HthetaYL), length(HxYL)])

figure,

subplot(2,2,1)

surf(meshgrid(hxL(1:d),hyL(1:d)))

title('Hamiloton(Large scale)')

subplot(2,2,2)

surf(meshgrid(hxhatL(1:d),hyhatL(1:d)))

title('Hamiloton(Large scale) predicted')

subplot(2,2,3)

plot(HthetaXL(1:d), HthetaYL(1:d))

xlabel('HthetaX(Large scale)')

ylabel('HthetaY(Large scale)')

subplot(2,2,4)

plot(HxXL(1:d), HxYL(1:d))

xlabel('HxX(Large scale)')

ylabel('HxY(Large scale)')

saveas(gca, [num2str(fig),'pqt.png']);

fig = fig+1;

ActionRatioL(i) = actionratioL;

ResL(i) = tolL;

TransitionRatioL(i, :) = [deltaL, deltahatL, gammaL, gammahatL];

TauL(i) = t;

CostL(i) = TauL(i)\*log(TauL(i));

ConvertRateL(i,:) ={mean(crL), mean(crhatL)};

TX0L(i,:) = {Tx0L};

TY0L(i,:) = {Ty0L};

TX0HatL(i,:) = {Tx0hatL};

TY0HatL(i,:) = {Ty0hatL};

TXcL(i,:) = {TxcL};

TYcL(i,:) = {TycL};

TXcHatL(i,:) = {TxchatL};

TYcHatL(i,:) = {TychatL};

for j = 1: length(sYL)-1

convergencyL(j,:) = [std(sXL(1,1:j+1)), std(sYL(1,1:j+1))];

end

CVL(i, :) = {convergencyL};

HxXhatL(i,:) = {HxXL};

HxYhatL(i,:) = {HxYL};

HthetaXhatL(i,:) = {HthetaXL};

HthetaYhatL(i,:) = {HthetaYL};

HXL(i,:) = {hxL};

HYL(i,:) = {hyL};

HXhatL(i,:) = {hxhatL};

HYhatL(i,:) = {hyhatL};

XHatL(i,:) = {XhatL};

YHatL(i,:) = {YhatL};

clear sXL sYL convergencyL HxXL HxYL HthetaXL HthetaYL C1L C2L C1hatL C2hatL

end

if abs(Action(i)) >= 0.01\*abs((mean(sX) + mean(sY))/2) | (abs((mean(sX) + mean(sY))/2) < 1000 & abs(ActionL(i)) >= abs((mean(sX) + mean(sY))/2) + 100) | (abs((mean(sX) + mean(sY))/2) < 100 & abs(Action(i)) >= abs((mean(sX) + mean(sY)/2) + 10))

Size\_t(i, :) = {[Size\_t{i,: }, [T, t, steps]]};

d = min([length(Xhat), length(Yhat), length(X), length(Px), length(Xhat), length(Pxhat), length(Px), length(Py), length(Pxhat), length(Pyhat), length(Y), length(Py), length(Pxhat), length(Pyhat)])

figure,

subplot(2,2,1)

plot(X,Y)

hold on

plot(Xhat(1:d), Yhat(1:d), 'g--')

xlabel('X normals')

ylabel('Y persisters')

subplot(2,2,2)

plot(X(1:d), Px(1:d))

hold on

plot(Xhat(1:d), Pxhat(1:d), 'g--')

xlabel('X normals')

ylabel('Px')

subplot(2,2,3)

plot(Px(1:d), Py(1:d))

hold on

plot(Pxhat(1:d), Pyhat(1:d), 'g--')

ylabel('Py')

xlabel('Px')

subplot(2,2,4)

plot(Y(1:d), Py(1:d));

hold on

plot(Yhat(1:d), Pyhat(1:d), 'g--');

ylabel('Py')

xlabel('Y persisters')

title( ['at time',num2str(t),' with size' ,num2str(T),' from interval',num2str(i)])

saveas(gca, [num2str(fig),'pqt.png']);

fig = fig +1;

d = min([length(Txc), length(Txchat), length(Tyc), length(Tychat), length(Tx0), length(Tx0hat), length(Ty0), length(Ty0hat)])

figure,

subplot(2,2,1)

plot(0:1/(length(Txc(1:d))-1):1, Txc(1:d))

hold on

plot(0:1/(length(Txchat(1:d))-1):1,Txchat(1:d), 'r--')

xlabel('Nx/Nxc')

ylabel('Txc')

subplot(2,2,2)

plot(0:1/(length(Tyc(1:d))-1):1, Tyc(1:d))

hold on

plot(0:1/(length(Tychat(1:d))-1):1, Tychat(1:d), 'r--')

xlabel('Ny/Nyc')

ylabel('Tyc')

subplot(2,2,3)

plot(0:1/(length(Tx0(1:d))-1):1, Tx0(1:d))

hold on

plot(0:1/(length(Tx0hat(1:d))-1):1, Tx0hat(1:d), 'r--')

xlabel('Nx/Ny0')

ylabel('Tx0')

subplot(2,2,4)

plot(0:1/(length(Ty0(1:d))-1):1, Ty0(1:d))

hold on

plot(0:1/(length(Ty0hat(1:d))-1):1, Ty0hat(1:d), 'r--')

xlabel('Ny/Nx0')

ylabel('Ty0')

title( ['at time',num2str(t),' with size' ,num2str(T),' from interval',num2str(i)])

saveas(gca, [num2str(fig),'pqt.png']);

fig = fig+1;

d = min([length(hx), length(hy), length(hxhat), length(hyhat), length(HthetaX), length(HthetaY), length(HxY)])

figure,

subplot(2,2,1)

surf(meshgrid(hx(1:d),hy(1:d)))

title('Hamiloton')

subplot(2,2,2)

surf(meshgrid(hxhat(1:d),hyhat(1:d)))

title('Hamiloton predicted')

subplot(2,2,3)

plot(HthetaX(1:d), HthetaY(1:d))

xlabel('HthetaX')

ylabel('HthetaY')

subplot(2,2,4)

plot(HxX(1:d), HxY(1:d))

xlabel('HxX')

ylabel('HxY')

saveas(gca, [num2str(fig),'pqt.png']);

fig = fig+1;

ActionRatio(i) = actionratio;

Res(i) = tol;

TransitionRatio(i,:) =[delta, deltahat, gamma, gammahat];

Tau(i) = t;

Cost(i) = Tau(i)\*log(Tau(i));

ConvertRate(i,:) ={mean(cr), mean(crhat)};

TX0(i,:) = {Tx0};

TY0(i,:) = {Ty0};

TX0Hat(i,:) = {Tx0hat};

TY0Hat(i,:) = {Ty0hat};

TXc(i,:) = {Txc};

TYc(i,:) = {Tyc};

TXcHat(i,:) = {Txchat};

TYcHat(i,:) = {Tychat};

for j = 1: length(sY)-1

convergency(j,:) = [std(sX(1,1:j+1)), std(sY(1,1:j+1))];

end

CV(i, :) = {convergency};

HxXhat(i,:) = {HxX};

HxYhat(i,:) = {HxY};

HthetaXhat(i,:) = {HthetaX};

HthetaYhat(i,:) = {HthetaY};

HX(i,:) = {hx};

HY(i,:) = {hy};

HXhat(i,:) = {hxhat};

HYhat(i,:) = {hyhat};

XHat(i,:) = {Xhat};

YHat(i,:) = {Yhat};

clear sX sY convergency2 HxX HxY HthetaX HthetaY C1 C2 C1hat C2hat

end

if ((XhatL(length(XhatL)) < b1(i+1)-1 & YhatL(length(YhatL)) < b2(i+1)-1) | (Xhat(length(Xhat)) < b1(i+1)-1 & Yhat(length(Yhat)) < b2(i+1)-1)) & t< tn & steps >0 %min(abs(tol), abs(tolL)) > 0.01

if fail > T-100

Size\_t(i, :) = {[Size\_t{i,: }, [T, t, steps]]};

T = length(X) + 10; % for example add large dose of drug...

display('fail to simulate within natural continuous microstates, give stimulation');

steps = steps -1;

clear Xhat;

else

fail = fail + 1; % not so useful iteration

steps = steps - 1;

clear Xhat;

end

elseif XhatL(length(XhatL)) >= b1(i+1) | YhatL(length(YhatL)) >= b2(i+1)

Xhat = XhatL;

Size\_t(i, :) = {[Size\_t{i,: }, [T, t, steps]]};

% steps = 50;

t = 0;

break;

elseif Xhat(length(Xhat)) >= b1(i+1) | Yhat(length(Yhat)) >= b2(i+1)

XhatL = Xhat;

Size\_t(i, :) = {[Size\_t{i,: }, [T, t, steps]]};

% steps = 50;

t = 0;

break;

end

end

% i = i+1;

end

%computation of rewards of continuous Markov Chain

%Use conversion rate computed as transition rate in matrix R

R = []

% figure,

% subplot()

% c1 = x1(length(x1)) - x1(1);%for a1, transcription mRNA and translation to proteins

% c2 = x2(length(c2)) - c2(1);%for a2,...

% gamma1 = c2/c1; %asociatiate rate

% w1 = c1\*N1./(1+(y1/K1/N1).^n);

% w2 = c2\*N1./(1+(x1/K2/N1).^m);

% px = log(mu1\*N1\*(1+(x1/K1/N1).^n)./c1);

% py = log(mu2\*N1\*(1+(y1/K2/N1).^m)./c2);

% dy = w2\*B2\*exp(py)-mu2\*B2\*N1\*exp(-py);

% dx = w1\*B1\*exp(px)-mu1\*B1\*N1\*exp(-px);

% cr1= ones(size(x1))./(m\*(x1/N1).^(m-1).\*(1+(y1/N1).^n).^2./(n\*(y1/N1).^(n-1))./(1+(x1/N1).^m).^2); %dy/dx

% end

% if X(length(X)) < b1(i+1) & Y(length(Y)) < b2(i+1) & X(1) > b1(1) & Y(1) > b2(1)

% delta = gamma;

% if thetax == 1

% directionx = 0.6;

% elseif thetay == 1

% directiony = 0.6;

% elseif thetax == l

% directionx = 0.4;

% elseif thetay == l

% directiony = 0.4;

% else

% directionx = rand(1);

% directiony = rand(1);

% end

% t = t + dt;

% if thetax == i & thetay == i

% if i == 1

% thetax = thetax + dt;

% thetay = thetay + dt;

% end

%

% if directionx >0.5

% thetax = thetax +dt;

% rx = C1\*B1\*W1;

% rpx = 0;% state of x change -1

% ry = C2\*B2\*W2;

% rpy = exp(Px)-1;%s tate of x change +1

% else

% thetax = thetax -dt;

% rx = -mu1\*B1\*N;

% rpx = -W1.^2\*m\*(X/K2/N).^(m-1)./C1\*exp(Py) ;

% ry = -mu2\*B2\*N;% state of x change -1

% rpy = -W2.^2\*n\*(Y/K1/N).^(n-1)./C2\*exp(Px) ;

% end

% elseif thetax == i

% if directiony >0.5

% thetay = thetay +dt;

% ry = B2\*W2;

% dy = exp(Py)-1;%state of y change +1

% rpy = 0;

% dpy = 0;% state of y change -1

% else

% thetay = thetay -dt;

% rpy = mu2\*N\*Y;

% dpy = exp(-Py)-1;% state of y change -1

% ry = 0;

% dy = 0;%state of y change +1

% end

% if thetax ~= i & thetay ~= i & CR1\*gamma >1

%

% end

% HthetaX = rx.\*Px.\*exp(Px)-rpx.\*Px.\*exp(-Px);

% HthetaY = ry.\*Py.\*exp(Py)-rpy.\*Py.\*exp(-Py);

% HxX = mu1\*N\*(exp(-Px)-1) - C2\*N\*m\*X.^(m-1).\*(exp(-Py) -1).\*(1+(X/K2/N).^m);

% HxY = mu2\*N\*(exp(-Py)-1) - C1\*N\*n\*Y.^(n-1).\*(exp(-Px) -1).\*(1+(Y/K1/N).^n);

% sX = Px.\*[X(1),(X(2:length(X))-X(1:length(X)-1))] - HthetaX -HxX;

% sY = Py.\*[Y(1),(Y(2:length(Y))-Y(1:length(Y)-1))] - HthetaY -HxY;

% actionratio = mean(abs((delta\*sY).^2./sX));

% tol = abs(mean(abs((delta\*sY).^2./sX))-gamma)/gamma;

% else

% HthetaX = [HthetaX;rx.\*Px.\*exp(Px)-rpx.\*Px.\*exp(-Px)];

% HthetaY = [HthetaY;ry.\*Py.\*exp(Py)-rpy.\*Py.\*exp(-Py)];

% HxX = [HxX;mu1\*N\*(exp(-Px)-1) - C2\*N\*m\*X.^(m-1).\*(exp(-Py) -1).\*(1+(X/K2/N).^m)];

% HxY = [HxY;mu2\*N\*(exp(-Py)-1) - C1\*N\*n\*Y.^(n-1).\*(exp(-Px) -1).\*(1+(Y/K1/N).^n)];

% sX = Px.\*[X(1),(X(2:length(X))-X(1:length(X)-1))] - HthetaX -HxX;

% sY = Py.\*[Y(1),(Y(2:length(Y))-Y(1:length(Y)-1))] - HthetaY -HxY;

% actionratio = mean(abs((delta\*sY).^2./sX));

% end

% while T > 0

% dl = (a1(length(a1)) - a1(1))/(T-1);

% %psudotime

% thetax = 1;

% thetay = 1;

% t = 1;

% %x as normal, y as persistor

% X = a1(1):dl:a1(length(a1));

% Y = spline(b1,b2,X);

% C1 = X(length(X)) - X(1);%for a, transcription mRNA and translation to proteins

% C2 = Y(length(Y)) - Y(1);%for ,...

% gamma = C2/C1; %asociatiate rate

% CR1= ones(size(X))./(m\*(X/N).^(m-1).\*(1+(Y/N).^n).^2./(n\*(Y/N).^(n-1))./(1+(X/N).^m).^2); %dy/dx

% % CR2 = (C2./(1+(X/N).^m)-mu2\*N).\*exp(Py-Px)./(C1./(1+(Y/N).^n)-mu1\*N);

% ConvertRate(t,:) = {CR1}; %dy/dx

% Px = log(mu1\*N\*(1+(X/K1/N).^n)./C1);

% Py = log(mu2\*N\*(1+(Y/K2/N).^m)./C2);

% figure,

% subplot(2,2,1)

% plot(X, Y)

% subplot(2,2,2)

% plot(X, Px);

% subplot(2,2,3)

% plot(Px, Py)

% subplot(2,2,4)

% plot(Y, Py);

% W1 = C1\*N./(1+(Y/K1/N).^n);

% W2 = C2\*N./(1+(X/K2/N).^m);

% end

% x = x\*delta;

% y = y\*delta;

% px = px \*delta;

% py = py\*delta;

% x(Pxdx(px-x +1) - eps \*(gamma\*x-y)\*(px-py))

%s = integral(px\*dx+py\*dy-h\*dT)

figure,

subplot(2,2,1)

surf(meshgrid(HthetaXhat{1},HthetaYhat{1}))

subplot(2,2,2)

surf(meshgrid(HxXhat{1},HxYhat{1}))

subplot(2,2,3)

polar(HthetaXhat{1},HthetaYhat{1})

subplot(2,2,4)

polar(HxXhat{1},HxYhat{1})

figure,

subplot(2,2,1)

surf(meshgrid(HthetaXhat{2},HthetaYhat{2}))

subplot(2,2,2)

surf(meshgrid(HxXhat{2},HxYhat{2}))

subplot(2,2,3)

polar(HthetaXhat{2},HthetaYhat{2})

subplot(2,2,4)

polar(HxXhat{2},HxYhat{2})

figure,

subplot(2,2,1)

surf(meshgrid(HthetaXhat{3},HthetaYhat{3}))

subplot(2,2,2)

surf(meshgrid(HxXhat{3},HxYhat{3}))

subplot(2,2,3)

polar(HthetaXhat{3},HthetaYhat{3})

subplot(2,2,4)

polar(HxXhat{3},HxYhat{3})

figure,

subplot(2,2,1)

surf(meshgrid(HthetaXhat{4},HthetaYhat{4}))

subplot(2,2,2)

surf(meshgrid(HxXhat{4},HxYhat{4}))

subplot(2,2,3)

polar(HthetaXhat{4},HthetaYhat{4})

subplot(2,2,4)

polar(HxXhat{4},HxYhat{4})

figure,

subplot(2,2,1)

surf(meshgrid(HthetaXhat{5},HthetaYhat{5}))

subplot(2,2,2)

surf(meshgrid(HxXhat{5},HxYhat{5}))

subplot(2,2,3)

polar(HthetaXhat{5},HthetaYhat{5})

subplot(2,2,4)

polar(HxXhat{5},HxYhat{5})

figure,

subplot(2,2,1)

surf(meshgrid(HthetaXhat{6},HthetaYhat{6}))

subplot(2,2,2)

surf(meshgrid(HxXhat{6},HxYhat{6}))

subplot(2,2,3)

polar(HthetaXhat{6},HthetaYhat{6})

subplot(2,2,4)

polar(HxXhat{6},HxYhat{6})