รายงาน

เรื่อง

**กลไกการเคลื่อนที่ของขากิ้งก่าบาซิลิสก์**

เสนอ

**รศ.ดร.รัชทิน จันทร์เจริญ**

**รศ.ดร.นภดนัย อาชวาคม**

จัดทำโดย

นายณัฐกมล พงศ์เต็มสุข รหัสนิสิต 5930159321

นางสาวนันทัชพร นันทปิยะวรรณ รหัสนิสิต 5930281121

นายสุเจตน์ โพดาพล รหัสนิสิต 5930537921

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา 2103322 MECH MACHINERY

ภาคการศึกษาต้น ปีการศึกษา2561

จุฬาลงกรณ์มหาวิทยาลัย

**สารบัญ**

**Part 1 : Introduction and Data preparation 2**

* Description 2
* Design parameter 3
* Reference data 4
* Calculation part 5
* Data plot 10

**Part 2 : 1st Simscape run for reference 11**

* Matlab script 11
* Simscape multibody model 14

**Part 3 : 2nd Simscape run for optimization 15**

* Matlab script 15
* Optimization result 17

**Part 4 : Conclusion 18**

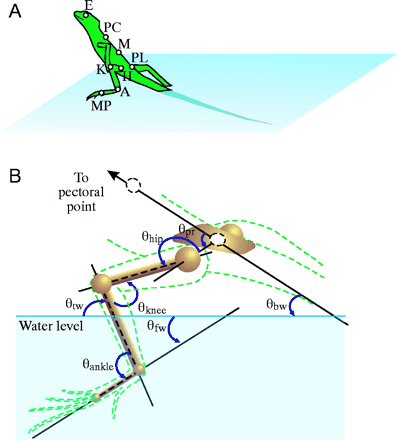
* Conclusion 18
* Problem and Solution 18

**Part 1 : Introduction and Data preparation**

**Description**

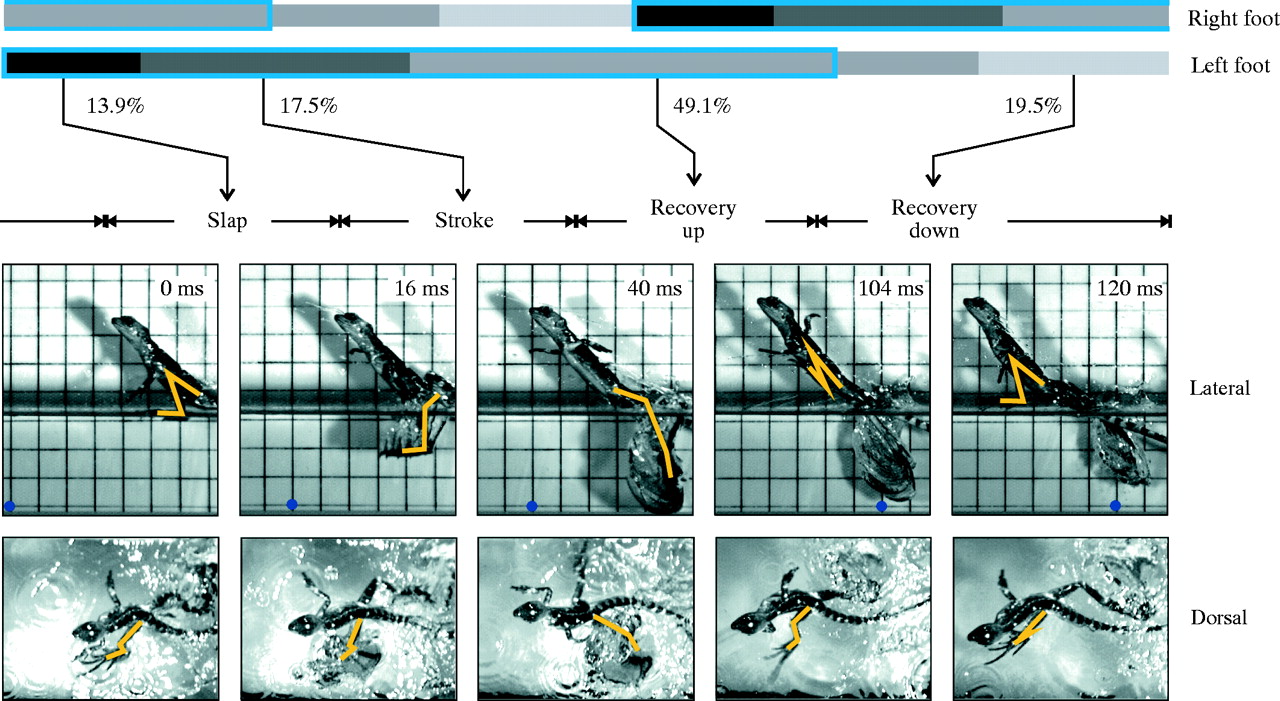
**กิ้งก่าบาซิลิสก์(Basilisk Lizard)**

รายงานฉบับนี้จัดทำขึ้นเพื่อศึกษาการเคลื่อนที่ของกิ้งก่าบาซิลิสก์ ซึ่งจะนำไปใช้ในการออกแบบและปรับปรุงกลไกที่จำลองมาจากการเคลื่อนที่ดังกล่าว โดยในการศึกษาครั้งนี้จะสนใจเฉพาะกลไกการเคลื่อนที่ของขาหลังด้านซ้ายของกิ้งก่าบาซิลิสก์ ซึ่งถูกใช้ในการวิ่งเหนือผิวน้ำ ดังรูปที่ 1 จุด 4 จุดได้แก่จุด H, K, A, MP ใช้แทนข้อต่อ(joint)ของกลไก

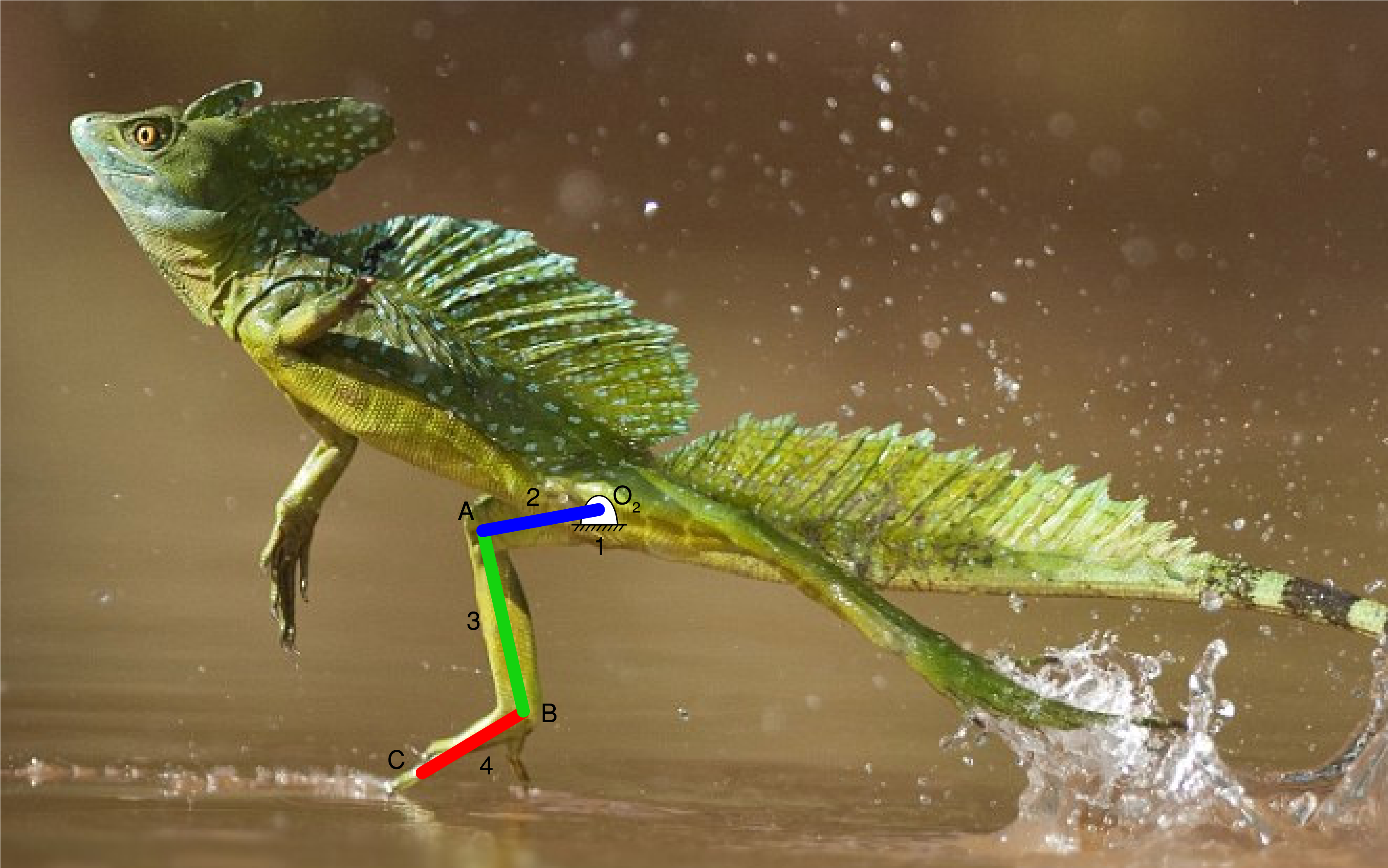
*รูปที่ 1 การเคลื่อนที่จริงและแผนภาพคิเนเมติกส์จำลองการเคลื่อนที่ของกิ้งก่าบาซิลิสก์*

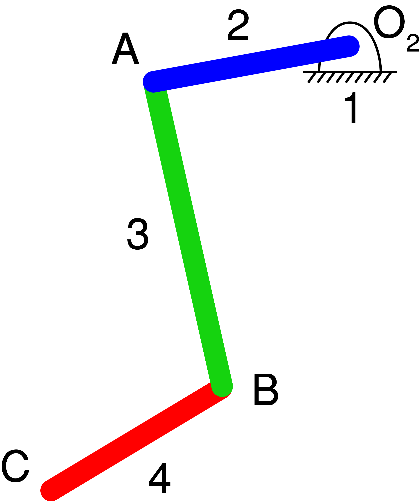
กิ้งก่าบาซิลิสก์นั้นมีการเคลื่อนที่ที่เป็นเอกลักษณ์เฉพาะตัว นั่นก็คือการวิ่งเหนือผิวน้ำ ซึ่งในการก้าวแต่ละครั้งจะมีลักษณะเหมือนการตะกุยน้ำไปข้างหน้า เริ่มจากจังหวะที่เท้าสัมผัสกับผิวน้ำ จากนั้นเท้าจะกดลงไปในน้ำโดยมีทิศทางเฉียงไปทางตรงกันข้ามกับการเคลื่อนที่ของลำตัว โดยใช้เท้าซึ่งทำหน้าที่คล้ายใบพายกวาดลงไปในน้ำเพื่อดันให้ตัวเคลื่อนที่ไปข้างหน้า ก่อนจะยกเท้ากลับขึ้นมาเหนือผิวน้ำและหมุนกลับมาแตะกับผิวน้ำอีกครั้ง นับเป็นการเคลื่อนที่ครบ 1 รอบ ดังรูปที่ 2



*รูปที่ 2 การเคลื่อนที่ของขาหลังของกิ้งก่าบาซิลิสก์ 1 รอบการเคลื่อนที่*

**Design parameter**

เมื่อพิจารณาการเคลื่อนที่ขาหลังของกิ้งก่าบาซิลิสก์ จะสามารถเขียนเป็น kinematic diagram ได้โดยสามารถพิจารณาได้เป็น 4 link (รวม ground) และ 3 revolute joint ดังนี้ โดย assume ให้ hip เป็น ground



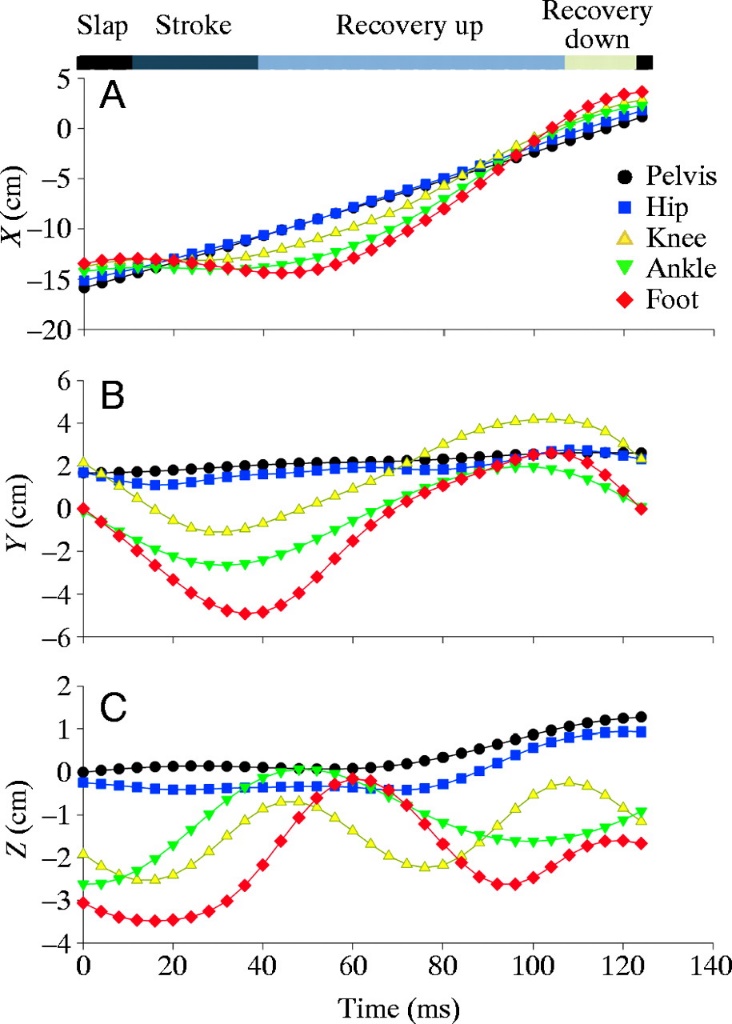
*รูปที่3 kinematic diagram ของขาหลังของกิ้งก่าบาซิลิสก์*

Parameter ทั้งหมดในการเคลื่อนที่ของกิ้งก่ามีดังนี้

 ความยาวของ link 2 [ cm ]  
 ความยาวของ link 3 [ cm ]  
 ความยาวของ link 4 [ cm ]  
มวลของ link 2 [ g ]  
มวลของ link 3 [ g ]  
มวลของ link 4 [ g ]  
 แรงบิดที่ใส่ที่ joint O2 [ Ncm ]  
 แรงบิดที่ใส่ที่ joint A [ Ncm ]  
 แรงบิดที่ใส่ที่ joint B [ Ncm ]  
 แรงเสียดทานที่ joint O2 [ N ]  
 แรงเสียดทานที่ joint A [ N ]  
 แรงเสียดทานที่ joint B [ N ]

ในการศึกษาครั้งนี้เราสนใจเพียงแต่ kinetics ของระบบ และการเคลื่อนที่ของจุด C ให้เป็นไปตาม trajectory ที่เราต้องการ ดังนั้นจึงเลือก design parameters เพื่อการ optimize คือ 

**Reference data**



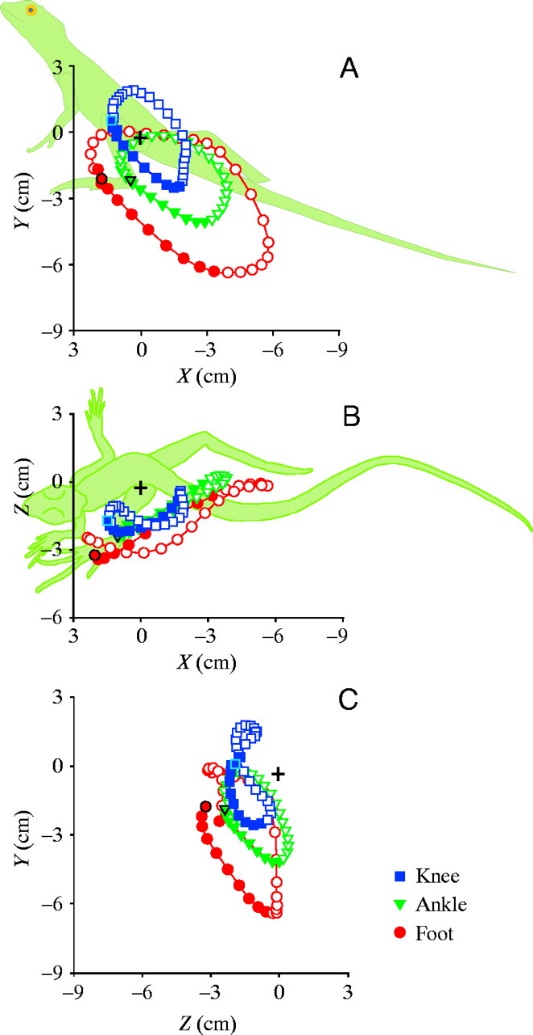
*กราฟที่1 แสดงถึงตำแหน่งของจุด pelvis, hip, knee, ankle, foot ตามแนวแกน x, y, z ตามลำดับ ณ เวลาต่างๆ*

*กราฟที่2 แสดงถึงตำแหน่งของจุด knee, ankle, foot เมื่อเทียบกับตำแหน่งของ hip ในระนาบ xy, xz, yz ตามลำดับ ในลักษณะวงปิด*

**อัตราส่วนความยาวขา**

อัตราส่วนความยาวขาท่อนบน ( Femur ) : ความยาวขาท่อนล่าง ( Tibia ) : ความยาวเท้า ( Foot ) : ความยาวขาหลังทั้งหมด ( the total hindlimb length ) มีค่าเท่ากับ 0.283 : 0.275 : 0.442 : 1 โดยมีความขาทั้งหมดเป็น 7.984803 cm ดังนั้นจะได้  2.1958 cm , 2.2597 cm , 3.5293 cm

**Source : http://jeb.biologists.org/content/206/23/4363**



**Calculation part**

**Bottom line xy**

p=polyfit(x,y,13);

x1=linspace(-2.3,5.65);

y1=polyval(p,x1);

figure

plot(x,y,'o')

hold on

plot(x1,y1)

xlabel('Position x [cm]');

ylabel('Position y [cm]');

title('Trajectory x-y');

p =

Columns 1 through 7

0.0000 0.0000 -0.0001 0.0007 -0.0026 -0.0001 0.0264

Columns 8 through 14

-0.0468 -0.0681 0.2119 0.0196 -0.1846 -0.9420 -4.2006

ได้สมการเส้น xy (เส้นล่าง) คือ +

**Upper line xy**

p1=polyfit(x2,y2,11);

x3=linspace(-2.3,5.65);

y3=polyval(p1,x3);

hold on;

plot(x2,y2,'o')

hold on;

plot(x3,y3)

hold off

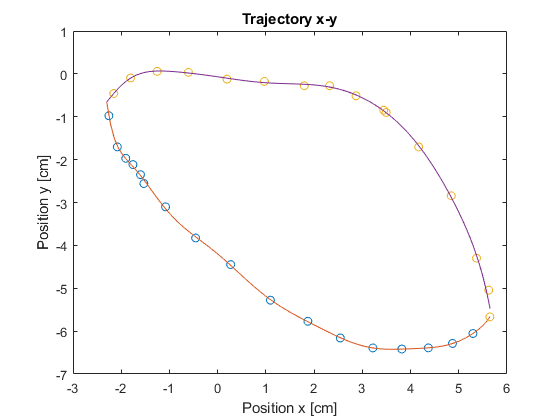
p1 =

Columns 1 through 7

-0.0000 0.0001 -0.0003 0.0003 0.0036 -0.0093 -0.0068

Columns 8 through 12

0.0208 0.0344 -0.0172 -0.1642 -0.0681

ได้สมการเส้นxy(เส้นบน) คือ

*กราฟที่3 แสดงถึงตำแหน่งของจุด foot เมื่อเทียบกับตำแหน่งของ hip ในระนาบ xy ในลักษณะวงปิด*

**x(t)**

p=polyfit(t1,xt,10);

T1=linspace(0,124.5);

xt1=polyval(p,T1);

figure

plot(t1,xt,'o')

hold on

plot(T1,xt1)

xlabel('Time [ms]');

ylabel('Position x [cm]');

title('Velocity in x axis');

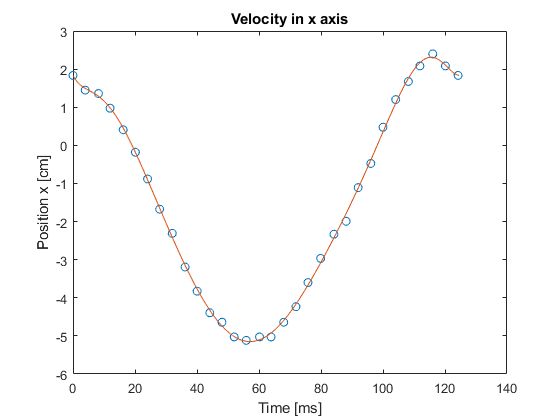
p =

Columns 1 through 7

0.0000 -0.0000 0.0000 -0.0000 0.0000 -0.0000 0.0001

Columns 8 through 11

-0.0026 0.0245 -0.1467 1.8241

ได้สมการเส้น x(t) คือ

*กราฟที่4 แสดงถึงตำแหน่งของจุด foot เทียบกับจุด hip ตามแนวแกน x ณ เวลาต่างๆ*

**y(t)**

p=polyfit(t2,yt,19);

T2=linspace(0,124.5);

yt1=polyval(p,T2);

figure

plot(t2,yt,'o')

hold on

plot(T2,yt1)

xlabel('Time [ms]');

ylabel('Position y [cm]');

title('Velocity in y axis');

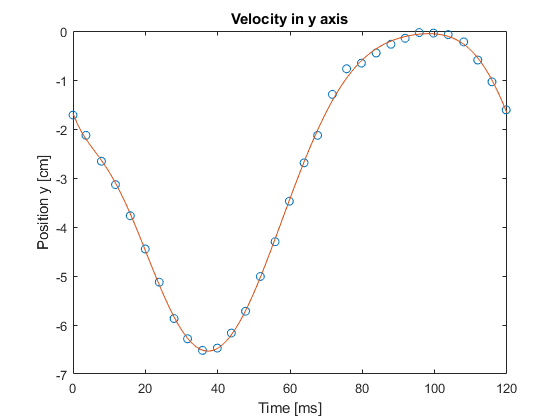
p =

Columns 1 through 7

0.0000 -0.0000 0.0000 -0.0000 0.0001 -0.0018 0.0204

Columns 8 through 9

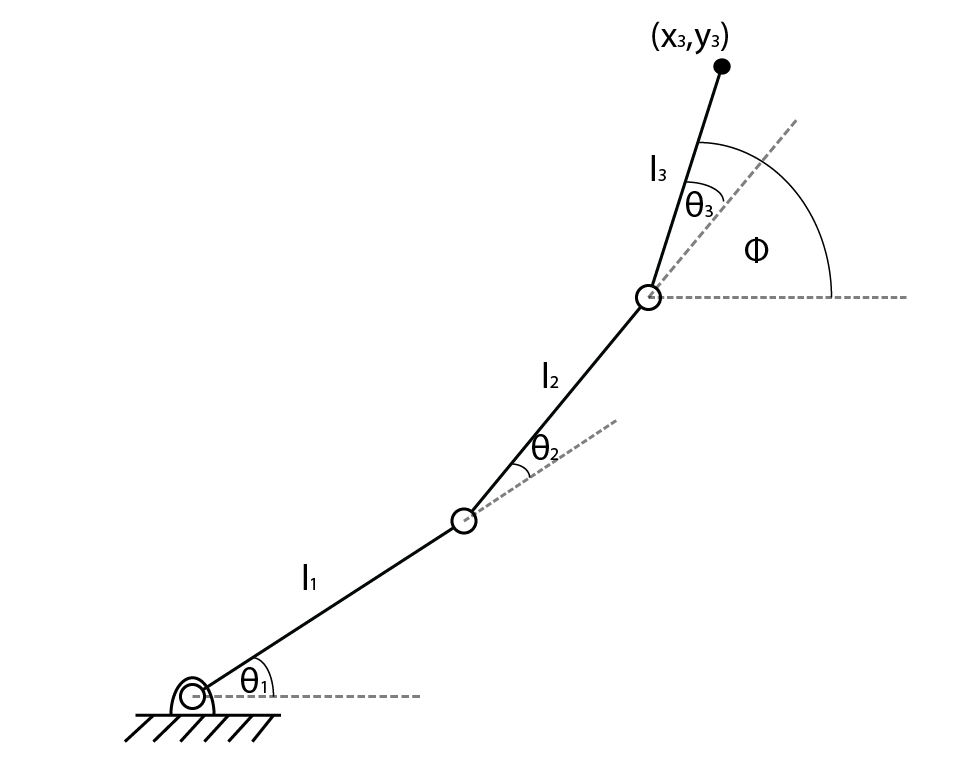
-0.1991 -1.6819

ได้สมการเส้น y(t) คือ

*กราฟที่5 แสดงถึงตำแหน่งของจุด foot เทียบกับจุด hip ตามแนวแกน y ณ เวลาต่างๆ*

**Inverse kinematics**

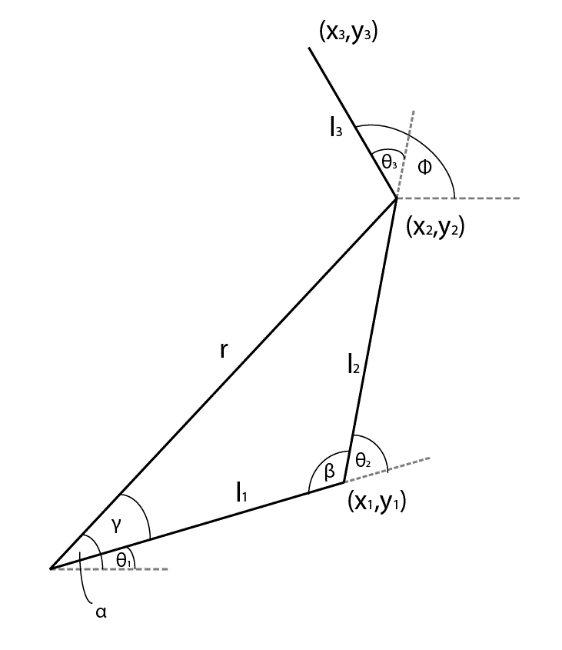
เราทราบค่า  ก Reference data แล้วจึงนำมาคำนวณเพื่อหาค่าได้ดังนี้

การเคลื่อนที่แบบ Forward

*รูปที่4 แสดงตำแหน่งของlinkทั้ง 3*

*สำหรับการเคลื่อนที่แบบforward*

การเคลื่อนที่แบบ Inverse

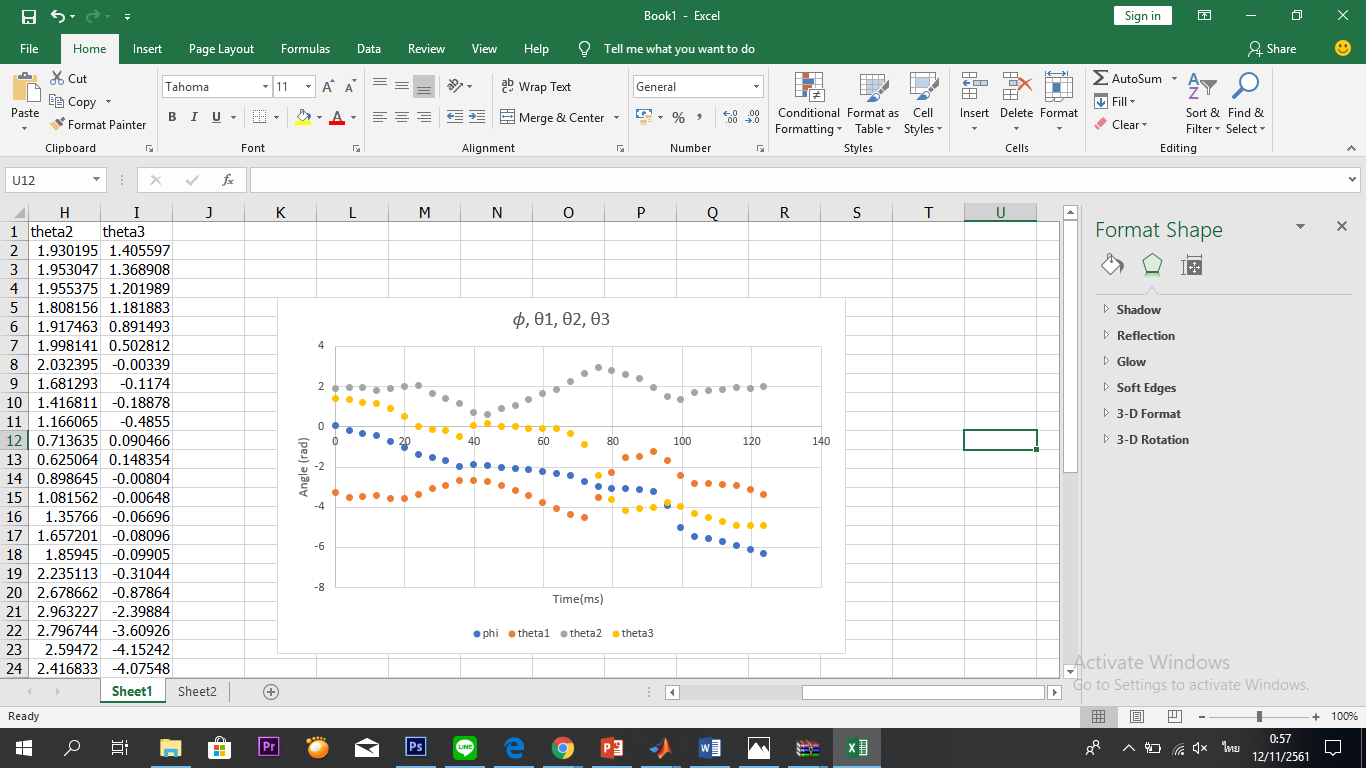


*รูปที่5 แสดงตำแหน่งของlinkทั้ง 3*

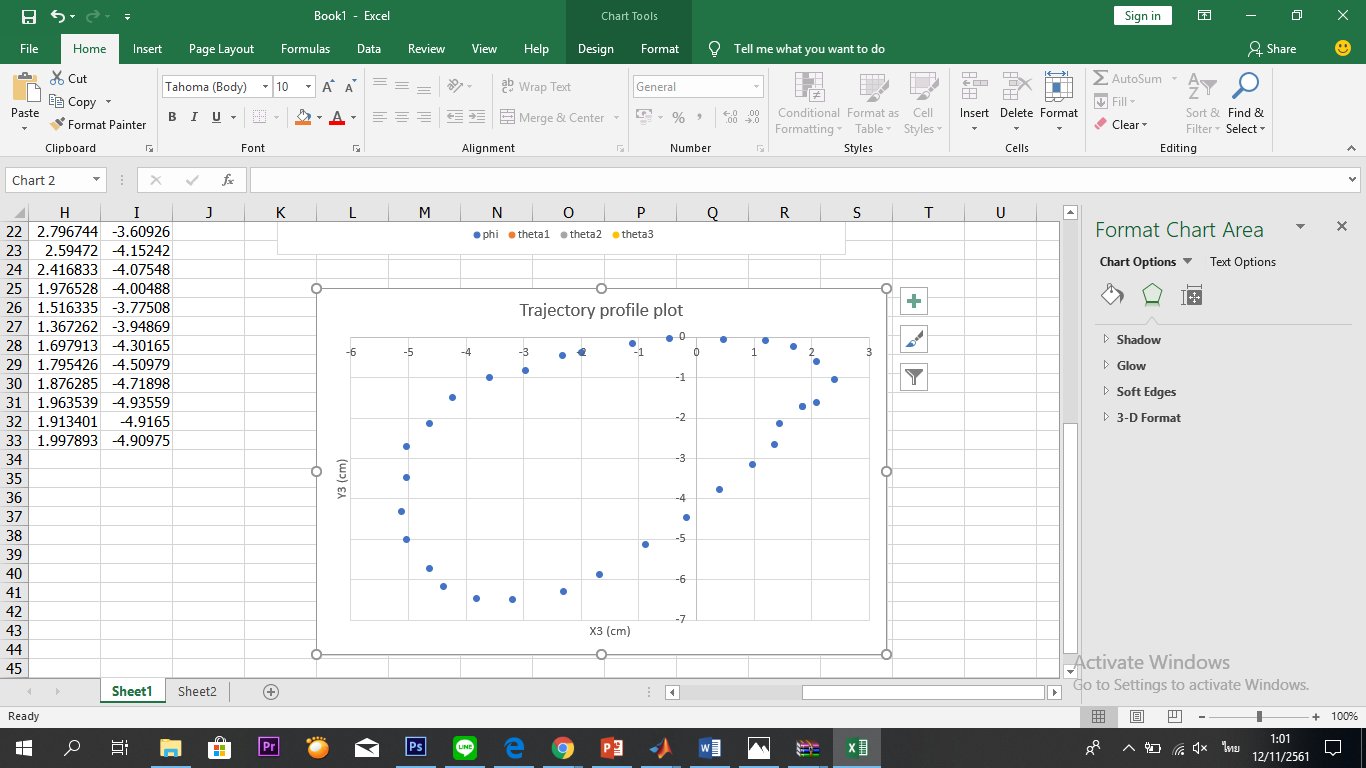
*สำหรับการเคลื่อนที่แบบinverse*

โดยจากรูปที่4 และ 5 สามารถเขียนสมการออกมาได้ดังนี้

**Data Plot**



*กราฟที่6 แสดงตำแหน่งของ* ***ณ เวลาต่างๆใน 1 รอบการเคลื่อนที่***



*กราฟที่7 แสดงถึงตำแหน่งของจุด foot เมื่อเทียบกับตำแหน่งของ hip ในระนาบ xy ในลักษณะวงปิด*

**Part 2 : 1st Simscape run for reference**

**Matlab Script**

clc;

close all;

clear all;

## Initial parameter

L = 7.984803; %leg length[cm]

R = L/200; %cylinder radius[cm]

L1 = 0.275\*L; %femur length[cm]

L2 = 0.283\*L; %tibia length[cm]

L3 = 0.442\*L; %foot length[cm]

clc;

## Load data

load('x3y3phical.mat');

## Inverse kinematics

x2 = 0;

y2 = 0;

theta1 = 0;

theta2 = 0;

theta3 = 0;

for i = 1:32

x2(i) = x3(i) - L3\*cos(phi(i));

y2(i) = y3(i) - L3\*sin(phi(i));

r = sqrt(x2(i)^2 + y2(i)^2);

theta1(i) = atan2(y2(i),x2(i)) - acos((L1^2 + r^2 - L2^2)/(2\*L1\*r));

theta2(i) = pi - acos((L1^2 + L2^2 - r^2)/(2\*L1\*L2));

theta1 = unwrap(theta1);

theta2 = unwrap(theta2);

theta3(i) = phi(i) - theta1(i) - theta2(i);

theta3 = unwrap(theta3);

end

theta1 = theta1';

theta2 = theta2';

theta3 = theta3';

## Input

time = linspace(0,10,32);

th1 = theta1; %pelvis angle[rad]

th2 = theta2; %knee angle[rad]

th3 = theta3; %ankle angle[rad]

th1x=interp1(time,th1,[0:0.00001:10],'spline');

th2x=interp1(time,th2,[0:0.00001:10],'spline');

th3x=interp1(time,th3,[0:0.00001:10],'spline');

timex=linspace(0,10,1000001);

## Simulation

sim('kinematic\_test.slx');

## Coupler curve

figure('Name','Coupler Curve','NumberTitle','off');

posx = coupler\_curve.Data(:,1);

posz = coupler\_curve.Data(:,2);

time\_coupler = coupler\_curve.Time;

hold on;

plot(posx,posz);

## Torque profile

torque\_pelvis = torqueout.Data(:,1);

torque\_knee = torqueout.Data(:,2);

torque\_ankle = torqueout.Data(:,3);

time\_torque = torqueout.Time;

N = length(time\_torque);

torque\_pelvis(N) = torque\_pelvis(N-1);

torque\_knee(N) = torque\_knee(N-1);

torque\_ankle(N) = torque\_ankle(N-1);

figure;

subplot(3,1,1);

plot(time\_torque,torque\_pelvis);

xlabel('time [s]');

ylabel('torque [N\*cm]');

title(sprintf('Torque at pelvis'));

subplot(3,1,2);

plot(time\_torque,torque\_knee,'r');

xlabel('time [s]');

ylabel('torque [N\*cm]');

title(sprintf('Torque at knee'));

subplot(3,1,3);

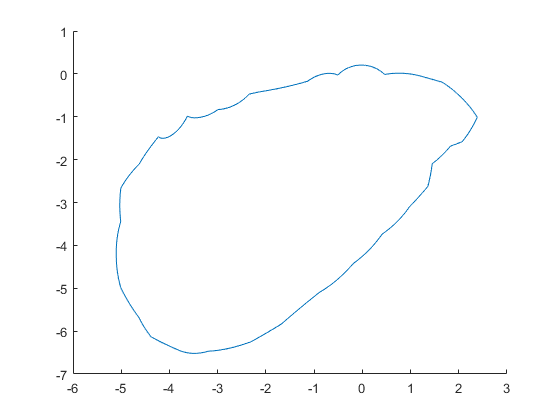
plot(time\_torque,torque\_ankle,'g');

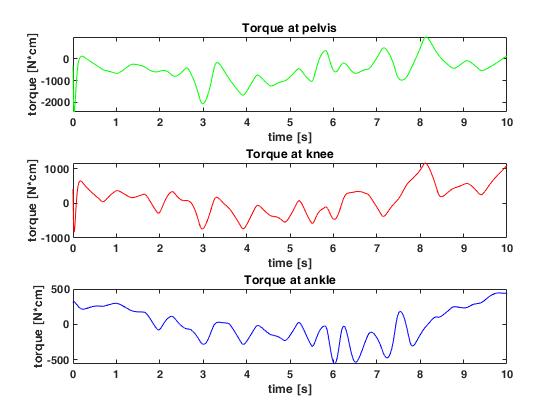
xlabel('time [s]');

ylabel('torque [N\*cm]');

title(sprintf('Torque at ankle'));

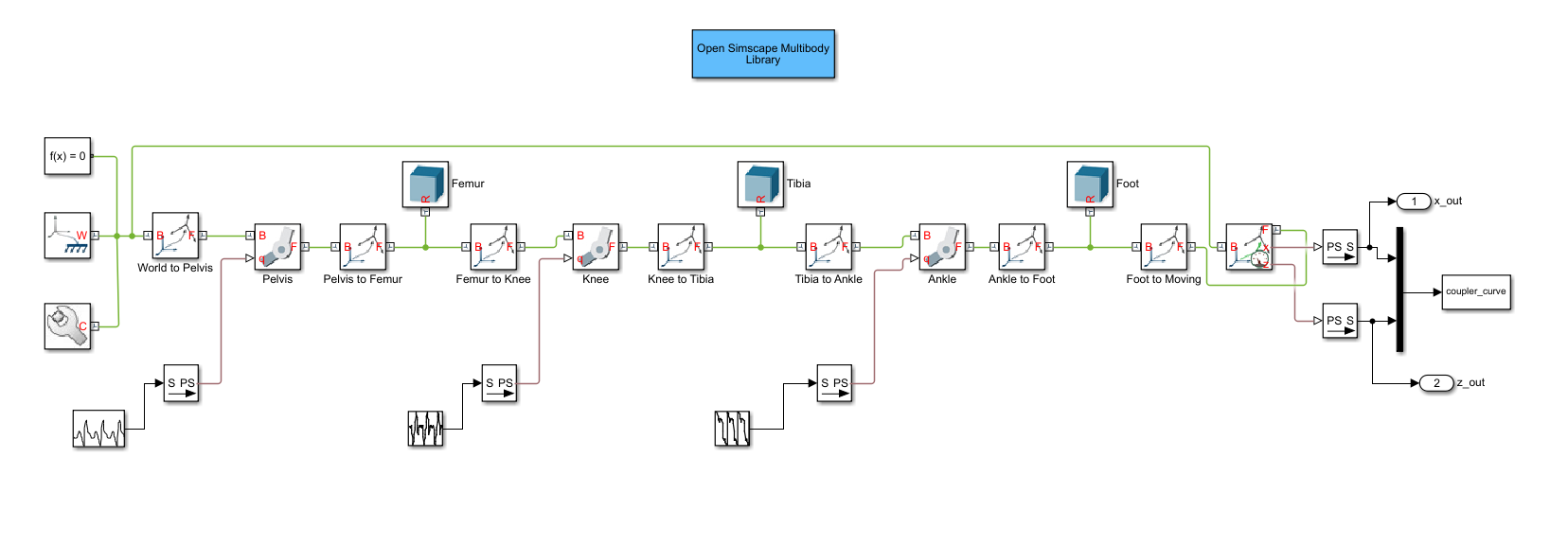
time2 = time\_torque';



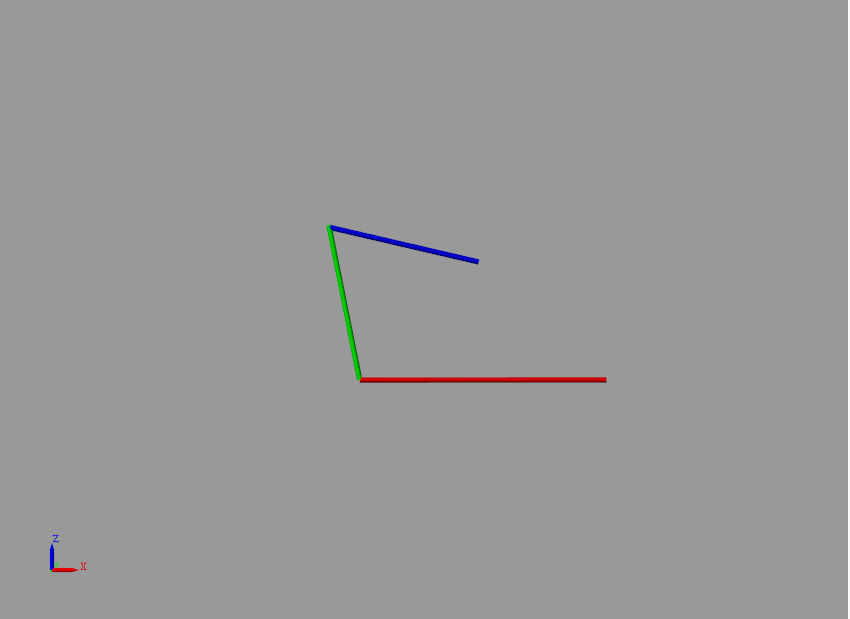


*กราฟที่8 แสดงถึงตำแหน่งของจุด foot เมื่อเทียบกับตำแหน่งของ hip ในระนาบ xy ในลักษณะวงปิด จากการsimulateในsimulink*

*กราฟที่9 แสดงถึงค่า Torque ที่เวลาต่างๆที่ pelvis, knee และ ankle*

**Simscape Multibody Model**

*รูปที่6 แสดงSimscape Multibody Model*



*รูปทึ่7 แสดงการsimulate Simscape Multibody Model*

**Part 3 : 2nd Simscape run for optimization**

**Optimization part**

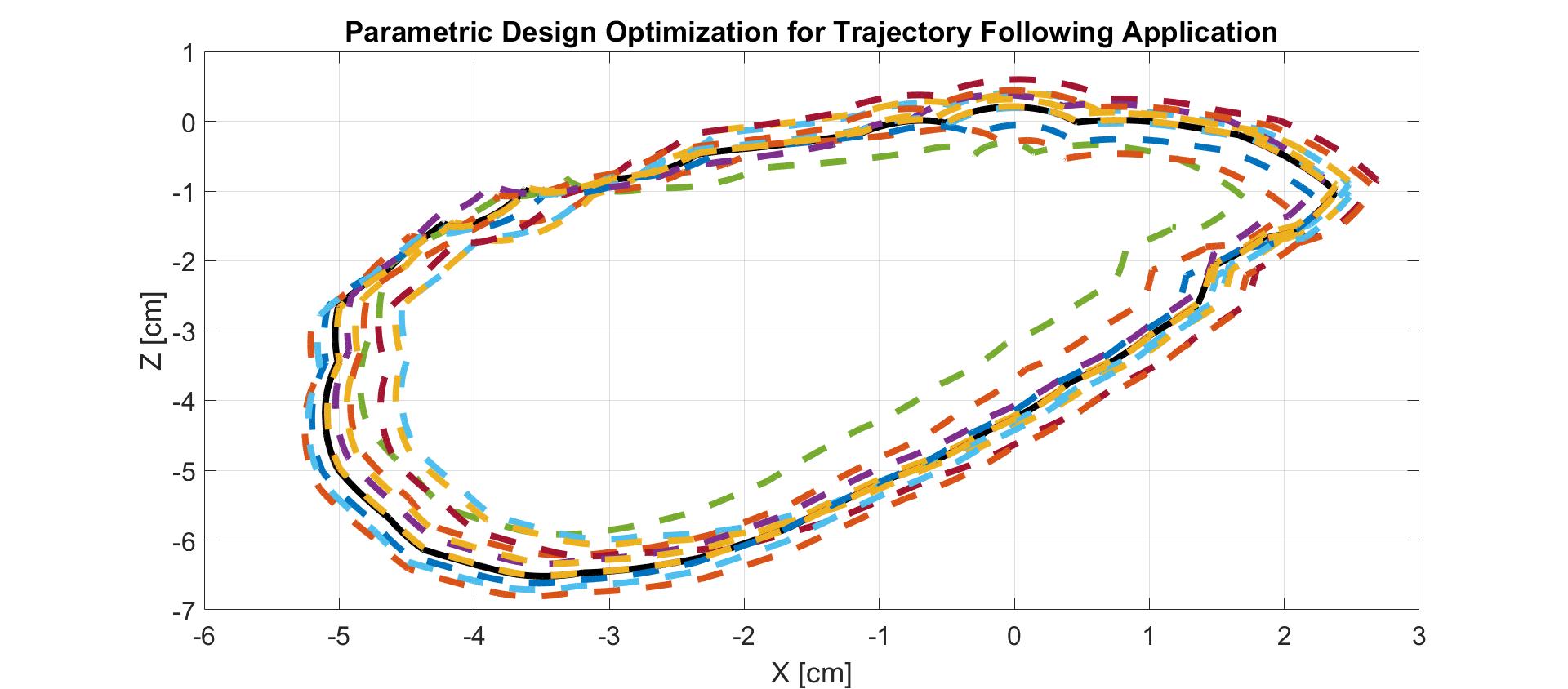
**Optimization**

clc;  
close all;  
  
X0 = [2.3 1.85 3]; %initial guess  
lower\_bound = [1.8, 1.8, 2];  
upper\_bound = [2.5, 2.5, 4];  
objective = @(X)cost(X, [x\_ref z\_ref]);  
options = optimset('Display','iter','TolX',1e-8, 'MaxIter', 50, 'PlotFcns', {'optimplotx', 'optimplotfval' }); % optimplotfval  
mdl = 'kinematic.slx';  
open(mdl);  
  
figure(101);  
set(gcf, 'Position', [1000 200 2560 1280]/2);  
plot(x\_ref, z\_ref, 'LineWidth', 4, 'Color', 'k');  
hold on;  
xlabel('X [m]');  
ylabel('Z [m]');  
title('Parametric Design Optimization for Trajectory Following Application');  
grid on;  
set(gca, 'FontSize', 16);  
  
% Solve Optimization Problem  
[X\_opt, fval] = fmincon(objective, X0, [], [], [], [], lower\_bound, upper\_bound,[], options)  
L1\_opt = X\_opt(1);  
L2\_opt = X\_opt(2);  
L3\_opt = X\_opt(3);  
  
% Display Result  
disp(sprintf('Optimal Design Parameters\nL1 = %.4f [cm]\nL2 = %.4f [cm]\nL3 = %.4f [cm]', L1\_opt, L2\_opt,L3\_opt));

**Cost function**

function J = cost(X, XZ\_ref)  
 % Given L123;  
 load('profileforopt.mat');  
 Ts = 10/3250;  
 L1 = X(1);  
 L2 = X(2);  
 L3 = X(3);  
 simopt = simset('SrcWorkspace','Current');  
 set\_param(gcs,'SimulationCommand','Update'); % Update Model  
 [~, ~, Yout]= sim('kinematic.slx', [0:1:3249]'\*Ts, simopt);  
 x = Yout(:,1);  
 z = Yout(:,2);  
 x\_ref = XZ\_ref(:,1);  
 z\_ref = XZ\_ref(:,2);  
 J = sum( (x - x\_ref).^2 + (z - z\_ref).^2 );  
  
 % Plot  
 figure(101);  
 plot(x, z, 'LineWidth', 4, 'LineStyle', '--');  
end

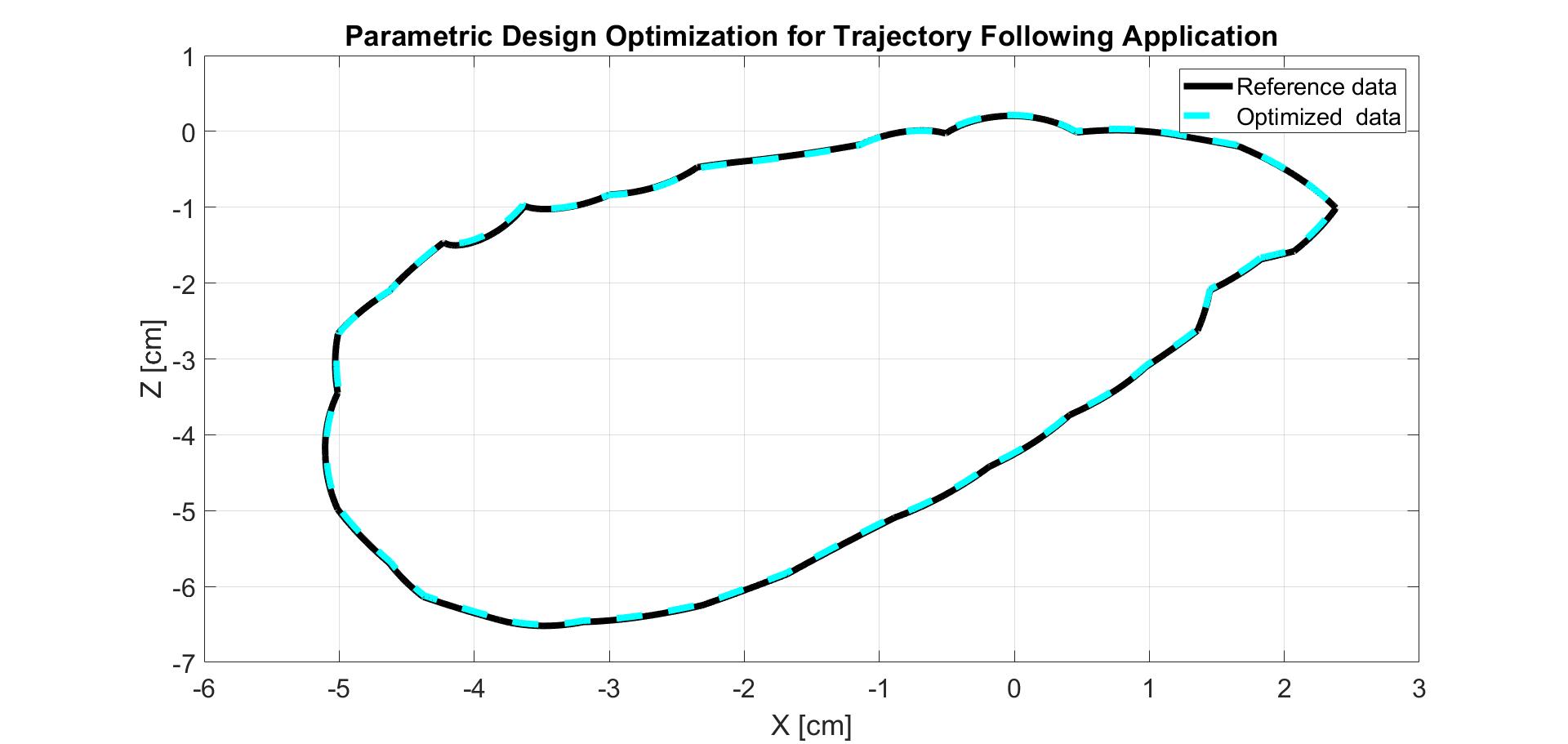
**Optimization result**

**

*กราฟที่10 แสดงการ optimize ตำแหน่งของจุด foot ซึ่ง simulate ผ่าน simscape multibody*

ซึ่งจะได้ว่า Reference Design Parameters : L1 = 2.1958 cm ,L2 = 2.2597 cm ,L3 = 3.5293 cm

Optimal Design Parameters : L1 = 2.1803 cm ,L2 = 2.2802 cm ,L3 = 3.5226 cm



*กราฟที่11 แสดงตำแหน่งของจุด foot ของกลไกซึ่งผ่านการ optimized แล้วเทียบกับ ข้อมูลอ้างอิง*

**Part 4 : Conclusion**

**Conclusion**

จากการทำ Optimization ใน Simscape พบว่าค่าความยาว L1, L2 และ L3 มีค่าเป็น 2.1803 cm, 2.2802 cm และ 3.5226 cm ตามลำดับ โดยเมื่อเทียบกับค่าความยาว L1, L2 และ L3 จริงซึ่งมีค่าเท่ากับ 2.1958 cm, 2.2597 cm และ 3.5293 cm ค่าความยาว L1, L2 และ L3 ที่ได้จากการ Optimization มีค่าคลาดเคลื่อนจากของจริงไปเพียงเล็กน้อยซึ่งก็คือ 0.706%, 3.514% และ 0.190% ตามลำดับ

**Problem and Solution**

1. การเคลื่อนที่ของขากิ้งก่าเป็นลักษณะการเคลื่อนที่ไม่อยู่บนระนาบ ทำให้เมื่อprojectการเคลื่อนที่ของขากิ้งก่าลงบนระนาบเพื่อหาความยาวขาของกิ้งก่าจะได้ไม่ตรงกับค่าความยาวขากิ้งก่าจริง

วิธีการแก้ปัญหา : เลือกการเคลื่อนที่ในระนาบ xy ซึ่งเป็นการเคลื่อนที่อยู่บนระนาบมากที่สุด

1. ค่า x, y ของ Foot ในกราฟ x-t, y-t ( Reference data ) เป็นค่า x, y เทียบกับพื้นโลก แต่ค่า x, y ที่เราต้องการคือค่า x, y เทียบกับตัวกิ้งก่า

วิธีการแก้ปัญหา : หาค่า x, y ของ Hip จากกราฟ x-t, y-t แล้วนำมาใส่สมการ x = xFoot - xHip, y = yFoot - yHip เพื่อหาค่า x, y ตามลำดับ

1. ในการคิดค่าเพื่อนำมาเป็น Input ใน Simulink โดยคิดค่า , , จาก, , ตาม Reference data จะได้ค่าค่า ออกมาเป็น complex number

สาเหตุของปัญหา : เพราะค่าและควรจะมีการเท่ากันที่จุดที่ , originมากที่สุด แต่ค่าและ ที่จุดนั้นไม่เท่ากันเพราะ ได้มาจาก Reference data แต่ได้มาจากการคำนวณ

วิธีการแก้ปัญหา : ปรับให้ค่าที่จุด, originมากที่สุด มีค่าเท่ากับค่า ที่จุดนั้น

1. วิธีการตั้งแกนของกราฟใน Reference data ไม่ตรงกับแกนของการคำนวณใน Hand Calculation

วิธีการแก้ปัญหา : ปรับแกนให้ตรงกัน

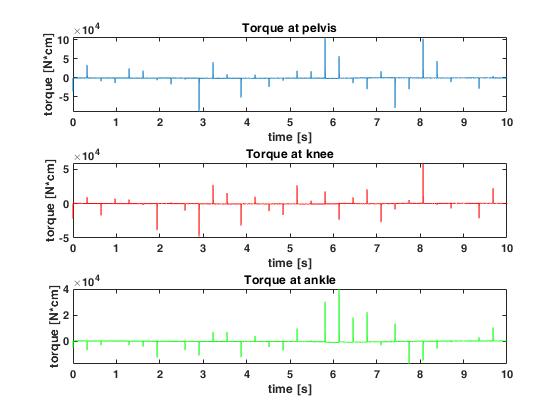
1. ภายในโปรแกรม MATLAB เมื่อแทนค่า, , ลงใน Inverse Kinematics แล้ว ค่า , , ที่ได้ มีค่าไม่ต่อเนื่องกัน ซึ่งทำให้การเคลื่อนที่ของกลไกมีลักษณะกลับไปมา

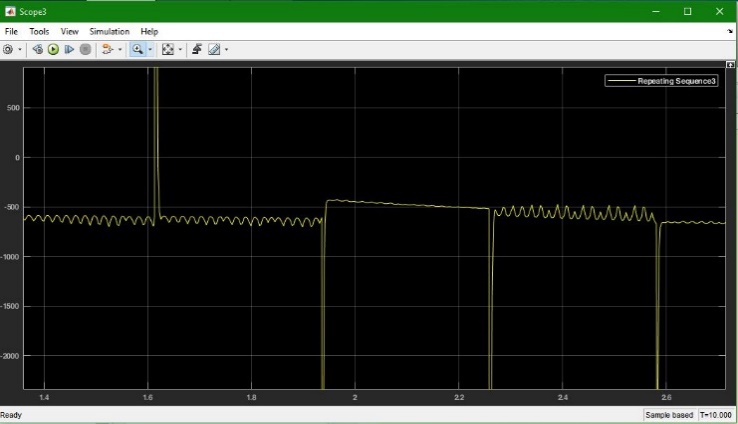
วิธีการแก้ปัญหา : ใช้ฟังก์ชั่น unwrap

1. เมื่อใส่ Input ใน Simulink เป็น Torque แล้ว จะได้การเคลื่อนที่ของขากิ้งก่าไม่ตรงกับ Reference Data

สาเหตุของปัญหา : Torqueที่ได้จากการวัดในการ Simulate การเคลื่อนที่ของขากิ้งก่าโดยใส่ Input เป็น, , มีลักษณะเป็น Abrupt change และ มี Noise ดังที่แสดงในกราฟที่12 และ รูปที่8 ตามลำดับ

วิธีการแก้ปัญหา : กลับไปใช้ , , เป็น Input เหมือนเดิม และ ใช้คำสั่ง interp1 ใน MATLAB เพื่อให้ค่า thetaที่ใส่เข้าไปเป็นinput มีจำนวนเยอะขึ้น เพื่อใน Torqueที่ได้ออกมามีลักษณะ Smooth ไม่เป็น Abrupt change ดังที่แสดงในกราฟที่9





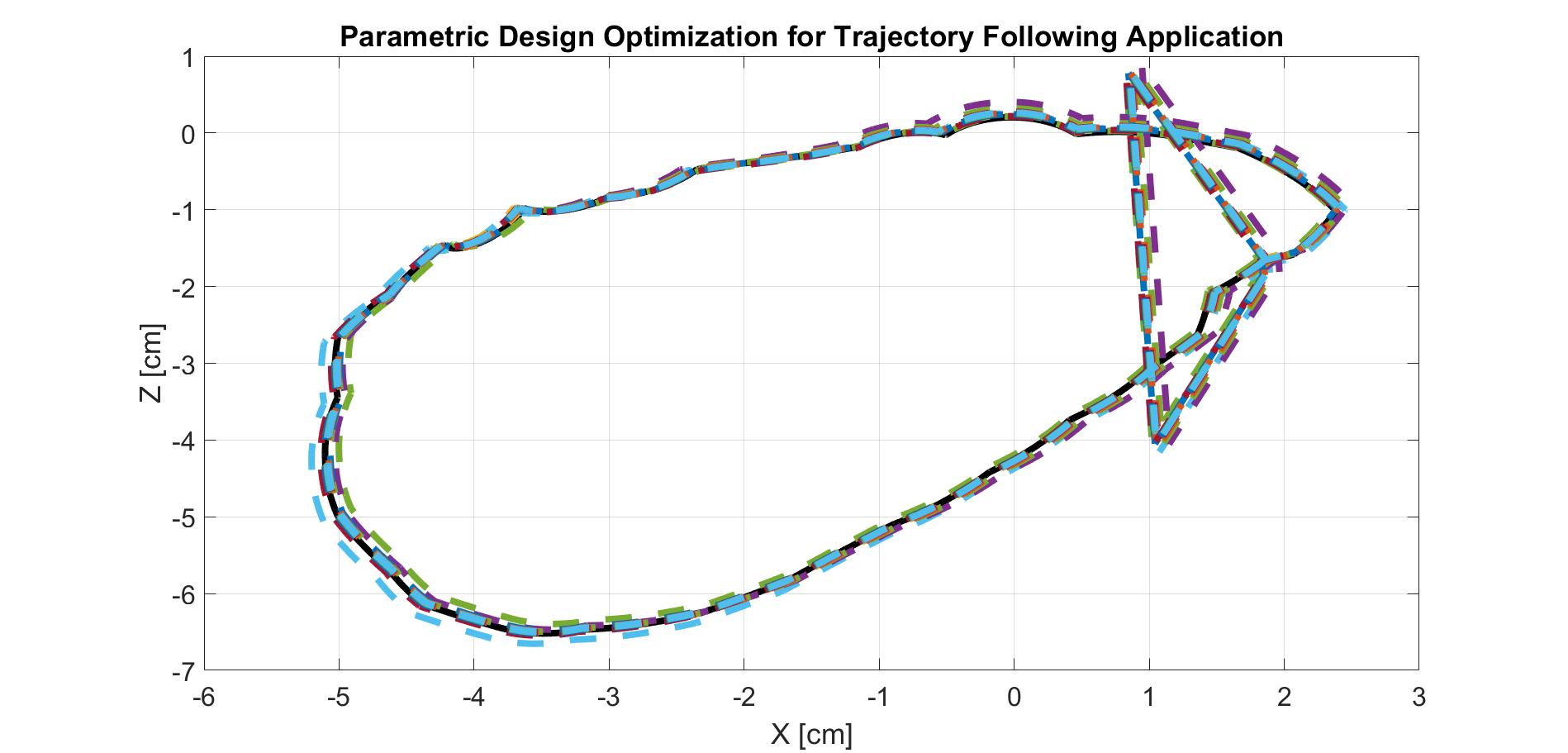
*รูปที่8 แสดง Noise ที่เกิดขึ้นในกราฟ Torque*

*กราฟที่12 แสดงTorque ณ เวลาต่างๆ ที่Pelvis, Knee, Ankle*

1. ในการทำ Optimization กราฟที่ได้เกิดรูปสามเหลี่ยมขึ้น ในการ Iteration แต่ละครั้ง

สาเหตุของปัญหา : ค่า Ts ที่ใส่ในมีความความละเอียดไม่เพียงพอ ( ทศนิยม 4 ตำแหน่ง )

วิธีการแก้ปัญหา : ใส่ค่า Ts เป็นเศษส่วน เพื่อให้ค่า Ts มีความละเอียดมากพอ



*กราฟที่ 13 แสดงถึงตำแหน่งของจุด foot เมื่อเทียบกับตำแหน่งของ hip ในระนาบ xy ในลักษณะวงปิด*

*จากการOptimization เทียบกับ กราฟที่ได้Simulink*