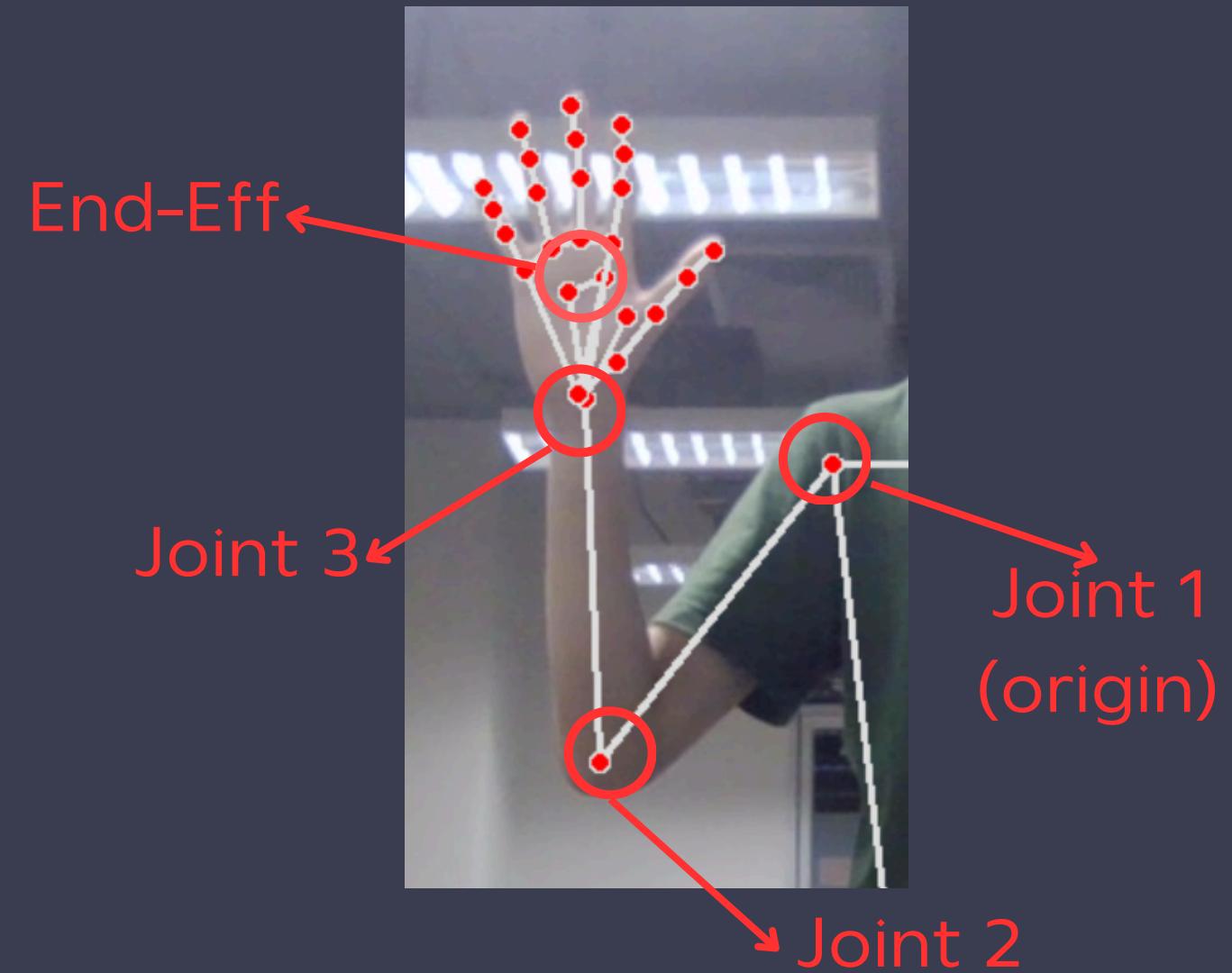
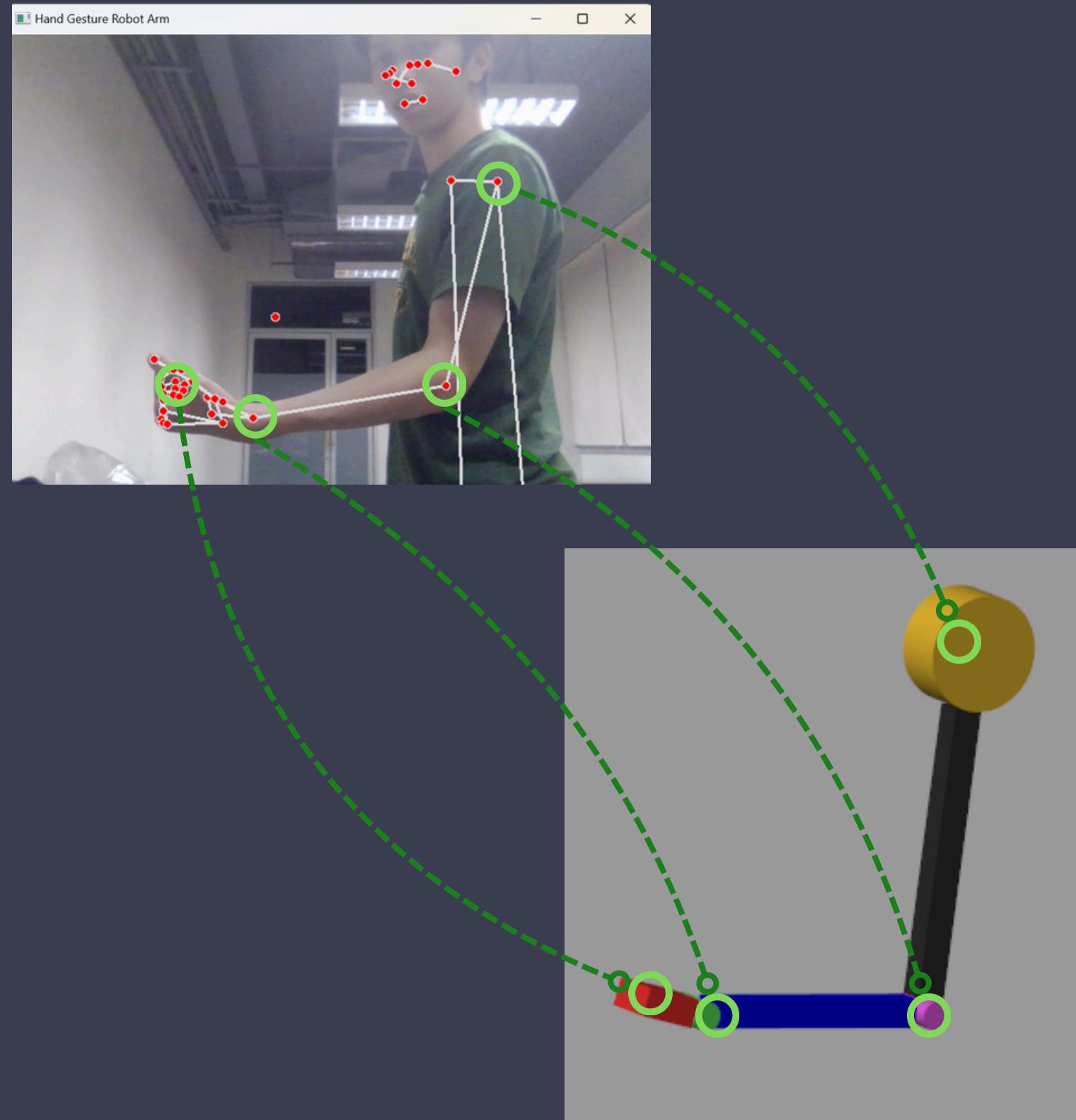


GROUP 18

TRANSFORMATION OF HUMAN MOVEMENT TO ROBOT MOTION

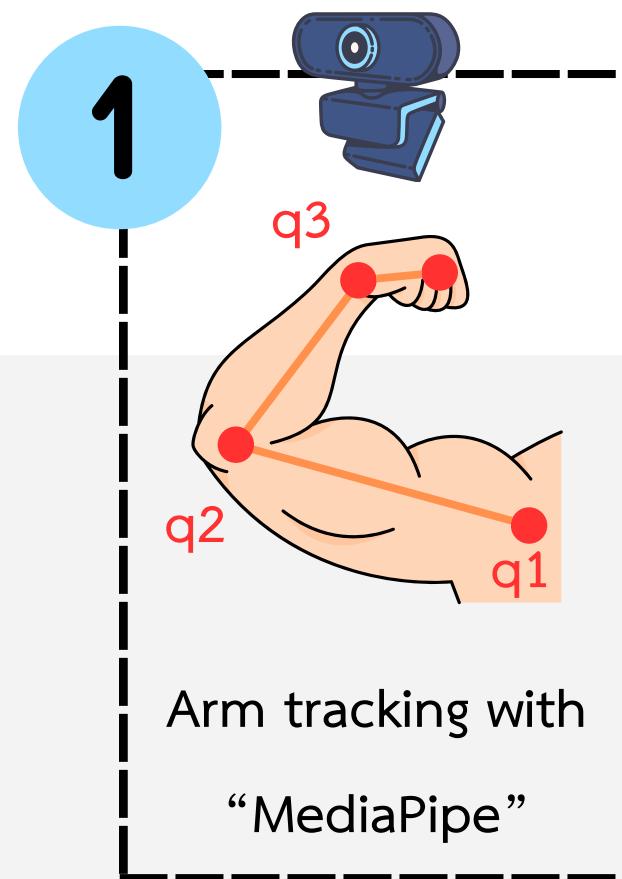
FRA333 KINEMATICS OF ROBOTICS SYSTEM

OVERVIEW

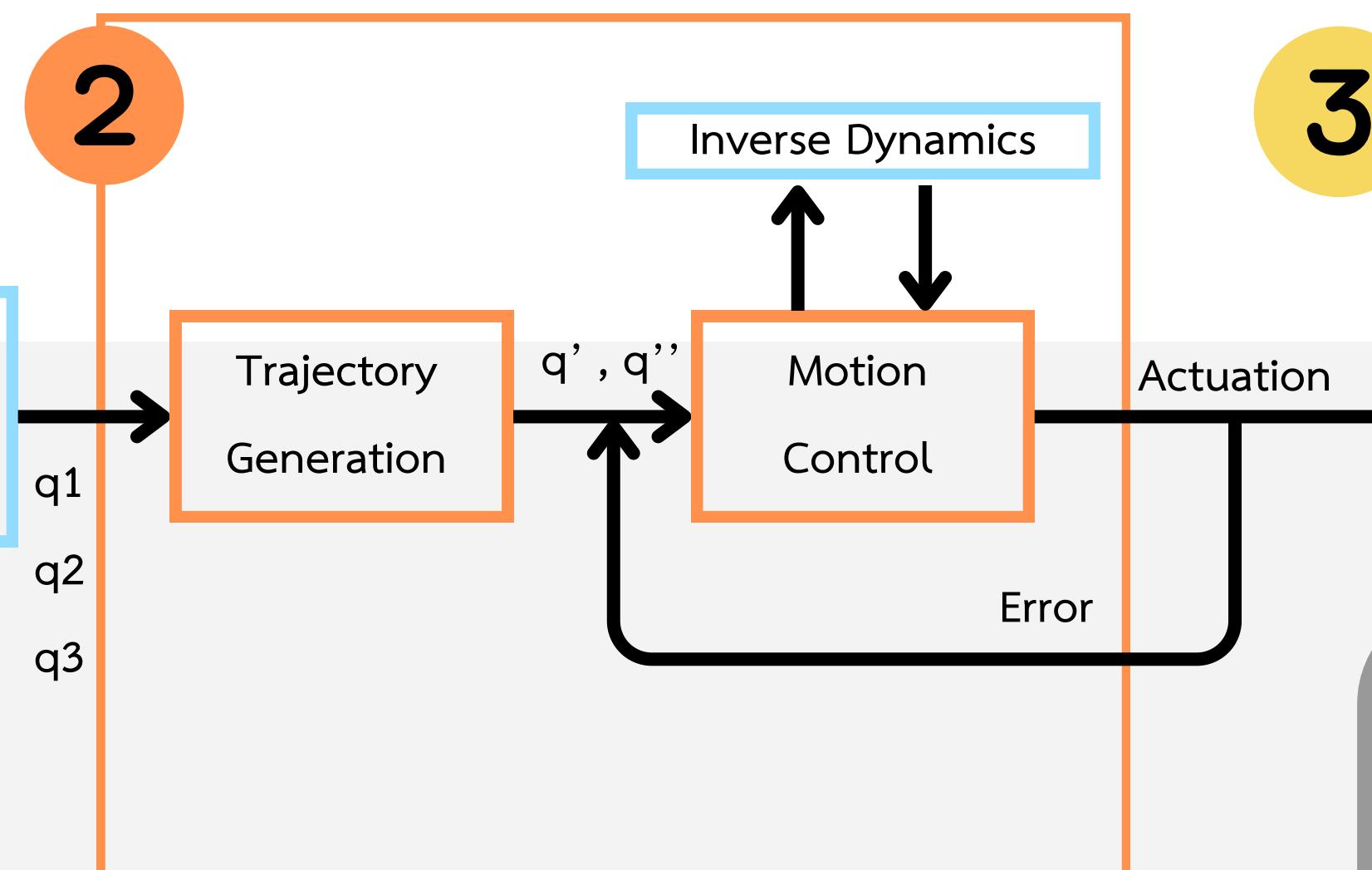


DIAGRAM

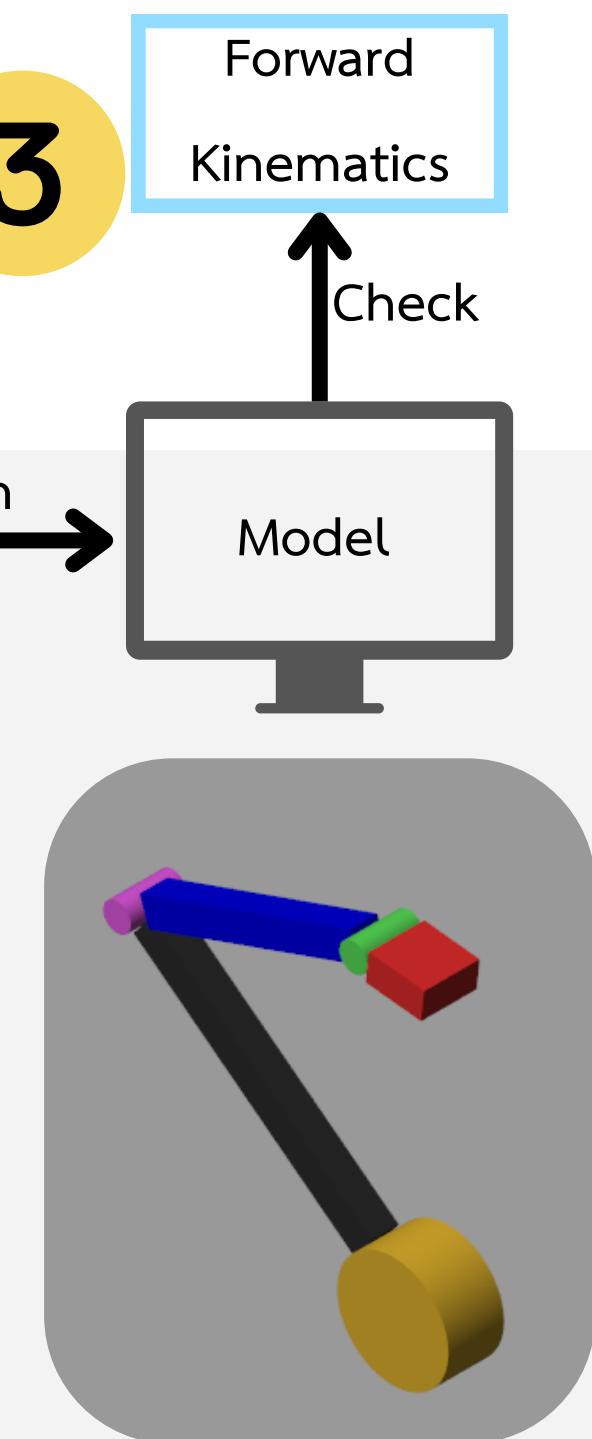
Arm Detection



Control



Modeling



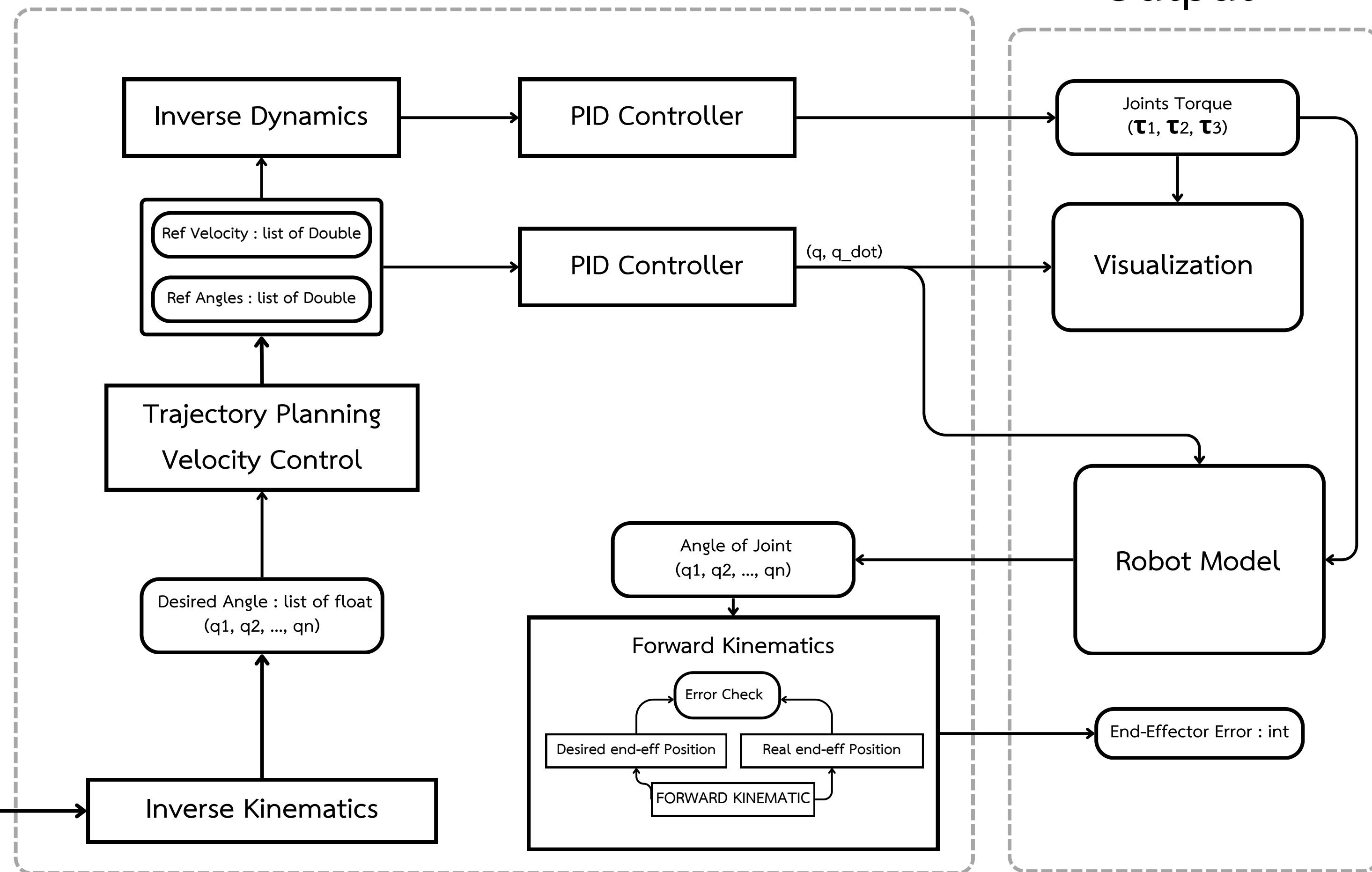
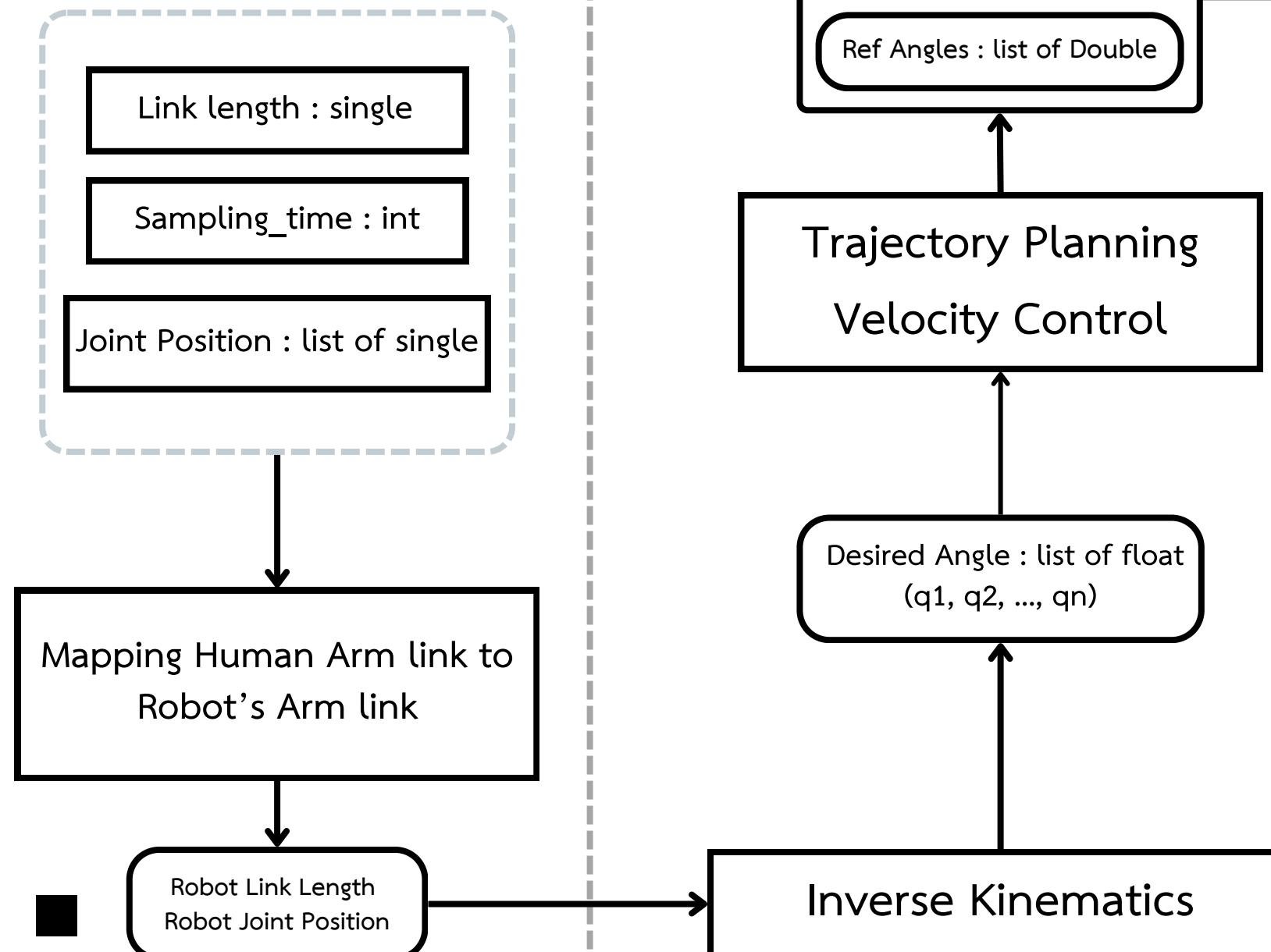
DIAGRAM

Control System

Output

INPUT

Human Motion from MediaPipe



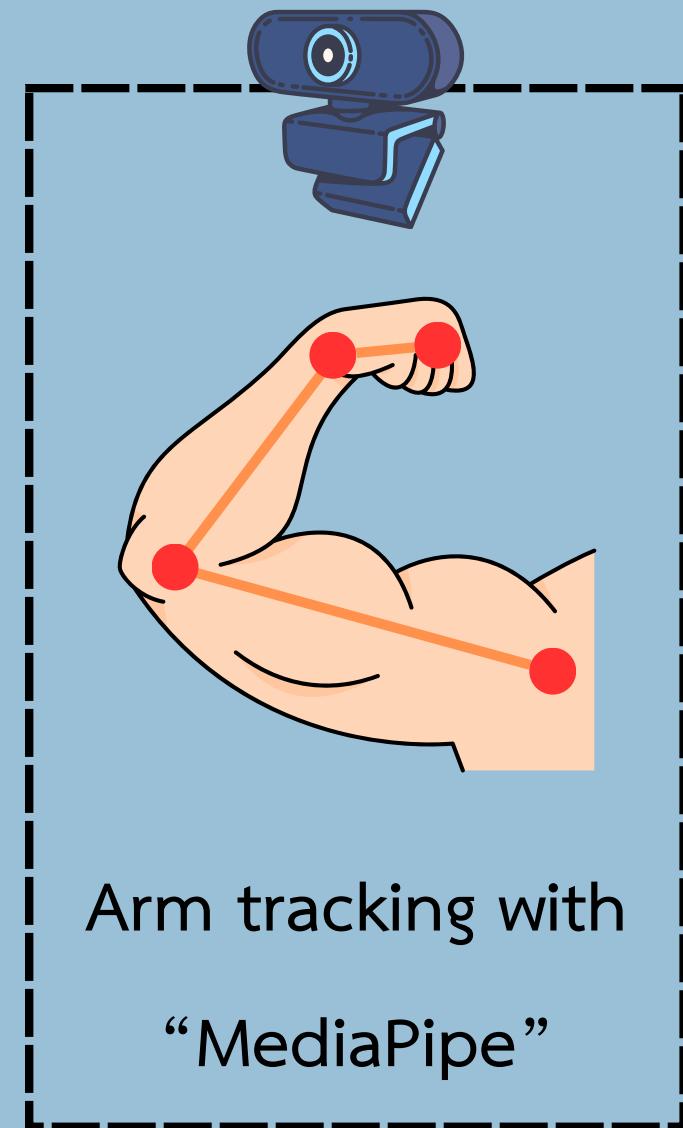
■ PROGRESS

Mediapipe	<ul style="list-style-type: none">อ่านค่าตำแหน่งข้อต่อแขนเพียง 1 ข้างรับ Input เป็นตำแหน่ง โดยไม่ได้คำนึงถึงความเร็ว หรือ ความเร่ง ในระบบจริง	100%
Inverse Kinematics	<ul style="list-style-type: none">แปลง Task space เป็น (q_1, q_2, q_3)	100%
Trajectory	<ul style="list-style-type: none">ออกแบบให้กราฟองศาการเคลื่อนที่มีความราบลื่น	100%
Inverse Dynamics	<ul style="list-style-type: none">คำนวณ Effort Torque จากน้ำหนัก Link (ขาด validate)	70%
Modeling	<ul style="list-style-type: none">สร้างโมเดลจำลอง 3 Dof แบบ Planar	75%
Forward Kinematics	<ul style="list-style-type: none">แปลงค่าจาก Configuration space เป็น Task space	100%

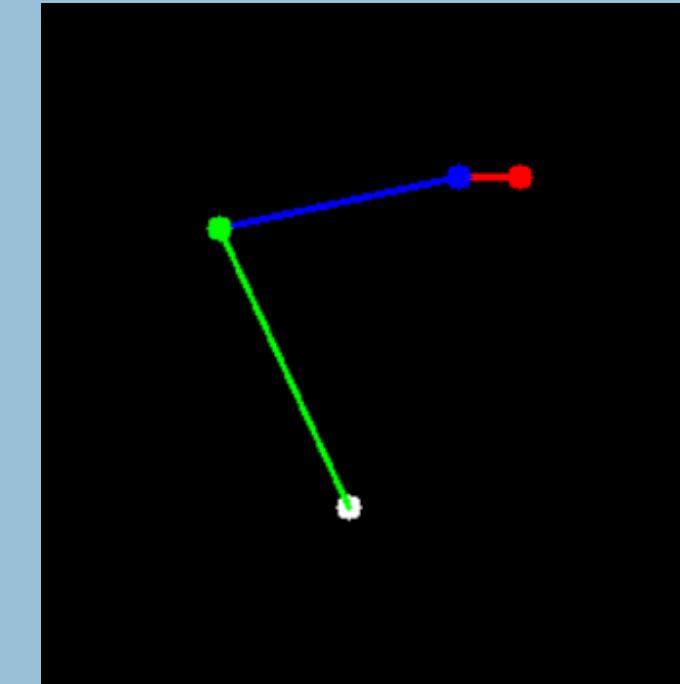
MEDIAPIPE

Overview

Input: Motion Capture



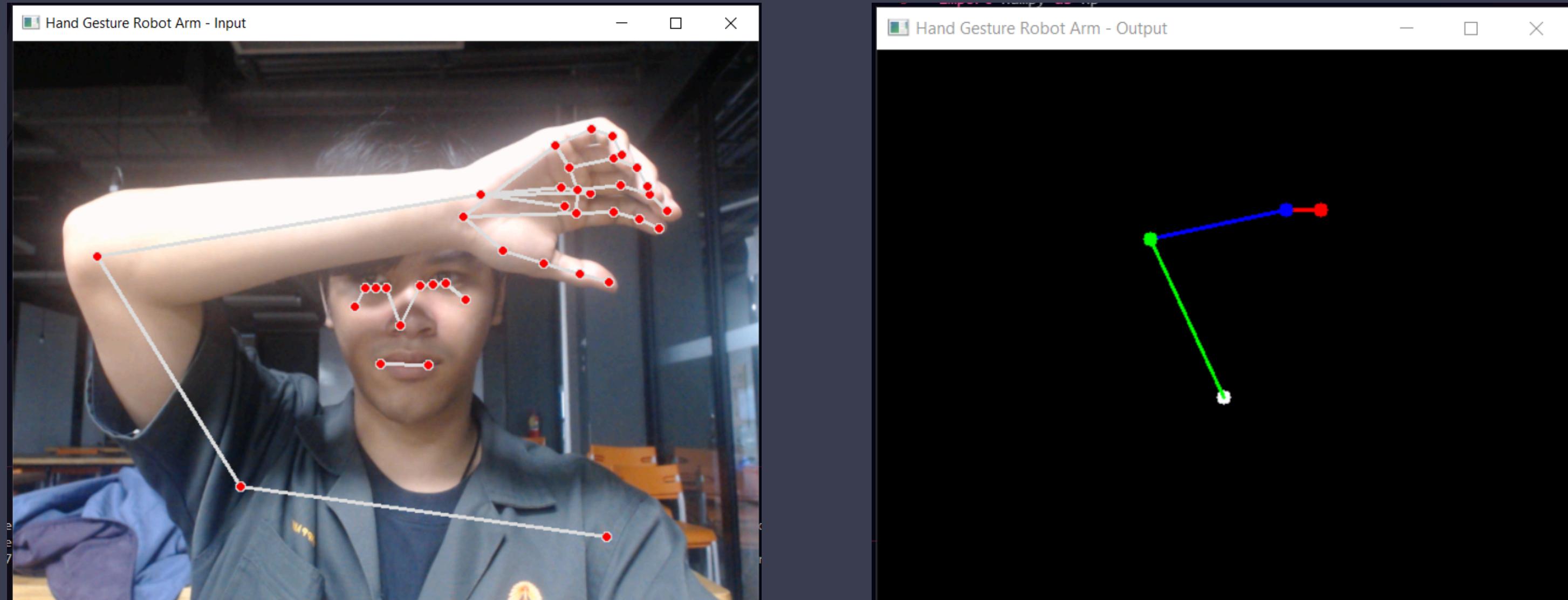
Output: Preprocessed
Position values



นำตำแหน่งในแกน X,Y กีผ่านการ Preprocess
ส่งไปยัง Matlab

MEDIAPIPE

Model Mapping Example



Mapping Human Arm into Robot Arm length
(Normalize & Rescaling)

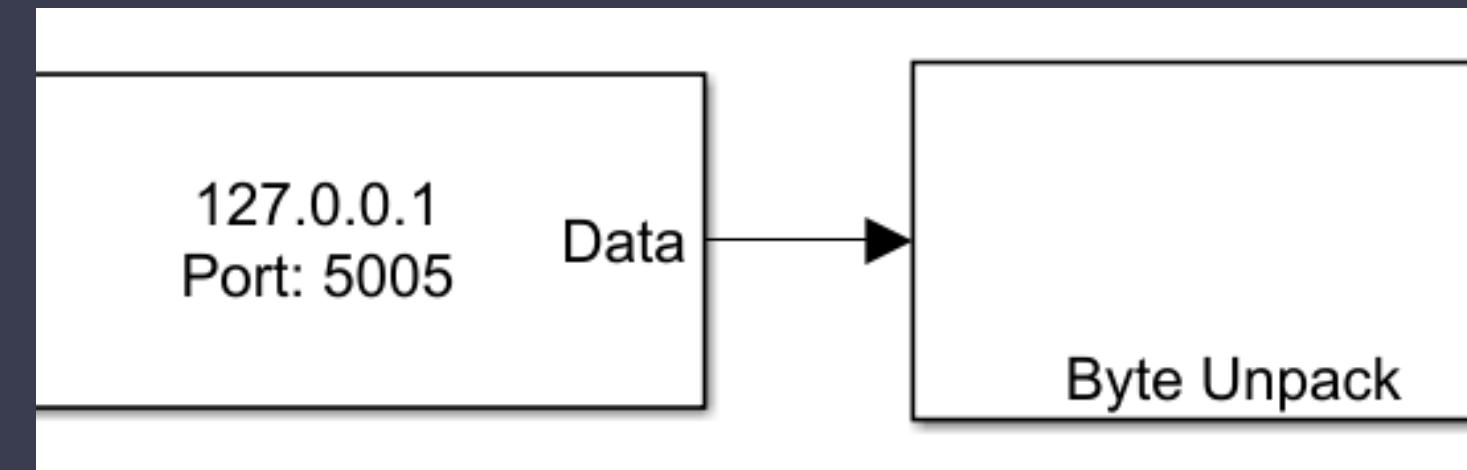
MEDIAPIPE

Transmission to Matlab/Simulink

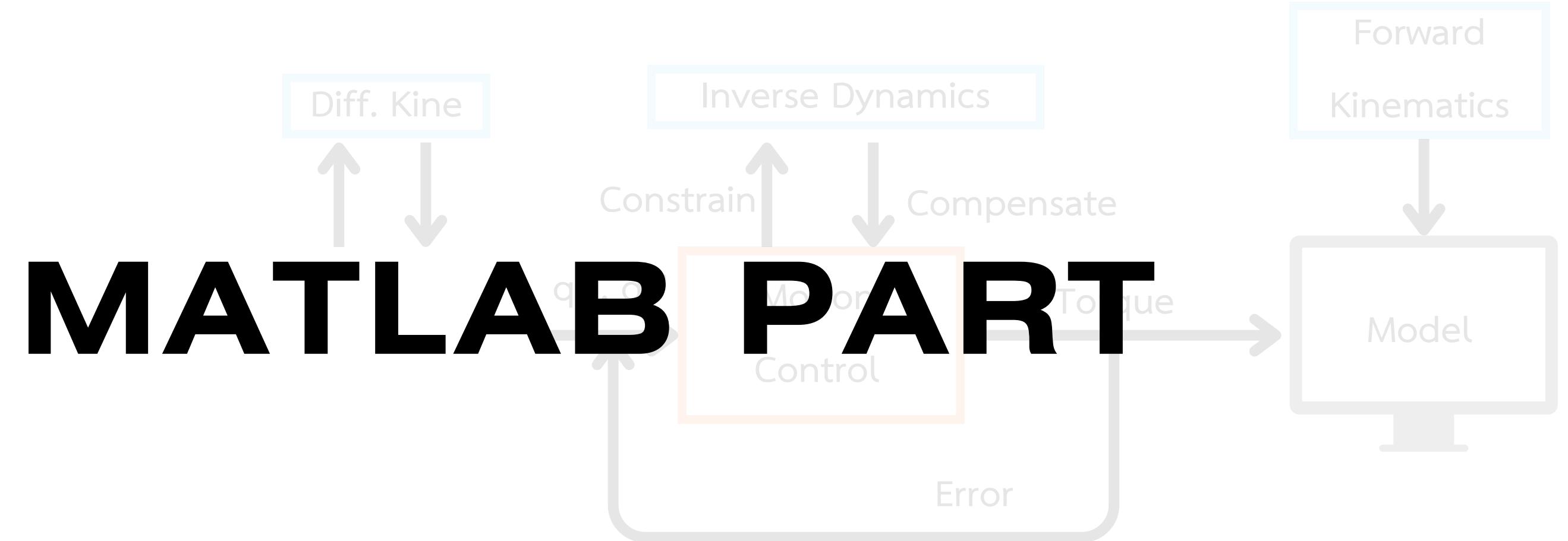
Transmit

```
# UDP transmission
data = struct.pack('<ffffffffffff',
                    0, 0, 0, # Shoulder (relative origin)
                    elbow_coords[0], elbow_coords[1], elbow_coords[2],
                    wrist_coords[0], wrist_coords[1], wrist_coords[2],
                    endf_coords[0], endf_coords[1], endf_coords[2])
sock.sendto(data, (UDP_IP, UDP_PORT))
#print(f"End Effector: X={endf_coords[0]}, Y={endf_coords[1]}, Z={endf_c
last_transmission_time = current_time
```

Receive



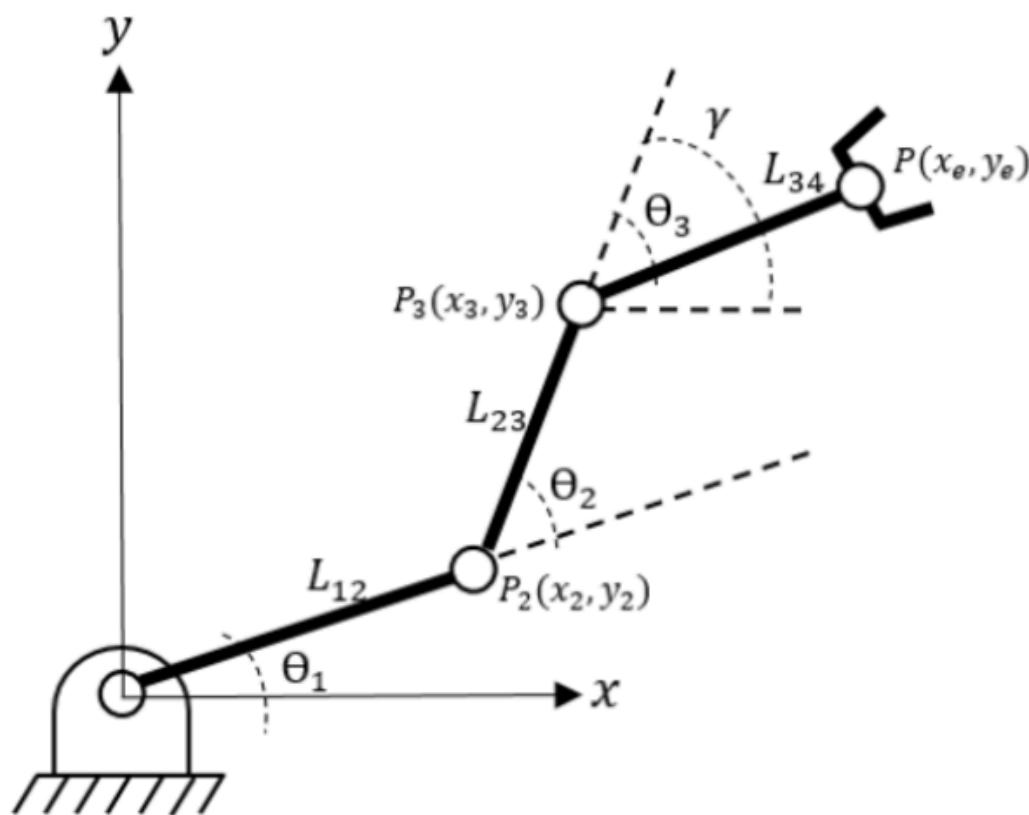
Transmission delay 0.1 sec



Inverse Kinematics

หาค่ามุมข้อต่อ จาก

- ตำแหน่งปลายแขน (x, y)
- กิจทางของปลายแขน (ϕ)



- ตำแหน่งข้อมือ

$$x_w = x - L_3 \cos(\phi)$$

$$y_w = y - L_3 \sin(\phi)$$

- ระยะดึงข้อมือ

$$r = \sqrt{x_w^2 + y_w^2}$$

- ตรวจสอบความสามารถในการเข้าถึง

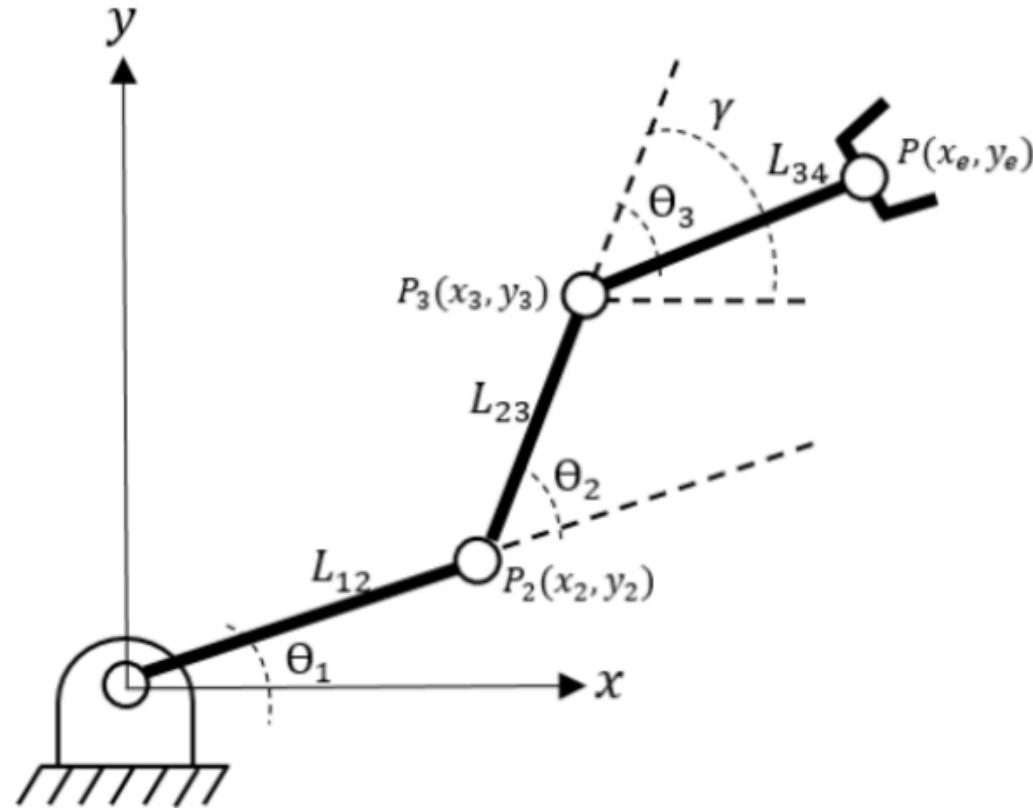
$$r > L_1 + L_2 \quad \text{เกินระยะที่แขนไปถึงได้}$$

$$r < |L_1 - L_2| \quad \text{ใกล้เกินไป ไม่สามารถเข้าถึงได้}$$

Inverse Kinematics

หาค่ามุมข้อต่อ จาก

- ตำแหน่งปลายแขน (x, y)
- ทิศทางของปลายแขน (ϕ)



• คำนวณ θ_2

$$\cos(\theta_2) = \frac{r^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$\sin(\theta_2) = \sqrt{1 - \cos^2(\theta_2)}$$

$$\theta_2 = \tan^{-1} \left(\frac{\sin(\theta_2)}{\cos(\theta_2)} \right)$$

• คำนวณ θ_3

$$\theta_3 = \phi - (\theta_1 + \theta_2)$$

• คำนวณ θ_1

$$\alpha = \tan^{-1} \left(\frac{y_w}{x_w} \right)$$

$$\beta = \tan^{-1} \left(\frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)} \right)$$

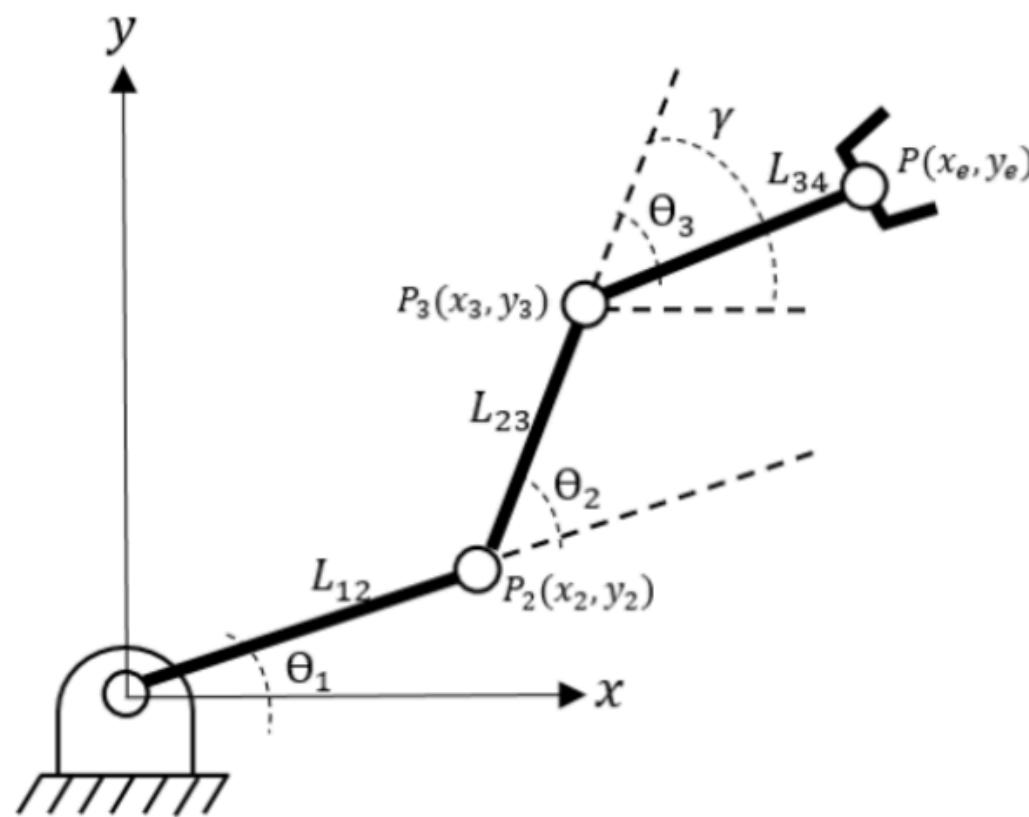
$$\theta_1 = \alpha - \beta$$

Output : theta 1,2,3

Inverse Kinematics

หาค่ามุมข้อต่อ จาก

- ตำแหน่งปลายแขน (x, y)
- ทิศทางของปลายแขน (ϕ)



- การตรวจสอบ Singularities

คำนวณ Determinant

- Singularities เกิดขึ้นเมื่อ Determinant ของ Jacobian matrix = 0

$$\det(J) = L_1 L_2 \sin(\theta_2)$$

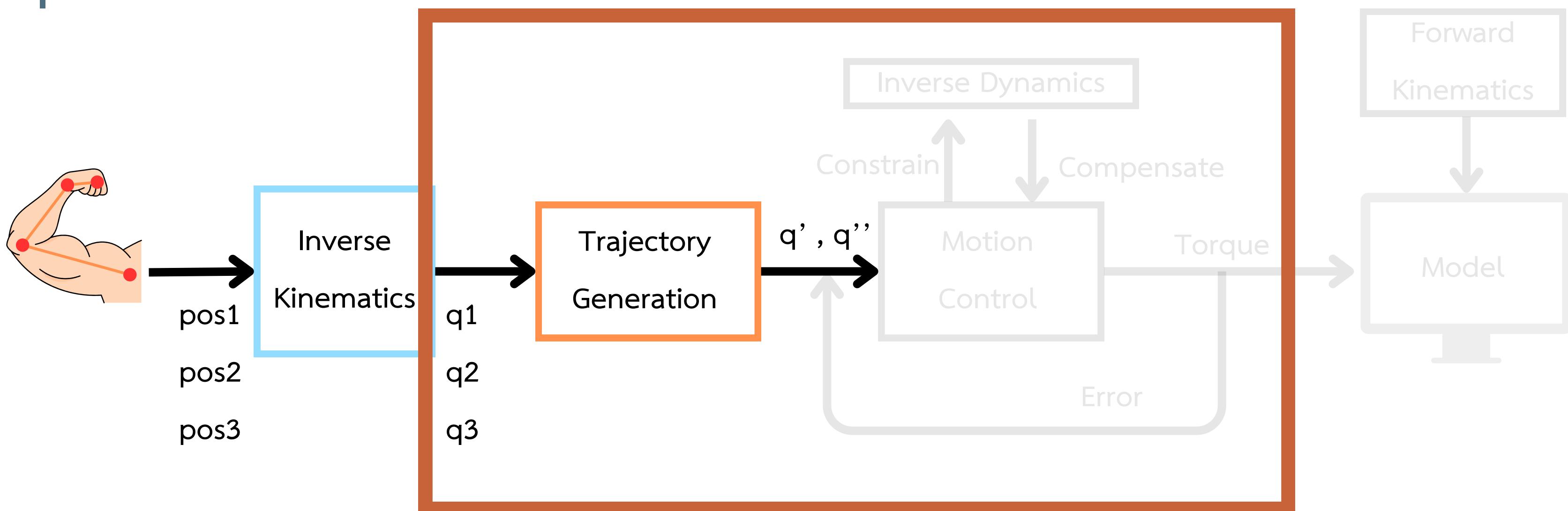
เงื่อนไขของ Singularities

- เงื่อนไขที่ 1: $\sin(\theta_2) = 0$
 - กรณีที่เกิดขึ้น
 - $\theta_2=0$: ข้อต่อทั้งสองยึดตรง
 - $\theta_2=\pi$: ข้อต่อทั้งสองพับซ้อนกัน

Output : theta 1,2,3

MATLAB PART (1/2)

TRAJECTORY & CONTROL



■ TRAJECTORY GENERATION

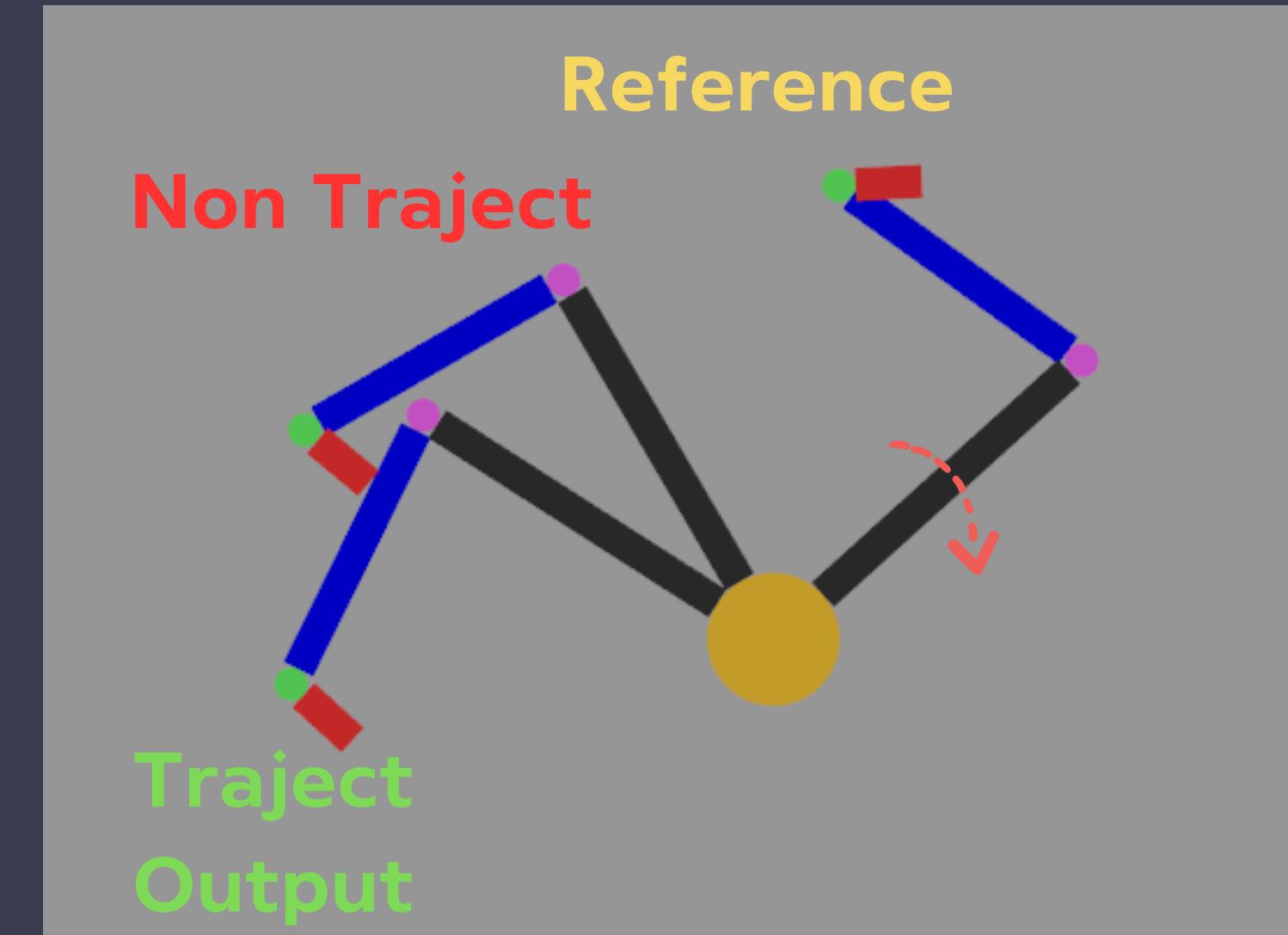
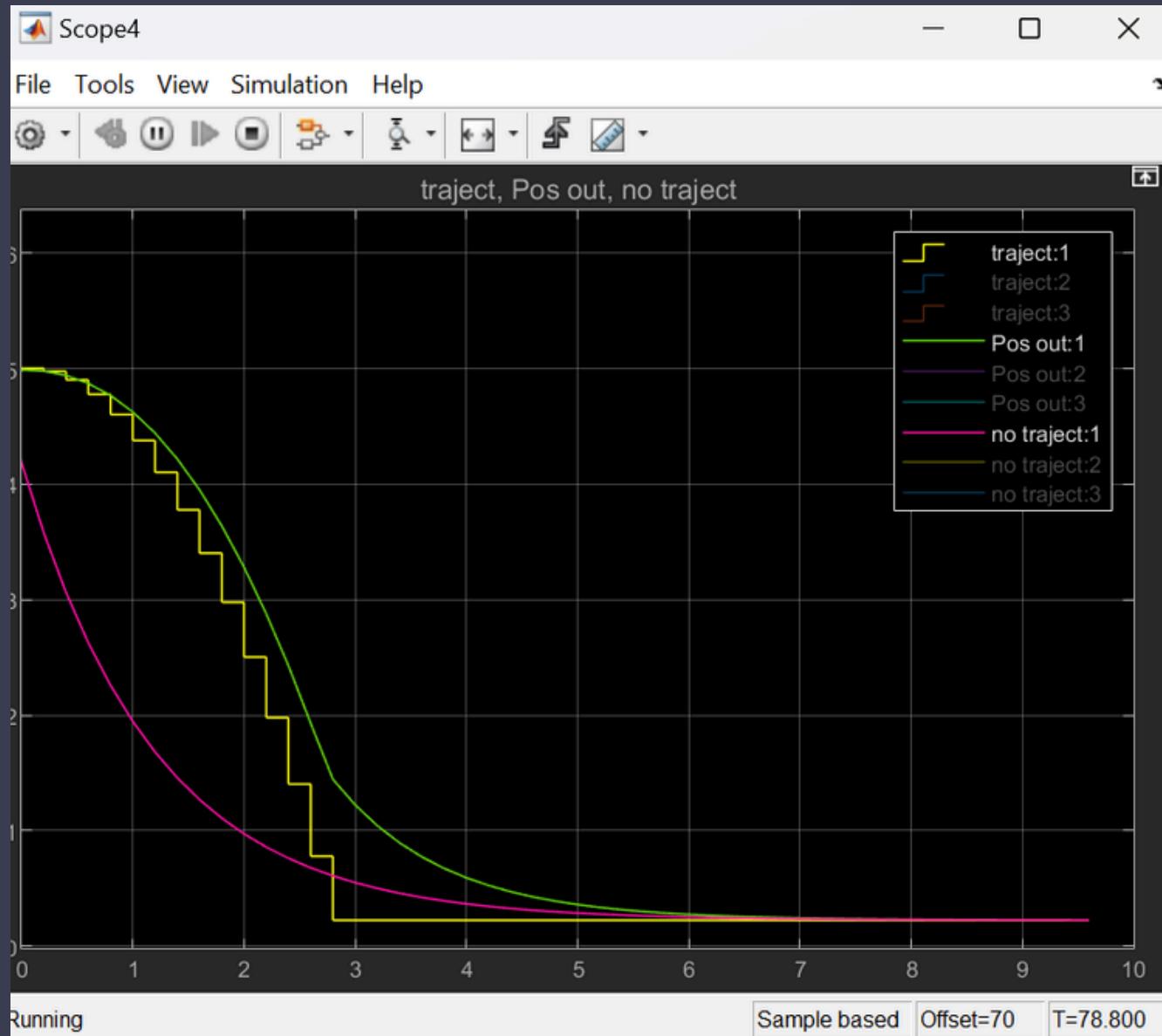
ใช้ **Trapezoidal Trajectory** ผสมกับ **Smoothing Trajectory**
เพื่อให้เข้ากับ Input ที่ update แบบ Real-time

โดยกำหนด

- **Max Angular Velocity = 0.5 rad/s**
- **Max Angular Accel = 3.57 rad^2/s**

โดยคำนวณให้ใกล้เคียงกับ แทนมนุษย์ก้าวไป ($v = wr$)

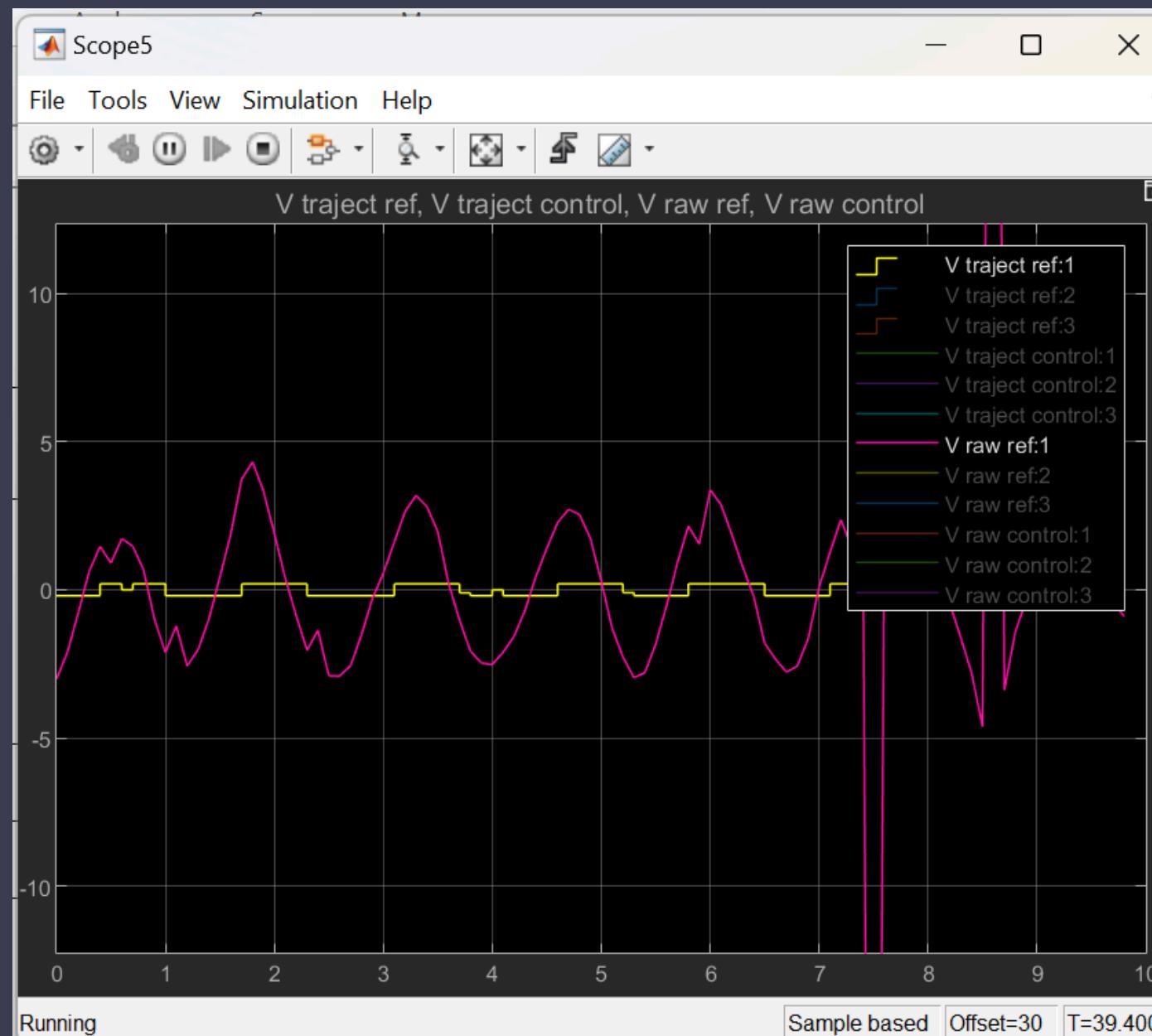
TRAJECTORY TESTING



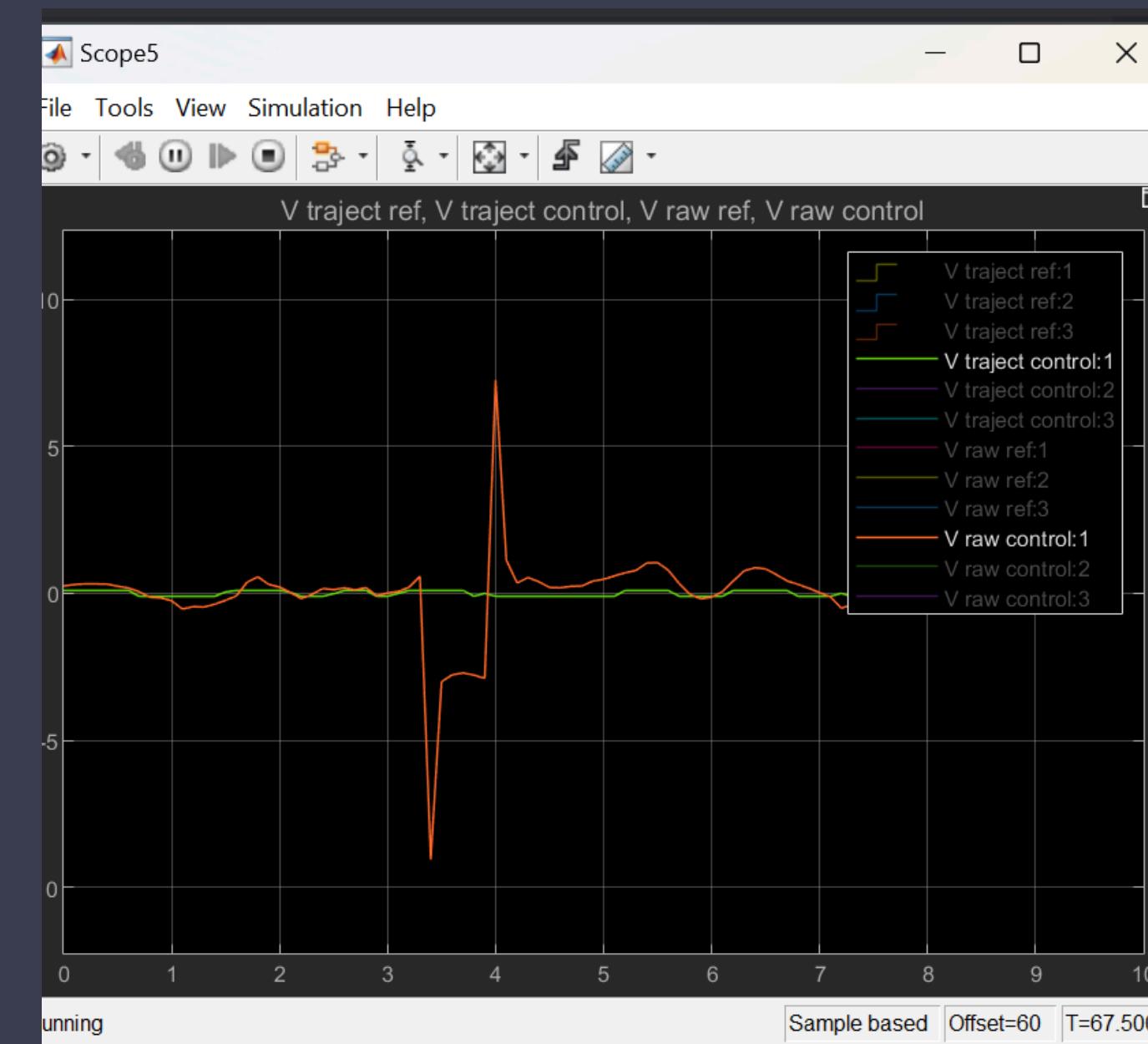
Traject Ref Vs Traject Vs No Traject

TRAJECTORY PERFORMANCE

RAW VELOCITY VS TRAJECT VELOCITY



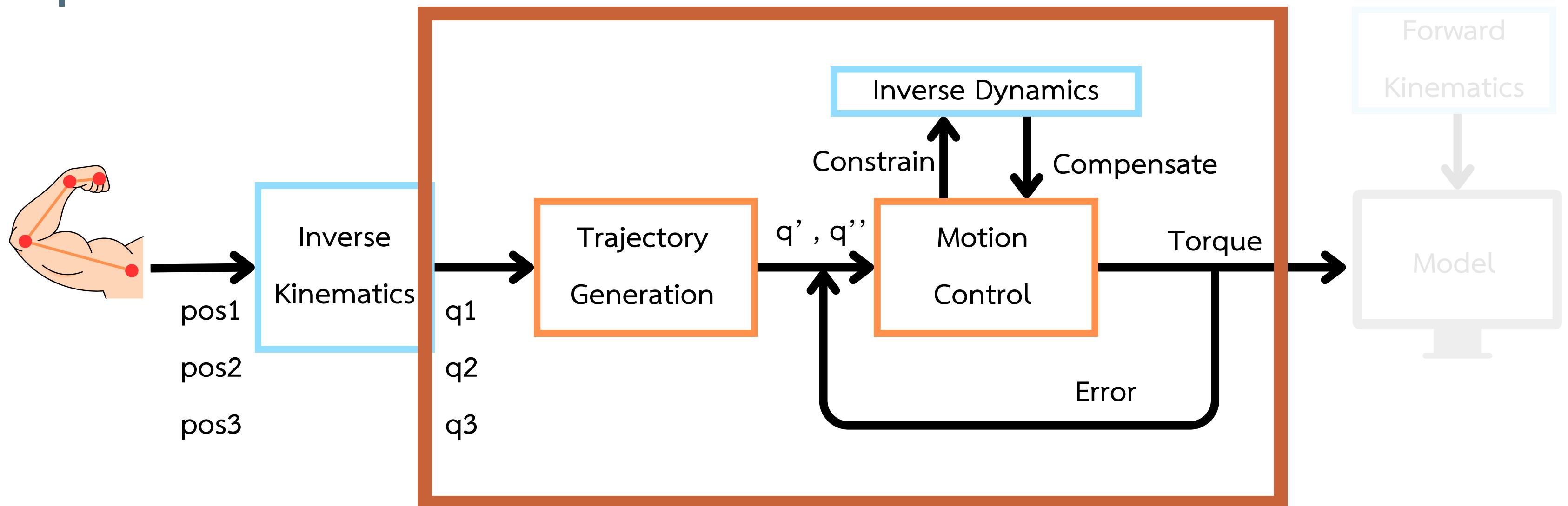
INPUT (NO PID)



OUTPUT (PID)

MATLAB PART (2/2)

DYNAMICS



Inverse Dynamics

สมการ Inverse Dynamic ของหุน 3 DOF Planar : $M(q)\ddot{q} + B(q, \dot{q})\dot{q} + g(q) = \tau$

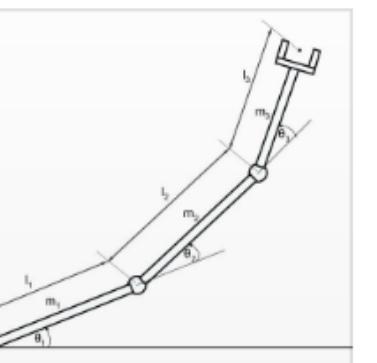
Conference Paper Full-text available

Redundancy control of robot manipulators using task priority

February 2007

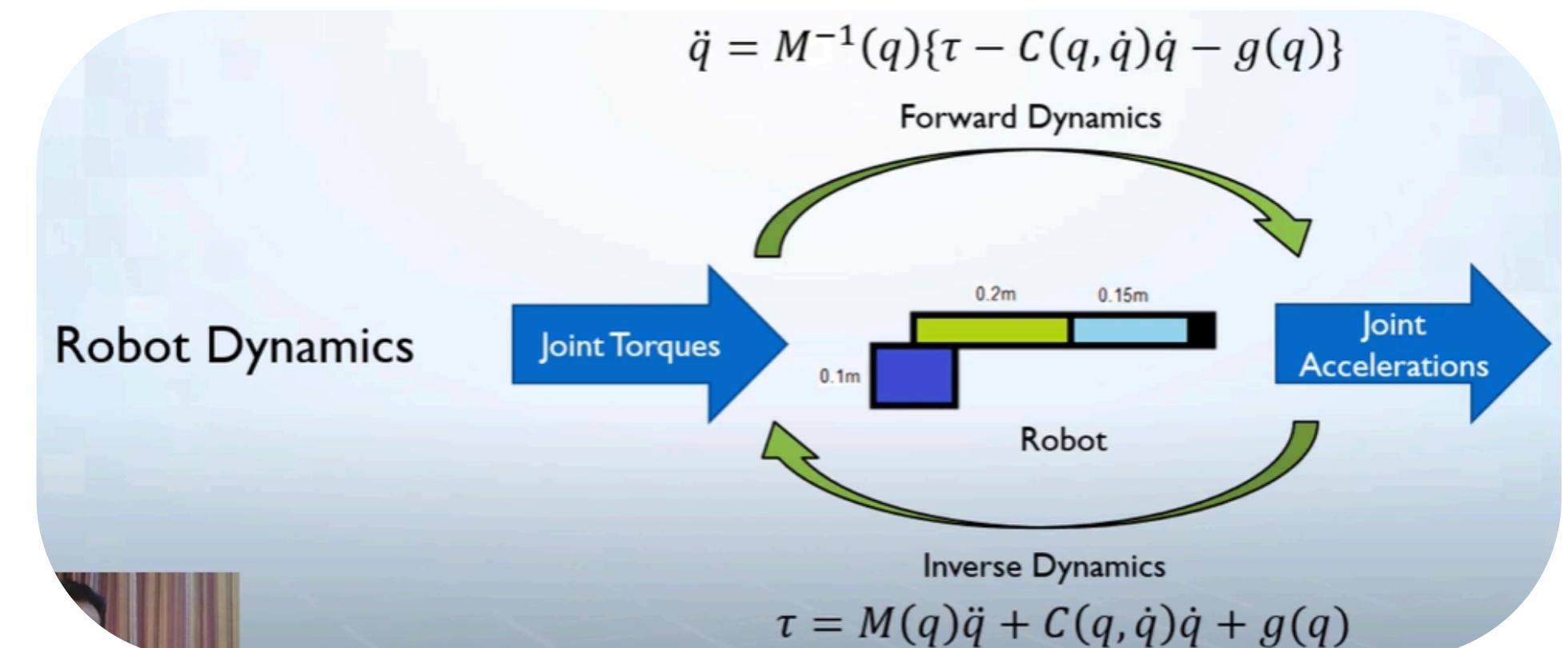
Conference: Proceedings of International Conference on Advances in Optimization of Dynamical Systems (ACODS'2007) · At: IISC Bangalore
Volume: pp. 72-78

Ashish Singla · Sandeep Kumar · Prasad Kulkarni · Bhaskar



https://www.researchgate.net/publication/281374317_Redundancy_control_of_robot_manipulators_using_task_priority

https://youtu.be/4XqWtfk4qwE?si=nyO0y4x6Z_eJ5w0w



Output: แปลงข้อมูลการควบคุมเป็นแรงบิด (Torque)
เพื่อควบคุมการเคลื่อนไหวของหุนยนต์

Problem with Dynamics

- Inverse Dynamics ด้วย Model จะได้ผลลัพธ์มาเพียงตัวเลข **Effort Torque**
- แต่เนื่องจากปัญหาเกิด **Singularity** ภายใน Model ขณะสั่งการด้วย Torque ซึ่งไม่สามารถแก้ไขได้ด้วยการ Control หรือ อีก Singularity ก่อนหน้า
- การทดสอบด้วย Model จึงมีเพียงการ Feed ค่า Trajectory Position เท่านั้น

Forward Kinematics (FK)

for Error Validation

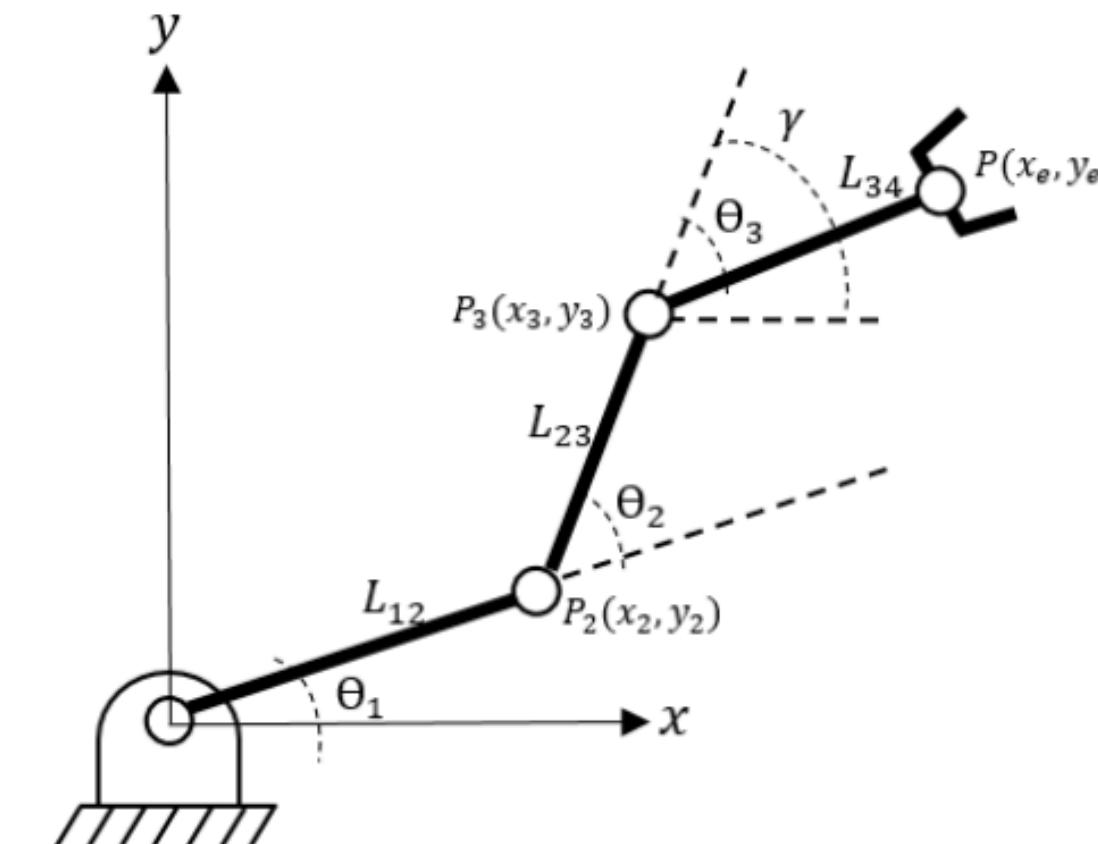
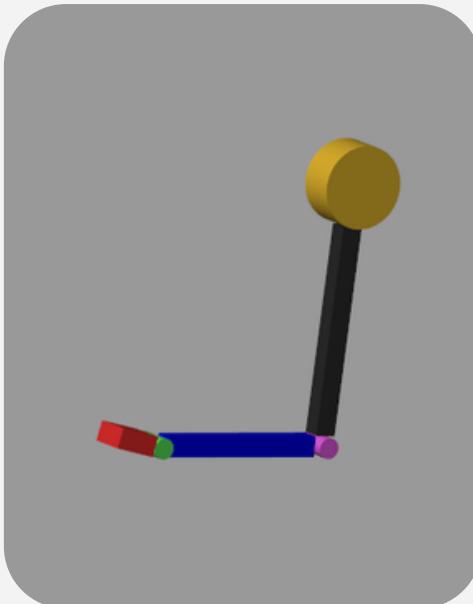
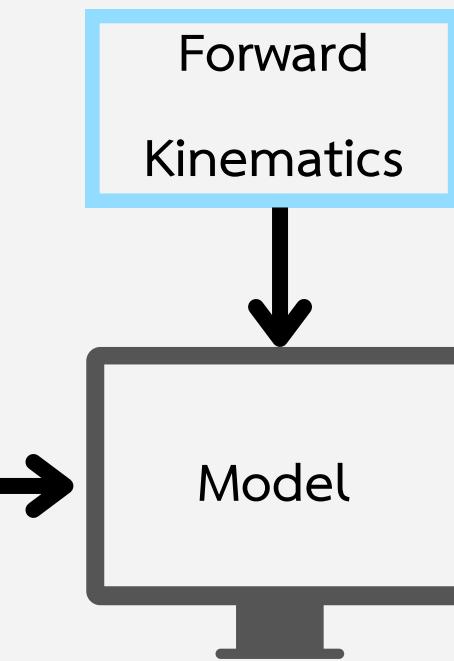


Fig. 1: 3R Planar Manipulator

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

MODEL DEMO

Check ReadMe file for
Detailed Modeling & Demo



GROUP 18

THANK YOU

ศิริประภา อุปภาค 65340500053

อพฟาน อัครามีน 65340500063

ชุติบันก์ พิพัฒนกิวงศ์ 65340500068

สิงคปรับบันก์ สรรณ์คุณแก้ว 65340500076