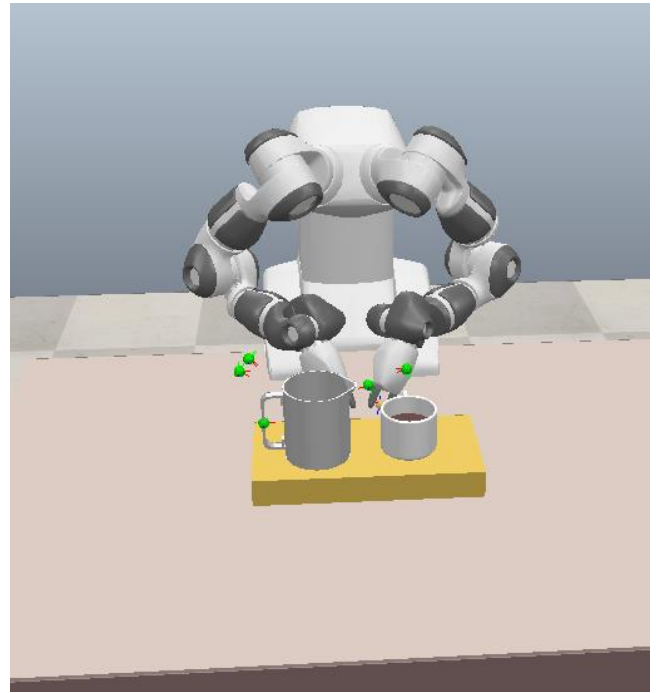


Automatic Coffee Maker Robot Arm

Trần Thanh Tuấn, Vũ Đức Hiếu, Nguyễn Văn Diễn

Ngành kỹ thuật Robot, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

Abstract: Hiện nay đang trong thời kỳ phát triển mạnh mẽ của khoa học kỹ thuật và công nghệ. Việc một số ngành nghề đang dần được thay thế bởi robot do khả năng, thời gian hoạt động lâu hơn cũng như đạt được hiệu suất và kết quả đáng kể so với con người. Tận dụng sự cung cấp trực tiếp nguồn robot từ ABB với mô hình sản xuất nhỏ và linh hoạt, thì công nghệ Robot ABB được sử dụng trong các ứng dụng lắp ráp và xử lý vật liệu. Giới thiệu robot cộng tác đầu tiên trên thế giới (Yumi)- được sử dụng vào các công việc cần độ chính xác cao cũng như hiệu suất lớn. Dự án sẽ trình bày những nguyên lý cơ bản của cánh tay máy đồng thời tiến hành các thực nghiệm robot và ứng dụng của robot để đánh giá tính khả dụng của nền tảng robot Yumi – ABB. Trong bài cáo này nhóm em xin trình bày về nguyên lý cơ bản hoạt động của robot, cũng như việc thiết kế hệ thống sao cho robot hoàn thành 1 nhiệm vụ cụ thể đó là mô phỏng lại động tác pha chế Latte và tạo hình bằng tay máy.



(Hình 1: Hình ảnh mô phỏng robot yumi)

Từ khóa: Yumi ABB, Latte

1. Giới thiệu

A, Đối tượng và phương pháp

- Đối tượng: Hệ thống robot Yumi, hệ thống điều

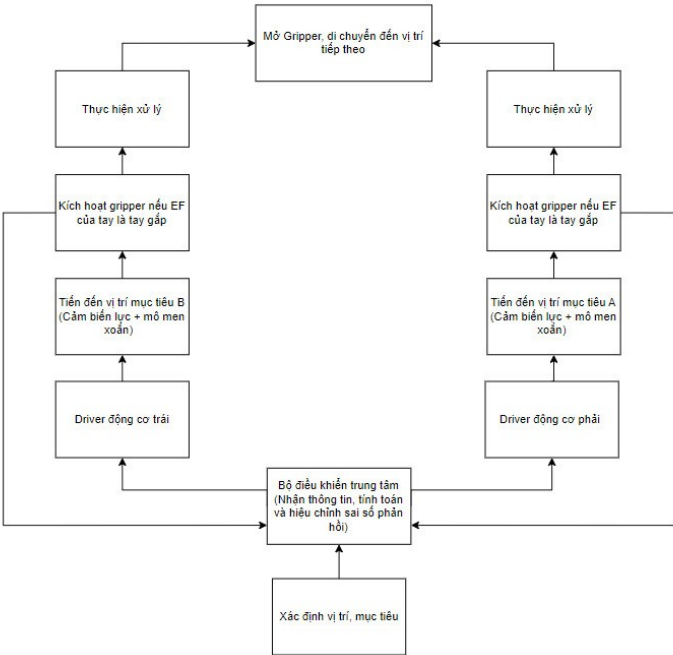
khiển, ...

- Phương pháp nghiên cứu: Tính toán, thiết kế và mô

phỏng robot trên hệ 3D, nghiên cứu động học và động lực học của robot.

B, Nội dung nghiên cứu

- Để có thể thiết kế một và điều khiển robot Yumi hoàn thành một tác vụ bất kỳ nào, phải xác định rõ: Mô hình hệ thống khi xây dựng, điều cần tránh khi điều khiển, phương thức điều khiển, thuật toán điều khiển, ... Trước tiên để điều khiển được hệ robot, phải xác định rõ các cấu tạo thành phần cơ bản của robot như arm, joints, ...

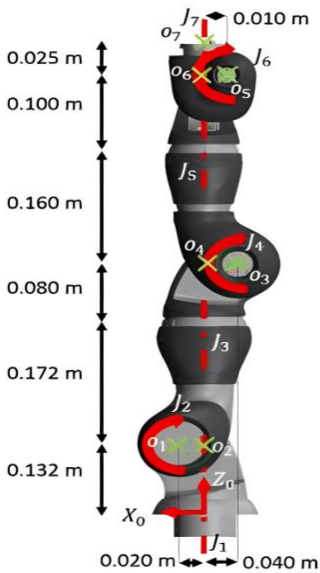


(Hình 2: Sơ đồ điều khiển robot Yumi)

Xác định chính xác giới hạn khối lượng của vật (mục tiêu), khoảng xác, vị trí, tọa độ rõ ràng của vật để giảm thiểu việc sai số cũng như hiệu suất của robot. Khi điều khiển robot, ta phải hiểu rõ các cơ cấu của các khớp liên kết cũng như hệ thống của chúng để thiết kế rõ ràng một pipeline cụ thể cụ thể để từ đó hướng tới vấn đề chung cần giải quyết. Phần quan trọng nhất là thuật toán cũng như phương thức điều khiển, ta tính toán hệ động học thuận và

ngược của robot vì nó ảnh hưởng tới kết quả công việc.

2. Tổng quan về nghiên cứu

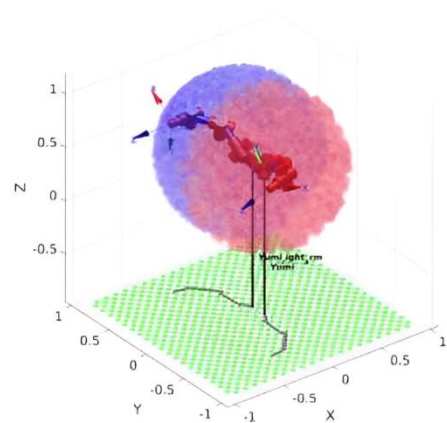


(Hình 3: Tay của Yumi với 7 khớp)

- Phần này trình bày về thiết kế mô hình vẽ latte được điều khiển đồng bộ theo sơ đồ hình:

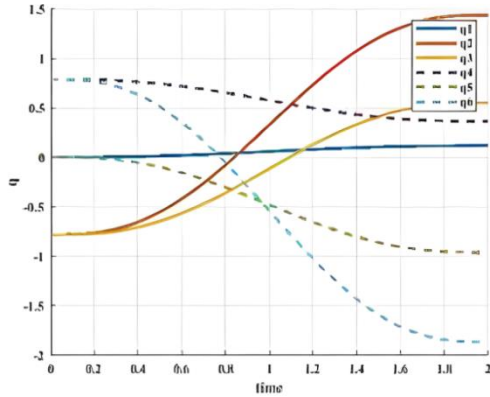
Articulation	α_i (rad)	d_i (cm)	r_i (cm)	θ_i (rad)
1	0	0	16.6	$q1-\pi$
2	$\pi/2$	3	0	$q2-\pi$
3	$\pi/2$	3	25.15	$q1$
4	$-\pi/2$	4.05	0	$q4-\pi/2$
5	$-\pi/2$	4.05	26.5	$q5+\pi$
6	$-\pi/2$	2.7	0	$q6-\pi$
7	$-\pi/2$	2.7	3.6	$q7+\pi$

(Hình 4: Bảng D-H cho tay phải của Yumi)

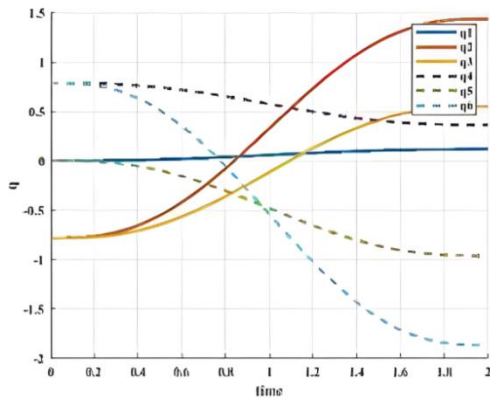


(Hình 5: Mô phỏng workspace của Yumi)

- Xác định workspace vào bảng D-H ta có thể hình dung rõ ràng được phạm vi hoạt động của robot, từ đó nhận định rằng mục tiêu đề ra có thể thực hiện được hay không.

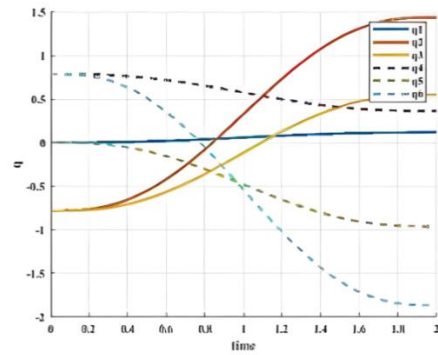


(Hình 6: Không gian khớp cánh tay trái)

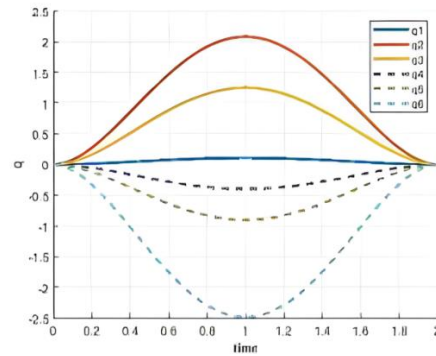


(Hình 7: Không gian khớp cánh tay phải)

Không gian khớp của cánh tay trái không gian khớp của cánh tay phải. Ở cánh tay trái, khớp thứ hai và khớp thứ sáu có độ dịch chuyển tối đa, nghĩa là chúng quay ở mức tối đa. Nó cũng xác định rằng khớp thứ nhất và thứ năm ở mức 0 trong quá trình khởi động.

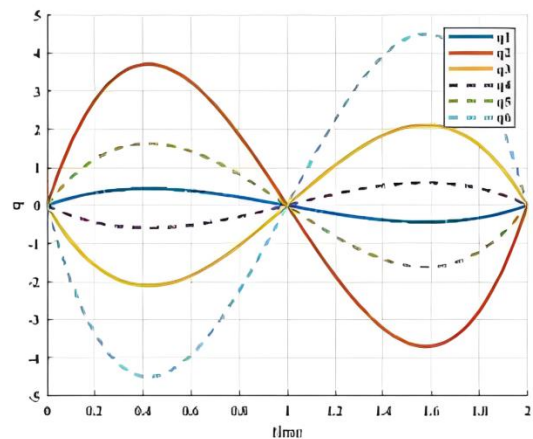


(Hình 8: Không gian quỹ đạo vận tốc tay trái)

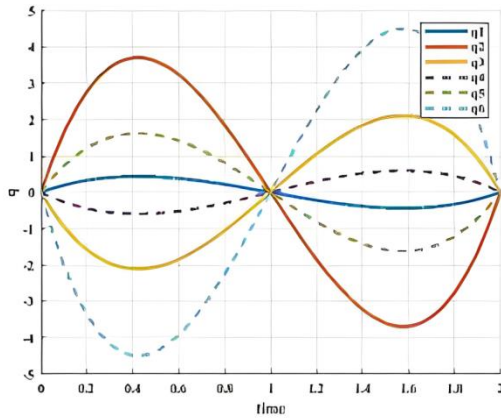


(Hình 9: Không gian quỹ đạo vận tốc tay phải)

- Điều xảy ra ở cả hai cánh tay: . Tại tay trái vận tốc của khớp thứ ba dao động âm, trong khi vận tốc của khớp thứ năm khớp dao động theo chiều dương. Điều này ngược lại ở cánh tay phải.



(Hình 10: Không gian quỹ đạo gia tốc tay trái)



(Hình 11: Không gian quỹ đạo gia tốc tay phải)

- Quan sát ta thấy, gia tốc tại khớp thứ 6 luôn đạt gia tốc tối đa còn tại gia tốc của khớp thứ nhất thì lại là nhỏ nhất.

3, Methodology

a, Cách hoạt động của yumi trong nhiệm vụ vẽ latte.

- Đầu tiên, ta xác định vị trí của robot và vị trí của vật thể (mục tiêu) trong không gian làm việc (hình 1). Vị trí của vật thể có thể được xác định trước hoặc ngẫu nhiên bằng việc để robot xác định bằng cảm biến cũng như camera được trang bị sẵn. Tiếp đó các thông tin này sẽ được gửi đến trung tâm điều khiển để thực hiện tính toán động học thuận cho các khớp, tiếp đó kích hoạt tay trái phải tiến đến mục tiêu từ góc khớp đã được xác định. Sau khi tiến đến vị trí vật thể, cảm biến lực (Force Sensors), cảm biến momen xoắn (Torque Sensors) và hình ảnh được kích hoạt trong việc xác định đúng vị trí của vật thể, hình ảnh trích xuất các đặc điểm của đối tượng. Đây là cách hoạt động chi tiết:

- Các cảm biến này được gắn trên các khớp của robot, giúp đo lực và momen tác động lên các trục khi robot thực hiện các chuyển động.

+ Khi robot cố gắng nhấc một vật thể, cảm biến lực có thể đo được lực cần thiết để nhấc vật thể đó. Nếu lực này vượt quá một ngưỡng nhất định, robot có thể nhận biết rằng vật thể không được nhấc đúng cách.

- Hiệu chỉnh dựa trên cảm biến lực:

- Nếu robot cảm nhận được lực không đồng đều khi gấp vật thể, nó có thể điều chỉnh vị trí của gripper để đảm bảo lực phân bố đều hơn, từ đó giúp gấp chính xác hơn.

- Nếu cảm biến lực phát hiện lực tác động khác thường (ví dụ: quá lớn hoặc quá nhỏ), robot có thể điều chỉnh vị trí của mình hoặc thay đổi cách tiếp cận vật thể.

- Nhận biết momen tác động lên các trục, giúp điều chỉnh lực để tránh làm hỏng vật thể hoặc robot.

+ Ví dụ: Khi xoay một vật thể, cảm biến momen xoắn có thể giúp robot điều chỉnh lực xoay để đảm bảo vật thể không bị xoay quá mức hoặc không bị tuột.

b, Động học thuận và động học ngược.

- Động học thuận:

+ Ma trận Denavit-Hartenberg cục bộ:

$$A_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i a_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i a_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

+ Ma trận chuyển đổi thuần nhất:

$$T_7^0 = A_1^0 \cdot A_2^1 \cdot A_3^2 \cdot A_4^3 \cdot A_5^4 \cdot A_6^5 \cdot A_7^6$$

$$= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ R_{goal} : ma trận xoay, P_{goal} : vector vị trí

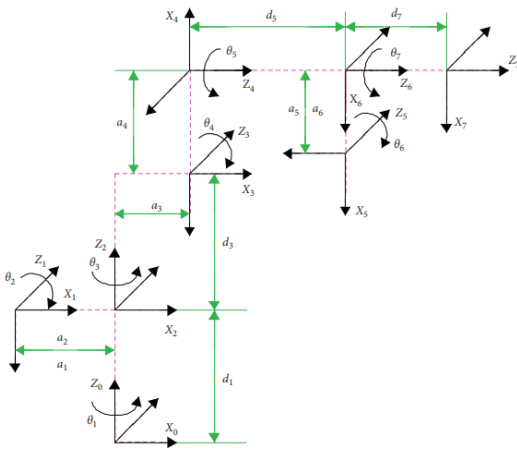
$$T_{goal} = \begin{bmatrix} R_{goal} & P_{goal} \\ 0 & 1 \end{bmatrix}$$

$$p_x = a_1 c \theta_1 - d_7 (c \theta_6 (c \theta_4 (s \theta_1 s \theta_3 - c \theta_1 c \theta_2 c \theta_3) + c \theta_1 s \theta_2 s \theta_4) - s \theta_6 (c \theta_5 (s \theta_4 (s \theta_1 s \theta_3 - c \theta_1 c \theta_2 c \theta_3) - c \theta_1 c \theta_2 s \theta_2) - c \theta_1 c \theta_4 s \theta_2) - s \theta_5 (c \theta_3 s \theta_1 + c \theta_1 c \theta_2 s \theta_3))) - d_5 (c \theta_4 (s \theta_1 s \theta_3 - c \theta_1 c \theta_2 c \theta_3) + c \theta_1 s \theta_2 s \theta_4) - a_4 s \theta_4 (s \theta_1 s \theta_3 - c \theta_1 c \theta_2 c \theta_3) - a_5 s \theta_5 (c \theta_3 s \theta_1 + c \theta_1 c \theta_2 s \theta_3) + a_2 c \theta_1 c \theta_2 + d_3 c \theta_1 s \theta_2 - a_3 s \theta_1 s \theta_3 + a_5 c \theta_5 (s \theta_4 (s \theta_1 s \theta_3 - c \theta_1 c \theta_2 c \theta_3) - c \theta_1 c \theta_4 s \theta_2) + a_6 s \theta_6 (c \theta_4 (s \theta_1 s \theta_3 - c \theta_1 c \theta_2 c \theta_3) + c \theta_1 s \theta_2 s \theta_4) + a_6 c \theta_6 (c \theta_5 (s \theta_4 (s \theta_1 s \theta_3 - c \theta_1 c \theta_2 c \theta_3) - c \theta_1 c \theta_4 s \theta_2) - s \theta_5 (c \theta_3 s \theta_1 + c \theta_1 c \theta_2 s \theta_3))) + a_3 c \theta_1 c \theta_2 c \theta_3 + a_4 c \theta_1 c \theta_4 s \theta_2,$$

$$p_y = d_7 (c \theta_6 (c \theta_4 (c \theta_1 s \theta_3 + c \theta_2 c \theta_3 s \theta_1) - s \theta_6 s \theta_2 s \theta_4) - s \theta_6 (c \theta_5 (s \theta_4 (c \theta_1 s \theta_3 + c \theta_2 c \theta_3 s \theta_1) + c \theta_4 s \theta_1 s \theta_2) - c \theta_2 c \theta_3 s \theta_1) + c \theta_4 s \theta_1 s \theta_2) - s \theta_5 (c \theta_1 c \theta_3 - c \theta_2 s \theta_1 s \theta_3)) + a_4 s \theta_4 (c \theta_1 s \theta_3 + c \theta_2 c \theta_3 s \theta_1) + a_5 s \theta_5 (c \theta_1 c \theta_3 - c \theta_2 s \theta_1 s \theta_3) + a_2 c \theta_2 s \theta_1 + a_3 c \theta_1 s \theta_3 + d_3 s \theta_1 s \theta_2 - a_5 c \theta_5 (s \theta_4 (c \theta_1 s \theta_3 + c \theta_2 c \theta_3 s \theta_1) + c \theta_4 s \theta_1 s \theta_2) - a_6 s \theta_6 (c \theta_4 (c \theta_1 s \theta_3 + c \theta_2 c \theta_3 s \theta_1) - s \theta_6 s \theta_2 s \theta_4) + a_3 c \theta_2 c \theta_3 s \theta_1 + a_4 c \theta_4 s \theta_1 s \theta_2,$$

$$p_z = d_1 - d_5 (c \theta_2 s \theta_4 + c \theta_3 c \theta_4 s \theta_2) - d_7 (s \theta_6 (c \theta_5 (c \theta_2 c \theta_4 - c \theta_3 s \theta_2 s \theta_4) - s \theta_2 s \theta_3 s \theta_5) + c \theta_6 (c \theta_2 s \theta_4 + c \theta_3 c \theta_4 s \theta_2) + d_3 c \theta_2 - a_2 s \theta_2 - a_5 c \theta_5 (c \theta_2 c \theta_4 - c \theta_3 s \theta_2 s \theta_4) + a_6 s \theta_6 (c \theta_2 s \theta_4 + c \theta_3 c \theta_4 s \theta_2) + a_4 c \theta_2 c \theta_4 - a_3 c \theta_3 s \theta_2 - a_6 c \theta_6 (c \theta_5 (c \theta_2 c \theta_4 - c \theta_3 s \theta_2 s \theta_4) - s \theta_2 s \theta_3 s \theta_5) - a_4 c \theta_3 s \theta_2 s \theta_4 + a_5 s \theta_2 s \theta_3 s \theta_5,$$

- Động học ngược:



(Hình 17: Minh họa cấu trúc tay bên trái của YuMi)

Các khớp vai, cổ tay và khuỷu tay có thể đặt tùy ý.

Đặt q1 và q2 ta tìm được góc q3:

$$n_{SEW}^T R_{02} R_{23} p_{34} = -n_{SEW}^T (R_{01} p_{12} + R_{02} p_{23}).$$

Tìm q5, q6 và q7:

$$-p_{67} + R_{67}^T R_{07}^T (p_{07} - p_{14}) = R_{65} (p_{56} + R_{54} p_{45})$$

Tìm q4:

$$R_{34} p = R_{03}^T R_{07}^T R_{47}^T p,$$

+ Vị trí của cổ tay:

$$P_{wrist} = P_{goal} - d_{EF} \cdot R_{goal} \cdot \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

4. Results and discussion

- Nội dung đề ra:

+ Hoàn thành mô phỏng Yumi, Cup, Table

trên CoppeliaSim.

+ Hoàn thành các phương trình Forward kinematics và Inverse kinematics.

+ Hoàn thành nhiệm vụ latte art action.

- Kết quả:

+ Tất cả các mục tiêu đề ra cả nhóm đã hoàn thành.

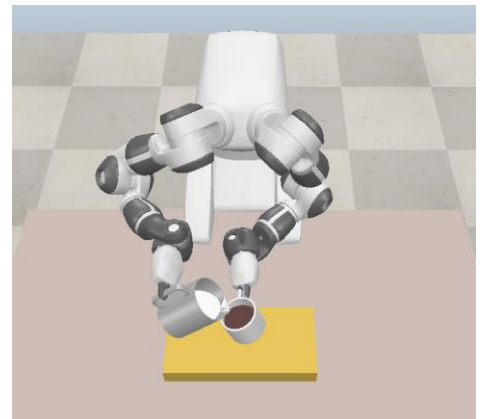
- Khó khăn:

+ Chưa thể mô phỏng đúng thực tế với việc vẽ tạo hình cũng như kết hợp 2 loại nước.

5. Conclusion

- Bài tập được triển khai dựa trên robot yumi thực hiện xem xét đánh giá các công việc có thể được ứng dụng nhiều trong cả kỹ thuật và dịch vụ để tăng năng suất cũng như hiệu suất công việc. Báo cáo đã trình bày khái quát về việc lên kết hoạch, thiết kế điều khiển cũng như các tính toán cơ bản để có thể điều khiển robot thực hiện 1 công việc đơn giản như pha cũng như vẽ latte trên café, đồng thời trình bày những nhược điểm còn tồn đọng trong việc chỉ triển khai trên nền tảng mô phỏng thay vì thực nghiệm thực tế. Kết luận lại, công việc được đề ra đã có thể hoàn thành và triển khai tốt trên nền tảng mô phỏng với sai số chấp nhận được.

c, Thực nghiệm trên nền tảng CoppeliaSim.



(Hình 12: Yumi thực hiện vẽ latte)

5 - Động học thuận với chức năng tính toán vị trí,

hướng của EF thông qua góc của các khớp cho trước:

```
function leftMoveToConfigCallback(config, velocity, accel, auxData)
    for i=1, #auxData.leftJoints, 1 do
        local jh=auxData.leftJoints[i]
        if sim.isDynamicallyEnabled(jh) then
            sim.setJointTargetPosition(jh, config[i])
        else
            sim.setJointPosition(jh, config[i])
        end
    end
end
```

(Hình 13: Code cập nhật vị trí dựa trên đầu vào)

+ Sử dụng 1 hàm trên

LeftMoveToConfigCallback để cập nhật vị trí các joints dựa trên các đầu vào đã cho (velocity, accel,..) thông qua sim.setJointTargetPosition và sim.setJointPosition tương tự cho tay phải.

```
function leftMoveToConfig_viaFK(maxVelocity, maxAcceleration, maxJerk, targetP)
    local startConfig={}
    for i=1, 7, 1 do
        startConfig[i]=sim.getJointPosition(auxData.leftJoints[i])
    end
    sim.moveToConfig(-1, startConfig, nil, nil, maxVelocity, maxAcceleration, maxJerk, targetP)
end
```

(Hình 14: Code di chuyển 2 tay dựa trên vị trí đã xác định)

+ Sử dụng Sim.MoveToConfig để di chuyển tay trái và phải đến vị trí mong muốn

- Với động học ngược ta biết tọa độ của vật thể, EF ta xác định các góc khớp để điều khiển robot thực hiện thành động

```
function rightMoveToPoseCallback(pose, velocity, accel, auxData)
    sim.setObjectPose(auxData.rightTarget, pose)
    simIK.handleGroup(auxData.iEnv, auxData.rightIkGroup, {syncWorlds=true})
end

function leftMoveToPose_viaIK(maxVelocity, maxAcceleration, maxJerk, targetP)
    local currentPose=sim.getObjectPose(auxData.leftTip)
    return sim.moveToPose(-1, currentPose, maxVelocity, maxAcceleration, maxJerk, targetP)
end
```

(Hình 15: Code minh họa điều khiển robot dựa trên IK)

```
local maxIkVel={0.45, 0.45, 0.45, 4.5}
local maxIkAccel={0.13, 0.13, 0.13, 1.24}
local maxIkJerk={0.1, 0.1, 0.1, 0.2}
```

(Hình 16: Data cho IK)

+ Sử dụng simIK.handleGroup và sim.moveToPose để thực hiện cập nhật tư thế, di chuyển các mục tiêu bên trái và bên phải theo mong muốn

6. References

- [1] <https://arxiv.org/pdf/2307.13122>
- [2] <https://manual.coppeliarobotics.com/en/apiFunctions.htm>
- [3] <https://www.sciencedirect.com/science/article/abs/pii/S0016003220307894>
- [4] https://www.researchgate.net/figure/7-DOF-YuMi-14000-ABB-industrial-robot_fig1_341227664
- [5] <https://iopscience.iop.org/article/10.1088/1757-899X/1012/1/012004/pdf>
- [6] <https://new.abb.com/products/robotics/robots/collaborative-robots/faqs/yumi>
- [7] <https://control.com/technical-articles/common-applications-of-force-and-torque-sensors-for-robotic-automation/>
- [8] <https://industrialmachinerydigest.com/imd/integrated-force-control-real-time-feedback-from-abb-robotics/>

