**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

****

**BÁO CÁO BÀI TẬP LỚN GIỮA KỲ**

**Sinh viên: Lê Đức Lâm – 21020922**

**Khóa: QH-2021-I/CQ-R**

**HỌC PHẦN: MÔ HÌNH HÓA ĐỘNG LỰC HỌC VÀ ĐIỀU KHIỂN ROBOT**

**Mã học phần: RBE3011**

**Ngành: Kỹ thuật Robot**

Hà Nội, 2023

**MỤC LỤC**

[**PHẦN 1. GIỚI THIỆU** 3](#_Toc149157432)

[**PHẦN 2. MÔ HÌNH ĐỘNG HỌC** 4](#_Toc149157433)

[**1.** **Động học thuận** 4](#_Toc149157434)

[**2.** **Động học ngược** 4](#_Toc149157435)

[**PHẦN 3. LẬP QUỸ ĐẠO CHO CÁNH TAY ROBOT UR10** 4](#_Toc149157436)

[**PHẦN 4. ĐIỀU KHIỂN VỊ TRÍ VÀ QUỸ ĐẠO ROBOT** 7](#_Toc149157437)

[**PHẦN 5. KẾT QUẢ CẢI TIẾN** 9](#_Toc149157438)

# **PHẦN 1. GIỚI THIỆU**

1. **Bài toán:**

Lập quỹ đạo chuyển động của tay máy UR10 và điều khiển robot di chuyển bám theo quỹ đạo.

1. **Yêu cầu:**
2. Lập quỹ đạo trên Matlab; kết nối Matlab với VREP để điều khiển UR10 qua API.
3. Áp dụng với trường hợp cho robot viết chữ trên mặt phẳng 2D (chọn chữ bất kỳ, chẳng hạn UET, FET,..
4. **Tóm tắt quy trình thực hiện (dự kiến)**

* Bước 1: Kết nối MATLAB với V-REP thông qua Remote APIs.
* Bước 2: Thêm và sắp xếp các đối tượng cần thiết vào môi trường làm việc (đã có sẵn trong thư viện V-REP): UR10, bút vẽ, mặt phẳng vẽ.
* Bước 3: Xác định giá trị gần đúng của các góc khớp ban đầu cho UR10 tạo điểm đặt bút đầu (Điểm mà đầu bút vuông góc mặt phẳng vẽ).
* Bước 4: Tính lại góc khớp ban đầu và xác định điểm gốc (tọa độ của End- effector để bút vẽ ở điểm đặt bút đầu).
* Bước 5: Lập quỹ đạo chuyển động của UR10 trên Matlab và chạy chương trình điều khiển UR10 bám theo quỹ đạo chuyển động đã lập.

1. **Các vấn đề gặp phải**

# **PHẦN 2. MÔ HÌNH ĐỘNG HỌC**

## **Động học thuận**

Phương trình động học thuận cho tay máy UR10 được thể hiện trong file **UR10\_forwardKinematic.m**.

## **Động học ngược**

Thực hiện bằng cách nhân ma trận Jacobi tựa nghịch đảo với tọa độ của khâu thao tác cuối, sau đó tạo vòng lặp tính toán để hiệu chỉnh sai số.

Ma trận Jacobi nghịch đảo được tính bằng hàm **computeJnd.m**.

# **PHẦN 3. LẬP QUỸ ĐẠO CHO CÁNH TAY ROBOT UR10**

1. **Xác định điểm bắt đầu của quỹ đạo:**

* Sử dụng động học thuận xác định tổ hợp 6 góc khớp ban đầu sao cho bút vuông góc và đầu bút chạm vào mặt phẳng vẽ.
* Vị trí đặt bút đầu tiên được xác định:

xE = -1.0024

yE = -0.2653

zE = 0.38823

* Ứng vơi giá trị gần đúng góc khớp ban đầu (đơn vị radian):

q1 = 1.6708

q2 = 0.62

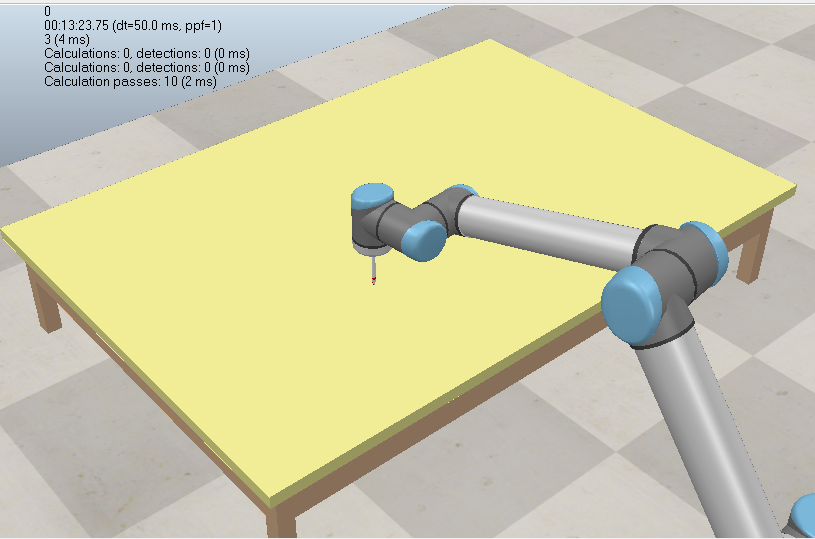
q3 = 1.208

q4 = -0.25

q5 = -1.5708

q6 = 0

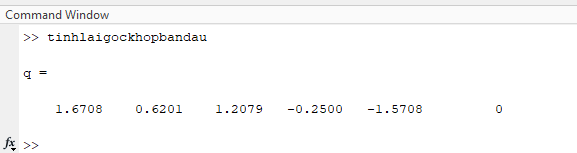
* Vị trí đặt bút đầu tiên trong CoppeliaSim:



* Sử dụng động học ngược để kiểm tra tính chính xác của vị trí các góc khớp:

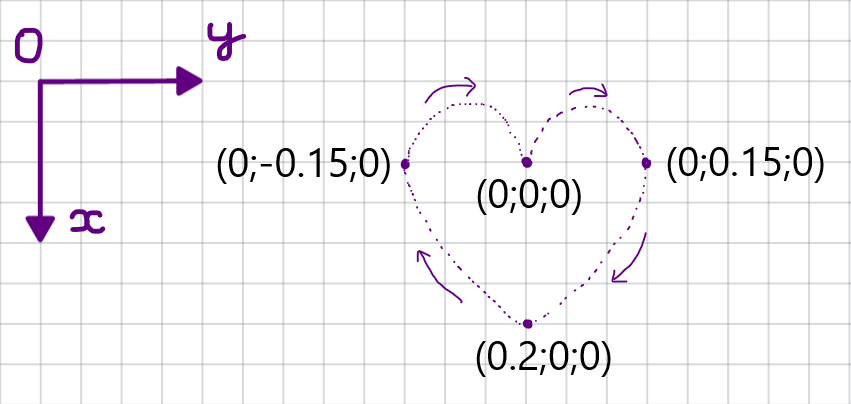
Thực hiện chạy file **tinhlaigockhopbandau.m** để kiểm tra:

Kết quả:

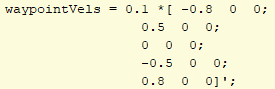


1. **Lập quỹ đạo cho robot**
   1. Lập quỹ đạo hình trái tim:

* Xác định các điểm đại diện cho quỹ đạo:



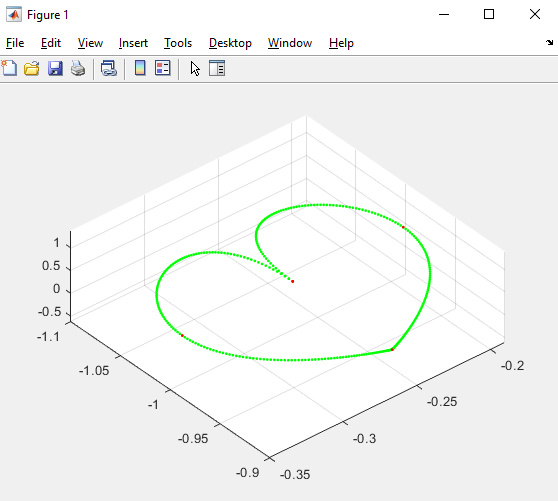
* Sử dụng quỹ đạo bậc 3 (cubic) cho các điểm, vector vận tốc được điều chỉnh để có được đường cong mong muốn:
* Thông số vận tốc được điều chỉnh trong hàm createMyWaypointData.m:



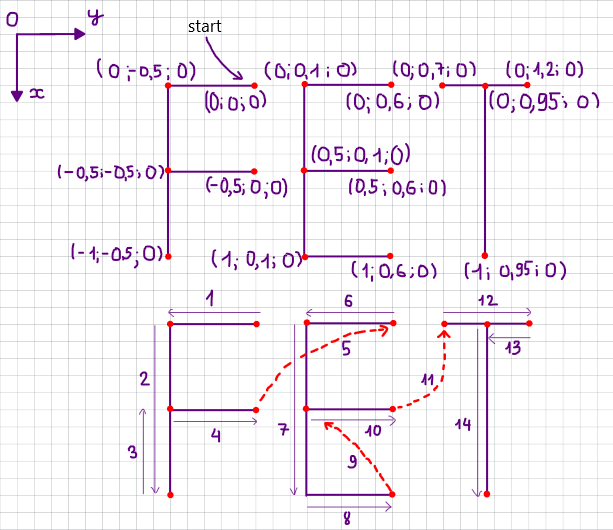
* Tham số đầu vào cho hàm **createMyWaypointData.m** là tọa độ điểm đặt bút đầu tiên và mảng lưu vị trí các điểm đại diện cho quỹ đạo:



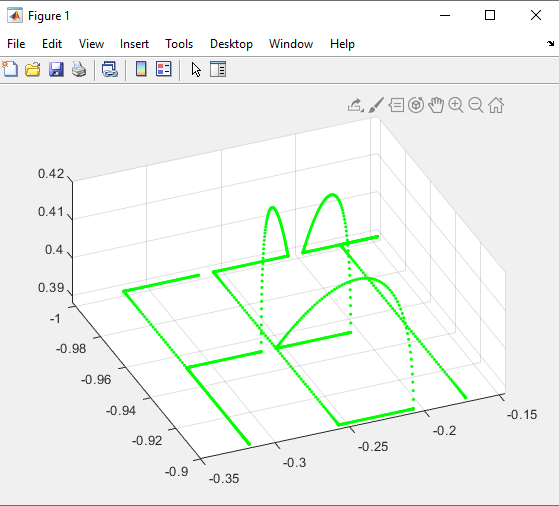
* Chạy file createTrajectory.m để tạo quỹ đạo bậc 3 cho hình trái tim, file **draw\_trajectory.m** để kiểm tra hình dạng của quỹ đạo:
* Kết quả khi thiết kế quỹ đạo bậc 3 cho hình trái tim:



* 1. Xác định quỹ đạo chữ ‘FET’:
* Xác định waypoints cho từng chữ:



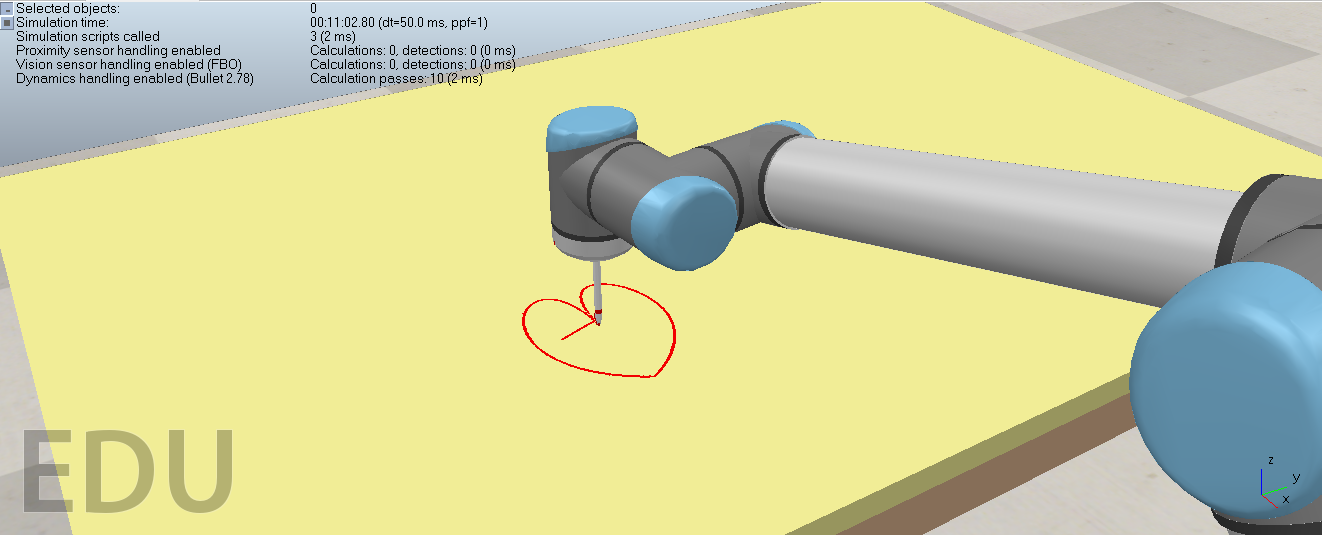
* Sử dụng quỹ đạo 2 1 2 để thiết kế nét cho từng chữ (màu xanh).
* Sử dụng quỹ đạo bậc 3 để thực hiện nhấc bút khỏi mặt bàn khi chuyển giữa các chữ (đường nét đứt màu đỏ).
* Kết quả khi thiết kế quỹ đạo:



# **PHẦN 4. ĐIỀU KHIỂN VỊ TRÍ VÀ QUỸ ĐẠO ROBOT**

1. Hình trái tim

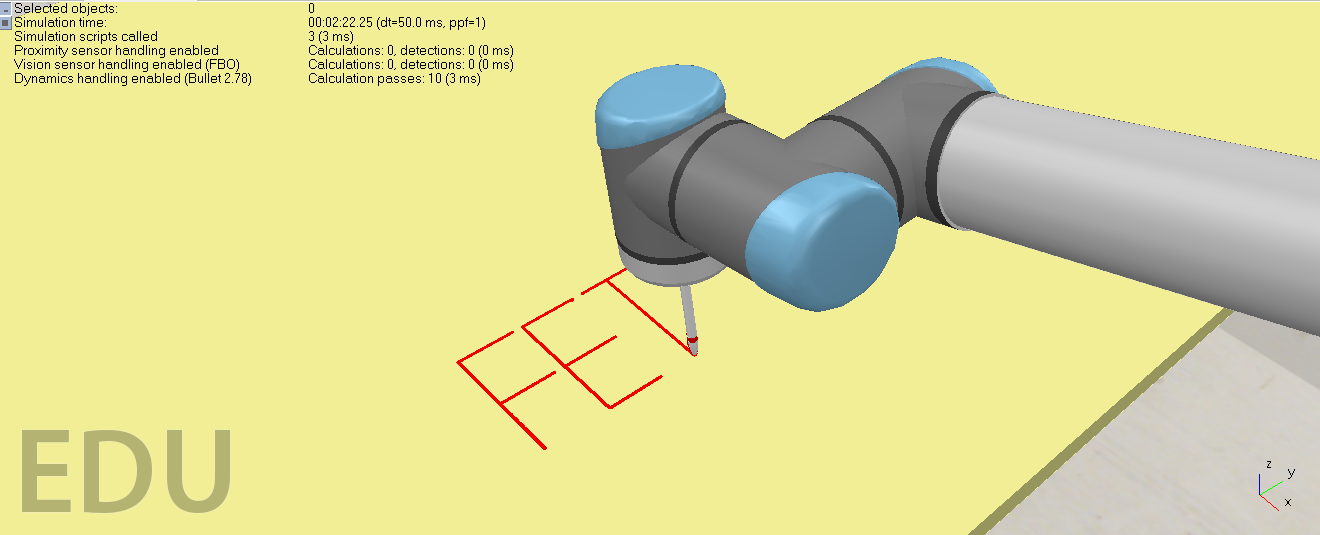
* Sau khi có được quỹ đạo mong muốn, vị trí khâu tác cuối theo thời gian sẽ được lưu vào mảng ‘qqq’ với kích thước là 3xN với N là số điểm mà robot sẽ đi qua trên quỹ đạo.
* Tạo vòng lặp tính toán động học ngược với đầu vào là mảng ‘qqq’ đã lưu từ trước, sau đó lưu kết quả là vị trí của 6 góc khớp theo thời gian vào mảng ‘qqqE’ với kích thước là 6xN. Thực hiện bằng cách chạy file **UR10\_draw.m**.
* Kết quả khi vẽ trên CoppeliaSim:



1. Chữ ‘FET’

* Tương tự với hình trái tim, thay đổi biến ‘myTrajectory’ trong file **createTrajectory.m** rồi sau đó chạy file **UR10\_draw.m**.





# **PHẦN 5. KẾT LUẬN**

Giải động học ngược như trên tốn nhiều thời gian khiến cho việc vẽ quỹ đạo trở nên lâu hơn, tốn nhiều thời gian.

Việc tính động học ngược chưa tính đến trường hợp giữ cho đầu bút luôn vuông góc với mặt phẳng cần vẽ, do đó dù giải động học ngược đúng nhưng nếu yêu cầu tăng kích thước của quỹ đạo cần vẽ thì sẽ có một số đoạn sẽ không được liền mạch.

Kết quả khi tăng kích thước hình trái tim lên 2 lần:

