Image Guided BioMedical Robotic Manipulation

BME/ECE 7400 Medical Robotics and Systems
Dr. Abhilash Pandya

Ron Hamameh
Sherif Matta
Vern Woldhuis
Overview

- Introduction
- Project Goal
- System Overview
- Project Objectives
- Project Objectives Implemented
- Final Project
- Examples
- Demonstration
- Questions
Introduction

- Robotic Systems in Medical Field
  - da Vinci
  - ROBODOC
  - Dental CAD/CAM systems
- All systems serve as assistants
- All can be developed further
Problem Statement

- Current Medical Robotic Systems are semi automated and serve only as human assistants
  - Use external camera or scanner to capture image
  - MANUALLY import data to image analysis software
  - MANUALLY use software to calculate robot path
  - MANUALLY export path
  - MANUALLY load path to medical robot
  - Run robotic path

- This process requires a high amount of manual interaction
Project Goal

- The goal of this project is to capture an image via a webcam and use a robot arm to reproduce that image.

- Symbolizes a fully automated system which could be used to perform robotic surgery without human intervention.
  - Multiple uses including:
    - Dental CAD/CAM
    - Milling femoral pockets for in total hip replacement
System Hardware

Camera → Computer → Microcontroller & Wireless Setup → Robot
Overall System Flowchart

Matlab GUI

Import Image → Process Image → Extract Cartesian Coordinates → Generate Inverse Kinematics → Filter Results

Robot Motion

Download Robot Path to Microcontroller
Project Objectives

- **Objective 1:** Capture an image of a shape using a web cam live feed and MATLAB's Video Processing Toolkit
- **Objective 2:** Import the image into MATLAB via GUI and MATLAB's Image Processing Toolkit
- **Objective 3:** Using MATLAB's Image Processing Toolkit analyze the image and create Cartesian coordinates or G-codes for a robot path
- **Objective 4:** Robot modeling and work envelope determination
Project Objectives Continued

- **Objective 5**: Robot Path Planning
- **Objective 6**: Calculate Inverse Kinematics
- **Objective 7**: Program microcontroller to control robot and communicate with MATLAB
- **Objective 8**: Robot moves along path and draws shape that was captured by the camera
Objective 1

- An image of a shape is captured using a web cam live feed and Matlab’s Video Processing Toolkit
Objective 2

- Image is imported into Matlab via GUI and Matlab’s Image Processing Toolkit
Objective 3

- Analyze the image and extract Cartesian coordinates for a robot path using Matlab’s Image Processing Toolkit
Objective 3

- This was not a simple task. How many points did we need? What commands and coding was correct? Two paths were taken until a clear winner was chosen.

Code utilizing `perim` & `boundaries` commands

Custom code to find corners only and extrapolate points in between
Objective 4

Modeling – DH Kinematics

DH kinematics model is derived.

\[
\begin{align*}
L1 &= \text{link}([\pi/2, 0, 0, 3.11, 0, 0], 'standard') \\
L2 &= \text{link}([0, 3.75, 0, 0, 0, \pi/2], 'standard') \\
L3 &= \text{link}([0, 3.75, 0, 0, 0, -\pi/2], 'standard') \\
L4 &= \text{link}([0, 5, 0, 0, 0, 0], 'standard')
\end{align*}
\]
Objective 4

Modeling – Reach Envelope Generation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE_HIGHT</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BASE_MIN</td>
<td>-60</td>
<td></td>
</tr>
<tr>
<td>SHOULDER_MIN</td>
<td>-90</td>
<td></td>
</tr>
<tr>
<td>ELBOW_MIN</td>
<td>-65</td>
<td></td>
</tr>
<tr>
<td>WRIST_MIN</td>
<td>-90</td>
<td></td>
</tr>
<tr>
<td>BASE_RADIUS</td>
<td>2;</td>
<td></td>
</tr>
<tr>
<td>BASE_MAX</td>
<td>+60</td>
<td></td>
</tr>
<tr>
<td>BASE_MAX</td>
<td></td>
<td>+60</td>
</tr>
<tr>
<td>SHOULDER_MAX</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>ELBOW_MAX</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>WRIST_MAX</td>
<td>90</td>
<td></td>
</tr>
</tbody>
</table>

The blue dots show the variation for q1 and q2 over the working range assuming q3 = π/2; q4 = 0;

The red dots show the variation of q1 and q4 assuming q2,q3 equal π/2.
Lynx 5 Robot calibration

Joint 3 Calibration

Joint 2 Calibration
Objective 5

- Path planning (trajectory generation)
- Plan the path of the robot end-effector
Objective 6

- Inverse Kinematics
- Compute the joint angles for the robot

\[ \mathbf{QQ} = \text{ikine}(\text{MY\_ROBOT, TC\_ALL, [0 0 -\pi/2 0],[1 1 0 1 0])} \]

- Initial estimation of the solution
- TC\_ALL: Contain all Cartesian coordinates path
- Mask to Ignore some DOF if solution doesn’t converge

Computed Joint Angles

Computer

Matlab
Joints to PWM

Micro seconds

-95deg -60deg Centre +60deg +95deg

0 500 1000 1500 2000 2500 3000

Ground
Supply +5V
PWM signal
Examples

Planned vs. Actual spiral design
Objective 7

- Interfacing & Firmware
- Serial interface was developed between MATLAB and Robot Microcontroller
- Motors move instantaneously after microcontroller receives signals
- Speed rate is controlled by MATLAB

Diagram:
- Computer
- Microcontroller & Wireless Setup
- Serial Cable RS232
Communication frame

<table>
<thead>
<tr>
<th>2 Byte</th>
<th>1 Byte</th>
<th>1 Byte</th>
<th>1 Byte</th>
<th>1 Byte</th>
<th>1 Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOF</td>
<td>Joint 1</td>
<td>Joint 2</td>
<td>Joint 3</td>
<td>Joint 4</td>
<td>EOF</td>
</tr>
</tbody>
</table>

Baud rate 115 Kbps

Every frame is at a specified delay configured by the user
PC – Microcontroller Communication

- PC (Matlab)
- Table of Desired Angles
  - Get Next row of Angles and convert to PWM
    - RS232 Buffer
      - Timer
        - Delay with respect to the slower motor
      - Servo Motor 1 (Base)
      - Servo Motor 2 (Shoulder)
      - Servo Motor 3 (Elbow)
      - Servo Motor 4 (Wrist)
Objective 8

- Move Robot
- Robot moves along path and draws the shape that was captured by the camera
Graphical User Interface
Example

Real Image

Planned trajectory

Output

Real vs. Planned vs. Output
Demonstration
Future Work

- Perform project with stable robot
  - Lynx 5 robotic arm introduced too much error
- Create automatic scale recognition
  - Image import currently has no accurate scaling method
- Add shape detection
  - Shape detection would symbolize recognition of cell types, particularly unwanted cancerous cells
Questions?
¿Preguntas?
أسماء؟
问题？
सवाल?
Вопросы?