Dupli Checker

PLAGIARISM SCAN REPORT

0% Plagiarised	100% Unique	**Date**	2024-03-24
		Words	926
		Characters	5867

Content Checked For Plagiarism

3.5 Fabrication of 3D Printed Hand

Here we used PLA because PLA is a common 3D printing material to print components. PLA is widely used because of its ease of availability, ecological footprint, and affordability.

PLA is a biodegradable substance derived from food crops like sugarcane, corn, and jowar. Here are some key characteristics of PLA and the 3D printer [9]:

Table 1: Features of PLA[9][10]

S.N.	Property	Value
1	Melting Point	Low (150°C and 180°C)
2	Thermal Expansion	Low (68 μm/m-K)
3	Layer Adhesion	Moderate
4	Heat Resistance	Low(55–65 °C)
5	Tensile Strength	High(39.9 MPa to 52.5 MPa)
6	Compressive Strength	High (48.2 MPa to 62.0 MPa)
7	Dimensional Accuracy	High

Table 2: Features of 3D Printer[11]

Model	Ender 3
Physical dimensions	(w) 40 cm x (d) 22 cm x (h) 46 cm
Maximum printing area	(w) 20 cm x (d) 22 cm x (h) 22 cm
Wire diameter	0.2mm
Nozzle diameter	0.4mm
Platform temperature	~100 °C
Nozzle printing temperature	~200 °C
Cooling method	Air Cool
Motor drive	Stepper Motor Drive

Now let us discuss the specification of the 3D Printed Hand

Table 3: Technical Specification of 3D Printed Parts[12]

No.	Name of the parts	No. of joints/parts	Length (cm)
1	Thumb finger	2	5.5
2	Index finger	3	6
3	Middle finger	3	8.5
4	Ring finger	3	7.5
5	Pinky finger	3	5.5
6	Palm	1	10
7	Wrist	-	2.3
8	The diameter of the end of the wrist	-	10
9	The total length of the arm	-	30

Fig. 12 Isometric projection of the 3D Printed Hand [11]

Fig. 13 After assembling all the parts of the hand

4 Algorithms and Their Detailed Analysis

4.1 Algorithm of the Arduino Code

Initialization:

- Set up the system parameters including the header files, sampling rate, baud rate, and pin configurations.
- Initialize the communication interfaces (e.g., Serial) for monitoring and debugging purposes.

Setup:

- Attach the servo motors to their respective pins and set their initial positions.
- Define the WritePin
- Configure the EMG sensor input pin and any auxiliary pins required for system operation.
- Define the threshold voltage (different for each person)

Main Loop:

- Check WritePin is High or Low

If High, Rotate the thumb, index, and middle finger as such it can hold a pen for writing purposes
If Low, Continue to the next steps

- Continuously sample the EMG signal at the defined sampling rate.
- Apply a band-pass Butterworth IIR digital filter to the raw EMG signal to extract relevant muscle activity within the desired frequency range.
- Compute the EMG envelope using an envelope detection algorithm to estimate the magnitude of muscle activation.

Gesture Recognition:

- Compare the normalized envelope value with a predefined threshold to determine muscle activation and gesture recognition.
- Implement a hysteresis mechanism to prevent rapid toggling of the hand due to noise or minor fluctuations in muscle activity.
- Define specific thresholds for opening and closing gestures based on individual user characteristics and preferences.

Servo Control:

- Based on the detected gesture:
- If the muscle activation exceeds the closing threshold:
- Close the hand by rotating the servo motors to the predefined closed position.
- If the muscle activation falls below the opening threshold:
- Open the hand by rotating the servo motors to the predefined open position.
- Implement a gesture delay to prevent rapid and unintended toggling of the claw in response to minor fluctuations in muscle activity. [1]

4.2 Algorithm for Envelope Detection-

1. Subtract the previous EMG signal value from the sum.
2. Add the absolute value of the current EMG signal to the sum.
3. Store the absolute value of the current EMG signal in the circular buffer at the current index.
4. Update the data index to point to the next position in the circular buffer, wrapping around to the beginning if necessary.
5. Compute the average of the EMG signal values in the circular buffer by dividing the sum by the buffer size.
6. Multiply the average by 2 to scale the envelope signal.
7. Return the computed envelope signal.

Fig. 14 EMG Signal After Filter with The Detected Envelope [9]

4.3 Algorithm of EMG Band Pass Filter

Algorithm for the Band-Pass Butterworth IIR digital filter:

1. Initialization:

- Initialize the state variables z1 and z2 for each filter section to zero.

2. Filtering Process:

- For each input sample:
- For each filter section:
- Calculate the intermediate variable x using the difference equation of a second-order IIR filter.
- Update the output using the calculated x value and the previous state variables.
- Update the state variables z1 and z2 for the next iteration.

3. Output:

- Return the filtered output.

Here's a breakdown of the steps within the filtering process:

- For each filter section:

1. Calculate the intermediate variable x using the difference equation:

x = input - a1 * z1 - a2 * z2

where input is the current input sample, z1, and z2 are the previous state variables, and a1 and a2 are the filter coefficients.

2. Update the output using the calculated x value and the previous state variables:

output = b0 * x + b1 * z1 + b2 * z2

where b0, b1, and b2 are the filter coefficients for the output.

3. Update the state variables z1 and z2 for the next iteration:

z2 = z1 z1 = x

Repeating these steps for each input sample, obtained the filtered output of the Band-Pass Butterworth IIR digital filter. [1] [5] [6]

Matched Source

No plagiarism found