Princess Sumaya University for Technology

King Abdullah II Faculty of Engineering

Electrical Engineering Department

EMBEDDED SYSTEMS
PROJECT PROPOSAL
ROBOTIC HAND THAT SIMULATES SIGN LANGUAGE

Authors:	*ID*	*Major*	*Emails:*
Tala Howari	20240749	Computer Eng.	tal20240744@std.psut.edu.jo
Lana Hussam	20201117	Power and Electrical Eng.	lan20201117@std.psut.edu.jo
Zaina Hasan	20220512	Network Information and Security Eng.	zai20220512@std.psut.edu.jo

Tuesday, January 14, 2025

Table of Contents

Abstract

This report presents the design and implementation of a robotic hand that simulates basic sign language gestures. The system integrates a variety of electronic components such as input devices, sensors, actuators, and feedback mechanisms to mimic human hand movements. The robotic hand's main functionalities include precise motor control of finger movements, gesture recognition through flex sensors, and user input handling using push buttons. Additionally, safety features are incorporated, such as an ultrasonic proximity sensor to prevent accidents. This project aims to provide an accessible communication tool for individuals with hearing or speech impairments, combining mechanical, electrical, and software elements in a seamless embedded system. The report also discusses the challenges encountered during development, as well as the future recommendations for improving the system.

Introduction and Background

The development of a robotic hand capable of performing sign language gestures plays a critical role in bridging communication barriers for individuals with hearing or speech impairments. As the global population of people with disabilities increases, it becomes more important to create accessible technologies that promote inclusion and ease of communication. The robotic hand mimics human finger movements to simulate various sign language signs such as "No", "Goodbye", and "I Love You". This involves the use of multiple sensors to detect finger bending, motors to actuate the movements, and input devices to select gestures.

This project integrates key embedded systems principles such as PWM motor control, interrupt-driven tasks, and analog feedback systems to achieve high precision in gesture reproduction. The primary goal of this project is to enable accurate and real-time sign language gesture simulation with emphasis on usability, safety, and efficiency.

Mechanical Design

The mechanical design of the robotic hand focuses on simulating basic sign language gestures, with a primary emphasis on moving only the distal (tip) joints of the fingers. The fingers are constructed from plastic and are rigid from the middle joint to the tip, with only the area around the knuckles (near the base of the fingertips) capable of moving.

Each of the finger's end joints is actuated by a servo motor positioned at the base of the knuckle or close to it. These servos control the movement of the fingers' distal joints by bending the finger tips forward and backward. The servo motors are connected to the plastic fingers using linkages, which translate the rotational motion of the servo to a bending motion at the knuckle area. The servos' motion is limited to moving the tips of the fingers, ensuring the system can replicate simple gestures with precision.

For wrist rotation, a DC motor is used to control the wrist's movement. The motor adjusts the angle of the wrist, providing additional flexibility in simulating gestures that require the hand to rotate or change orientation, such as waving or rotating the hand for different sign language expressions.

Overall, the mechanical design is optimized to perform a small set of sign language gestures by using servo motors to control the movement of the fingertips while ensuring the rest of the hand structure remains fixed and stable.

Electrical Design

The electrical components of the robotic hand include sensors, actuators, and control devices that work together to accurately simulate sign language gestures. These components work in conjunction with the mechanical design to control the finger and wrist movements based on user input and environmental conditions. The key electrical components include:

- **Flex Sensors (Voltage Divider Configuration)**: The flex sensors are placed near the fingertips to measure the bending of the fingers. These sensors are used in a voltage divider configuration. When the finger bends, the resistance of the flex sensor changes, causing a voltage change that is read by the microcontroller. This voltage is then converted to a binary value, which is used to control the servo motors to move the finger tips accordingly.

- **Servo Motors**: The servo motors are connected at the base of the knuckles of each finger. These servos control the bending of the finger tips. Each servo motor receives PWM signals from the microcontroller, which adjusts the angle of the joints near the knuckles, thereby controlling the movement of the finger tips.

- **DC Motor**: The wrist movement is controlled by a DC motor. The DC motor rotates the wrist, allowing for greater flexibility in performing sign language gestures. The motor is controlled through an H-Bridge motor driver, which allows for controlling both the speed and direction of the motor.

- **H-Bridge Motor Driver**: The H-Bridge motor driver is used to manage the DC motor's speed and direction. The H-Bridge allows the DC motor to rotate the wrist in both directions, enabling the hand to perform gestures that require wrist flexibility (such as turning the wrist to simulate different sign language motions). The H-Bridge is controlled by the microcontroller, using PWM signals to regulate the motor’s speed and direction.

- **Ultrasonic Proximity Sensor**: The ultrasonic proximity sensor is responsible for detecting nearby objects and ensuring the safety of the robotic hand. When an obstacle is detected within a certain range, the microcontroller triggers a safety stop to halt the motor operations. This prevents potential damage to the robotic hand or objects in the surrounding environment. The sensor’s data is continuously monitored, and when the threshold is exceeded, the system initiates an emergency stop. The buzzer also sounds to alert the user of the potential danger.

- **Limit Switches**: Limit switches are placed at predetermined positions within the hand's mechanical structure to stop the motor or servo once the finger or wrist reaches its maximum allowable position. This prevents overextension or excessive movement of the robotic hand’s components, ensuring safe operation. The limit switches send signals to the microcontroller, which then halts the corresponding motor or servo when the limit is reached.

- **Push Buttons (User Input)**: The push buttons allow the user to select different predefined sign language gestures. When a button is pressed, the microcontroller receives the input and triggers the corresponding servo movements to simulate the selected gesture. These buttons serve as a simple and effective user interface for controlling the robotic hand.

- **Buzzer (Auditory Feedback)**: The buzzer provides auditory feedback to the user. It is activated when a gesture is successfully performed, or when an error or safety issue is detected. The buzzer serves as an alert system, notifying the user of successful gestures, errors, or the activation of the safety stop (such as when the proximity sensor detects an obstacle).

- **Reset Button**: The reset button is used to restart or reset the system in the event of a malfunction or to restore the robotic hand to its default state. If the system experiences a freeze or error, pressing the reset button will initiate a restart, allowing the user to quickly resume operation without needing to power cycle the entire system.

- **LEDs (Visual Feedback)**: LEDs are used to provide visual feedback on the robotic hand's status. When a specific gesture is being performed, the corresponding LED will light up, indicating to the user which gesture is active. These LEDs are important for providing clear visual cues to the user about the current state of the hand.

All of these components are connected to a **standalone embedded system circuit board**, which houses the microcontroller that manages the input devices, sensors, and actuators. The embedded system is powered by a **lithium-ion battery**, and a **voltage regulator** ensures that each component receives the appropriate voltage for stable operation.

Software Design

The software design for the robotic hand controls the movement of the fingers and wrist, processes user input, provides real-time feedback, and ensures safety during operation. The system's primary function is to accurately simulate sign language gestures based on user input, utilizing sensors, actuators, and feedback mechanisms.

Key Components:

1. **Flex Sensors for Gesture Detection**:
    - The **flex sensor** acts as the primary input for finger movement. This sensor is placed near the fingertips and is part of a **voltage divider configuration**. When the fingers bend, the resistance of the flex sensor changes, resulting in a change in voltage. The microcontroller reads this voltage, processes it, and converts it into a binary signal, which is then used to control the movement of the servo motors that actuate the fingers.
    - The software continuously monitors the flex sensor readings and translates them into specific movements for the robotic fingers, ensuring realistic hand gestures.

2. **Push Button Input for Gesture Selection**:
    - The **push buttons** serve as the user interface for selecting predefined sign language gestures (e.g., "No", "Goodbye", etc.). Each button is mapped to a specific gesture, and when a button is pressed, the microcontroller triggers the corresponding servo movements to simulate the selected gesture.
    - The software listens for button presses, processes the input, and executes the appropriate gesture. This is achieved by controlling the servo motors to adjust finger positions according to the predefined gesture's binary encoding.

3. **Servo Motors for Finger Control**:

- The **servo motors** are responsible for moving the fingers based on the flex sensor input or button presses. Each servo motor is connected to the base of the knuckles of the robotic fingers.
- The software sends Pulse Width Modulation (PWM) signals to the servos to control their rotation and move the finger joints. These movements are continuously adjusted based on input from the flex sensors or buttons, ensuring that the robotic hand mimics the sign language gestures accurately.

4. **DC Motor for Wrist Movement**:
    - The **DC motor** controls the wrist's rotational movement. This allows the hand to perform gestures that require wrist flexibility (e.g., rotating the hand).
    - The software controls the speed and direction of the DC motor using an **H-Bridge motor driver**. PWM signals from the microcontroller modulate the motor's speed, while the H-Bridge allows the motor to rotate in both directions, providing the necessary wrist motion for a variety of gestures.

5. **H-Bridge Motor Driver**:
    - The **H-Bridge motor driver** is controlled by the microcontroller and manages the speed and direction of the DC motor. The software uses PWM signals to vary the motor's speed and reverse its direction when necessary, ensuring smooth and responsive wrist movements.
    - The H-Bridge helps in achieving forward and reverse motion of the wrist for different sign language gestures that require wrist rotation.

6. **Ultrasonic Proximity Sensor for Safety**:
    - The **ultrasonic proximity sensor** detects objects that are too close to the robotic hand. The software continuously checks the sensor's readings and triggers a safety stop if an object is detected within a predefined range. This feature ensures that the robotic hand halts its motion if an obstacle is detected, preventing damage to the hand or surrounding objects.
    - The system also activates the **buzzer** in the event of an obstacle detection, alerting the user to the safety issue. The software processes sensor input, compares it against the threshold, and initiates an emergency stop if necessary.

7. **Limit Switches for Movement Limitation**:
   - The **limit switches** are used to prevent the robotic hand's components (such as the finger) from moving beyond a safe range. The software monitors the limit switches and stops the corresponding motor or servo when the end position is reached.
   - When the finger or wrist reaches its limit, the microcontroller receives a signal from the limit switch, halting further movement and preventing damage or overextension of the robotic hand.

8. **Buzzer for Feedback**:
   - The **buzzer** provides auditory feedback to the user and serves as an alert system for obstacle detection by the ultrasonic sensor.
   - The software triggers the buzzer at appropriate times, such as when a gesture is performed or when an error or safety issue occurs.

9. **LEDs for Visual Feedback**:
   - The **LEDs** are used to give visual feedback to the user, indicating which gesture is currently being performed. Each predefined gesture corresponds to a specific LED, which lights up to show the user the active gesture.
   - The software controls the LEDs based on the current gesture being performed. When a gesture is activated by the user (via push buttons), the corresponding LED lights up, providing the user with a visual cue of the current state of the robotic hand.

10. **Timer-Based Control**:
    - A **timer** is used to manage the duration of time a gesture is maintained. The software ensures that once a gesture is completed, the robotic hand returns to a neutral position after a specified duration. This prevents the hand from staying in a fixed position for too long, contributing to more fluid and natural movement.
    - The timer task works in conjunction with the other control tasks, ensuring smooth transitions between gestures and maintaining the hand in a ready position for the next input.

Software Workflow:

1. **Sensor Reading and Processing**:
    - Flex sensors are continuously monitored, and the data is used to calculate the angle of the fingers. This is processed into a binary format that controls the servos.
    - The ultrasonic sensor is constantly checked for proximity, and the limit switches are monitored to prevent overextension.

2. **User Input and Gesture Execution**:
    - Push buttons are continuously scanned for user input. When a button is pressed, the software triggers the corresponding gesture by sending PWM signals to the servos and DC motor, controlling finger and wrist movements.

3. **Safety Checks**:
    - The software continuously monitors the ultrasonic sensor and limit switches. If an obstacle is detected or the limit switches are triggered, the system immediately halts motion and activates the buzzer to alert the user.

4. **Feedback and Status Indication**:
    - As gestures are performed, the software updates the LEDs to show the active gesture. The buzzer provides auditory feedback whenever a gesture is successfully completed or when a safety issue is detected.

5. **Timer Task**:
    - After each gesture, the software uses a timer to ensure that the hand returns to a neutral position after a set period, ensuring smooth transitions between gestures.

I see! Here's the "Problems and Recommendations" section based on the issues you mentioned:

Problems and Recommendations

1. Software PWM Generation

Problem: The microcontroller used in the project lacks hardware PWM channels for controlling multiple servos and motors. As a result, PWM signals had to be generated in software, which can be less efficient and more processor-intensive.

Recommendation:

- **Optimize Software PWM Code**: To improve timing accuracy, implement the PWM generation using hardware timers or interrupts rather than using software delays. This approach ensures that PWM signals are generated with greater precision and reduces the processor load during signal generation.
- **Consider PWM Driver ICs**: If further PWM channels are needed, consider using external PWM driver ICs, such as the **PCA9685**, which provides up to 16 PWM channels via I2C communication, offloading the PWM generation from the microcontroller.

2. DC Motor Activation

Problem: There were initial issues with turning on the DC motor, which didn't work as expected despite proper wiring.

Recommendation:

- **Verify Wiring Connections**: Double-check the connections, especially the polarity of the DC motor and H-Bridge motor driver. Ensure that the H-Bridge is correctly connected to the motor, and verify the control signals from the microcontroller.
- **Check Power Supply**: Make sure the DC motor receives sufficient current from the power supply. DC motors can require more current than a typical microcontroller pin can provide, so use an adequate power source for the motor and motor driver.

3. Servo Positioning with Plastic Fingers

Problem: The robotic hand's fingers are made of plastic, which limits the movement to only the ends of the fingers. As a result, the servos had to be repositioned to the knuckle area to allow more effective movement of the fingertips.

Recommendation:

- **Redesign the Finger Structure**: To enhance the movement range and precision, consider using a more flexible material for the fingers or a more advanced mechanical design that allows joint movement at the mid or base of the fingers.
- **Modify Servo Mounting**: Position the servos closer to the knuckles, as done in the current solution, but ensure the mechanical linkage between the servo and fingertips is as rigid and precise as possible for optimal performance.

4. Sensor Reading Accuracy

Problem: There were some challenges with accurately reading the flex sensor data, which could lead to inconsistencies in finger movement.

Recommendation:

- **Calibrate Flex Sensors**: Implement a calibration process in software to map sensor readings accurately to finger positions. This can help reduce noise and ensure the flex sensors produce reliable data.
- **Use Averaging Techniques**: To reduce noise from the analog readings, average the sensor data over multiple readings or apply a low-pass filter to smooth the input signal.
- **Consider Different Sensor Types**: If flex sensors prove inaccurate over time, consider using alternative sensors, such as potentiometers or strain gauges, for more precise finger position detection.

Conclusion

The development of the robotic hand for simulating basic sign language gestures has proven to be a successful and rewarding project. By integrating various mechanical, electrical, and software components, the system is capable of mimicking simple hand movements, providing a practical solution for accessible communication for individuals with hearing or speech impairments.

The mechanical design faced challenges due to the plastic construction of the fingers, but adjustments, such as repositioning the servos to the knuckle area, resolved the issue and allowed for functional finger movements. The electrical components, including flex sensors, servos, and DC motors, worked together to execute the desired gestures, with the use of an H-Bridge motor driver and limit switches ensuring safe and controlled operations. The software design effectively processed sensor inputs and generated control signals to manipulate the actuators.

Despite some challenges, including software PWM generation and sensor accuracy, these were addressed through optimization and modifications in both hardware and software. This project not only demonstrates the potential of robotic systems in sign language translation but also offers a foundation for further improvements in accuracy, reliability, and efficiency.

In conclusion, the robotic hand project provides a significant contribution toward accessible communication, and the lessons learned from this development lay the groundwork for future enhancements and wider applications in the field.