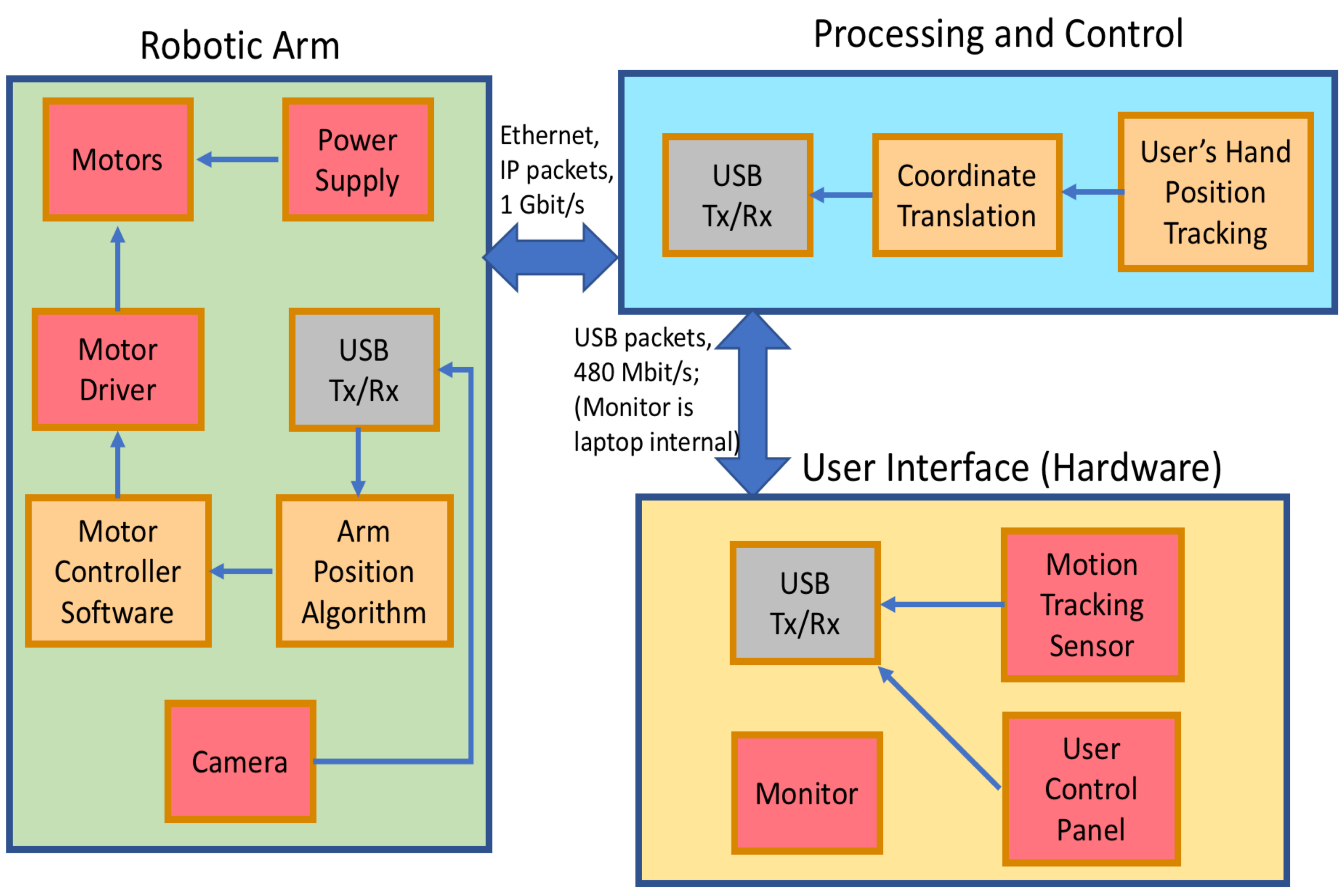
Helping Hand

Joshua Girard, Corey Ruderman,
Daniel Travis, Jacob Wyner
Faculty Advisor: Prof. Marco Duarte

Abstract

Helping Hand allows an inexperienced operator to control the movement of our robotic arm using intuitive hand movements and a few simple buttons alone. Our system interprets the live movements of the operator's arm and hand and translates those movements into commands for the robot. The robotic arm then moves to the correct position in order to mimic the operator's movements. Our system uses an external sensor to detect the operator's hand movements as opposed to hardware that must be worn by the operator. This system is intended to introduce a novel human robot interface that can be controlled by an operator with minimal training. Such a system has the potential to make complex robotic arms and similar systems accessible to essentially any operator while still allowing for precise operator control.

Block Diagram

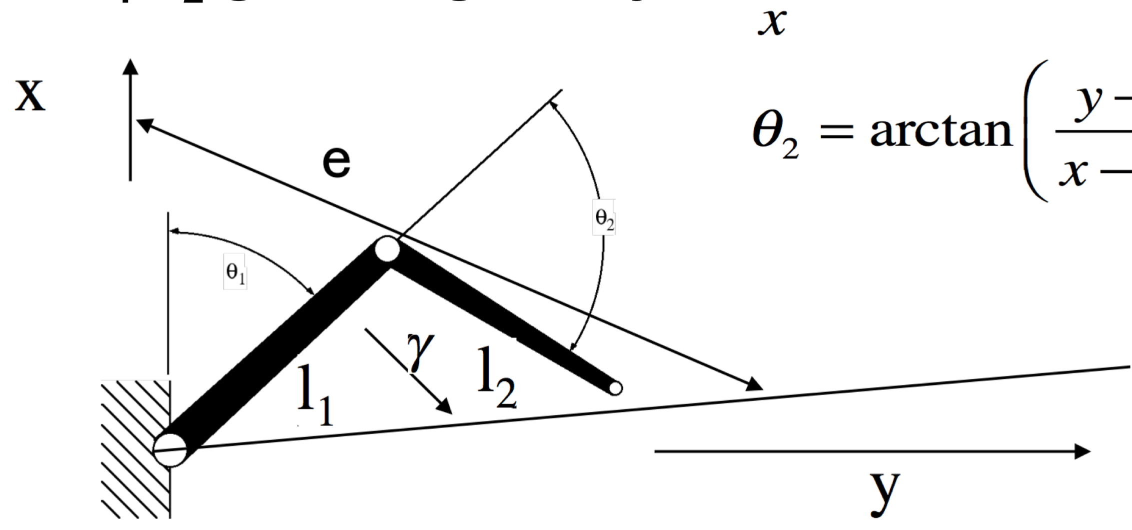


Inverse Kinematics

Inverse kinematics is the process by which angles are calculated for each joint in the arm based off of a desired x,y coordinate to be reached.
Pictured here is a 2D diagram of the arm in the x-y plane. To the right are the equations for calculating the shoulder and elbow joint angles θ_1, θ_2 given a goal x,y.

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

$$l = \sqrt{x^2 + y^2}$$
$$l_2^2 = l_1^2 + l^2 - 2 l_1 l \cos\gamma$$
$$\Rightarrow \gamma = \arccos\left(\frac{l^2 + l_1^2 - l_2^2}{2 l_1 l}\right)$$
$$\frac{y}{x} = \tan\varepsilon \quad \Rightarrow \quad \theta_1 = \arctan\frac{y}{x} - \gamma$$
$$\theta_2 = \arctan\left(\frac{y - l_1 \sin\theta}{x - l_1 \cos\theta_1}\right) - \theta_1$$



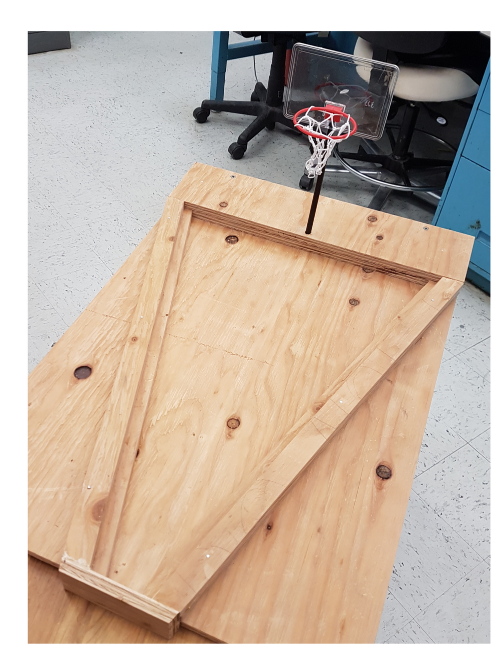
System Overview

The Helping Hand demonstrates the feasibility of constructing a remotely controlled robotic arm using a novel human-robot interface. Helping Hand uses inverse kinematics to mimic the operator's actions, resulting in an experience that is both precise and operator friendly. The operator is able to remotely control the arm due to the implementation of a live video feed from the arm to the operator, which is broadcast at roughly 60 FPS.

The robotic arm, with 4 degrees of freedom, is controlled by mimicking the operator's hand position with less than 250 milliseconds of latency and can move at least 5 inches per second in any direction. operator hand tracking is done by the Leap Motion at roughly 100 FPS, and is accurate up to 1" of the actual hand position. The gripper system was 3D printed, and is controlled solely by the operator opening and closing their hand. An operator control panel interfaces with the client software and has features such as pause/resume, power on/off, and emergency stop.

Results

In an attempt to prove the intuitiveness of the robot, Team Helping Hand performed a case study in which a random operator was selected to control the robot. This operator was assigned the task of dunking a ball, approximately 1.5" in diameter, into a basketball net that was approximately 2" in diameter. The operator had no prior experience with the robot, nor the Leap Motion sensor, and was able to successfully put the ball into the hoop on the first try.



Acknowledgements

Our group would like to thank our advisor, Professor Duarte, for giving us such great insight and advice throughout our project. A special thanks to Professor Hollot for his invaluable control systems knowledge. We would also like to thank Professor Frasier, Fran Caron and the staff at M5 for providing us with supplies.

System Specifications

Requirement	Specification
Lifting Strength	Should be able to lift at least 1 lb.
Range of Motion	The robot should be able to reach every point in a workspace 2’ wide, 2’ deep, and 1.5’ tall.
Latency	The robot should move within 250ms of the operator moving.
Speed	The movement of the robot should be at least 5 inches per second.
Functionality	An operator should be able to move five rocks from the workspace into a bowl in under 5 minutes.

Data Path

The data path begins at the operator’s end where sensor input is acquired from the Leap Motion and Operator Control Panel.
This data is processed on the local computer where the motion tracking algorithms run and coordinate translation and scaling (from operator’s coordinate space to robot’s) occurs.
Every 50ms, control packets with updated state are transmitted over TCP/IP to the robot’s onboard Raspberry Pi.
The Pi performs the inverse kinematics calculations and relays the target angles to the microcontroller.
The microcontroller implements a P controller for the linear actuators which move the arm vertically, and a lead compensator for the stepper motor attached to the base shaft.
Potentiometers in the linear actuators and on the base shaft provide feedback to the control system on the microcontroller.
A video camera mounted on the robot’s arm is connected to the Pi. This provides a live video feed back to the user by relaying the image stream back over the TCP/IP connection.

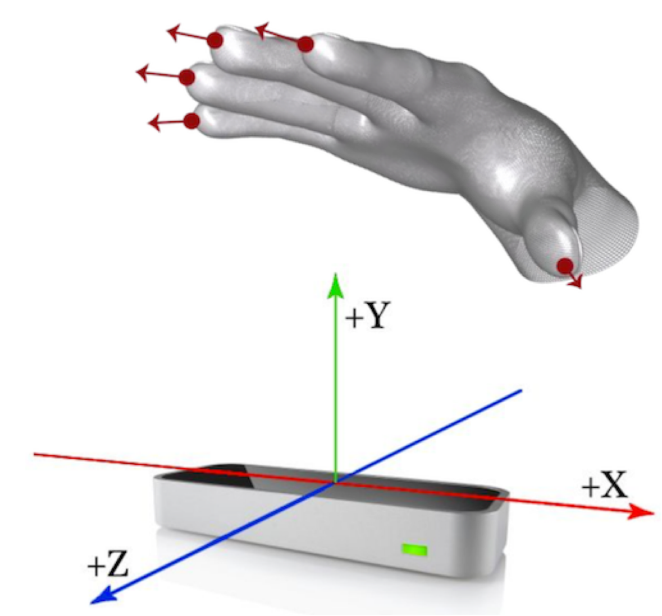
Costs

Part	Development	Production (1000)
Gripper System	$50	$20
Stepper Motor	$158	$100
Linear Actuator (2)	$260	$160
Servo Motor	$5	$2
Motor Driver	$28	$12
Custom PCB (2)	$60	$10
Raspberry Pi	$35	$20
Leap Motion	$60	$40
Webcam	$35	$15
Power Supply	$30	$10
8020 Aluminum (10ft)	$35	$10
Mounting Hardware	$30	$5
Total	$786	$404

Operator Interface

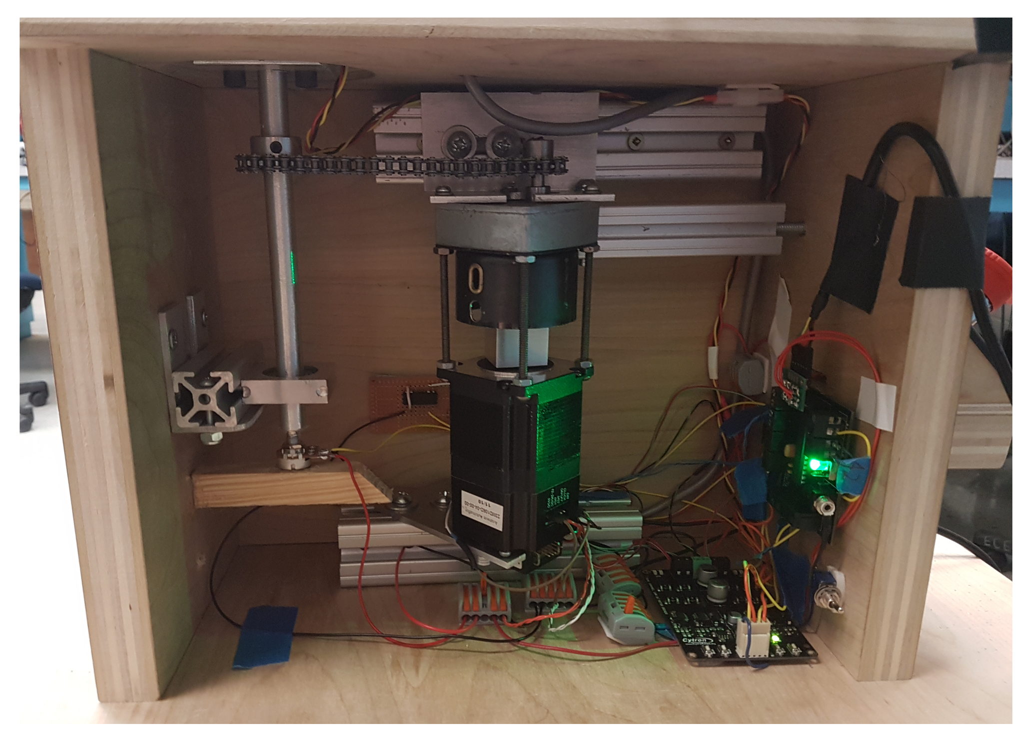
The Operator Interface consists of 3 components:

1. **Leap Motion Controller**
2. **Operator’s Control Panel**
3. **Live Video Feed**

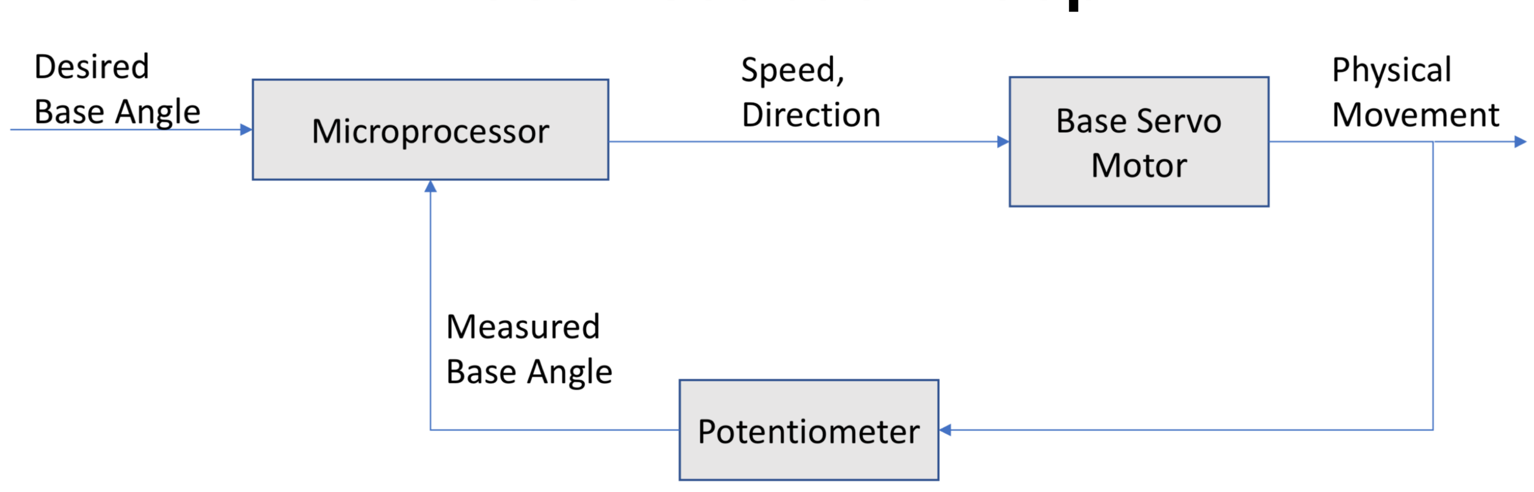


The Leap Motion tracks the operator’s 3D hand position in real time.
The Operator’s Control Panel allows the operator to perform important control actions.
The Live Video Feed enables the operator to observe the robot’s workspace in real time.

Rotational Movement



Base Feedback Loop



The robot’s base rotation movement is controlled with a stepper motor and a feedback control system. The feedback is angular position measured through a potentiometer and the control signal is speed and direction of the stepper motor.
Due to the large inertia when rotating at maximum speed, it is critical to implement a control system which minimizes jerk and overshoot.
The control system chosen for this application was a Lead Compensator.

Lead Compensator

$$\frac{u_{k+1} - u_k}{T} + bu_k = K[\frac{e_{k+1} - e_k}{T} + ae_k]$$

$$\Rightarrow u_{k+1} = T(K[\frac{e_{k+1} - e_k}{T} + ae_k] - bu_k) + u_k$$

Adapting Inverse Kinematics for 3D Rotational Movement

The inverse kinematics algorithm shown to the left works well for manipulation in 2D. However, in order to add support for 3D, it was adapted to work by first calculating the necessary base angle, then performing a mathematical rotational transformation using that base angle. This allows the shoulder and elbow joint angles to be calculated in the 2D plane facing the target point.