UMassAmherst

Helping Hand

Team 5

Team Members: Corey Ruderman, Dan Travis, Jacob Wyner, Joshua Girard

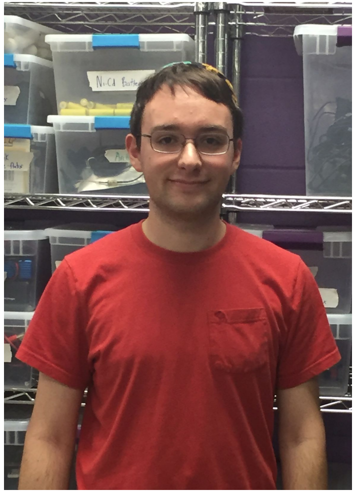
Advisor: Professor Duarte

Meet the Team

Corey Ruderman
CSE

Daniel Travis
CSE

Jacob Wyner
CSE



Joshua Girard
CSE, CS

The Problem:

- Robotic arms are used in everything from medical research to construction

- Remote control of robotic arms is complicated and unintuitive

Significance:

- As robotic technology becomes more prevalent in our society, more intuitive control approaches will be necessary
- This will help make the technology more accessible to all users with little to no training

Project Overview

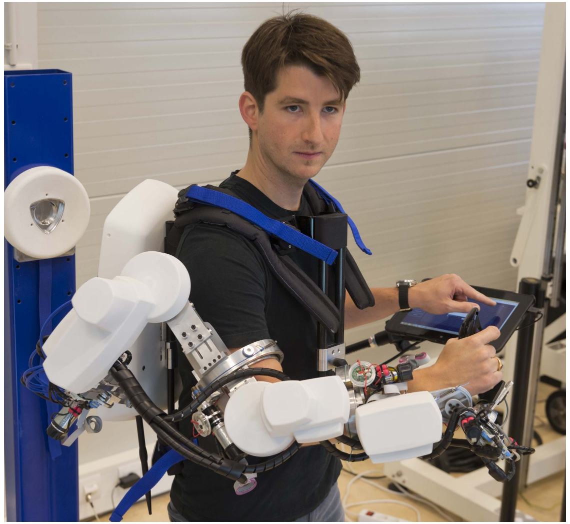
- Our team is designing and constructing a remotely controlled robotic arm using a novel human-robot interface that will be both precise and user friendly
- The arm will be controlled mainly by mimicking the user
- The user will also have a view of the arm from a live video feed

Current Solutions

- Control using a remote

- Control using a joystick

- Control using a wearable device

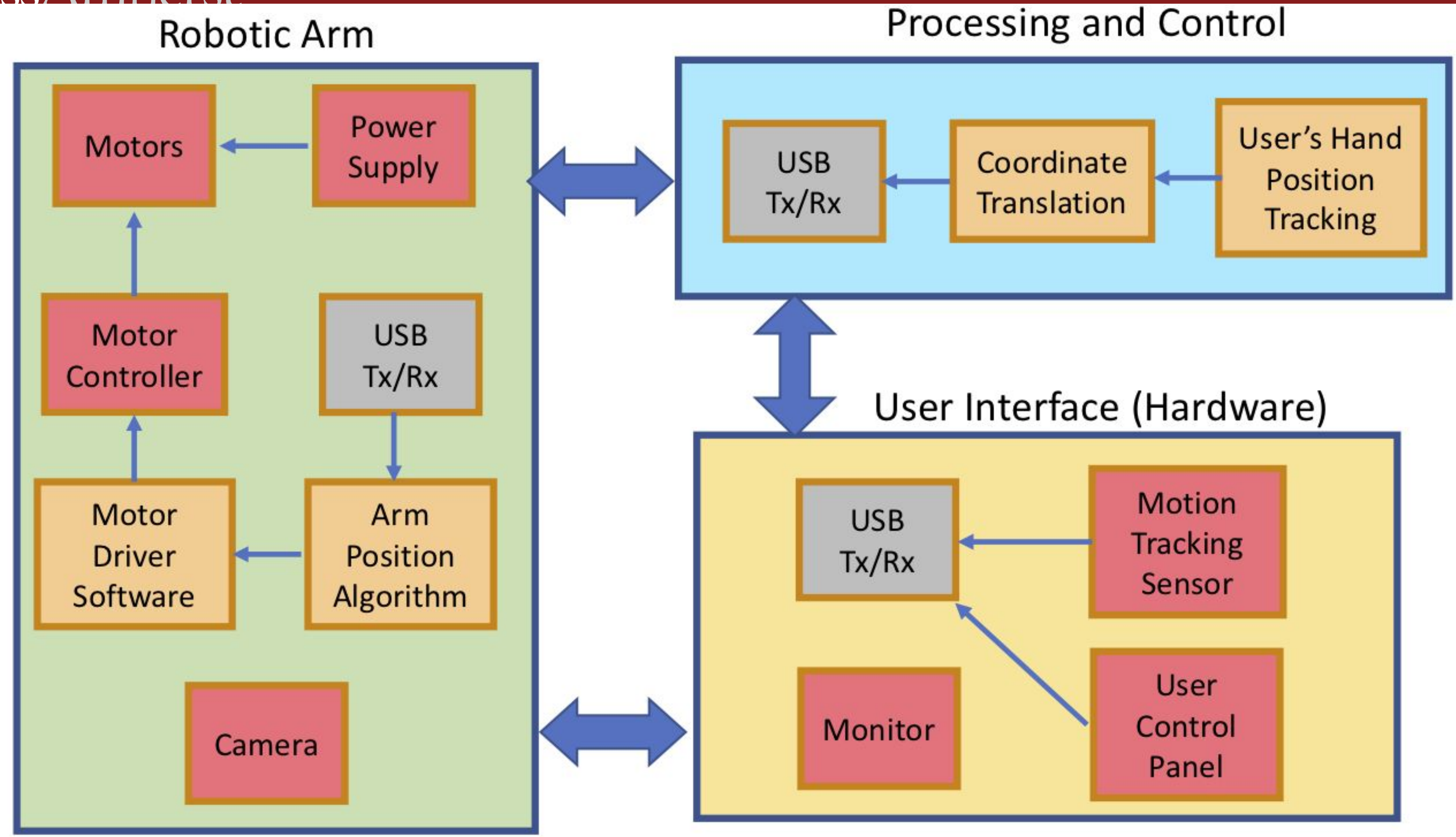


Arm Requirements and Specifications

- Arm will have a minimum range of motion defined by a rectangular prism 1.5′x1.5′ horizontally and 1′ vertically directly in front of the robot in 4 DOF
- Arm should mimic the user's arm position with <0.25 second latency
- Arm will be able to move at least 5 inches per second in any direction
- If user moves their arm faster than the robot arm can move, the robot must move at its fastest possible speed to the user's arm's final point
- Evaluation metric: Arm will be able to perform the task of moving 5 rocks (approx. size of a ping pong ball) placed randomly within the workspace of the arm into a ~3″ tall bowl of diameter ~8″

User Interface Requirements and Specifications

- Hand tracking -- Intuitive and easy to use
- Fast tracking rate (>20 FPS)
- Accurate tracking (within 1″ of actual hand position)
- Adequate range of motion (> 2′x2′x1′ tracking area)
- User Control Board should implement: on/off, emergency stop, pause/resume


Robotic Arm
Motors
Power Supply
Motor Controller
USB Tx/Rx
Motor Driver Software
Arm Position Algorithm
Camera
Processing and Control
USB Tx/Rx
Coordinate Translation
User's Hand Position Tracking
User Interface (Hardware)
USB Tx/Rx
Motion Tracking Sensor
Monitor
User Control Panel

Distribution of Responsibilities

Joshua:
User Input processing
and Desktop UI

Dan:
Arm construction,
User Interface PCB

Corey:
arm control
algorithms, website

Jacob:
Electronic hardware for
arm, arm control
algorithms

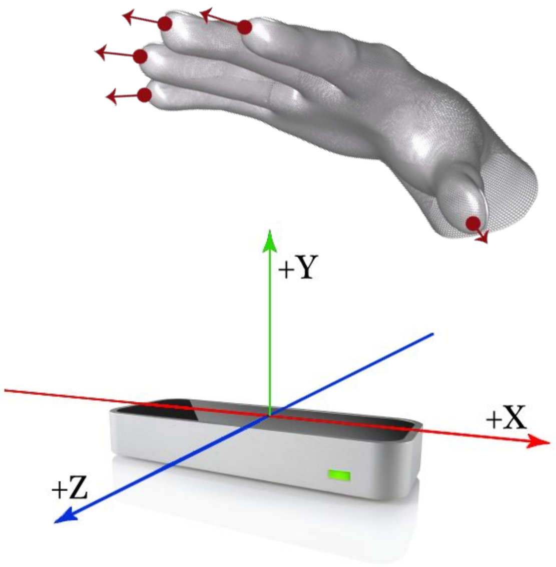
User Input & Processing- Joshua

- Motion tracking sensor
  - Leap Motion Controller
  - Or Microsoft Kinect v2

- Custom tracking software
  - Data processing and API integration
  - Smoothing (exponential filter)
  - Coordinate Translation

Motion Tracking: Leap Motion Controller -Joshua

- Uses optical sensors and infrared light to recognize and track hands and fingers
- 20-200 FPS
- Finger characteristics estimated based on recent observations and the anatomical model of the hand
- Bundled tracking software
- USB



Alternative Motion Tracking: Kinect

- Uses optical and infrared light for tracking
- >30 FPS
- 2mm tracking accuracy[1]
- Can detect open/closed hand
- Kinect tracking API
- USB 3.0

1. Evaluation of Pose Tracking Accuracy in the First and Second Generations of Microsoft Kinect https://arxiv.org/pdf/1512.04134.pdf

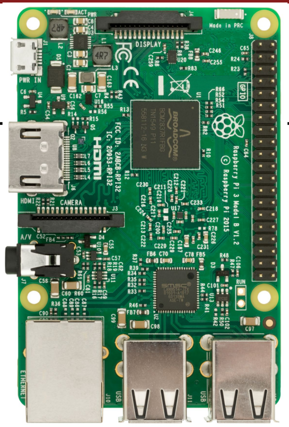
Physical Arm Construction- Dan

- Arm design V1: aluminum frame, uses DC gear motors/linear actuator, larger than design V2
- Arm design V2: controlled entirely by servos, smaller than design V1, backup if V1 fails
- Safety is being taken into consideration
- Focus of project is on user interface, not arm

Arm Electronic Hardware-Jacob

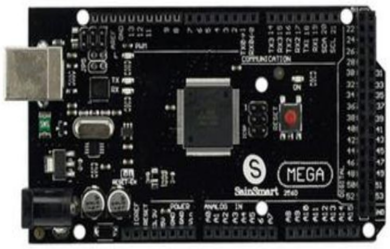
- 1) High level control: Embedded Computer
- 2) Low level control: Microcontroller
- 3) Live video feed from arm to UI

Arm Electronic Hardware Specifics



Jacob

- 1) Embedded Computer
  - Raspberry Pi Model 2: 1 GB RAM, 900 MHZ CPU
  - Requires level converter for connection to Arduino
- 2) Microcontroller
  - Atmega 2560: 256 KB flash storage, 16 MHZ clock
  - Hardware depends on arm design: For DC gear motors we will need an H-bridge, for stepper motors stepper motor driver
- 3) Live video feed from arm to UI
  - Sends a live video feed back to user using a webcam

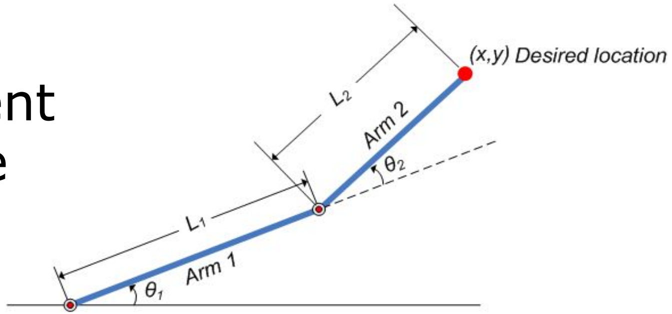


Arm Control Algorithm- Corey and Jacob

- The control algorithm consists of:

  1) Inverse Kinematics algorithm for 3 DOF arm

  2) Path planning algorithm: Given the current location of the arm and the current scaled location of the user’s arm give the robot arm directions



User Control Panel PCB - Dan

- Simple interface to give more control options to the user
- Power On/Off
- Emergency Stop
- Pause and Resume motion of the arm
- Interfaces with the Raspberry Pi via serial

Budget

▪ 3 DC gearbox motors or Linear Actuators: 3($40)	$120
▪ 1 Small DC gearbox motor or servo	$20
▪ Motor Controllers	$100
▪ Raspberry Pi Model 3	$35
▪ Arduino Mega	$35
▪ Leap Motion	$60
▪ User Interface PCB	$80
▪ 8ft 8020 Aluminum 1” ($2.76/ft)	$23
▪ Misc Hardware	$20
Total:	$493

MDR Deliverables

- Arm movement in 3 DOF (base + shoulder + elbow)
- Raw user input data is successfully received and processed
- User control board prototype complete

Questions?