UMassAmherst

Mid-year Design Review

Team 5: Helping Hand

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

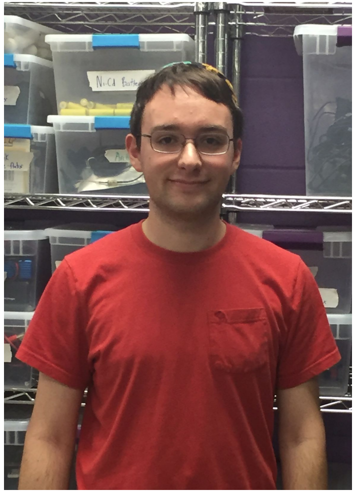
Advisor: Professor Duarte

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

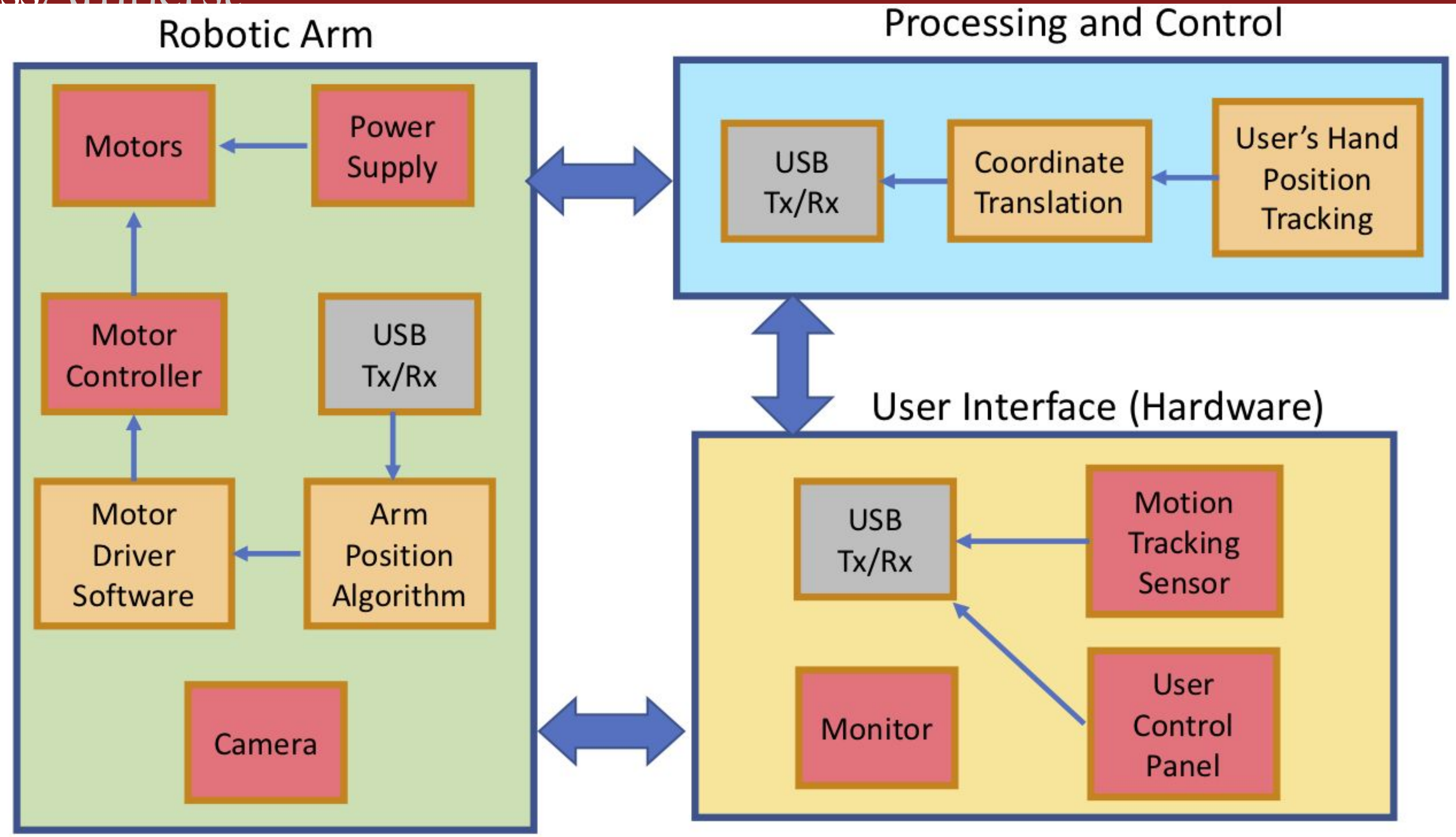
- Remote control of robotic arms is complicated and unintuitive

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
- Robot will move towards the user's current hand position as fast as possible rather than mimic all movements exactly
- Evaluation metric: Arm will 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″ within 5 min

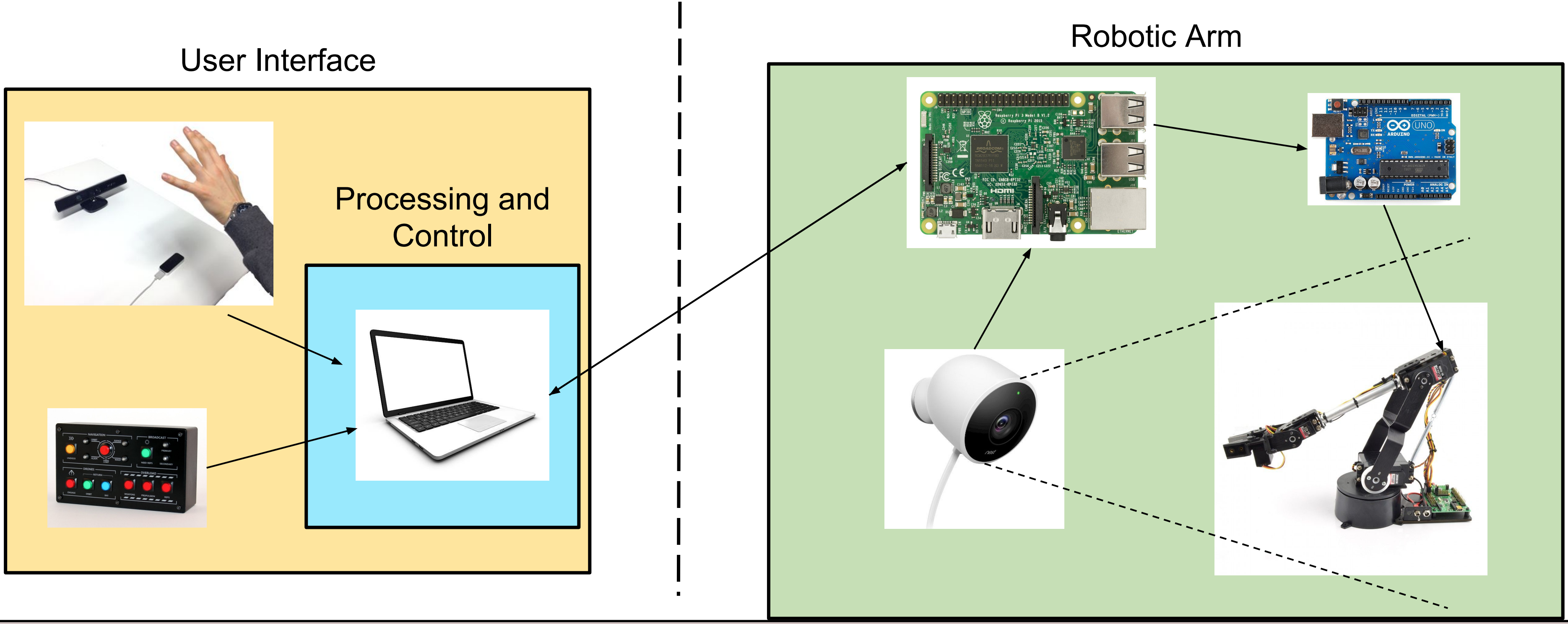
User Interface Requirements and Specifications

- 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

System Topology



MDR Deliverables

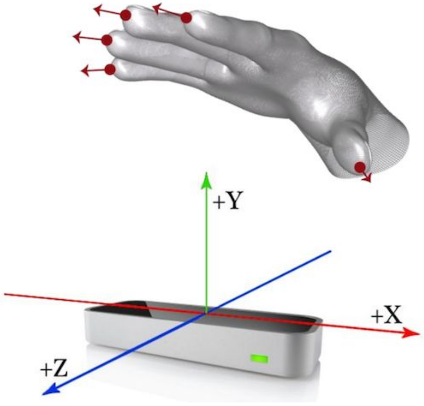
- Arm movement in 3 DOF (base + shoulder + elbow) (Jacob + Dan)
- Arm’s vertical movement controlled by integration of all major systems (All)
- Raw user input data is successfully received and processed (Joshua)
- User control board prototype complete (Dan)

MDR Deliverables

- ✓ Arm movement in 3 DOF (base + shoulder + elbow) (Jacob + Dan)
- ✓ Arm’s vertical movement controlled by integration of all major systems (All)
- ✓ Raw user input data is successfully received and processed (Joshua)
- ✓ User control board prototype complete (Dan)

Motion Tracking: Joshua

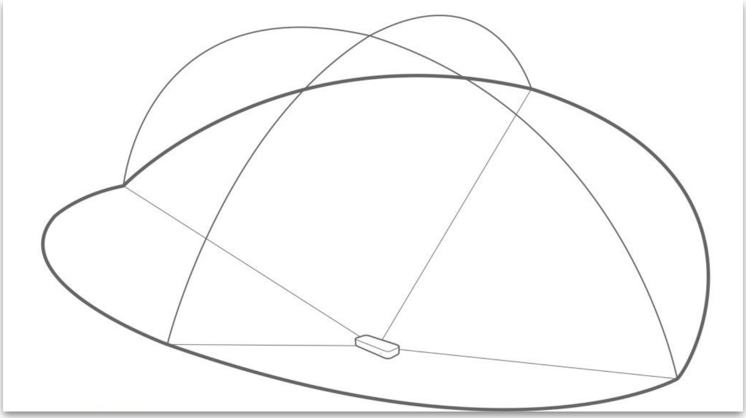
- Leap Motion Controller sensor
  - Effective range: 25 to 600 millimeters above the device (1 inch to 2 feet)
  - Field of view: ~150 degrees
- Mapping of user-space to robot-space coordinates
  - 1.07 : 1 ratio
- Tracking speed
  - ~100 FPS
  - Every 5 samples averaged
  - Transmitting latest coordinates to Raspberry Pi every 50 milliseconds (20x per second)



Design Specification: >20 FPS tracking ✓

Leap Motion Tracking Area

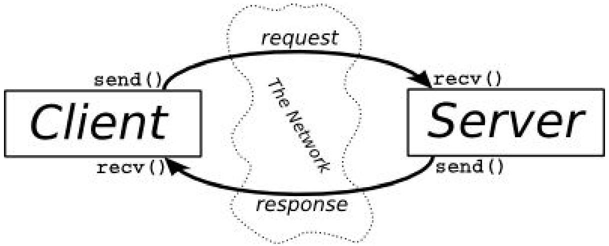
- Tracking Area
  - 2 feet above controller
  - 2 feet wide on each side (150º angle)
  - 2 feet deep on each side (120º angle)
- Design Specification:
  - > 2′x2′x1′ tracking area
  - Tracking area of Leap Motion is not a prism so corners are not within the tracked area



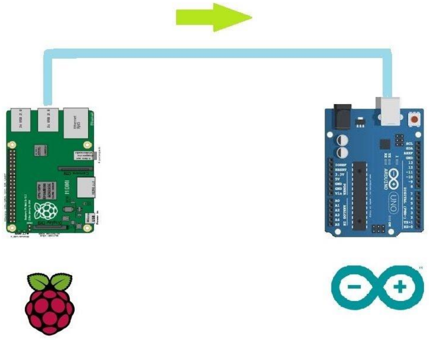
- Exploring Kinect tracking in Spring

Intersystem Communication

- Computer to Raspberry Pi
  - Ethernet: 100 Mbps
  - Client Server Model and TCP protocol

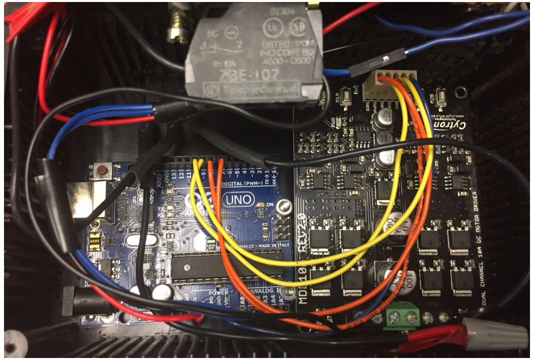


- Raspberry Pi to Microcontroller
  - Serial UART: 9600 bps



Arm Electronic Hardware: Jacob

- Raspberry Pi Model 3
  - Connected to Arduino Uno via Serial UART connection
- Atmega328P microcontroller
  - Controls motor controller using PWM signals
- MDD10A NMOS Dual Channel H-bridge
  - Supports 5V-30V
  - Max continuous current: 10A
  - Peak current: 30A
- Feedback sensors feed into 15-pin VGA cable
- Enclosure contains microcontroller, H-bridge, emergency stop



Arm Control Algorithms: Corey and Jacob

- 2 DOF inverse kinematics algorithm
  - Options: algebraic, iterative, inverse Jacobian
  - We chose to use algebraic method
  - Specific equation is from CS545 at USC

$$l = \sqrt{x^2 + y^2}$$

$$l_2^2 = l_1^2 + l^2 - 2l_1 l \cos\gamma$$

$$\Rightarrow \gamma = \arccos\left(\frac{l^2 + l_1^2 - l_2^2}{2l_1 l}\right)$$

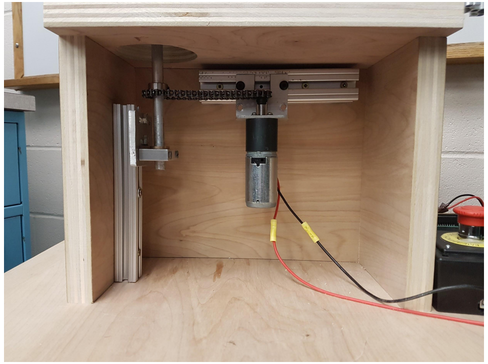
$$\frac{y}{x} = \tan\varepsilon \quad \Rightarrow \quad \theta_1 = \arctan\frac{y}{x} - \gamma$$

$$\theta_2 = \arctan\left(\frac{y - l_1\sin\theta}{x - l_1\cos\theta_1}\right) - \theta_1$$

- Mapping algorithm
  - Angle is input, sensor value is output so that arm can move to that point in it’s linear workspace
  - Calibrated using min/max position of arm

Physical Arm Construction: Dan

- Implemented arm design v1:
  - Used 8020 aluminum frame
  - Two linear actuators
  - One DC motor with gearbox
  - Chain drive for base rotation
  - Wooden Enclosure and base



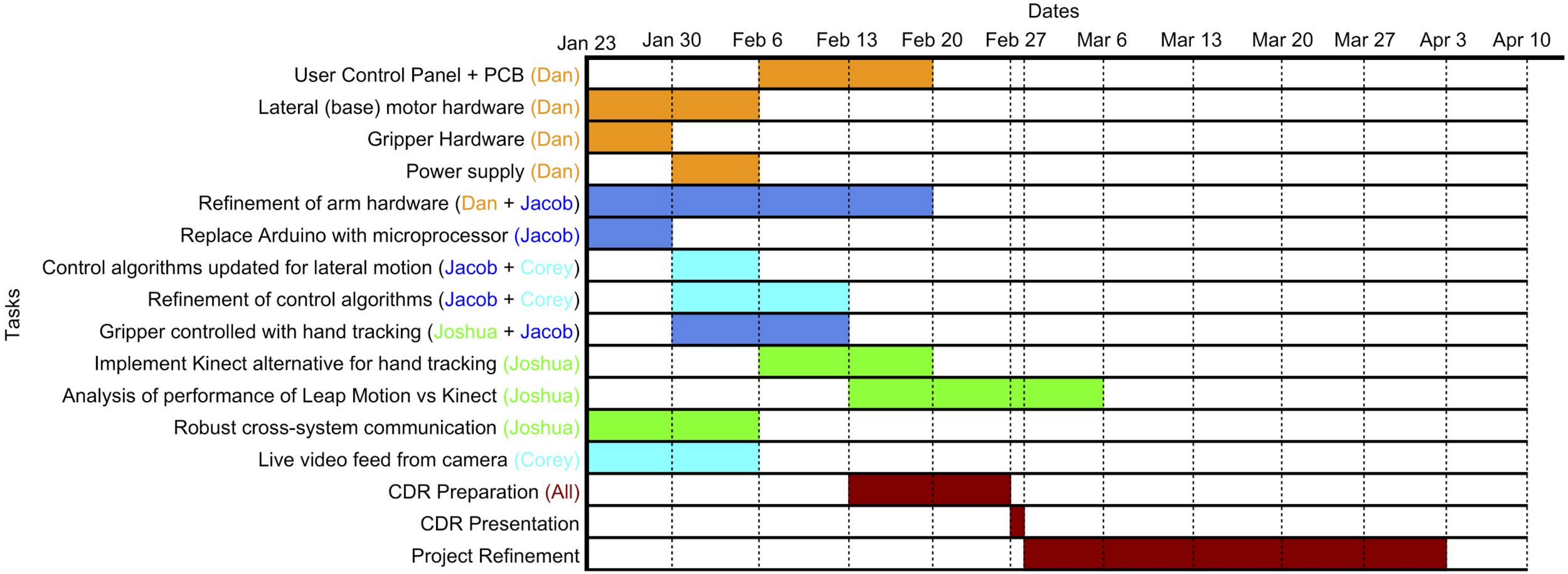
User Control Panel PCB: Dan

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

Proposed: CDR deliverables

- Integration of base motor into control algorithms to provide positioning in 3DOF
- Integration of gripper into system: Gripper state (open/closed) will be controlled by the user opening and closing their hand
- Implementation of live video feed from arm to user allowing them to use the arm remotely
- Arm will perform task as described in specifications slide within the 5 min timeframe

Gantt chart


Dates
Jan 23
Jan 30
Feb 6
Feb 13
Feb 20
Feb 27
Mar 6
Mar 13
Mar 20
Mar 27
Apr 3
Apr 10
Tasks
User Control Panel + PCB (Dan)
Lateral (base) motor hardware (Dan)
Gripper Hardware (Dan)
Power supply (Dan)
Refinement of arm hardware (Dan + Jacob)
Replace Arduino with microprocessor (Jacob)
Control algorithms updated for lateral motion (Jacob + Corey)
Refinement of control algorithms (Jacob + Corey)
Gripper controlled with hand tracking (Joshua + Jacob)
Implement Kinect alternative for hand tracking (Joshua)
Analysis of performance of Leap Motion vs Kinect (Joshua)
Robust cross-system communication (Joshua)
Live video feed from camera (Corey)
CDR Preparation (All)
CDR Presentation
Project Refinement

Demo

Thank You

Questions