# Group P22: Pick ‘n’ Place Robot Competition

Brian Herrmann, Albert Olea, Augustine Marfo, Tamal Kumar, Saman Hooshyar
Sponsor: Professor Brown
October 27, 2017

# Project Explanation

- Design a robotic arm capable of picking up an object from one location, and then placing the object in another location
- Arduino controlled servo/stepper motors
- Additive and/or Subtractive manufacturing for linkages
- $250.00 USD Budget
- Final arm must excel in the following areas:
  - Speed
  - Accuracy
  - Precision
  - Minimal Cost

# About the Sponsor

## Michael A. Brown, PhD, PE

- Experienced in computational dynamics, biomechanics, and a variety of aspects of manufacturing.

## Desired Outcomes of Project

The goal of this project is to gain a fundamental understanding of multiple DOF systems and controls. Designing a 6 DOF robotic arm will require knowledge of programming, kinematics, cost analysis, performance analysis, and electronics. A great deal of these topics are covered in the current curriculum, but some will require individual research efforts. Overall, by the end of the project, all group members will be able to demonstrate valuable engineering skills in the working world.

# Customer Requirements

## Description

A three-link servo/stepper motor controlled mechanism. The mechanism must be able to pick up an object, move it a certain distance, and place it at the new location without damaging said object. The following customer requirements must be met...

**Safe** - The mechanism should be able to be handled safely by a child

**Minimal cost** - The project should cost no more than $250 to assemble

**Functional** - The mechanism should be able to perform its objective efficiently without error

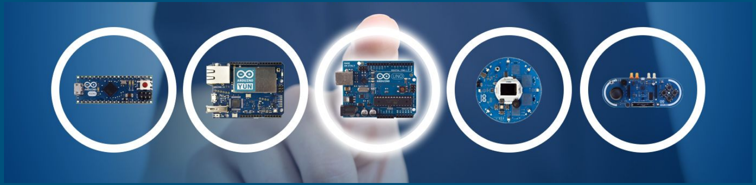
**Durable** - The mechanism should be able to perform multiple trials (100) without need for repairs.

**Aesthetic** - It should be appealing enough for someone to want to purchase a model.

# Background Information

Arduinos:

- Programming language based on C/C++
- Open-source software
- Open-source hardware
- Inexpensive: Official Arduinos range from $15-100 USD
    - Third party platforms start around $5 USD



# Arduinos cont.

Points of Consideration:

- Form Factor
- # of Analog In/Out
- # Digital IO/PWM
- RAM
- Flash Memory

https://www.arduino.cc/en/Products/Compare

| Name | Processor | Operating/Input Voltage | CPU Speed | Analog In/Out | Digital IO/PWM | EEPROM [kB] | SRAM [kB] | Flash [kB] | USB | UART |
|---|---|---|---|---|---|---|---|---|---|---|
| 101 | Intel® Curie | 3.3 V/ 7-12V | 32MHz | 6/0 | 14/4 | - | 24 | 196 | Regular | - |
| Gemma | ATtiny85 | 3.3 V / 4-16 V | 8 MHz | 1/0 | 3/2 | 0.5 | 0.5 | 8 | Micro | 0 |
| LilyPad | ATmega168V<br>ATmega328P | 2.7-5.5 V /<br>2.7-5.5 V | 8MHz | 6/0 | 14/6 | 0.512 | 1 | 16 | - | - |
| LilyPad SimpleSnap | ATmega328P | 2.7-5.5 V /<br>2.7-5.5 V | 8 MHz | 4/0 | 9/4 | 1 | 2 | 32 | - | - |
| LilyPad USB | ATmega32U4 | 3.3 V / 3.8-5 V | 8 MHz | 4/0 | 9/4 | 1 | 2.5 | 32 | Micro | - |
| Mega 2560 | ATmega2560 | 5 V / 7-12 V | 16 MHz | 16/0 | 54/15 | 4 | 8 | 256 | Regular | 4 |
| Micro | ATmega32U4 | 5 V / 7-12 V | 16 MHz | 12/0 | 20/7 | 1 | 2.5 | 32 | Micro | 1 |
| MKR1000 | SAMD21<br>Cortex-M0+ | 3.3 V/ 5V | 48MHz | 7/1 | 8/4 | - | 32 | 256 | Micro | 1 |
| Pro | ATmega168<br>ATmega328P | 3.3 V / 3.35-12 V<br>5 V / 5-12 V | 8 MHz<br>16 MHz | 6/0 | 14/6 | 0.512<br>1 | 1<br>2 | 16<br>32 | - | 1 |
| Pro Mini | ATmega328P | 3.3 V / 3.35-12 V<br>5 V / 5-12 V | 8 MHz<br>16 MHz | 6/0 | 14/6 | 1 | 2 | 32 | - | 1 |
| Uno | ATmega328P | 5 V / 7-12 V | 16 MHz | 6/0 | 14/6 | 1 | 2 | 32 | Regular | 1 |
| Zero | ATSAMD21G18 | 3.3 V / 7-12 V | 48 MHz | 6/1 | 14/10 | - | 32 | 256 | 2 Micro | 2 |

# Background Information cont.

**Materials** : Certain materials can be used in the design of our mechanism based on their respective properties.

- Carbon fiber - Expensive ($11-16 per Kg), Lightweight (Density = 1.6 g/cm3) , Strong (810 MPa Tensile strength, 200 MPa Yield Strength)
- Aluminum - Cheap ($2 per Kg), Lightweight (Density = 2.7 g/cm3), Strong (90 MPa tensile strength, 48.3MPa Yield strength), Corrosion resistant.
- Steel - Expensive ($6 per Kg), Heavy (Density =  8.05 g/cm3), Stronger than Al (696 MPa tensile strength, 301 MPa yield strength)
- Copper - Expensive ( $7 per Kg), Heavy (8.96 g/cm3), Strong (331 Mpa Tensile strength, 310 MPa Yield Strength)
- PVC - Cheap ($1.55 per Kg), Lightweight (1.48 g/cm3), Weak (52 MPa Tensile strength, 8 MPa Yield strength)

# Materials cont.

| | Unit | Steel | Aluminum | Magnesium | Titanium |
|---|---|---|---|---|---|
| Density, [kg/dm³] | ρ | 7.83 | 2.8 | 1.74 | 4.5 |
| Young's modulus, [GPa] | E | 210 | 70 | 45 | 110 |
| Tensile strength, [N/mm²] | Rm | 300-1200 | 150-680 | 100-380 | 910-1190 |
| Specific strength, [10³N mm/kg] | Rm/ρ | 38-153 | 54-243 | 57-218 | 202-264 |
| Specific stiffness, [10⁹Nmm/kg] | E/ρ | 26.8 | 25.0 | 25.9 | 24.4 |

Figure 2

**Mechanical Properties of Carbon Fibre Composite Materials, Fibre / Epoxy resin (120°C Cure)**

Fibres @ 0° (UD), 0/90° (fabric) to loading axis, Dry, Room Temperature, Vf = 60% (UD), 50% (fabric)

| | Symbol | Units | Std CF Fabric | HMCF Fabric | E glass Fabric | Kevlar Fabric | Std CF UD | HMCF UD | M55** UD | E glass UD | Kevlar UD | Boron UD | Steel S97 | Al. L65 | Tit. dtd 5173 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Young's Modulus 0° | E1 | GPa | 70 | 85 | 25 | 30 | 135 | 175 | 300 | 40 | 75 | 200 | 207 | 72 | 110 |
| Young's Modulus 90° | E2 | GPa | 70 | 85 | 25 | 30 | 10 | 8 | 12 | 8 | 6 | 15 | 207 | 72 | 110 |
| In-plane Shear Modulus | G12 | GPa | 5 | 5 | 4 | 5 | 5 | 5 | 5 | 4 | 2 | 5 | 80 | 25 | |
| Major Poisson's Ratio | v12 | | 0.10 | 0.10 | 0.20 | 0.20 | 0.30 | 0.30 | 0.30 | 0.25 | 0.34 | 0.23 | | | |
| Ult. Tensile Strength 0° | Xt | MPa | 600 | 350 | 440 | 480 | 1500 | 1000 | 1600 | 1000 | 1300 | 1400 | 990 | 460 | |
| Ult. Comp. Strength 0° | Xc | MPa | 570 | 150 | 425 | 190 | 1200 | 850 | 1300 | 600 | 280 | 2800 | | | |
| Ult. Tensile Strength 90° | Yt | MPa | 600 | 350 | 440 | 480 | 50 | 40 | 50 | 30 | 30 | 90 | | | |
| Ult. Comp. Strength 90° | Yc | MPa | 570 | 150 | 425 | 190 | 250 | 200 | 250 | 110 | 140 | 280 | | | |
| Ult. In-plane Shear Stren. | S | MPa | 90 | 35 | 40 | 50 | 70 | 60 | 75 | 40 | 60 | 140 | | | |
| Ult. Tensile Strain 0° | ext | % | 0.85 | 0.40 | 1.75 | 1.60 | 1.05 | 0.55 | | 2.50 | 1.70 | 0.70 | | | |
| Ult. Comp. Strain 0° | exc | % | 0.80 | 0.15 | 1.70 | 0.60 | 0.85 | 0.45 | | 1.50 | 0.35 | 1.40 | | | |
| Ult. Tensile Strain 90° | eyt | % | 0.85 | 0.40 | 1.75 | 1.60 | 0.50 | 0.50 | | 0.35 | 0.50 | 0.60 | | | |
| Ult. Comp. Strain 90° | eyc | % | 0.80 | 0.15 | 1.70 | 0.60 | 2.50 | 2.50 | | 1.35 | 2.30 | 1.85 | | | |
| Ult. In-plane shear strain | es | % | 1.80 | 0.70 | 1.00 | 1.00 | 1.40 | 1.20 | | 1.00 | 3.00 | 2.80 | | | |
| Thermal Exp. Co-ef. 0° | Alpha1 | Strain/K | 2.10 | 1.10 | 11.60 | 7.40 | -0.30 | -0.30 | -0.30 | 6.00 | 4.00 | 18.00 | | | |
| Thermal Exp. Co-ef. 90° | Alpha2 | Strain/K | 2.10 | 1.10 | 11.60 | 7.40 | 28.00 | 25.00 | 28.00 | 35.00 | 40.00 | 40.00 | | | |
| Moisture Exp. Co-ef 0° | Beta1 | Strain/K | 0.03 | 0.03 | 0.07 | 0.07 | 0.01 | 0.01 | | 0.01 | 0.04 | 0.01 | | | |
| Moisture Exp. Co-ef 90° | Beta2 | Strain/K | 0.03 | 0.03 | 0.07 | 0.07 | 0.30 | 0.30 | | 0.30 | 0.30 | 0.30 | | | |
| Density | | g/cc | 1.60 | 1.60 | 1.90 | 1.40 | 1.60 | 1.60 | 1.65 | 1.90 | 1.40 | 2.00 | | | |

*** Calculated figures*

Fibres @ +/-45 Deg. to loading axis, Dry, Room Temperature, Vf = 60% (UD), 50% (fabric)

| | Symbol | Units | Std. CF | HM CF | E Glass | Std. CF fabric | E Glass fabric | Steel | Al |
|---|---|---|---|---|---|---|---|---|---|
| Longitudinal Modulus | E1 | GPa | 17 | 17 | 12.3 | 19.1 | 12.2 | 207 | 72 |
| Transverse Modulus | E2 | GPa | 17 | 17 | 12.3 | 19.1 | 12.2 | 207 | 72 |
| In Plane Shear Modulus | G12 | GPa | 33 | 47 | 11 | 30 | 8 | 80 | 25 |
| Poisson's Ratio | v12 | | .77 | .83 | .53 | .74 | .53 | | |
| Tensile Strength | Xt | MPa | 110 | 110 | 90 | 120 | 120 | 990 | 460 |
| Compressive Strength | Xc | MPa | 110 | 110 | 90 | 120 | 120 | 990 | 460 |
| In Plane Shear Strength | S | MPa | 260 | 210 | 100 | 310 | 150 | | |
| Thermal Expansion Co-ef | Alpha1 | Strain/K | 2.15 E-6 | 0.9 E-6 | 12 E-6 | 4.9 E-6 | 10 E-6 | 11 E-6 | 23 E-6 |
| Moisture Co-ef | Beta1 | Strain/K | 3.22 E-4 | 2.49 E-4 | 6.9 E-4 | | | | |

*** Calculated figures*

These tables are for reference / information only and are **NOT** a guarantee of performance
1 GPa = 1000 MPa = 1000 N/mm² = 145,000 PSI

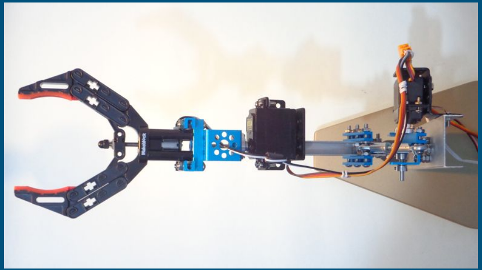
These tables relate to only 2 of the many fibre orientations possible. Most components are made using combinations of the above materials and with the fibre orientations being dictated by the performance requirements of the product. Performance Composites Ltd. can assist with the design of components where appropriate.

Figure 3

# Gripper Options


## Only Gears

- Two gears with equal teeth count with one gear screwed to the output spline of servo motor
- Most commonly used in similar projects because relatively simple and is fast
- https://youtu.be/sIP9ujByaLI?t=11s



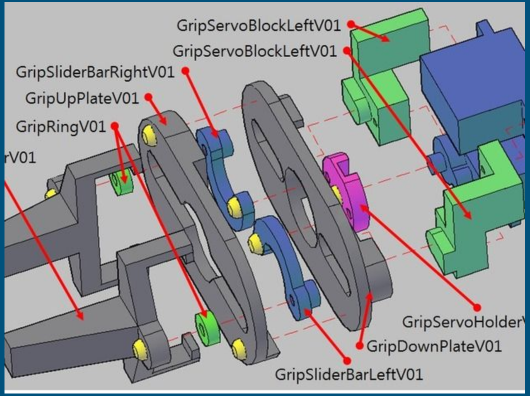
## Gear and Screw

- Threaded shaft connected to servo motor. When screw is lowered, the arms contract.
- Isn’t as common, but has a professional model to look off of
- Slower and not as strong as solely gear option
- https://www.youtube.com/watch?v=9xSvRSg7VZA

# Gripper Options Continued

## Through Connections

- No gears or screws. All parts connected to the one part rotating from servo motor.
- Small range of motion and not a strong grip
- Least amount of effort out of other options [https://youtu.be/aId8dsG8fkk?t=17s](https://youtu.be/aId8dsG8fkk?t=17s)




## General Info

- Shape of claw depends on the shape and size of the object
- Most similar projects add a motor at base of claw so that it can rotate 360 degrees
- Material will depend on object characteristics

# Motor Comparisons

## Servo Motors

- Functional at high speeds
- Accuracy/precision relies on the feedback/encoder system
- Maintains rated torque to about 90% of no load RPM
- Servos can malfunction if overloaded mechanically

## Stepper Motors

- High holding torque
- Lose strength at higher RPMs
- Ideal for repeatability and precision (provided microstepping is utilized)
- Encoder can be added to detect incremental or absolute system information
- With no feedback system, a stepper motor can lose positioning when overloaded

# Motor Comparisons Cont.

## Incremental Encoders

- Often simple systems
- Cost effective due to their simplicity
- Determine “If the position/velocity has changed”

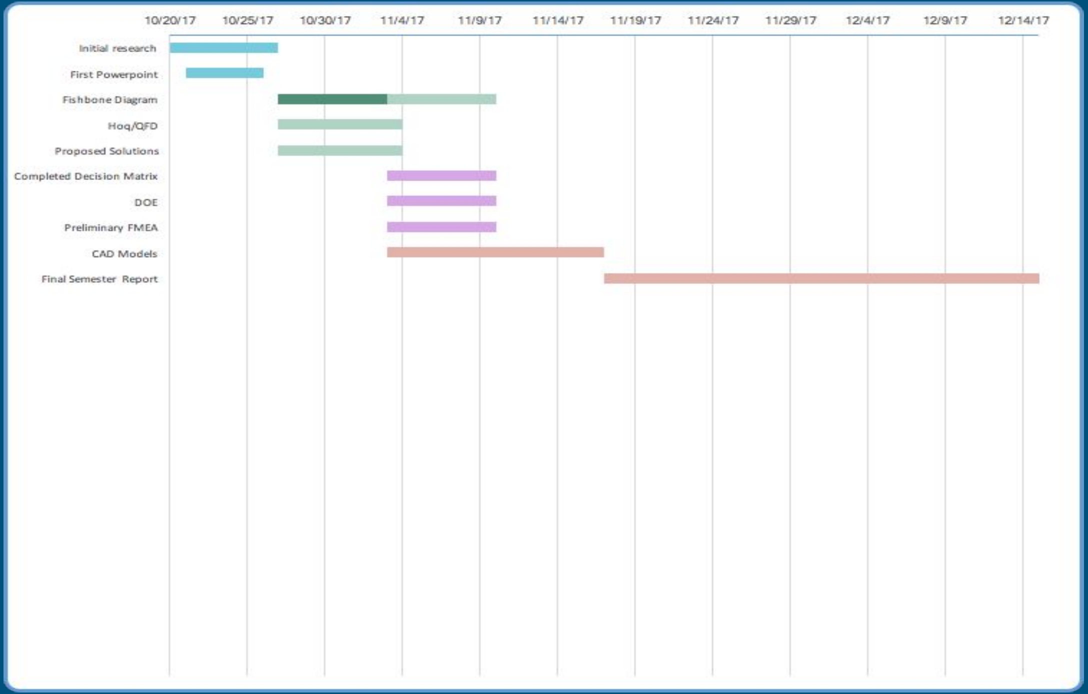
## Absolute Encoders

- Determines “How much has the position/velocity changed”
- Can still “remember” output information after a system powers down
- Often more complex systems and therefore costlier

## Resolvers

- Uses changes in magnetic fields to detect position changes
- Older technology (has been around since WW2)
- Requires additional specialized components to obtain useful output information, thus driving up costs

# Gantt Chart (As of 10/29/2017)



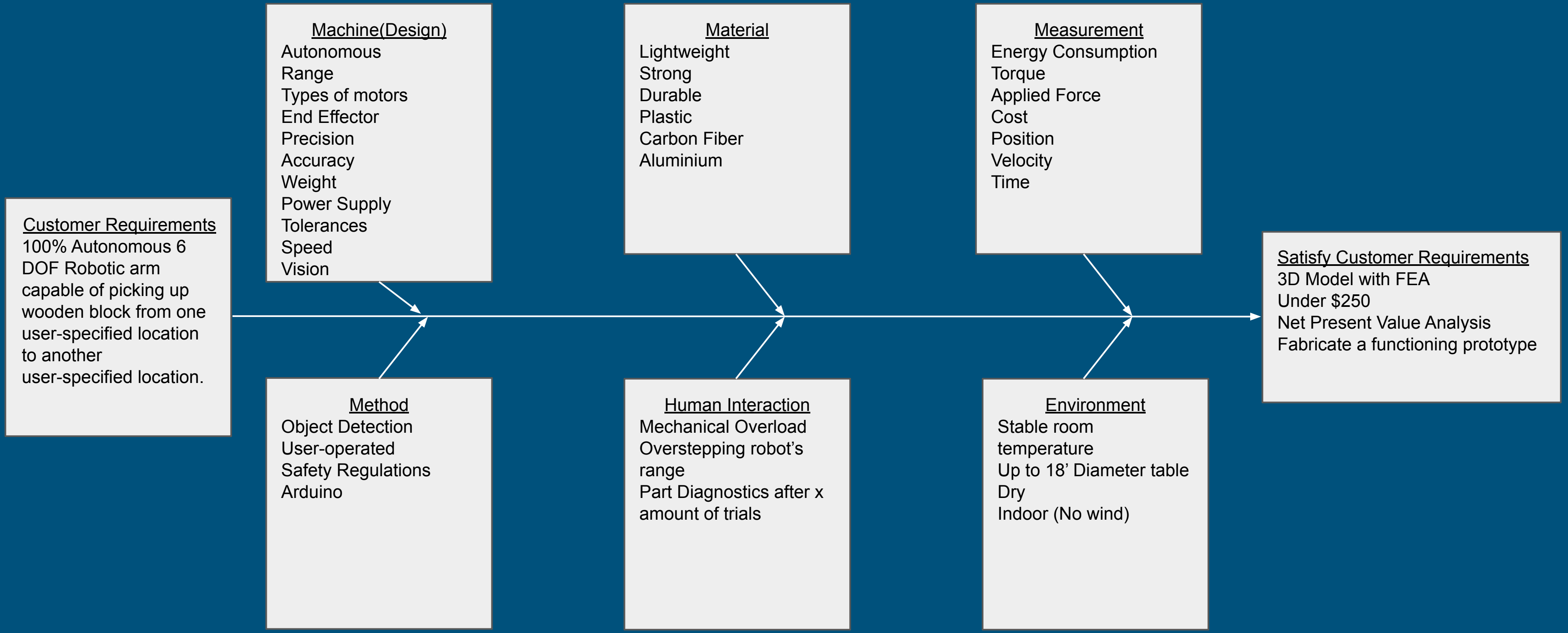
# Metrics

| Metric | Description |
|---|---|
| Cost | $ amount for entire product (parts, assembly etc.) |
| Speed | Time in seconds for each process (initial movement, pincher contraction etc.) |
| Strength | Maximum weight that can be lifted as well as the torque of each motor |
| Accuracy | Distance measurement of how close we place block to desired location |

# Design Considerations/Criteria

- Material for Arms: Carbon Fiber, ABS, PLA
- Arm Design: What the parts look like and how they fit together
- Motors Used: Stepper or servo motor used for arm movement
- Arduino Board: Ability to support multiple drivers for steppers, voltage/current limits
- “Gripper” Design: Pincher, suction, scoop

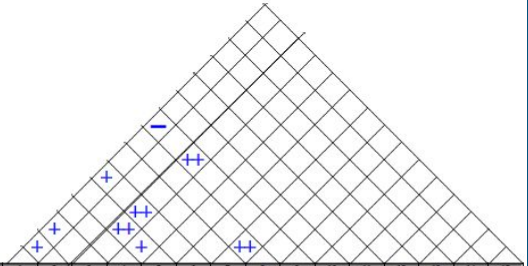
# Fishbone Diagram



# QFD/HoQ

Demanded Quality Vs The Means of Achievement

- Determining motors has the most weight
  - Motor leads to energy considerations
    - Speed Capabilities
- Object Detection is the second most concerning
- Cost will be minimized based on the parts needed




| | | | | Column # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Direction of Improvement: Minimize (▼), Maximize (▲), or Target (x) | ▼ | ▲ | x | ▼ | ▼ | ▼ | ▼ | ▼ | x | | | | | | |
| Row # | Max Relationship Value in Row | Relative Weight | Weight / Importance | Quality Characteristics (a.k.a. "Functional Requirements" or "Hows") / Demanded Quality (a.k.a. "Customer Requirements" or "Whats") | Material | Object Detection | Energy Requirement | Time for Arm Movement | Time for Gripper Movement | Weight of arms | Material Cost | Cost of Assembly | Motor Type | | | | | | |
| 1 | 9 | 8.0 | | Object Speed | O | ▲ | Θ | Θ | Θ | O | ▲ | ▲ | Θ | | | | | | |
| 2 | 9 | 8.0 | | Accuracy & Precision | ▲ | Θ | O | Θ | Θ | ▲ | ▲ | ▲ | Θ | | | | | | |
| 3 | 9 | 7.0 | | Two-Finger Gripper | O | O | Θ | ▲ | Θ | ▲ | O | O | Θ | | | | | | |
| 4 | 9 | 9.0 | | Cost | Θ | O | Θ | Θ | Θ | Θ | Θ | Θ | Θ | | | | | | |
| 5 | 9 | 6.0 | | Safety | Θ | ▲ | O | O | O | O | ▲ | ▲ | O | | | | | | |
| 6 | 9 | 8.0 | | Ease of Use | ▲ | Θ | Θ | Θ | Θ | ▲ | ▲ | ▲ | Θ | | | | | | |
| 7 | 9 | 9.0 | | Autonomus | ▲ | Θ | Θ | ▲ | ▲ | ▲ | Θ | Θ | Θ | | | | | | |
| 8 | 9 | 8.0 | | Strength | Θ | ▲ | Θ | O | O | Θ | Θ | Θ | Θ | | | | | | |
| 9 | 9 | 8.0 | | Durability | Θ | ▲ | Θ | O | O | Θ | Θ | Θ | Θ | | | | | | |
| 10 | 9 | 9.0 | | Range of motion | O | Θ | Θ | Θ | O | O | Θ | Θ | Θ | | | | | | |
| | | | | Target or Limit Value | | | | | | | | | | | | | | | |
| | | | | Difficulty (0=Easy to Accomplish, 10=Extremely Difficult) | 3 | 9 | 2 | 8 | 7 | 3 | 8 | 3 | 7 | | | | | | |
| | | | | Max Relationship Value in Column | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | | | | | | |
| | | | | Weight / Importance | 376.0 | 384.0 | 636.0 | 460.0 | 462.0 | 326.0 | 438.0 | 438.0 | 684.0 | | | | | | |
| | | | | Relative Weight | 8.9 | 9.1 | 15.1 | 10.9 | 11.0 | 7.8 | 10.4 | 10.4 | 16.3 | | | | | | |

# Proposed Solution (Saman Hooshyar)

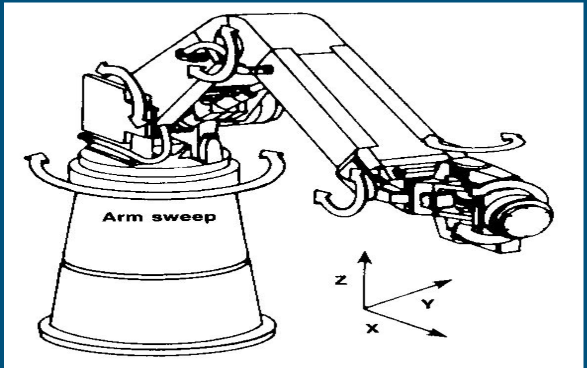
## Three link pick and place robot arm with camera

- Consists of four stepper motors and one servo motor.  Stepper motors will be implemented on rotational base, shoulder, upper arm, and lower arm.  Servo motor will be used to move two finger gripper.
  - This solution is based on initial project description which was picking rectangular block placed between two semi circular segments with 2” radius difference and place it at designated spot and bring it back to its initial position without crossing smaller semi circle.

- Employs finger tracking method [1]

- OpenCV and C programming language [1]

- Motors specifications and Arduino model will be decided after sponsor provides detailed description of new project.




Three linked pick and place robot arm with two finger gripper [2]

Robot arm with camera to detect hand gesture and desired object [3]

# Proposed Solution (Saman Hooshyar)

References

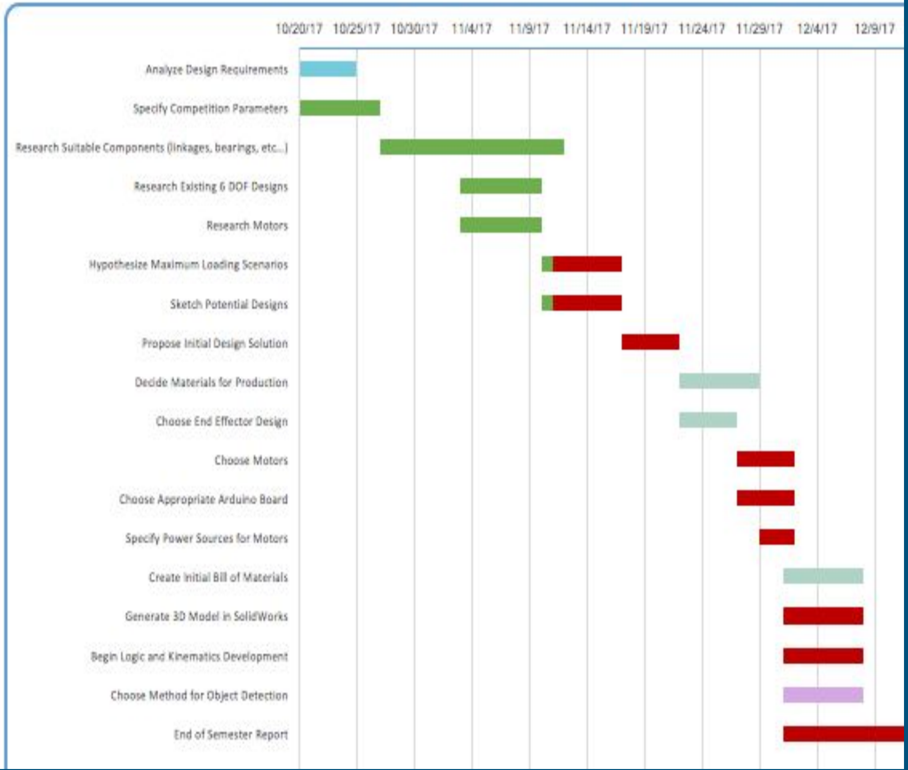
[1] A. Dhawan, A. Bhat, S. Sharma, and H. K. Kaura, "Automated Robot with Object Recognition and Handling Features", *International Journal of Electronics and Computer Science Engineering, vol. 2, no. 3, pp. 861-873.* [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.433.9442&rep=rep1&type=pdf

[2] *Material Handling Equipment,* September 1999. [Online]. Available: http://www4.ncsu.edu/~kay/mhetax/PosEq/index.htm

[3] J. Happich, "Depth-sensing robot arm camera improves situational awareness", May 2016. [Online]. Available: http://www.smart2zero.com/news/depth-sensing-robot-arm-camera-improves-situational-awareness

# Gantt Chart (As of 11/17/2017)

| | | | | | | |
|---|---|---|---|---|---|---|
| Analyze Design Requirements | 10/20/17 | 10/25/17 | 5 | 5.00 | 0.00 | 75% |
| Specify Competition Parameters | 10/20/17 | 10/27/17 | 7 | 7.00 | 0.00 | 50% |
| Research Suitable Components (linkages, bearings, etc...) | 10/27/17 | 11/10/17 | 16 | 16.00 | 0.00 | 75% |
| Research Existing 6 DOF Designs | 11/3/17 | 11/10/17 | 7 | 7.00 | 0.00 | 100% |
| Research Motors | 11/3/17 | 11/10/17 | 7 | 7.00 | 0.00 | 100% |
| Hypothesize Maximum Loading Scenarios | 11/10/17 | 11/17/17 | 7 | 1.00 | 6.00 | 10% |
| Sketch Potential Designs | 11/10/17 | 11/17/17 | 7 | 1.00 | 6.00 | 0% |
| Propose Initial Design Solution | 11/17/17 | 11/22/17 | 5 | 0.00 | 5.00 | 0% |
| Decide Materials for Production | 11/22/17 | 11/27/17 | 5 | 0.00 | 7.00 | 0% |
| Choose End Effector Design | 11/22/17 | 11/27/17 | 5 | 0.00 | 5.00 | 0% |
| Choose Motors | 11/27/17 | 12/1/17 | 5 | 0.00 | 5.00 | 0% |
| Choose Appropriate Arduino Board | 11/27/17 | 12/1/17 | 5 | 0.00 | 5.00 | 0% |
| Specify Power Sources for Motors | 11/29/17 | 12/1/17 | 3 | 0.00 | 3.00 | 0% |
| Create Initial Bill of Materials | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |
| Generate 3D Model in SolidWorks | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |
| Begin Logic and Kinematics Development | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |
| Choose Method for Object Detection | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |
| End of Semester Report | 12/1/17 | 12/15/17 | 14 | 0.00 | 14.00 | 0% |



# References

[1] Gingichashvili, Sarah. “A New Robotic Hand.” *The Future Of Things*, The Future Of Things, 6 Aug. 2007, thefutureofthings.com/5605-a-new-robotic-hand/.

[2] Staff, Atmel. “Arduino Board Comparisons: Picking the Right Board.” *PubNub*, PubNub, 17 Apr. 2015, www.pubnub.com/blog/2015-04-17-arduino-board-comparisons-picking-the-right-board/.

http://www.ni.com/white-paper/3656/enhhttps://search-proquest-com.proxy.cc.uic.edu/abiglobal/docview/218410879/fulltextPDF/1412805D25314D37PQ/1?accountid=14552ttp://www.ni.com/white-paper/3656/en/#toc1/#toc1

https://search-proquest-com.proxy.cc.uic.edu/abiglobal/docview/218410879/fulltextPDF/1412805D25314D37PQ/1?accountid=14552

Materials

http://homepages.which.net/~paul.hills/Materials/MaterialsBody.html

http://www.performance-composites.com/carbonfibre/mechanicalproperties_2.asp

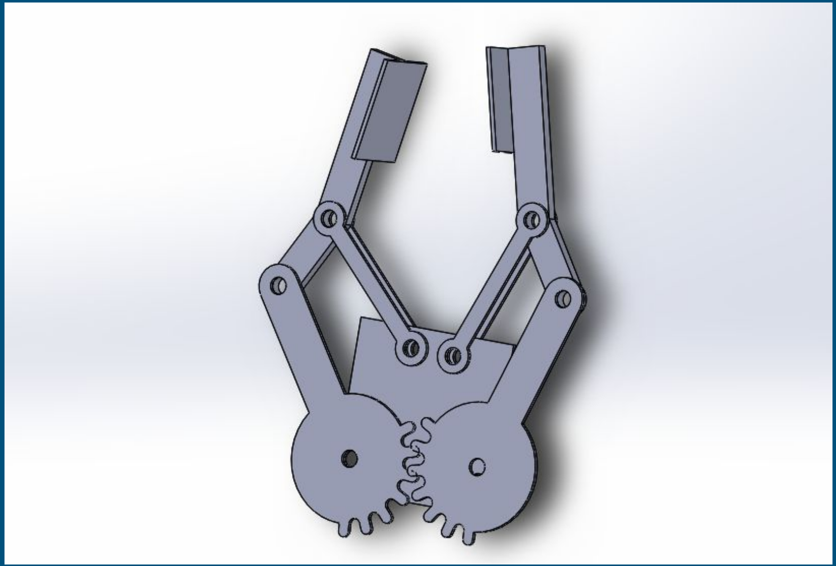
https://www.alibaba.com/showroom/pvc-price-kg.html

# Decision Matrix (Grippers)

| | Speed | Accuracy | Precision | Cost | Safety | Rank |
|---|---|---|---|---|---|---|
| **Weighting Factor** | **0.125** | **0.125** | **0.125** | **0.350** | **0.275** | **1.000** |
| **Suction** | 5 | 6 | 4 | 5 | 7 | |
| **Weight** | 0.625 | 0.75 | 0.5 | 1.75 | 1.925 | 5.550 |
| **Scooper** | 7 | 8 | 4 | 8 | 6 | |
| **Weight** | 0.875 | 1 | 0.5 | 2.8 | 1.65 | 6.825 |
| **Magnetic** | 10 | 5 | 7 | 6 | 5 | |
| **Weight** | 1.25 | 0.625 | 0.875 | 2.1 | 1.375 | 6.225 |
| **2-Finger Pincher** | 6 | 9 | 9 | 8 | 6 | |
| **Weight** | 0.75 | 1.125 | 1.125 | 2.8 | 1.65 | 7.450 |
| **Adhesive** | 8 | 7 | 6 | 4 | 9 | |
| **Weight** | 1 | 0.875 | 0.75 | 1.4 | 2.475 | 6.500 |

# 2-Finger Gripper

- 2 gears with servo attached to one gear
- Pinchers are wide to account for inaccuracy
- How wide the gripper can open depends on how many teeth are added to gears and by length of both stabilizing bars.
- Can contract to the point of direct contact between both pinchers
- https://youtu.be/sIP9ujByaLI
- Can be made of metal or plastic



# Gantt Chart (As of 11/27/17)

| Task Name | Start Date | End Date | Duration (Days) | Days Complete | Days Remaining | Percent Complete |
|---|---|---|---|---|---|---|
| Analyze Design Requirements | 10/20/17 | 10/25/17 | 5 | 5.00 | 0.00 | 75% |
| Specify Competition Parameters | 10/20/17 | 10/27/17 | 7 | 7.00 | 0.00 | 50% |
| Research Suitable Components (linkages, bearings, etc...) | 10/27/17 | 11/10/17 | 16 | 16.00 | 0.00 | 75% |
| Research Existing 6 DOF Designs | 11/3/17 | 11/10/17 | 7 | 7.00 | 0.00 | 100% |
| Research Motors | 11/3/17 | 11/10/17 | 7 | 7.00 | 0.00 | 100% |
| Hypothesize Maximum Loading Scenarios | 11/10/17 | 11/17/17 | 7 | 7.00 | 0.00 | 10% |
| Sketch Potential Designs | 11/10/17 | 11/17/17 | 7 | 7.00 | 0.00 | 25% |
| Propose Initial Design Solution | 11/17/17 | 11/22/17 | 5 | 5.00 | 0.00 | 0% |
| Decide Materials for Production | 11/22/17 | 11/27/17 | 5 | 5.00 | 0.00 | 25% |
| Choose End Effector Design | 11/22/17 | 11/27/17 | 5 | 5.00 | 0.00 | 100% |
| Choose Motors | 11/27/17 | 12/1/17 | 5 | 1.00 | 4.00 | 0% |
| Choose Appropriate Arduino Board | 11/27/17 | 12/1/17 | 5 | 1.00 | 4.00 | 0% |
| Specify Power Sources for Motors | 11/29/17 | 12/1/17 | 3 | 0.00 | 3.00 | 0% |
| Create Initial Bill of Materials | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |
| Generate 3D Model in SolidWorks | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |
| Begin Logic and Kinematics Development | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |
| Choose Method for Object Detection | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |
| End of Semester Report | 12/1/17 | 12/8/17 | 7 | 0.00 | 7.00 | 0% |

| Start Date in Number Form | 43028.00 |
|---|---|

*Use this number for the Minimum Bound of the Horizontal Axis to set the beginning of the chart.*

# Group P22: Pick ‘n’ Place Robot Competition

Brian Herrmann, Albert Olea, Augustine Marfo, Tamal Kumar, Saman Hooshyar
Sponsor: Professor Brown
Advisor Meeting: January 23, 2018

# Assignments

Product Sourcing

Continue Practicing Arduino Programming

Equations of Motion

# Work Performed

Sourced a vision system

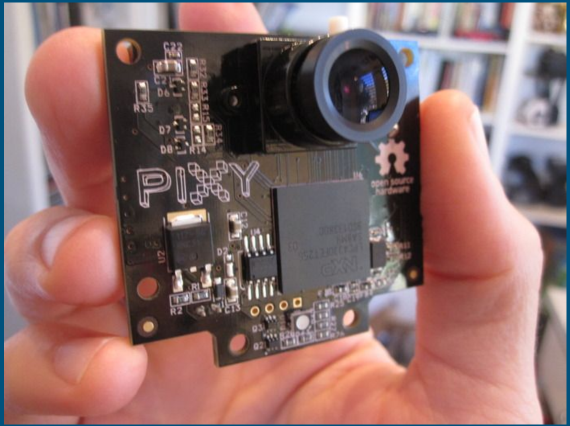
Pros:

- Small
- Connect to Arduino
- Object Learning
- Open-Source Software

Cons:

- Cost ($75)
- Hue Dependant

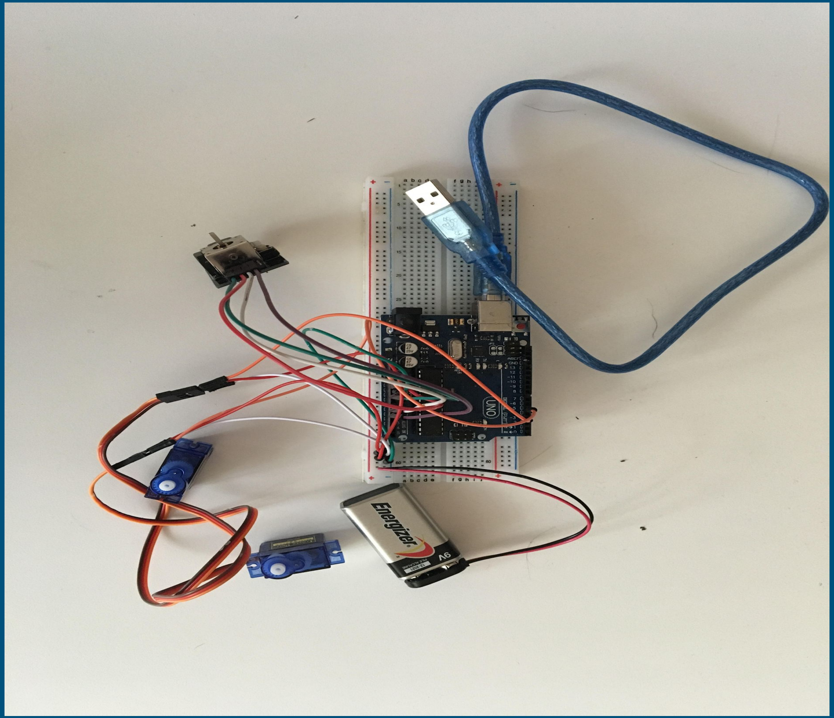
http://www.cmucam.org/projects/cmucam5




# Running Multiple Motors

Began running multiple motors using one arduino

- No sensors were used
- Single Toggle switch controlled two separate servo motors



# Future Work

**IMMEDIATE (ASAP)**

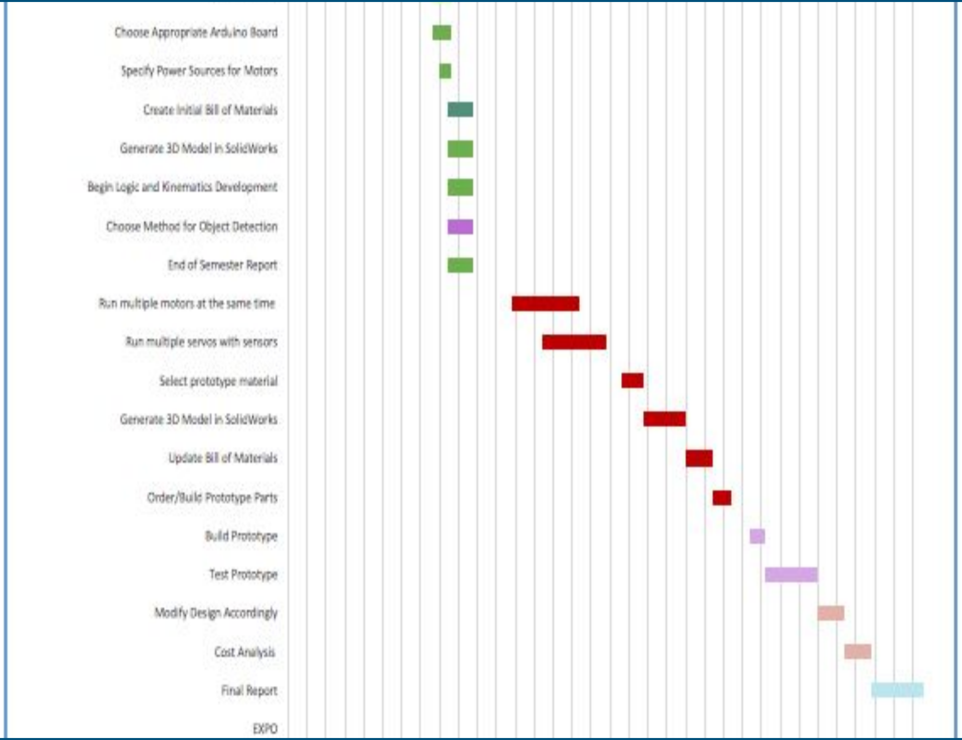
- Order materials / Begin filling out expense reports
- Update 3D Solidworks Model Design and Create code and wired arduino model that can use sensors, motors, ect. in a unified system

AFTER

- Begin building prototype
- Model the system and perform multiple test runs
- Win EXPO

# Gantt Chart (as of 1/23/18)

| | | | | | | |
|---|---|---|---|---|---|---|
| Specify Power Sources for Motors | 11/29/17 | 12/1/17 | 3 | 3.00 | 0.00 | 10% |
| Create Initial Bill of Materials | 12/1/17 | 12/8/17 | 7 | 7.00 | 0.00 | 0% |
| Generate 3D Model in SolidWorks | 12/1/17 | 12/8/17 | 7 | 7.00 | 0.00 | 25% |
| Begin Logic and Kinematics Development | 12/1/17 | 12/8/17 | 7 | 7.00 | 0.00 | 0% |
| Choose Method for Object Detection | 12/1/17 | 12/8/17 | 7 | 7.00 | 0.00 | 75% |
| End of Semester Report | 12/1/17 | 12/8/17 | 7 | 7.00 | 0.00 | 100% |
| Run multiple motors at the same time | 12/18/17 | 1/5/18 | 18 | 0.00 | 18.00 | 0% |
| Run multiple servos with sensors | 12/26/17 | 1/12/18 | 17 | 0.00 | 17.00 | 0% |
| Select prototype material | 1/16/18 | 1/22/18 | 6 | 0.00 | 6.00 | 0% |
| Generate 3D Model in SolidWorks | 1/22/18 | 2/2/18 | 11 | 0.00 | 11.00 | 0% |
| Update Bill of Materials | 2/2/18 | 2/9/18 | 7 | 0.00 | 7.00 | 0% |
| Order/Build Prototype Parts | 2/9/18 | 2/14/18 | 5 | 0.00 | 5.00 | 0% |
| Build Prototype | 2/19/18 | 2/23/18 | 4 | 0.00 | 4.00 | 0% |
| Test Prototype | 2/23/18 | 3/9/18 | 14 | 0.00 | 14.00 | 0% |
| Modify Design Accordingly | 3/9/18 | 3/16/18 | 7 | 0.00 | 7.00 | 0% |
| Cost Analysis | 3/16/18 | 3/23/18 | 7 | 0.00 | 7.00 | 0% |
| Final Report | 3/23/18 | 4/6/18 | 14 | 0.00 | 14.00 | 0% |
| EXPO | 4/28/18 | | | | | |



# Group P22: Pick ‘n’ Place Robot Competition

Brian Herrmann, Albert Olea, Augustine Marfo, Tamal Kumar, Saman Hooshyar
Sponsor: Professor Brown
Advisor Meeting: February 8, 2018

# Assignments

Completed Work

CAD Drawings

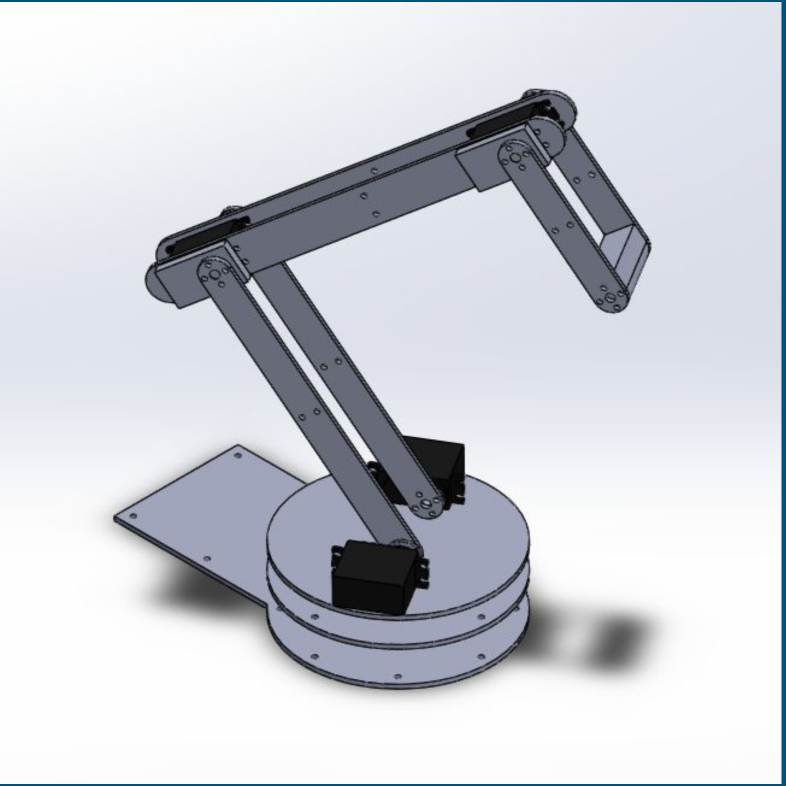
PixyCam Code

Material Selection

Inventory

# Work Performed: CAD

- 4x MG966R Motors
- 4x Aluminum Arm Horns
- 12x M3 6mm Female Hex Standoff
- 8x M3 22mm Male-Female Hex Standoff
- 4x M3 45mm Female Hex Standoff
- 2x M3 20mm Female Hex Standoff
- x M3 2.4mm ht. Hex Nut
- x M3 8mm Phillips Screws
- x M3 1mm Spacers
- 28449 mm^3 PLA
- 264324.73 mm^3 Acrylic



# Work Performed: Bill of Materials (BOM)

| Items | Quantity | Total Cost | Source |
|---|---|---|---|
| M3x6 Female hex standoff | 12 | $ 9.72 | McMasterCarr |
| M3x22 Male-Femal Hex Standoff | 8 | $ 14.96 | McMasterCarr |
| M3x45 Female Hex Standoff | 4 | $ 6.24 | McMasterCarr |
| M3x20 Female Hex Standoff | 2 | $ 1.72 | McMasterCarr |
| M3x8 Screws | 100 | $ 2.72 | McMasterCarr |
| M3 Nut | 100 | $ 2.12 | McMasterCarr |
| Washers | 100 | $ 1.62 | McMasterCarr |
| Arduino | 1 | $ 11.00 | Walmart |
| Servo Driver | 1 | $ 11.99 | Amazon |
| Servo | 5 | $ 37.48 | Amazon |
| Servo Horns | 5 | $ 8.99 | Amazon |
| Wire Leads | 10 | $ 7.95 | Amazon |
| Pixy CMUcam5 | 1 | $ 74.95 | Adafruit |
| Acrylic Sheets (24"x48") | 1 | $ 24.48 | ACME Plastics |
| Power Adapter for Servo | 1 | $ 5.99 | Amazon |
| Gripper Mechanism | 1 | ? | ? |
| | | | |
| Running Total | | $ 221.93 | |

# Work Performed: Pixy Cam

- Pixy cam not yet ordered
- Pixymon add on software downloaded
- Tracking code is being developed using pixy package in arduino
- Pixy cam is a fast digit sensor for robotic applications, it can
    - Detect specific objects
    - Detect hundreds of objects
    - Filter out unwanted objects
    - Connect to an arduino UNO

https://www.youtube.com/watch?v=DV4YK_Kk5IY

# Gripper Options

- All options only take 1 servo motor
- Vacuum-$8, Claw-$10, Scoop-$1
- Gripper is best, but hard to code
- Max Load might be affected by movement, vacuum viability unclear
- Recommend testing of vacuum which will be cheap

| | Speed | Accuracy & Precision | Difficulty to Code | Cost | Max Load | Rank |
|---|---|---|---|---|---|---|
| **Weighting Factor** | **0.175** | **0.175** | **0.125** | **0.250** | **0.275** | **1.000** |
| **Vacuum** | 10 | 8 | 9 | 6 | 7 | |
| **Weight** | 1.75 | 1.4 | 1.125 | 1.5 | 1.925 | 7.700 |
| **Claw** | 8 | 9 | 5 | 5 | 9 | |
| **Weight** | 1.4 | 1.575 | 0.625 | 1.25 | 2.475 | 7.325 |
| **Scoop** | 5 | 3 | 6 | 10 | 9 | |
| **Weight** | 0.875 | 0.525 | 0.75 | 2.5 | 2.475 | 7.125 |

# Future Work

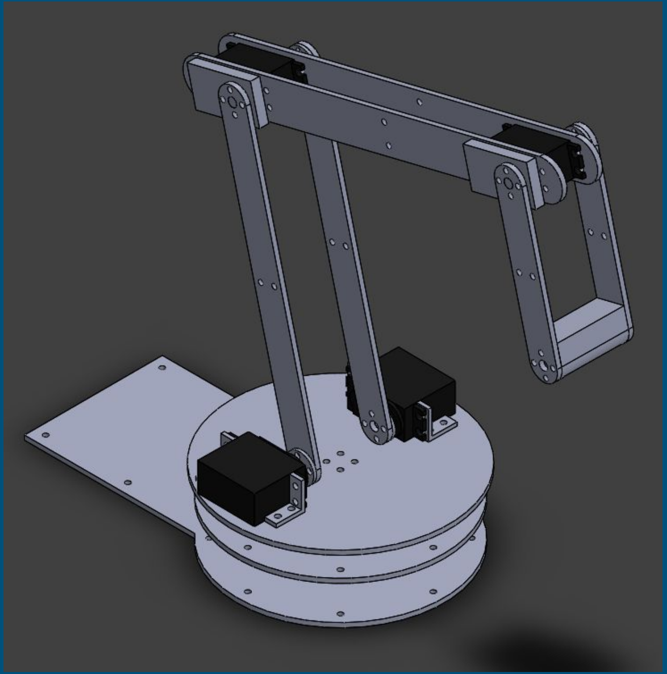
- Make Base motor brackets
- Holes for Pixy Mount
- Holes for Gripper Mount
- Download McMaster CAD Drawings for Screws etc.
- Submit buy orders
- Continue Coding

# Group P22: Pick ‘n’ Place Robot Competition

Brian Herrmann, Albert Olea, Augustine Marfo, Tamal Kumar, Saman Hooshyar
Sponsor: Professor Brown
Advisor Meeting: February 22, 2018

# Work Performed: CAD

- The base motors are now centered at point of rotation (this will simplify coding down the road)
- Mounting Brackets to be 3D printed
- Settled on 24” Maximum Reach (current model is about 20” without gripper)



# Work Performed: Bill of Materials (BOM)

- Significantly reduced cost from initial BOM
- All items have been bought except for gripper
- 3D Printing Cost?
- Shipping/Tax Cost?

| Item | Quantity | Total Cost |
|---|---|---|
| M3x8 Screw | 100 | $2.72 |
| M3 Nut | 100 | $2.12 |
| Washers | 100 | $1.62 |
| Arduino | 1 | $6.95 |
| Servo Driver | 1 | $5.75 |
| MG996R Servo | 8 | $34.50 |
| Servo Horns | 5 | $8.99 |
| 30mm Male-Female Jumper Wires | 40 | $4.99 |
| Pixy CMUcam5 | 1 | $74.95 |
| Acrylic Sheets (12"x24") | 1 | $9.82 |
| Power Adapter for Servo | 1 | $15.98 |
| Gripper Mechanism | 1 | ? |
| Standoff Assortment | 1 | $10.88 |
| Wire Wrap | 1 | $4.99 |
| | | |
| | | |
| | | $184.26 |

# Analyzing Motion

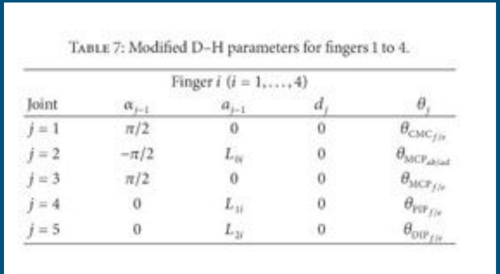
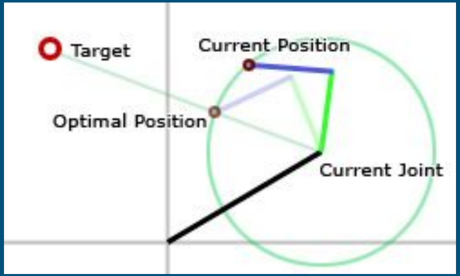


Table 7: Modified D-H parameters for fingers 1 to 4.

| | Finger $i$ ($i = 1, \ldots, 4$) | | | |
|---|---|---|---|---|
| Joint | $\alpha_{j-1}$ | $a_{j-1}$ | $d_j$ | $\theta_j$ |
| $j = 1$ | $\pi/2$ | 0 | 0 | $\theta_{CMC_{f/e}}$ |
| $j = 2$ | $-\pi/2$ | $L_{0i}$ | 0 | $\theta_{MCP_{ab/ad}}$ |
| $j = 3$ | $\pi/2$ | 0 | 0 | $\theta_{MCP_{f/e}}$ |
| $j = 4$ | 0 | $L_{1i}$ | 0 | $\theta_{PIP_{f/e}}$ |
| $j = 5$ | 0 | $L_{2i}$ | 0 | $\theta_{DIP_{f/e}}$ |




Forward (Direct) Kinematics

- Joint Space to Cartesian Space
- Denavit Hartenberg Convention
- Multiply a series of translation and rotation matrices to find matrix describing end effector location

Inverse Kinematics

- Cartesian Space to Joint Space (often more difficult to compute due to number of possible solutions
- Complexity increases with DoF
- Cyclic Coordinate Descent
- Jacobian Based Methods

# Work Performed: Max Torque

- 24” Maximum Reach Assumption
- Gripper and Object Weight Assumptions
- Most likely point of failure is 2nd motor

| Length(in) | Length (m) | Mass at Point (kg) | Produce Torque (kg*m) | Torqe on Motor (kg*cm) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 13.1415536 |
| 1 | 0.0254 | 0 | 0 | |
| 2 | 0.0508 | 0.03169 | 0.001609852 | |
| 3 | 0.0762 | 0 | 0 | |
| 4 | 0.1016 | 0 | 0 | |
| 5 | 0.127 | 0 | 0 | |
| 6 | 0.1524 | 0 | 0 | |
| 7 | 0.1778 | 0 | 0 | |
| 8 | 0.2032 | 0.06 | 0.012192 | 11.7613684 |
| 9 | 0.2286 | 0 | 0 | |
| 10 | 0.254 | 0 | 0 | |
| 11 | 0.2794 | 0 | 0 | |
| 12 | 0.3048 | 0.041 | 0.0124968 | |
| 13 | 0.3302 | 0 | 0 | |
| 14 | 0.3556 | 0 | 0 | |
| 15 | 0.381 | 0 | 0 | |
| 16 | 0.4064 | 0.06 | 0.024384 | 8.0732884 |
| 17 | 0.4318 | 0 | 0 | |
| 18 | 0.4572 | 0.01647 | 0.007530084 | |
| 19 | 0.4826 | 0 | 0 | |
| 20 | 0.508 | 0.0271 | 0.0137668 | |
| 21 | 0.5334 | 0.1 | 0.05334 | |
| 22 | 0.5588 | 0 | 0 | |
| 23 | 0.5842 | 0 | 0 | |
| 24 | 0.6096 | 0.01 | 0.006096 | |
| | | | | |
| | | Gripper Weight | | |
| | | Object Weight | | |

# Future Work

- Rework of Gripper base
  - Holes for Pixy Mount
  - Holes for Gripper Mount
- Incorporate McMaster CAD Drawings for Screws etc.
- Cut acrylic
- Moment of Inertia Calculations
- 3D Print Parts
- Continue Coding

# Group P22: Pick ‘n’ Place Robot Competition

Brian Herrmann, Albert Olea, Augustine Marfo, Tamal Kumar, Saman Hooshyar
Sponsor: Professor Brown
Advisor Meeting: March 8, 2018

# Current Order Status

| Item | Quantity | Total Cost | Source | Purchased | Consider Price |
|---|---|---|---|---|---|
| M3x8 Screw | 100 | $2.72 | https://www.mcmaster.com/#92005a118/=1bk8cf9 | | |
| M3 Nut | 100 | $2.12 | https://www.mcmaster.com/#90592a085/=1bk8dfi | | |
| Washers | 100 | $1.62 | https://www.mcmaster.com/#93475a210/=1bk6wqz | | |
| Arduino | 1 | $6.95 | https://www.ebay.com/itm/ATmega328P-CH340G-UNO-R3-Board-USB-Cable-Compatible-with-Arduino-CP/202004659027?hash=item2f086a7b53:g:-sAAAOSw~RhZf-Oe | | |
| Servo Driver | 1 | $5.75 | https://www.ebay.com/itm/PCA9685-16-Channel-12-bit-PWM-Servo-motor-Driver-I2C-Module-For-Servo-Arduino-US/222487276915?epid=726544424&hash=item33cd466d73:g:gsIAAOSwls5Y6xXj | | |
| MG996R Servo | 8 | $34.50 | https://www.ebay.com/itm/Lot-MG996R-Metal-Torque-Gear-Digital-Servo-For-RC-Truck-Car-Boat-Helicopter-AL/112673220235?hash=item1a3bd8e68b:m:mSxPhCsx6ju0uJGUNibCXcQ | | |
| Servo Horns | 5 | $8.99 | https://www.amazon.com/Mallofusa-Servo-Aluminum-Silvery-Helicopter/dp/B00NOGMK3M/ref=pd_bxgy_21_img_2?_encoding=UTF8&pd_rd_i=B00NOGMK3M&pd_rd_r=ZPFP86MEB41K9MNGKDNB&pd_rd_w=FRfIw&pd_rc | | |
| 30mm Male-Female Jumper Wires | 40 | $4.99 | https://www.ebay.com/itm/40pcs-10cm-20cm-30cm-50cm-2-54MM-Male-to-Female-Wire-Jumper-Cables-For-Arduino/253297077577?hash=item3af9aead49:m:mmXeifnrKxClcQxCN3h24Fw | | |
| Pixy CMUcam5 | 1 | $74.95 | https://www.adafruit.com/product/1906 | | |
| Acrylic Sheets (12"x24") | 1 | $9.82 | https://www.acmeplastics.com/acrylic-sheets/acrylic-extruded-clear-sheet | | |
| Power Adapter for Servo | 1 | $15.98 | https://www.amazon.com/KNACRO-100V-240V-Switching-Interface-Surveillance/dp/B01JI373AY/ref=sr_1_10?s=electronics&ie=UTF8&qid=1518588750&sr=1-10&keywords=dc+5v+10a+power+supply | | |
| Gripper Mechanism | 1 | ? | | | |
| Standoff Assortment | 1 | $10.88 | https://www.amazon.com/Jekewin-120pcs-Universal-Standoffs-Assortment/dp/B01AQSZ262/ref=sr_1_9?s=industrial&ie=UTF8&qid=1518143864&sr=1-9&keywords=m3+metal+standoff | | |
| Wire Wrap | 1 | $4.99 | https://www.ebay.com/itm/1-2-Polyethylene-Spiral-Cable-Wire-Wrap-Tube-Black-10FT-Length/222791824319?hash=item33df6d73bf:g:imcAAOSwdTJaV2wW | | |
| | | | | | |
| | | | | | |
| | | $184.26 | | | |

# Work Performed: Revised Max Torque

- 24” Maximum Reach Assumption
- Gripper and Object Weight Assumptions
- Most likely point of failure is 1st motor

| Points | Multiplier | Length(in) | Length from motor1 (m) | Length from motor2 (m) | Length from motor3 (m) | Mass at Point (kg) | Torque Motor1 (kg*m) | Torque Motor2 (kg*m) | Torque Motor3 (kg*m) | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | | | 0 | 0 | | | |
| 1 | 1 | 1 | 0.0254 | | | 0 | 0 | | | |
| 2 | 1 | 2 | 0.0508 | | | 0.03169 | 0.001609852 | | | |
| 3 | 1 | 3 | 0.0762 | | | 0 | 0 | | | |
| 4 | 1 | 4 | 0.1016 | | | 0 | 0 | | | |
| 5 | 1 | 5 | 0.127 | | | 0 | 0 | | | |
| 6 | 1 | 6 | 0.1524 | | | 0 | 0 | | | |
| 7 | 1 | 7 | 0.1778 | | | 0 | 0 | | | |
| 8 | 1 | 8 | 0.2032 | | | 0.06 | 0.012192 | | | |
| 9 | 1 | 9 | 0.2286 | 0.0254 | | 0 | 0 | 0 | | |
| 10 | 1 | 10 | 0.254 | 0.0508 | | 0 | 0 | 0 | | |
| 11 | 1 | 11 | 0.2794 | 0.0762 | | 0 | 0 | 0 | | |
| 12 | 1 | 12 | 0.3048 | 0.1016 | | 0.041 | 0.0124968 | 0.0041656 | | |
| 13 | 1 | 13 | 0.3302 | 0.127 | | 0 | 0 | 0 | | |
| 14 | 1 | 14 | 0.3556 | 0.1524 | | 0 | 0 | 0 | | |
| 15 | 1 | 15 | 0.381 | 0.1778 | | 0 | 0 | 0 | | |
| 16 | 1 | 16 | 0.4064 | 0.2032 | | 0.06 | 0.024384 | 0.012192 | | |
| 17 | 1 | 17 | 0.4318 | 0.2286 | 0.0254 | 0 | 0 | 0 | 0 | |
| 18 | 1 | 18 | 0.4572 | 0.254 | 0.0508 | 0.01647 | 0.007530084 | 0.00418338 | 0.000836676 | |
| 19 | 1 | 19 | 0.4826 | 0.2794 | 0.0762 | 0 | 0 | 0 | 0 | |
| 20 | 1 | 20 | 0.508 | 0.3048 | 0.1016 | 0.0271 | 0.0137668 | 0.00826008 | 0.00275336 | |
| 21 | 1 | 21 | 0.5334 | 0.3302 | 0.127 | 0.1 | 0.05334 | 0.03302 | 0.0127 | |
| 22 | 1 | 22 | 0.5588 | 0.3556 | 0.1524 | 0 | 0 | 0 | 0 | |
| 23 | 1 | 23 | 0.5842 | 0.381 | 0.1778 | 0 | 0 | 0 | 0 | |
| 24 | 1 | 24 | 0.6096 | 0.4064 | 0.2032 | 0.001 | 0.0006096 | 0.0004064 | 0.0002032 | |
| | | | | | | | 12.5929136 | 6.222746 | 1.6493236 | Static Loading |
| | | | | | | | 18.8893704 | 9.334119 | 2.4739854 | Dynamic Loading |

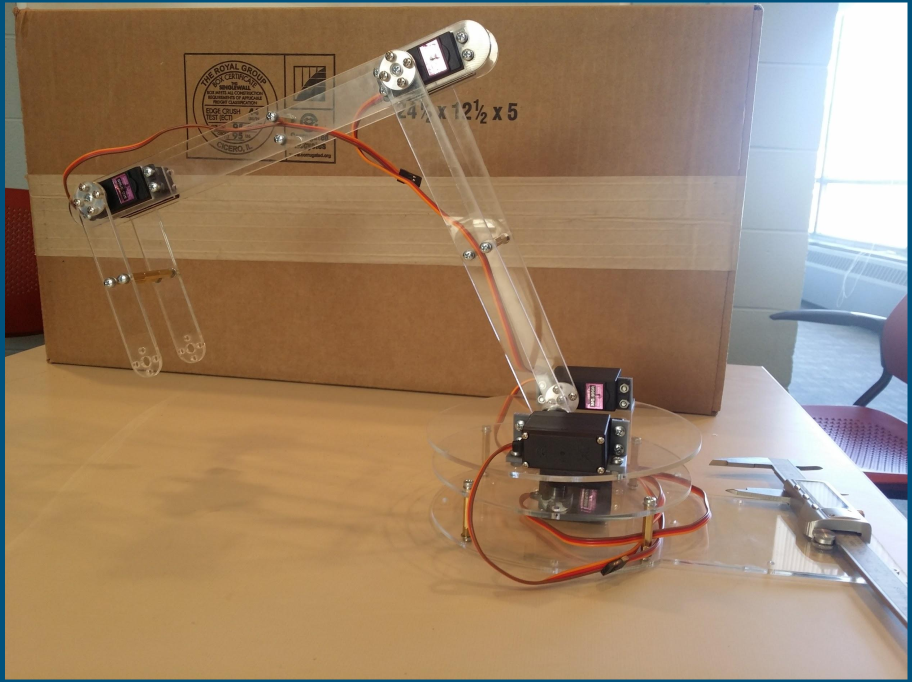
# Work Performed: Operating Conditions

- 16.97” Horizontal Max Reach assumption
- 24” at 45 Degrees
- Gripper and Object Weight Assumptions
- Should not fail

| Points | Multiplier | Length(in) | Length from motor1 (m) | Length from motor2 (m) | Length from motor3 (m) | Mass at Point (kg) | Torque Motor1 (kg*m) | Torque Motor2 (kg*m) | Torque Motor3 (kg*m) | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.707106 | 0 | 0 | | | 0 | 0 | | | |
| 1 | 0.707106 | 0.70710678 | 0.01796051224 | | | 0 | 0 | | | |
| 2 | 0.707106 | 1.41421356 | 0.03592102448 | | | 0.03169 | 0.001138337266 | | | |
| 3 | 0.707106 | 2.12132034 | 0.05388153673 | | | 0 | 0 | | | |
| 4 | 0.707106 | 2.82842712 | 0.07184204897 | | | 0 | 0 | | | |
| 5 | 0.707106 | 3.53553390 | 0.08980256121 | | | 0 | 0 | | | |
| 6 | 0.707106 | 4.24264068 | 0.1077630735 | | | 0 | 0 | | | |
| 7 | 0.707106 | 4.94974746 | 0.1257235857 | | | 0 | 0 | | | |
| 8 | 0.707106 | 5.65685424 | 0.1436840979 | | | 0.06 | 0.008621045876 | | | |
| 9 | 0.707106 | 6.36396103 | 0.1616446102 | 0.01796051224 | | 0 | 0 | 0 | | |
| 10 | 0.707106 | 7.07106781 | 0.1796051224 | 0.03592102448 | | 0 | 0 | 0 | | |
| 11 | 0.707106 | 7.77817459 | 0.1975656347 | 0.05388153673 | | 0 | 0 | 0 | | |
| 12 | 0.707106 | 8.48528137 | 0.2155261469 | 0.07184204897 | | 0.041 | 0.008836572023 | 0.002945524008 | | |
| 13 | 0.707106 | 9.19238815 | 0.2334866591 | 0.08980256121 | | 0 | 0 | 0 | | |
| 14 | 0.707106 | 9.89949493 | 0.2514471714 | 0.1077630735 | | 0 | 0 | 0 | | |
| 15 | 0.707106 | 10.6066017 | 0.2694076836 | 0.1257235857 | | 0 | 0 | 0 | | |
| 16 | 0.707106 | 11.3137085 | 0.2873681959 | 0.1436840979 | | 0.06 | 0.01724209175 | 0.008621045876 | | |
| 17 | 0.707106 | 12.0208152 | 0.3053287081 | 0.1616446102 | 0.01796051224 | 0 | 0 | 0 | 0 | |
| 18 | 0.707106 | 12.7279220 | 0.3232892204 | 0.1796051224 | 0.03592102448 | 0.01647 | 0.005324573459 | 0.002958096366 | 0.0005916192733 | |
| 19 | 0.707106 | 13.4350288 | 0.3412497326 | 0.1975656347 | 0.05388153673 | 0 | 0 | 0 | 0 | |
| 20 | 0.707106 | 14.1421356 | 0.3592102448 | 0.2155261469 | 0.07184204897 | 0.0271 | 0.009734597635 | 0.005840758581 | 0.001946919527 | |
| 21 | 0.707106 | 14.8492424 | 0.3771707571 | 0.2334866591 | 0.08980256121 | 0.1 | 0.03771707571 | 0.02334866591 | 0.008980256121 | |
| 22 | 0.707106 | 15.5563491 | 0.3951312693 | 0.2514471714 | 0.1077630735 | 0 | 0 | 0 | 0 | |
| 23 | 0.707106 | 16.2634559 | 0.4130917816 | 0.2694076836 | 0.1257235857 | 0 | 0 | 0 | 0 | |
| 24 | 0.707106 | 16.9705627 | 0.4310522938 | 0.2873681959 | 0.1436840979 | 0.025 | 0.01077630735 | 0.007184204897 | 0.003592102448 | |
| | | | | | | | 9.939060107 | 5.089829564 | 1.511089737 | Static Loading |
| | | | | | | | 14.90859016 | 7.634744346 | 2.266634605 | Dynamic Loading |

# Work Performed: Assembly

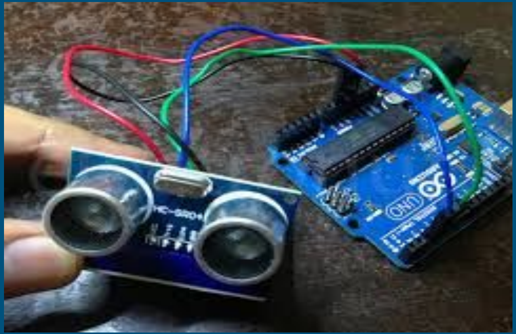
- Had to melt a hole into base for arm horn screw (CAD model updated)
- Mounting Brackets are 3D printed
- Rotation brackets redesigned to lower profile

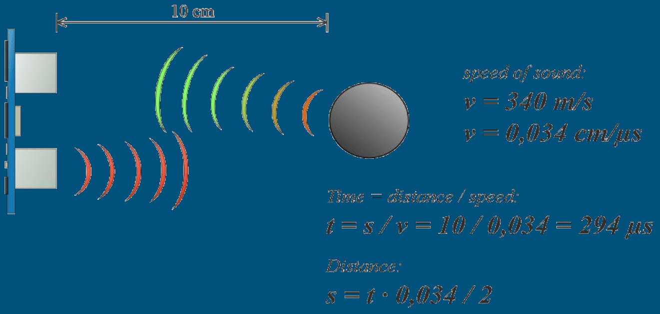


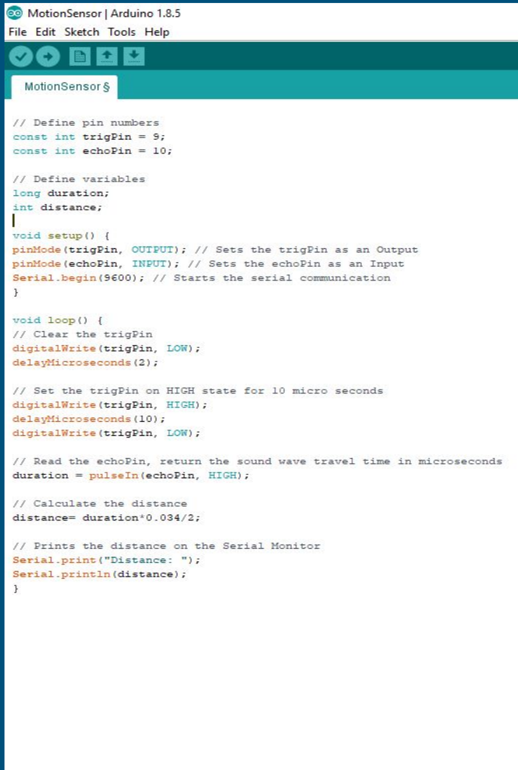
# Fallback Vision System
# (absolute worst case scenario)

HC-SR04 UltraSonicSensor

- Emmits 40,000 Hz UltraSound which bounces back from objects
- Echo pin receives the “bounceback” signal and outputs the waves travel time in microseconds
- Travel time is then used to calculate distance from echo pin to object.




```
// Define pin numbers
const int trigPin = 9;
const int echoPin = 10;

// Define variables
long duration;
int distance;

void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600); // Starts the serial communication
}

void loop() {
// Clear the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// Set the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Read the echoPin, return the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculate the distance
distance= duration*0.034/2;

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

# Multiple Servo Motor Control Code

- Need servo driver to run more than two servos at a time.
- Each individual servo is connected to a board pin and a command is issued to each servo pin to rotate the desired degrees.
- Use if...else control structure in arduino to read pixycam detection and move each servo (x) degrees based on recorded distances.

  [https://www.arduino.cc/reference/en/](https://www.arduino.cc/reference/en/)

multipeservo | Arduino 1.8.5

File Edit Sketch Tools Help

multipeservo

```
///Add the servo library and the servo variable

#include <Servo.h>

Servo servo1;
Servo servo2;


//Servo position in degrees
int servoPos = 0;

void setup() {
  //Define servo signal inputs (Digital PWM 3-5)
  servo1.attach(3);
  servo2.attach(5);
}

void loop() {
  //scan from 0 to 180 degrees
  for(servoPos = 0; servoPos < 180; servoPos++)
  {
    servo1.write(servoPos);
    servo2.write(servoPos);
    delay(10);
  }

 //now scan back from 180 to 0

  for(servoPos = 180; servoPos > 0; servoPos--)
  {
    servo1.write(servoPos);
    servo2.write(servoPos);
    delay(10);
  }

}
```

# Future Work

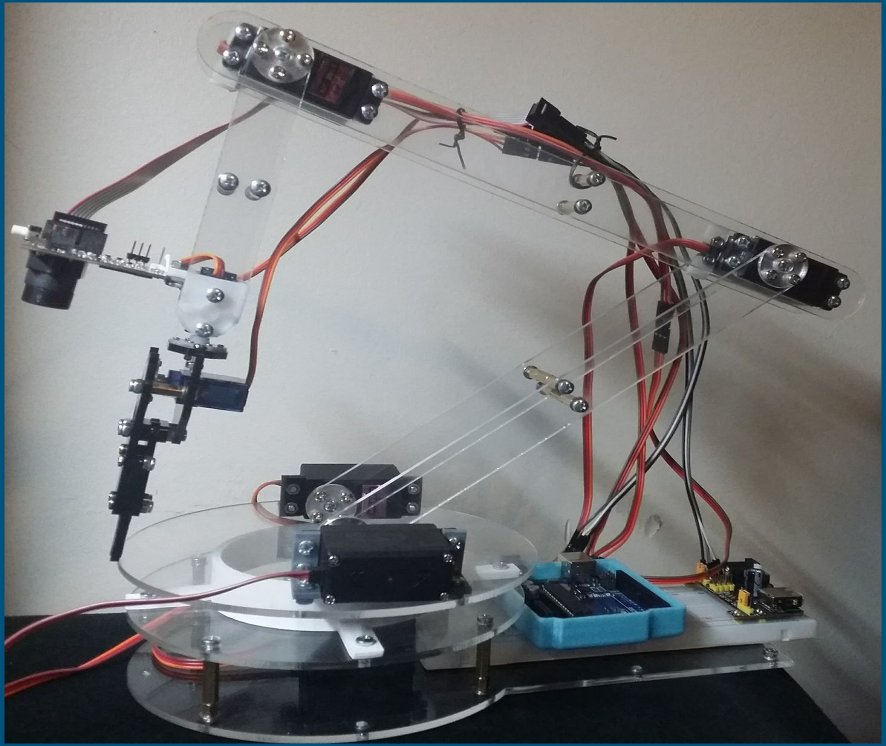
- Rework of Gripper base
  - Holes for Pixy Mount
  - Holes for Gripper Mount
- Incorporate McMaster CAD Drawings for Screws etc.
- 3D Print Rotating Parts
- Continue Coding
- Look into roller support for rotating base

# Group P22: Pick ‘n’ Place Robot Competition

Brian Herrmann, Albert Olea, Augustine Marfo, Tamal Kumar, Saman Hooshyar
Sponsor: Professor Brown
Advisor Meeting: March 22, 2018

# Work Performed: Assembly

- Gripper Mount Redesigned
- Mounting Brackets are redesigned and 3D printed
- Stabilization base helps with tilting (needs to be about 1.5mm taller)
- Motors attached according to rotation limitations

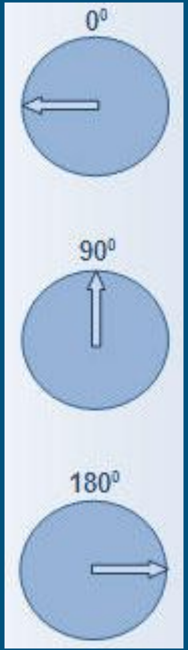


# Work Performed: Motor Alignment

- Each motor is advertised for 0 - 180°
- Actual limitation 0 ~ 170-179°
- Find 0° of motor, then attach it to the corresponding angle on the arm

ServoLimitationFinder §

```
#include <Servo.h>    // Create servo object to control a servo

Servo Servo_0;
Servo Servo_1;

void setup()
{
  //Start the serial.
  Serial.begin(9600);

  // Stops the motors from shaking when turning the arduino on
  delay (3000);

  //Attach the servos on pins to the servo object
  Servo_0.attach(4);
  Servo_1.attach(5);
}

void loop()
{
  //Set the initial positions of steering gears
  Servo_0.write(0);
  Servo_1.write(180);
  while(1);
}
```



# Forward Kinematics

Denavit–Hartenberg parameters (DH Parameters)

- Most common convention for defining angles and lengths of spatial chain in robotics
- Fundamental to Forward (Direct Kinematics)
- Uses transformation matrices to define location of end effector relative to a global frame

Cyclic Coordinate Descent

- Relatively new and simple method for inverse kinematics
- Typically requires less iterations than other methods (such as Jacobian) to reach a sufficient solution

# Forward Kinematics (Matlab Code)

Editor - C:\UIC\Senior Design I\KinematicsMatlabCode.m

KinematicsMatlabCode.m

```matlab
clc
theta_range = linspace(0,(170*pi)/180,1000);
l1 = 0.15;
l2 = 0.2;
l3 = 0.1;
theta = theta_range.*ones(3,1);
arm_length = [l1; l2; l3].*ones(1,length(theta_range));
alfa = zeros(3,length(theta_range));
distance = zeros(3,length(theta_range));
A=cell(1,3);
for i=1:3
    A{i} = [cos(theta(i,:)),-sin(theta(i,:)).*cos(alfa(i,:)),sin(theta(i,:)).*sin(alfa(i,:)),arm_length(i,:).*cos(theta(i,:));
        sin(theta(i,:)),cos(theta(i,:)).*cos(alfa(i,:)),-cos(theta(i,:)).*sin(alfa(i,:)),arm_length(i,:).*sin(theta(i,:));
        zeros(1,length(theta_range)),sin(alfa(i,:)),cos(alfa(i,:)),distance(i,:);
        zeros(1,length(theta_range)),zeros(1,length(theta_range)),zeros(1,length(theta_range)),ones(1,length(theta_range))];
end
T=cell(1,1000);
for j=1:1000
    T{j} = A{1}(:,4*(j-1)+1:4*j)*A{2}(:,4*(j-1)+1:4*j)*A{3}(:,4*(j-1)+1:4*j);
end
```

Workspace

| Name | Value |
|---|---|
| A | 1x3 cell |
| alfa | 3x1000 doubl |
| arm_length | 3x1000 doubl |
| distance | 3x1000 doubl |
| i | 3 |
| j | 1000 |
| l1 | 0.1500 |
| l2 | 0.2000 |
| l3 | 0.1000 |
| T | 1x1000 cell |
| theta | 3x1000 doubl |
| theta_range | 1x1000 doubl |

# Controlling Multiple Servos

- Adafruit 16-channel servo driver
- Add more in the void loop
- Final code would not be a loop

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

#define MIN_PULSE_WIDTH       650
#define MAX_PULSE_WIDTH       2350
#define DEFAULT_PULSE_WIDTH   1500
#define FREQUENCY             50

void setup() {
  pwm.begin();
  pwm.setPWMFreq(FREQUENCY);
}

void loop() {
  pwm.setPWM(0, 0, pulseWidth(0));
  delay(1000);
  pwm.setPWM(0, 0, pulseWidth(180));
  delay(1000);
}

int pulseWidth(int angle)
{
  int pulse_wide, analog_value;
  pulse_wide   = map(angle, 0, 180, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
  analog_value = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);
  return analog_value;
}
```

# Future Work

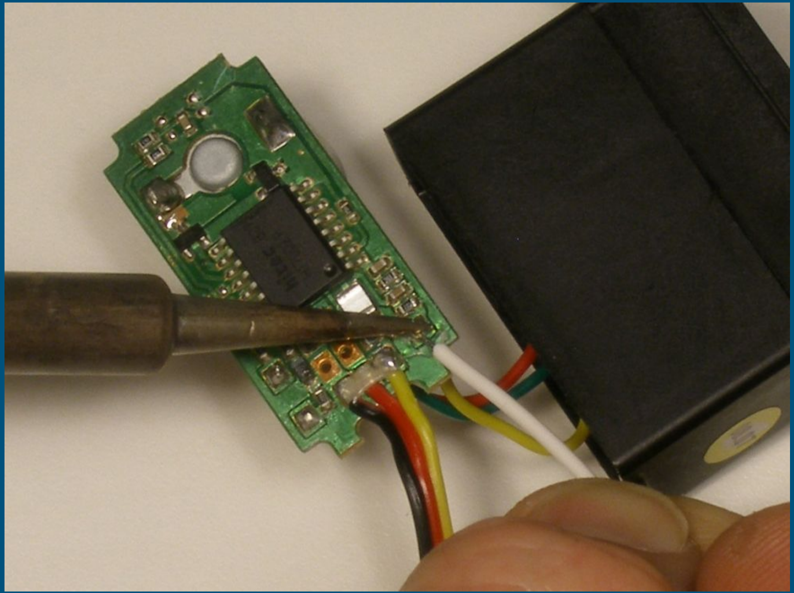
- Rework of Stabilization Base (1.5mm taller)
- Incorporate McMaster CAD Drawings for Screws etc.
- Coding

# Group P22: Pick ‘n’ Place Robot Competition

Brian Herrmann, Albert Olea, Augustine Marfo, Tamal Kumar, Saman Hooshyar
Sponsor: Professor Brown
Advisor Meeting: April 11, 2018

# Work Performed: Motor Feedback Modification

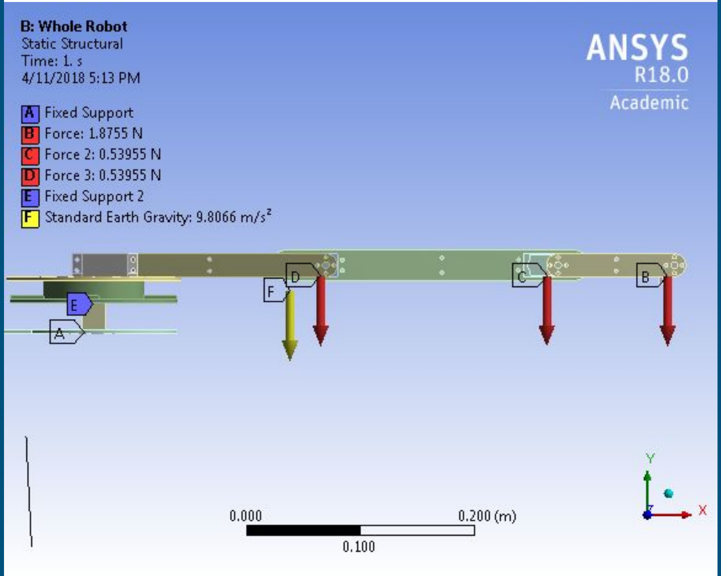
- Each motor is advertised for 0 - 180°
- Actual limitation 0 ~ 170-179°
- When loads are applied angles are not exactly when the code tells them to go
- Use internal potentiometer to see actual physical position
- Current test run says there is a 4-7° delta between software and hardware value



# ANSYS Analysis

Began ANSYS Analysis

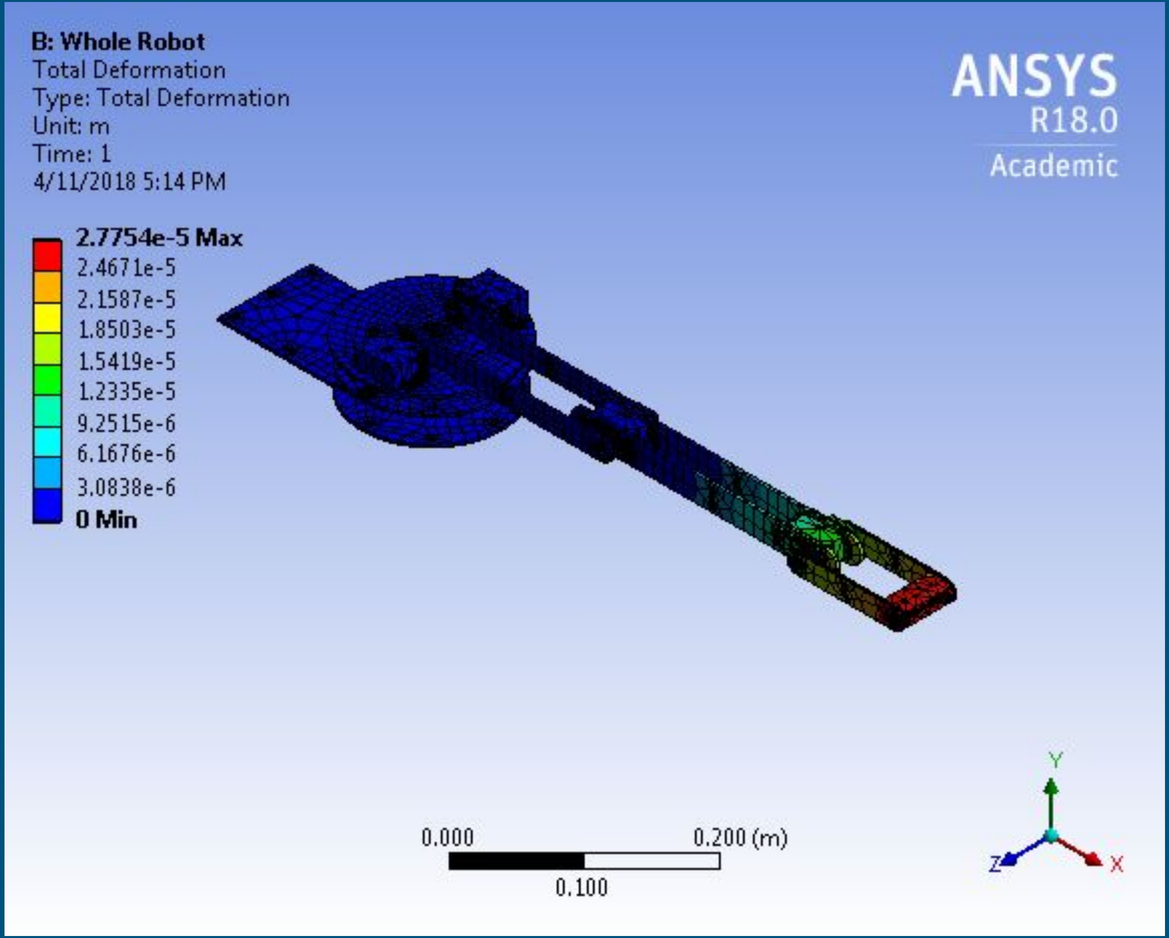
- Dynamic body analysis is very tricky and time consuming in ANSYS
- Decided to perform static analysis with arm at position that generates most torque at base (see image to right)
- The end result will solve for deformation, max stress, safety factor, as well as moment reactions at each joint



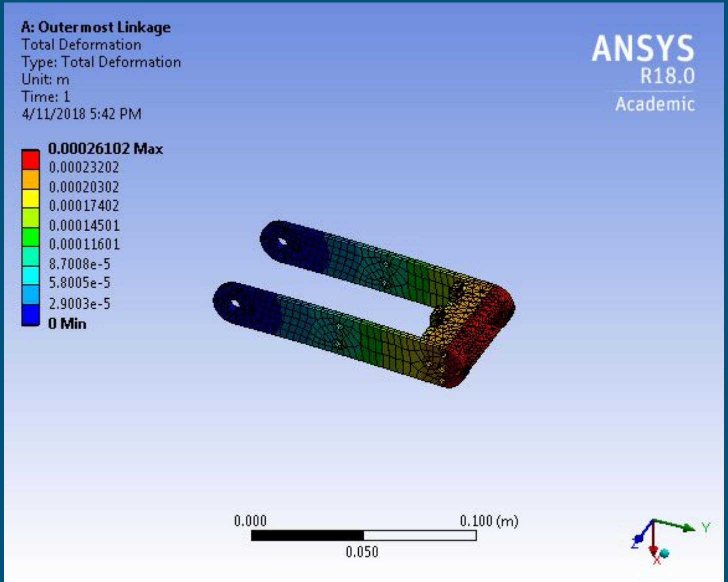
# ANSYS Analysis (Cont.)

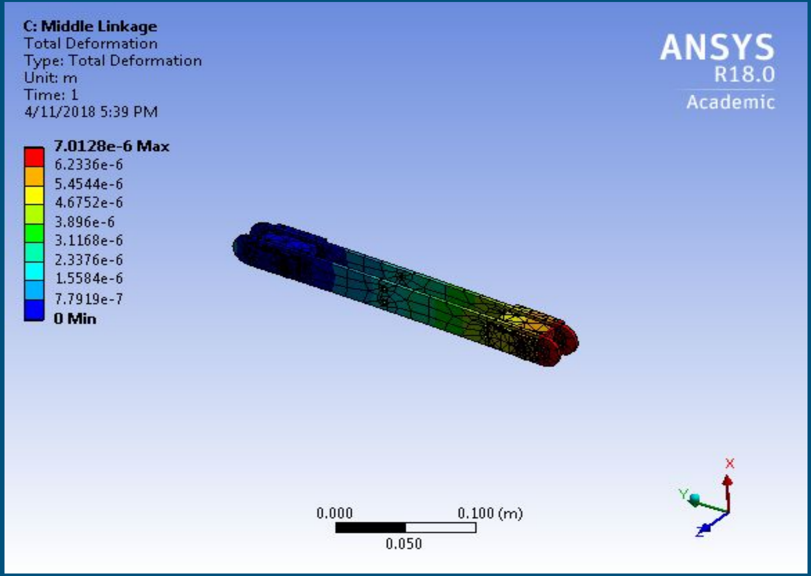
What is needed to complete analysis

- Update material properties
- Fix solidworks assembly
- Add screws to the assembly
- Ensure proper boundary conditions and fixed supports
- Proper mesh generation



# ANSYS Analysis (Cont.)



# Future Work

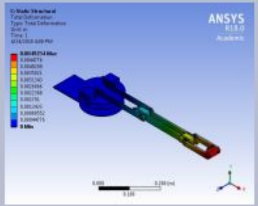
- Gather parts for expo demonstration
- Poster rework
- Incorporate McMaster CAD Drawings for Screws etc.
- Coding
- 3D print wiring cover

# Group P22: Pick ‘n’ Place Robot Competition

Brian Herrmann, Albert Olea, Augustine Marfo, Tamal Kumar, Saman Hooshyar
Sponsor: Professor Brown
Advisor Meeting: April 26, 2018

## Applications

Pick n place machines already exist in numerous fields such as assembly lines in factories or in the fabrication of computer hardware. Those devices differ in design however. This product would be more suited for sorting objects. A possible application could be in a sorting center for packages. The main need for the device is color distinction between objects. If that is present, the scale and range of the device can be altered to fit a number of different industries.




## Contact Information

Saman Hooshyar: shoosh2@uic.edu
Tamal Kumar: tkumar3@uic.edu
Augustine Marfo: amarfo2@uic.edu
Alberto Olea: aolea2@uic.edu
Brian Herrmann: bherrm3@uic.edu

MECHANICAL
AND INDUSTRIAL
ENGINEERING
COLLEGE OF
ENGINEERING
UIC

# Autonomous Sorting Robot

Saman Hooshyar
Tamal Kumar
Augustine Marfo
Alberto Olea
Brian Herrmann
Faculty Advisor: Michael Brown
Technical Advisor: Atif Mehmet Yardimci

## What Does it Do?

The robot arm is able to autonomously determine what object to move and where to place it. The robot will have a maximum horizontal reach of 24 inches and a vertical reach of 16 inches. The operating conditions include the following: Being able to sort several different objects of different colors, and being able to sort within the area of a 180° rotation up to 16 inches from the base of the automated arm.

The arm should be also able to perform this sorting in a time efficient manner comparable to a manually operated machine. The autonomous sorting consists of picking up the object from a random position near the base diameter, and then moving it onto a new designated storage location near the edge of the defined operating range. This process will be completed without direct user input.

## How it Works

This robotic arm's main components are linkages, motors, a camera, and an Arduino board. The Arduino board is what controls all of the motors which in turn move the linkages. The selected motors can rotate 180°.


Through lengthy Arduino code that applies equations of motion, the arm is able to move

The motors control the movement and the Arduino code control what the motors do. The robotic arm knows where to move through the use of the Pixy Camera.

## Pixy Camera

One way to detect objects is through PixyCam. It is an image sensor board that uses an external camera to get raw image data, thus acting as a visual sensor. Many image sensors output a lot of data that can easily override processors, however Pixy comes with its own processor that filters data so that only the most relevant data is sent to the microcontroller. It works primarily with Arduino and has all Arduino libraries. Pixy is small enough to be mounted onto a robot arm if needed. It uses color signatures to detect objects (up to 7 color signatures). Another feature is that it can be taught to recognize specific objects based on its color. The PixyCam has open source software, making it easier to work with.

**Pixy Camera**

# Future Work

- Competition Wednesday