



# Animatronic Kinect Bear

Computer Engineering Senior Project

Winter - Spring 2017

Under the Advisement of Dr. Hugh Smith

Christopher Barth

Emily Lopez

Luis Manjarrez

---

<b>Overview</b>	<b>2</b>
Goals	2
Background	2
<b>Specifications</b>	<b>4</b>
Project Outcome and Deliverables	4
Requirements	4
<b>Design Components</b>	<b>5</b>
Skeletal Structure	5
Servo Control	7
Kinect	8
Movement Mapping	9
Software Implementation	11
Hardware Implementation	13
<b>System Integration and Testing</b>	<b>13</b>
Design Verification	13
Analysis	16
Skeletal Structure	16
Kinect	18
Arduino	19
<b>Future Works</b>	<b>20</b>
<b>Appendices</b>	<b>22</b>
Bill of Materials	22
FMEA	23
Setup Manual	24
Personal Contributions	26
Chris Barth	26
Emily Lopez	26
Luis Manjarrez	26
Analysis of Senior Project Design	27



## Overview

The objective of the project was to build an interactive robotic bear through the application of Computer Engineering related software and hardware concepts.


Utilizing the body recognition features of XBox One Kinect, the bear will be able to mimic a user's motions in real time.

## Goals

1. Interface an Xbox One Kinect to capture a user's movement
2. Build a skeletal structure to support a 53-inch Costco Bear
3. Integrate 6 servos to allow for shoulder and elbow movement in at least 3 degrees of freedom
4. Create a system that can be easily maintained by the Computer Engineering Department for future use

## Background

The Kinect Bear is a reiteration of a previous senior project also advised by Dr. Smith. Although it has been done before, none of the code or movement mapping algorithms were published by the original group. Dr. Smith wanted to revisit the project, with the intention of creating a foundation for an ongoing project that future groups could build upon.



The original infrastructure utilized an Xbox One Kinect, Arduino Uno microcontroller, and five servos - 2 in each shoulder and one in the head. Dr. Smith's personal design had six servos - 2 in each shoulder and one in each elbow. Our final design drew from both structural designs, but adopted Dr. Smith's servo configuration.

The integration of the Kinect was the next major component of the project. Microsoft's Kinect was originally released in 2010, designed to enhance interactive gaming and the user's experience with the Xbox console. It provided a HD camera and voice recognition, supporting motion and voice commands. In 2011, Microsoft released a software development kit, allowing users to access the Kinect's features and create user interactive applications. Our team's development focused on accessing the camera's ability to track a user's movement.

The SDK included an overview guide, development packages, sample applications and corresponding code. The provided applications came in both C++ and C#, with the built projects in Visual Studio. These samples provided the most help during the initial development stages. Online resources on the Microsoft site and other developer forums were also crucial in learning which objects and components we needed to utilize in the Kinect Library. The Visual Studio support, alongside with its debug and diagnostic tools, helped during development. However learning how to work in Visual Studio and access the debug info, in conjunction with the Kinect interfacing also required research in online forums.

---

## Specifications

All requirements for the project were discussed and given by our advisor, Dr. Hugh Smith. The final product is intended for use by the Computer Engineering Department to display within the office and showcase at department events.

## Project Outcome and Deliverables

Upon completion, the bear will have a skeletal structure with two shoulder and one elbow joints to comprise each arm. The code and executable will be loaded onto a department computer, allowing for easy setup. A user manual will detail how to interface and connect all components including the bear, kinect, arduino, computer.

## Requirements

1. The system shall be designed to animate at 53-inch Costco Bear.
2. The bear shall be able to mimic natural arm movement.
3. The bear shall be able to follow along with a user with minimal delay.
4. The system must follow the movement of one user at a time.
5. The system shall run from a computer station.
6. The system shall use an Xbox Kinect to detect a user's movement.
7. The internal system of the bear must run on a power supply.

---

## Design Components

The design of the bear consists of 3 major components: the skeletal structure, the Kinect sensor, and the servos. Each part presented the challenge of integrating of mechanical, hardware, and software components.

### Skeletal Structure

The skeletal structure was modeled off of a skeletal structure Dr Smith created. The frame needed to provide the structural support for an 53" stuffed bear. The final structure was composed of pvc pipe, pvc connectors, and wooden dowels. The frame measured about 16" wide by 44" tall with 16" long legs. For the spine, two cross PVC connectors created a rectangular section with two 4" pvc pipes and two 34" pvc pipes. The distributed weight of the shoulders and arms relieved pressure on the torso providing better support than a single pipe.. The height of the bear was also taken into account when making the frame since we wanted the head of the bear to be supported.

In order to connect the arms to the frame, we created custom aluminum brackets that were screwed into the frame for each shoulder joint. Each arm of the skeletal structure is made up of three servos, two wooden dowels, and an aluminum flat bar. To create the shoulder movements for the structure we used two high torque servos, one was screwed into the aluminum metal bracket that was connected to the frame and the other servo was attached to the first servo to allow for upper

arm movement. A 10" wooden dowel was then screwed into the servo horn of the second servo that gave an upper arm shape. Using a custom drilled aluminum bar, we attached the third servo to the bottom of the 10" wooden dowel. The final part to the arm was attaching an 8" wooden dowel to the third servo to represent the forearm.

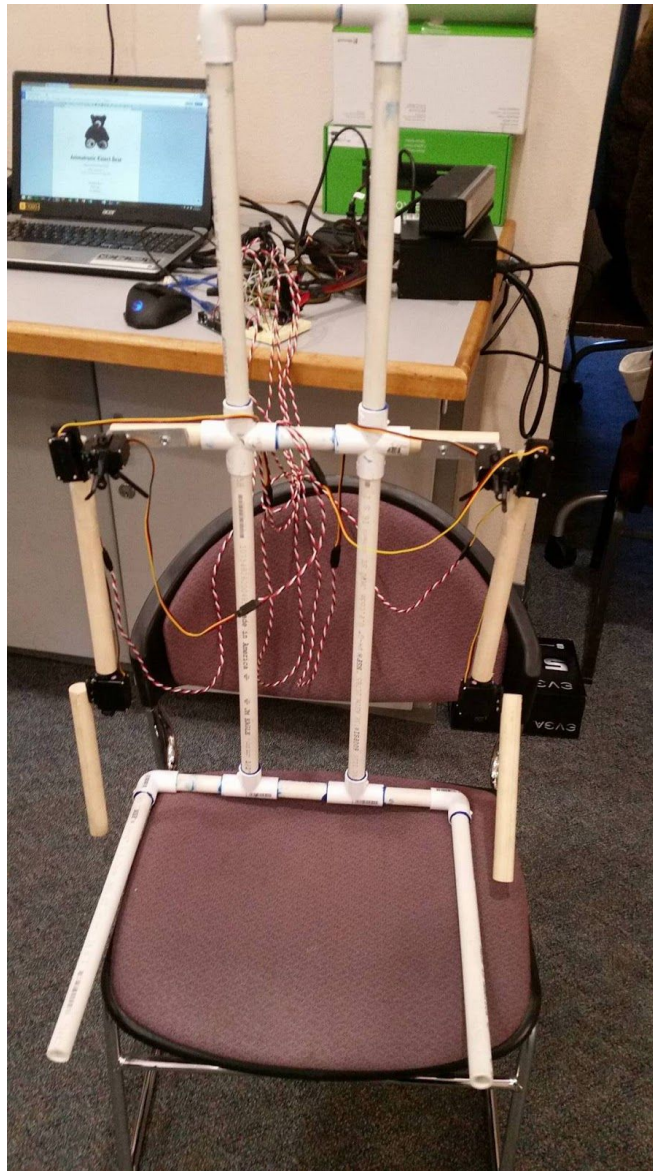


Figure 1: Final Skeletal Structure

## Servo Control

To replicate natural arm movement, we focused on the three degrees of freedom in the shoulder: the shoulder pitch, shoulder roll, and elbow pitch (Figure 8). To access these three degrees of freedom, we needed three servos per arm - six servos total.

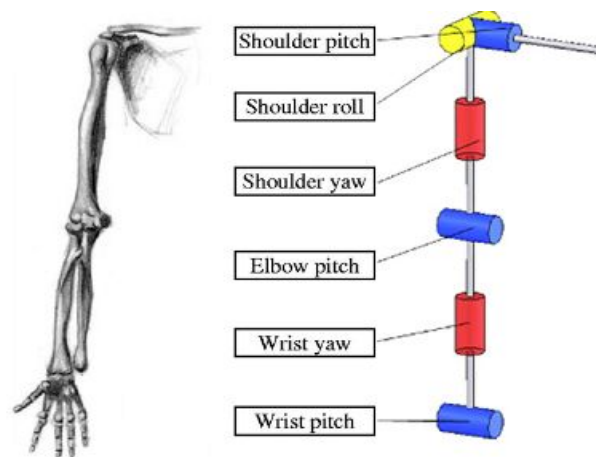


Figure 2: Degrees of Freedom in Arm

Source: <http://reset.etsii.upm.es/en/projects/robotic-arm/>

To control the servos, we used an Arduino Uno because of its 6 pwm pins. In order to communicate angles from the computer to the Arduino, we established a serial connection between the two devices. The serial connection passed the angles calculated on the computer and the corresponding servo to be moved to the arduino. Then using the Arduino servo library, we wrote the angle to each servo. The Arduino servo library was used to create servo objects set to a specific pwm pin. The servo library comes with a few ways to write an angle to a servo, such as sending the number of microseconds, or sending the degree at which the servo



should be at. Since we were able to calculate an angle, we set the angle of the servo by sending it the number of degrees.

## Kinect

In order to set the position of the servos, we need to access the data captured by the Kinect's body mapping capabilities. To do so, we utilized Microsoft's SDK for Windows. We began by adapting a program *Body Basics*: a simple application that visualizes a body frame by connecting each of the 27 recognized joints with a line (Figure 2). Through the implementation of *Body Basics*, we learned how to access the joint structures and familiarize ourselves with the Kinect Library.

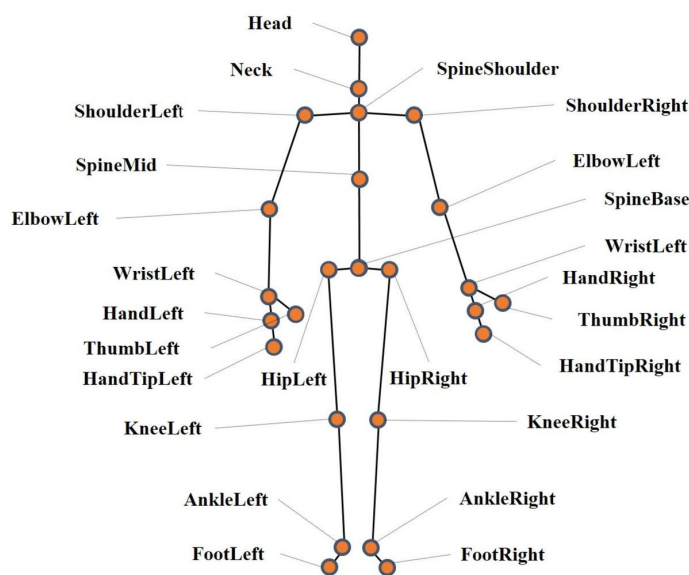


Figure 3: Kinect Recognized Joints

Source: <https://doi.org/10.7717/peerj.2364/fig-4>

Each of the 6 joint's' XYZ coordinates are captured and continuously updated by Kinect. These values are then sent to our movement mapping functions to calculate the necessary servo angles to achieve the same position. Throughout

development, we continued to utilize *Body Basics* to compare the Kinect readings with the bear's output.

## Movement Mapping

The Kinect collects an XYZ coordinate in space for each joint in relation to the center of its lens (Figure 3). In order to replicate natural movement, two servos are dedicated for a shoulder joint and one for the elbow joint.

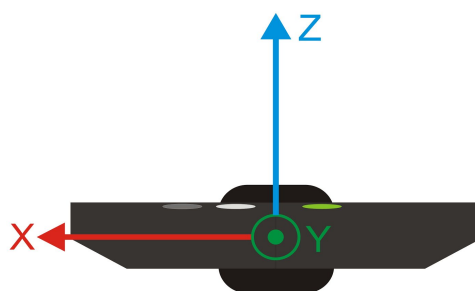


Figure 4 : XYZ Access Orientation in Relation to the Kinect

Source: <http://www.depthbiomechanics.co.uk/?p=2496>

In attempt to simplify the shoulders, individual servos were dedicated to control movement in the XY and YZ planes. To calculate the angle in each plane, the arctan is taken with respect to the  $\Delta X$  or  $\Delta Z$  over  $\Delta Y$  between the shoulder and elbow joints.

$$\theta = \arctan\left(\frac{X_{\text{shoulder}} - X_{\text{elbow}}}{Y_{\text{shoulder}} - Y_{\text{elbow}}}\right)$$

$$\theta = \arctan\left(\frac{Z_{\text{shoulder}} - Z_{\text{elbow}}}{Y_{\text{shoulder}} - Y_{\text{elbow}}}\right)$$

For the elbow, we are only using one designated servo and can not isolate the joint's orientation in 2 directions. The position needed to be found in 3D space. We utilized vector equations, taking the positions of the 3 joints into consideration. To calculate the angle we take the arccosine of the dot product between the 2D dimensional vectors formed between the wrist - elbow and shoulder - elbow .

$$\theta = 180 - \arccos\left(\frac{\bar{x} \cdot \bar{y}}{\|\bar{x}\| \cdot \|\bar{y}\|}\right) \cdot \frac{180}{\Pi}$$

where:

$$\bar{x} = (X_{elbow}, Y_{elbow}, Z_{elbow}) - (X_{shoulder}, Y_{shoulder}, Z_{shoulder}) = (X_1, Y_1, Z_1)$$

$$\bar{y} = (X_{elbow}, Y_{elbow}, Z_{elbow}) - (X_{wrist}, Y_{wrist}, Z_{wrist}) = (X_2, Y_2, Z_2)$$

## Software Implementation

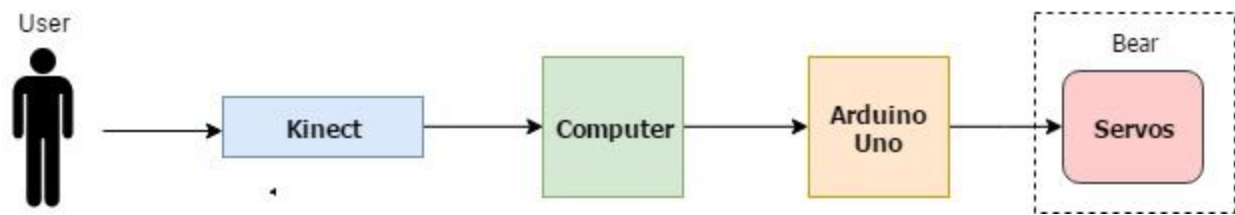


Figure 5: High Level Flow Diagram

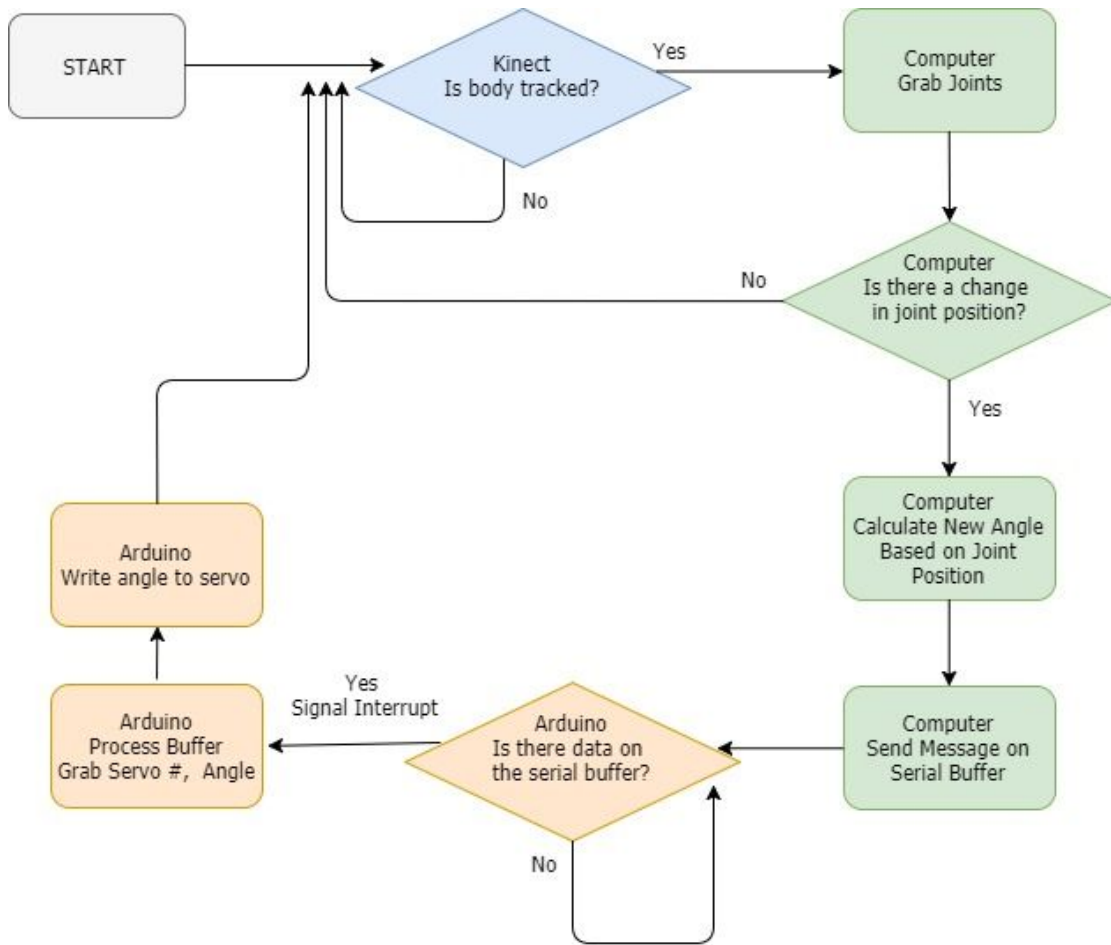


Figure 6: Software Flow Diagram

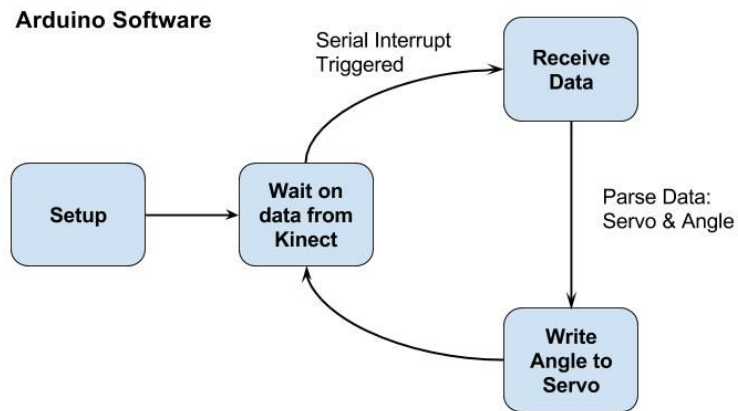


Figure 7: Arduino State Diagram

## Hardware Implementation

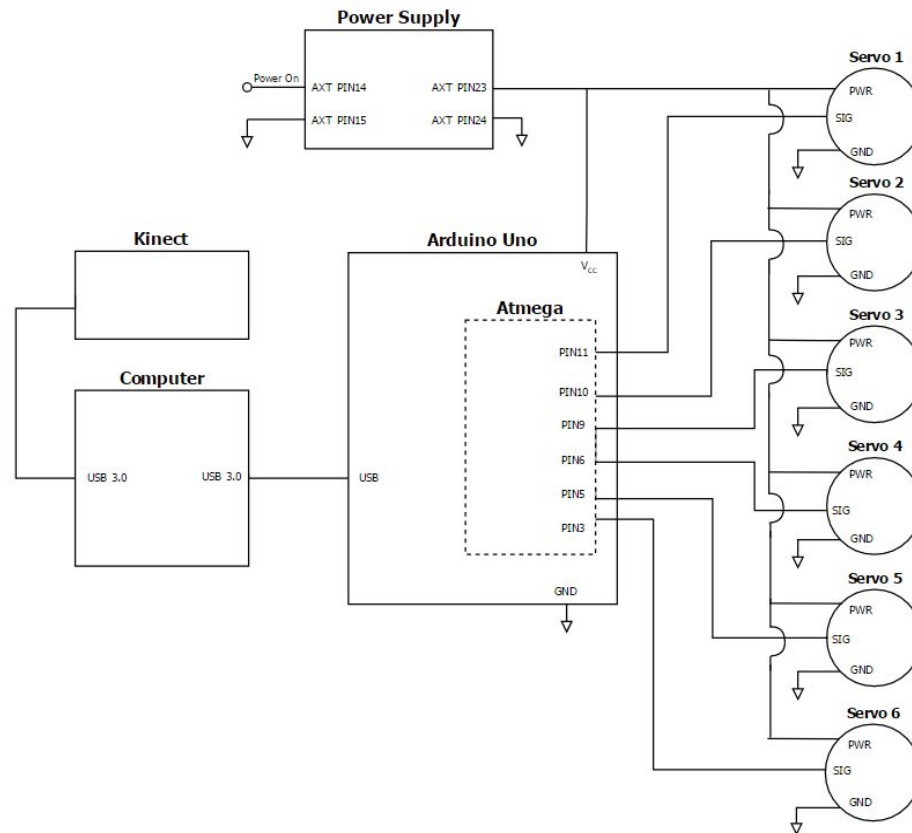


Figure 8: Block Diagram

## System Integration and Testing

### Design Verification

**Test Case:** Start Up

**Objective:** Verify upon startup bear is at resting position (arms straight down)

**Materials:** Kinect, Computer Station, Power Supply, Arduino, Bear

**Safety:** Verify all wires are secure and grounds are connected.

**Procedure:**

1. Power on supply
2. Verify Arduino power light is on
3. Launch Visual Studio Project Application
4. Verify new window opens for body tracking
5. Verify bear's arms are straight down

**Test Case:** Shoulder Movement in Z Plane

**Objective:** Verify the Bear can mimic shoulder movement in 2D space

**Materials:** Kinect, Computer Station, Power Supply, Arduino, Bear

**Safety:** Verify all wires are secure and grounds are connected.

**Procedure:**

1. Power on supply
2. Launch Application on Visual Studio.
3. Stand in front of Kinect with arm straight down at side
4. Move left arm straight in front of you
5. Verify bear's right arm moves straight out in front.
6. Verify bear's elbow does not bend
7. Verify bear's shoulder does not open outward
8. Move left arm straight above head
9. Verify bear's right arm moves straight up.
10. Verify bear's elbow does not bend
11. Verify bear's shoulder does not open outward
12. Move arm back down to side
13. Verify bear's arm returns to starting position
14. Repeat with right arm

**Additional Steps:** Run test moving arm up and down at different angles. Run test moving both arms at same time. Run test moving arm up and down at different speeds.

**Test Case:** Shoulder Movement in X Plane

**Objective:** Verify the Bear can mimic shoulder movement in 2D space

**Materials:** Kinect, Computer Station, Power Supply, Arduino, Bear

**Safety:** Verify all wires are secure and grounds are connected.

**Procedure:**

1. Power on supply
2. Launch Application on Visual Studio.
3. Stand in front of Kinect with arm straight down at side
4. Move left arm straight out to to your side
5. Verify bear's right arm moves outward to side.
6. Verify bear's elbow does not bend
7. Verify bear's shoulder does not move toward the front
8. Move left arm straight above head
9. Verify bear's right arm moves straight up.
10. Verify bear's elbow does not bend
11. Move arm back down to side
12. Verify bear's arm returns to starting position

13. Repeat with right arm

**Additional Steps:** Run test moving arm up and down at different angles. Run test moving both arms at same time. Run test moving arm up and down at different speeds.

**Test Case:** Shoulder Movement in 3D Space

**Objective:** Verify the Bear can mimic shoulder movement in 3D space

**Materials:** Kinect, Computer Station, Power Supply, Arduino, Bear

**Safety:** Verify all wires are secure and grounds are connected.

**Procedure:**

1. Power on supply
2. Launch Application on Visual Studio.
3. Stand in front of Kinect with arm straight down at side
4. Move left arm straight out to a lower 45 degree angle
5. Verify bear's right arm moves outward to a lower 45 degree angle.
6. Verify bear's elbow does not bend
7. Move left arm straight out to a high 45 degree angle
8. Verify bear's right arm moves outward to a high 45 degree angle.
9. Verify bear's elbow does not bend.
10. Move arm back down to side
11. Verify bear's arm returns to starting position
12. Repeat with right arm

**Additional Steps:** Run test moving arm straight out in different points in 3D space. Run test moving both arms at same time. Run test moving arm at different speeds.

**Test Case:** Elbow

**Objective:** Verify the Bear can mimic forearm movement in 3D space

**Materials:** Kinect, Computer Station, Power Supply, Arduino, Bear

**Safety:** Verify all wires are secure and grounds are connected.

**Procedure:**

1. Power on supply
2. Launch Application on Visual Studio.
3. Stand in front of Kinect with arm straight down at side
4. Bend elbow at 90 degree angle
5. Verify bear's right arm bends at elbow at 90 degree angle
6. Move arm up straight out, keeping elbow bent at 90 degree angle
7. Verify bear's arm moves straight out in front
8. Verify bear's elbow remains bent
9. Open shoulder outward to a 45 degree angle
10. Bend Elbow back to a 120 angle
11. Verify bear's elbow is bent to a 120 degrees
12. Move arm back down to side





13. Verify bear's arm returns to starting position


14. Repeat with right arm

**Additional Steps:** Run test bending elbow at various angles while moving arm in 3D space. Run test moving both arms at same time. Run test moving arm up and down at different speeds.

## Analysis

### Skeletal Structure


One of the main challenges for this project was designing a skeletal structure that could support the weight of the bear and the arms of the structure. The design we chose for the skeletal structure for the bear was a modified version of the skeleton Dr. Smith had created. The modifications we made to the design allowed for more stability in the body of the frame, support in the head of the bear, and less strain on the servo motors.



The first modification that was made to the design was adding a second pvc pipe to the body of the frame, which gave the body a rectangular shape. This allowed for us to split up the piping for the shoulders into three segments. Because of these modifications, the weight from the arms was better distributed across the body. The next modification was adding pvc pipe above the torso in a rectangular like fashion, which would be used to hold up the head of the bear.

The other major modification made to the design was using wooden dowels to construct the arms. We chose to use wooden dowels because the material was lightweight yet sturdy. For connecting the shoulder segments of the frame to the arms, we used wooden dowels in place of pvc. The main reason for this change was due to the weight of the arms and the amount of force from moving the arms that would be exerted on the shoulder segments. Lastly, for the arm segments we used wooden dowels over pvc and aluminum to make the arms lightweight and sturdy.

Although our design changes aimed to improve issues experienced with Dr. Smith's design, new complications arose for our iteration. One of the challenges was the need for custom made parts. There were two parts that had to be custom made for each arm. The first being an aluminum bracket to attach the shoulder servo to the frame and the second an aluminum bar that was used to attach the elbow servo to the upper arm. Our search for a metal bracket to attach the servo to the shoulder came up with no results. It was during the second quarter of working on this senior project we learned a red tag would give us access to Cal Poly's Machine shop,




where we could make our own custom parts. After one of our members received their red tag the assembling the bear move forward .

The next issue we came across with this project was finding a more stable way to attach the arm to the frame. Toward the final rounds of testing, the left arm would pop off of the aluminum bracket holding it. The arm was attached by two screws that then screwed into the bracket and the shoulder servo. The bracket did not provide any support to the servo. Over time, after testing the movement of the arms, the amount of force exerted on the servo and the two screws caused the servo to slip off. The short term solution we found was putting zip ties around the servo and the screws to force the servo to stay in place. In the future, designing a new bracket that supports all four mounting holes, while holding the bottom of the shoulder servo, could be a long term solution.

The last frame issue we encountered was a result of switching from pvc to dowels. Although the dowels provided stronger arms, the material was slightly heavier than the pvc. The structure was top heavy and rapid arm movement would cause the structure to rock. If dowels are used in the next iteration, adding weight to the legs to keep the structure grounded would help with the overall stability.

## Kinect

Developing for the kinect proved the greatest challenge in the project. Although the SDK provided a lot of sample code, the functions and objects we not documented well. We spent a large portion of the first quarter, going through the Kinect library



and identifying what we needed. The *Joint* object held the xyz coordinates of a given joint. Joints were identified through enums called *JointType*. The following joints are what we used in are mapping equation:

Shoulder Left	=	4
Elbow Left	=	5
Wrist Left	=	6
Shoulder Right	=	8
Elbow Right	=	9
WristRight	=	10

In order to grab to the data collected from the kinect, we read from the array of joints updated by the function *GetJoints*. Once we pinpointed where the joints were updated, we were able to develop our angle algorithms .

Developing from the structure of the *Body Basics* application also helped us throughout development. We were able to compare our movement to what the kinect register to what the servos output. Utilizing Visual Studio Debug Output also allowed us to display the coordinates. Using all 4 points of reference, determining where values were incorrect proved fairly simple.

### Arduino

An Arduino ATmega was purchased for the project, but the team ran into some technical difficulties when it came to interfacing it with our computers. Therefore, an Arduino Uno was used instead. The Uno provided enough processing power and enough PWM pins to handle our requirements. However for future iterations, if additional servos are added, additional pins available on the ATmega will be required.


## Future Works

Along with the code and the frame that was built for the project, the system requires a computer with USB 3.0 capabilities. The computer itself doesn't need to be super powerful in performance. It would help facilitate the process if a computer was purchased and had the Arduino IDE and Microsoft Studio installed on it. That would allow for the project to be worked on one central computer that could be passed on between team members, if more than one person will be continuing the project.

Due to the fact that we did not place the frame inside the bear itself, it is important to test if the servos can handle the additional stress added by the weight of the bear. The servos used for the shoulders might require a bit more of stability in order to maximize performance.

To increase the range of motion of the frame, an additional servo should be added to focus on the shoulder yaw, as indicated in Figure 1. This would focus on the rotational motion in the bicep area that would allow the frame to more accurately mimic a human's motion. We believe that little to no change would have to be done to the shoulder and elbow calculations. In fact, the same way the shoulders were calculated should be similar to the way this new degree of freedom is added, using the elbow and wrist joints instead.

When the bear is finalized, the goal was to have the Kinect lock on to the first skeleton, as it can process up to 6 skeletons. This would ensure an optimal



experience for a single user without confusing the bear into reacting to multiple users at one time.

Something that might be worth looking into is a software program called Vitruvius. Released about a year ago, by Microsoft and Channel, Vitruvius is an advanced Kinect framework. The program detects the movement of 1-6 people and calculates the angles between 3 joints using a single line of code, and provides a lot of additional functionalities not related for our project. There are multiple tiers of paid versions, though it appears that the free version would be sufficient for this use. The application would perform a lot of the same core functionality that we implemented. If a future group was able to utilize this software, they could then explore and implement more advanced features of the Kinect.

Finally, an original goal was to look into voice features using the Kinect. The Kinect can handle voice options but the challenging part would be translating what effect that would cause on the movement. Certain voice commands could trigger a certain set pattern or could do simple movements.

## Appendices

### Bill of Materials

Item #	Description	Unit Price	Quantity	Total Price
1	Arduino Uno	\$24.95	1	\$24.95
2	HS-5585MH Servo	\$55.99	4	\$223.96
3	90 Degree Universal Servo Mount (2 pack)	\$1.95	5	\$9.75
4	#4 - 3/8" Zinc - Plated Steel Flat Head Phillips Self-Tapping Screws	\$0.05	10	\$0.50
5	#4 - 1/2" Zinc - Flat Head Phillip Wood Screw (100pk)	\$3.28	1	\$3.28
6	HS-5496MH Servo	\$27.99	2	\$55.98
7	13/16" Wood Dowel	\$3.99	2	\$7.98
8	3/4" x 10' PVC Pipe	\$2.97	1	\$2.97
9	3/4" PVC 90-Degree Elbow	\$0.37	4	\$1.48
10	3/4" PVC Tee	\$0.53	2	\$1.06
11	3/4" PVC Cross	\$2.08	2	\$4.16
12	3/4" x 36" Aluminum Flat Bar with 1/8" Thick	\$4.28	1	\$4.28
13	8oz. PVC All-Purpose Cement	\$5.80	1	\$5.80
14	#6 - 1/2" box of screws	\$6.20	1	\$6.20
			Total	\$352.35

FAILURE MODE AND EFFECTS ANALYSIS

Item: Animotronic Bear Responsibility: All  
 Model: Current Prepared by: All

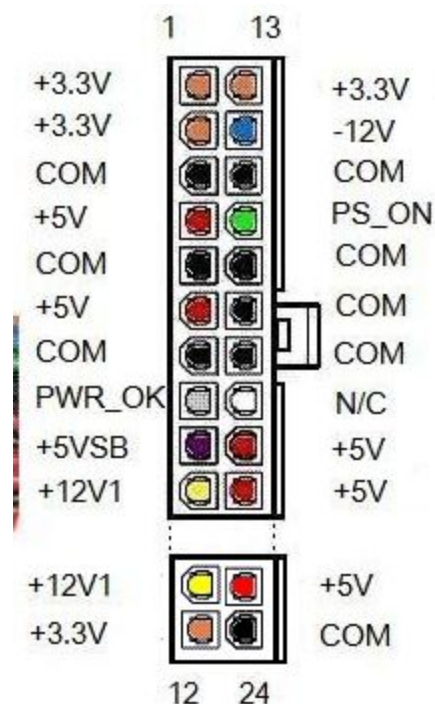
FMEA number: 1  
 Page: 1 of 1  
 FMEA Date (Orig): 6/1/2017 Rev: 1

Process Function	Potential Failure Mode	Potential Effect(s) of Failure	S e v e r i t y	C i a s e s	P o t e n t i a l C a u s e ( s )/ M e c h a n i s m ( s ) o f F a i l u r e	O c c u r r e n c e	C u r r e n t P r o c e s s C o n t r o l s	D e t e c t e d	R i s k P o t e n t i a l	R e c o m m e n d e d A c t i o n ( s )	R e s p o n s i b i l i t y a n d T a r g e t C o m p l e t i o n D a t e	Action Results			
												A c t i o n s T a k e n	S e c u r e	O c c u r r e n c e	R e p a r t u r e
Frame Follows User's Movement	Servo blows out	Movement can't be mirrored	9	2	Overuse of too much power	2	Use servos intended for use	9	162	Replace Servos	Next team, ASAP	0	0	0	0
	Arm falls off frame	Movement can't be mirrored	9	3	Improper machine set up	3	Ensure structure stability	9	243	Reattach Arm	Next team, ASAP	0	0	0	0
Kinect Detects User Skeleton	Camera Isn't Responsive	User can't be tracked	7	2	Broken Kinect	2	None	9	126	Replace Kinect	Next team, ASAP	0	0	0	0
Arduino Processes Data	Board can't process data	Data can't be sent to servos	8	2	Too much data being sent	2	Control data flow	7	112	Check data rate needed	Next team, ASAP	0	0	0	0
Power Supplied from Supply	Power can't be supplied	Not enough power is supplied	9	2	Wire gets disconnected	2	Ensure all connections are secure	8	144	Reconnect any loose connections	Next team, ASAP				0



## Setup Manual

1. Download Kinect SDK from Microsoft Website
2. Download Visual Studio Community
3. Sign-in to Visual Studio using Cal Poly Email
4. 2 - Options
  - a. Option 1: Download Arduino IDE
  - b. Option 2: Download Visual Micro (add-on to Visual Studio)
5. Download BodyBasics-D2D from OneDrive: Animatronic Kinect Bear Folder
6. Open up visual Studio
7. Import BodyBasics-D2D project
8. Open Arduino Development environment (Option 1 or 2)
9. Open ServoControl.c
10. Setup Hardware
  - a. Connect servos to Arduino Uno using the Hardware Implementation Diagram
  - b. Connect power supply to the breadboard using the 20+4 ATX Connector: PIN23 to power. PIN24 to GND.
  - c. Create a short (or a switch) between the PS\_ON and GND on PIN14 and PIN15, respectively.



- d. Plug in Kinect
  - e. Connect Kinect to Computer station using a USB 3.0 port
11. Upload Code to Arduino Uno
- a. Turn on power supply
  - b. Connect Arduino Uno to Computer station
  - c. Set Board Setting to "Arduino Uno"
  - d. Set Port to that of the usb connection
  - e. Upload program to the Arduino Uno
12. Setup BodyBasics-D2D Serial Connection
- a. Find line 111 in BodyBasics.cpp  
Line 111 : `serialPort = new Serial("COM6");`
  - b. Change COM6 to the COM Port the Arduino Uno is connected to.
13. Before Running BodyBasics.cpp
- a. Stand about 2' away from the front or sides so that no one gets injured or the skeleton does not break.
  - b. Check skeletal structure to make sure it is well supported
  - c. Check condition of skeletal structure.
    - i. If not, fix skeletal structure then proceed
14. Skeletal structure is in good condition then click "Local Window Debugger" in Visual Studio.
15. To Stop, Click the stop button on the Visual Studio.
16. Unplug Arduino
17. Turn off Power Supply

---

## Personal Contributions

### Chris Barth

There are three parts to our project: designing and creating the skeletal structure, the software on the Arduino Uno, and the software for the Kinect. In each of these parts I had a significant role. In order to build the skeletal structure we needed to create custom parts. I took a 3-hour test to get my red tag so I could use Cal Poly's Machine Shop to make the custom parts we needed. With the help of Luis, we designed the skeletal structure and using those plans, I built the skeletal structure. I researched and bought most of the parts to build the frame. On the software side, I wrote the code for the Arduino Uno to move the servos and setup the serial connection on the Arduino Uno's end. On the Kinect's software side, I helped to setup the serial communication between the board and Kinect, implement the theory for the elbow and shoulders, debug 2D space of the shoulder movements, and get 3D space working for the arms. I was highly involved in debugging the software and fixing problems with the skeletal structure when they arose.

### Emily Lopez

My bulk of contribution to the project involved the Kinect. In the beginning, I researched Microsoft's SDK and available documentation. I documented all the required software and hardware we needed to interface with the Kinect. Once I was able to successfully run a sample program, I then helped the rest of the group get set up. My next steps involved understanding the code. I went through the provided Kinect library and documented which functions and objects we would need to use to grab joints. I isolated where the data points were updated and print them out to the Visual Studio's output console. I also did the calculations to determine the angles for both the shoulders, and the elbow alongside Luis. Once the equations were implemented in the code, I worked with Chris on debugging the movement, comparing what the Kinect read and what the servos were being sent.

### Luis Manjarrez

My contributions mainly included helping with the skeletal design and the mapping of the elbow movements. With the help of Chris, we measured the desired lengths for each component and decided on what materials to use. I originally tested the controls of the servos using the Arduino IDE to learn more about how to operate each servo. After that, I focused on how to accurately map a user's movement in the elbows using 3D vector math. All three of us were involved in the troubleshooting when it came to the movement mapping of the shoulders.

## Analysis of Senior Project Design

Please provide the following information regarding your Senior Project and submit to your advisor along with your final report. Attach additional sheets for your responses to the questions below.

**Project Title:** Animatronic Kinect Bear      **Quarter / Year Submitted:** Spring 2017

1. **Student:** (Print Name) Christopher Barth      (Sign) \_\_\_\_\_
2. **Student:** (Print Name) Emily Lopez      (Sign) \_\_\_\_\_
3. **Student:** (Print Name) Luis Manjarrez      (Sign) \_\_\_\_\_

**Advisor:** (Print Name) Hugh Smith      (Initial) \_\_\_\_\_ Date: \_\_\_\_\_

### Functional Requirements:

Describe the overall capabilities of functions of your project or design. Describe what your project does. (Do not describe how you designed it.)

- Interface an Xbox One Kinect to capture a user's movement
- Build a skeletal structure to support a 53-inch Costco Bear
- Integrate 6 servos to allow for shoulder and elbow movement in at least 3 degrees of freedom
- Create a system that can be easily maintained by the Computer Engineering Department for use

### Primary Constraints:

Describe significant challenges or difficulties associated with your project or implementation.


For example, what were limiting factors or other issues that impacted your approach? What made your project difficult? What parameters or specifications limited your options or directed your approach?

### Economic:

- o Original estimated cost of component parts (as of the start of your project)
- o Actual final cost of component parts (at the end of your project)
- o Attach a final bill of materials for all components
- o Additional equipment costs (any equipment needed for development?)
- o Original estimated development time (as of the start of your project)
- o Actual development time (at the end of your project)

Original Estimated cost: \$200

Final Cost: \$352.35



Final Bill of Materials: Refer to the appendix

Original Estimated Development Time: 3 Quarters

Actual Development Time: 2 Quarters

**Manufactured for a commercial basis?**

- o Estimated number of devices to be sold per year
- o Estimated manufacturing cost for each device
- o Estimated purchase price for each device
- o Estimated profit per year
- o Estimated cost for user to operate device, per unit time (specify time interval)

Our project is not to be manufactured for a commercial basis.

**Environmental:**

Describe any environmental impact associated with manufacturing or use.

The system is environmentally safe as it uses a regulated amount of power for the system.

**Manufacturability:**


Describe any issues or challenges associated with manufacturing.

The component we manufactured was the skeletal frame. The challenge here was building a structure that could support the combined weight of the servos and the weight of the bear, while providing optimal usage.

**Sustainability:**

- o Describe any issues or challenges associated with maintaining the completed device or system.
- o Describe how the project impacts the sustainable use of resources.
- o Describe any upgrades that would improve the design of the project.
- o Describe any issues or challenges associated with upgrading the design.

One issue with our skeletal design is the attachment of the arm to the skeletal frame. Our current method of attachment is screwing two screws into the servo mounting holes and an aluminum bracket. There are two problems with this method. The bracket does not provide any support to the servo causing all the pressure to be on the point where the servo is screwed to to the bracket. The second problem is due to all the force coming from moving the arms the servo



pops off of the screws causing the whole arm to fall off with it. Our short-term solution for this problem was using zip ties to force the servo to stay attached to the screws. A long term solution might be designing and making a new bracket that supports all four mounting holes and holds the bottom of the shoulder servo. The challenge the long term solution is having to create a custom bracket.

### **Ethical:**

Describe ethical implications relating to the design, manufacture, use or misuse of the project.

Our project has no ethical implications.

### **Health and Safety:**

Describe any health and safety concerns associated with design, manufacture or use.

- Do not touch any of the wires when the power supply is on.
- Do not stand within 2 feet of the front or sides of the frame because the arms can extend and may hit the user.

### **Social and Political:**

Describe any social and political concerns associated with design, manufacture or use.

There are no social or political concerns

### **Development:**

Describe any new tools or techniques used for either development or analysis that you learned independently during the course of your project.

Software Development:

- Visual Studio
- Visual Micro (add-on for Arduinos)
- Arduino IDE

Mechanical Development:

- Cal Poly Machine Shop
- Red Tag