Basketball Return Optimizer

Brian Acker, CSE, Derek Foster, EE, Devon O'Rourke, CSE, and Adam Paranay, EE.

Abstract—**In an era where automation is becoming a necessity in almost all facets of life, it is no surprise that the Basketball Return Optimizer's (BRO) main feature is automation via target tracking. Basketball return systems have been developed to help return the basketball to the player after they have made a shot. These return systems are meant to maximize the player's time shooting while limiting the time they have to retrieve the shot basketball. Unfortunately, the player still must manually adjust these systems to control where on the court the ball will be returned. This is an inefficiency that BRO addresses. As it stands, no recreational or professional system can track a player and return the ball to said player no matter where they stand on the court. BRO is a traditional return funnel system that is modified to maximize the players shooting time by using automatic tracking. By taking the funnel, mechanizing it and integrating a webcam that tracks the player, our team has created a system that allows the player to freely move around the court and have the ball returned to them regardless of position.**

I. Introduction

TRADITIONAL basketball return systems, whether recreational, commercial, or professional do not maximize the player's time shooting because of the limitation of where the ball is returned on the court. The main problem that our team is addressing is that time spent practicing in basketball is often wasted by retrieving the basketball after shooting attempts. This problem does not need to be addressed however, we as a team feel that with our solution, the sport itself could reach new heights in terms of refining player's skill levels. Many existing products do this and we hope to build off of these designs to create an even more efficient system.

Figure A: iC3 Basketball Return System by Airborne Athletics

According to a study done by Airborne Athletics, their return system, the iC3 manages to triple the amount of shots possible within an hour [1]. Our system works on a recreational and professional level. Current return systems that professionals use only allow for pre-programmed return positions. With our system, professionals would have the freedom to shoot wherever they want while increasing their shots per hour. Our system is designed to better utilize a player's practice time so that they can take more shots per hour, allowing them to develop their skills faster and more efficiently.

TABLE I
GENERAL REQUIREMENTS

Specification	Value
Tracking Distance	5-25 feet from rim
Tracking Accuracy	100%
Operation Time	>1 hour
System Integrity	Withstands direct hit from basketball
Weight	<15lbs
Setup/Teardown	<5 minutes

As mentioned above, Airborne Athletics have created a system that effectively triples the shots a player can take per hour. Yet their system does not support free-form movement around the court because the iC3 only returns the basketball in one direction and requires manual adjustment to change that direction. The iC3 system has a retail price of $349.99. Another company, Goalrilla, has created a basketball return system using one large net that acts as a ramp. This ramp allows for the ball to be returned to anywhere on the free throw line. This system, like the iC3, does not accommodate free movement about the court because the ball is returned to a predefined location. This system retails at $79.50 [2]. The last system we have used for reference is the Dr. Dish Rebel. This system is a top of the line product meant for professional use. This system returns the ball using pre-programmed spots that the player or coach decides upon before starting up the machine. The Rebel not only returns the basketball but it does so in a chest-pass form, which represents how the shooter would realistically receive their passes. This system retails for $3,999.99 [3].

By studying the current market, our team was able to identify a shortcoming common to existing products. No existing system can return the basketball to the player regardless of their position on the court in real-time. Our system implements the solution to this problem by tracking the player on the court via a camera and processing the image to find the direction in which the ball should be returned. Our team believes that this feature

D. Foster from Stoneham, MA (e-mail: derekf@umass.edu).
A. Paranay from Dracut, MA (e-mail: aparanay@umass.edu).
B. Acker from Norton, MA (e-mail: backer@umass.edu).
D. O'Rourke from Bourne, MA (e-mail: dporourk@umass.edu).

is desirable and marketable because it allows basketball players increased flexibility in where they shoot from on the court. Unlike existing systems, BRO ensures that the ball is always returned to the player without requiring the player to manually adjust the system.

II. Design

A. Overview

Our overall design revolves around the SKLZ funnel return system [4]. This funnel attaches to the rim using four hooks and straps. The straps connect to a disc around which the funnel rotates. The funnel is manually set to return the ball to a certain location on the court. By taking this $29.99 return system and modifying it, we can make it possible to track a player in real-time while returning the ball to the player at any position in front of the basketball hoop.

Figure B: SKLZ Funnel Return System

By replacing the funnel's disc with a 3D printed gear, we can attach a motor and pinion gear that will rotate the funnel around the 3D printed gear. The motor will be controlled by power signals regulated by the BeagleBone Black. The BeagleBone Black will decide what signals to send to the motor with the help of the camera [5]. If the player is in the middle of the camera frame, then the funnel does not need to move so no power is supplied to the motor.

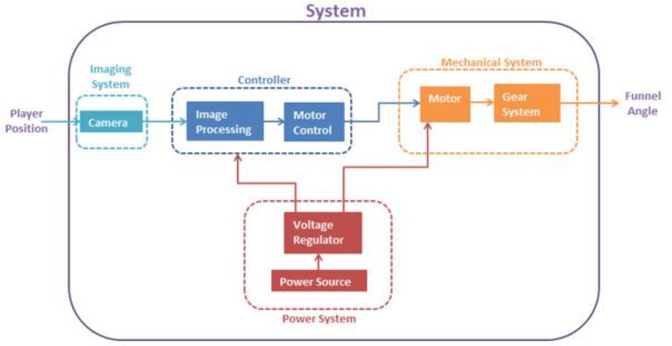

Figure C: Block Diagram

If the player is to the left or right of the camera's center, then either positive or negative power will be sent to the DC motor resulting in either a clockwise or counterclockwise rotation of the funnel.

The BRO will consist of four sub-systems: imaging, controller, power, and mechanical. We will further break down each of the sub-systems. The imaging sub-system will consist of the webcam, which will take the pictures at least five times per second. The controller sub-system will process the images and send a signal that tells the motor what voltage to pull. The power system will regulate the different powers and voltages needed to run the BeagleBone and the motor. The mechanical system will turn the funnel using the motor and 3D printed gears.

B. Block 1: Imaging System

The imaging system is the simplest block of our overall system and only consists of a camera for capturing images of the shooting area of the basketball court. These images are then sent to the controller for processing. The camera must be able to take images with enough clarity to analyze pixel colors of objects at distances up to twenty-five feet, and it must be able to take at least five pictures per second, which will be limited primarily by the controller's processing speed.

The camera we chose to use is the Logitech c270 720p HD Webcam [6]. This camera is perfect for our project because it is a small, lightweight and inexpensive. Thus, it will not weigh down our system and it leaves us with plenty of money left to spend on other components of our project. The 720p resolution is clear enough for us to see color details of objects at far distances and having fewer pixels than a 1080p camera allows us to process images significantly faster than we could with a HD webcam that captures more pixels. The webcam plugs into

TABLE II
Imaging System Requirements

Specification	Value
Image capture distance	5-25ft
Capture rate	>=5 frames/sec
Imaging processing time	<=200ms
Resolution	720p

our BeagleBone Black controller via USB which allows for easy integration with the controller and easy connection even when mounted on our funnel.

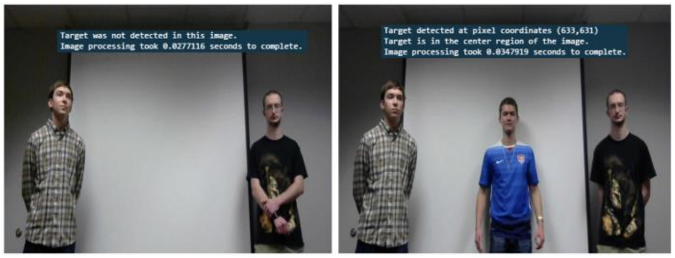

Figure D: Demo of Imaging System Basic Concepts

Due to the simplicity of the Imaging System, this block has already been built and integrated with the controller. We tested this block by taking images and sending them to the controller for processing, which worked without issue. We will further analyze the efficiency of this camera by observing how many images per second it can take when our image processing system is integrated with the mechanical system.

C. Block 2: Controller

The controller block of our system is a very significant part as it is responsible for processing the images taken by the camera and sending signals to the mechanical system indicating how it should move. The technology that we chose to use for this block is the BeagleBone Black Rev C microcontroller. We chose this controller over other options, such as Arduino and Raspberry Pi, primarily for its processor speed of 1 GHz and its ability to run a Linux operating system.

Constructing this block is primarily composed of writing software in C++ capable of performing quick image processing and integrating a controller with a mechanical system. As such, techniques from ECE 373 (Software Intensive Engineering) and ECE 354 (Computer Systems Lab II) were and will continue to be used to build this block. Most of our knowledge and experience writing C++ code and running scripts on Linux-based systems was gained from our studies in ECE 373, and ECE 354 taught us all about embedded systems and specifically writing programs capable of simple image processing techniques. All of these skills are essential to the completion of this block of our project and have been extremely helpful thus far.

The purpose of the image processing in our system is to determine the position of the shooter on the basketball court. While we considered many different image processing techniques, we found that the most efficient method for our target detection is to have the shooter wear a jersey with a specific color pattern on it and have the camera look for that pattern. Thus, the image processing technique that we chose for our target detection is color filtering. The color filtering approach is exactly what it sounds like-the camera captures an image, and then the controller filters through the different color values in that image until it finds the value or range that it is looking for and then performs some action.

Our color filtering code is written in C++ and it is run on our BeagleBone Black's Debian Linux 7 operating system using the board's bash command line from the Cloud9 IDE, which is hosted by the BeagleBone itself. We decided to use the OpenCV [7] and Video4Linux2 [8] libraries to complete our image processing. The V4L2 functions allow us to access the camera and its information from the BeagleBone, while the OpenCV functions provide us with data structures, methods, and API's specifically for accessing the pixels of captured images and performing image processing.

TABLE III
CONTROLLER SYSTEM REQUIREMENTS

Specification	Value
Capture Distance	5 – 25 ft.
Capture Rate	>=5 frames per second
Processing Time	<=200ms

For MDR, we promised to have a color filtering code for simple target detection, and we accomplished this. Our simple target detection code begins by sampling a desired target color that is instructed to be placed in the center of the camera's vision. The code determines whether the target color is a clear blue, clear green, clear red, or a mixture of the three and then sets target detection conditions based on that decision. Any image taken after the sample is processed as the code goes through each image pixel by pixel, analyzing each pixel value and comparing it to the detection conditions set initially. We used an OpenCV mat data structure to format each pixel of every image to contain three bytes: one for blue value, one for red value, and one for green value. Thus, detection conditions can weigh specific color values against each other, or set a range on each byte for values to fall within depending on what the code determined when sampling the target color. Once the desired color is found, the code determines whether the pixel coordinates are in the left, center, or right region of the image and indicates how the camera must be moved to center the target. This logic will later be used to tell the motor how to move to keep the shooter centered in the image to keep the funnel facing them at all times.

Presently, our image processing takes between twenty and forty milliseconds to complete, which would allow for twenty-five to fifty frames per second, which is well over the minimum of five that we require. However, our code will become more complex after we choose a jersey pattern and integrate our system. We found that our code is very good at finding distinct colors such as clear blues, clear greens, and clear reds, but can have issues finding mixed colors at different distances and lighting due to illumination. Therefore, we plan to design our jersey with horizontal stripes or sections comprised of a clear blue, a clear green, and a clear red color. The code will then look for these colors next to each other. This should eliminate any problems with illumination as it can easily find these three colors regardless of lighting and distance. It should also make detection errors nearly impossible as the probability of finding the jersey pattern in a place on a basketball court that is not the jersey itself will be nearly zero. The code will become more complex, though, as detection conditions and search algorithms will have to be a lot more particular for pattern detection. This will likely increase processing time. However, since our goal is to have processing take less than two hundred milliseconds and our current processing is taking less than forty, we should be able to adapt our code with increased complexity and still be able to capture at least five frames per second as desired.

D. Block 3: Power System

The power system component of the system must regulate our input voltage to power the motor and BeagleBone. On the high end, we estimate that the motor will draw 2A and the BeagleBone Black Controller will draw 1A for a total current draw of 3A. Also, note that the 2A estimate for the motor is during motor startup. When DC motors start moving they instantaneously pull a large amount of current called "inrush current." Because the motor in our system constantly changes direction, it is crucial that our electronics can provide this amount of current. We estimate the continuous running current of the motor to be closer to 1A, but are designing around the 2A inrush current estimate. All motor current estimates are extrapolated from our initial motor tests.

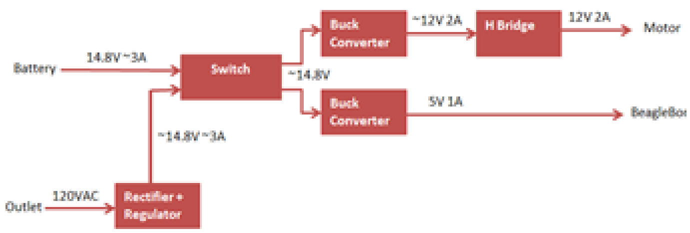

Figure E: Further Breakdown of Power Block

The block diagram for this subsystem is shown in Figure E. The system is powered in one of two modes, as selected by a switch. In outlet mode, the system can be plugged into a 120VAC outlet. While this mode restricts the locations in which the system can be used, it also allows the system to be run for an indefinite period. This mode is ideal for when the user has access to outlet power, possibly while practicing in the gym or close to a house. In battery mode, the system runs off a 4 cell, 14.8V lithium polymer (LIPO) rechargeable battery pack. In this mode, system life is limited but the system itself is self-contained, allowing for its deployment in a variety of locations.

We selected a LIPO battery for battery mode because it has the highest energy density of rechargeable batteries on the market today [9]. This allows us to run the system for long periods without costing us much in terms of system weight. Moreover, even small LIPO batteries are rated to supply the necessary current of 3A to power the system for at least an hour. We selected a 5000mAh LIPO battery to ensure that we meet this requirement.

In the subsequent stages in the block diagram we plan to use buck converters to convert our input voltage of 14.8V down to 12V and 5V to power the motor and BeagleBone respectively. We have selected chips by Alpha & Omega Semiconductor to perform as our buck converters. The

TABLE IV
POWER SYSTEM REQUIREMENTS

Specification	Value
Motor Voltage	12V
BeagleBone Voltage	5V
Operation Time	>=1 hour
Current	>=3A

AOZ1031Al has a maximum output current of 3A and can accept input voltages from 4.5-18V. At our voltage and current estimates, we expect to achieve about 90% power efficiency using this chip to regulate the voltage to the motor (12V, ~2A) [10]. This chip is also suitable as a buck converter for the BeagleBone though it is less efficient at such a relatively light load. For this reason, we purchased the Alpha & Omega AOZ1280, which can provide only 1.2A of current for input voltages between 3-26V. The chip is optimized for lower currents, so we can also achieve 90% power efficiency to the BeagleBone using this chip [11]. With the AOZ1031Al we may only achieve 80-85% efficiency when used for the BeagleBone.

Finally, we must also design an H Bridge circuit to control the motor direction. For this, we purchased the Texas Instruments DRV8801 Full-Bridge Motor Driver. The chip can be controlled by the BeagleBone, is rated for an 8-36V operating voltage, and can provide up to 2.8A of current to the motor in either direction [12].

The design of this subsystem will be led by Adam and Derek. As the EEs in the group, they will use their knowledge of electronics and buck converters from ECE 324 to design the circuitry around our selected ICs. We had to learn a lot about battery types, DC motor inrush current, buck converter chips, and H Bridges to design this block and will continue to learn more about these subjects as we begin implementing the design. We will also have to learn about PCB design for the final product.

We have already performed power measurement tests on the motor to obtain estimates for motor current draw. Moving forward we plan to design and test each component individually before system integration. Fortunately, the datasheets for our components provide design procedures and example applications to aid us in this process. During integration, we must ensure that the system can run on battery power for at least one hour. We must also ensure that the system operates properly in both power modes and that both the BeagleBone and the motor receive their proper voltages and currents.

The results of these tests will tell us if our design is operating properly and efficiently. We will analyze the results to find potential bottlenecks in the system so that we can optimize the design to achieve better performance.

E. Block 4: Mechanical System

This project has a heavy mechanical aspect, which is why it requires its own block. The mechanical system is composed of the funnel, the motor, and the gears. It also consists of mounting and casing designs.

One of the main focuses of the semester was to design the gears used to rotate the funnel and determine which motor we would use to rotate the gears. The decision on the motor was led by Devon O'Rourke while the gear design was led by Adam Paranay.

As Electrical and Computer Systems Engineers, we rarely work with motors during our undergraduate career. When we did, we didn't have to choose one from scratch. We were given

TABLE V
MECHANICAL SYSTEM REQUIREMENTS

Specification	Value
Weight	<15lbs
RPM	45deg/s – 55deg/s
Gear 1 diameter	13in
Gear 2 teeth	>=20

one as well as the specific function in which it would work correctly. In this project we needed to find a motor that would work with the requirements we set. This meant we needed to learn specific things about motors that we were not taught in our classes.

There are many different variations of motors that are sold commercially and we needed to calculate specific needs of the desired motor before shopping around. The requirements from Table V guided this decision significantly. Of the many characteristics of motors, the most important was the maximum RPM and the stall torque. We needed to make sure that our desired RPM and required torque matched well with that of the motor. Our desired RPM was set with the help of Computer Science Professor Rod Grupen in the Computer Science Department here at UMass. He suggested we use a RPM of around 50deg/s. We decided we would use a range that would be able to track the player quickly enough, seen in Table V. The torque needed to rotate the funnel was determined with the help of Francis Caron and the MIE Department. By using a force gauge from the MIE Department, Adam and Derek were able to measure the force needed to rotate the funnel while under a simulated weight. They measured ~8lbs. By multiplying this by the radius of Gear 1 we received our minimum torque needed to spin the funnel. With the help of mechanical engineering student Joe Howard and some research on gear properties, we decided upon a radius of 1in for our pinion gear, Gear 2 [13]. Using this radius and the minimum required teeth on a pinion gear, we chose 20 teeth for Gear 2. This gives us all of our knowns for the motor.

The next step of the process was to put all of our knowns in an Excel sheet, plug them into formulas (see formulas below), and generate the expected values needed for minimum stall torque and maximum RPM (without load). The main formulas we used took in stall torque and maximum RPM without load and outputted a RPM with our estimated load. We used stall torques and RPMs of motors from a well-known motor hobbyist website [14]. After thorough discussion as a team, we decided upon a 60RPM HD Planetary Gear Motor because it offered the highest stall torque with the closet RPM to what we wanted when under load. This meant we could run the motor at its nominal voltage of 12V.

Figure F: Motor and Pinion Gear Mounted to SKLZ Unit

<u>Formulas</u>
RPM_a*Teeth_a = RPM_b*Teeth_b
RPM_b = (RPM_a)*(Teeth_a/Teeth_b)
Torque Ratio → (Teeth_a/Teeth_b) = (RPM_a/RPM_b)

We needed to design a mechanical system that could rotate the outer shell of the SKLZ system around its inner ring that attaches to the rim of a basketball hoop. Being a team of electrical and computer systems engineers, we didn’t have too much experience with designing gears or mounting systems, and we couldn’t look to any of our course material for guidance either. Still, we were confident in our ability to come up with a solution to our problem. Our initial idea of creating a system of two gears, driven by the smaller motor mounted pinion gear, was what we decided to implement in our project. We knew it was important to look to other possible approaches before moving ahead with an idea, so we considered several other ideas before moving ahead with the gear design.

The only other major design idea we came up with was that of implementing a friction wheel design. Instead of a gear attached to the motor, we would have a small wheel that would be pushed upwards into the stationary ring. The motor would need to be positioned underneath the track of the ring and mounted to the body of the shell, since that is what needs to spin. When the motor turns the friction wheel, it would “grab” hold of the ring and roll the shell around it. We immediately found problems with this idea. First, the mechanical design necessary to apply the correct amount of force to the wheel/ring pair would be very complex. If too much upward force is applied to the wheel, then the system would likely jam and wouldn’t rotate as desired. If too little upward force was applied to the wheel, then the wheel would just slip on the ring and the system wouldn’t rotate either. Therefore, we would need to design a properly tuned spring loaded mount for the friction wheel so that the correct amount of force would always be applied. This seemed much too complicated a solution for our project. In addition, there is the concern that the wheel/ring pair would degrade over time, since the system would operate on the friction between them.

Instead, we decided that a gear system would be perfect for our project because it is easier to implement and much more reliable. We would replace the inner ring that came with the system (that the straps attach to) with a near identical ring with gear teeth along its outside edge. In the back of the SKLZ system, we cut a two-inch window in the ring track where the small gear (attached to the motor shaft) would interface with the redesigned inner ring gear. Both of our gears would be designed by us and 3D printed here on campus, since you can’t find a 13-inch diameter ring gear online for purchase. The motor is mounted to the SKLZ system underneath the gear window to allow the two gears to mesh fully. Using one of the motor mounts that Devon had purchased along with our motor, Adam was able to create a mounting bracket and attach the motor and pinion gear to the system.

The actual design of the gears was done on Autodesk Inventor by Adam. After learning the basics of the program, he learned how to use the Inventor gear creation tool. Using this tool, you can specify several known parameters and have the software calculate the last unknown. He was able to input the number of teeth, gear ratio, and center distance (which indirectly specified the diameter of each gear), and have the creation tool calculate the module of each gear (the tooth size and pitch).

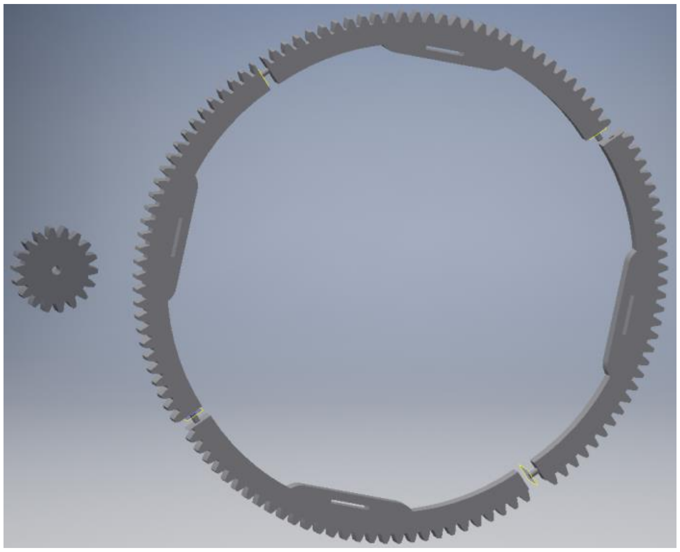

Figure G: 3D Model of Pinion Gear and Inner Gear

After creating each gear using the tool, it was important to check that the gears would mesh properly before moving ahead with further modifications and eventually printing. By using physical constrains, he was able to constrain faces, axis, and planes of the two gears to achieve rotation within the program. By doing this, he verified that the gears mated well and prevented the need for additional 3D printing in the future due to poorly designed gears. He then moved ahead and created a D shaped motor shaft hole in the two-inch diameter pinion gear. Also, he made the large gear into a ring by removing the center area of the gear. He was then able to split the large gear into four quarters, since the singular gear was too big to 3D print as one piece. He added connection pegs and holes to the faces of the split gear so they would be easy to piece together and then acetone to make a solid finished product. Finally, he created the strap mounts and added one to each quarter of the gear ring so the system could be hung and tested.

We were skeptical of the strength of the 3D printed gears and figured that there may be problems with our first print anyways. Therefore, we were expecting to have to print another round of gears. We were extremely happy to find that the finished product, after using acetone to melt the four quarters together into one large gear, was near perfect! The gears meshed very smoothly and there was no way we were going to have a gear tooth break off. In addition, the pinion gear fit onto the motor shaft well, and we secured it with Guerrilla glue to make sure the metal motor shaft would not eat into the plastic gear. After completing the gears and getting the motor/pinion gear mounted to the body of the SKLZ system, we were eager to test it. We used a ladder to hang the system with completed mechanical components, and powered the system with a ±12V powered breadboard. By adding a switch, we could test the effect of rapidly changing directions by switching between +12V and -12V. All in all, we found that the design we implemented had no problems rotating. We tested the system under normal operation, under a simulated load (by adding weight where we are planning on mounting the battery, camera, etc.), and under the effect of a basketball being shot at and through the system. The integrity of the gears has been maintained throughout testing and we are excited to move onto CDR.

Figure H: MDR Demo System with Motor Integration

III. Project Management

As a team, we were able to meet all of our Midway Design Review deliverables. The table below shows the goals we had promised at Preliminary Design Review as well as their current status. For Cumulative Design Review, we will deliver at least three things: full integration of the webcam, BeagleBone Black, and the motor, a completed power system breadboard design, and a mounting design for the integrated sub-units.

TABLE VI
MDR Goals

Goal	Status
Demo of rotating funnel	Completed
Decision on power system	Completed
Webcam/controller integration	Completed
Image processing target detection	Completed

Team 10 works well together. We constantly meet the deadlines we establish for ourselves as well as the deadlines of our advisor/faculty evaluators. Derek Foster and Adam Paranay are EEs while Brain Acker and Devon O'Rourke are CSEs. This gives us a solid background in both hardware and software. Each of us has worked in a professional engineering setting giving us a leg up in terms of working responsibly within a team.

As a team, we worked to set goals for the semester early on. After setting the goals and assigning leads for each one, we individually worked on our parts. While we achieved all the goals we set for ourselves, the roles of each member in each

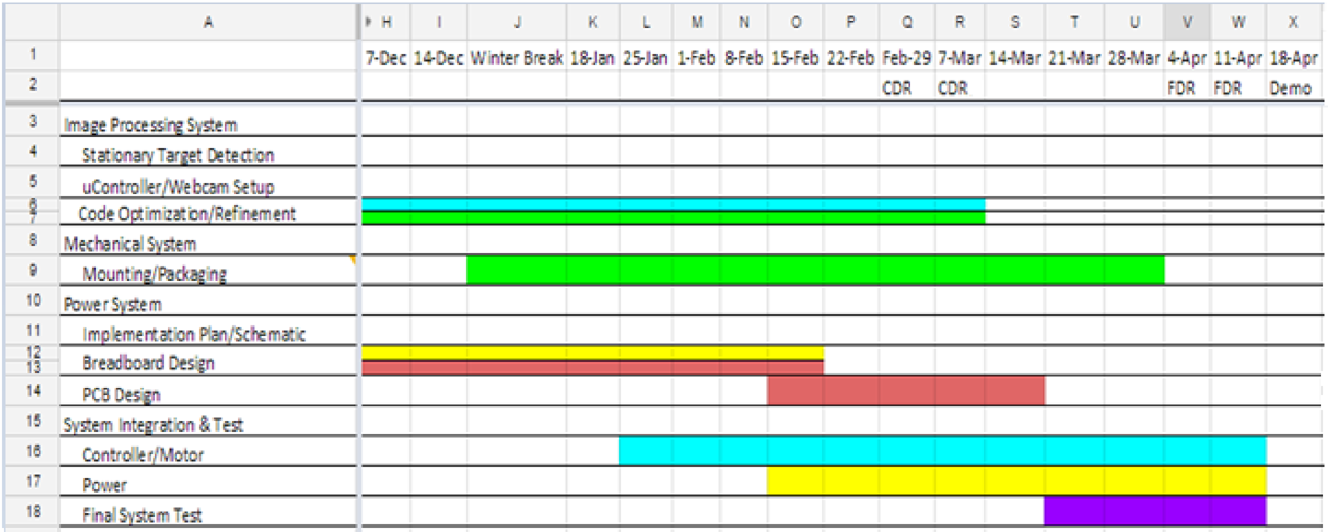

Figure I: Gantt Chart

sub-unit changed slightly from the projected roles assigned. We understand that this project will require adaptation and a dynamic outlook. Everyone is helping out with every aspect in the project. The communication we have as a team has contributed heavily to our success. Our weekly meetings allow us to check in with each other's progress. In addition, this gives us an avenue to help each other and offer insight on every part of the project.

IV. Conclusion

As of now, Team 10 is right on track with what we projected for ourselves. We completed all of our MDR deliverables and received impressive reviews from our advisor and faculty evaluators.

The project itself is moving along quite nicely. We have most of the individual systems completed and ready for integration with the other sub-systems. As mentioned in the previous section we have three main goals for CDR with the focus relying heavily on the integration of the mechanical sub-system and the imaging sub-system. As with MDR, each deliverable will have specific leads but we will work as a team even more so during the integration phase. Our goals for CDR can be found in Table VII.

TABLE VII
CDR Goals

Goal	Status
Integration of the imaging and mechanicals sub-systems.	In Progress
Power system.	In Progress
The mounting design.	In Progress

Acknowledgment

We want to thank Professor Hossein Pishro-Nik for his tireless support and the professional and personal insight he has provided our team. Acknowledgements also go out to: Joe Howard, Professors Kelly, Holcomb, Duarte, and Grupen.

References

[1] 2016. [Online]. Available: [1] https://drdishbasketball.com/ic3/. [Accessed: 19- Jan- 2016].

[2] Amazon, 2016. [Online]. Available: http://www.amazon.com/Goalrilla-B2608W-Goaliath-Basketball-Return/dp/B002UPWBUK/ref=sr_1_34?ie=UTF8&qid=1452970711&sr=8-34&keywords=goalrilla. [Accessed: 19- Jan- 2016].

[3] D. Rebel, "Basketball Shooting Machine | Dr. Dish Rebel", *Dr. Dish*, 2016. [Online]. Available: https://drdishbasketball.com/product/dr-dish-rebel/. [Accessed: 20- Jan- 2016].

[4] Amazon.com, "Amazon.com : SKLZ Shoot Around 180° Ball Return : Sports & Outdoors", 2016. [Online]. Available: http://www.amazon.com/SKLZ-Shoot-Around-180%C2%B0-Return/dp/B013B8ICNY/ref=sr_1_78?srs=2600676011&ie=UTF8&qid=1453175005&sr=8-78&keywords=basketball. [Accessed: 19- Jan- 2016].

[5] Sparkfun.com, "Beaglebone Black - Rev C - DEV-12857 - SparkFun Electronics", 2016. [Online]. Available: https://www.sparkfun.com/products/12857. [Accessed: 19- Jan- 2016].

[6] Amazon.com, "Amazon.com: Logitech HD Webcam C270, 720p Widescreen Video Calling and Recording: Computers & Accessories", 2016. [Online]. Available: http://www.amazon.com/Logitech-Webcam-Widescreen-Calling-Recording/dp/B004FHO5Y6. [Accessed: 20- Jan- 2016].

[7] Opencv.org, "OpenCV | OpenCV", 2016. [Online]. Available: http://opencv.org/. [Accessed: 19- Jan- 2016].

[8] Wikipedia, "Video4Linux", 2016. [Online]. Available: https://en.wikipedia.org/wiki/Video4Linux. [Accessed: 19- Jan- 2016].

[9] L. Keith Arauo - Epec, 'Primary and Rechargeable Battery Chemistries with Energy Density', *Epectec.com*, 2015. [Online]. Available: http://www.epectec.com/batteries/chemistry/. [Accessed: 11- Dec- 2015].

[10] Alpha & Omega Semiconductor. (2010, March). "*AOZ1031Al EZBuck 3A Buck Regulator"* [Online]. Available: http://aosmd.com/res/data_sheets/AOZ1031AI.pdf [Dec. 11, 2015].

[11] Alpha & Omega Semiconductor. (2011, Aug.). "*AOZ1280 EZBuck 1.2A Simple Buck Regulator"* [Online]. Available: http://aosmd.com/res/data_sheets/AOZ1031AI.pdf [Dec. 11, 2015].

[12] Texas Instruments. (2015, March). "*DRV880x DMOS Full-Bridge Motor Drivers."* [Online]. Available: http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=drv8801&fileType=pdf [Dec. 11, 2015].

[13] Understanding Gear Ratios, 2016. [Online]. Available: https://www.teamassociated.com/pdf/cars_and_trucks/shared/gear_ratios.pdf. [Accessed: 20- Jan- 2016].

[14] Servocity.com, "60 RPM HD Precision Planetary Gear Motor", 2016. [Online]. Available: https://www.servocity.com/html/60_rpm_hd_precision_planetary_.html. [Accessed: 20- Jan- 2016].

FIGURES

[A] I.ytimg.com, 2016. [Online]. Available: https://i.ytimg.com/vi/QuUIJmdrvsM/maxresdefault.jpg. [Accessed: 19- Jan- 2016].

[B]Amazon.com, "Amazon.com : SKLZ Shoot Around 180Â° Ball Return : Sports & Outdoors", 2016. [Online]. Available: http://www.amazon.com/SKLZ-Shoot-Around-180%C2%B0-Return/dp/B013B8ICNY/ref=sr_1_78?srs=2600676011&ie=UTF8&qid=1453175005&sr=8-78&keywords=basketball. [Accessed: 19- Jan- 2016].