# ARK: MDR Report

Ethan Miller (EE), Jacqueline Lagasse (CSE), Matthew Bolognese (EE), and Charles Klinefelter (CSE)

*Abstract*—**Augmented Reality Kick (ARK) is a gaming peripheral and Android app. A virtual reality headset holds a phone in front of a user's face, while running the ARK app. The app itself displays what the rear camera can see, and overlays a virtual ball and net in a suitably flat area. Utilizing an inertial sensor on the foot, a user can then kick the virtual ball into the net to score points.**

## I. Introduction

AT the core of this project is augmented reality (AR). AR is the dynamic response of a computer to its environment. Using computer vision or object recognition, a computer "augments" the world it sees, most often through overlaid sounds and visuals. While a user requires goggles or a screen to use this technology similarly to Virtual Reality (VR), they are not the same [1]. VR relies on constructing a world around the user, whereas AR must first understand its surroundings and construct from there.

AR is of growing popularity among tech companies, particularly Microsoft, Google, Apple. While Microsoft has the standalone HoloLens product, Apple and Google have primarily focused on mobile phone integration. Both have developer-supported AR platforms (ARKit and ARCore, respectively) which enable developers to utilize tools within smartphones that detect and track surfaces, as well as placing models on them.

The potential for gaming with AR has been demonstrated by the massively popular and successful *Pokemon Go!* Released by Niantic in 2016. As of July 2017 it has been downloaded almost 10 million times. The premise of this game is that as you walk around in real life, you can find and catch virtual pokemon with your smartphone.

One major limitation of this gaming setup is the lack of a controller. Currently there are no peripherals for augmented reality on smartphones outside of the headsets users can wear. Given that wearing a smartphone means you cannot interact with it via the touchscreen, this means that the inputs for any controls have to come from an onboard sensor such as an accelerometer or GPS. It follows that since the phone is attached to the user's head, it can only follow the direction or position of the head, and cannot sense the extremities without seeing them with the camera. It is proposed that inertial sensors allow for movement of extremities to be sensed, recorded, and acted upon.

Inertial sensors have been used before in gait reconstruction and limb tracking [4],[6],[7]. The applications of this technology have proven useful in physical training, whereby trainers and physicians examine how a person moves their body, but with more detail than a visual inspection. As shown by Thiel et al. [8], the use of inertial sensors can demonstrate a congruence between the lateral tilt of a ballet dancer's torso and lower scores from judges. Such observations can also be used in physical therapy; when a physically injured person demonstrates motion to a doctor, inertial sensors can provide indicators of strong or weak healing of muscles or tendons.

For an inertial sensor to be used in a more active environment, some requirements must be met. The sensor package must fit on a user's shoe, without overhanging so much as to trip the user or become damaged from clipping an object. The sensor package must also be lightweight, as too much weight on a user's foot can upset their balance or become outright uncomfortable. Lastly, the system must last a long enough time on a single charge for a user to have multiple sessions using the device. Below are the requirements, quantified:

TABLE 1
SENSOR PACKAGE REQUIREMENTS

| Specification | Value |
|---|---|
| Length | ≤ 7 cm |
| Width | ≤ 7 cm |
| Height | ≤ 2 cm |
| Weight | < 200g |
| Battery Life | > 5 hours/charge |

## II. Design

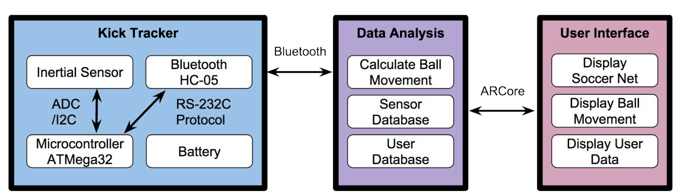

FIGURE 1. BLOCK DIAGRAM.

To create a sensor package as described above, referred to from here on out as the "kick tracker," we will need a few core devices, as introduced in Figure 1. These include an inertial sensor, a microprocessor, a wireless module, and a battery. These components will be described in further detail in the following blocks.

To start, it was necessary to consider what minimum data we would need to represent the motion of a human extremity- namely a foot in our use case, and particularly the motion of kicking. At a basic level, acceleration data can be used to determine whether a user is performing a kicking movement or not. In order to determine more about a kick than simply whether or not it is happening, we must also incorporate direction, which will require the use of a gyroscope. With these two components, we will be able to determine what type of kick the user has made, and reflect this in-game.

This acceleration and rotational data must be pulled constantly from the inertial sensor. There are multiple possible protocols through which this could be implemented. We initially tested with analog sensors, due to the simplicity of interfacing with them. However, other possibilities include Serial Peripheral Interface (SPI) and Inter-Integrated Circuit ($I^2C$), both of which output sensor data in a digital format. We have decided to use $I^2C$, because although both protocols are quite similar, $I^2C$ makes sense for our purposes because it uses a single data line in all cases, which SPI does not. Since we are only receiving data from the sensor and not writing to it, with the exception of ACK messages from the processor, it is a slightly simpler setup for our purposes. Our current sensor will output 16-bit datagrams to be sent to the processor.

From the processor, we will use Bluetooth LE to transfer the data from the sensor to the phone. While WiFi is also an option, it is important to note that this would limit usability, as it requires the system to be on a network to be used. Bluetooth allows the user to connect the kick tracker directly to their device at any time. Additionally, the latency of the BLE data transfer is around 6ms. Given our need for end-to-end latency to be <100ms in order to retain user immersion, this allows plenty of flexibility in additional processing time, while still staying below our delay cutoff.

For the purpose of our virtual game, we must qualify what constitutes a kick. This includes setting acceleration thresholds to determine the strength of a kick, as well as using the pitch, yaw, and roll to determine what kind of kick the user is trying to make. Each piece of data must be compared to these profiles and match with one in order for a kick to happen. Once a piece of inertial data has been transferred to the phone, these calculations will occur. At that point, user movement will result in movement of the virtual ball if our system determines that a user has performed a sufficient kick.

All of this needs to be displayed on a smartphone, being held in a generic AR headset. Given very recent advances in AR from both the iOS and Android communities, either option would have been suitable for our end product, but we have chosen to work with the Android OS given our accessibility to AR-capable devices as well as its higher global market share.

### *A. Inertial Sensor*

The most fundamental piece of hardware is the inertial sensor. Our design will include an Adafruit sensor known as the GY-521 MPU-6050, an inexpensive, thin, and low-powered chip that consists of both a triple-axis accelerometer and a triple-axis gyroscope, as acceleration data and rotational data are essential to our data processing. The digital output accelerometer built into this sensor is capable of measuring a full scale range of ±2g, ±4g, ±8g, and ±16g. The digital output gyroscope also contains a user-programmable full scale range of ±250deg/sec, ±500deg/sec, ±1000deg/sec, and ±2000deg/sec. This sensor also makes use of three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. These parts allow for precise tracking of both fast and slow motion. The MPU-6050 has a communication interface of I2C, which allows it to read data from external sensors such as magnetometers. This interface will be used to communicate with our choice of a Microcontroller in our design.

### *B. Microprocessor*

The next piece to our kick tracker is a sufficient microprocessor. We have chosen Atmel's low-power CMOS 8-bit microcontroller ATMega32/L due to its inexpensive cost, its versatility in interfacing, and reasonable size. This processor contains several communication interfaces that will be used for our design which includes I2C and ADC for collecting sensor data from the MPU-6050 sensor, and a serial programmable USART to communicate with the Bluetooth module. This processor is also capable of obtaining a high throughput of up to 16 MIPS (million instructions per second) at a clock rate of 16MHz.

### *C. Wireless Module*

Our wireless interfacing device is the HC-05 Wireless Bluetooth Serial Transceiver, a simple Bluetooth module that provides the link between kick tracker and smartphone. This part is also inexpensive with a USART interface that will allow it to communicate with the ATmega32 processor.

### *D. Power System*

Lastly, we will be powering the system with a 3.7V 18650 lithium ion rechargeable battery cell. These batteries can be purchased at cheap prices (~$5 to $9) and contain a nominal capacity typically between 2500mAh and 3000mAh, which is more than enough time to power our system; they can also be recharged up to 1200 times. Some safety features included within these batteries consist of a safety vent to prevent from an excessive buildup of pressure and a built-in PTC (Pressure, Temperature, Current Switch) to protect against current surges. The battery will also be enclosed to prevent it from being damaged when attached to the "kick tracker."

### *E. Data Analysis*

Data analysis will occur on the phone itself, with the most significant component of this block being the calculation of the ball movement. As described before, this will require comparisons between incoming data and predetermined

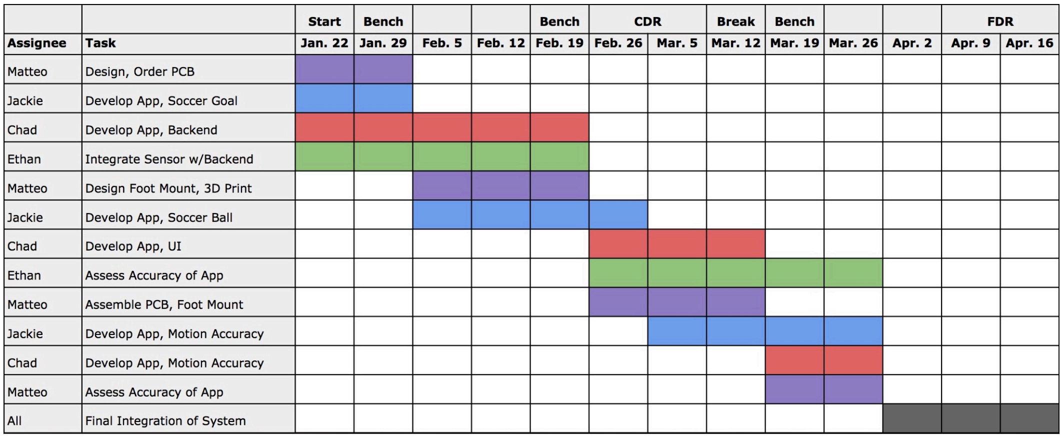


FIGURE 2. GANTT CHART.

thresholds. A small space will be allocated for sensor data storage in order to make sure that all the necessary datagrams to constitute a single kick are updated. Additionally, space will be allocated to store user data in reference to gameplay.

These calculations will occur on the phone itself, so that we can retrieve the raw sensor data from the processor as quickly as possible, then store it until all of the necessary three-axis data has been retrieved.

### F. *User Interface*

Using Google's ARCore API, we will implement a basic soccer ball/net environment, overlaid on the physical space that the user's smartphone camera can see from its headset mounting. This system then is best used in a relatively open space, in order to provide realistic depth perception.

While the user will see both the virtual ball and net, as well as their leg if it comes into view, the kicking will not be vision-based. The app's display of a kick will be entirely based on the input from the kick tracker.

## III. PROJECT MANAGEMENT

The following is a table of our proposed MDR deliverables and their completion status at the time of our MDR presentation:

TABLE 2
MDR DELIVERABLES

| Deliverable | Status | Team Member |
|---|---|---|
| Select and purchase inertial sensor, microcontroller, Bluetooth module, and battery | Completed (except battery) | Matteo & Ethan |
| Assemble purchased components together | Completed | Matteo, Jackie, Chad, & Ethan |
| Send data from inertial sensors to phone over Bluetooth | Completed | Jackie, Chad, & Ethan |
| Display data from sensors in Android App | Completed | Jackie & Chad |

In the week following MDR, we were able to address ordering the battery, so now all listed MDR deliverables have been completed.

The first deliverable was accomplished by researching all components in detail and selecting appropriate options for our project. More details can be found for the components in the design section of this paper. The next deliverable was to assemble all the components onto a breadboard correctly, which we were able to do with ease. The third deliverable was the most challenging of all, which required us to properly receive data over Bluetooth. The main challenge was with the complexity of the sensor we selected, which used the I2C protocol and also required additional programming. We ultimately realized that this sensor would not be optimal to use with our design and we were able use an analog sensor to complete this deliverable in time for MDR. However, an analog sensor is not ideal for this system so we will be using a simplified I2C sensor instead. We will need to interface with this new I2C sensor by the beginning of the spring semester. The last deliverable was achieved by creating an Android app that could sample the analog data and use it to create a real time graph. The data was displayed in three separate charts, one for each axis, and displayed realistic data from the analog sensor according to how it was moved in real time.

TABLE 3
CDR DELIVERABLES

| Deliverable | Member |
|---|---|
| Augmented reality must be implemented to show soccer goal and soccer ball | Jackie |
| Backend of app must store sensor data, model data to analyze foot movement | Chad & Ethan |
| Kick accuracy measurement | Ethan |
| Design PCB, send for fabrication | Matteo |
| Design foot mount structure and enclosure | Matteo |

Our proposed CDR deliverables are listed in Table 3. These deliverables are also displayed chronologically in a Gantt chart shown in Figure 2. The team will be separated as follows so that the team members with strongest software abilities will be working with the app, and the team members with the strongest electrical and mechanical backgrounds will be working on creating the physical device and packaging it so that it can be attached to one's foot.

Jackie and Chad, who have the strongest background in software, will be focused with programming the augmented reality and backend data management of the app respectively. Ethan, who is strong in both software and hardware, will be assisting Chad with the backend of the app and will also be working on testing the device for accuracy. Matteo, who is the strongest with electrical and mechanical components and wiring, will be devoted to working on the PCB and physical structure of the device. Our team communicates each week for one hour with our adviser and together for about one hour without our adviser.

## IV. Conclusion

After PDR, there were some setbacks. It was initially intended that an omnidirectional treadmill for use in VR would be created, but it proved infeasible to innovate on given the current number of existing solutions and variety of implementations. It was therefore decided to change the direction of the project from VR to AR, with the intent on overcoming the obstacle of creating a controller for a person to use. After further research the use of inertial sensors for research purposes was discovered. Originally used for gait reconstruction or motion analysis on dancers, it was clear that there was an ability to extrapolate information from this form of sensing.

Given our project-changing PDR, we hope to make rapid progress in the initial weeks of spring semester. We plan on making the prototype portable and attaching it to the foot, collecting data in January so that we can create the motion mappings as soon as possible. The most difficult part of the project is expected to be the dynamic detection of the kicks. While it could be relatively trivial to have a small set of possible ways to kick the ball (e.g. high, mid, low), ideally we would like to have a system that responds as much as possible to the user's foot. What we are attempting to discover is the limit of extrapolation from a single inertial sensor in tandem with a camera for location detection. We do not yet know how feasible it will be to discern the angle of a foot mid-kick, or how similar a small punt might look like a normal stride. The only way we can find out is by getting the sensor on-foot as soon as possible and seeing what data we can collect from it.

Part of the design philosophy behind sending the inertial data directly to the phone is that modern smartphones are incredibly capable devices, and have much more power and versatility in the calculations they can perform than any microcontroller. This in turn allows us the flexibility to decide exactly how our data will be processed without having to worry about the limitations of our processor on the foot.

## References

[1] "Everything You Need to Know About Augmented Reality Now That It's Invading Your Phone," Gizmodo, 2017 [Online]. Available: http://fieldguide.gizmodo.com/everything-you-need-to-know-about-augmented-reality-now-1809069515

[2] J. D. Hol, T. B. Schon, F. Gustafsson and P. J. Slycke, "Sensor Fusion for Augmented Reality," *2006 9th International Conference on Information Fusion*, Florence, 2006, pp. 1-6.

[3] M. Kanbara, H. Fujii, H. Takemura and N. Yokoya, "A stereo vision-based augmented reality system with an inertial sensor," *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, Munich, 2000, pp. 97-100.

[4] R. Slyper, J. K. Hodgins, "Action Capture with Accelerometers," Eurographics/ ACM SIGGRAPH Symposium on Computer Animation, 2008

[5] H. Yan and D. A. Johns, "An open source inertial sensor network with Bluetooth Smart," *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Halifax, NS, 2015, pp. 1018-1023.

[6] H. Yang and J. Ye, "A calibration process for tracking upper limb motion with inertial sensors," *2011 IEEE International Conference on Mechatronics and Automation*, Beijing, 2011, pp. 618-623

[7] A. Ahmadi *et al.*, "3D Human Gait Reconstruction and Monitoring Using Body-Worn Inertial Sensors and Kinematic Modeling," in *IEEE Sensors Journal*, vol. 16, no. 24, pp. 8823-8831, Dec.15, 15 2016.

[8] David V. Thiel, Julian Quandt, Sarah J.L. Carter, Gene Moyle, Accelerometer based Performance Assessment of Basic Routines in Classical Ballet, In Procedia Engineering, Volume 72, 2014, Pages 14-19, ISSN 1877-7058, https://doi.org/10.1016/j.proeng.