Active Electronic Assault System

Alex Balogh, Ryan Sivek, and Karena Stout

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Active Electronic Assault System (AEAS) is the senior design project for Group 16 overseen by the College of Engineering and Computer Science at the University of Central Florida. The project aims to provide an accurate firefight simulation tool that could potentially lead to the replacement of currently used, out-of-date laser based systems. AEAS will use various sensors and communications subsystems to determine location and orientation of a weapon in the field. This data will be communicated to a main server that calculates virtual bullet trajectories. The server can be accessed over a network to view simulation data or display a target wall using a projector.**

***Index Terms* — Weapons, Guns, Military Equipment, Military Training, Radiofrequency, Integrated Circuits, Wireless Communication, Inertial Measurement Units, Accelerometers, Gyroscopes, Network Server.**

I. Introduction

Many current US Army trainers assert the low fidelity and negative training issues associated with current laser combat training systems. High cost, low realism, impenetrability of foliage or other soft obstructions, and low fidelity 'hit' notification mechanisms are some of the more prevalent issues with current laser based systems.

The Multiple Integrated Laser Engagement System (MILES) is the most widely used laser based training system in the world. The system uses laser emitters attached to rifle barrels and laser receptors on soldiers' helmets and harnesses to simulate combat. The US Communication Code Standard for this technology as maintained by PEO-STRI defines that laser pulses encode weapon type, ammunition type, player identification, and weapon/ammunition lethality effects which are then decoded by the receiving sensor. Also, beam divergence (especially in direct sunlight) contributes a significant margin of laser signal reception error. A 'shot' from a single rifle at a certain distance can potentially 'hit' multiple targets simultaneously if they are close enough together.

The MILES system (including the more recent MILES 2000 program) has provided soldiers with a fairly effective means of training for the past 30 years. However, the laser technology is extraordinarily expensive and out of date. Due to these negative training issues, many have commented that alternative training mechanisms can and should be achieved with state-of-the-art technology. A more accurate 'virtual bullet' solution is desirable to create more effective training scenarios.

This project will be an effort to explore the possibility of providing a low cost, easy to use, high fidelity virtual weapon system that will allow soldiers to train more effectively and in more realistic conditions. Using a combination of sensors and position tracking mechanisms, we are confident in our ability to develop an effective simulation that could potentially rival the laser systems currently provided. This vision is based on the idea that given accurate enough information about weapon position and orientation in the playing field, 'shot' trajectories can be effectively estimated using simple Newtonian principles and projectile motion calculations.

Using such a system, the realism of firefight simulation training becomes much more easily enhanced with improvements to technology and the introduction of more efficient and advanced software algorithms. Due to the time and budget constraints of our project, we were not able to provide a complete virtual bullet training solution. Our efforts, however, provide a first step, proof-of concept prototype that shows the capabilities currently achievable with inexpensive components.

II. System Components

The implementation of AEAS will consist of three main parts, as shown in Figure 1. A weapon attachment will be available to those participating in the simulation which will communicate the orientation and position of the device to the server. The device is able to pinpoint its location using the ultrasonic signals sent by the positioning beacons. All of the data captured by the server is available on a web accessible GUI, and a client can also be set up to drive the projector, which will display various shooting exercises for the user.

Each of these devices will need to communicate with each other in real time to transmit various scenario data. Due to the interference that wires would cause, a wireless system is the most effective solution for communication between the weapon attachment and the AEAS server. The communications subsystem will be unobtrusive and various current models allow simple integration.

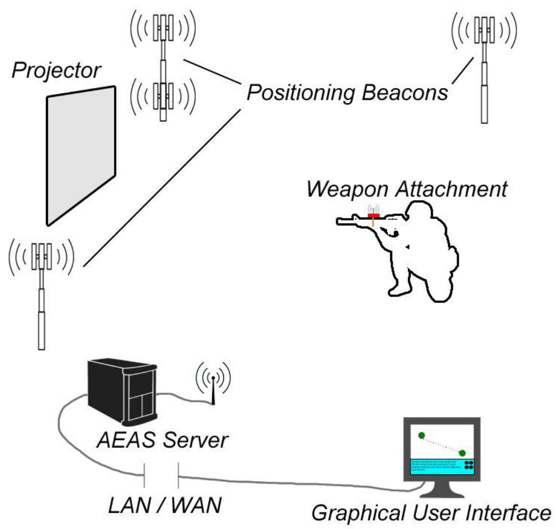


Fig. 1. Concept of the AEAS system

A. Weapon Attachment Device

This is the main component to AEAS and is an electronic device that can easily be attached to any particular weapon with a trigger, specifically military weapons. This will allow soldiers to use their own personal weapons in combat training, which is a significant qualification we had in its design. It is powered completely on a single 9V battery. The device provides a means of pinpointing the weapon's location and orientation in 3-dimensional space and will communicate this information to the server at regular intervals during the simulation. A 'shot' signal will be sent to the main server when the user presses the weapon trigger.

B. Main Server

The server is a distinct hardware and software system that pieces the combat scenario together. When a 'shot' signal is received from the weapon attachment device, data concerning the weapon orientation and position will be captured and used to mathematically determine the 'shot' trajectory based on predefined weapon specifications. This server will also provide a front end web accessible GUI that will allow trainers to view the scenario as it progresses in real time, as well as keep logs of all of the data so that it can theoretically be used in soldier's After Action Reviews.

B. Positioning Beacons

The mechanism by which the attachments can pinpoint their locations in three dimensional space will be known as a positioning beacon. A number of ultrasonic transmitters will be set up at specific points as beacons. The main server and the weapon attachment will be aware of the coordinates of these beacons. This is a reference based positioning system which can also be augmented with dead-reckoning via inertial measurement data to provide more accurate measurements.

III. System Concepts

A. Weapon Attachment Device

The weapon attachment will be a small unobtrusive 'box' designed to measure the position and orientation of the firearm as shown in Figure 2 below. This device will also send fire signals when the user presses the trigger. It will have consistent communication with the main server to send position, orientation, and fire reports with which the system will be able to piece together the scenario and calculate bullet trajectories during simulations.

Fig. 2. Concept of Weapon Attachment Device with an M16A2

This device will consist of the orientation subsystem, positioning subsystem, communications subsystem, and trigger sensing mechanism as shown in (2). The means of attaching the unit to the weapon will be a modified, releasable cable tie harness to provide a strong, unmoving attachment while also allowing the user to attach and remove the device with relative ease.

B. Main Server

Upon certain conditions, such as the notification that a 'shot' has been fired by a weapon attachment, the main server will acquire orientation and position information for a weapon, and perform precise floating point calculations to determine the trajectory of the fired 'shot'. The trajectory of the 'shot' will be calculated based on the firing weapon's specification, position, and orientation. Using this information, the main server will then determine if any user is positioned along the trajectory. The main server also will also allow trainers to monitor scenario in real-time through a front-end web accessible GUI while maintaining per user logs of the scenario.

C. Positioning System

The positioning system is an essential part to the AEAS system to determine where units and weapons are in the playing field at any given moment of time, and especially when a virtual bullet is fired. The system needs to be accurate enough to appear realistic compared to current Army training systems. In an effort to keep project costs to a minimum, and still achieve the accuracy and precision required, an ultrasonic active beacon navigation system was determined to be the most effective solution. Active beacon systems allow high sampling rates and reliability with minimal processing.

Ultrasonic transmitters and receivers are currently quite inexpensive and typically used for range finding applications in robotics.

The system causes sound waves to be emitted from prepositioned beacons at certain time intervals. These sound waves are then picked up by the receiver which calculates the time of arrival of the signal, which can then be used to calculate the distance between the receiver and the beacon. Using four or more beacons, one can use trilateration to pinpoint the receiver's coordinates in three dimensional space. Jan Dyre Bjerknes and colleagues from Bristol Robotics Laboratories developed such an ultrasonic positioning system for mobile robots and recorded an absolute error of within 3 centimeters [1]. This ultrasonic solution exceeds the needs of our system and cost requirements. Therefore, we have adapted a version of their ultrasonic positioning system design for the AEAS project.

Four ultrasonic beacons are placed along the coordinate axes and provide the required anchor points from which to measure. Aligning these beacons to the axes greatly reduces the amount of computation that the ATmega328 must perform to compute its location. This simplification allows the microcontroller to perform the trilateration algorithm in less than 10 milliseconds with four beacon points.

(1) Trilateration. Trilateration in three dimensional space is determining a position by knowing your distance from at least four known points as shown in Figure 3. In our system, those known points will be the ultrasonic beacons.
(2) The computations needed to calculate the position of our weapon (A in Figure 3) is given by the following simplified equations:

$$x = \frac{r_1^2 - r_2^2 + x_2^2}{2x_2} \qquad (1)$$

$$y = \frac{r_1^2 - r_3^2 + y_3^2}{2y_3} \qquad (2)$$

$$z = \frac{r_1^2 - r_4^2 + z_4^2}{2z_4} \qquad (3)$$

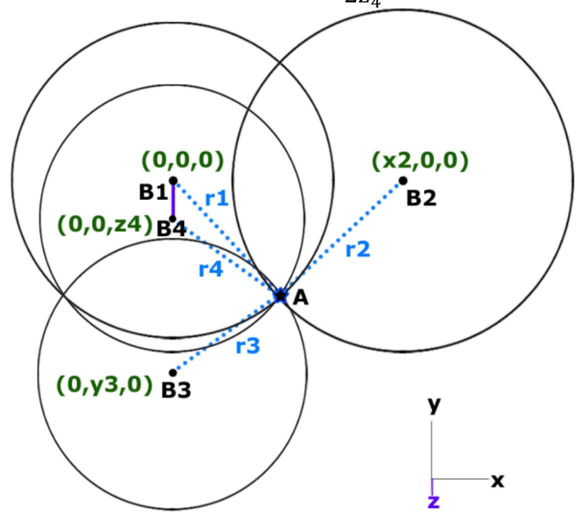


Fig. 3. Trilateration Depiction

(3) Beacons will be controlled by an Arduino Uno connected to the server via a USB port. The Arduino and the weapon attachment are time synced when a signal is sent to each device from the server. When this signal is received from the server, the Arduino fires each of the four beacons at 60ms intervals, completing in 240ms. The weapon attachment computes the position after the last signal is received and relays the result to the server.

Each of the beacons except B4 is mounted approximately 5 feet above the ground plane on self-standing plastic rods. The outer rods are connected by means of two custom 16 foot cables that are then connected together at the center beacon. This final Y connection brings all of the power, ground, and signal lines for each beacon to one single cable which is connected directly to the Arduino Uno.

IV. Hardware Design

The majority of the hardware resides on the weapon attachment. Additionally, we discuss the hardware of the positioning beacons and the server here.

A. Weapon Attachment Device

In an attempt to keep our design simple and effective, we pressed forward with our design with the intention of meeting system requirements with the minimum amount of hardware and power requirements.
(3) Microcontroller ATmega328. The 32K of program space, 20 MHz maximum clock rate, 6 ADC channels, and

23 I/O pins should be more than adequate for our purposes [3]. The internal clock for this chip runs at 8MHz, which is more than adequate for our uses. However, the calibration performed in the Atmel factory is made at a fixed operating voltage and precisely 25 degrees Celsius [4]. The internal oscillator is affected by operating voltage and temperature; therefore in order to get the level of accuracy we need, we connected an external oscillator.

(4) The orientation subsystem consists of an IMU (inertial measurement unit) which uses accelerometer and gyroscope data to make its calculations. This unit resides on the PCB with the microcontroller and provides orientation data that will be processed and then forwarded on to the main server. For the IMU we chose the Invensense MPU-6050 3-Axis Gryo/Accelerometer Motion Processing Unit (MPU) due to its low cost, ease of use, and adequate precision. The integrated circuit includes a Digital Motion Processor (DMP) which offloads the computation of motion processing algorithms from the host processor. The chip supports the I2C communication protocol, which allows for possible expansion of the module's capabilities with additional I2C devices. The device also has an auxiliary serial interface for communicating to an off-chip 3-Axis digital output magnetometer, which is useful if more precision is needed in the orientation calculations

(5) To detect when the user presses the trigger, we will use a modified FlexiForce® 25lb pressure sensor. When pressure is applied to the head section, resistance decreases from infinite to approximately 300KΩ. The sensor itself is flexible and thin, but resistance does not change while flexing. Due to its flexible design, it can be attached to the trigger with minimal interference to the user, as shown in Figure 4. The sensor will attached to wires which lead to the attachment unit. A threshold pressure of 4.5lbs was set to determine when to send a 'shot' fired signal.

Fig. 4. Trigger Sensor Placement

(6) The communication system will provide the link between the weapon and body attachments to get the server the information it needs to create the scenario. The AEAS system is designed to be used to simulate military combat, which includes a lot of running, twisting, ducking, and jumping. Obviously the AEAS system will rely on wireless communication to link both the weapon and body attachments to the server. The particular wireless communication device that we be used for the AEAS system is the XBee-Pro RF module. The hardware implementation of the communication system will be different for the base module and the field modules.

The base station will reside on the XBee USB Adapter, discussed in the Prototyping section, coupled with the server. This adapter incorporates the antenna, power, and EMC filtering circuits. The server does will not experience the space constraints of the weapon attachments, leaving the attachment board to be the best choice for the base station. This also adds the indicative LEDs as a permanent debugger when using the AEAS system. Additionally, the adapter boards are offered in bulk at a reduced price, making it also cost effective. For similar reasons the field modules will reside on adapter boards, connected to the PCB through a simple and minimal interface.

Another big consideration is the type of antenna to be used on each of the XBee-Pros in the AEAS system. The XBee-PRO RF Module accommodates four antenna styles including the Dipole Chip Antenna, the Attached Monopole Whip, the U.FL, and the RPSMA. All of these antennas are omni-directional. The wireless communication system will implement PCB chip antennas to save space in the weapon attachment, since space has become a growing consideration. Since the other antennas have longer ranges we used the whip antenna from for the base module to extend the network further. With the whip antenna and using the XBee-Pro instead of the XBee, the communication network has enough range using the PCB chip antenna to meet the requirements of the AEAS system.

(7) A typical alkaline 9V battery rated at 565 mA is more than adequate for our purposes. In practice, we achieve a much higher battery life when devices are configured to go into sleep mode when not used. The XBee can be put into a power-down mode which greatly reduces the amount of current draw of the device. To connect the battery to the circuit, we will use a simple 9V snap connector. The battery will be secured in a Radio Shack 9-Volt battery holder attached to the module box.

(8) When designing the PCB to connect these components, we chose to keep the connections minimalistic in order to keep the size of the weapon attachment as small as possible, and to reduce cost. For convenience and cost, we kept a few circuits on adapter boards, including the XBee-Pro circuit, the ultrasonic

receiver circuit, and the MPU circuit. This keeps the length and width of the PCB small. Following along with this theme, we kept the board to two layers to compensate for the adapter boards attached to the top. Our design paid off, resulting in a 1.5'' by 3'' PCB. The PCB was designed in EagleCAD, and it was fabricated through OSH Park, as suggested by SparkFun for their cheaper pricing and fast delivery.

B. Main Server

Embedded computer (such as a Raspberry Pi Model B), or a standard x86-based computer. The choice of server hardware depends upon the scale and responsiveness desired for the system. The simplest case scenario of one or two users can be handled by a single Raspberry Pi, but for large numbers of users, more robust hardware is highly recommended.

The performance of floating point computations is a one of the most important considerations for the server. In order for the system to remain responsive and accurate, native hardware support for floating point operations is required. In addition, it must be ensured that consistent results are provided from the various computations performed by the server. It has been found that ARM processors with floating point (such as the Raspberry Pi) as well as standard Intel processors (SU7300 and Q8300) meet or exceed the required processing capability. It is advantageous to use more robust processors, however, in the event of limited resources; there are many readily available components which can perform the computations quickly and with the desired level of precision.

C. Positioning System

(1) Ultrasonic Beacons. The beacons are ultrasonic transmitters placed in specific locations to allow the receiving weapon attachment to compute its location in three dimensional space. The team determined that using Maxbotix MB1200 XL-Maxsonar EZ0 to be the optimal solution for the positioning system. They operate on 3.3-5V, emit a 42kHz wave, have the widest beam range of any Maxbotix Rangefinder, and an advertized maximum range depth of approximately 7.6 meters. This guarantees that our receiver will be able to obtain sufficiently powerful signals at any point of our sensing zone.

(2) Ultrasonic Receiver. The ultrasonic receiver relays beacon signals to the weapon attachment so that trilateration calculations can be made to determine location. The receiver consists of a custom PCB designed and sold by EngineeringShock Electronics, with a modified ultrasonic receiver. We replaced the kit's receiver with a Murata MA40S4R receiver which has better attenuation at indirect angles as shown in Table 1. It also has a peak attenuation frequency of 42 kHz, which is perfect for our chosen transmitters.

Deviation Angle (degrees)	Attenuation (dB)
0	0
15	-1.5
30	-4
45	-8
60	-14
75	-20
90	-25

Table 1. MA40S4R Directivity in Sensitivity (Estimates) [4]

Even though the replacement ultrasonic receiver has improved directivity in sensitivity over the provided receiver, it still has significant signal loss at angles larger than 45 degrees. Therefore, to allow the device to optimally receive signals at all angles in a 360 degree radius, a small metallic cone is attached to the face of the receiver. The receiver is then positioned upward on the weapon attachment as shown in Figure 5.

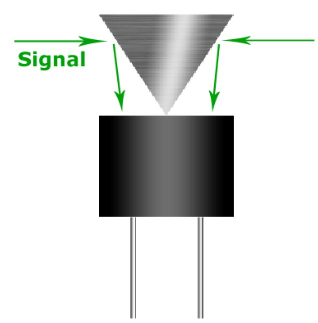


Fig. 5. Ultrasonic Receiver with cone attachment

V. SOFTWARE DESIGN

The majority of the AEAS software resides on the server, but the weapon attachment does carry some of the load, as described below.

A. Weapon Attachment Data Processing

The ATmega328 needs to communicate with four external devices for the weapon module: the MPU-6050 Motion Processing Unit (MPU), the FlexiForce sensor, the XBee Pro RF Module, and the ultrasonic receiver.

The MPU conveniently provides its own processor on board which means minimal orientation processing for the host microcontroller. All the microcontroller needs to do is apply dead reckoning calculations after programming the onboard DMP to provide a more accurate reading to the server when needed. This device also simply requires two wire connections for the I2C interface: serial data (SDA) and serial clock (SCL). The MPU will always operate naturally as a slave device to the microcontroller with a maximum bus speed of 400 kHz.

Accelerometer data may also be used to augment position data by dead reckoning [5]. Measuring acceleration deviations over time between the last received position update, and the current time can give a more precise location report. Once acceleration data is obtained, it can be integrated to obtain inertial frame velocity and position. Since this data will be received at discrete time intervals, velocity and position can be estimated by the following equations:

$$v = v_{prev} + a * T \tag{4}$$
$$R = R_{prev} + v * T \tag{5}$$

where T is the sampling period. Applying this calculation to each of the axes in three dimensional space we get the following:

$$v_x = v_{x\,prev} + a_x * T \tag{6}$$
$$R_x = R_{x\,prev} + v_x * T \tag{7}$$

$$v_y = v_{y\,prev} + a_y * T \tag{8}$$
$$R_y = R_{y\,prev} + v_y * T \tag{9}$$

$$v_z = v_{z\,prev} + a_z * T \tag{10}$$
$$R_z = R_{z\,prev} + v_z * T \tag{11}$$

The initial position R_0 for our uses will be given by the most recent position update received by the positioning system. In this way, we reduce error due to sensor inaccuracy. A trajectory error of even a single degree will cause the estimated velocity to be off by more than a 17 meters after 10 seconds [5].

B. XBee-Pro Configuration

The free X-CTU software is used to configure the base and field modules in the AEAS system. Remote configuration of the field modules is performed by the base module using the X-CTU software. Since the field modules will be designed to always communicate with the same base module, all of the configuration settings will be the same each time the AEAS system is used. We used all the default setting regarding baud rate, parity bits, etc. The XBee-Pro supports three different network topologies, including point-to-point, star, and mesh. For the AEAS system to maximize reliability, scalability, operational efficiency, and budget a star network will be implemented as the architecture model for the communication system. This makes remote configuration simple, since each field module communicates directly with the base module. It also takes full advantage of the base module having the whip antenna with the longer range.

C. Server Operating System

The software developed to perform the server trajectory processing and web functionality is written to be highly portable, thus the AEAS system is not constrained to any specific operating system. As long as the operating system is supported by the selected server hardware, and can run the Node.js framework, then the AEAS software can be utilized. This results in compatibility with Windows, Mac, and many distributions of the Linux operating system.

D. Server Software

(1) Data Structures are a large component or the server software. The server receives position and orientation data from the weapon attachment through a serial connection to a wireless XBee communications device. The data is received in the form of a buffer object, and the desired quantities are parsed from the raw data. The position and orientation data are then placed into separate high-precision floating point arrays.

$$[position_x, position_y, position_z]$$
$$[orientation_{yaw}, orientation_{pitch}, orientation_{roll}]$$

Each quantity is a three element vector, which is passed to the client web interface via a web socket connection.

In the event of a fired shot, an additional Boolean true or false parameter may be passed with the previously mentioned quantities to instantiate the computation of the trajectory and determine if any hit on the target has occurred.

For shots that have been fired and have hit the target, the position of the hit is translated to pixel coordinates and displayed using an HTML5 Canvas element. Additionally, the position of the weapon is also displayed using the same process. These resulting coordinates are stored in the same way as the previous position, except in each case one component will be removed since a simple two dimensional position is displayed.

The AEAS software relies heavily upon the use of standardized high-precision floating point values. Much of the data which is operated on exists in this form, and is crucial to ensuring the proper operation of the defined system. In all computations performed by the software, accuracy is the highest concern, so great effort has been expended to ensure that the values are appropriately formatted at each point in the system data path.

(2) Algorithms are another important component of the server software. To perform the trajectory computation from the available position and orientation data, the target areas of the room are established as planes in a coordinate system which models the room using the top-left corner as the origin, and uses only positive values to represent locations. The use of such a coordinate system eliminates the need to perform complicated translations from real-world coordinates to pixel coordinates on the display. This optimization reduces some of the computation required by the system, and thus does not introduce any additional latency.

Upon receiving and parsing the data from the attachment, the server software first converts the orientation data from yaw, pitch, and roll angles to a single three dimensional direction vector. The conversion is shown below.

$$x_{dir} = \cos(yaw) * \cos(pitch) \quad (12)$$
$$y_{dir} = \sin(yaw) * \cos(pitch) \quad (13)$$
$$z_{dir} = \sin(pitch) \quad (14)$$

Using the position point, the direction vector, and the target plane (defined by three points real-world points), it can be determined if there exists an intersection of the trajectory on the plane. If such an intersection exists, the point at which it occurs can be found through the use of linear algebra. The trajectory algorithm thus relies upon known planes to check for intersections. By modeling each wall of a room as a plane, the system can be easily extended to handle more complicated firefight scenarios.

Using the three points which define the plane, three vectors can be produced by subtracting one unique point from another unique point. The cross product of these two vectors then gives the normal vector to the plane on which the points lie. The equation of the plane is thus

$$N_x(x + P_{0x}) + N_y\left(y + P_{0y}\right) + N_z(z + P_{0z}) \quad (15)$$

where P0 represents a three dimensional point on the plane.

The x, y, and z parameters are found from the parameterization of the line between the weapon and the point of intersection on the plane (if it exists).

$$x = position_x + orientation_x * t \quad (16)$$
$$y = position_y + orientation_y * t \quad (17)$$
$$z = position_z + orienation_z * t \quad (18)$$

By rearranging the equation of the plane to solve for t after substitution, the x, y, z location of the intersection can thus be determined through reuse of the equations above.

For the situation where no solution exists, the result is often infinity, and is among the conditions checked before forwarding any resulting data to the visualization. In addition, checks are also implemented to verify that only hits and positions within the designated area are displayed.

(1) Visualization

Using the supplied and computed information, the current position and direction of the weapon can be displayed on a 2D plane using the HTML5 Canvas. The real world coordinates are scaled to pixel coordinates by simply normalizing each position with respect to the length, width, and height of the scenario area, and then scaling based on the display resolution entered for the Canvas. The end result simply displays either the weapon's location on the playing field from above or the point on a plane where a "bullet" has impacted if the visualization is being used as the target.

For example, to represent the location of a weapon from a top-down view only the x and y components of the position are required. The weapon visualization can then be expressed as

$$World\ coordinate = \left[position_x, position_y\right] \quad (19)$$

$$To\ pixel\ coordinate = \frac{position_x}{\frac{field\ length, position_z}{field\ width}} \quad (20)$$

$$Pixel\ coordinate = [normalizedx \times xpixels, normalizedy \times ypixels] \quad (21)$$

For the case that a fired shot that has intersected with a target, the result can be expressed by converting the point (positionx, positiony, positionz) to a set of pixel coordinates. In displaying a fired shot, the y component of the position is not necessary, and thus can be discarded for display on the two dimensional target.

$$World\ coordinate: [position_x, position_z] \quad (23)$$

$$To\ pixel\ coordinate: [position_x / field\ length, position_z / field\ height] \quad (24)$$

$$pixel\ coordinate: [normalizedx * xpixels, normalizedz * ypixels] \quad (25)$$

In this case, "xpixels" and "ypixels" represent the number of pixels measured horizontally or vertically on the visualization element. For the simulation area and visualization element, it is assumed that all lengths are equal, thus the simulation area can be modeled by a cube and the mentioned visualized quantities are simply projections onto a face of the cube. As with many other elements in the AEAS system, this restriction can be modified with relative ease to accommodate more complicated scenarios as desired.

VI. Conclusion

The integration of the weapon attachment, server, and custom ultrasonic positioning system provides a training capability that is potentially much more effective than current live simulations. This includes allowing soldiers to use their own weapons during training, and implementing a capability that allows more complete simulation data capture. This system also maintains that virtually all of the simulation could be controlled by software, which means easy modeling of various weapon types, damage assessment, ricochet, or any other realistic modeling that is desired by the trainers, to be limited only by what is achievable with current technology. The success and accuracy of the AEAS system will lay the groundwork for future research and development to eventually deploy the virtual bullet capability in future training simulations.

Acknowledgements

The authors wish to acknowledge the assistance and support of Dr. Richie as well as all of the instructors in the Department of Electrical Engineering and Computer Science. We also wish to acknowledge Jeff Rowberg for his incredibly useful MPU6050 Arduino libraries which are being used for orientation calculations.

References

[1] Bjerknes, J. D., Liu, W., Winfield, A. F., and Melhuish, C. (2007). "Low Cost Ultrasonic Positioning System for Mobile Robots." In Wilson, M.S., Labrosse, F., Nehmzow, U., Melhuish, C., and Witkowski, M., editors, *Proceeding of Towards Autonomous Robotic Systems*, pages 107 - 114, Aberystwyth, UK.

[2] Atmel Corporation. "8-bit AVR Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash". http://www.atmel.com/Images/doc8161.pdf. Rev. 8161D-AVR. Oct. 2009. Web. 24 Nov. 2013.

[3] Atmel Corporation. "AVR053: Calibration of the internal RC oscillator. " http://www.atmel.com/Images/doc2555.pdf. *8-bit AVR Microcontrollers Application Note*. Rev. 2555G-AVR. May, 2006. Web. 11 Nov. 2014

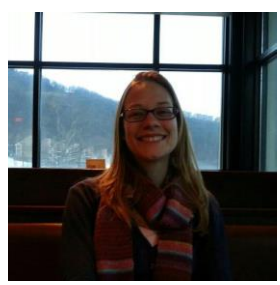
[4] Murata Electronics. "MA40S4R: Directivity in Sensitivity". http://search.murata.co.jp/Ceramy/CatalogshowpageAction.do?sParam=MA40S&sKno=G011&sTblid=A091&sDirnm=A09X&sFilnm=MAS4G03&sType=1&sLang=en&sNHinnm=MA40S4R&sCapt=Directivity_in_Sensitivity 2013. Web. 9 Nov. 2013.

[5] CH Robotics. "Estimating Velocity and Position Using Accelerometers" http://www.pololu.com/file/0J587/AN-1007-EstimatingVelocityAndPositionUsingAccelerometers.pdf. AN-1007 rev. 1.0. 19 Oct, 2012. Web. 11 Nov. 2014.

Biographies

Alex Balogh is a 22 year old Computer Engineering student. After gaining some experience in software engineering, Alex hopes to pursue a career as a software or hardware development engineer at a company such as Intel or NVIDIA focused on the development of processors and GPUs.

Ryan Sivek is a 25 year old Computer Engineering student with a minor in Psychology who is interested in pursuing his Master's degree in Biomedical Engineering. He currently works at the MITRE Corporation as a Computer Science Associate performing various tasks on a number of different projects. In particular he has worked on the Squad Overmatch Project which is surveying technological and operational solutions for more fidelity and stress exposure to enhance Army training. This project is what prompted him to pursue the AEAS system for his senior design project.



Karena Stout is a 23 year old Electrical Engineer student. She is a member of the Society of Women Engineers and the Association of Professional Women. She works as a CWEP intern at Lockheed Martin designing FPGAs and plans to pursue this path after college. She will continue to work for Lockheed in Orlando as an FPGE designer with hopes to transferring within the company to the Norfolk, Virginia area where here fiancé is stationed in the Navy.