OPTICAL ENGINEERING SIMULATOR DEVELOPMENT FOR EFFICIENT DESIGN OF MILES GEAR

Seon Han Choi[(a)] , Changbeom Choi[(b)], Tag Gon Kim[(c)]

[(a),(c)]Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology
[(b)]Sch. of Creative Convergence Technology, Handong Global University

[(a)]gigohan01@kaist.ac.kr, [(b)]cbchoi@handong.edu, [(c)]tkim0303@kaist.ac.kr

ABSTRACT
Most of live military training systems are based on the MILES gear. In order to simulate the real engagements, it simulates a gunfire using a laser beam, and death by being shot is judged by sensors, which are attached on the trainer's body. Therefore, one of the important design consideration is to choose proper specifications of MILES gears to embrace the properties of real firearm. To decide the specification of MILES gear, the designer should decide several things. The problem is that conducting this with real experiments needs a lot of time and cost because of too many combinations of specifications. This paper suggests an optical engineering simulator to compute an efficient design of MILES gear. The simulator is based on the domain knowledge of the laser beam and the sensors to acquire high fidelity results, so that the designer of MILES gear may find the proper specifications easier.

Keywords: MILES gear, Modeling and Simulation, Simulation Based Acquisition

1. INTRODUCTION

Multiple Integrated Laser Engagement System (MILES) is widely used in about 32 national militaries to conduct live military training.(Wikipedia 2015) In order to give immersive experience to the trainees, MILES gear simulates real engagements with a laser beam and multiple laser sensors. The laser beam describes a bullet with a beam width, and it is emitted from the transmitter mounted on a real firearm. The diameter of beam region is decided by the beam width, and all the sensors are activated in the beam region. The sensor attached on a body detects the laser beam and decides whether the bullet hits the body or not (Jones, Huang, and Bian 2008).

To increase the effectiveness of training, several properties of MILES gear should be the identical as that of a real firearm. Among the properties, a hit rate is one of the important property. The hit rate means chance to hit (or detected by sensors) a target when you fire an aimed shot to it. In case of a real firearm, the hit rate is close to 100 percent, because all area of trainee's body is effective area for real bullets. However, it is hard that the hit rate of MILES gear achieves 100 percent because of the beam width, the limited number of sensors, and restricted location of attaching sensors. If the designer wants to make the hit rate of MILES gear to be 100 percent, the beam width should be converge to zero without decreasing the range of the beam. On the other hand, the number of sensors should be infinite to cover entire area of the trainee's body.

Reducing the beam width without decreasing its range can be achieved by increasing the initial energy to generate the beam, and, unfortunately, it is dangerous to the trainees. Whereas, the large number of sensors may restrict the movement of the trainees. If the designer decides to attach the large number of sensors to the trainees, the sensors should be small enough so that the trainee can attend training without any restriction. However, it leads to a budget problem. Therefore, the designer of the MILES gear should comprise aforementioned considerations to choose proper specifications of MILES gear to simulate real battle using the MILES gear.

Among several design considerations of the MILES gear, the important design considerations of the MILES gear are 1) the beam width; 2) the number of sensors; and 3) the location of the sensors. The designer should consider the combinations of the design considerations. However, combinations of specifications are too many so that the designer cannot test each combination of the consideration in real environment. To tackle the problem, this paper proposes an optical engineering simulator to decide proper specification of the MILES gear. It is based on the domain knowledge of a laser beam and sensors and it models physical characteristics of the laser beam and the sensors to get high fidelity results. As a result, simulator helps decision makers and designers to find proper specifications of MILES gears.

This paper is organized as follows. Section 2 illustrates the differences and the similarities between real firearm and the optical engineering models of the proposed simulator such as the hit rate. In Section 3, we introduce mathematical definitions of the component models for MILES gear, and in Section 4, we proposes the optical engineering simulator using these models. Several simulation results are showed in Section 5 and finally, we conclude in Section 6.

2. COMPARASION WITH REAL FIREARM

As we mentioned in the Section 1, the hit rate of MILES gear should be the identical as that of a real firearm to increase an effectiveness of training. When the number of sensors is limited, increasing the beam width makes the hit rate similar to a real firearm. However, MILES gear has the additional hit rate called a near-hit rate caused by the beam width.

To define the near-hit rate, Figure 1 classifies the beam according to relationships about beam, sensors, and target. The beam is classified into 3 groups according to relationship between beam and target: *not-hit*, *hit*, *near-hit*. The *not-hit* means the beam does not overlap with the target. In case of overlapping, the beam is classified into 2 groups according to the location of center of beam. If the center is on the target, the beam is classified as *hit*. Otherwise, it is classified as *near-hit*.

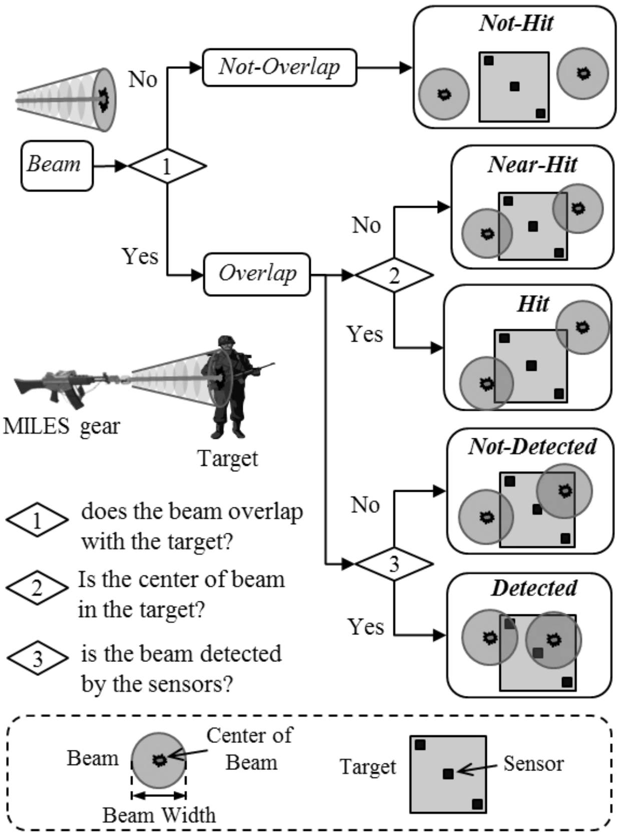


Figure 1: Classification of Beam

For the *hit* and *near-hit* beams, they are classified into 2 groups according to relationship between beam and sensors: *detected*, and *not-detected*. If sensors can detect the beam, it is categorized as *detected*. Otherwise, it is categorized as *not-detected*. The hit rate and near-hit rate is the rate of the *detected* beams among the *hit* beams and the *near-hit* beams. Using these classification, the hit rate(P_H) and near-hit rate(P_N) are defined as below.

$$P_H = \frac{n(Hit \cap Detected)}{n(Hit)} \quad (1)$$

$$P_N = \frac{n(Near\text{-}Hit \cap Detected)}{n(Near\text{-}Hit)} \quad (2)$$

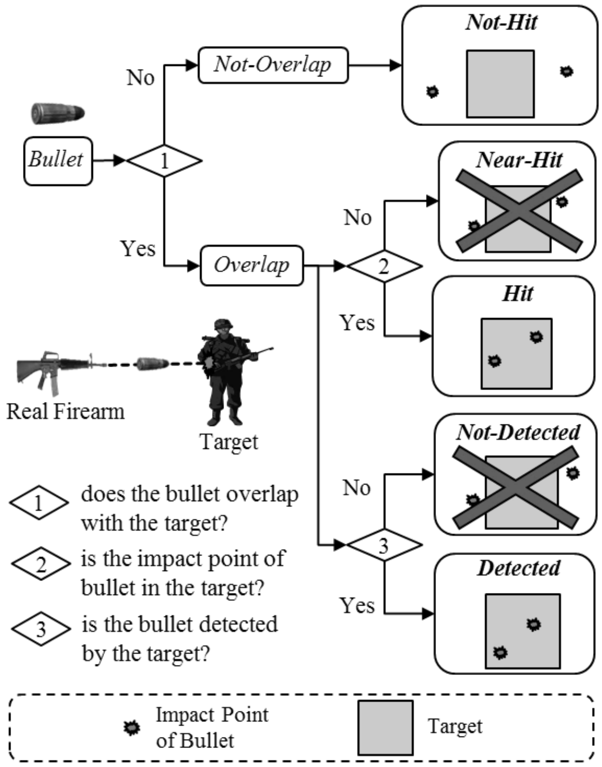


Figure 2: Classification of Bullet

Comparing MILES gear with a real firearm, a bullet is not classified as *near-hit* because of the size of bullet. The size of bullet is almost 0 compared to the beam width. (i.e. center of beam is considered as the impact point of bullet.) Also, since all of the parts of the target can detect the bullet, *not-detected* cannot be occurred.(see Figure 2) Therefore, the hit rate of a real fire arm is 100 percent without any side-effects such as wind, aiming error, gravity, and so on. Also, the near-hit rate is 0 percent because of no bullet classified as *near-hit*. However, the hit rate of MILES gear is below 100 percent and the near-hit rate of it is over 0 percent because of the beam width and the limited number of sensors. Figure 3 shows this using a Venn diagram.

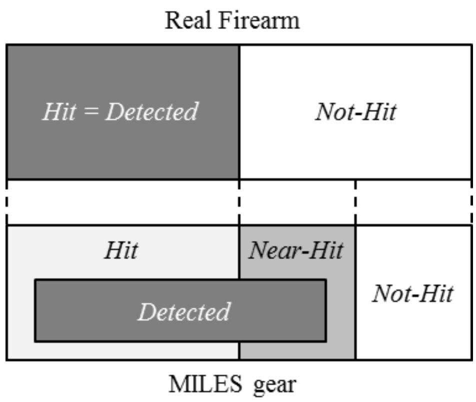


Figure 3: Comparison with Real Firearm

To design MILES gear similar to a real firearm, the hit rate should go up to 100 percent and the near-hit rate should go down 0 percent. To achieve the hit rate as 100 percent, increasing the beam width or the number of sensors is a simple way. However, this leads to rising the near-hit rate. On the contrary to this, decreasing the beam width or the number of sensors to achieve the near-hit as 0 percent, lowers the hit rate. Ideally, achieving the hit rate and near-hit rate to 100 and 0 percent are accomplished by reducing the beam width to almost 0 and increasing the number of sensors on the target infinitely. However, it is impossible because of the several reasons as mentioned in Section 1. Therefore, when you designs MILES gear, chooses proper specifications to maximize the hit rate and minimize the near-hit rate. The following sections will explain the optical engineering simulator for doing this easily.

3. COMPONENTS MODELING

This section will explain about 3 component models of MILES gear such as laser beam, sensor, and target for calculating the hit rate and near-hit rate.

3.1. Laser Beam Model

3.1.1. General Gaussian Beam Model

Gaussian Beam Model is a general model for describing a laser beam.(Quimby 2006) Mathematical representation of the model and parameter descriptions are below.(see Figure 4)

$$I(r,z) = I_0 \left(\frac{w_0}{w(z)}\right)^2 \exp\left(-\frac{2r^2}{w^2(z)}\right) \quad (3)$$

$$w(z) = w_0\sqrt{1+(z/z_R)^2} \cong \sqrt{w_0^2 + \theta_0^2 z^2} \quad (4)$$

- $I(r,z)$[W/m^2]: intensity of beam
- z[m]: distance from the initial point of beam to the center of beam
- r[m]: radial distance from the center of beam
- θ_0[rad]: divergence angle of beam($\cong w_0/z_R$)
- w_0[m]: waist of beam
- z_R[m]: Rayleigh range

The intensity of beam$\left(I(r,z)\right)$ following Gaussian is diffused as the distance(z) increases. The points apart from the center of beam as the same radial distance(r) have equal intensity and that makes cross section of beam circular. The center of beam means an intersection point between the center axis of beam and a plane. (The plane is one of the faces that construct a target.) Using the representation, we define the beam width (R) mathematically. The beam width is a diameter of area(circle) where the intensity of beam is above the intensity threshold of sensor(D_L). In other words, sensors which are in the area can be activated by the beam. Mathematical representation of the beam width is below.

$$R(z) = 2\sqrt{-\frac{w^2(z)}{2}\, ln\left(\frac{D_L}{I(0,z)}\right)} \quad (5)$$

- D_L[W/m^2]: intensity threshold of sensor

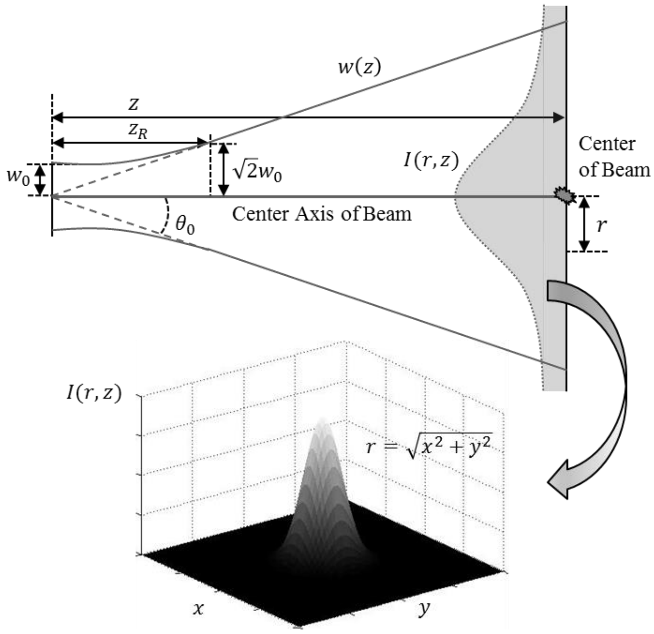


Figure 4: Gaussian Beam Model Parameters & 3D Plot of Intensity$\left(I(r,z)\right)$

The beam width is decided by the distance and the intensity threshold. Figure 5 shows the beam width according to the distance. It increases as the distance increases until at a certain point, and decreases after the point. If the distance is over the maximum distance(z_{max}), the beam width becomes 0. Therefore, when designing MILES gear to simulate a real firearm, deciding the maximum distance with consideration for the maximum range of it is important.

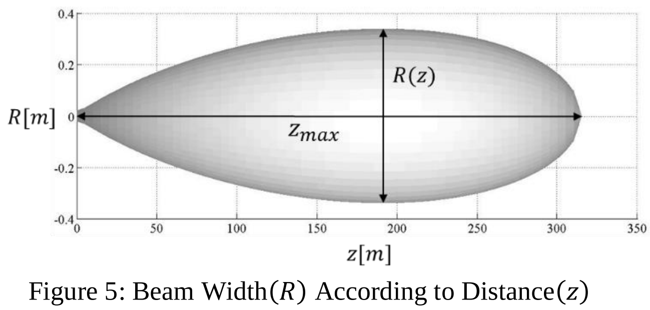


Figure 5: Beam Width(R) According to Distance(z)

3.1.2. Extended Gaussian Beam Model

In real cases, the beam does not come at a right angle to a target (i.e. the incidence angle(θ_I) is 0°). However, the general model does not deal an incidence angle. To increase fidelity of the simulator, this paper proposes an Extended Gaussian Beam Model including the incidence angle. To simple calculation and expression, the paper puts an assumption that all of the axes constructing the beam have the same incidence angle. (i.e. all of the axes are parallel each other.) The extended mathematical representation of the model and additional parameter descriptions are below.

$$I(x,y,z,\theta_I) = I_0\left(\frac{w_0}{w(z)}\right)^2 \cos\theta_I \exp\left(-\frac{2x^2+2y^2\cos^2\theta_I}{w^2(z)}\right) \qquad (6)$$

- θ_I[rad]: incidence angle
- (x,y): coordinate from the center of beam

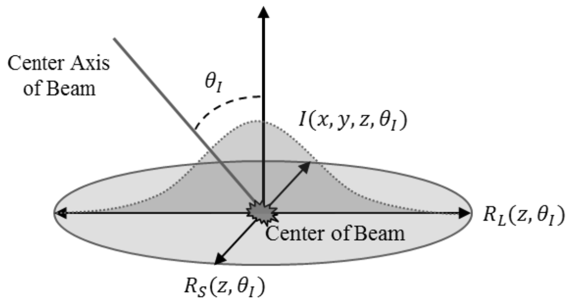


Figure 6: Extended Gaussian Beam Model

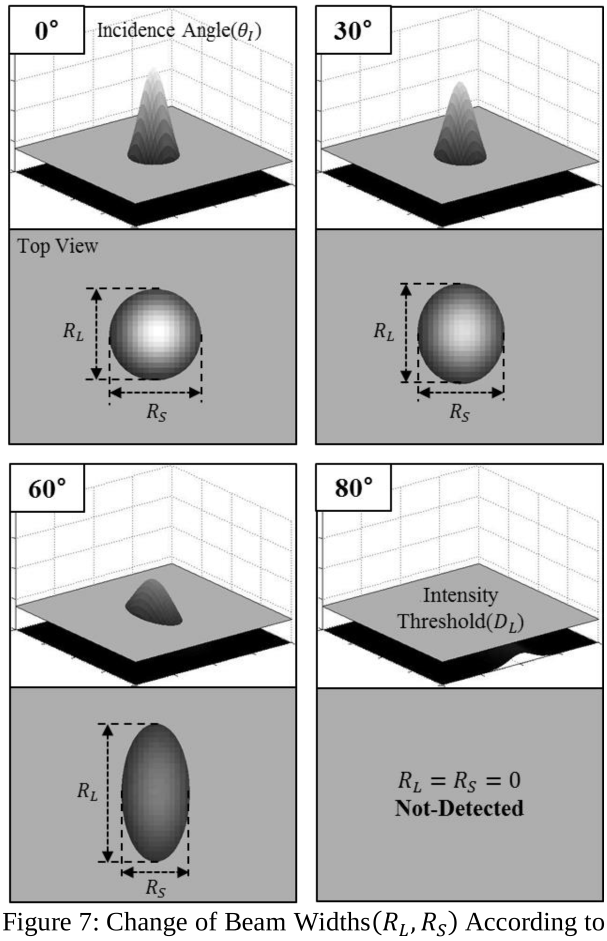


Figure 7: Change of Beam Widths(R_L, R_S) According to Incidence Angel(θ_I)

Because the intensity is diffused to the incidence direction, the intensity of points apart from the center of beam as the same radial distance are not equal. That means the cross section of the beam transforms from a circle to an ellipse. The radial distance is replaced by the coordinate from the center of beam and the beam width is also divided into 2 different parts: long beam width (R_L) and short beam width (R_S).(see Figure 6) They are respectively twice the major axis and the minor axis of the ellipse. Mathematical representation of these are below.

$$R_L(z,\theta_I) = 2\sqrt{-\frac{w^2(z)}{2\cos^2\theta_I} ln\left(\frac{D_L}{I(0,z)\cos\theta_I}\right)} \qquad (7)$$

$$R_S(z,\theta_I) = 2\sqrt{-\frac{w^2(z)}{2} ln\left(\frac{D_L}{I(0,z)\cos\theta_I}\right)} \qquad (8)$$

Based on the representation (7) and (8), we define the limit incidence angle(θ_L) that means the long width and the short width remain 0 after the angle. If the incidence angle is more than the limited angle, the intensity is diffused so widely that no points are over the intensity threshold of sensor.(see Figure 7) The limit incidence angle decreases as the distance increases. Therefore, when designing a sensor, take care of the limited angle and decide the parameters of the sensor. Mathematical representation of the limited angle is below.

$$\theta_L(z) = \cos^{-1}\left(\frac{D_L}{I_0(w_0/w(z))^2}\right) \qquad (9)$$

In the simulator, a beam consists of a coordinate which is on a virtual sphere, and a vector that indicate direction of the beam. Size of the vector is the same as the distance of beam. The simulator generates and classifies the beam using the Extended Gaussian Beam Model. Section 4 will explain this in detail.

3.2. Sensor Model

A sensor model is quite simple. In the simulator, it is abstracted to a point because its size is very small as compared to the beam width. The sensor model has 2 parameters: intensity threshold(D_L), angle threshold(θ_D). As mentioned previously, the intensity threshold of sensor is used to define the beam width. The angle threshold of sensor, different with the limited incidence angle (θ_L) mentioned in the last section, is a limit incidence angle in terms of sensor. That means the sensor can detect the beam whose incidence angle is under the minimum between the limited incidence angle(θ_L) and angle threshold(θ_D). Therefore, the sensor detects the beam whose intensity on the sensor is over the intensity threshold and incidence angle is under the angle threshold. Simply put, the sensor where is in the ellipse of the beam, is activated.

In the simulator, a sensor consists of a location coordinate on s target and a vector that indicates direction of the sensor. The vector is equal to the normal vector of the face where the sensor is on. Using the mentioned sensor model, the simulator decides whether the sensor detects a beam or not.

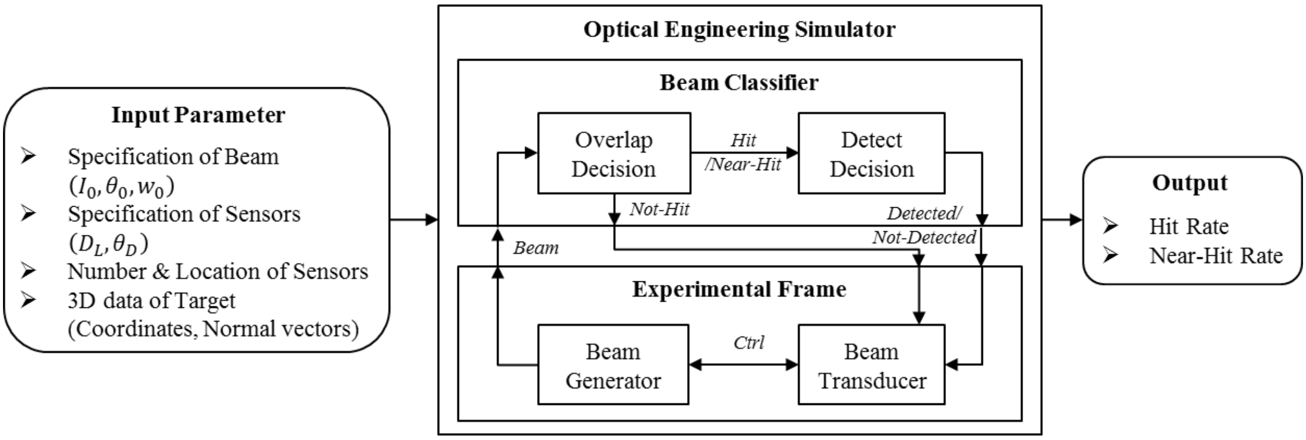


Figure 8: Structure of Optical Engineering Simulator

3.3. Target Model

A target model is a 3D mesh model. It consists of coordinates of the faces constructing the target and normal vectors whose direction is outside of the faces. The sensors can be on the faces. The simulator calculates the center of beam, the incidence angle and the beam widths about all of the faces to check whether the beam overlaps with the faces. The following section will explain this in detail.

4. OPTICAL ENGINEERING SIMULATOR

The optical engineering simulator calculates the hit rate and near-hit rate of MILES gear in given input parameters. The parameters are about the beam model, the sensor model and the target model mentioned in the previous section. Figure 8 shows the structure of the optical engineering simulator. The simulator consists of 2 main models which are the Experimental Frame and the Beam Classifier.

4.1.1. Experimental Frame

The experimental frame based on Zeigler, Praehofer, and Kim (2000) generates beams and analysis them. It has 2 sub models which are Beam Generator and Beam Transducer. The beam generator generates beams to the beam classifier and the beam transducer calculates the rates from the classification result of the classifier.

4.1.2. Beam Generator

Because the number of *hit* and *near-hit* beams of a target is infinite, the beam generator generates beams using the Monte-Carlo simulation method.(Mooney 1997) In other words, the generator chooses the finite number of beams from a set of all *hit* and *near-hit* beams, and generates them to calculate the hit and the near-hit rate. To get the accurate rates, the set should include all of *hit* and *near-hit* beams. For this, the paper proposed a concept of virtual sphere.

The virtual sphere surrounds a target which is on the center of the sphere. If the size of the virtual sphere is enough large, all of the *hit* and *near-hit* beams of the target have the center of beam on the inside of the sphere. Inversely, to simulate all of these beams the generator generates beams whose center of beam are on the inside of the sphere. Therefore, the generator generates a beam using 4 random parameters: ①center of beam, ②theta of beam(θ_{Beam}), ③phi of beam(ϕ_{Beam}), ④distance of beam(z_{Beam}).(see Figure 9) The generated vector of the beam(V_{Beam}) is below. (The subscript '$_{Beam}$' is omitted.)

$$V_{Beam} = -(z \cdot sin\theta cos\phi, z \cdot sin\theta sin\phi, zcos\theta) \qquad (10)$$

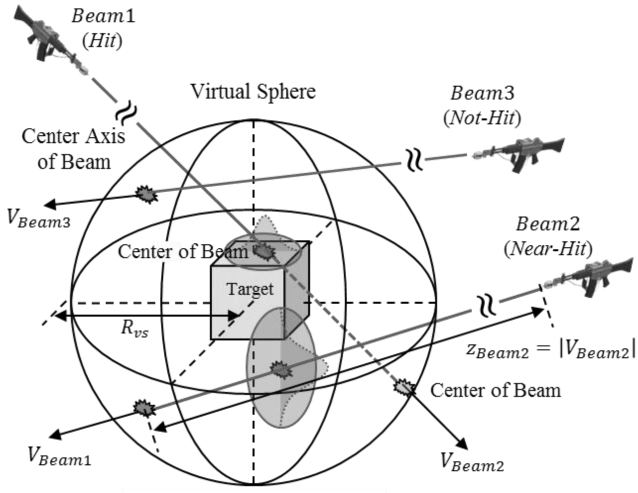


Figure 9: Virtual Sphere

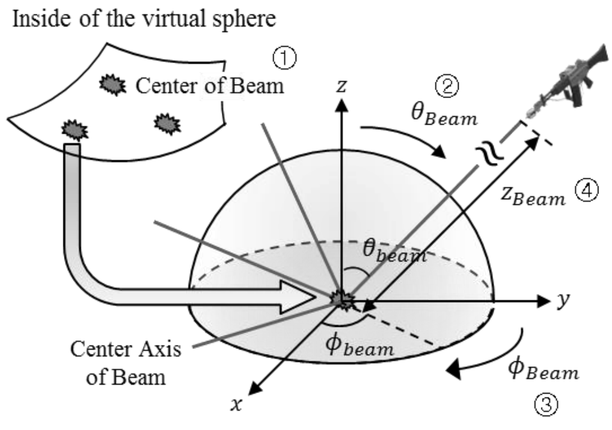


Figure 10: Generating Parameters of Beam

As mentioned previously, the radius of virtual sphere should be enough long to include all of the *hit* and *near-hit* beams of the target. If the radius is too long, the sphere include excessive *not-hit* beam additionally. That makes the simulator generate more beams to get the results and reduces the performance of the simulator. Otherwise, if the radius is too small, the sphere cannot include all of the *hit* and *near-hit* beams. That makes the simulator draw out wrong results. Therefore, the appropriate radius is the minimum value among radiuses including these beams. The representation of appropriate radius is below. $Distance$ means a distance from the center of the target (usually $(0,0,0)$) to a point on the target.

$$R_{vs} = \max(Distance) + \max\big(R_L(z, \theta_I)\big) \qquad (11)$$

4.1.3. Beam Transducer

The beam transducer collects classification result from the beam classifier and calculates the hit rate and near-hit rate using the equations (1) and (2). Also using *Ctrl*, it controls the generator such as stop and go.

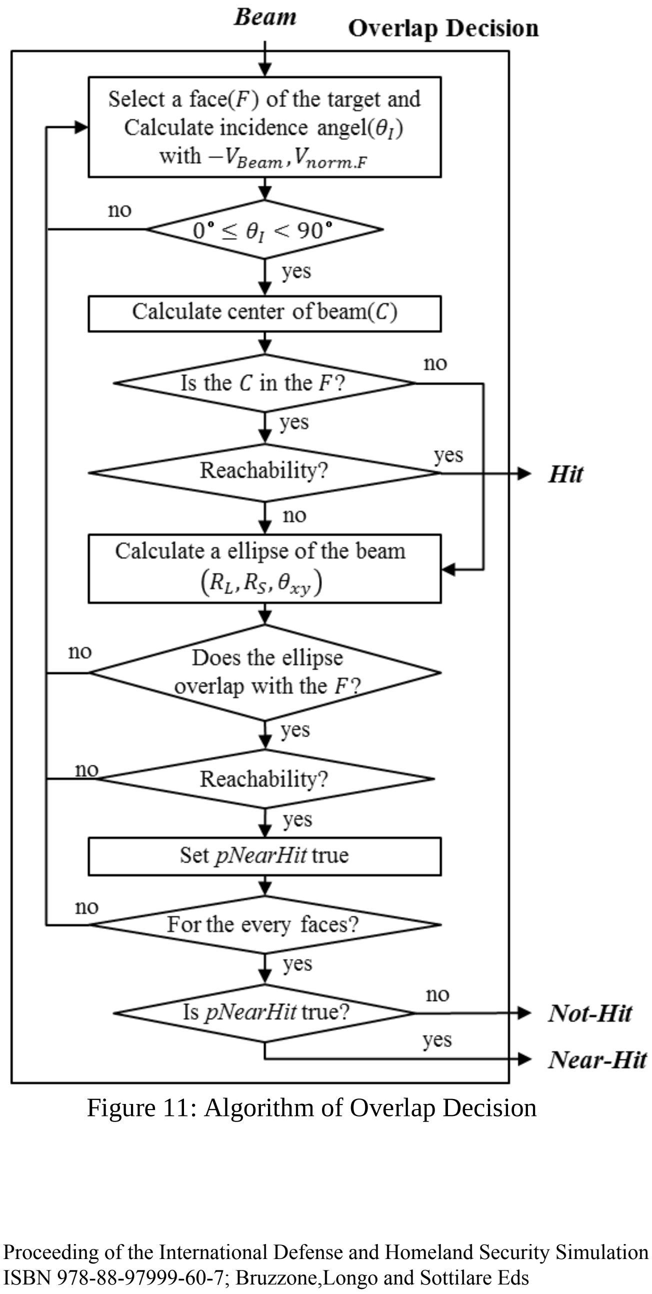


Figure 11: Algorithm of Overlap Decision

4.2. Beam Classifier

The beam classifier categorizes a beam according to the mentioned classification.(see Figure 1) It has 2 sub models witch are Overlap Decision and Detect Decision. The overlap decision decides whether a beam overlaps with a target and classifies the beam into 3 groups: *Not-Hit*, *Hit* and *Near-Hit*. For the beam classified as Hit or *Near-Hit*, the detection decision decides whether the beam is detected by sensors on a target and classifies the beam into 2 groups: *Detected* and *Not-Detected*.

4.2.1. Overlap Decision

The overlap decision is an algorithm model to classify a beam into 3 groups: *Not-Hit*, *Hit* and *Near-Hit*. Figure 11 shows the algorithm of the model and Figure 12 shows graphical representation of the algorithm. When a beam as an input enters to the overlap decision, it chooses one of the faces of a target and calculates an incidence angle between vector of the beam and normal vector of the face. If the incidence angle is between 0° and 90°, the beam can arrive at the face. Then it calculates the center of the beam which means the intersection point between the center axis of beam and the plane including the face. If the center of beam is in the face, then it checks reachability of the beam. The reachability means the beam can reach at the face without any interruption of the other faces. When the beam has reachability, the overlap decision categorizes the beam as *hit* and makes output.

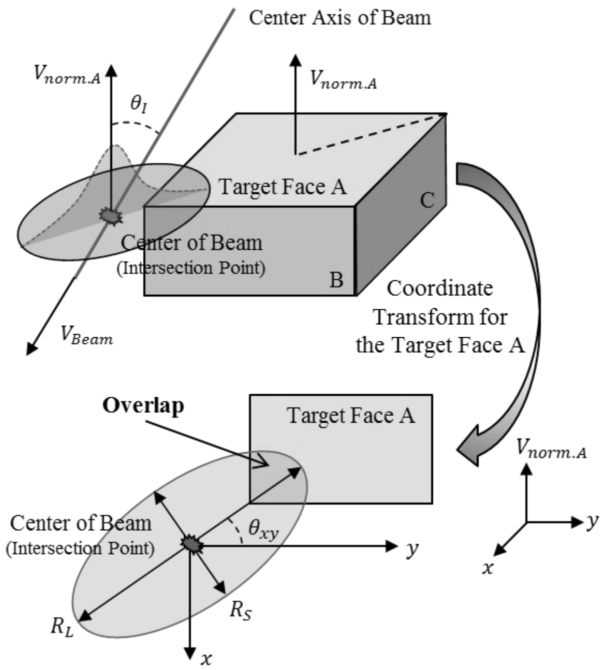


Figure 12: Graphical Representation of Overlap Decision

Otherwise, it calculates the ellipse of the beam (R_L, R_S, θ_{xy}) and decides whether the ellipse overlaps with the face. In case of overlapping, the overlap decision checks reachability of the beam. When the beam has reachability, the overlap decision sets *pNearHit* true. Because the beam can be classified into *hit* for the other faces, it cannot classify the beam into *near-hit* before checking all of the faces. Therefore, if the beam does not classified into *hit* until checking all of the faces and *pNearHit* is true, then the overlap decision classifies the beam into *near-hit* beam and makes output. Otherwise, if *pNearHit* is false, then it classifies the beam into *not-hit* and makes output.

4.2.2. Detect Decision

The detect decision is also an algorithm model to decide whether the *hit* or *near-hit* beam is detected by sensors. Figure 13 shows the algorithm of the model. The detect decision is similar to the overlap decision but more simple. When a *hit* or *near-hit* beam as an input enters to the detect decision, it chooses one of the sensors on a target and calculates an incidence angle between vector of the beam and vector of the sensor. The vector of sensor is the same as the vector of the face where the sensor is on. If the incidence angle is between 0° and the angle threshold(θ_D), it can arrive at the sensor and activate that with the intensity over the intensity threshold(D_L).

To check whether the intensity is over the threshold, the detect decision calculates the coordinate of the sensor (x, y) and the intensity $\big(I(x, y, z, \theta_I)\big)$ on that. When the intensity is over the threshold, it checks reachability of the beam to the sensor. In case the beam has reachability, the detect decision classifies the beam as *detected* and makes output. Otherwise, it chooses another sensor and checks that sensor again using the algorithm. If all of the sensors on a target does not detect the beam, the beam is classified into *not-detected*.

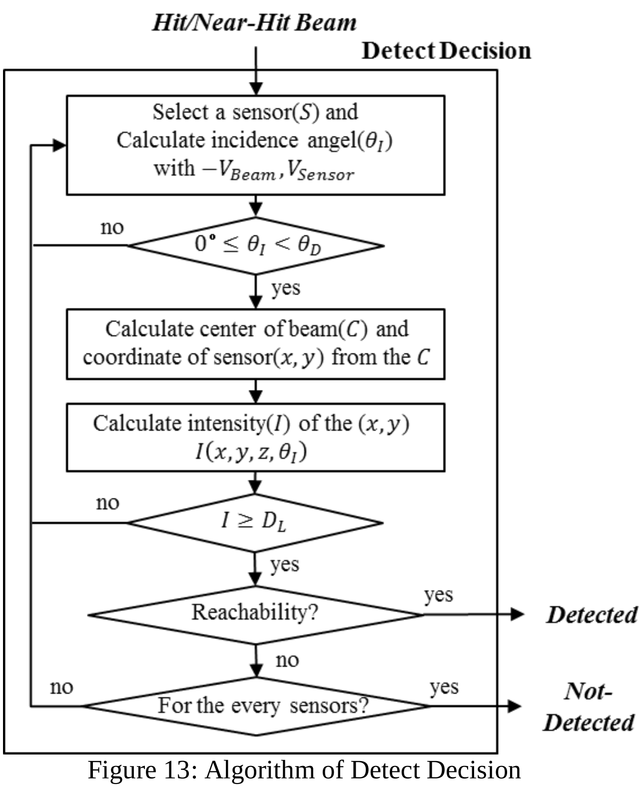


Figure 13: Algorithm of Detect Decision

5. SIMULATION RESULT

This Section shows how much effective the simulator is to design MILES gear. Using the simulator, it is easy to calculating the hit rate and near-hit rate for various input parameters without real experiments. Also, designers can get some useful insights of MILES gear from the results. Figure 14 presents a simple result of the simulator which is the hit rate and near-hit rate according to the distance of beam(z). The parameters of beam and the intensity threshold of sensor are set to achieve the beam width 60cm at the distance 250m and the maximum distance 350m.(the beam simulates K-1 rifle of ROK army) The angle threshold of sensor is set to 90°. The target is an infantry who has 6 sensor modules: 2 modules are on the head, another 2 modules are on the front body and the rest is on the rear body.(see Figure 14)

Meanwhile, the sensor module is a kind of cubic-shaped module which has 2 sensors in each faces except the attaching face. The module can detect more beams than a single sensor because it is virtually unaffected by the angle threshold (θ_D) and the limited incidence angle(θ_L). For example, the single sensor cannot detect a beam whose incidence angle is 80°, because of the limited incidence angle. However, the module can detect the beam because incidence angle between the beam and the sensors on the side face of the module is just 10°. Therefore the beam can be detected by the module. (Actually the sensors on the side face of the module detect the beam.)

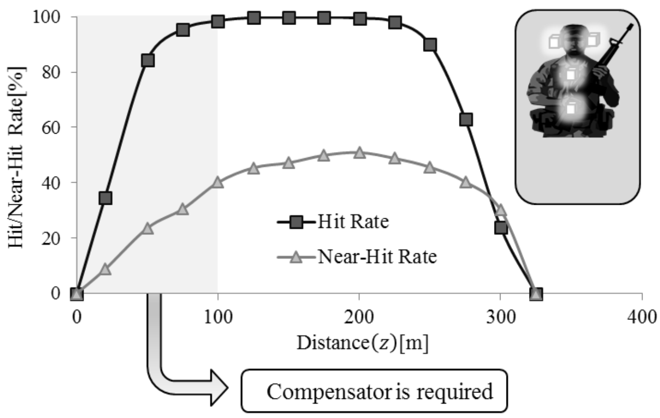


Figure 14: Hit/Near-Hit Rate According to Distance of Beam

The hit rate and near-hit rate increase as the distance increases until at 100m, then the hit rate keeps 100 percent and the near-hit rate rises slowly. After about 200m, they decrease rapidly and go to 0 percent at roughly 330m which is approximation of the maximum distance 350m. The shape of the graph is similar to that of the beam width, because the hit rate and near-hit rate are greatly affected by the beam width.(see Figure 5) This gives a useful insight of MILES gear which is a requirement of compensator. In case of a real firearm, the hit rate at close range is 100 percent even though considering side-effects such as wind, aiming error, and so on. However, because the beam width at close range is too small to activate sensors, the hit rate is low at the close range. Therefore, an additional beam is required as the compensator to increase the hit rate at the close range.

Table 1: Hit/Near-Hit Rate According to Target Model

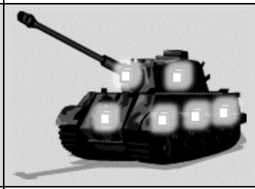
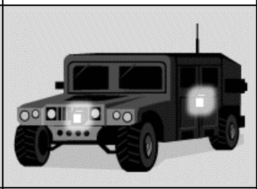
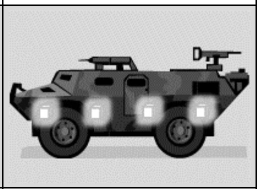
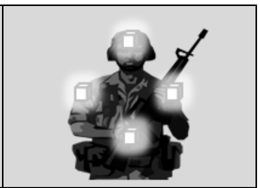
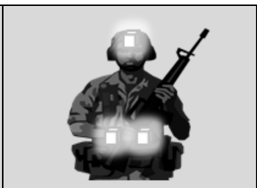
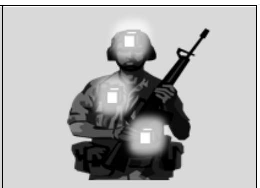
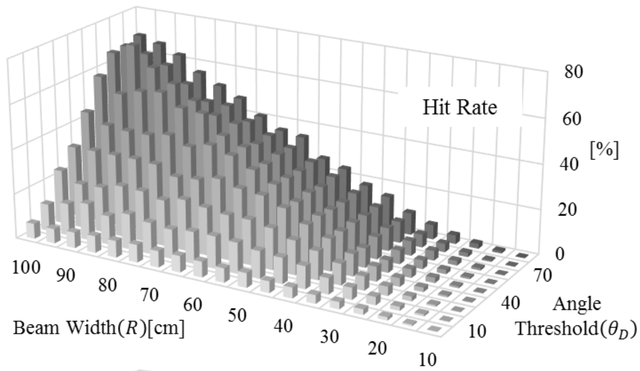
Target Model	**Infantry**	**Tank(K-1)**
Loc. of Sensors		
# of Sensors	6 modules	12 modules
Hit Rate	89.7% at 250m	50.3% at 250m
Near-Hit Rate	46.1% at 250m	16.6% at 250m
Target Model	**Vehicle(K-111)**	**Vehicle2(K200)**
Loc. of Sensors		
# of Sensors	4 modules	6 modules
Hit Rate	62.0% at 250m	46.1% at 250m
Near-Hit Rate	20.1% at 250m	13.1% at 250m

Table 2: Hit/Near-Hit Rate According to Location of Sensors

Loc. of Sensors		
# of Sensors	6 modules	6 modules
Hit Rate	89.7% at 250m	99.2% at 250m
Near-Hit Rate	46.1% at 250m	52.4% at 250m
Loc. of Sensors		
# of Sensors	6 modules	6 modules
Hit Rate	84.0% at 250m	89.3% at 250m
Near-Hit Rate	46.6% at 250m	47.9% at 250m

Besides getting some insights, the simulator can give results for various input parameters. Table 1 shows that the hit rate and near-hit rate according to various target models. Except the case of infantry, the parameters of beam and the intensity threshold of sensor are set to achieve the beam width 150cm at the distance 250m and the maximum distance 350m.(the beam simulates M72LAW antitank weapon of ROK army) The angle threshold of sensor is set to 60°. The parameters of infantry cases are the same as Figure 14's one. Each target models have several sensor modules and the location of modules is in Table 1. All of the hit rates and the near-hit rates of targets are calculated at the distance 250m.

Table 2 shows that the hit rate and near-hit rate according to the location of sensors. The parameters of beam and sensor are the same as Figure 14's one. Each infantry targets have 6 modules attached at different locations. All of the hit rates and the near-hit rates of targets are calculated at the distance 250m.



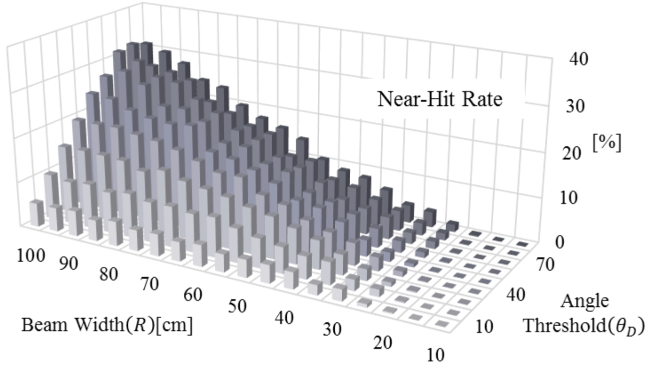


Figure 15: Hit/Near-Hit Rate According to Beam Width and Angle Threshold

Figure 15 shows that the hit rate and near-hit rate according to the beam width and the angle threshold. The number and location of sensors are the same as Figure 14's one except that the modules are replaced with single sensors. Because of this, the hit rate is lower than Figure 14's one in the same condition. Among the parameters of beam model, the initial intensity of beam(I_0) is changed to achieve the beam width and the others are invariable. The beam width and the angle threshold are increased from 10cm to 100cm and from 0° to 90° respectively. All of the hit rates and near-hit rates of targets are calculated at the distance 250m. Figure 15 demonstrates that the hit rate and near-hit rate are increased as these parameters are increased.

Like all of these results, the simulator can give the hit rate and near-hit rate for various input parameters, and sometimes it can give useful insights of MILES gear. Designers can find the proper specifications of MILES gear efficiently using the simulator, and that reduces the time and cost for real experiments. In addition, if some constraints are given, designers can find the optimal combination of parameters using the simulator and optimization methods: simulated annealing, genetic algorithm, and so on.(Gosavi 2014)

6. CONCLUSION

This paper proposes the Optical Engineering Simulator for an efficient design of MILES gear. The simulator calculates the hit rate and near-hit rate in given parameters of component models: beam model, sensor model, and target model. The beam model and sensor model calculates the beam width based on the Extended Gaussian Beam Model which includes the incidence angle. The target model is a 3D mesh model and has the sensor models on the faces of it. Using these component models, the simulator makes the output through the Experimental Frame and Beam Classifier. The experimental frame generates beams to the classifier and analyzes the classification results. The beam classifier is an algorithm model for categorizing the beam. The simulator reduces the time and cost for real experiments and make it easier to choose proper specifications of MILES gear.

There are 2 future works of the simulator. The first work is to expand the simulator, to deal with a moving target. In an actual training process, all of the soldiers are moving continually. However, the current simulator can deal with only a stalled target. The results from this simulator is difficult to apply it to an actual training process. The works will allow that the simulator is used more practically. The second works is to add MILES Communication Code(MCC).(U.S. Army's PEO-STRI 2011) MCC is a kind of signal that is transmitted by a laser beam, and used to assess damage of a target. Considering the code will increase the capability of the simulator.

ACKNOWLEDGMENTS

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract UD140022PD, Korea.

REFERENCES

Wikipedia, 2015. Multiple Integrated Laser Engagement System. Wikipedia. Available from: https://en.wikipedia.org/wiki/Multiple_Integrated_Laser_Engagement_System [Accessed 29 June 2015]

Jones G.D., Huang B.C., Bian Q., et al., 2008. Simulated Firearm Having A Multiple Integrated Laser Engagement System. U.S. Patent Application 12/024,693.

Quimby, R.S. 2006. Photonics and lasers: an introduction. Hoboken, New Jersey:John Wiley & Sons.

Zeigler B.P., Praehofer H., Kim T.G., 2000. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems. San Diego, CA:Academic press.

Mooney C.Z., 1997. Monte carlo simulation. Thousand Oaks, CA:Saga Publications.

Gosavi A., 2014. Simulation-based optimization: parametric optimization techniques and reinforcement learning. New York:Springer.

U.S. Army's PEO-STRI, 2011. MILES Communication Code (MCC), PMT 90-S002M standard. U.S. Army's PEO-STRI. Available from: http://www.peostri.army.mil/PRODUCTS/MCC/ECP_FILES/MCC_Standard_PMT_90-S002M-8Feb2011.pdf. [Accessed 8 Feb 2011]