UNIVERSITY OF BELGRADE

FACULTY OF MECHANICAL ENGINEERING

Saif AlAmeri

# Missile Guidance Navigation and Control Algorithms Design Using Machine Learning

Ph.D. Dissertation

Belgrade, 2019.

УНИВЕРЗИТЕТ У БЕОГРАДУ

МАШИНСКИ ФАКУЛТЕТ БЕОГРАД

Saif AlAmeri

# Синтеза алгоритама навигације и вођења пројектила заснованих на машинском учењу

Докторска дисертација

Београд, 2019.

Подаци о ментору и члановима комисије

Ментор: др Драган Лазић редовни професор, Универзитет у Београду, Машински факултет

Комисија:

др Милан Ристановић, ванредни професор, Универзитет у Београду, Машински факултет

др Ивана Тодић, доцент, Универзитет у Београду, Машински факултет

др Марко Милош, редовни професор, Универзитет у Београду, Машински факултет

др Жарко Ћојбашић, редовни професор, Универзитет у Нишу, Машински факултет

Датум одбране:

# Missile Guidance Navigation and Control Algorithms Design Using Machine Learning

Summary:

This thesis discusses the use of machine learning to design guidance, navigation, and control algorithms as an alternative to traditional algorithms for a missile system. The machine learning algorithm used in this thesis is the neural network. It is trained using the Neuro Evolution of Augmenting Topologies algorithm. Furthermore, the missile system and its environment have been modeled in order to simulate and compare the missile performances. The terminal guidance neural network will be compared to the proportional navigation algorithm. In addition, the neural network GPS/INS integration will be compared to the Kalman filter GPS/INS integration. Moreover, the neural network roll, pitch, and yaw autopilots will be compared to the traditional PID roll, pitch, and yaw autopilots. The goal of this thesis is to design neural network guidance, navigation, and control solutions which is expected to perform similar or better than their traditional counterparts. Thereby, the viability of the neural network designs as a guidance, navigation, or control solution will be verified.

Keywords: Missile, Guidance, Navigation, Control, Design, Machine Learning, PID, INS, GPS, IMU, Algorithm, Neural Network, NEAT, Genetic Algorithm, Modeling, Simulation

Scientific Field: Technical Sciences – Mechanical Engineering

Scientific Subfield: Weapon Systems

UDK: 623.46:681.5(043.3)

# Синтеза алгоритама навигације и вођења пројектила заснованих на машинском учењу

Резиме:

У овој докторској тези се разматра употреба машинског учења у синтези алгоритама навигације, управљања и вођења ракете, као алтернативи традиционалним алгоритмима. Алгоритам машинског учења који се користи у овој докторској тези је заснован на примени неуронских мрежа. Неуронска мрежа се обучава Неуро еволуционим алгоритмом са прwhiширеном топологијом. Осим тога, извршено је математичко моделовање вођеног пројектила и његовог окружења како би се извршиле нумеричке симулације и упоредиле његове перформансе. Извршено је поређење неуронске мреже алгоритма вођења терминалне фазе са алгоритмом пропорционалне навигације. Осим тога, интеграција GPS/INS-а на бази неуронских мрежа је упоређена са Калмановим филтром. На крају је дато поређење аутопилота по каналима ваљања, пропињања и скретања реализованих неуронским мрежама насупрот традиционалним аутопилотима са ПИД управљачким алгоритмима. Циљ ове докторске тезе је синтеза алгоритма вођења и управљања пројектила применом неуронских мрежа које треба да покаже слично или боље понашање од традиционалних решења. Притом, верификује се одрживост решења примене неуронских мрежа у синтези алгоритама управљања и вођења.

Кључне речи: пројектил, вођење, управљање, навигација, синтеза, машинско учење, ПИД, ИНС, GPS, IMU, алгоритам, неуронска мрежа, NEAT, генетски алгоритам, моделовање, симулација

Научна област: Техничке науке - Машинство

Ужа научна област: Системи наоружања

УДК: 623.46:681.5(043.3)

## Table of Contents

## Table of Figures

# Acknowledgements

I would like to express my deepest gratitude and appreciation to Allah for all the known and hidden blessings in my life. In addition, I would also like to thank Him for His guidance and giving me the patience to persevere.

I would also like to express my sincere appreciation to the United Arab Emirates and Their Highnesses The Rulers of United Arab Emirates for their tremendous supports and providing me with the enormous learning opportunities. In addition, I would like to extend my appreciation to Emirates Advanced Research and Technology Holdings for their support and for providing me with the opportunity to learn as well.

Life will not be the same without the best family in the world. They have always supported me, encouraged me, and believed in me even through the most difficult times when everything seemed impossible. I will not be here today without you.

They say "If you want to walk fast walk alone. If you want to walk far walk together". This proverb captures the essence of the relationships and friendships that kept me going through the adventurous journey. Thank you all for sharing every moment and riding the roller coaster of life with me.

Furthermore, I would like to take this opportunity to express my immense gratitude to the University of Belgrade and its professors for all the knowledge I have gained. In addition, a special mention to four professors who played a major role in helping me through the doctorial journey. Thank you Professor Dragan Lazic, Professor Milan Ristanovic, Professor Ivana Todic, and Professor Stevica Graovac. Your constant guidance and believe in me helped make this thesis possible. I would also like to thank all the professors and teachers that taught me from as little as a single letter to as much as an ocean of knowledge.

# 1. Introduction

Machine learning has become one of the most popular topics in modern technology. It is used by major technological companies such as Google, Microsoft, and Apple. That is because machine learning has shown to have great potential in solving engineering problems that are otherwise difficult to solve. For instance, facial recognition and content suggestion all use machine learning as it performs much better than other algorithms.

The purpose of this research is to investigate the potential benefits of machine learning in missile guidance, navigation, and control system applications. More specifically, this thesis will focus on investigating whether machine learning can be used in terminal guidance as an alternative to proportional navigation, GPS/INS integration, as an alternative to Kalman filter, and roll, pitch, and yaw autopilot as an alternative to PID controllers.

There are several assumptions made about the thesis. The major assumption is that the models used to simulate the missile and its environment is accurate. Moreover, the guidance computer behavior is assumed to be realistic. This means that the microprocessor is expected to behave exactly like the simulation. This includes all guidance, navigation, and control algorithms as well as the trained neural network algorithm counterpart. Hence, the performances are also assumed to be realistic.

Furthermore, a deep understanding of machine learning and the missile system is required. Hence, this thesis will discuss machine learning and its training. Then, it will discuss the modeling of the missile system and its environment. It will also present the neural network guidance, navigation, and control designs. Once an understanding has been established, the performances of the traditional and neural network guidance, navigation, and control designs will be compared and analyzed quantitatively. This allows for the evaluation of the neural network designs as a viable solution for the missile system.

## 1.1. Methodology

### 1.1.1. Missile Simulation

Since the missile is an autonomous dynamic system, it needs to be accurately modeled and simulated. Therefore, Simulink will be used to simulate the environment and the missile. The environment includes Earth's gravity, atmosphere, and transformation matrices. Hence, Earth's gravity will be modeled using the WGS84 standard. In addition, the COESA standard will be used to model the variations in wind, pressure, and temperature with altitude. Furthermore, the transformation matrices will include the Coriolis correction to account for Earth's rotation.

Once the environment is modeled, the propulsion of the missile will be modeled. This is to account for the thrust forces as well as moments due to misalignment. Then, the structural elements such as the variable mass and moment of inertia will be modeled. Moreover, the aerodynamic forces and moments will be modeled. This generates the physical properties of the missile.

The last step is to model the guidance computer and the actuation system. The actuation system consists of all the actuator models of the missile. Moreover, the guidance computer contains the traditional missile guidance, navigation, and control models. The missile guidance and control models allow the missile to reach the target. Additionally, the IMU, GPS, and INS algorithm models allow the missile to find its location.

### 1.1.2. Neural Network Design

For each of the guidance, navigation, and control, a unique neural network will be trained. Then, these neural networks will be integrated into the simulation replacing their traditional counterparts. This allows for the viability of the neural network as a solution to be verified. Furthermore, unlike the traditional neural network which is trained by back propagation, this thesis will use Neuro Evolution of Augmenting Topologies method to train the neural network. The uniqueness

of this training method is that it is an online training method that grows a neural network to optimally perform the intended purpose.

### 1.1.3. Data Generation and Collection

The simulation with traditional guidance, navigation, and control will be executed with variable initial conditions and target location. Then, the simulation with the neural network guidance, navigation, or control will be executed with the corresponding variable conditions. The performance of each of the guidance, navigation, and control of both traditional and neural network simulations will be recorded. Thereby, this completes the data generation and collection.

### 1.1.4. Data Analysis

Once the data collection is completed, the performances of each corresponding guidance, navigation, or control will be analyzed. The performances of the proportional navigation and terminal guidance neural network will be compared. In addition, the pure INS, Kalman filter integration, and neural network integration errors will be compared to the 6 DOF data. Furthermore, the traditional roll, pitch, and yaw autopilot performances will be compared to the neural network autopilots. A figure of merit will be used to evaluate the performance of the traditional and neural network methods. The figure of merit for guidance is the miss distance. On the other hand, the figure of merit for the navigation error and the autopilot performances is the mean square error. This allows the performances of both traditional and neural network methods to be quantitatively evaluated.

### 1.1.5. Expected Results

The expected results are that the neural network guidance, navigation, and control designs will perform similar or better than their traditional counterparts. This means that the viability of the neural network as a guidance, navigation, or control solution will be verified. In addition, the thesis is expected to overcome the challenges with neural network generalization in order to apply the solutions in real life.

# 2. Introduction to Machine Learning

The advent of machine learning allowed computers to learn from past experiences. This means that computers can learn to adapt mathematical model behaviors and solve complex problems. Hence, this means that computers can solve problems that was otherwise difficult to solve using mathematics. In addition, it can reduce the solving complexity of the algorithm. Machine learning is a vast and growing field that is divided into multiple categories. Figure 1 shows an example of the machine learning categories.
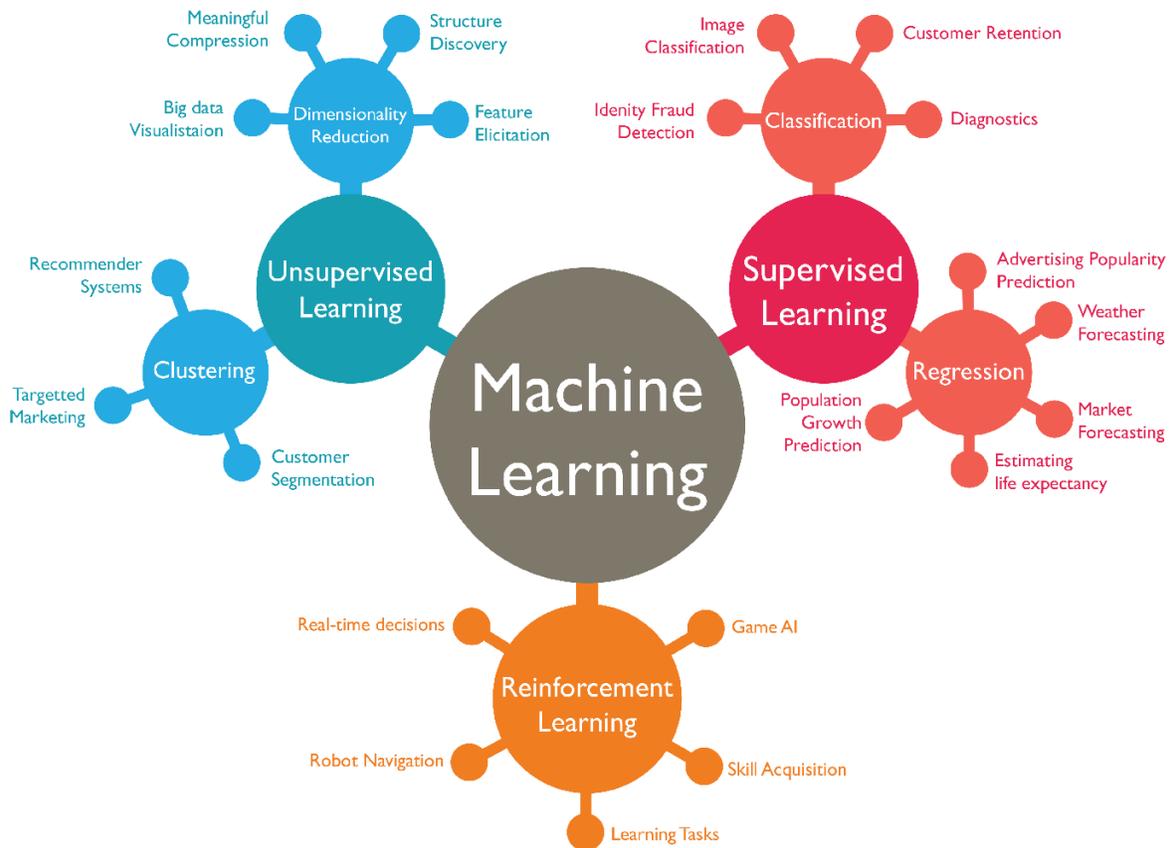


Figure 1 - Machine Learning Fields (1)

The three major categories of machine learning are supervised, unsupervised, and reinforcement learning. Supervised learning happens when a set of data is used to train the machine learning algorithm where the algorithm receives feedback of its performance. On the other hand, unsupervised learning also happens when a set of data is used for

training the machine learning algorithm. However, it learns without the supervision. In other words, the machine learning algorithm needs to figure out its performance independently. Lastly, reinforcement learning happens when the machine learning algorithm interacts with the environment. Based on that interaction, the algorithm is awarded a reward. By maximizing the reward, the optimal algorithm is trained.

Furthermore, machine learning algorithms can be further classified as shown in Figure 2. Here, the supervised learning can be divided into classification and regression. Classification happens when a set of input are mapped to a certain output. For instance, a set of image feature can be classified to be a cat. On the other hand, regression happens when the set of input is best fitted to produce an output, similar to linear or nonlinear regression. Moreover, unsupervised learning can be labeled clustering. That is because the machine learning algorithm tries to make sense of the data and groups them in to different clusters.
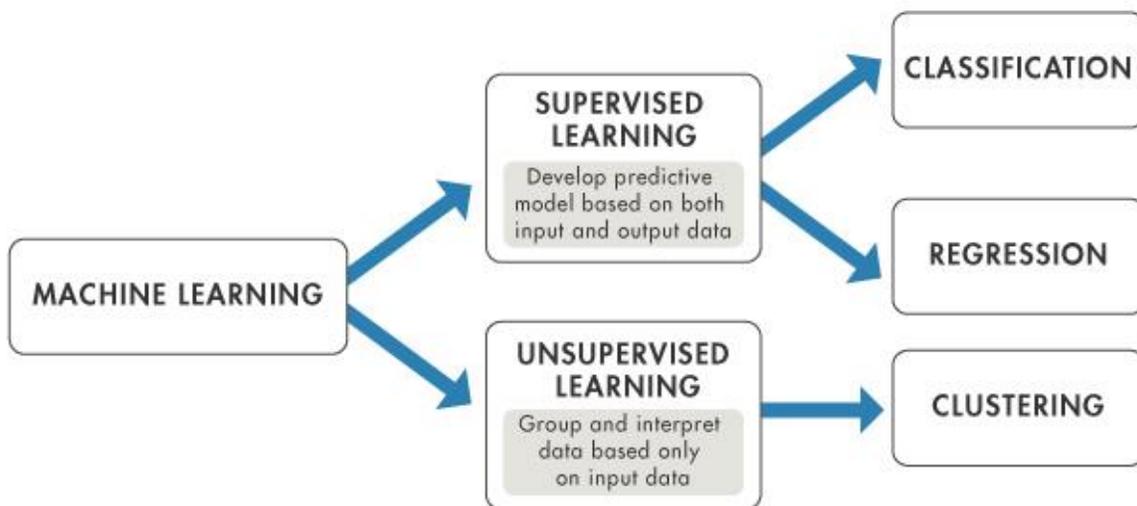


Figure 2 - Machine Learning Sub-Categories (2)

Additionally, machine learning algorithm can be categorized based on algorithms as seen in Figure 3. Here algorithms such as Support Vector Machines (SVM) and Naïve Bayes are used for classification. In addition, algorithms such as Support Vector Regressor (SVR) and Decision Tree can be used for regression. Moreover, algorithms such as K-means and Hidden

Markov Model can be used for clustering. Furthermore, Q-learning algorithm can be used for reinforcement learning.

It is important to note that the neural networks are a rather unique family of algorithms. That is because it can be used for classification, regression, and clustering. In addition, it can be used for reinforcement learning as well. Hence, it can be seen that the neural networks are adaptable. It is this exclusive property that is the reason that neural networks are the leading algorithms in modern science.
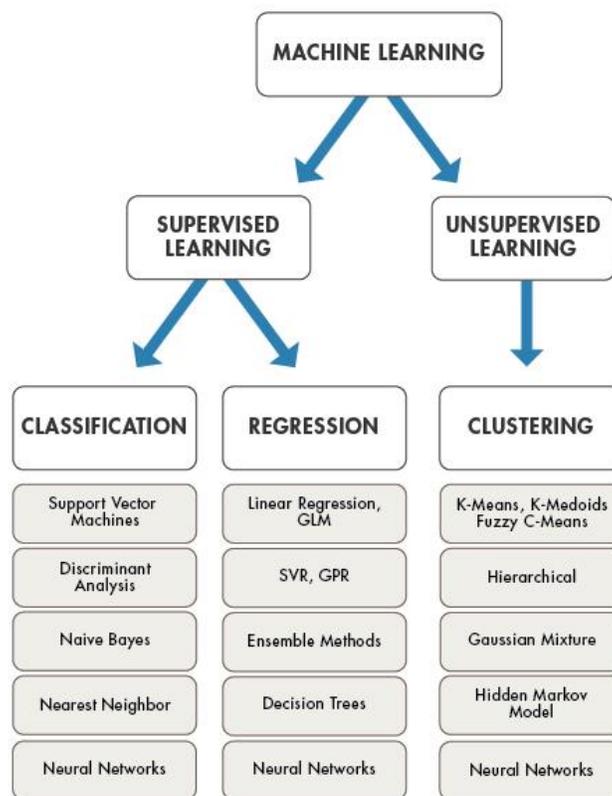


Figure 3 - Machine Learning Algorithms (2)

## 2.1. Neural Network

The machine learning category used in this thesis is the reinforcement learning. That is because reinforcement learning allows for the machine learning algorithm in the missile to interact with the environment. In addition, neural networks algorithm is chosen because of its adaptability to different types of learning.

## 2.1.1. Neural Network Design

Neural networks algorithm is inspired by the brain's network of neurons. Figure 4 shows the biological neuron. Here, the dendrites serve as an input to the biological neuron. On the other hand, the axons serve as an output to the biological neuron. These networks of neurons are linked together as shown in Figure 5 where the dendrites are connected to axons. The flow of electric pulses in the brain allows the neurons to communicate. This is thought to be the bases of memories and decisions.
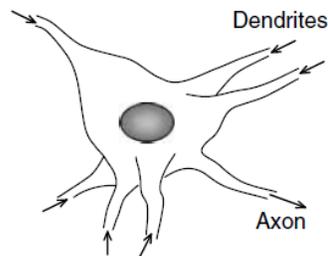
Figure 4 - Biological Neuron (3)
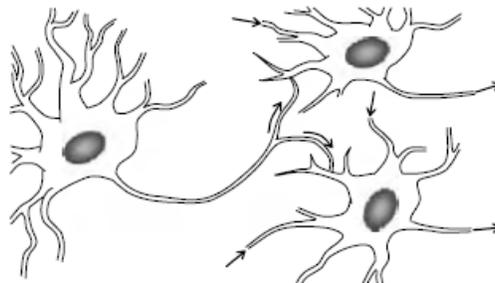
Figure 5 - Biological Neural Network (3)

The artificial neuron shown in Figure 6 is very similar to the biological neuron. Here, there are multiple input ports to the artificial neuron similar to the dendrites. In addition, there are output ports from the artificial neuron similar to the axons. These networks of artificial neurons are linked together as shown in Figure 7 where the input and output ports are connected to form the Artificial Neuronal Network (ANN) or Neural Network (NN). The flow of logic through the neural network is how decision is made.
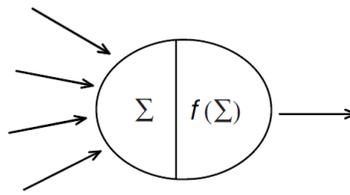
Figure 6 - Artificial Neuron (3)
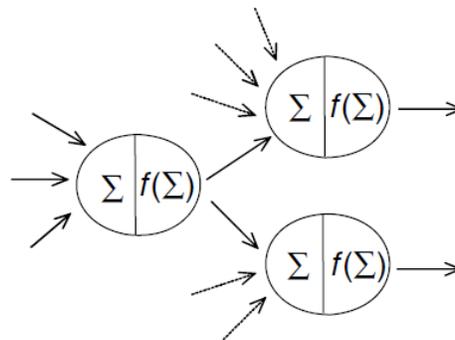


Figure 7 – Artificial Neural Network (3)

The artificial neuron operational concept is shown in Figure 8. Here, each individual input is weighted then summed with all the inputs. The sum of the weighted inputs is put through an input-output function. The most common function is the sigmoid function. That is because it produces a normalized output for the sum of the inputs. These normalized outputs are then inputted into the next set of artificial neurons.
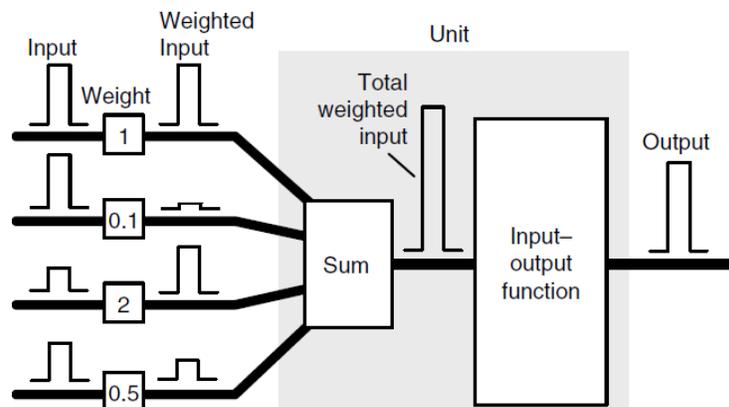


Figure 8 – Artificial Neuron Operational Concept (3)

Despite the similarity, the artificial neural network is much simpler than its biological counterpart. Figure 9 shows the evolution of the artificial network. Here, it can be seen that the biological network is simplified into computational neuroscience. These early models were computational heavy. Hence, they were simplified in order to form the modern artificial neural networks.
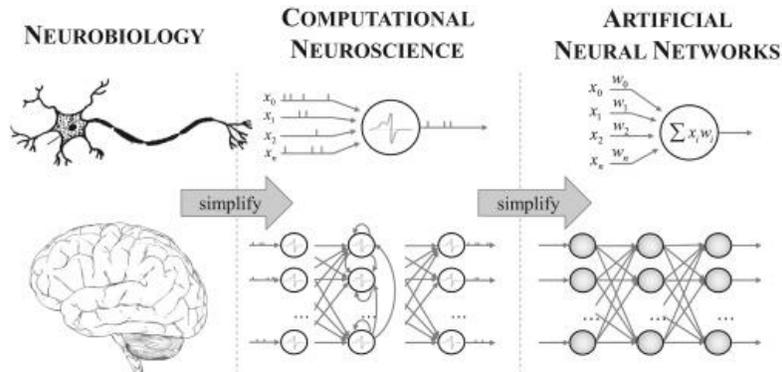


Figure 9 - Biological to Artificial Network (4)

The artificial neural network can be organized as a set of layers. The first layer is known as the input layer which is denoted with I. The input layer consists of a set of neurons that take in the input to the network. The length of the input layer is identical to the number of inputs. The second set of layers are the hidden layers which is denoted by H. These set of layers can have variable length and width and serve as the main computational logic for the network. The last layer is the output layer which is denoted by O. These layers serve as the last adjustment before the result of the neural network computation is exported. Moreover, the length of the layer is identical to the number of outputs.
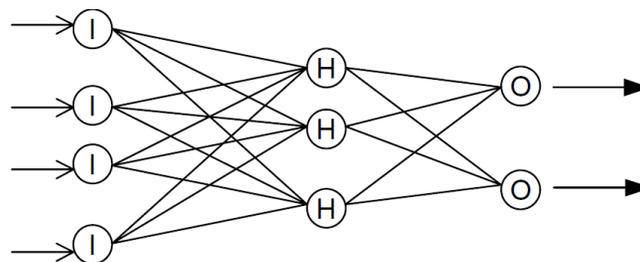


Figure 10 - Neural Network Layers (3)

## 2.1.2. Training

There are vast amounts of training method for neural networks. Depending on the type of the neural network design, the best training method differs. Popular training methods such as Bayesian Regularization and Scaled Conjugate Gradient all have their strengths and weaknesses. However, since this thesis uses a traditional feed forward neural network, the most effective training function used for this type of networks is the Levenberg-Marquardt (LM) training method (5).

The LM algorithm is an iterative method for training neural network that is depended on back propagation technique. This means that the differential error of the neural network prediction is compared to the supervised results. This enables the iterative method to adjust the neuron weights and thereby train the neural network (5).

The LM algorithm is an improvement on the Gauss-Newton algorithm that is shown in Equation 1. Here, $W_{i+1}$ is the updated weight after each iteration. In addition, the $W_i$ is the current weight of the network. The current weight is subtracted from the correcting factor. Moreover, the Jacobian matrix J consists of all the first derivative weights and bias errors. Additionally, the vector of network error is represented by e (5).

| | |
|---|---|
| $$W_{i+1} = W_i - [J^T(x)J(x)]^{-1} J^T(x) \cdot e$$ | Equation 1 <br> (5) |

The LM algorithm shown in Equation 2 adds a correction term to improve the training performance. Here, the $\mu_k$ term which can be updated with each iteration makes the correcting factor invertible (5).

| | |
|---|---|
| $$W_{i+1} = W_i - [J^T(x)J(x) + \mu_k I]^{-1} J^T(x) \cdot e$$ | Equation 2 <br> (5) |

## 2.1.3. Genetic Algorithm

The genetic algorithm (GA) is a biological search algorithm. The algorithm is modeled based on the natural selection. This means that the biological search algorithm chooses the fittest of the solutions and grows it into the next generation and eliminates the unfit solutions. In addition, the genetic algorithm also cross breads solutions and randomizes it in order to search for a more optimal solution (6).

Figure 11 show an example of a two-dimensional solution space. Here, there are a couple of maxima and minima. Assuming that the solution space represents the errors of the design, the task becomes a minimization problem. The goal of the genetic algorithm is to search for all the minima. Hence, by mimicking biological evolution, the genetic algorithm eliminates solutions that are not minima. This is achieved by evaluating the solution using fitness function. In addition, it is important to note that the genetic algorithm does not have a unique solution because it can converge into different local minimum.
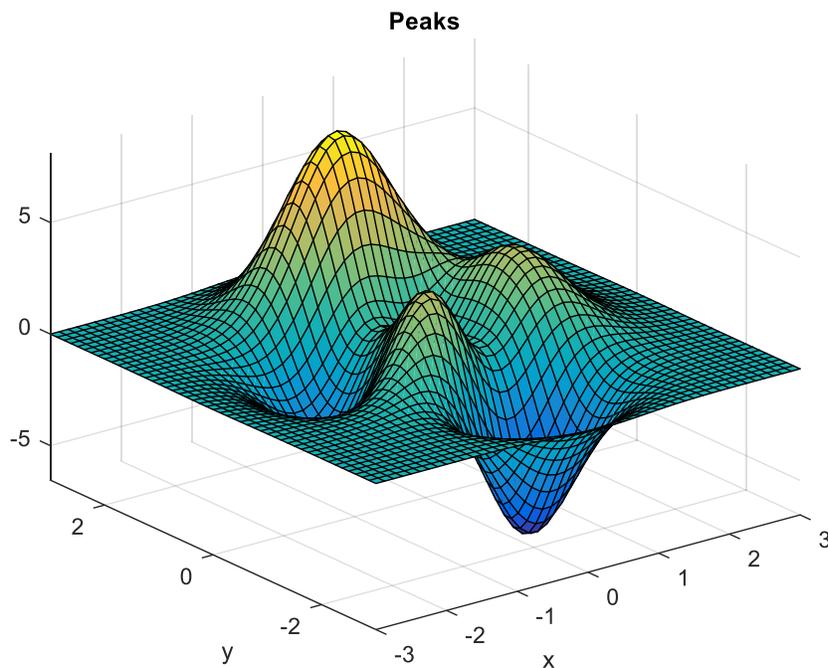


Figure 11 - Peak Graph (7)

## 2.1.3.1. Fitness Function

The fitness function is used as an evaluation criterion for the genetic algorithm. The goal of the fitness function is to provide a quantitative measurement of the genetic algorithm solution performance. It is important to note that output of the fitness function results is a single value. Hence, it has to be carefully designed to properly evaluate the genetic algorithm with a single value output. This also means that a weight $W_n$ should be assigned to different evaluation criteria that sums up the fitness function as shown in Equation 3.

| | |
|---|---|
| $Fitness = W_1 Criterion_1 + W_2 Criterion_2 + \cdots + W_n Criterion_n$ | Equation 3 |

One example of an evaluation criteria is the Mean Square Error (MSE) shown in Equation 4. Here, the MSE takes the input from two sources, the desire and response for instance. Then, it outputs a quantitative numerical value that indicates the closeness the two sources are from each other. This performs very well when comparing the actual performances of a missile system with the desired performances.

| | |
|---|---|
| $$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2$$ | Equation 4 <br> (8) |

## 2.1.3.2. Stop Criteria

It is important to identify and choose the correct stop criteria for genetic algorithm. That is because the genetic algorithm can converge to a local minimum. It can also diverge to infinity. Moreover, it can also get stuck in a loop that has very minimal or no improvement from a generation to another.

The stop criteria can be as a form of time limit for the biological search. In addition, it can be the average change of solution between generations. Moreover, the search algorithm should also be limited within a certain search range. These criteria can be set together or alone in order to achieve maximum optimization.

## 2.1.4. NeuroEvolution of Augmenting Topologies

The NeuroEvolution of Augmenting Topologies (NEAT) is a technique that utilizes the concept of genetic algorithm and reinforcement learning to train the neural network. The weight of the neural network is varied with the genetic algorithm. Here, every generation randomizes some weights of the neural network while inheriting some others from the previous generation. In addition, the neural network acts as an agent that is connected live to the missile environment. This means that the neural network algorithm in the guidance computer interacts with the missile environment in order to achieve the desired performances. These performances are evaluated using fitness function in order to identify the fittest and unfit solutions. Therefore, growing the neural network to perform the desired functionality. This process can be seen in Figure 12 (9).
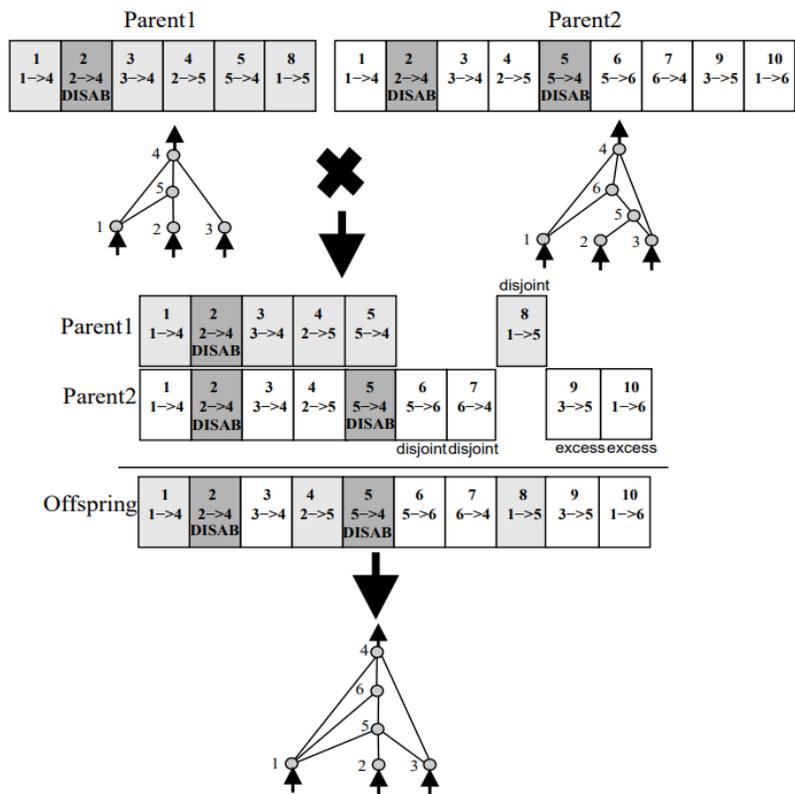


Figure 12 – Neural Network Evolution Concept (9)

## 2.1.5. Challenges

The neural network generalization problem can present an issue in designing a neural network solution. That is because neural networks perform very well for repetitive patterns. However, since the missile is a dynamic system, it can be a challenge. Moreover, not every application can be trained using neural network. The challenge is to find a way to train the neural network that will successfully perform adequately for all situations. In addition, the selection of inputs and outputs for the training is challenging. That is because there are no agreed upon rules for choosing the inputs and the outputs for the neural network. Consequently, the selection of the incorrect inputs and outputs can result in very different neural network performance.

Since there are several neural network designs, choosing the optimal neural network design is a challenge. That is because the design of the neural network affects the performance greatly. For instance, a feedforward network might be simpler but prone to errors. However, a cascade forward network reduces the error but adds complexity to the system. The complexity of the network design also extends to the number of layers of the network. Once again, the lower network size might not be sufficient to produce the optimal result while high network size might make the network too generalized.

Additionally, since the neural network field is rather modern, there aren't many platforms that supports it. And some that support it have limited functions or require deep knowledge of the field. For instance, MATLAB and Simulink support neural network design. However, the support is limited. This means that in order to have full control of the neural network, deep knowledge is required.

Moreover, the training of neural networks requires a lot of computation power. This means that it will require a lot of time to design any network, let along optimizing it. In addition, high computation power is not readily available. This means that every training needs to be understood properly. Moreover, it also means that the results should be predicted before the initiation of the training to optimize time.

# 3. Mathematical Modeling

## 3.1. Missile Environment

Since the missile is flying on Earth, its environment needs to be modeled. This is achieved by using the gravitational model as well as the atmospheric model. In addition, the missile needs to be seen from different perspective. Hence, coordinates frames allow for such point of view. Moreover, Transformation matrices allow for the transition between different coordinate frames. Furthermore, quaternions allow for a more efficient way to calculate the transition between the coordinate frames.

### 3.1.1. Gravitational Model

Earth's gravity is not consistent throughout the planet. Hence, a mathematical model was developed to model these changes. The estimation of the changes in gravity ranges from basic to more complex models. The basic models enable easier computations while the more complex ones are more accurate.

#### 3.1.1.1. Flat Earth Model

The flat earth model assumes that Earth is not rotating and that the gravitational acceleration g is constant. This is the simplest model and it allows for a quick estimation of the missile's trajectory. This model is important for short range missiles where the gravitational acceleration and Earth's rotation has minimal effects on the trajectory. The gravitational acceleration can be calculated using Newton's law of gravity as shown in Equation 5.

| | |
|---|---|
| $$g = G\frac{M}{r^2}$$ *where Gravitational Constant* $(G) = 6.67 \times 10^{-11} \text{Nm}^2\text{kg}^{-2}$ $M$ *mass of EARTH* $6 \times 10^{24}\text{kg}$ $r$ *to earth surface distance* $= 6.371 \times 10^6\text{m}$ | Equation 5 (10) |

## 3.1.1.2. Spherical Earth Model

The spherical earth model also assumes that Earth's gravity is constant. However, unlike the flat earth model, the spherical earth model considers the rotation of Earth as seen in Figure 13. In addition, the effect of Earth's curvature is also modeled. This model is important for medium range missiles where the rotation of Earth has an effect on the missile's trajectory. However, the gravitational acceleration effects are minimal.
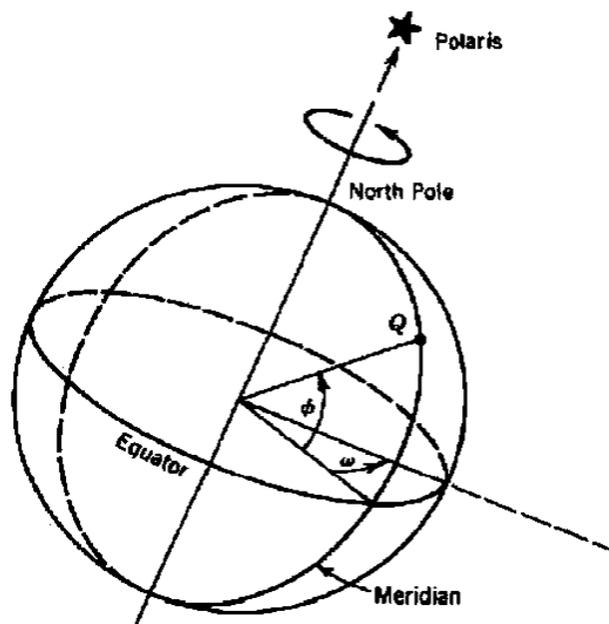
Figure 13 - Spherical Earth Model (11)

## 3.1.1.3. Elliptical Earth Model

The most popular earth model is the elliptical earth model. That is because it considers the changes in gravitational acceleration as well as the rotation of Earth. Although Earth is not elliptical, the estimation obtained from the elliptical model is accurate enough. This model is important for long range missiles where the rotation of Earth and gravitational acceleration have an effect on its trajectory. One of the most popular elliptical models is the WGS84 as shown in Figure 14.
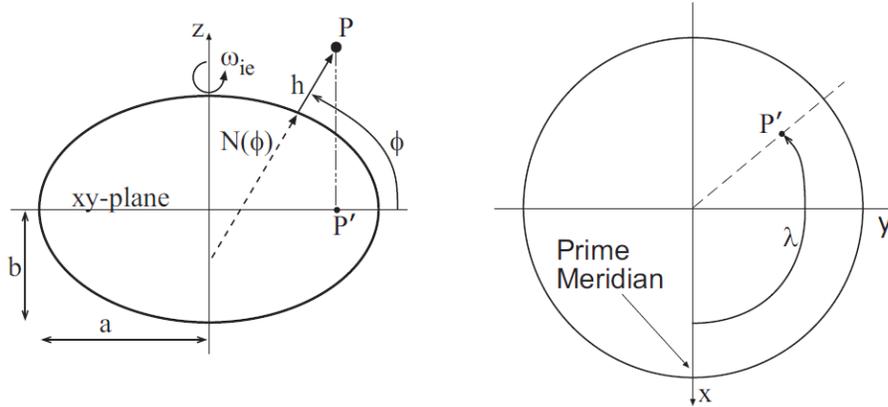
Figure 14 - Elliptical Earth Model (12)

The WGS84 model defines the ellipsoid using the following characteristic equations.

| | |
|---|---|
| $Equatorial\ Radius\ a = 6378137\mathrm{m}$ | |
| $Reciprocal\ Flattening \dfrac{1}{f} = 298.25$ | |
| $Flatness\ f = 0.00335281$ | |
| $Semiminor\ Axis\ b = a(1-f) = 6356752.314\mathrm{m}$ | |
| $Eccentricity\ e = \sqrt{f(2-f)} = 0.08181919$ | |
| $Angular\ Rate = \omega_{ie} = 7.29 \text{ x } 10^{-5}\,\dfrac{\mathrm{rad}}{\mathrm{s}}$ | |
| $Meridian\ Radius\ R_M(\phi) = \dfrac{a(1-e^2)}{(1-e^2\sin^2(\phi))^{\frac{3}{2}}}$ | Equation 6 – WGS84 (12) |
| $Normal\ Radius\ R_N(\phi) = \dfrac{a}{(1-e^2\sin^2(\phi))^{\frac{1}{2}}}$ | |
| $Gravitational\ Acceleration\ g = \begin{bmatrix} 0 \\ 0 \\ \gamma(\phi) \end{bmatrix} + \begin{bmatrix} \zeta_g \\ \eta_g \\ \delta_g \end{bmatrix}$  *Where*  $\gamma_e = 9.7803267715\,\dfrac{\mathrm{m}}{\mathrm{s^2}}\ and\ \gamma(\phi) = \gamma_e\dfrac{1+0.0019\sin^2(\phi)}{\sqrt{1-0.00669\sin^2(\phi)}}$  $\begin{bmatrix} \zeta_g \\ \eta_g \\ \delta_g \end{bmatrix}$ *Are The Local Pertubation of Gravity Vector* | |

## 3.1.2. Atmospheric Model

As altitude increases, the temperature, pressure, and density of air changes. There is no universal standard to model the changes of the atmosphere. However, the most popular standards are the International Standard Atmosphere (ISA) and the Committee on Extension to the Standard Atmosphere (COESA). This thesis will use the COESA standard which is shown in Table 1. The atmospheric model is important because the air pressure and density significantly affects the performance of the missile. Hence, the accuracy of the model is essential.

| Height (km) | Temperature (K) | Pressure (mb) | Height (km) | Temperature (K) | Pressure (mb) |
|---|---|---|---|---|---|
| 0.0 | 288.2 | 1013.2 | 11.0 | 216.8 | 227.0 |
| 1.0 | 281.7 | 898.8 | 12.0 | 216.6 | 194.0 |
| 2.0 | 275.2 | 795.0 | 14.0 | 216.6 | 141.7 |
| 3.0 | 268.7 | 701.2 | 16.0 | 216.6 | 103.5 |
| 4.0 | 262.2 | 616.6 | 18.0 | 216.6 | 75.65 |
| 5.0 | 255.7 | 540.5 | 20.0 | 216.6 | 55.29 |
| 6.0 | 249.2 | 472.2 | 25.0 | 221.6 | 25.49 |
| 7.0 | 242.6 | 411.0 | 30.0 | 226.5 | 11.97 |
| 8.0 | 236.2 | 356.5 | 35.0 | 236.5 | 5.746 |
| 9.0 | 229.7 | 308.0 | 40.0 | 250.4 | 2.871 |
| 10.0 | 223.3 | 265.0 | 50.0 | 270.6 | 0.798 |

Table 1 - COESA Atmospheric Model (13)

## 3.1.3. Coordinate Frames

There are several types of coordinate frames each with a unique perspective design for a particular usage. There is no limit to number of coordinate frames that could be defined. Nonetheless, this thesis will discuss the most commonly used frames in the missile applications.

### 3.1.3.1. Body Fixed Frame

The body frame is located at the center of gravity (CG) of the missile as seen on Figure 15. This frame is used to see from the missile's perspective. It is useful because aerodynamic and thrust forces and moments act along the frame's axis. Therefore, the forces and moments are more intuitive.
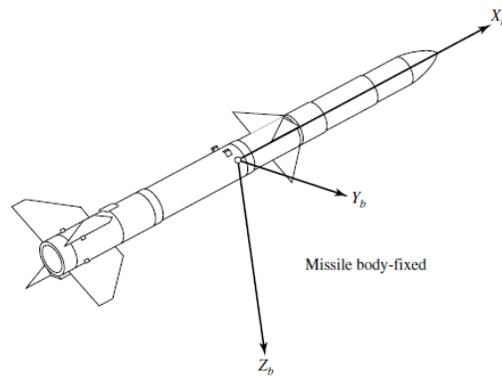
Figure 15 - Body Fixed Frame (14)

## 3.1.3.2. Sensor Frame

Modern missiles depend on the Inertial Measure Unit (IMU), shown in Figure 16, to estimate its orientation and position. Since the IMU is not aligned with the CG of the missile but is mounted on it, a unique frame is defined. This frame enables the navigation algorithm to compensate the difference between the sensor placement and the body frame.
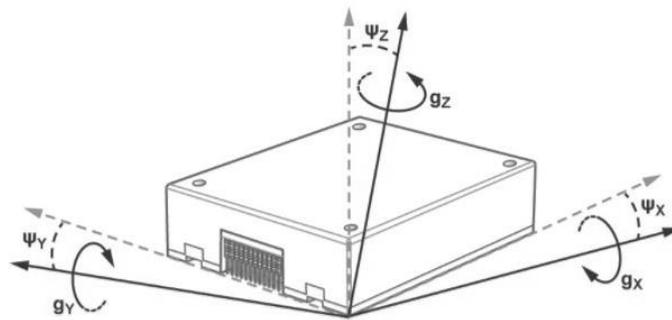


Figure 16 - Strapped Down IMU (15)

## 3.1.3.3. Navigation or North East Down (NED) Frame

Figure 17 shows the navigation frame placed on Earth where one axis is pointed directly north, one east, and one down. This aligns the navigation frame with the planet. The navigation frame is centered at the launching point of the missile. This is so that the missile always fires at initially zero coordinates. Hence, this allows for the distance travelled and the deviation to be more intuitive.
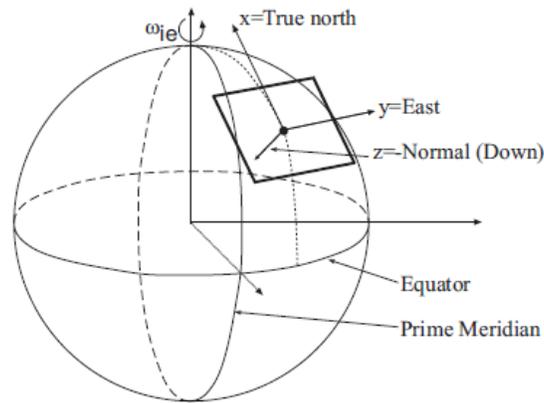
Figure 17 - Navigation Frame (12)

## 3.1.3.4. Wander Frame

The wander frame is identical to the navigation frame with one key difference. Since the navigation frame always points towards north and east, unless the missile is fired directly north or east, the distance travelled will have an azimuth angle component. In order to remove the effect of the angle, the wander frame is created. As seen in Figure 18, the wander frame is tilted away from the navigation frame, this means that as long as the missile is travelled forward the Y component will increase and as long as there is a right deviation, the X component will increase. It is important to note that the X and Y conventions here are for the navigation frame with East North Up (ENU) convention.
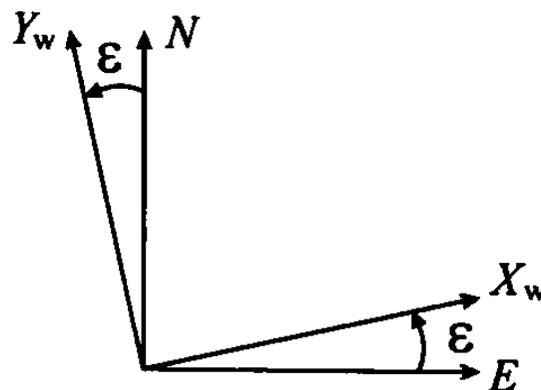


Figure 18 - Wander Frame (16)

## 3.1.3.5. Earth Centered Earth Fixed (ECEF) Frame

The ECEF frame is the frame that represents Earth without rotation. As seen in Figure 19, the ECEF frame is center at the planet's core with a plane on the equator and another on the prime meridian. The frame is fixed to the planet, and hence, moves with it. This means that it cannot see the Earth's rotation. This frame is important because it allows the missile to be absolutely localized within the planet.
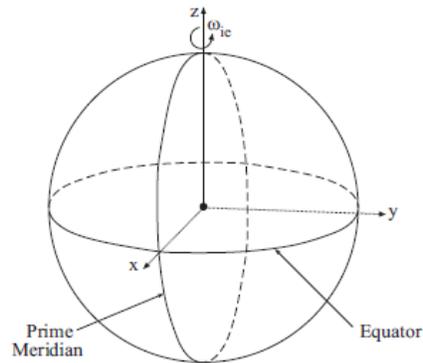
Figure 19 - ECEF Frame (17)

## 3.1.3.6. Earth Centered Inertial (ECI) Frame

The ECI frame is also centered at the core of the planet. However, unlike the ECEF frame, one of the ECI planes are on the Equinox. This means that the frame remains stationary while Earth rotates which allows it to see Earth's rotation. This frame is especially important for long distance flights where Earth's rotation affects the flight trajectory.
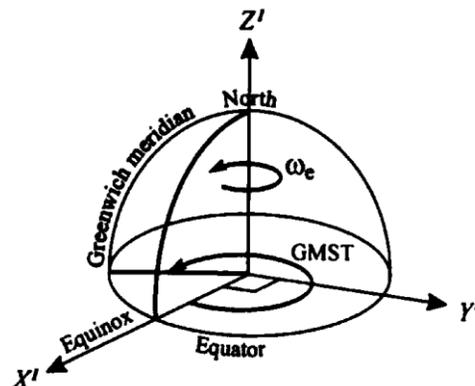
Figure 20 - ECI Frame (16)

## 3.1.3.7. Geodetic Frame

Another frame that has its center on Earth's core is the geodetic frame. Similar to the ECEF frame, the planes of the geodetic frame are on the meridian and the equator. However, the geodetic frame uses a polar system to represent the coordinates. This allows for easier computation. In addition, global positioning systems use the geodetic frame which makes it very essential.
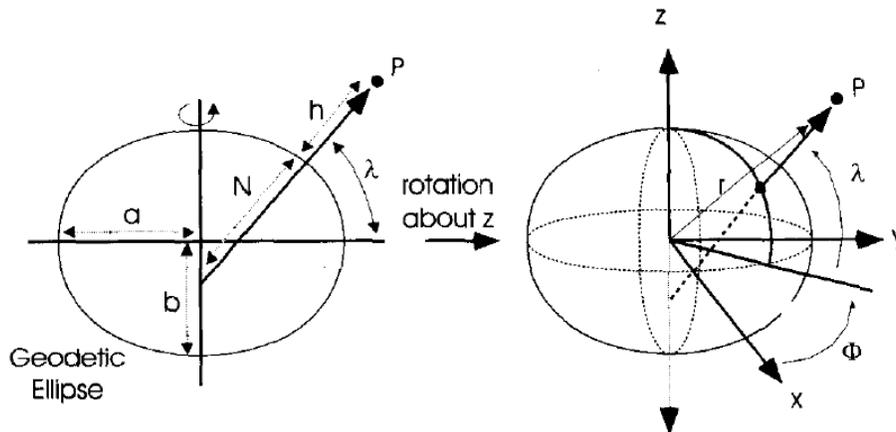


Figure 21 – Geodetic Frame (17)

## 3.1.4. Transformation Matrix

The transformation matrix allows the transition between the coordinate frames. This is important because it allows for the missile's forces and moments to be seen from a different perspective.

## 3.1.4.1. Body to Navigation Frame

The transformation matrix from the body frame to navigation frame is shown in Equation 7. Since there are three rotations, the transformation matrix is broken down into its components of roll, pitch, and yaw. Each transformation matrix component can be multiplied with forces or moments in order to change their perspective. It is important to note that the order of multiplication matters. Equation 7 shows the yaw-pitch-roll order. In addition, it is essential to keep in mind the ranges specified in Equation 7. That is because the pitch angle $\theta$ can align the roll and yaw axis which causes singularity in the XYZ or ZYX Euler angle notations. This could happen when $\theta$ is $\frac{\pi}{2}$ rad or $\frac{3\pi}{2}$ rad.

| | |
|---|---|
| $$Roll\ Transformation\ Matrix\ \ T_{Roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$ | |
| $$Pitch\ Transformation\ Matrix\ \ T_{Pitch} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$ | |
| $$Yaw\ Transformation\ Matrix\ \ T_{Yaw} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ | Equation 7 (14) |
| Body to Navigation Transformation Matrix $T_{Body}^{Navigation}$ $$T_{Body}^{Navigation} = T_{Roll} * T_{Pitch} * T_{Yaw} =$$ $$\begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}$$ | |
| $$\begin{aligned} -\pi \le \phi < \pi \quad &\text{or} \quad 0 \le \phi < 2\pi, \\ -\pi \le \psi < \pi, & \\ -\pi/2 \le \theta \le \pi/2 \quad &\text{or} \quad 0 \le \psi < 2\pi. \end{aligned}$$ | |

## 3.1.4.2. ECEF To Navigation Frame

Equation 8 shows the transformation matrix from ECEF to Navigation frame. There are two transitions between the two frames. The latitude transformation matrix $T_\phi$ and the longitude transformation matrix $T_\lambda$ are multiplied in longitude-latitude order. The transformation is illustrated on Figure 22.

| | |
|---|---|
| $$T_\lambda = \begin{bmatrix} \cos\lambda & \sin\lambda & 0 \\ -\sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ | |
| $$T_\phi = \begin{bmatrix} \cos(\phi + \frac{\pi}{2}) & 0 & \sin(\phi + \frac{\pi}{2}) \\ 0 & 1 & 0 \\ -\sin(\phi + \frac{\pi}{2}) & 0 & \cos(\phi + \frac{\pi}{2}) \end{bmatrix} = \begin{bmatrix} -\sin\phi & 0 & \cos\phi \\ 0 & 1 & 0 \\ -\cos\phi & 0 & -\sin\phi \end{bmatrix}$$ | Equation 8 (12) |
| $$T_{ECEF}^{Navigation} = T_\phi * T_\lambda = \begin{bmatrix} -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\phi\cos\lambda & -\cos\phi\sin\lambda & -\sin\phi \end{bmatrix}$$ | |
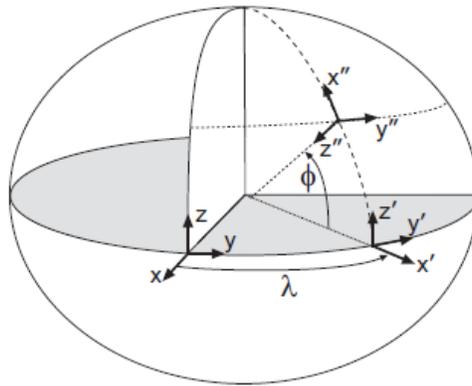
Figure 22 - Transformation Matrix ECEF to Navigation (12)

### 3.1.4.3. Geodetic to ECEF Frame

The transformation from geodetic to ECEF Frame can be completed with a system of equations shown in Equation 9. Here, the WGS84 model's parameters are used to transform the coordinates.

| |  |
|---|---|
| $x = (R_N + h)\cos\phi\cos\lambda$ | Equation 9 |
| $y = (R_N + h)\cos\phi\sin\lambda$ | (12) |
| $z = (R_N(1 - e^2) + h)\sin\phi$ | |

### 3.1.4.4. Properties of Transformation Matrix

A unique property of the transformation matrix is the fact that it is orthogonal. This means that the determinant of the matrix is one. In addition, it means that the transpose of the matrix is also its inverse. This allows for simplified mathematical operations.

### 3.1.4.5. Puasson Equation

Earth's rotation has an effect on the derivative of the transformation matrix. Hence, the effect is compensated using Puasson's equation shown in Equation 10. $R_I$ is an arbitrary matrix in an inertial frame, or a frame where the point of origin is moving. On the other hand, $R_m$ is an arbitrary matrix in a non-inertial frame, or a frame that its point of origin is not moving. Here, $\breve{\omega}_m$ is the skew matrix of the rotational rates.

| | |
|---|---|
| $$\dot{R}_I = \dot{R}_m + \breve{\omega}_m R_I$$ | Equation 10 |
| $$\breve{\omega}_m = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$ | (16) |

### 3.1.5. Quaternions Matrix

An alternative way to calculate the transition between coordinates frame is by using quaternions. The challenge with transformation matrix is that it requires the computation of trigonometry functions, which can be computationally heavy. Hence, a four-dimensional complex number method is used as shown in Equation 11.

| | |
|---|---|
| $$Q = q_0 + q_1 i + q_2 j + q_3 k$$ | Equation 11 (16) |

### 3.1.5.1. Euler Angles to Quaternions

In order to use quaternions, Euler Angles are used to initialize quaternions. This can be accomplished using Equation 12. Here, $\mu$ is the intensity of rotation, $\alpha$, $\beta$, and $\gamma$ are the rotation angles respectively.

| | |
|---|---|
| $$q_0 = \cos\frac{\mu}{2}$$ | |
| $$q_1 = \sin\frac{\mu}{2}\cos\alpha$$ | Equation 12 (16) |
| $$q_2 = \sin\frac{\mu}{2}\cos\beta$$ | |
| $$q_3 = \sin\frac{\mu}{2}\cos\gamma$$ | |

### 3.1.5.2. Quaternions to Euler Angles

Quaternion can also be converted back into Euler Angles. This is done using Equation 13. $\phi$, $\theta$, and $\psi$ are roll, pitch, yaw, angles respectively.

| | |
|---|---|
| $\sin\theta = -2(q_2 q_0 + q_1 q_3)$ | Equation 13 |
| $\phi = \arctan2[2(q_2 q_3 - q_1 q_0), 1 - 2(q_1^2 + q_2^2)]$ | (17) |
| $\psi = \arctan2[2(q_1 q_2 - q_3 q_0), 1 - 2(q_2^2 + q_3^2)]$ | |

### 3.1.5.3. Transformation Matrix to Quaternions

An alternative way of initializing quaternions is by using the already established transformation matrix $T$. Here, the elements of the transformation matrix are used to calculate the initial quaternions as shown in Equation 14.

| | |
|---|---|
| $$Q = \begin{bmatrix} \dfrac{T[3,2] - T[2,3]}{4q_0} \\ \dfrac{T[1,3] - T[3,1]}{4q_0} \\ \dfrac{T[2,1] - T[1,2]}{4q_0} \\ \dfrac{1}{2}\sqrt{1 + T[1,1] + T[2,2] + T[3,3]} \end{bmatrix}$$ | Equation 14 (17) |

### 3.1.5.4. Quaternions to Transformation Matrix

Similarly, quaternions can be converted into transformation matrix. This can be accomplished using Equation 15.

| | |
|---|---|
| $$T = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_3 q_0) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$ | Equation 15 (16) |

### 3.1.5.5. Rotation Rates to Quaternion Derivative

Rotational rates p, q, and r can be used to calculate the next quaternion step. However, it is calculated with a differential order. The quaternion can be obtained by integration of the derivative of quaternion.

| | |
|---|---|
| $$\dot{Q} = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$ | Equation 16 (17) |

### 3.1.5.6. Quaternions Normalization

In order to improve computational performance, quaternions can be normalized using Equation 17. This aids the computation from quaternions to transformation matrix.

| | |
|---|---|
| $$N(Q) = \frac{q_0 + q_1 i + q_2 j + q_3 k}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}}$$ | Equation 17 (18) |

### 3.1.5.7. Quaternions Puasson Equation

The Puasson's equation can also be represented using quaternion. This allows for the computation to be done in the quaternion domain using Equation 18.

| | |
|---|---|
| $$\dot{Q} = \frac{1}{2} Q \omega$$ | Equation 18 (16) |

### 3.1.5.8. Quaternions Coordinate Transformation

Coordinate transformation can be done with quaternions $Q$ and its conjugate $Q^*$. Equation 19 shows the transformation from an arbitrary inertial frame $R_{Inertial}$ to a non-inertial frame $R_{Non\,Inertial}$.

| | |
|---|---|
| $$R_{Non\,Inertial} = Q R_{Inertial} Q^*$$ | Equation 19 (16) |

## 3.2. Missile Kinematics

Missile Kinematics is the description of how forces and moments impact the missile's body. Since the missile is considered as a rigid body, traditional kinematics laws can be used to describe the equations of motion.

### 3.2.1. Forces and Moments

Typically, there are six Degrees of Freedom (6DOF) in a missile. There are three translational forces and three rotational moments. The corresponding equations in vector form are shown in Equation 20.

| | |
|---|---|
| $Translational\ Force\ \sum F = ma$ | Equation 20 |
| $Rotational\ Moment\ \sum \tau = \dfrac{d}{dt}(r \times mV)$ | (14) |

Figure 23 illustrates the 6DOF of the missile. Here, u, v, and w represent the translational velocities while P, Q, and R represents the rotational velocity.
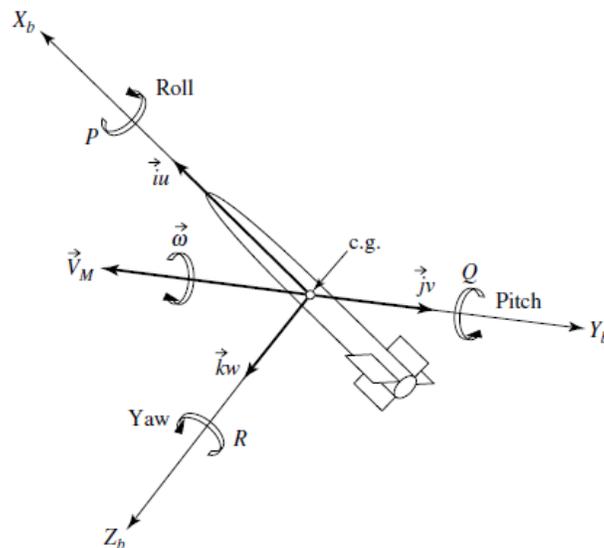


Figure 23 - 6 Degrees of freedom (14)

The force $F$ can be broken down into three components $F_x$, $F_y$, and $F_z$. Hence, the rewritten form is shown in Equation 21.

| | |
|---|---|
| $$F_x = \frac{\mathrm{d}(mu)}{\mathrm{dt}}, F_y = \frac{\mathrm{d}(mv)}{\mathrm{dt}}, F_z = \frac{\mathrm{d}(mw)}{\mathrm{dt}}$$ | Equation 21 (14) |

In addition, the moment $\tau$ can be broken down into three components $L$, $M$, and $N$. Hence, the rewritten form is shown in Equation 22. Here, $H$ is the moment of momentum.

| | |
|---|---|
| $$L = \frac{\mathrm{d}H_x}{\mathrm{dt}}, M = \frac{\mathrm{d}H_y}{\mathrm{dt}}, N = \frac{\mathrm{d}H_z}{\mathrm{dt}}$$ | Equation 22 (14) |

## 3.2.2. Inertial Effects on Forces

Since the missile body is on an inertial frame, the Puasson's form of the force formula is shown in Equation 23.

| | |
|---|---|
| $$F = m \left[\frac{\mathrm{d}V_M}{\mathrm{dt}}\right]_{body} + m(\omega \times V_M),$$ where $V_M$ is missile's velocity $\omega$ is rotational rate | Equation 23 (14) |

The cross multiplication of the missile's velocity and the rotational rates results in Equation 24.

| | |
|---|---|
| $$\omega \times V_M = (wQ - vR)\mathrm{i} + (uR - wp)\mathrm{j} + (vP - uQ)\mathrm{k}$$ | Equation 24 (14) |

This means that summation form of Equation 23 can be broken down into components as shown in Equation 25.

| | |
|---|---|
| $$\sum F_x = m(\dot{u} + w - vR)$$ | Equation 25 |
| $$\sum F_y = m(\dot{v} + uR - wP)$$ | (14) |
| $$\sum F_z = m(\dot{w} + vP - uQ)$$ | |

### 3.2.3. Inertial Effects on Moments

In addition, there are inertial effects on the missile's moments. That is because the velocity components of Equation 20 for rotational moments is affected by inertial. Hence, the Puasson form of the equation needs to be considered as shown in Equation 26.

| | |
|---|---|
| $$\left[\frac{dr}{dt}\right]_{body} = \left[\frac{dr}{dt}\right]_{None\ Inetial} + \omega \times r$$ | Equation 26 |
| $$Since\ \left[\frac{dr}{dt}\right]_{None\ Inetial} = 0$$ | (14) |
| $$V = \omega \times r$$ | |

Considering the effect of the inertial on the velocity, the updated moment of momentum equation is shown in Equation 27.

| | |
|---|---|
| $$H = r \times mV = mr \times (\omega \times r)$$ | Equation 27 (14) |

The cross multiplication of the moment distance and the rotational rates results in Equation 28.

| | |
|---|---|
| $$r \times (\omega \times r) = [(y^2 + z^2)P - xyQ - xZR]\mathrm{i} + [(z^2 + x^2)Q - yzR - xyP]\mathrm{j} + [(x^2 + y^2)R - xzP - yzQ]\mathrm{k}$$ | Equation 28 (14) |

It is important to note that Equation 22 holds true if the object of the rotation is a particle. However, since missile is an entire body, the moment of momentum equations is shown in Equation 29.

| | |
|---|---|
| $$H = \sum r \times mV = \sum mr \times (\omega \times r)$$ $$H = \left(\sum mr^2\right)\omega - \sum mr(r \cdot \omega)$$ $$H = \sum \delta H = \sum (r \times V)dm + \sum[r \times (\omega \times r)]\,dm$$ $$Since \;\; \sum (r \times V)dm = 0$$ $$\delta H = \sum[r \times (\omega \times r)]dm$$ $$H = \int r \times (\omega \times r)dm$$ | Equation 29 (14) |

By using Equation 29, the components of the moment of momentum can be written as in Equation 30.

| | |
|---|---|
| $$H_x = P \int (y^2 + z^2)dm - R \int xz \; dm = PI_{xx} - RI_{xz}$$ | |
| $$H_y = Q \int (x^2 + z^2)dm = QI_{yy}$$ | Equation 30 (14) |
| $$H_z = R \int (x^2 + y^2)dm - P \int xz \; dm = RI_{zz} - PI_{xz}$$ | |

Hence, the derivative can be taken to result in Equation 31.

| | |
|---|---|
| $$\frac{dH_x}{dt} = \frac{dP}{dt}I_{xx} - \frac{dR}{dt}I_{xz}$$ | |
| $$\frac{dH_y}{dt} = \frac{dQ}{dt}I_{yy}$$ | Equation 31 (14) |
| $$\frac{dH_z}{dt} = \frac{dR}{dt}I_{zz} - \frac{dP}{dt}I_{zz}$$ | |

The Puasson form of the moment equation is presented in Equation 32.

| | |
|---|---|
| $$\sum M = \overrightarrow{\left(\frac{dH}{dt}\right)} + \omega \times H$$ | Equation 32 (14) |

Here, the result of the cross multiplication of the rotation rates and moment of momentum is shown in Equation 33.

| | |
|---|---|
| $$\omega \times H = (QH_z - RH_y)\mathrm{i} + (RH_x - PH_z)\mathrm{j} + (PH_y - QH_x)\mathrm{k}$$ | Equation 33 (14) |

By substituting Equation 33 and Equation 31 in Equation 32, the resultant final summation form of the moment equations components is shown in Equation 34.

| | |
|---|---|
| $$\sum L = \dot{P}I_x - \dot{R}I_{xz} + QR(I_z - I_y) - PQI_{xz}$$ | |
| $$\sum M = \dot{Q}I_y + PR(I_x - I_z) + (P^2 - R^2)I_{xz}$$ | Equation 34 (14) |
| $$\sum N = \dot{R}I_z - \dot{P}I_{xz} + PQ(I_y - I_x) + QRI_{xz}$$ | |

### 3.2.4. Translational Equations of Motion

The acceleration of the missile can be calculated from the forces and rotation rates as presented in Equation 35.

| | |
|---|---|
| $$\frac{\mathrm{d}u}{\mathrm{dt}} = Rv - Qw + \frac{F_x}{m}$$ | |
| $$\frac{\mathrm{d}v}{\mathrm{dt}} = Pw - Ru + \frac{F_y}{m}$$ | Equation 35 (14) |
| $$\frac{\mathrm{d}w}{\mathrm{dt}} = Qu - Pv + \frac{F_z}{m}$$ | |

## 3.2.5. Rotational Equations of Motion

Although unintuitive, the derivative of Euler angles cannot be converted directly into rotation rates. The proper way of converting derivative of Euler angles to rotation rates, with the matrix form, is presented in Equation 36.

| | |
|---|---|
| $$P = \frac{\mathrm{d}\phi}{\mathrm{dt}} - \left(\frac{\mathrm{d}\psi}{\mathrm{dt}}\right)\sin\theta$$ | |
| $$Q = \left(\frac{\mathrm{d}\theta}{\mathrm{dt}}\right)\cos\phi + \left(\frac{\mathrm{d}\psi}{\mathrm{dt}}\right)\cos\theta\sin\phi$$ | Equation 36 (14) |
| $$R = \left(\frac{\mathrm{d}\psi}{\mathrm{dt}}\right)\cos\theta\cos\phi - \left(\frac{\mathrm{d}\theta}{\mathrm{dt}}\right)\sin\phi$$ | |
| $$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$ | |

## 3.3. Propulsion Model

The missile propulsion system consists of two subsystems. The booster and the sustainer. The purpose of the booster is to launch the missile and give it the push it needs to get off the ground and reach a desired state. The sustainer on the other hand is to maintain the velocity of the missile in order to extend its range. Figure 24 presents a simplified model of the turbojet engine. Here, it can be seen that turbojet engine is made up of compressor, fuel injection, combustion section, turbine section, and nozzle.
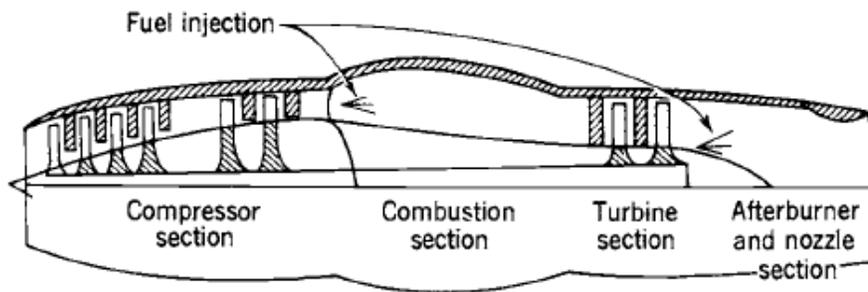


Figure 24 - Simplified Turbojet Engine (19)

This thesis will not discuss the turbojet engine modeling. However, the modeling principles are similar. The focus of this thesis will be the modeling of a rocket motor engine shown in Figure 25. The rocket motor consists of a combustion chamber and nozzle.
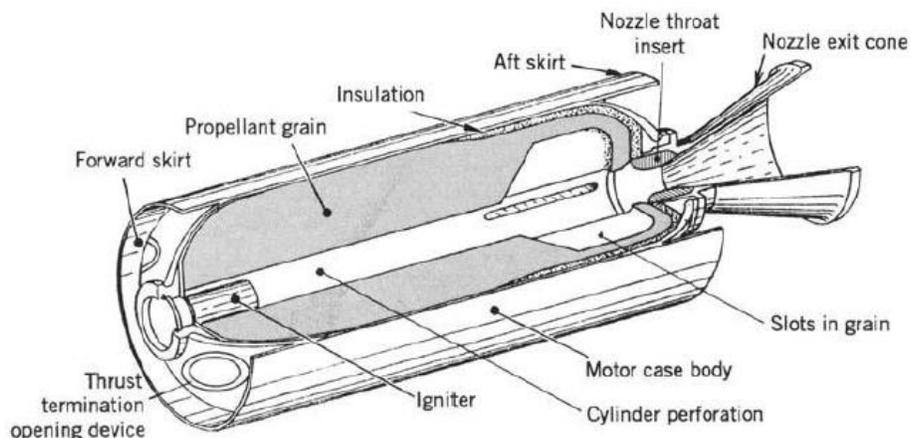


Figure 25 – Simplified Rocket Motor (19)

## 3.3.1. Thrust Profile

Typically, the missile propulsion is modeled using the thrust profile. Figure 26 shows the thrust profile of the booster used in this paper. The thrust profile is a plot of the magnitude of thrust force $F^T$ in (dN) with time.
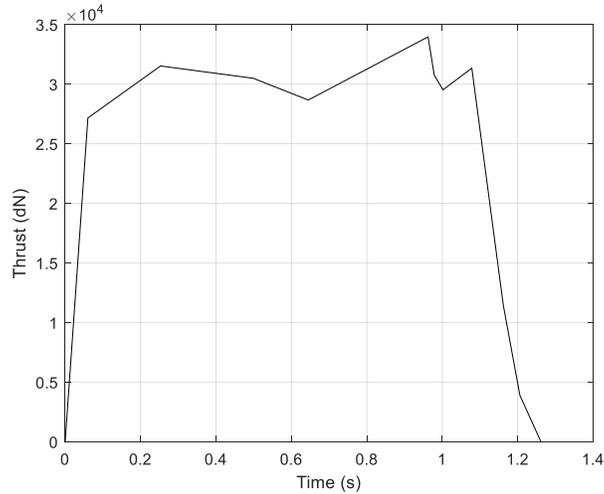


Figure 26 - Booster Thrust Profile

By using the thrust profile, the total impulse $I_t$ can be calculated as shown in Equation 37. The total impulse is the amount of thrust energy in the booster.

| | |
|---|---|
| $$I_t = \int_0^t F^T dt \sim F^T t$$ | Equation 37 (19) |

Another important parameter is the specific impulse $I_s$ shown in Equation 38. The specific impulse is the measurement of the quality of the propellant. Here $m_p$ and $w_p$ is the mass and weight of propellent respectively.

| | |
|---|---|
| $$I_s = \frac{I_t}{m_p g} = \frac{I_t}{w_p}$$ $$I_s = \frac{F}{\dot{m}_p g} = \frac{F}{\dot{w}_p}$$ | Equation 38 (19) |

## 3.3.2. Thrust Forces, Moments, and Misalignment

Ideally, the booster acting point $X_T$ should be aligned perfectly with the missile's center of gravity $X_{cg}$. However, in reality it is virtually impossible to align the two points. That is because $X_{cg}$ is changing throughout the flight as the propellent is being consumed. This in essence caused unwanted forces and moments on the missile. Hence, it is important to consider them in modeling the booster. Figure 27 show the representation of thrust misalignment. Here, $\epsilon_y$ and $\epsilon_z$ are the misalignment angles.
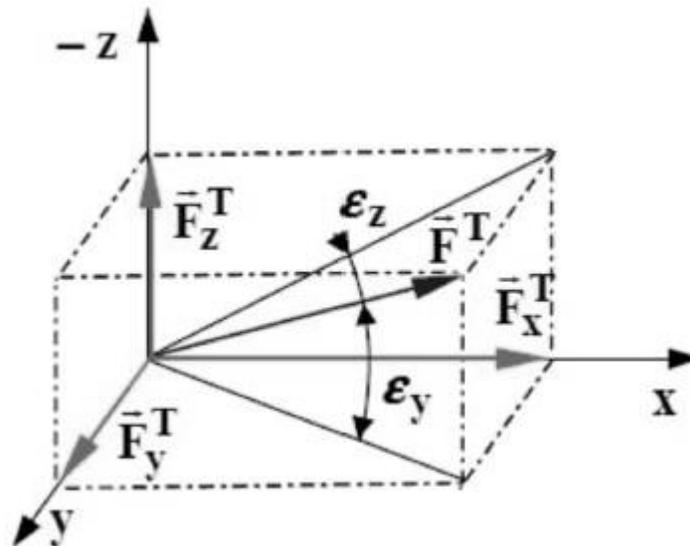


*Figure 27 – Thrust Misalignment (20)*

By analyzing Figure 27, Equation 39 can be developed. Here, the thrust force is broken down into smaller thrust components while taking misalignment effects into consideration.

| | |
|---|---|
| $F_x^T = F^T \sqrt{1 - \sin^2 \epsilon_y - \sin^2 \epsilon_z}$ | Equation 39 (20) |
| $F_y^T = F^T \sin\epsilon_z$ | |
| $F_z^T = -F^T \sin\epsilon_y$ | |
| $F^T = \dot{m} I_{sp} - A_e p_a$ | |

In addition to force misalignment effects, there are moment effects due to misalignments $M_T$. Similar to the force case, the thrust moment can be broken down into components that takes into consideration the misalignment effects as shown in Equation 40.

| | |
|---|---|
| $M_x^T = 0$ | Equation 40 |
| $M_y^T = F_z^T(X_T - X_{cg})$ | (20) |
| $M_y^T = -F_y^T(X_T - X_{cg})$ | |

## 3.4. Aerodynamic Model

The missile aerodynamics can be modeled by observing the behavior of the missile due to the changes in conditions. For instance, the pitch moment of the missile can be observed during the change of angle of attack or velocity. The observed behavior can be written as an equation with a set of aerodynamic derivatives. The observed behavior can be aerodynamics forces or moments. It is important to note that the aerodynamics forces and moments act on the center of pressure (CP) as shown in Figure 28.
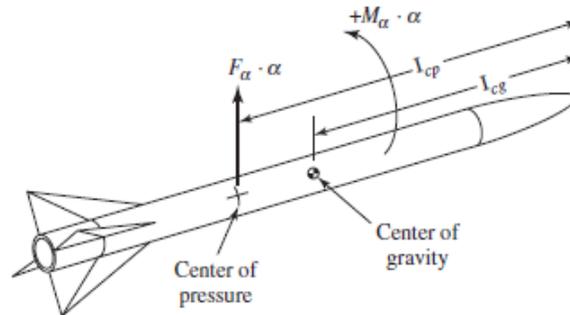


Figure 28 - Missile Center of Pressure (14)

## 3.4.1. Aerodynamics Derivatives

The equation made from aerodynamic derivatives can be linear or non-linear. The linear equations are good for initial approximation. However, since accuracy is a priority, non-linear equations are preferred. Hence, Equation 41 shows the aerodynamics equations and derivatives used to model the missile in this thesis. It is important to note that the equations provided are of general form in order to be comprehensive.

$$C_{ld} = C_{lp}\frac{pd}{V}, \ C_{nd} = C_{mq}\frac{rd}{V}, \ C_{md} = C_{mq}\frac{qd}{V}$$

$$\boldsymbol{\alpha}^2 = \alpha^2 + \beta^2$$

$$C_{m0} = C_{m00} + C_{m\alpha^2 0}\boldsymbol{\alpha}^2$$

$$C_{n0} = C_{n00} + C_{n\alpha^2 0}\boldsymbol{\alpha}^2$$

$$C_{N0} = C_{N00} + C_{N\alpha^2 0}\boldsymbol{\alpha}^2$$

$$C_{Y0} = C_{Y00} + C_{Y\alpha^2 0}\boldsymbol{\alpha}^2$$

$$C_{l0} = C_{l00} + C_{l\alpha^2}\boldsymbol{\alpha}^2$$

$$\Delta C_m = \left[\Delta C_{m\alpha} + \Delta C_{m\alpha^3}\left(\alpha^2 + \beta^2\right)\right]\alpha + \left[\Delta C_{m\delta} + \Delta C_{m\alpha^2\delta}\left(\alpha^2 + \beta^2\right)\right]\delta_m$$

$$\Delta C_N = \left[\Delta C_{N\alpha} + \Delta C_{N\alpha^3}\left(\alpha^2 + \beta^2\right)\right]\alpha + \left[\Delta C_{N\delta} + \Delta C_{N\alpha^2\delta}\left(\alpha^2 + \beta^2\right)\right]\delta_m$$

$$C_A = C_{A00} + C_{A\delta^2}\left(\delta_n^2 + \delta_m^2\right) + C_{A\delta_\ell}\delta_\ell^2 + C_{A\alpha^2}\left(\alpha^2 + \beta^2\right) + C_{A\alpha\delta}\left(\alpha\delta_m - \beta\delta_n\right)$$

$$C_N = \left[C_{N\alpha} + C_{N\alpha^3}\left(\alpha^2 + \beta^2\right)\right]\alpha + \left[C_{N\delta} + C_{N\alpha^2\delta}\alpha^2 + C_{Y\alpha^2\delta}\beta^2\right]\delta_m -$$
$$-\left(C_{N\alpha^2\delta} - C_{Y\alpha^2\delta}\right)\alpha\beta\delta_n + \Delta C_N + C_{N0}$$

$$C_Y = -\left[C_{N\alpha} + C_{N\alpha^3}\left(\alpha^2 + \beta^2\right)\right]\beta + \left[C_{N\delta} + C_{N\alpha^2\delta}\beta^2 + C_{Y\alpha^2\delta}\alpha^2\right]\delta_n -$$
$$-\left(C_{N\alpha^2\delta} - C_{Y\alpha^2\delta}\right)\alpha\beta\delta_m + C_{Y0}$$

$$C_\ell = \left[C_{\ell\delta_\ell} + C_{\ell\alpha^2\delta_\ell}\left(\alpha^2 + \beta^2\right)\right]\delta_\ell + C_{\ell\alpha\delta}\left(\beta\delta_m + \alpha\delta_n\right) + C_{l0} + C_{ld}$$

$$C_n = -\left[C_{m\alpha} + C_{m\alpha^3}\left(\alpha^2 + \beta^2\right)\right]\beta + \left[C_{m\delta} + C_{m\alpha^2\delta}\beta^2 + C_{n\alpha^2\delta}\alpha^2\right]\delta_n -$$
$$-\left(C_{m\alpha^2\delta} - C_{n\alpha^2\delta}\right)\alpha\beta\delta_m + C_{n0} + C_{nd}$$

$$C_m = \left[C_{m\alpha} + C_{m\alpha^3}\left(\alpha^2 + \beta^2\right)\right]\alpha + \left[C_{m\delta} + C_{m\alpha^2\delta}\alpha^2 + C_{n\alpha^2\delta}\beta^2\right]\delta_m$$
$$-\left(C_{m\alpha^2\delta} - C_{n\alpha^2\delta}\right)\alpha\beta\delta_n + \Delta C_m + C_{m0} + C_{md}$$

Equation 41

(21)

## 3.4.2. Aerodynamics Forces

Once the equations made from aerodynamic derivatives result in aerodynamic force coefficients. These coefficients are used to calculate the aerodynamic forces using Equation 42.

| | |
|---|---|
| $$F_X = -F_{Axial} = \frac{1}{2}\rho V^2 S C_A$$ | |
| $$F_Y = F_{Side} = \frac{1}{2}\rho V^2 S C_Y$$ | Equation 42 (14) |
| $$F_Z = -F_{Normal} = \frac{1}{2}\rho V^2 S C_N$$ | |

## 3.4.3. Aerodynamics Moments

Similar to the aerodynamic force, the aerodynamic derivatives equations can result in aerodynamic moments coefficients. These coefficients are used to calculate aerodynamic moment using Equation 43.

| | |
|---|---|
| $$L = \frac{1}{2}\rho V^2 S C_l d$$ | |
| $$M = \frac{1}{2}\rho V^2 S C_m d$$ | Equation 43 (14) |
| $$N = \frac{1}{2}\rho V^2 S C_n d$$ | |

## 3.4.4. Aerodynamics Transfer Functions

Aerodynamic transfer functions allow for the missile behavior to be simplified. This simplification allows for quick analysis of the missile. In addition, the linear time invariant model serves as a reference for the missile control loop. This allows for the missile autopilot to be design and tested before implementing it with the complete dynamics.

### 3.4.4.1. Dynamic Variables

In order to define the aerodynamic transfer function, a set of dynamic variables, shown in Equation 44, should be defined. These variables are used to simplify the complex representation of the transfer function.

| | |
|---|---|
| $$z_\eta = \frac{-QS.C_{N\eta}}{m}$$ $$m_w = \frac{QSl}{J_y} \cdot \frac{l}{U} \cdot \left(C_{m\alpha} + \frac{l}{V} \cdot C_{m\dot{\alpha}} z_w\right)$$ $$m_\eta = \frac{QSl}{J_y} \cdot (C_{m\eta} - \frac{l}{U^2} \cdot C_{m\dot{\alpha}} z_\eta)$$ | Equation 44 (22) |

### 3.4.4.2. Control Variables

There are two important control variables for transfer function which as natural frequency $\omega_n^2$ and damping factor $\zeta_n$ and shown in Equation 45.

| | |
|---|---|
| $$\omega_n^2 = -(m_w U - z_w m_q)$$ $$\zeta_n = -\frac{m_q + z_w}{2\omega_n}$$ | Equation 45 (22) |

### 3.4.4.3. Roll Transfer Function

The roll factor of reinforcement $K_\phi$ and time constant $T_\phi$ in Equation 46 are used to simplify the roll transfer function.

| | |
|---|---|
| $$K_\phi = -\frac{l_\xi}{l_p}$$ $$T_\phi = -\frac{1}{l_p}$$ | Equation 46 (22) |

Hence, with all the parameters at hand. The roll transfer function can be represented as shown in Equation 47.

| | |
|---|---|
| $$Roll\ Rate\ TF: \frac{p(s)}{\xi(s)} = \frac{K_\phi}{T_\phi s + 1}$$ | Equation 47 (22) |
| $$Roll\ Angle\ TF: \frac{\phi(s)}{\xi(s)} = \frac{1}{s}\frac{p(s)}{\xi(s)} = \frac{K_\phi}{T_\phi s + 1}$$ | |

### 3.4.4.4. Pitch or Yaw Transfer Function

The pitch or yaw factor of reinforcement $K_q$ and time constant $T_a$ in Equation 48 are used to simplify the pitch or yaw transfer function.

| | |
|---|---|
| $$K_q = \frac{z_\eta m_w - z_w m_\eta}{\omega_n^2}$$ $$T_a = \frac{m_\eta}{z_\eta m_w - z_w m_\eta}$$ | Equation 48 <br><br> (22) |

Similarly, with the required parameters, the pitch or yaw transfer function can be represented in Equation 49.

| | |
|---|---|
| $Pitch\ or\ Yaw\ Rate\ TF$: $\dfrac{q(s)}{\eta(s)} = \dfrac{\omega_n^2 K_q(T_q s + 1)}{s^2 + 2\zeta_n \omega_n s + \omega_n^2}$ | Equation 49 <br><br> (22) |
| $Pitch\ or\ Yaw\ Angle\ TF$: $\dfrac{\theta(s)}{\eta(s)} = \dfrac{1}{s}\dfrac{q(s)}{\eta(s)} = \dfrac{\omega_n^2 K_q(T_q s + 1)}{s(s^2 + 2\zeta_n \omega_n s + \omega_n^2)}$ | |

### 3.4.4.5. Normal Acceleration Transfer Function

Although contrary to intuition, the pitch or yaw autopilot in this thesis regulates the normal acceleration for lateral control. Hence, the normal acceleration transfer function is required. Equation 50 shows the reinforcement factor and time constant for the angle of attack.

| | |
|---|---|
| $$K_\alpha = \frac{1}{U} \cdot \frac{m_\eta U - m_q z_\eta}{\omega_n^2}$$ $$T_\alpha = \frac{z_\eta}{m_\eta U - m_q z_\eta}$$ | Equation 50 <br><br> (22) |

In addition, Equation 51 shows the time constant and damping ratio for the normal acceleration.

| | |
|---|---|
| $$T_\gamma = \sqrt{\frac{K_\alpha}{K_q} \cdot T_\alpha}$$ $$\zeta_\gamma = \frac{1}{2T_\gamma}\left(\frac{K_\alpha}{K_q} - T_\alpha\right)$$ | Equation 51 <br><br> (22) |

Hence, with all the available parameters, the normal force transfer function can be represented as shown in Equation 52.

| | |
|---|---|
| *Normal Acceleration TF*: $\dfrac{a_z(s)}{\eta(s)} = \dfrac{U\omega_n^2 K_q (T_\gamma^2 s^2 + 2\zeta_\gamma T_\gamma s + 1)}{s^2 + 2\zeta_n \omega_n s + \omega_n^2}$ | Equation 52 |
| *Normal Load TF*: $\dfrac{n_z(s)}{\eta(s)} = \dfrac{U}{g}\dfrac{\omega_n^2 K_q (T_\gamma^2 s^2 + 2\zeta_\gamma T_\gamma s + 1)}{s^2 + 2\zeta_n \omega_n s + \omega_n^2}$ | (22) |

## 3.5. Actuator Model

In order to control the missile's canards, there has to be an actuating system. Therefore, it is necessary to model the actuating system. To reduce the complexity of the system, the actuator system is modeled using a first order transfer function shown in Equation 53. Here, $T_a$ is the actuator time constant.

| | |
|---|---|
| $$-\dfrac{1}{T_a s + 1}$$ | Equation 53 |
| | (23) |

Since the missile has four canards, the deflection convention needs to be agreed upon. Figure 29 shows the deflection convention in this thesis. Here, four canards are deflected for roll and two canards for pitch or yaw respectively. In addition, Equation 54 shows the mathematical formula to calculate the total deflection of roll $\delta_p$, pitch $\delta_q$, and yaw $\delta_r$.
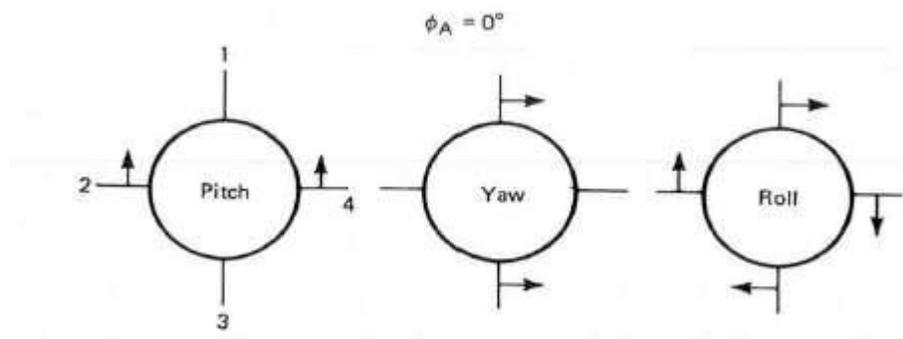


Figure 29 - Canard Deflection Convention (24)

| | |
|---|---|
| $$\delta_p = \frac{\delta_1 + \delta_2 + \delta_3 + \delta_4}{4}$$ $$\delta_q = \frac{\delta_2 - \delta_4}{2}$$ $$\delta_r = \frac{\delta_1 - \delta_3}{2}$$ | Equation 54 (24) |

# 4. Autopilot Design

## 4.1. Roll Autopilot

As implied by the name, the function of the roll autopilot is to stabilize the roll angle of the missile. This is the most important missile autopilot despite the fact that it is the simplest. That is because without the stability of the roll angle, the pitch and yaw planes will be off. Hence, the command received on those autopilots will be operating at an incorrect orientation.

### 4.1.1. Roll Autopilot Loop

The roll autopilot used in this missile is a modified proportional-derivative controller design. Figure 30 shows the autopilot design where $T_a$ is the actuator time constant, $G_A$ is the roll rate transfer function from Equation 47. The proportional gain of the loop is $K$ and the derivative gain is $G_R$.
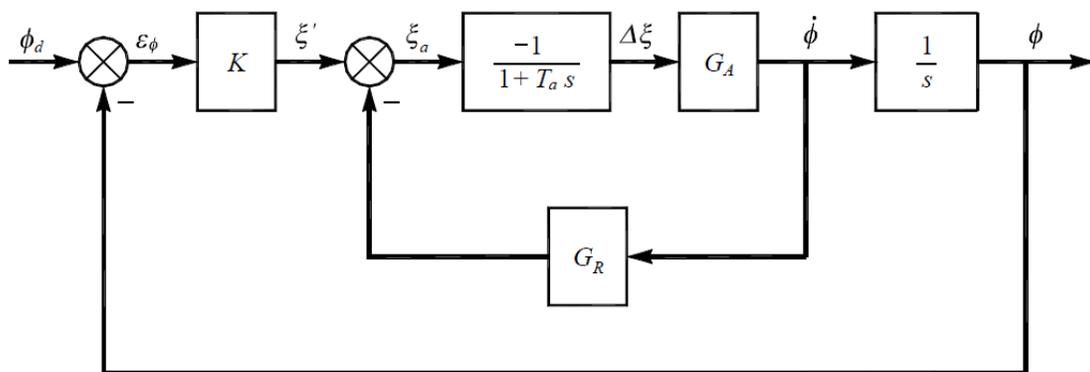


Figure 30 - Roll Autopilot (25)

## 4.1.2. Roll Gain Calculation

The roll gains for the autopilot was tuned manually using Ziegler-Nichols method. The method works by first setting all gains to zero except $K_p$. Then, gradually increase $K_p$ until the response starts oscillating. This value of $K_p$ is now set as $K_c$. The oscillation period of the response is $T_c$. By using the rules provided in Table 2, the PD controller gains can be calculated. It is important to note that since the PD controller rules is not in the table, the PID rules is used to calculate the gains while ignoring the integral gain.

| Controller Type | $K_p$ | $K_i$ | $K_d$ |
|:---:|:---:|:---:|:---:|
| P | $0.5K_c$ | | |
| PI | $0.4K_c$ | $0.8T_c$ | |
| PID | $0.6K_c$ | $0.5T_c$ | $0.125T_c$ |

Table 2 - Ziegler-Nichols Method Rules (26)

## 4.2. Pitch and Yaw Autopilot

The pitch and yaw autopilots are used to stabilize and control the missile during its flight. The pitch and yaw autopilots are identical because the missile in this thesis is cruciform. This means that the pitch and yaw dynamics are identical. Since its easier to adjust the acceleration to obtain the desired position, the normal acceleration transfer function from Equation 52 is used.

## 4.2.1. Pitch or Yaw Autopilot Loop

The pitch or yaw autopilot loop is a modified version of a proportional-integral-derivative controller. Figure 31 shows the autopilot loop design. Here, $G_{A_1}$ is the transfer function of the derivative of the flight path angle. In addition, $G_{A_2}$ is the conversion from derivative of the flight path angle to pitch rate transfer function as presented in Equation 55. Lastly, in order to simplify the autopilot, a is set to zero.

$$G_{A_1} = \frac{\dot{\gamma}}{\eta} = \frac{\omega_n^2 K_q}{s^2 + 2\zeta_n \omega_n s + \omega_n^2}$$

|  | Equation 55 |
|---|---|
| $$G_{A_2} = \frac{q}{\eta} = G_{A_1}(1 + T_q s)$$ | (23) |

It is important to note that $K = K_a G_N U_0$ is the proportional gain, $GR = K_a G_R$ is the differential gain, and $G_N$ is the integral gain. In addition, $U_0$ is the velocity of the missile.
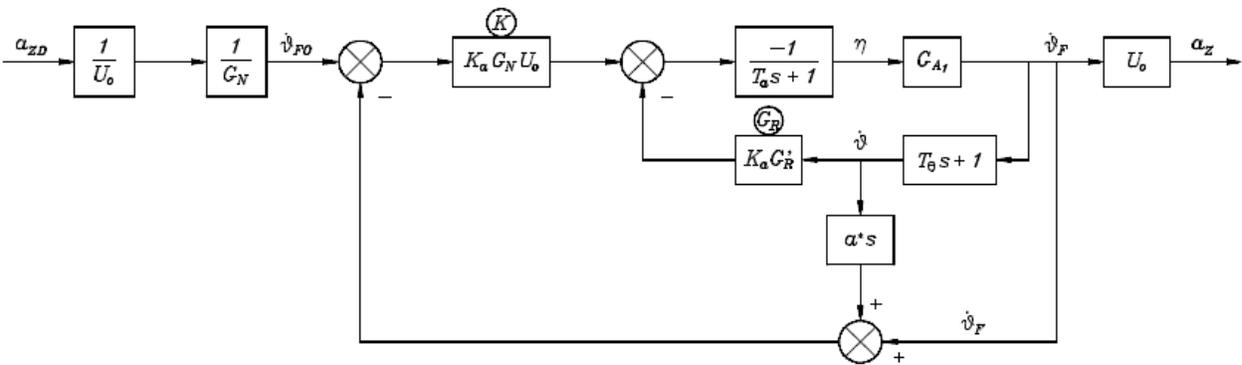


Figure 31 – Pitch or Yaw Autopilot (23)

## 4.2.2. Pitch or Yaw Gain Calculation

The pitch or yaw autopilot can be evaluated analytically. By transforming the autopilot loop and solving for the gain, the resultant gain equations are presented in Equation 56.

$$\omega_0 = \frac{1}{1.75}\left(\frac{1}{T_a} + 2\zeta_n \omega_n\right)$$

$$G_R = \frac{1}{-K_q}\frac{T_a}{T_q}\left(\left(\frac{\omega_0}{\omega_n}\right)^2\left(2.15 - \frac{2\zeta_n}{T_a \omega_0}\frac{\omega_n}{\omega_0}\right) - 1\right)$$

|  | Equation 56 |
|---|---|
| $$K = -\frac{1}{K_q}\left(\omega_0 T_a\left(\frac{\omega_0}{\omega_n}\right)^2 - 1\right) - G_R$$ | (23) |

$$G_N = -K\frac{K_q}{1 - K_q(G_R + K)}$$

## 4.3. Neural Network Autopilot

The purpose of the neural network autopilot (NNA) is to stabilize and control the missile during its flight. This is achieved by designing two unique neural networks that mimic the behavior of the traditional missile autopilot. One autopilot design is responsible for roll control and the other is for lateral pitch or yaw control. It is important to note that although the lateral design can be used for pitch or yaw control, the trainings and implementations are done separately.

### 4.3.1. Roll Neural Network Autopilot Algorithm

The Roll Neural Network Autopilot (RNNA) algorithm is shown in Figure 32. Here, the roll angle $\phi$ is subtracted from the Roll Demand $Roll_{Dem}$ of zero. The resultant can be denoted as $\Delta\phi$. Thereby, it creates the roll angle error. In addition, the roll angle error as well as the roll rate $P$ are inputted into the RNNA where the roll command $Roll_{Cmd}$ is calculated.
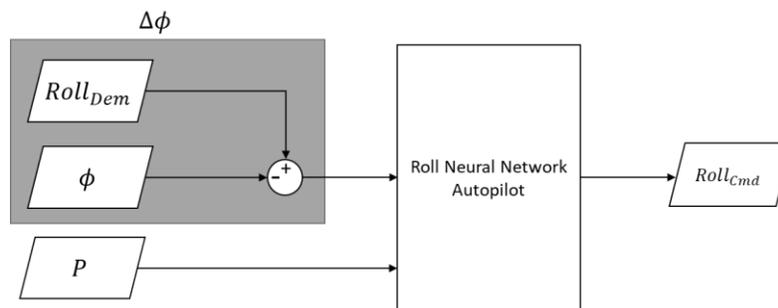


Figure 32 – Roll Neural Network Autopilot Algorithm

### 4.3.2. RNNA Design

The design of the RNNA is a traditional feedforward network shown in Figure 33. The input neuron consists of roll angle error and roll rate. The output neuron consists of the roll command. The number of hidden layers has been varied during the training process where 10 layers delivered the optimum result. The length of each hidden layers as well as the connections between the layers, hence the weight and biases, are also varied during the training to obtain the optimum result.
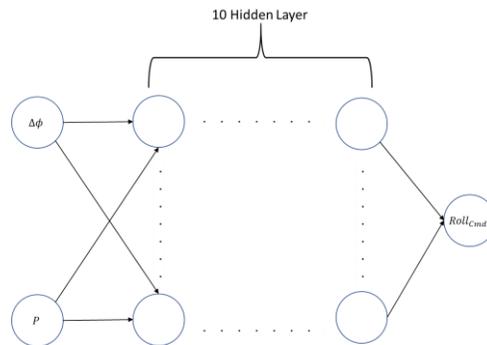
Figure 33 – Roll Neural Network Autopilot Design

## 4.3.3. RNNA Training Environment

In order to train the RNNA using NEAT Training method, the missile environment simulation must be prepared. Although traditional control missile dynamics can be simplified with transfer functions, the generalization nature of neural network can present a challenge. That is because the performance of the RNNA is affected by the time variant and non-linear missile dynamics. Hence, the full simulation should be used. Nevertheless, in order to simplify the training, some simulation parameters were fixed. The RNNA simulation is shown in Figure 34. Here, the RNNA algorithm commands the Actuator Systems which consists of all the four actuators. However, the pitch and yaw commands are set to zero. In addition, the aerodynamics assumes that the center of mass, side slip angle, velocity, and altitude are constant. Additionally, the missile body assumes that the inertia and the mass is constant. Moreover, the gravity is assumed to be zero as well. The simulation is set to run for a maximum time, 10s in this case, and the roll demand is an initial roll angle of 20degs and step function to 0deg at 1 second. The velocity is set at 0.5 Mach. In addition, the training session is run for 48 hours. This is to ensure that there is sufficient time for the optimal solution to be found.
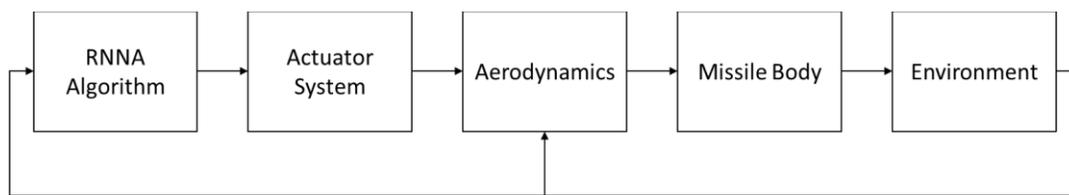

Figure 34 – Roll Neural Network Autopilot Simulation

### 4.3.4. RNNA Training Genetic Algorithm Parameters

The GA fitness function $Fitness_{RNNA}$ shown in Equation 57 is used to evaluate the quality of RNNA. $Fitness_{RNNA}$ consists of the Mean Square Error of the Roll Demand $Roll_{Dem}$ and Roll Response $Roll_{Res}$. This forces the algorithm to match $Roll_{Res}$ to $Roll_{Dem}$ in order to obtain minimum MSE. In addition, the overshoot term minimizes the overshoot of the RNNA. Moreover, there are two terms that are used to quickly ignore bad RNNA results. The roll angle error term $Angle_e$ is zero except when the missile roll angle is too high in the simulation during training. The value of $Angle_e$ is $200$ in this experiment. Additionally, the zero error term $Zero_e$ is zero except when the output of the RNNA is not zero when all the inputs are zero. The value of $Zero_e$ is 500 in this experiment. This is to minimize the neural network static error when all inputs are zero.

| $Fitness_{RNNA} = \text{MSE}(Roll_{Dem}, Roll_{Res}) + Overshoot^2 + Angle_e + Zero_e$ | Equation 57 |
|---|---|

### 4.3.5. Lateral Neural Network Autopilot Algorithm

The Lateral Neural Network Autopilot (LNNA) algorithm is shown in Figure 35. Here, the design of the lateral acceleration $Acc_{Lat}$ is subtracted from the acceleration demand $Acc_{Dem}$. The resultant can be denoted as $\Delta Acc$. Thereby, it creates the acceleration error. In addition, the acceleration error, the lateral rate, and Mach number are inputted into the LNNA where the acceleration command $Lateral_{Cmd}$ is calculated.
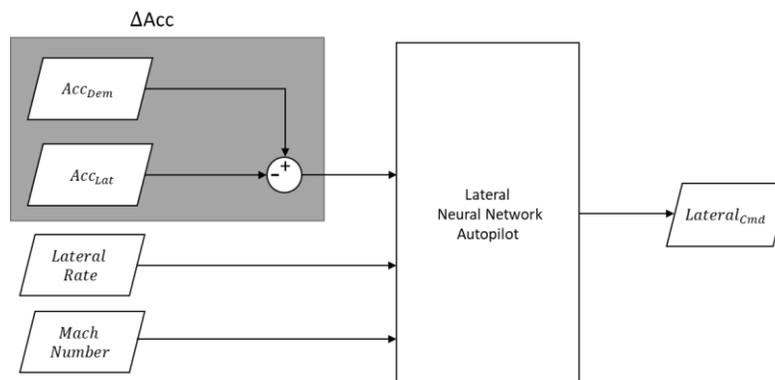


Figure 35 - Lateral Neural Network Autopilot Algorithm

## 4.3.6. LNNA Design

The design of the LNNA is a traditional feedforward network shown in Figure 36. The input neurons consist of the acceleration error, the lateral rate, and the Mach number. The output neuron consists of the lateral acceleration command. The number of hidden layers has been varied during the training process where 15 layers delivered the optimum result. The length of each hidden layers as well as the connections between the layers, hence the weight and biases, are also varied during the training to obtain the optimum result.
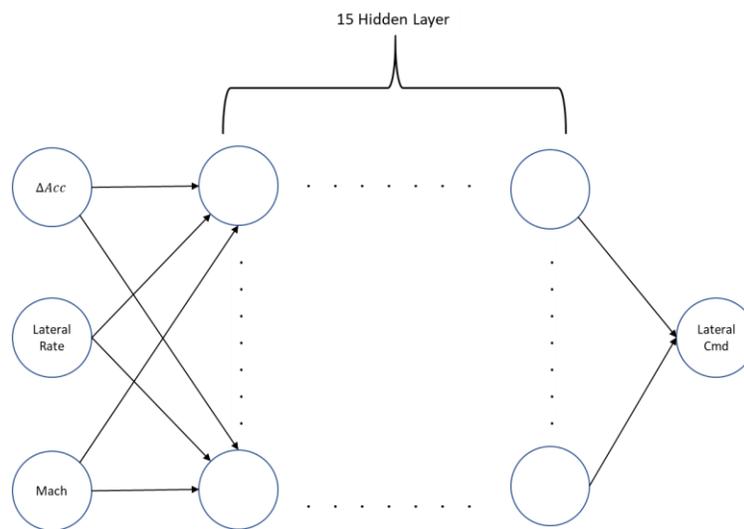


Figure 36 – Lateral Neural Network Autopilot Design

## 4.3.7. LNNA Training Environment

In order to train the LNNA using NEAT Training method, the missile environment simulation must be prepared. Similar to the RNNA, the full simulation should be used. Nevertheless, in order to simplify the training, some simulation parameters were fixed for either pitch or yaw simulations. The LNNA simulation is shown in Figure 37. Although the RNNA training runs the simulation once, the LNNA requires the simulation to be run multiple times. That is because the Mach number of the simulation is changed for each time it is run. This is to reduce the generalization problem of neural networks. This loop is seen in Figure 37 where the Mach number is varied from 0.1 to 1.3 with a step of 0.2.

Furthermore, the Actuator Systems consists of all the four actuators. However, for the pitch case, the yaw and roll commands are set to zero. In addition, for the yaw case, the pitch and roll commands are set to zero. In addition, the aerodynamics assumes the center of mass and altitude are constant. Unlike the RNNA, the side slip angle and velocity vary. The side slip angle changes dynamically with the simulation. On the other hand, the velocity is constant for each simulation run. Additionally, the missile body assumes that the inertia and the mass are constant. Moreover, the gravity is assumed to be zero as well. The simulation is set to run for a maximum time, 10s in this case. The initial pitch or yaw acceleration is zero. However, the acceleration demand slopes to -12 at 1 sec. Then, it slopes to 0 at 4sec. Lastly, it slopes to 12 at 7 seconds. This behavior is to account for the missile acceleration needs. Moreover, the training session is run for 4 days. This is to ensure that there is sufficient time for the optimal solution to be found.
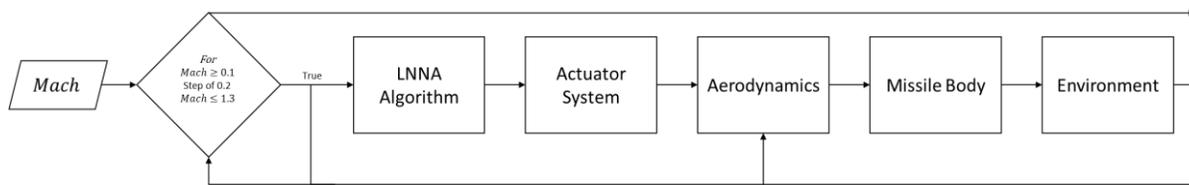


Figure 37 - Lateral Neural Network Autopilot Simulation

## 4.3.8.  LNNA Training Genetic Algorithm Parameters

The GA fitness function $Fitness_{LNNA}$ shown in Equation 58 is used to evaluate the quality of LNNA. $Fitness_{LNNA}$ consists of the summation of all the Mean Square Error of the Lateral Demand $Lat_{Dem}$ and Lateral Response $Lateral_{Res}$ as well as the lateral angle error $Angle_e$. The MSE forces the algorithm to match $Lat_{Res}$ to $Lat_{Dem}$ in order to obtain minimum MSE. On the other hand, $Angle_e$ is used to quickly ignore bad LNNA results. The lateral angle error term $Angle_e$ is zero except when the missile lateral angle is too high in the simulation during training. The value of $Angle_e$ is 200 in this experiment. This sum reduces the generalization problem and it allows for a global quantification of the performance of the LNNA.

| $Fitness_{LNNA} = \sum_{t=0}^{n}(\text{MSE}(Lat_{Dem}, Lat_{Res}) + Angle_e)_t$ | Equation 58 |
|---|---|

# 5. Guidance Design

A guidance algorithm is needed to calculate the command given to the autopilot to perform the required maneuver to reach the target. There are many forms of guidance algorithm. Nevertheless, this thesis discusses the trajectory guidance and proportional guidance.

## 5.1. Trajectory Guidance

The trajectory guidance is used for the missile to maintain a constant trajectory parallel to the ground. Although the missile discussed in this thesis does not have sustainer, trajectory guidance can still be used for a short period. For instance, the missile can maintain a constant altitude when it is launched from air switching guidance algorithms. Figure 38 shows the missile or aircraft flying above the desired trajectory. By setting a reference points and geometrical relationship, the trajectory guidance algorithm can be derived as shown in Equation 59. This equation is valid when the reference point is perpendicular to the lateral acceleration $a_{S_{cmd}}$.
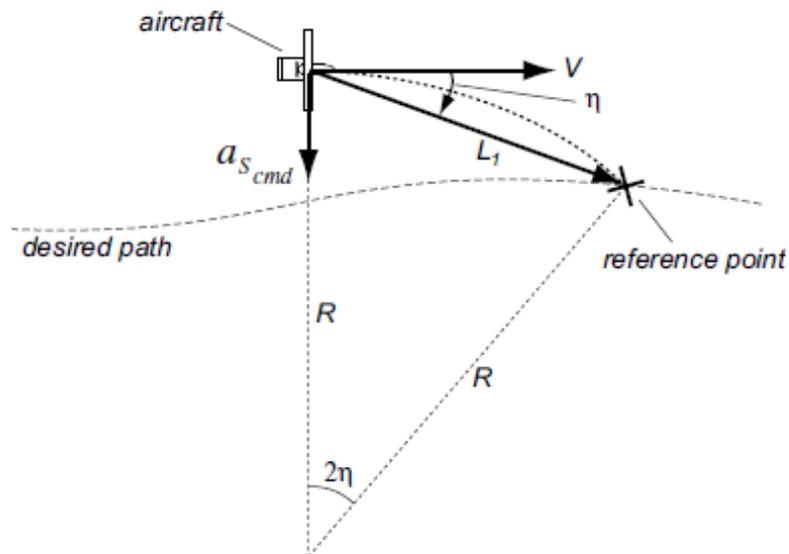


Figure 38 - Trajectory Guidance (27)

| | |
|---|---|
| $$a_{s_{cmd}} = 2\frac{V^2}{L_1}\sin\eta$$ | Equation 59 (27) |

Therefore, in order to account for the situation where the missile is not perpendicular to the references point as seen in Figure 39, a more general equation should be developed.
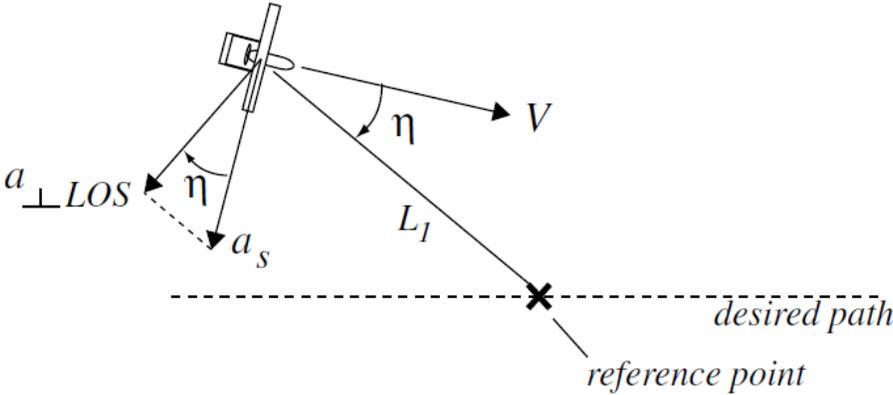


Figure 39 - Trajectory Guidance for Line of Sight (27)

By using the relation between line of sight acceleration $a_{\perp LOS}$ and lateral acceleration $a_s$, Equation 60 can be developed. Here, the equation accounts for the line of sight of the missile and the reference point does not need to be perpendicular to the lateral acceleration.

| | |
|---|---|
| $$a_{\perp LOS} = a_s \cos\eta$$ $$Since \ a_s = 2\frac{V^2}{L_1}\sin\eta$$ $$a_{\perp LOS} = 2\frac{V^2}{L_1}\sin\eta\cos\eta = 2\,(V\cos\eta)\left(\frac{V}{L_1}\sin\eta\right)$$ | Equation 60 (27) |

## 5.2. Proportional Navigation

The Proportional Navigation (PN) guidance algorithm is used to guide the missile towards the target during the terminal stage. Here, the missile uses the line of sight between itself and the target to calculate the acceleration needed to reach the target. Figure 40 shows the missile, target, and geometry properties relating the two entities.
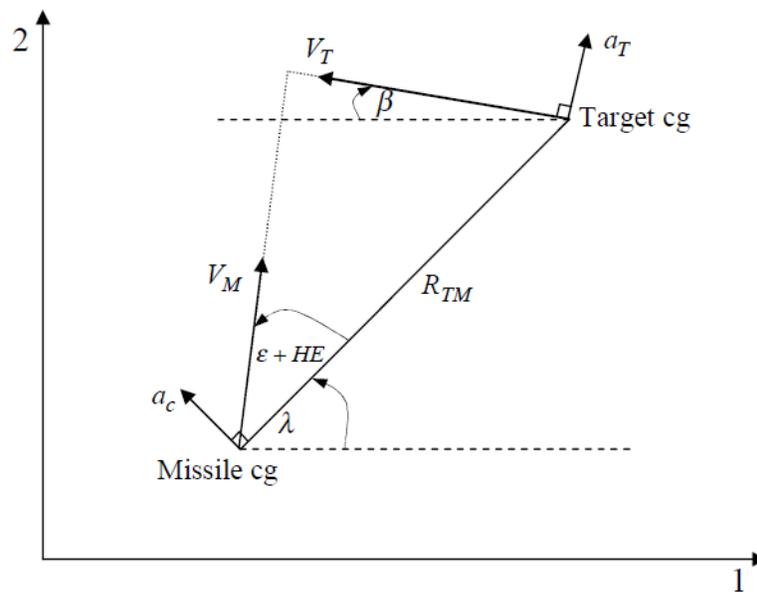


Figure 40 – Proportional Navigation (28)

The PN algorithm is shown in Equation 61. Here, the lateral acceleration $a_c$ is calculated with the closing velocity $V_c$, the line of sign rate $\dot{\lambda}_m$, and the navigation constant $N'$ that usually ranges between 3 and 5.

| | Equation 61 |
|---|---|
| $$a_c = N'V_c\dot{\lambda}_m$$ | (28) |

Although as simple as it seems, the parameters of the PN algorithm are not readily available. This means that the parameters need to be estimated in order to calculate the required lateral acceleration. By using the approximation in Equation 62, the missing parameters can be calculated. It is important to keep in mind the effects resulted from the approximation of the PN algorithm parameters.

$$R_{TM1} = R_{T1} - R_{M1}$$

$$R_{TM2} = R_{T2} - R_{M2}$$

$$R_{TM}^2 = \sqrt{R_{TM1}^2 + R_{TM2}^2}$$

$$V_{TM1} = V_{T1} - V_{M1}$$

$$V_{TM2} = V_{T2} - V_{M2}$$

$$\lambda = \tan^{-1}\frac{R_{TM2}}{R_{TM1}}$$

$$\dot{\lambda} = \left[\frac{R_{TM1}V_{TM2} - R_{TM2}V_{TM1}}{R_{TM}^2}\right]$$

$$V_c = \frac{-(R_{TM1}V_{TM1} + R_{TM2}V_{TM2})}{R_{TM}}$$

Equation 62

(28)

## 5.3. Terminal Guidance Neural Network

The purpose of the Terminal Guidance Neural Network (TGNN) is to guide the missile towards the target during the terminal phase of the flight. This is achieved by designing a neural network that mimics and improves the functionality of terminal guidance.

### 5.3.1. TGNN Algorithm

The TGNN algorithm is shown in Figure 41. Here, the line of sight rate $\dot{\lambda}$ is subtracted from the demand line of sight rate of zero. The resultant can be denoted as $\Delta\dot{\lambda}$. Thereby, it creates the line of sight rate error. The error is fed into the TGNN where the acceleration command $A_c$ is calculated. The acceleration command can be fed into the lateral autopilot. It is important to note that the line of sight rate can be either read by seeker or calculated using Equation 62 as in this case.
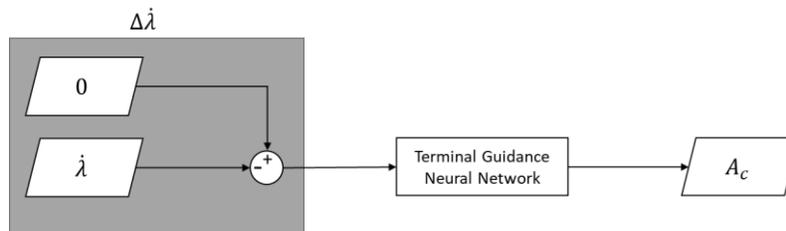


Figure 41 – Terminal Guidance Neural Network Algorithm

## 5.3.2. TGNN Design

The design of TGNN is a traditional feedforward network as shown in Figure 42. The input neuron consists of the line of sight rate error and the output neuron consists of the acceleration command. The output neuron consists of the acceleration command. The number of hidden layers has been varied during the training process where 10 layers delivered the optimum result. The length of each hidden layers as well as the connections between the layers, hence the weight and biases, are also varied during the training to obtain the optimum result.
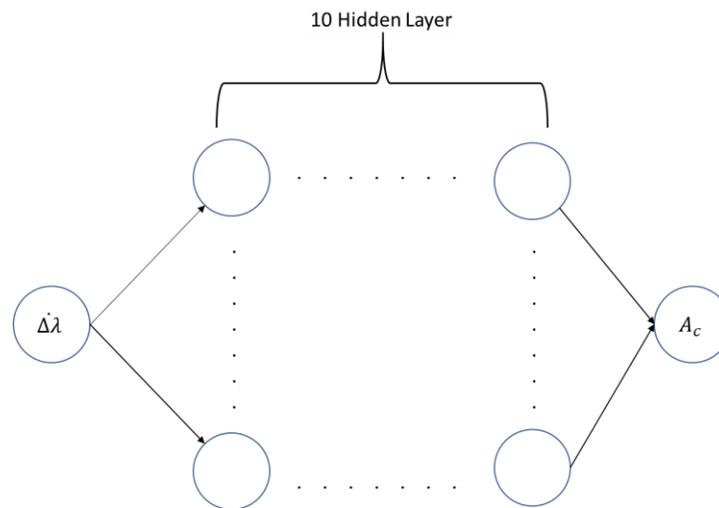


Figure 42 - Terminal Guidance Neural Network Design

## 5.3.3. TGNN Training Environment

In order to train the Terminal Guidance Neural Network using NEAT Training method, the missile environment simulation must be prepared. Figure 43 shows the simulation used to train the network. Here, the line of sight rate error is fed into the neural network. Then, the acceleration command is saturated with the missile lateral tolerance. The acceleration command is used to calculate the missile dynamics.

A simplified missile dynamics model is used in the training. The purpose of the simplification is to speed up the training process. A second order Pitch or Yaw Rate transfer function shown in Equation 49 is used. The

calculated pitch or yaw rate is used to calculate the angle of the missile. In addition, a constant magnitude of velocity, 200m/s, is used for simplification. The magnitude of velocity and missile angle are used to calculate the components of velocity. The components of the velocity are integrated to calculate the position. Both velocity and position are used to calculate the line of sight rate error.
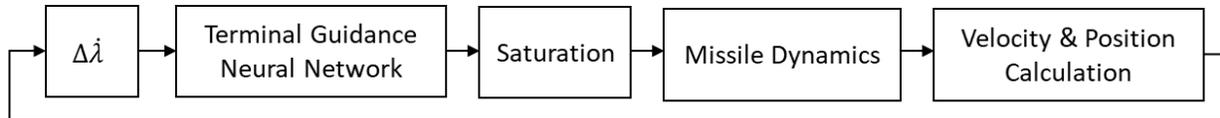


Figure 43 – Terminal Guidance Neural Network Simulation

The simulation is set to run for a maximum time, 100s in this case. This is to prevent infinite loop. In addition, the missile is fired from a constant position, xm = 0m and ym = 3000m, and 0 to a constant the target, xt =10km and yt = 0m. The initial launch angle of the missile is at 0 deg. This is to remove any errors due to the initial angle position. In addition, the training session is run for 24 hours. This is to ensure that there is sufficient time for the optimal solution to be found.

## 5.3.4. TGNN Training Genetic Algorithm Parameters

The GA fitness function $Fitness_{TGNN}$ shown in Equation 63 is used to evaluate the quality of the TGNN. $Fitness_{TGNN}$ consists of the Mean Square Error of the line of sight rate demand $\dot{\lambda}_{Dem}$ of zero and $\dot{\lambda}_{Res}$ response. This forces the algorithm to match $\dot{\lambda}_{Res}$ to $\dot{\lambda}_{Dem}$ in order to obtain minimum MSE. In addition, it includes the absolute value of the vertical and horizontal miss distance denoted with $|X|_{Miss}$ and $|Y|_{Miss}$ respectively. The MSE of the idea acceleration of zero $Az_{ideal}$ and the acceleration response $Az_{Res}$ are included. This minimizes the acceleration command generated by the TGNN to minimize lateral loading on the missile.

| | |
|---|---|
| $Fitness_{TGNN} = \text{MSE}(\dot{\lambda}_{Dem}, \dot{\lambda}_{Res}) + |X|_{Miss} + |Y|_{Miss} +$ $\text{MSE}(Az_{ideal}, Az_{Res}) + Alt_e + Angle_e + Range_e$ | Equation 63 |

Furthermore, there are a couple of terms that are used to quickly ignore bad TGNN results. The altitude error term $Alt_e$ is zero except when the missile flies too high in the simulation during training. The value of $Alt_e$ is set to be huge, $1 \times 10^4$ in this experiment. This signifies that the missile headed in the wrong direction. In addition, the angle error term $Angle_e$ is zero except when the missile angle is above or below $\pm 80$ degs. This is to prevent the missile from turning with too high of an angle. Lastly, the range error $Range_e$ is zero except when the range is negative. This is to prevent the missile from flying backwards.

# 6. Navigation Design

Missile navigation allows for the missile to locate itself and the target in space. The missile measures its linear acceleration as well as its rotational rate using the inertial measurement unit (IMU). The output of the IMU is then fed into the Inertial Navigation System (INS) algorithm, thereby calculating the location of the missile. In addition, in order to improve the accuracy of the INS algorithm, a GPS sensor is integrated with the INS algorithm.

## 6.1. Inertial Measurement Unit Modeling

At the core level, the function of the IMU is to read the acceleration and rotation rates of the missile. There are several sensor technologies that can accomplish this such as gimballed, strapdown, and laser IMUs. Each technology has its challenges.

### 6.1.1. Gimballed IMU

Figure 44 show the inner working of the gimballed IMU. Here, the entire platform is gyro stabilized. This means that they are rotating so far that it has the tendency to maintain its precession. The changes in the precession is measured to estimate the missile's change of orientation. The challenge with this technology is that it takes up space, the gyro precession drifts with time, and the risk of gimbal lock, which is when two gimbals axis overlap.
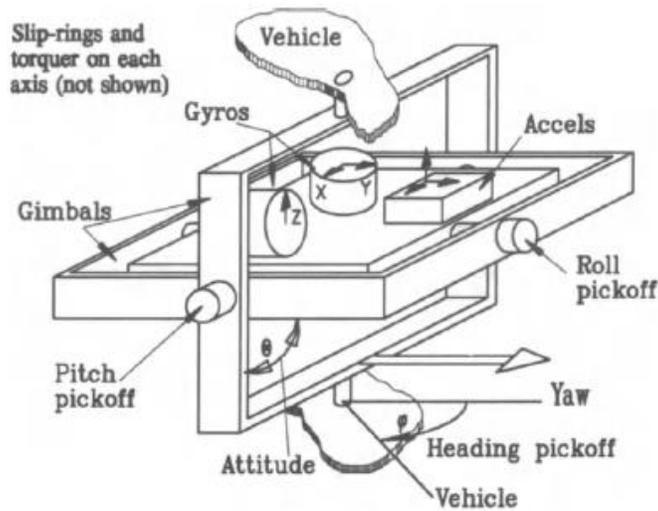
Figure 44 - Gimballed Gyro (29)

## 6.1.2. Strapdown IMU

The strapdown IMU is shown in Figure 45. Unlike the gyro IMU, there is no gimbaled platform. Instead, there is a set of electronic sensors that are strapped down to the missile body which measures the acceleration and rotation of the missile. The challenge with these sensors is the accuracy of the measurement. However, since they can be small and cheap to produce while providing a good enough accuracy, it is the most popular sensor in modern missiles. Hence, this sensor is assumed in this thesis.
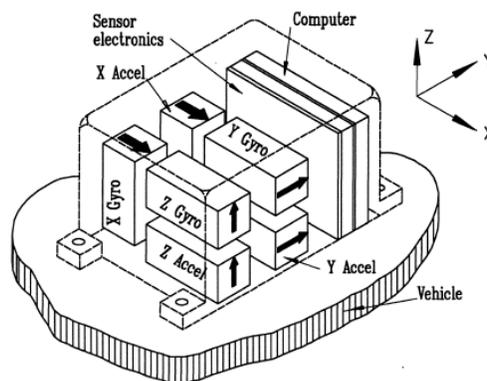


Figure 45 - Strapdown IMU (29)

## 6.1.3. Ring Laser IMU

When high accuracy is required, the ring laser IMU sensor can be used. Figure 46 shows the ring laser gyro. Here, the acceleration and rates are measured by the variation in the laser's paths. The challenges of this technology are that the sensor is complex and the cost of such sensor is high.
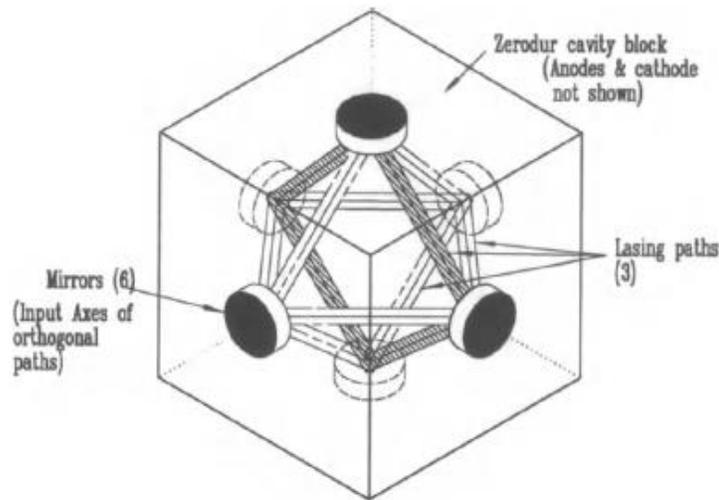


Figure 46 - Ring Laser Gyro (29)

## 6.1.4. IMU Errors

The IMU can be modeled by introducing a set of errors to the true missile acceleration and rotation rates. Equation 64 shows the IMU error model. Here, the $g_r$ is the rotation rate output of the IMU, $r$ is the true rotational rate, $c_r$ is the offset, $b_r$ is the bias, and $w_{gyro}$ is the white noise associated wit the gyro. Similarly, $a_{\ddot{x}}$ is the acceleration output of the IMU, $\ddot{x}$ is the true acceleration, $c_{\ddot{x}}$ is the offset, $b_{\ddot{x}}$ is the bias, $w_{accel}$ is the white noise. In addition, to the defined error, the sensitivity, quantization, saturation, and delay of the IMU error can be included to improve the accuracy of the IMU model.

| | |
|---|---|
| $g_r = r + c_r + b_r + w_{gyro}$ | Equation 64 |
| $a_{\ddot{x}} = \ddot{x} + c_{\ddot{x}} + b_{\ddot{x}} + w_{accel}$ | (30) |

## 6.2. Inertial Navigation Algorithm

This thesis uses the INS algorithm proposed by Salychev. The flow chart of the algorithm is shown in Figure 48 and Figure 49. The Salychev's INS algorithm addresses the errors associated with the IMU and compensates it. Hence, by using this algorithm, the position and orientation of the missile can be calculated more accurately.

### 6.2.1. Reading IMU Data

The output of the IMU data consists of three gyroscope data and three accelerometer data. A series of four or eight data in sequence is stored as it is needed for the algorithm. The data is read by two separate channels as shown in the flowchart.

### 6.2.2. Compensation of Gyro and Acceleration

By using the IMU gyro error in Equation 64, the read gyro and acceleration data can be corrected for the error. It is important to note that an accurate model for the offset, bias, and white noise is needed to ensure the optimal correction. In the case where an accurate model is not feasible, then this step of the algorithm can be ignored as an incorrect model might decrease the accuracy.

### 6.2.3. Calculation of Angle Increment

Once the compensation is completed, the angular rate can be converted to angle increments using Equation 65. It is important to note that since computational speed is important in a missile, the integration can be simplified as a multiplication by sampling time $T_s$.

| | |
|---|---|
| $$\alpha = \int_{t_k}^{t_k+h_{N1}} \omega \, dt \sim \omega T_s$$ | Equation 65 (16) |

## 6.2.4. Calculation of Velocity Increment

Similarly, once the compensation is completed, the acceleration can be converted to velocity increments using Equation 66. A shorter form of the integration can be used here as well.

| | |
|---|---|
| $$\Delta W = \int_{t_k}^{t_k + h_{N1}} a \, dt \sim a T_s$$ | Equation 66 <br><br> (16) |

## 6.2.5. Coning Correction

Since the rotational vector $\dot{\phi}$ does not match the true rotation rate of the missile $\omega$ as discussed in section 3.2.5, the coning correction shown in Equation 67 is required.

| | |
|---|---|
| $$\Delta\phi = \begin{bmatrix} \Delta\phi_x \\ \Delta\phi_y \\ \Delta\phi z \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{4} \alpha_x(j) \\ \sum_{j=1}^{4} \alpha_y(j) \\ \sum_{j=1}^{4} \alpha_z(j) \end{bmatrix} + $$ $$\frac{2}{3}\left\{ P_1 \begin{bmatrix} \alpha_x(2) \\ \alpha_y(2) \\ \alpha_z(2) \end{bmatrix} + P_3 \begin{bmatrix} \alpha_x(4) \\ \alpha_y(4) \\ \alpha_z(4) \end{bmatrix} \right\} + $$ $$\frac{1}{2}(P_1 + P_2)\left\{ \begin{bmatrix} \alpha_x(3) \\ \alpha_y(3) \\ \alpha_z(3) \end{bmatrix} + \begin{bmatrix} \alpha_x(4) \\ \alpha_y(4) \\ \alpha_z(4) \end{bmatrix} \right\} + $$ $$\frac{1}{30}(P_1 - P_2)\left\{ \begin{bmatrix} \alpha_x(3) \\ \alpha_y(3) \\ \alpha_z(3) \end{bmatrix} - \begin{bmatrix} \alpha_x(4) \\ \alpha_y(4) \\ \alpha_z(4) \end{bmatrix} \right\} + $$ $$where \; P_j = \begin{bmatrix} 0 & -\alpha_z(j) & \alpha_y(j) \\ \alpha_z(j) & 0 & -\alpha_x(j) \\ -\alpha_y(j) & \alpha_x(j) & 0 \end{bmatrix}$$ | Equation 67 <br><br> (16) |

## 6.2.6. Quaternion Calculation

Once the coning correction is completed, the quaternions can be determined. The quaternion calculation and correction use the algorithm presented in Figure 47. Therefore, the quaternion calculation is broken into two parts to reduce calculation error.

$$Q^P_{n+1} = Q^f_n \, \Delta\lambda$$

$$Q^f_{n+1} = \Delta m^* \, Q^P_{n+1}$$
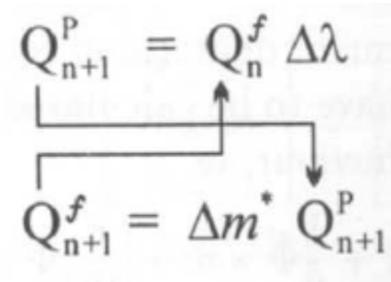
Figure 47 -Quaternion Calculation (16)

Equation 68 presents the discrete version of Euler angles to quaternion. Here, $\Delta\lambda$ is the quaternion, $\Delta\phi$ is the intensity of rotation, $\Delta\phi_x$, $\Delta\phi_y$, and $\Delta\phi_z$ are the rotation angles.

| | |
|---|---|
| $\Delta\lambda = \lambda_0 + \Delta\lambda_1 \mathrm{i} + \Delta\lambda_2 \mathrm{j} + \Delta\lambda_3 \mathrm{k}$ $$\Delta\lambda_0 = \cos\frac{\Delta\phi}{2}$$ $$\Delta\lambda_1 = \frac{\Delta\phi_x}{\Delta\phi}\sin\frac{\Delta\phi}{2}$$ $$\Delta\lambda_2 = \frac{\Delta\phi_y}{\Delta\phi}\sin\frac{\Delta\phi}{2}$$ $$\Delta\lambda_3 = \frac{\Delta\phi_z}{\Delta\phi}\sin\frac{\Delta\phi}{2}$$ | Equation 68 (16) |

The initial condition of the final quaternion $Q^f_n$ is set to form the transformation matrix between body to navigation frame. Once calculated, the final quaternion can be used to calculate the preliminary quaternion using Equation 69.

| | |
|---|---|
| $$Q^P_{n+1} = Q^f_n \Delta\lambda$$ | Equation 69 (16) |

The conjugate of the quaternion $\Delta m^*$ is shown in Equation 70. The conjugate is needed to calculate the final quaternions. $h_{N3}$ is the sampling interval and $\omega$ is the absolute angular velocity.

| | |
|---|---|
| $$\Delta m^* = \Delta m_0 - \Delta m_1 i - \Delta m_2 j - \Delta m_3 k$$ $$\Delta m_0 = \cos\frac{\omega h_{N3}}{2}$$ $$\Delta m_1 = \frac{\omega_x}{\omega}\sin\frac{\omega h_{N3}}{2}$$ $$\Delta m_2 = \frac{\omega_y}{\omega}\sin\frac{\omega h_{N3}}{2}$$ $$\Delta m_3 = \frac{\omega_z}{\omega}\sin\frac{\omega h_{N3}}{2}$$ | Equation 70 (16) |

Once the conjugate and preliminary quaternion are obtained, Equation 71 can be used to obtain the final quaternion.

| | |
|---|---|
| $$Q_{n+1}^f = \Delta m^* Q_{n+1}^P$$ | Equation 71 (16) |

## 6.2.7. Quaternion to Transformation Matrix

The final quaternion is first normalized using Equation 17. Then, Equation 15 is used to calculate the transformation matrix $T_{Body}^{Navigation}$ from quaternion.

## 6.2.8. Attitude Calculation

The resultant transformation matrix can be used to calculate the orientation, or attitude, of the missile. This is accomplished by taking parts of the matrix and its geometry property as shown in Equation 72.

$$T_{Body}^{Navigation}[0] = \sqrt{T_{Body}^{Navigation}[3,1]^2 + T_{Body}^{Navigation}[3,3]^2}$$

$$\theta = \text{atan} \frac{T_{Body}^{Navigation}[3,2]}{T_{Body}^{Navigation}[0]}$$

$$\gamma = -\text{atan} \frac{T_{Body}^{Navigation}[3,1]}{T_{Body}^{Navigation}[3,3]}$$

$$\psi = \text{atan} \frac{T_{Body}^{Navigation}[1,2]}{T_{Body}^{Navigation}[2,2]}$$

| | |
|---|---|
| (equations above) | Equation 72 (16) |

## 6.2.9. Sculling Compensation

Since the measured body acceleration is in an inertial frame, the Puasson's version of the equation should be used. However, since computational speed is essential, a discrete version of the body acceleration shown in Equation 73 is used.

$$W_{x,k} = W_{x,k-1} + W_{y,k-1}\alpha_{z,k} - W_{z,k-1}\alpha_{y,k} + \Delta W_{x,k}$$

$$W_{y,k} = W_{y,k-1} + W_{z,k-1}\alpha_{x,k} - W_{x,k-1}\alpha_{z,k} + \Delta W_{y,k}$$

$$W_{z,k} = W_{z,k-1} + W_{x,k-1}\alpha_{y,k} - W_{y,k-1}\alpha_{x,k} + \Delta W_{z,k}$$

$$W_{z,k} = W_{z,k-1} + W_{x,k}\alpha_{y,k} - W_{y,k}\alpha_{x,k} + \Delta W_{z,k}$$

$$W_{y,k} = W_{y,k-1} + W_{z,k}\alpha_{x,k} - W_{x,k}\alpha_{z,k} + \Delta W_{y,k}$$

$$W_{x,k} = W_{x,k-1} + W_{y,k}\alpha_{z,k} - W_{z,k}\alpha_{y,k} + \Delta W_{x,k}$$

$$Where\ Initial\ Condision, W_x = W_y = W_z = 0$$

Equation 73 (16)

## 6.2.10. Velocity Increment to Navigation Frame

Once the sculling correction is completed, the velocity increment can be transformed, using the transformation matrix from section 6.2.7, to the navigation frame using Equation 74.

$$\begin{bmatrix} \Delta W_x \\ \Delta W_y \\ \Delta W_z \end{bmatrix}_{Navigation} = T_{Body}^{Navigation} \begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix}_{Body}$$

Equation 74 (16)

## 6.2.11. Velocity Calculation

The velocity of the missile can be calculated with respect to the projection of velocity to Earth by using Equation 75. $\Delta W_x^*$, $\Delta W_y^*$, and $\Delta W_z^*$ are the running sum of the velocity increment. $U_x$, $U_y$, $U_z$ are the projection of Earth rotation on the none-inertial frame. $\Omega_x$ and $\Omega_y$ are the projection of angular velocity on the none-inertial frame.

| | |
|---|---|
| $V_x = \Delta W_x^* + \int_{t_0}^{t} (V_y 2U_z - V_z(\Omega_y + 2U_y)) \mathrm{dt}$ | Equation 75 |
| $V_y = \Delta W_y^* - \int_{t_0}^{t} (V_x 2U_z - V_z(\Omega_x + 2U_x)) \mathrm{dt}$ | (16) |
| $V_z = \Delta W_z^* - \int_{t_0}^{t} V_y(\Omega_x + 2U_x) \mathrm{dt} - \int_{t_0}^{t} V_x(\Omega_y + 2U_y) \mathrm{dt} + g$ | |

The angular velocity projection can be calculated using Equation 76. Here, T is the transformation matrix from ECEF to navigation frame from section 3.1.4.2.

| | |
|---|---|
| $\Omega_x = -\dfrac{V_y}{R_y} - \dfrac{V_x}{a} e^2 T[1,2]T[2,3]$ | Equation 76 |
| $\Omega_y = \dfrac{V_x}{R_x} + \dfrac{V_y}{a} e^2 T[1,3]T[2,3]$ | (16) |

The elliptical Earth model radii are calculated using Equation 77. Here a and e are from the elliptical earth model from section 3.1.1.3. In addition, H is the altitude of the missile.

| | |
|---|---|
| $\dfrac{1}{R_x} = \dfrac{1}{a}(1 - e^2 \dfrac{T[3,3]^2}{2} + e^2 T[1,3]^2 - \dfrac{H}{a})$ | Equation 77 |
| $\dfrac{1}{R_Y} = \dfrac{1}{a}(1 - e^2 \dfrac{T[3,3]^2}{2} + e^2 T[2,3]^2 - \dfrac{H}{a})$ | (16) |

Equation 78 calculates the projection of the absolute angular velocity.

| | |
|---|---|
| $\omega_x = \Omega_x + U_x$ | Equation 78 |
| $\omega_y = \Omega_y + U_y$ | (16) |
| $\omega_z = U_z$ | |

Earth's angular velocity components can be calculated using Equation 79.

| | |
|---|---|
| $U_x = UT[1,3]$ | Equation 79 |
| $U_y = UT[2,3]$ | (16) |
| $U_z = UT[3,3]$ | |

## 6.2.12. Puasson Discrete Transformation Matrix

Once the velocity is calculated, the transformation matrix should be put in Puasson form as shown in Equation 80.

| | |
|---|---|
| $T[1,2]_N = T[1,2]_{N-1} - \Omega_y T[3,2]_{N-1} h_{N3}$ $T[2,2]_N = T[2,2]_{N-1} + \Omega_x T[3,2]_{N-1} h_{N3}$ $T[3,2]_N = T[3,2]_{N-1} + (\Omega_y T[1,2]_{N-1} - \Omega_x T[2,2]_{N-1}) h_{N3}$ $T[1,3]_N = T[1,3]_{N-1} - \Omega_y T[3,3]_{N-1} h_{N3}$ $T[2,3]_N = T[2,3]_{N-1} + \Omega_x T[3,3]_{N-1} h_{N3}$ $T[3,3]_N = T[3,3]_{N-1} + (\Omega_y T[1,3]_{N-1} - \Omega_x T[2,3]_{N-1}) h_{N3}$ $T[3,1]_N = T[1,2]_N T[2,3]_N - T[2,2]_N T[1,3]_N$ | Equation 80 (16) |

## 6.2.13. Coordinate Calculation

Now that all the parameters are found, the coordinate of the missile can be calculated. Here $\phi$ is the latitude, $\lambda$ is the longitude, and $\epsilon$ is the azimuth of the missile.

| | |
|---|---|
| $T[0] = \sqrt{T[3,1]^2 + T[2,3]^2}$ | |
| $\phi = \text{atan} \dfrac{T[3,3]}{T[0]}$ | Equation 81 |
| $\lambda = \text{atan} \dfrac{T[3,2]}{T[3,1]}$ | (16) |
| $\epsilon = \text{atan} \dfrac{T[1,3]}{T[2,3]}$ | |

IMU | $G_{xb}$ | $G_{yb}$ | $G_{zb}$ | | $A_{zb}$ | $A_{yb}$ | $A_{xb}$

$\omega_{xb}$ $\omega_{yb}$ $\omega_{zb}$  $a_{xb}$ $a_{yb}$ $a_{zb}$

**Compensation of gyro errors according to model (6.1)**

**Compensation of acceleration errors according to model (6.1)**

**Calculation of angle increments**

$$\alpha_{xb,yb,zb} = \int_{t_k}^{t_k+h_{NI}} \omega_{xb,yb,zb}\,dt$$

$h_{N1}$

**Calculation of velocity increments**

$$\Delta W_{xb,yb,zb} = \int_{t_k}^{t_k+h_{NI}} \alpha_{xb,yb,zb}\,dt$$

$h_{N1}$

$\alpha_{xb,yb,zb}$  $\Delta W_{xb,yb,zb}$

$h_{N3}$ **Coning correction (equation 4.10)**  **Sculling compensation (equation 4.4)** $h_{N3}$

$\Delta\Phi_{xb}$ $\Delta\Phi_{yb}$ $\Delta\Phi_{zb}$  $W_{xb}$ $W_{yb}$ $W_{zb}$

alignment
$q_0, q_i^c$
$q_i^{fin}$

$q_i^f$

**Quaternion calculation -1**

$$Q_{n+1}^p = Q_n^f \,\Delta\lambda$$

$$\Delta\lambda_0 = \cos\frac{\Delta\Phi}{2}$$

$$\Delta\lambda_1 = \frac{\Delta\Phi_{xb}}{\Delta\Phi}\sin\frac{\Delta\Phi}{2}$$

$$\Delta\lambda_2 = \frac{\Delta\Phi_{yb}}{\Delta\Phi}\sin\frac{\Delta\Phi}{2}$$

$$\Delta\lambda_2 = \frac{\Delta\Phi_{zb}}{\Delta\Phi}\sin\frac{\Delta\Phi}{2}$$

**Recalculation of velocity increments into navigation frame**

$\Delta W_{xb}$

alignment

$$\begin{bmatrix} \Delta W_{xb} \\ \Delta W_{yb} \\ \Delta W_{zb} \end{bmatrix} = C_b^N \begin{bmatrix} W_{xb} \\ W_{yb} \\ W_{zb} \end{bmatrix}$$

$C_{ij}$

Figure 48 - INS Algorithm Part 1 (16)
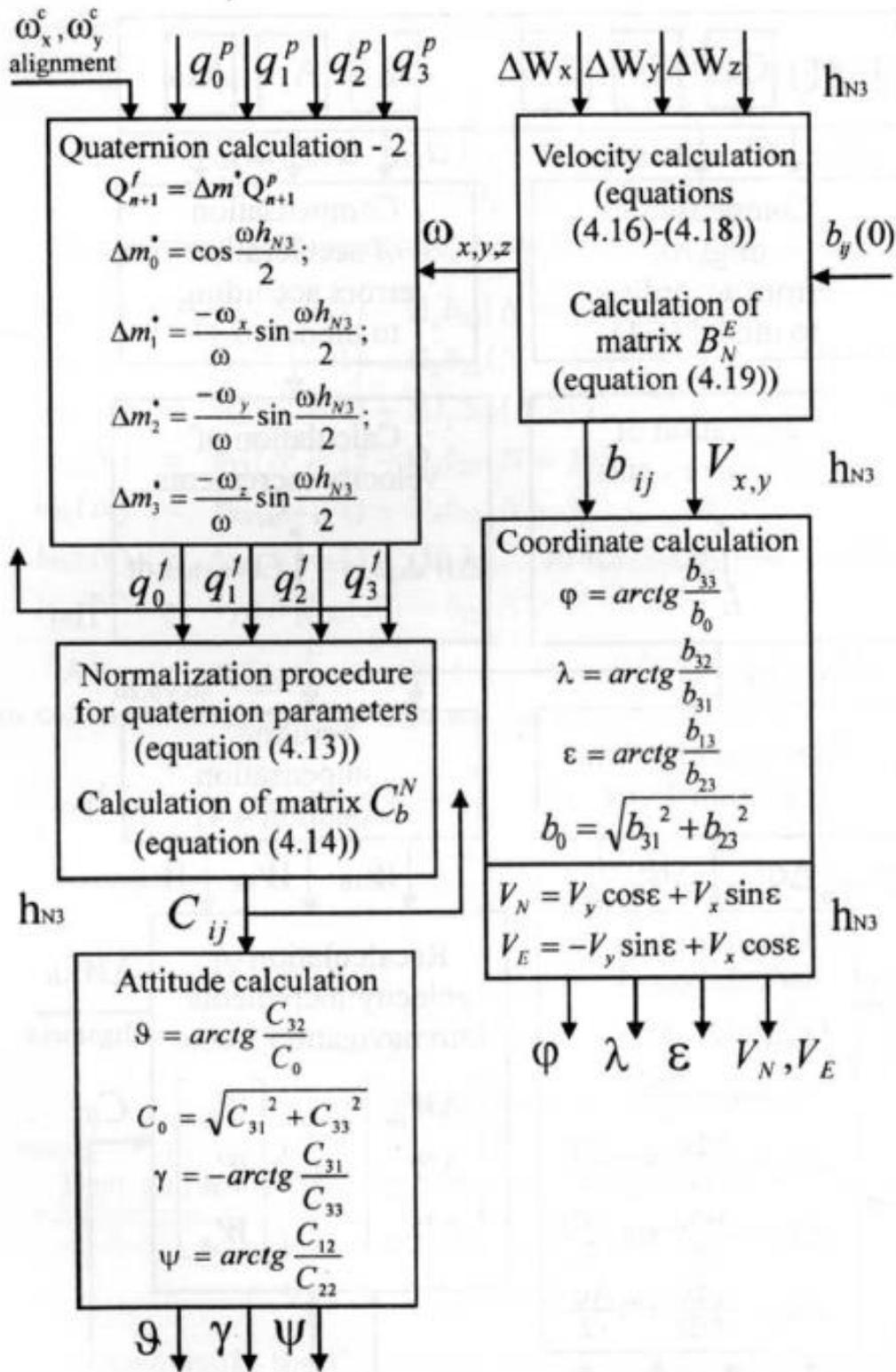
Figure 49 - INS Algorithm Part 2 (16)

## 6.3. GPS/INS Integration

### 6.3.1. GPS Modeling

There is a wide range of models that can accurately represent the GPS. Similar to the IMU modeling, the GPS error modeling is shown in Equation 82. Here the range from missile to each satellite $R_i$ consists of the true range $r_i$, $X_A$ antennae phase error, $X_P$ position error, $X_V$ velocity error, and $X_{TSF}$ scale factor error. Each error is multiplied with a coefficient $C$ that is shorter form of the transformation matrix and vector presented in the original paper.

| | |
|---|---|
| $$R_i = r_i + C_0 X_A + C_1 X_P + C_2 X_V + C_4 X_{TSF}$$ | Equation 82 (31) |

Once the range is obtained for at least three satellites, and since the position of each satellite $X_i$, $Y_i$, $Z_i$ is known, a system of three equation can be generated from Equation 83 to solve for the missile position, x, y, and z.

| | |
|---|---|
| $$R_i = \sqrt{(x - X_i)^2 + (y - Y_i)^2 + (z - Z_i)^2}$$ | Equation 83 (32) |

Nevertheless, a simpler yet effective GPS modeling method is used in this thesis. Since the chosen GPS has the sampling frequency $f_s$ of 10Hz, and the position of the missile in the simulation can be obtained, the GPS is modeled using a delayed sampling time shown in Equation 84.

| | |
|---|---|
| $$x_s = \sum_{n=-\infty}^{\infty} x_c \delta(t - nT_s)$$ $$\Omega_s = \frac{2\pi}{T_s} \frac{\text{rad}}{\text{s}}$$ $$f_s = \frac{1}{T_s} \text{ Hz}$$ | Equation 84 (33) |

## 6.3.2. Kalman Filter

In order to make sure of the GPS, a Kalman filter is used to integrate both INS algorithm data and GPS data. Hence, it is essential to understand the Kalman filter. The Kalman filter consists of two major part, the state space model and the Kalman equation.

### 6.3.2.1. State Space Model

The first part of the state space model, shown in Equation 85, is used to predict the behavior of the missile. This serves as the reference behavior of the missile. Here, $x_k$ is the state vector, $\Phi_{k,k-1}$ is the transition matrix, $G_{k,k-1}$ is the input matrix, and $\omega_{k-1}$ is the input white noise with covariance matrix $Q$.

| | |
|---|---|
| $$x_k = \Phi_{k,k-1}x_{k-1} + G_{k,k-1}\omega_{k-1}$$ | Equation 85 <br><br> (16) |

In addition, the second part is the measurement part, where the behavior of the missile is measured and computed using Equation 86. Here, $z_k$ is the measurement vector, $H$ is the output matrix, and $v_k$ is the measurement white noise with covariance matrix $R$.

| | |
|---|---|
| $$z_k = Hx_k + v_k$$ | Equation 86 <br><br> (16) |

### 6.3.2.2. Kalman Equation

The traditional Kalman equation is shown in Equation 87. Here, $\hat{x}_k$ is the optimal estimation and $K_k$ is the Kalman gain.

| | |
|---|---|
| $$\hat{x}_k = \phi_{k,k-1}\hat{x}_{k-1} + K_k(z_k - H\Phi_{k,k-1}\hat{x}_{k-1})$$ | Equation 87 <br><br> (16) |

## 6.3.2.3. Kalman Filter Integration Algorithm

One approach to integrate the GPS and INS is the use Kalman filter to predict the error instead of the position and velocity of the missile. This algorithm is presented in Figure 50.
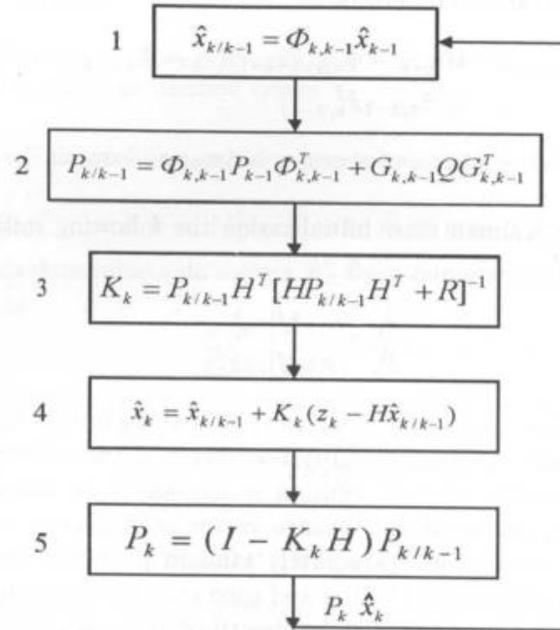


Figure 50 - GPS/INS Integration Algorithm (16)

The first step of the algorithm is to find the previous prediction $\hat{x}_{k-1}$. The previous prediction for position is set to be the difference, $\delta N$, $\delta E$, and $\delta U$ between the INS algorithm output and the GPS output as stated in Equation 88.

| | |
|---|---|
| $$\begin{bmatrix} \delta N \\ \delta E \\ \delta U \end{bmatrix} = \begin{bmatrix} N_{INS} - N_{GPS} \\ E_{INS} - E_{GPS} \\ U_{INS} - U_{GPS} \end{bmatrix}$$ | Equation 88 (16) |

In addition, the velocity difference $\delta V_N$, $\delta V_E$, and $\delta V_U$ is also computed as stated in Equation 89.

| | |
|---|---|
| $$\begin{bmatrix} \delta V_N \\ \delta V_E \\ \delta V_U \end{bmatrix} = \begin{bmatrix} V_{N\_INS} - V_{N\_GPS} \\ V_{E\_INS} - V_{E\_GPS} \\ V_{U\_INS} - V_{U\_GPS} \end{bmatrix}$$ | Equation 89 (16) |

Moreover, Equation 90 computes the angular difference $\phi_N$, $\phi_E$, and $\phi_U$.

$$\begin{bmatrix} \phi_N \\ \phi_E \\ \phi_U \end{bmatrix} = \begin{bmatrix} -\delta\gamma \\ -\delta\theta \\ \delta H \end{bmatrix}$$

Equation 90

(16)

Equation 91 shows the three previous prediction $\hat{x}_{k-1}$ which are represented as $North_{k-1}$, $East_{k-1}$, and $Up_{k-1}$. Here, the Kalman filter algorithm is computed three times. Each time for a navigation axis channel.

$$North_{k-1} = \begin{bmatrix} \delta N \\ \delta V_N \\ \phi_E \end{bmatrix}$$

$$East_{k-1} = \begin{bmatrix} \delta E \\ \delta V_E \\ \phi_N \end{bmatrix}$$

Equation 91

(16)

$$Up_{k-1} = \begin{bmatrix} \delta U \\ \delta V_U \\ \phi_U \end{bmatrix}$$

The transfer matrix $\Phi_k$ for each of the individual channel is presented in Equation 92. This is the state space model of the behavior of the missile.

$$\Phi_{N\_k} = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & -gT_s \\ 0 & \dfrac{T_s}{R_{EARTH}} & 1 \end{bmatrix}$$

$$\Phi_{E\_k} = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & -gT_s \\ 0 & \dfrac{T_s}{R_{EARTH}} & 1 \end{bmatrix}$$

Equation 92

(16)

$$\Phi_{U\_k} = \begin{bmatrix} 1 & T_s & 0 & 0 \\ \dfrac{-2g}{R_{EARTH}}T_s & 1 & 0 & 0 \\ 0 & 0 & \dfrac{T}{R}\tan\Xi & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The state place model can be multiplied by the previous prediction to obtain the updated previous prediction $\hat{x}_{k-1}$. It is made of three channels as seen in Equation 93. It is important to note that $b_a$ is the accelerometer bias error.

| | |
|---|---|
| $\delta North_k = \Phi_{N\_k} North_{k-1}$ | |
| $\delta East_k = \Phi_{E\_k} East_{k-1}$ | Equation 93 |
| $\delta Up_k = \Phi_{U\_k} Up_{k-1} + \begin{bmatrix} 0 \\ b_a \\ 0 \\ 0 \end{bmatrix}$ | (16) |

Before progressing to the second step, a couple of parameters needs to be defined. These parameters are tuned for the missile presented in this thesis. Hence, it is essential to consider the proper parameters for other missiles. Equation 94 shows the output matrix for the missile. Here, it is set to identity matrix because the output is assumed to be not skewed.

| | |
|---|---|
| $H_{North} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | |
| $H_{East} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | Equation 94 |
| $H_{Up} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | (16) |

In addition, the measurement noise matrix $R$ is shown in Equation 95.

| | |
|---|---|
| $R_{North} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$ | |
| $R_{East} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$ | Equation 95 |
| $R_{Up} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$ | (16) |

Moreover, the initial input noise $Q$ is presented in Equation 96.

| | |
|---|---|
| $Q_{North} = P_{k\_North\_Initial} = \begin{bmatrix} 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \\ 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \\ 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \end{bmatrix}$ | |
| $Q_{East} = P_{k\_East\_Initial} = \begin{bmatrix} 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \\ 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \\ 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \end{bmatrix}$ | Equation 96 (16) |
| $Q_{Up} = P_{k\_Up\_Initial} = \begin{bmatrix} 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \\ 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \\ 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \\ 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} & 1 \times 10^{-3} \end{bmatrix}$ | |

Lastly, the measurement vector $z_k$ is shown in Equation 97.

| | |
|---|---|
| $z_{North} = \begin{bmatrix} \delta N \\ \delta V_N \\ \phi_E \end{bmatrix}$ | |
| $z_{East} = \begin{bmatrix} \delta E \\ \delta V_E \\ \phi_N \end{bmatrix}$ | Equation 97 (16) |
| $z_{Up} = \begin{bmatrix} \delta U \\ \delta V_U \\ \delta V_E \\ \phi_U \end{bmatrix}$ | |

The second step of the algorithm is to calculate the input noise. This is accomplished by using Equation 98 where the previous input noise is multiplied by the transfer matrix and its conjugates.

| | |
|---|---|
| $P_{k\_North} = \Phi_{N\_k} P_{k-1\_North} \Phi'_{N\_k}$ | |
| $P_{k\_East} = \Phi_{E\_k} P_{k-1\_East} \Phi'_{E\_k}$ | Equation 98 (16) |
| $P_{k\_Up} = \Phi_{U\_k} P_{k-1\_Up} \Phi'_{U\_k}$ | |

The third step is to calculate the Kalman gain as shown by Equation 99.

| | |
|---|---|
| $$K_{North} = P_{k\_North}\, H'_{North}\, inv(H_{North}\, P_{k_{North}}H'_{North} + R_{North})$$ | |
| $$K_{East} = P_{k\_East}\, H'_{East}\, inv(H_{East}\, P_{k_{East}}H'_{East} + R_{East})$$ | Equation 99 (16) |
| $$K_{Up} = P_{k\_Up}\, H'_{Up}\, inv(H_{Up}\, P_{k_{Up}}H'_{Up} + R_{Up})$$ | |

After obtaining all the parameters, the fourth step is to calculate the optimal prediction $\hat{x}_k$ in Equation 100.

| | |
|---|---|
| $$\begin{bmatrix} \delta\hat{N} \\ \delta\widehat{V_N} \\ \delta\widehat{\phi_E} \end{bmatrix} = \delta North_k + K_{North}(z_{North} - H_{North}\delta North_k)$$ | |
| $$\begin{bmatrix} \delta\hat{E} \\ \delta\widehat{V_E} \\ \delta\widehat{\phi_N} \end{bmatrix} = \delta East_k + K_{East}(z_{East} - H_{East}\delta East_k)$$ | Equation 100 (16) |
| $$\begin{bmatrix} \delta\hat{U} \\ \delta\widehat{V_U} \\ \delta\widehat{\phi_U} \end{bmatrix} = \delta Up_k + K_{Up}(z_{Up} - H_{Up}\delta Up_k)$$ | |

Equation 101 updates the input noise as the fifth step of the algorithm.

| | |
|---|---|
| $$P_{k-1\_North} = (I - K_{North}H_{North})P_{k\_North}$$ | |
| $$P_{k-1\_East} = (I - K_{East}H_{East})P_{k\_East}$$ | Equation 101 (16) |
| $$P_{k-1\_Up} = (I - K_{Up}H_{Up})P_{k\_Up}$$ | |

Finally, the GPS/INS integration output can be obtained using Equation 102. Here the INS algorithm output is subtracted by the optimal prediction. Hence, this corrects the error that was generated from the INS algorithm.

| | |
|---|---|
| $North_{GPSINS} = N_{INS} - \delta\hat{N}$ | |
| $East_{GPSINS} = E_{INS} - \delta\hat{E}$ | Equation 102 (16) |
| $North_{GPSINS} = U_{INS} - \delta\hat{U}$ | |

Similarly, Equation 103 shows that the GPS/INS velocity can be obtained from the subtraction of the INS algorithm velocity and the optimally predicted velocity. Thereby, concluding the GPS/INS integration algorithm.

| | |
|---|---|
| $Velcoity_{N\_GPSINS} = V_{N\_INS} - \delta\widehat{V_N}$ | |
| $Velcoity_{E\_GPSINS} = V_{E\_INS} - \delta\widehat{V_E}$ | Equation 103 (16) |
| $Velcoity_{U\_GPSINS} = V_{U\_INS} - \delta\widehat{V_U}$ | |

## 6.3.3. Neural Network GPS/INS Integration

The purpose of the Neural Network GPS/INS Integration (NNI) is to integrate GPS and INS data during the missile flight. This is achieved by designing a neural network that mimics and improves the functionality of GPS/INS integration.

## 6.3.3.1. NNI Algorithm

The NNI algorithm is shown in Figure 51. Here, the GPS data is subtracted from the INS data. The resultant can be denoted as $\Delta NED$. Hence, it creates the NED frame error. The error is fed into the NNI. In addition, a delayed sample of the NED frame error is also fed into the NNI. This is to allow the NNI to obtain more information when predicting the NED coordinates. The larger the previous error the bigger the correction should become. Moreover, the previous correction data is also fed into the NNI. This is to allow the NNI to know how far the previous correction was from the actual NED frame coordinates. The larger the correction, the more NNI prediction was incorrect. Hence, it needs to compensate for it.
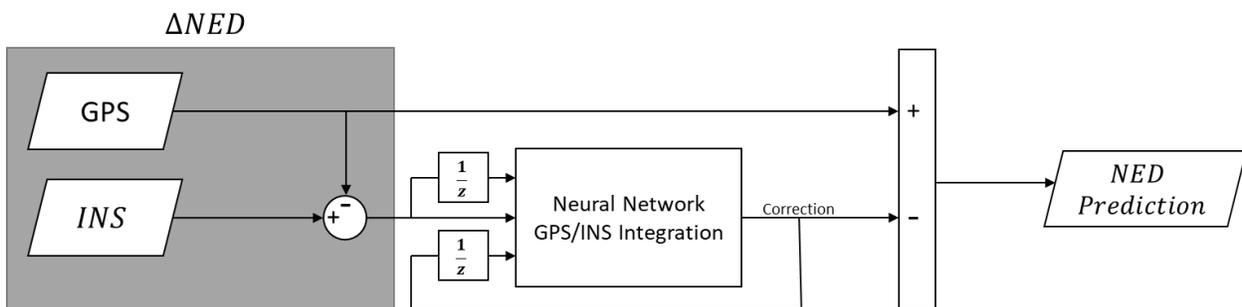


Figure 51 - Neural Network Integration Algorithm

Once all the three data are imported into the NNI, it produces a correction prediction. The correction prediction is subtracted from the GPS data. This results in the NED prediction which can be used in the guidance and autopilot of the missile. The reason the GPS is used as a point of reference is because it is assumed to be the most accurate. It is important to keep in consideration that with this design, a significantly faulty GPS data will greatly impact the performance of the NNI. However, assuming the GPS performance is acceptable, the NNI performs as designed. The GPS data is produced using the GPS model in Equation 84. In addition, the IMU model in Equation 64 is used to produce gyro rates and acceleration. The results are fed into the INS algorithm in Figure 48 and Figure 49 to produce the INS data.

## 6.3.3.2. NNI Design

The design of NNI is a traditional feedforward network as shown in Figure 52. The input neuron consists of previous NED frame error, current NED frame error, and previous correction. The output neuron consists of the correction prediction. The number of hidden layers has been varied during the training process where 10 layers delivered the optimum result. The length of each hidden layers as well as the connections between the layers, hence the weight and biases, are also varied during the training to obtain the optimum result.
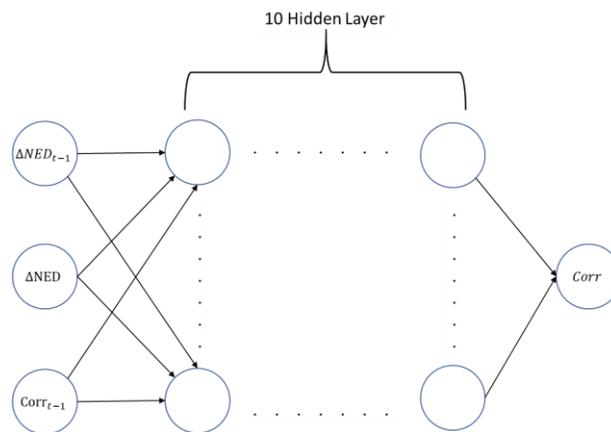


Figure 52 - Neural Network Integration Design

## 6.3.3.3. NNI Training Environment

In order to train the Neural Network NNI using NEAT Training method, the missile environment simulation must be prepared. However, unlike with TGNN Training Environment, the missile simulation model cannot be simplified. That is because the time variant and non-linear dynamics affect the NNI performance greatly. Hence, a full simulation needs to be used. Since the full simulation of the missile will take very long to execute, the simulation is pre-executed and the resulted INS Data and GPS Data are recorded. Furthermore, one of the biggest challenges with neural network is generalization. The generalization problem can be reduced by evaluating the performance of the NNI across a range of pre-executed simulation. Therefore, the NNI simulation, shown in Figure 53, consists of a loop. The simulation reads the launch angle $\theta_0$. Then, based

on a look up table of the previously executed full missile simulation, the NNI algorithm, shown in Figure 52, reads the corresponding GPS and INS Data to obtain the NED prediction. The launch angle is varied from 10deg to 80deg in order to include a comprehensive set of INS and GPS behaviors. It is important to note that each of the three position and three velocity channels are trained separately. In addition, the training for the side component fixes the launch angle at 45degs and the initial Azimuth $Az_0$ is varied from -80 to 80 deg as shown in Figure 54. In addition, the training session is run for 4 hours which proved to be sufficient to obtain the optimum results.
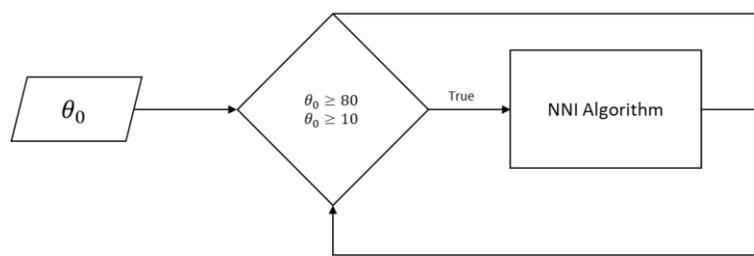


Figure 53 - NNI Simulation Launch Angle Variation
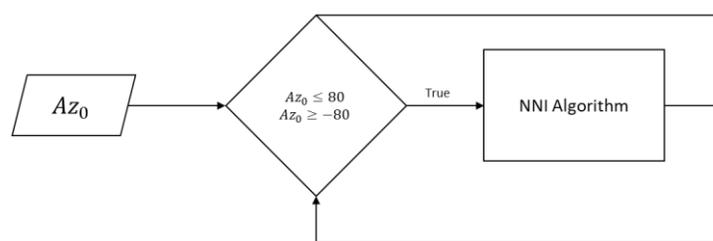


Figure 54 – NNI Simulation Azimuth Angle Variation

## 6.3.3.4. NNI Training Genetic Algorithm Parameters

The GA fitness function $Fitness_{NNI}$ shown in Equation 104 is used to evaluate the quality of NNI. $Fitness_{NNI}$ consists of the summation of all the Means Square Error of the Simulation Real XNED $XNED_{6DOF}$ and the XNED Prediction $XNED_{\text{Prediction}}$. This sum reduces the generalization problem and it allows for a global quantification of the performance of the NNI.

| $Fitness_{NNI} = \sum_{t=0}^{n} \left( \text{MSE}(XNED_{6DOF}, XNED_{Prediction}) \right)_t$ | Equation 104 |
|---|---|

# 7. Implementation and Verification

## 7.1. High Level Simulation

Figure 55 shows the highest-level simulation block diagram. Here, it can be seen that the booster affects the missile body. In addition, the gravitational model and aerodynamics also affects the missile body. The missile body flies within the environment where the coordinate transformation, actual missile velocity, position, and rotation rates are calculated. The environment effects are felt by the aerodynamics and guidance computer. The guidance computer control output affects the actuator. The actuator deflection affects the aerodynamics. Hence, this completes the high-level missile simulation cycle.
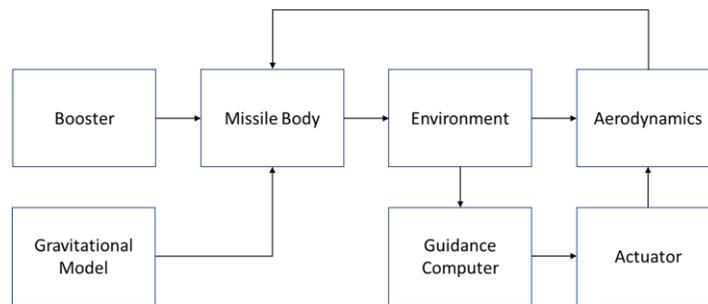


Figure 55 – High Level Simulation

## 7.1.1. Booster Simulation

Figure 56 shows the booster simulation block diagram. Here, the Thrust profile from Figure 26 is read using a look-up table. The misalignment calculation uses Equation 39 to calculate the thrust forces. Moreover, the thrust moments are calculated using Equation 40. They are outputted in body frame.
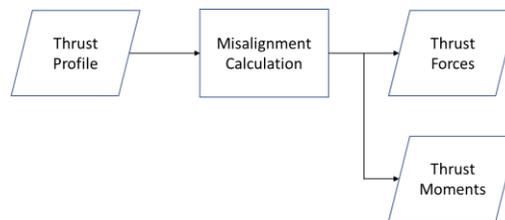


Figure 56 – Booster Simulation

## 7.1.2. Gravitational Model Simulation

Figure 57 shows the gravitational model simulation block diagram. Here, the earth gravity parameters are read. It could be either for a flat, spherical, or elliptical model. The gravity is calculated using Equation 5 for flat and spherical earth models. However, it uses the WGS84 standard in Equation 6 for the elliptical model. The gravity is then outputted in body frame.
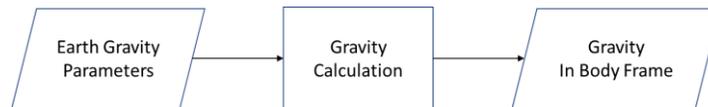


Figure 57 – Gravitational Simulation

## 7.1.3. Aerodynamics Simulation

Figure 58 shows the aerodynamic simulation block diagram. Here, structure parameter such as the center of mass is read. Moreover, the environmental parameters such as angle of attack, side angle, missile velocity and Mach, dynamic pressure, and rotation rates are read. In addition, the aerodynamic derivatives are read as well. These are predetermined by the missile model. Additionally, the actuator deflections are inputted too. These are the results of the guidance computer commands. The aerodynamic calculation is then performed using Equation 41 to obtain both aerodynamic force and moment coefficients. The aerodynamic forces are calculated using Equation 42. On the other hand, the aerodynamic moments are calculated using Equation 43. Both force and moments are outputted in body frame.
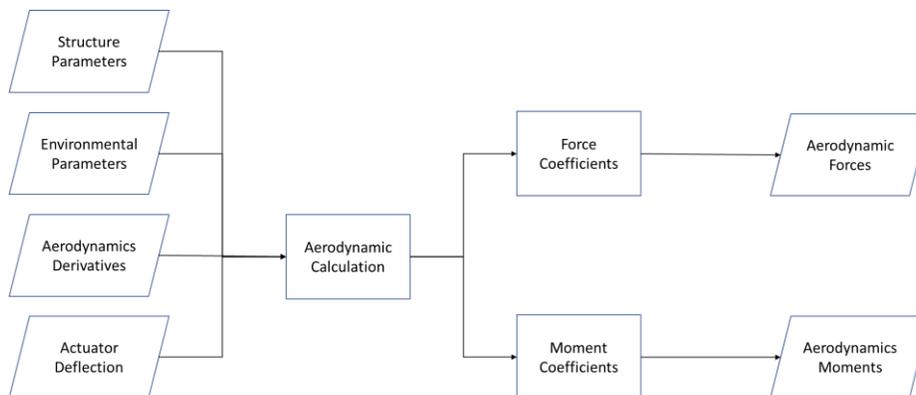


Figure 58 – Aerodynamic Simulation

### 7.1.4. Missile Body Simulation

Figure 59 shows the missile body simulation block diagram. Here, the gravitational forces, aerodynamic forces, and thrust forces are summed together to form the total force on the missile. In addition, the aerodynamic moments and the thrust moments are summed together to form a total moment on the missile. The total moments and forces are outputted in body frame. Furthermore, the structure parameters such as the missile body inertia, mass, and center of mass as a function of time is read and outputted.
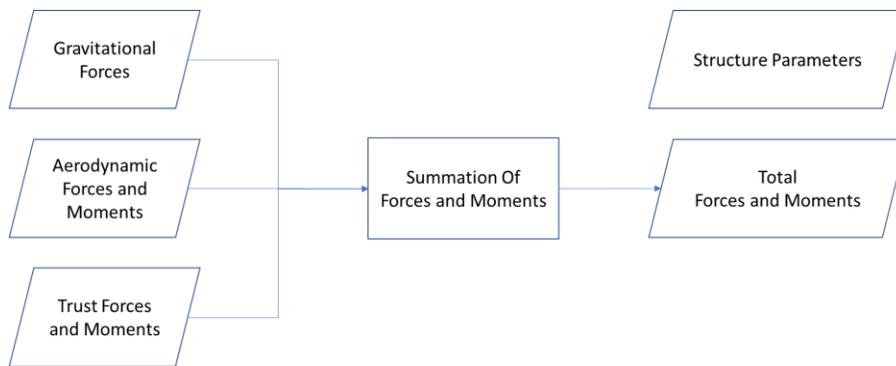


Figure 59 - Missile Body Simulation

### 7.1.5. Environment Simulation

Figure 60 shows the environment simulation block diagram. Here, the atmospheric model shown in Table 1 is used to find the atmospheric parameters such as angle of attack, side angle, missile velocity and Mach, and dynamic pressure. In addition, the mass and the total force are used to calculate the body acceleration using Equation 20. However, for an accurate simulation Equation 21 can be used. Moreover, the inertial and total moments are read to calculate the rotation rates using Equation 34. The rotation rates are used to calculate the quaternion derivative using Equation 16. Then, the quaternion is used to calculate the missile angles using Equation 13. Furthermore, the quaternion can be used to calculate the transformation matrix Body to NED using the transpose of Equation 15.

The body acceleration and transformation matrix are corrected using Pausson's equation shown in Equation 10. In addition, the corrected velocity and position can be obtained as well. These localization parameters can be converted into ECEF frame using transpose of Equation 8. Moreover, it can be converted into geodetic frame using Equation 9. It is important to note that in case wander frame is used it can be converted into wander frame as well using the yaw transportation matrix of Equation 7.
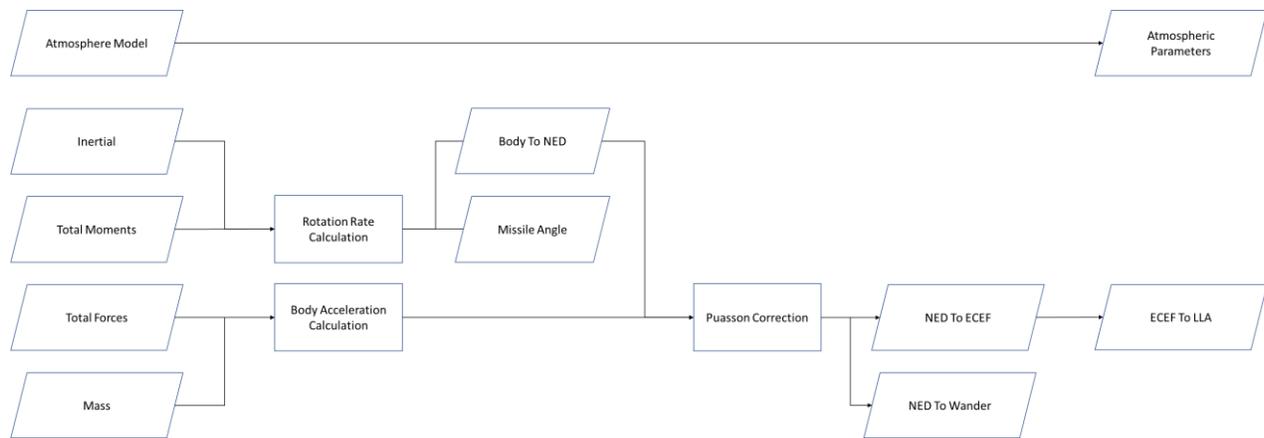


Figure 60 - Environment Simulation

## 7.1.6. Actuator Simulation

Figure 61 shows the actuator simulation block diagram. Here, the roll, pitch, and yaw commands are received from the guidance computer. These commands are summed together while prioritizing the roll command. That is because if the roll angle is not stabilized, the pitch and yaw commands could be incorrect since the guidance does not consider the roll angle. Each sum is inputted into an actuator model. Equation 53 is used to model the actuating system. In addition, the output response of the roll, pitch, and yaw deflection is calculated using Equation 54.
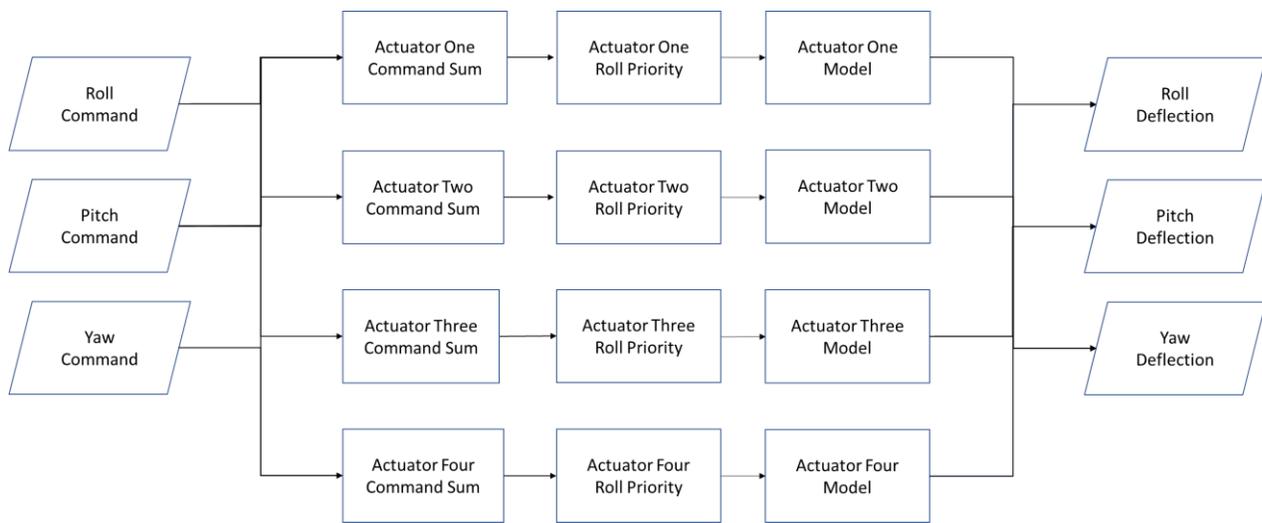
Figure 61 – Actuator Simulation

## 7.1.7. Guidance Computer Simulation

Figure 62 shows the guidance computer simulation block diagram. Here, the environmental parameters such as acceleration and rotation rates are read into the navigation. The navigation calculates the velocity, position, and angle of the missile. The navigation output is fed into both the guidance and the control. On the other hand, the seeker block outputs the target location. This simulation assumes an ideal seeker where the target location is exactly known.  The seeker output is fed into the guidance. The guidance calculates the guidance command then it feeds it to the control where the roll, pitch, and yaw commands are calculated. The control commands are then outputted.
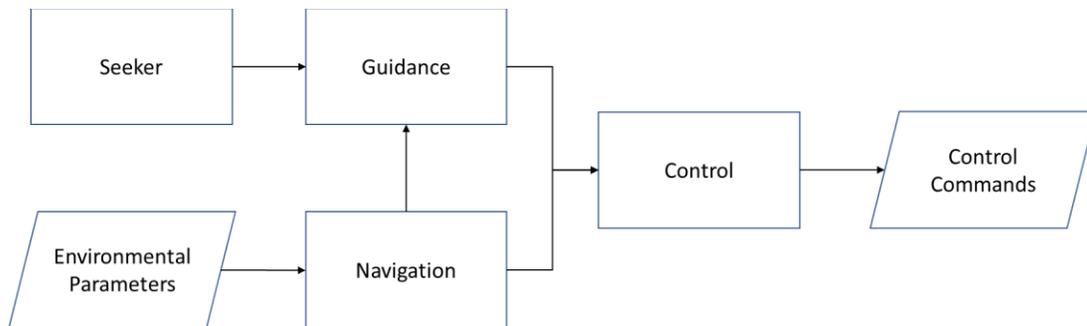


Figure 62 – Guidance Computer Simulation

## 7.1.8. Navigation Simulation

Figure 63 shows the navigation simulation block diagram. Here, the environmental parameters such as the acceleration and rotation rates are fed into the IMU. The IMU parameter is then fed into the INS block. In addition, the environmental parameter such as the missile position in geodetic frame is fed into the GPS. Then, the GPS parameter and INS parameter are fed into the GPS/INS integration.
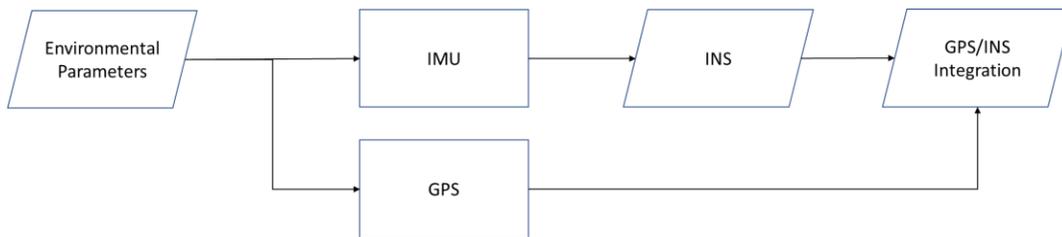
Figure 63 – Navigation Simulation

## 7.1.9. IMU Simulation

Figure 64 shows the IMU simulation block diagram. Here, the environmental parameters such as acceleration and rotation rates are read. The IMU model from Equation 64 is used to calculate the IMU parameters. Then, the IMU parameters such as the noisy acceleration and noisy rotation rates are outputted.
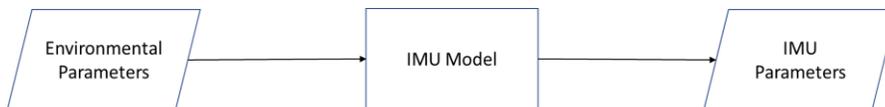
Figure 64 - IMU Simulation

## 7.1.10. GPS Simulation

Figure 65 shows the GPS simulation block diagram. Here, the environmental parameters such as the missile position in geodetic frame is read. The GPS model from Equation 84 is used to calculate the GPS parameters. Then, the GPS parameters such as noisy position and velocity are outputted.
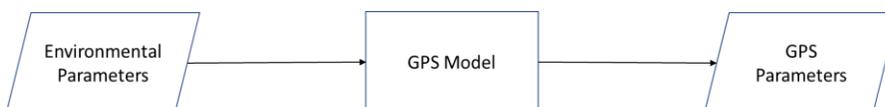
Figure 65 - IMU Simulation

# 7.1.11. Inertial Navigation System Simulation

Figure 66 shows the INS simulation block diagram. Here, the IMU parameters such as acceleration and rotating rates are read into the INS algorithm. The INS algorithm from Figure 48 and Figure 49 calculates the estimated missile velocity, position, and angle in NED and geodetic frame. These INS parameters are then outputted. It is important to note that in the case of GPS/INS integration with reset, the INS reset trigger is used to reset the INS. Moreover, the GPS/INS integration prediction is inputted to the INS algorithm for reset.
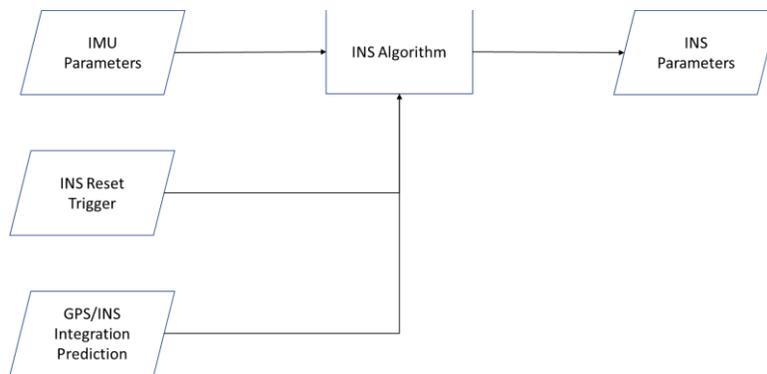


Figure 66 – Inertial Navigation System Simulation

# 7.1.12. GPS/INS Integration Simulation

Figure 67 shows the GPS/INS Integration simulation block diagram. Here, the INS parameters such as the missile velocity, position, and angle are read. Moreover, the GPS parameters such as the velocity, position, and geodetic coordinates are read. Then, the GPS/INS integration algorithm calculates the prediction using Figure 50 for Kalman filter method. On the other hand, the GPS/INS integration algorithm calculates the prediction using Figure 51 for the NNI. The GPS/INS integration prediction is then outputted. It is important to note that the desired algorithm is pre-selected before the simulation is executed.
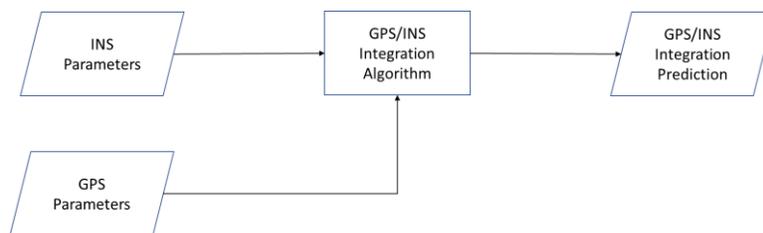


Figure 67 – GPS/INS Integration Simulation

# 7.1.13. Guidance Simulation

Figure 68 shows the guidance simulation block diagram. Here, the predefined trajectory is fed into the trajectory guidance block. In addition, the navigation parameters such as the missile position is also fed into the trajectory guidance block. This is where the trajectory guidance command is calculated using Equation 59.

Moreover, the seeker parameters, such as target location, and the navigation parameters such as the missile position and velocity are fed into the terminal guidance block. Here, the terminal guidance command is calculated using Equation 61 for Proportional Navigation algorithm. However, the terminal guidance command is calculated using Figure 41 for TGNN. It is important to note that the type of terminal guidance algorithm is pre-selected before the simulation is executed.

In addition, the terminal guidance and trajectory guidance cannot be happening at the same time. Hence, an exclusive OR block is used to allow only a single type of the guidance to be used. It is also important to note that the type of guidance is pre-selected before the simulation execution. However, it can be coded to switch during the missile flight.
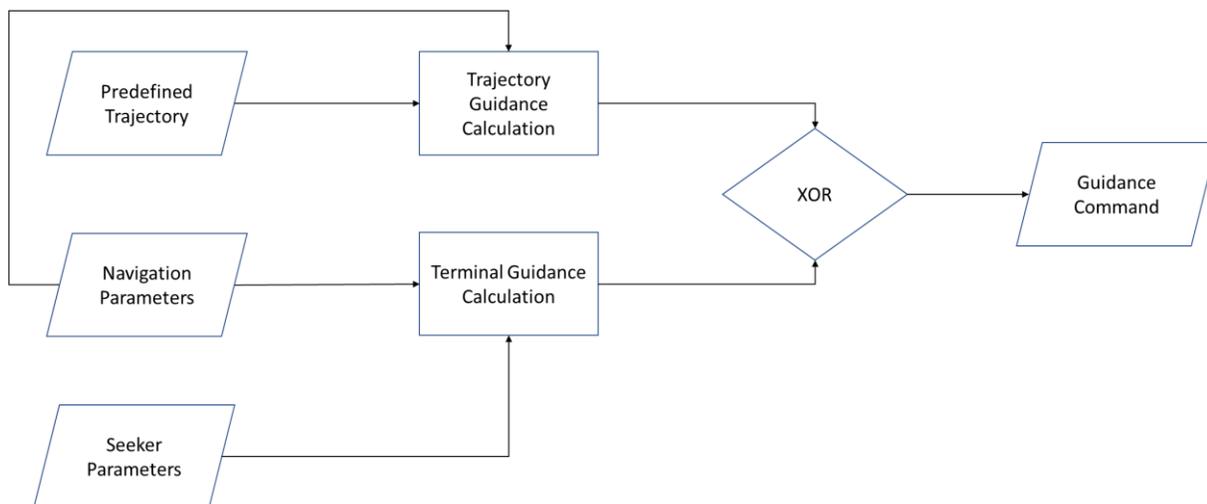


Figure 68 –Guidance Simulation

## 7.1.14. Control Simulation

Figure 69 shows the control simulation block diagram. Here, the roll demand of zero is inputted into the roll autopilot. In addition, the navigation parameters such as roll angle and rate are also inputted into the roll autopilot. The roll autopilot calculates the command using Figure 30. However, its uses Figure 32 for Roll Neural Network Autopilot. Moreover, the pitch demand from guidance is inputted into the pitch autopilot. The navigation parameters such as vertical acceleration, velocity, and pitch rate are also inputted into the pitch autopilot. The pitch autopilot calculates the command using Figure 31. However, it uses Figure 35 for Pitch Neural Network Autopilot. Furthermore, the yaw demand from guidance is inputted into the yaw autopilot. The navigation parameters such as side acceleration, velocity, and yaw rate are also inputted into the yaw autopilot. The yaw autopilot calculates the command using Figure 31. However, it uses Figure 35 for Yaw Neural Network Autopilot. It is important to note that the gains are varied with velocity for traditional autopilot and the type of autopilot is pre-selected before the simulation execution.
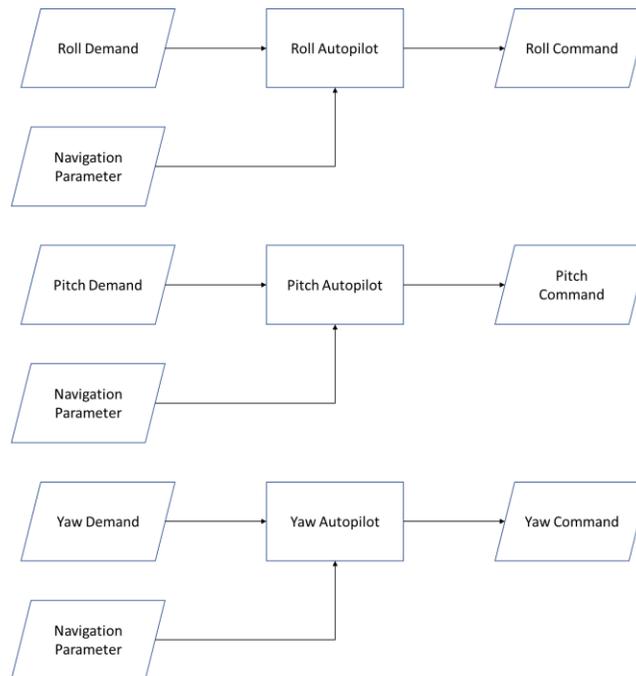
Figure 69 – Control Simulation

## 7.2. Performance Comparison

In order to verify the dependability of the Guidance, Navigation, and Control neural network algorithm, they will be compared to traditional algorithms. The comparison is done by implementing both neural network and traditional algorithms in a full missile simulation that is time variant and non-linear. This allows for an adequate estimation of the algorithms performance for a real system.

### 7.2.1. TGNN vs PN Guidance Comparison

The missile is launched at a constant 45degs launch angle. This angle is chosen for the comparison because it allows the missile to fly to the maximum range. In addition, it ensures sufficient altitude to evaluate the performance of the missile. However, it is important to note that since the launch angle is fixed, minimum range will cause problems. Hence, the minimum target range for this comparison is 8000m while the maximum target range is 14000m. Moreover, the minimum target deviation for this comparison is 0m while the maximum target deviation is 10000m in the negative direction.

Furthermore, the terminal guidance algorithm along with the lateral autopilot is turned on at 20 seconds as shown in Figure 70. This is to allow the missile to reach the apex before the guidance starts. These guidance scenarios are repeated for both terminal guidance algorithms. The missile miss distance will be used as the evaluation criteria.



Guidance Start
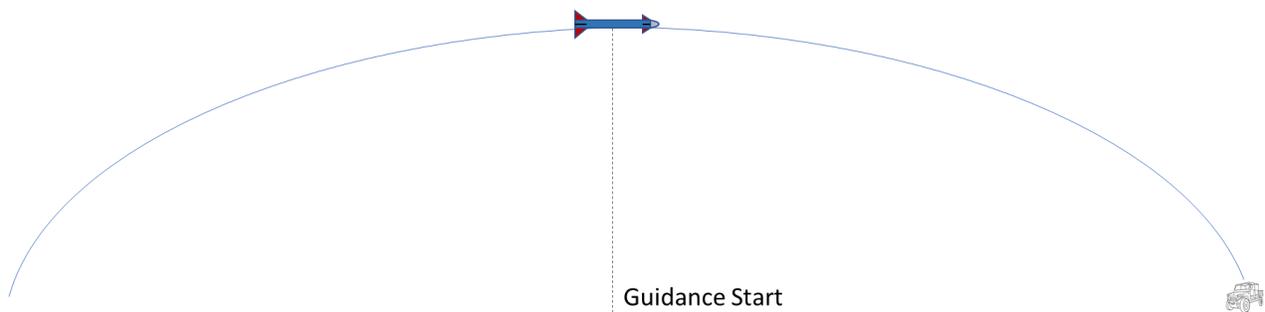
Figure 70 - Guidance Comparison Scenario

Figure 71 and Figure 72 show the performance of the missile for the target at 8000m range and 0m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 202m while PN is 1724m. However, PN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 928m while PN is 102m. This is because the TGNN algorithms design struggles with zero command. Moreover, TGNN took longer than PN to reach the altitude of the target.
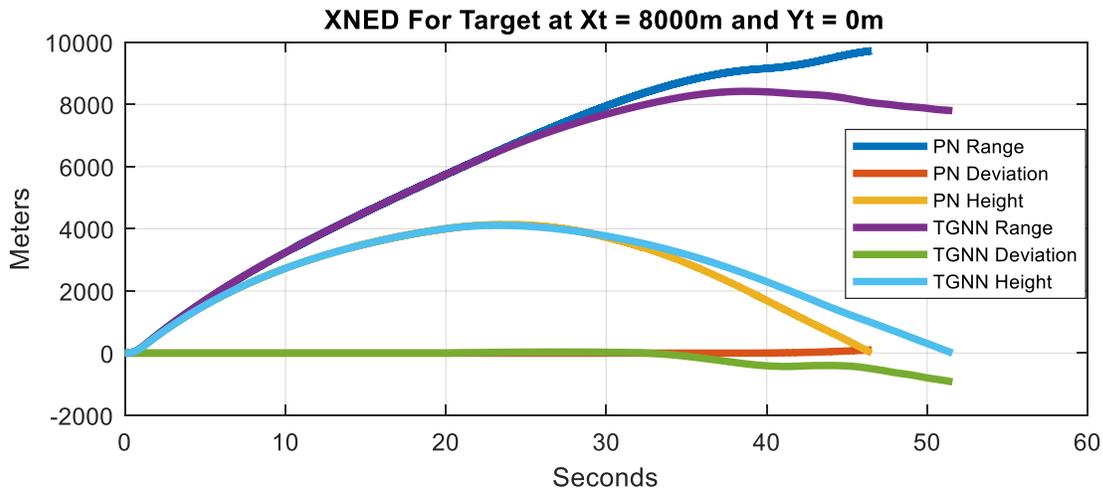


Figure 71 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 8000m and Yt = 0m
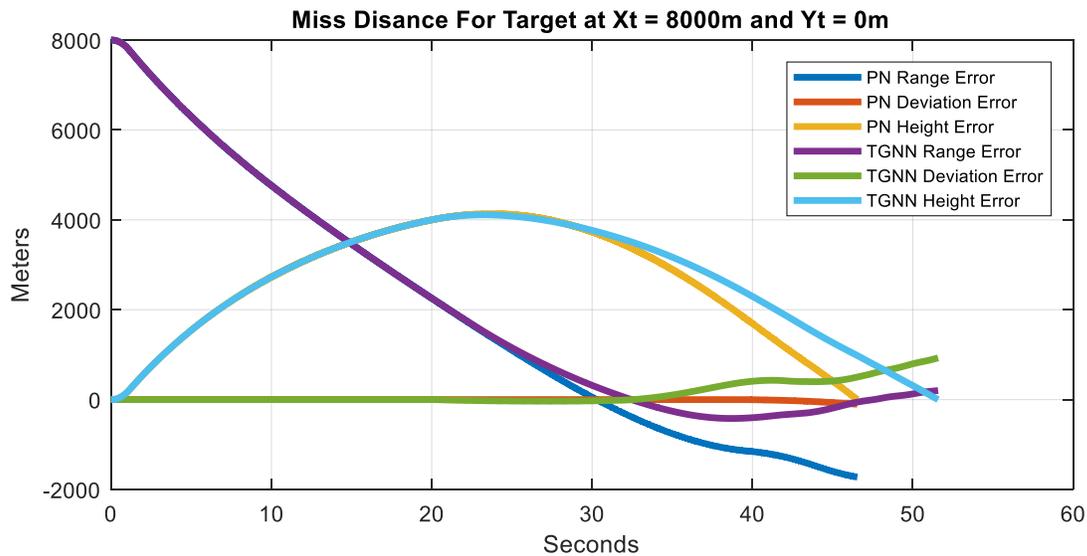


Figure 72 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 8000m and Yt = 0m

Figure 73 and Figure 74 show the performance of the missile for the target at 8000m range and -2000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 248m while PN is 801m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 205m while PN is 458m. It is important to note that TGNN took longer to reach the altitude of the target.
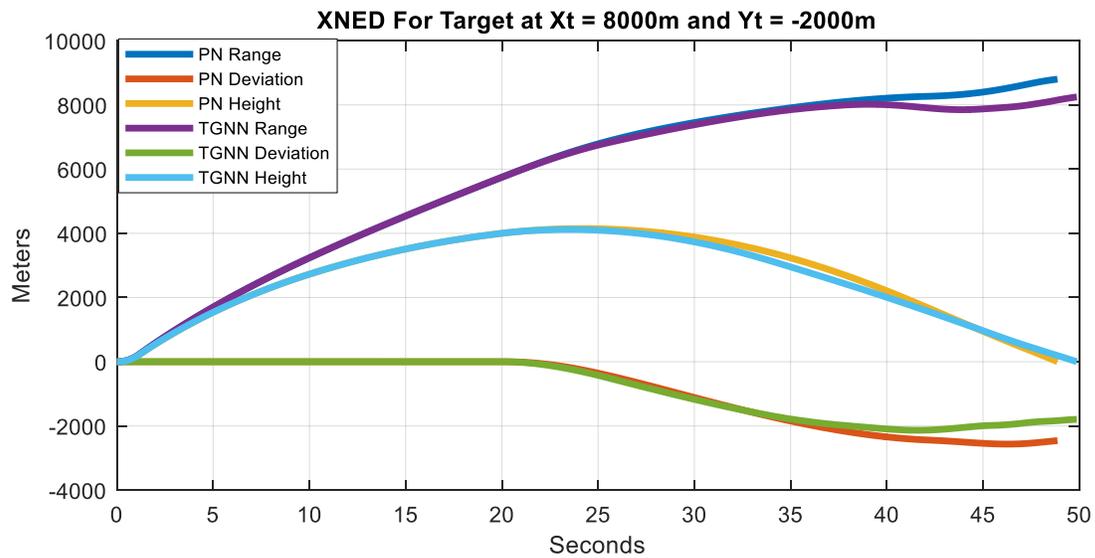


Figure 73 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 8000m and Yt = -2000m
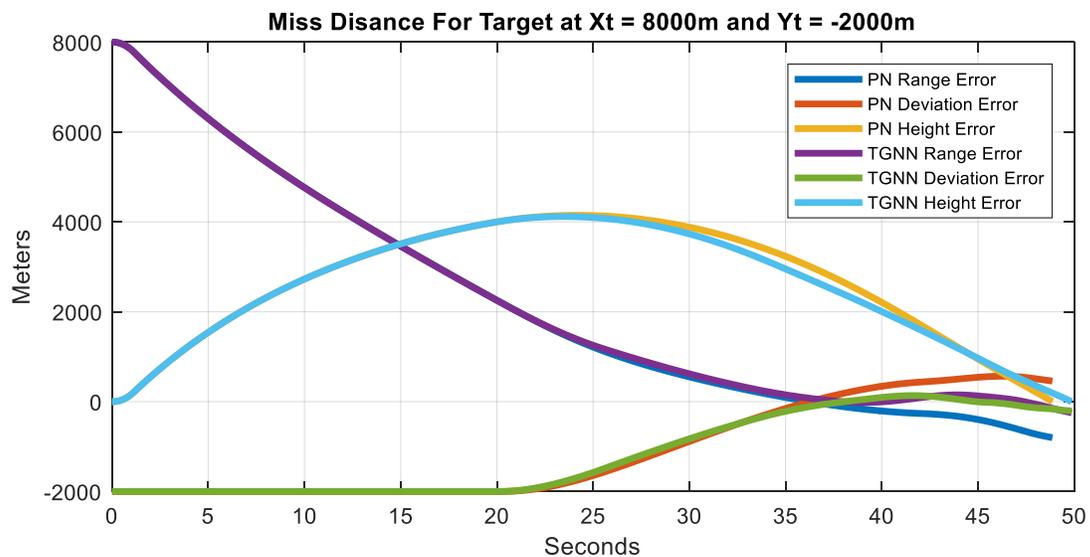


Figure 74 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 8000m and Yt = -2000m

Figure 75 and Figure 76 show the performance of the missile for the target at 8000m range and -4000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 0m while PN is 7m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 0m while PN is 623m. It is important to note that both TGNN and PN took the same amount of time to reach the altitude of the target.
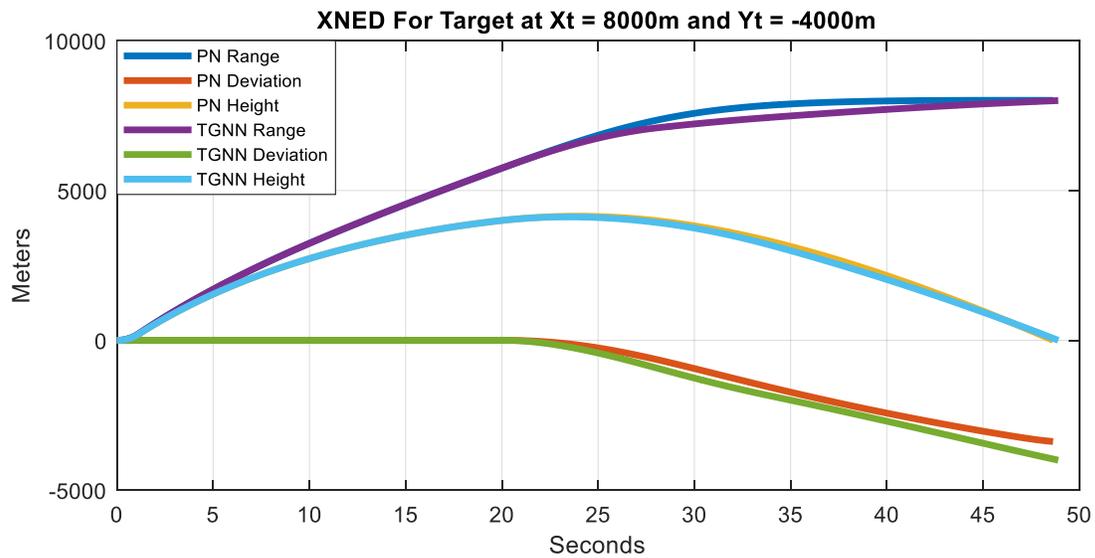


Figure 75 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 8000m and Yt = -4000m
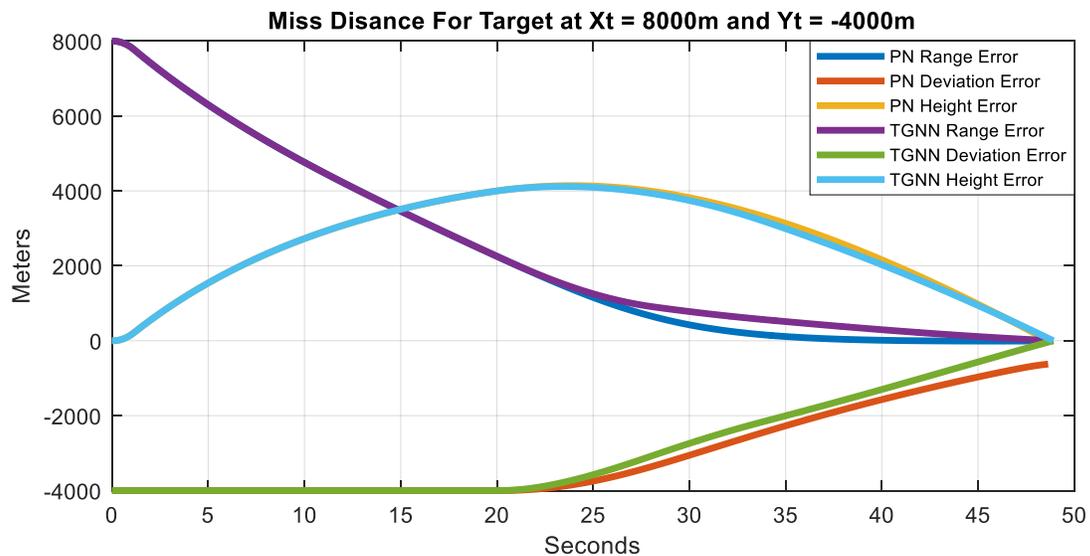


Figure 76 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 8000m and Yt = -4000m

Figure 77 and Figure 78 show the performance of the missile for the target at 8000m range and -6000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 3m while PN is 365m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 19m while PN is 4595m. It is important to note that TGNN took longer to reach the altitude of the target.
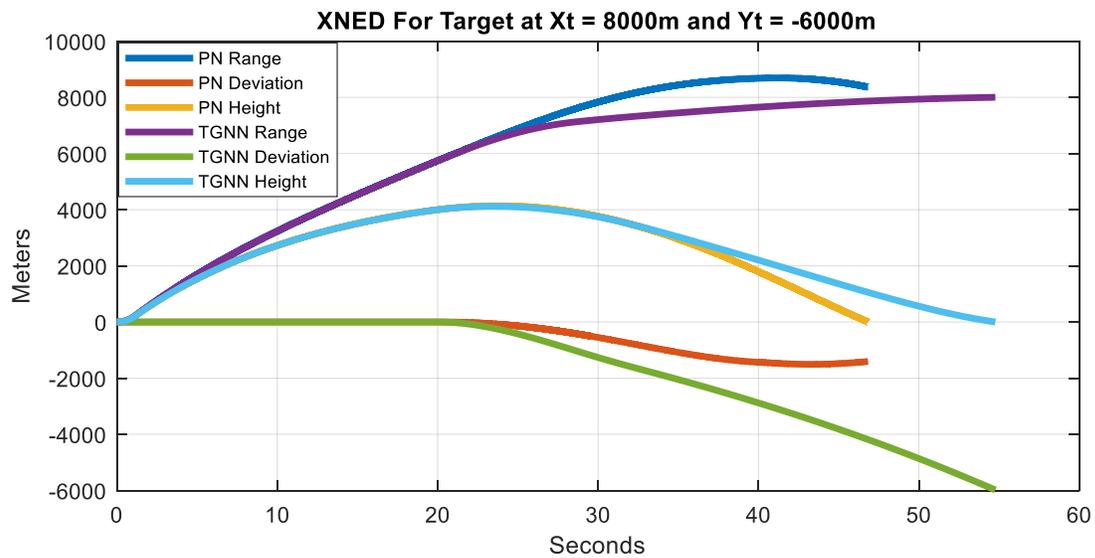


Figure 77 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 8000m and Yt = -6000m
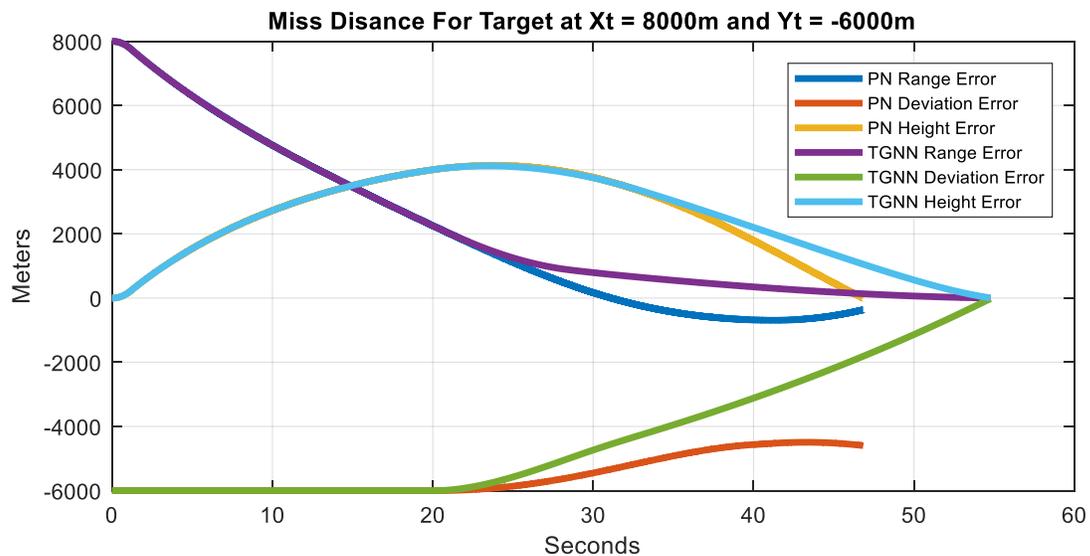


Figure 78 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 8000m and Yt = -6000m

Figure 79 and Figure 80 show the performance of the missile for the target at 8000m range and -8000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 0m while PN is 1540m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 3069m while PN is 6895m. It is important to note that TGNN took longer to reach the altitude of the target.
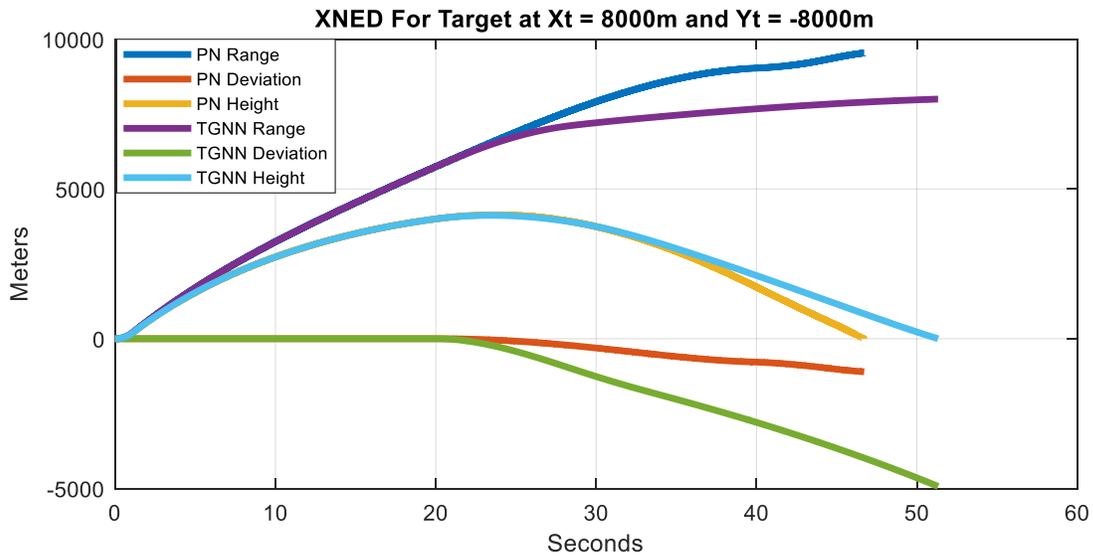


Figure 79 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 8000m and Yt = -8000m
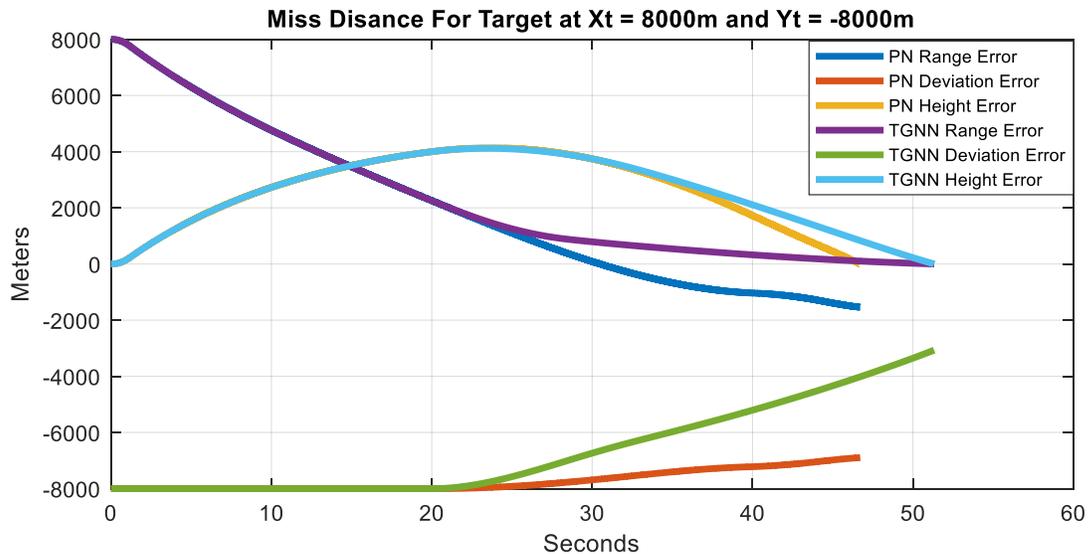


Figure 80 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 8000m and Yt = -8000m

Figure 81 and Figure 82 show the performance of the missile for the target at 8000m range and -10000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 0m while PN is 1683m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 6924m while PN is 9430m. It is important to note that TGNN took longer to reach the altitude of the target.
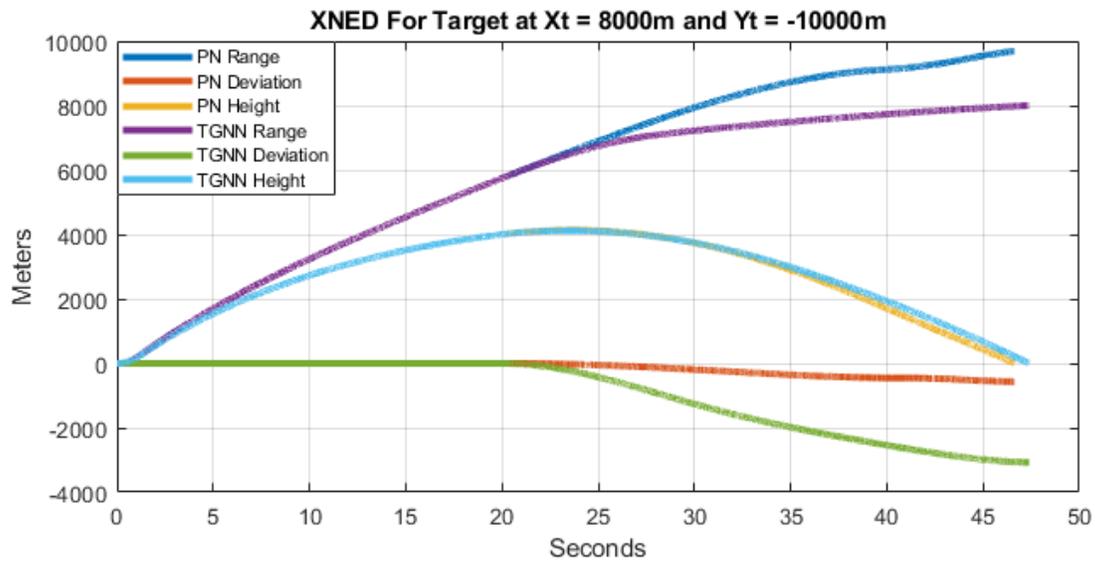


Figure 81 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 8000m and Yt = -10000m
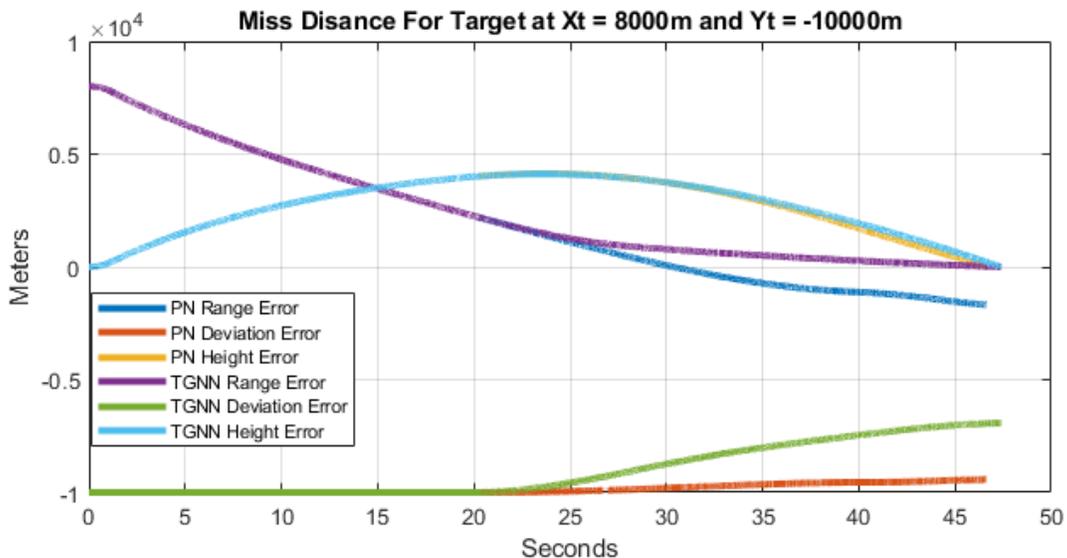


Figure 82 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 8000m and Yt = -10000m

Figure 83 and Figure 84 show the performance of the missile for the target at 10000m range and 0m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. In addition, TGNN and PN performed similarly for deviation. It can be seen that the deviation error for TGNN is 0m while PN is also 0m. However, TGNN deviated from the center axis before settling back to zero. It is important to note that TGNN and PN took the same amount of time to reach the altitude of the target.
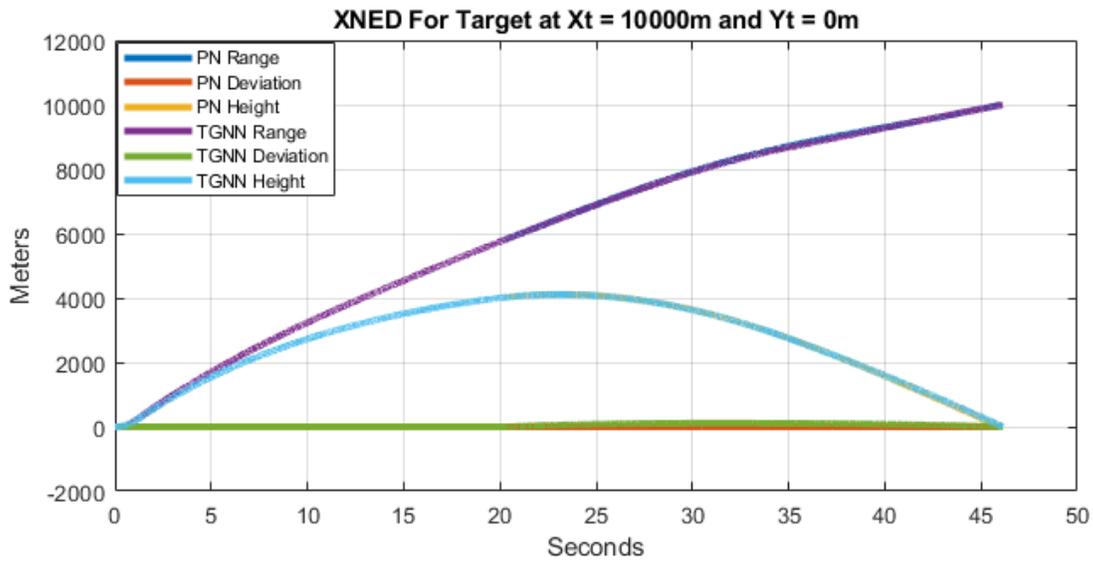


Figure 83 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 10000m and Yt = 0m
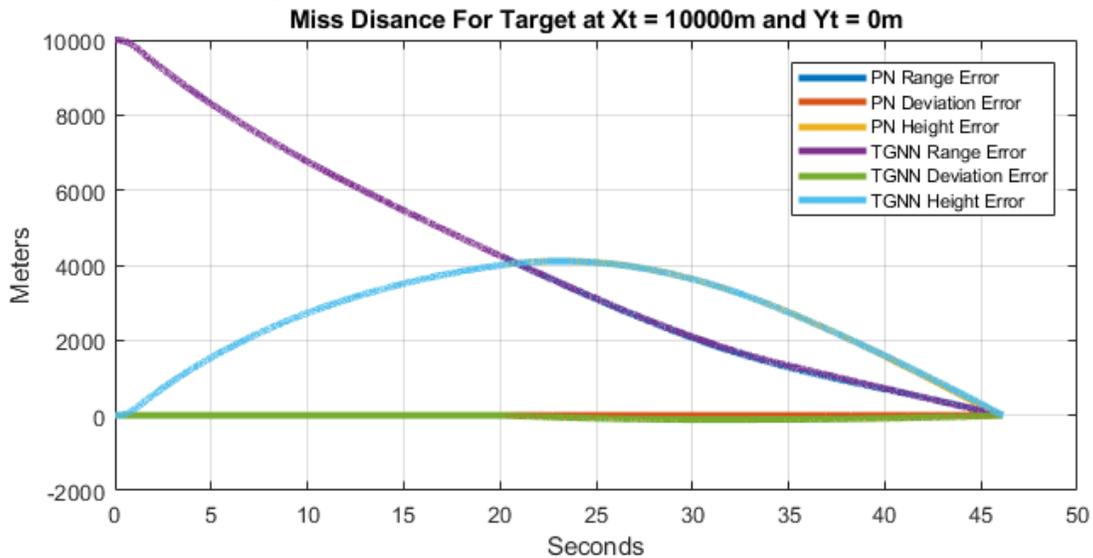


Figure 84 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 10000m and Yt = 0m

Figure 85 and Figure 86 show the performance of the missile for the target at 10000m range and -2000m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. In addition, TGNN and PN performed similarly for deviation. It can be seen that the deviation error for TGNN is 0m while PN is also 0m. It is important to note that TGNN and PN took the same amount of time to reach the altitude of the target.
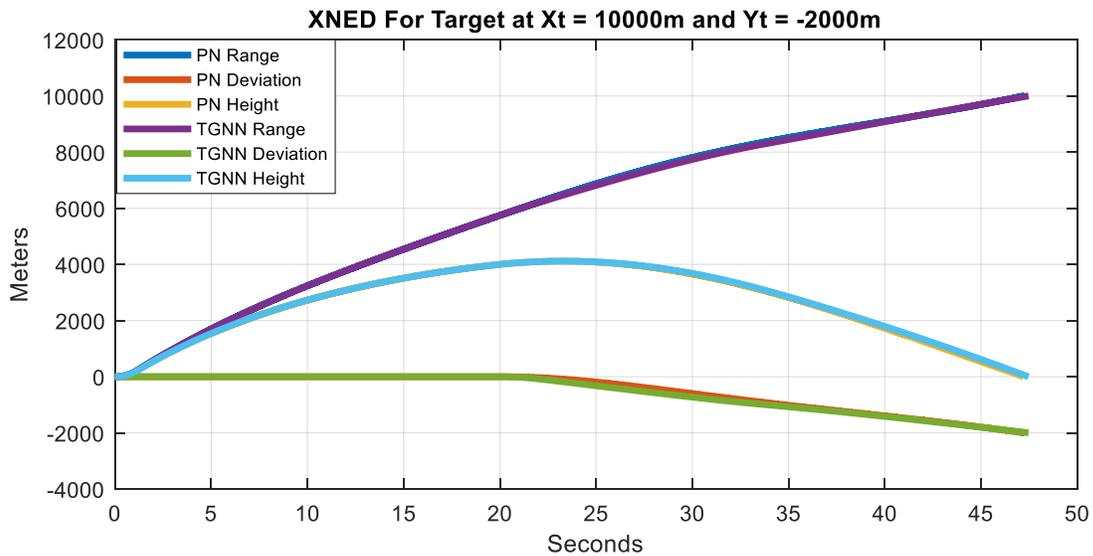


Figure 85 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 10000m and Yt = -2000m
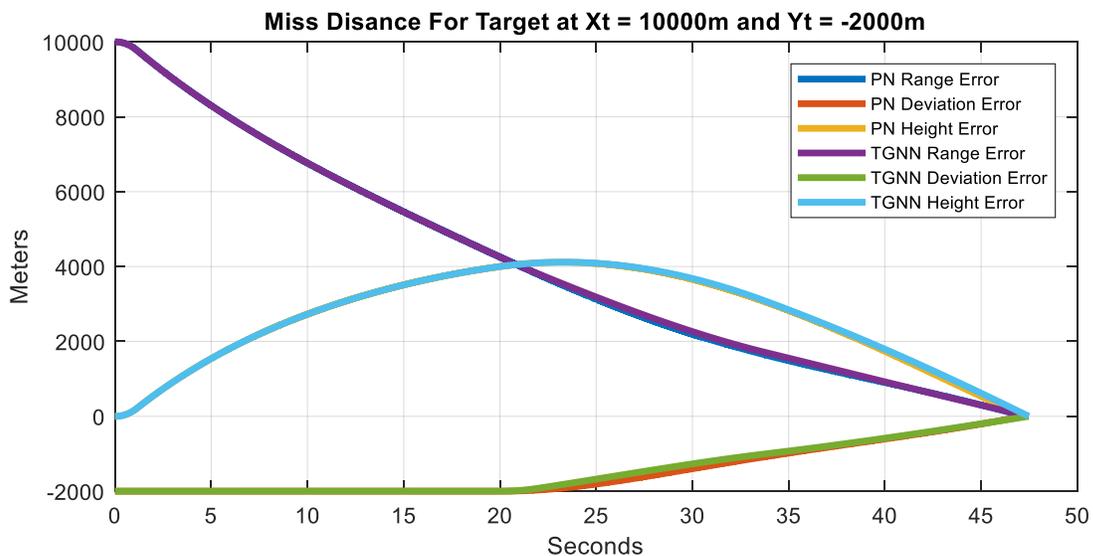


Figure 86 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 10000m and Yt = -2000m

Figure 87 and Figure 88 show the performance of the missile for the target at 10000m range and -4000m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. In addition, PN performed a little bit better than TGNN for deviation. It can be seen that the deviation error for TGNN is 0.5m while PN is 0m. It is important to note that TGNN and PN took the same amount of time to reach the altitude of the target.
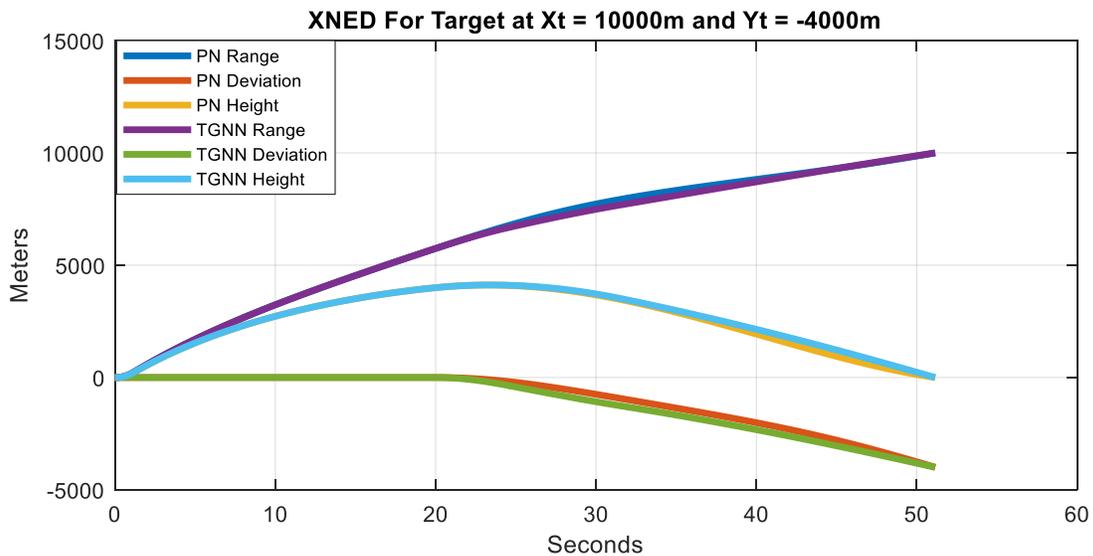


Figure 87 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 10000m and Yt = -4000m
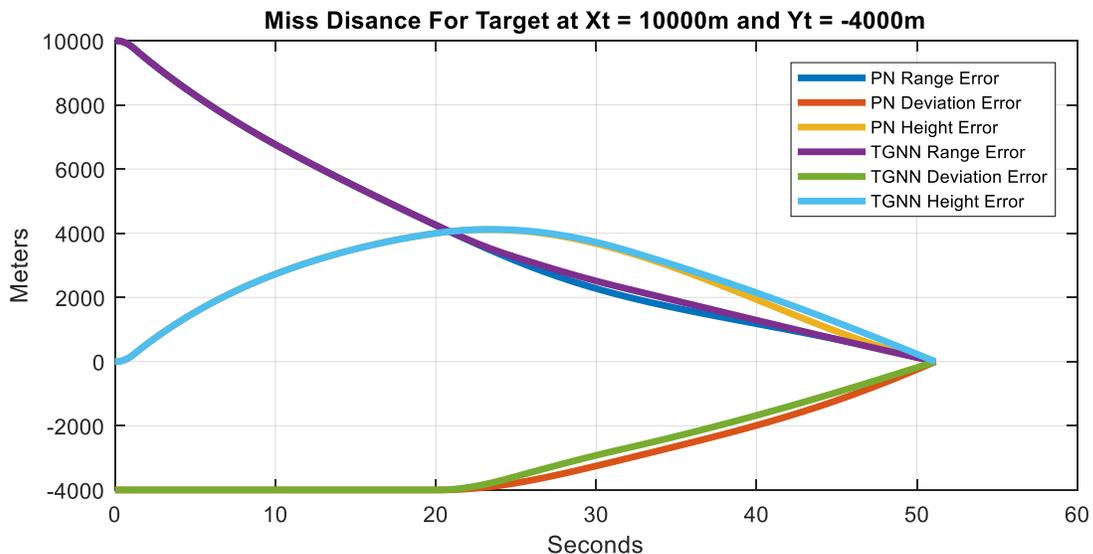


Figure 88 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 10000m and Yt = -4000m

Figure 89 and Figure 90 show the performance of the missile for the target at 10000m range and -6000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 1.3m while PN is 107m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 0m while PN is 538m. It is important to note that TGNN and PN took the same amount of time to reach the altitude of the target.
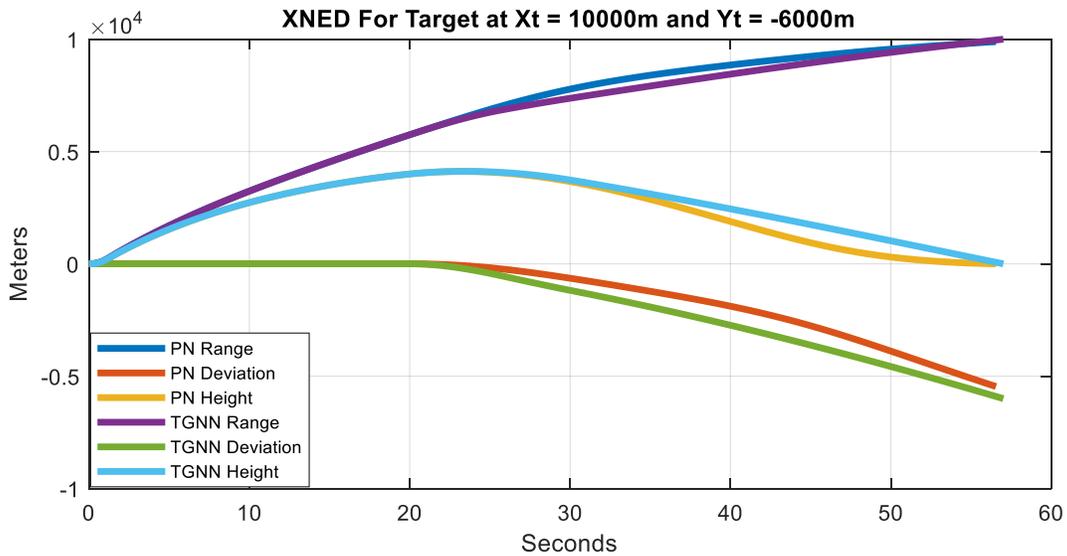


Figure 89 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 10000m and Yt = -6000m
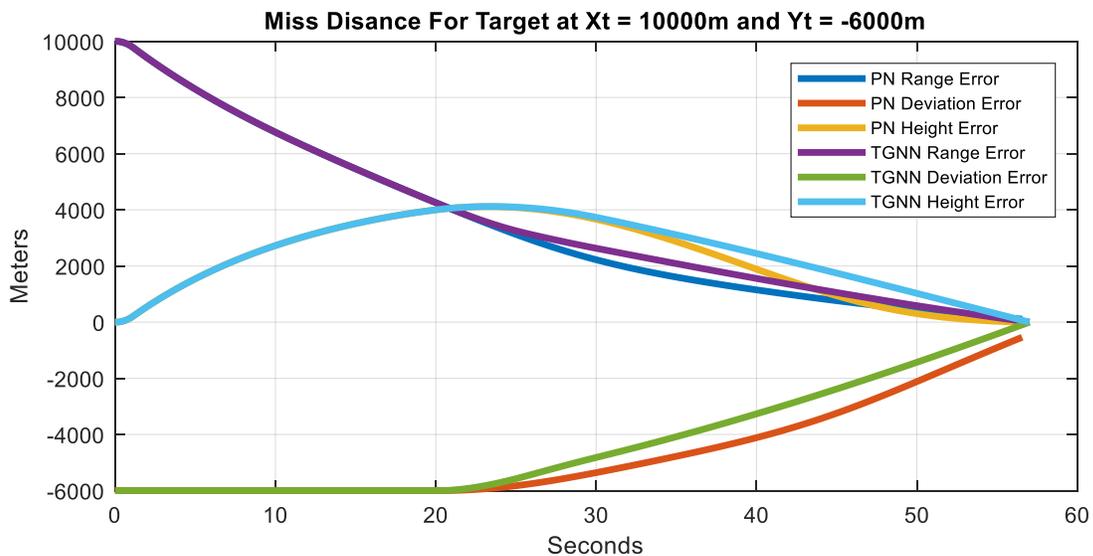


Figure 90 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 10000m and Yt = -6000m

Figure 91 and Figure 92 show the performance of the missile for the target at 10000m range and -8000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 4m while PN is 124m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 14m while PN is 4831m. It is important to note that TGNN took longer to reach the altitude of the target.
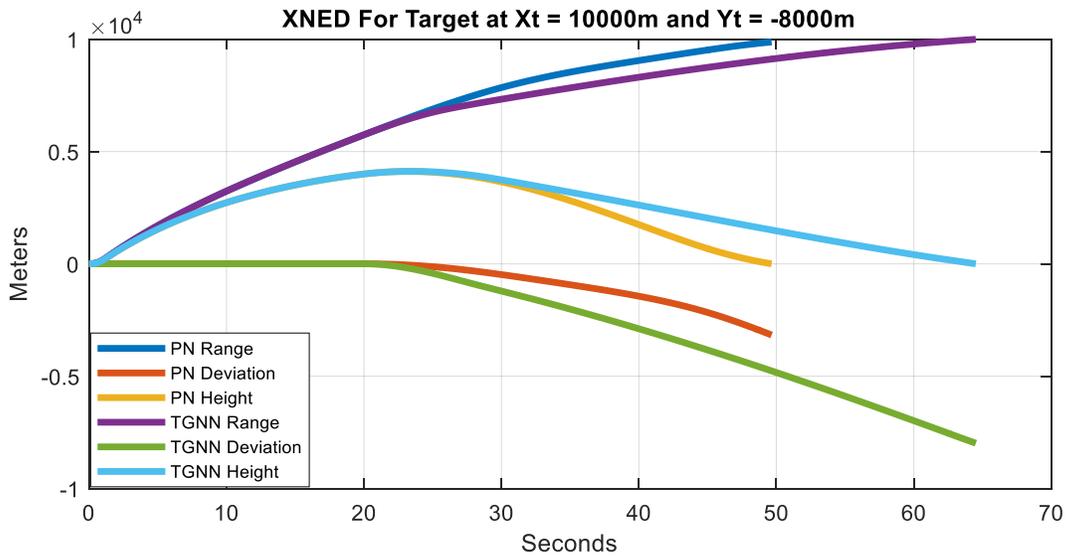


Figure 91 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 10000m and Yt = -8000m
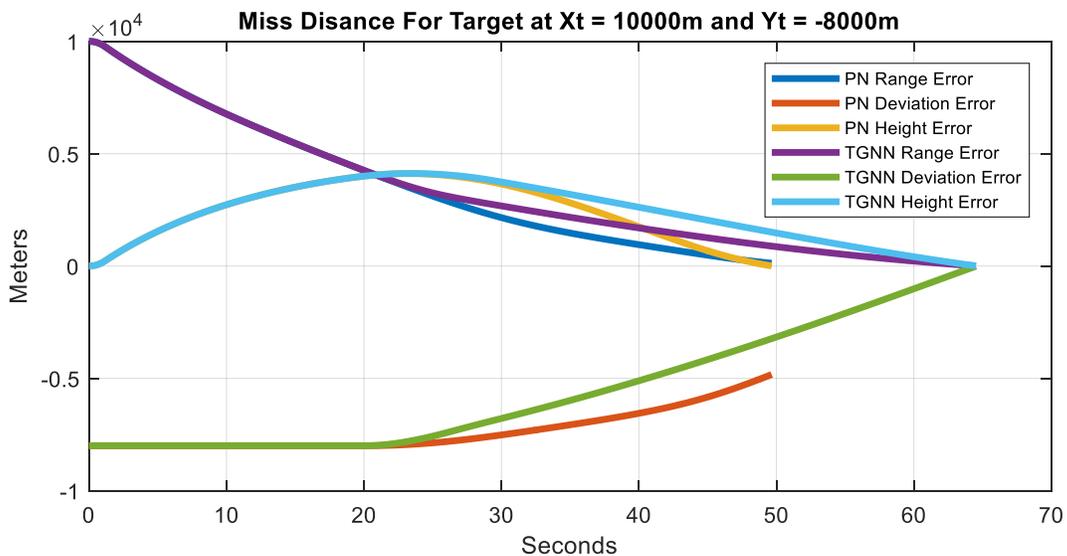


Figure 92 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 10000m and Yt = -8000m

Figure 93 and Figure 94 show the performance of the missile for the target at 10000m range and -10000m deviation. Here, it can be seen that for the range PN performed better than TGNN. The TGNN range error is 10m while PN is 2m. However, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 114m while PN is 8226m. It is important to note that TGNN took longer to reach the altitude of the target.
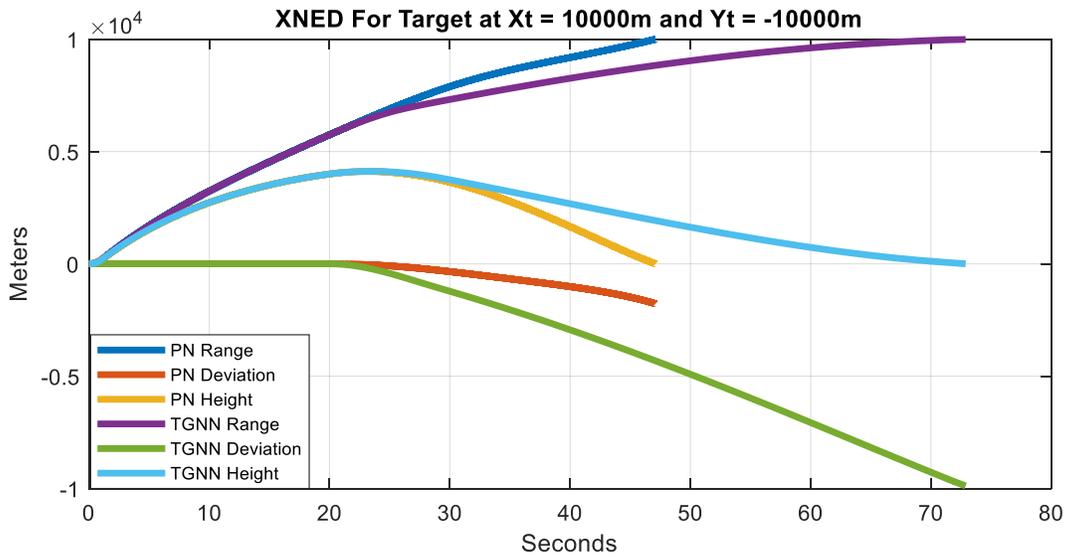


Figure 93 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 10000m and Yt = -10000m
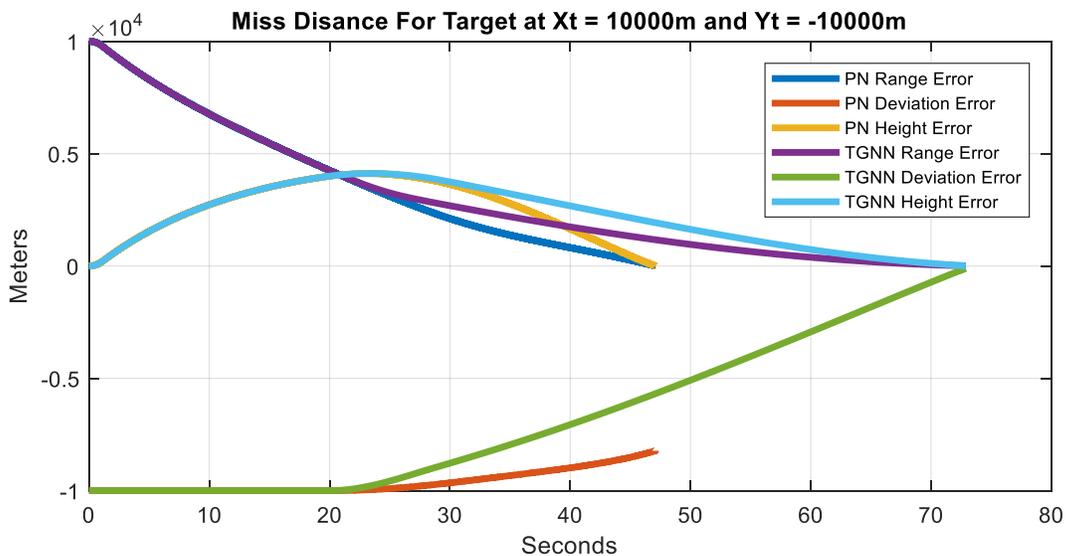


Figure 94 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 10000m and Yt = -10000m

Figure 95 and Figure 96 show the performance of the missile for the target at 12000m range and 0m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. In addition, TGNN and PN performed similarly for deviation. It can be seen that the deviation error for TGNN is 0m while PN is also 0m. However, TGNN deviated from the center axis before settling back to zero. It is important to note that TGNN and PN took the same amount of time to reach the altitude of the target.
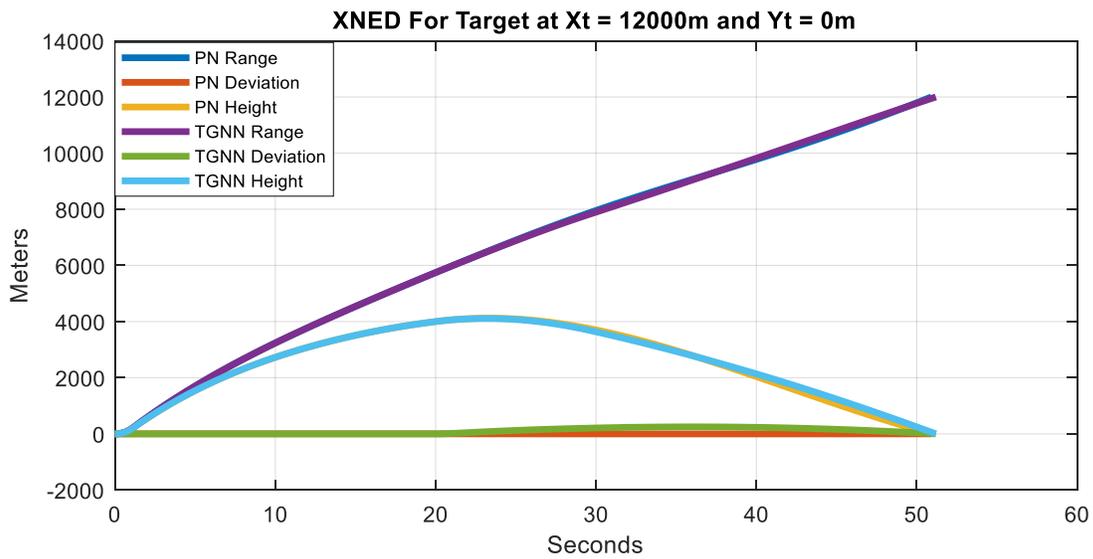


Figure 95 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 12000m and Yt = 0m
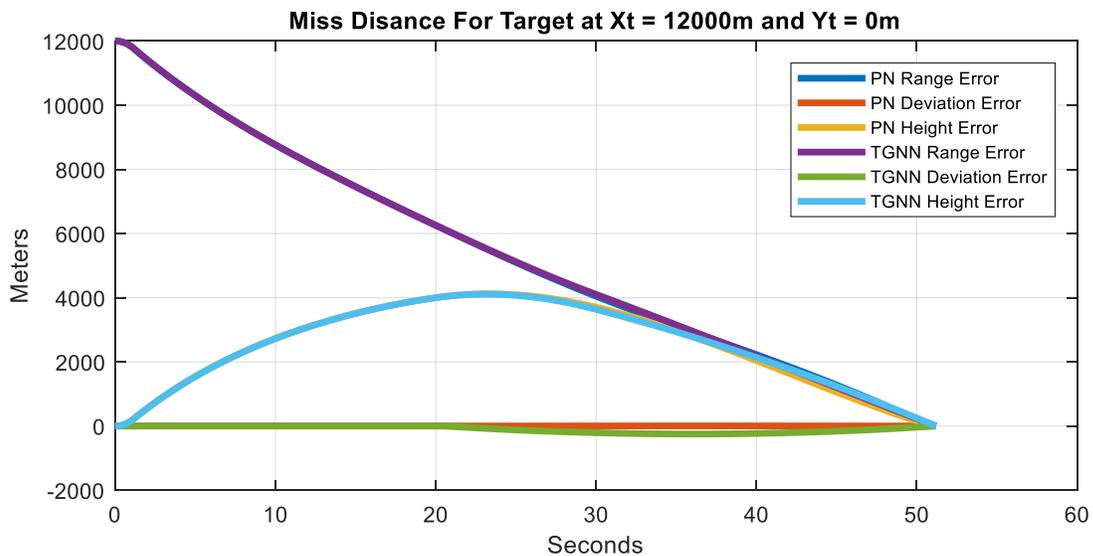


Figure 96 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 12000m and Yt = 0m

Figure 97 and Figure 98 show the performance of the missile for the target at 12000m range and -2000m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. In addition, TGNN and PN performed similarly for deviation. It can be seen that the deviation error for TGNN is 0m while PN is also 0m. It is important to note that TGNN and PN took the same amount of time to reach the altitude of the target.
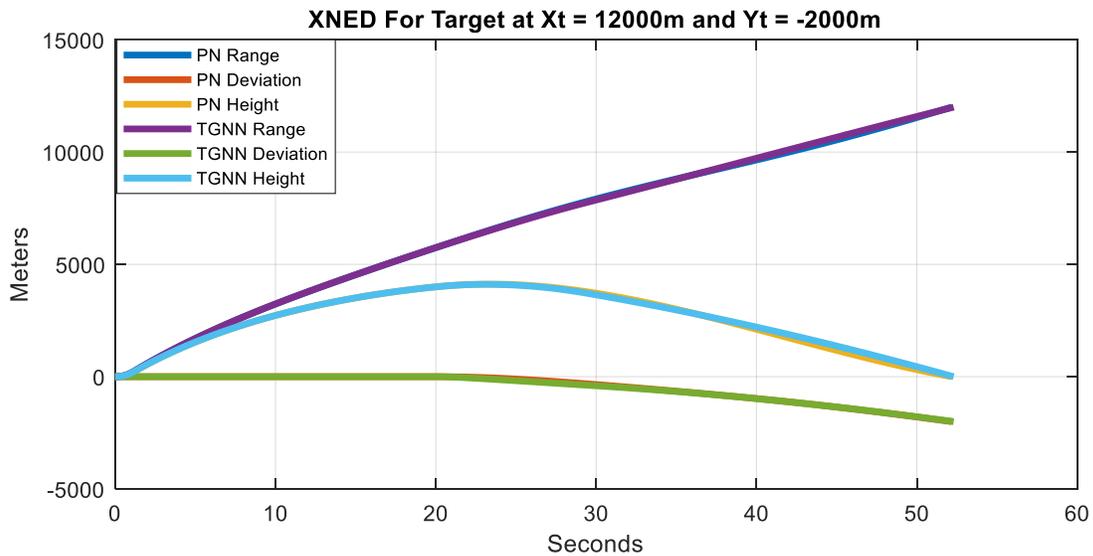


Figure 97 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 12000m and Yt = -2000m
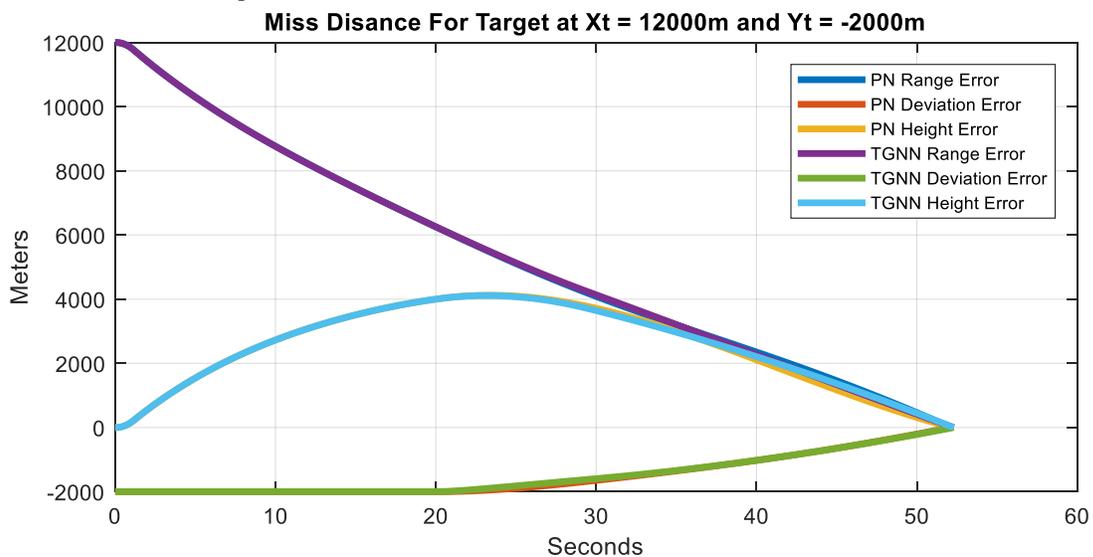


Figure 98 Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 12000m and Yt = -2000m

Figure 99 and Figure 100 show the performance of the missile for the target at 12000m range and -4000m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. In addition, TGNN and PN performed similarly for deviation. It can be seen that the deviation error for TGNN is 0m while PN is also 0m. It is important to note that TGNN and PN took the same amount of time to reach the altitude of the target.
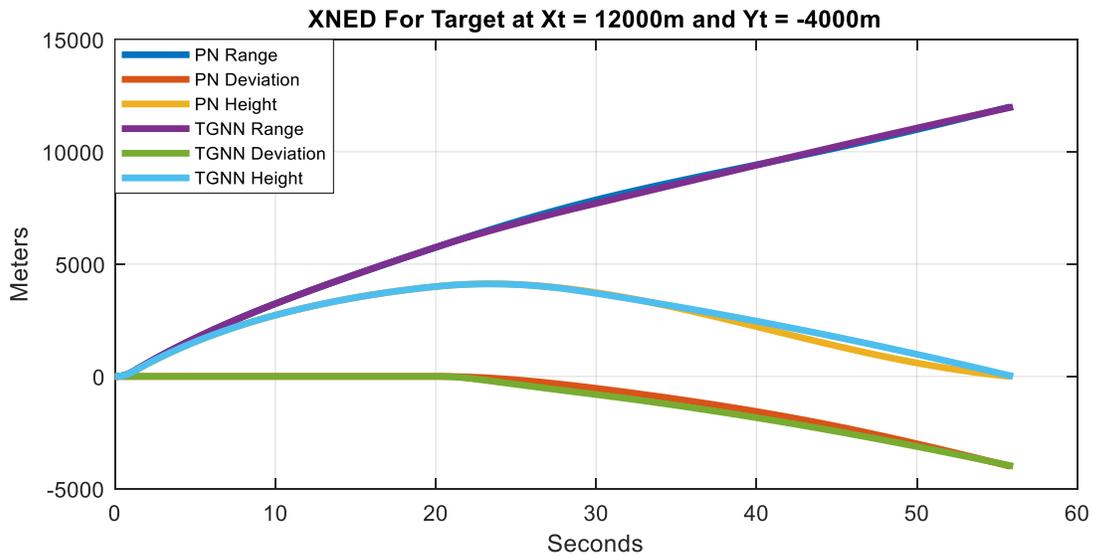


Figure 99 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 12000m and Yt = -4000m
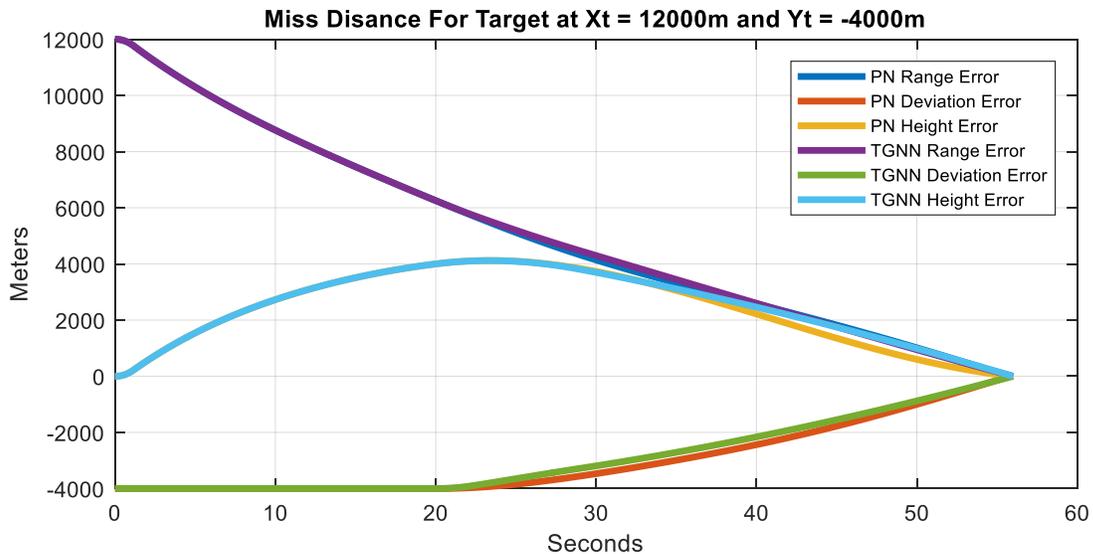


Figure 100 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 12000m and Yt = -4000m

Figure 101 and Figure 102 show the performance of the missile for the target at 12000m range and -6000m deviation. Here, it can be seen that for TGNN performed better than PN. The TGNN range error is 3m while PN is also 79m. In addition, TGNN performed better than PN for deviation. It can be seen that the deviation error for TGNN is 0m while PN is 144m. It is important to note that TGNN and PN took the same amount of time to reach the altitude of the target.
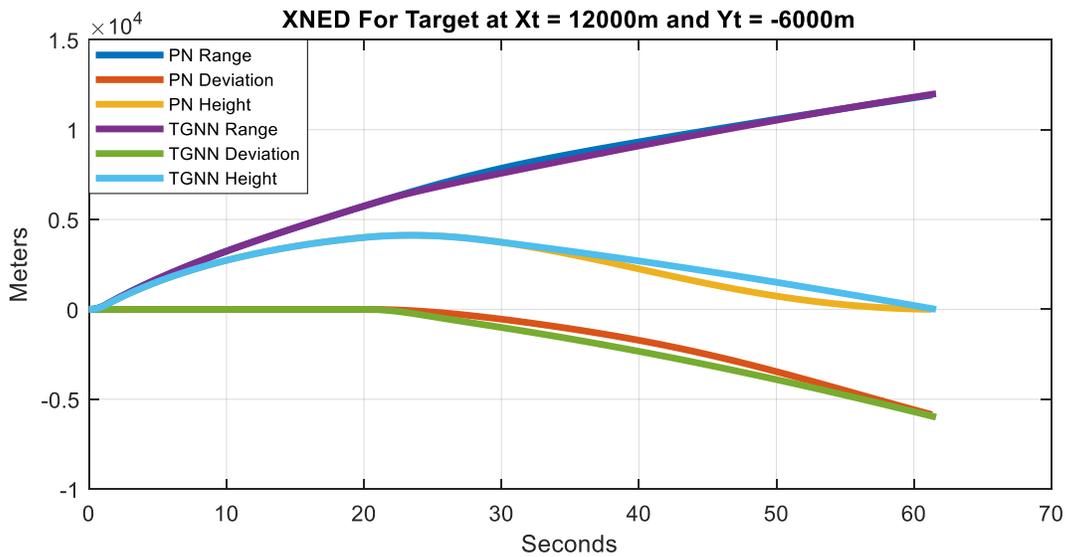


Figure 101 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 12000m and Yt = -6000m
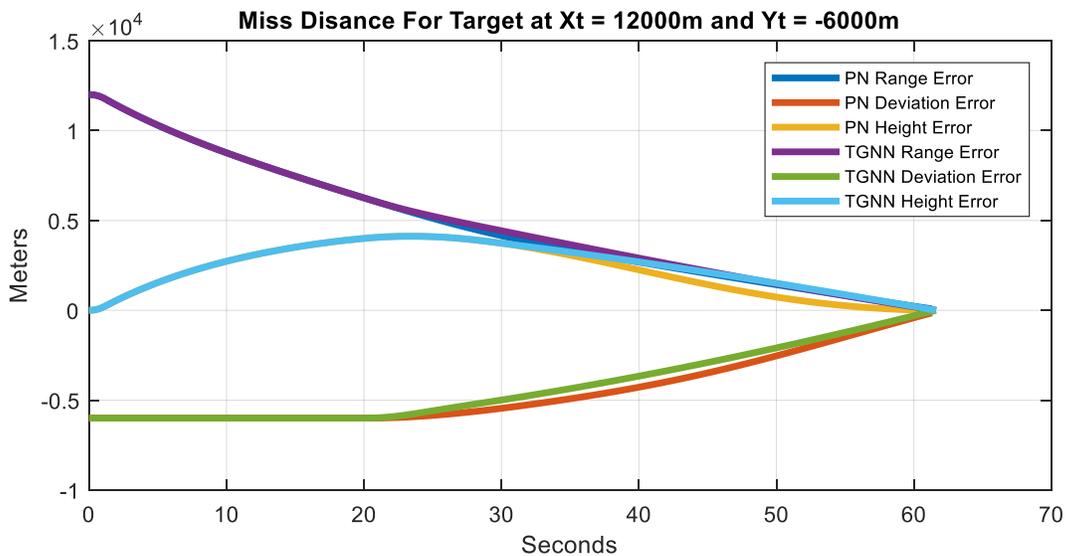


Figure 102 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 12000m and Yt = -6000m

Figure 103 and Figure 104 show the performance of the missile for the target at 12000m range and -8000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 5m while PN is 691m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 8m while PN is 2408m. It is important to note that TGNN took longer to reach the altitude of the target.
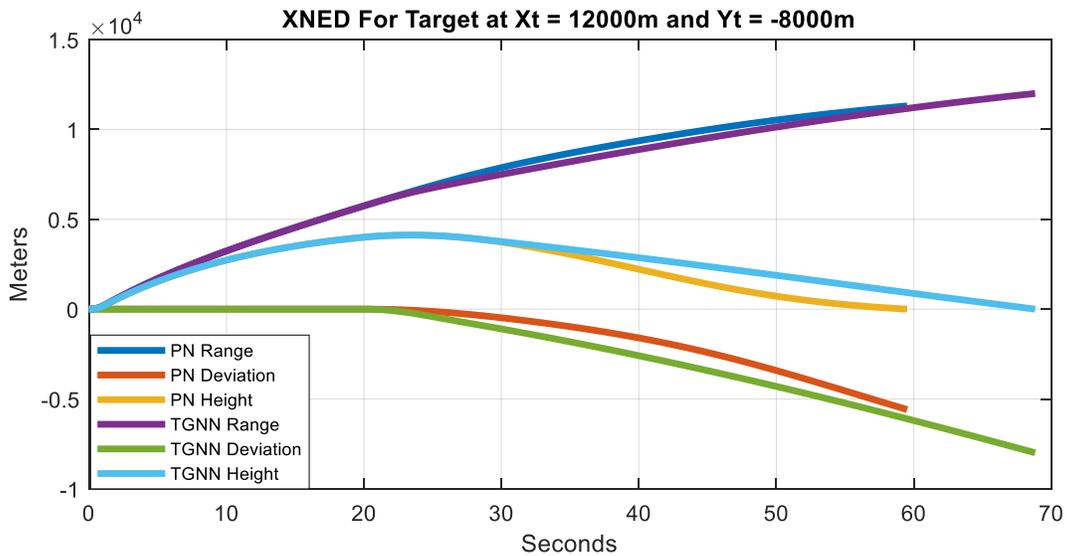


Figure 103 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 12000m and Yt = -8000m
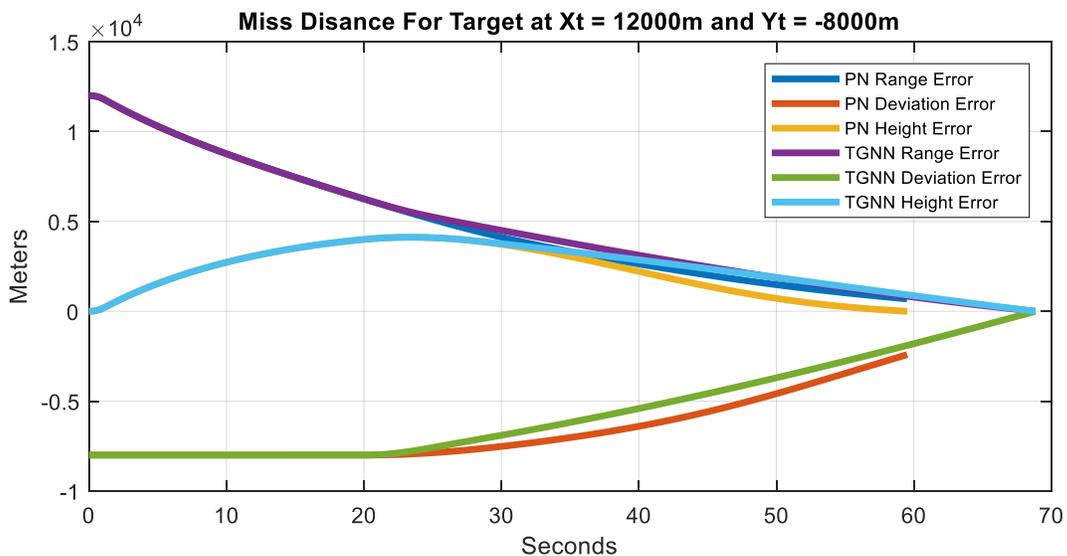


Figure 104 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 12000m and Yt = -8000m

Figure 105 and Figure 106 show the performance of the missile for the target at 12000m range and -10000m deviation. Here, it can be seen that for the range TGNN performed better than PN. The TGNN range error is 8m while PN is 599m. In addition, TGNN performed better for deviation. Here, it can be seen that the deviation error for TGNN is 28m while PN is 4928m. It is important to note that TGNN took longer to reach the altitude of the target.
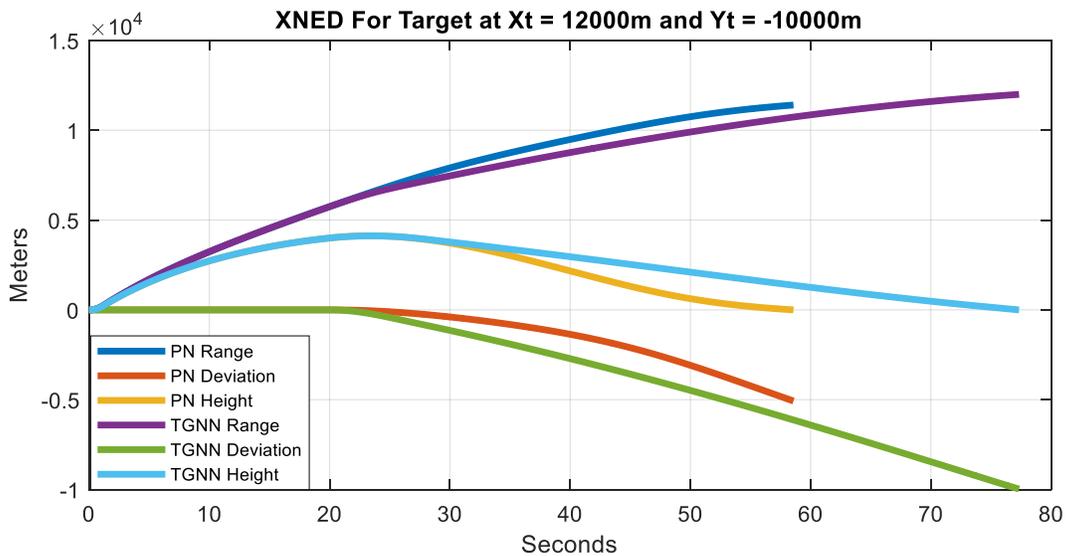


Figure 105 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 12000m and Yt = -10000m
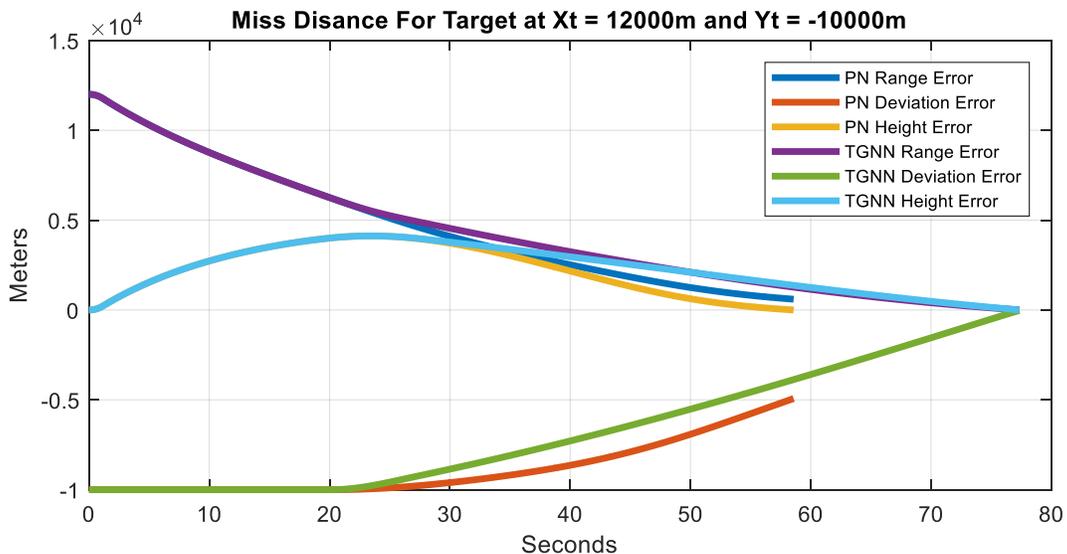


Figure 106 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 12000m and Yt = -10000m

Figure 107 and Figure 108 show the performance of the missile for the target at 14000m range and 0m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. In addition, TGNN and PN performed similarly for deviation. It can be seen that the deviation error for TGNN is 0m while PN is also 0m. However, TGNN deviated from the center axis before settling back to zero. It is important to note that TGNN took a little bit longer to reach the altitude of the target.
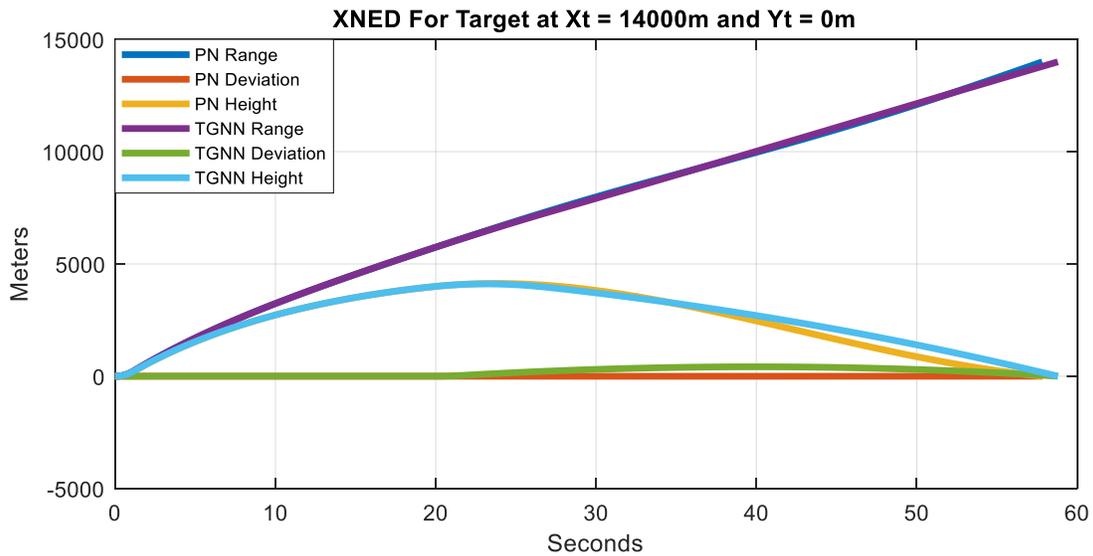


Figure 107 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 14000m and Yt = 0m
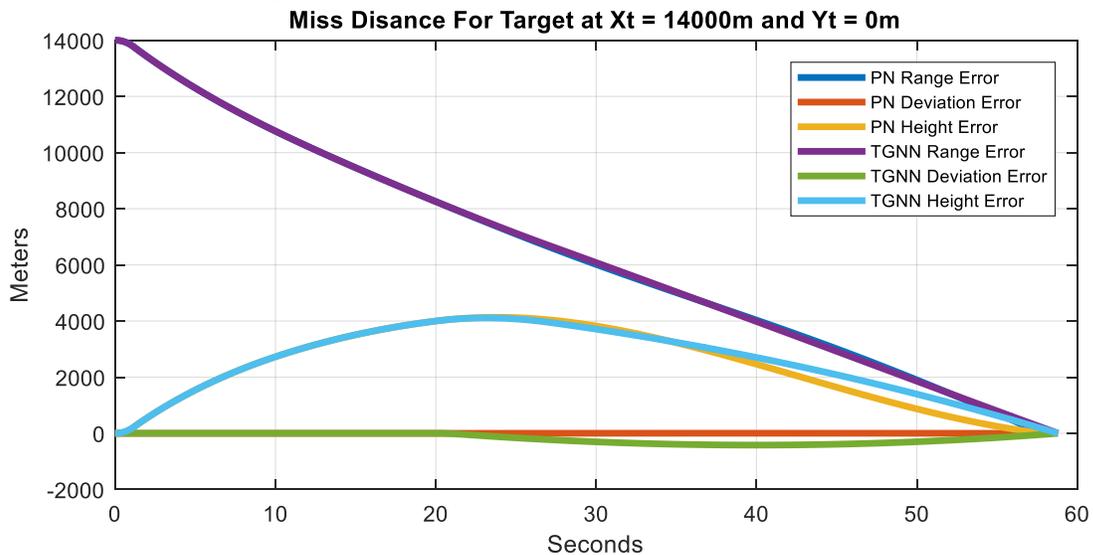


Figure 108 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 14000m and Yt = 0m

Figure 109 and Figure 110 show the performance of the missile for the target at 14000m range and -2000m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. In addition, TGNN and PN performed similarly for deviation. It can be seen that the deviation error for TGNN is 0m while PN is also 0m. It is important to note that TGNN took a little bit longer to reach the altitude of the target.
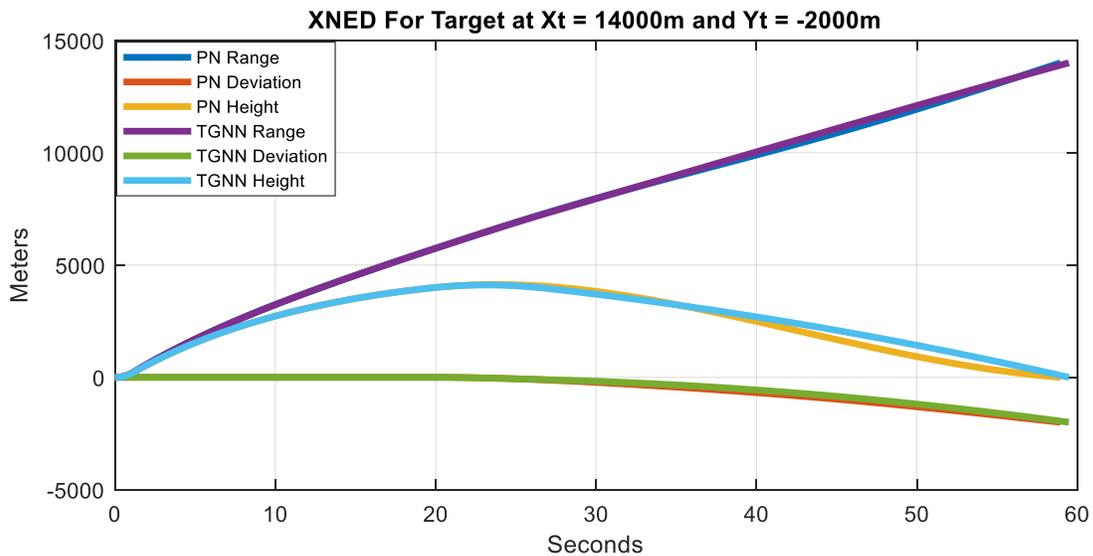


Figure 109 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 14000m and Yt = -2000m
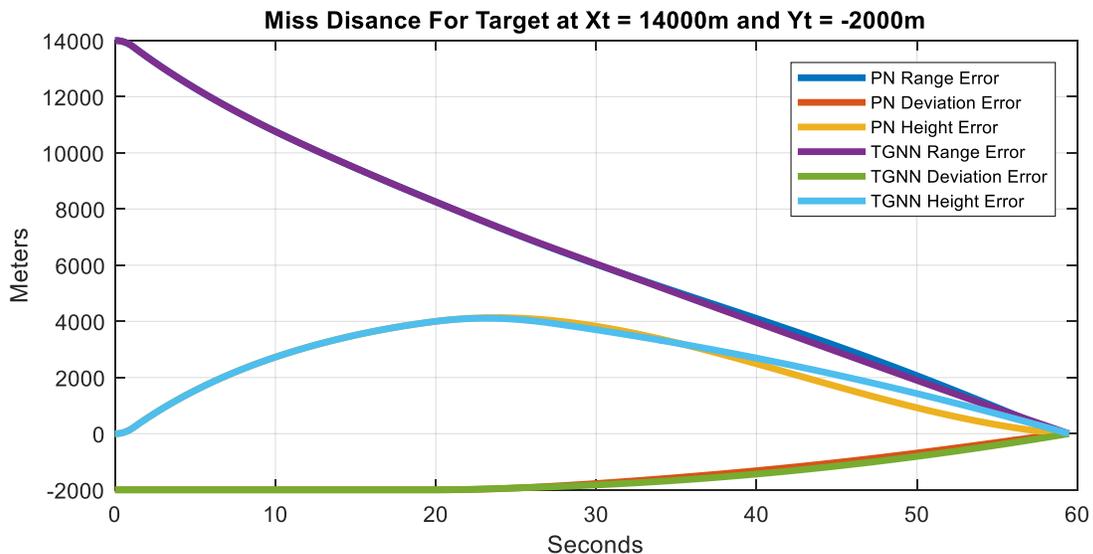


Figure 110 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 14000m and Yt = -2000m

Figure 111 and Figure 112 show the performance of the missile for the target at 14000m range and -4000m deviation. Here, it can be seen that for the both TGNN and PN performed similarly. The TGNN range error is 0m while PN is also 0m. However, TGNN performed better than PN for deviation. It can be seen that the deviation error for TGNN is 0m while PN is 4m. It is important to note that TGNN took a little bit longer to reach the altitude of the target.
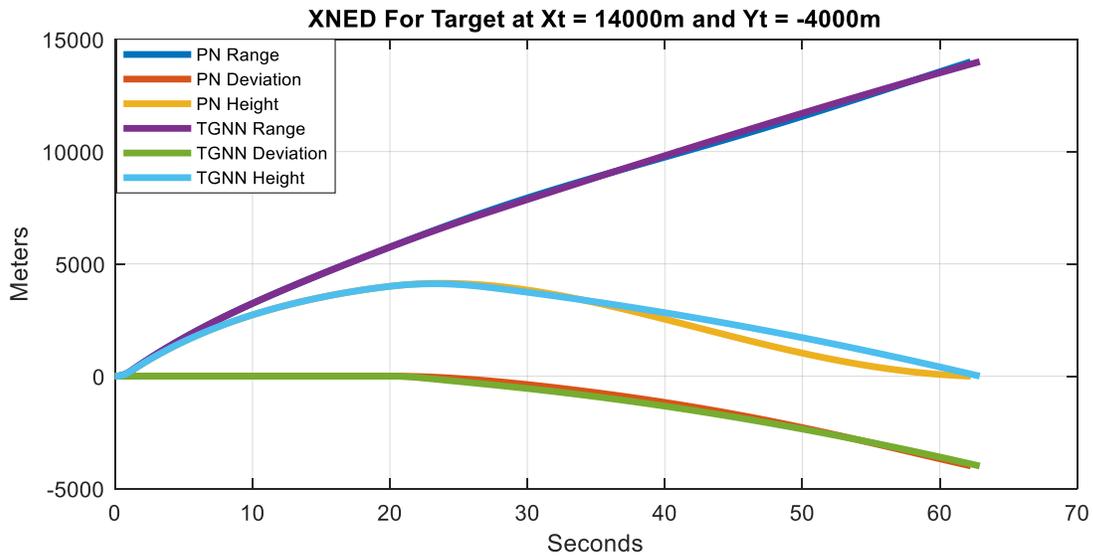


Figure 111 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 14000m and Yt = -4000m
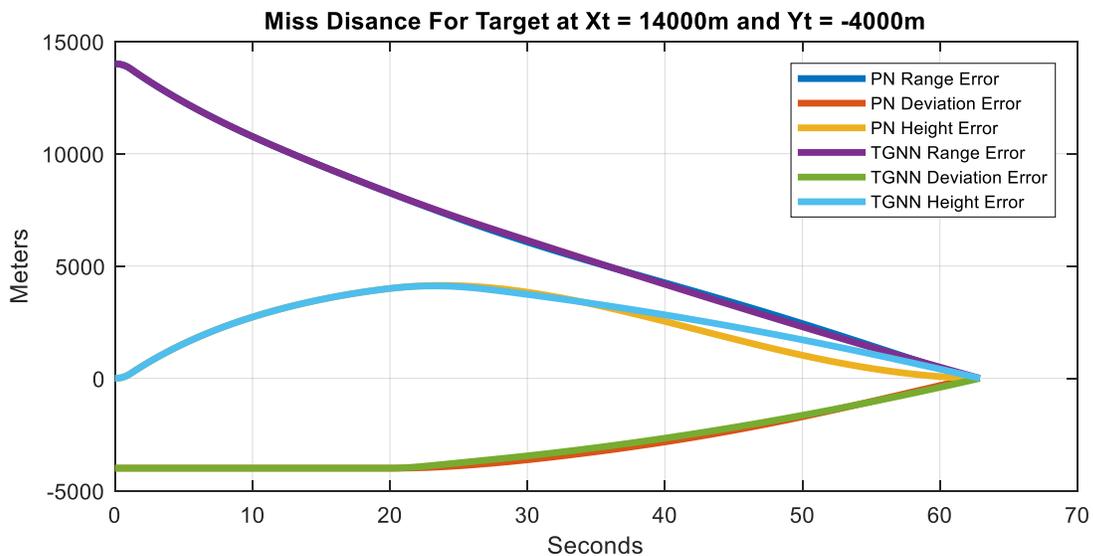


Figure 112 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 14000m and Yt = -4000m

Figure 113 and Figure 114 show the performance of the missile for the target at 14000m range and -6000m deviation. Here, it can be seen that TGNN performed better than PN. The TGNN range error is 1m while PN is also 615m. In addition, TGNN performed better than PN for deviation. It can be seen that the deviation error for TGNN is 1.3m while PN is 733m. It is important to note that TGNN took longer to reach the altitude of the target.
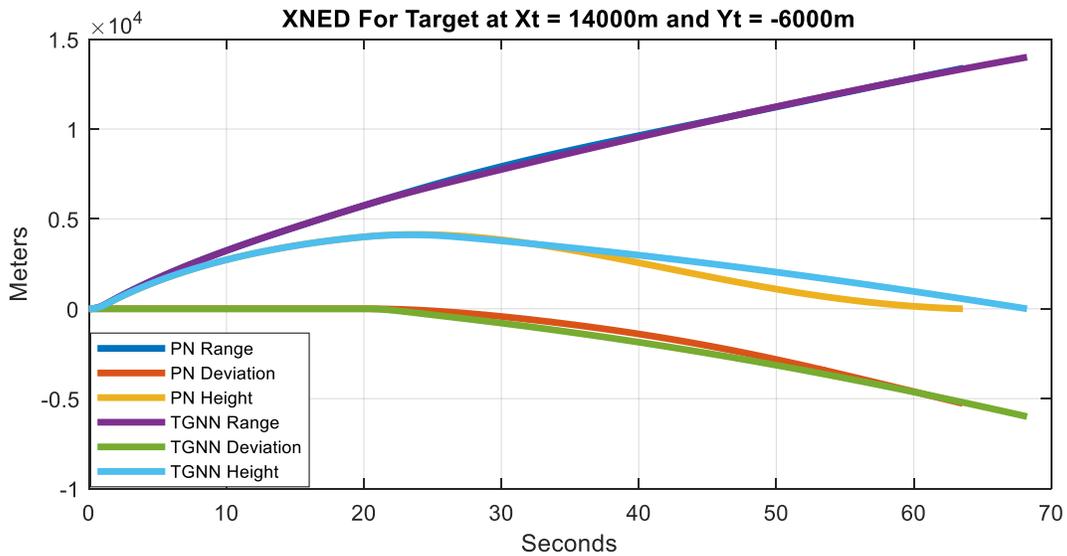


Figure 113 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 14000m and Yt = -6000m
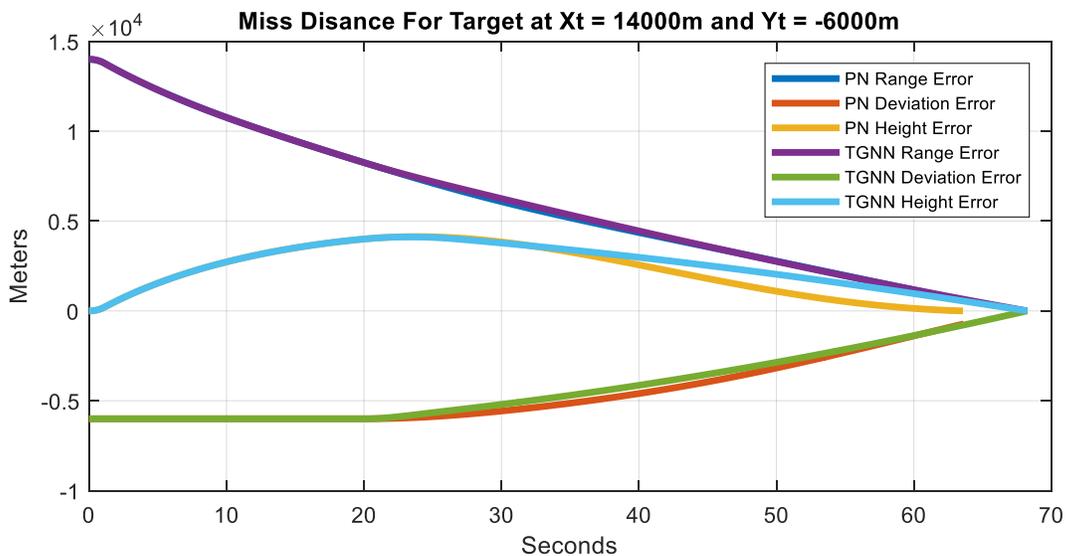


Figure 114 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 14000m and Yt = -6000m

Figure 115 and Figure 116 show the performance of the missile for the target at 14000m range and -8000m deviation. Here, it can be seen that TGNN performed better than PN. The TGNN range error is 3m while PN is also 1383m. In addition, TGNN performed better than PN for deviation. It can be seen that the deviation error for TGNN is 1.5m while PN is 2571m. It is important to note that TGNN took longer to reach the altitude of the target.
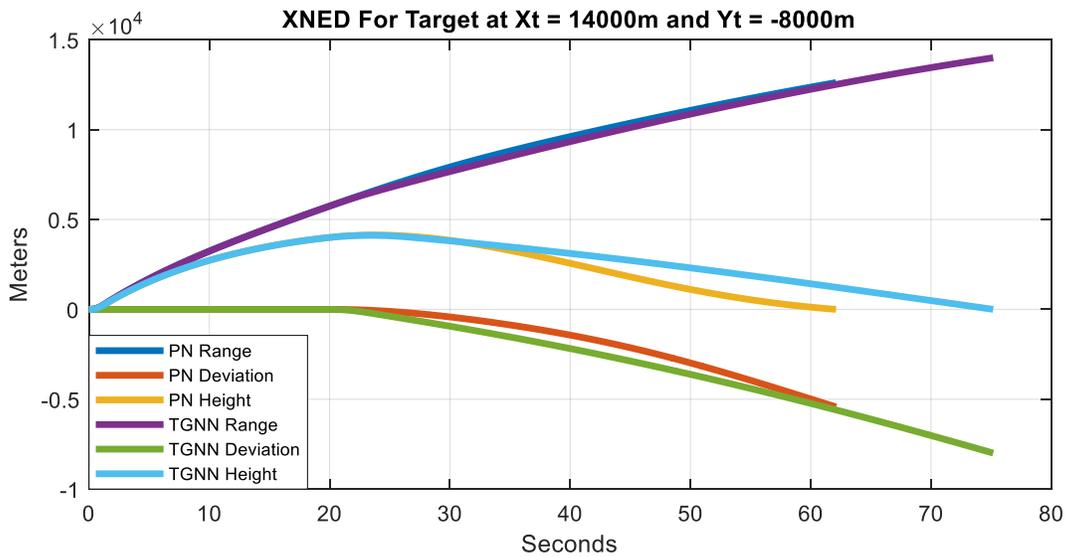


Figure 115 – Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 14000m and Yt = -8000m
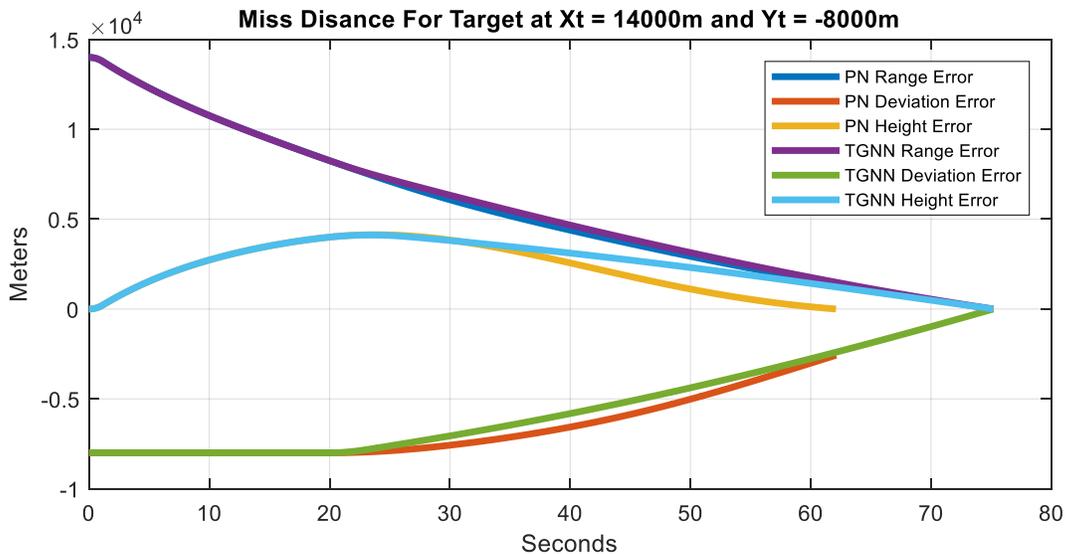


Figure 116 – Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 14000m and Yt = -8000m

Figure 117 and Figure 118 show the performance of the missile for the target at 14000m range and -10000m deviation. Here, it can be seen that TGNN performed better than PN. The TGNN range error is 4m while PN is also 1632m. In addition, TGNN performed better than PN for deviation. It can be seen that the deviation error for TGNN is 16m while PN is 4719m. It is important to note that TGNN took longer to reach the altitude of the target.
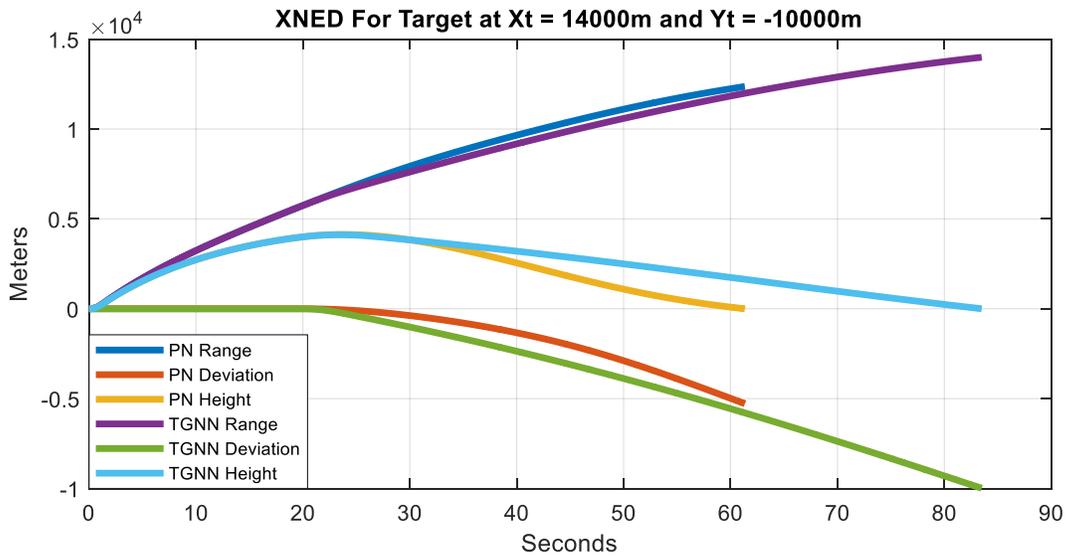


Figure 117 - Comparison of NED Frame Position Between TGNN and PN for Target Located at Xt = 14000m and Yt = -10000m
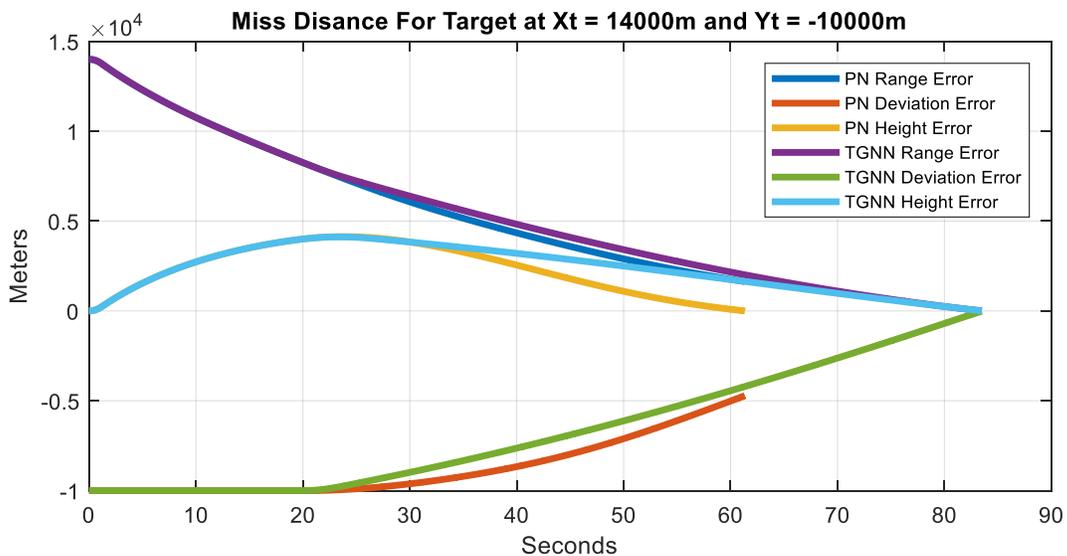


Figure 118 - Comparison of NED Frame Position Miss Distance Between TGNN and PN for Target Located at Xt = 14000m and Yt = -10000m

Figure 119 and Figure 120 show the comparison of the histogram distribution between PN and TGNN for range and deviation miss distances. It can be seen that PN has a higher mean and a higher standard of deviation. This means that for both cases PN has more error. This leads to the conclusion that TGNN is a viable if not better alternative to PN for Terminal Guidance.
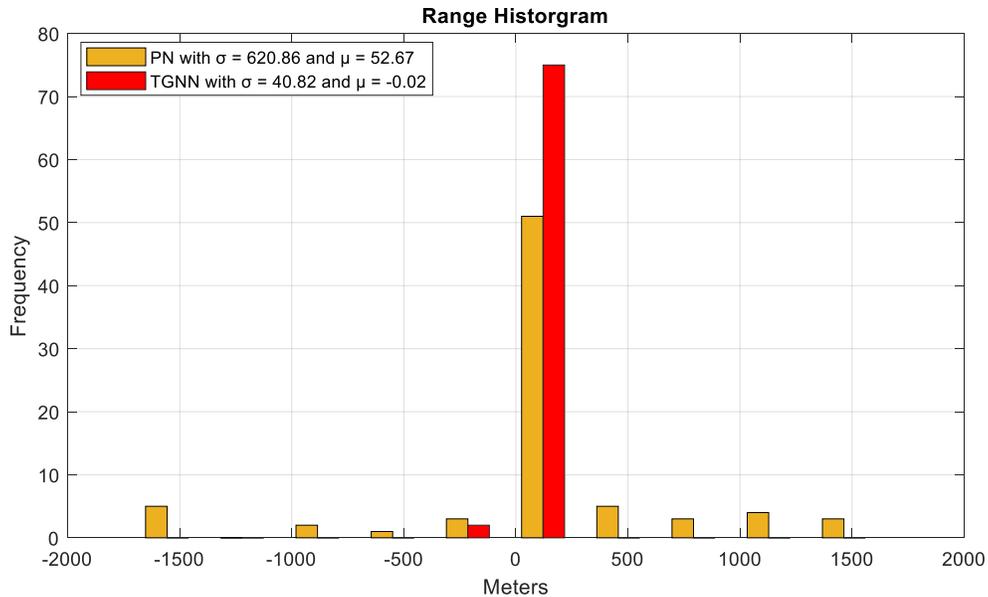


Figure 119 – Comparison of The Histogram Distribution Between PN and TGNN For Range Miss Distance
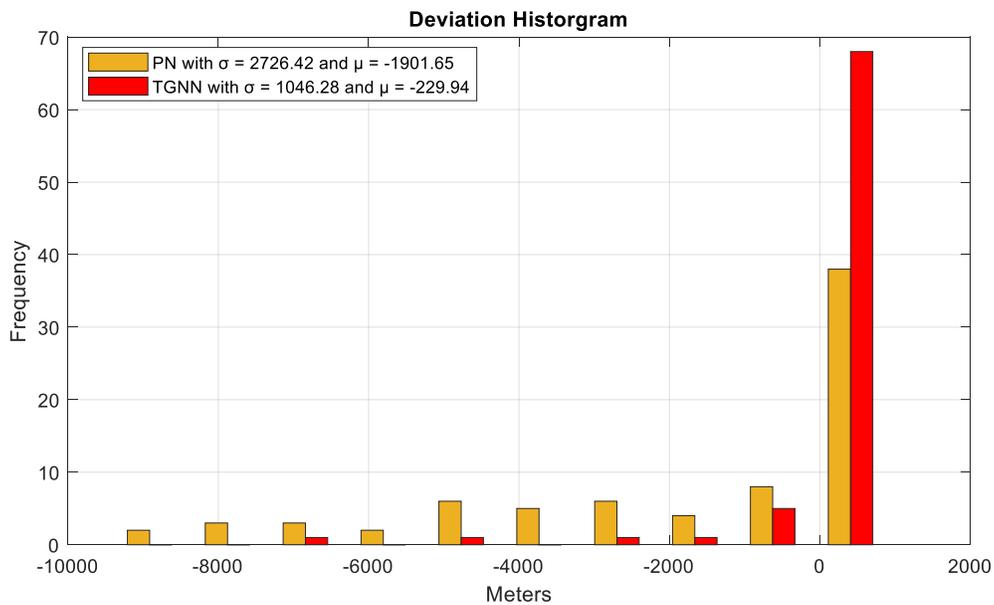


Figure 120 Comparison of The Histogram Distribution Between PN and TGNN For Deviation Miss Distance

## 7.2.2. Autopilot NN vs Autopilot

Similar to the guidance comparison, the missile is launched at 45degs launch angle. The angle is chosen for the comparison because it allows the missile to fly to the maximum range. In addition, it ensures sufficient amount of altitude to evaluate performance of the missile. The proportional navigation algorithm is used to guide the missile towards the target. In order to reduce the terminal guidance limitations, the target range is varied between 10000m and 14000m. Moreover, the target deviation is varied between 0m and 3000m. It is important to note that the missile guidance and autopilot starts at 20 seconds after launch as shown in Figure 121. This is to ensure that the missile reaches the apex before the guidance start. In addition, the scenario will be repeated with a maximum actuator deflection of 15degs and 20degs. This will allow the controllers to be tested with and without aerodynamic limitations caused by the maximum actuator deflection of 15degs. Additionally, the IMU refresh rate is set to 0.2mili seconds in order for the NNA to work properly.

Both PID and Neural Network Autopilot will be used to control the missile in these scenarios. Unlike the guidance comparison where only the miss distance of the missile is used as the evaluation criteria, the Mean Square Error will be used as well. Here, the MSE of both algorithms for roll, pitch, and yaw responses will be compared. Hence, this creates a quantifiable criterion to see which controller followed the demand better. Furthermore, the Euler angles of the missile will be presented in order to see the smoothness of each of the controllers. This is important because an oscillating missile is not desirable.



Autopilot Start
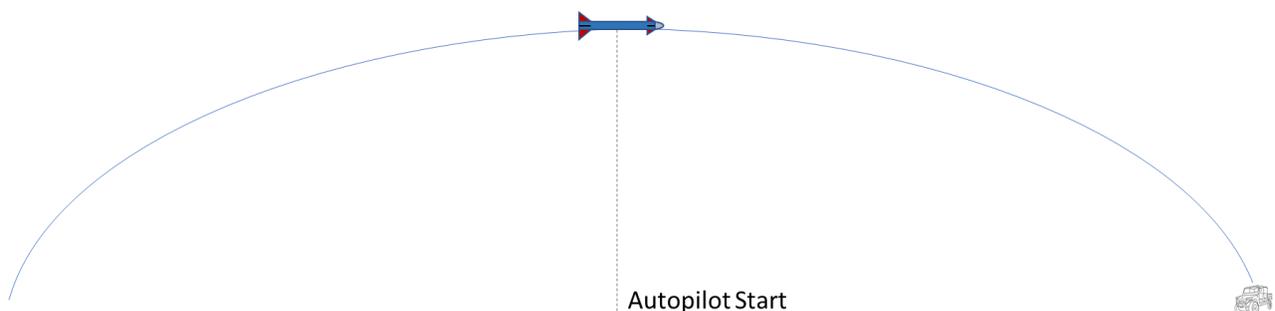
Figure 121 – Autopilot Comparison Scenario

Figure 122 and Figure 123 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 10000m range and 0m deviation. Here, it can be seen that both PID and NNA algorithms performed almost perfectly. However, PID has a small range miss distance.
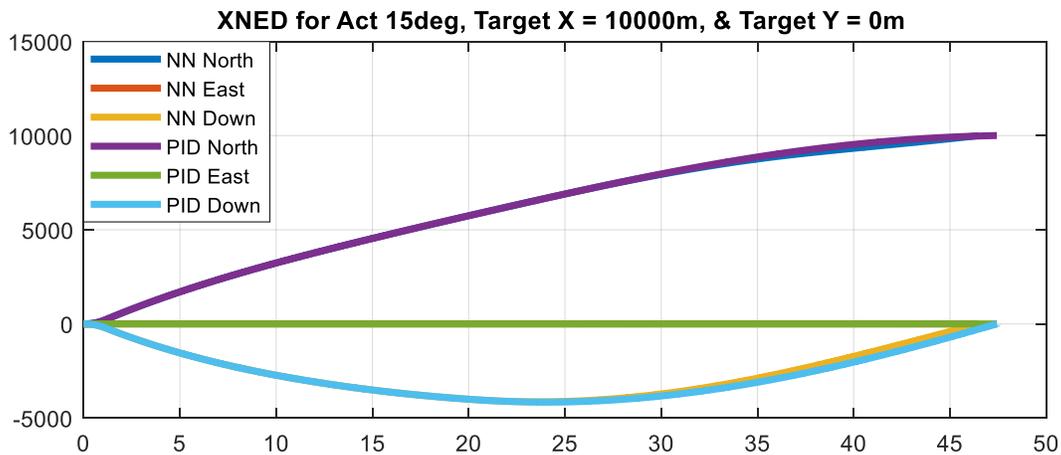


Figure 122 – Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = 0m
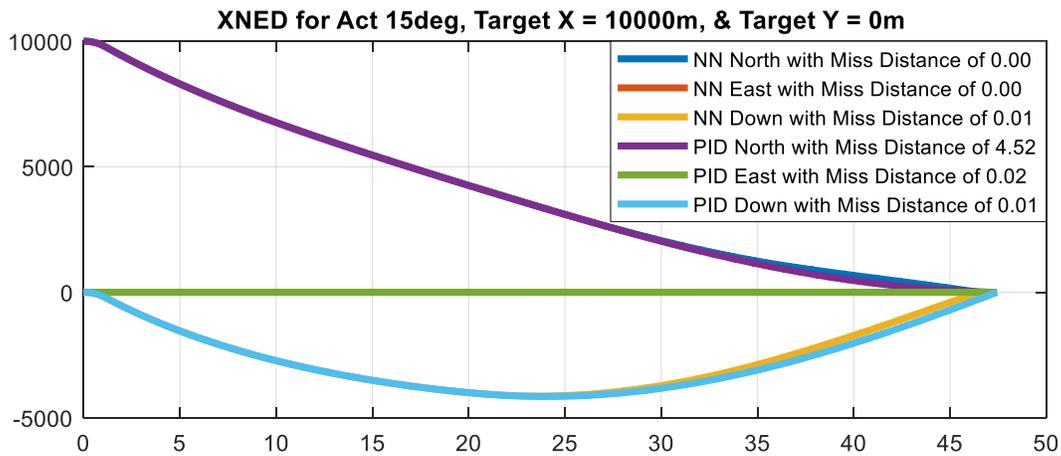


Figure 123 – Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = 0m

Figure 124 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and 0m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
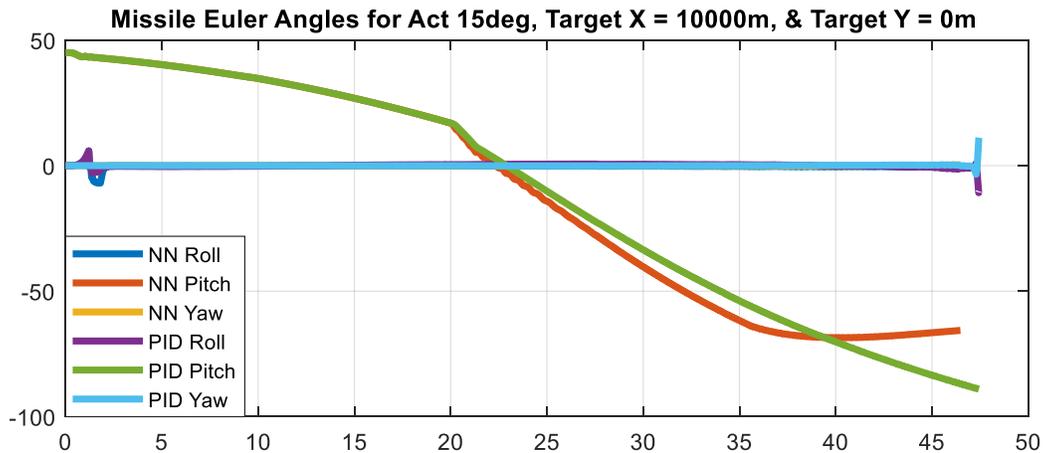


Figure 124 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = 0m

Figure 125 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and 0m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
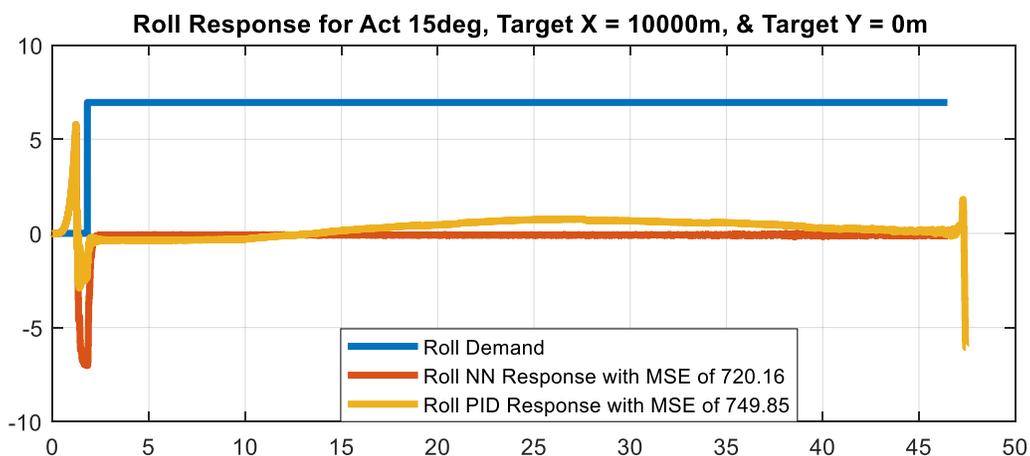


Figure 125 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = 0m

Figure 126 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and 0m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller shows an unusual behavior which contributed to the small miss distance.



**Pitch Response for Act 15deg, Target X = 10000m, & Target Y = 0m**
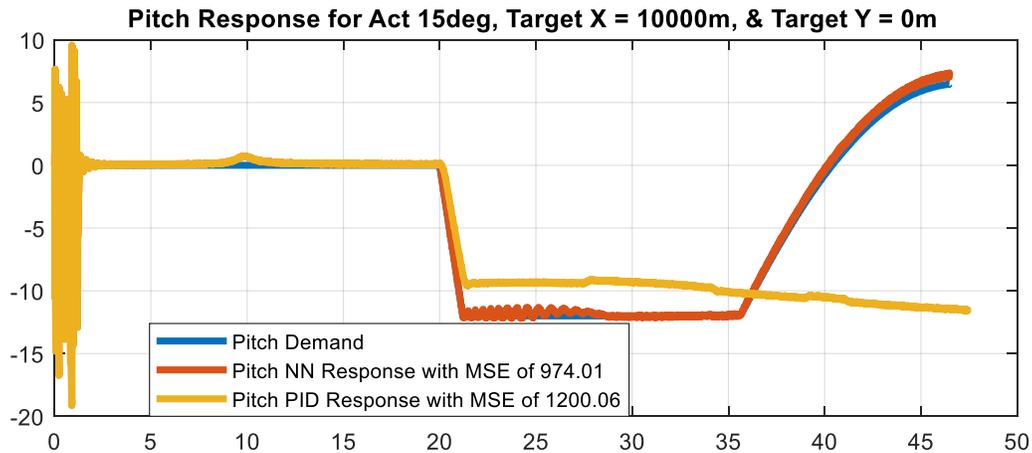
Figure 126 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = 0m

Figure 127 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and 0m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 15deg, Target X = 10000m, & Target Y = 0m**
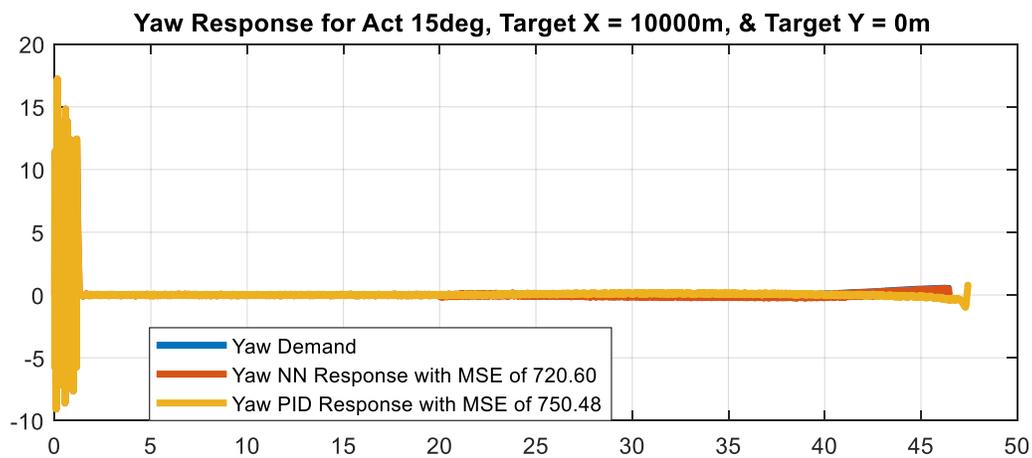
Figure 127 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = 0m

Figure 128 and Figure 129 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 10000m range and -1000m deviation. Here, it can be seen that both PID and NNA algorithms performed almost perfectly.
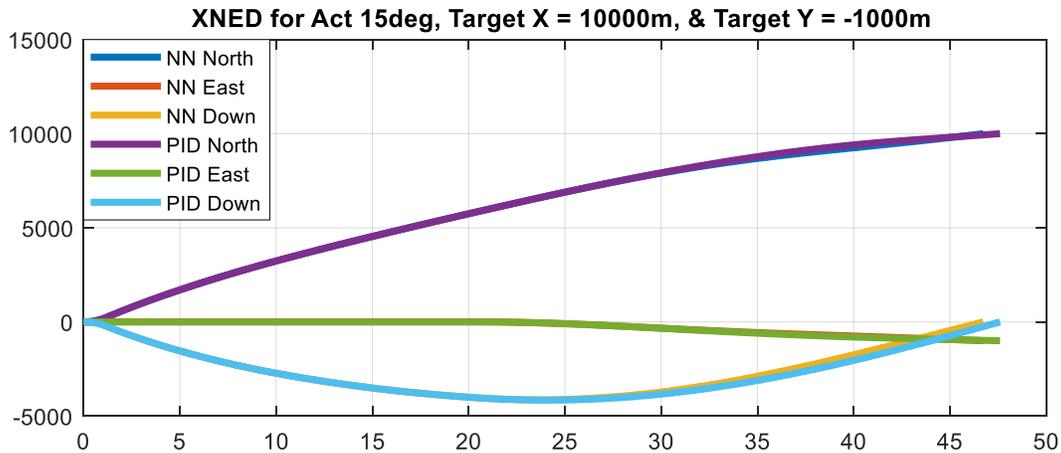


Figure 128 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -1000m
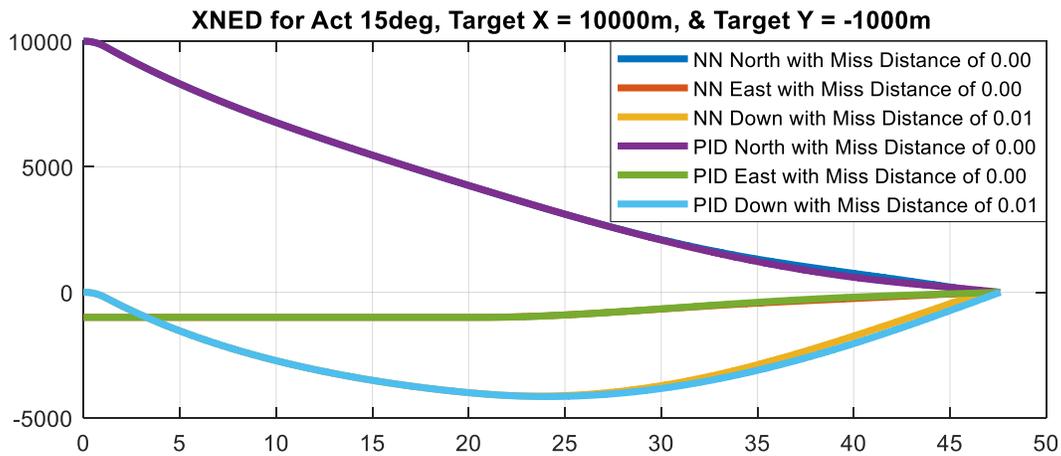


Figure 129 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 130 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and -1000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
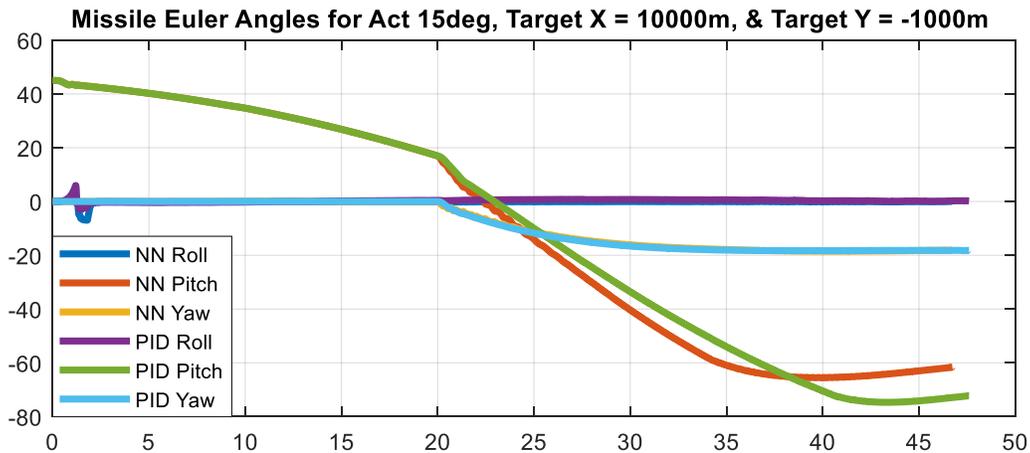


Figure 130 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 131 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and -1000m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
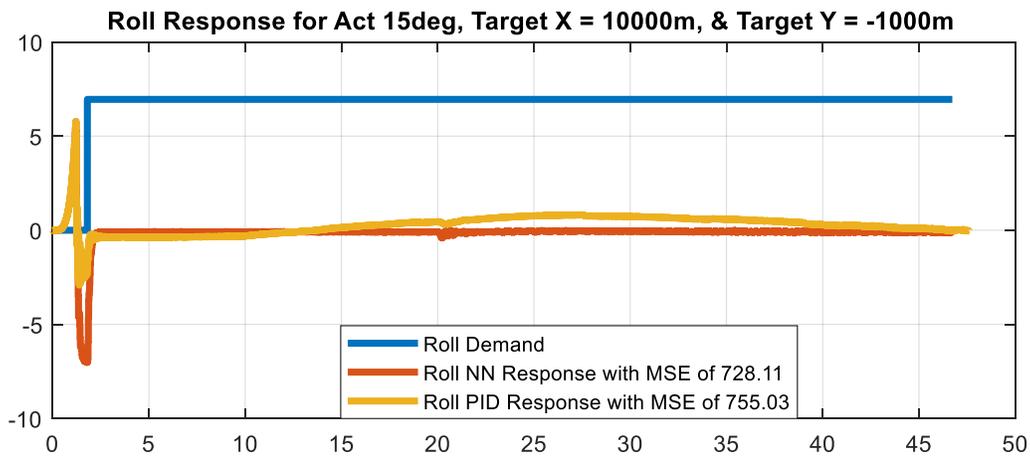


Figure 131 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 132 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and -1000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.
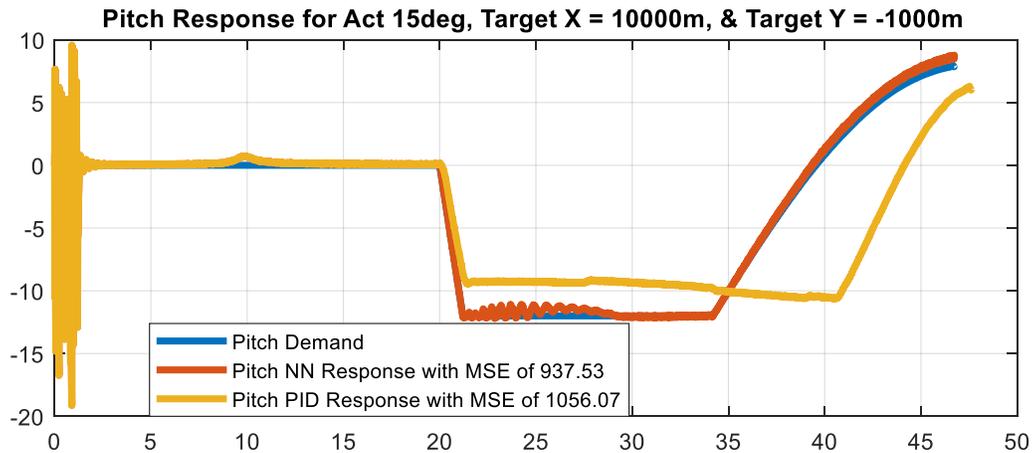


Figure 132 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 133 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and -1000m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.
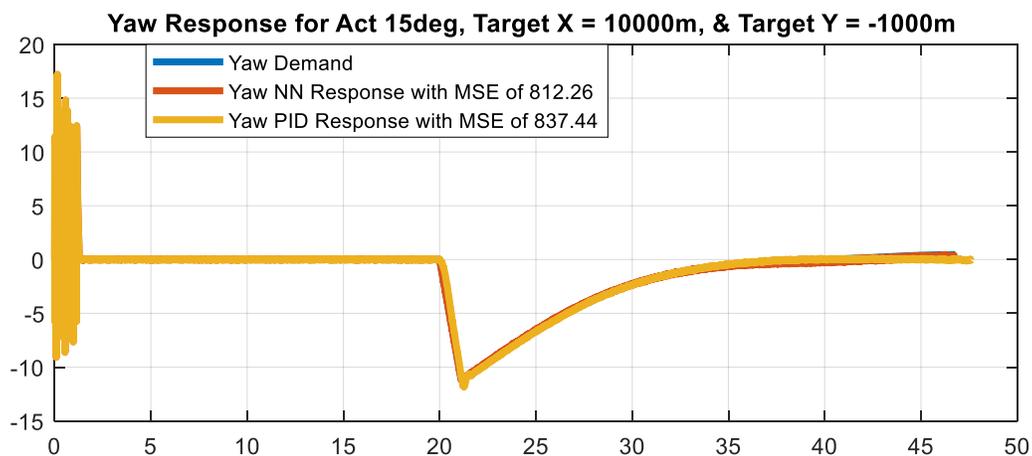


Figure 133 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 134 and Figure 135 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the NNA controller performed better for range with less miss distance. However, the PID controller performed better for deviation with less miss distance.
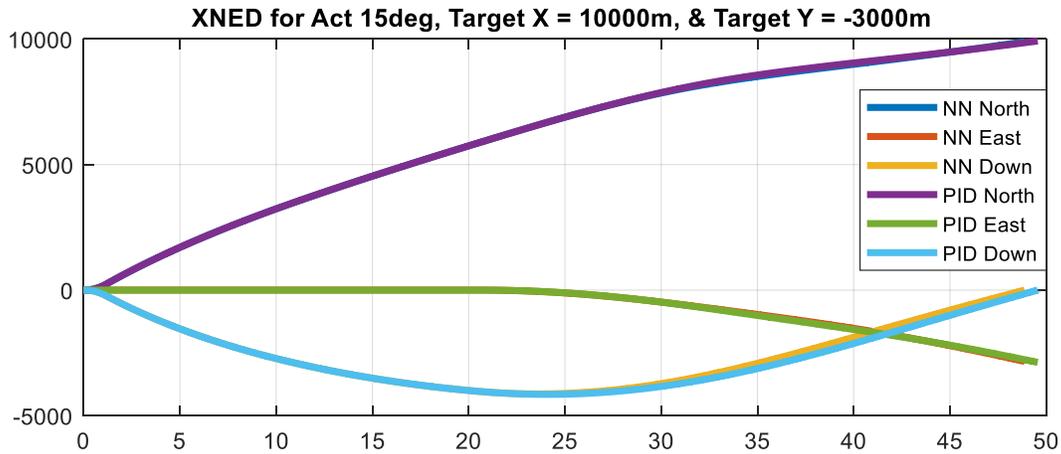


Figure 134 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -3000m
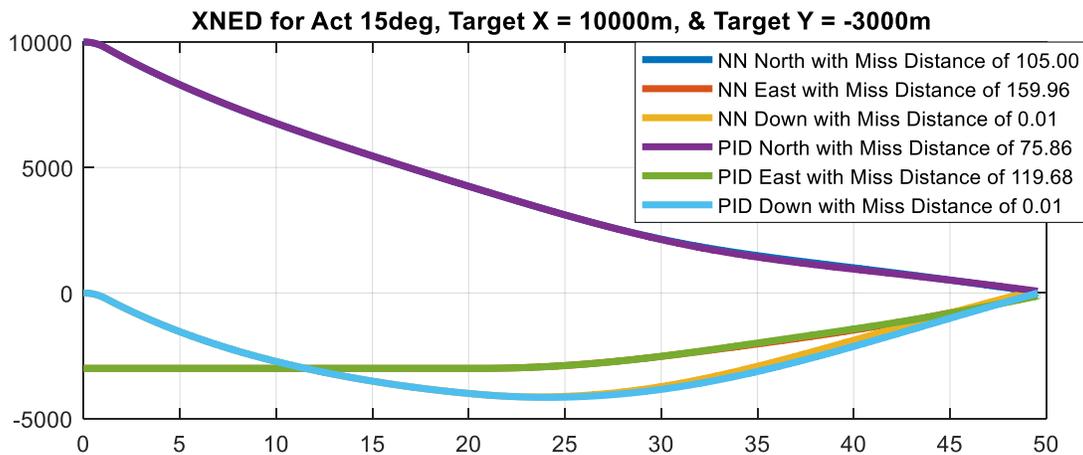


Figure 135 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 136 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
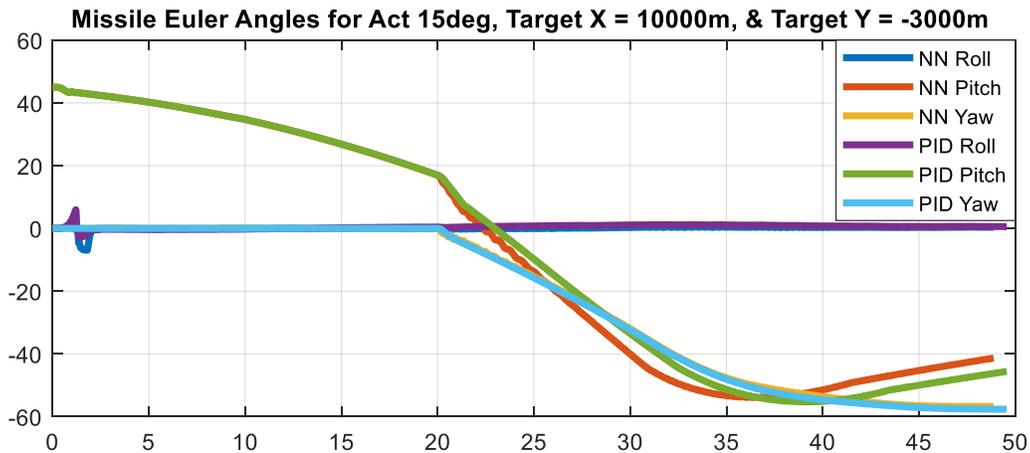


Figure 136 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 137 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
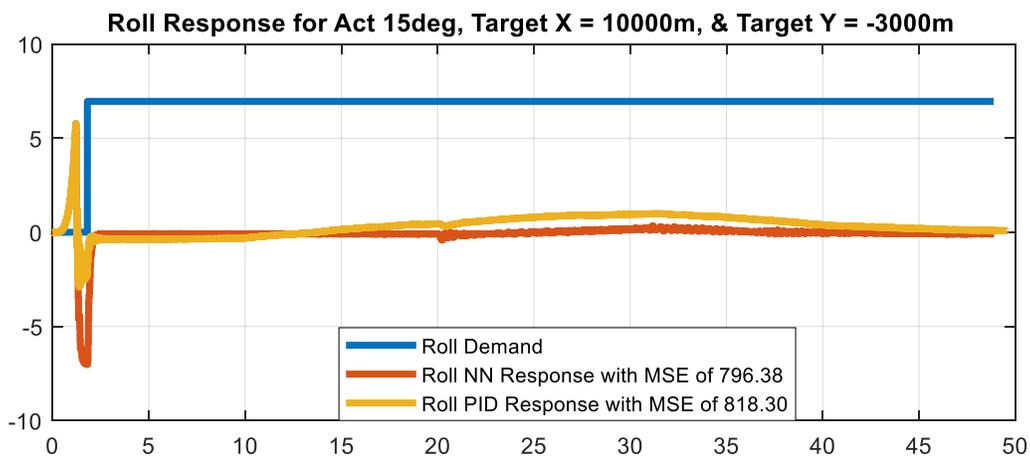


Figure 137 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 138 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the Pitch LNNA controllers performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.



**Pitch Response for Act 15deg, Target X = 10000m, & Target Y = -3000m**

Figure 138 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 139 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. It is interesting because despite Yaw LNNA controller having a better performance, the PID resulted in less range miss distance. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 15deg, Target X = 10000m, & Target Y = -3000m**
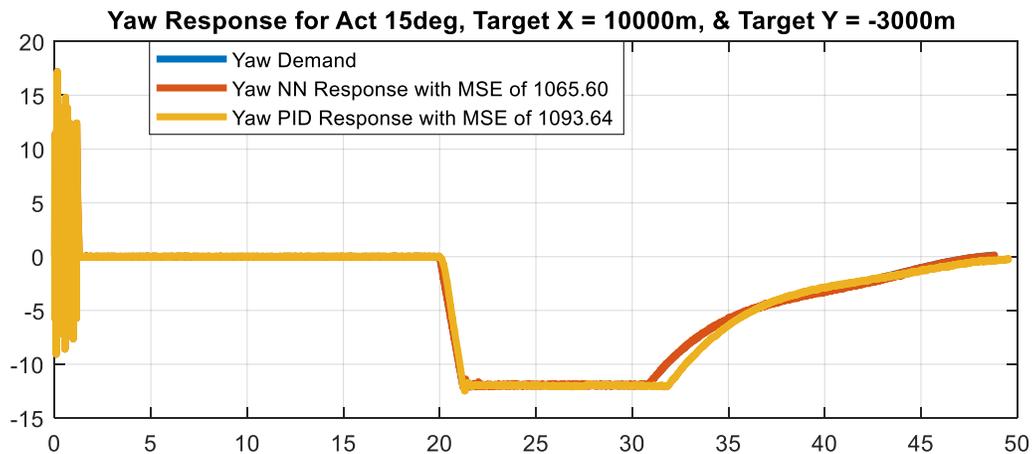
Figure 139 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 140 and Figure 141 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 12000m range and 0m deviation. Here, it can be seen that the NNA controller performed better for range with less miss distance. However, the PID controller performed better for deviation with less miss distance.
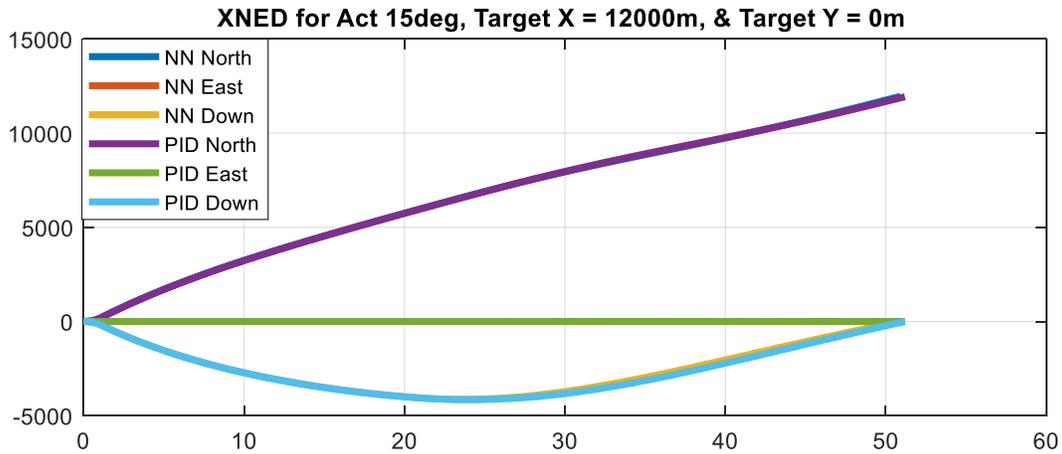


Figure 140 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = 0m
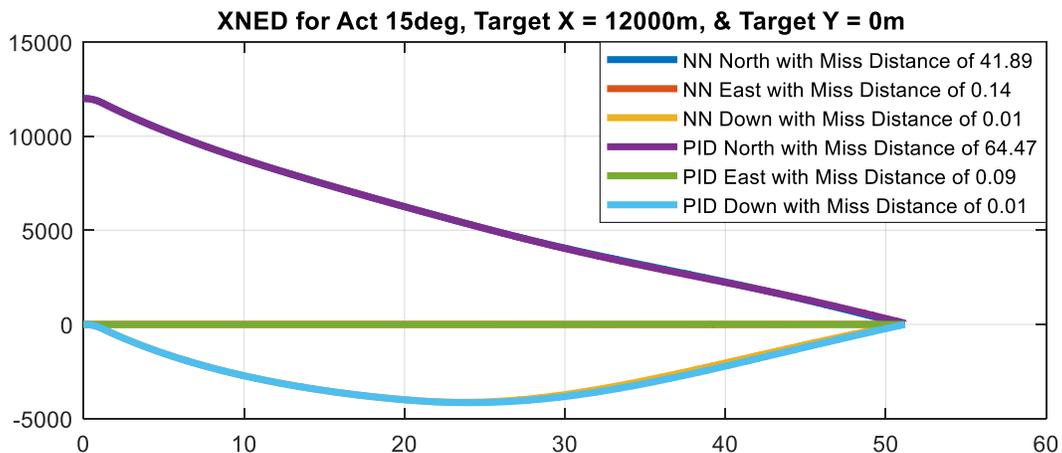


Figure 141 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = 0m

Figure 142 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and 0m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
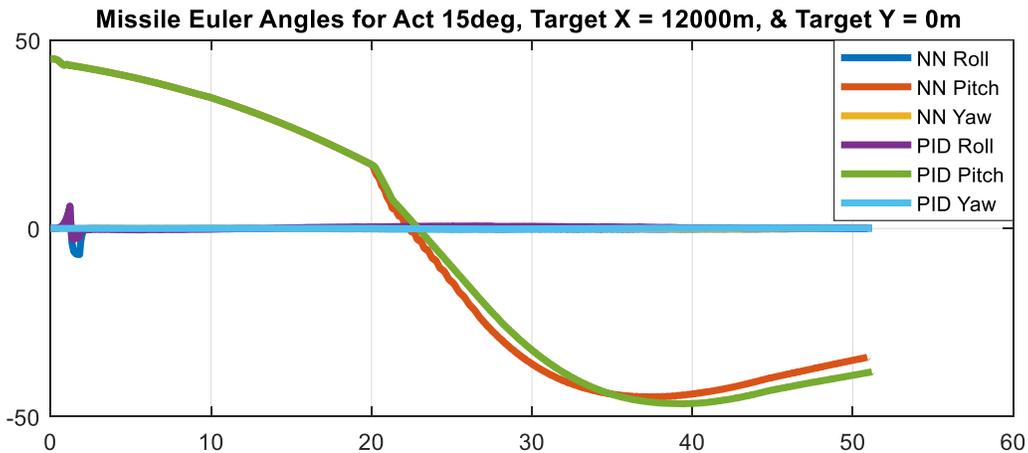


Figure 142 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = 0m

Figure 143 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and 0m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
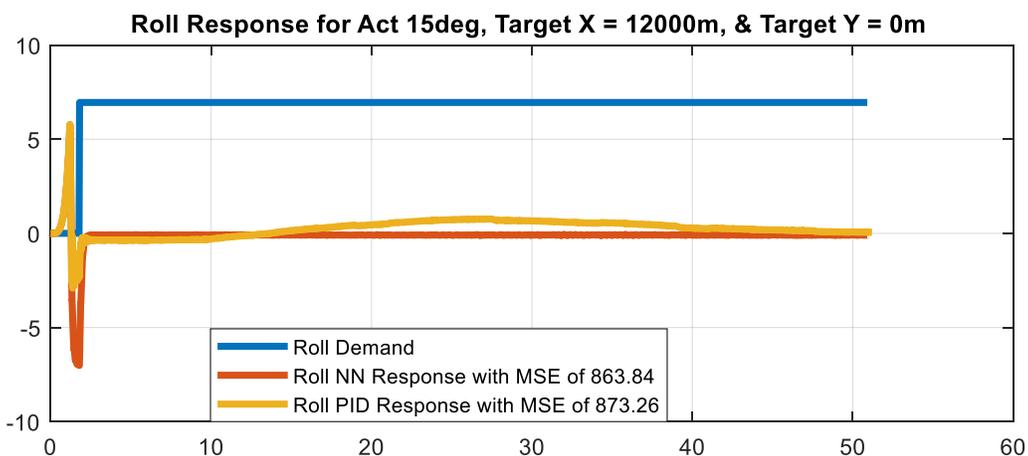


Figure 143 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = 0m

Figure 144 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and 0m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.



**Pitch Response for Act 15deg, Target X = 12000m, & Target Y = 0m**
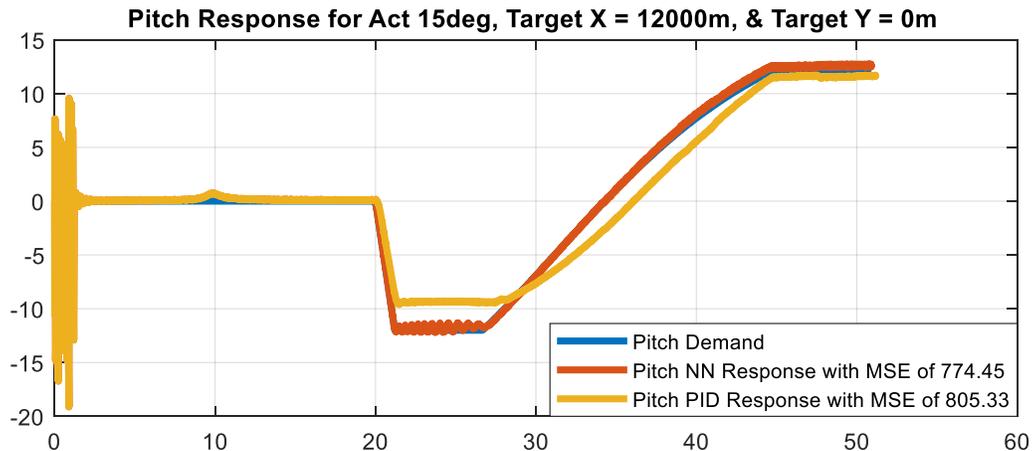
Figure 144 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = 0m

Figure 145 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and 0m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. It is interesting because despite Yaw LNNA controller having a better performance, the PID resulted in less range miss distance. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 15deg, Target X = 12000m, & Target Y = 0m**
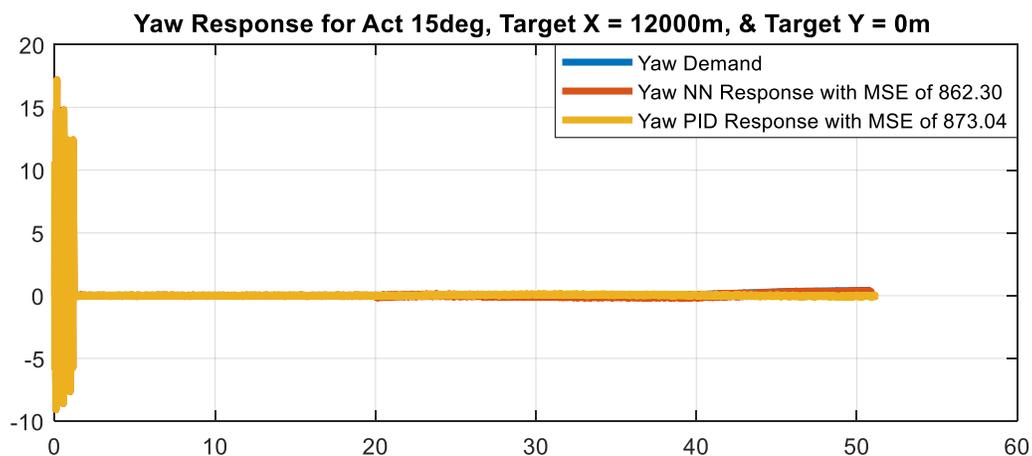
Figure 145 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = 0m

Figure 146 and Figure 147 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the NNA controller performed better for both range and deviation.
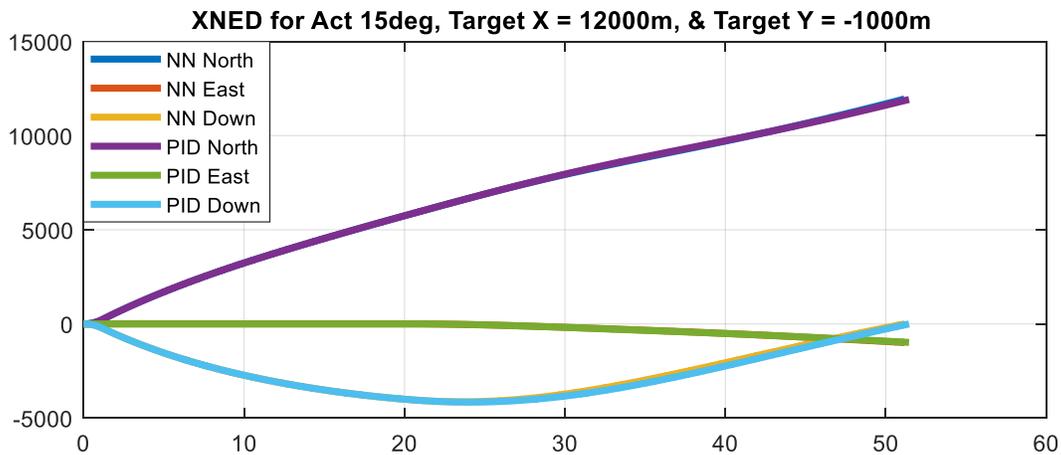


Figure 146 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -1000m
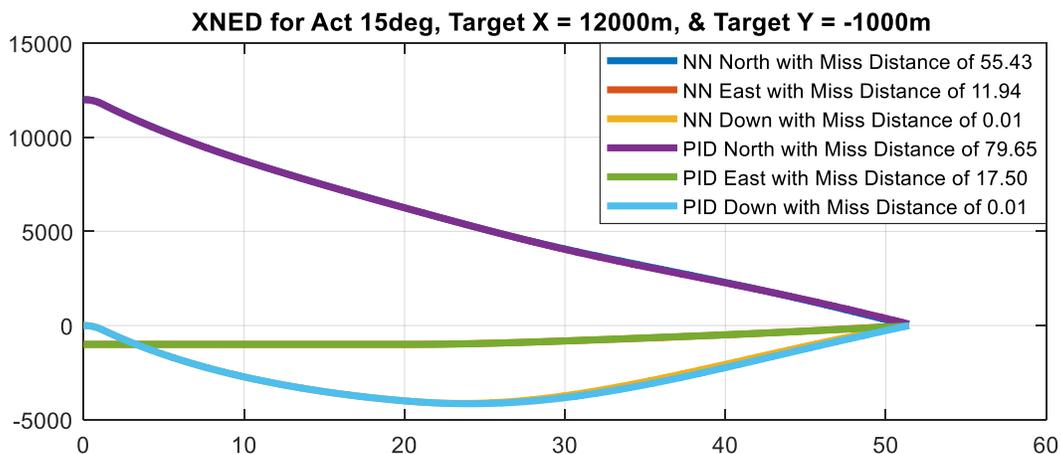


Figure 147 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 148 shows the missile Euler angles comparisons between NNA and PID controller for a maximum actuator deflection of 15deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
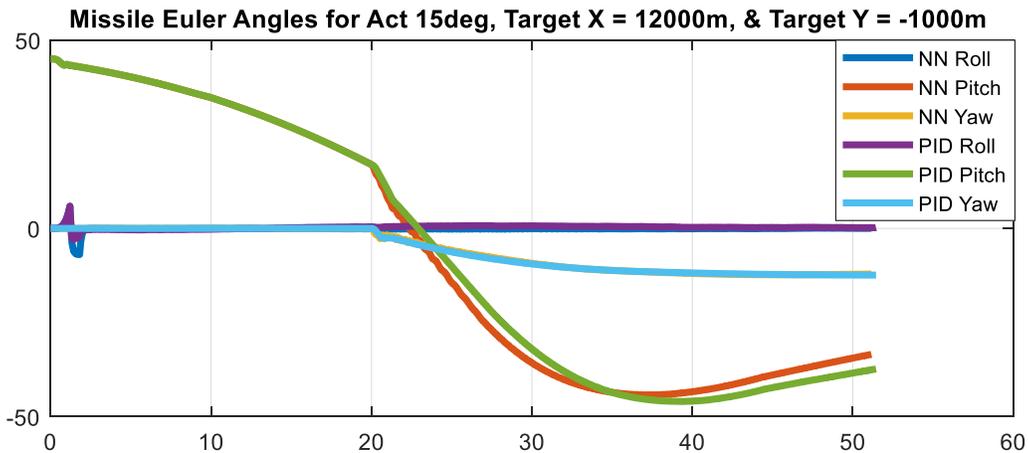


Figure 148 - Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 149 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
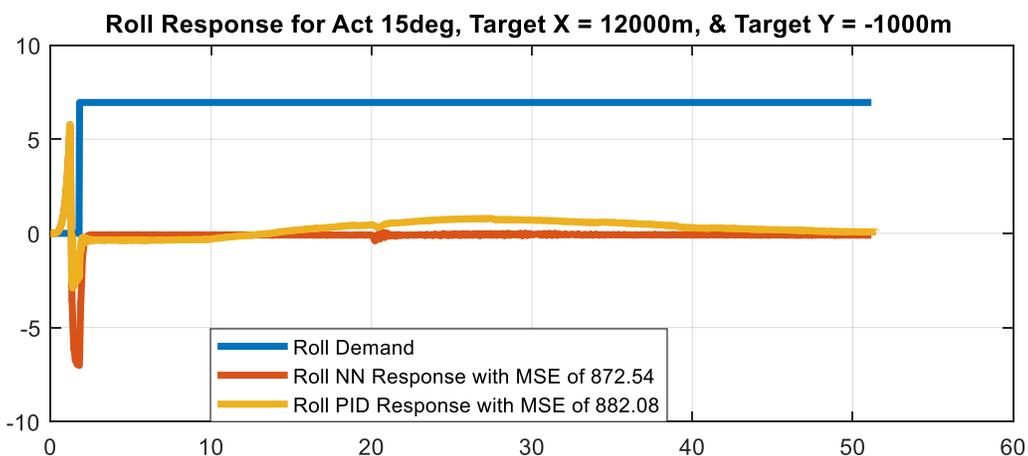


Figure 149 - Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 150 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.



**Pitch Response for Act 15deg, Target X = 12000m, & Target Y = -1000m**
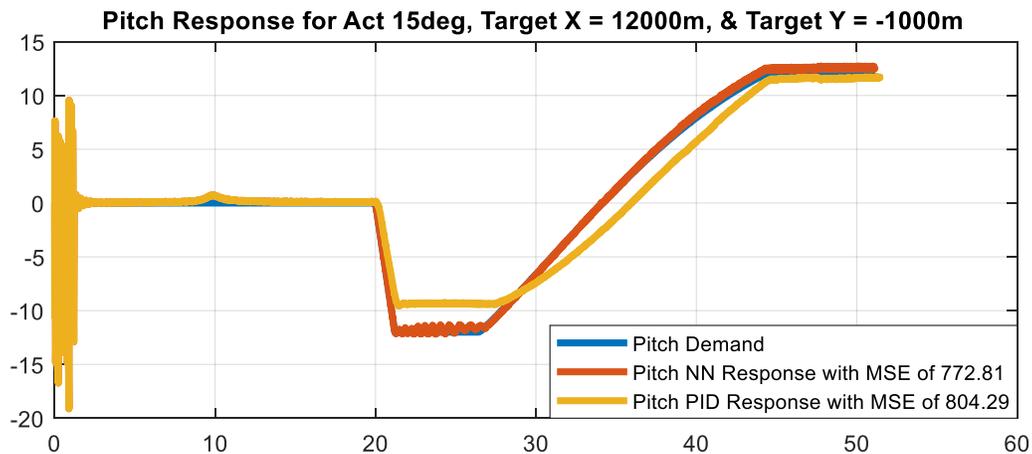
Figure 150 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 151 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 15deg, Target X = 12000m, & Target Y = -1000m**
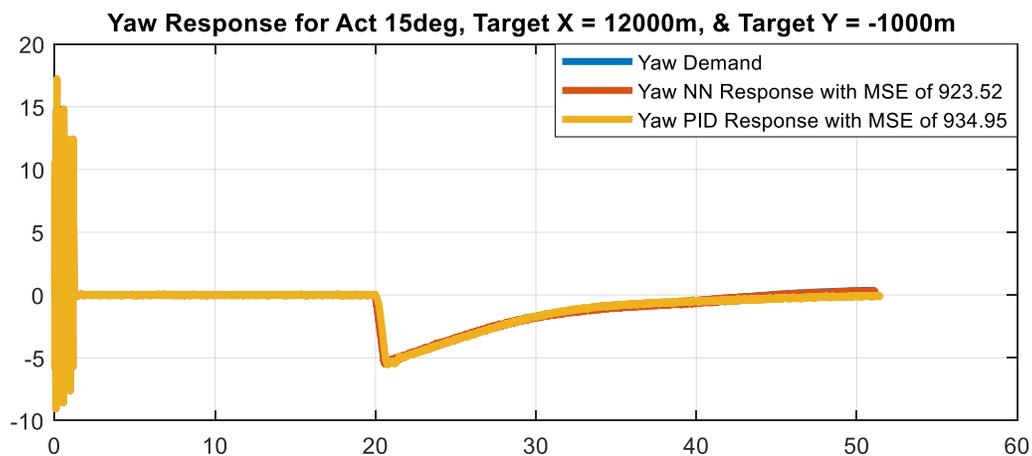
Figure 151 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 152 and Figure 153 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the NNA controller performed better for both range and deviation.
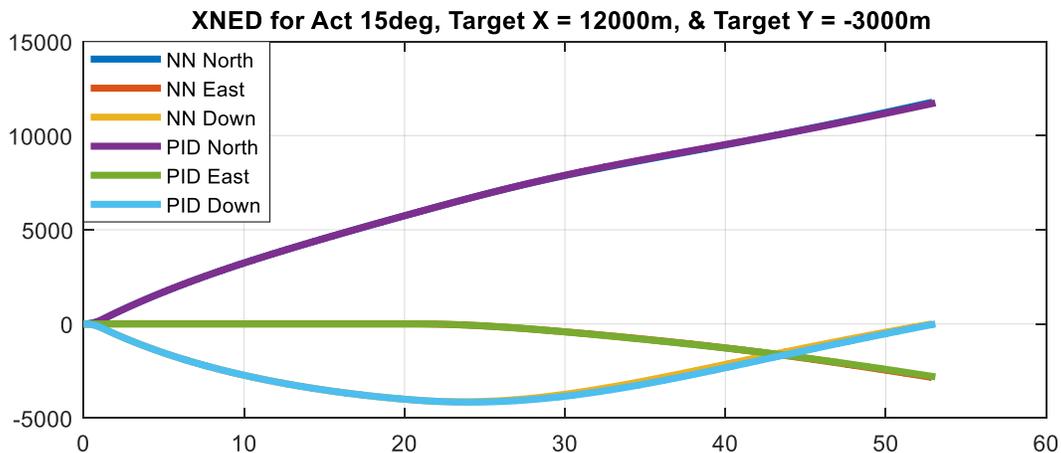


Figure 152 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -3000m
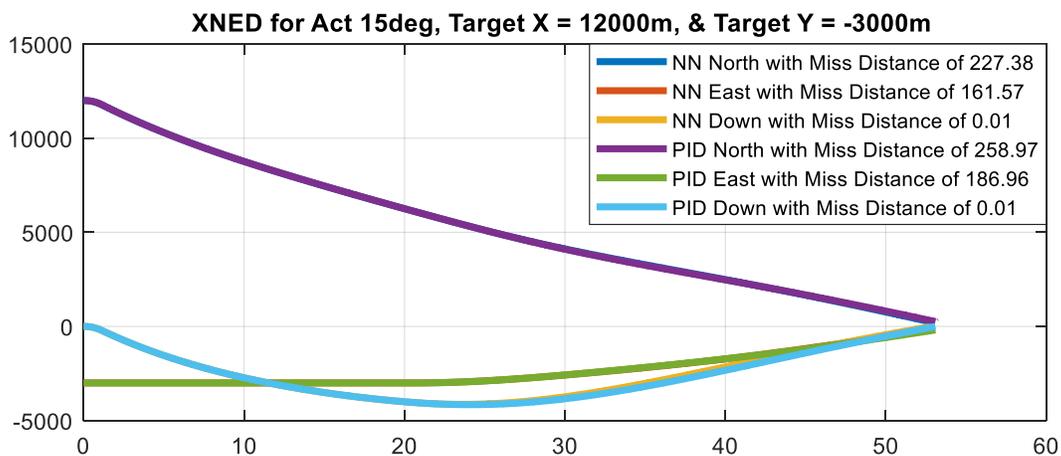


Figure 153 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 154 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.

**Missile Euler Angles for Act 15deg, Target X = 12000m, & Target Y = -3000m**
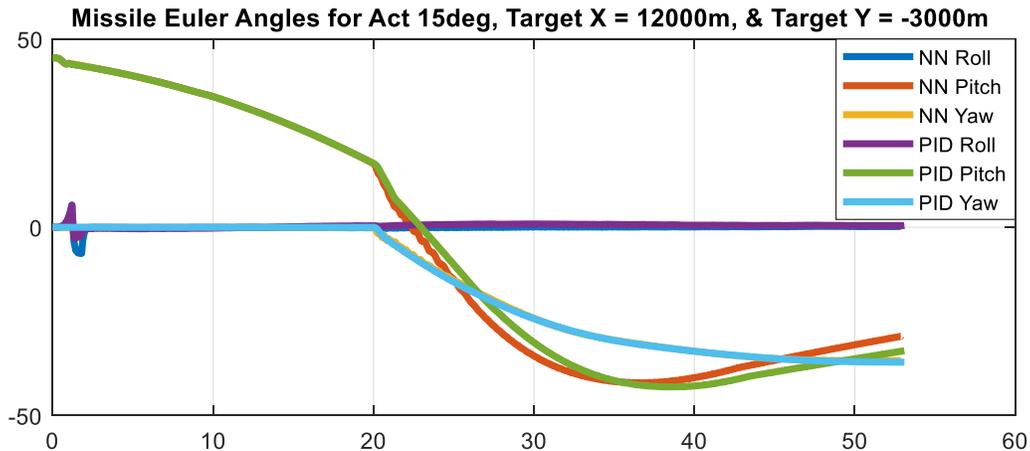


Figure 154 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 155 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.

**Roll Response for Act 15deg, Target X = 12000m, & Target Y = -3000m**
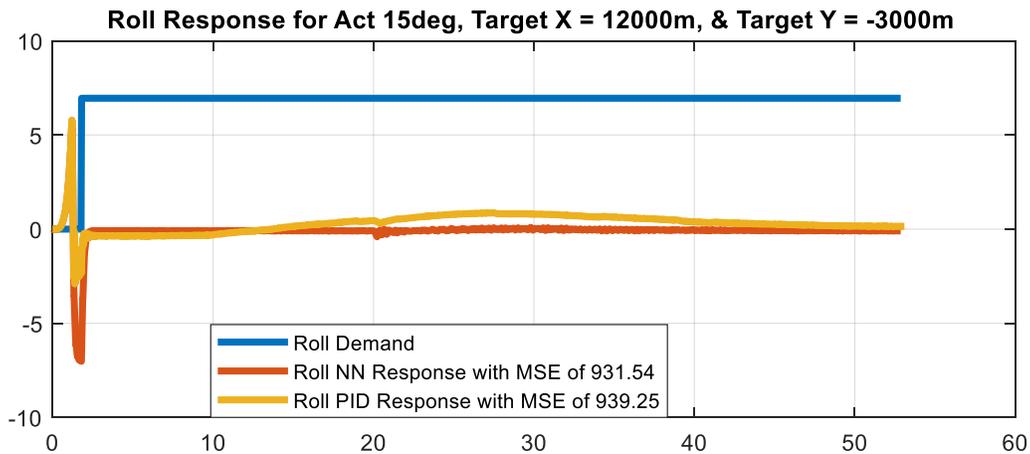


Figure 155 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 156 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.
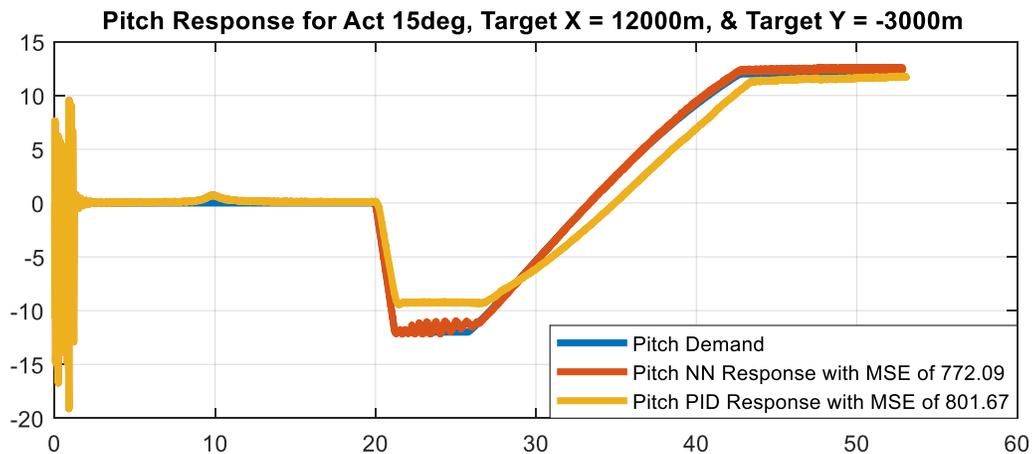


Figure 156 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 157 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



Figure 157 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 158 and Figure 159 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 14000m range and 0m deviation. Here, it can be seen that the NNA controller performed better for range. However, the PID controller performed better for deviation.
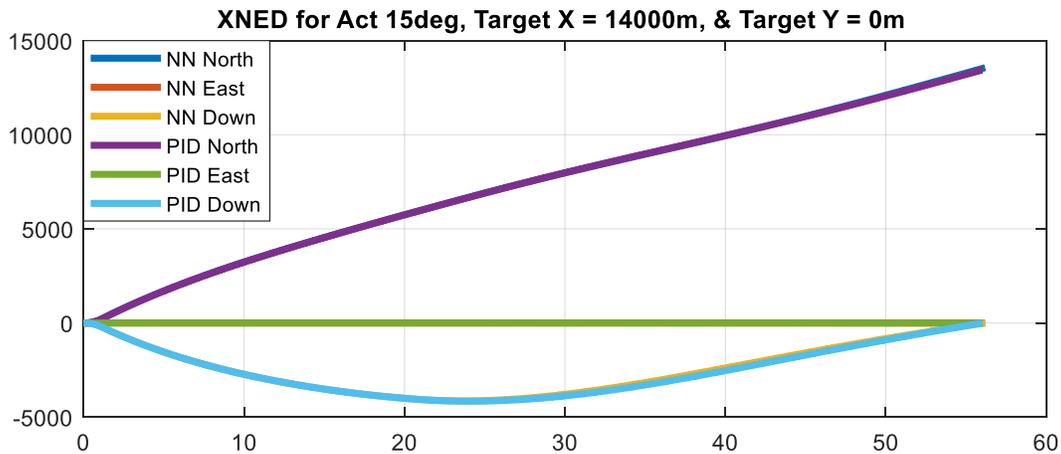


Figure 158 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = 0m
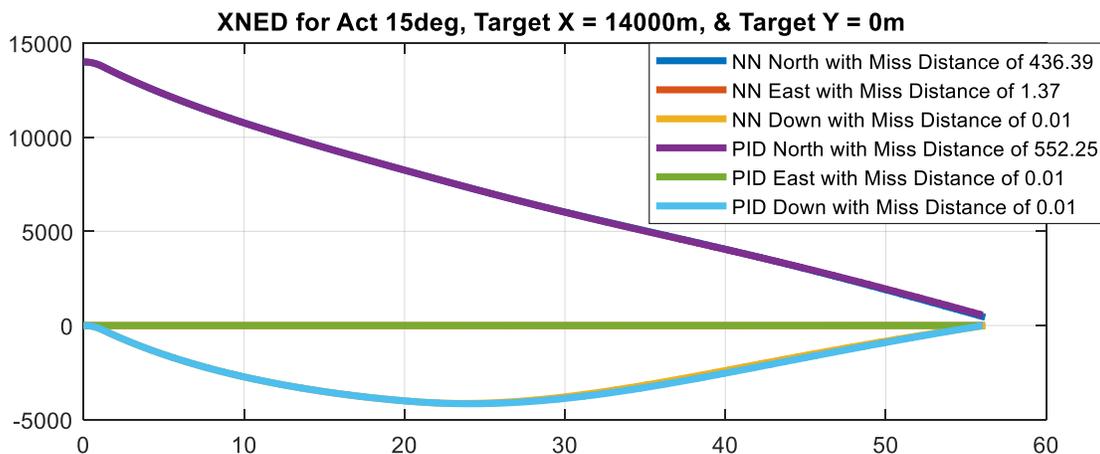


Figure 159 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = 0m

Figure 160 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and 0m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.



**Missile Euler Angles for Act 15deg, Target X = 14000m, & Target Y = 0m**
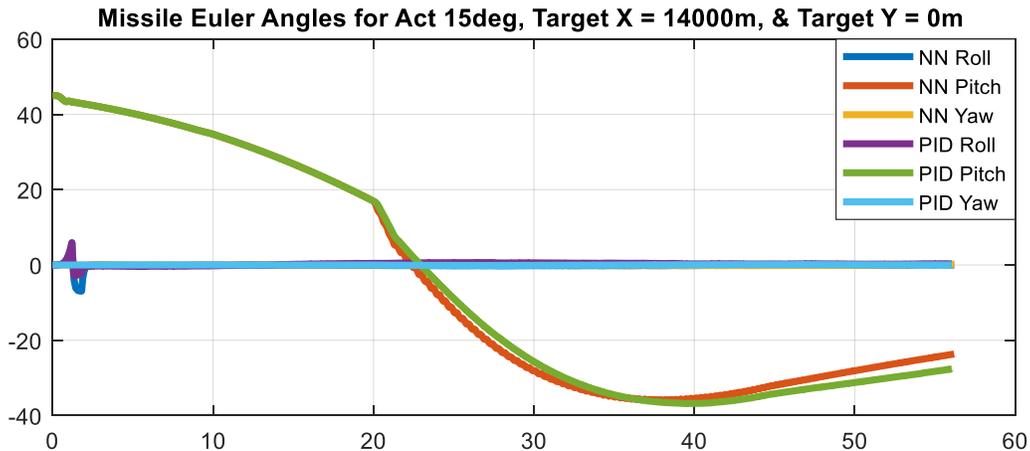
Figure 160 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = 0m

Figure 161 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and 0m deviation. Here, it can be seen that the PID controller performed better than the RNNA controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.



**Roll Response for Act 15deg, Target X = 14000m, & Target Y = 0m**
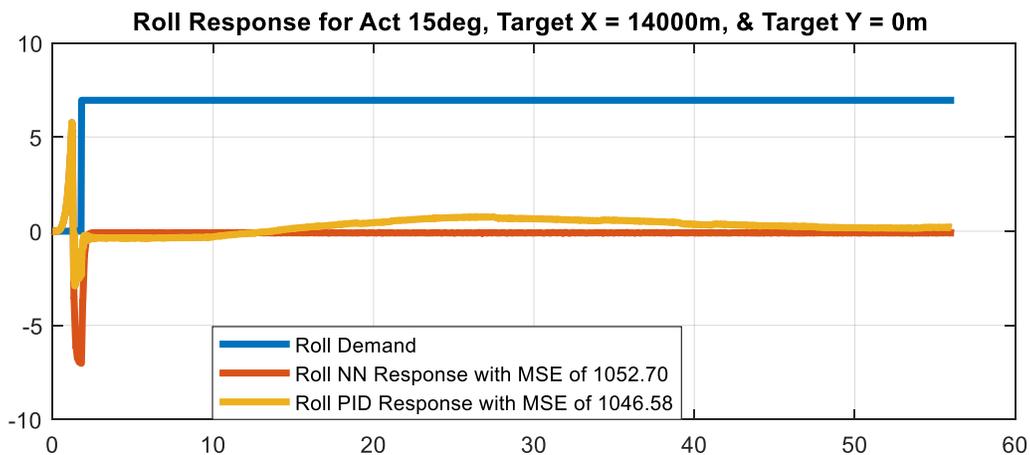
Figure 161 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = 0m

Figure 162 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and 0m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.
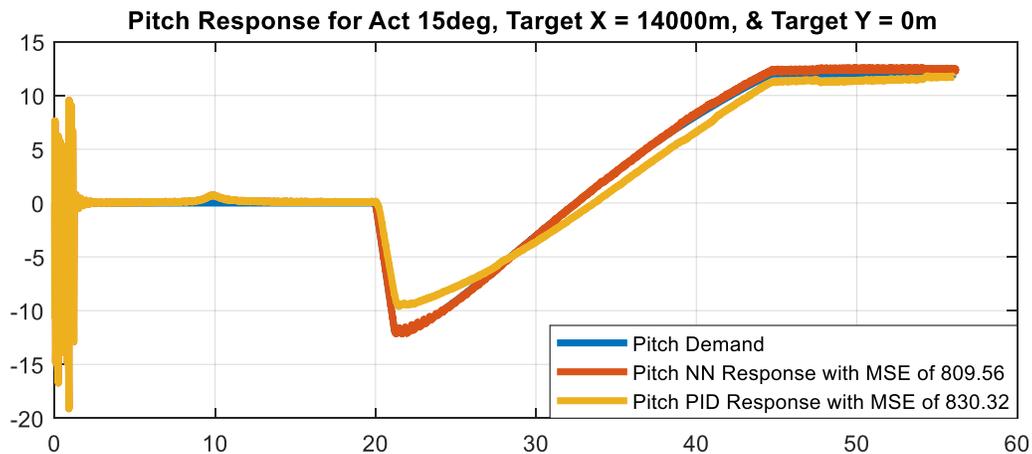


Figure 162 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = 0m

Figure 163 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and 0m deviation. Here, it can be seen that the PID controller performed better than the Yaw LNNA controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.
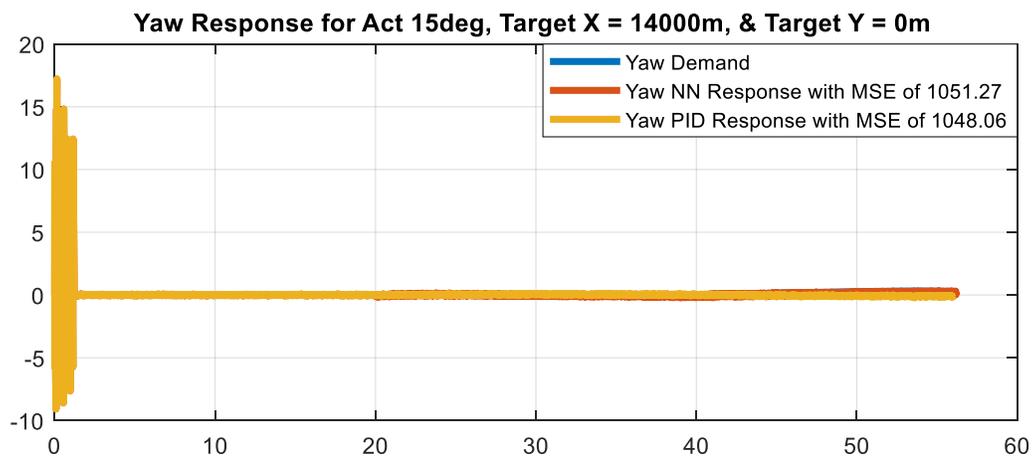


Figure 163 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = 0m

Figure 164 and Figure 165 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the NNA controller performed better for range. However, the PID controller performed better for deviation.
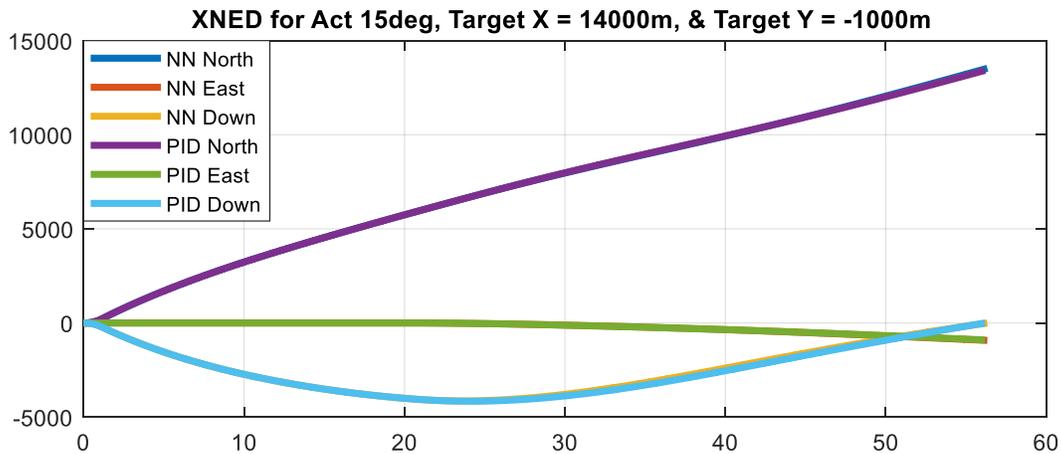


Figure 164 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -1000m
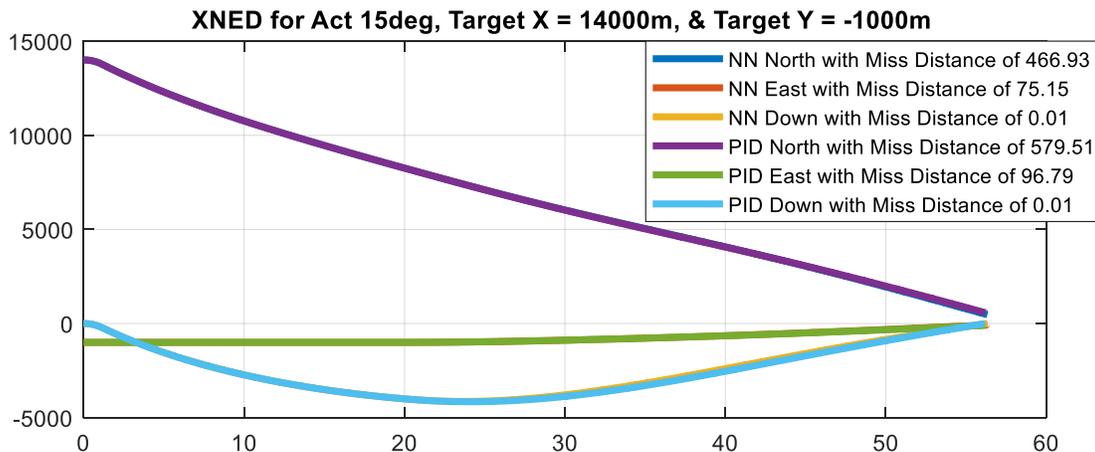


Figure 165 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 166 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
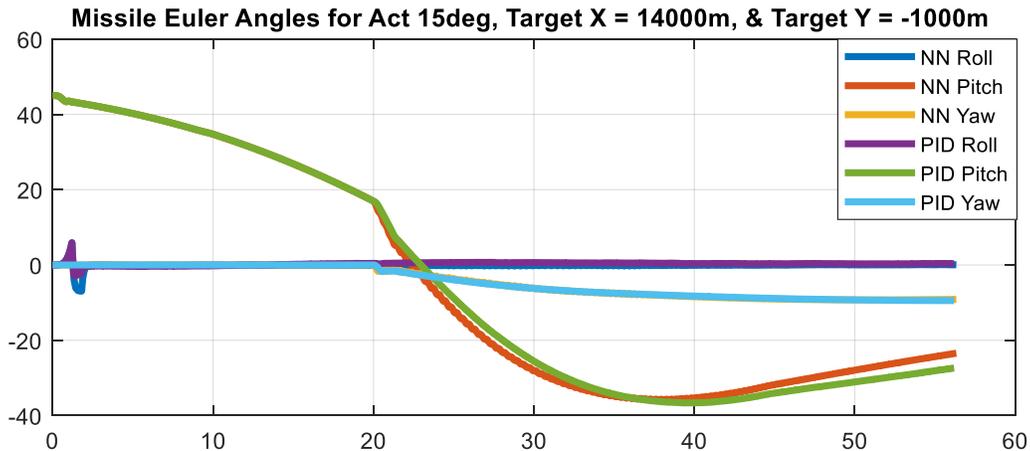


Figure 166 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 167 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the PID controller performed better than the RNNA controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
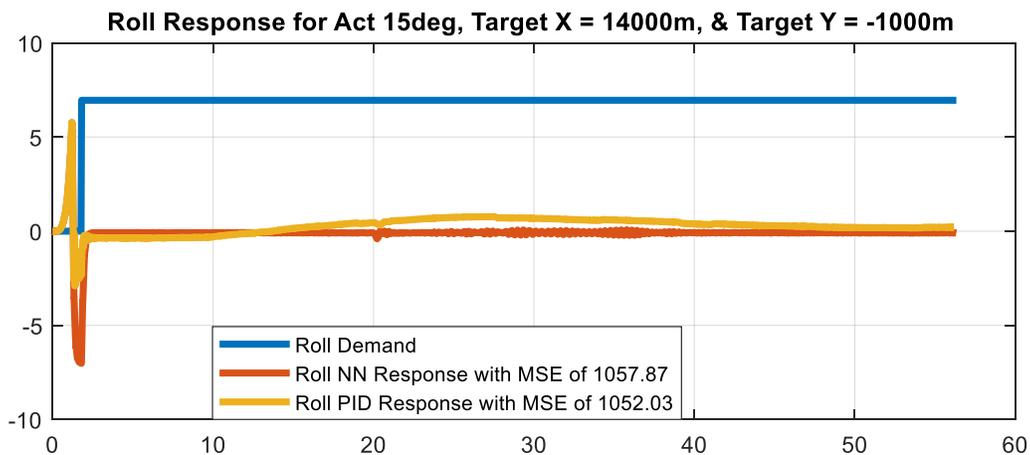


Figure 167 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 168 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.
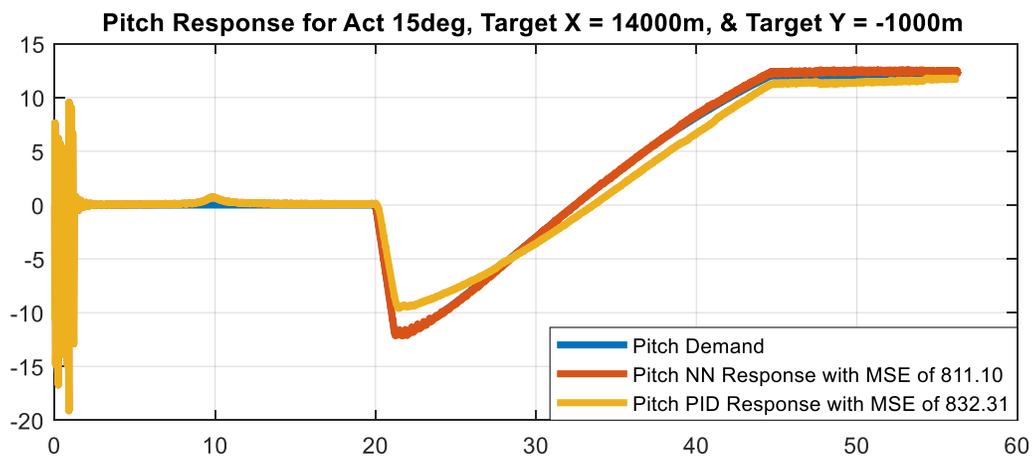


Figure 168 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 169 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the PID controller performed better than the Yaw LNNA controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



Figure 169 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 170 and Figure 171 show the autopilot performance of the missile for a maximum actuator deflection of 15deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the NNA controller performed better for both range and deviation.
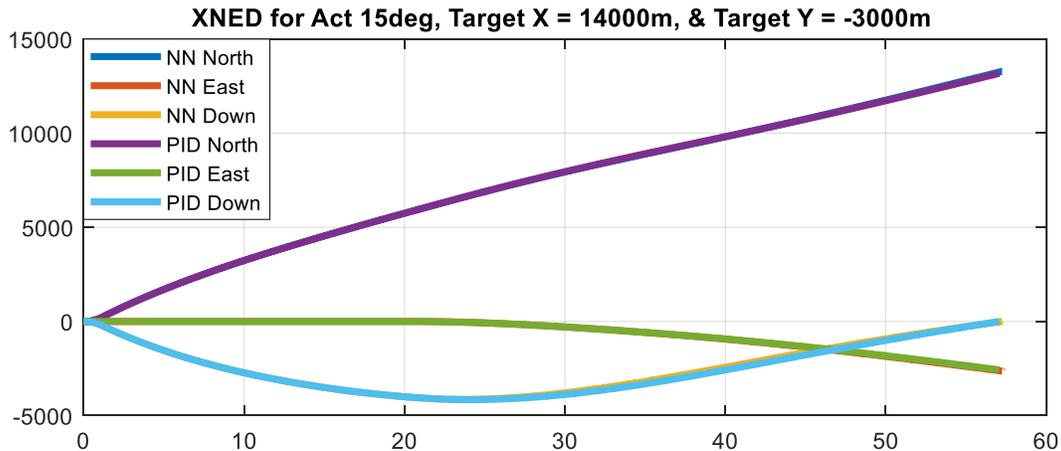


Figure 170 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -3000m
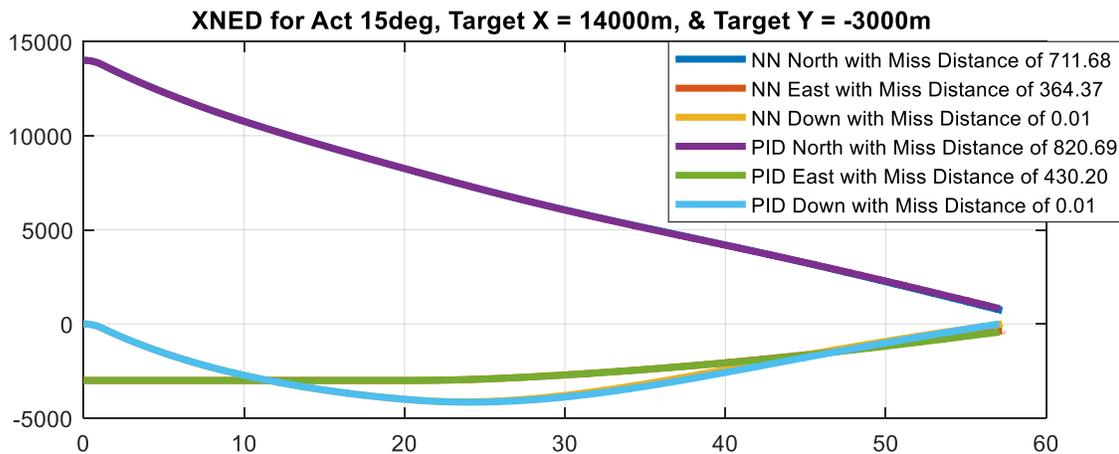


Figure 171 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 172 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.



**Missile Euler Angles for Act 15deg, Target X = 14000m, & Target Y = -3000m**
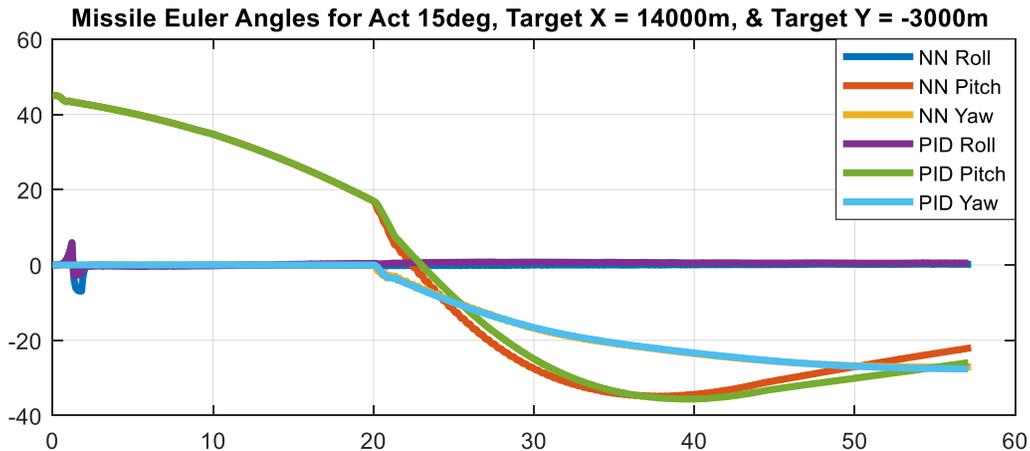
Figure 172 - Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 173 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the PID controller performed better than the RNNA controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.



**Roll Response for Act 15deg, Target X = 14000m, & Target Y = -3000m**
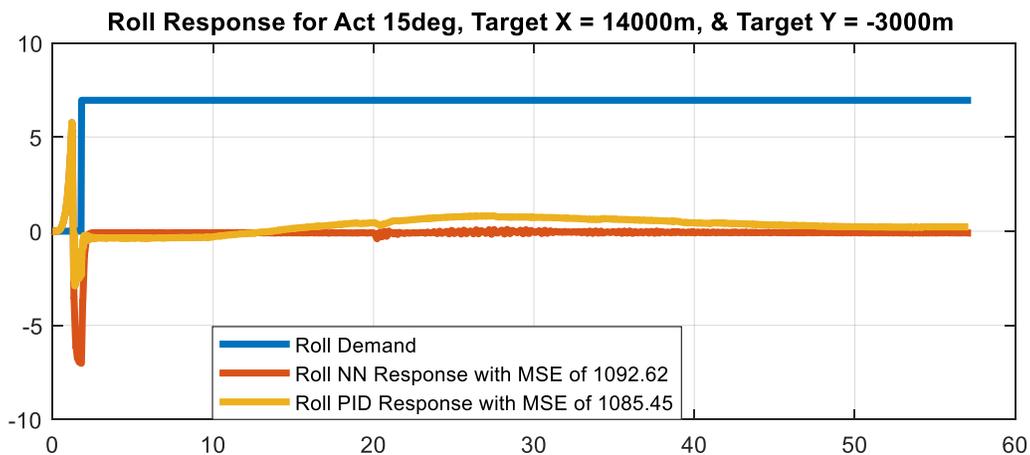
Figure 173 - Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 174 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.
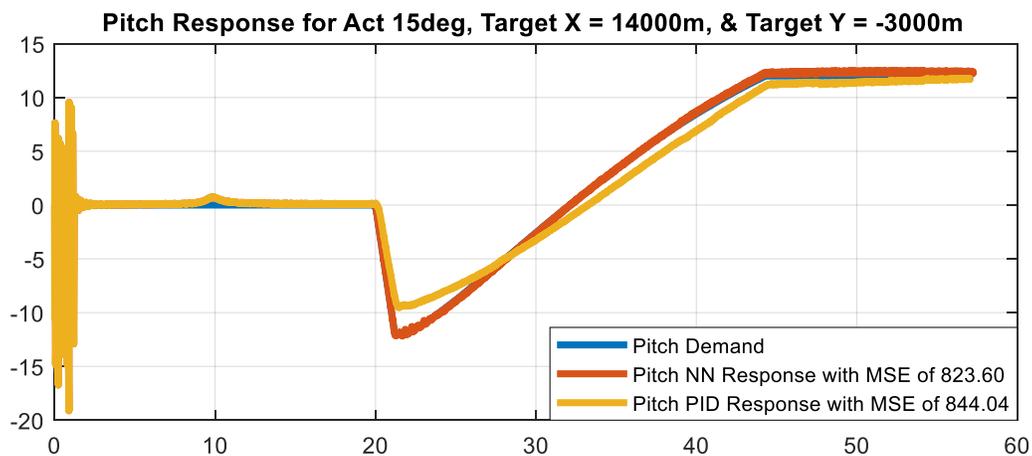


Figure 174 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 175 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 15deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the PID controller performed better than the Yaw LNNA controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



Figure 175 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 15deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 176 shows the comparison of the histogram distribution between PID and NNA controllers for range and deviation miss distances with a maximum actuator deflection of 15deg. It can be seen that PID has a higher mean and a higher standard of deviation. This means that for both cases PID resulted in more error. This leads to the conclusion that NNA is a viable if not a better alternative to the PID controller.
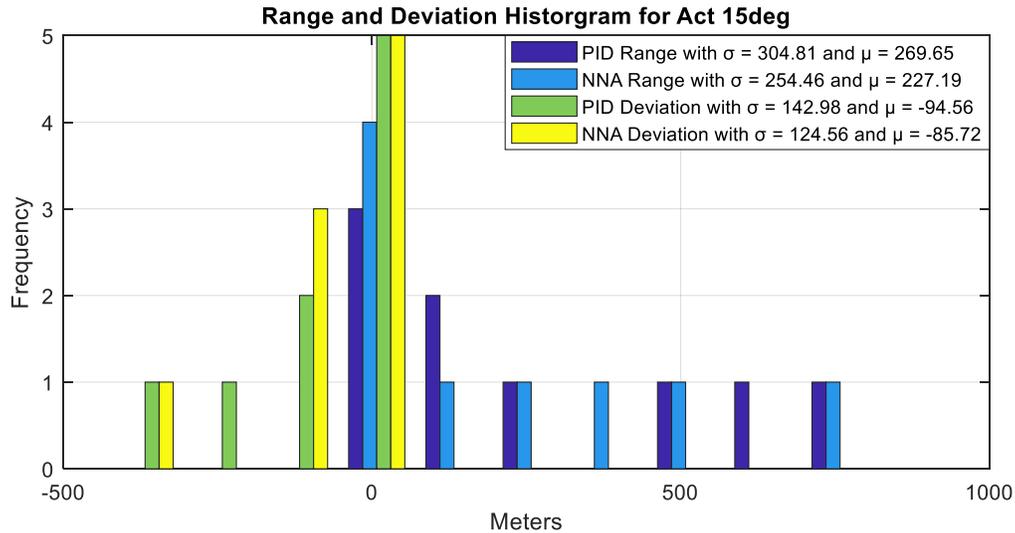


Figure 176 - Comparison of The Histogram Distribution Between PID and NNA Controllers for Range and Deviation Miss Distance

Figure 176 shows the comparison of the histogram distribution between the performances of the Roll PID and RNNA controllers. It can be seen that the RNNA controller has a lower mean. This means that the RNNA controller is more precise. On the other hand, the PID controller has a lower standard of deviation. This means that the PID controller is more accurate.
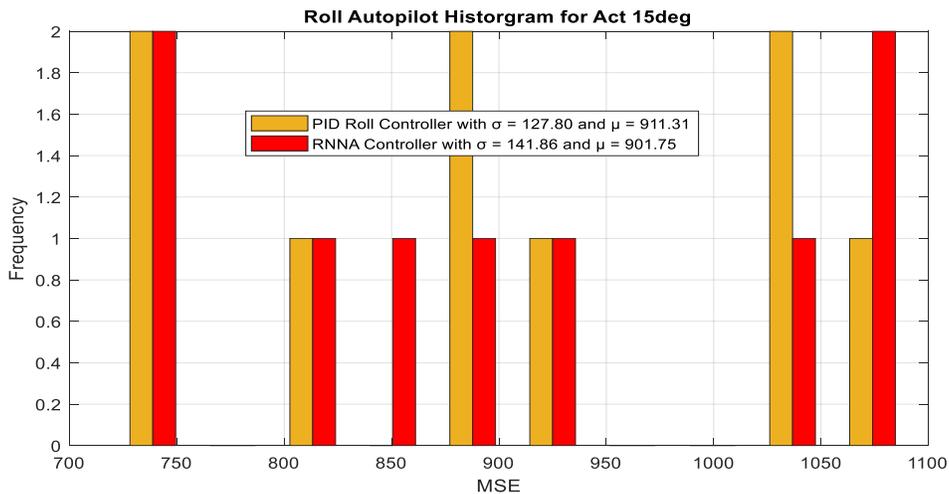


Figure 177 - Comparison of The Histogram Distribution Between The Performances of Roll PID and RNNA Controllers

Figure *178* shows the comparison of the histogram distribution between the performances of the Pitch PID and LNNA controllers. It can be seen that the PID controller has a higher mean and standard of deviation. This means that Pitch LNNA performed better overall.
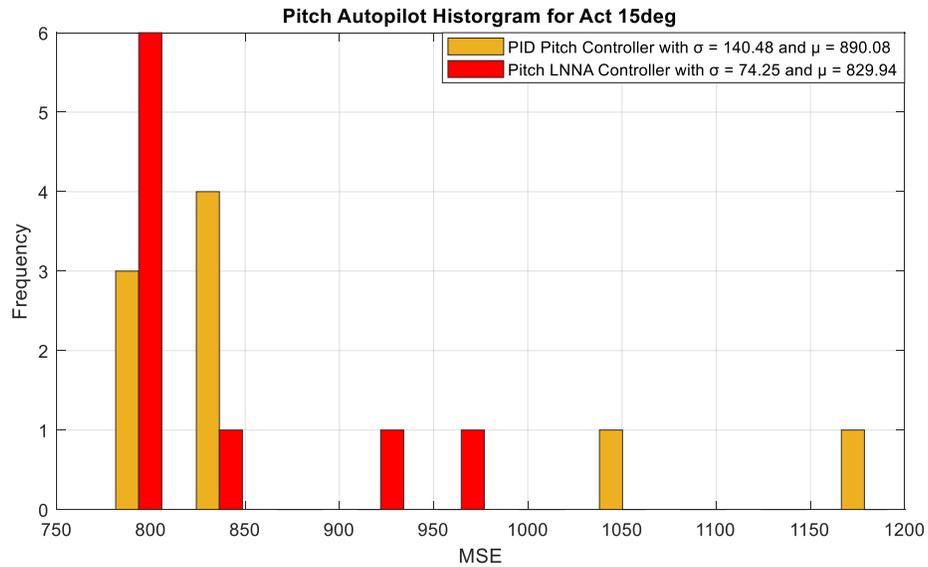


Figure 178 – Comparison of The Histogram Distribution Between The Performances of Pitch PID and LNNA Controllers

Figure *179* shows the comparison of the histogram distribution between the performances of the Yaw PID and LNNA controllers. It can be seen that the Yaw LNNA controller has a lower mean. This means that the Yaw LNNA controller is more precise. On the other hand, the PID controller has a lower standard of deviation. This means that the PID controller is more accurate.
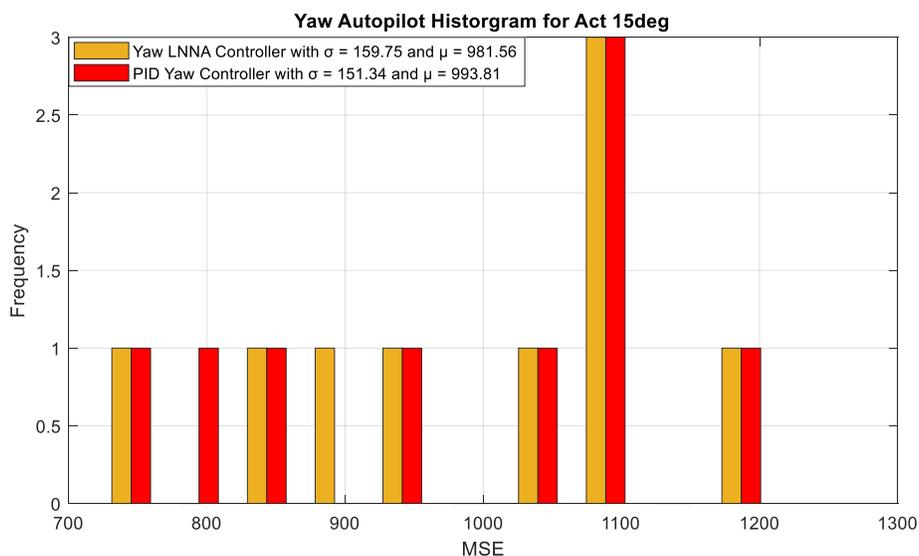


Figure 179 – Comparison of The Histogram Distribution Between The Performances of Yaw PID and LNNA Controllers

Figure 181 and Figure 186 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 10000m range and 0m deviation. Here, it can be seen that both PID and NNA algorithms performed almost perfectly. However, PID has a small range miss distance.
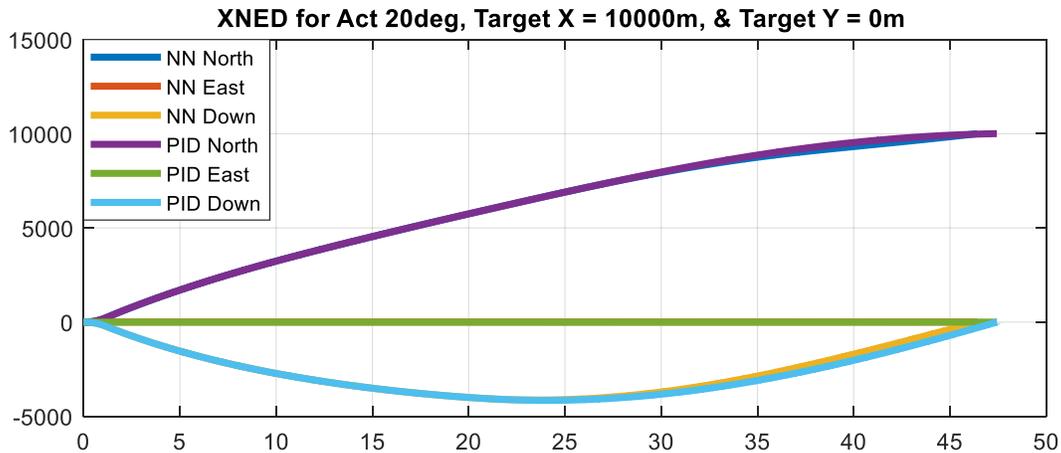


Figure 180 – Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = 0m
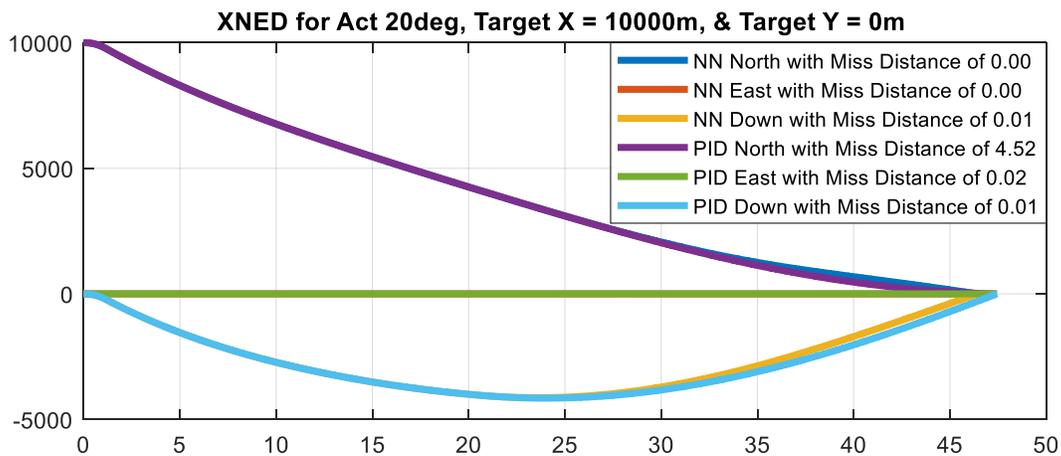


Figure 181 – Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = 0m

Figure 182 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and 0m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
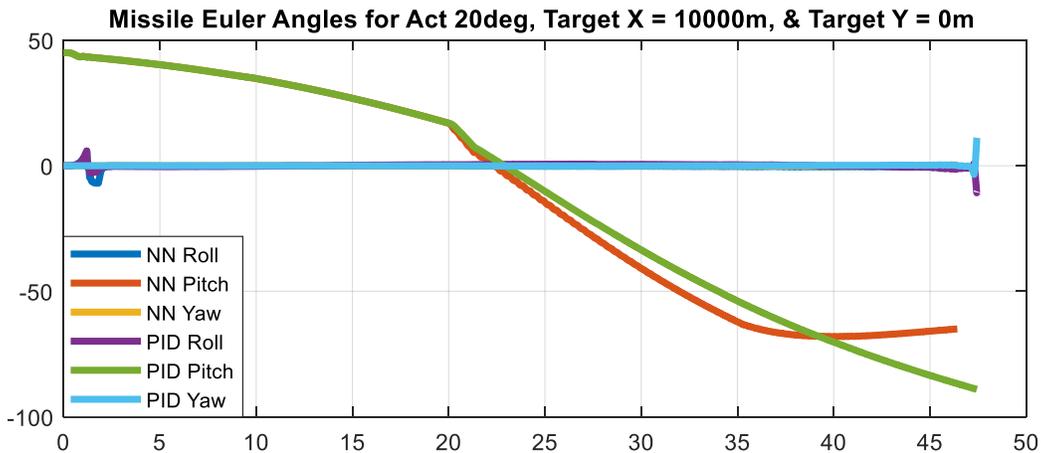


Figure 182 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = 0m

Figure 183 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and 0m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
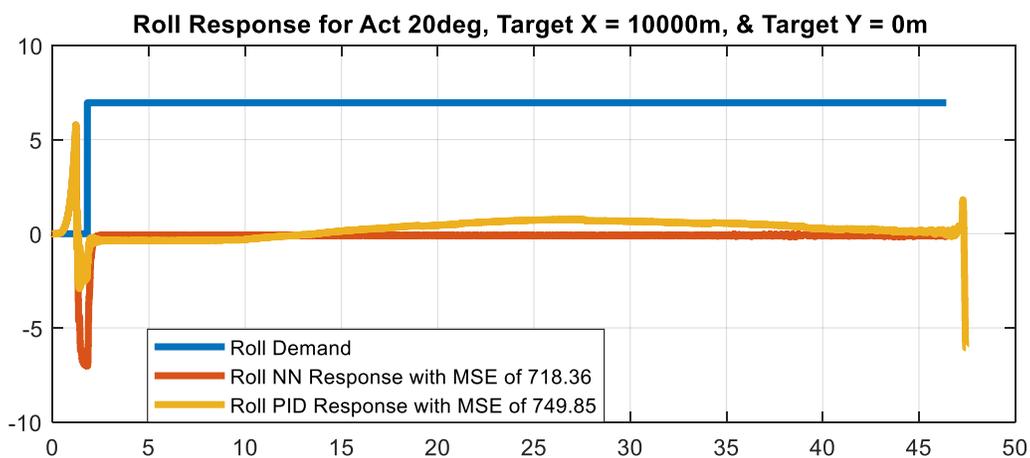


Figure 183 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = 0m

Figure 184 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and 0m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller shows an unusual behavior which contributed to the small miss distance.



**Pitch Response for Act 20deg, Target X = 10000m, & Target Y = 0m**
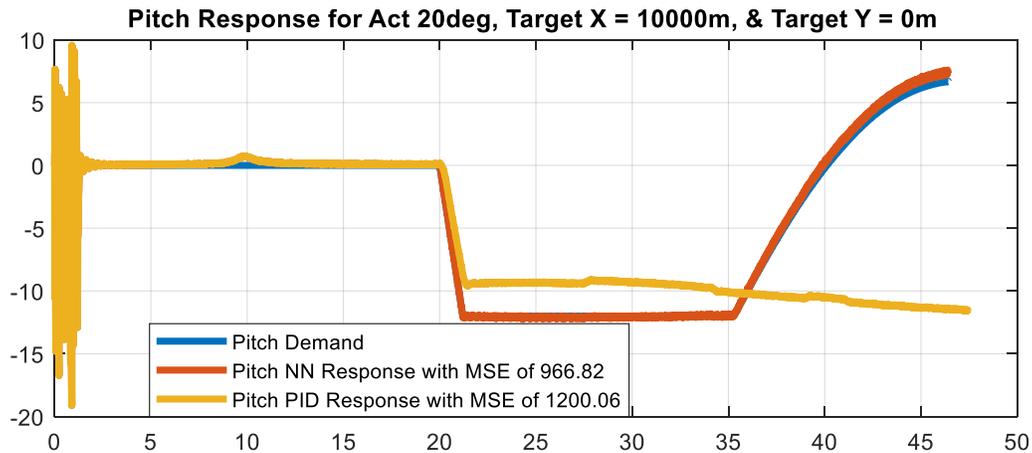
Figure 184 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = 0m

Figure 185 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and 0m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 20deg, Target X = 10000m, & Target Y = 0m**
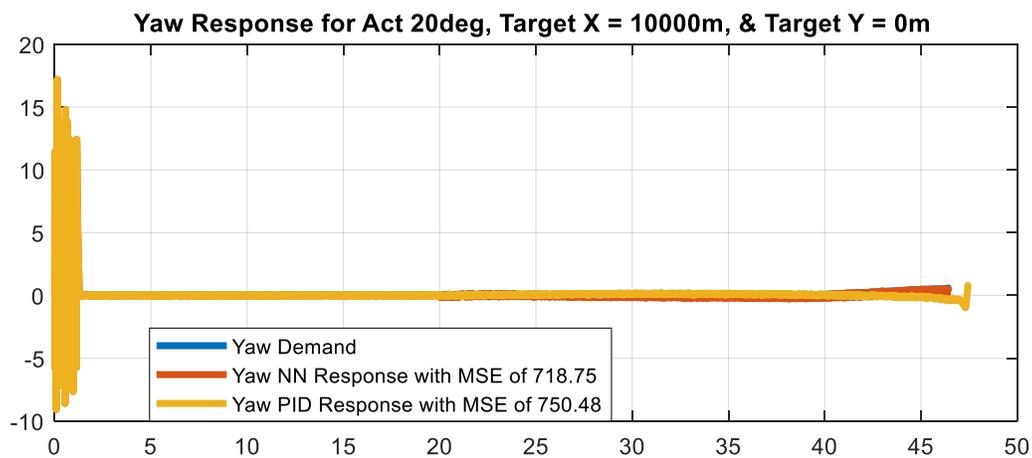
Figure 185 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = 0m

Figure 186 and Figure 187 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 10000m range and -1000m deviation. Here, it can be seen that both PID and NNA algorithms performed almost perfectly.
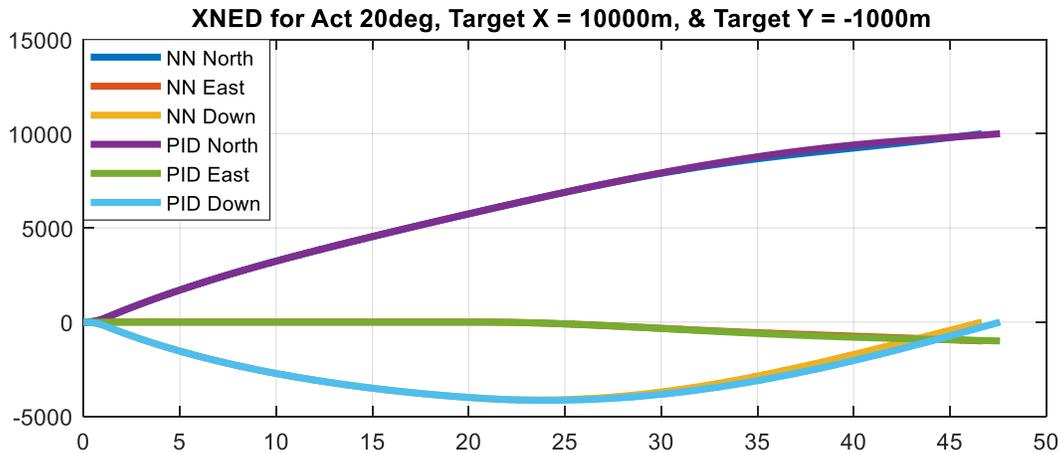


Figure 186 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -1000m
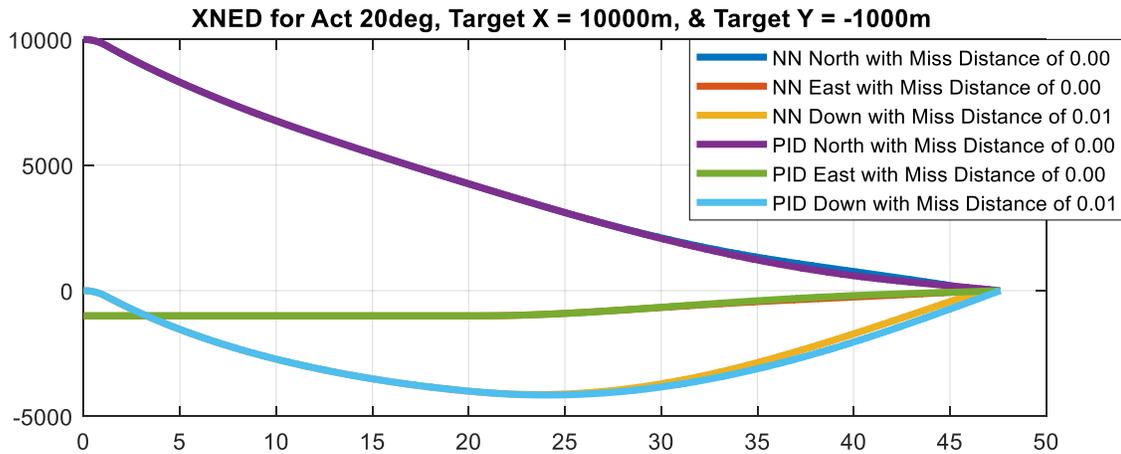


Figure 187 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 188 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and -1000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
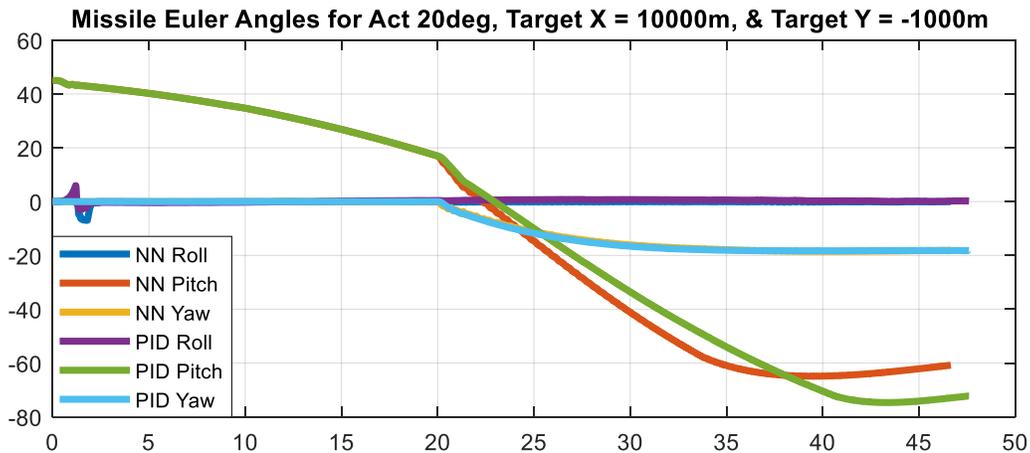


Figure 188 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 189 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and -1000m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
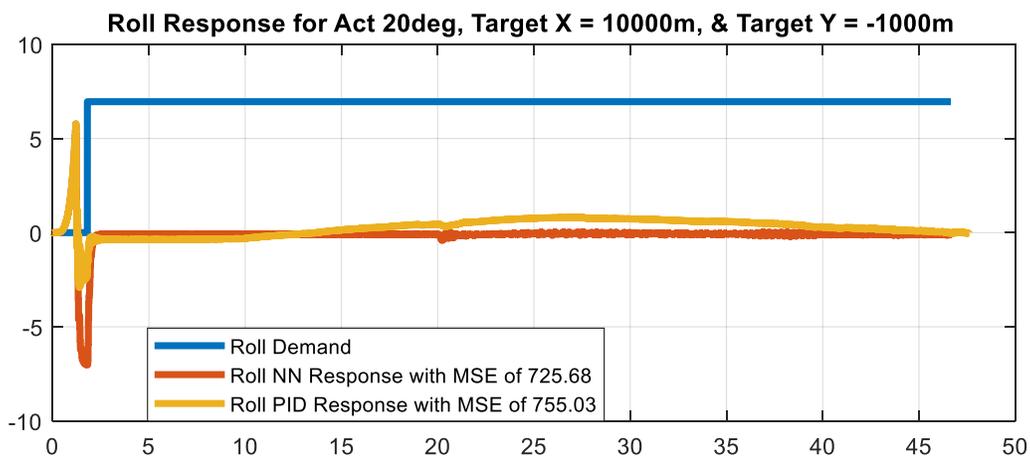


Figure 189 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 190 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and -1000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.



**Pitch Response for Act 20deg, Target X = 10000m, & Target Y = -1000m**

Legend:
- Pitch Demand
- Pitch NN Response with MSE of 928.92
- Pitch PID Response with MSE of 1056.07
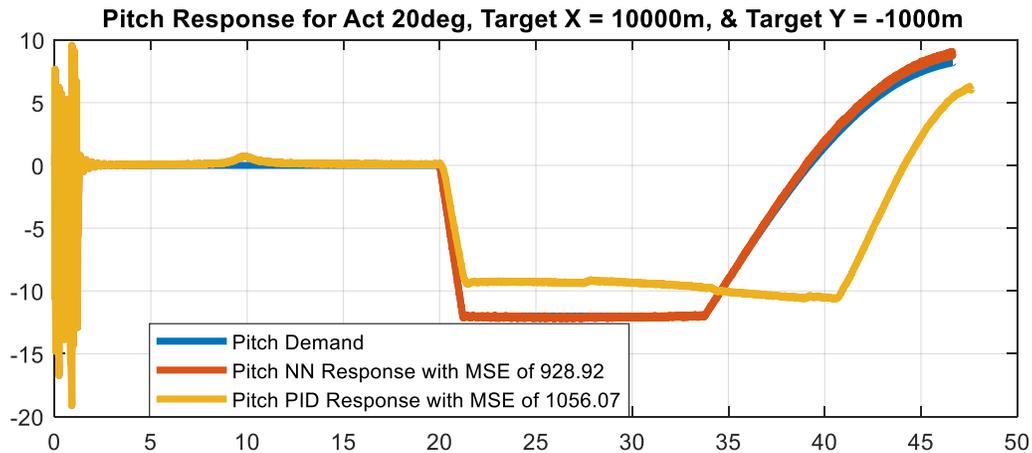
Figure 190 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 191 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and -1000m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 20deg, Target X = 10000m, & Target Y = -1000m**

Legend:
- Yaw Demand
- Yaw NN Response with MSE of 809.72
- Yaw PID Response with MSE of 837.44
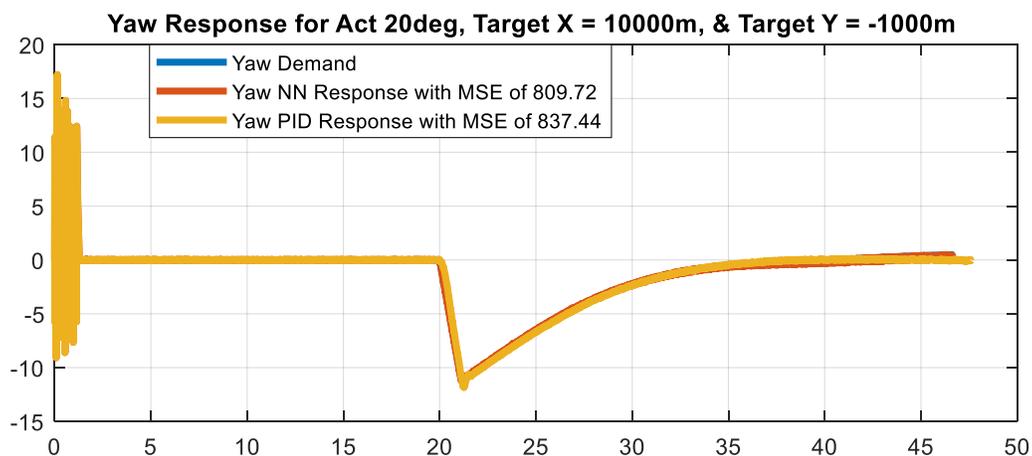
Figure 191 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -1000m

Figure 192 and Figure 193 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the PID controller performed better for both range and deviation with less miss distance.
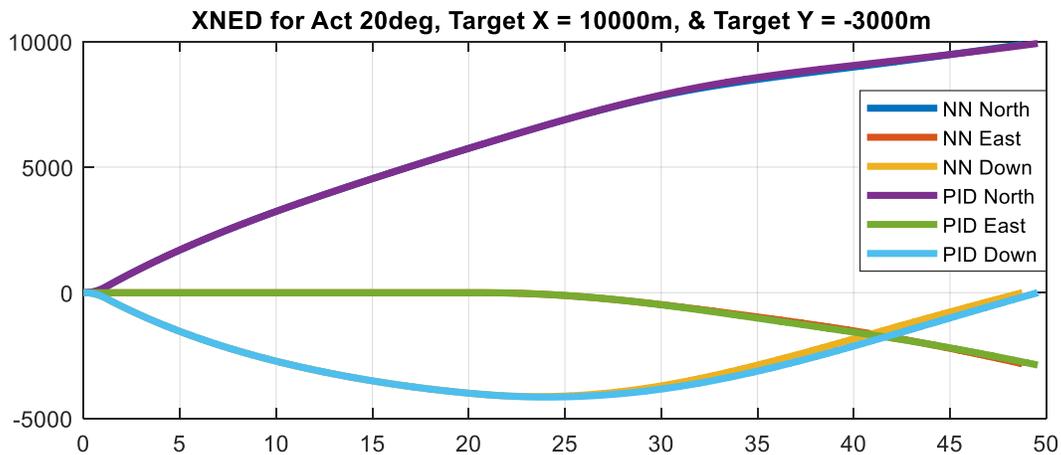


Figure 192 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -3000m
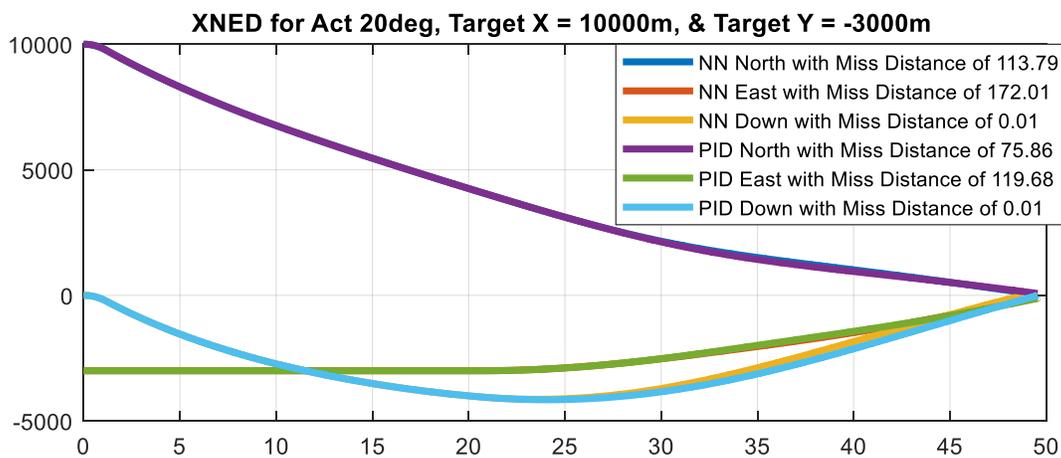


Figure 193 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 194 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.

**Missile Euler Angles for Act 20deg, Target X = 10000m, & Target Y = -3000m**
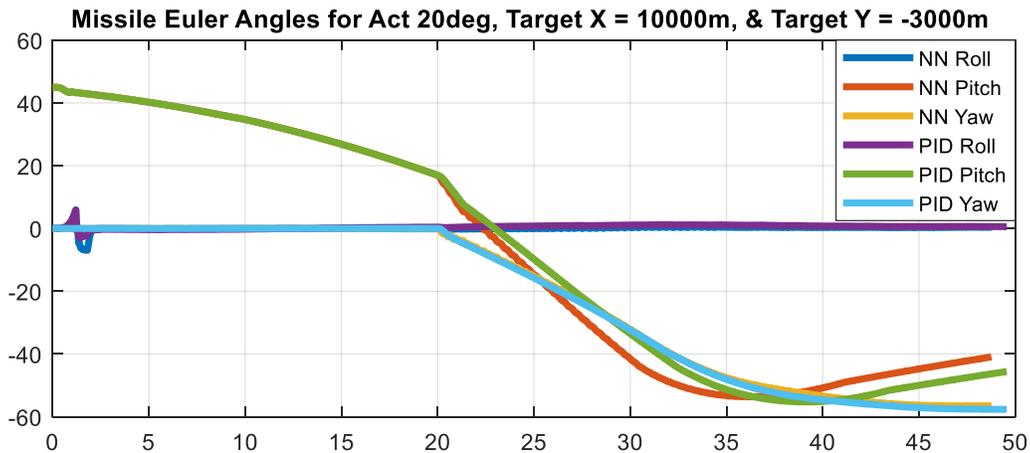


Figure 194 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 195 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.

**Roll Response for Act 20deg, Target X = 10000m, & Target Y = -3000m**
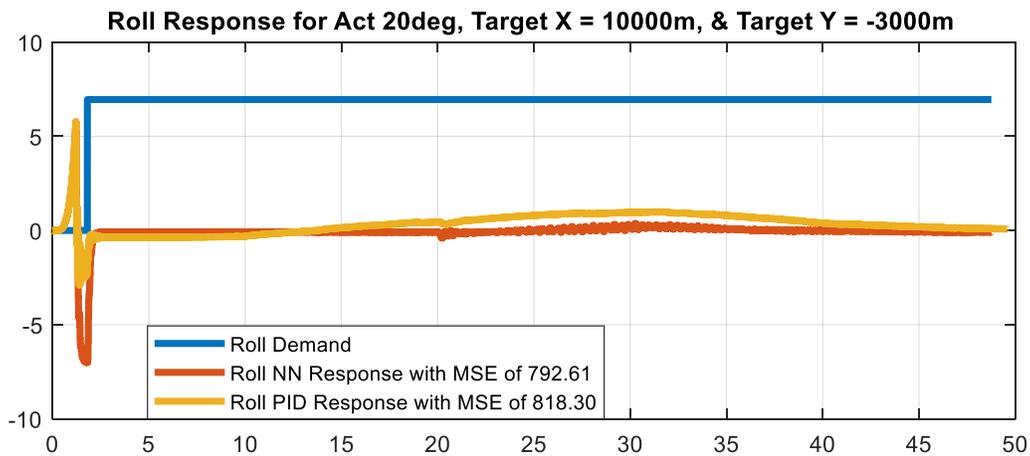


Figure 195 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 196 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the Pitch LNNA controllers performed better than the PID controller as it has a lower MSE value. It is interesting because despite Pitch LNNA controller having a better performance, the PID resulted in less range miss distance. The PID controller does not match as well because it was tuned for the transfer function.

**Pitch Response for Act 20deg, Target X = 10000m, & Target Y = -3000m**



Figure 196 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 197 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 10000m range and -3000m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. It is interesting because despite Yaw LNNA controller having a better performance, the PID resulted in less range miss distance. Nevertheless, the performance of both controllers is very similar.

**Yaw Response for Act 20deg, Target X = 10000m, & Target Y = -3000m**
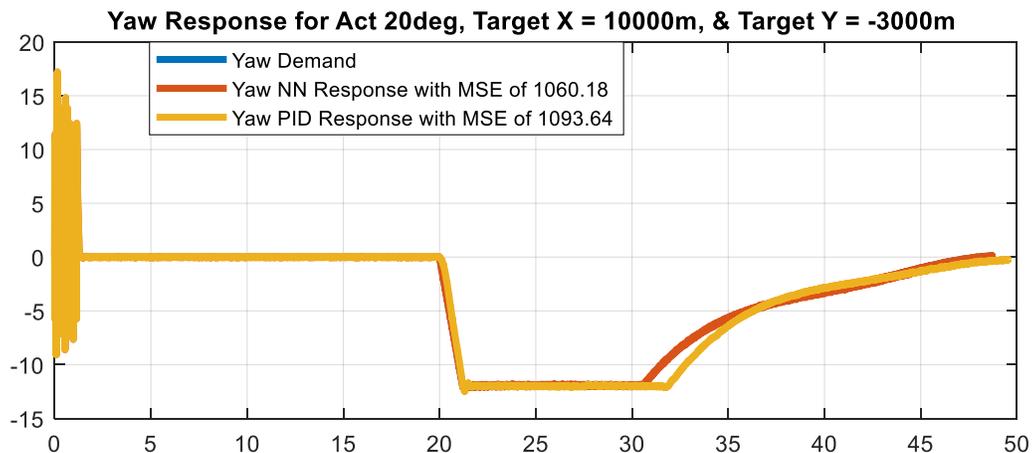


Figure 197 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 10000m and Yt = -3000m

Figure 198 and Figure 199 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 12000m range and 0m deviation. Here, it can be seen that the NNA controller performed better for range with less miss distance. However, the PID controller performed better for deviation with less miss distance.
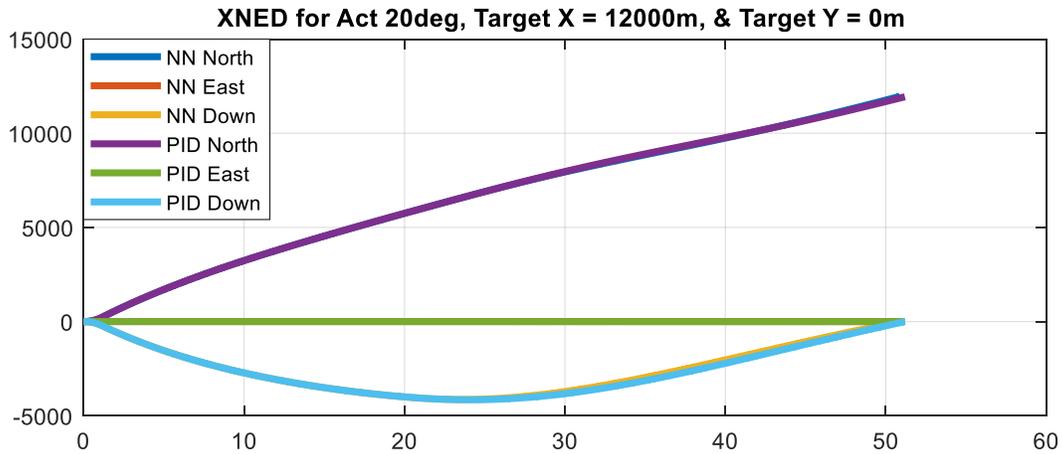


Figure 198 – Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = 0m
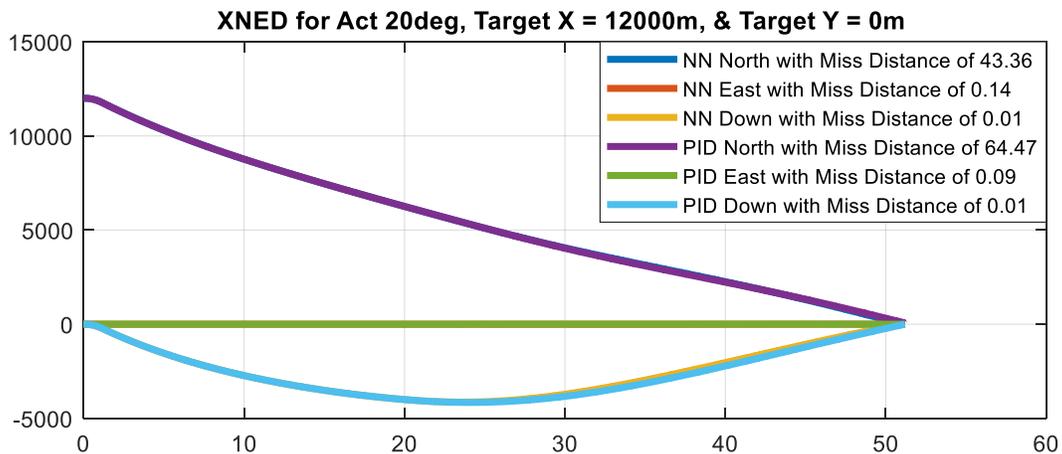


Figure 199 – Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = 0m

Figure 200 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and 0m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
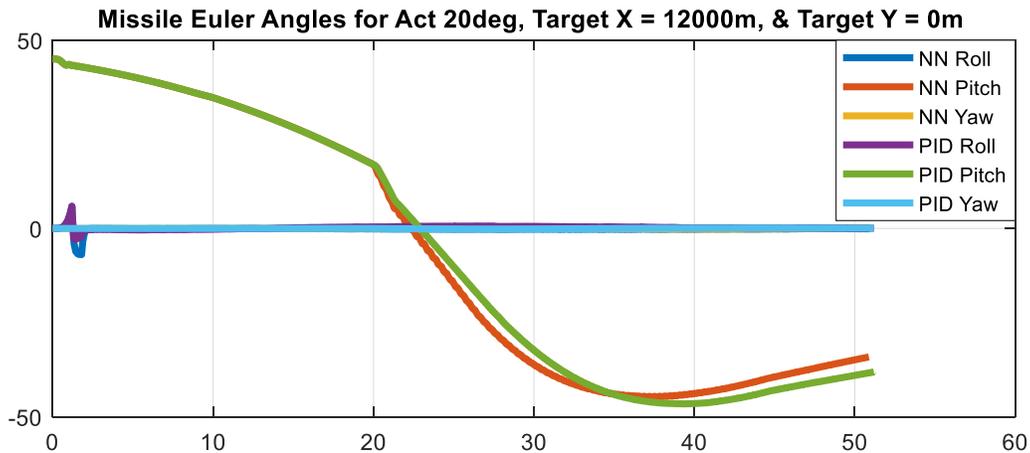


Figure 200 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = 0m

Figure 201 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and 0m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
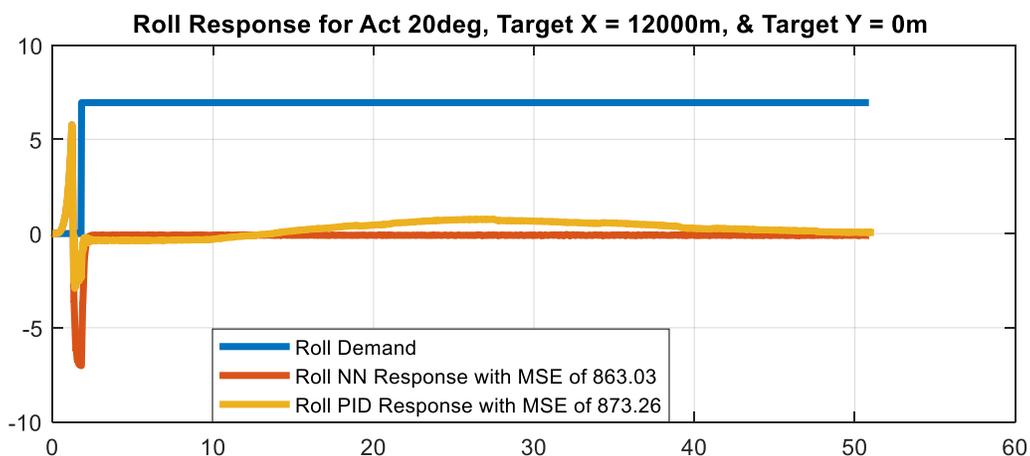


Figure 201 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = 0m

Figure 202 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and 0m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.
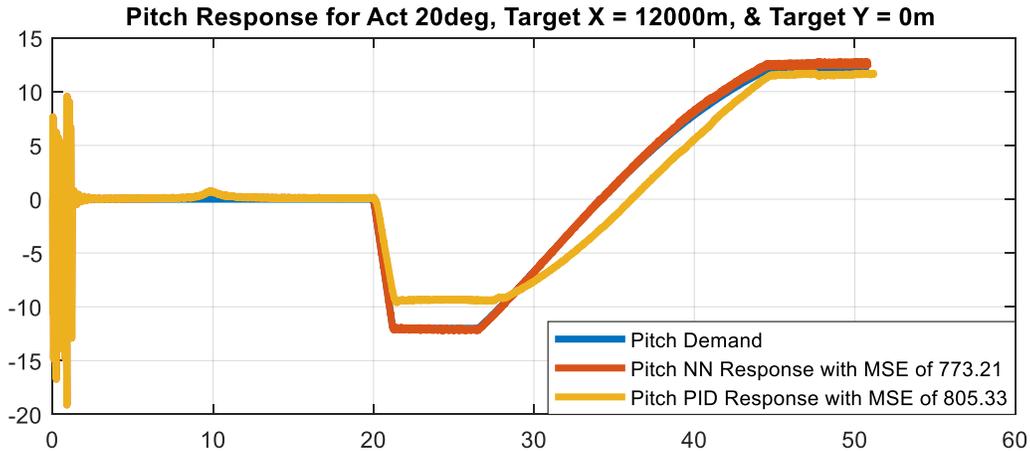


Figure 202 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = 0m

Figure 203 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and 0m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. It is interesting because despite Yaw LNNA controller having a better performance, the PID resulted in less range miss distance. Nevertheless, the performance of both controllers is very similar.
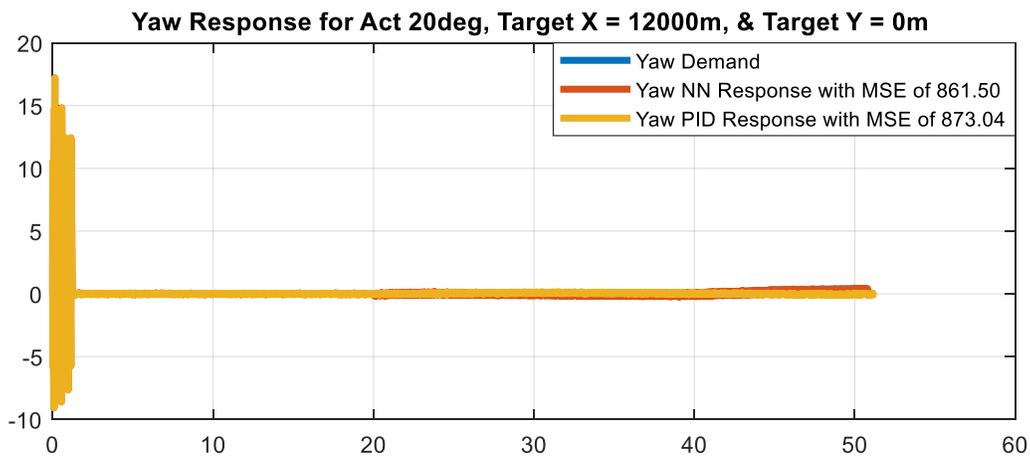


Figure 203 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = 0m

Figure 204 and Figure 205 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the NNA controller performed better for both range and deviation.
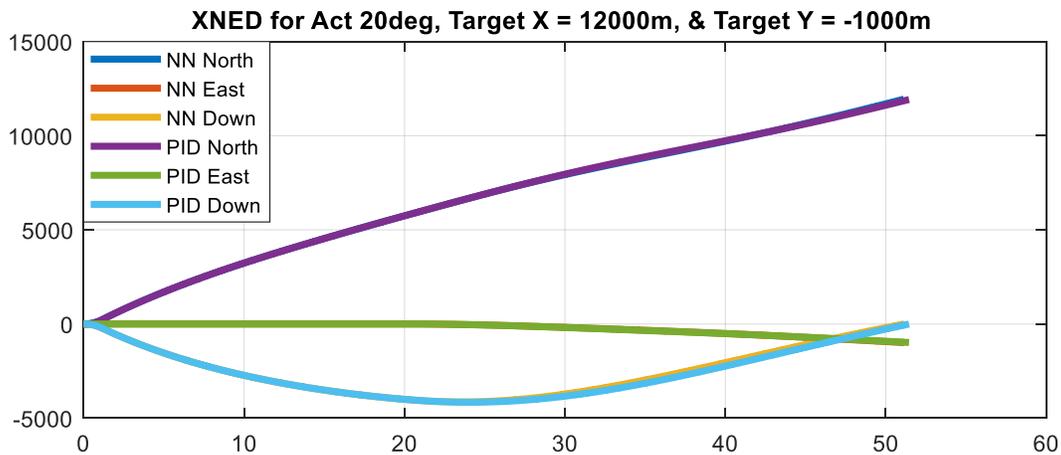


Figure 204 – Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -1000m
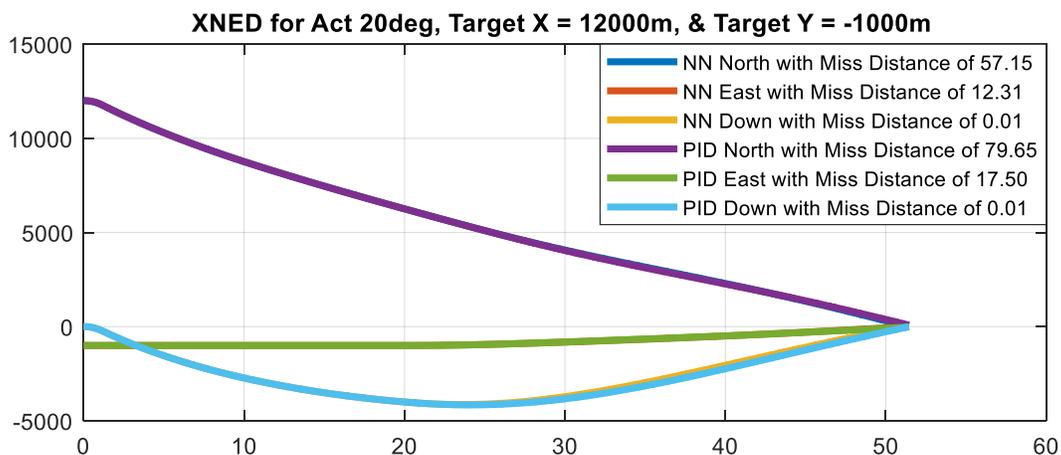


Figure 205 – Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 206 shows the missile Euler angles comparisons between NNA and PID controller for a maximum actuator deflection of 20deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.

**Missile Euler Angles for Act 20deg, Target X = 12000m, & Target Y = -1000m**
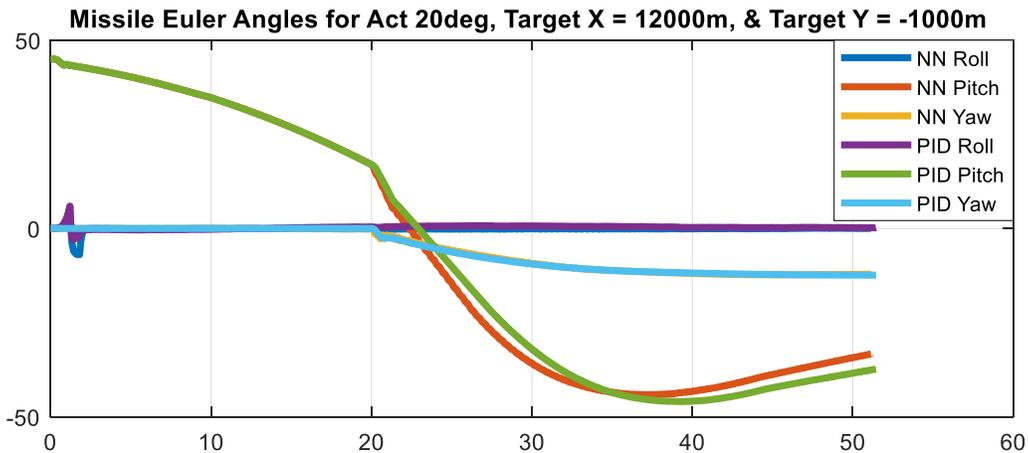


Figure 206 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 207 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.

**Roll Response for Act 20deg, Target X = 12000m, & Target Y = -1000m**
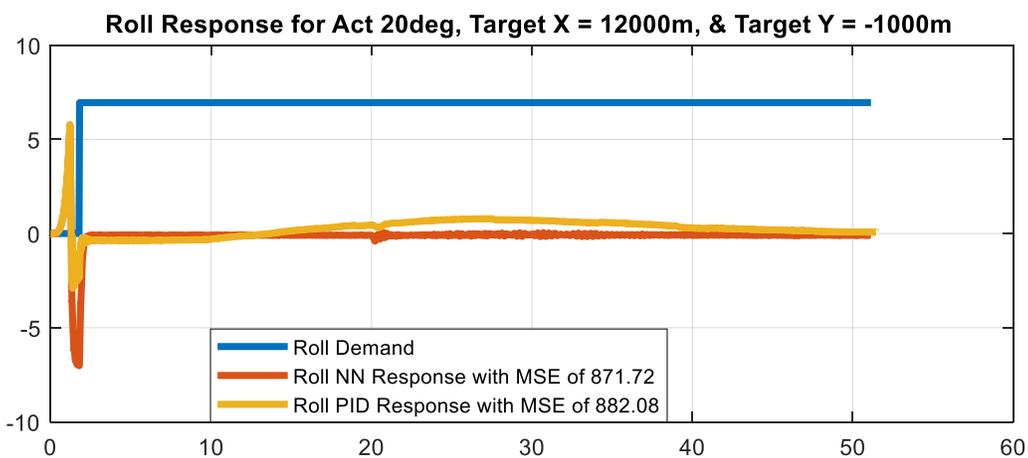


Figure 207 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 208 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.
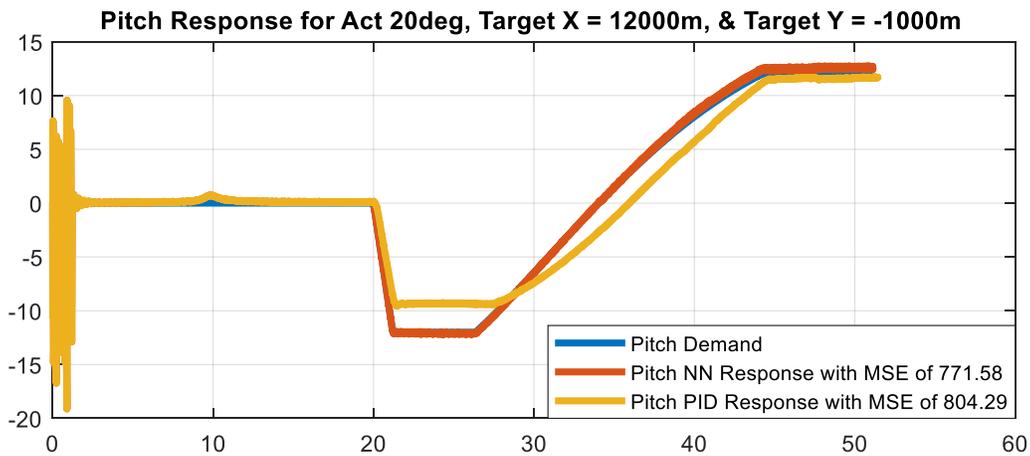


Figure 208 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 209 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and -1000m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.
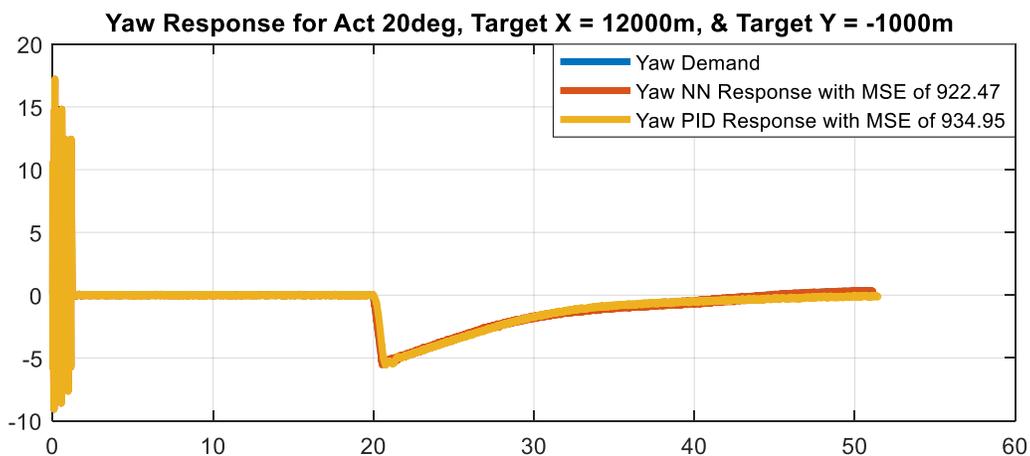


Figure 209  – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -1000m

Figure 210 and Figure 211 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the NNA controller performed better for both range and deviation.
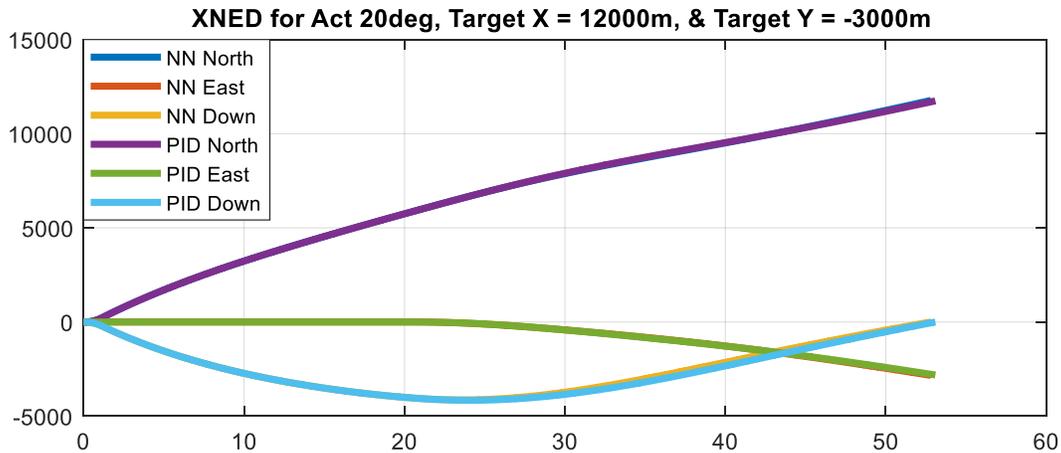


Figure 210 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -3000m
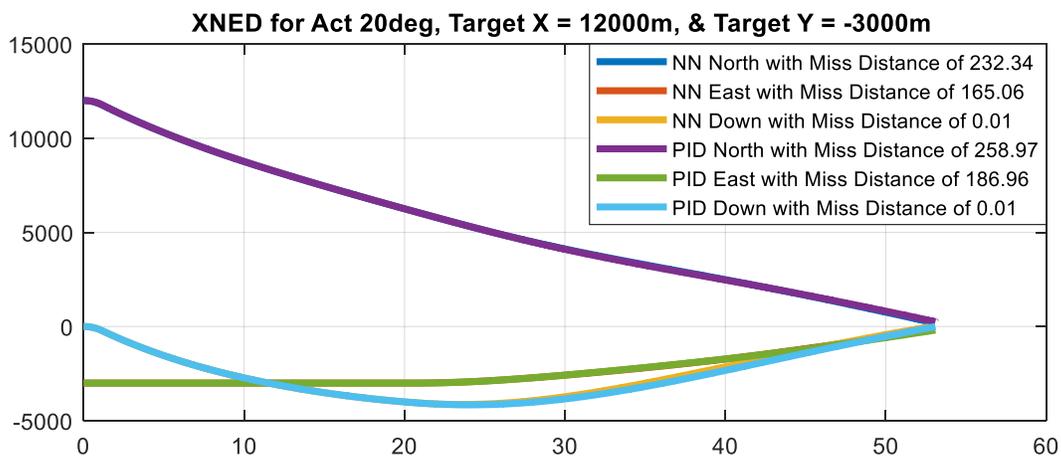


Figure 211 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 212 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
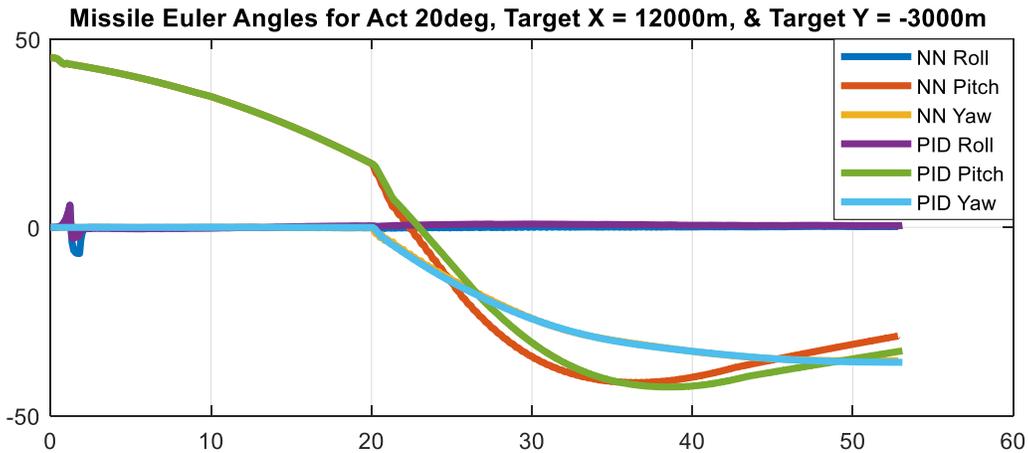


Figure 212 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 213 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the RNNA controller performed better than the PID controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
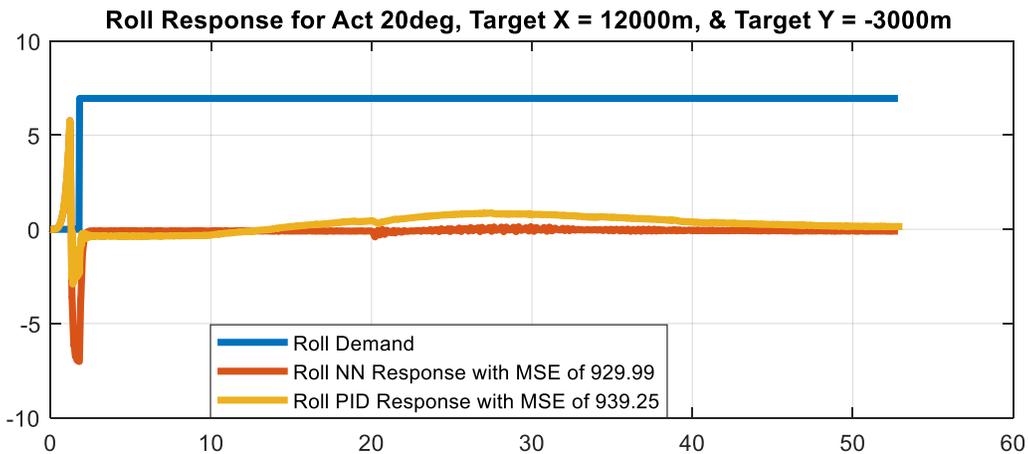


Figure 213 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 214 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.



**Pitch Response for Act 20deg, Target X = 12000m, & Target Y = -3000m**

Legend:
- Pitch Demand
- Pitch NN Response with MSE of 770.78
- Pitch PID Response with MSE of 801.67
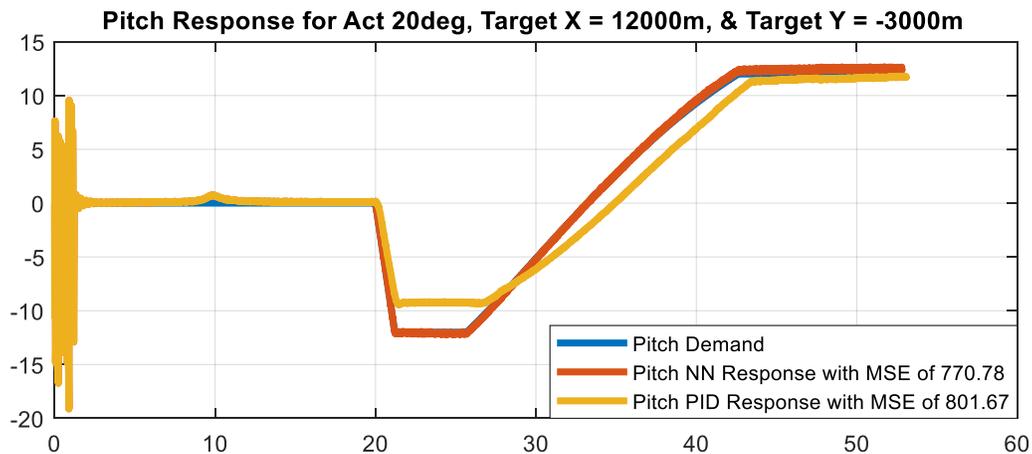
Figure 214 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 215 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 12000m range and -3000m deviation. Here, it can be seen that the Yaw LNNA controller performed better than the PID controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 20deg, Target X = 12000m, & Target Y = -3000m**

Legend:
- Yaw Demand
- Yaw NN Response with MSE of 1090.61
- Yaw PID Response with MSE of 1103.36

Figure 215 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 12000m and Yt = -3000m

Figure 216 and Figure 217 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 14000m range and 0m deviation. Here, it can be seen that the NNA controller performed better for range. However, the PID controller performed better for deviation.
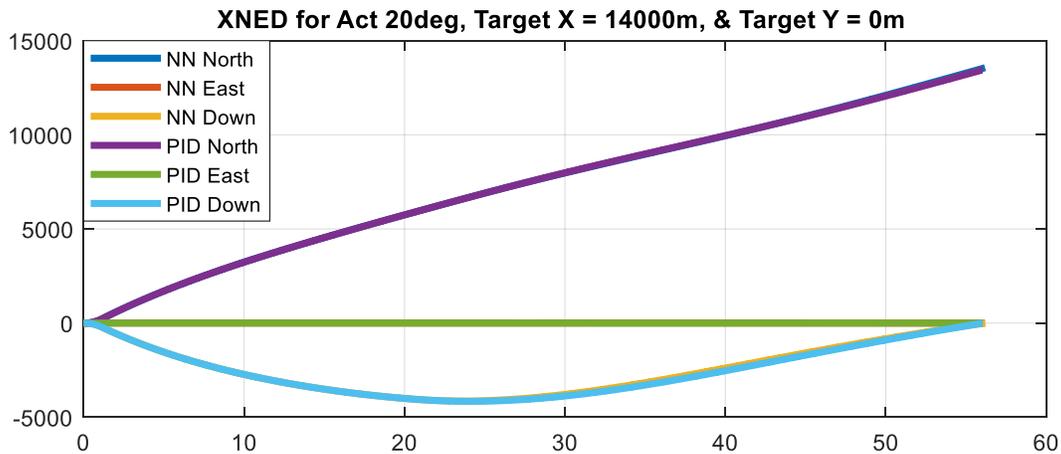


Figure 216 – Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = 0m
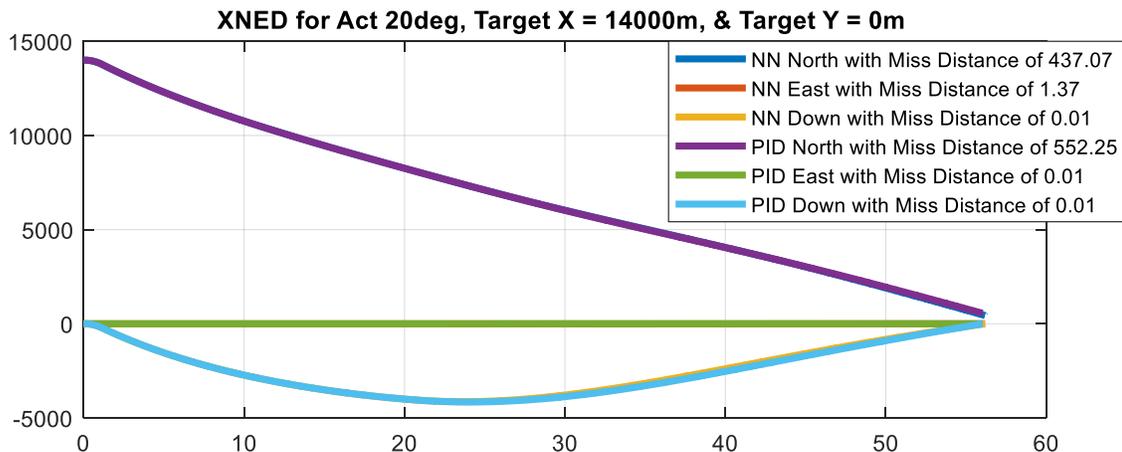


Figure 217 – Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = 0m

Figure 218 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and 0m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.



**Missile Euler Angles for Act 20deg, Target X = 14000m, & Target Y = 0m**
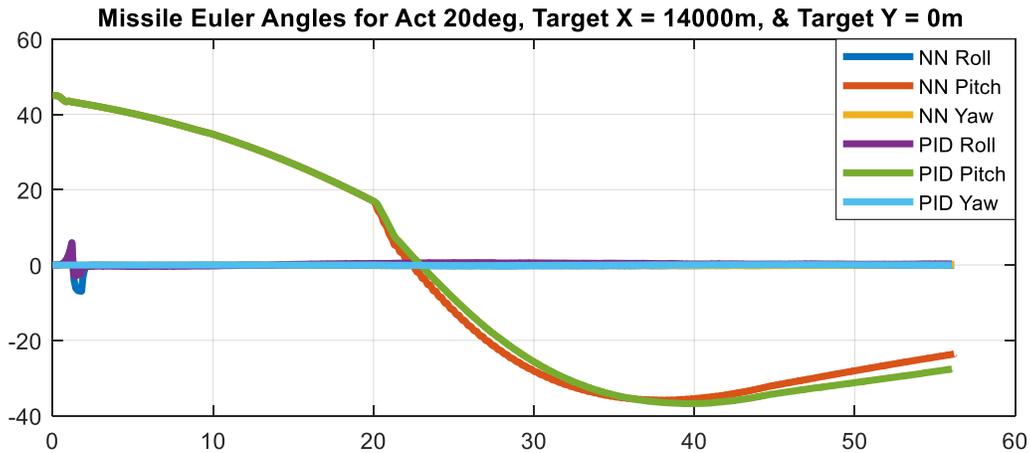
Figure 218 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = 0m

Figure 219 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and 0m deviation. Here, it can be seen that the PID controller performed better than the RNNA controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.



**Roll Response for Act 20deg, Target X = 14000m, & Target Y = 0m**
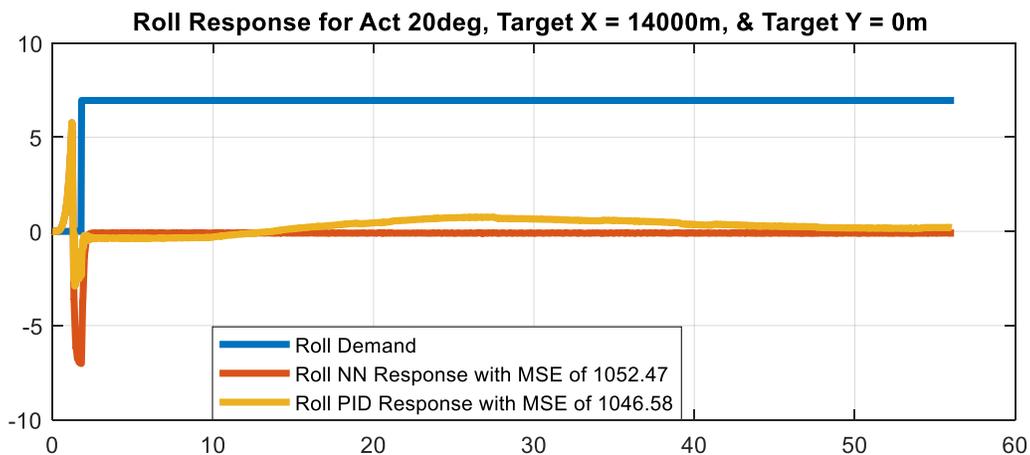
Figure 219 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = 0m

Figure 220 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and 0m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.



**Pitch Response for Act 20deg, Target X = 14000m, & Target Y = 0m**

Legend:
- Pitch Demand
- Pitch NN Response with MSE of 809.27
- Pitch PID Response with MSE of 830.32
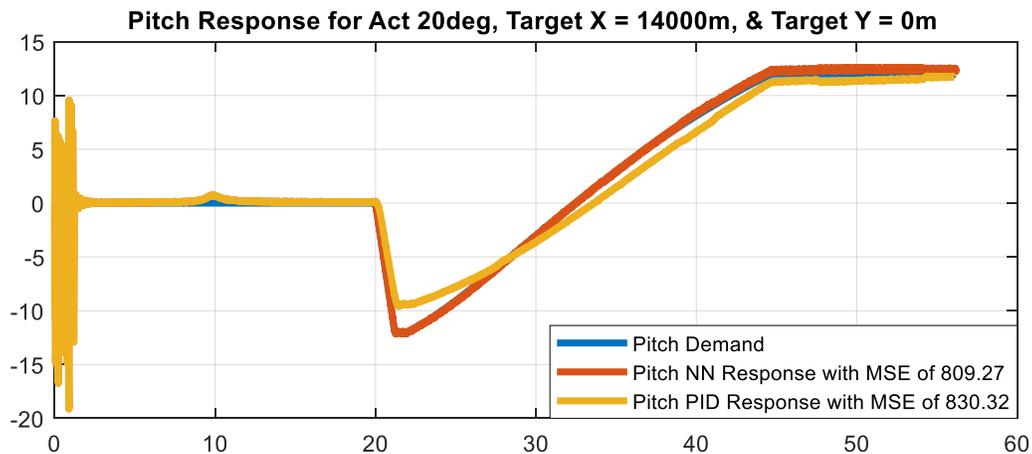
Figure 220 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = 0m

Figure 221 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and 0m deviation. Here, it can be seen that the PID controller performed better than the Yaw LNNA controller as it has a lower MSE value. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 20deg, Target X = 14000m, & Target Y = 0m**

Legend:
- Yaw Demand
- Yaw NN Response with MSE of 1050.90
- Yaw PID Response with MSE of 1048.06
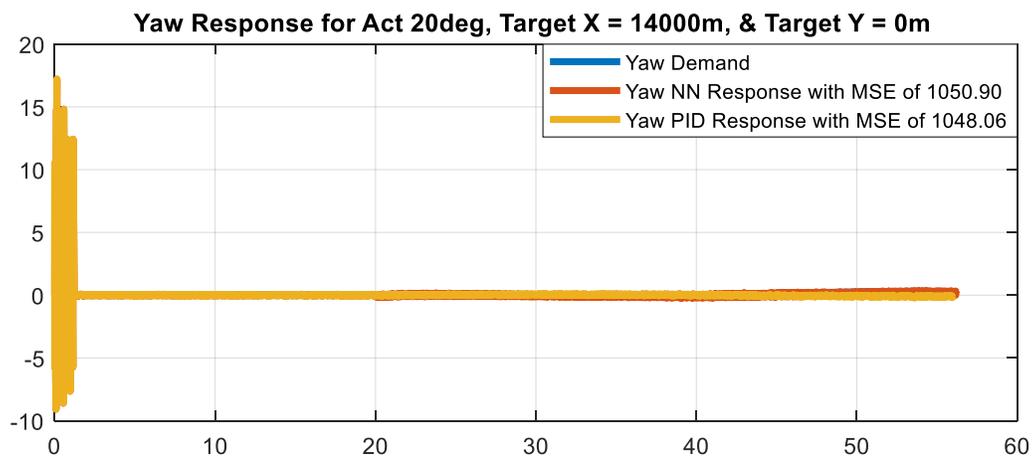
Figure 221 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = 0m

Figure 222 and Figure 223 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the NNA controller performed better for both range and deviation.
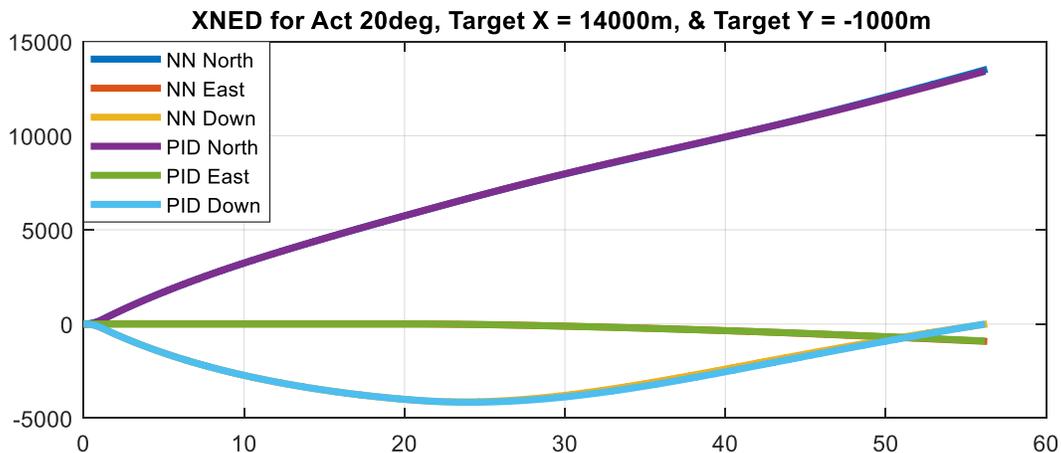


Figure 222 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -1000m
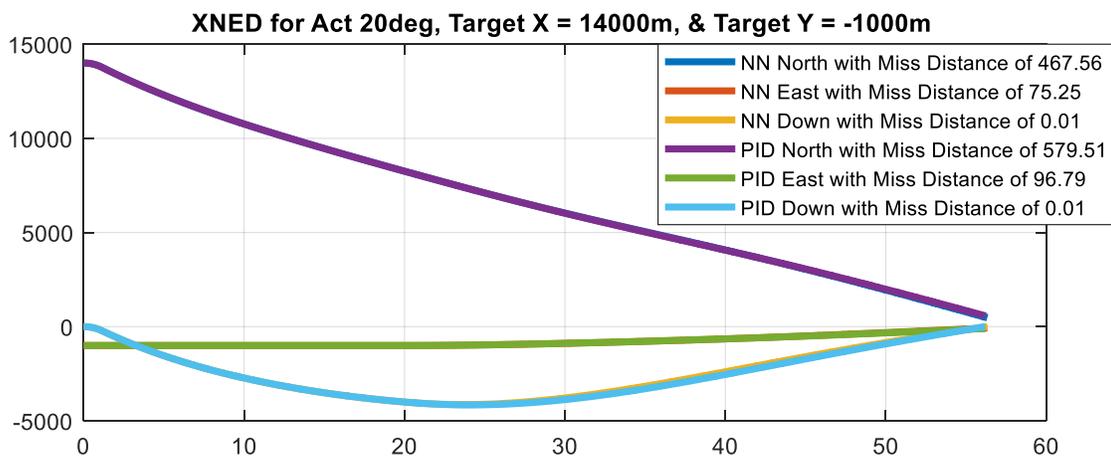


Figure 223 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 224 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
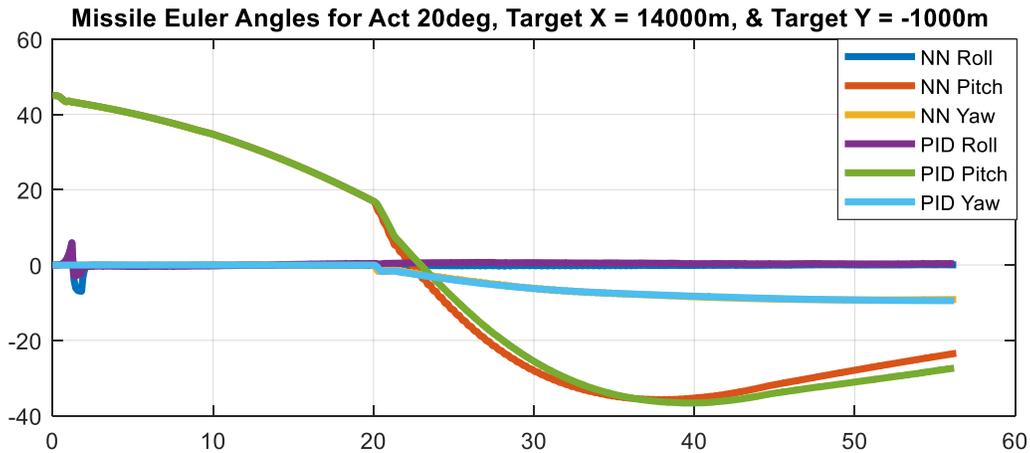


Figure 224 – Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 225 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the PID controller performed better than the RNNA controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
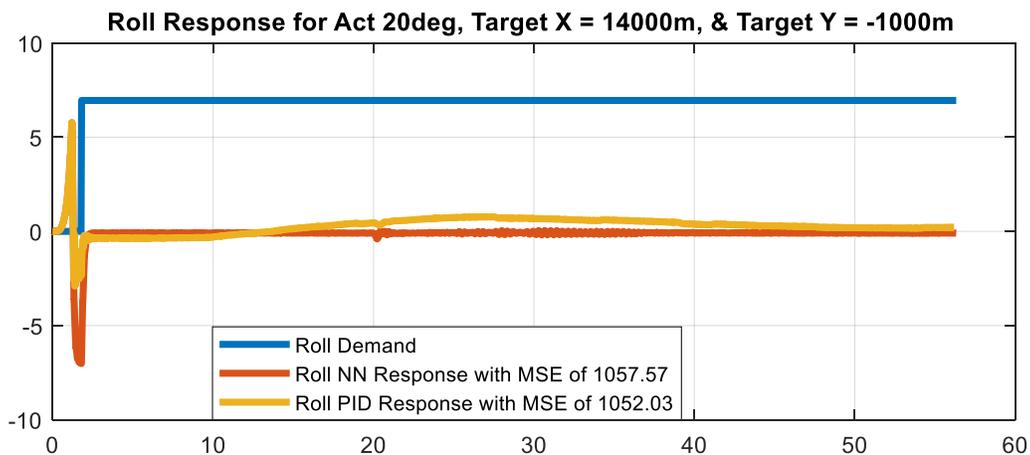


Figure 225 – Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 226 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.



**Pitch Response for Act 20deg, Target X = 14000m, & Target Y = -1000m**
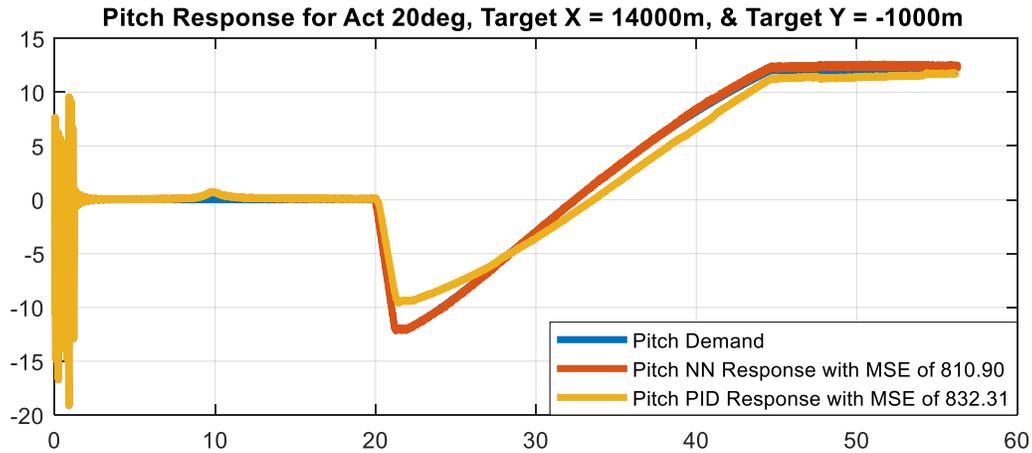
Figure 226 – Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 227 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and -1000m deviation. Here, it can be seen that the PID controller performed better than the Yaw LNNA controller as it has a lower MSE value. It is interesting because despite PID controller having a better performance, the Yaw LNNA resulted in less range miss distance. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 20deg, Target X = 14000m, & Target Y = -1000m**

Figure 227 – Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -1000m

Figure 228 and Figure 229 show the autopilot performance of the missile for a maximum actuator deflection of 20deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the NNA controller performed better for both range and deviation.
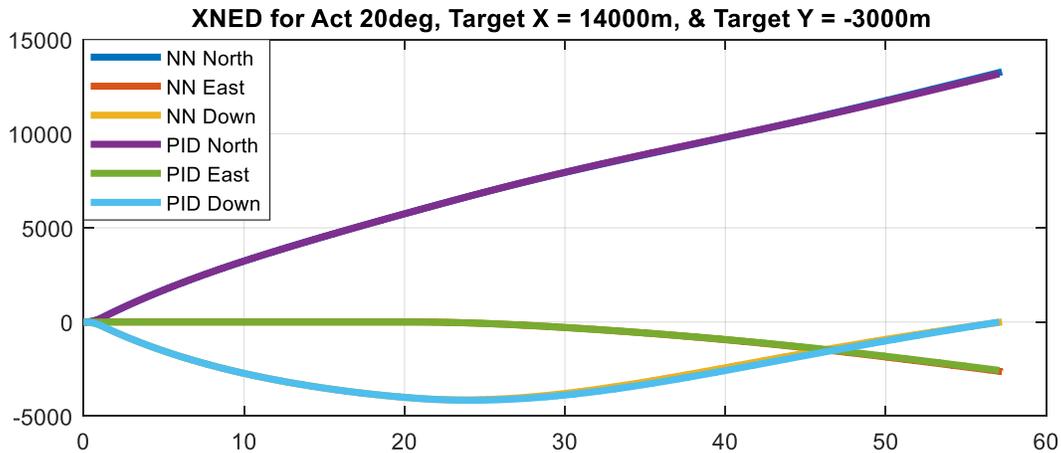


Figure 228 - Comparison of NED Frame Position Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -3000m
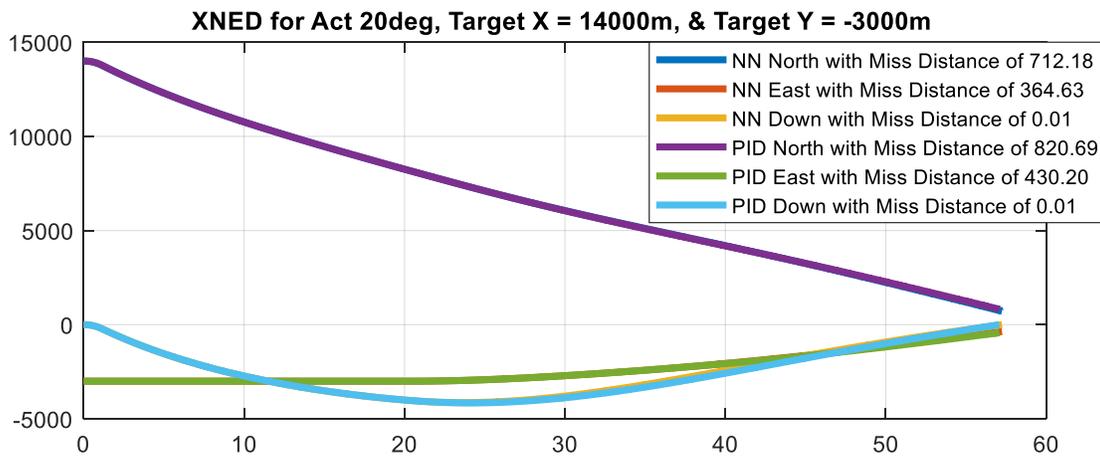


Figure 229 - Comparison of NED Frame Position Miss Distance Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 230 shows the missile Euler angles comparisons between NNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the PID controller is smoother than the NNA controller, which has a small oscillation.
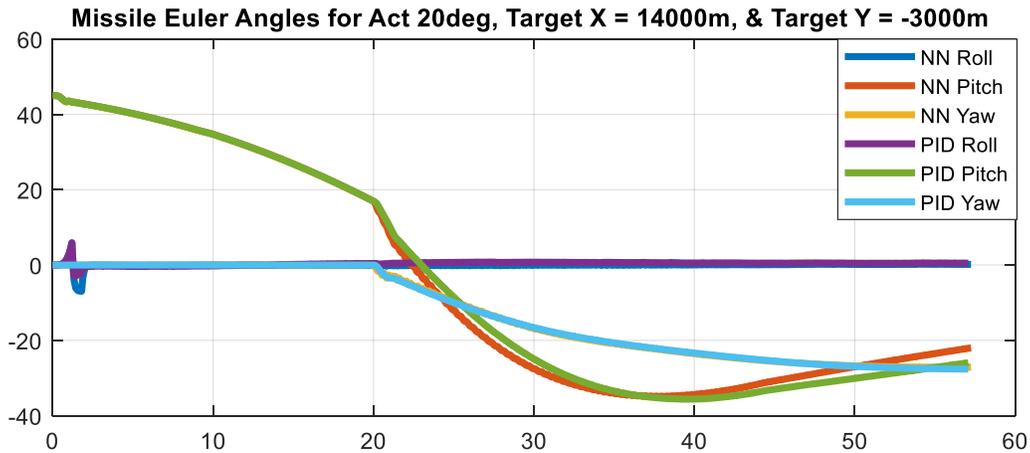


Figure 230 - Comparison of Missile Euler Angles Between NNA and PID for a Maximum Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 231 shows the comparison between RNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the PID controller performed better than the RNNA controller as it has a lower MSE value. It is important to note that the roll demand is adjusted to account for the static error.
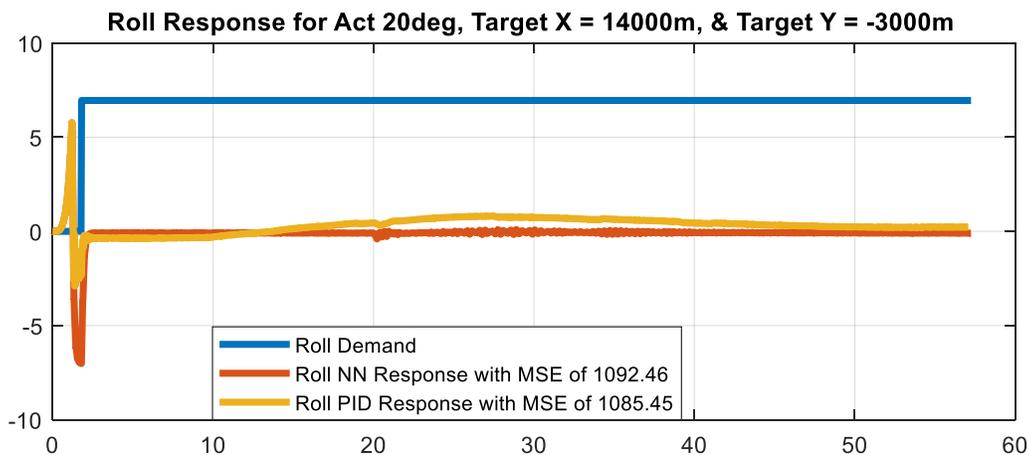


Figure 231 - Comparison of Roll Response Between RNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 232 shows the comparison between Pitch LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the Pitch LNNA controller performed better than the PID controller as it has a lower MSE value. The PID controller does not match as well because it was tuned for the transfer function.



**Pitch Response for Act 20deg, Target X = 14000m, & Target Y = -3000m**
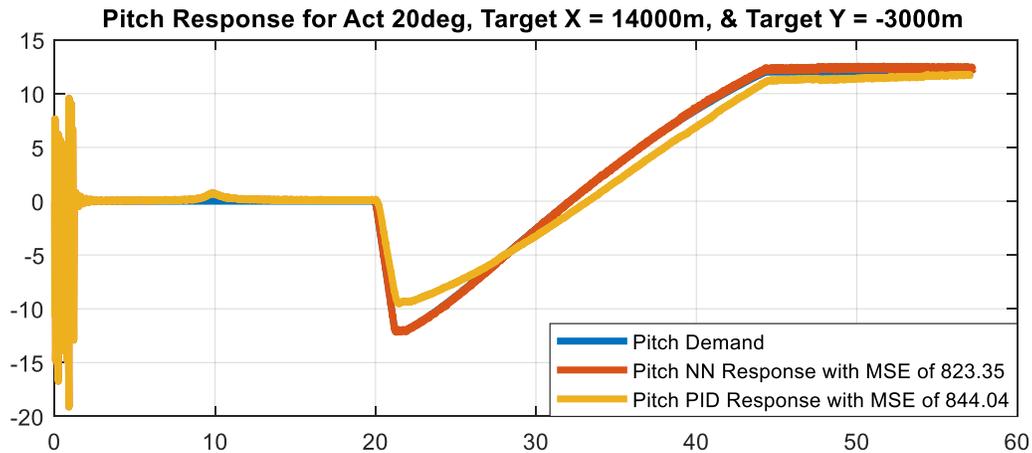
Figure 232 - Comparison of Pitch Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 233 shows the comparison between Yaw LNNA and PID controllers for a maximum actuator deflection of 20deg and the target at 14000m range and -3000m deviation. Here, it can be seen that the PID controller performed better than the Yaw LNNA controller as it has a lower MSE value. It is interesting because despite PID controller having a better performance, the Yaw LNNA resulted in less range miss distance. Nevertheless, the performance of both controllers is very similar.



**Yaw Response for Act 20deg, Target X = 14000m, & Target Y = -3000m**
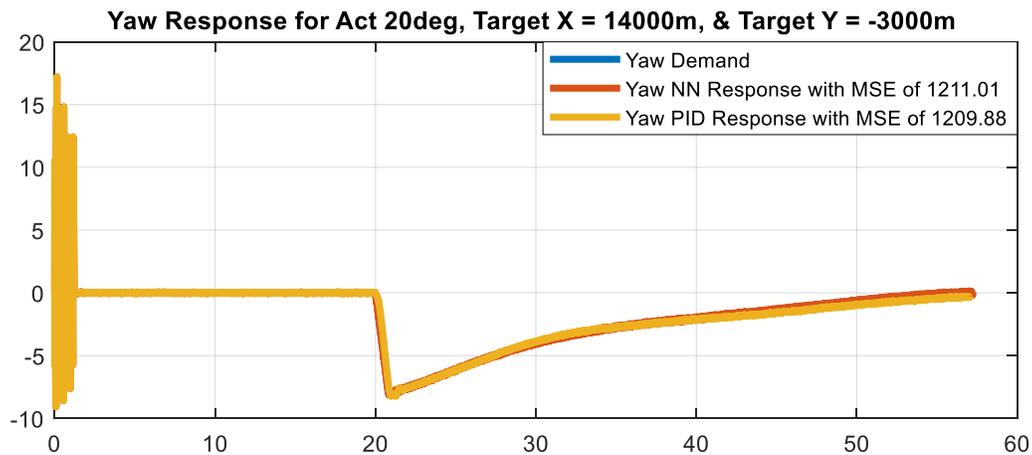
Figure 233 - Comparison of Yaw Response Between LNNA and PID for Max Actuator Deflection of 20deg and Target Located at Xt = 14000m and Yt = -3000m

Figure 234 shows the comparison of the histogram distribution between PID and NNA controllers for range and deviation miss distances with a maximum actuator deflection of 20deg. It can be seen that PID has a higher mean and a higher standard of deviation. This means that for both cases PID resulted in more error. This leads to the conclusion that NNA is a viable if not a better alternative to the PID controller.
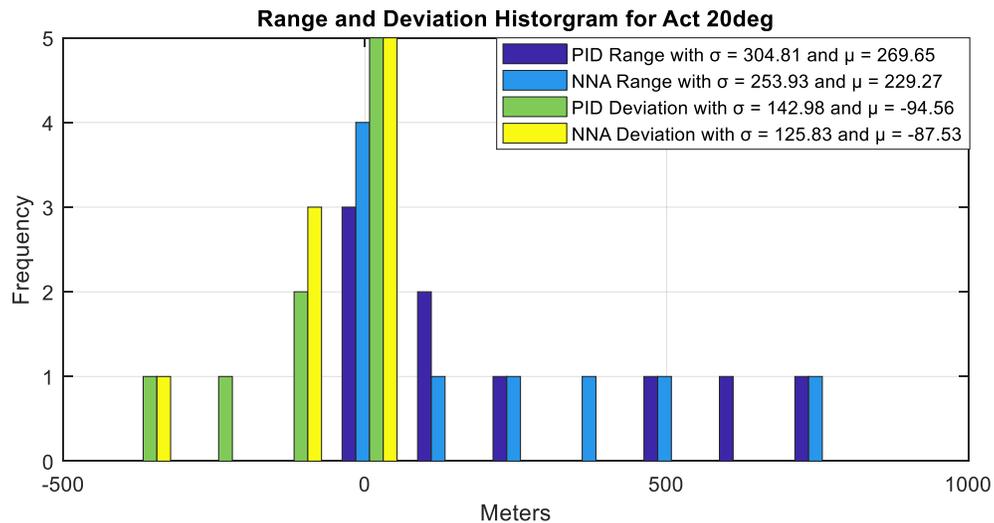


Figure 234 – Comparison of The Histogram Distribution Between PID and NNA Controllers for Range and Deviation Miss Distance

Figure *235* shows the comparison of the histogram distribution between the performances of the Roll PID and RNNA controllers. It can be seen that the RNNA controller has a lower mean. This means that the RNNA controller is more precise. On the other hand, the PID controller has a lower standard of deviation. This means that the PID controller is more accurate.
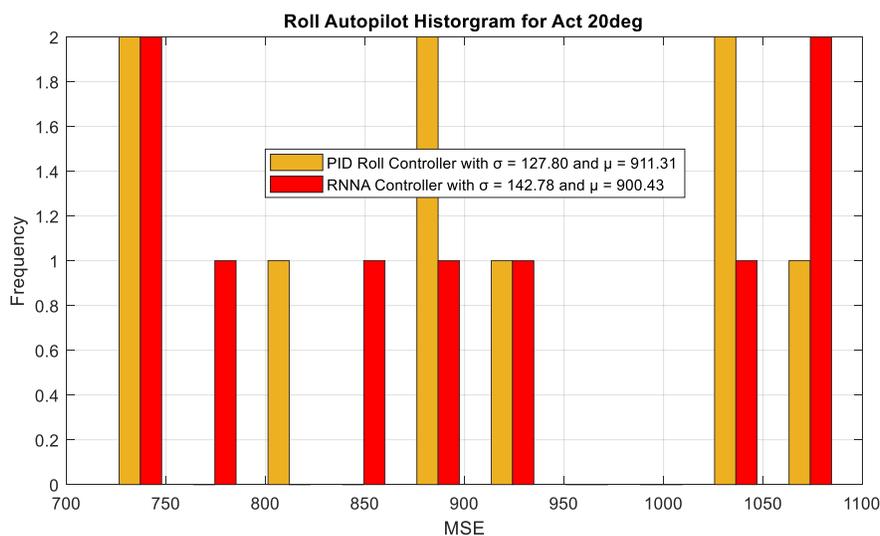


Figure 235 – Comparison of The Histogram Distribution Between The Performances of Roll PID and RNNA Controllers

Figure *236* shows the comparison of the histogram distribution between the performances of the Pitch PID and LNNA controllers. It can be seen that the PID controller has a higher mean and standard of deviation. This means that Pitch LNNA performed better overall.
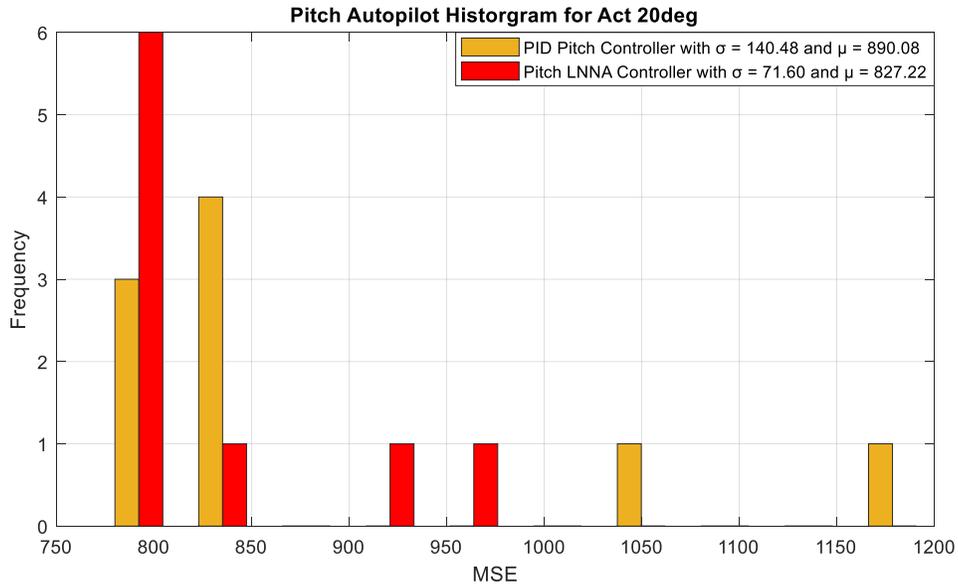


Figure 236 – Comparison of The Histogram Distribution Between The Performances of Pitch PID and LNNA Controllers

Figure *237* shows the comparison of the histogram distribution between the performances of the Yaw PID and LNNA controllers. It can be seen that the Yaw LNNA controller has a lower mean. This means that the Yaw LNNA controller is more precise. On the other hand, the PID controller has a lower standard of deviation. This means that the PID controller is more accurate.
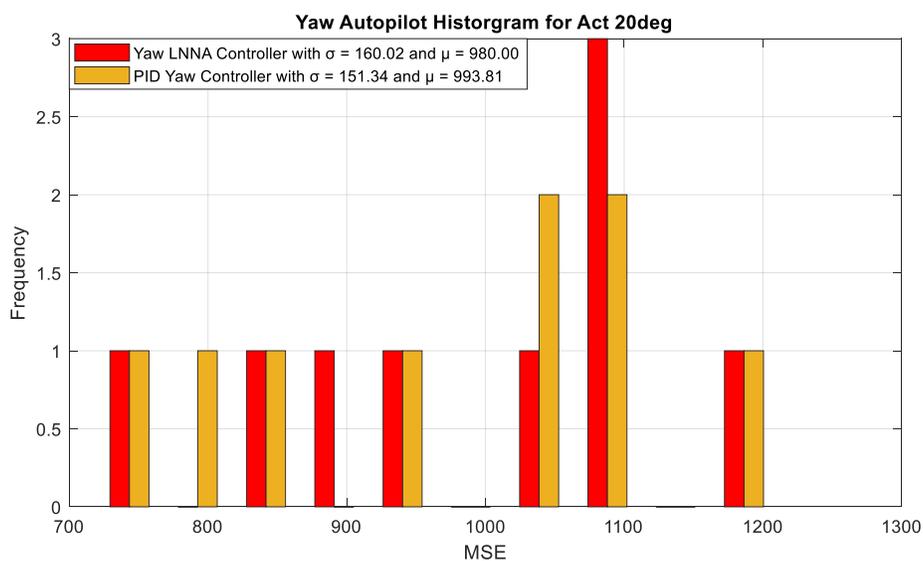


Figure 237 – Comparison of The Histogram Distribution Between The Performances of Yaw PID and LNNA Controllers

## 7.2.3. NNI vs GPS/INS Integration Comparison

The missile is launched at varying launch angles from 10deg to 80deg with a step of 10 deg. At each launch angle, the target is located at either 0m or 2000m. The reason for varying the launch angle and target location is to see the effects it has on the GPS/INS integration performance. It is important to note that the GPS/INS integration can only happen if the GPS signal is available. Since the simulation assumes a commercial GPS, the limitation has to be taken into consideration.

Therefore, Figure 238 shows the GPS/INS Comparison Scenario when the GPS signal is available. The GPS first becomes available once the missile is launched and its velocity drops below 500m/s. That is because commercial GPS is software limited to not report data faster than the 500m/s velocity limit. Since the GPS signal can be lost during the flight, it should be tested in the simulation. In addition, the reacquisition of the GPS signal is important. Hence, it is also tested in the simulation.
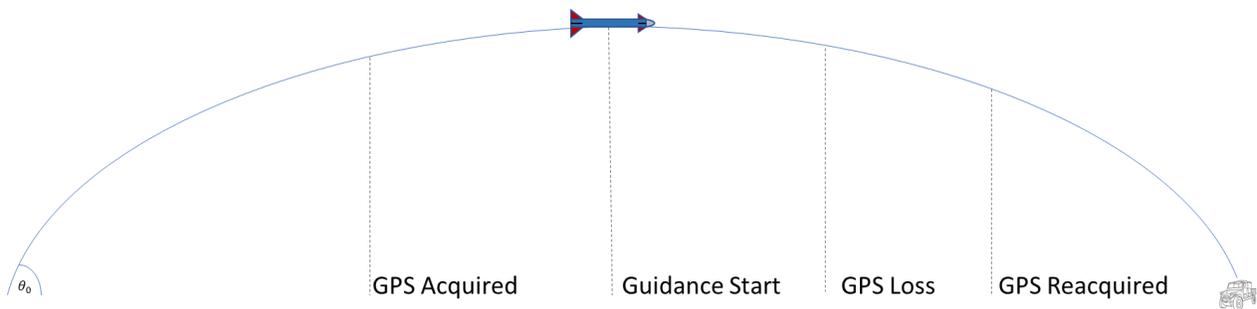


Figure 238 – GPS/INS Comparison Scenario

Furthermore, there are two methods of handling the loss of GPS signal. One method is to switch the INS algorithm with no reset. The No Reset method keeps the INS estimation as a backup localization of the missile. The Rest method will reset the INS with the predicted estimation. Therefore, the INS is corrected with the prediction although no backup localization is kept. The performance of the two methods will be presented separately in the comparison. That is because each method has its unique advantage. The performance of the No Reset method will assume

constant GPS signal. This is to ensure that error from not resetting does not overshadow the integration performance. However, for the Reset method, the performance will include the GPS loss of signal. That is because it is important to see the effects of resetting the INS on the integration algorithm performance. In addition, it is also the more popular method used in the missile industry. Moreover, the performance of the integration algorithm will be evaluated using Mean Square Error. This is to allow an overall quantifiable comparison criterion.

Figure 239 shows the no reset GPS/INS integration concept. Here, the blue line is the pure INS performance. In addition, the green line is the actual location of the missile. Moreover, the red line is the GPS/INS integration with no reset. It can be seen that the GPS/INS integration performance is better than INS. However, once the GPS signal is lost, the algorithm switches directly to the INS. This makes the performance gains useless if the GPS signal is lost. The algorithm switches back once the GPS signal is regained. On the other hand, Figure 240 shows the reset GPS/INS integration concept. Similarly, green line is the actual location of the missile. However, there is no pure INS blue line because it is being reset at every iteration. The red line is the GPS/INS integration with reset. It can be seen that once the GPS signal is lost, the algorithm switches to the cyan pure INS performance. Then, once the GPS is regained, it continues with the algorithm from where it left off.
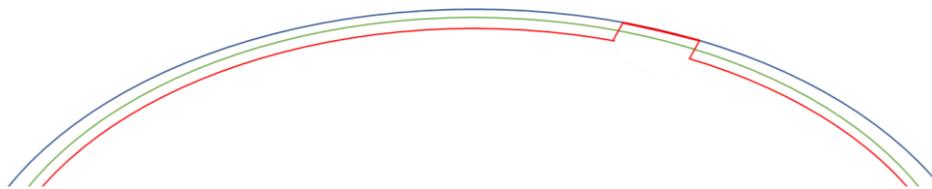


Figure 239 - GPS/INS Integration Concept of Operation with No INS Reset
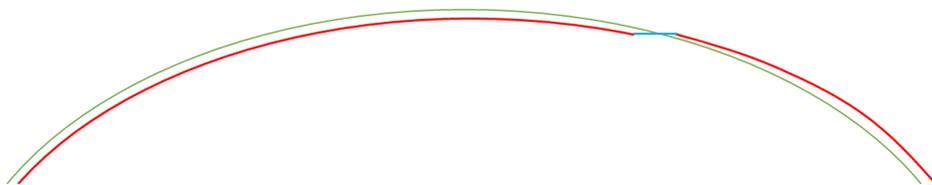


Figure 240 - GPS/INS Integration Concept of Operation with INS Reset

Figure 241 and Figure 242 show the comparison of down position GPS/INS integration in NED frame without reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the Kalman filter performed fine as well. However, it can be seen that the INS struggled with a high MSE. It is important to note that the deviation did not affect the performance much.
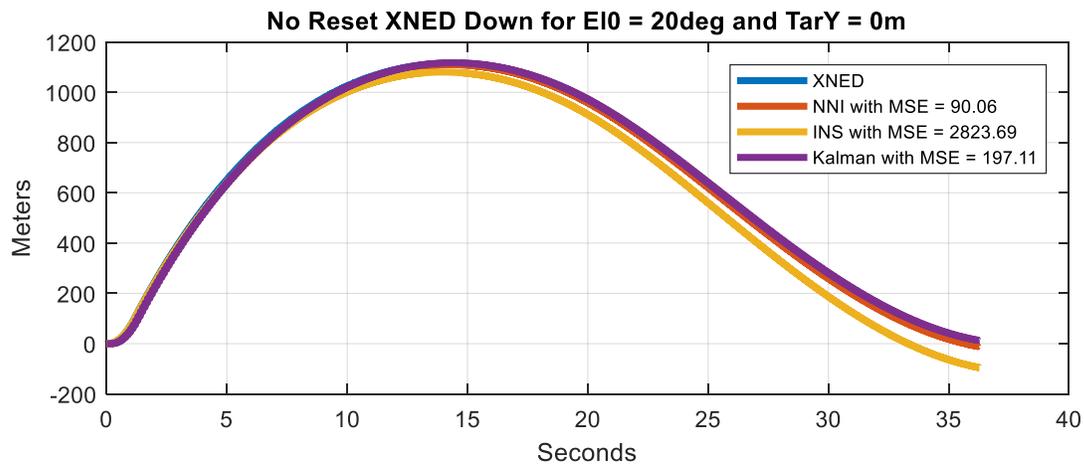


Figure 241 – Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20deg and Target Y = 0m
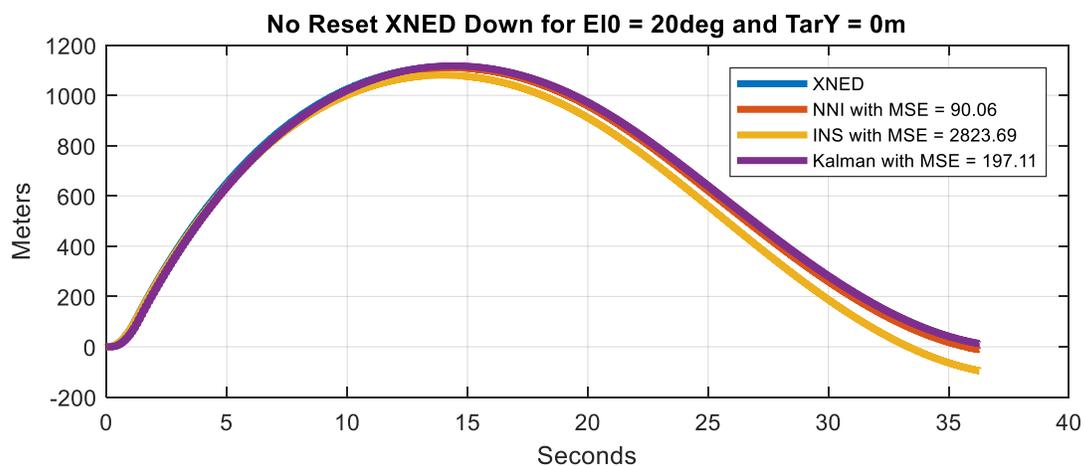


Figure 242 – Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20deg and Target Y = 2000m

Figure 243 and Figure 244 show the comparison of down position GPS/INS integration in NED frame without reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the Kalman filter performed fine as well. However, it can be seen that the INS struggled with a high MSE. It is important to note that the deviation did not affect the performance much.
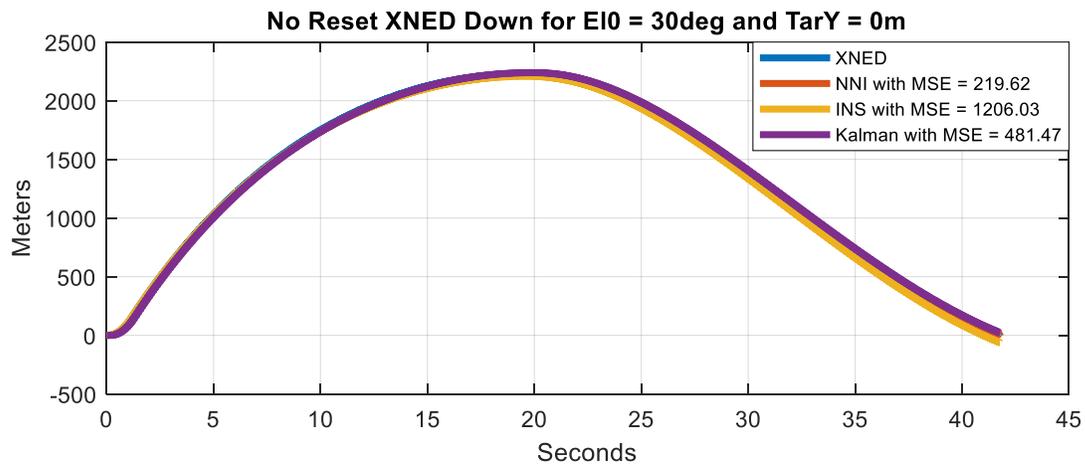


Figure 243 - Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30deg and Target Y = 0m
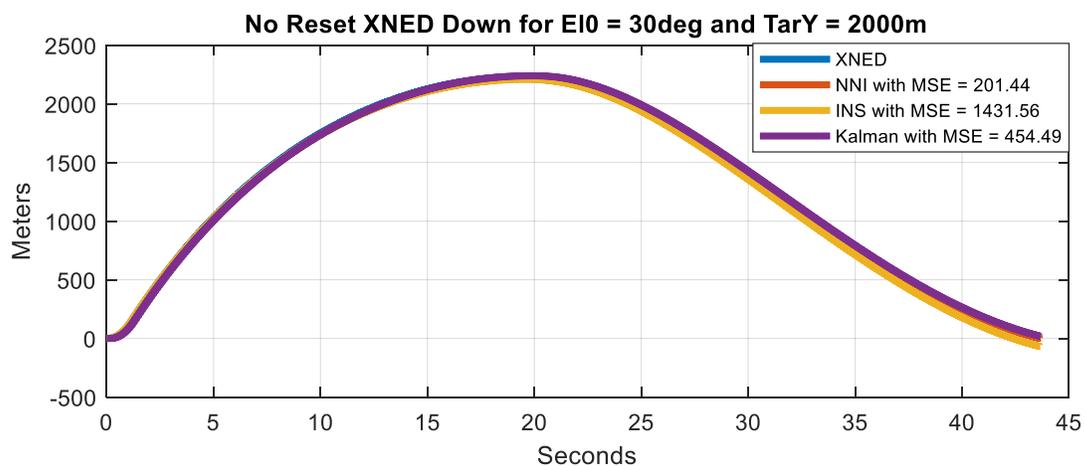


Figure 244 - Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30deg and Target Y = 2000m

Figure 245 and Figure 246 show the comparison of down position GPS/INS integration in NED frame without reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the INS performed the best for both cases with the least and small MSE. On the other hand, although NNI performed fine and better than the Kalman filter, the GPS error worsens the navigation performance. Hence, pure INS was a better choice for this particular case. It is important to note that the deviation did not affect the performance much.
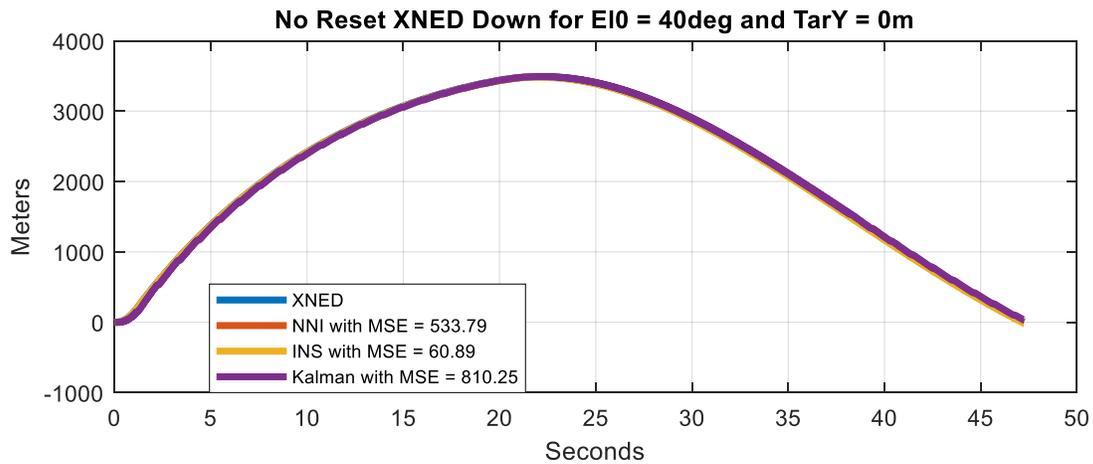


Figure 245 – Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40deg and Target Y = 0m
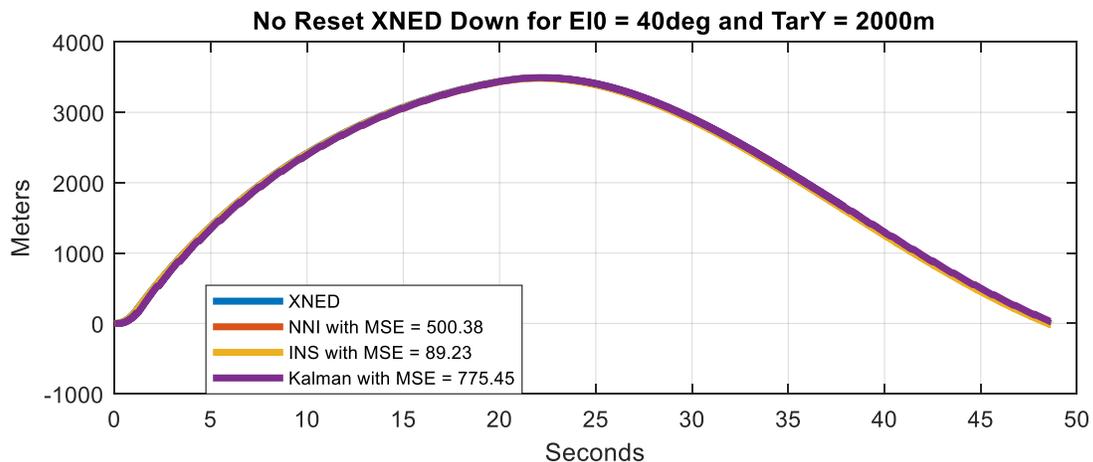


Figure 246 – Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40deg and Target Y = 2000m

Figure 247 and Figure 248 show the comparison of down position GPS/INS integration in NED frame without reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the INS performed the best for both cases with the least MSE but also close to NNI performance. The NNI performed fine and better than the Kalman filter. However, the GPS error worsens the navigation performance. Hence, pure INS was a better choice but not much better. It is important to note that the deviation did not affect the performance much.
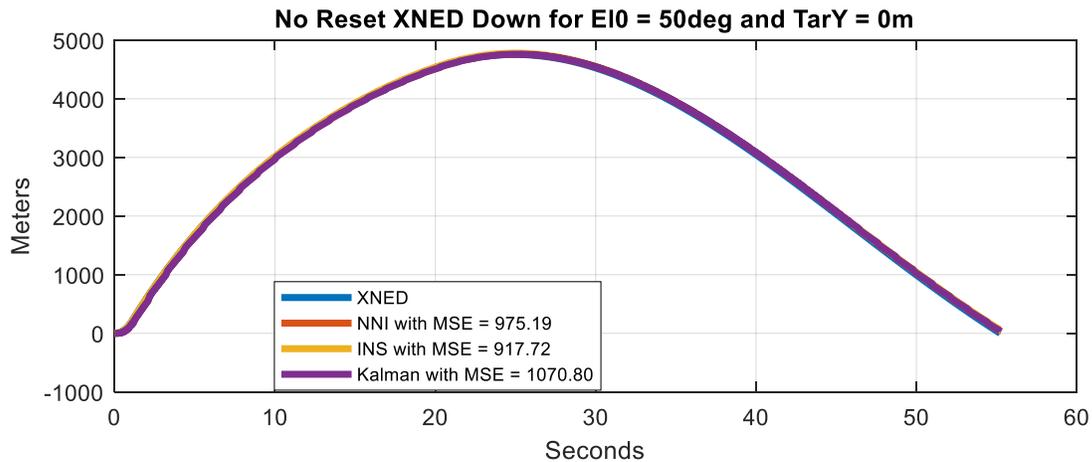


Figure 247 – Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50deg and Target Y = 0m
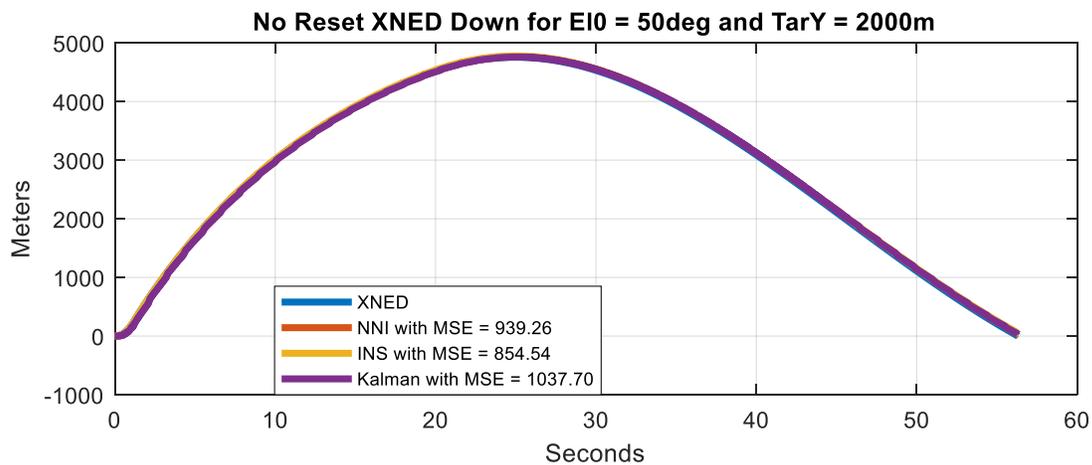


Figure 248 – Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50deg and Target Y = 2000m

Figure 249 and Figure 250 show the comparison of down position GPS/INS integration in NED frame without reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for both cases with the least MSE. In addition, the NNI performed fine as well. However, it can be seen that the INS struggled with a high MSE. It is important to note that the deviation did not affect the performance much.
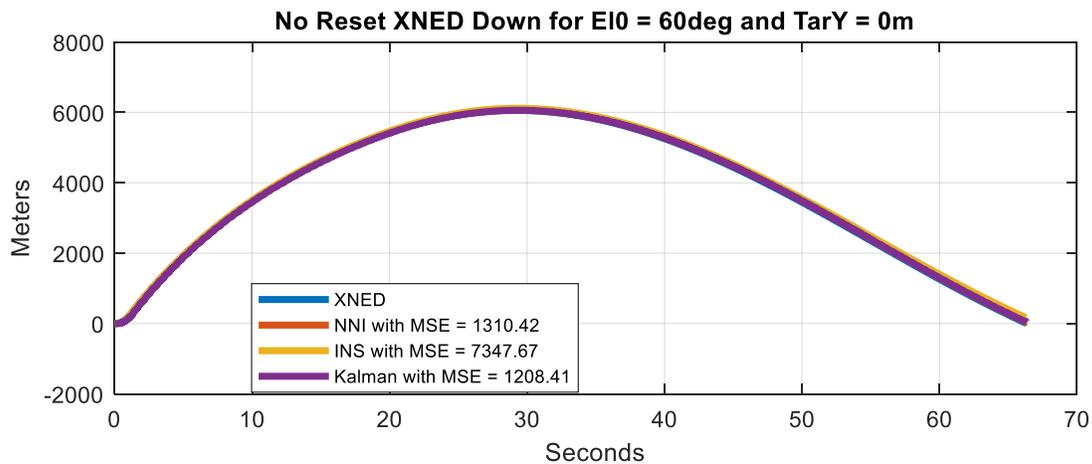


Figure 249 - Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 0m
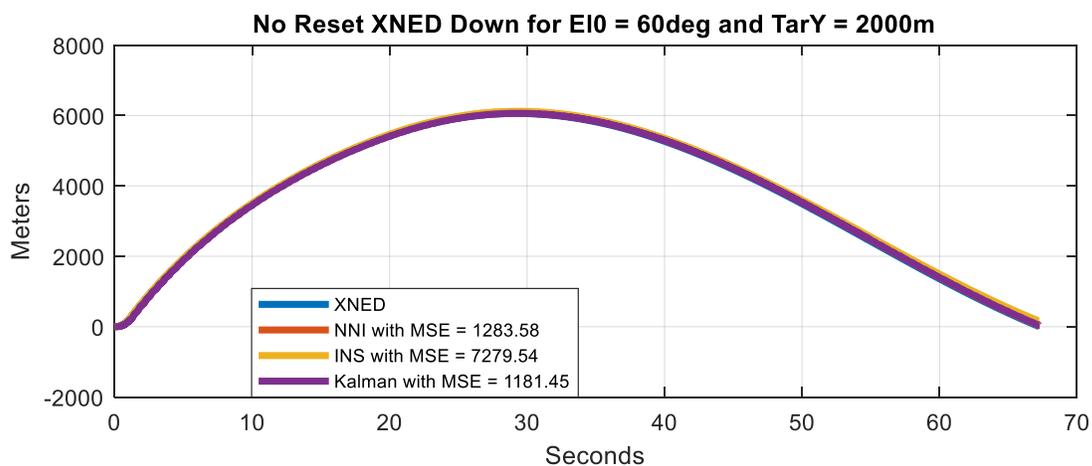


Figure 250 - Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 2000m

Figure 251 and Figure 252 show the comparison of down position GPS/INS integration in NED frame without reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE that is also twice better than the Kalman filter. However, the Kalman filter did perform fine. In addition, it can be seen that the INS struggled with a very high MSE. It is important to note that the deviation did not affect the performance much.
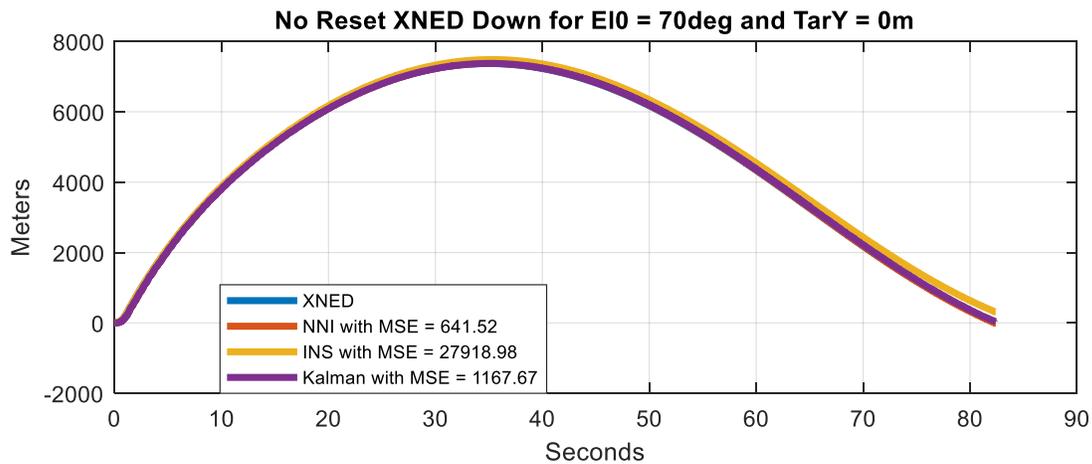


Figure 251 – Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 0m
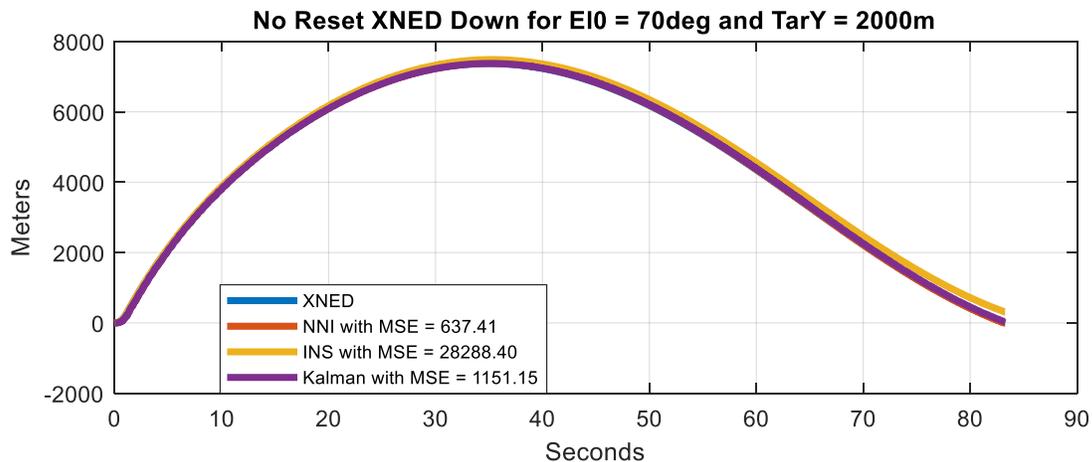


Figure 252 – Comparison of Down Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 2000m

Figure 253 and Figure 254 show the comparison of east position GPS/INS integration in NED frame without reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for no deviation. However, the NNI algorithm performed best for the case with deviation. That is because the NNI design struggles with zero input condition by introducing a bias. In addition, it can be seen that the INS struggled with a high MSE.
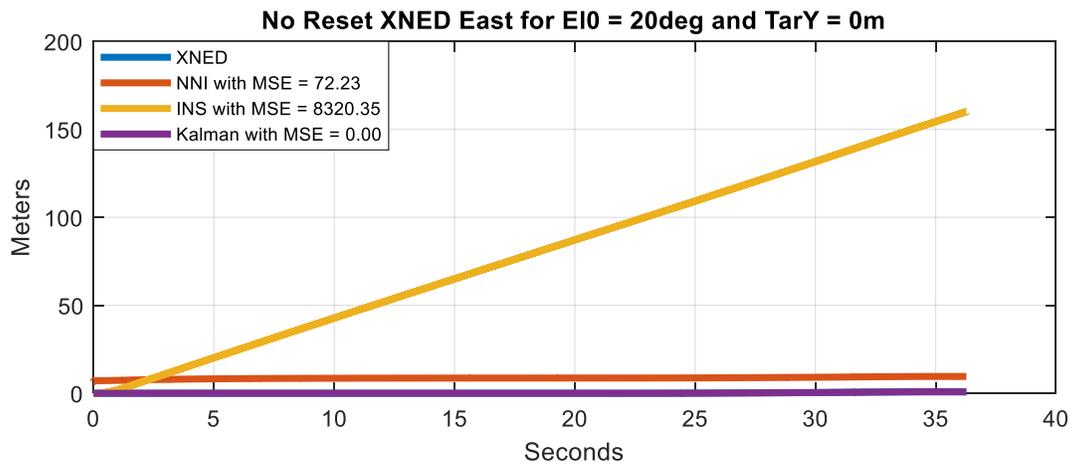


Figure 253 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 0m
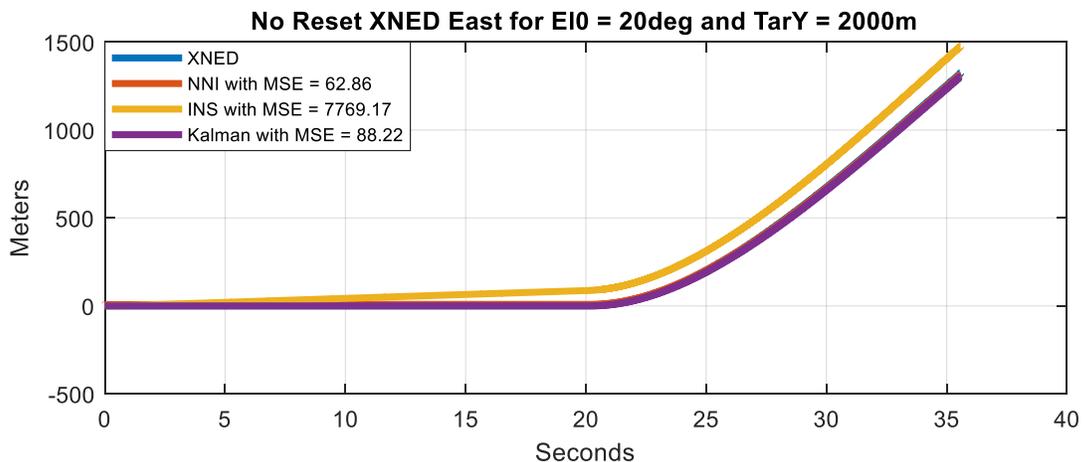


Figure 254 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 2000m

Figure 255 and Figure 256 show the comparison of east position GPS/INS integration in NED frame without reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for no deviation. However, the NNI algorithm performed best for the case with deviation. That is because the NNI design struggles with zero input condition by introducing a bias. In addition, it can be seen that the INS struggled with a very high MSE.

**No Reset XNED East for El0 = 30deg and TarY = 0m**
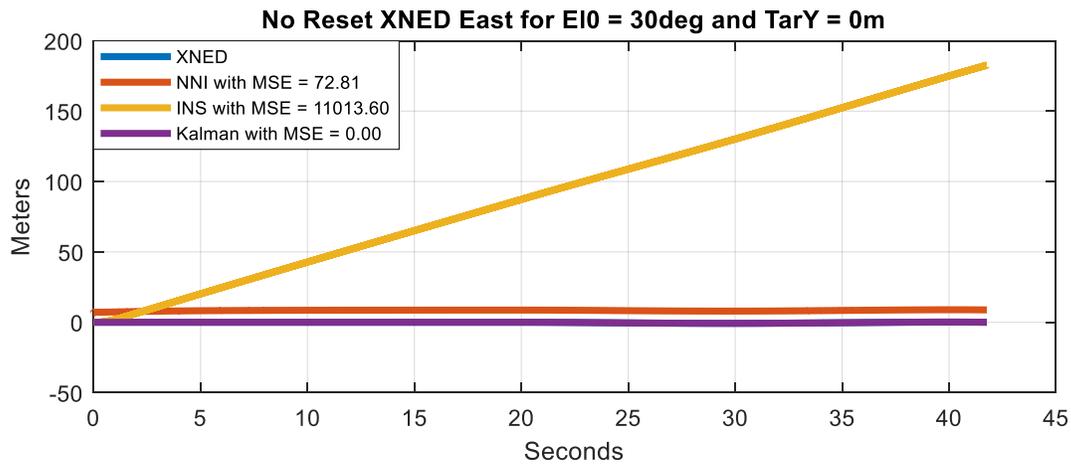


Figure 255 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 0m

**No Reset XNED East for El0 = 30deg and TarY = 2000m**
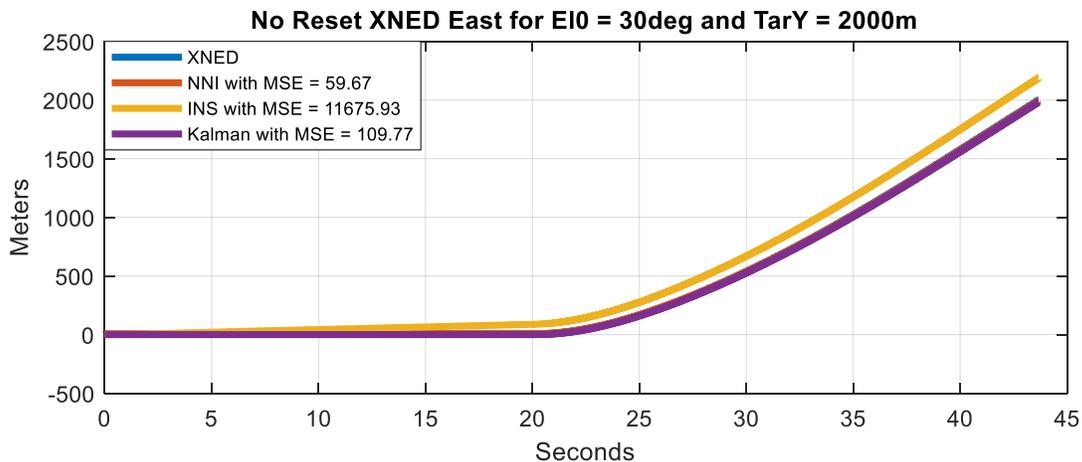


Figure 256 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 2000m

*Figure 257* and *Figure 258* show the comparison of east position GPS/INS integration in NED frame without reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for no deviation. However, the NNI algorithm performed best for the case with deviation. That is because the NNI design struggles with zero input condition by introducing a bias. In addition, it can be seen that the INS struggled with a very high MSE.
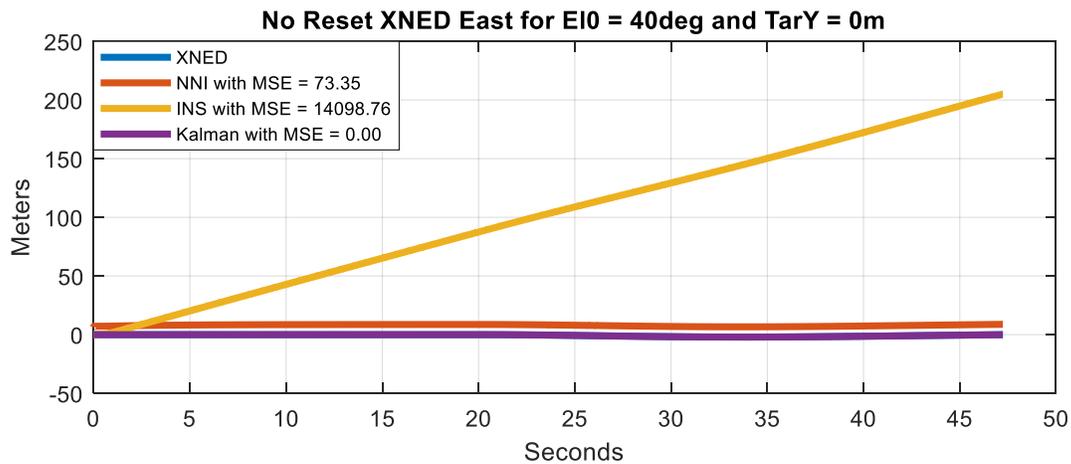


Figure 257 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 0m
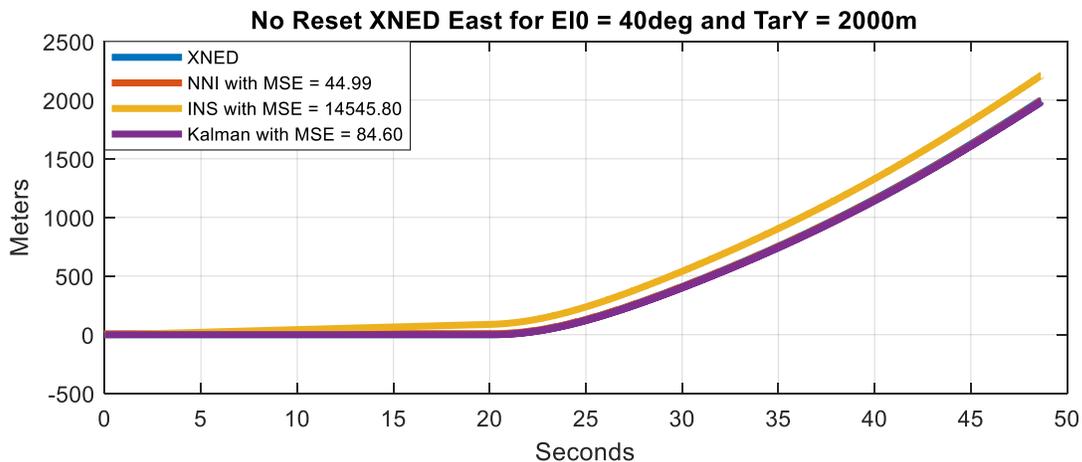


Figure 258 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 2000m

Figure 259 and Figure 260 show the comparison of east position GPS/INS integration in NED frame without reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for no deviation. However, the NNI algorithm performed best for the case with deviation. That is because the NNI design struggles with zero input condition by introducing a bias. In addition, it can be seen that the INS struggled with a very high MSE.
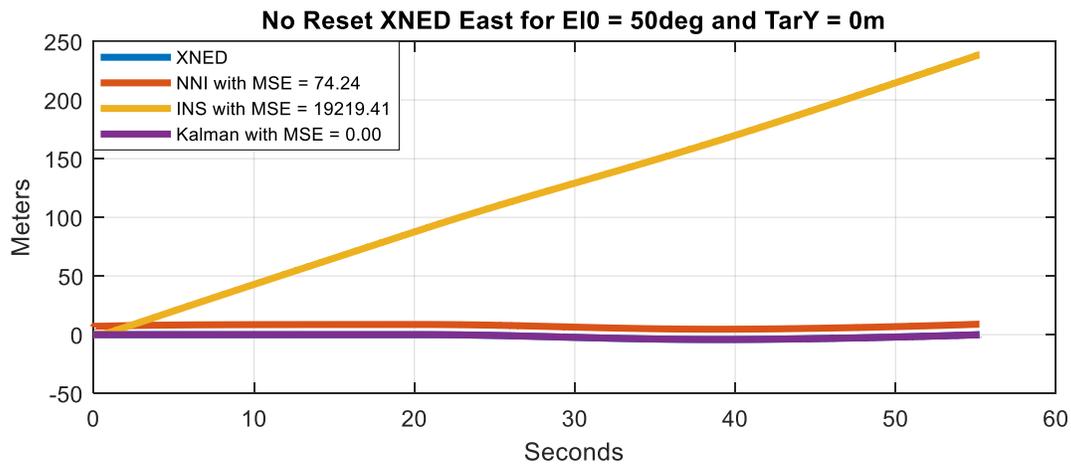


Figure 259 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 0m
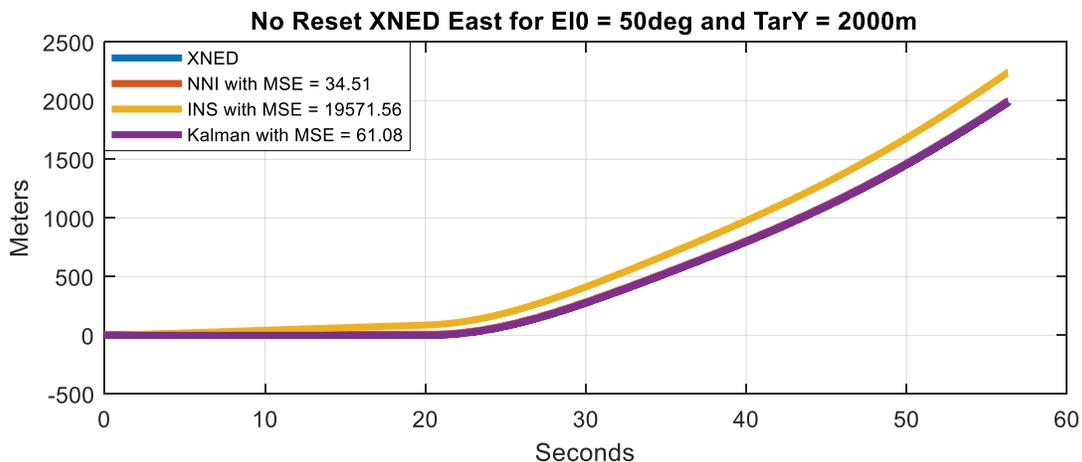


Figure 260 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 2000m

Figure 261 and Figure 262 show the comparison of east position GPS/INS integration in NED frame without reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for no deviation. However, the NNI algorithm performed best for the case with deviation. That is because the NNI design struggles with zero input condition by introducing a bias. In addition, it can be seen that the INS struggled with a very high MSE.
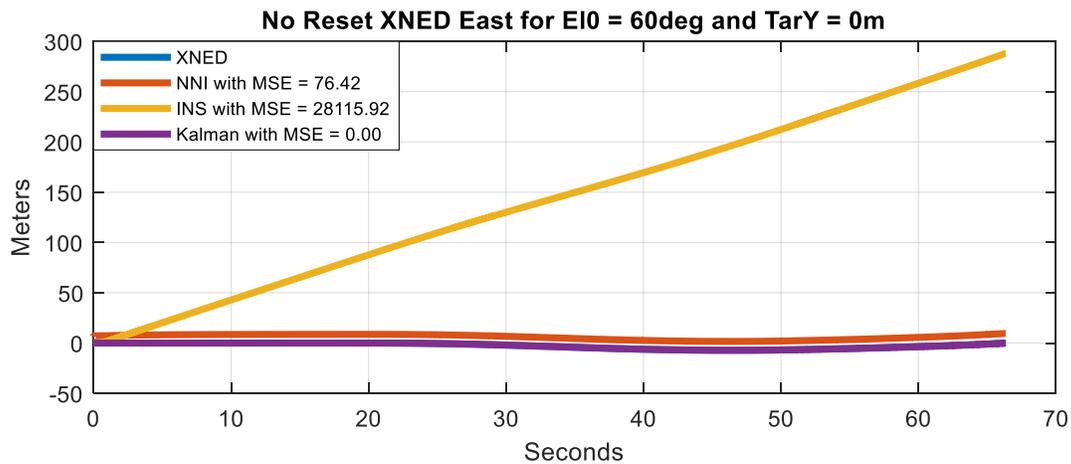


Figure 261 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 0m
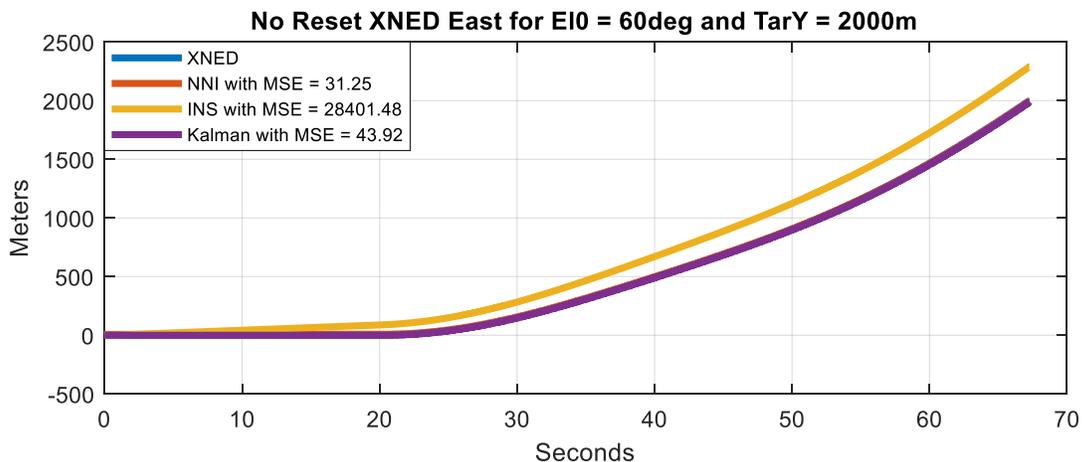


Figure 262 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 2000m

Figure 263 and Figure 264 show the comparison of east position GPS/INS integration in NED frame without reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed best for both cases. However, the NNI algorithm performed fine and close to Kalman filter for the deviation case. In addition, it still struggles with zero input with a bias. In addition, it can be seen that the INS struggled with a very high MSE.
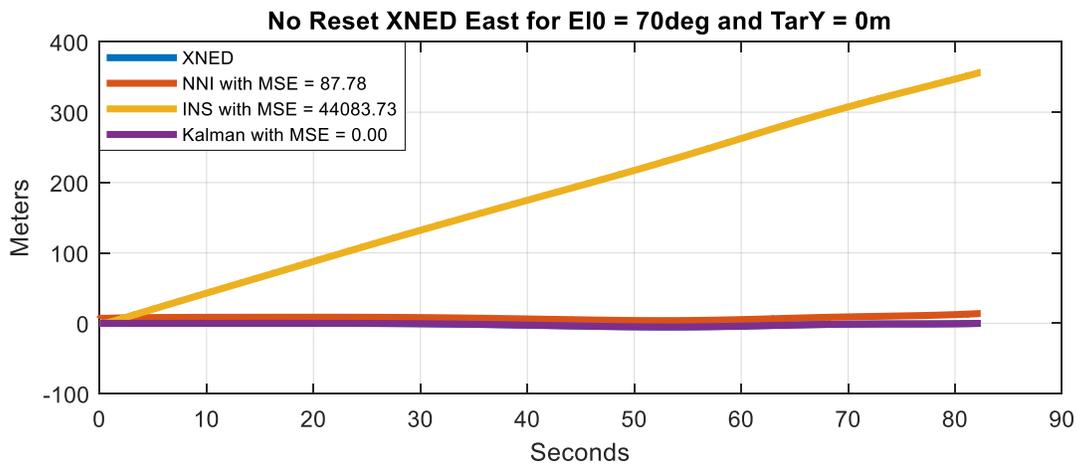


Figure 263 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 0m
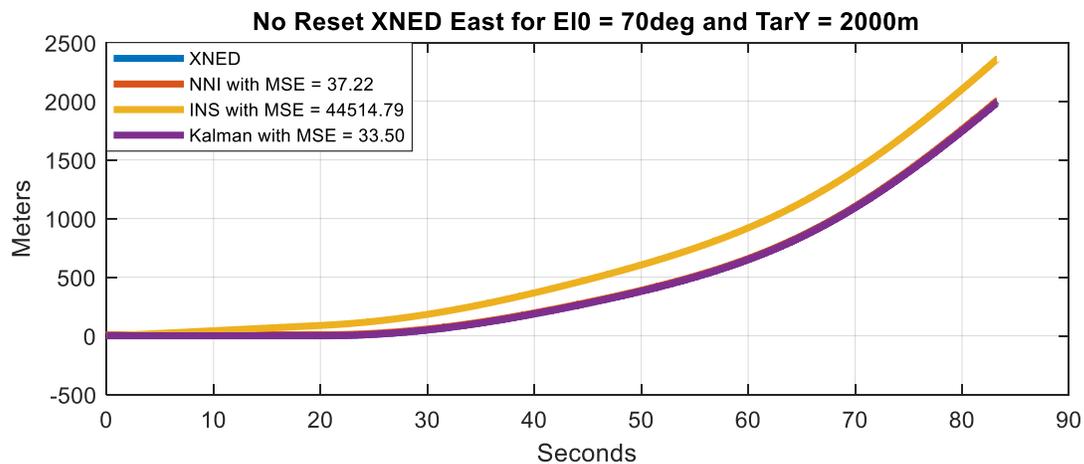


Figure 264 – Comparison of East Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 2000m

Figure *265* and Figure *266* show the comparison of north position GPS/INS integration in NED frame without reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. However, the Kalman filter performed fine. In addition, it can be seen that the INS struggled with a very high MSE. It is important to note that the deviation did not affect the performance much.
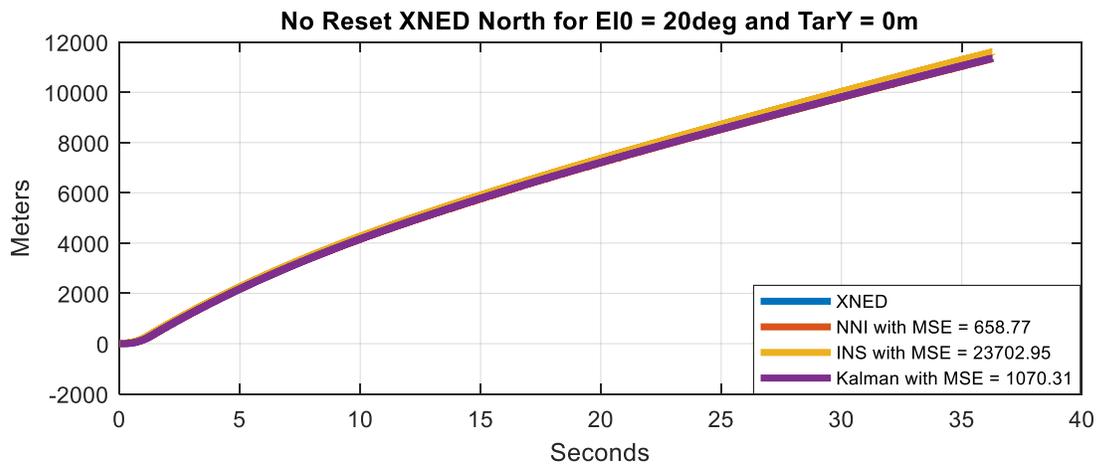


Figure 265 – Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 0m
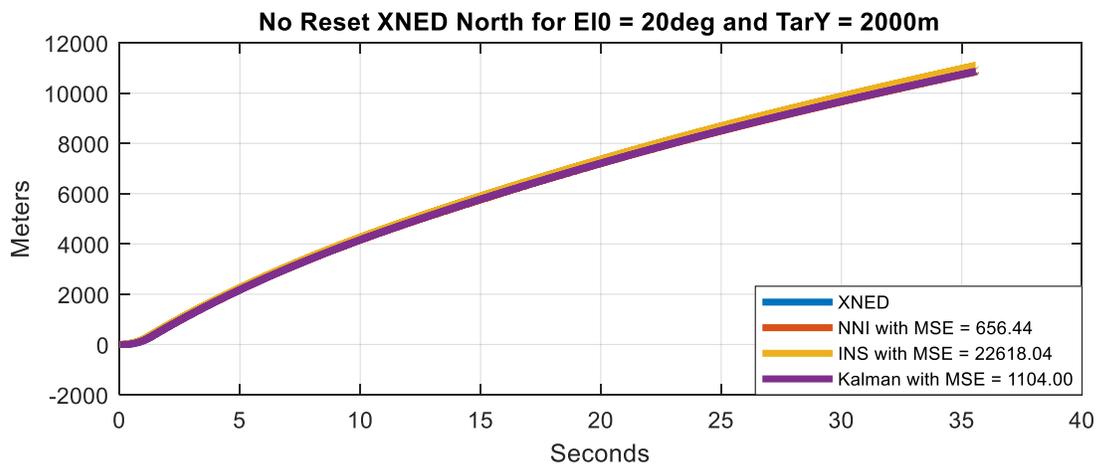


Figure 266 – Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 2000m

Figure *267* and Figure *268* show the comparison of north position GPS/INS integration in NED frame without reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. However, the Kalman filter performed fine. In addition, it can be seen that the INS struggled with a very high MSE. It is important to note that the deviation did not affect the performance much.
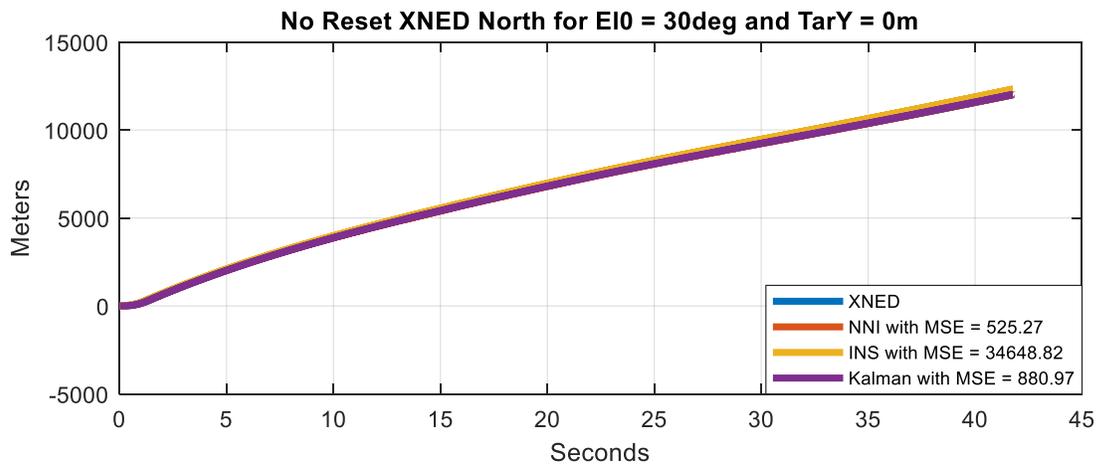


Figure 267 - Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 0m
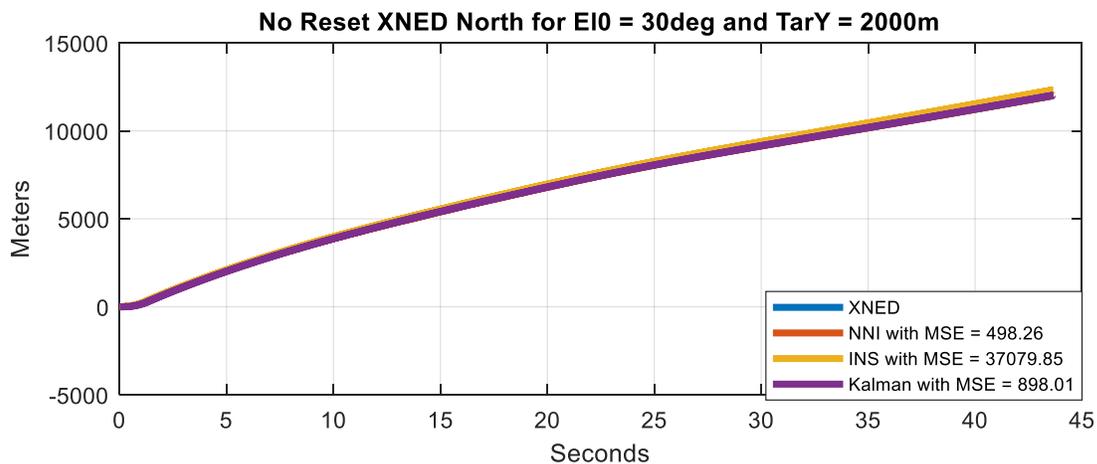


Figure 268 - Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 2000m

Figure *269* and Figure *270* show the comparison of north position GPS/INS integration in NED frame without reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. However, the Kalman filter performed fine. In addition, it can be seen that the INS struggled with a very high MSE. It is important to note that the deviation did not affect the performance much.
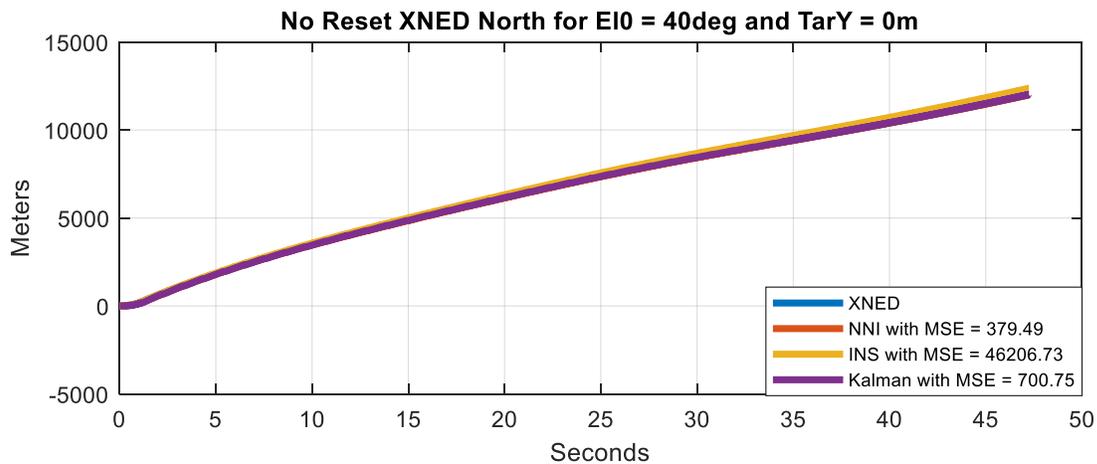


Figure 269 – Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 0m
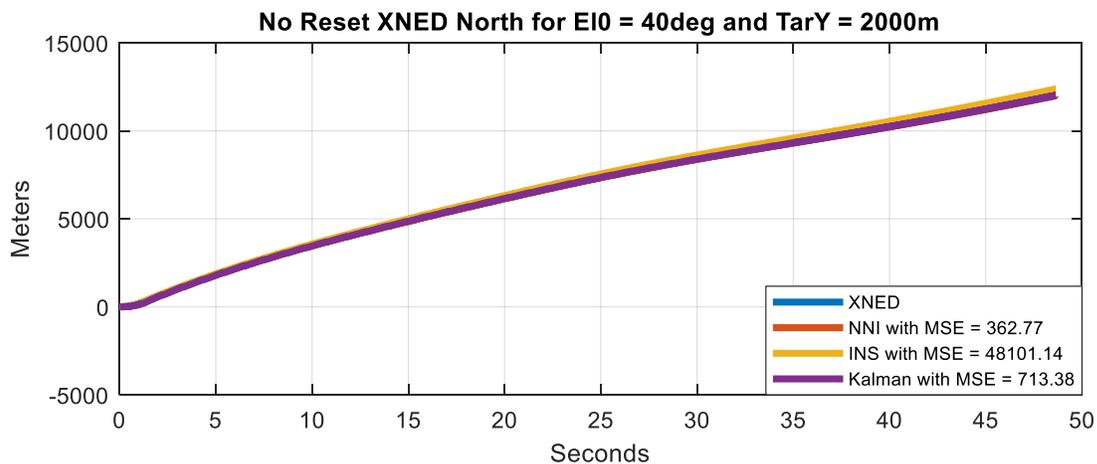


Figure 270 – Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 2000m

Figure *271* and Figure *272* show the comparison of north position GPS/INS integration in NED frame without reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. However, the Kalman filter performed fine. In addition, it can be seen that the INS struggled with a very high MSE. It is important to note that the deviation did not affect the performance much.
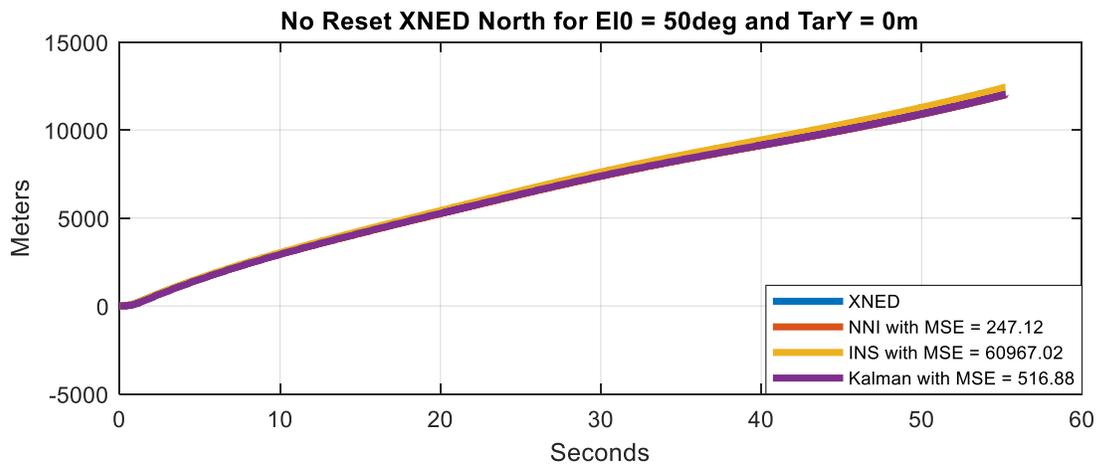


Figure 271 - Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 0m
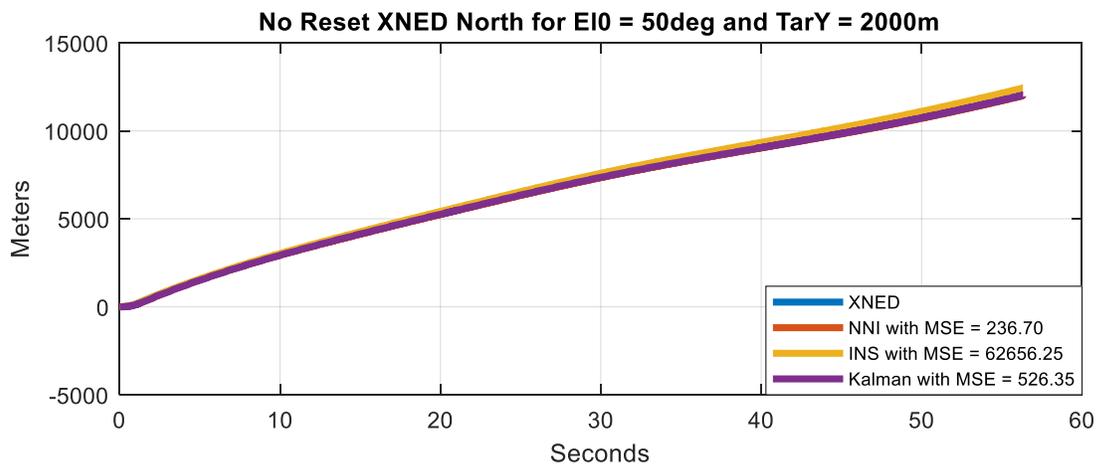


Figure 272 - Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 2000m

Figure *273* and Figure *274* show the comparison of north position GPS/INS integration in NED frame without reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. However, the Kalman filter performed fine. In addition, it can be seen that the INS struggled with a very high MSE. It is important to note that the deviation did not affect the performance much.
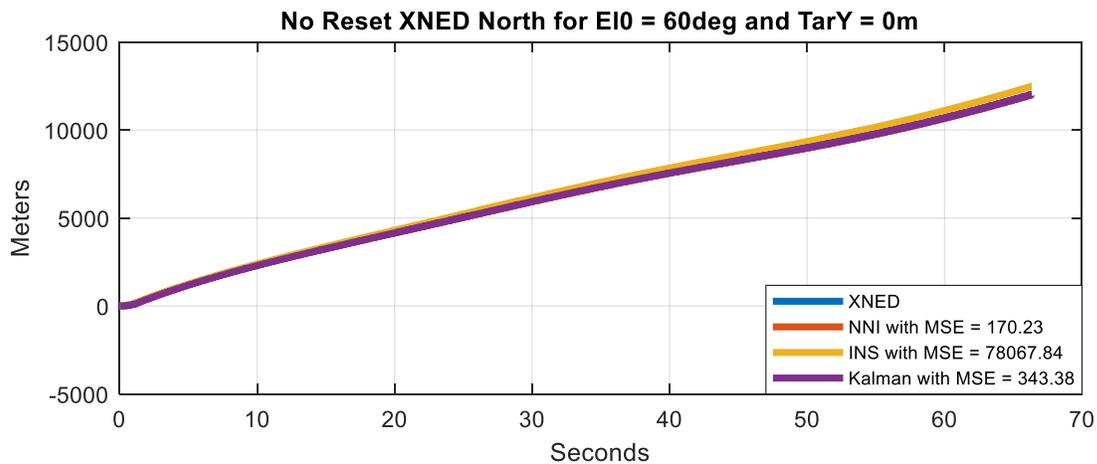


Figure 273 - Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 0m
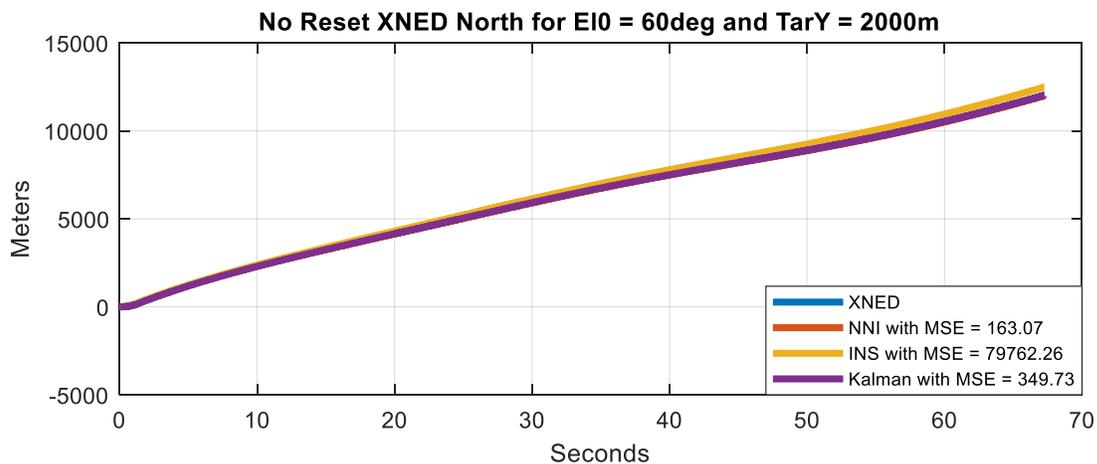


Figure 274 - Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 2000m

Figure *275* and Figure *276* show the comparison of north position GPS/INS integration in NED frame without reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for both cases with the least MSE. However, the NNI filter performed fine. In addition, it can be seen that the INS struggled with a very high MSE. It is important to note that the deviation did not affect the performance much.
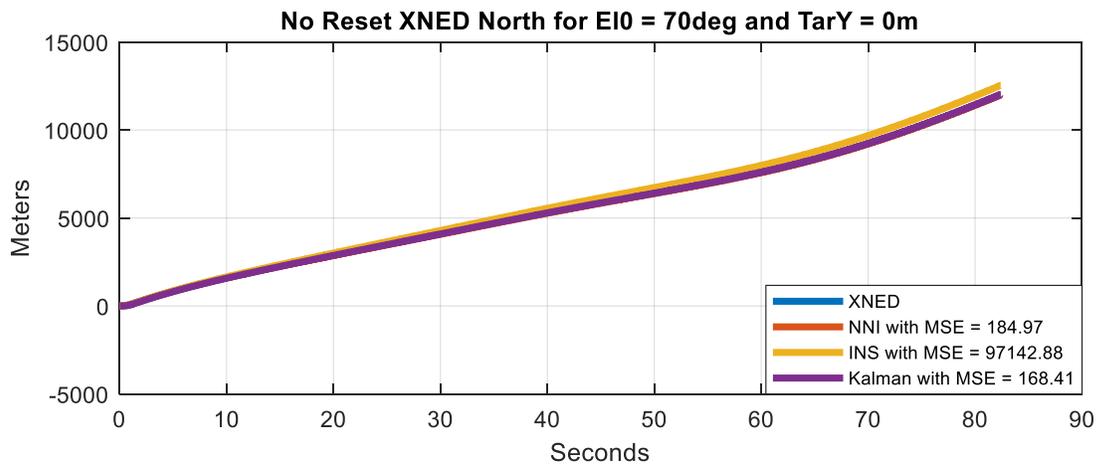


Figure 275 – Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 0m
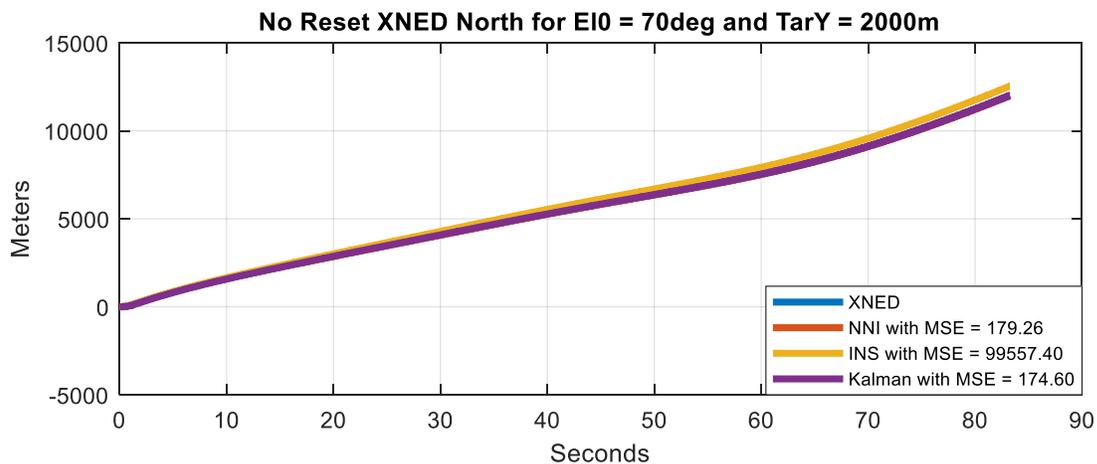


Figure 276 – Comparison of North Position GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 2000m

Figure *277* and Figure *278* show the comparison of down velocity GPS/INS integration in NED frame without reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the INS performed fine but with almost double the MSE. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
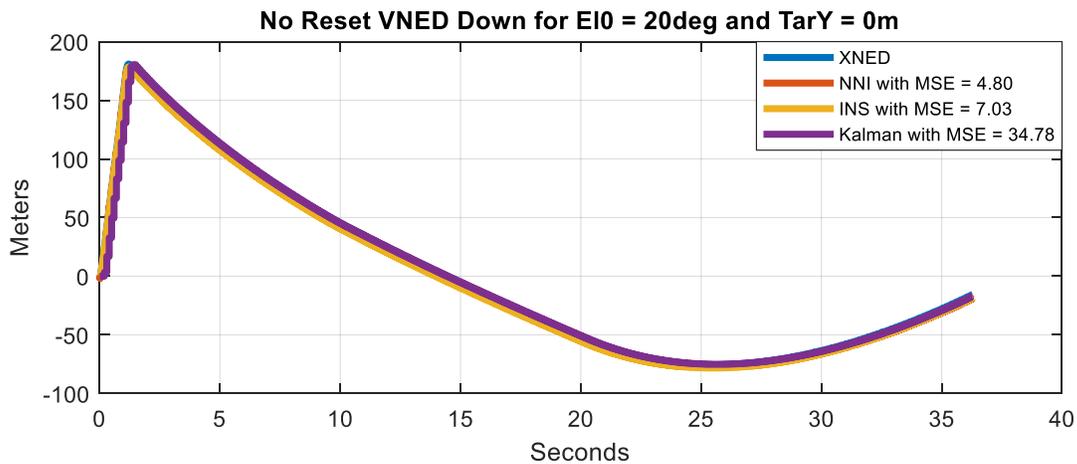


Figure 277 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 0m
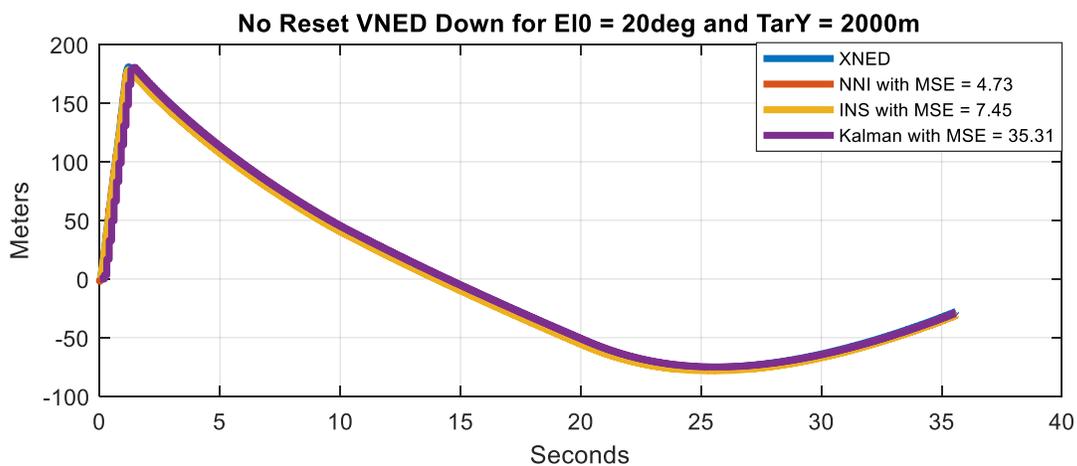


Figure 278 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 2000m

Figure *279* and Figure *280* show the comparison of down velocity GPS/INS integration in NED frame without reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the INS performed the best for both cases with the least MSE. The GPS error worsened the navigation performance. Nevertheless, the NNI performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
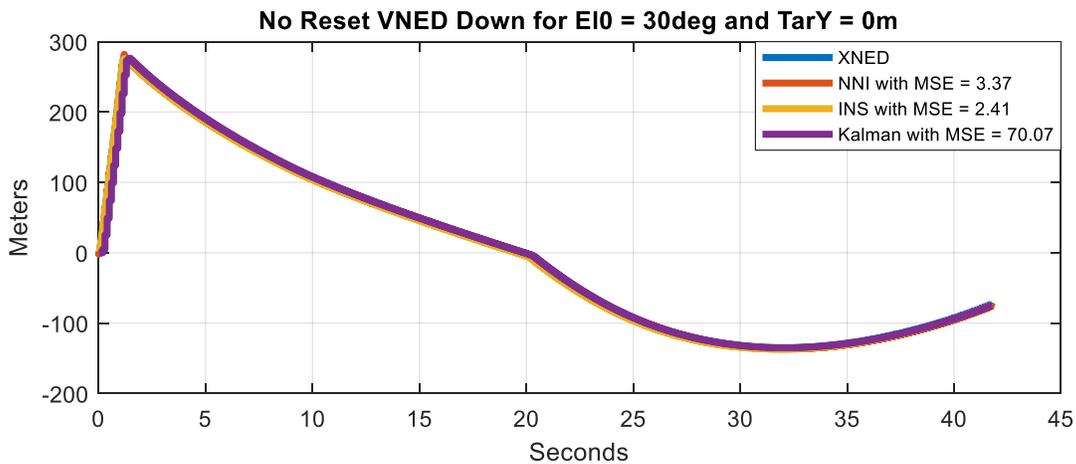


Figure 279 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 0m
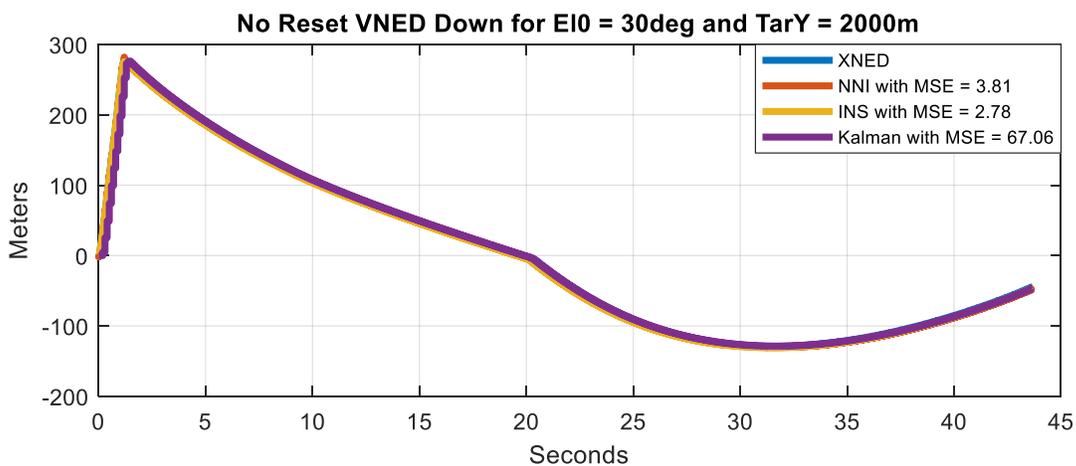


Figure 280 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 2000m

Figure *281* and Figure *282* show the comparison of down velocity GPS/INS integration in NED frame without reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the INS performed the best for both cases with the least MSE. The GPS error worsened the navigation performance. Nevertheless, the NNI performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
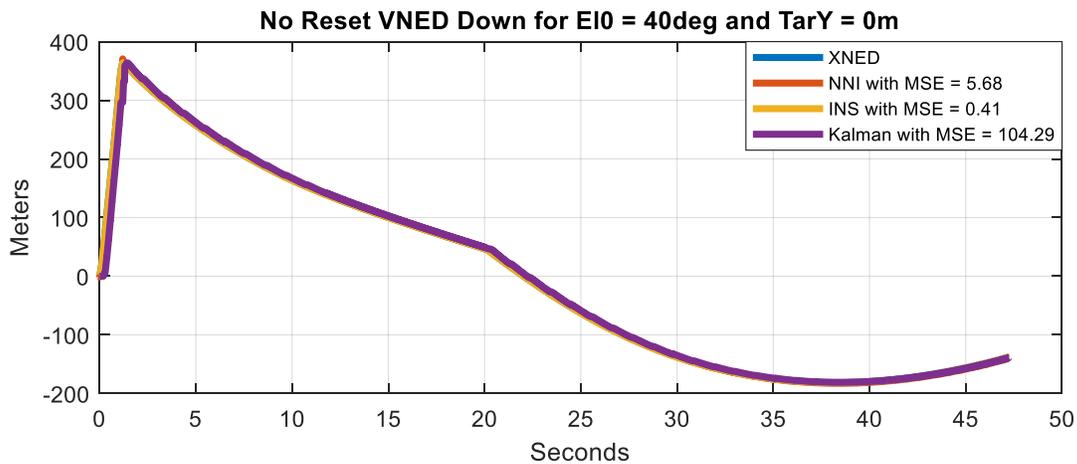


Figure 281 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 0m
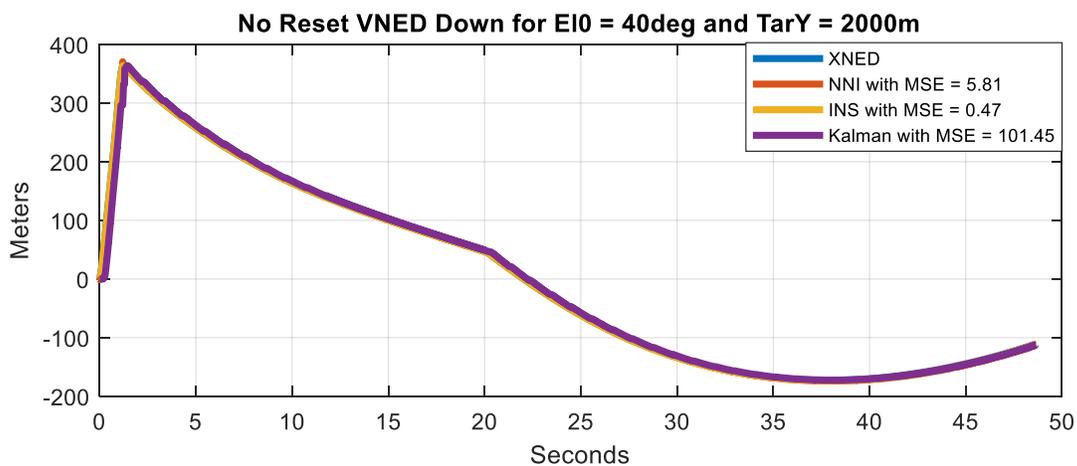


Figure 282 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 2000m

Figure *283* and Figure *284* show the comparison of down velocity GPS/INS integration in NED frame without reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the INS performed the best for both cases with the least MSE. The GPS error worsened the navigation performance. Nevertheless, the NNI performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
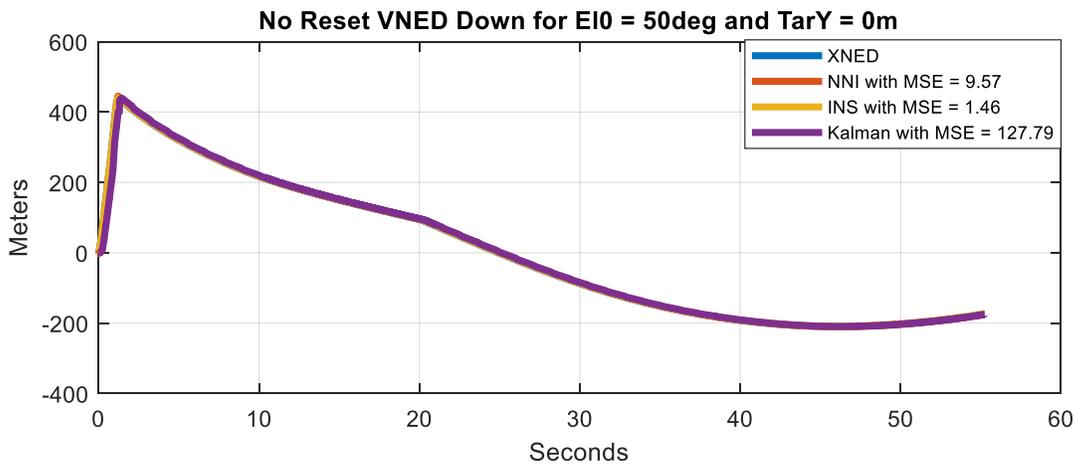


Figure 283 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 0m
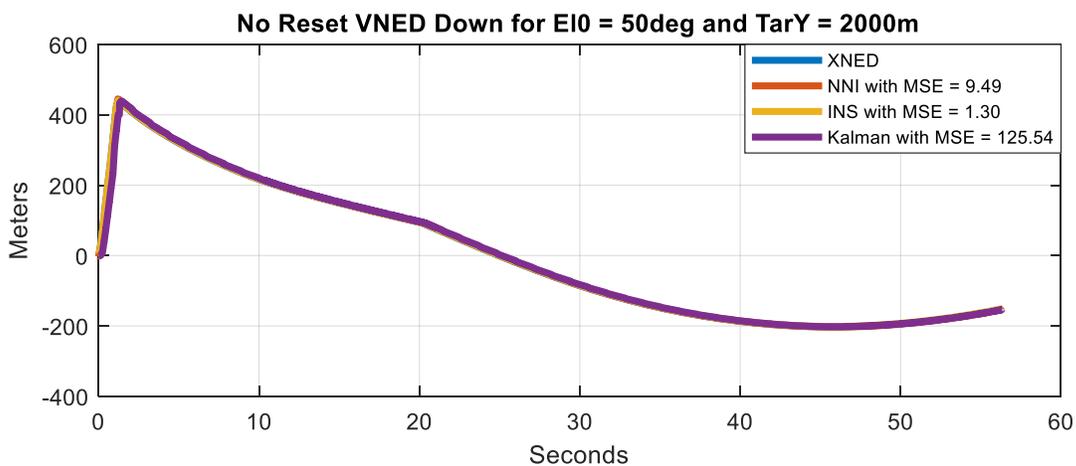


Figure 284 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 2000m

Figure *285* and Figure *286* show the comparison of down velocity GPS/INS integration in NED frame without reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the INS performed the best for both cases with the least MSE. The GPS error worsened the navigation performance. Nevertheless, the NNI performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
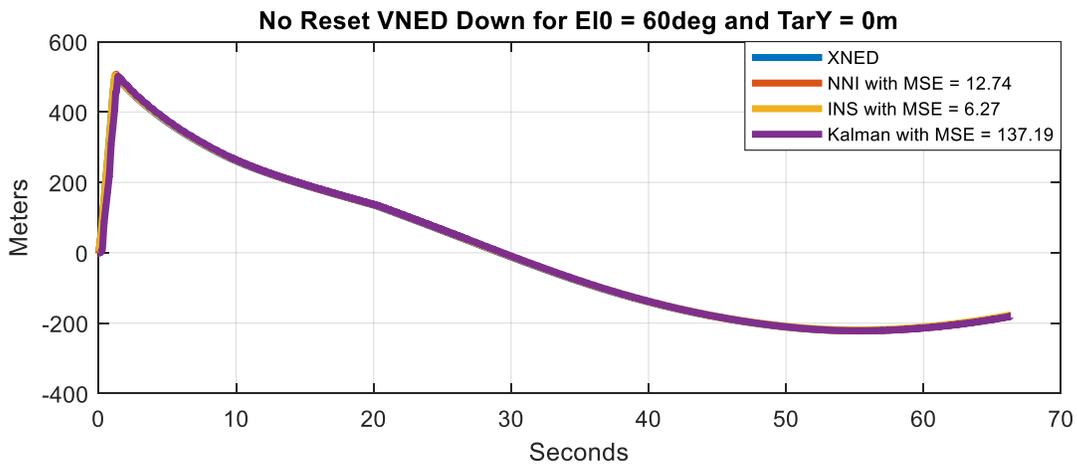


Figure 285 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 0m
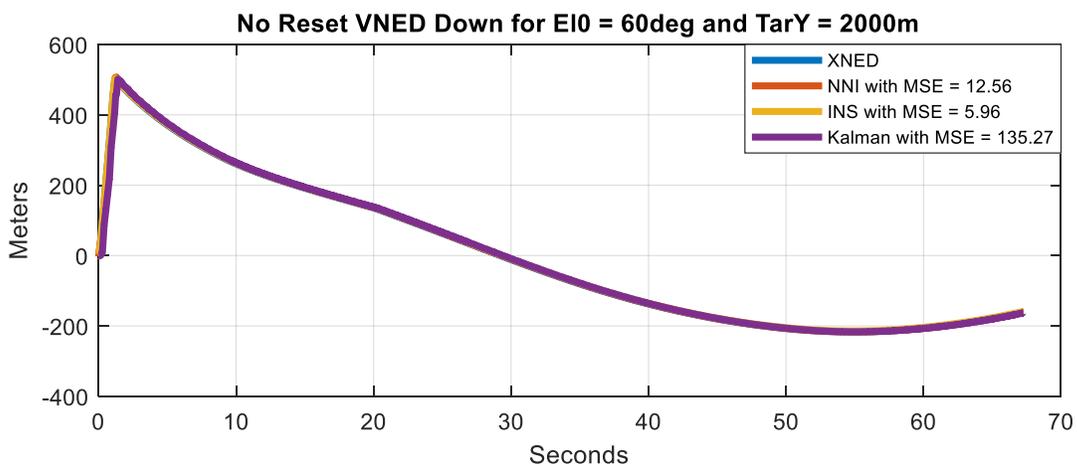


Figure 286 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 2000m

Figure *287* and Figure *288* show the comparison of down velocity GPS/INS integration in NED frame without reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI performed the best for both cases with the least MSE. Nevertheless, the INS performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
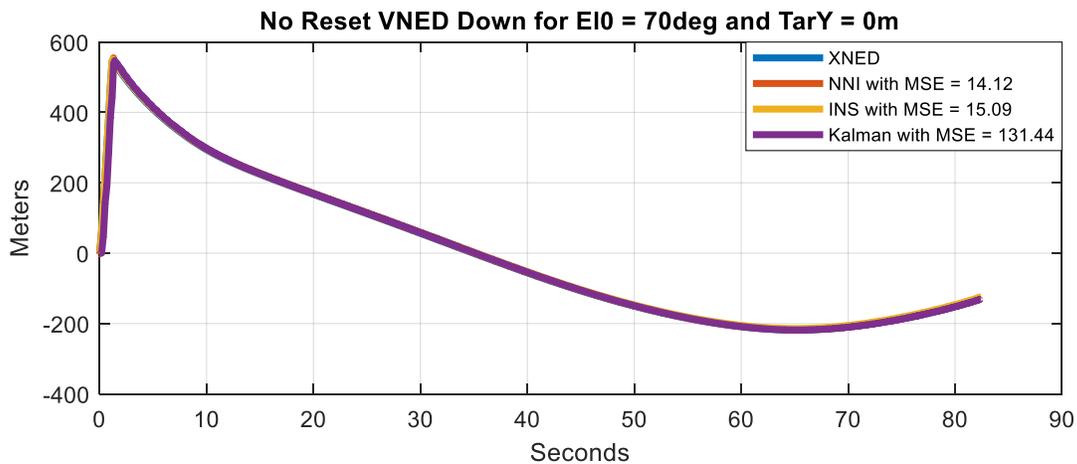


Figure 287 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 0m
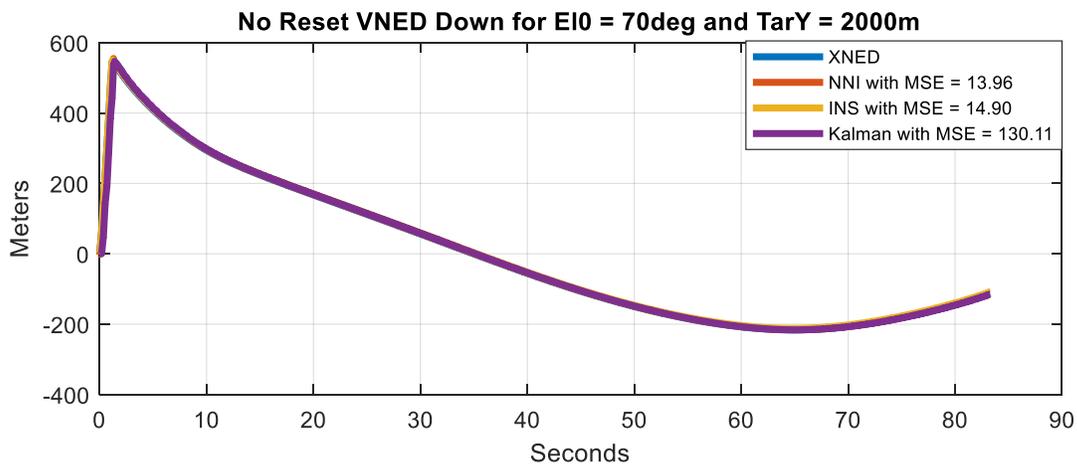


Figure 288 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 2000m

Figure *289* and Figure *290* show the comparison of east velocity GPS/INS integration in NED frame without reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter performed the best for the case with no deviation. However, the NNI performed best for the case with deviation. That is because the NNI algorithm struggles with zero input with a bias. In addition, the INS algorithm performed fine although with a higher MSE.
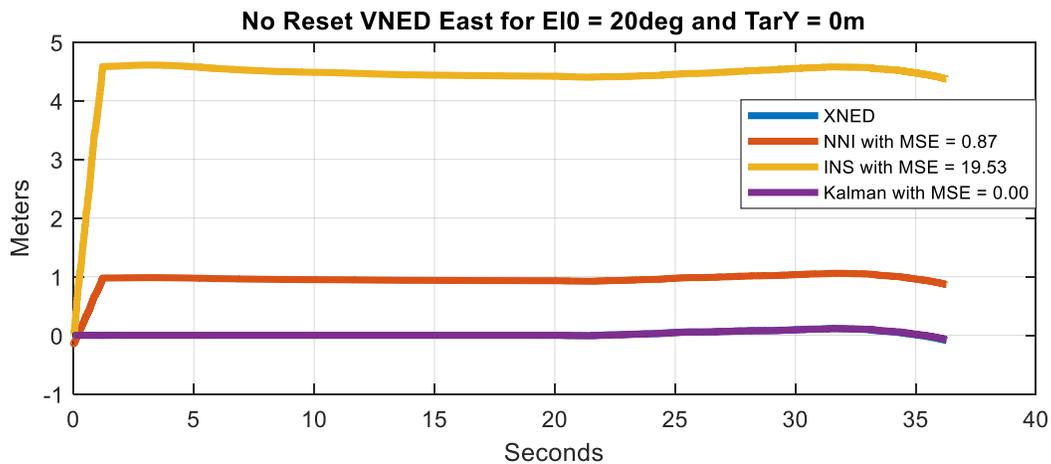


Figure 289 – Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 0m
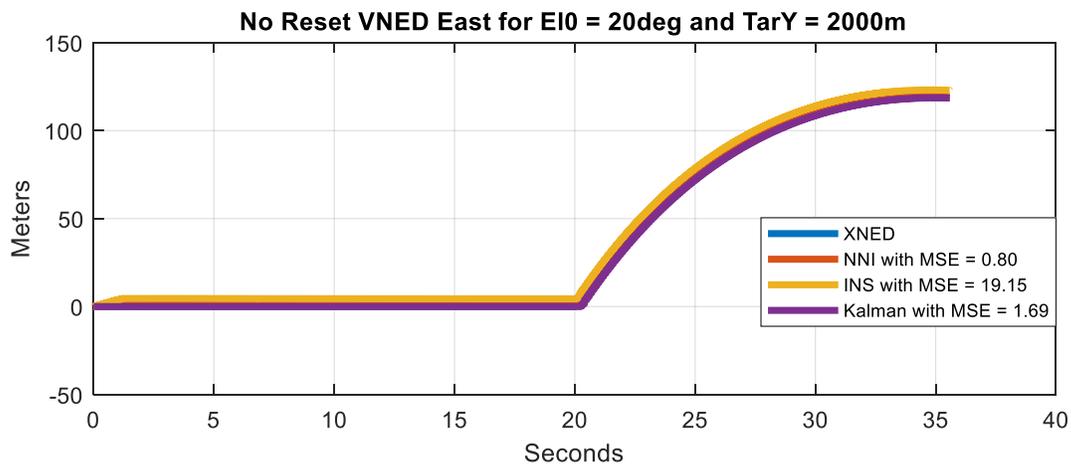


Figure 290 – Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 2000m

Figure *291* and Figure *292* show the comparison of east velocity GPS/INS integration in NED frame without reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter performed the best for the case with no deviation. However, the NNI performed best for the case with deviation. That is because the NNI algorithm struggles with zero input with a bias. In addition, the INS algorithm performed fine although with a higher MSE.
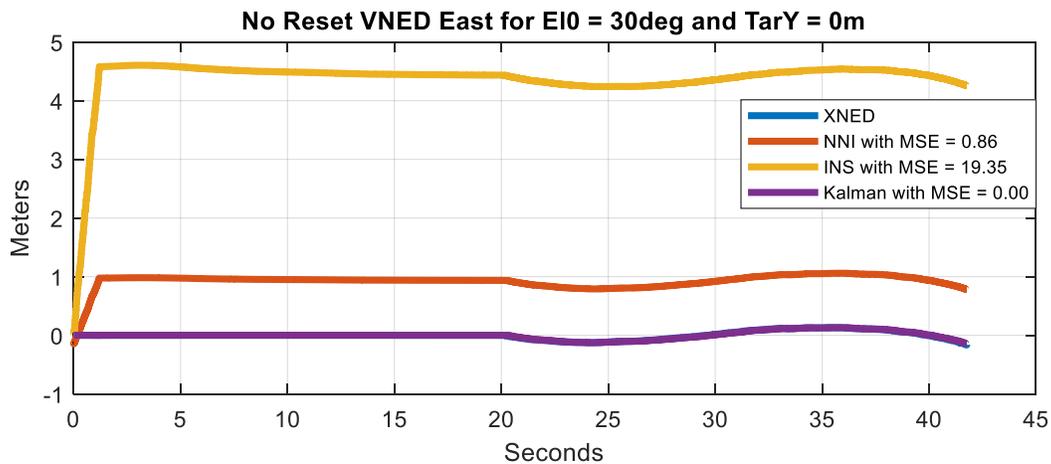


Figure 291 - Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 0m
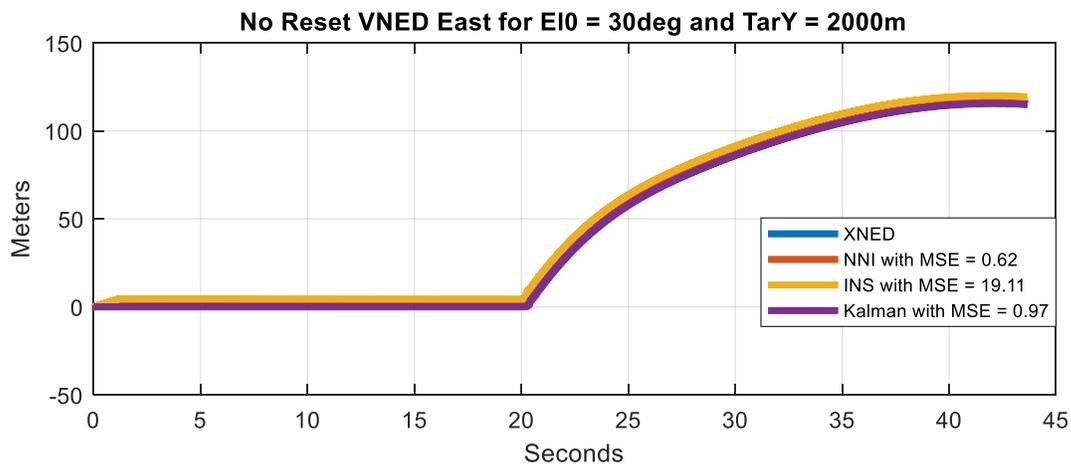


Figure 292 - Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 2000m

Figure *293* and Figure *294* show the comparison of east velocity GPS/INS integration in NED frame without reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter performed the best for the case with no deviation. However, the NNI performed best for the case with deviation by a little. That is because the NNI algorithm struggles with zero input with a bias. In addition, the INS algorithm performed fine although with a higher MSE.
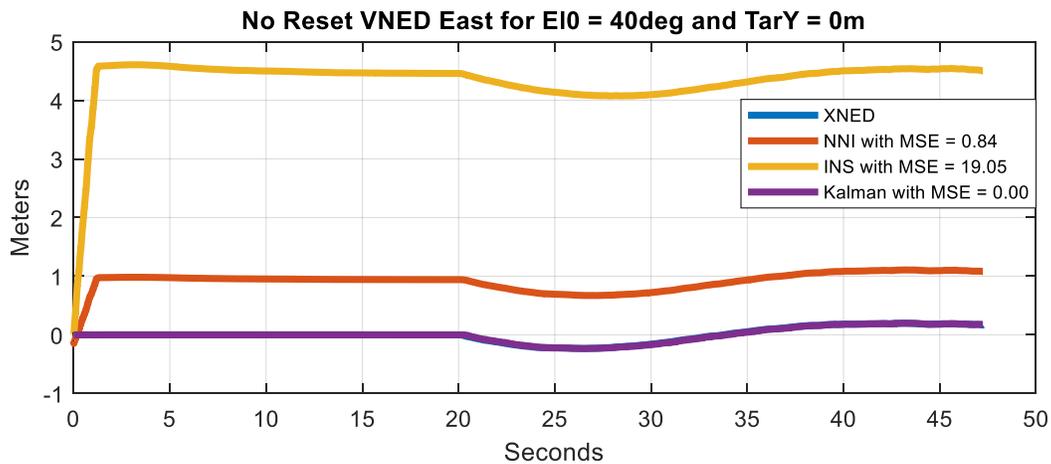


Figure 293 – Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 0m
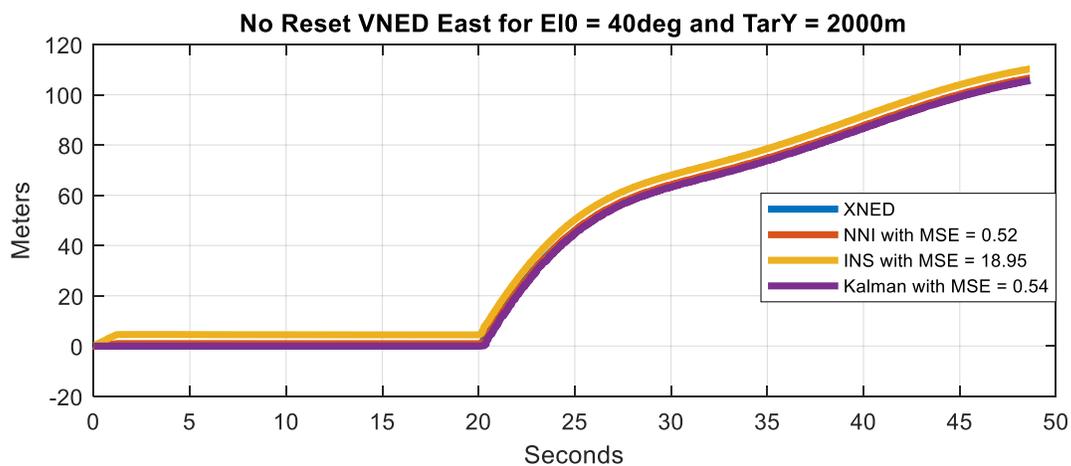


Figure 294 – Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 2000m

Figure *295* and Figure *296* show the comparison of east velocity GPS/INS integration in NED frame without reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter performed the best for both cases. However, the NNI performed fine with a bigger error with the deviation case. That is because the NNI algorithm struggles with zero input with a bias. In addition, the INS algorithm performed fine although with a higher MSE.
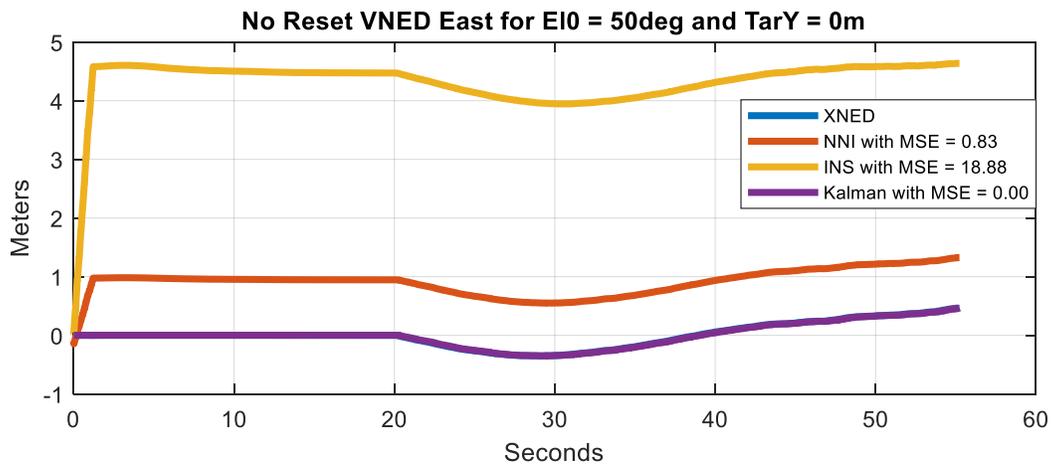


Figure 295 – Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 0m
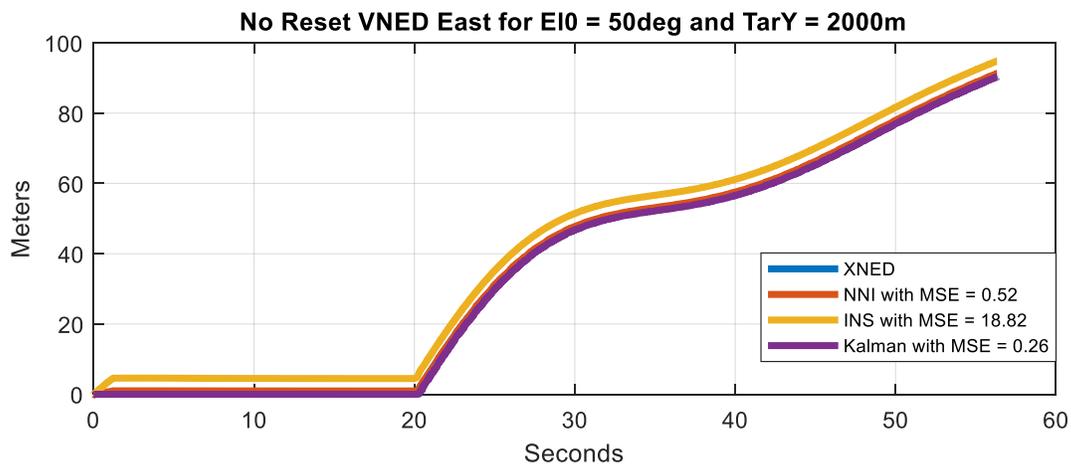


Figure 296 – Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 2000m

Figure *297* and Figure *298* show the comparison of east velocity GPS/INS integration in NED frame without reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter performed the best for both cases. However, the NNI performed fine with a bigger error with the deviation case. That is because the NNI algorithm struggles with zero input with a bias. In addition, the INS algorithm performed fine although with a higher MSE.
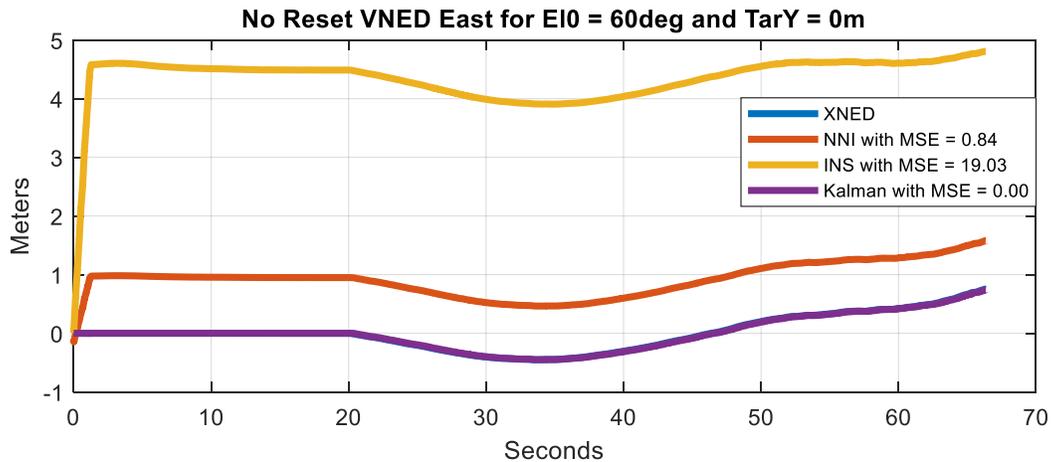


Figure 297 – Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 0m
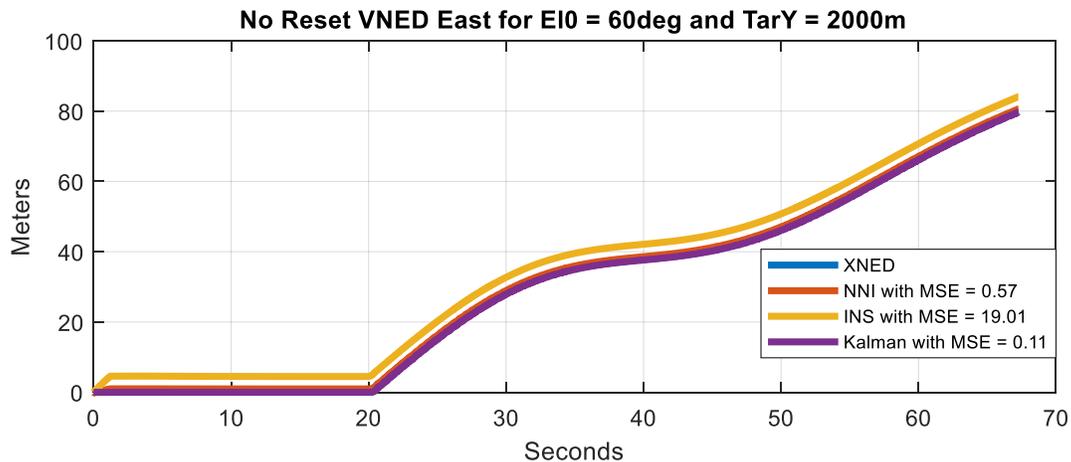


Figure 298 – Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 2000m

Figure *299* and Figure *300* show the comparison of east velocity GPS/INS integration in NED frame without reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter performed the best for both cases. However, the NNI performed fine with a bigger error with the deviation case. That is because the NNI algorithm struggles with zero input with a bias. In addition, the INS algorithm performed fine although with a higher MSE.
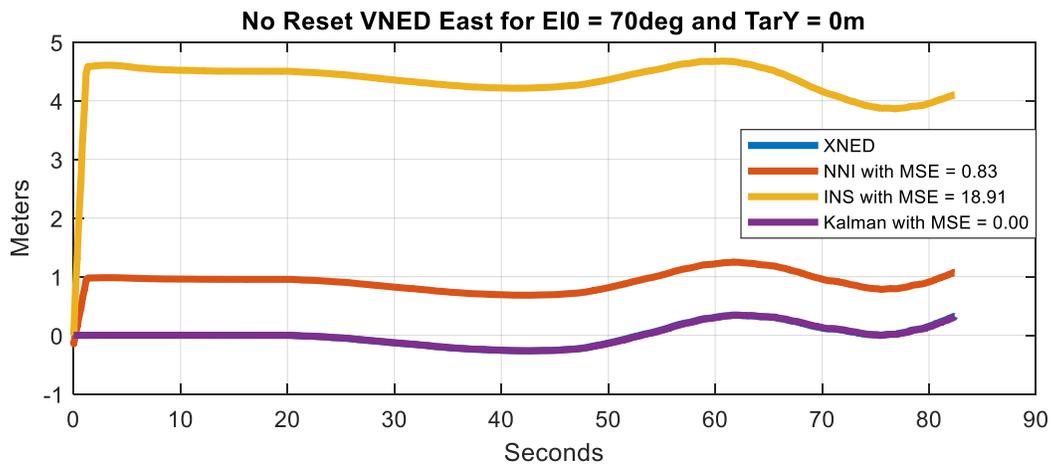


Figure 299 - Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 0m
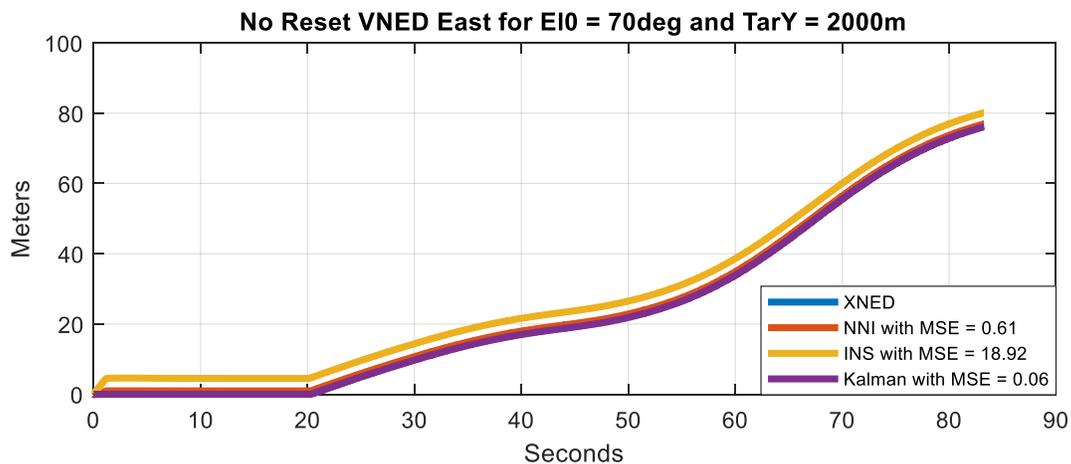


Figure 300 - Comparison of East Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 2000m

Figure *301* and Figure *302* show the comparison of north velocity GPS/INS integration in NED frame without reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI performed the best for both cases with the least MSE. Nevertheless, the INS performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
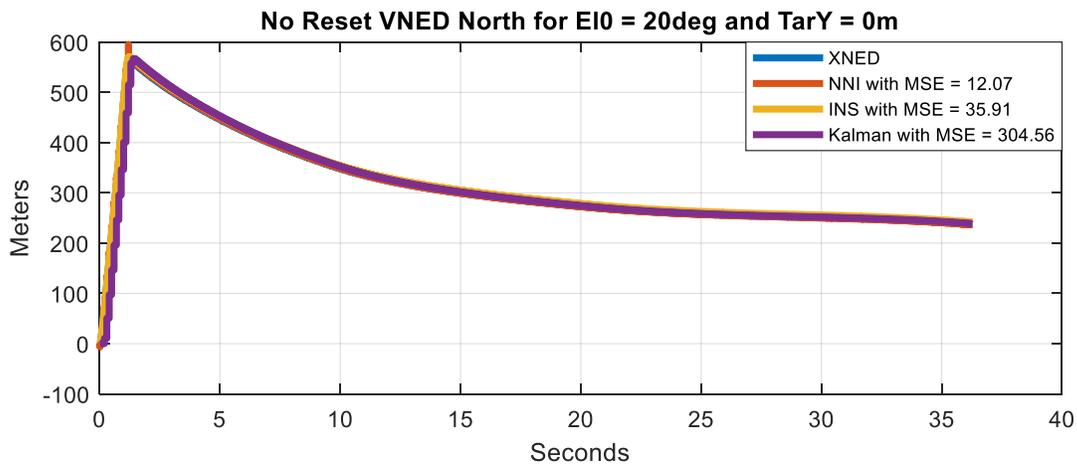


Figure 301 - Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 0m
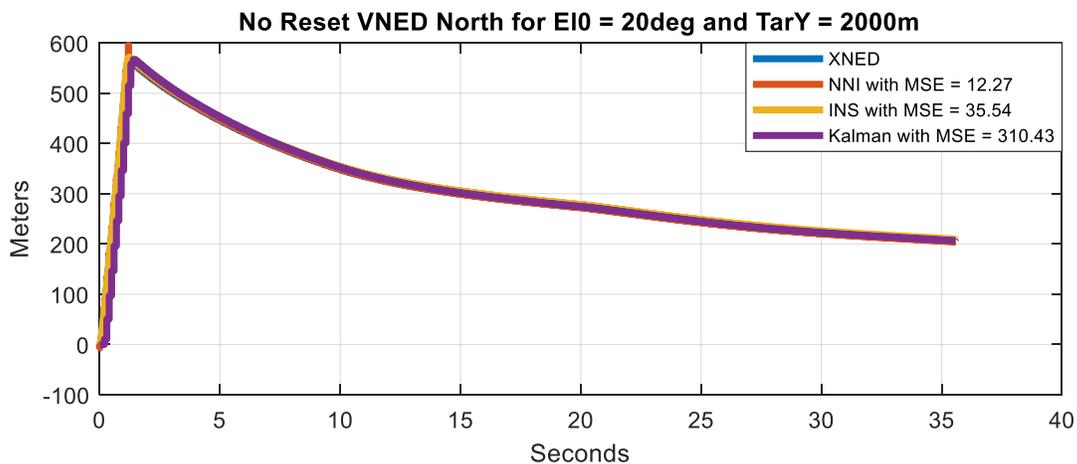


Figure 302 - Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 20 deg and Target Y = 2000m

Figure *303* and Figure *304* show the comparison of north velocity GPS/INS integration in NED frame without reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI performed the best for both cases with the least MSE. Nevertheless, the INS performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
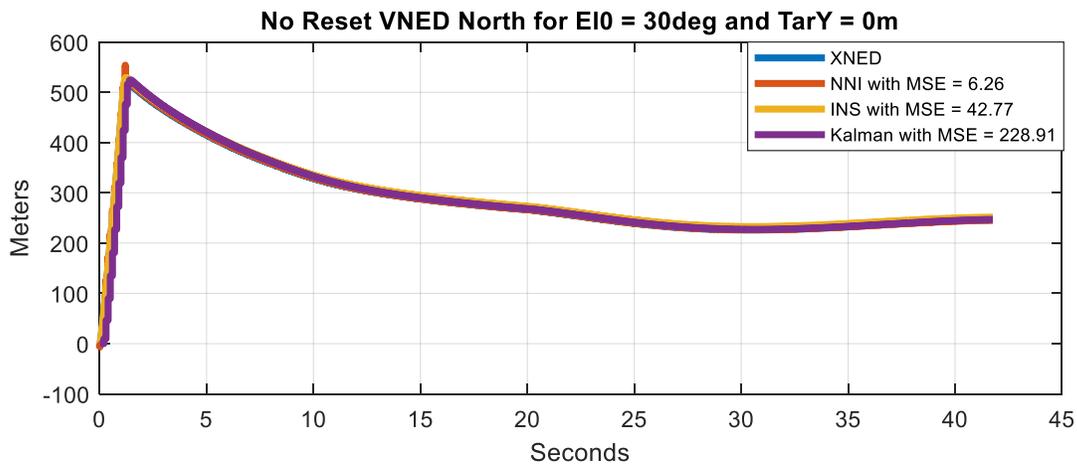


Figure 303 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 0m
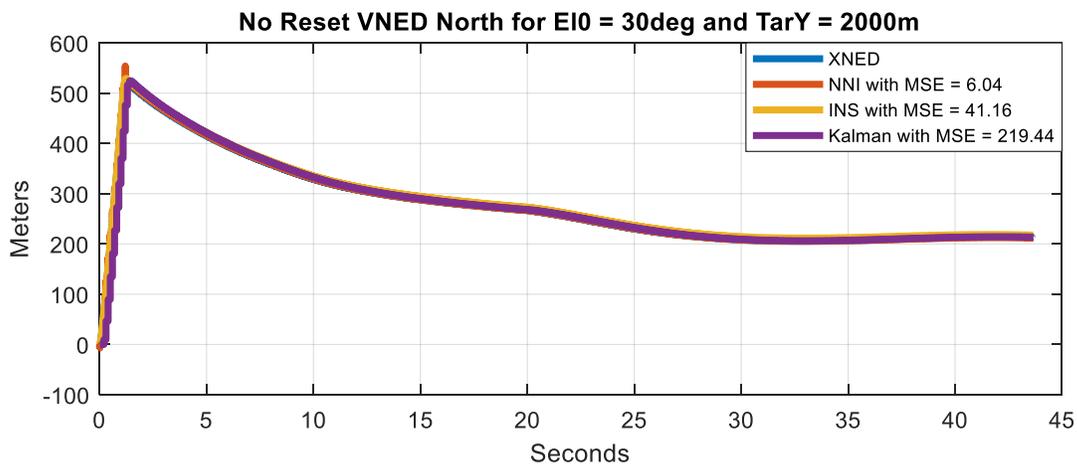


Figure 304 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 30 deg and Target Y = 2000m

Figure *305* and Figure *306* show the comparison of north velocity GPS/INS integration in NED frame without reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI performed the best for both cases with the least MSE. Nevertheless, the INS performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
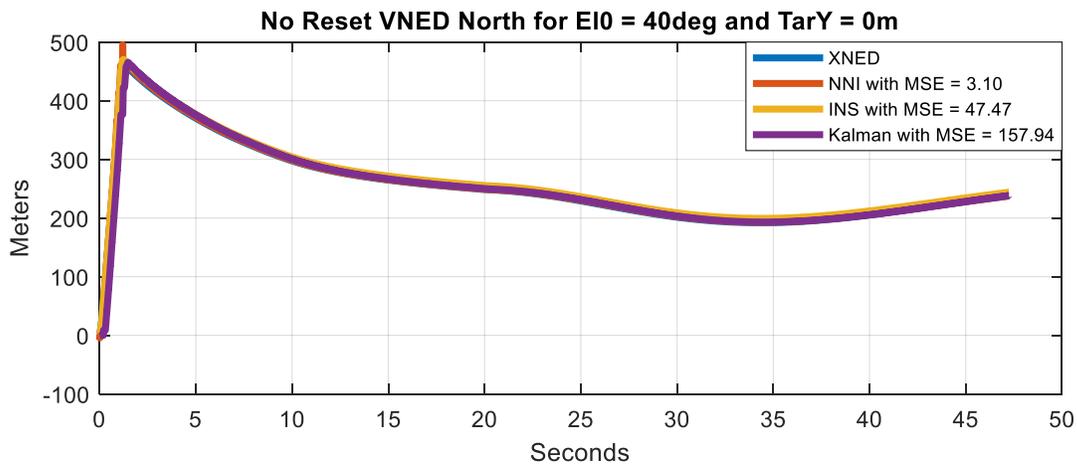


Figure 305 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 0m
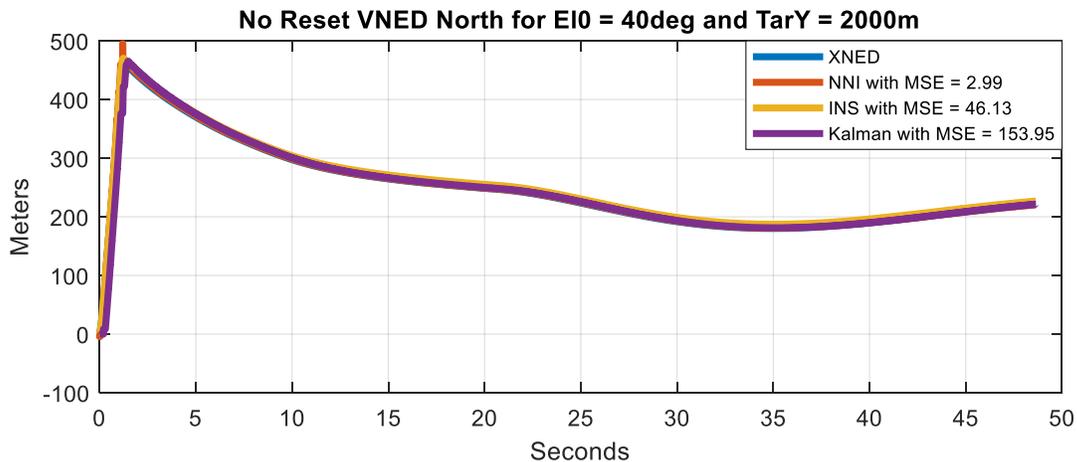


Figure 306 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 40 deg and Target Y = 2000m

Figure *307* and Figure *308* show the comparison of north velocity GPS/INS integration in NED frame without reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI performed the best for both cases with the least MSE. Nevertheless, the INS performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
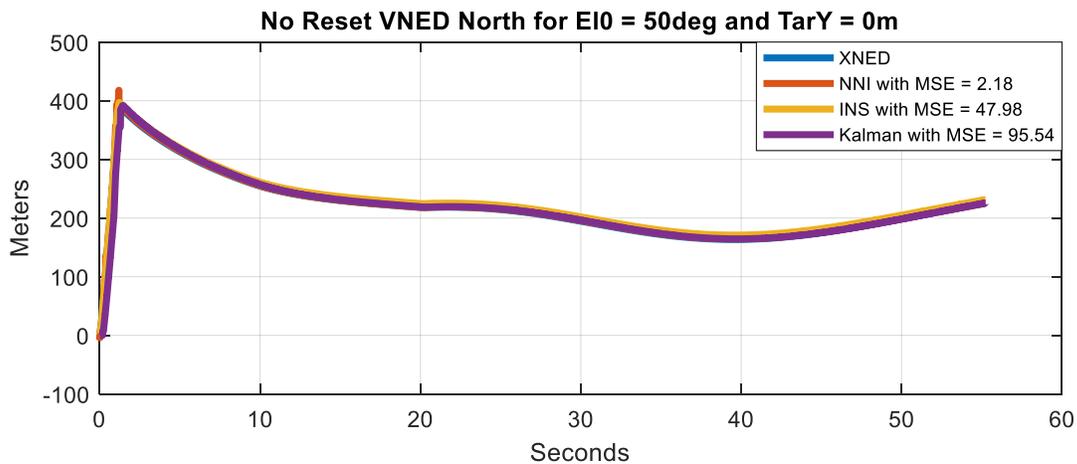


Figure 307 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 0m
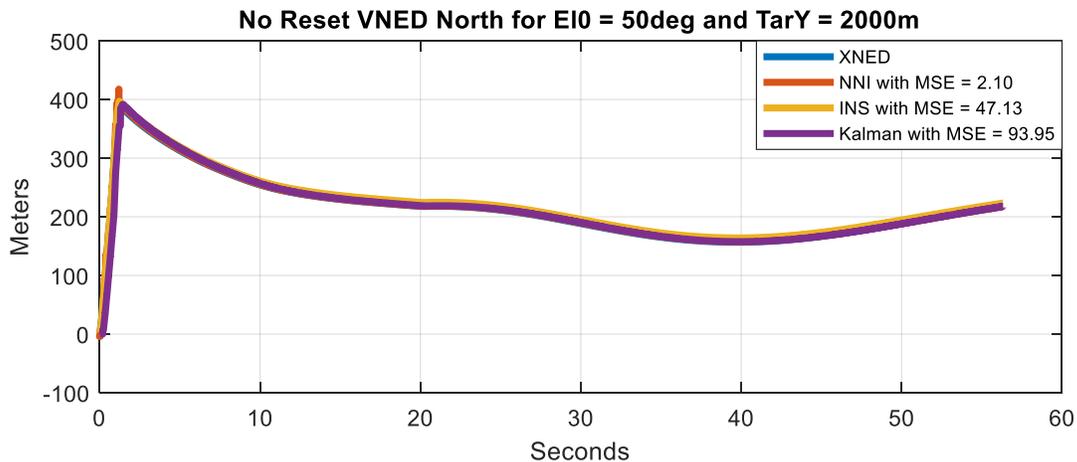


Figure 308 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 50 deg and Target Y = 2000m

Figure *309* and Figure *310* show the comparison of north velocity GPS/INS integration in NED frame without reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI performed the best for both cases with the least MSE. Nevertheless, the INS performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.
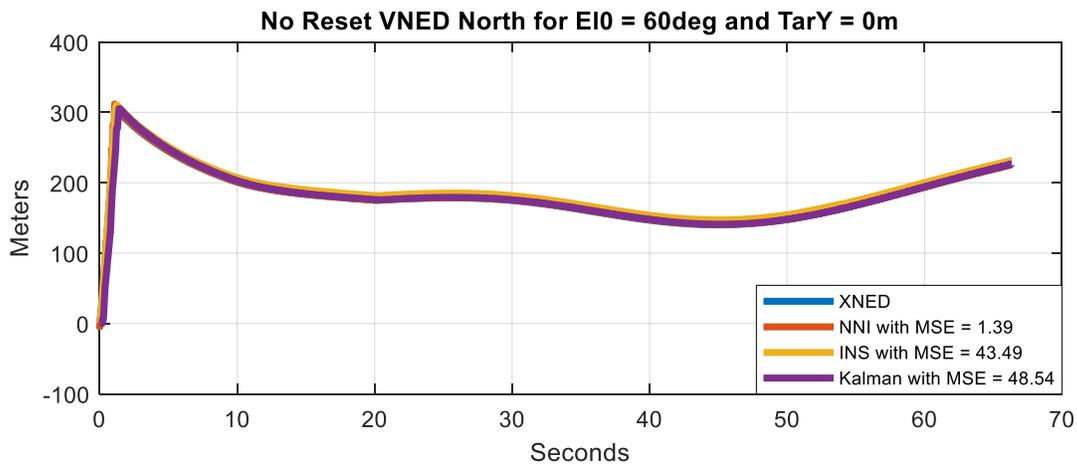


Figure 309 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 0m
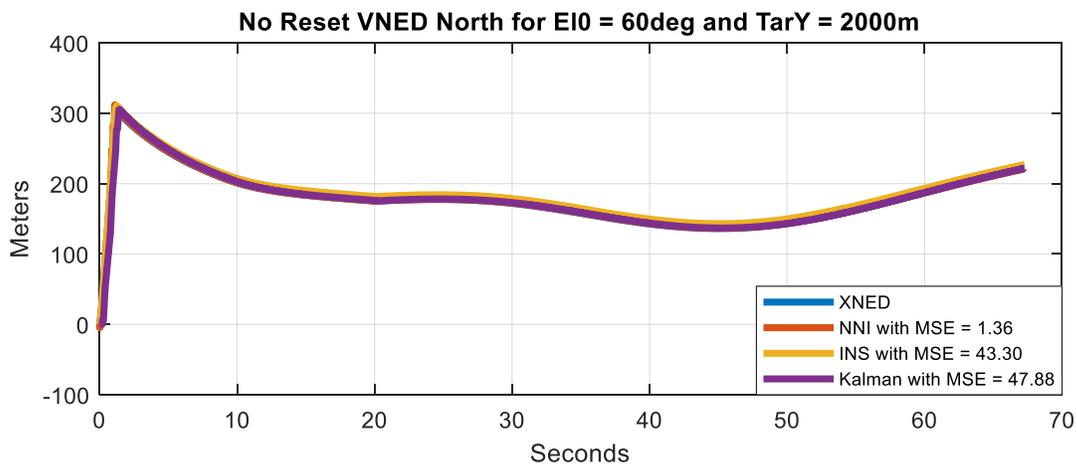


Figure 310 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 60 deg and Target Y = 2000m

Figure *311* and Figure *312* show the comparison of north velocity GPS/INS integration in NED frame without reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI performed the best for both cases with the least MSE. Nevertheless, the INS performed fine. Moreover, the Kalman filter also performed fine but with a higher MSE. It is important to note that the deviation did not affect the performance much.



**No Reset VNED North for El0 = 70deg and TarY = 0m**
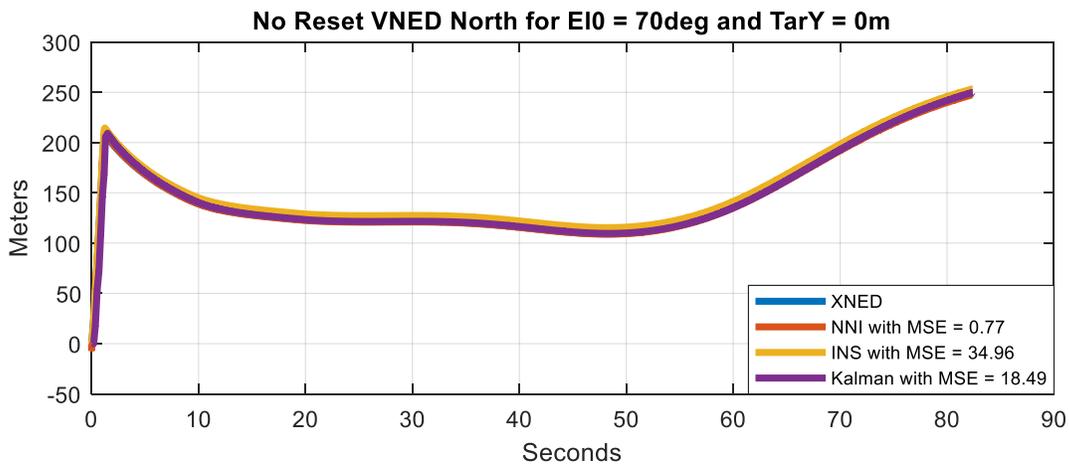
Figure 311 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 0m



**No Reset VNED North for El0 = 70deg and TarY = 2000m**
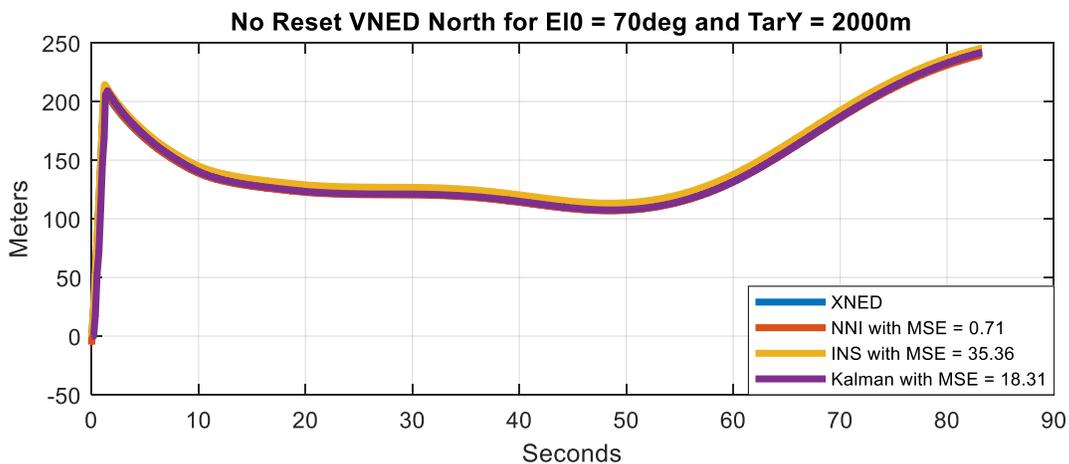
Figure 312 – Comparison of North Velocity GPS/INS Integration in NED Frame Between INS, Kalman Filter, and NNI Without Reset for El0 = 70 deg and Target Y = 2000m

Figure 313 shows a comparison of the histogram distribution of down position GPS/INS integration in NED frame without reset for INS, NNI, and Kalman filter. It can be seen that INS has the highest mean and standard of deviation. This means that it is the least accurate overall despite having better performances in some cases. In addition, the NNI has the lowest mean and deviation. This means that is the most accurate despite having cases where it did not perform the best. Moreover, it can be seen that the Kalman filter performed fine overall.



**No Reset XNED Down Historgram**

Legend:
- INS with σ = 20133.70 and μ = 11549.28
- NNI with σ = 417.52 and μ = 517.33
- Kalman with σ = 438.64 and μ = 708.64
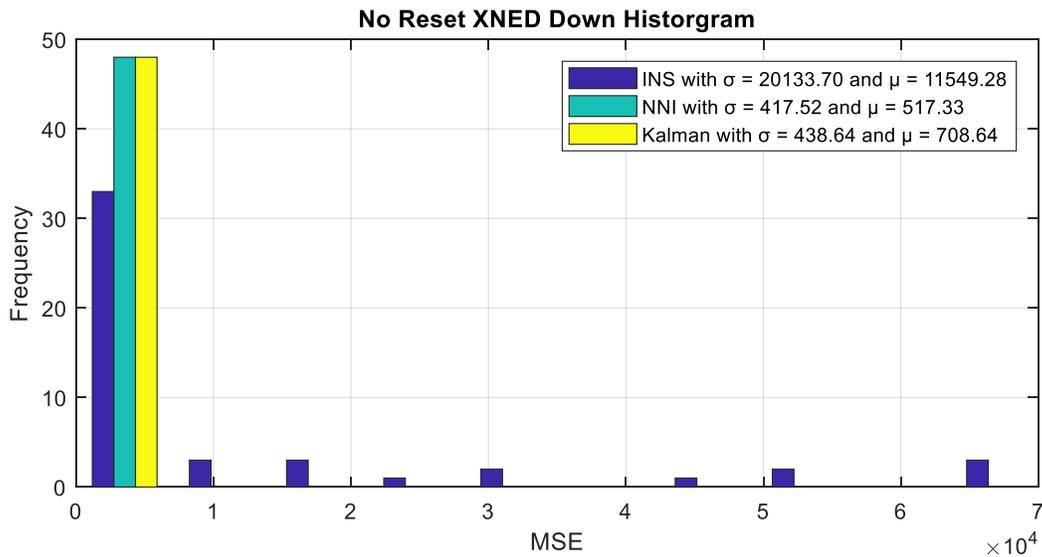
*Figure 313* – Comparison of The Histogram Distribution for Down Position GPS/INS Integration in NED Frame Without Reset Between INS, NNI, and Kalman Filter.

Figure *314* shows a comparison of the histogram distribution of east position GPS/INS integration in NED frame without reset for INS, NNI, and Kalman filter. It can be seen that INS has the highest mean and standard of deviation. This means that it is the least accurate overall which was clearly seen in its performances. In addition, the Kalman filter has the lowest mean and deviation. This means that it is the most accurate despite having cases where it did not perform the best. Moreover, it can be seen that the NNI performed fine overall. However, the NNI distribution is skewed because of its challenges with zero input. Thus, making the Kalman filter a better solution for side navigation integration.

**No Reset XNED East Historgram**
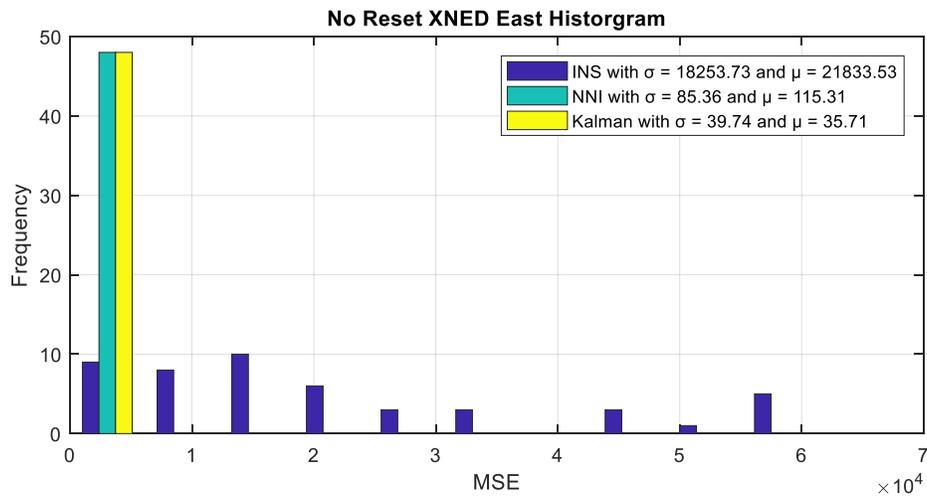
Figure 314 – Comparison of The Histogram Distribution for East Position GPS/INS Integration in NED Frame Without Reset Between INS, NNI, and Kalman Filter.

Figure *315* shows a comparison of the histogram distribution of north position GPS/INS integration in NED frame without reset for INS, NNI, and Kalman filter. It can be seen that INS has the highest mean and standard of deviation. This means that it is the least accurate overall which was clearly seen in its performances. In addition, the NNI has the lowest mean and deviation. This means that it is the most accurate despite which was also seen in its performances. Moreover, it can be seen that the Kalman filter performed fine overall.



**No Reset XNED North Historgram**
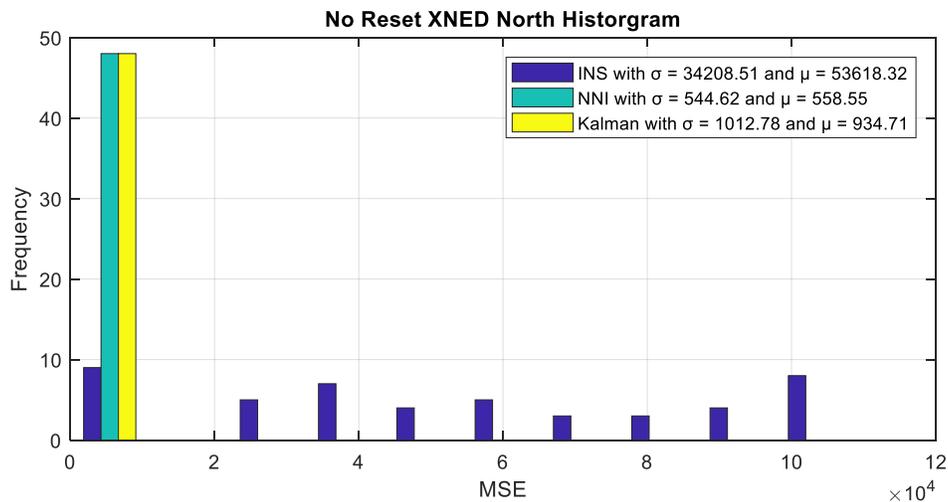
Figure 315 – Comparison of The Histogram Distribution for North Position GPS/INS Integration in NED Frame Without Reset Between INS, NNI, and Kalman Filter.

Figure *316* shows a comparison of the histogram distribution of down velocity GPS/INS integration in NED frame without reset for INS, NNI, and Kalman filter. It can be seen that Kalman filter has the highest mean and standard of deviation. This means that it is the least accurate overall which was seen in its performances. In addition, the NNI has the lowest mean and deviation. This means that it is the most accurate despite having cases where it did not perform the best. In addition, it can be seen that the INS filter performed fine overall.



**No Reset VNED Down Historgram**

Legend:
- INS with σ = 7.59 and μ = 8.51
- NNI with σ = 3.93 and μ = 8.87
- Kalman with σ = 46.00 and μ = 88.64
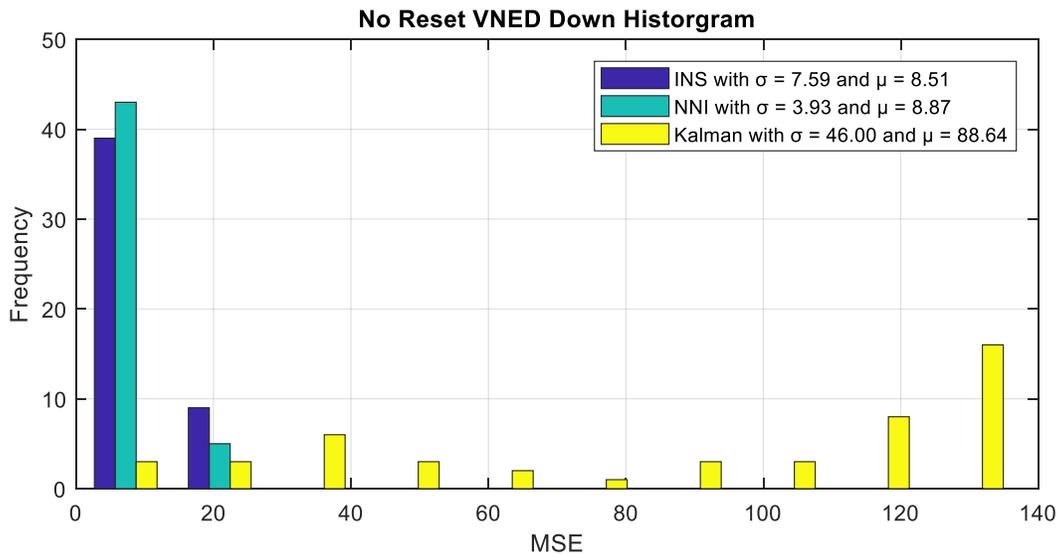
Figure 316 – Comparison of The Histogram Distribution for Down Velocity GPS/INS Integration in NED Frame Without Reset Between INS, NNI, and Kalman Filter.

Figure *317* shows a comparison of the histogram distribution of east velocity GPS/INS integration in NED frame without reset for INS, NNI, and Kalman filter. It can be seen that INS has the highest mean but the lowest standard of deviation. This means that it is the least accurate but the most precise. In addition, the Kalman filter has the lowest mean and a low stand of deviation. This means that it is the most accurate as seen from its performances. In addition, the NNI performed fine. However, its distribution is mostly skewed because of its challenges with zero input.

**No Reset VNED East Historgram**
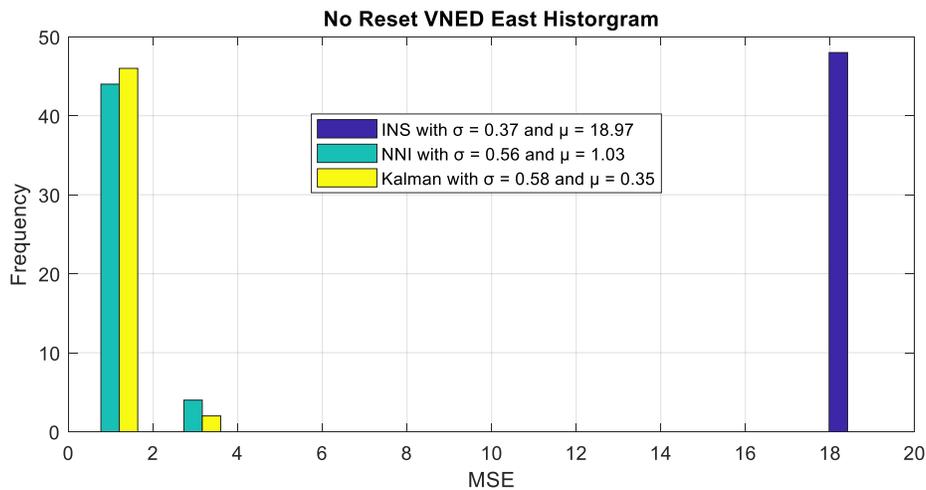


Figure 317 – Comparison of The Histogram Distribution for East Velocity GPS/INS Integration in NED Frame Without Reset Between INS, NNI, and Kalman Filter.

Figure *318* shows a comparison of the histogram distribution of north velocity GPS/INS integration in NED frame without reset for INS, NNI, and Kalman filter. It can be seen that INS has the highest mean but the lowest standard of deviation. This means that it is the least accurate but most precise. In addition, the NNI has the lowest mean and a low stand of deviation. This means that it is the most accurate as seen from its performances. In addition, the Kalman filter performed fine. However, it is the least accurate and least precise.

**No Reset VNED North Historgram**
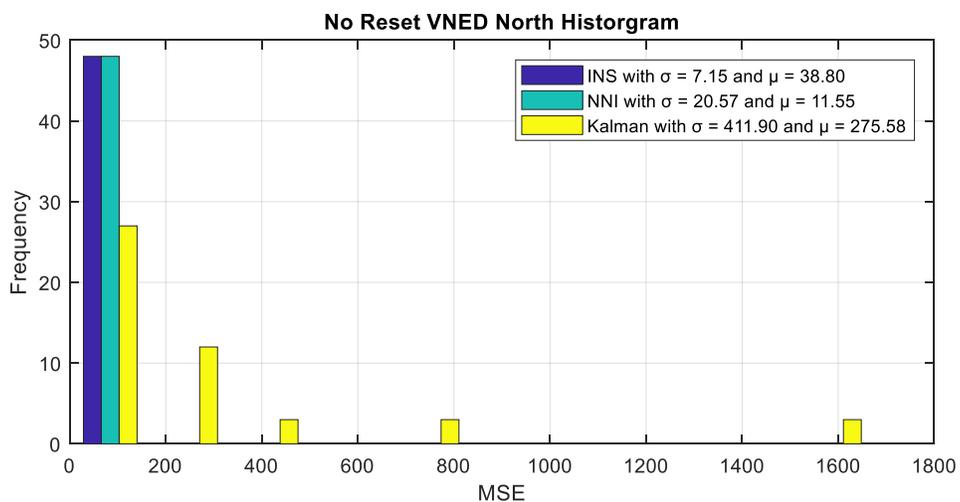


Figure 318 – Comparison of The Histogram Distribution for North Velocity GPS/INS Integration in NED Frame Without Reset Between INS, NNI, and Kalman Filter.

Figure *319* and Figure *320* show the comparison of down position GPS/INS integration in NED frame with reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
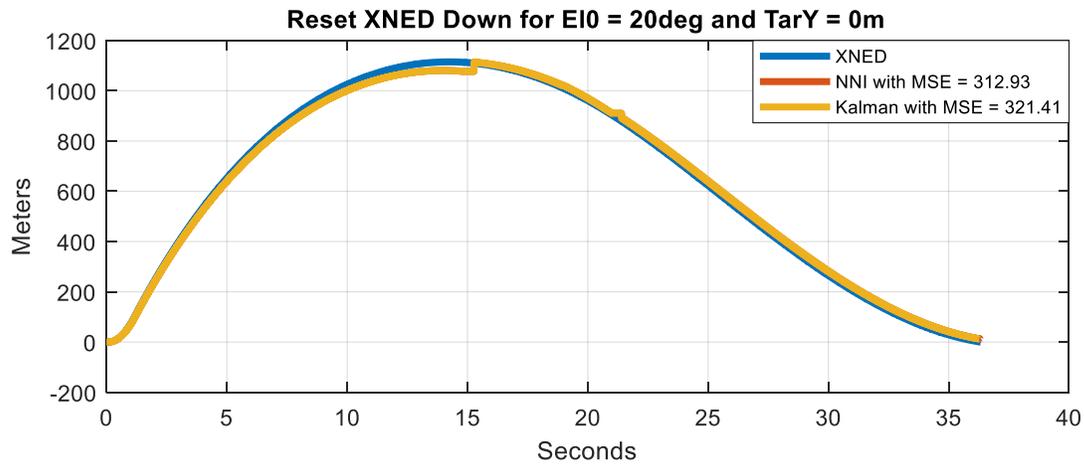


Figure 319 - Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 0m
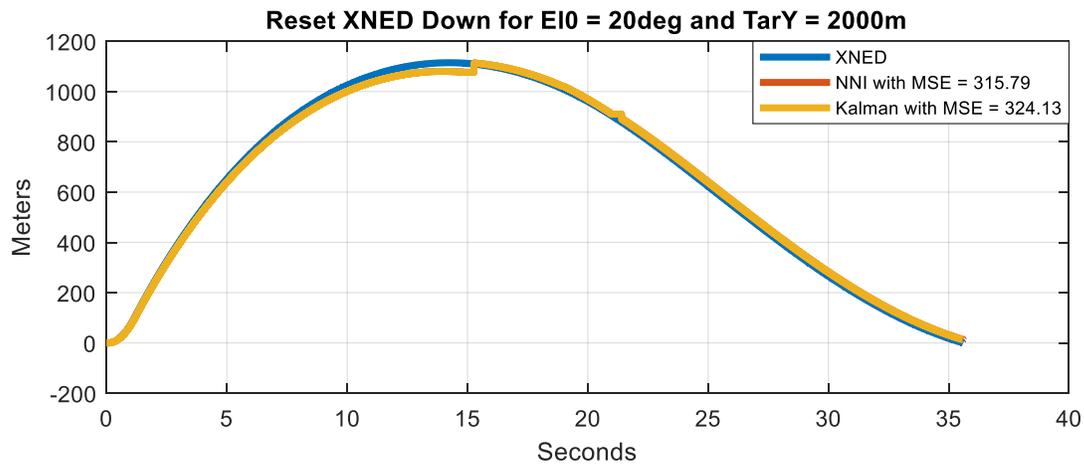


Figure 320 - Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 2000m

Figure *321* and Figure *322* show the comparison of down position GPS/INS integration in NED frame with reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
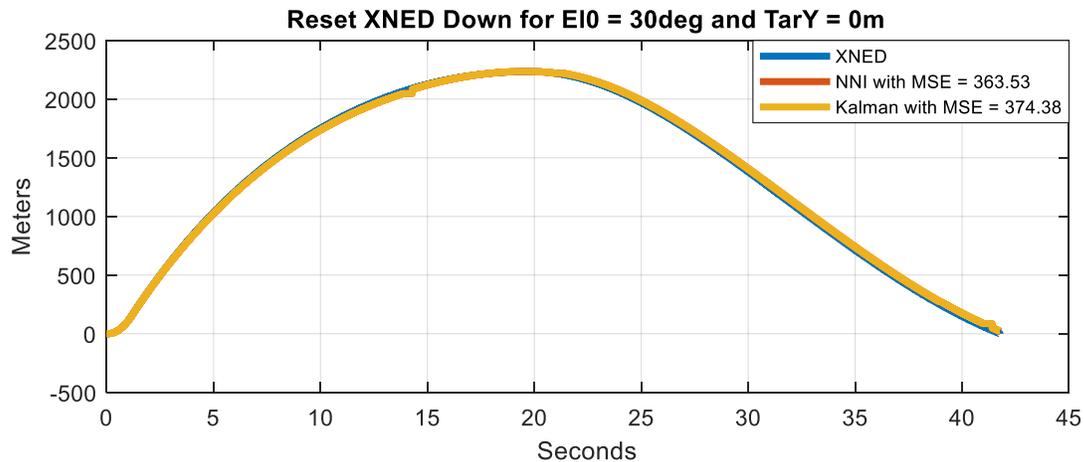


Figure 321 – Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 0m
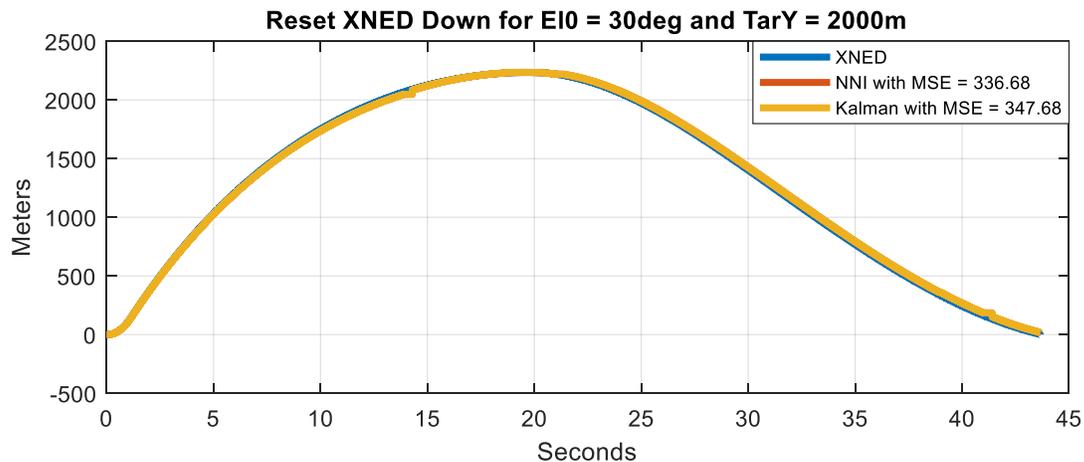


Figure 322 – Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 2000m

Figure *323* and Figure *324* show the comparison of down position GPS/INS integration in NED frame with reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
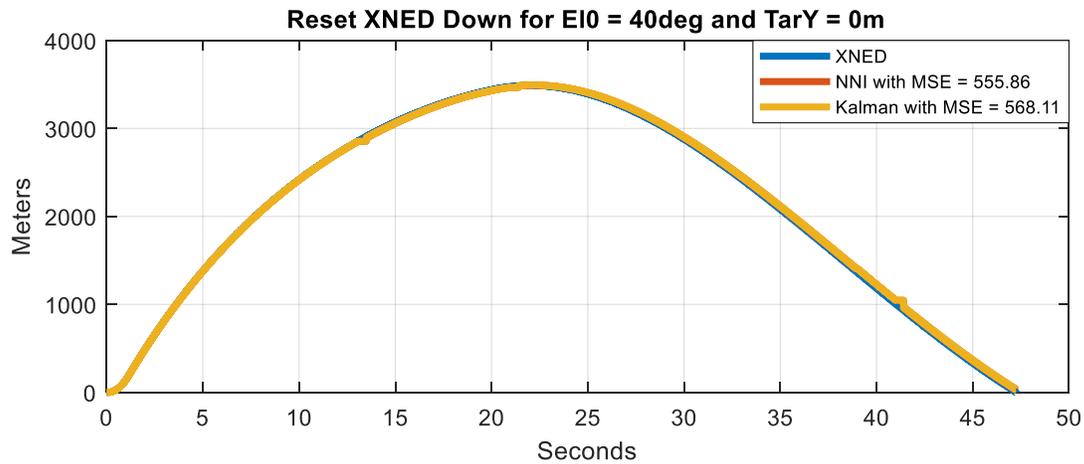


Figure 323 - Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 0m
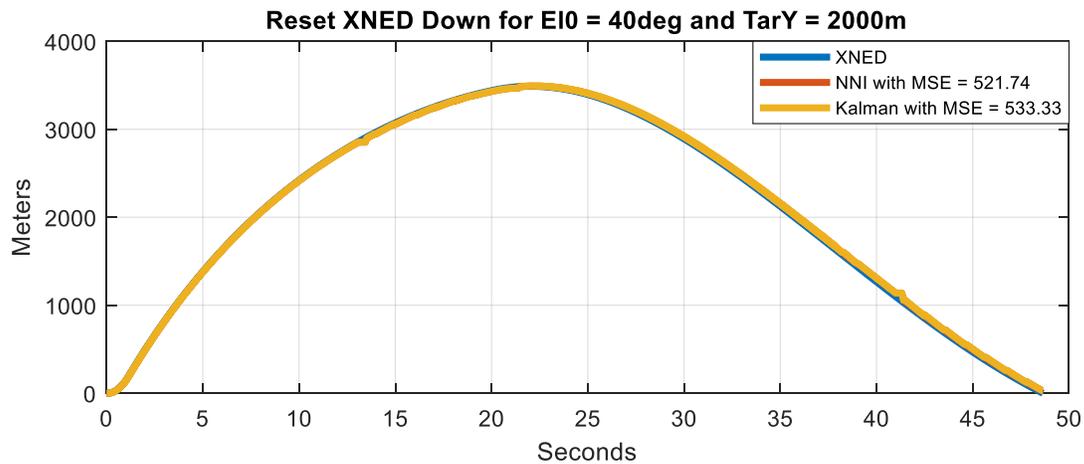


Figure 324 - Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 2000m

Figure *325* and Figure *326* show the comparison of down position GPS/INS integration in NED frame with reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
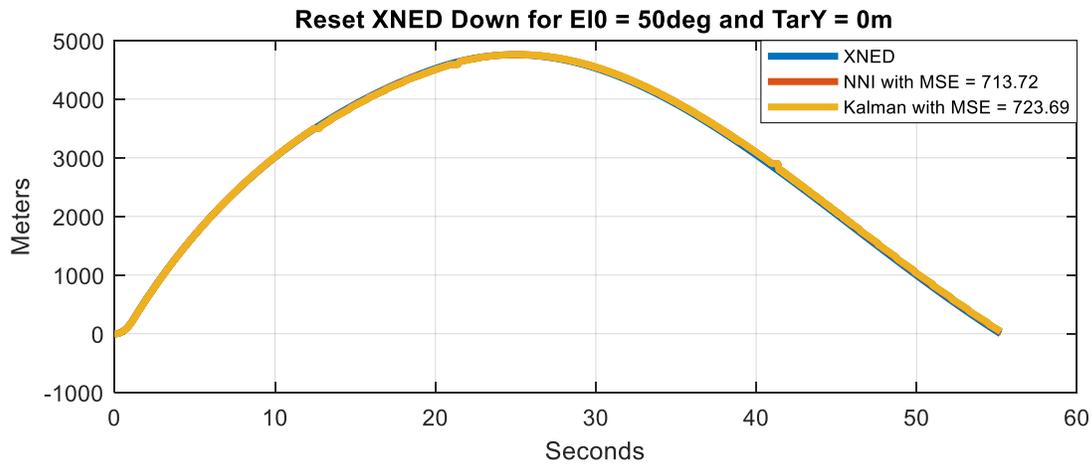


Figure 325 – Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 0m
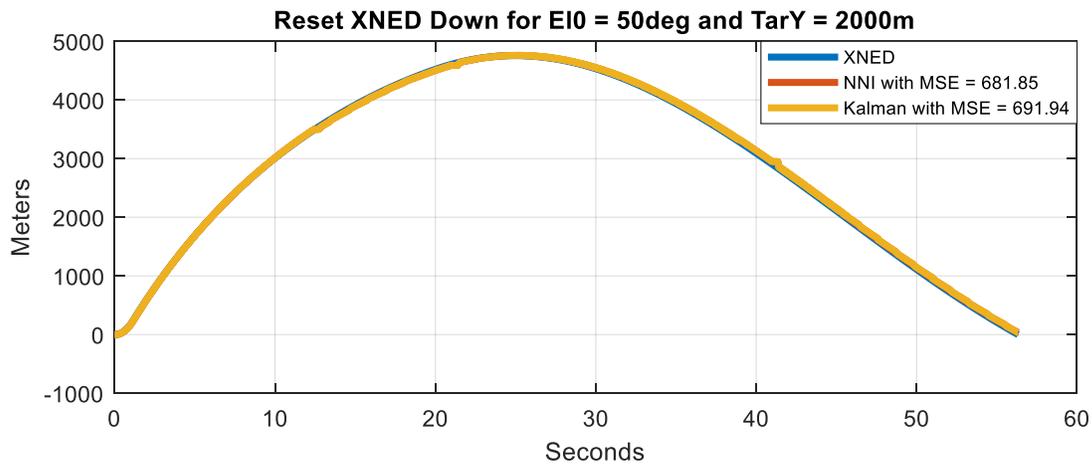


Figure 326 – Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 2000m

Figure *327* and Figure *328* show the comparison of down position GPS/INS integration in NED frame with reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
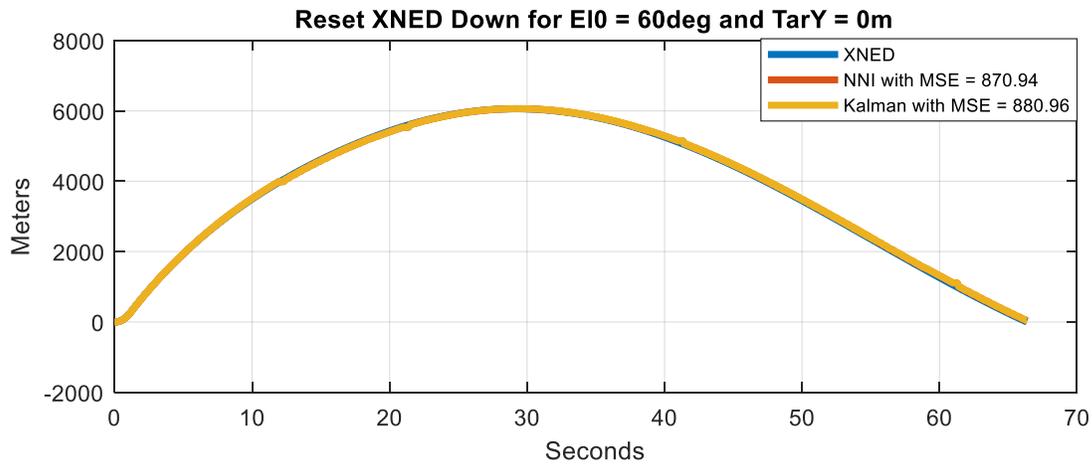


Figure 327 – Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 0m
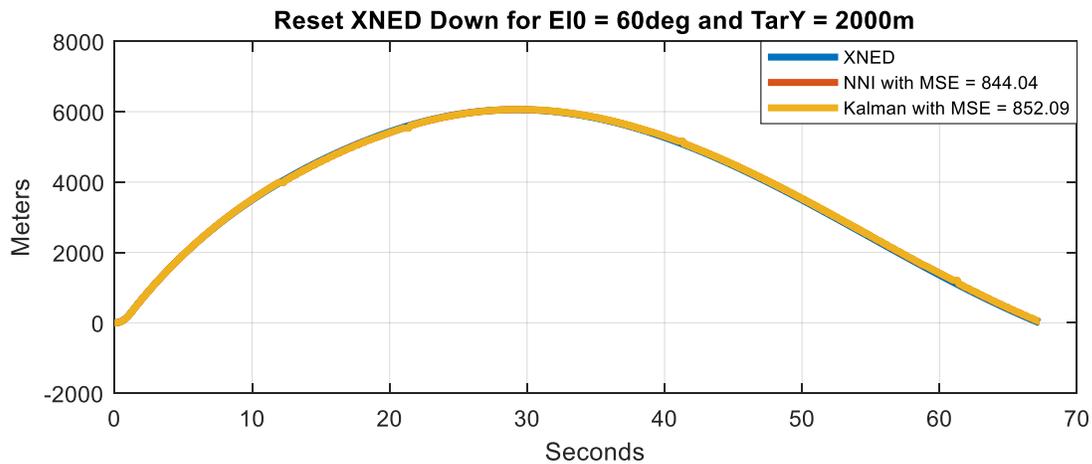


Figure 328 – Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 2000m

Figure *329* and Figure *330* show the comparison of down position GPS/INS integration in NED frame with reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. In addition, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
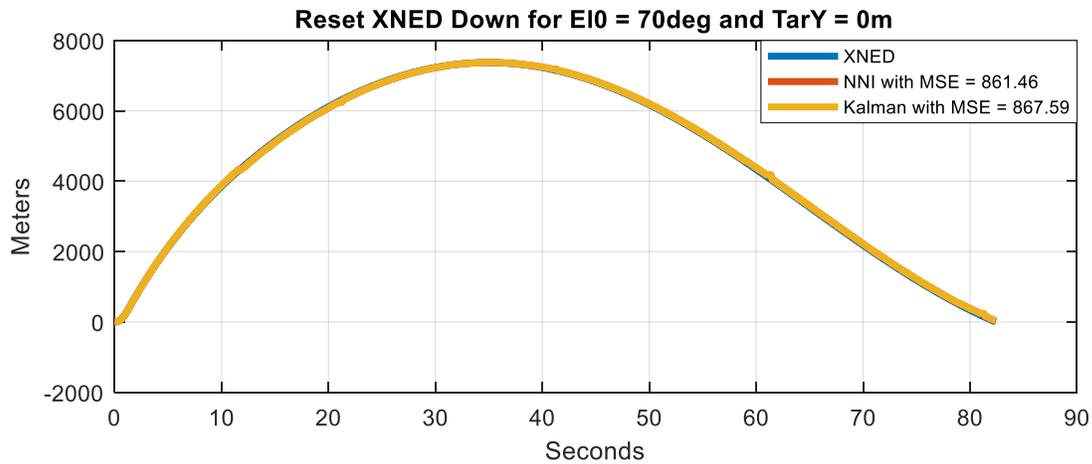


Figure 329 – Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 0m
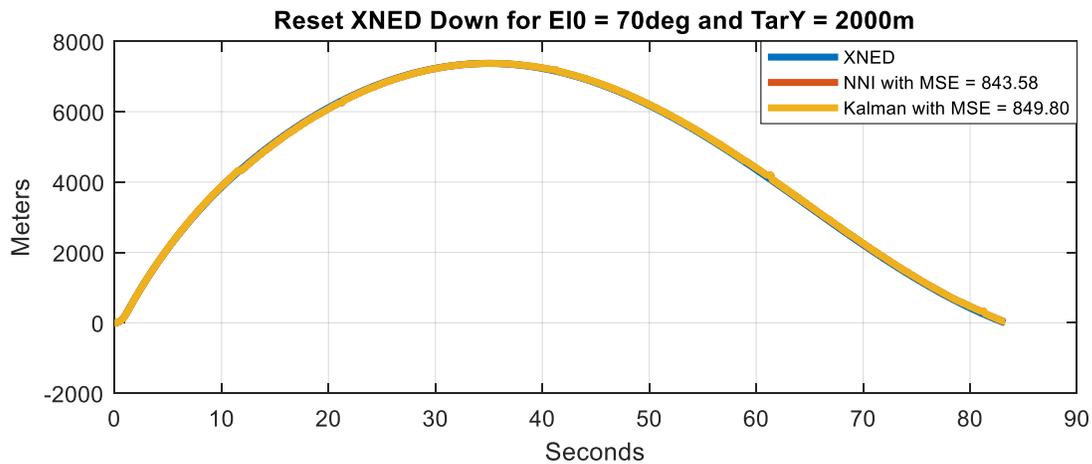


Figure 330 – Comparison of Down Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 2000m

Figure *331* and Figure *332* show the comparison of east position GPS/INS integration in NED frame with reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.
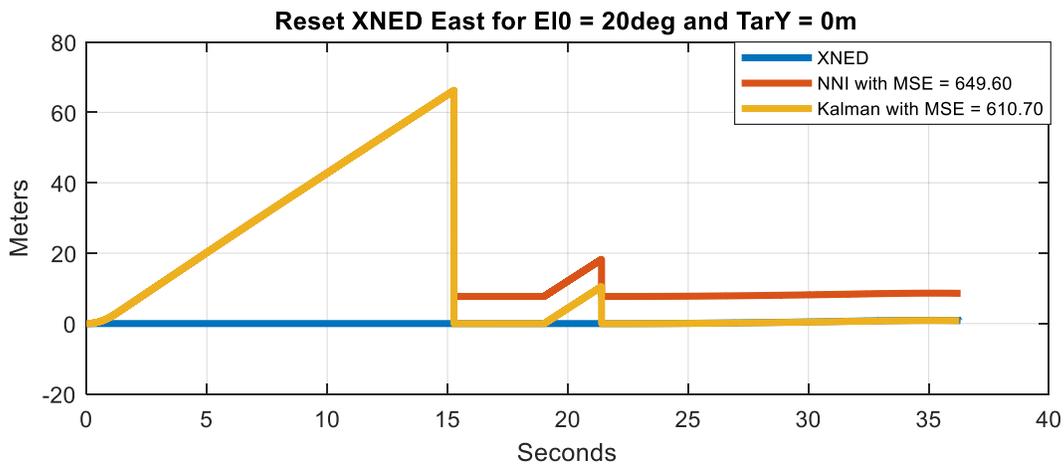


Figure 331 – Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 0m
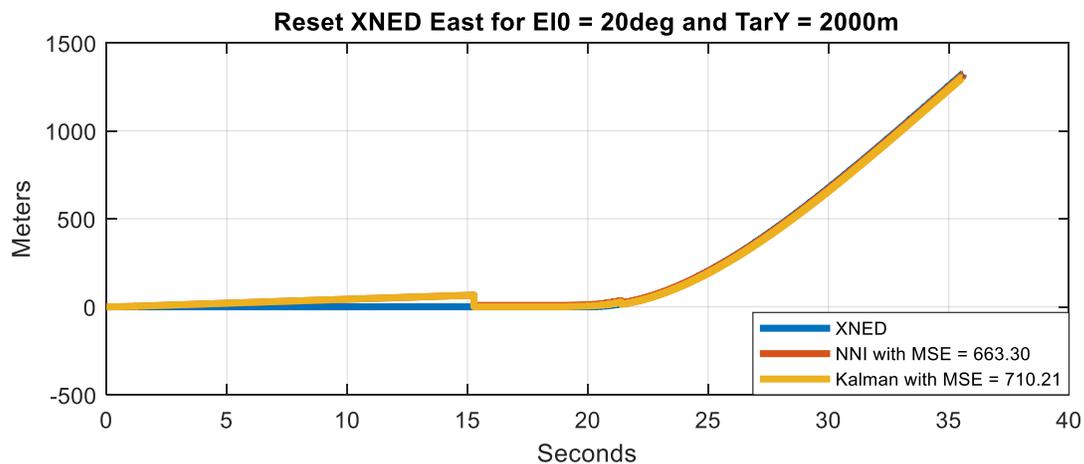


Figure 332 – Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 2000m

Figure *333* and Figure *334* show the comparison of east position GPS/INS integration in NED frame with reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.
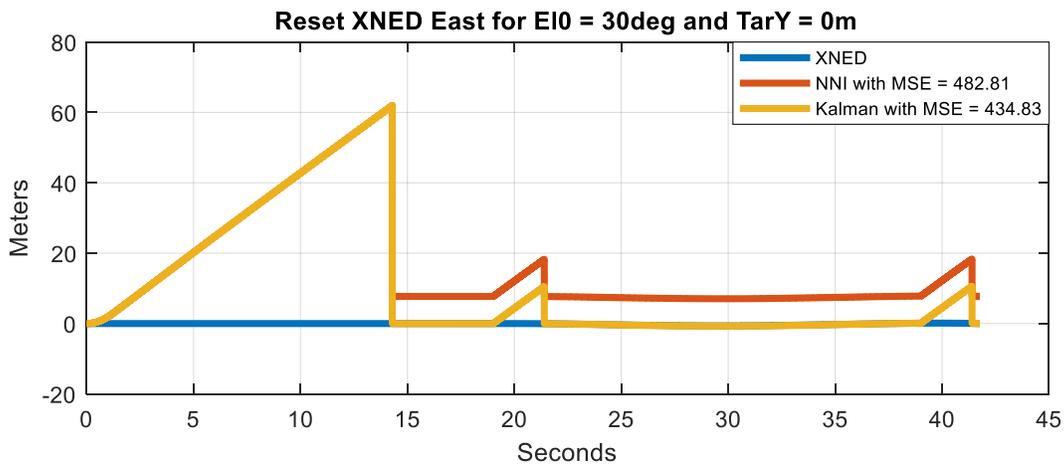


Figure 333 - Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 0m
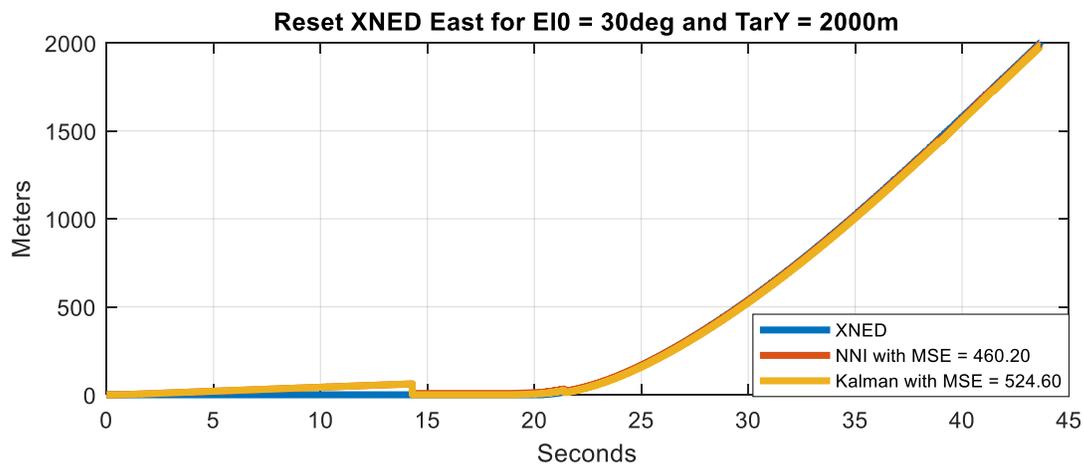


Figure 334 - Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 2000m

Figure *335* and Figure *336* show the comparison of east position GPS/INS integration in NED frame with reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.
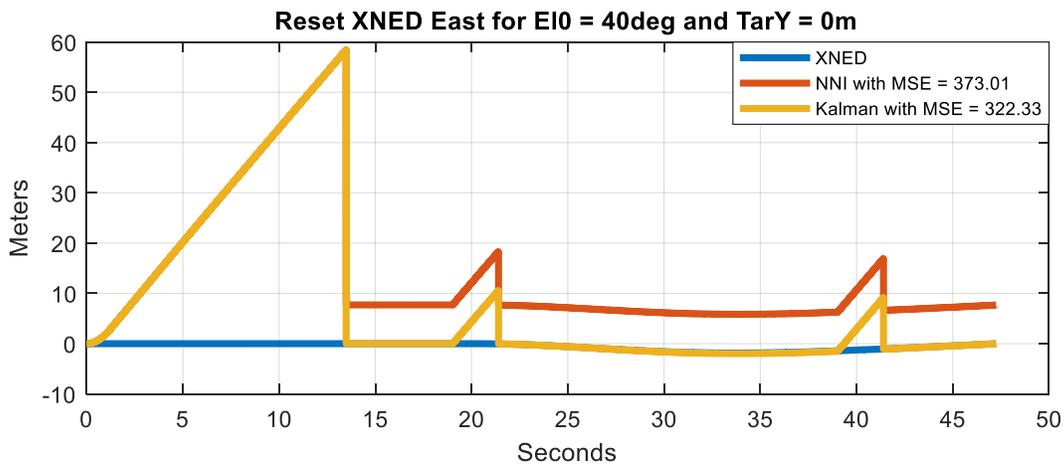


Figure 335 – Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 0m
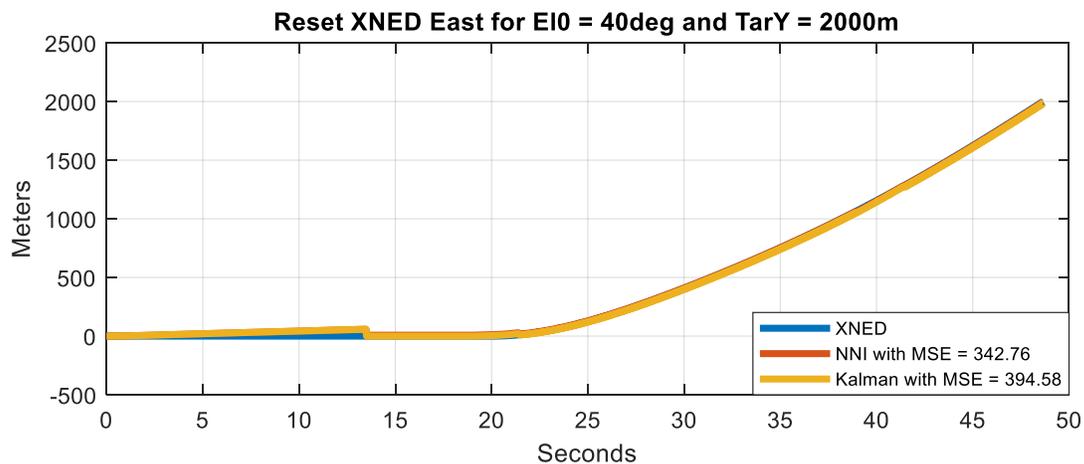


Figure 336 – Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 2000m

Figure *337* and Figure *338* show the comparison of east position GPS/INS integration in NED frame with reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.
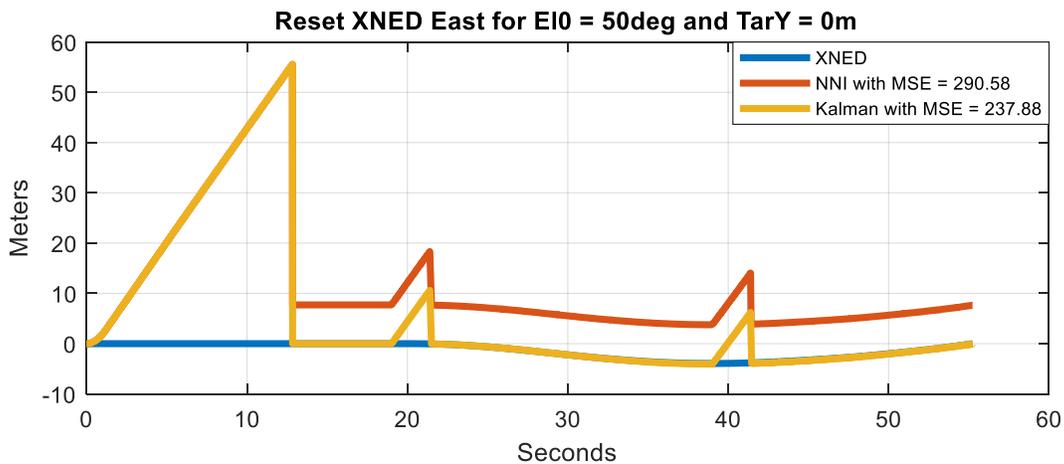


Figure 337 – Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 0m
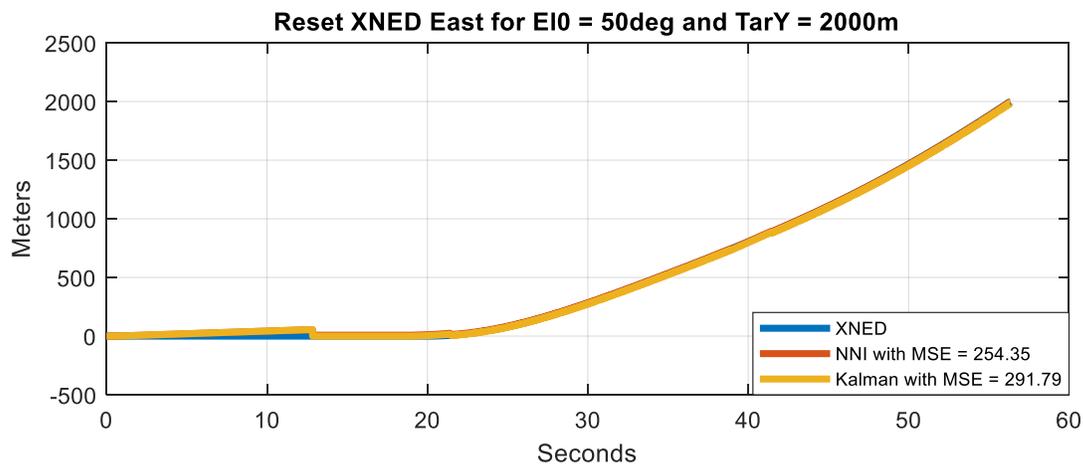


Figure 338 – Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 2000m

Figure *339* and Figure *340* show the comparison of east position GPS/INS integration in NED frame with reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.
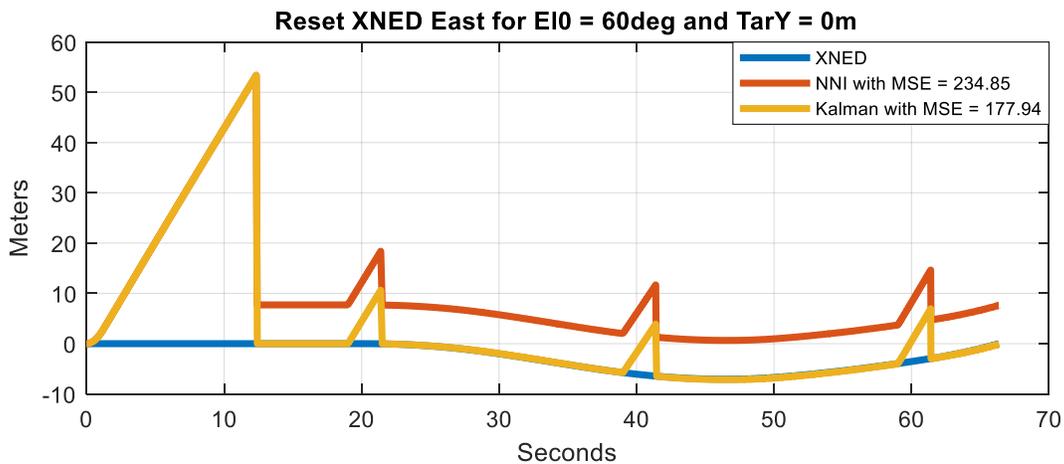


Figure 339 - Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 0m
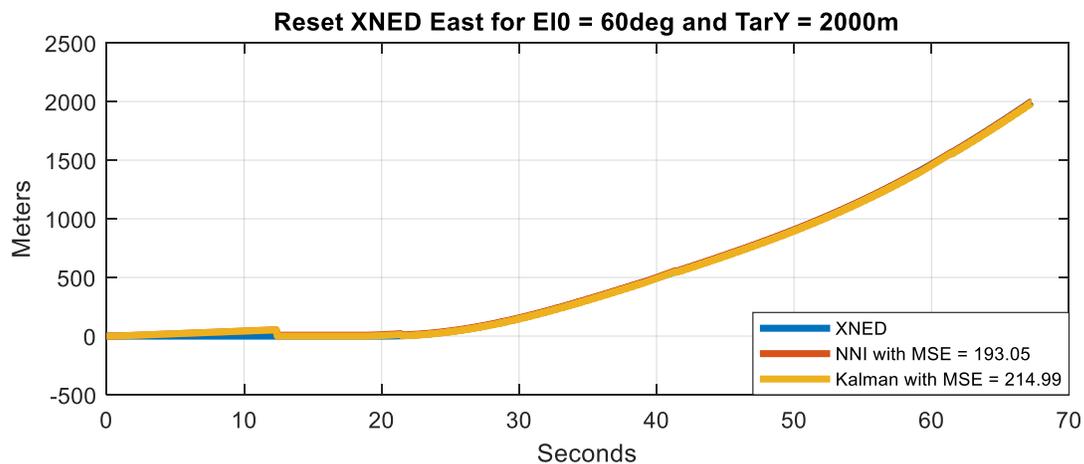


Figure 340 - Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 2000m

Figure *341* and Figure *342* show the comparison of east position GPS/INS integration in NED frame with reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.
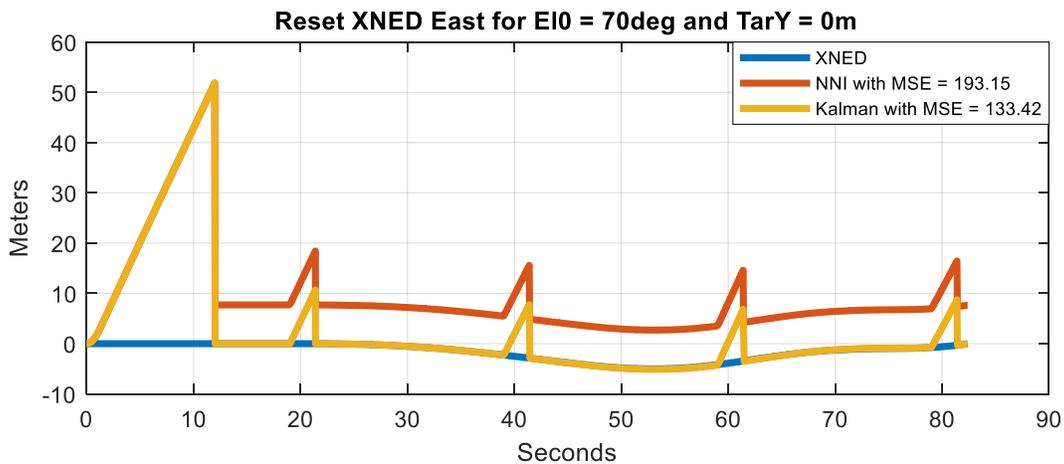


Figure 341 – Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 0m
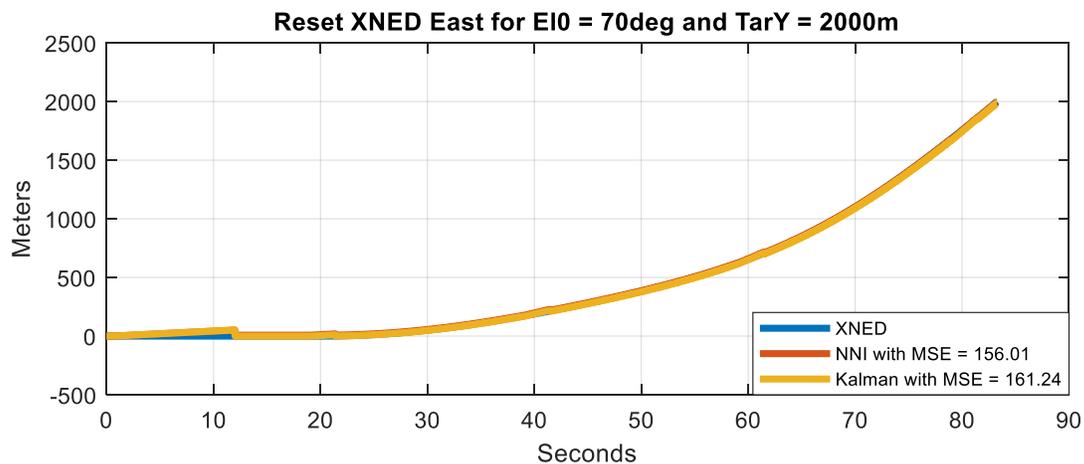


Figure 342 – Comparison of East Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 2000m

Figure *343* and Figure *344* show the comparison of north position GPS/INS integration in NED frame with reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman algorithm performed the best for both cases with the least MSE. Nevertheless, the NNI performed fine as well. It is an interesting phenomenon because with the No Reset case NNI performed better. It is important to note that the deviation did not affect the performance much.
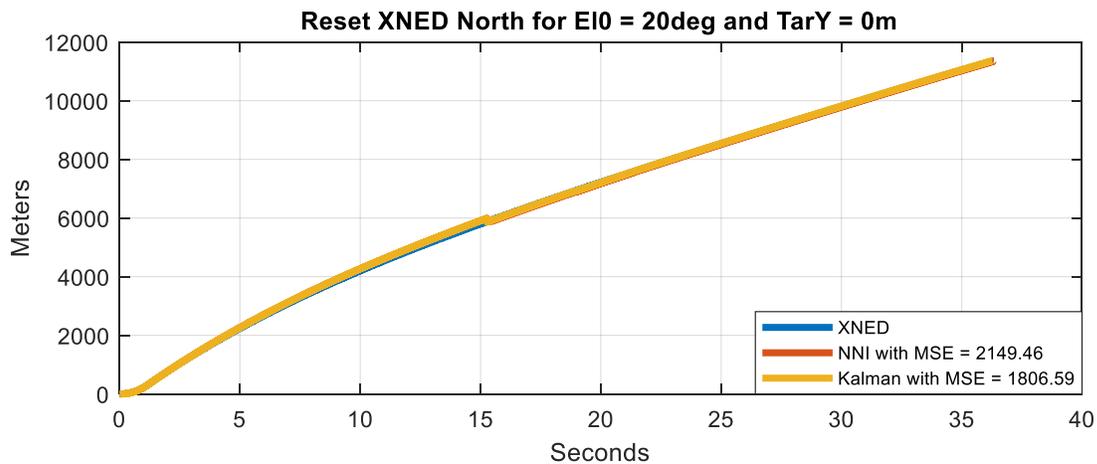


Figure 343 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 0m
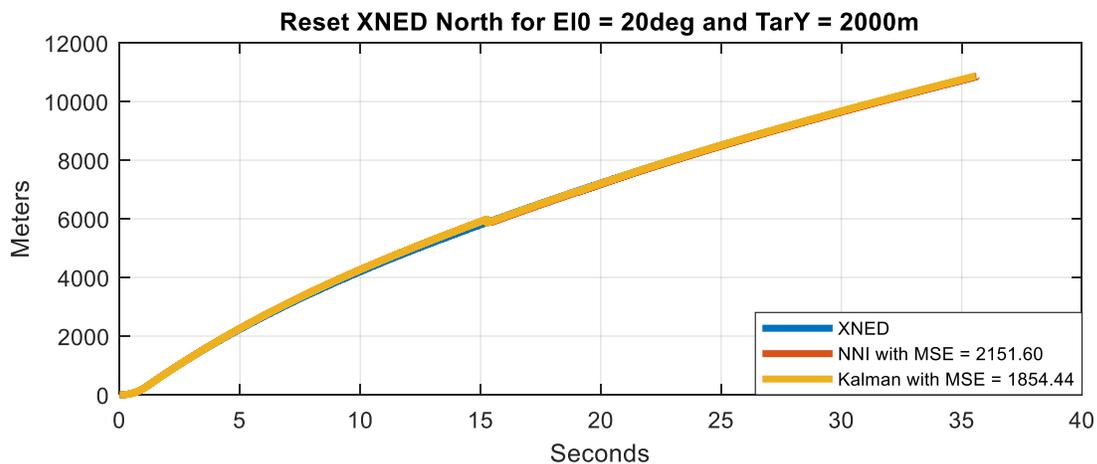


Figure 344 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 2000m

Figure *345* and Figure *346* show the comparison of north position GPS/INS integration in NED frame with reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman algorithm performed the best for both cases with the least MSE. Nevertheless, the NNI performed fine as well. It is an interesting phenomenon because with the No Reset case NNI performed better. It is important to note that the deviation did not affect the performance much.

**Reset XNED North for El0 = 30deg and TarY = 0m**
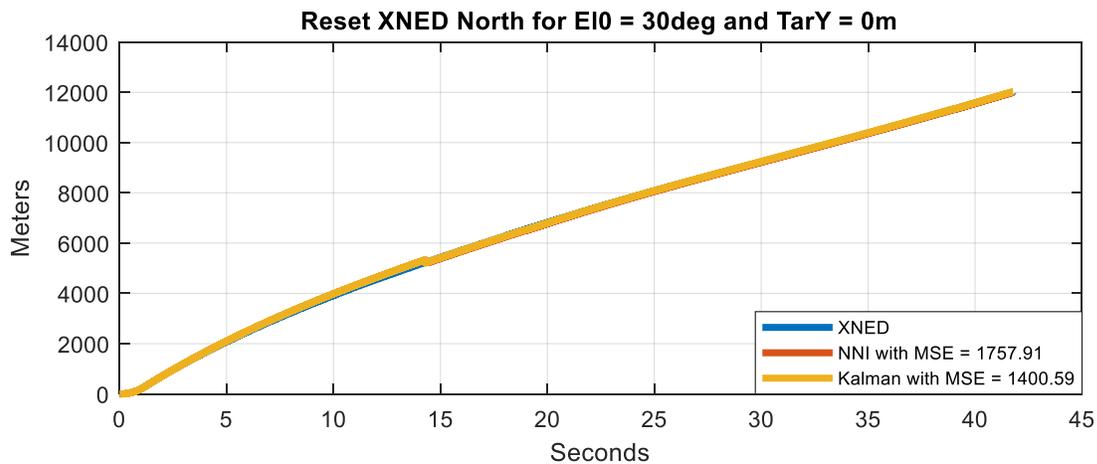
Figure 345 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 0m

**Reset XNED North for El0 = 30deg and TarY = 2000m**
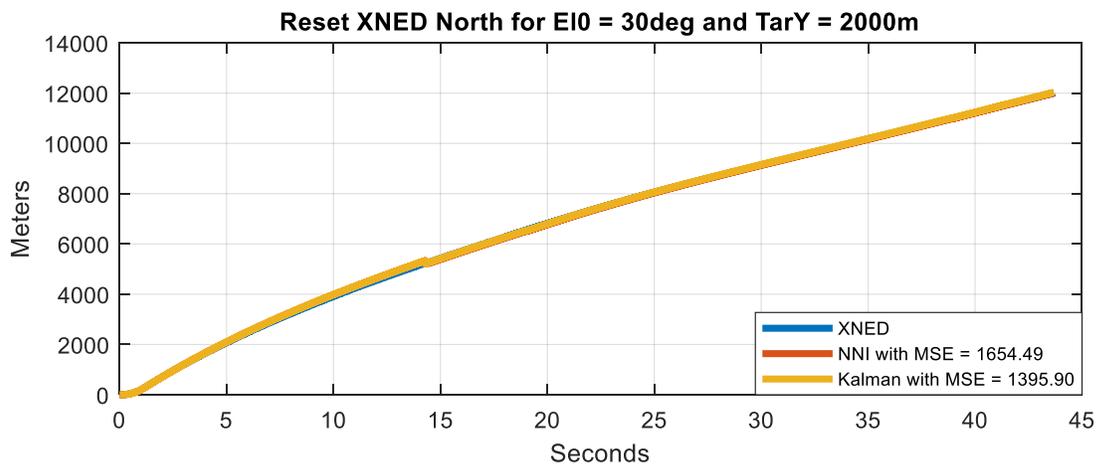
Figure 346 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 2000m

Figure *347* and Figure *348* show the comparison of north position GPS/INS integration in NED frame with reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman algorithm performed the best for both cases with the least MSE. Nevertheless, the NNI performed fine as well. It is an interesting phenomenon because with the No Reset case NNI performed better. It is important to note that the deviation did not affect the performance much.
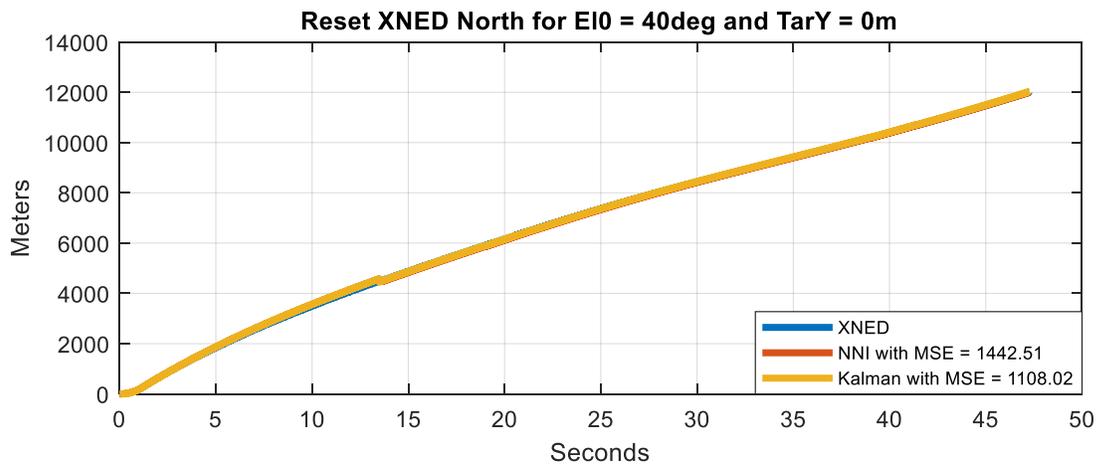


Figure 347 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 0m
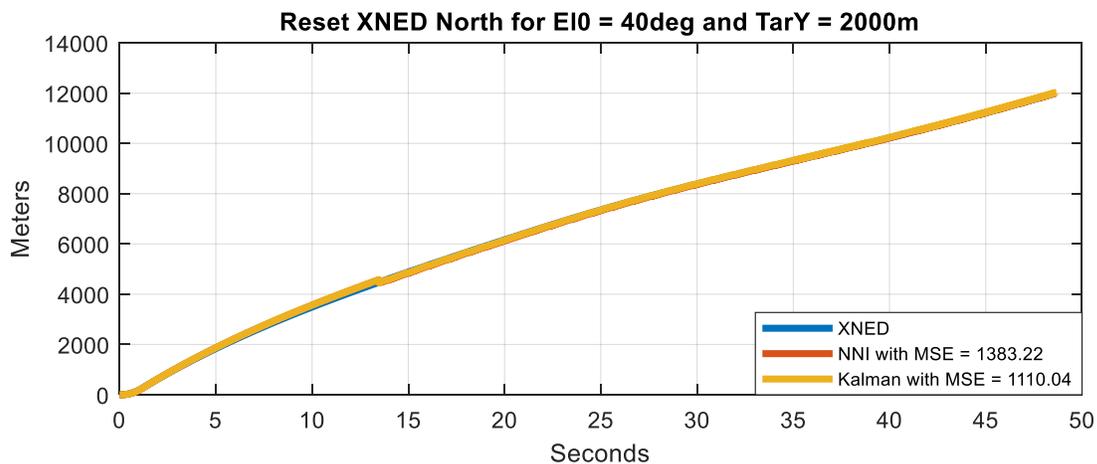


Figure 348 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 2000m

Figure *349* and Figure *350* show the comparison of north position GPS/INS integration in NED frame with reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman algorithm performed the best for both cases with the least MSE. Nevertheless, the NNI performed fine as well. It is an interesting phenomenon because with the No Reset case NNI performed better. It is important to note that the deviation did not affect the performance much.
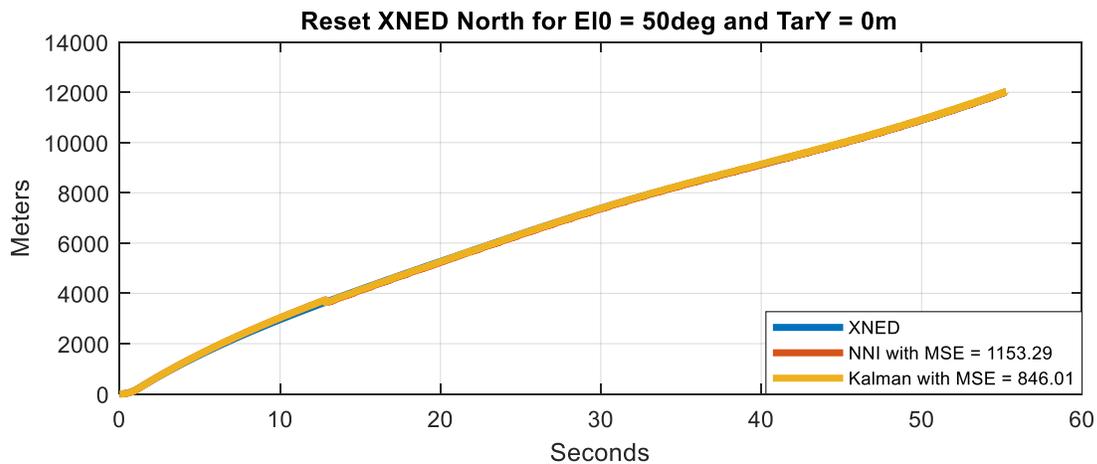


Figure 349 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 0m
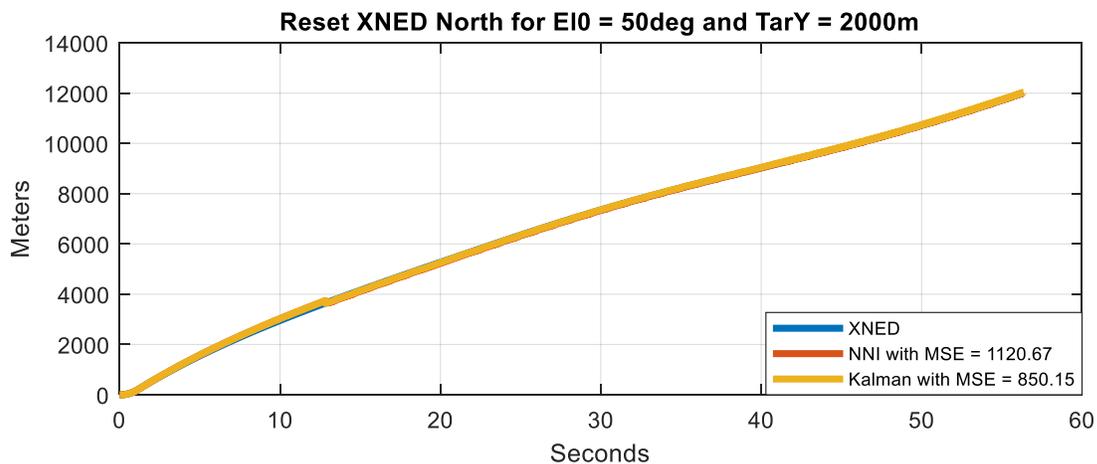


Figure 350 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 2000m

Figure *351* and Figure *352* show the comparison of north position GPS/INS integration in NED frame with reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman algorithm performed the best for both cases with the least MSE. Nevertheless, the NNI performed fine as well. It is an interesting phenomenon because with the No Reset case NNI performed better. It is important to note that the deviation did not affect the performance much.
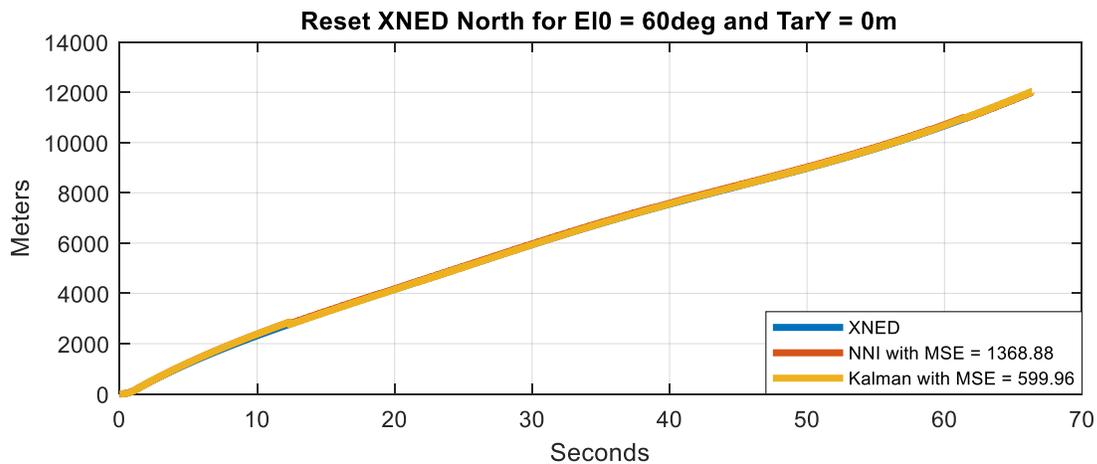


Figure 351 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 0m
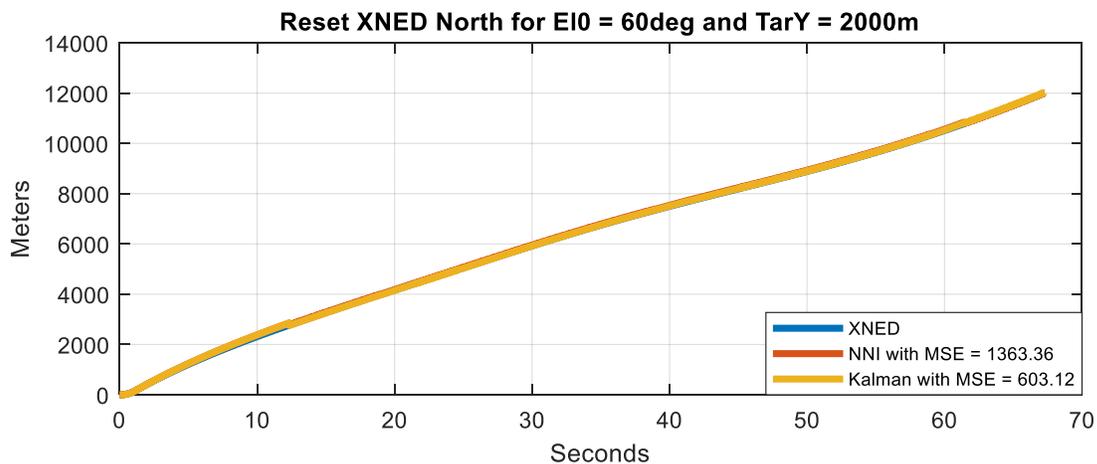


Figure 352 – Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 2000m

Figure *353* and Figure *354* show the comparison of north position GPS/INS integration in NED frame with reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman algorithm performed the best for both cases with the least MSE. Nevertheless, the NNI performed fine as well. It is an interesting phenomenon because with the No Reset case NNI performed better. It is important to note that the deviation did not affect the performance much.
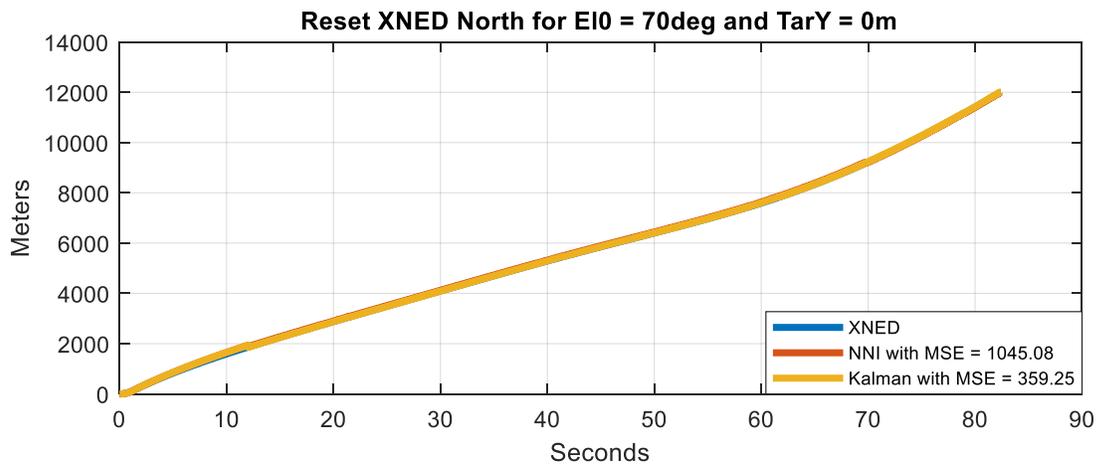


Figure 353 - Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 0m
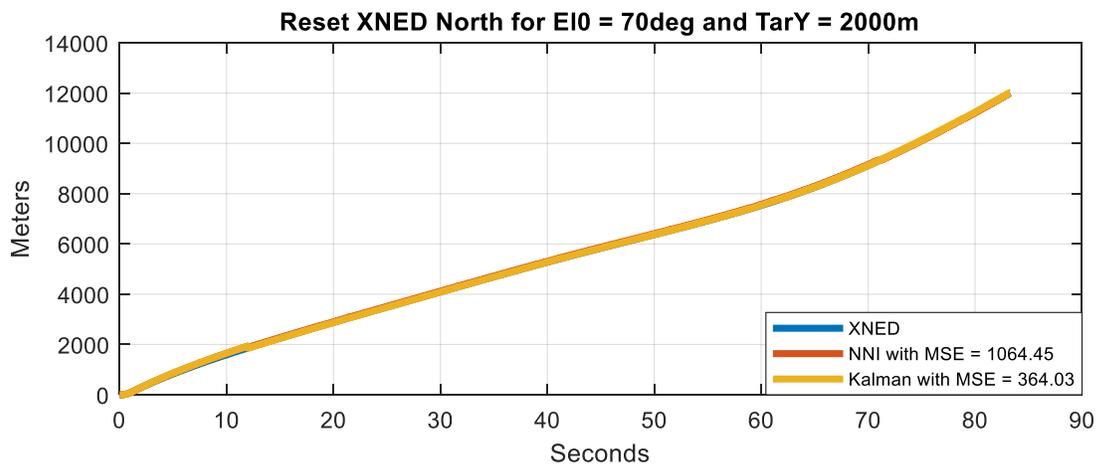


Figure 354 - Comparison of North Position GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 2000m

Figure *355* and Figure *356* show the comparison of down velocity GPS/INS integration in NED frame with reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman algorithm performed the best for both cases with the least MSE. Nevertheless, the NNI performed fine as well. It is important to note that the deviation did not affect the performance much.
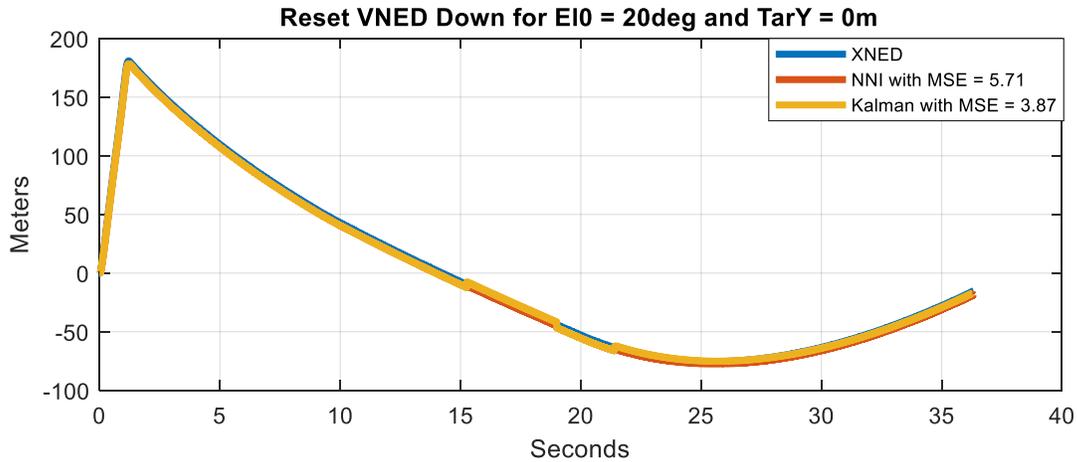


Figure 355 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 0m
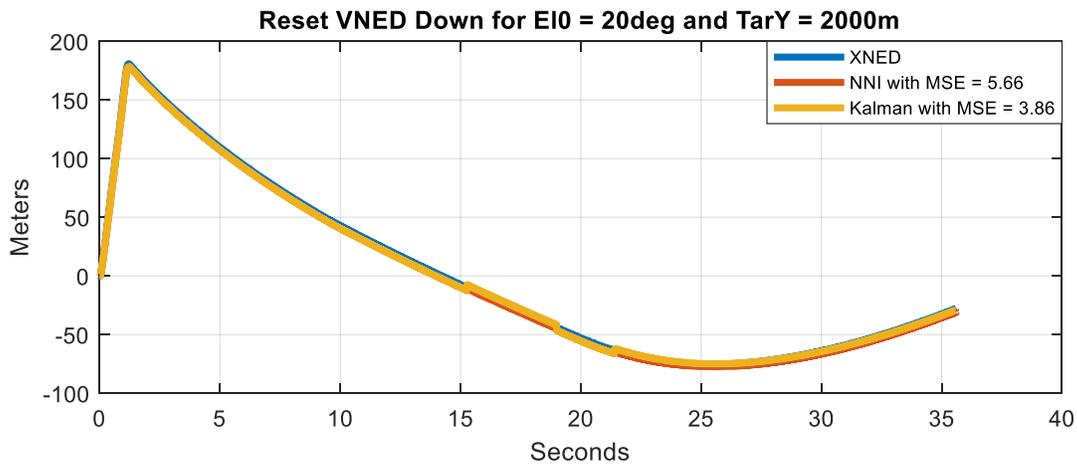


Figure 356 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 2000m

Figure *357* and Figure *358* show the comparison of down velocity GPS/INS integration in NED frame with reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
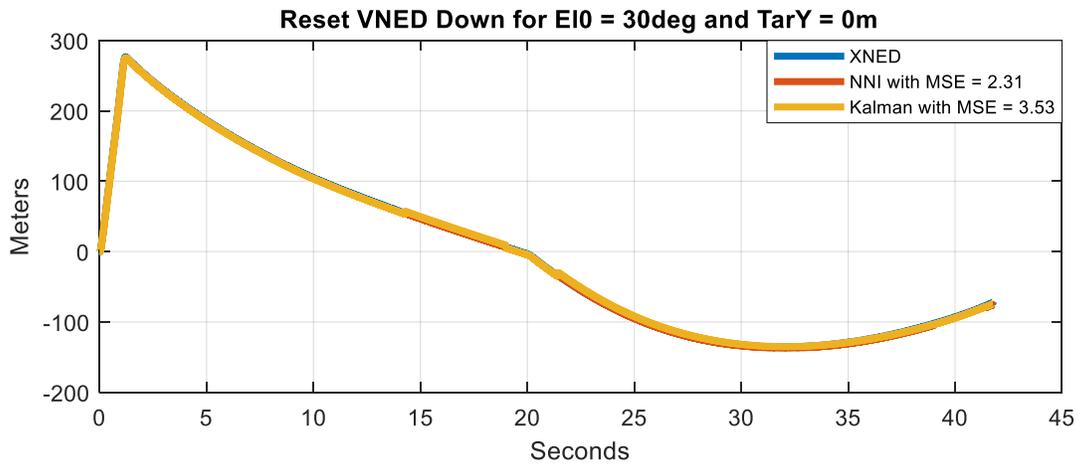


Figure 357 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 0m
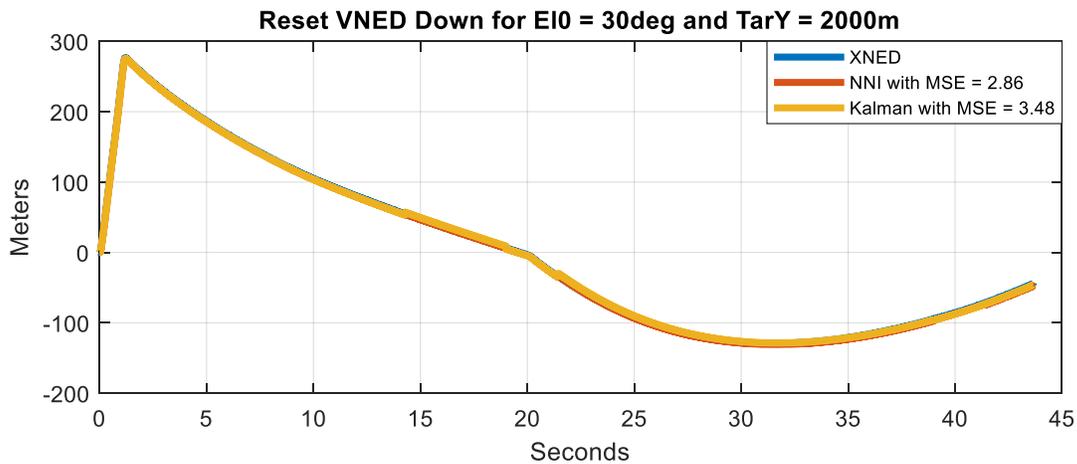


Figure 358 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 2000m

Figure *359* and Figure *360* show the comparison of down velocity GPS/INS integration in NED frame with reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
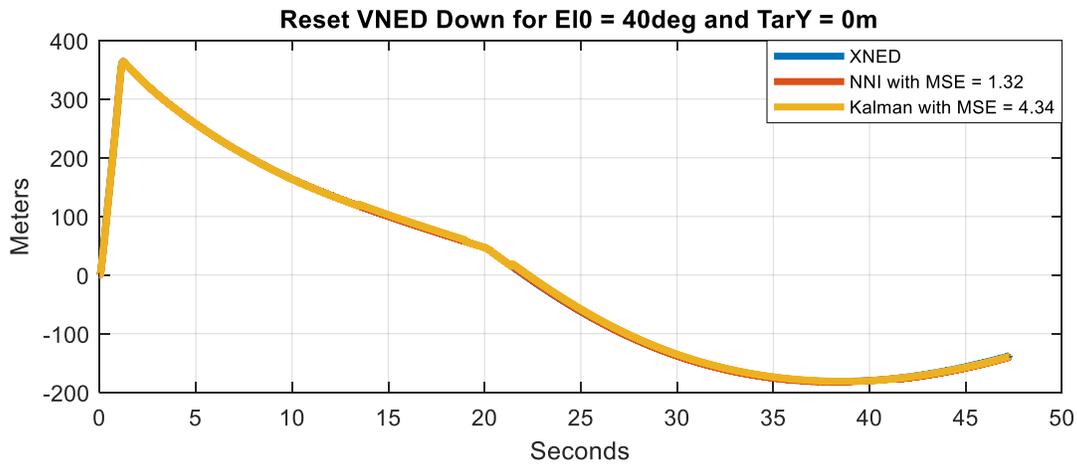


Figure 359 - Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 0m
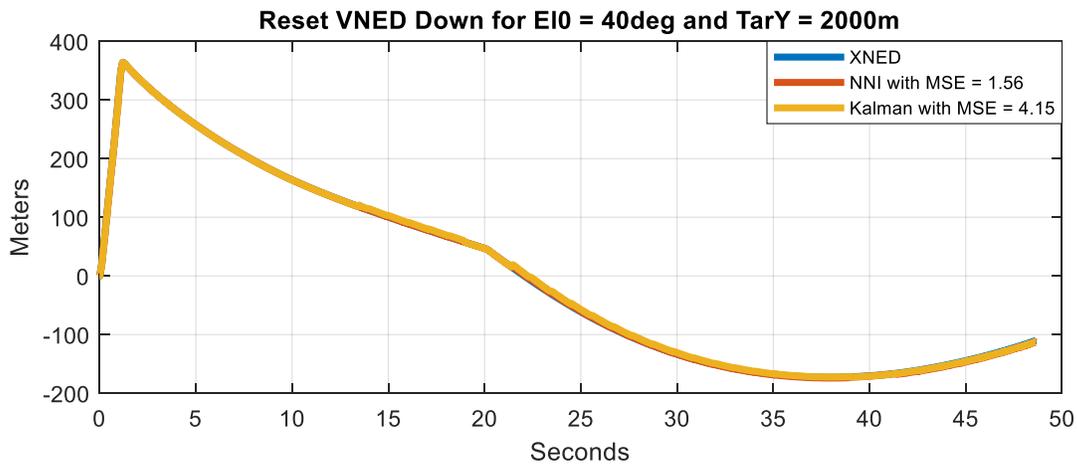


Figure 360 - Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 2000m

Figure *361* and Figure *362* show the comparison of down velocity GPS/INS integration in NED frame with reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
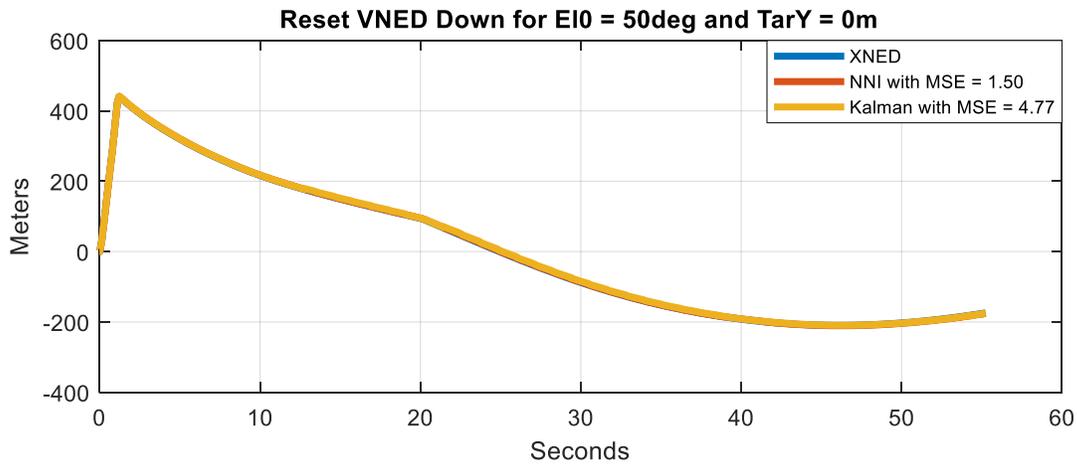


Figure 361 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 0m
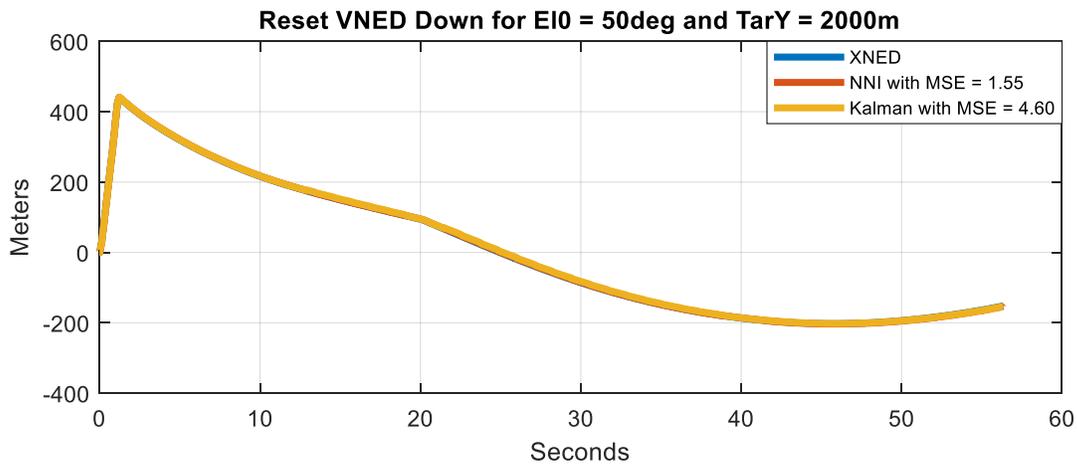


Figure 362 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 2000m

Figure *363* and Figure *364* show the comparison of down velocity GPS/INS integration in NED frame with reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
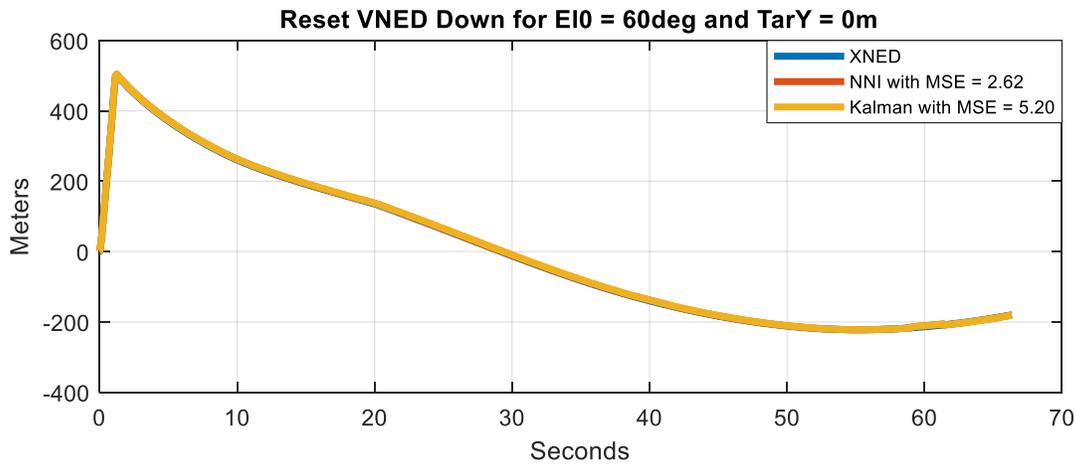


Figure 363 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 0m
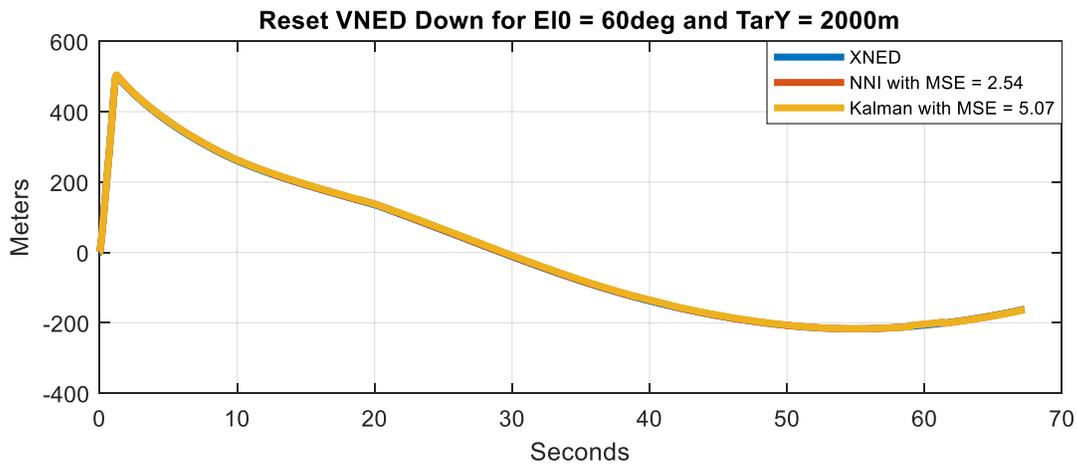


Figure 364 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 2000m

Figure *365* and Figure *366* show the comparison of down velocity GPS/INS integration in NED frame with reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
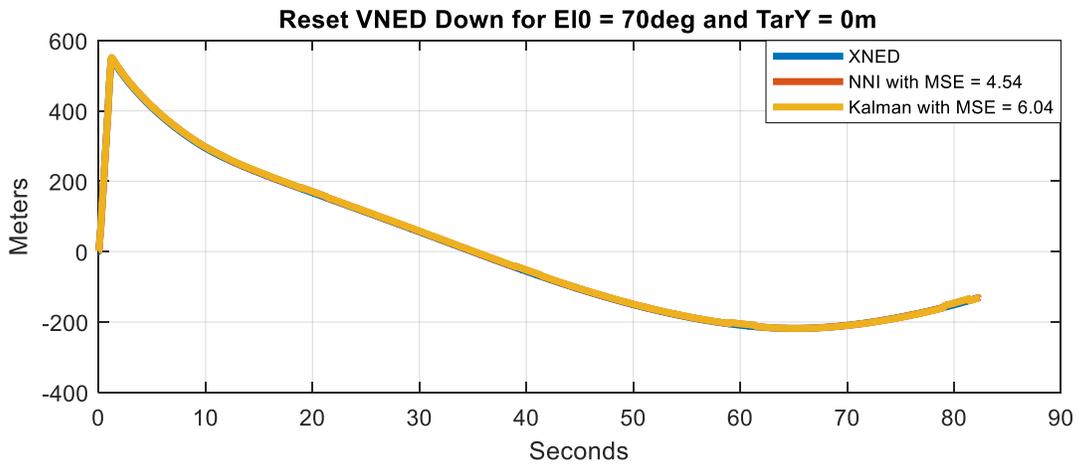


Figure 365 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 0m
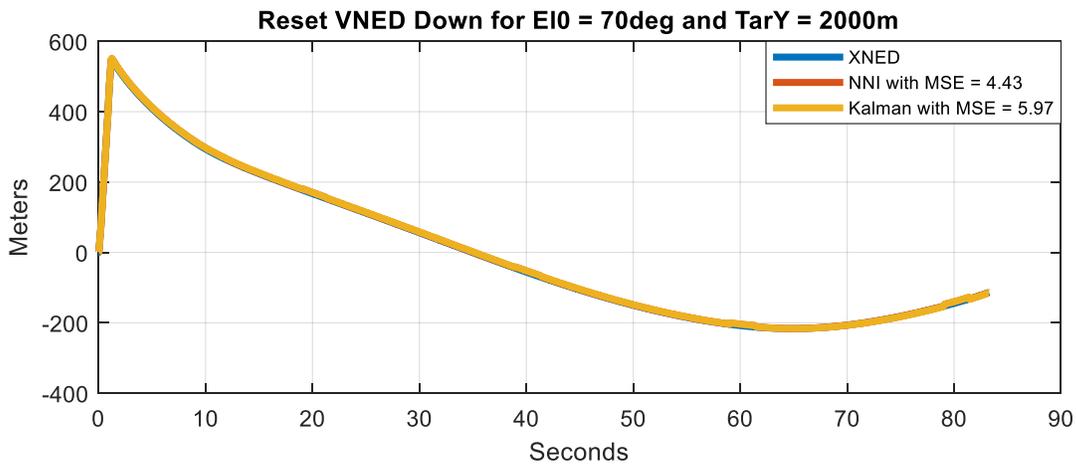


Figure 366 – Comparison of Down Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 2000m

Figure *367* and Figure *368* show the comparison of east velocity GPS/INS integration in NED frame with reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.



**Reset VNED East for El0 = 20deg and TarY = 0m**

Legend:
- XNED
- NNI with MSE = 9.94
- Kalman with MSE = 9.50
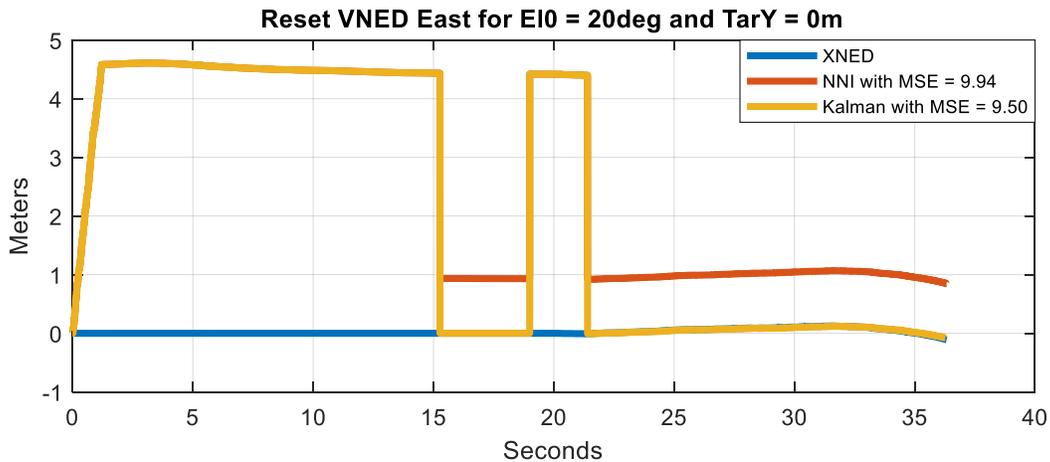
Figure 367 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 0m



**Reset VNED East for El0 = 20deg and TarY = 2000m**

Legend:
- XNED
- NNI with MSE = 9.93
- Kalman with MSE = 10.83
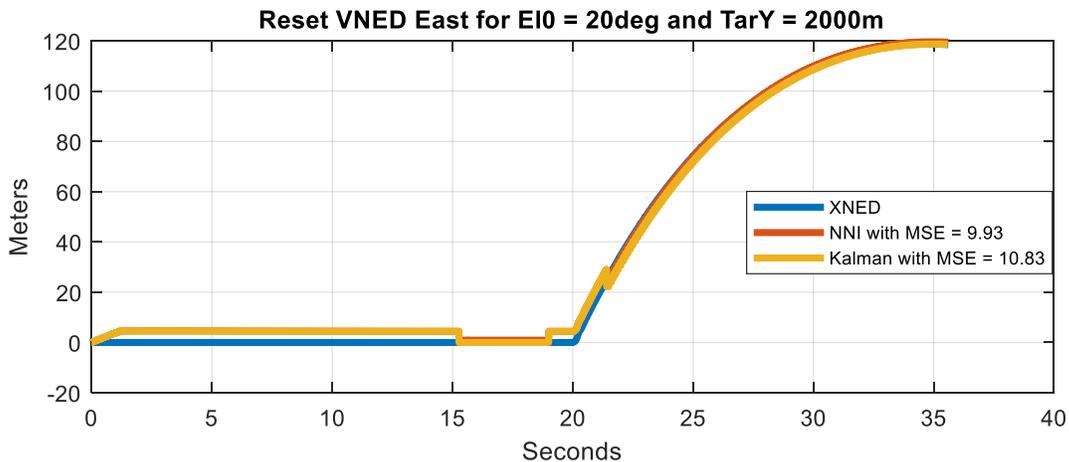
Figure 368 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 2000m

Figure *369* and Figure *370* show the comparison of east velocity GPS/INS integration in NED frame with reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.
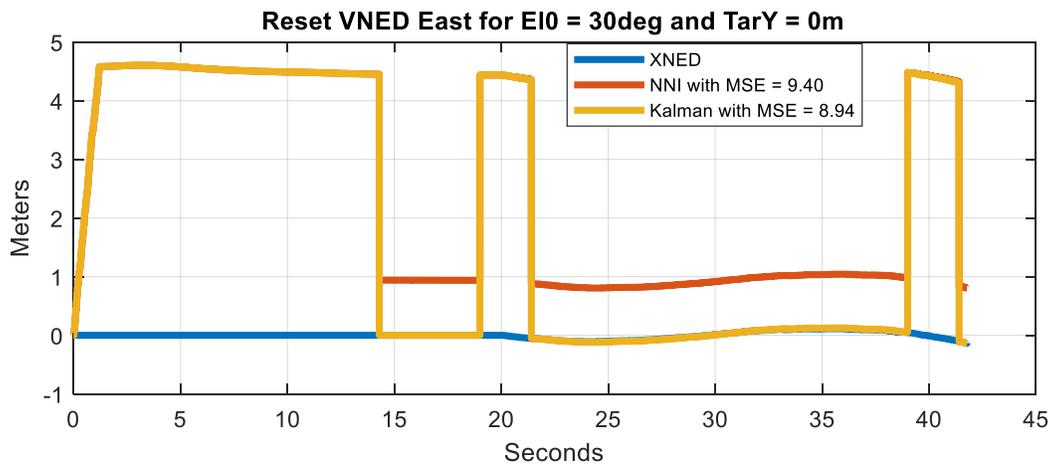


Figure 369 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 0m
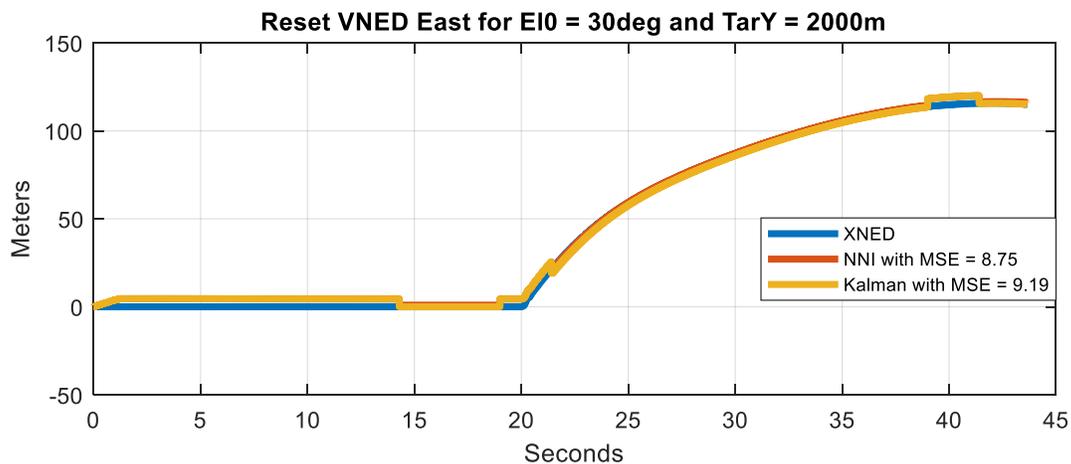


Figure 370 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 2000m

Figure *371* and Figure *372* show the comparison of east velocity GPS/INS integration in NED frame with reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for the case with no deviation. On the other hand, the NNI performed better in the deviation case. That is because the NNI has challenges with zero input for no deviation. Nevertheless, the NNI performed close to the Kalman filter.
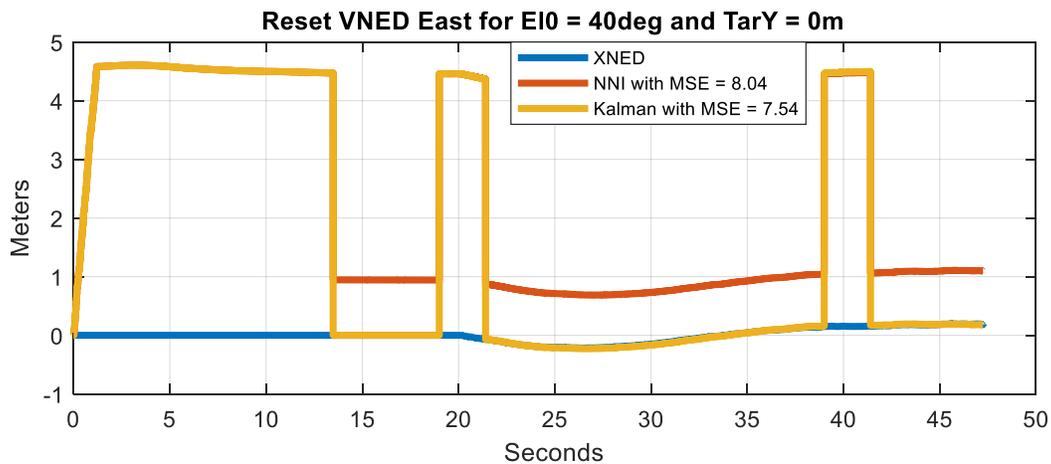


Figure 371 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 0m
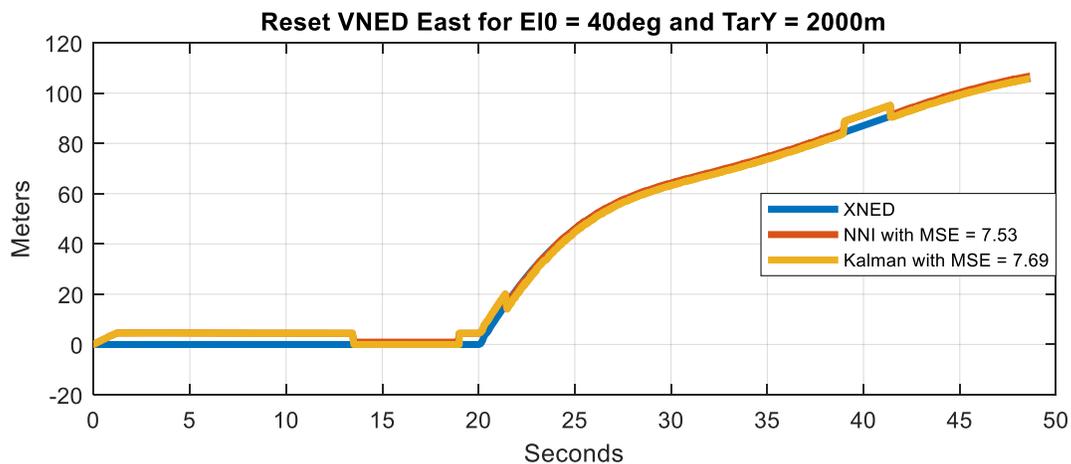


Figure 372 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 2000m

Figure *373* and Figure *374* show the comparison of east velocity GPS/INS integration in NED frame with reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for both cases. Nevertheless, the NNI performed fine and close to the Kalman filter.
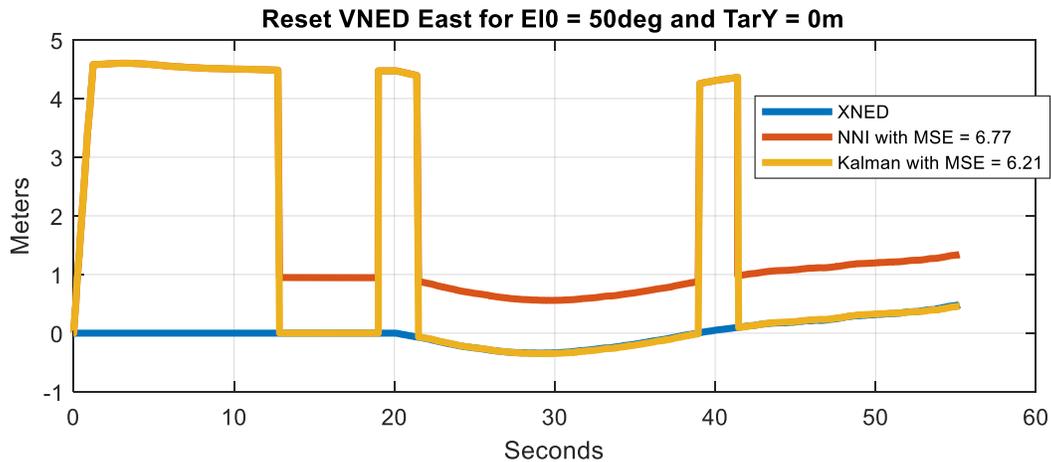


Figure 373 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 0m
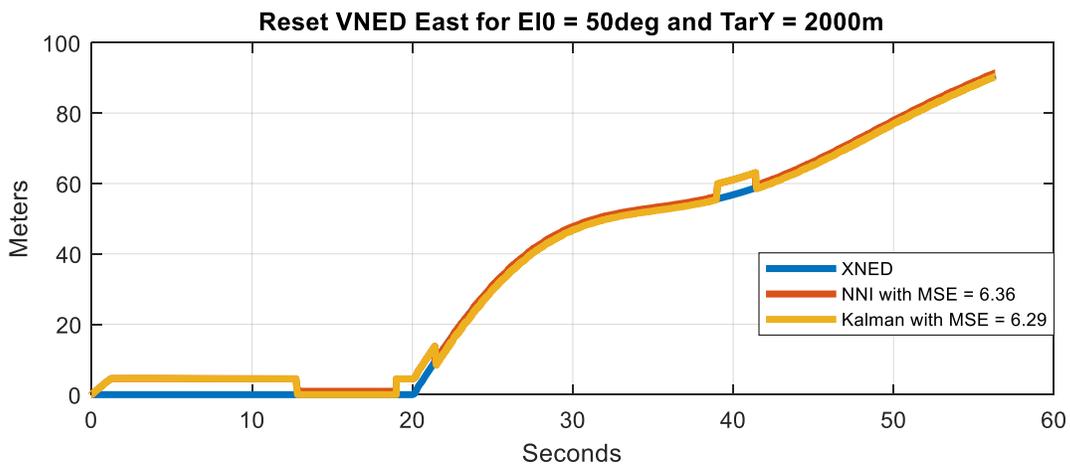


Figure 374 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 2000m

Figure *375* and Figure *376* show the comparison of east velocity GPS/INS integration in NED frame with reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for both cases. Nevertheless, the NNI performed fine and close to the Kalman filter.
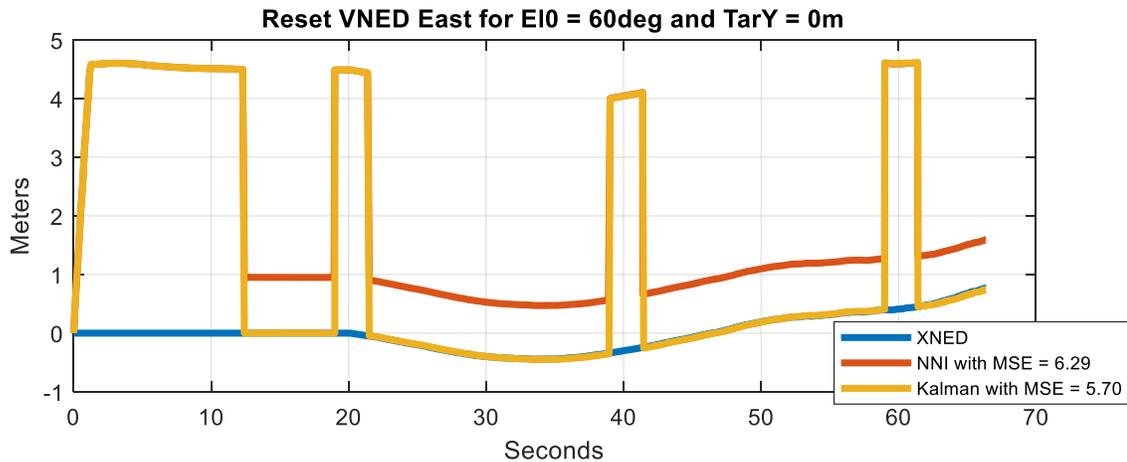


Figure 375 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 0m
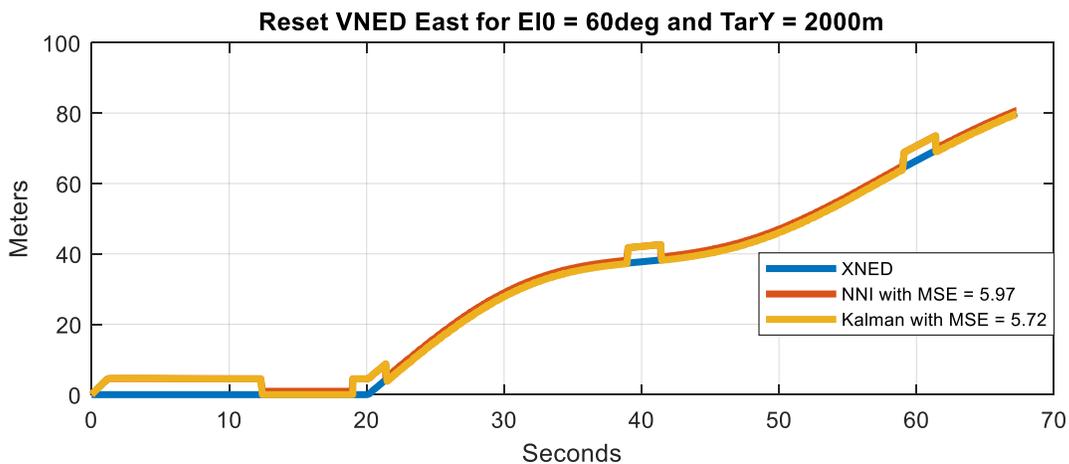


Figure 376 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 2000m

Figure *377* and Figure *378* show the comparison of east velocity GPS/INS integration in NED frame with reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for both cases. Nevertheless, the NNI performed fine and close to the Kalman filter.
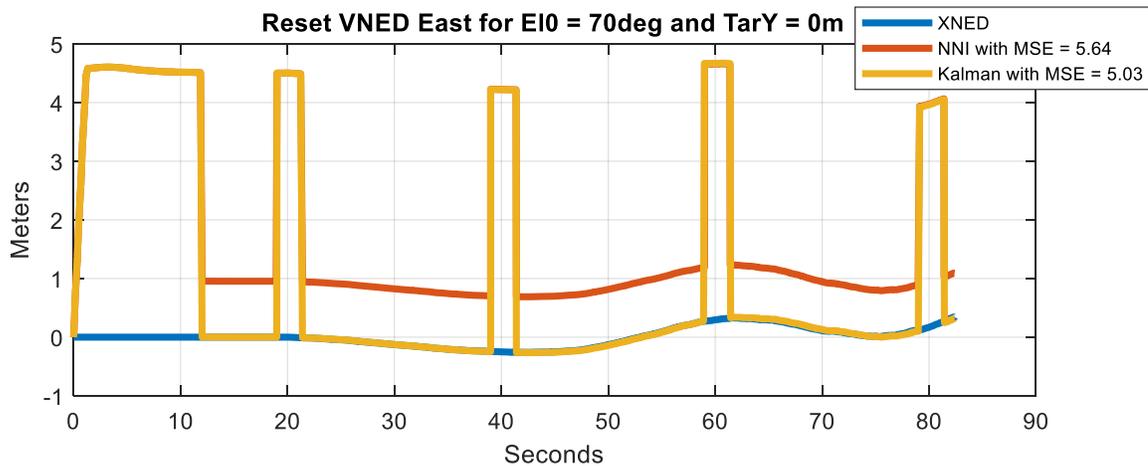


Figure 377 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 0m
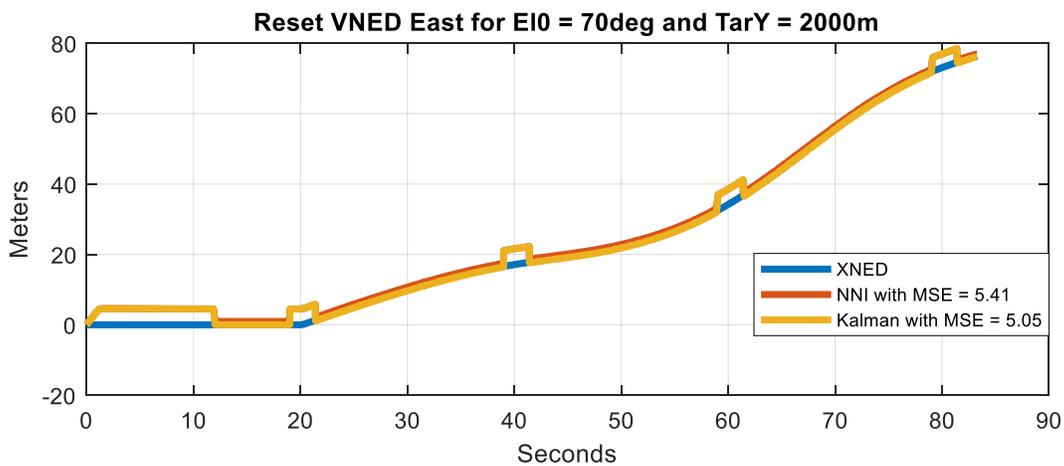


Figure 378 – Comparison of East Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 2000m

Figure *379* and Figure *380* show the comparison of north velocity GPS/INS integration in NED frame with reset for the missile launched at 20deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
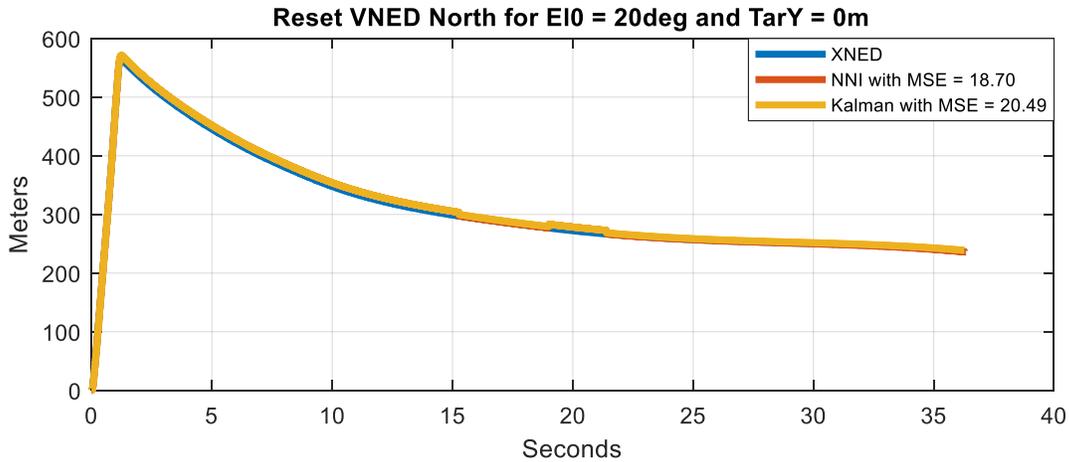


Figure 379 – Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 0m
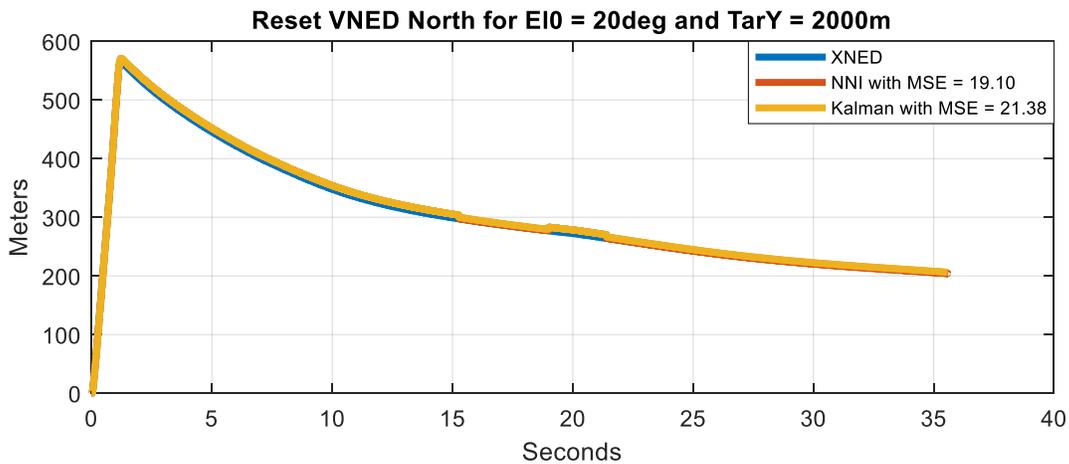


Figure 380 – Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 20 deg and Target Y = 2000m

Figure *381* and Figure *382* show the comparison of north velocity GPS/INS integration in NED frame with reset for the missile launched at 30deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
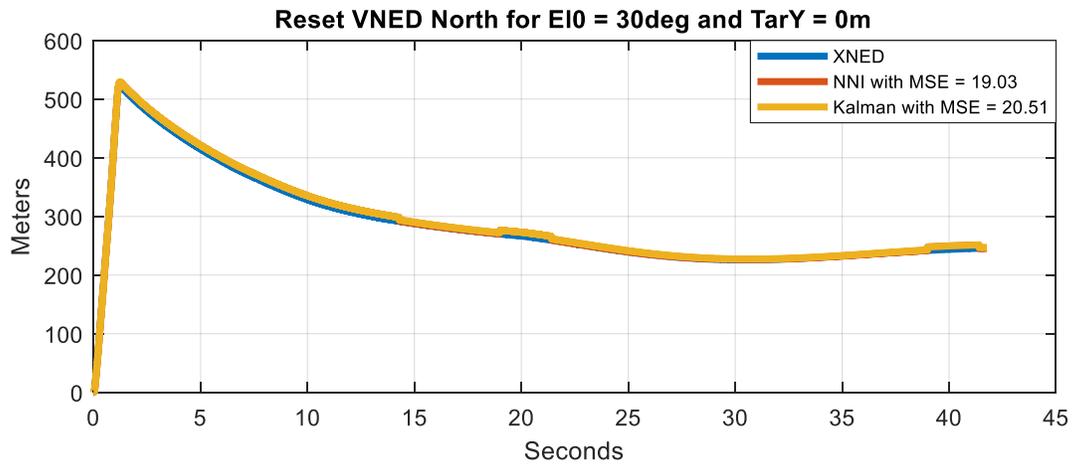


Figure 381 - Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 0m
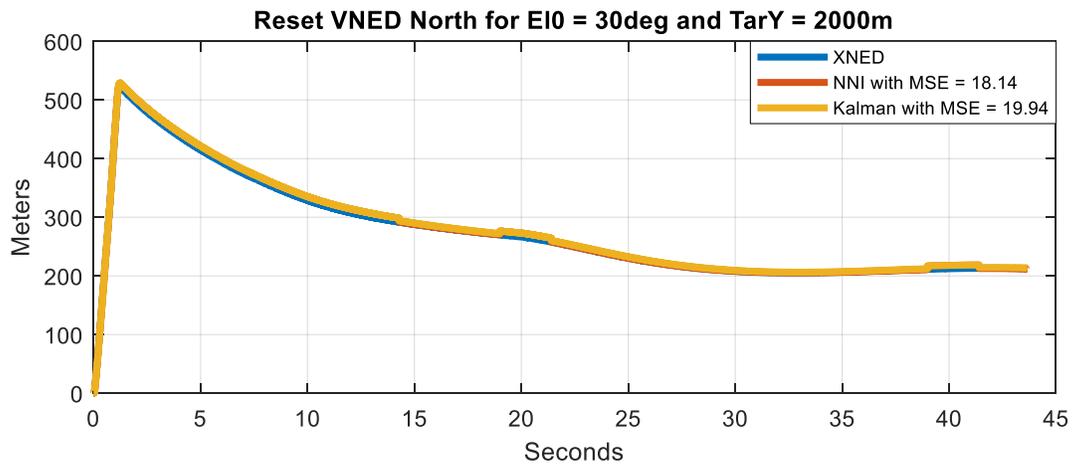


Figure 382 - Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 30 deg and Target Y = 2000m

Figure *383* and Figure *384* show the comparison of north velocity GPS/INS integration in NED frame with reset for the missile launched at 40deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
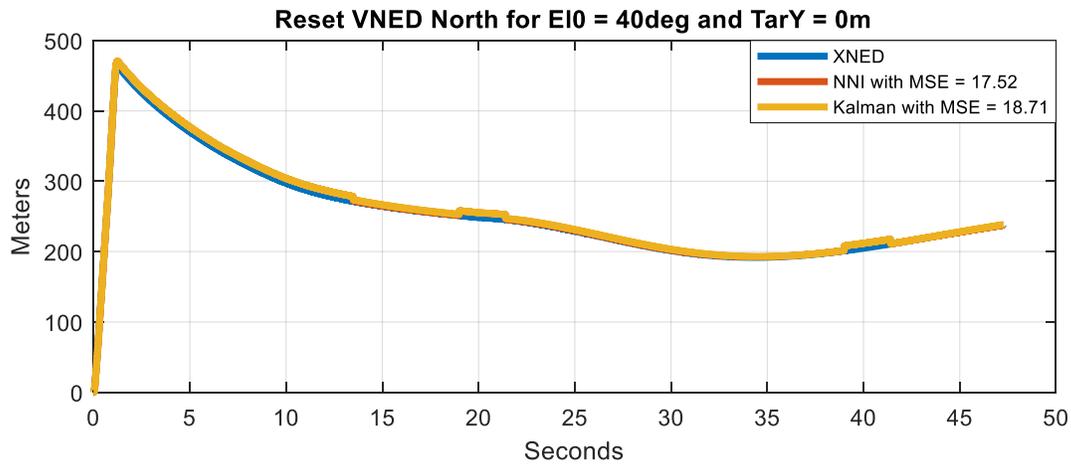


Figure 383 - Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 0m
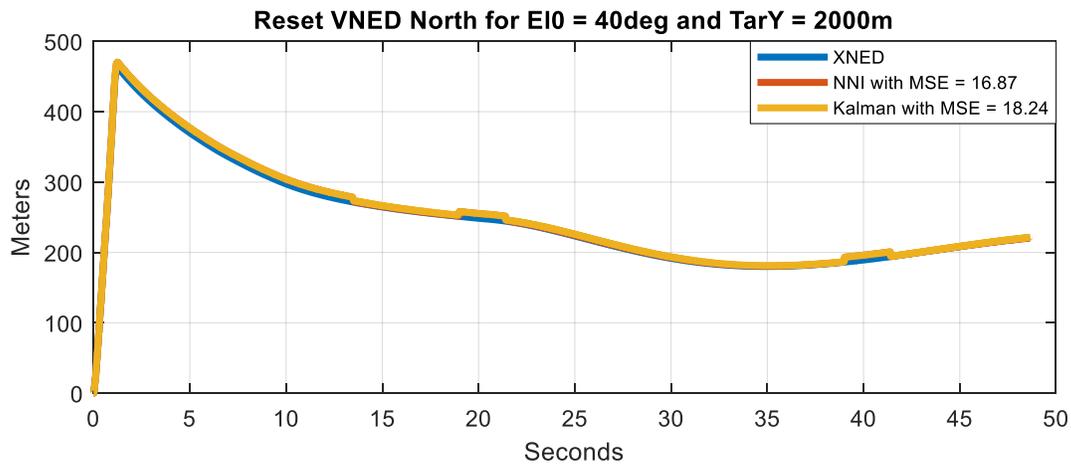


Figure 384 - Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 40 deg and Target Y = 2000m

Figure *385* and Figure *386* show the comparison of north velocity GPS/INS integration in NED frame with reset for the missile launched at 50deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
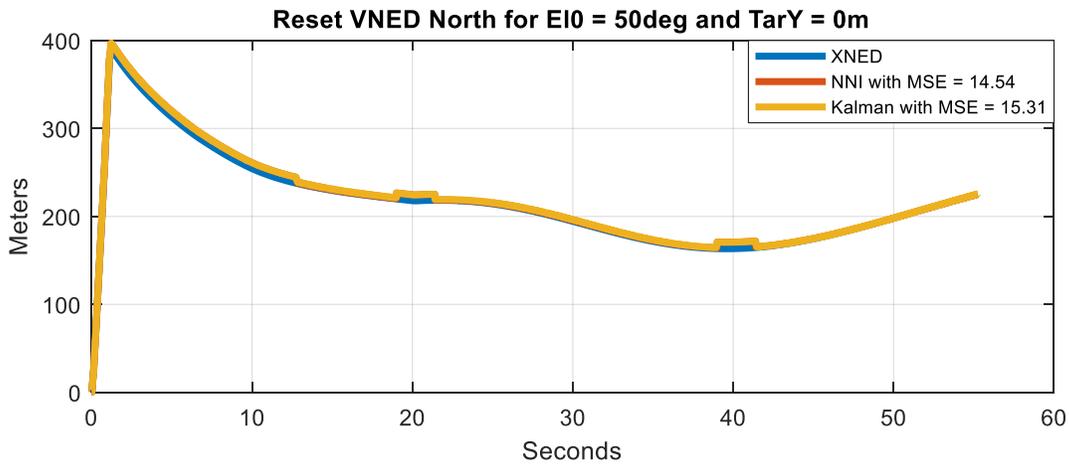


Figure 385 - Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 0m
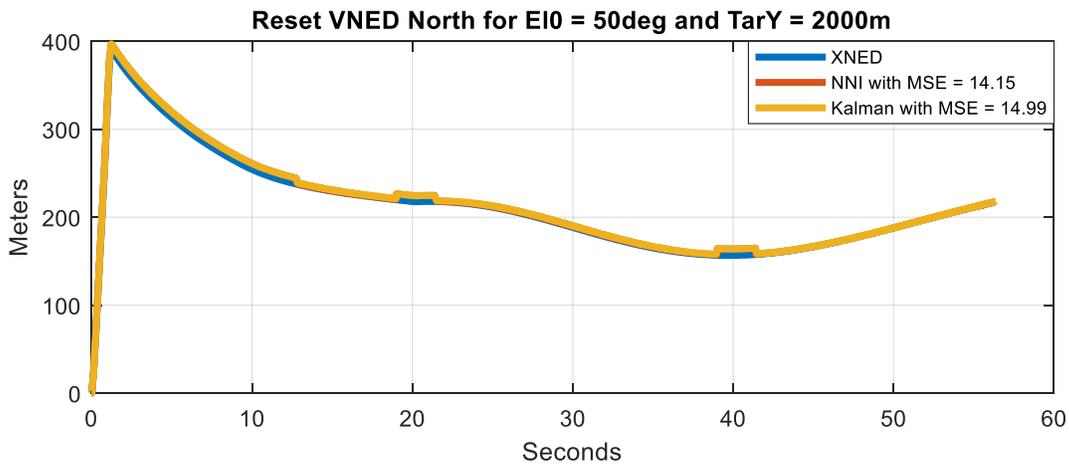


Figure 386 - Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 50 deg and Target Y = 2000m

Figure *387* and Figure *388* show the comparison of north velocity GPS/INS integration in NED frame with reset for the missile launched at 60deg and target located at 0m and 2000m deviation. Here, it can be seen that the NNI algorithm performed the best for both cases with the least MSE. Nevertheless, the Kalman filter performed fine as well. It is important to note that the deviation did not affect the performance much.
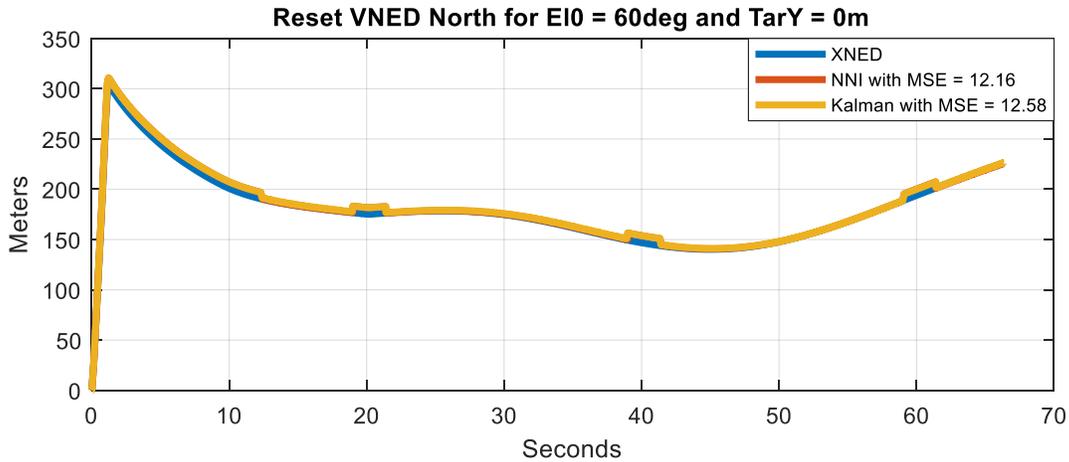


Figure 387 - Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 0m
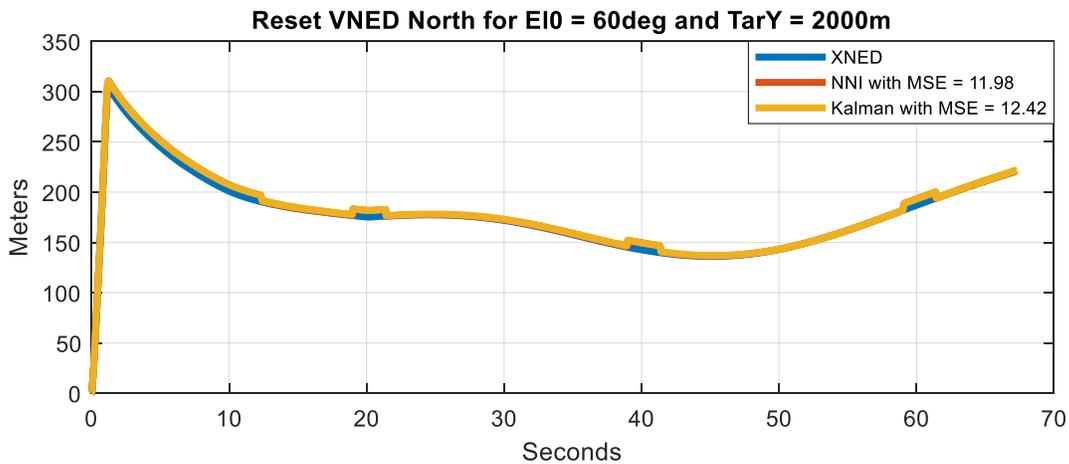


Figure 388 - Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 60 deg and Target Y = 2000m

Figure *389* and Figure *390* show the comparison of north velocity GPS/INS integration in NED frame with reset for the missile launched at 70deg and target located at 0m and 2000m deviation. Here, it can be seen that the Kalman filter algorithm performed the best for both cases with the least MSE. Nevertheless, the NNI performed fine as well. It is important to note that the deviation did not affect the performance much.
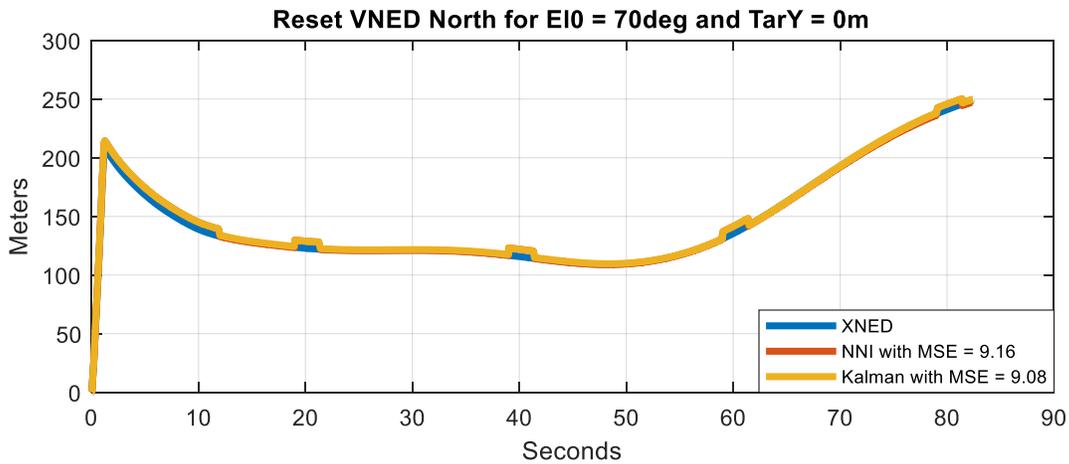


Figure 389 – Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 0m
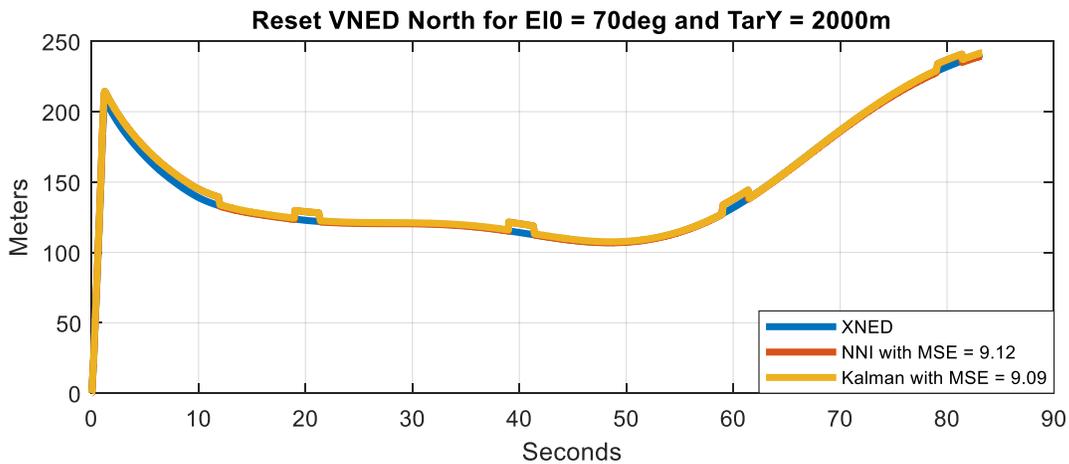


Figure 390 – Comparison of North Velocity GPS/INS Integration in NED Frame Between Kalman and NNI With Reset for El0 = 70 deg and Target Y = 2000m

Figure 391 shows a comparison of the histogram distribution of down position GPS/INS integration in NED frame with reset between NNI and Kalman filter. It can be seen that Kalman filter has the higher mean. This means that it is less accurate overall. In addition, the NNI has the lower mean. This means that it is more accurate. Since the difference between the standard deviation of the Kalman filter and NNI is small, it can be assumed that they are equally precise.



**Reset XNED Down Historgram**

Legend:
- NNI with σ = 231.41 and μ = 611.68
- Kalman with σ = 231.05 and μ = 619.73
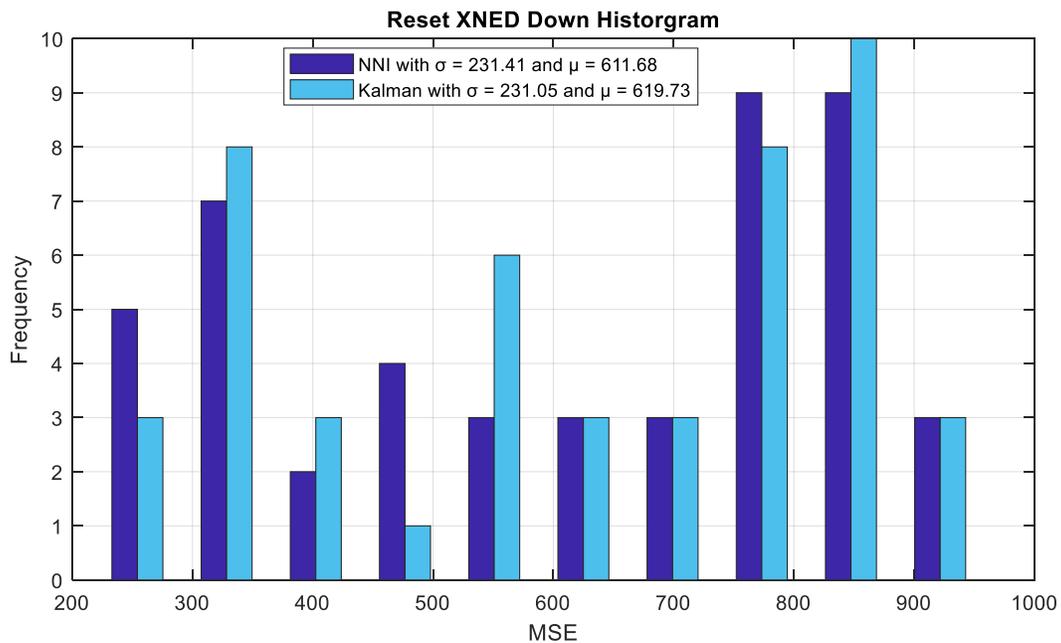
Figure 391 – Comparison of The Histogram Distribution for Down Position GPS/INS Integration in NED Frame With Reset Between NNI and Kalman Filter.

Figure 392 shows a comparison of the histogram distribution of east position GPS/INS integration in NED frame with reset between NNI and Kalman filter. It can be seen that NNI has the higher mean. This means that it is less accurate overall. This could be because of the challenges the NNI has with zero input for the no deviation case. In addition, the Kalman filter has the lower mean. This means that it is more accurate. Since the standard deviation of the NNI is smaller than the Kalman filter, it is more precise.
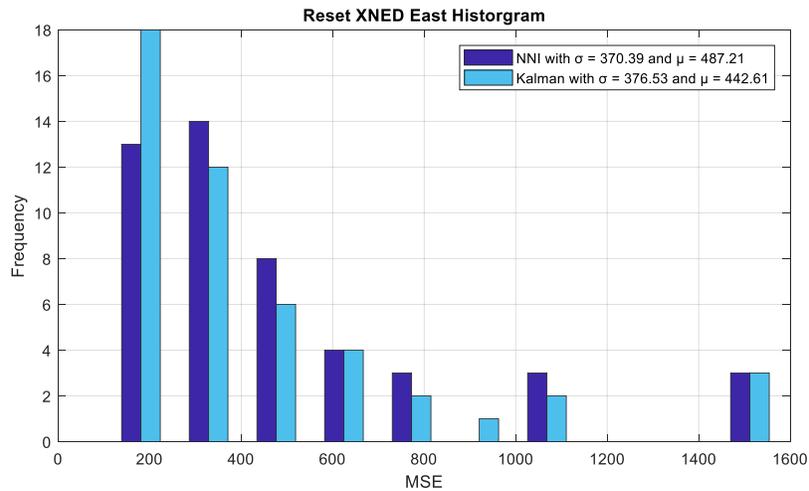
Figure 392 – Comparison of The Histogram Distribution for East Position GPS/INS Integration in NED Frame With Reset Between NNI and Kalman Filter.

Figure 393 shows a comparison of the histogram distribution of north position GPS/INS integration in NED frame with reset between NNI and Kalman filter. It can be seen that NNI has the higher mean. This means that it is less accurate overall. In addition, the Kalman filter has the lower mean. This means that it is more accurate. Since the standard deviation of the NNI is lower than the Kalman filter, it is more precise.
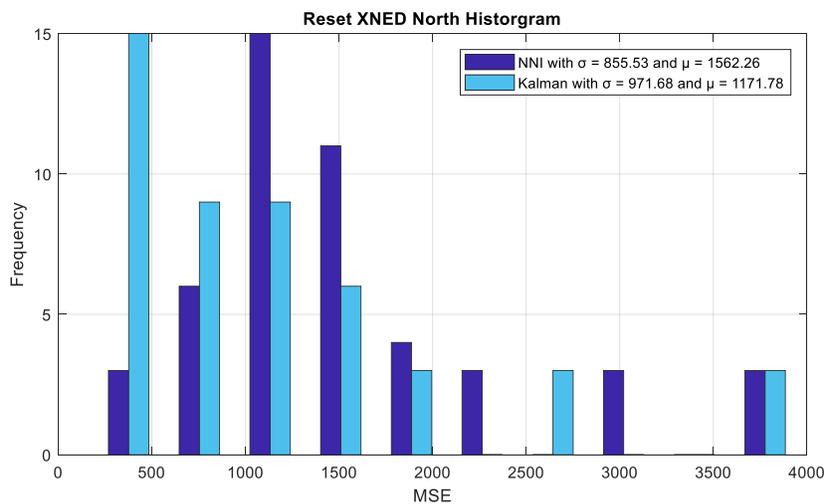


Figure 393 – Comparison of The Histogram Distribution for North Position GPS/INS Integration in NED Frame With Reset Between NNI and Kalman Filter.

Figure 394 shows a comparison of the histogram distribution of down velocity GPS/INS integration in NED frame with reset between NNI and Kalman filter. This means that it is less accurate overall. In addition, the NNI has the lower mean. This means that it is more accurate. Since the standard deviation of the Kalman filter is lower than the NNI, it is more precise.
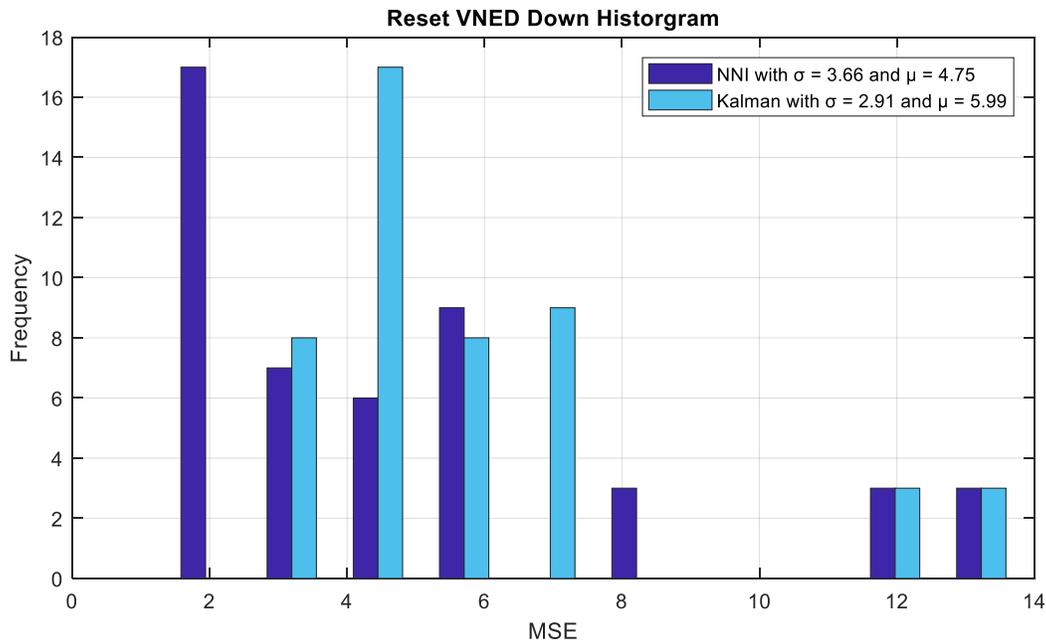


Figure 394 – Comparison of The Histogram Distribution for Down Velocity GPS/INS Integration in NED Frame With Reset Between NNI and Kalman Filter.

Figure 395 shows a comparison of the histogram distribution of east velocity GPS/INS integration in NED frame with reset for between NNI and Kalman filter. It can be seen that NNI has the higher mean. This means that it is less accurate overall. In addition, the Kalman filter has the lower mean. This means that it is more accurate. Since the standard deviation of the NNI is lower than the Kalman filter, it is more precise.

**Reset VNED East Histogram**
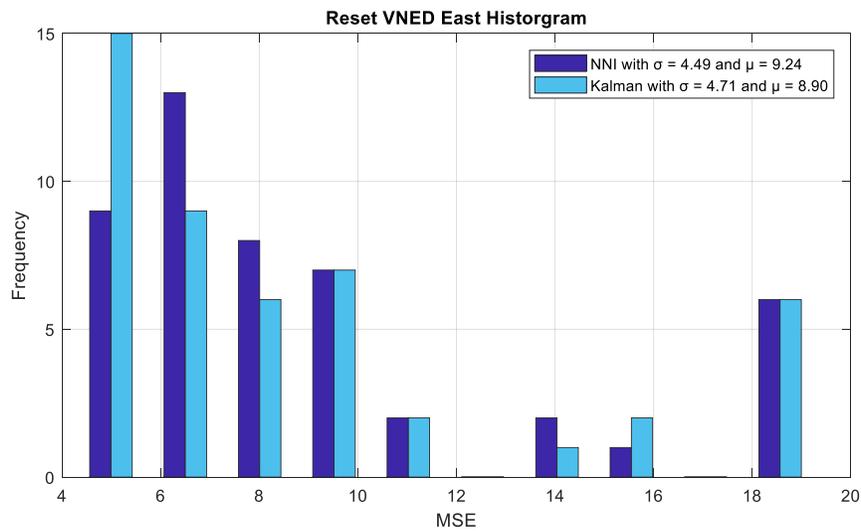
Figure 395 – Comparison of The Histogram Distribution for East Velocity GPS/INS Integration in NED Frame With Reset Between NNI and Kalman Filter.

Figure 396 shows a comparison of the histogram distribution of north velocity GPS/INS integration in NED frame with reset between NNI and Kalman filter. It can be seen that Kalman filter has the higher mean. This means that it is less accurate overall. In addition, the NNI has the lower mean. This means that it is more accurate. Since the standard deviation of the NNI is lower than the Kalman filter, it is more precise.



**Reset VNED North Histogram**
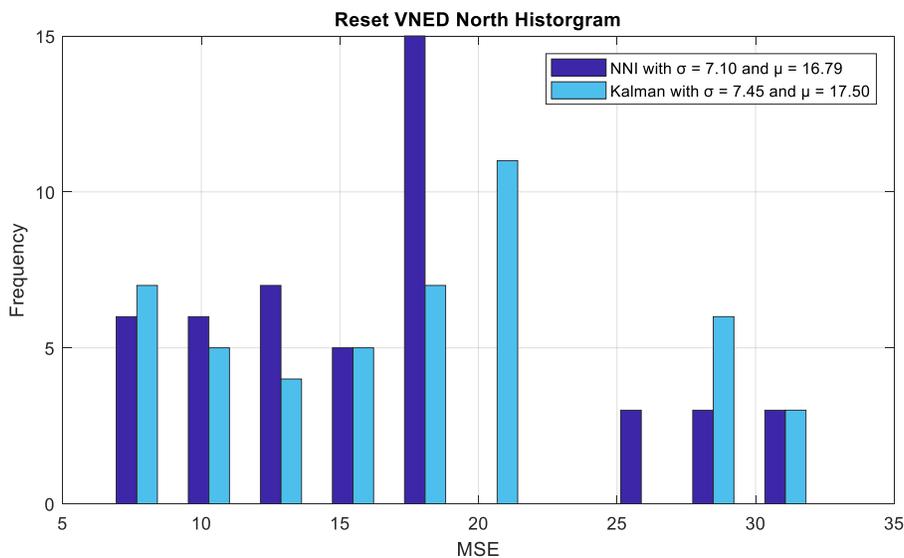
Figure 396 – Comparison of The Histogram Distribution for North Velocity GPS/INS Integration in NED Frame With Reset Between NNI and Kalman Filter.

# 8. Conclusion

This thesis investigated the possibility of using machine learning to design missile guidance, navigation, and control algorithms. This was accomplished by modeling the missile and its environment. In addition, the missile system and traditional guidance, navigation, and control algorithms were modeled. This allowed for a benchmark to test the algorithms designed by machine learning. It is important to keep in mind that the thesis assumes that the simulation is accurate enough to represent a physical missile system.

Proportional navigation was used as the benchmark for the terminal guidance. The Terminal Guidance Neural Network algorithm was developed using machine learning to perform the task of terminal guidance. The TGNN algorithm resulted in a more accurate and precise terminal guidance algorithm than traditional PN. In fact, when pushing the limits of both algorithms, the TGNN algorithm resulted in significant improvements over the PN. This means that the TGNN is a viable if not a better alternative to the traditional PN algorithm.

In addition, a modified version of the roll, pitch, and yaw PID controller was used as the benchmark for the missile autopilot. The missile was tested for 15deg and 20deg maximum actuator deflection. In both cases, the Roll Neural Network Autopilot controller was more precise than the PID controller. On the other hand, the Roll PID controller was more accurate. Additionally, the Pitch Lateral Neural Network Autopilot controller was more precise and accurate than the Pitch PID controller. Moreover, the Yaw Lateral Neural Network Autopilot controller was more precise. However, the Yaw PID controller was more accurate. This means that the Neural Network Autopilot algorithms designed by machine learning is also a viable alternative if not a better alternative to the traditional PID controller.

Furthermore, the Kalman filter was used as the benchmark for the GPS/INS integration. The integration was tested using No Reset and Reset methods. In the No Reset Method, both north and down position channels were more

accurate and precise on Neural Network Integration algorithm. However, because of the challenge with zero input condition, Kalman filter was more accurate and precise for the east position channel. The same behavior holds true for the NED velocities. However, the east velocity is more accurate on the NNI than the Kalman filter. On the other hand, the behavior changes with the Reset case. The Kalman filter is more accurate and precise for down position channel. Although the Kalman filter was more precise, the NNI was more accurate for the east and north position channels. The north velocity channel was more precise and accurate on the NNI algorithm. Nonetheless, the Kalman filter was more precise for the east velocity channel but less accurate. In addition, the NNI was more accurate on the down velocity channel but less accurate. This competitive behavior could be resulted from the fact that the NNI was trained only for the No Reset case despite being tested on the Reset case. Hence, with proper training, the NNI algorithm could be improved further. Since the NNI showed improvements on the No Reset case and was competitive with Kalman filter on the Reset case, the NNI algorithm is a viable alternative to the Kalman filter for GPS/INS integration.

The lesson learned from this thesis is that machine learning is a viable solution to design missile guidance, navigation, and control algorithm. However, in addition to its advantages in improving the performances of the traditional algorithms, it is important to keep in mind the time it took to design and train the solution. Since the training time ranged from several hours to several days, traditional algorithm might be a more practical solution. That is because in the industry where development is limited with time and budget, time needed to find an adequate solution might not be acceptable. However, this could be combated by using super computers and parallel processing in order to speed up the training process. Moreover, another lesson is that the neural network algorithm is a tool and not a magical algorithm that will solve any problem. It requires deep understanding of both the matter and the network design. The best results were achieved not when the neural networks were working alone, but when the neural network worked with

traditional mathematical model such as the case with TGNN where the line of sight rate was calculated. Lastly, it is difficult to predict generalization challenge of the neural network. Nevertheless, one solution for this problem is by allowing the network to interact with as many cases as possible so that it can learn to interact with changes.

The future step for this thesis is to test the neural network algorithm with physical systems. That is because despite the assumptions, the physical system behaves differently for the simulated one. That is because of all elements that has not been considered in the simulation. In addition, it is because mathematical models are not identical to the physical system and is limited to the understanding of the system. Additionally, the neural network guidance, navigation, and control algorithms can be further improved in order to increase its accuracy and design. For instance, the NNA can be improved to reduce oscillation and the NNI can be improved to work better with Reset case and solve the zero-input condition challenge. In conclusion, the field of research and development is a growing field where new challenges and opportunities present itself every day. Hence, there will always be wonderful opportunities to expand the knowledge and travel beyond the horizons.

# References

1. Granville, Vincent. Types of Machine Learning Algorithms in One Picture. Data Science Central. [Online] 07 27, 2017. [Cited: 08 13, 2018.] https://www.datasciencecentral.com/profiles/blogs/types-of-machine-learning-algorithms-in-one-picture.

2. Machine Learning in MATLAB. Mathworks. [Online] [Cited: 08 13, 2018.] https://www.mathworks.com/help/stats/machine-learning-in-matlab.html.

3. Samarasinghe, Sandhya. Neaural Network For Applied Sciences and Engineering. Boca Raton : Taylor & Francis Group, 2007.

4. Review Of Advances In Neural Networks: Neural Design Technology Stack. Almasi, Adela Diana, et al. 2016, ELSEVIER, pp. 31-41.

5. On the Levenberg-Marquardt Training Method for Feed-Forward Neural Networks. Liu, Hongwei. Beijing : s.n., 2010. 2010 Sixth International Conference on Natural Computation .

6. Goldberg, David E. Genetic Algorithm in Search, Optimization & Machine Learning. s.l. : Addison-Wesley Publishing Company, 1989.

7. Mathworks. Genetic Algorithm. [Online]

8. Mean Squared Error. [Online] Dec 2016.

9. Evolving Neural Networks Through Augmenting Topologies. Stanley, Kenneth O and Miikkulainen, Risto. 2002, The MIT Press Journals, pp. 99-127.

10. Forces and Gravity. Princeton Univeristy. [Online] [Cited: Feb 11, 2018.] http://wwwphy.princeton.edu/~steinh/ph115/Chapter03D.pdf.

11. Enchanced Model of One Axis-Two Positions Manual Tracking Photovoltaic Panel For Lighting Projects In Palestine. Hussein, Mohammed T. and Albarqouni, Shadi N. Alberta : s.n., 2010. Second LASTED Interational Conference. pp. 937-943.

12. Farrell, Jay A. Aided Navigation: GPS with Hight Rate Sensors. s.l. : McGraw-Hill, 2008.

13. Oliver, John E. Encyclopedia of World Climatology. Dordrecht : Springer, 2005.

14. Siouris, George M. Missile Guidance and Control Systems. New York : Springer, 2004.

15. Looney, Mark. The Basics of MEMS IMU/Gyroscope Alignment. [Online] [Cited: Feb 18, 2018.] http://www.analog.com/en/analog-dialogue/articles/mems-imu-gyroscope-alignment.html.

16. Salychev, Oleg S. Applied Inertial Navigation: Problems and Solution. Moscow : BMSTU Press, 2004.

17. Farrell, Jay A. and Barth, Matthew. The Global Positioning System & Inertial Navigation. s.l. : McGraw-Hill, 1999.

18. MathWorks. Quaternion Normalize. [Online] [Cited: Feb 13, 2018.] https://www.mathworks.com/help/aeroblks/quaternionnormalize.html.

19. Sutton, George P. and Biblarz, Oscar. Rocket Propulsion Elements. s.l. : John Wiley & Sons, 2001.

20. Propulsion Force.

21. Čuk, Danilo.

22. AERODINAMIČKE FUNKCIJE PRENOSA.

23. AUTOPILOT PROPINJANJA (SKRETANJA).

24. Seebass, A. Richard. Tactical Missile Aerodynamics: General Topics. Washington : American Institute of Aeronautics and Astronautics, 1992.

25. AUTOPILOT VALJANJA.

26. Astrom, K. J. and Murray, R. M. Feeback Systems. New Jersey : Princeton University PRess, 2009.

27. A New Nonlinear Guidance Logic for Trajectory Tracking. Park, Sanghyuk, Deyst, John and How, Jonathan P. American Institue of Aeronautics and Astronautics.

28. A Full Missile Homing System Design Based On Proportional Navigation Guidance Law And Electro-Optical Tracking System. Al-Ashtari, Waleed Kh. 2008, Journal of Engineering, pp. 2806-2824.

29. Lawrence, Anthony. Modern Inertial Technology Navigation, Guidance, and Control. New York : Springer, 1993.

30. Warren S. Flenniken, IV. Modelling Inertial Measurement Units and Analyzing The Effect of Their Errors in Navigation Application. Auburn : Auburn University, 2005. Master of Science Thesis.

31. Dougherty, John J., et al. GPS Modeling For Designing Aerospace Vehicle Navigation System. s.l. : IEEE, 1995.

32. Grewal, Mohinder S., Weill, Lawrence R. and Andrews, Angus P. Global Positioning Systems, Inertial Navigation, and Integration. s.l. : John Wiley & Sons, Inc., 2001.

33. Oppenheim, Alan V and Schafer, Ronald W. Discrete-Time Signal Processing. Upper Saddler River : Pearson Higher Education Inc., 2010.

# Изјава о ауторству

Потписани **Saif AlAmeri**

број индекса **D36/2015**

<p align="center">Изјављујем</p>

да је докторска дисертација под насловом

**Синтеза алгоритама навигације и вођења пројектила заснованих на машинском учењу (Missile Guidance Navigation and Control Algorithms Design Using Machine Learning)**

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, 27/01/2019

_____

# Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора **Saif AlAmeri**

Број индекса **D36/2015**

Студијски програм  **Докторске Системи Наоружања Студије**

Наслов рада **Синтеза алгоритама навигације и вођења пројектила заснованих на машинском учењу (Missile Guidance Navigation and Control Algorithms Design Using Machine Learning)**

Ментор  **др Драган Лазић редовни професор**

Потписани  **Saif AlAmeri**

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 27/01/2019

# Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић" да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

**Синтеза алгоритама навигације и вођења пројектила заснованих на машинском учењу (Missile Guidance Navigation and Control Algorithms Design Using Machine Learning)**

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

**1. Ауторство**

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство –  без прераде

6. Ауторство –  делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

**Потпис докторанда**

У Београду, 27/01/2019

_____

1. Ауторство - Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство - некомерцијално – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство - некомерцијално – делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство - делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.