

ORIGINAL RESEARCH

Intelligent missile guidance using artificial neural networks

Arvind Rajagopalan^{*1}, Farhan A. Faruqi², D (Nanda) Nandagopal¹

¹Division of Information Technology, Engineering and the Environment, University of South Australia, Australia

²Defence Science and Technology Organisation (DSTO), WCCSD, Edinburgh

Received: November 18, 2014

Accepted: February 2, 2015

Online Published: March 2, 2015

DOI: 10.5430/air.v4n1p60

URL: <http://dx.doi.org/10.5430/air.v4n1p60>

Abstract

Missile guidance systems using the Proportional Navigation (PN) guidance law is limited in performance in supporting wide class of engagement scenarios with varying mission and target parameters. For surpassing this limitation, an Artificial Neural Network (ANN) to substitute the PN guidance is proposed by the authors. The ANN based system enables learning, adaptation, and faster throughput and thus equips the guidance system with capability akin to intelligent biological organisms. This improvement could remove the barrier of limitations with allowable mission scope. In this paper, a Multi-Layer Perceptron (MLP) has been selected to implement the ANN based approach for replacing PN guidance. Attempts to replace PN guidance using MLP are limited in the literature and warrant greater attention due to significant theoretical development with the MLP field in recent times. It is shown in this paper, that the MLP based guidance law can effectively substitute PN for a wide range of engagement scenarios with variations in initial conditions. A foundational argument to justify using an MLP for substituting PN is provided. Besides this, the design, training and simulation based testing approach for an MLP to replace PN has been devised and described. The potential for faster throughput is possible as the MLP nodes process information in parallel when generating PN like guidance commands. The results clearly demonstrate the potential of MLP in future applications to effectively replace and thus upgrade a wide spectrum of modern missile guidance laws.

Key Words: Artificial neural networks, Function approximation, Proportional navigation, Missile guidance, Intelligent guidance, Non-linear regression, Machine learning, Terminal guidance of missiles

1 Introduction

A staple missile guidance law that has remained robust for most homing missile engagements is Proportional Navigation (PN).^[1] However, this robustness could decrease due to a higher level of challenge being forced on homing missiles in the modern mission environment. The challenges can be attributed to needing to engage targets that are improved in their ability to maneuver in complex fashion, provide for better concealment against threats and being capable of duping attacking homing missiles.^[2,3] Guidance systems mak-

ing use of laws such as PN thus require upgrades to overcome this significant problem. To compound the difficulty, any upgrade proposed, needs to be cost effective in order to become practically useful.

One approach to achieve these upgrades is to enable the missile guidance system to perform automatic learning, adaptation, and improve throughput when generating guidance law commands. This has led to research and design effort being directed towards the discipline of Intelligent Systems techniques.^[4] This discipline has been advancing in order to up-

^{*}**Correspondence:** Arvind Rajagopalan; Email: rajay011@mymail.unisa.edu.au; Address: Department of Information Technology, Engineering and the Environment, University of South Australia, Australia

grade a system's agility at low cost. Therefore, the authors have identified it as a correct direction for investigations for upgrading PN. The Intelligent systems discipline is actually a collection of a number of fields, which includes Artificial Neural Networks, Bayesian estimation, and Game theory to name a few.

For this study, an Artificial neural network (ANN) based approach was selected for substituting and upgrading PN. This is because it appeared to be the best fit for enabling automatic learning, adaptation, as well as faster throughput.^[4,5] In particular, the authors selected one variety of ANN, namely the Multi-Layer-Perceptron (MLP). The selection of MLP can be justified through the following argument: Missile guidance laws such as PN and many others make use of mathematical formulae to represent the guidance law.^[6] A generalized equation is shown as an example in Equation 1. This equation describes acceleration output of a guidance law as a continuous function of inputs.

$$Accn_{demanded} = F(Guidance\ law\ inputs) \quad (1)$$

MLP theory is well developed^[4] and has been widely applied for approximation of continuous mathematical functions to high accuracy. Furthermore, MLP ANNs have been proven mathematically as universal function approximators, able to achieve an arbitrary level of accuracy in matching a given continuous differentiable function.^[7,8] Therefore, the choice to use MLP for substituting PN was justified.

A number of research and design questions were posed when considering replacing PN with an MLP. The main questions raised and answered through this study are provided below. The starting point was to determine the test scenario that was required to compare the performance of PN against an MLP based guidance approach. It is well known that PN can be applied for a broad range of terminal engagements.^[1] A terminal engagement scenario focusses on the last portion of flight where the missile is usually tracking the target within its own on-board sensor and homing onto it. This usually requires the proximity to the target to be small relative to the rest of the missile's flight. The small proximity in turn constraints the time to take correct action by the homing system in response to changes in target behavior. So far, PN has been applied successfully in many such engagement scenarios.^[1,6] This is especially the case with many types of air-to-ground engagements where a guidance system using PN has been applied to successfully intercept a relatively slow moving ground target. For this scenario, PN has been found to cope with any variation in initial conditions such as heading and range and still intercept the target with small miss distances.^[1] However, when expanding mission scope beyond air-to-ground to air-to-air and so on, PN has limitations in being able to achieve intercept with small miss distances. Therefore, as a first step, an MLP design that purports to replace and upgrade PN would need to perform just as well as PN for the type of mission

scenario where PN is adequate. Consequently, the air-to-ground scenario was a selected for the test-bed to compare degree of match between PN and an MLP based guidance law. Furthermore, the authors have devised the test-bed as well as mission scenario to be scalable to support other types of engagements, which would be useful for future studies such as air-to-air and so on.

The next question was to determine a suitable MLP network design. This involved taking into consideration a number of factors. This included selection of suitable inputs and outputs, selecting number of hidden layers and number of neurons per hidden layer and selecting the activation function for these neurons. To answer this question, a closer look at classical PN's inputs and outputs as well as the nature of relationship (linear/non-linear) between input and output variables needed consideration. This needed to be associated with mathematical proofs that enabled an MLP to be a universal approximator for continuous differentiable functions. Besides this, the training of the designed MLP was also a major focus for the investigation. Training has the objective of closely matching the behavior of the mathematical function being mapped via the MLP and needs to achieve this outcome efficiently in terms of computation time. There is already a lot of criteria available in literature which describes various types of MLP training algorithms.^[4,5] Using this and relating it to the PN replacement problem, the first major decision was to select between applying either a supervised or an unsupervised training paradigm.^[4,5,9] Within these paradigms, there are a larger number of classes of algorithms, so there needed to be further selection for choosing the particular class of either supervised or unsupervised training algorithm. This needed to take into account training speed and under or over-fitting trade-offs^[10,11] which impacted correct generalization (prediction) potential.^[4,5,12] Since there was input-output data available via running classical PN over a set of air-to-ground engagements, it was easy to choose the supervised training approach, however choosing the particular type of supervised training algorithm was not a trivial process and is explored in this study.

To assist with selections, a number of variations in scenarios^[1,6] under the theme of air-to-ground terminal engagement needed to be devised and tested with both PN and MLP based guidance. The approach to systematically test was critical in order to yield the best choice for MLP design that was compact and efficient to execute and could replace PN with high accuracy across all of the scenarios tested. For all of these tests, it was important to keep referring to a relevant set of metrics in order to compare PN and MLP performance. Selecting relevant metrics was an investigation in itself as there are already many metrics for evaluating MLP in the literature.^[4,5] This was useful to serve as reference but needed to be filtered to select the appropriate ones that made sense in context of missile guidance system performance analysis. Finally, it was important to carefully

select the training data for all the scenarios tested because some choices would result in the MLP guidance only being able to substitute PN for data in the neighborhood of training data collected, rather than match correctly across the entire set of input-output patterns that processing using PN allows for. This was briefly investigated with some limited solutions described by referring to literature such as.^[13]

The consequence of an MLP successfully substituting for PN can open the doorway to guidance law design that is much more flexible and robust for support demanding missions or an expanded mission scope. To consider the the major impacts, the assessment can be divided into three areas which can be stated as impact of enabling learning, impact of enabling adaptation and impact of faster throughput respectively. The ability to cater for learning^[4,5,9] provides for a whole range of different and complex continuous guidance law functions to be effectively copied by an MLP. Since the MLP can be further tuned and can generalize depending on choice of training algorithm, this allows for increased flexibility. It means that a single network can effectively be used to represent a wide range of guidance laws. For missions, this can be useful as the mission stages change or for situation where custom requirements for terminal phase involving some modification to PN etc becomes mandatory. The ability to provide for adaption is because an MLP can be tuned in real-time^[4,5] as well within allowable limitations to cope with some degree of novel environmental conditions not encountered during training. This allows the mission to progress without failing in such circumstances. Using classical guidance laws such as PN here would otherwise limit mission scope. Finally, the ability to support faster throughput via MLP can mean that guidance commands are computed faster. This allows for trajectory correction in a timely manner against highly agile targets. Since modern targets are becoming equipped with better maneuvering capability,^[2,6] an MLP based guidance can allow for robustness under these conditions.

The current paper is organized as follows: Section 2 describes a survey of recent work where an ANN substitute has been used in place of a classical PN missile guidance law. Apart from gaining an understanding of how other authors have applied ANN for purpose of missile guidance, the authors of this paper have also identified some gaps in approaches used by others. These gaps are described and are mostly addressed in this paper. Section 3 provides a description of a guidance test-bed framework for testing and comparing the ANN based guidance system against classical guidance approaches. Comparison simulations were conducted for a variety of mission scenarios via the test-bed. Section 4 provides relevant ANN theory, which is suitable for application for substituting missile guidance laws via referring to work done by others as well as understanding level of development with MLP and missile guidance fields. Section 5 discusses all of the critical simulation tests conducted

to design, refine, and benchmark the ANN guidance law. Finally, Section 6 provides conclusions, and discusses the significance of the current work. A summary of future research directions is also given.

2 Backgroundd for ANN based missile guidance

In the open literature, only a limited number of articles are available describing missile guidance using ANN. The authors of this paper have reported on some of the relevant and significant works by others in order to discover the strategies, suggestions for progression and gaps in applying neural network techniques in place of classical missile guidance laws. All of the research questions described in the introduction section were used to guide this analysis.

The most notable paper surveyed on ANN based missile guidance was Ref.^[14] Here, an MLP was used to replace a classical game theoretic pursuit-evasion guidance law, which is a generalized version of PN. The argument made to introduce an MLP substitute was that the classical pursuit-evasion guidance law was unsuitable for real-time implementation. To overcome this problem, an MLP substitute was deemed a suitable replacement. However, this justification was not addressed in detail. This has provided the motivation for authors of the current paper to provide systematic investigation into the arguments for introducing ANN based guidance techniques as appropriate substitutes in place of "difficult-to-implement" classical guidance laws. The inputs applied for the ANN substitute in Ref.^[14] were range, range rate, line-of-sight rate, and heading error. For the current paper, this set of inputs was seen to be excessive and was reduced to simply consist of line-of-sight rates in azimuth and elevation with range-rate (also expressed as closing velocity) becoming an optional input because of low variation over mission scenario. Heading error and range information was determined to secondary to critical information captured by the line of sight rate inputs, hence they were omitted or designated as optional. The outputs were the acceleration demands in azimuth and elevation required to be achieved by the missile to home in to the target. The same set of outputs was selected in this paper, as it was the obvious choice. However, with the current paper, the authors used inertial axis as well as made use of direction cosines transformation to convert acceleration demands to missile body axis and vice-versa as required.^[1] The network used in^[14] was an MLP with a single hidden layer. Sigmoid activation functions were used by the neurons the hidden layer and linear neurons in the output layer. Such an approach is also used in this paper, as it is the typical approach when making use of MLP for function substitution problems.

The training algorithm reported in Ref.^[14] was Levenberg Marquardt (LM). Once again, this selection was not justified in detail. For this paper, the MATLAB neural network tool-

box^[15] was used and does provide the Levenberg Marquardt (LM) training algorithm as one choice for performing supervised training. However, selection of training algorithm was performed using a systematic approach by authors of the current paper. Table 1^[14] provides a means to compare the flight times between using classical PN and Neural network configurations. There is some information in the prose about miss distance comparison, but this is not reported in tabular form and so is difficult to follow. The manner of reporting will be improved in the current paper, so it is easier to follow for the reader. Both miss distance and flight time are critical metrics to evaluate any guidance law so they will continue to be used for this paper. Besides this, Ref.^[14] states in section 4.1 that the maximum difference in miss between the PN and NN cases is 0.2 m and the difference in flight times was 2 ms. This is a small difference and shows evidence of high degree of match between PN and ANN approaches. However, the training time required to build up the ANN is not reported. This will be reported in the current paper, as it is important to take note of when designing training scenarios. Some scenarios may lead to long and prohibitive training times and thus can restrict complexity of the network architecture as well as degree of testing to match against PN. The suggestion made by the researchers at the end of the paper is to consider 3-D engagements. This has been performed in the current paper, which means that the current paper somewhat extends the work.^[14]

Table 1: Simulations per choice of hidden layer neurons

Test Description	Average RMS Error over flight (m)	Maximum Error over flight (m)
3 HL Neurons Run 1	0.2438	1.079
3 HL Neurons Run 2	0.1924	0.5299
3 HL Neurons Run 3	0.4224	15.83
5 HL Neurons Run 1	0.331	5.599
5 HL Neurons Run 2	0.1872	1.202
5 HL Neurons Run 3	0.4419	1.752
7 HL Neurons Run 1	0.7165	1.888
7 HL Neurons Run 2	2.11	53.31
7 HL Neurons Run 3	0.3642	1.683
9 HL Neurons Run 1	0.4837	3.972
9 HL Neurons Run 2	0.5564	2.08
9 HL Neurons Run 3	0.4323	1.933

Another notable paper reviewed was Ref.^[16] In this paper, the authors employed an ANN based controller for the guidance law using a design strategy known as adaptive-critic. This strategy falls into the domain of following an approximate dynamic programming approach.^[17] It is an advanced approach where two separate networks are used, namely, the action network and critic network. The role of the action network is to generate a guidance law, which is used to update the missile plant model, which in turn produces an output that is then evaluated using a critic network. The critic network's function is to approximate a value related function, which, in this case will affect estimated miss distance

and energy available at the end of the missile flight. The action's network can be tuned to best match the guidance law that will result in some base cost. The critic network can be tuned to turn this base cost and drive it down a minimal cost and in doing so will provide an updated set of parameters such as change to navigation ratio to optimize the guidance law. The optimized guidance law is then encoded via further tuning of the action network. This cycle is repeated over and over to meet some strict error tolerance criteria set. For the current paper, the authors have stuck with the approach of Ref.^[14] rather than Ref.^[16] This is because approximate dynamic programming is required only when dealing with more advanced non-linear guidance laws such as the Optimal Guidance Law (OGL).^[18] The focus for this paper is PN, which does not require this sophistication. However, the remaining sections^[16] was useful for discovering the approaches taken when considering the other questions considered in the introduction of this paper.

The inputs chosen for the action network^[16] were 4 inputs, which were associated with 4 guidance gains required to be calculated in order to work out acceleration demands via OGL. For the current paper, since PN is being used, there is no need to make use of such gains. The outputs chosen in Ref.^[16] were two outputs, which denoted acceleration values in down and cross range respectively. This is same as Ref.^[14] and will be used in the current paper. The type of network used in Ref.^[16] was an MLP with two hidden layers with 6 hidden layer neurons per hidden layer. Each of the hidden layer neurons used sigmoid activation functions. The reasoning for this configuration was not provided. Choosing number of hidden layer neurons and number of hidden layers will be subject of systematic investigation in the current paper.

The training algorithm used by in Ref.^[16] was back-propagation, which was selected because the authors stated that it was simple, and in widespread use. Just like Ref.,^[16] there was not much discussion on selection of back-propagation besides these comments. In the current paper, the justification for training algorithm is being addressed. The metrics used in Ref.^[16] to compare performance between MLP and OGL was by calculating the impact on miss distance, optimal cost function performance index and flight time. Figure 2 in Ref.^[16] describes the difference in trajectory between classical OGL and when making using of the adaptive critic. It is evident that for most of the flight, there is some difference between the trajectories but towards the end of the flight, the trajectories begin to converge. It was concerning to see that the trajectories in between were deviating from the classical case. This may or may not be acceptable when factoring in real mission conditions. In the current paper, the attempt is made to match the whole trajectory of PN as closely as possible. The result reported was that the adaptive critic outperformed by the OGL by producing on average a miss that was 5 m smaller. For the current

paper, the manner of comparing the trajectories will be same i.e. to compare trajectory and miss. However since there is a 3-D engagement, the trajectory comparison will take place in all three axes and the miss as impact on flight time will be compared. Ref.^[16] also provides some information on number of training trials but it not at all obvious how much time was actually taken for training and execution of the ANN based guidance law. This will be clearly reported for the current paper. The suggestion for future research by Ref.^[16] was to make use of unsupervised training algorithms by allowing the critic network to learn online. Such a suggestion does not link with the research effort in this paper but may be useful for advanced work in later studies.

The other papers found in the survey were less relevant and detailed but have been mentioned briefly. In Ref.^[19] Radial-Basis Function (RBF) neural network architecture with fuzzy logic was applied to classical substitute PN guidance. Fuzzy logic^[20] is another subfield of Intelligent systems but is outside scope of study for this paper. Scanning through the remainder of Ref.^[19] it was highly similar to the work done in Ref.^[14] except that it was bit more limited and brief. Once again, the reasoning for specific choice of neural network architecture was not discussed. Other authors such as Ref.^[14,21,22] have discussed approaches to substitute advanced classical guidance laws (OGL) using ANN. Just like has been mentioned when evaluating^[16] attempting to use neural networks to represent advanced classical guidance laws has been left for the future.

3 Test-bed framework

A test-bed framework was developed for benchmarking novel guidance law formulations against classical guidance laws. The benchmarking was completed for a variety of mission scenarios simulations, which are likely to occur in real-world settings. The test-bed framework is depicted in Figure 1. It shows that the measurements of target information can be transmitted and processed via the ANN based guidance law and the PN guidance law separately so that two sets of acceleration demands, one for ANN and PN respectively can be produced. These outputs can be used independently to direct the missile to home in onto the target in separate simulations. Mission data was collected and stored to compare the two approaches when running the simulations. The comparison was performed by considering the following:

- The trajectories for the missile when applying PN and ANN guidance respectively were compared by taking the difference between them over the flight time of the missile and calculating an RMS error throughout that period.
- The closest point of approach to the target (referred to as "miss-distance"^[6]) and the time at which this occurred was noted and compared to determine the

degree of closeness of results.

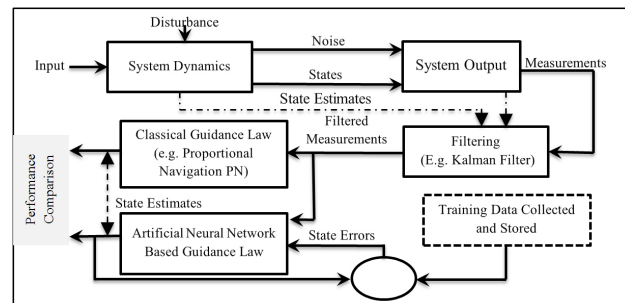


Figure 1: Test-Bed Framework for testing ANN and PN Guidance Laws

These two aspects were deemed sufficient for comparing the two guidance approaches. Other tests such as generating a kinematic boundary^[23] and stability analysis^[24] have been deferred to future studies as they were deemed more useful when considering substitution for more advanced guidance laws such as OGL where non-linearity and complex trajectory means that studying the impact on these aspects is necessary.

3.1 Simulation scenario

The engagement geometry applied to the test-bed is depicted in Figure 2. This figure shows the geometrical relationship between a missile and target. It shows the important quantities (angles/vectors) to take into consideration when devising a guidance method. The engagement geometry shown is the case of a generalized 3-D intercept where the missile or target can be on the ground, in the air, or in the ocean. In the current paper, this geometry was modified to the case of a missile directed to strike a relatively slow moving ground target. The target was placed some distance in both down-range and cross-range from the missile as an initial condition. This made the geometry, a 3-D engagement scenario. In addition, this kind of initial condition represents the beginning of the terminal phase of engagement. Terminal phase refers to the final stage of intercept where the missile has exhausted its rocket fuel and is relying completely on built up kinetic energy and maneuvering capability (via fins, for example) to home in to the target via gliding.^[6] The engagement geometry was selected with the following attributes:

The terminal phase of an engagement has the goal of intercepting the designated target.

- Time to achieve intercept is only in the order of a few seconds due to the close proximity to the target.
- The missile has to cope with constraints during this phase such as heading error with respect to the calculated collision course at the beginning of the terminal phase. This forces the requirement of rapid and accurate adjustments in missile motion when homing in

onto the target. With the earlier stages of flight, i.e. the bestlaunch and mid-course guidance phases, the demands, on the missile to guide precisely under high level of time constraints is not serious a concern, as it is with the terminal phase.^[6] Hence, the focus has been placed on terminal phase engagement only.

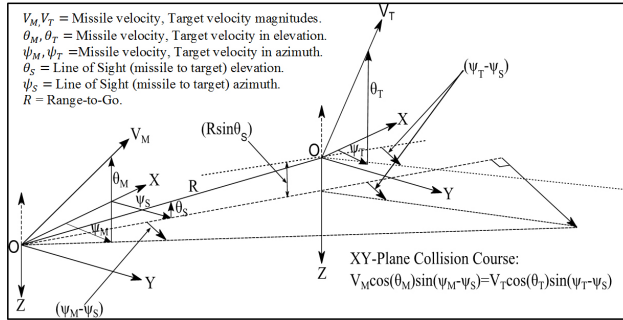


Figure 2: Engagement Geometry between ANN and PN

3.2 PN guidance

PN was chosen as the classical guidance law to compare with the ANN guidance approach. The following motivated the selection:

- PN is widely applied and well understood and is available in analytical form.^[1]
- It is valid to apply it in the test-bed framework scenario for an air-to-ground engagement.^[1,6]
- The analytical expression is non-linear but limited in complexity compared to advanced classical Guidance Laws such as the Optimal Guidance Law (OGL).

Therefore, PN forms an appropriate starting point prior to attempting to substitute more advanced guidance laws using ANN. The form of the PN guidance law is as follows:

$$a_{i_d}^b = (N \times \hat{w}_{s_{ji}}) \tag{2}$$

In (2), a_i^b is the inertial acceleration demanded by the missile defined in missile body axis (**b**). **N** is the navigation constant and $\hat{w}_{s_{ji}}$ is the estimated line of sight (LOS) rate from missile to the target (s_{ij}) in inertial/missile-body axis.^[6]

3.3 Validation of test-bed

Classical PN was run on the test-bed for the simulation scenario. The results generated were then compared against published results for the same scenario from Ref.^[1] This approach taken to validate the test-bed. The graph in Figure 3 shows the result for normalized miss distance vs normalized flight time for variety of heading errors for classical PN. This response agreed with published results from Ref.^[1] The figure also shows that there is limited variation in the

graph when heading error was varied. This level of variation was deemed to be within tolerable limits (less than 1m, set conservatively) in order to not affect successful intercept goals. Following successful validation of the test-bed, it was possible to use it to evaluate the performance of an ANN guidance law substitute for PN.

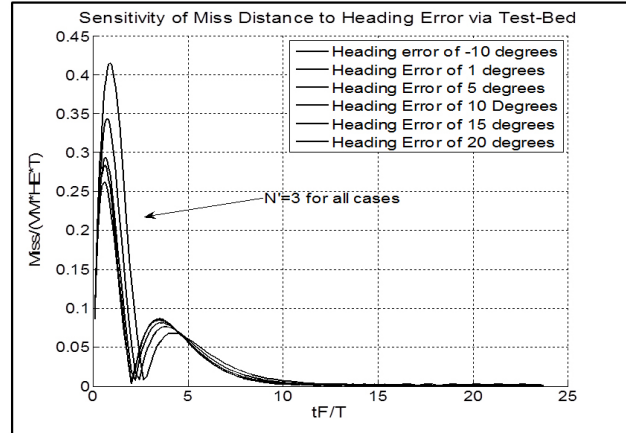


Figure 3: Sensitivity of Miss to missile heading error

4 ANN based missile guidance

4.1 Classical PN function

The form of the PN guidance function to be substituted via ANN is displayed graphically in Figure 4 and Figure 5.

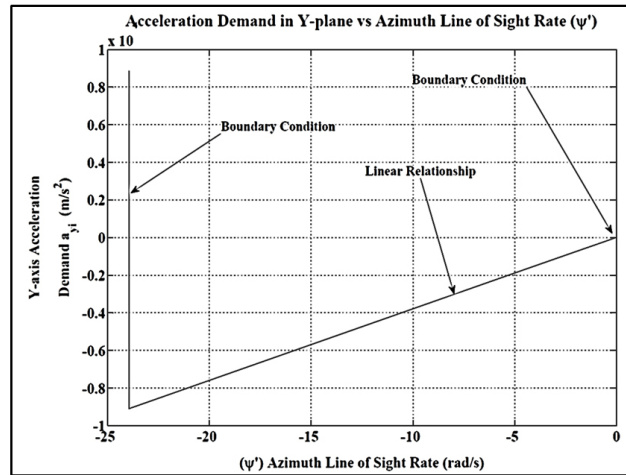


Figure 4: PN Guidance in Azimuth Plane

It can be observed, that for all cases, apart from extremities in the x-axes (line-of-sight), the relationship between inputs and outputs for PN is linear. The inconsistency at the extremity in can be attributed to rapid line of sight variation near interception the target as the target crosses the missile's path from one side to the other in short time span. The combined speed of missile and target is large and contributes to

this. This feature should not pose a problem with the substitution via ANN. This is because of its capability to be able to substitute for non-linearity, as well as coping with some degree of discontinuity without suffering loss of accuracy when substituting a function.^[25] Therefore, it is acceptable to view PN guidance law function as square-integrable and hence it falls with the class of L^2 functions.^[26] This fact is noteworthy because it means that the universal approximation theorem in Ref.^[7,27] can now be applied to enable substitution of the classical PN guidance function. This can be accomplished using a multi-layer perceptron (MLP) ANN architecture.^[8]

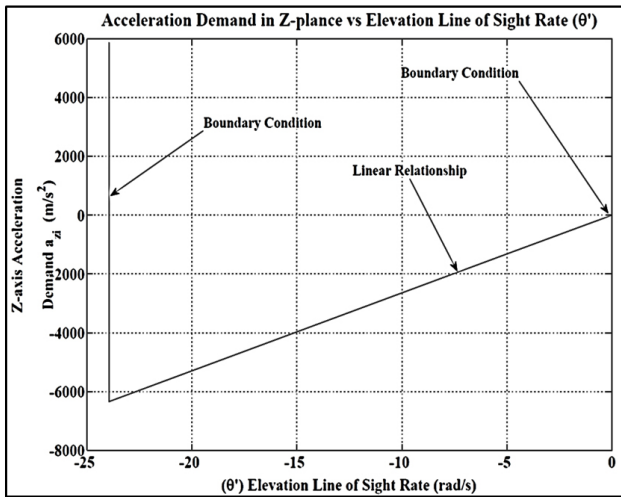


Figure 5: PN Guidance in Elevation Plane

For the current paper, MLP architecture has been selected for the guidance law substitution. This is because of reported issues with extrapolation for the other potential ANN candidate, namely, the RBF. This issue is described in Ref.^[28] Extrapolation would be necessary for supporting generalization, a desirable feature for any trained neural network to cope with correctly mapping inputs to outputs, which were not provided during training. The next section describes a summary of the design process used to construct a suitable MLP ANN.

4.2 Benchmarking against classical regression

The general form of the multi-variate linear regression (MvLR) as well the specific way it was applied for this problem is given in equation (3) and equation (4) respectively.

$$\text{General Form : } [Y] = [X] * [\beta] + \epsilon \tag{3}$$

$$\text{Specific Form: } \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} 1 & \hat{\theta}_{LOS} & \hat{\phi}_{LOS} \end{bmatrix} * \begin{bmatrix} \beta_{10} & \beta_{11} & \beta_{21} \\ \beta_{20} & \beta_{21} & \beta_{22} \\ \beta_{30} & \beta_{31} & \beta_{32} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}$$

Dependant variables (Y)
Predictor variables (X)
Regression Coefficients (β)
Residual error

$$\tag{4}$$

The MLP ANN based substitution needs to be compared against applying a classical regression technique to perform the substitution for PN. This is required in order to evaluate whether a classical regression is adequate as well as determining whether the MLP ANN approach offers a superior solution.

A MvLR was chosen as the classical regression technique to substitute PN because

- There are multiple predictor and multiple dependent variables
- The relationship between predictor and dependent variables is linear as can be seen in Figures 4 and 5.

The regression was performed by collecting the data when applying PN for line of sight rates as well as associated accelerated demands and using this to calculate the regression coefficients (β) via ordinary least squares minimization.

4.3 MLP Guidance solution design approach

The approach used for designing the MLP ANN is summarized schematically in Figure 6. Each of the steps outlined in the figure is elaborated in the following section.

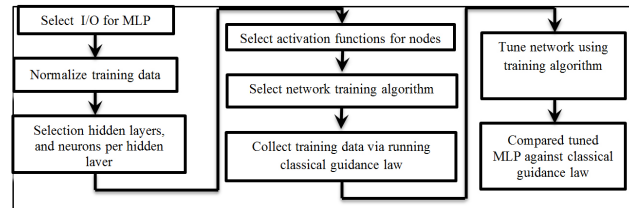


Figure 6: MLP design approach

Detailed explanation of MLP ANN design approach

Step 1: To begin with, the number of inputs and outputs for the network needed to be selected. The line of sight rates in azimuth and elevation between missile and target were chosen as the inputs for the network. Closing velocity was not included as an input. This was because it does not vary greatly over the course of the terminal phase. In addition, the effects of gravity, wind resistance etc. was not included in the modelling. The acceleration demand produced by the PN guidance law was the obvious choice for the network output. Since the engagement scenario was in 3-D, the demands needed to be applied in x, y and z axes. In summary, two inputs (LOS rates) and three outputs (planar Acceleration demands) were used for the MLP ANN.

Step 2: The inputs and outputs chosen required modification via a normalization pre and post processor. Normalization scales the inputs and outputs into the range 0 to 1. This step was introduced in order to provide for best choice of weights end the end of training. Min-max normalization has been applied to the network inputs and outputs Ref.^[12]

Step 3: The MLP ANN was configured with one hidden layer and one output layer as this was determined to be adequate for the guidance function being substituted. This is because one hidden layer is considered sufficient for representing data that may not be linearly-separable.^[29] Guidance functions requiring substitution can fall into this category, because they need not be Boolean in nature.^[30] The number of neurons in the hidden layer was decided upon via trial and error by making an initial guess at number of neurons required and then pruning or growing neurons as necessary depending on observing the values of weights connecting to the particular neuron at the end of training. If the weights were close to zero, the neuron could be removed because it had insignificant impact on the mapping. If the Mean squared error (MSE) between actual and network outputs at the end of training was not sufficiently small, another neuron could be added and training repeated, until the MSE was brought within tolerable margin. This strategy was adopted following consultation of the "Optimal Brain Damage" approach.^[31] An output layer was also used in order to serve as a scaling layer in addition to normalization treatment on inputs and outputs.^[32] A diagram of the MLP ANN network architecture with these layers is depicted in Figure 7.

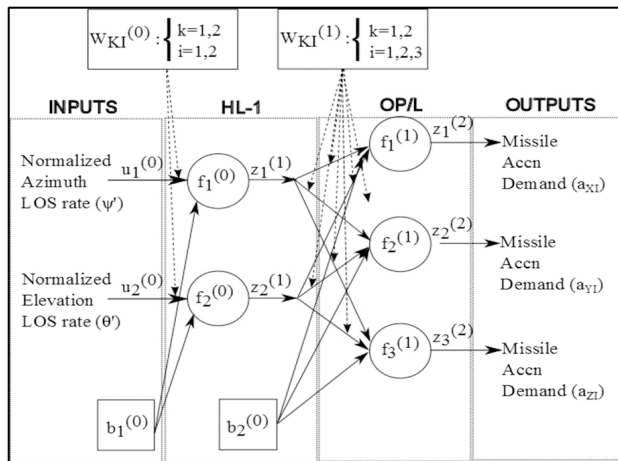


Figure 7: Multi-Layer Perceptron (MLP) network architecture

Step 4: Following selection of network architecture, the design of the internals of neurons (activation functions) for the hidden as well as output layer was chosen. For the hidden layer, a tan-sigmoid function was chosen as the activation function in order to be able to account for non-linear regions of guidance function being substituted.^[8] For the output layer a linear function was chosen as activation function in order to provide for scaling of the signals from hidden layer to the relevant magnitudes beyond range of 0 to 1 when transmitting to network outputs.^[32]

Step 5: The training algorithm to tune the MLP ANN was chosen initially to be back-propagation, the simplest and

appropriate training algorithm. One detailed walkthrough of back-propagation is provided in Ref.^[33] Within MATLAB's Neural Network Toolbox,^[15] the tool used for this study, the Levenberg–Marquardt supervised training algorithm was selected. This algorithm was reported to be fast and stable for small and medium sized non-linear regression problems Ref.^[34,35] It was also reported to yield a better match to the guidance law being substituted compared to other back-propagation algorithms. Some tests were conducted to confirm this for this application. Most other authors surveyed did not delve into selection of training algorithm and suitable training data. From the papers surveyed, only Ref.^[14] has gone on to state that the (LM) algorithm was put to use for training the networks because it is robust popular for minimization purposes, the same reason for which in this study it has been selected. However, in our study, we have gone further to make use of extra features such as Bayesian regularization,^[36] and early stopping via cross validation to minimize the changes of over-fitting taking place. This has not been covered by Ref.^[14] or any of the other authors surveyed. In Ref.,^[14] however, there is some treatment given to describe how to select the training data to be representative in detail. In this paper, the principles for improved selection of training data are mentioned without delving into detailed exploration. This has been left as a future research pathway.

Step 6: The comparison between classical PN and MLP ANN substitute was performed as follows:

- (1) Use classical PN as guidance and collect LOS and missile acceleration data as inputs and outputs to be stored for training the network.
- (2) Initialize an MLP ANN and train the network using training data available until MSE is within tolerable level.
- (3) Substitute the classical PN with trained MLP ANN and run the engagement simulation again.
- (4) Compare match between classical PN and MLP ANN results.
- (5) Repeat all of these steps for variations in engagement geometry in order to refine the MLP ANN design.

5 Simulations

This section describes the simulations conducted to test the performance of the ANN based Guidance Law against the conventional PN for the scenarios described in Section 3.

5.1 General simulation settings

In all of these simulations, the following assumptions hold:

- Both missile and target are treated as point masses. The missile's airframe response is a first-order lag when homing in to the target.
- Only single missile vs single target scenario was used for the tests.

- The target was constrained to be non-accelerating, but was allowed to have motion at constant velocity.
- The missile was constrained to allow lateral acceleration only (i.e. in azimuth and elevation direction w.r.t to the missile body axis).
- The missile has lateral acceleration-limits applied in order to make the scenario more realistic.
- Gravity and drag forces were not taken into consideration.

All of these settings were applied in order to setup the simplest but adequate configuration to be able to carry out a sensible comparison.

5.2 Performance criteria for comparison

The performance criterion to evaluate the performance of the MLP relative to classical PN has been conducted via considering the following factors. The labels for the factor as well as its type are provided for reader benefit:

- Effect on Miss Distance (\mathbf{R}^+ in metres)
- Effect on Flight Time (\mathbf{R}^+ in seconds)
- RMS error for difference in trajectory between PN and MLP or other regression scheme over whole flight (\mathbf{R}^+ , time series in metres)
- Time to train MLP or determine coefficients for classical regression. (\mathbf{R}^{++} in seconds)
- Time to execute MLP or classical regression scheme in comparison with PN (\mathbf{R}^+ in seconds)
- Network Complexity Metric: VC dimension.^[37]
- Training data similarity metric: matrix norm,^[38] Dynamic Time Warping (DTW) distance.^[39]

5.3 Simulation results & observations

5.3.1 Simulation 1: Simple collision test

A. Description

In this test, the missile was placed on collision course with slow moving ground target. This is depicted via the planar engagement geometry shown in Figure 8. This figure does show the provision for heading error, which was set to zero for the current test, which is what makes it a collision course.

For collision condition in Figure 8 to be satisfied, the movement, normal to the line of sight should be same for both missile and target within a given time duration. The following equations describe the collision course calculation as well its relationship to missile heading.

$$V_M \sin(\theta_s - \theta_{Mcc}) = V_T \sin(\theta_s - \theta_T) \quad (5)$$

$$\rightarrow \theta_{Mcc} = \theta_s - \sin^{-1}(V_T/V_M \sin(\theta_s - \theta_T)) \quad (6)$$

$$\text{From Geometry : } \theta_M = \theta_{Mcc} - \theta_{HE} \quad (7)$$

In (5), (6) and (7), V_M is the velocity of the missile, V_T is the velocity of the target, θ_s is the initial line of sight angle, θ_{MCC} is the collision course, θ_M is the missile heading error, θ_{HE} is the missile heading error.

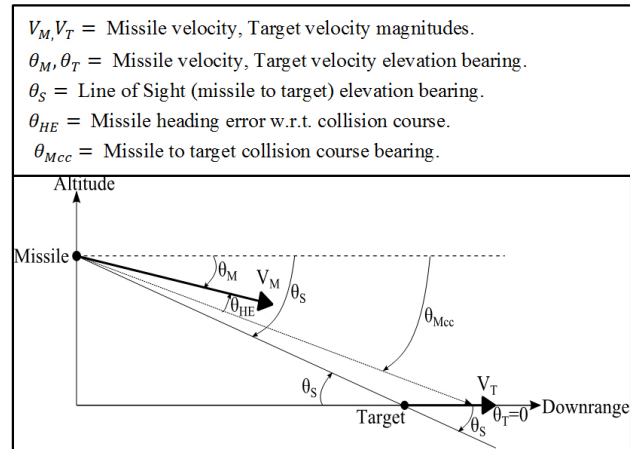


Figure 8: Collision course diagram

The geometry shows a missile and target at the beginning of the terminal phase of an engagement. Accompanying equations to calculate the collision course trajectory are provided in (4), (5), and (6). The target is shown as moving along the ground with constant velocity and the missile is aimed towards the target. The line of sight between missile and target is displayed in Figure 8. The actual heading for the missile is also displayed via the missile velocity vector Figure 8. The direction for this vector may not be coincident with the angle that describes the "zero-miss collision course".^[6] The collision course is shown in Figure 8 via the line joining the missile and a point ahead of the target on the ground where the missile is expected to hit the target in the case of zero heading error being applied at the beginning of the terminal phase. The missile was configured to commence the terminal engagement with a velocity of 250 m/s. The target was configured to move along the ground at constant velocity of 50 m/s away from the missile. This is an example of a typical terminal phase engagement for guided missiles against moving land targets.^[1]

B. Guidance system performance results and discussion

In Figure 9, two plots are displayed. The plot at the top shows the decrease in separation range over the flight time for the missile as it homes in on the target. There are three graphs present in this plot, each corresponding to a different initial value for separation range between missile and target. There are in fact six lines in the top plot of Figure 9. This can be seen via the accompanying legend. Three of these lines are associated with applying classical PN and

the other three are associated with application of the MLP ANN substitute for PN. The bottom plot of Figure 9 shows the Root Mean Square error (RMS)^[40] when comparing the PN against ANN results. This was computed by taking the difference between the range-to-go values produced via application of PN against the range-to-go value produced via application of the ANN substitute over the whole flight time, followed by taking the RMS of this difference.

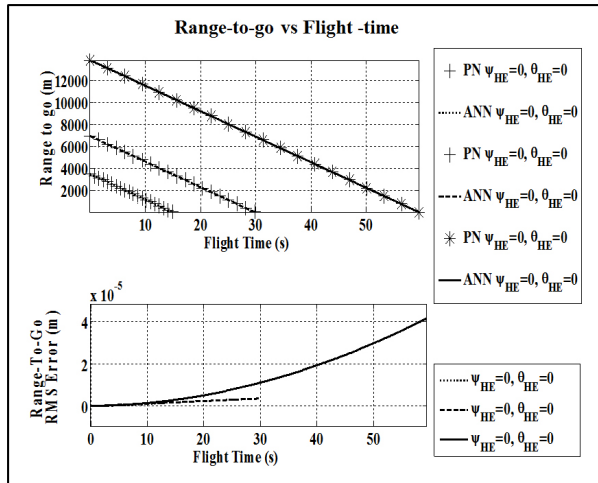


Figure 9: Planar view for collision course

The legends show the initial conditions in terms of heading error that was applied to the missile as well serving as means to identify the associated graph. In this case, zero heading error was applied in azimuth and elevation directions respectively at the beginning of the terminal phase. This is indicated in Table 2. The table also shows the comparison between final miss values and miss times between

PN and ANN. In Figure 10, a 3-dimensional view of the same scenario is displayed, once again with two plots. The plot at the top displays the paths of the missile and target over the flight time. The plot at the bottom shows the RMS error that describes the difference between the classical PN vs ANN paths. Once again, there are three simulation runs, consistent with Figure 9. This pattern of presenting results is repeated through the rest of the simulations. From Figure 9 and Figure 10, it can be observed that the RMS error is small (of the order of tens of micrometers). This level of precision in match was deemed close enough match between PN and ANN. Table 2 serves to reinforce this via close agreement between PN and ANN minimum miss distance and miss times.

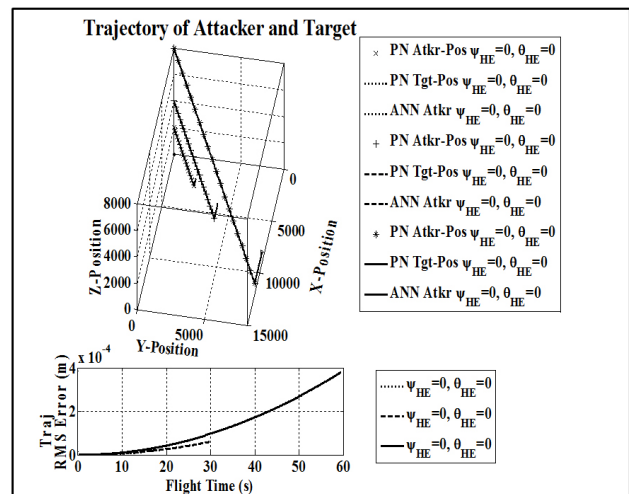


Figure 10: Engagement Trajectory for collision course

Table 2: Collision Course Settings and Results

Flt No	Initial Range To Go (m)	Azimuth HE (deg)	Elevation HE (deg)	PN Miss Time (s)	PN Miss (m)	ANN Miss Time (s)	ANN Miss (m)
1	3464	0.0	0.0	14.87	0.03018	14.87	0.03018
2	6928	0.0	0.0	29.74	0.05613	29.74	0.05613
3	13860	0.0	0.0	59.48	0.004233	59.48	0.004291

5.3.2 Simulation 2: Simulation repeatability test

A. Description

This simulation was conducted in order to verify that it is possible to repeat Simulation 1.

B. Results and observations

Figure 11 shows two plots. The top plot provides a description of range-to-go versus time and the bottom plot provides the RMS error between PN and MLP ANN range-to-go plot lines. The results in the top plot show that results can be repeated because the running of two consecutive simulations

with the same settings results in identical plots. Via this observation, it was deemed that the ability to repeat results when running the same simulation was successful and the simulation program was now ready for rigorous testing of MLP ANN vs classical PN performance comparison studies.

5.3.3 Simulation 3: Comparison with regression test

A. Description

In order to benchmark the MLP based substitution, a MvLR based approximation to PN was produced. A comparison

between the trajectories using PN, MvLR and MLP are provided as well RMS error differences in comparison with PN. A summary of execution times for MLP and MvLR schemes has been collected and used for the performance analysis.

network to harness its inherent potential for fast execution. The network used had 5 neurons in the hidden layer and 3 neurons in the output layer.

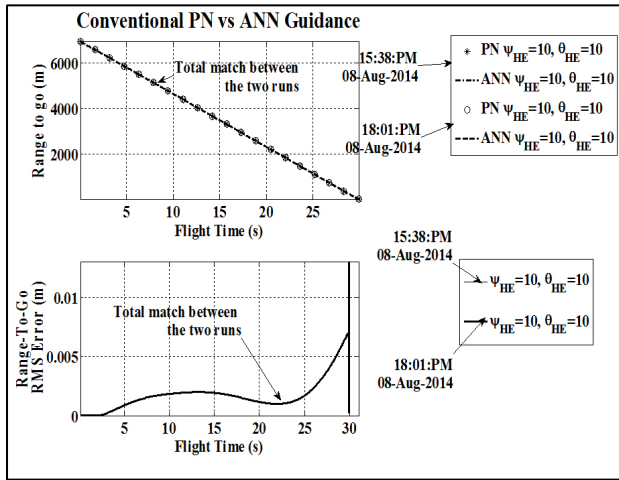


Figure 11: Repetition of Collision course simulation

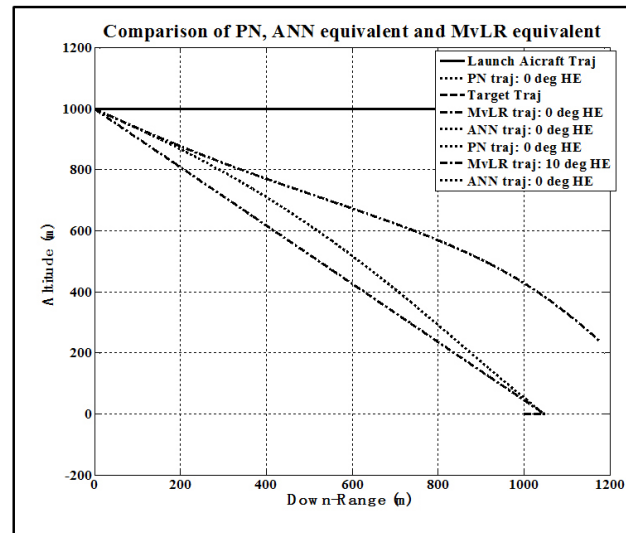


Figure 12: RMS error comparing regression types with PN

B. Results and observations

In Figure 12, the multi-variate linear regression performed well for the collision course case which is the scenario where there was no heading error applied during missile launch. In this case, the MLP ANN also performed just as well in matching results with using classical PN. When a heading error of 10 degrees in elevation was applied however, the performance of the MvLR became worse in comparison with PN. This is depicted in both Figure 12 and Figure 13. This is because an ordinary least squares approach was used when estimating parameters for the MvLR and it has a limitation in being able to estimate the coefficients for the regression.^[41] Meanwhile, the ANN with 5 neurons in hidden layer, performed much better in matching the PN trajectory. It has been deferred to a future study to investigate improving the classical regression.

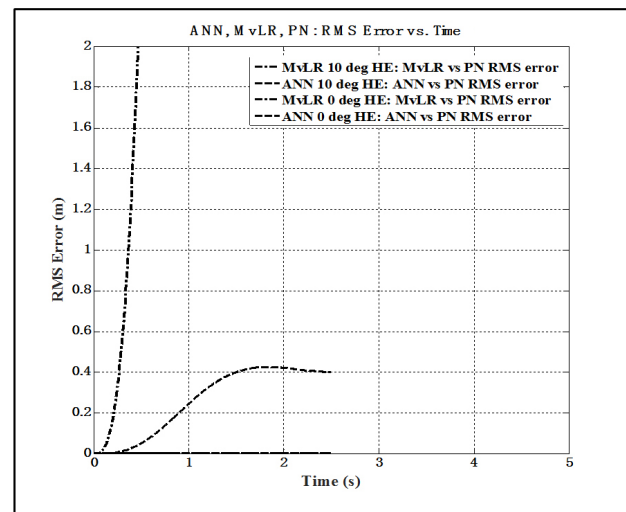


Figure 13: RMS error comparing PN and regression varieties

Table 3: Execution times recorded for executing PN, MvLR, and MLP ANN respectively

PN Guidance Law Type	Guidance Law Execution Time per iteration (seconds)
PN	1.4463e-05
Multi-Variate Linear Regression (MvLR)	4.1252e-06
MLP ANN	0.00013943

5.3.4 Simulation 4: Selection of neurons and hidden layers

A. Description

Researchers such as Ref.^[14] have suggested to use of analytical formulas for selecting number of hidden layers and neurons per hidden layer. For this paper, the authors have instead resorted to discover the best combination of hidden layers and nodes via trial and error.

B. Results and observations

Figure 14 shows two plots. The top plot appears to show that there is no difference brought about by varying number of hidden layer neurons. However, the bottom plot with its finer y-axis scale does show that RMS of the difference between PN and ANN does change as the number of hidden layer neurons is varied. One hidden layer neuron produces a large error considering the scale of millimeters used by the bottom plot y-axis.

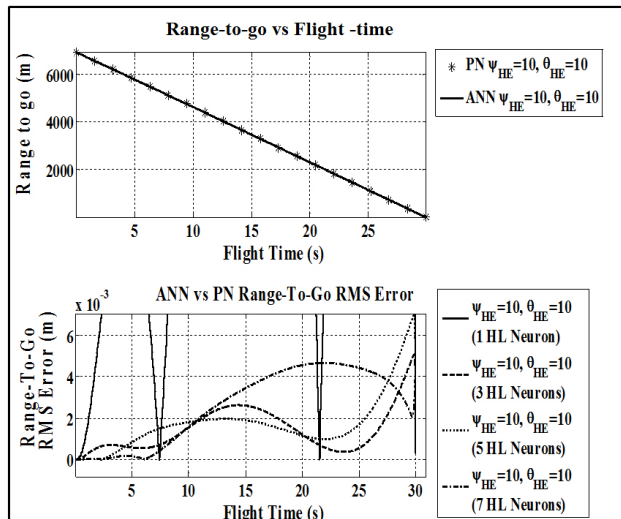


Figure 14: Effect when hidden layer neurons varied

Increasing the number of hidden layer neurons from 1 to 3 to 5 to 7 does decrease the RMS error compared to just using 1 hidden layer neuron. However, when looking at the bottom plot of Figure 14, it was still unclear as to how many additional neurons sufficed as adequate to produce a sufficiently small RMS error. In order to resolve this, it was decided to repeat the simulation for the cases of 3, 5, and 7 hidden layer neurons 3 times each. For each of the three runs, the sample being used of training was varied via collecting a different set of training data by changing the random sampler configuration. The results collected in this manner for the case of running 3 times, the scenario with 3 hidden layer neurons is shown below in Figure 15. This was done for all choices of hidden layer neurons, i.e. 5 and 7. The results are displayed in Table 4.

The observation from these results was that three neurons in

the hidden layer were the adequate. This was because; increasing the number of neurons in the hidden layer beyond three neurons did not cause the average RMS error to decrease significantly. Also, increasing the number of hidden layer neurons led to longer computation time for training which needed to be avoided as much as possible.

The reason why, a small number of neurons was adequate was because, the classical PN guidance law being substituted had a linear relationship between line of sight rate and commanded acceleration. The number of hidden layer neurons did not need to be altered when varying heading error. This was because the linear relationship for classical PN would not be violated even if heading error were changed. However, this needed to be verified.

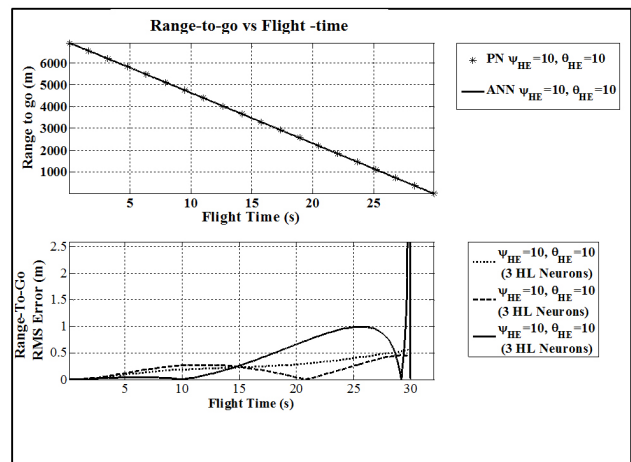


Figure 15: Selecting number of hidden layer neurons

Table 4: Comparing MLP training algorithm performance for small heading error

10 degrees Heading Error				
Training Algorithm	Training Time (s)	Execution Time (s)	Flight Time (s)	Max RMS error deviation compared to PN (m)
Trainrp	2.3331	0.0001384	1092.63	>10
Trainlm	127.5842	0.00013813	1046.38	0.3
Trainbr	244.302	0.00013861	1048.35	0.4
PN	N/A	0.000015	1048.09	0.0

5.3.5 Simulation 5: Selecting training algorithm

A. Description

This test was conducted to discover the most appropriate training algorithm for the MLP to use out of a large bucket of supervised training algorithm available from literature and within the Neural network toolbox.^[15] This was not addressed sufficiently in other papers and forms part of the new contribution.

B. Results and observations

LM and BR appear superior compared to RP in matching

closely with PN. This is seen in Figure 16 and Figure 17. However, execution time for LM and BR are much longer compared to RP as shown in Table 4. The accuracy is matching PN is critical, hence LM and BR are better candidates for choice of training algorithm compared to RP.

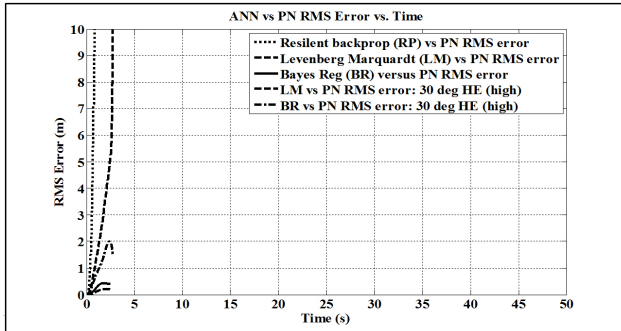


Figure 16: RMS error comparison between approximation approaches

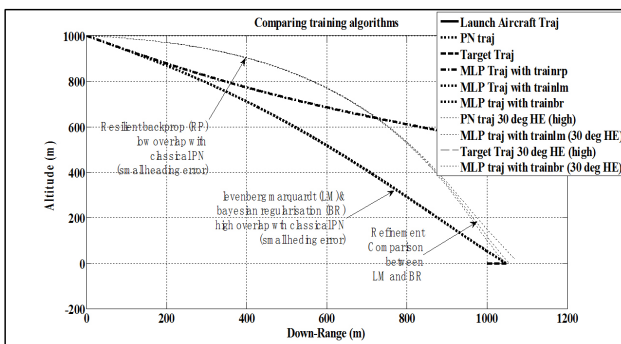


Figure 17: Comparison of MLP training algorithms

Further selection between BR and LM needed to be performed. In order to this, a large heading error was applied to PN and then BR and LM were used for substitution. As Table 4 and Figure 17 show, LM was outperformed significantly by BR in terms of lower RMS error even if it took slightly longer to complete. For a smaller heading error, LM did outperform BR as shown in Figure 17. However, it was only marginally better whereas with heading error BR outperformed LM by a very large difference. For this reason, BR was selected as training algorithm. The reason was attributed to the better generalization performance of BR compared to LM. When the trained network is applied, the data it sees is effectively new data as the differences between the MLP outputs and PN outputs drift apart slightly as iterations progress. So the MLP based guidance system has to generalize correctly in the presence of new data which means that generalization performance becomes important. Hence the BR algorithm with its superior generalization performance works better than LM. For this reason BR was finally selected as choice of MLP training algorithm.

5.3.6 Simulation 6: Varying heading error

A. Description

In order to determine the degree of sensitivity to changes in initial conditions, the missile heading error with respect to the computed collision course was varied in azimuth and elevation between -30 and 30 degrees. This is depicted in Figure 18.

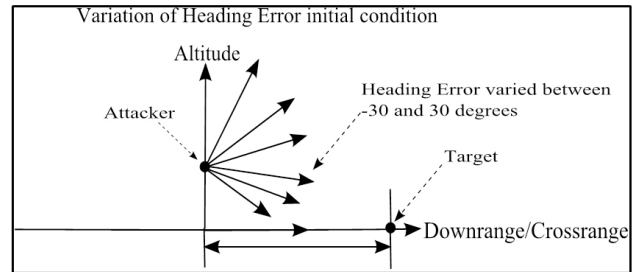


Figure 18: Sensitivity analysis testing plan

B. Results and observations

For each of the heading errors applied, PN and ANN results follow closely as shown in Figure 19. This is because the ANN and PN trajectories match closely in the diagrams shown. This result demonstrates that ANN guidance is insensitive to change in heading error within the ranges tested. The ranges tested are typical of missile heading errors that are possible during real-life air-to-ground engagements.

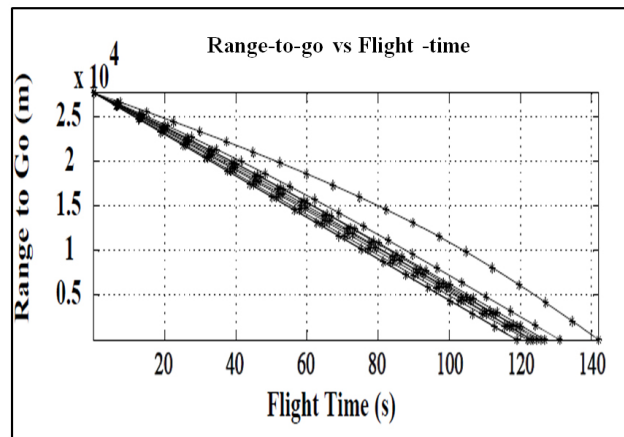


Figure 19: Impact of heading error variation

5.3.7 Simulation 7: Applying large heading error

A. Description

This test was devised in order to assess whether MLP ANN results could still match closely with classical PN for the rare scenario where a large heading error was applied. In this case, the heading error was increased to 40 degrees in both azimuth and elevation.

B. Results and observations

The bottom plot in Figure 20 shows that the match between classical PN and MLP ANN remains close with only small RMS error throughout the flight. This observation was used to conclude that the MLP ANN devised, robustly substituted classical PN.

5.3.8 Simulation 8: Performance of ANN outside training scenarios

A. Description

The test was performed to see if an MLP ANN trained for one choice of initial separation range (3,464 m) would suffice even when applied to situation of larger separation range. Three initial separation ranges were trialed.

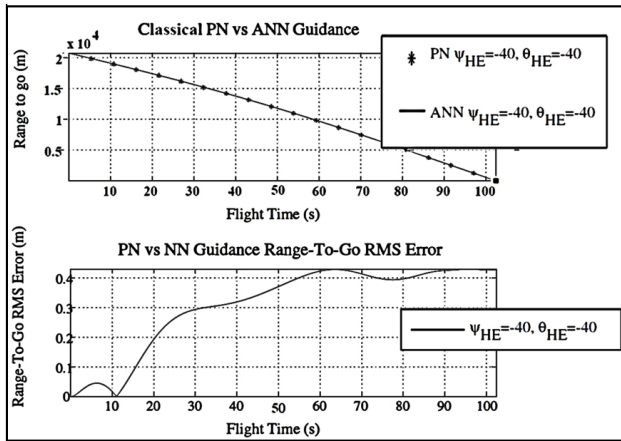


Figure 20: Impact of 40 degrees heading error

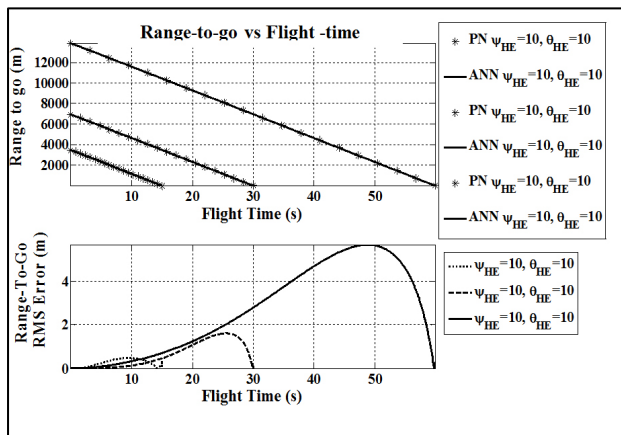


Figure 21: Generalization test applying symmetric range scaling

B. Results and observations

As the top plot in Figure 21 shows, an MLP ANN trained with data collected for just one choice of separation range

(3,464 m) was still able to closely match the classical PN results for different separation ranges. It can be observed, however that, there is an increase in error when the network is applied for engagements where it has not been trained. In this case, the increase in error increases to a maximum of around 5 m sometime during the middle of the flight, as seen in the bottom plot of Figure 21. This may be still acceptable as it is not a large increase in error. The results provided in Table 3 associated with the figure, further confirm this since PN and ANN miss times and miss values closely match. Thus, it was concluded that robustness of the ANN was maintained in substituting classical PN. The implication of being able to only needing training for one separation range was that it provides for savings in training time for multiple scenarios. When designing an ANN guidance substitute for classical guidance law for a wide range of scenarios, the time savings for training may be substantial and lead to significant cost savings.

5.3.9 Simulation 9: Further generalization testing

A. Description

This test was used to determine sensitivity of a trained neural network’s performance when the initial conditions were varied in the close neighbourhood of the conditions for which training was applied.

B. Results and observations

Figure 23 depicts the performance of a network trained just once to cope with small changes in launch conditions. The small changes applied with minor deviations in initial down-range and cross range values. These changes are summarized in Table 6. The training time for the network recorded here was approximately 712 seconds. The execution time following training was 0.0021 seconds. Since the network could be successfully reused with small RMS in all cases as depicted in Table 7, a saving of $4 \times 712 = 2828$ seconds could be achieved. This is a large timesaving and suggests that complete substitution of PN via the MLP ANN across a range of scenarios is achievable at low computation cost. In order to get more confidence about need for small amount of training time required to substitute PN, some additional simulations were performed as depicted in Figure 22.

Table 5: Comparing MLP training algorithm performance for large heading error

30 degrees Heading Error				
Training Algorithm	Training Time (s)	Execution Time (s)	Flight Time (s)	Max RMS error deviation compared to PN (m)
Trainlm	36.2644	0001426	1068.50	>10
Trainbr	107.0161	0.00014647	1053.54	2
PN	N/A	0.000015	1047.02	0.0

In Figure 22, the dashed lines represent the PN trajectories and the unbroken lines, the attempts with equivalent MLP

ANN substitutes. Apart from the scenario of carrying out training, in all other cases there is significant deviation between the ANN and PN trajectories. So this means that generalization via avoiding retraining does not work well even if there is match for some portion of the trajectory in each case. The large RMS error difference between the PN and MLP ANN shown in Table 6 further reinforce this. This means that savings in training time is limited when attempting to substitute PN. In order to work-around this limitation, it is necessary to understand how to compare training data matrices for two separate trajectories via using concepts such as norm^[40] and/or using dynamic time warping algorithms^[43] in order to understand how to simply train for the difference in data sets and thus reducing training time. Apart from this, choosing the training data carefully to be representative of range for the PN function can mean avoiding need for further training. Such a procedure requires detailed research in itself as can be seen with another example in Ref.^[44]

Table 6: Generalization when perturbing about case where training was applied

Network training applied or reused without training? (Yes/No-reused with training)	Initial range to target Down-range (m)	Initial range to target Cross-range (m)	Max RMS error deviation (m)
Yes	2000	2150	0.0002
No-reused	2040	1500	0.0004
No-reused	2040	2490	0.0004
No-reused	2150	1000	1.2
No-reused	1700	3000	2.3

Table 7: Results when launching missile from different directions w.r.t. target

Network training applied or reused without training? (Yes/No-reused with training)	Initial range to target Down-range (m)	Initial range to target Cross-range (m)	Max RMS error deviation approx. (m)
Yes	2000	2000	0.0002
No-reused	2000	0.0	200
No-reused	0.0	2000	>200
No-reused	2000	-2000	>200
No-reused	0.0	-2000	>200
No-reused	-2000	-2000	>200
No-reused	-2000	0.0	>200
No-reused	-2000	2000	200

Here the authors have pointed out that for a training dataset to be effective in reducing training it needs to display properties such as:

- Being able to capture as much of the number of patterns that subgroups of data can be associated with
- Being able to adequately represent statistical variation of inputs and parameters

When training offline, PN and other guidance laws can be scrutinized to help generate data that complies with the desirable training data requirements stated. In some instances, due to constraints, only a limited training data set may be employable, such as for example when considering on-line data collection and learning where very small time constraints to re-tune network during mission may in turn constrain correct mappings to guidance law demands in the context of new data that network has not been trained with to be presented to the MLP guidance law network. In summary, without extra intervention, the generalization achievable when training for a single case is limited to scenario where separation range is simply scaled or in nearby vicinity of case for which training was applied.

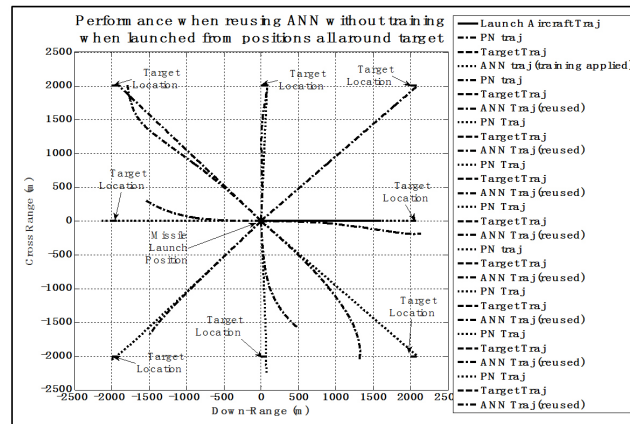


Figure 22: Generalization test when varying approach direction

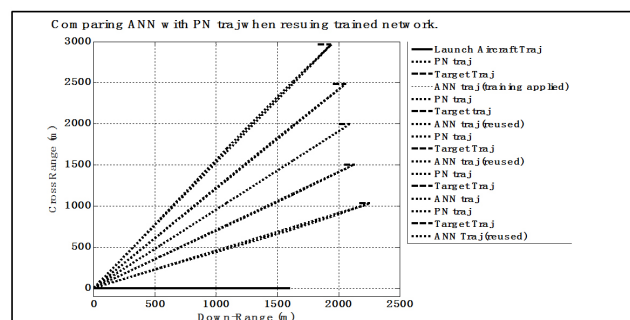


Figure 23: Generalization test in neighborhood of training scenario

6 Conclusion

The focus of this paper was to develop and evaluate a guidance law using Artificial Neural Networks that is able to re-

place classical PN. This has been demonstrated to be achievable via a series of simulations applied to a scenario with some minor variations where classical PN is usually applied to guide an air-to-ground missile to intercept a slow moving (constant velocity) ground target.

The simulation results demonstrate that ANN results follow closely with PN for a wide range of initial conditions. Any differences encountered were overcome via strategies to change to number of hidden layer neurons, fine-tuning the application of the training algorithm when generating the simulation results. Refinements in selecting training data have been withheld for this study because that requires detailed treatment of spanning all data patterns encompassed by PN or any other classical guidance laws being upgraded. The considerations for such a treatment have been listed but have been left for a future research project.

From a theoretical point of view, the authors here have researched the foundations of whether ANN internal behaviour can be explained. This was discovered to be possible via the literature survey results provided in Ref.^[27] The visual proof of ANN as being able to substitute any continuous function with a single hidden layer and sufficient hidden layer neurons to bring the accuracy up to prescribed standard was discovered minimization of MSE could be established as a mathematically sound non-linear regression method via consulting the universal approximation theorem Ref.^[8,45] Previously, this has been a limiting factor for many researchers to adopt ANN for emulating missile guidance

laws because of the safety critical nature of application with strict requirements to minimize collateral damage.

Finally, there was testing performed to investigate the generalization potential of MLP based missile guidance laws. It was demonstrated that an MLP was able to substitute for a subset of PN behaviour by just training once and without requiring further retraining. However, without further refinement of the methodology, many training runs are still required to span PN as the launch conditions are varied. It was identified that researchers would need to investigate the similarity between two or more sets of training data via using concepts such as the norm or dynamic time warping algorithms and so on in order to then understand how to minimize training time for associated MLP that correctly provide mappings for the training data. This is a research pathway for the future.

While the result of classical PN being substituted successfully via an MLP ANN was shown as possible, the real payoff for making use of ANN in place of classical guidance laws would come about only if it was able to be used in circumstances where the analytical guidance law being emulated is not suitable for real-time application. This is not a limitation with classical PN whereas it could be the case with advanced classical guidance laws such as OGL.^[14] If the ANN Guidance law approximation were to successfully apply in such cases, it would pave way for genuine paradigm expansion for applying advanced missile guidance laws into the practical applications.

References

- [1] Zarchan P. Tactical and strategic missile guidance: American Institute of Aeronautics and Astronautics; 1997.
- [2] Cloutier JR, Evers JH, Feeley JJ. Assessment of air-to-air missile guidance and control technology. 1989. <http://dx.doi.org/10.1109/37.41440>
- [3] Spezio AE. Electronic warfare systems. Microwave Theory and Techniques, IEEE Transactions on. 2002; 50(3): 633-44. <http://dx.doi.org/10.1109/22.989948>
- [4] Negnevitsky M. Artificial intelligence: a guide to intelligent systems: Pearson Education; 2005.
- [5] Haykin S, Network N. A comprehensive foundation. Neural Networks. 2004; 2(2004).
- [6] Palumbo NF, Blauwkamp RA, Lloyd JM. Modern homing missile guidance theory and techniques. Johns Hopkins APL Technical Digest. 2010; 29(1): 42-59.
- [7] Cybenko G. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems. 1989; 2(4): 303-14. <http://dx.doi.org/10.1007/BF02551274>
- [8] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural networks. 1989; 2(5): 359-66. [http://dx.doi.org/10.1016/0893-6080\(89\)90020-8](http://dx.doi.org/10.1016/0893-6080(89)90020-8)
- [9] Hagan MT, Demuth HB, Beale MH. Neural network design: Pws Boston; 1996.
- [10] Cheng B, Titterton DM. Neural networks: A review from a statistical perspective. Statistical science. 1994: 2-30.
- [11] Wang S. An insight into the standard back-propagation neural network model for regression analysis. Omega. 1998; 26(1): 133-40.
- [12] Martin T. Hagan OSU, Howard B. Demuth UoI, Mark Hudson Beale M. Inc Improve Neural Network Generalization and Avoid Overfitting. 2013b ed: MathWorks 2014. p. Retrieved from Neural Network Toolbox User's Guide (2014a).
- [13] Borovicka T, Jirina Jr M, Kordik P, et al. Selecting representative data sets. 2012. <http://dx.doi.org/10.5772/50787>
- [14] Choi H-L, Lee H-G, Tahk M-J, et al. A Three-dimensional differential game missile guidance law using neural networks. 2001.
- [15] Demuth H, Beale M. Neural network toolbox for use with MATLAB. 1993.
- [16] Dalton J, Balakrishnan S. A neighboring optimal adaptive critic for missile guidance. Mathematical and computer modelling. 1996; 23(1): 175-88. [http://dx.doi.org/10.1016/0895-7177\(95\)00226-X](http://dx.doi.org/10.1016/0895-7177(95)00226-X)
- [17] Bertsekas DP. Approximate dynamic programming. 2011.
- [18] Savkin AV, Pathirana PN, Faruqi FA. Problem of precision missile guidance: LQR and H control frameworks. Aerospace and Electronic Systems, IEEE Transactions on. 2003; 39(3): 901-10.
- [19] Yu J, Xu Q, Zhi Y, et al. A self-adaptive region fuzzy guidance law based on RBF neural network for attacking UAV. Computer Research and Development (ICCRD), 2011 3rd International Conference on; 2011: IEEE.
- [20] Klir G, Yuan B. Fuzzy sets and fuzzy logic: Prentice Hall New Jersey; 1995.

- [21] Rahbar N, Bahrami M. Synthesis of optimal feedback guidance law for homing missiles using neural networks. *Optimal Control Applications and Methods*. 2000;21(3):137-42. [http://dx.doi.org/10.1002/1099-1514\(200005/06\)21:3%3C137::AID-0CA668%3E3.0.CO;2-E](http://dx.doi.org/10.1002/1099-1514(200005/06)21:3%3C137::AID-0CA668%3E3.0.CO;2-E)
- [22] ZHOU R. Design of closed loop optimal guidance law using neural networks. *Chinese Journal of Aeronautics*. 2002; 15(2): 98-102. [http://dx.doi.org/10.1016/S1000-9361\(11\)60137-4](http://dx.doi.org/10.1016/S1000-9361(11)60137-4)
- [23] Perh D. A study into advanced guidance laws using computational methods. California: DTIC Document, MONTEREY NPS; 2011.
- [24] Song S, Hong J, Ha I, et al. L₂ Stability and Performance Analysis of Missile Systems with Linear Autopilot and PN Guidance Laws. 2005.
- [25] Rojas R. *Statistics and Neural Networks*. Berlin: Springer-Verlag; 1996 [cited 2014 July 2014].
- [26] Rowland T. L₂ Function: Wolfram Research; 2014 [cited 2014 August 2014]. Available from: <http://mathworld.wolfram.com/L2-Function.html>
- [27] Chen T, Chen H, Liu R-w. A constructive proof and an extension of Cybenko's approximation theorem. *Computing science and statistics*: Springer; 1992. p. 163-8.
- [28] Hassoun MH. *Fundamentals of Artificial Neural Networks*: MIT Press; 1995.
- [29] Medin DL, Schwanenflugel PJ. Linear separability in classification learning. *Journal of Experimental Psychology: Human Learning and Memory*. 1981; 7(5): 355. <http://dx.doi.org/10.1037/0278-7393.7.5.355>
- [30] Blum AL, Langley P. Selection of relevant features and examples in machine learning. *Artificial intelligence*. 1997; 97(1): 245-71. [http://dx.doi.org/10.1016/S0004-3702\(97\)00063-5](http://dx.doi.org/10.1016/S0004-3702(97)00063-5)
- [31] LeCun Y, Denker JS, Solla SA, et al. Optimal brain damage. *NIPS*; 1989.
- [32] Chen DS, Jain RC. A robust backpropagation learning algorithm for function approximation. *Neural Networks, IEEE Transactions on*. 1994; 5(3): 467-79. <http://dx.doi.org/10.1109/72.286917>
- [33] Rojas R. *The Backpropagation Algorithm*. *Neural Networks: A systematic Introduction*. 1996.
- [34] Collantes WA. *Faster Training*. academia.edu, 2014.
- [35] Hao Yu BMW. *Levenberg-Marquardt Training*. 2011 [cited 2014 June 2014].
- [36] Girosi F, Jones M, Poggio T. Regularization theory and neural networks architectures. *Neural computation*. 1995; 7(2): 219-69.
- [37] Vapnik V, Levin E, Le Cun Y. Measuring the VC-dimension of a learning machine. *Neural Computation*. 1994; 6(5):851-76.
- [38] Mathias R. The spectral norm of a nonnegative matrix. *Linear Algebra and its Applications*. 1990; 139: 269-84.
- [39] Vidal Ruiz E, Casacuberta Nolla F, Rulot Segovia H. Is the DTW "distance" really a metric? An algorithm reducing the number of DTW comparisons in isolated word recognition. *Speech Communication*. 1985; 4(4): 333-44.
- [40] Kreyszig E. *Advanced engineering mathematics*: John Wiley & Sons; 2007.
- [41] White H. Using least squares to approximate unknown regression functions. *International Economic Review*. 1980: 149-70.
- [42] Stathakis D. How many hidden layers and nodes? *International Journal of Remote Sensing*. 2009; 30(8): 2133-47. <http://dx.doi.org/10.1080/01431160802549278>
- [43] Müller M. Dynamic time warping. *Information retrieval for music and motion*. 2007: 69-84.
- [44] Thirumalainambi R, Bardina J. Training data requirement for a neural network to predict aerodynamic coefficients. *AeroSense 2003*; 2003: International Society for Optics and Photonics.
- [45] Specht DF. A general regression neural network. *Neural Networks, IEEE Transactions on*. 1991; 2(6): 568-76. <http://dx.doi.org/10.1109/72.97934>