Latin American Journal of
Solids and Structures
www.lajss.org

# Tip Deflection Determination of a Barrel for the Effect of an Accelerating Projectile Before Firing Using Finite Element and Artificial Neural Network Combined Algorithm

## Abstract

For realistic applications, design and control engineers have limited modelling options in dealing with some vibration problems that hold many nonlinearity such as non-uniform geometry, variable velocity loadings, indefinite damping cases, etc. For these reasons numerous time consuming experimental studies at high costs must be done for determining the actual behaviour such nonlinear systems. However, using advantages of multiple computational methods like Finite Element Method (FEM) together with an Artificial Intelligence (ANN), many complicated engineering problems can be handled and solved to some extent. This study, proposes a new collective method to deal with the nonlinear vibrations of the barrels in order to fulfil accurate shooting expectancy. Using known analytical methods, in practical, to determine dynamic behaviour of the barrel beam is not possible for all conditions of firing that include numerous varieties of ammunition for different purposes, and each projectile of different ammunition has different mass and exit velocity. In order to cover all cases this study proposes a new method that combines a precise FEM with ANN, and can be used for determining the exact dynamic behaviour of a barrel for some cases and then for precisely predicting the behaviour for all other possible cases of firing. In this study, the whole nonlinear behaviour of an antiaircraft barrel were obtained with 3.5% accuracy errors by ANN trained by FEM using calculated analysis results of ammunitions for a particular range. The proposed FEM-ANN combined method can be very useful for design and control engineers in design and control of barrels in order to compensate the effect of nonlinear vibrations of a barrel for achieving a higher shooting accuracy; and can reduce high-cost experimental works.

## Keywords

Nonlinear vibration modelling, Vibration of continuous systems, Artificial neural networks, Gun barrels, Finite element method

Mehmet Akif Koç [a]
İsmail Esen [b]
Yusuf Çay [c]

[a] Department of Mechanical Engineering, Sakarya University, Sakarya, Turkey. makoc@sakarya.edu.tr
[b] Department of Mechanical Engineering, Karabük University, Karabük, Turkey iesen@karabuk.edu.tr
[c] Department of Mechanical Engineering, Sakarya University, Sakarya, Turkey. ycay@sakarya.edu.tr

http://dx.doi.org/10.1590/1679-78252718

Received 17.12.2015
In revised form 05.04.2016
Accepted 06.05.2016
Available online 18.05.2016

# 1 INTRODUCTION

The non-linear dynamic behaviour of a structure due to the effect of an accelerating mass is still a research interest with the applications of it in the new fields such as defence and transportation engineering. The effects of moving loads on the dynamics of the structures have been widely studied in literature, for example, (Dehestani et al., 2009; Lee, 1996; Michaltsos, 2002; Niaz and Nikkhoo, 2015; Omolofe and Oni, 2015; Wang, 2009) have investigated the subject. For the application to the bridge engineering (Michaltsos et al., 1996) have studied the effect of accelerating vehicles on the bridge beams, considering highway bridges and high speed rail road construction. Some more accurate tools of engineering calculations of the dynamic interaction using FEM have been proposed by(Esen, 2011, 2015, 2013; Kahya, 2012) . Using analytical methods for simple cases neglecting damping effects and assuming uniform beam cross sections (Esmailzadeh and Jalili, 2003; Liu et al., 2015; Wyss et al., 2011) have studied the subject in terms of vehicle structure interaction problems. Using a two-axle half car model, (Lou, 2005) has studied the wheel - rail interaction and dynamics of a railway bridge. For both simple and realistic models of moving vehicles can be found in(Azimi et al., 2013; Lou and Au, 2013; Yang et al., 2013).

The moving mass problem is also vital in defence field, but the studies in this field are rarely available in literature due to the confidentiality. Using a precise FEM model(Esen and Koç, 2015a) have presented the interaction of an accelerating projectile and a barrel of a cannon. Effect of stepped barrels on the stability and the dynamics of barrels have been investigated by (Balla, 2011; Tawfik, 2008). In order to understand interaction between projectile and barrel, (Alexander, 2007) has prepared an ABAQUS explicit dynamic finite element model, and then compared analysis results to the test data 155 mm cannon. M. Stiavnicky and P. Lisy, 2013 have investigated numerical simulation to determine influence of the barrel fixing on barrel vibration when bullet exits barrel. Another aspect of reducing the vibrations is the usage of dynamic vibration absorbers, and (Esen and Koç, 2015b; Kathe, 1997; Littlefield et al., 2002) have studied the vibration reduction of barrels.

ANN also known as ‘parallel distributed processing’ is a powerful artificial intelligence for solving complicated engineering problems. This method can be applied to predict the desired output parameters when the database of the problem represents all relationships. ANN have been used in different engineering applications such as mechanical vibrations (Koide et al., 2014; Lagaros and Papadrakakis, 2012; Liu et al., 2015; Martínez-Martínez et al., 2015; Perez-Ramirez et al., 2016) rail rolling processing (Altınkaya et al., 2014), creep modelling (Düğenci et al., 2015), steel projectile penetration depth (Hosseini and Dalvand, 2014) and internal combustion engines to estimate some important parameters of fuels on emissions (Cay, 2013; Czarnigowski, 2010). The uses of ANN in the field of defence systems have recently begun to increase. ANN models have played an important role in the development of military automatic target recognition (ATR) (Rogers et al., 1995).

Considering all the nonlinearity such as projectile/barrel interaction, different masses, and exit velocities of projectiles, non-uniform barrel cross-sections, inertia effects of the accelerating mass, precise damping model of whole system, the design engineers need more methods that are accurate in order to determine actual behaviour to satisfy the perfect shooting expectancy. In practical, to determine the dynamic behaviour of a barrel beam is not possible for all conditions of firing. For example, for a tank system, there are various classes of ammunition for different purposes, and each projectile of different ammunition has different mass and exit velocity. In order to cover all cases, this study proposes a new method that combines a precise FEM with an artificial intelligence technique ANN, and can be used for determining the dynamic behaviour of the barrel for some cases and then for precisely predicting the behaviour for all other possible cases of firing. The proposed FEM-ANN combined method can be very useful for control engineers in design of fire control algorithm of weapon in order to compensate the effect of nonlinear vibration of a barrel for achieving a higher shooting accuracy. There should be many compensation sub systems in a weapon system in order to satisfy the shooting accuracy that is the most important property in such systems. However, in order to design a proper compensation system, engineers need very large data about the dynamic behaviour of the barrels. The needed data can be created by means of experimental studies, but experimental studies are generally time-consuming and expensive. As an alternative to the experimental studies, one of the economic ways of creating accurate data is the modelling using the prediction power of artificial intelligence techniques. In this study, the mass and exit velocity of a projectile are used as input, while the tip deflection as output, and the predictions of the deflections have been achieved with an acceptable accuracy. Where, the $R^2$ is 0.99 for training and testing; the MSE for training is $8.25x10^{-4}$, for testing is 0.03767; the MEP, for training is 0.5%, for test is 0.1%.

Without omitting all the nonlinearities including damping, this method can also easily be adapted to other problems of the structural dynamics such as vehicle bridge interaction, wheel/rail interaction, high-speed precise machining, and flexible run-ways of robotic systems, etc. Being capable of predicting the nonlinear behaviour for many cases, this technique can reduce research and development costs by reducing costly and time-consuming experimental studies.

## 2 MATHEMATICAL MODELLING

In the formulation, the following assumptions will be adopted (Fig. 1)

- The mass inertia is considered.
- The mass is always in contact with the beam.
- The beam is thin and small displacements in the beam occurred according to thin beam theory.
- The beam is of variable thickness and the material properties are constant trough length of the beam.
- The trajectory of the mass is defined by time-dependent $x_p(t)$

Based on the above assumptions, the motion equation of the barrel beam due to the effect of the projectile located at the time-dependent point $x_p$ within the barrel beam, is provided by Eq. (1) (L. Fryba, 1999):

$$\begin{gathered} EI(x)\frac{\partial^4 w_z(x,t)}{\partial x^4} + \rho A(x)\frac{\partial^2 w_z(x,t)}{\partial t^2} + \sigma\frac{\partial w_z(x,t)}{\partial t} = F(x,t), \\ F(x,t) = m_p g\delta(x - x_p(\mathrm{t})) - m_p\left(\frac{\mathrm{d}^2 w_z(x_p,t)}{\mathrm{d}t^2}\right)\delta(x - x_p(\mathrm{t})) \end{gathered} \tag{1}$$

The left hand side of Eq. (1) represents the resisting internal stiffness, inertia and damping forces due to the external forces on the right side. Where, ρ is the density, *A(x)* is the none-uniform cross-sectional area, $\sigma$ is the equivalent viscous damping coefficient, $E$ is the Young's modulus of elasticity; *I(x)* is the area moment of inertia, while $x$ represents the central coordinate of the barrel system; $t$ represents time; *w(x, t)* is the vertical displacement of the barrel; $m_p$ is the mass of the projectile; $m_p g\delta(x - x_p(\mathrm{t}))$ is the force applied to the unit length of the barrel by the projectile (as a moving mass); while $g$ and $\delta$ represent gravitational acceleration and the Dirac delta function, respectively; and $\mathrm{d}^2 w(x_p, t)/\mathrm{d}t^2$ represents the acceleration of the barrel at the contact point of lumped projectile mass. For inclined positions of the barrel beam one can refer to the study by Esen and Koç, 2015b.

The initial and boundary conditions of the barrel beam are:

$$w(x,t=0)=\frac{\partial w(x,t=0)}{\partial t}=0 \tag{2a}$$

$$\begin{aligned} &w(x=0,t)=0, \frac{\partial w(x=0,t)}{\partial x}=0 \\ &\frac{\partial^2 w(x=L,t)}{\partial x^2}=0,\ \frac{\partial^3 w(x=L,t)}{\partial x^3}=0 \end{aligned} \tag{2b}$$

In order to determine time dependent displacements of the barrel tip, which is very important for shooting accuracy of the weapon, a rough analytical solution of the motion Eq. (1) can be obtained through some simplifications by ignoring the effects of inertia and damping, and accepting that the cross-section area is uniform and the projectile moves with a constant velocity. For simplified cases, that omits geometric and dynamic nonlinearities, such moving load problems have been extensively studied in the literature by numerous researchers. From this perspective, the proposed FEM-ANN combined method can be very useful for design and control engineers if they pay attention to the requirements of modelling as described below.

The model of the barrel with an accelerating projectile is shown in Figure 1. The interaction of the barrel and the projectile are determined in both vertical and horizontal directions.

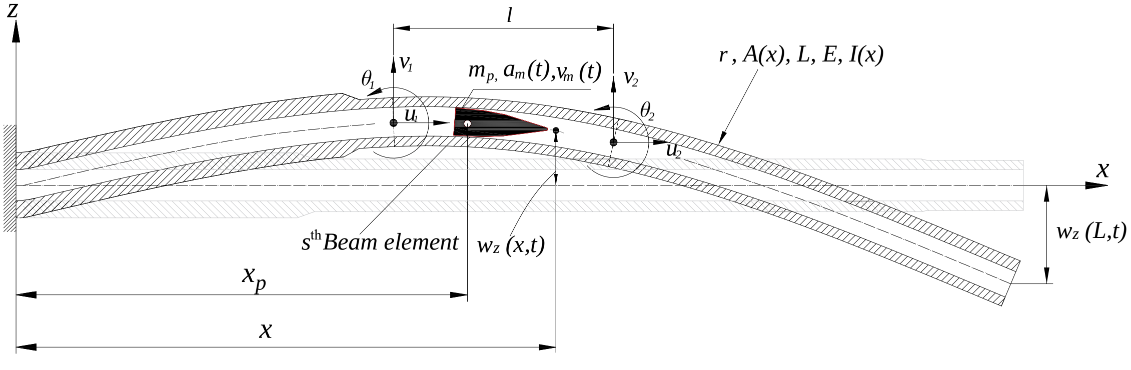

**Figure 1**: Model of a barrel with an accelerating projectile.

While the barrel is vibrating, the transverse and longitudinal interaction forces between the barrel and the projectile that is accelerating trough the deflected barrel geometry can be determined by using the following equation (Cifuentes, 1989; L. Fryba, 1999):

$$f_z(x,t) = [m_p g - m_p \left[ \begin{matrix} \dfrac{\partial^2 w_z(x,t)}{\partial t^2} + 2\dfrac{\partial^2 w_z(x,t)}{\partial x\,\partial t}\dfrac{\mathrm{d}x_p}{\mathrm{d}t} \\ + \dfrac{\partial^2 w_z(x,t)}{\partial x^2}\left(\dfrac{\mathrm{d}x_p}{\mathrm{d}t}\right)^2 + \dfrac{\partial w_z(x,t)}{\partial x}\dfrac{\mathrm{d}^2 x_p}{\mathrm{d}t^2} \end{matrix} \right]_{x=x_p} ]\delta(x-x_p)$$

$$f_x(x,t) = m_p \frac{\mathrm{d}^2 w_x(x_p,t)}{\mathrm{d}t^2}\delta(x-x_p) \tag{3}$$

$$x_p = x_0 + v_0 t + \frac{a_m t^2}{2};\ \frac{\mathrm{d}x_p}{\mathrm{d}t} = v_0 + a_m t;\ \frac{\mathrm{d}^2 x_p}{\mathrm{d}t^2} = a_m$$

In above equation $f_z(x,t)$ and $f_x(x,t)$ respectively, are transverse and horizontal contact forces between the barrel and the projectile accelerating at point $x_p$ on the axis, while $t$ represents time, while $\delta(x\text{-}x_p)$ and $g$ are the Dirac-delta function and the gravitational acceleration, respectively. The parameters $x_0$ and $v_0$ are the initial position and initial speed of the projectile at time $t=0$, respectively. On the other hand, $a_m$ is the average acceleration of the projectile within the barrel. Under the effect of the accelerating projectile, the equivalent nodal forces of the barrel element (Figure 2 a and b) and the relationships between the shape functions and the transverse and longitudinal deflection functions and the nodal displacements of the $s^{th}$ element at position $x_m(t)$ at time $t$ are as follows:(Clough R.W; Penzien J., 2003):

$$\begin{gathered} f_i = \psi_i m_p \ddot{w}_x \quad (i=1,\ 4) \\ f_i = \psi_i m_p(\ddot{w}_z + 2\dot{w}_z'(v_0 + a_m t) + w_z''(v_0 + a_m t)^2 + a_m w_z' + g) \qquad (i=2,\ 3,\ 5,\ 6) \\ w_x(x,t) = \psi_1 u_1 + \ \psi_4 u_2 \\ w_z(x,t) = \psi_2 v_1 + \ \psi_3 \theta_1 + \ \psi_5 v_2 + \ \psi_6 \theta_2 \end{gathered} \tag{4}$$

$$\begin{aligned} \psi_1 &= 1 - \xi(t) & \psi_4 &= \xi(t) \\ \psi_2 &= 1 - 3\xi(t)^2 + 2\xi(t)^3 & \psi_5 &= 3\xi(t)^2 - 2\xi(t)^3 \\ \psi_3 &= \left(\xi(t) - 2\xi(t)^2 + \xi(t)^3\right)l & \psi_6 &= \left(-\xi(t)^2 + \xi(t)^3\right)l \\ \xi(t) &= \frac{x_m(t)}{l} \end{aligned} \tag{5}$$

In these expressions, " ′ " and " · " represent the spatial and time derivatives of the displacement function, respectively. In addition, $w_z$=$w_z(x,t)$ and $w_x$=$w_x(x,t)$ represent the vertical displacement ($z$) and Longitudinal ($x$) on the coordinate plane of the barrel at coordinate $x$ and time $t$. The nodal displacements of the beam element are at left node $u_1$, $v_1$, $\theta_1$ and at right node $u_2$, $v_2$, $\theta_2$, that are represent axial displacement, vertical displacement and slope, respectively. $\psi_i$ *(i=1-6)* are the shape functions of the beam element (Clough R.W; Penzien J., 2003) . The length of the element is *l.* and $x_m(t)$ is the variable distance between the accelerating projectile and the left end of the $s^{th}$ element at time t as shown in Figure 2b.

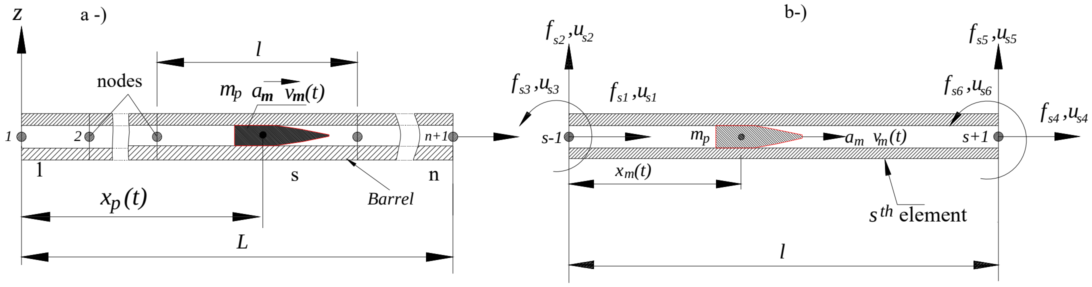

**Figure 2**: Modelling of the barrel and projectile interaction using FEM a-) FEM discretion of the barrel system b-) Beam element s over which the projectile $m_p$ passes at time t.

The property matrices of a beam element in Figure 2, having transverse and longitudinal nodal forces and displacements, can be derived from the procedure of the principle of virtual work and the relation of the kinetic and internal potential energies of the element.

In the case of interaction with a projectile, any stiffness coefficient associated with beam flexure and axial displacements for the element on which the projectile locates is as follows:

$$\begin{gathered} k_{ij} = \int_0^l EI(x)\psi_i''(x)\psi_j''(x)dx + v(t)^2\psi_i\psi_j'' + a_m\psi_i\psi_j', \quad v(t) = v_0 + a_m t, \ \left(i\ , j = 2,3,5,6\right) \\ k_{ij} = -\int_0^l EA\psi_i'(x)\psi_j'(x)dx, \ \left(i, j = 1,4\right) \end{gathered} \tag{6}$$

In the same manner, for the relation between nodal accelerations and resisting inertial forces, the elemental balance equation can be obtained. Including the effect of the projectile, any mass coefficient associated with beam flexural and axial accelerations are as follows:

$$\begin{gathered} m_{ij} = \int_0^l m(x)\psi_i(x)\psi_j(x)dx + m_p\left[0 \quad \psi_2 \quad \psi_3 \quad 0 \quad \psi_5 \quad \psi_6\right]\left[0 \quad \psi_2 \quad \psi_3 \quad 0 \quad \psi_5 \quad \psi_6\right]^T \\ i, j = (2,3,5,6) \\ m_{ij} = \int_0^l m(x)\psi_i(x)\psi_j(x)dx + m_p\left[\psi_1 \quad 0 \quad 0 \quad \psi_4 \quad 0 \quad 0\right]\left[\psi_1 \quad 0 \quad 0 \quad \psi_4 \quad 0 \quad 0\right]^T \\ i, j = (1,4) \end{gathered} \tag{7}$$

The damping in the system can be any type; and for structural systems, hysteric and structural damping may be applied. However, considering practical usage in engineering any type of damping can be modelled as viscous damping using equivalent damping approximations. In this study, the damping is modelled as equivalent viscous damping using Rayleigh's proportional damping theory in which the damping matrix is proportional to the combination of the mass and stiffness matrices; and including the effect of accelerating projectile, the coefficients of time dependent damping matrix can be formed as follows:

$$\left[c_{i,j}\right] = \alpha\left[m_{i,j}\right] + \kappa\left[k_{i,j}\right] + 2m_p v(t)\left[0 \quad \psi_2 \quad \psi_3 \quad 0 \quad \psi_5 \quad \psi_6\right]\left[0 \quad \psi_2' \quad \psi_3' \quad 0 \quad \psi_5' \quad \psi_6'\right]^T$$
$$\alpha = \frac{2\omega_i\omega_j\left(\zeta_i\omega_j - \xi_j\omega_i\right)}{\omega_j^{\,2} - \omega_i^{\,2}}, \quad \kappa = \frac{2\left(\zeta_j\omega_j - \zeta_i\omega_i\right)}{\omega_j^{\,2} - \omega_i^{\,2}} \tag{8}$$

where $\zeta_i$ and $\zeta_j$ are the damping ratios of the structural system for any corresponding natural frequencies $\omega_i$ and $\omega_j$. The instantaneous equation of motion for the entire system is can be expressed as:

$$\left[M\right]\left\{\ddot{U}(t)\right\} + \left[C\right]\left\{\dot{U}(t)\right\} + \left[K\right]\left\{U(t)\right\} = \left\{F(t)\right\} \tag{9}$$

where $[M]$, $[C\ ]$ and $[K\ ]$ are, respectively, the instantaneous overall mass, damping and stiffness matrices, while $\{\ddot{U}(t)\}$, $\{\dot{U}(t)\}$ and $\{U(t)\}$ are, respectively, the acceleration, velocity, and displacement vectors. Besides, $\left\{F(t)\right\}$ is the overall external force vector of the system at time *t*. For the obtaining the matrices of [M], [K], and [C], one can determine the elemental property matrices; and then can assemble them properly using the conventional FEM approach. In case of an accelerating projectile the time dependent elemental matrices of the beam element *s* are determined using the coefficients given Eqs. (6, 7 and 8). For calculation of time dependent property matrices, the instantaneous values of $x_m(t)$ and *s* are:

$$x_m(t) = x_p(t) - (s-1)l \tag{10a}$$

$$s = (\text{integer part of } x_p(t) / L) + 1, \quad s = (1-n) \tag{10b}$$

Embedding the other inertia, centripetal and Coriolis forces in the left side of the system equation, only the vertical gravitational and longitudinal acceleration force components of moving projectile should be applied as external forces; thus, the instantaneous overall force vector becomes as follows:

$$\left\{F(t)\right\} = \left[0 \quad \cdots \quad f_{s1} \quad f_{s2} \quad f_{s3} \quad f_{s4} \quad f_{s5} \quad f_{s6} \quad \cdots \quad 0\right]^T,$$
$$f_{s\,i} = m_p g\phi_i \quad (i = 2,\ 3,\ 5,\ 6), \quad f_{s\,i} = m_p a_m \phi_i \quad (i = 1, 4) \tag{11}$$

## 3 ARTIFICIAL NEURAL NETWORK MODELLING TO PREDICT THE AMOUNT OF TIP DISPLACEMENT

Recent developments have also increased the use of computer systems for military purposes. In the beginning, they are used to perform complex operations in a short time, but today computers are used to make predictions on the relationship between separate events using big data analogous to the human brain. Thanks to the discovery of new information, learning refers to the process of improvement in behaviour in living beings. On the other hand, Machine learning refers to a situation where all of these operations are conducted by a computer. To be able to learn, computers require a dataset on the event in question, thus learning through artificial neural networks requires a training set. Examples in the training set are based on previous experience with the problem in question. Learning is achieved by introducing these examples to the neural network in an order. Artificial neural networks are computer systems that are able to learn and to react to stimuli in the environment by making use of previous examples implemented by humans. At a most basic level, the task of an artificial neural network is to produce a set of outputs that correspond to given inputs. For the artificial neural network to be able to do that, the network needs to be trained by existing examples of events representing the engineering problem in question, and eventually needs to acquire the ability to generalize.

The most important parameter that affects the shooting accuracy of a weapon system is the vertical movement of the muzzle, which happens during shooting, also known as a muzzle displacement. These movements cause the barrel axis to displace, and have an adverse effect on the shooting accuracy of the weapon system. Considering the long ranges of contemporary weapon systems, small muzzle displacements can result in large deviations from the target. Muzzle displacements are affected by two basic parameters. The first of these parameters is the velocity of the projectile as it departs from the muzzle, and the second is the force of gravity perpendicular to the muzzle axis, due to the mass of the projectile. Contemporary firearms, in particular, are designed to have larger projectile masses and higher muzzle velocities in order to be more effective against rapidly moving and manoeuvring targets. However, larger projectile masses and higher muzzle velocities have generated serious problems on the target accuracies of weapons.

To deal with this problem, a number of active and passive control systems have been designed. For example, some studies (Esen and Koç, 2015a; Littlefield et al., 2002) report that when a mass-spring system is added to the muzzle, it is able to decrease muzzle displacement by around 50%. However, this technique was not able to eliminate muzzle displacement, which could be due to a number of reasons. One reason is that the projectile forces the barrel to change its frequency continuously until it departs the muzzle, making it difficult to design an appropriate absorber. In addition, even if an absorber were to be designed that matches the forcing frequency of the projectile, the natural frequencies of the whole system change when the absorber is mounted on the barrel. All of these factors limit the use of passive vibration absorbers on gun barrels. Active control systems have also been designed to prevent muzzles from dipping. However, these systems are very expensive and time consuming, and have limitations of their own.

Different types of ammunition can be used in a weapon system for different purposes. Each type of ammunition would have its own projectile mass and exit velocity. Thus, barrel dynamics would react differently to the different types used. For illustrative purposes, Table 1 reports the different types of ammunition used in a 120 mm tank cannon. As the table shows, projectile mass varies between 5.5 and 12.2 kg, and the exit velocity of the projectile varies between 1140 and 1170 m/s. Similarly, Table 2 reports the characteristics of the different types of ammunition used in 35 mm anti-aircraft cannon.

| | Exit velocity<br>(m/s) | Projectile<br>mass (kg) | Cartridge<br>mass (kg) | Propellant<br>mass (kg) |
|---|---|---|---|---|
| M829A3 | 1555 | 10.00 | 22.3 | 8.10 |
| M830A1 | 1400 | 11.40 | 22.3 | 7.10 |
| M831A1 TP-T | 1140 | 12.20 | 24.0 | 6.35 |
| M865 TPCSDS-T | 1700 | 5.50 | 17.0 | 7.20 |
| M1002 | 1375 | 10.60 | 22.6 | 7.90 |
| M1028 | - | - | - | 7.20 |

**Table 1**: Experimental data of some ammunition for 120 mm tank.

| | Exit velocity<br>(m/s) | Projectile<br>mass (kg) | Propellant<br>mass (kg) |
|---|---|---|---|
| HEI-T | 1175 | 0.535 | 0.33 |
| HEI | 1175 | 0.550 | 0.33 |
| HEI (BF) | 1175 | 0.550 | 0.33 |
| SAOHEI-T | 1175 | 0.550 | 0.33 |
| FAPDS | 1440 | 0.375 | 0.33 |
| TP-T/TP | 1175 | 0.550 | 0.33 |
| AHEAD | 1050 | 0.750 | 0.33 |

**Table 2**: Experimental data of some ammunition for 35 mm anti-aircraft.

Using an artificial neural network, this study aims to predict the amount of tip displacement of a barrel in a weapon system, due to an accelerating projectile. Using this artificial neural network, designed for many different projectile masses and exit velocities, the amount of muzzle displacement that occurs when different types of ammunition are used can be predicted prior to shooting. This way, deviations from the target can be calculated and the needed adjustments can be made for the barrel position to eliminate all deviation.

### 3.1 The Structure of the Artificial Neural Network

An artificial neural network consists of a large number of process elements connected to each other and called parallel process structures. Each process element in turn, consists of five components: inputs, weights, aggregation function, activation function, and outputs. Figure 3 provides the structure of a process element.

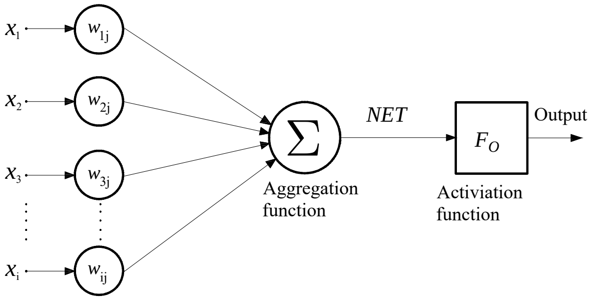

**Figure 3**: The representation of an artificial neuron.

Inputs are the information that is input to the process element from the outside world. A process element can receive inputs from other process elements as well as from the outside world. Weights represent the effect of the incoming information on that process element. Aggregation function calculates the net input received by the process element. There are many different types of aggregation functions (multiplication, maximum, majority, etc.). This study uses a weighted aggregation function. This function calculates the net input to a process element by taking the weights of inputs into consideration, as follows:

$$NET = \sum_{i=1}^{N} (I_p)_i w_i \tag{12}$$

In Eq. (12), $I_p$ represents inputs and $w$ represents weights. $N$ is the total number of inputs received by a process element. The NET value that results is then sent to the activation function. The activation function processes the net input received by the process element, and calculates the output that will be produced by the process element for this input. As is the case with the aggregation function, many different formulas can be used in an activation function to calculate the output. Because this study aims to create a multi-layered neural network model, the sigmoid function was used, which is a differentiable function and is expressed as follows:

$$F_O(\text{NET}) = \frac{1}{1+e^{-NET}} \tag{13}$$

The output of a process element is the value produced by the activation function. The output of a process element can be sent to the outside world, or can be used as an input to another process element.

## 3.2 Multi-Layered Neural Network Structure

Figure 4 describes the three-layered neural network model used in this study. The layers that structure the neural network are the input layer, the hidden layer, and the output layer. The input layer consists of two process elements, one of which represents the projectile mass $m_p$ (kg) and the other represents the exit velocity $v$ (m/s) of the projectile. The input data is transmitted to the hidden layer without undergoing any processing in the input layer.

The hidden layer contains six process elements. The task of the hidden layer is to process the information received from the input layer, and transmit it to the next layer after processing. As Figure 4 shows, this study uses a single hidden layer with six process elements. The output layer processes the information-received form the hidden layer, and calculates the output to be produced by the network in response to the input received at the input layer. In the output layer, this study uses the process element described in Figure 3. This process element represents the amount of muzzle displacement at the moment the projectile leaves the muzzle $w_z(x=L,t)$.

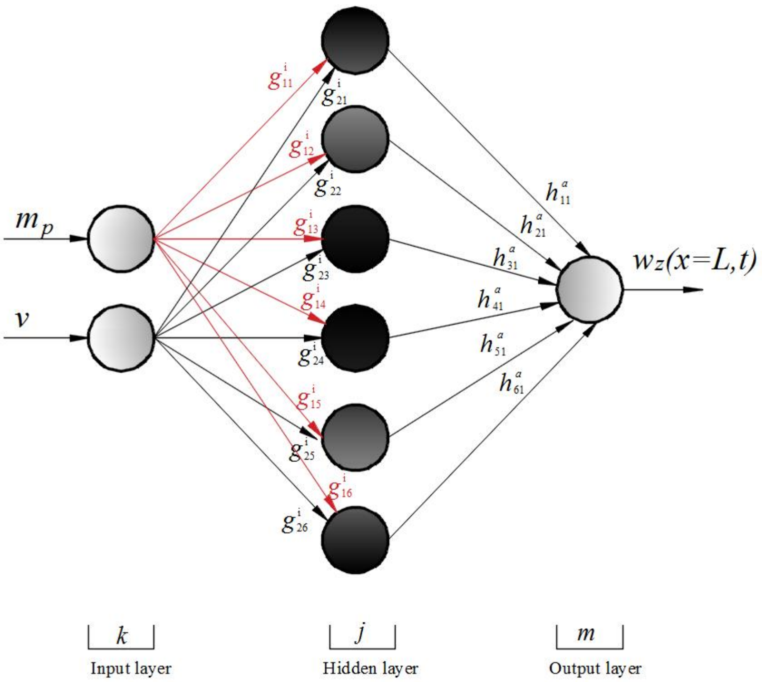

**Figure 4**: The ANN model used in this study for prediction tip displacement of the barrel.

### 3.3 Learning Rule of the Multi-Layered Neural Network

A supervised learning strategy was used in the neural network model for muzzle displacements. The generalized delta rule (GDR) used for network learning consists of two parts. The first part is forward propagation and the second part is backward propagation.

#### 3.3.1 Forward Propagation

At this stage, the first example in the training set is introduced to the network. Because there is no data processing in the input layer, incoming data is directly transmitted to the hidden layer. The input of the $k^{th}$ process element in the hidden layer is calculated as follows:

$$NET_j^a = \sum_{k=1}^{N} G_{kj} O_k^i \tag{15}$$

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} & \cdots & g_{1j} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{k1} & g_{k2} & g_{k3} & \cdots & k_{kj} \end{bmatrix}, (k=2, j=6) \tag{16}$$

In Eq. (15), $G_{kj}$ represents the weight of the link between the $k^{th}$ process element in the input layer and the $j^{th}$ process element in the hidden layer, as shown in Eq. (16). The output of the $j^{th}$ process element in the hidden layer, on the other hand, is calculated by processing the net input to this element through the sigmoid function. Accordingly, the output of the $j^{th}$ process element in the hidden layer is calculated as follows:

$$I_j^a = \frac{1}{1+e^{-NET_j^a}} \tag{17}$$

All process elements in the hidden layer are similarly related to the process elements in the output layer. The output of a process element in the output layer is also calculated by first calculating the net data received by this element, and processing that data through the sigmoid function, as follows:

$$I_m = \frac{1}{1+e^{-NET_m}}, \quad NET_m = \sum_{k=1}^{N} H_{jm} O_j \tag{18}$$

In Eq. (18), $H_{jm}$ represents weight of the link between the hidden layer and the output layer, and is expressed as in Eq. (19).

$$H_{jm} = \begin{bmatrix} h_{11} & h_{21} & h_{31} & \cdots & h_{j1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{k1} & h_{k2} & h_{k3} & \cdots & h_{kj} \end{bmatrix}, (k=1, j=6) \tag{19}$$

### 3.3.2 Backward Propagation

After the first example in the training set is introduced to the network and the output of the network is calculated, this output is compared to the expected output, and the difference between the two is called the error. Training an artificial neural network means reducing this error further with each new example in the training set. The error for the $m^{th}$ process element in the output layer is calculated as follows:

$$E_m = B_m - O_m, \; m=1. \tag{20}$$

In Eq. (20), $B_m$ represents the expected output value for the $m^{th}$ process element, and $O_m$ represents the output value produced by the neural network for this process element. This value is the error value for a process element. To reduce the error, weights of the links between the hidden layer and the output layer, and between the hidden layer and the input layer are changed. The error of the neural network is expressed using the mean squared error (MSE), and the absolute fraction of variance $R^2$ and the mean error percentage MEP are respectively given by:

$$MSE = \frac{1}{N_p} \sum_{i=1}^{N_p} (y_i - y_k)^2, \tag{21a}$$

$$R^2 = 1 - \left( \frac{\sum_{i=1}^{N_p} (y_i - y_k)^2}{\sum_{i=1}^{N_p} y_i^2} \right), \tag{21b}$$

$$MEP = \frac{\sum_{i=1}^{N_p} \left( \frac{y_k - y_i}{y_k} \right) x100}{N_p}, \tag{21c}$$

In Eq. (21), $N_p$ represents the total number of examples in the training set, $y_{\mathrm{i}}$ represents the value produced by the neural network for the $i^{th}$ example in the training set, and $y_k$ represents the actual value. The amount of change in iteration $t_N$ in the weight of the link between the $j^{th}$ process element in the hidden layer and the $m^{th}$ process element in the output layer, $\Delta H^a$, is expressed as follows:

$$\Delta H^a_{jm}(t_N) = \lambda \Lambda_m O^a_j + \Phi \Delta H^a_{jm}(t_N - 1) \tag{22}$$

In Eq. (22), $\lambda$ represents learning coefficient, and $\Phi$ represents momentum coefficient. In addition, $\Lambda_m$ represents the error of the $m^{th}$ output unit, and it is calculated as follows when the sigmoid function is used as the activation function:

$$\Lambda_m = O_m (1 - \mathrm{O}_m) \mathrm{E}_m \tag{23}$$

After $\Delta H^a(\mathrm{t})$, the amount of change at iteration $t_N$, is calculated, the new value of the weights at iteration $t_N$ is calculated as follows:

$$H^a_{jm}(\mathrm{t}_N) = \lambda O_m (1 - \mathrm{O}_m) \mathrm{E}_m O^a_j + \Phi \Delta H^a_{jm}(\mathrm{t}_N - 1) + H^a_{jm}(\mathrm{t}_N - 1) \tag{24}$$

Once the new weights of the links between the process elements in the hidden layer and in the output layer are calculated, the new weights of the links between the process elements in the hidden layer and in the input layer are calculated. The amount of change in the links between the hidden level and the input layer, $\Delta G^i$, is expressed as follows:

$$\Delta G^i_{kj}(\mathrm{t}_N) = \lambda \Lambda^a_j O^i_k + \Lambda \Delta G^i_{kj}(\mathrm{t}_N - 1) \tag{25}$$

In Eq. (25), the error term $\Lambda^a$ is calculated as follows, assuming the activation function is the sigmoid function:

$$\Lambda^a_j = O^a_j (1 - \mathrm{O}^a_j) \sum_m \Lambda_m H^a_{jm} \tag{26}$$

After the error value is calculated, the new values for the weights of the links between the process elements in the hidden layer and in the input layer are calculated as follows:

$$G^i_{kj}(\mathrm{t}_N) = \lambda O^a_j(1-\mathrm{O}^a_j)\left(\sum_m \Lambda_m H^a_{jm}\right)O^i_k + \Phi\Delta G^i_{kj}(\mathrm{t}_N - 1) + G^i_{kj}(\mathrm{t}_N - 1) \tag{27}$$

### 3.4 Training of the Multi-Layered Neural Network

Training an artificial neural network means reducing, for each example in the training set, the difference between the actual output value and the output value produced by the neural network, to below error tolerance. The training process is the process of adjusting weights in the neural network until the expected outputs are achieved for each example in the training set. Before using (testing) a neural network, the network needs to be trained well. Table 3 reports the calculated amounts of muzzle displacement for different projectile masses and muzzle velocities, which were used to train the neural network, calculated based on the theories explained in Section 2.

| Number data | $m_p$ (kg) | $v_m$ (m/s) | $w_z(L,t)$ (mm) FEM | Number data | $m_p$ (kg) | $v_m$ (m/s) | $w_z(L,t)$ (mm) FEM |
|---|---|---|---|---|---|---|---|
| 1 | 1.5 | 1000 | 8.1119 | 14 | 0.5 | 1200 | 2.3778 |
| 2 | 1.5 | 1200 | 7.3752 | 15 | 0.5 | 1300 | 1.9550 |
| 3 | 1.5 | 1300 | 6.0703 | 16 | 0.8 | 1450 | 1.9080 |
| 4 | 0.5 | 1600 | 5.9531 | 17 | 1.5 | 1600 | 1.6993 |
| 5 | 1.1 | 1000 | 5.8878 | 18 | 1.1 | 1600 | 1.2566 |
| 6 | 1.1 | 1200 | 5.3347 | 19 | 0.5 | 1450 | 1.1811 |
| 7 | 1.1 | 1300 | 4.3874 | 20 | 0.2 | 1000 | 1.0391 |
| 8 | 0.8 | 1000 | 4.2493 | 21 | 0.2 | 1200 | 0.9422 |
| 9 | 0.8 | 1200 | 3.8414 | 22 | 0.8 | 1600 | 0.9309 |
| 10 | 1.5 | 1450 | 3.6664 | 23 | 0.2 | 1300 | 0.7747 |
| 11 | 0.8 | 1300 | 3.1587 | 24 | 0.2 | 1450 | 0.4680 |
| 12 | 1.1 | 1450 | 2.6499 | 25 | 0.2 | 1600 | 0.2441 |
| 13 | 0.5 | 1000 | 2.6355 | | | | |

**Table 3**: The training set for 35 mm anti-aircraft cannon barrel.

In the training set, the projectile mass varies between 0.5 kg and 1.5 kg, whereas exit velocity of the projectile varies between 1000 and 1600 m/s. To deal with this imbalance between the parameters in the training set, all values were scaled between 0 and 1, within their own groups. For example, for the parameter of velocity, 1 represents 1600 m/s, which is the highest value, and 0 represents 1000 m/s, which is lowest value. The following formula was used to scale the input values:

$$x_r' = \frac{x_r - x_{\min}}{x_{\max} - x_{\min}} \tag{28}$$

In Eq. (28), $x_r$ represents the input value to be scaled, $x_{\min}$ represents the minimum value in the input set, $x_{\max}$ represents the maximum value in the input set, and $x_r'$ represents the scale $x_r$ input value. For example, for a projectile's exit velocity $x_r$=1300 m/s, the scaled $x$ value is calculated as $x_r^{'}$=0.5 using Eq. (28), because $x_{min}$=1000 m/s and $x_{max}$=1600 m/s.

Because inputs in the training set and expected outputs are presented to the neural network in scaled format, the output values produced by the neural network will also be scaled between 0 and 1. To translate these values back to their original format, Eq. (28) is expressed as follows:

$$x_r = x_r^{'}(x_{\max} - x_{\min}) + x_{\min} \tag{29}$$

### 3.4 Defining Stopping Criteria

In an artificial neural network, training needs to be stopped once the values of the weights are able to represent the problem space. The reason for this is that if the training continues after the weights become able to represent the problem space, further changes in the weights of the network may result in lower performance. There are two algorithms used to decide when to stop training. In the first, training is stopped when the error values calculated for all the examples in the training set are reduced to below a pre-defined level. In the second, training is stopped after a certain number of iterations, which requires a few trials to be made, to determine the number of iterations. This study uses the second algorithm. Although determining the appropriate number of iterations is a laborious process, the result was worth the effort.

## 4 NUMERICAL EXAMPLES

In this paper, the Newmark direct integration method (Wilson, 2002) is used along with the time step $\Delta t$ = 0.0001, β=0.25 and γ=0.5 values to obtain the solution of Eq. (11), where β and γ are parameters that manage the sensitiveness and stability of the Newmark procedure. When $\beta$ takes 0.25 value and γ 0.5, this numerical procedure is unconditionally stable.

**Example 1:** Let us take a simple supported isotropic beam-plate transversed by a $F$ = 4.4 N moving load. The dimensional and material specifications of the plate are identical with those chosen in (Reddy, 1984), i.e. $l_x$ = 10.36 cm; $l_y$= 0.635 cm, $h$ = 0.635 cm; $E$ = 206.8 GPa, $\rho$ = 10686.9 kg/m$^3$; $T_f$= 8.149 s, where $T_f$ is the fundamental period. In Table 4, dynamic amplification factors (DAF), which are defined as the ratio of the maximum dynamic deflection to the maximum static deflection, are compared with several previous numerical, analytical, and experimental results available in literature. It is noted that $T$ is the required time for moving load to travel the plate. It is seen that the results obtained by the new finite element (column 3) are very close to the analytical solution (Meirovitch, 1967), and the results of first order shear deformation theory (FSDT) method (Kadivar, 1998).

**Example 2:** For numerical verification, Table 3 reports the training set created for a 35 mm anti-aircraft barrel, based on the theory explained in Section 2. The training set includes expected values for muzzle displacement for different projectile masses and muzzle velocities. The training set contains 25 examples. The examples in the training set were presented to the neural network in order, starting from example one. The neural network was trained using a special *m.file* written in MATLAB, with a learning coefficient of λ=0.5 and momentum coefficient of $\Phi$=0.8, and then tested. Here, learning coefficient represents the amount of change in weights. Momentum coefficient, on the other hand, represents the proportion of the amount of change in the previous iteration that is added to the new amount of change.

| V(m/s) | $T_f / T$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 15.6 | 0.125 | 1.047 | 1.025 | 1.063 | 1.045 |
| 31.2 | 0.25 | 1.354 | 1.121 | 1.151 | 1.350 |
| 62.4 | 0.5 | 1.270 | 1.258 | 1.281 | 1.273 |
| 93.6 | 0.75 | 1.575 | 1.572 | 1.586 | 1.572 |
| 124.8 | 1 | 1.706 | 1.701 | 1.704 | 1.704 |
| 156 | 1.25 | 1.711 | 1.719 | 1.727 | 1.716 |
| 187.2 | 1.5 | 1.547 | - | - | - |
| 250 | 2 | 1.538 | 1.548 | 1.542 | 1.542 |

**Table 4**: Dynamic amplification factors (DAF) versus velocity. (1) Present method. (2) Analytical solution from Ref.(Meirovitch, 1967). (3) From Ref. (Kadivar and Mohebpour, 1998). (4) From Ref. (Esen, 2013).

As Figure 4 shows, the topological structure of the neural network created contains an input layer, hidden layer, and an output layer. There are two process elements in the input layer, representing, respectively, the inputs of projectile mass and departure velocity. The hidden layer, on the other hand, contains six process elements. The output layer contains a single process element. This process element represents the amount of muzzle dip $w_z(x=L,t)$ at time $t$ and projectile location $x=L$. Sigmoid function was used both in the hidden layer and in the output layer as the activation function. Figure 5 shows flowchart of the ANN and FEM combined algorithm for predict barrel tip displacement.

The training of the network was completed in 90,000 training rounds. Each round consisted of 25 iterations. Figure 5 displays the errors that resulted when the MSE expression given in Eq. (21a) was used. As the graph shows, the MSE dropped from 0.138 to less than 0.02 at the end of 10,000 training rounds. At the end of 90,000 training rounds, the MSE value was 0.000825, at which point the training was stopped. The effects of the various processing elements in the intermediate layer by GDR algorithm are presented in Figure 6. In this study, $9\text{x}10^4$ training cycles were used. However, the graph show $5\text{x}10^3$ iterations of the first portion to be understood more clearly. In addition, Figure 6a shows the change in the value of MSE for different process element, and in Figure 6b, the change in the value of $R^2$ for various process elements usage is presented.

Figure 7 displays the errors that resulted for some of the examples in the training set, by the number of training rounds. Figure 7a shows the error (E=B-O) for examples 4, 5, and 7 in the training set. Figure 7b shows the change in the errors for examples 9, 11, 13 and 14 by the number of training rounds; Figure 7c shows the change in the errors for examples 15, 17, 19 and 20; and Figure 7d shows the change in the errors for examples 21, 22, 23, and 25. As these graphs show, the error decreased as the number of training rounds increased.

Once the neural network was trained using 90,000 training rounds, the neural network was tested using examples that were not included in the training set. Eqs. (30) and (31) provide the weights of the links between the input layer and the hidden layer, and between the hidden layer and the output layer, respectively, after the training was completed.

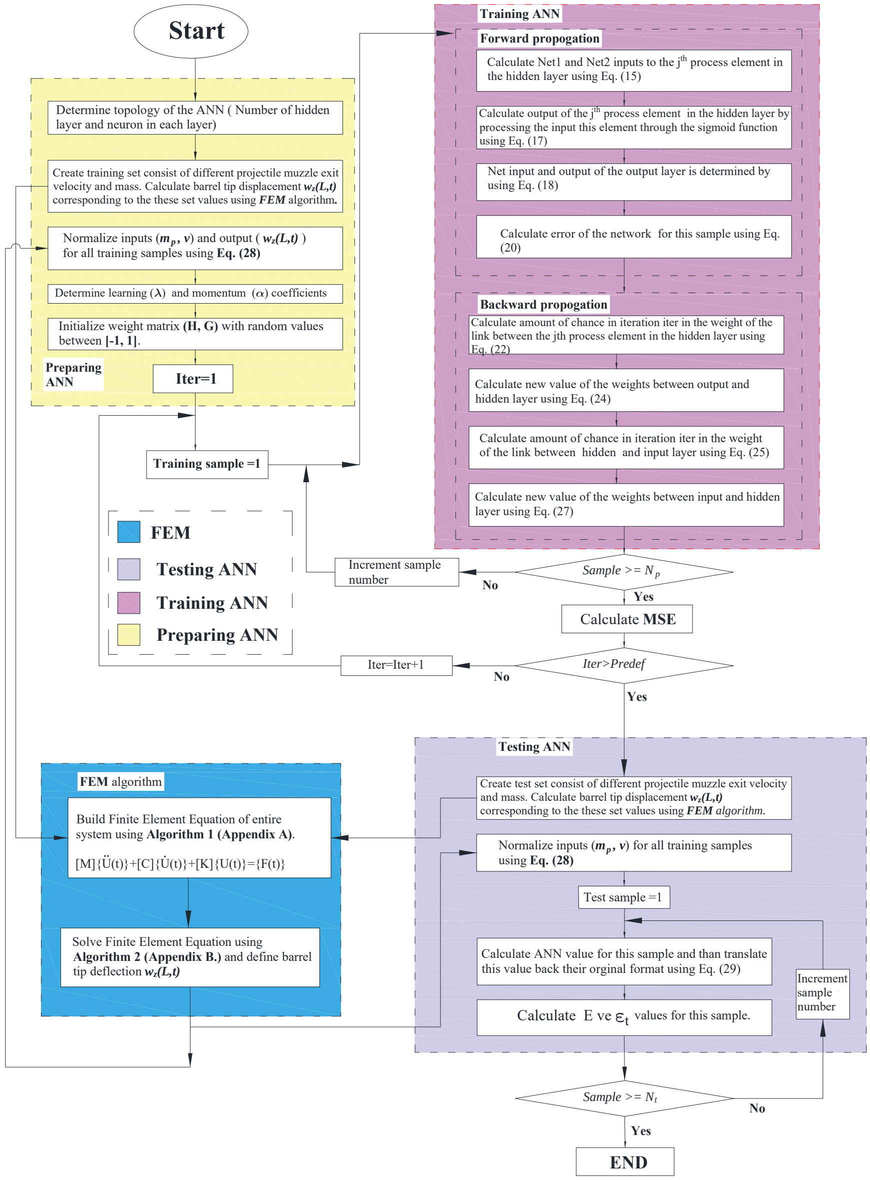
Start
Determine topology of the ANN ( Number of hidden layer and neuron in each layer)
Create training set consist of different projectile muzzle exit velocity and mass. Calculate barrel tip displacement wz(L,t) corresponding to the these set values using FEM algorithm.
Normalize inputs (mp, v) and output ( wz(L,t) ) for all training samples using Eq. (28)
Determine learning (λ) and momentum (α) coefficients
Initialize weight matrix (H, G) with random values between [-1, 1].
Preparing ANN
Iter=1
Training sample =1
Training ANN
Forward propogation
Calculate Net1 and Net2 inputs to the jth process element in the hidden layer using Eq. (15)
Calculate output of the jth process element in the hidden layer by processing the input this element through the sigmoid function using Eq. (17)
Net input and output of the output layer is determined by using Eq. (18)
Calculate error of the network for this sample using Eq. (20)
Backward propogation
Calculate amount of chance in iteration iter in the weight of the link between the jth process element in the hidden layer using Eq. (22)
Calculate new value of the weights between output and hidden layer using Eq. (24)
Calculate amount of chance in iteration iter in the weight of the link between hidden and input layer using Eq. (25)
Calculate new value of the weights between input and hidden layer using Eq. (27)
FEM
Testing ANN
Training ANN
Preparing ANN
Increment sample number
Sample >= Np
No
Yes
Calculate MSE
Iter=Iter+1
Iter>Predef
No
Yes
FEM algorithm
Build Finite Element Equation of entire system using Algorithm 1 (Appendix A).
[M]{Ü(t)}+[C]{U̇(t)}+[K]{U(t)}={F(t)}
Solve Finite Element Equation using Algorithm 2 (Appendix B.) and define barrel tip deflection wz(L,t)
Testing ANN
Create test set consist of different projectile muzzle exit velocity and mass. Calculate barrel tip displacement wz(L,t) corresponding to the these set values using FEM algorithm.
Normalize inputs (mp, v) for all training samples using Eq. (28)
Test sample =1
Calculate ANN value for this sample and than translate this value back their orginal format using Eq. (29)
Calculate E ve εt values for this sample.
Increment sample number
Sample >= Nt
No
Yes
END

**Figure 5**: The flowchart of the ANN and FEM combined algorithm for predict barrel tip displacement.

$$G^{i}_{kj} = \begin{bmatrix} 0.6708 & 0.6768 & -0.7513 & -0.6389 & -37.5684 & -52.3438 \\ -2.2072 & -2.2437 & -3.2869 & 1.8058 & 8.0246 & 12.6402 \end{bmatrix} \tag{30}$$

$$H_{jm}^{a} = \left[5.2717 \quad 5.3392 \quad -10.4707 \quad -4.0228 \quad -13.9874 \quad 12.8721\right] \tag{31}$$

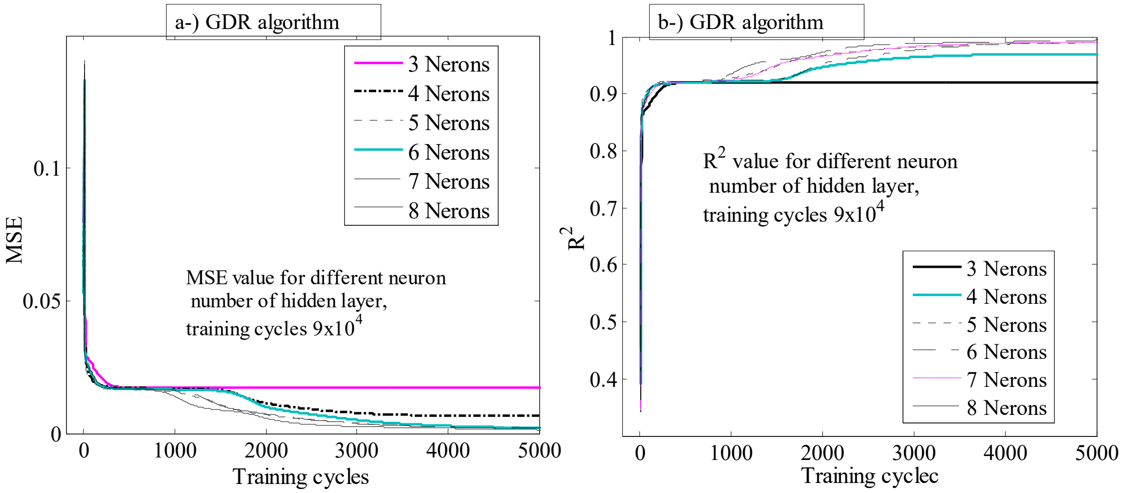

**Figure 6**: Performance of proposed ANN for different neuron number of hidden layer.

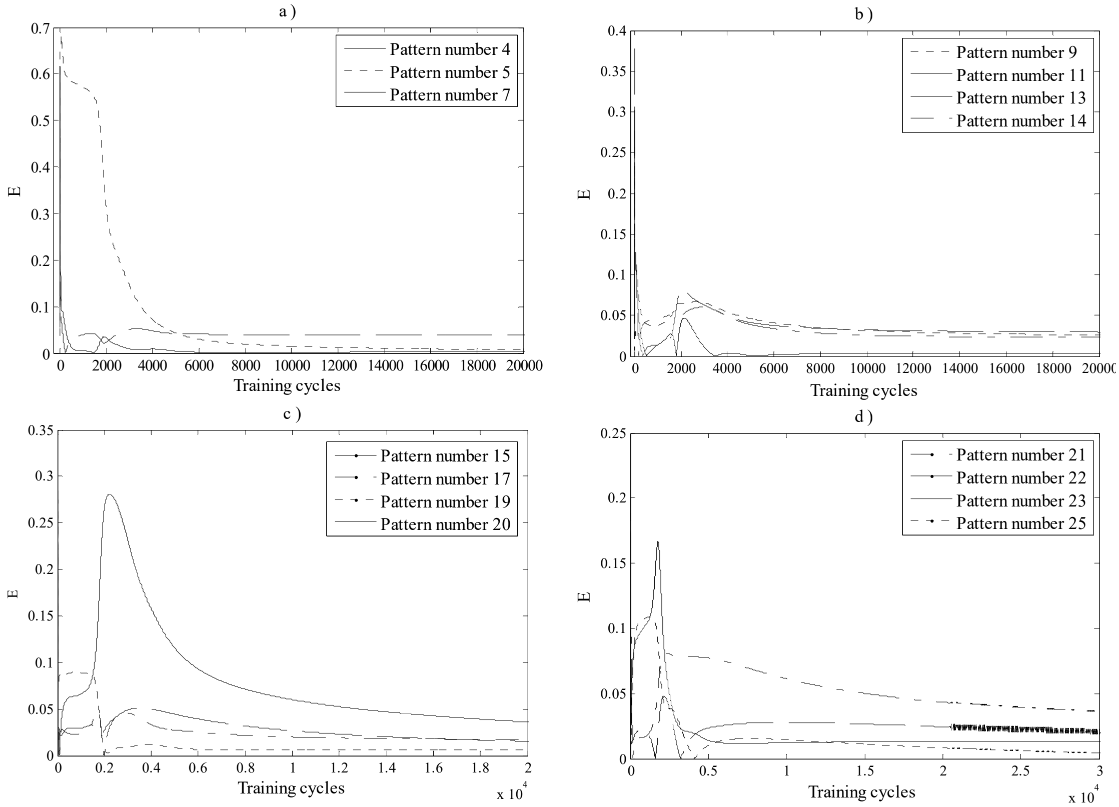

**Figure 7**: The error in training pattern during training process. a) Pattern number (4, 5, 7) b) Pattern number (9, 11, 13, 14) c) Pattern number (15, 17, 19, 20) d) Pattern number (21, 22, 23, 25).

The most experienced problem during the training of multi-layered network is the very long period of learning. Many parameters affect the training time such as learning coefficient (λ), momentum coefficient (Φ), the number of iterations, the initial value of the weight vector between the input layer and middle layer; and between middle layer and output layer. There is no precise information about the optimal number of cycles to complete the training. This varies according to the problem applied to the neural network. For some problems, the training of the network can take more than $10^7$ cycles, while for some others the training can be done at 100 cycles. In this study, 90,000 training cycle on a computer medium capacity (i7 processor, 32 GB RAM) has taken about 10 minutes. Table 5 shows the change in the mean error, for the training cycle from 50000 to 140000 with a 5000 interval increase; and errors for the examples and training set (4, 5, 9, 11, 15, 17, 21, and 22). The mean error MSE is decreased 4.29% for a $4x10^4$ increase in the education cycle from $5x10^4$ to $9x10^4$ that are $8.62x10^{-4}$ and $8.25x10^{-4}$. However, when $14x10^4$ training cycle have reached, the MSE value is $8.11x10^{-4}$, it decreased 1.6% compared to the situation in $9x10^4$ only. For a training cycle, $5x10^5$ and after this point, the value of MSE may decrease between 0.3-0.5 percent. However, a training cycle $5x10^5$ is not preferred due to the necessity of long time and memory capacity of the computer.

| Training cycle. | *MSE* | Training Pattern number | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 4 | 5 | 9 | 11 | 15 | 17 | 21 | 22 |
| 50000 | $8.62x10^{-4}$ | 0.0193 | 0.0432 | 0.0127 | 0.0168 | 0.0180 | 0.0136 | 0.0147 | 0.0127 |
| 55000 | $8.54\ x10^{-4}$ | 0.0184 | 0.0433 | 0.0126 | 0.0168 | 0.0172 | 0.0135 | 0.0140 | 0.0126 |
| 60000 | $8.48\ x10^{-4}$ | 0.0176 | 0.0435 | 0.0126 | 0.0168 | 0.0165 | 0.0134 | 0.0133 | 0.0126 |
| 65000 | $8.42\ x10^{-4}$ | 0.0169 | 0.0436 | 0.0126 | 0.0168 | 0.0159 | 0.0134 | 0.0126 | 0.0125 |
| 70000 | $8.38\ x10^{-4}$ | 0.0162 | 0.0437 | 0.0125 | 0.0168 | 0.0154 | 0.0133 | 0.0120 | 0.0125 |
| 75000 | $8.34\ x10^{-4}$ | 0.0156 | 0.0438 | 0.0125 | 0.0168 | 0.0149 | 0.0133 | 0.0115 | 0.0124 |
| 80000 | $8.31\ x10^{-4}$ | 0.0151 | 0.0439 | 0.0125 | 0.0168 | 0.0145 | 0.0132 | 0.0110 | 0.0124 |
| 85000 | $8.28\ x10^{-4}$ | 0.0146 | 0.0440 | 0.0125 | 0.0167 | 0.0141 | 0.0132 | 0.0106 | 0.0124 |
| 90000 | $8.25\ x10^{-4}$ | 0.0141 | 0.0440 | 0.0124 | 0.0167 | 0.0137 | 0.0132 | 0.0101 | 0.0123 |
| 95000 | $8.23\ x10^{-4}$ | 0.0137 | 0.0441 | 0.0124 | 0.0167 | 0.0134 | 0.0131 | 0.0098 | 0.0123 |
| 100000 | $8.21\ x10^{-4}$ | 0.0134 | 0.0441 | 0.0124 | 0.0167 | 0.0131 | 0.0131 | 0.0094 | 0.0123 |
| 110000 | $8.18\ x10^{-4}$ | 0.0127 | 0.0442 | 0.0124 | 0.0168 | 0.0125 | 0.0131 | 0.0087 | 0.0122 |
| 120000 | $8.15\ x10^{-4}$ | 0.0121 | 0.0443 | 0.0123 | 0.0168 | 0.0120 | 0.0130 | 0.0082 | 0.0121 |
| 130000 | $8.13\ x10^{-4}$ | 0.0115 | 0.0444 | 0.0123 | 0.0168 | 0.0116 | 0.0130 | 0.0077 | 0.0121 |
| 140000 | $8.11\ x10^{-4}$ | 0.0111 | 0.0445 | 0.0123 | 0.0168 | 0.0112 | 0.0129 | 0.0072 | 0.0120 |

**Table 5**: The effect of different training round upon MSE and training pattern error.

In Table 6, the effect of the different training cycle of network, on the performance is presented. Performance of the network is tested using test samples in the test set (1, 6, 8, 10, 22, 31) for the three different training cycle ($5x10^4$, $9x10^4$, $14x10^4$). The performances of the network for all samples in the test set are different in different training cycles. For example, when the training cycle is decreased from 9x10$^4$ to 5x10$^4$ for test samples of (1, 6, 10 and 22), the error rate has been decreased between 0.1-0.4%, while for the test sample 8 and 31 it has increased of between 0.05-0.2 percent. Moreover, increasing of the training cycle from 9x10$^4$ to 14x10$^4$ has reduced the error rate between 0.1-0.3% for 6, 8, and 31. However, for (1, 10, 22) it has increased by approximately 0.2-0.3%. The reason of this behaviour is the learning performance of each example in training set can be different for different training cycles. For some examples, learning can be completed at the beginning of the training process, but it will continue until the specified tolerance of MSE is satisfied. In this case, the learning performance at the beginning of the training process with a very low error rate may decrease by increasing the error rate gradually. What is important for the network is not only to learn an example well, but also is to learn generally for all samples at low error rates. The other analyses made for all the other pairs in the test sample set have showed that similar results are valid.

| Number data | $m_p$ (kg) | $v_p$ (m/s) | FEM $w_z(L,t)$ (mm) | Number of iteration | ANN (Predict) | E | $\varepsilon_t$ (%) |
|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 1100 | 2.6276 | 50000 | 2.6035 | 0.0241 | 0.9168 |
| | | | | 90000 | 2.5925 | 0.03508 | 1.3354 |
| | | | | 140000 | 2.5838 | 0.0438 | 1.6684 |
| 6 | 0.65 | 1350 | 2.2291 | 50000 | 2.1399 | 0.0892 | 3.9996 |
| | | | | 90000 | 2.1379 | 0.09116 | 4.0899 |
| | | | | 140000 | 2.1353 | 0.0838 | 3.7776 |
| 8 | 0.65 | 1600 | 0.7648 | 50000 | 0.9267 | -0.1619 | 21.1654 |
| | | | | 90000 | 0.9241 | -0.1593 | 20.837 |
| | | | | 140000 | 0.9210 | -0.1562 | 20.4287 |
| 10 | 0.8 | 1250 | 3.5282 | 50000 | 3.5252 | 0.003 | 0.0861 |
| | | | | 90000 | 3.5185 | 0.0096 | 0.2729 |
| | | | | 140000 | 3.5128 | 0.0154 | 0.4326 |
| 22 | 1.3 | 1350 | 4.5635 | 50000 | 4.5408 | 0.0227 | 0.4971 |
| | | | | 90000 | 4.5365 | 0.02691 | 0.5898 |
| | | | | 140000 | 4.5335 | 0.03 | 0.6579 |
| 31 | 1.5 | 1500 | 2.8458 | 50000 | 2.9460 | -0.1002 | 3.5214 |
| | | | | 90000 | 2.9447 | -0.0989 | 3.4779 |
| | | | | 140000 | 2.9444 | -0.0986 | 3.4637 |

**Table 6**: The performance of the ANN for different training round.

Table 7 reports the test results that were obtained after the training of the neural network was completed, using examples that were not included in the training set. The first column in Table 7 shows the number of the training set, and the second column shows the projectile mass tested, and the third column shows the exit velocity of the projectile, the fourth column shows the value of the amount of muzzle displacement calculated using FEM, the fifth column shows muzzle displacements predicted using ANN, the sixth column shows the difference between the actual value and the expected value, that is to say the error term, and the last column shows the relative error.

For each example in the test set, relative error is calculated as follows:

$$\varepsilon_t = \frac{|E|}{B}100 \tag{32}$$

The data reported in Table 7 show that relative error is usually below 5%. The only exception is observed in the test set example 8, where the expected value was 0.7648 for a projectile mass of $m_p$=0.65 kg and projectile exit velocity of 1600 m/s, but the ANN predicted a value of 0.9241. The relative error in this case was 20%. However, the relative errors for the rest of the test set examples show that overall the learning was very successful. Figure 8 shows the expected amount of muzzle displacement according to the theory and the amount predicted by the ANN.

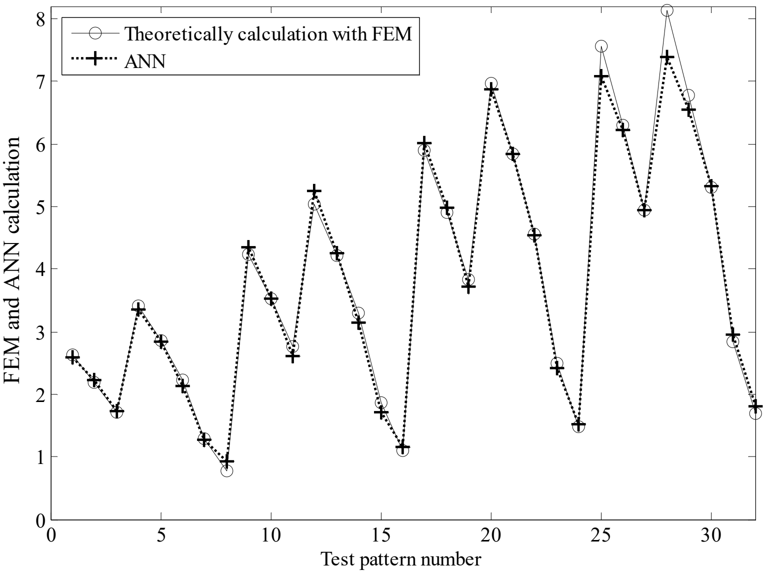

**Figure 8**: FEM and ANN calculation for 35 mm anti-aircraft barrel tip deflection.

In this study, during the firing of a gun barrel, the displacement at the end of a barrel was estimated by the artificial neural network. The obtained values were compared with the FEM model. The performance of the GDR algorithm used in design of ANN was compared with scaled conjugate gradient learning algorithm (SCH) used in literature. A comparison of the two algorithms GDR and SCH is given in Table 8, for the number of neurons in the hidden layer of the neural network from 3 to 8, and two different training cycle ($9x10^4$, $14x10^4$). After training is completed, as shown in table, the result of the testing of the samples contained in the test kit was obtained at the lowest average error, for 6 processing elements in middle layer and training cycle $9x10^4$.

| Number data | $m_p$ (kg) | $v_p$ (m/s) | FEM<br>$w_z(L,t)$ (mm) | ANN (Predict) | E | $\varepsilon_t$ (%) |
|---|---|---|---|---|---|---|
| 1 | 0.5 | 1100 | 2.6276 | 2.5925 | 0.03508 | 1.3354 |
| 2 | 0.5 | 1250 | 2.1836 | 2.2209 | -0.0373 | 1.7113 |
| 3 | 0.5 | 1350 | 1.7060 | 1.7381 | -0.0321 | 1.8844 |
| 4 | 0.65 | 1000 | 3.4191 | 3.3510 | 0.06805 | 1.9905 |
| 5 | 0.65 | 1250 | 2.8525 | 2.8293 | 0.02315 | 0.8117 |
| 6 | 0.65 | 1350 | 2.2291 | 2.1379 | 0.09116 | 4.0899 |
| 7 | 0.65 | 1500 | 1.2907 | 1.2713 | 0.01939 | 1.5025 |
| 8 | 0.65 | 1600 | 0.7648 | 0.9241 | -0.1593 | 20.837 |
| 9 | 0.8 | 1100 | 4.2427 | 4.3564 | -0.1137 | 2.6820 |
| 10 | 0.8 | 1250 | 3.5282 | 3.5185 | 0.0096 | 0.2729 |
| 11 | 0.8 | 1350 | 2.7577 | 2.6084 | 0.14926 | 5.4125 |
| 12 | 0.95 | 1000 | 5.0390 | 5.2533 | -0.2143 | 4.2535 |
| 13 | 0.95 | 1250 | 4.2108 | 4.2491 | -0.0383 | 0.9096 |
| 14 | 0.95 | 1350 | 3.2922 | 3.1429 | 0.14922 | 4.5327 |
| 15 | 0.95 | 1500 | 1.8614 | 1.7079 | 0.15348 | 8.2456 |
| 16 | 0.95 | 1600 | 1.0944 | 1.1496 | -0.0552 | 5.0462 |
| 17 | 1.1 | 1100 | 5.8881 | 6.0129 | -0.1248 | 2.1199 |
| 18 | 1.1 | 1250 | 4.9006 | 4.9711 | -0.0705 | 1.4390 |
| 19 | 1.1 | 1350 | 3.8328 | 3.7259 | 0.10687 | 2.7884 |
| 20 | 1.3 | 1000 | 6.9635 | 6.8682 | 0.09527 | 1.3682 |
| 21 | 1.3 | 1250 | 5.8320 | 5.8409 | -0.0089 | 0.1528 |
| 22 | 1.3 | 1350 | 4.5635 | 4.5365 | 0.02691 | 0.5898 |
| 23 | 1.3 | 1500 | 2.4884 | 2.4249 | 0.06345 | 2.5500 |
| 24 | 1.3 | 1600 | 1.4744 | 1.5237 | -0.0493 | 3.3444 |
| 25 | 1.4 | 1100 | 7.5654 | 7.0859 | 0.4794 | 6.3368 |
| 26 | 1.4 | 1250 | 6.3029 | 6.2149 | 0.0879 | 1.3951 |
| 27 | 1.4 | 1350 | 4.9334 | 4.9370 | -0.0036 | 0.0738 |
| 28 | 1.5 | 1100 | 8.1318 | 7.3832 | 0.7485 | 9.2047 |
| 29 | 1.5 | 1250 | 6.7774 | 6.5428 | 0.2345 | 3.4610 |
| 30 | 1.5 | 1350 | 5.3063 | 5.3230 | -0.0167 | 0.3150 |
| 31 | 1.5 | 1500 | 2.8458 | 2.9447 | -0.0989 | 3.4779 |
| 32 | 1.5 | 1600 | 1.6993 | 1.8081 | -0.1088 | 6.4078 |

**Table 7**: The testing set for 35 mm anti-aircraft cannon barrel and comparison of results.

The average error (MEP) in the algorithm SCR is obtained as 0.09235, while for GDR algorithm, (0.09512) it is also very close to previous value. It is observed that the average error is higher, where the number of processing elements in the intermediate layer is less than 6 or more. The cause of this is related to the topological structure of the network. It is not possible to create a single network topology that can represent all the engineering problems. The determination of network topology depends on the type of the problem, and the best network topology that will represent the problem should be determined by designers using some trial and error methods. Therefore, a large number of neuron does not mean that it will certainly represent the problem well. Likewise, possession of a small number of neuron does not mean that the representation of problem is weak.

| Algorithm | Number of neurons | Training cycle. | Training data | | | Test data | | |
|---|---|---|---|---|---|---|---|---|
| | | | MSE x $10^{-4}$ | $R^2$ | Average error (%) | MSE | $R^2$ | Average error (%) |
| GDR | 3 | $9x10^4$ | 170.819 | 0.92015 | 11.8485 | 0.13000 | 0.99301 | 1.03567 |
| GDR | 3 | $14x10^4$ | 170.853 | 0.92015 | 11.84856 | 0.13000 | 0.99301 | 1.03567 |
| GDR | 4 | $9x10^4$ | 61.1896 | 0.97139 | 0.00314 | 0.43018 | 0.97689 | 7.66544 |
| GDR | 4 | $14x10^4$ | 60.7863 | 0.97161 | 0.15147 | 0.42689 | 0.97707 | 7.51316 |
| GDR | 5 | $9x10^4$ | 8.04889 | 0.99623 | 1.60089 | 0.11297 | 0.99393 | 2.72106 |
| GDR | 5 | $14x10^4$ | 7.84242 | 0.99633 | 1.25377 | 0.13243 | 0.99288 | 3.15141 |
| GDR | 6 | $9x10^4$ | 8.25235 | 0.99613 | 0.52192 | 0.03767 | 0.99797 | 0.09512 |
| GDR | 6 | $14x10^4$ | 8.11695 | 0.99620 | 0.22156 | 0.03792 | 0.99796 | 0.19779 |
| GDR | 7 | $9x10^4$ | 8.09059 | 09.9621 | 0.08644 | 1.85827 | 0.90020 | 12.82969 |
| GDR | 7 | $14x10^4$ | 7.86834 | 0.99632 | 0.00328 | 1.61409 | 0.91331 | 13.21768 |
| GDR | 8 | $9x10^4$ | 4.50216 | 0.99789 | 0.13103 | 16.22256 | 0.12881 | 18.60012 |
| GDR | 8 | $14x10^4$ | 4.46487 | 0.99791 | 0.07493 | 16.43801 | 0.11724 | 18.87232 |
| SCG | 3 | $9x10^4$ | 125.311 | 0.93015 | 10.8385 | 0.115280 | 0.99589 | 1.25963 |
| SCG | 3 | $14x10^4$ | 109.103 | 0.92015 | 10.81258 | 0.125698 | 0.99457 | 1.35962 |
| SCG | 4 | $9x10^4$ | 72.2589 | 0.97139 | 0.00418 | 0.58692 | 0.98529 | 6.95862 |
| SCG | 4 | $14x10^4$ | 58.1936 | 0.99161 | 0.05637 | 0.40259 | 0.98301 | 7.02569 |
| SCG | 5 | $9x10^4$ | 8.12569 | 0.99221 | 1.70283 | 0.20569 | 0.98697 | 202596 |
| SCG | 5 | $14x10^4$ | 7.69253 | 0.99263 | 1.03698 | 0.1502 | 0.98995 | 3.32569 |
| SCG | 6 | $9x10^4$ | 6.25987 | 0.99301 | 0.69583 | 0.06952 | 0.99105 | 0.09235 |
| SCG | 6 | $14x10^4$ | 7.16391 | 0.99345 | 0.23594 | 0.04201 | 0.99304 | 0.12569 |
| SCG | 7 | $9x10^4$ | 7.09059 | 09.9477 | 0.06965 | 1.52589 | 0.92038 | 10.2596 |
| SCG | 7 | $14x10^4$ | 6.98696 | 0.99405 | 0.002589 | 1.32569 | 0.91658 | 11.1258 |
| SCG | 8 | $9x10^4$ | 6.97652 | 0.99258 | 0.09687 | 15.2584 | 0.10881 | 13.60209 |
| SCG | 8 | $14x10^4$ | 6.91256 | 0.99013 | 0.05896 | 15.2648 | 0.12724 | 13.23546 |

**Table 8**. Statistical data for the barrel tip displacement using two different algorithms.

## 5 CONCLUSION

The use of different types of ammunition is a military necessity, because each type of ammunition has its specific intended use for operational purposes, and each type of ammunition has its specific weight and chemical content. This means that projectiles in different types of ammunition have different masses and muzzle exit velocities. Thus, muzzle behaviour during shooting varies by the type of ammunition used. The most important parameters affecting a weapon system's dynamics are the projectile-mass, acceleration, and exit velocity.

The purpose of this study is to develop an artificial neural network to predict the amount of muzzle displacement, which is due to the force created by a projectile accelerating inside the barrel, and which reduces the shooting accuracy of a weapon system. Using the proposed method, one can determine the amount of muzzle displacement prior to shooting. In this method, the projectile mass and exit velocity are used as the input parameters of the neural network, while the amount of muzzle displacement $w_z(x=L,t)$ is the output of the model. A training set is created to characterize the problem consisting of 25 examples from the problem space. At the end of the training process, which consisted of 90,000 training rounds, both the MSE and the individual errors E for the examples in the training set were reduced to a very low level. The test set prepared to test the artificial neural network consisted of 32 examples covering the training space. Relative errors for some of the examples (Test pattern numbers 21, 27, and 30) were between 0.1- 0.2%, corresponding to about 0.009 mm, which is negligible for engineering purposes. In some of the test set examples (15 and 24), on the other hand, the relative error was about 8-9%, corresponding to a miscalculation of 0.15-0.7 mm. Only in one test set example (Test pattern number 8), the error was about 20%.

The method developed in this study makes it possible to examine the effect of different types of ammunition on the barrel using computers and eliminates the need for time consuming and costly tests. In addition, by integrating an artificial neural network trained according to barrel characteristics to the software, which is controlling barrel position, the shooting accuracy and strike power of the weapon system can be increased by simply adjusting the initial position of the barrel. This would make it possible to design weapons that are lighter and more effective against targets. The velocity of a projectile inside the barrel varies by time and forces the barrel to change its natural frequencies continuously. This means that for different projectiles and muzzle velocities, different vibration modes are created in the barrel. For example, the muzzle displacement value is positive at some muzzle velocities, and negative at others. In addition, predicting the amount of muzzle displacement in a weapon barrel may not be sufficient sometimes, predicting the angle of inclination of the barrel may also be required. The neural network modelled in this study does not require many complex systems to make prediction, but an artificial neural network with at least two hidden layers is required only, and a preparation of a larger training set that represents the problem space are needed to predict both positive and negative muzzle displacements. Using the proposed method may help engineers in improving the target accuracy of a weapon system.

## References

Alexander, E.J., 2007. AGS Gun and Projectile Dynamic Modeling Correlation to Test Data Branch Manager , Applied Mechanics. US Army Armament Syst. Div. 480–496.

Altınkaya, H., Orak, İ.M., Esen, İ., 2014. Artificial neural network application for modeling the rail rolling process. Expert Syst. Appl. 41, 7135–7146. doi:10.1016/j.eswa.2014.06.014

Azimi, H., Galal, K., Pekau, O.A., 2013. A numerical element for vehicle-bridge interaction analysis of vehicles experiencing sudden deceleration. Eng. Struct. 49, 792–805. doi:10.1016/j.engstruct.2012.12.031

Balla, J., 2011. Dynamics of mounted automatic cannon on track vehicle. Int. J. Math. Model. Methods Appl. Sci. 5.

Cay, Y., 2013. Prediction of a gasoline engine performance with artificial neural network. Fuel 111, 324–331. doi:10.1016/j.fuel.2012.12.040

Cifuentes, A.O., 1989. Dynamic response of a beam excited by a moving mass. Finite Elem. Anal. Des. 5, 237–246. doi:10.1016/0168-874X(89)90046-2

Clough R.W; Penzien J., 2003. Dynamics of Structures, Dynamics of Structures. doi:10.1002/9781118599792

Czarnigowski, J., 2010. A neural network model-based observer for idle speed control of ignition in SI engine. Eng. Appl. Artif. Intell. 23, 1–7. doi:10.1016/j.engappai.2009.09.008

Dehestani, M., Mofid, M., Vafai, A., 2009. Investigation of critical influential speed for moving mass problems on beams. Appl. Math. Model. 33, 3885–3895. doi:10.1016/j.apm.2009.01.003

Düğenci, M., Aydemir, A., Esen, İ., Aydın, M.E., 2015. Creep modelling of polypropylenes using artificial neural networks trained with Bee algorithms. Eng. Appl. Artif. Intell. 45, 71–79. doi:10.1016/j.engappai.2015.06.016

Esen, I., 2011. Dynamic response of a beam due to an accelerating moving mass using moving finite element approximation. Math. Comput. Appl. 16, 171–182.

Esen, İ., 2015. A new FEM procedure for transverse and longitudinal vibration analysis of thin rectangular plates subjected to a variable velocity moving load along an arbitrary trajectory. Lat. Am. J. Solids Struct. 12, 808–830.

Esen, İ., 2013. A new finite element for transverse vibration of rectangular thin plates under a moving mass. Finite Elem. Anal. Des. 66, 26–35. doi:10.1016/j.finel.2012.11.005

Esen, İ., Koç, M.A., 2015a. Optimization of a passive vibration absorber for a barrel using the genetic algorithm. Expert Syst. Appl. 42, 894–905. doi:10.1016/j.eswa.2014.08.038

Esen, İ., Koç, M.A., 2015b. Dynamic response of a 120 mm smoothbore tank barrel during horizontal and inclined firing positions. Lat. Am. J. Solids Struct. 12, 1462–1486.

Esmailzadeh, E., Jalili, N., 2003. Vehicle–passenger–structure interaction of uniform bridges traversed by moving vehicles. J. Sound Vib. 260, 611–635. doi:10.1016/S0022-460X(02)00960-4

Hosseini, M., Dalvand, A., 2014. Neural Network Approach for Estimation of Penetration Depth in Concrete Targets by Ogive-nose Steel Projectiles. Lat. Am. J. Solids Struct. 12, 492–506.

Kadivar, M.H., Mohebpour, S.R., 1998. Finite element dynamic analysis of unsymmetric composite laminated beams with shear effect and rotary inertia under the action of moving loads. Finite Elem. Anal. Des. 29, 259–273. doi:10.1016/S0168-874X(98)00024-9

Kahya, V., 2012. Dynamic analysis of laminated composite beams under moving loads using finite element method. Nucl. Eng. Des. 243, 41–48. doi:10.1016/j.nucengdes.2011.12.015

Kathe, E., 1997. Design and validation of a gun barrel vibration absorber. New York.

Koide, R.M., Ferreira, A.P.C.S., Luersen, M.A., 2014. Laminated Composites Buckling Analysis Using Lamination Parameters, Neural Networks and Support Vector Regression. Lat. Am. J. Solids Struct. 12, 271–294.

L. Fryba, 1999. Vibration solids and structures under moving loads. Thomas Telford House.

Lagaros, N.D., Papadrakakis, M., 2012. Neural network based prediction schemes of the non-linear seismic response of 3D buildings. Adv. Eng. Softw. 44, 92–115. doi:10.1016/j.advengsoft.2011.05.033

Lee, H.P., 1996. Transverse vibration of a Timoshenko beam acted on by an accelerating mass. Appl. Acoust. 47, 319–330. doi:10.1016/0003-682X(95)00067-J

Littlefield, A., Kathe, E., Messier, R., Olsen, K., 2002. Gun barrel vibration absorber to increase accuracy. New York.

Liu, B., Wang, Y., Hu, P., Yuan, Q., 2015. Impact coefficient and reliability of mid-span continuous beam bridge under action of extra heavy vehicle with low speed. J. Cent. South Univ. 22, 1510–1520. doi:10.1007/s11771-015-2668-6

Lou, P., 2005. A vehicle-track-bridge interaction element considering vehicle's pitching effect. Finite Elem. Anal. Des. 41, 397–427. doi:10.1016/j.finel.2004.07.004

Lou, P., Au, F.T.K., 2013. Finite element formulae for internal forces of Bernoulli-Euler beams under moving vehicles. J. Sound Vib. 332, 1533–1552. doi:10.1016/j.jsv.2012.11.011

M. Štiavnický, P. Lisý, 2013. influence of Barrel Vibration on the Barrel Muzzle Position at the Moment when Bullet Exits Barrel. Adv. Mil. Technol. 8, 89–102.

Martínez-Martínez, V., Gomez-Gil, F.J., Gomez-Gil, J., Ruiz-Gonzalez, R., 2015. An Artificial Neural Network based expert system fitted with Genetic Algorithms for detecting the status of several rotary components in agro-industrial machines using a single vibration signal. Expert Syst. Appl. 42, 6433–6441. doi:10.1016/j.eswa.2015.04.018

Meirovitch, L., 1967. Analytical methods in vibrations. The Macmillan Company, New York.

Michaltsos, G., 2002. Dynamic behaviour of a single-span beam subjected to loads moving with variable speeds. J. Sound Vib. 258, 359–372. doi:10.1006/jsvi.5141

Michaltsos, G., Sophianopoulos, D., Kounadis, A.N., 1996. The Effect of a Moving Mass and Other Parameters on the Dynamic Response of a Simply Supported Beam. J. Sound Vib. 191, 357–362. doi:10.1006/jsvi.1996.0127

Niaz, M., Nikkhoo, A., 2015. Inspection of a rectangular plate dynamics under a moving mass with varying velocity utilizing BCOPs. Lat. Am. J. Solids Struct. 12, 317–332.

Omolofe, B., Oni, S.T., 2015. Transverse Motions of Rectangular Plates Resting on Elastic Foundation and Under Concentrated Masses Moving at Varying Velocities. Lat. Am. J. Solids Struct. 12, 1296–1318.

Perez-Ramirez, C.A., Amezquita-Sanchez, J.P., Adeli, H., Valtierra-Rodriguez, M., Camarena-Martinez, D., Romero-Troncoso, R.J., 2016. New methodology for modal parameters identification of smart civil structures using ambient vibrations and synchrosqueezed wavelet transform. Eng. Appl. Artif. Intell. 48, 1–12. doi:10.1016/j.engappai.2015.10.005

Rogers, S.K., Colombi, J.M., Martin, C.E., Gainey, J.C., Fielding, K.H., Burns, T.J., Ruck, D.W., Kabrisky, M., Oxley, M., 1995. Neural networks for automatic target recognition. Neural Networks 8, 1153–1184. doi:10.1016/0893-6080(95)00050-X

Tawfik, M., 2008. Dynamics and Stability of Stepped Gun-Barrels with Moving Bullets. Adv. Acoust. Vib. 2008, 1–6. doi:10.1155/2008/483857

Wang, Y., 2009. The transient dynamics of a moving accelerating / decelerating mass traveling on a periodic-array non-homogeneous composite beam. Eur. J. Mech. A/Solids 28, 827–840. doi:10.1016/j.euromechsol.2009.03.005

Wilson, E.L., 2002. Static and Dynamic Analysis of Structures. Computers and Structures Inc., Berkeley.

Wyss, J.C., Su, D., Fujino, Y., 2011. Prediction of vehicle-induced local responses and application to a skewed girder bridge. Eng. Struct. 33, 1088–1097. doi:10.1016/j.engstruct.2010.12.020

Yang, Y.B., Cheng, M.C., Chang, K.C., 2013. Frequency Variation in Vehicle–Bridge Interaction Systems. Int. J. Struct. Stab. Dyn. 13, 1350019. doi:10.1142/S0219455413500193

## APPENDIX A

### Algorithm 1: Deriving Equation of motion

For the calculation of the instantaneous overall mass and stiffness matrices of the entire system at every time step of $\Delta t$, one may use the following steps:

1. Determine the mass and stiffness matrices of each barrel beam element.
2. For time $t$, determine the element $s$ on which the moving projectile locates with (10).

3. Determine $x_m(t)$ which is the time dependent position of the moving projectile on the $s^{\text{th}}$ element with (10).
4. Calculate the time dependent interpolation functions with (5) by substituting the value $x_m(t)$ which is defined in the previous step.
5. Calculate mass, stiffness and damping matrices of the time-dependent finite element using of Eqs. (6-8).
6. Calculate the instantaneous overall mass and stiffness matrices of the entire system by combining the mass and stiffness matrices of each beam element, and then impose boundary conditions. If necessary, the Eigen solution of these matrices gives instantaneous natural frequency of the entire system at time $t$.
7. For $t+\Delta t$ go to step 2

## APPENDIX B

### Algorithm 2: Solution of Equation of motion

Using Newmark's integration method (Wilson, 2002), the solution of Eq. (9) can be obtained according to the following steps:

1. Determine the integration parameters β and γ and magnitude of the time interval *Δt*. Calculate integration constants:

$$\begin{aligned} a_0 &= \frac{1}{\beta\Delta t^2}, \quad a_1 = \frac{\gamma}{\beta\Delta t}, \quad a_2 = \frac{1}{\beta\Delta t}, \quad a_3 = \frac{1}{2\beta} - 1, \\ a_4 &= \frac{\gamma}{\beta} - 1, \quad a_5 = \frac{\Delta t}{2}(\frac{\gamma}{\beta} - 2), \quad a_6 = \Delta t(1-\gamma), \quad a_7 = \gamma\Delta t \end{aligned} \tag{B.1}$$

2. Assembling all element matrices including time dependent finite element, define the mass, stiffness and damping $[M]$, $[K]$ and $[C]$ matrices at $t_n = (t_{n-1} + \Delta t)$ time.
3. Calculate effective stiffness matrix at ($t_n = t_{n-1} + \Delta t$) time:

$$[\widehat{K}] = [K] + a_0[M] + a_1[C] \tag{B.2}$$

$$\begin{aligned} \{\widehat{F}(t_n)\} = \{F(t_n)\} + [M](a_0\{U(t_{n-1})\} + a_2\{\dot{U}(t_{n-1})\} + a_3\{\ddot{U}(t_{n-1})\}) \\ +[C](a_1\{U(t_{n-1})\} + a_4\{\dot{U}(t_{n-1})\} + a_5\{\ddot{U}(t_{n-1})\}) \end{aligned} \tag{B.3}$$

Where $\{\ddot{U}(t_{n-1})\}$ , $\{\dot{U}(t_{n-1})\}$ , $\{U(t_{n-1})\}$ are, respectively, the initial conditions for the accelerations, velocities and displacements of the structural system at time $t = t_0 = 0$.

4. Calculate deflections at $t_n$ time:

$$\{U(t_n)\} = [\widehat{K}]^{-1}\{\widehat{F}(t_n)\} \tag{B.4}$$

5. Calculate accelerations and velocities at $t_n$ time:

$$\{\ddot{U}(t_n)\} = a_0(\{U(t_n)\} - \{U(t_{n-1})\}) - a_2\{\dot{U}(t_{n-1})\} - a_3\{\ddot{U}(t_{n-1})\} \tag{B.5}$$

$$\{\dot{U}(t_n)\} = \{\dot{U}(t_{n-1})\} + a_6\{\ddot{U}(t_{n-1})\} + a_7\{\ddot{U}(t_n)\} \tag{B.6}$$

Steps 3-7, $t$=$t_n$ =$t_{n-1}$+ $\Delta t$ ($n$=1, 2, 3, and $t_0$=0) are repeated for all time steps, for deflections $\{U(t_{n-1})\}$, velocities $\{\dot{U}(t_n)\}$ and accelerations $\{\ddot{U}(t_n)\}$ of the entire system.