*IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*
*e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 11, Issue 4 Ver. III (Jul. – Aug. 2016), PP 66-81*
*www.iosrjournals.org*

# ANN Robot Energy Modeling

Fernando Rios-Gutierrez, Adel El-Shahat, Mudasser Wahab
*(Department of Electrical Engineering Georgia Southern University Statesboro, Georgia 30458)*

***Abstract:*** *This paper proposes energy modeling for robot based on real measurements data. First, the paper proposes six preliminary ANN Models on both carpet and hard floor. These models' Inputs: Theoretical Time, Theoretical Velocity and Output: The Current; then with Inputs: Theoretical Time, Theoretical Velocity and Output: The Current, The Voltage; and finally with Inputs: Time, Real Linear Velocity, Rotational Velocity and Output: The Current, The Voltage. Second, a global ANN model with time and speed as inputs and current, voltage, linear speed, rotational speed on carpet, along with current, voltage, linear speed, rotational speed on hard floor. This general model is presented in the form of Simulink model after care selection of number in neurons in hidden layer. This model has the capability to predict and simulate the robot energy characteristics under different conditions. This real data measurements on both hard floor and carpet are presented to be used as training data for Neural Network. All the ANN models are checked in the form of minimum error, accuracy, good regression constants and comparisons between real and predicted data. ANN with feed forward back-propagation technique is used to implement the models. It is adopted to make benefits from its ability of interpolation. ANN models with Back - Propagation (BP) technique is created with suitable numbers of layers and neurons. The last model will be used with the aid of Genetic Algorithm to improve and optimize the energy efficiency of robots.*

***Keywords:*** *Energy, Modeling, Simulink, Artificial Neural Network, Estimation.*

## I. Introduction

It is a very important issue to model Mobile robots in order to make complete performance analysis and optimize this performance. The purpose of proper modeling and optimization is to solve the problem of limited amount of robots energy. Many important contributions from other researchers related to robots modeling are reviewed with advantages and drawbacks before starting this work [1], [2], [3], [4], [5]. Robots design, control, and implementation work are done using ANN like [6], [7]. Due to the enormous development in the field of robotics, robots and especially autonomous mobile robots have found their use in a lot of applications. These applications include search and rescue, security, rehabilitation, cleaning and delivery. Autonomous mobile robots most of the time rely on batteries as their energy source and batteries have very limited energy capacity. This finite amount of energy can make the robot work only for a limited time and this is why the use of these robots in complicated missions is not feasible. Although these robots could be refueled while they are operating and their time of operation can be increased; the cost of replacement of their energy source is too high to be realistic. So, our global ANN model will become a very important one especially for optimization analysis. Especially, this model accomplishes all important performance characteristics and based on real experimental measurements.

## II. Experimental Data

The experimental data are drawn from the robot and associated measurements devices which used in [1] for the same authors as shown in the following samples figures. The used robot was built at the first author's laboratory by his previous students as shown in Fig. 1 to measure required data.

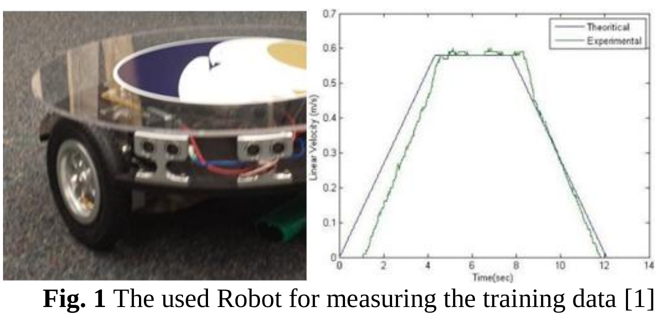


**Fig. 1** The used Robot for measuring the training data [1]

The current is measured by an ACS712 current sensor as shown in Fig. 2. Current data is fed into an Arduino Mega 2560 during the robot's motion. The sampling time was set at 26 *ms* because our robot's maximum speed is 0.58 *m/s*

**Fig. 2.** ACS712 current sensor module [1]

This data introduced in the following figures and more is used as training data for ANN models.

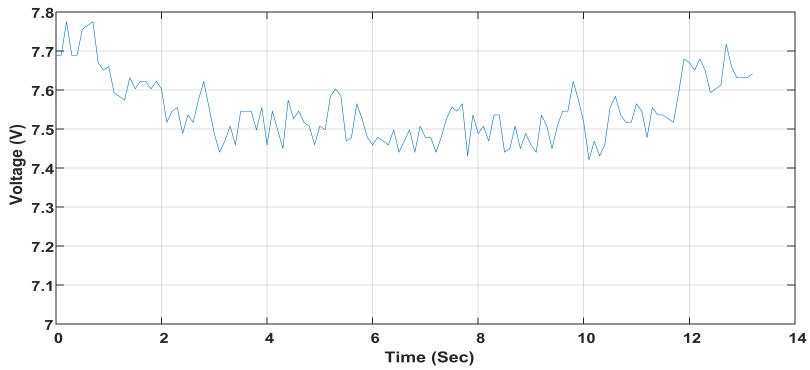


**Fig. 3** Voltage and Time Relation on Carpet.

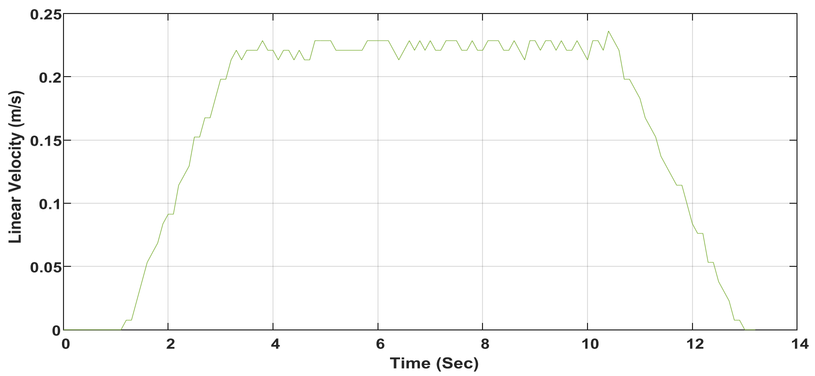


**Fig. 4** Linear Velocity and Time Relation on Carpet.

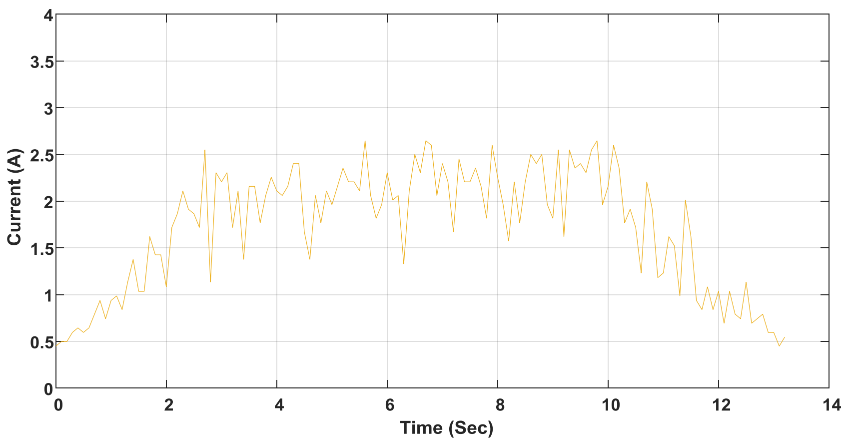


**Fig. 5** Current and Time Relation on Carpet.

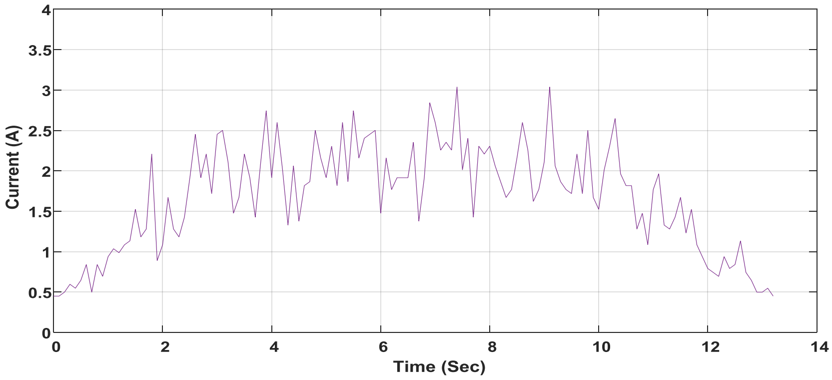


**Fig. 6** Current and Time Relation on Hard-Floor.

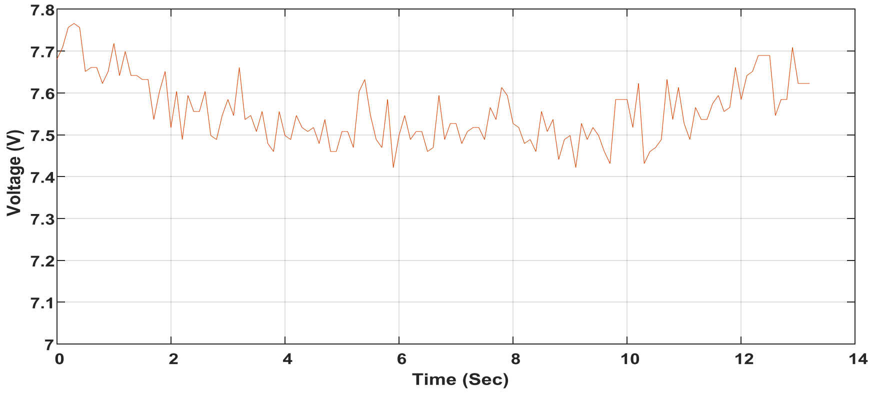


**Fig. 7** Voltage and Time Relation on Hard-Floor.

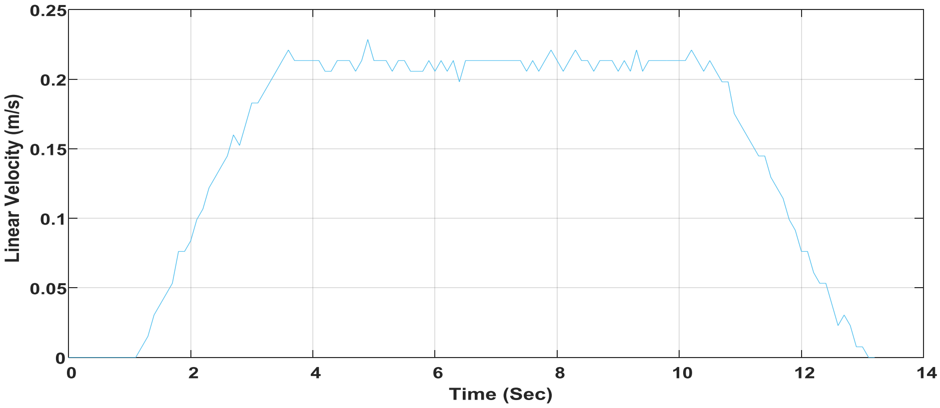


**Fig. 8** Linear Velocity and Time Relation on Hard-Floor.

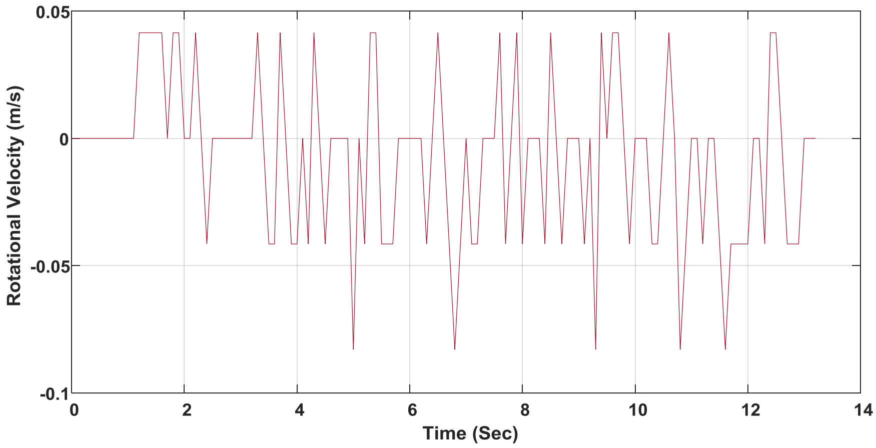


**Fig. 9** Rotational Velocity and Time Relation on Carpet.

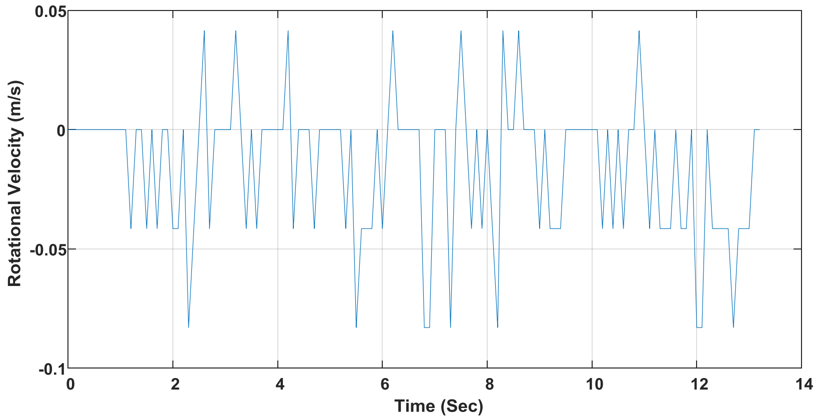


**Fig. 10** Rotational Velocity and Time Relation on Hard-Floor.

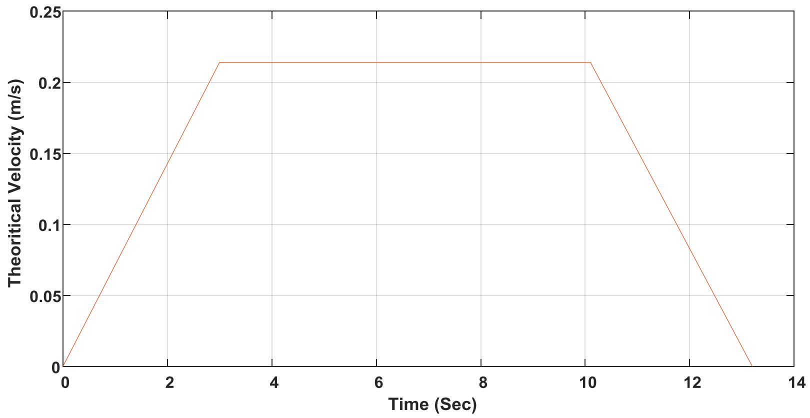


**Fig. 11** Theoretical Velocity and Time Relation.

## III. Ann Models

Artificial Neural Network with feed forward back-propagation technique is used to implement the various ANN models [8]-[11]. All the models consist of one hidden contains log-sigmoid function and other is output layer contains pure-line function. Carpet Model 1: Inputs: Time, Velocity, Output: Current; with 7 neurons and 1 neuron. (R= 0.99105). 2nd Carpet Model: The same i/ps as 1st model, O/ps: Current, Voltage; 8 neurons and 2 neurons. (R= 0.9766). 3rd Carpet Model: I/ps: Time, Linear Velocity, Rotational Velocity; o/ps: Current, Voltage; 7 neurons and 2 neurons. (R = 0.99511). Hard floor: Model 1: I/ps: Time, Velocity; O/p: Current; 6 neurons, 1 neuron. (R= 0.98189). Model 2: same i/ps; O/p: Current, Voltage; 11 neurons, 2 neurons. (R= 0.98188). Model 3: I/ps: Time, Linear Velocity, Rotational Velocity, O/ps: Current, Voltage, 7 neurons, 2 neurons. (R= 0.97472).

### *A. Carpet Models*

***ANN Model 1 for the carpet:***

Inputs: Theoretical Time, and Theoretical Velocity

Output: The Current

Neural Network consists of two layers one hidden contains log-sigmoid function with seven neurons and the other is the output layer contains pure-line function with one neuron.

The normalized inputs eq.n are:

$$\text{Time}_n = (\text{Time} - 1.5000) / (0.9092)$$
$$\text{Velocity}_n = (\text{Velocity} - 0.1070) / (0.0649) \qquad (1)$$

Equation (1) presents the normalized input for the power and the following equations lead to the required derived equation.

n: Subscript denotes normalized parameters

Ei: Sum of input with input weight and input bias for each node in hidden layer in neural network

Fi: Output from each node in hidden layer to output layer according to transfer function here is logsig

$E1= -0.0206 \ \text{Time}_n + 6.2084 \ \text{Velocity}_n + 0.8792$

$F1=1 / (1 + \exp(-E1))$
$E2= -41.9445\ Time_n - 36.2368\ Velocity_n + 107.7541$
$F2=1 / (1 + \exp(-E2))$
$E3= 1.8870\ Time_n + 4.3406\ Velocity_n + 1.3116$
$F3=1 / (1 + \exp(-E3))$
$E4= 12.2653\ Time_n + 7.9388\ Velocity_n - 2.9239$ (2)
$F4=1 / (1 + \exp(-E4))$
$E5= -51.7383\ Time_n - 58.0546\ Velocity_n + 71.4314$
$F5=1 / (1 + \exp(-E5))$
$E6= -63.1920\ Time_n - 68.1760\ Velocity_n + 180.9475$
$F6=1 / (1 + \exp(-E6))$
$E7= -3.7685\ Time_n - 2.7997\ Velocity_n - 1.7873$
$F7=1 / (1 + \exp(-E7))$

The normalized current relation from ANN:

$I_n = -171.7834\ F1 - 114.3126\ F2 + 318.7558\ F3 + 4.4619\ F4 - 1.4824\ F5 + 113.6865\ F6 + 150.6382\ F7 - 149.7478$ (3)

The un- normalized output (Current) on Carpet

$I = 0.5977*I_n + 1.2486$ (4)

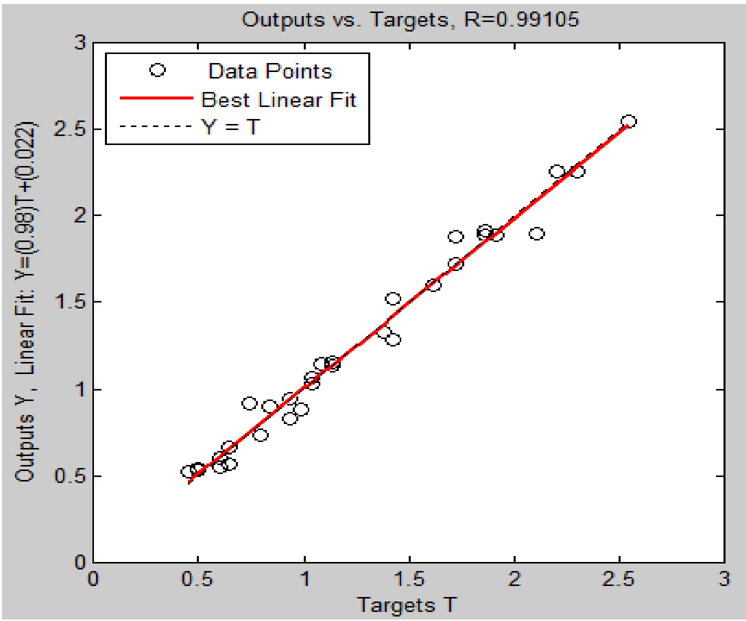


**Fig. 12** Output VS Target for ANN Model testing data

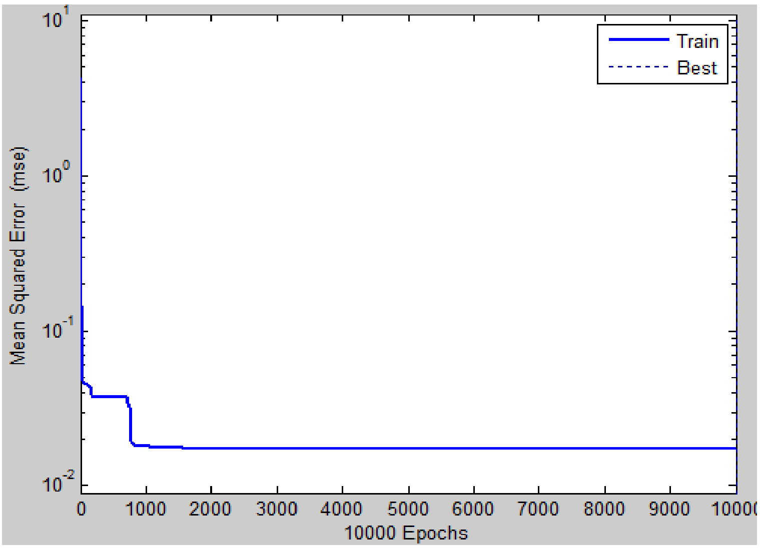


**Fig. 13** Performance for the Model

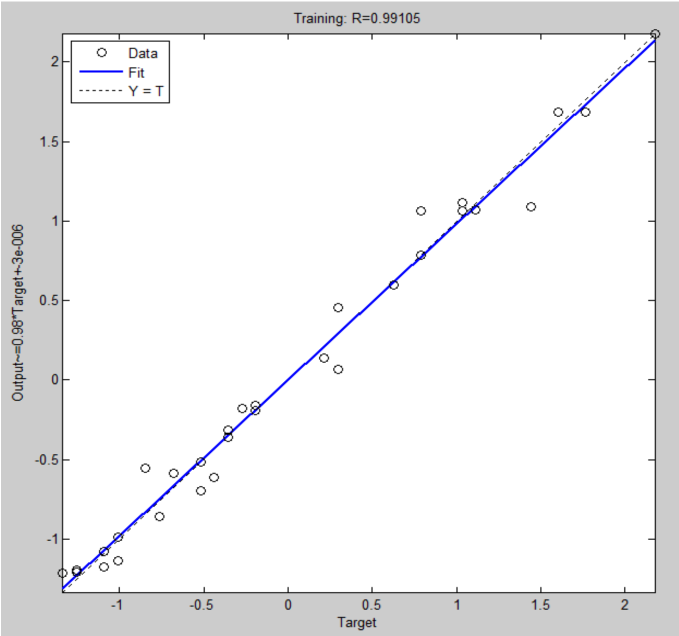


**Fig. 14** Regression for ANN Model

**2nd *ANN Model for the carpet:***

Inputs: Theoretical Time, and Theoretical Velocity

Output: The Current, The Voltage

Neural Network consists of two layers one hidden contains log-sigmoid function with eight neurons and the other is the output layer contains pure-line function with two neurons.

Equation (1) presents the normalized input for the power and the following equations lead to the required derived equation.

$$E1= -30.3556\ Time_n - 34.0484\ Velocity_n + 88.0686$$
$$F1=1 / (1 + \exp(-E1))$$
$$E2= 112.9999\ Time_n + 110.6328\ Velocity_n - 145.2920$$
$$F2=1 / (1 + \exp(-E2))$$
$$E3= 99.6837\ Time_n + 104.0222\ Velocity_n - 287.2837$$
$$F3=1 / (1 + \exp(-E3))$$
$$E4= 110.4102\ Time_n + 103.6221\ Velocity_n + 45.5007$$
$$F4=1 / (1 + \exp(-E4))$$
$$E5= 103.9192\ Time_n + 97.3830\ Velocity_n - 250.3730 \quad (5)$$
$$F5=1 / (1 + \exp(-E5))$$
$$E6= 9.4533\ Time_n + 2.8637\ Velocity_n + 14.2774$$
$$F6=1 / (1 + \exp(-E6))$$
$$E7= -90.8838\ Time_n - 84.2455\ Velocity_n - 269.1573$$
$$F7=1 / (1 + \exp(-E7))$$
$$E8= 0.4986\ Time_n + 2.5868\ Velocity_n + 4.4950$$
$$F8=1 / (1 + \exp(-E8))$$

The normalized current relation from ANN:

$$I_n = 1.0e{+}004 *(-1.8034\ F1 + 0.0001\ F2 - 1.7231\ F3 + 0.00004\ F4 - 0.0801\ F5 - 0.0001\ F6 + 0.0001\ F7 + 0.0003\ F8 + 1.8030) \quad (6)$$

$$V_n = 1.0e{+}004 *(1.2374\ F1 - 0.0001\ F2 + 1.1822\ F3 + 0.0001\ F4 + 0.0551\ F5 + 0.0007\ F6 - 0.0004\ F7 - 0.0019\ F8 - 1.2362) \quad (7)$$

The un- normalized outputs (Current & Voltage) for carpet

$$I = 0.5977*I_n + 1.2486 \quad (8)$$
$$V = 0.0808*V_n + 7.6220 \quad (9)$$

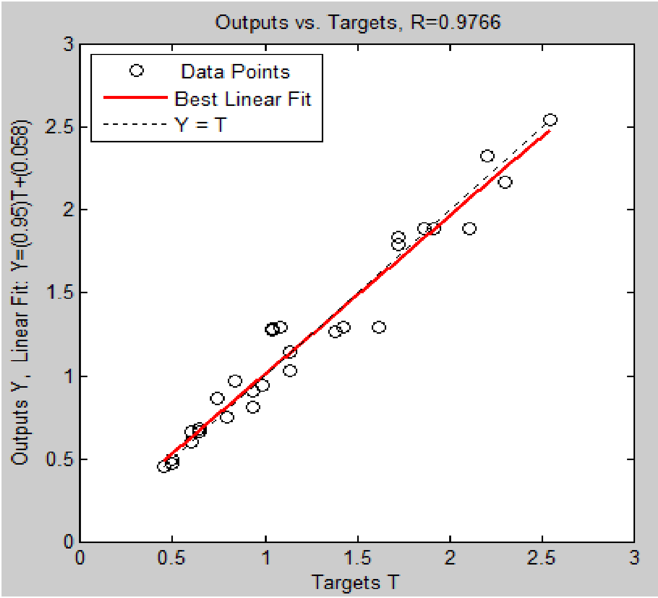


**Fig. 15** Output VS Target for ANN Model testing data

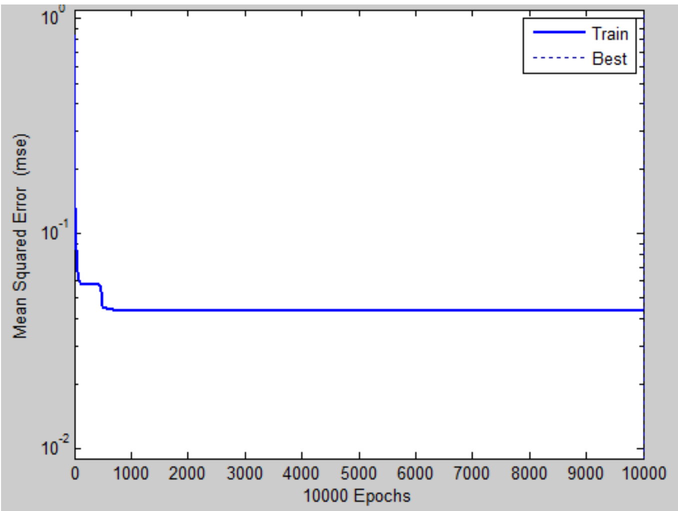


**Fig. 16** Performance for the Model

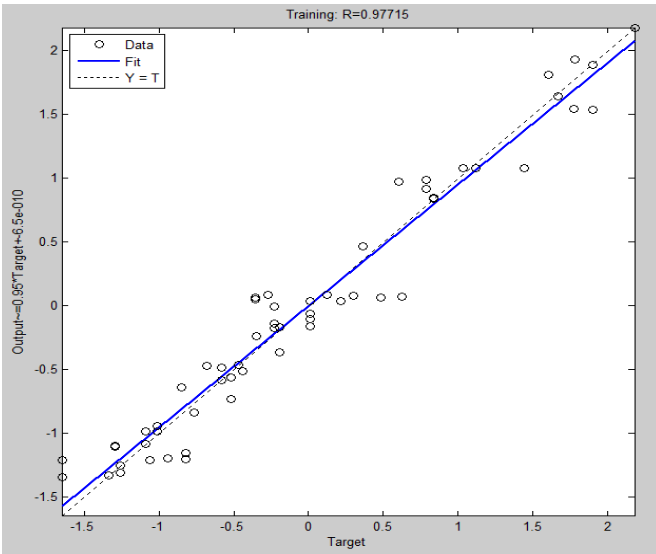


**Fig. 17** Regression for ANN Model

**$3^{rd}$ *ANN Model for the carpet:***

Inputs: Time, Linear Velocity, and Rotational Velocity

Output: The Current, The Voltage

Neural Network consists of two layers one hidden contains log-sigmoid function with seven neurons and the other is the output layer contains pure-line function with two neurons.

The normalized inputs eq.n are:

$Time_n$ = (Time - 1.5000) / (0.9092)

$Linear\_Velocity_n$ = (Linear_Velocity - 0.0617) / (0.0681)

$Rotational\_Velocity_n$ = (Rotational _Velocity - 0.0094) / (0.0206) (10)

Equation (10) presents the normalized input for the power and the following equations lead to the required derived equation.

E1= - 15.5297 $Time_n$ - 5.3634 $Linear\_Velocity_n$ + 2.8217 $Rotational\_Velocity_n$ + 29.0958

F1=1 / (1 + exp (- E1))

E2= - 9.7170 $Time_n$ + 25.6755 $Linear\_Velocity_n$ + 6.7923 $Rotational\_Velocity_n$ + 5.5822

F2=1 / (1 + exp (- E2))

E3= - 185.6356 $Time_n$ + 77.7530 $Linear\_Velocity_n$ + 77.9753 $Rotational\_Velocity_n$ - 38.3748

F3=1 / (1 + exp (- E3))

E4= 3.2609 $Time_n$ - 99.2405 $Linear\_Velocity_n$ - 112.4696 $Rotational\_Velocity_n$ + 97.7252

F4=1 / (1 + exp (- E4))

E5= 17.2092 $Time_n$ - 10.0285 $Linear\_Velocity_n$ - 2.0355 $Rotational\_Velocity_n$ - 4.6875 (11)

F5=1 / (1 + exp (- E5))

E6= - 0.1548 $Time_n$ + 1.3294 $Linear\_Velocity_n$ - 0.7397 $Rotational\_Velocity_n$ - 1.0455

F6=1 / (1 + exp (- E6))

E7= - 0.1677 $Time_n$ + 1.3318 $Linear\_Velocity_n$ - 0.7223 $Rotational\_Velocity_n$ - 1.0656

F7=1 / (1 + exp (- E7))

The normalized current relation from ANN:

$I_n$ = 9.6341 F1 - 7.3508 F2 - 0.0096 F3 - 9.4918 F4 + 4.7189 F5 + 449.9340 F6 - 444.7533 F7 - 2.7859 (12)

$V_n$ = - 2.6704 F1 + 0.8331 F2 + 1.6853 F3 + 0.2252 F4 - 8.1039 F5 + 869.4779 F6 - 869.6830 F7 + 0.5736 (13)

The un- normalized outputs (Current & Voltage) for carpet

I = 0.5977*$I_n$ + 1.2486 (14)

V = 0.0808*$V_n$ + 7.6220 (15)

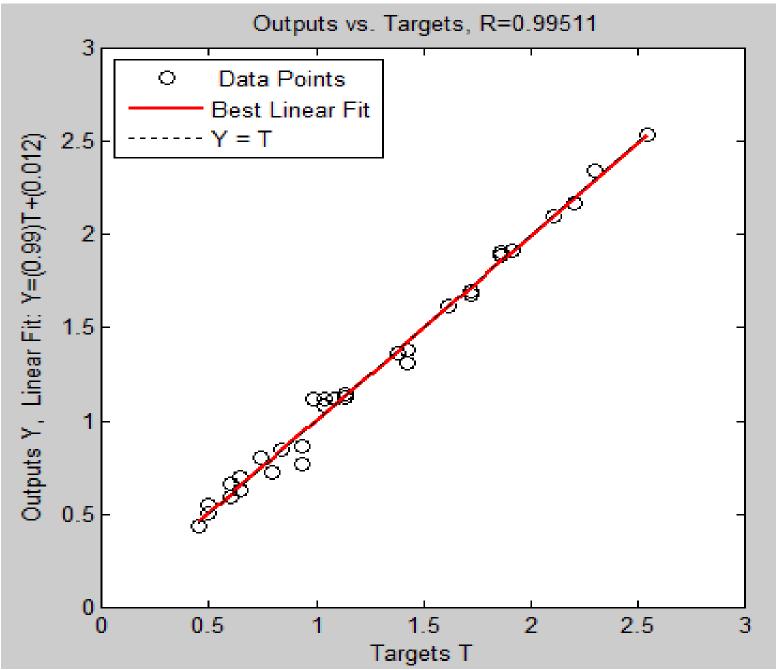


**Fig. 18** Output VS Target for ANN Model testing data

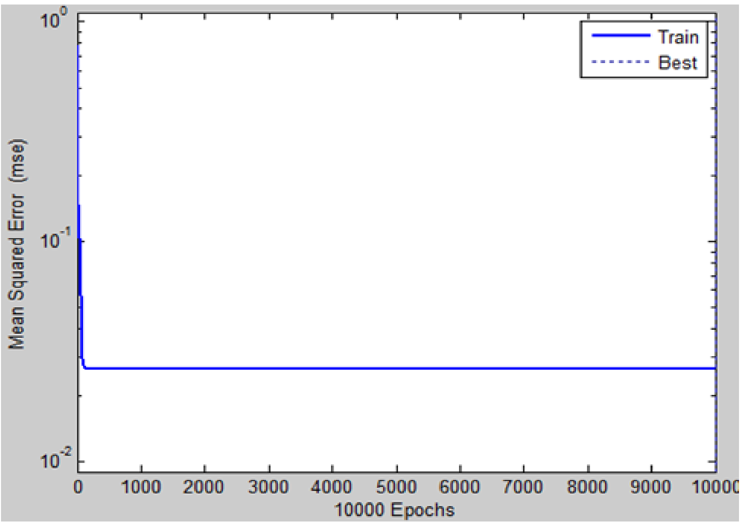


**Fig. 19** Performance for the Model

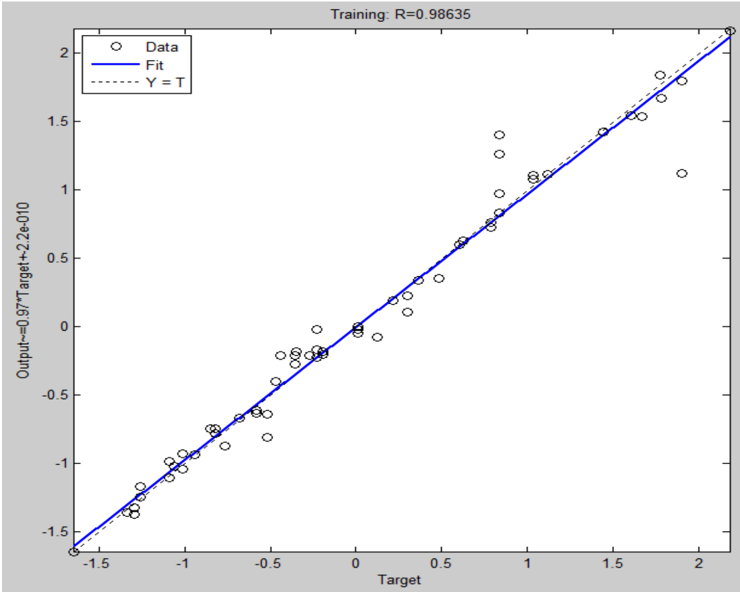


**Fig. 20** Regression for ANN Model

### *B. Hard floor Models*

***ANN Model 1 for the hard floor:***

Inputs: Theoretical Time, and Theoretical Velocity

Output: The Current

Neural Network consists of two layers one hidden contains log-sigmoid function with six neurons and the other is the output layer contains pure-line function with one neuron.

The normalized inputs eq.n are:

$Time_n$ = (Time - 1.5000) / (0.9092)

$Velocity_n$ = (Velocity - 0.1070) / (0.0649) (16)

Equation (16) presents the normalized input for the power and the following equations lead to the required derived equation.

N: Subscript denotes normalized parameters

Ei: Sum of input with input weight and input bias for each node in hidden layer in neural network

Fi: Output from each node in hidden layer to output layer according to transfer function here is logsig

E1= 3.5120 $Time_n$ - 2.1097 $Velocity_n$ - 5.2000

F1=1 / (1 + exp (- E1))

E2= - 4.5842 $Time_n$ - 1.4050 $Velocity_n$ + 6.5328

F2=1 / (1 + exp (- E2))

$E3 = 5.1731 \ Time_n + 0.7477 \ Velocity_n - 6.4648$
$F3 = 1 / (1 + \exp(-E3))$
$E4 = 35.3574 \ Time_n + 38.8344 \ Velocity_n - 32.6172$ (17)
$F4 = 1 / (1 + \exp(-E4))$
$E5 = 180.2645 \ Time_n + 181.4834 \ Velocity_n - 21.6003$
$F5 = 1 / (1 + \exp(-E5))$
$E6 = 100.4598 \ Time_n + 94.6200 \ Velocity_n - 85.7998$
$F6 = 1 / (1 + \exp(-E6))$

The normalized current relation from ANN:

$I_n = 1.0e{+}003 * (0.3371\,F1 - 1.3742\,F2 - 1.3922\,F3 + 5.0763\,F4 - 0.0005\,F5 - 5.0749\,F6 + 1.3728)$ (18)

The un- normalized output (Current) on Hard Floor

$I = 0.6009 * I_n + 1.2139$ (19)

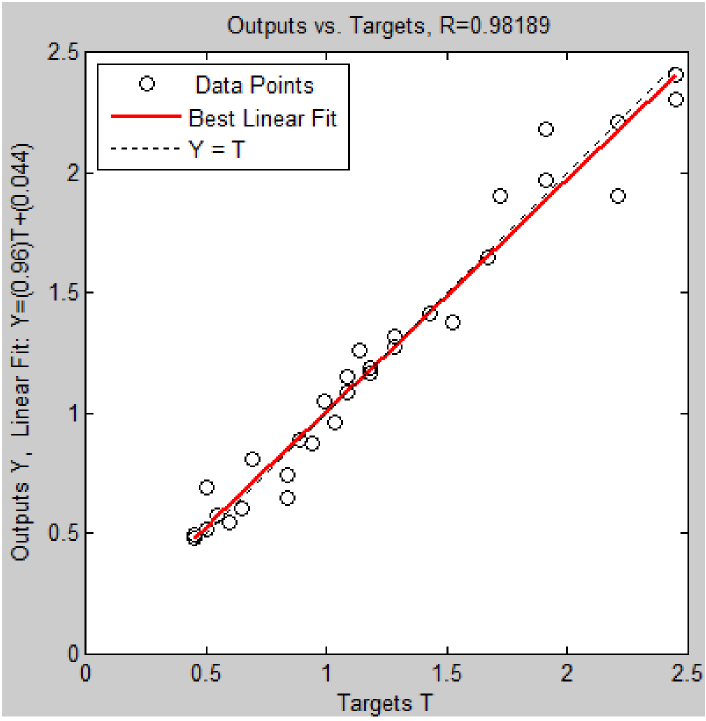


**Fig. 21** Output VS Target for ANN Model testing data

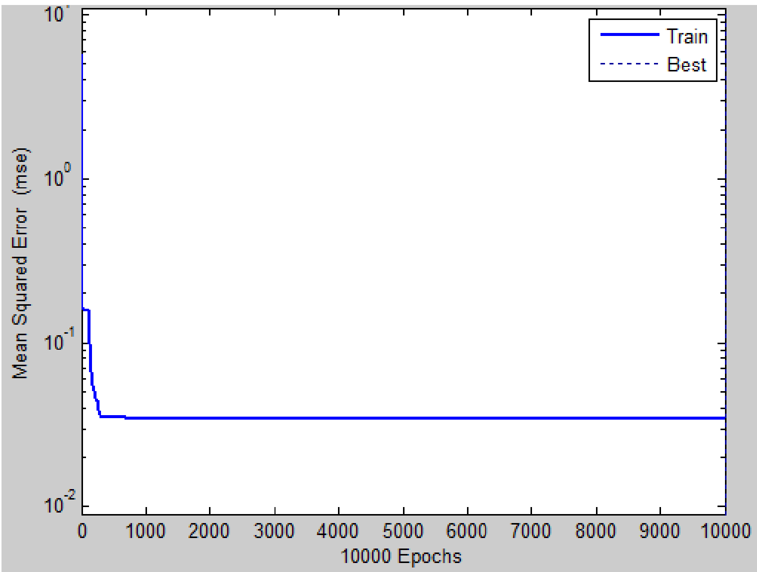


**Fig. 22** Performance for the Model

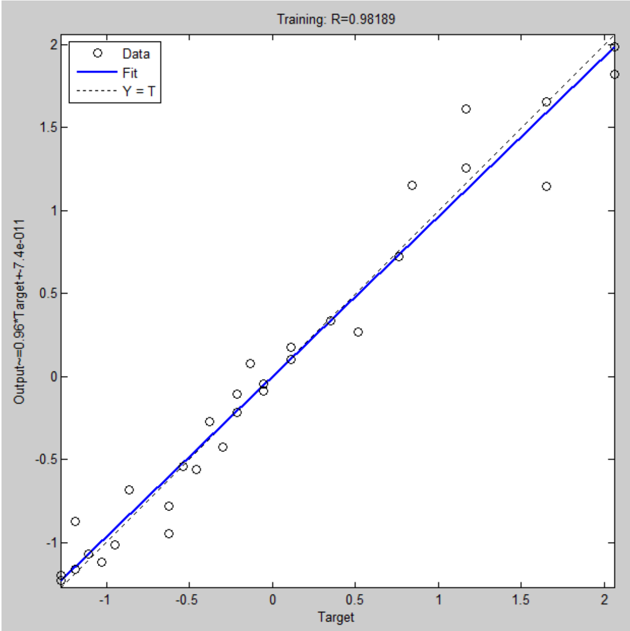


**Fig. 23** Regression for ANN Model

**2nd *ANN Model for the HARD FLOOR:***

Inputs: Theoretical Time, and Theoretical Velocity

Output: The Current, The Voltage ON Hard floor

Neural Network consists of two layers one hidden contains log-sigmoid function with eleven neurons and the other is the output layer contains pure-line function with two neurons.

The normalized inputs eq.n are:

$Time_n$ = (Time - 1.5000) / (0.9092)

$Velocity_n$ = (Velocity - 0.1070) / (0.0649) (20)

Equation (20) presents the normalized input for the power and the following equations lead to the required derived equation.

E1= 6.9451 $Time_n$ - 1.0164 $Velocity_n$ - 6.4228
F1=1 / (1 + exp (- E1))
E2= 3.5580 $Time_n$ + 5.6321 $Velocity_n$ - 5.3752
F2=1 / (1 + exp (- E2))
E3= 91.7443 $Time_n$ + 88.5816 $Velocity_n$ - 40.7935
F3=1 / (1 + exp (- E3))
E4= - 3.6263 $Time_n$ + 4.1488 $Velocity_n$ - 6.0022
F4=1 / (1 + exp (- E4))
E5= 34.0280 $Time_n$ + 30.4938 $Velocity_n$ - 15.3230
F5=1 / (1 + exp (- E5))
E6= 7.6541 $Time_n$ + 0.7352 $Velocity_n$ - 9.3942
F6=1 / (1 + exp (- E6))
E7= - 0.7983 $Time_n$ - 8.6454 $Velocity_n$ + 5.5002
F7=1 / (1 + exp (- E7))
E8= - 14.4237 $Time_n$ - 7.8443 $Velocity_n$ - 33.7609 (21)
F8=1 / (1 + exp (- E8))
E9= - 88.2193 $Time_n$ - 87.1098 $Velocity_n$ - 115.4809
F9=1 / (1 + exp (- E9))
E10= - 3.8894 $Time_n$ - 3.0128 $Velocity_n$ + 7.7715
F10=1 / (1 + exp (- E10))
E11= - 0.0014 $Time_n$ + 7.9550 $Velocity_n$ - 9.9963
F11=1 / (1 + exp (- E11))

The normalized current relation from ANN:

$I_n$ = 1.0e+004 *(0.1844 F1 - 0.1605 F2 - 0.0097 F3 + 0.0555 F4 + 0.0102 F5 0.1222 F6 - 0.1521 F7 - 0.00004 F8 - 0.00004 F9 + 0.3384 F10 + 0.0402 F11 - 0.18650) (22)

$V_n$ = 1.0e+004 *(0.6147 F1 - 0.4426 F2 - 0.3387 F3 - 0.4618 F4 + 0.3406 F5 + 0.4067 F6 - 0.4165 F7 - 0.0002 F8 - 0.0002 F9 + 1.1320 F10 + 0.1380 F11 – 0.71457) (23)

The un- normalized outputs (Current & Voltage) for HARD FLOOR

$I = 0.6009*I_n + 1.2139$ (24)

$V = 0.0768*V_n + 7.6241$ (25)

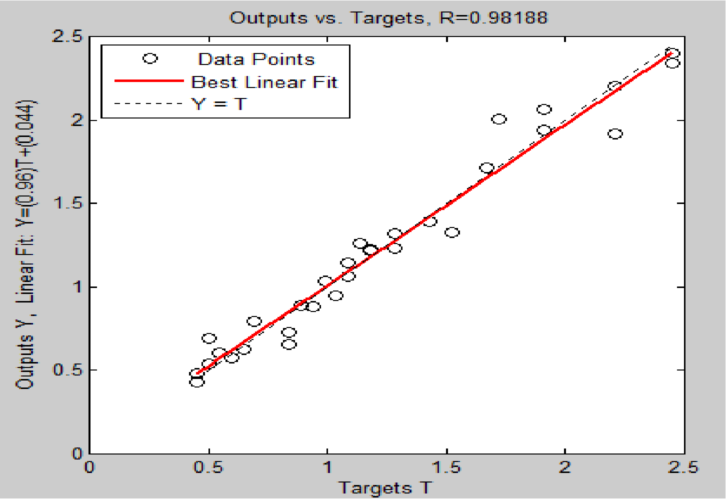


**Fig. 24** Output VS Target for ANN Model testing data

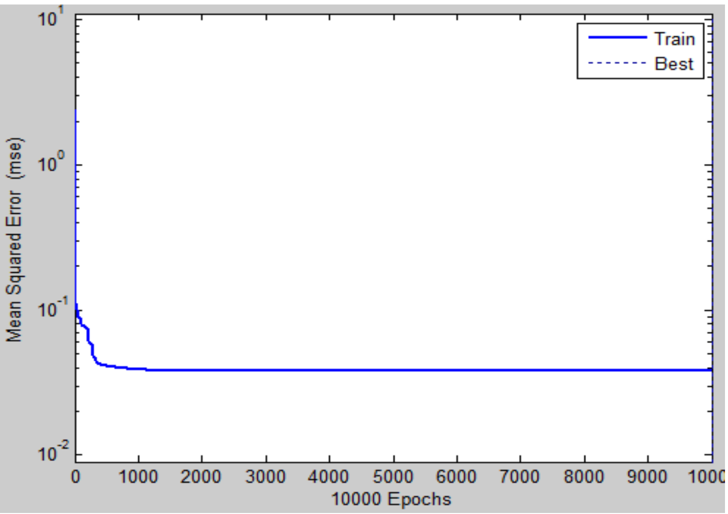


**Fig. 25** Performance for the Model

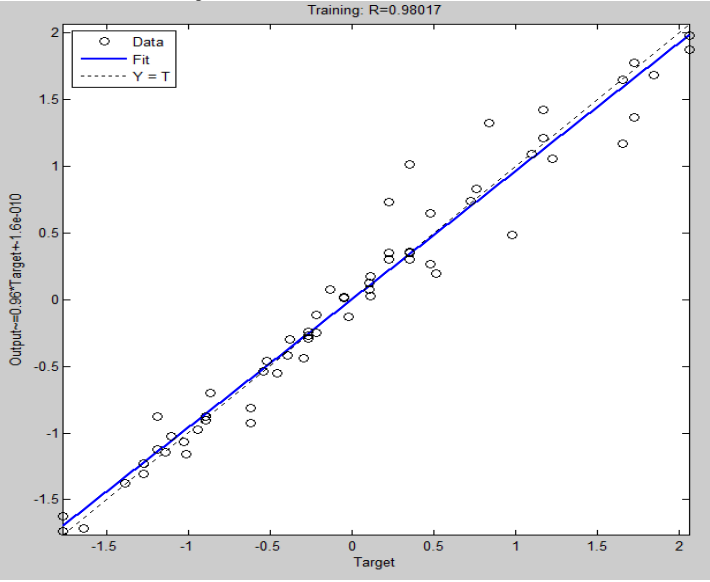


**Fig. 26** Regression for ANN Model

**3rd *ANN Model for the HARD FLOOR:***

Inputs: Time, Linear Velocity, and Rotational Velocity

Output: The Current, The Voltage

Neural Network consists of two layers one hidden contains log-sigmoid function with seven neurons and the other is the output layer contains pure-line function with two neurons.

The normalized inputs eq.n are:

$Time_n = (Time - 1.5000) / (0.9092)$

$Linear\_Velocity_n = (Linear\_Velocity - 0.0590) / (0.0636)$

$Rotational\_Velocity_n = (Rotational\_Velocity + 0.0107) / (0.0239)$ (26)

Equation (26) presents the normalized input for the power and the following equations lead to the required derived equation.

$E1= 6.1011\ Time_n - 4.3725\ Linear\_Velocity_n - 0.5366\ Rotational\_Velocity_n - 7.8921$

$F1=1 / (1 + \exp(- E1))$

$E2= 7.4497\ Time_n - 24.2664\ Linear\_Velocity_n + 2.0769\ Rotational\_Velocity_n + 1.1058$

$F2=1 / (1 + \exp(- E2))$

$E3= - 10.4953\ Time_n + 15.7299\ Linear\_Velocity_n + 3.3590\ Rotational\_Velocity_n - 3.4718$

$F3=1 / (1 + \exp(- E3))$

$E4= 7.1575\ Time_n - 62.6552\ Linear\_Velocity_n + 10.0593\ Rotational\_Velocity_n + 8.0336$ (27)

$F4=1 / (1 + \exp(- E4))$

$E5= - 6.5531\ Time_n + 9.4656\ Linear\_Velocity_n + 1.4800\ Rotational\_Velocity_n - 3.4261$

$F5=1 / (1 + \exp(- E5))$

$E6= - 45.3300\ Time_n + 22.5199\ Linear\_Velocity_n - 23.0080\ Rotational\_Velocity_n - 22.0751$

$F6=1 / (1 + \exp(- E6))$

$E7= 138.7388\ Time_n - 49.9340\ Linear\_Velocity_n + 10.4912\ Rotational\_Velocity_n + 38.7297$

$F7=1 / (1 + \exp(- E7))$

The normalized current relation from ANN:

$I_n = - 31.3817\ F1 - 580.7507\ F2 - 1.5190\ F3 + 580.0468\ F4 + 2.8268\ F5 - 0.2776\ F6 + 0.6847\ F7 - 0.1780$ (28)

$V_n = - 551.2813\ F1 + 106.8142\ F2 - 3.2684\ F3 - 106.6242\ F4 + 2.5206\ F5 + 1.6734\ F6 + 0.7518\ F7 + 0.1575$ (29)

The un- normalized outputs (Current & Voltage) for carpet

$I = 0.5977*I_n + 1.2486$ (30)

$V = 0.0808*V_n + 7.6220$ (31)

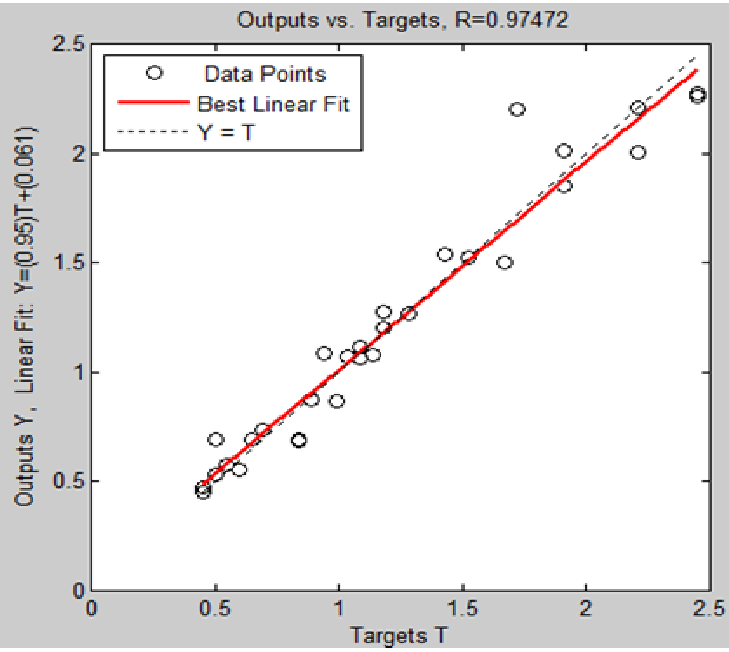


**Fig. 27** Output VS Target for ANN Model testing data

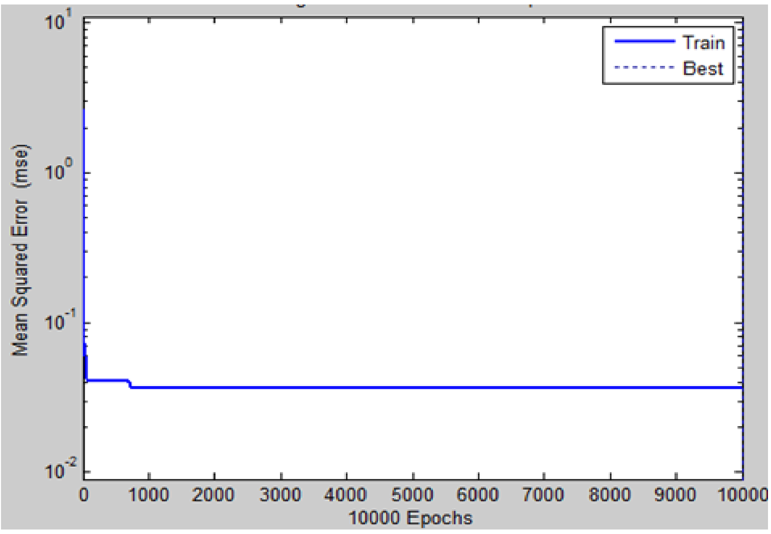


**Fig. 28** Performance for the Model

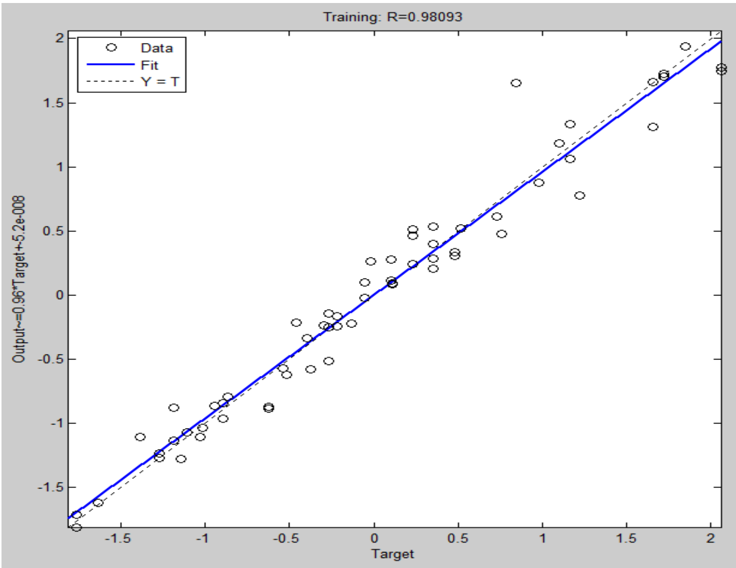


**Fig. 29** Regression for ANN Model

Global model: ANN model with time and speed as inputs and current, voltage, linear speed, rotational speed on carpet, with current, voltage, linear speed, rotational speed on hard floor.

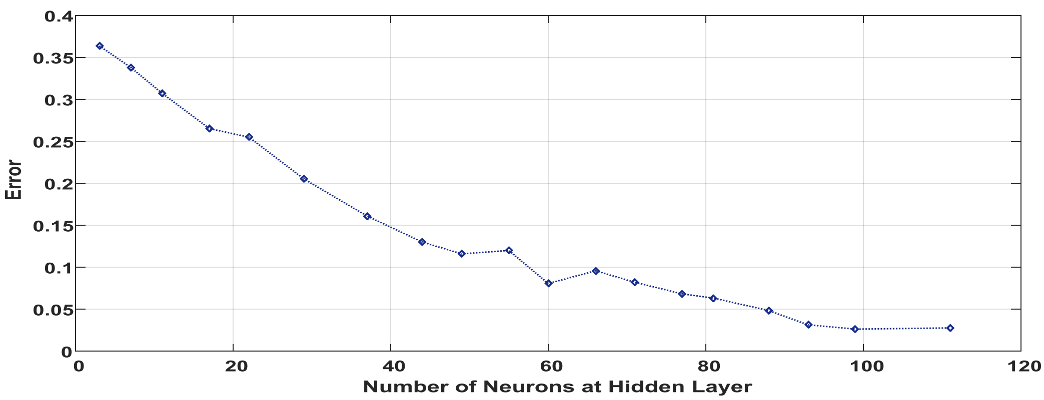


**Fig. 30** Error and Number of Neurons Relation.

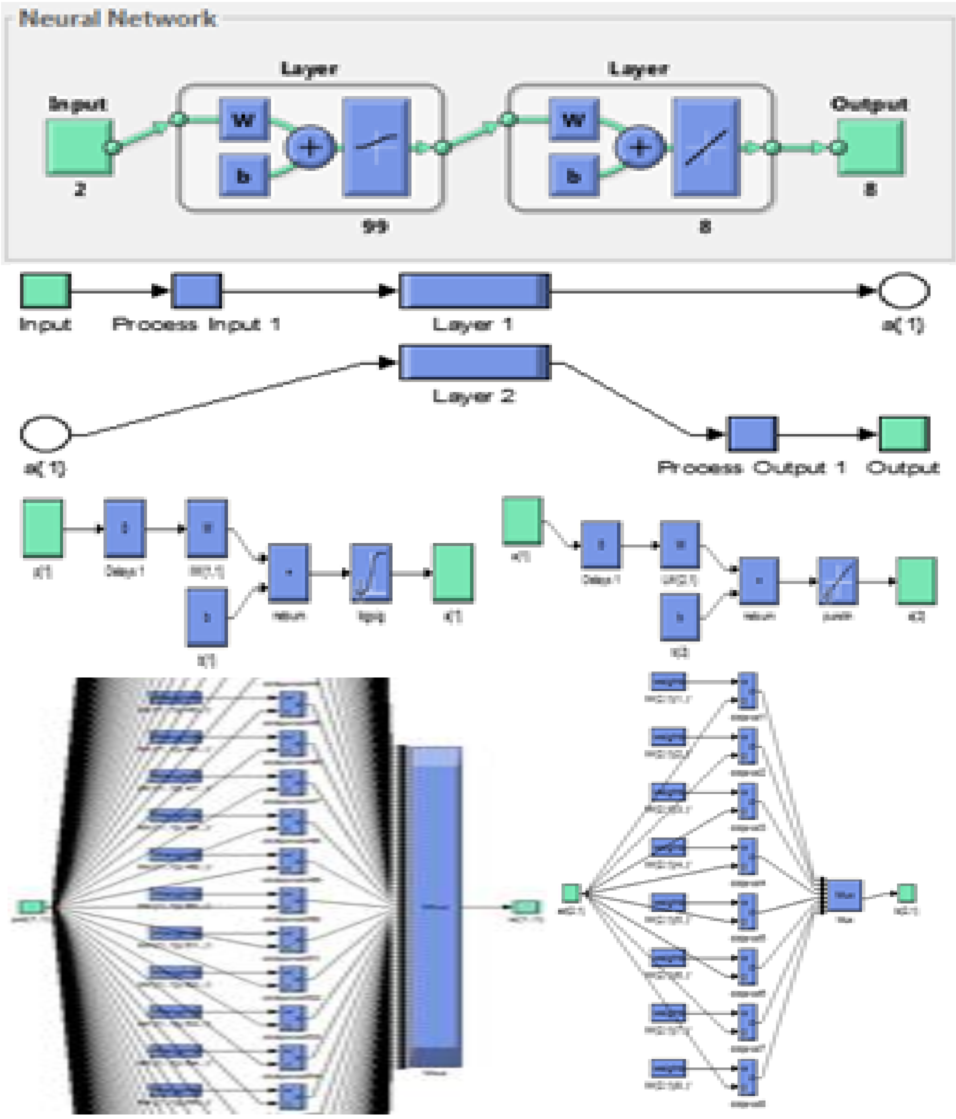


**Fig. 31** Full detailed ANN Simulink Model

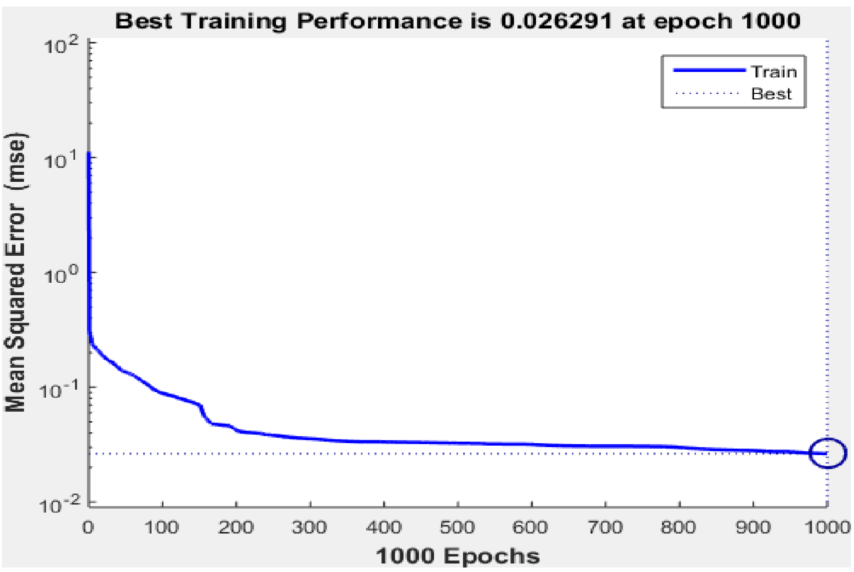


**Fig. 32** ANN Model Error

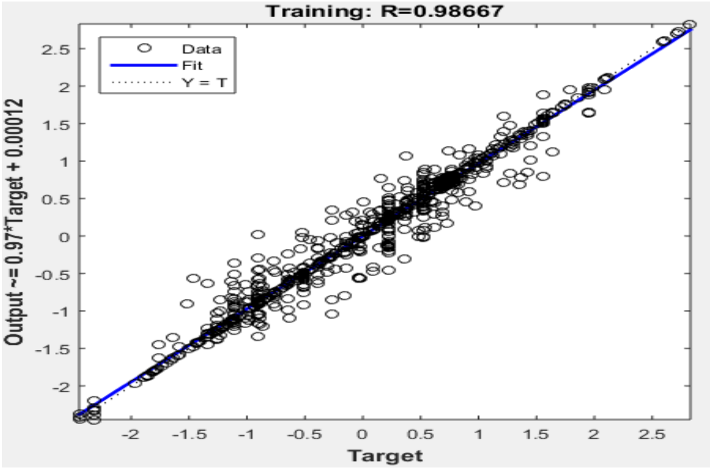


**Fig. 33** ANN Model Regression Constant

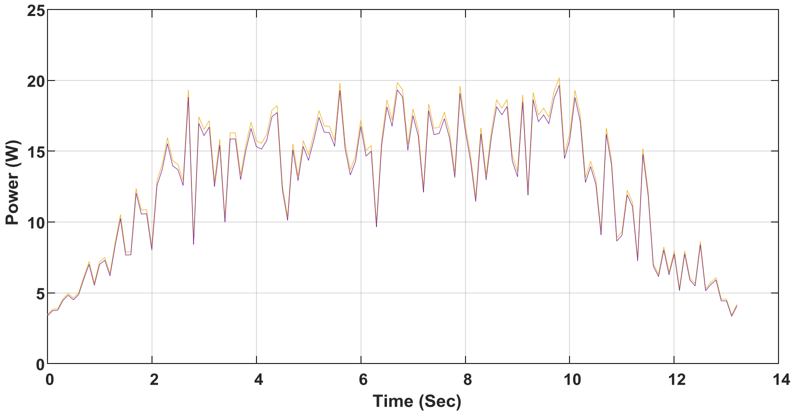


**Fig. 34** ANN Model Power Comparison

## IV. Conclusion

Using the Artificial Neural Network (ANN), with feed forward back-propagation technique to introduce Robot model. This is done to use the ability of neural network for interpolation. It can predict the characteristics and performance of this mobile robot properly. The model are checked and verified by comparing actual and predicted ANN values, with good error and excellent regression factor. Finally, the algebraic equations could be deduced to use them without training the neural unit in each time or using the Simulink model to use them for optimization purposes for future work.

## References


[1] M. Wahab, F. Rios-Gutierrez, A. El Shahat, "Energy Modeling of Differential Drive Robots", IEEE SoutheastCon 2015 Conference, April 9-12, 2015 in Fort Lauderdale, Florida.

[2] S. Liu and D. Sun, "Minimizing energy consumption of wheeled mobile robots via optimal motion planning," Mechatronics, IEEE/ASME Transactions on, vol. 19, no. 2, pp. 401–411, April 2014.

[3] S. Liu and D. Sun, "Modeling and experimental study for minimization of energy consumption of a mobile robot," in Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on, July 2012, pp. 708–713.

[4] S. Liu and D. Sun, "Optimal motion planning of a mobile robot with minimum energy consumption," in Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on, July 2011, pp. 43–48.

[5] S. Derammelaere, S. Dereyne, P. Defreyne, E. Algoet, F. Verbelen, and K. Stockman, "Energy efficiency measurement procedure for gearboxes in their entire operating range," in Industry Applications Society Annual Meeting, 2014 IEEE, Oct 2014, pp. 1–9.

[6] Zeyan Hu, Xiaoguang Zhou, Shimin Wei, "The modeling and controller design of an angular servo robot based on the RBF neural network adaptive control", 2014 International Conference on Advanced Mechatronic Systems (ICAMechS), 10-12 Aug. 2014, pp: 319 - 323

[7] Yanbo Cui, Lei Guo, Shimin Wei, Qizheng Liao, "Design and implementation of a kind of neural networks robustness controller for variable structure bicycle robot's track-stand motion," in Information and Automation (ICIA), 2014 IEEE International Conference on , vol., no., pp.689-694, 28-30 July 2014.

[8] Adel El Shahat, "Smart Homes Systems Technology", Scholar Press Publishing, 2015.

[9] Adel El Shahat, "Artificial Neural Network (ANN): Smart & Energy Systems Applications", Scholar Press Publishing, 2014.

[10] Adel El Shahat, "PV Module Optimum Operation Modeling", Journal of Power Technologies, Vol. 94, No 1, 2014, pp. 50–66.

[11] Adel El Shahat, Rami Haddad, Youakim Kalaani, "An Artificial Neural Network Model for Wind Energy Estimation", IEEE SoutheastCon 2015 Conference, April 9-12, 2015 in Fort *Note that the journal title, volume number and issue number are set in italics.*