Journal of Engineering, Science and Technology
ISSN: 2714-2396, Volume 9 Issue 1

SUPERVISED TRAINING OF INSECT-GAIT MIMICKING HEXAPODS USING DIGITAL TERRAIN MAPPING DATA AND JOINT DATA

E. N. Iyekekpolo [1*], and P.E. Orukpe [2]

[1] Department of Electrical/Electronic Engineering, University of Benin
[2] Department of Electrical/Electronic Engineering, University of Benin

Corresponding email: enosa.iyekekpolo@uniben.edu

Received: 28 April 2026 **Accepted**: Date 30 May 2026 **Published**: 16 June 2026

Abstract. This paper describes the development of a method by which Hexapods with locomotion controllers having machine-learning ability, and whose legs are designed to have the same segment proportions and functional degrees of freedom as a given insect, can have their locomotion controllers trained using supervised learning techniques to closely mimic the locomotion of the insect on which they are modelled. This method can be utilized with little customizations to train any hexapod designed to closely mimic the gait of an insect, hence imbuing the hexapod with a similar level of motion efficiency and smoothness as the insect it is modelled on. The method utilizes Digital Elevation Models (DEM) of terrain over which the model/ training insect walks along with its leg joint variables to create an input/output data structure that can be used for supervised training of the hexapod's controller. The technique involves capturing an insect's motion over a wide range and variety of terrain, after which the insect's leg lengths along with the DEM's will be scaled up to the dimensions of the hexapod to be trained, before being applied with the leg-joint angles as training data to the hexapod's locomotion controllers. This method was validated by modelling and simulation using MATLAB/Simulink to develop a Fine Tree model of the controller, and it was able to predict the correct leg-joint angle in response to the terrain DEMs with a RMSE of 1.3406, after being trained for 23.004 seconds with 101 sets of inputs/output for that particular leg-joint phase.

.

Keywords: DEM (Digital Elevation Model), Hexapod, Joint-angle, Locomotion, Kinematic-Chain.

1. Introduction

The idea of developing robotic versions of animals is an old one that has persisted till current times [1]. This is not surprising considering the admiration that is often evoked by animals over their beauty, efficiency, effectiveness and graceful bearing [2]. Likewise, the use of animals as a means of conveyance has a very long history [3]. So, it is not surprising that ideas on using robotic versions of animals as a means of conveyance have a long history as well [4]. Amongst animal-mimicking robots, hexapods in particular have attracted an appreciable amount of interest because of certain associative, design and operational reasons [5].

Hexapods are commonly associated with walking insects [6]- including ants which are well known for their remarkably agile and graceful gaits, even when bearing burdens that are heavier than their own weight [7]. The design of hexapods is considerably simplified by the relative abundance of

research information on insects upon which hexapod designs can be developed [8]. From an operations point of view, because hexapods possess six legs, they have greater intrinsic mechanical stability than two-legged and four-legged robots [9] – as they can be viewed as a stable tripod that is complementarily stabilized by its mirror image tripod. Also, they do not have so many legs as to make the power consumed by moving the legs impractical [10].

In spite of the significant research interest shown in this type of robot, there exist appreciable difficulty in developing a hexapod to closely mimic the locomotive gaits of insects [11]. This feat is particularly important for hexapods designed to convey delicate or fragile loads that cannot withstand the rigors of a rough 'robotic ride', but require smooth and graceful locomotion gaits even in rough terrain to ensure the safe conveyance of their burdens. This difficulty in closely mimicking insect locomotion gaits persists even in modern artificially intelligent hexapods that possess locomotion controllers with the capacity to learn and improve their performance [12].

This paper describes a methodology by which artificially intelligent hexapods with legs having sections whose lengths are insect-like in proportion and whose functional walking degrees of freedom (DOF) are also insect-like, can be trained using supervised-learning to closely mimic the walking gaits of the insects on which their designs are based. A major objective of this work is to simplify the high-level of complexity in implementing the sensing and control functions required by traditional methods in imbuing hexapods with a high degree of insect-gait mimicry [13]. The method involves three steps compatible with the procedure for using MATLAB's machine learning toolbox for developing machine learning algorithms [14]. The steps are: (1) acquisition of training and validation data, (2) structuring the acquired training and validation data and (3) utilization of the structured data to train (or fit) Machine Learning (ML) models. These steps were simulated in the MATLAB/ Simulink ® 2024 modelling environment and the results indicated that the proposed methodology was viable in achieving its aim. Details of this methodology are presented in section 2. The results of simulating the methodology is discussed in section 3. And the conclusion drawn from the research is stated in Section 4.

2. Materials and Method

This section presents the method used in implementing the steps undertaken to achieve the aim of the research. The method is illustrated on a pair of legs on opposite sides of the body. The pair of legs sampled could be the front-pair, middle pair, or rear-pair of legs. In this method, each pair of legs along with the trunk interconnecting them is treated as a single kinematic chain. The purpose of this method is that at any given time during which the hexapod is 'walking', each leg of this kinematic chain must take a single step exactly like the insect (on which the hexapod is modelled) will do when walking over the same swath of terrain. This is illustrated for the front-leg-pair kinematic chain in Fig. 1. Each of the stages listed in section one, undertaken to train the hexapod's locomotion controller to achieve this, is explained in order in the following subsections.

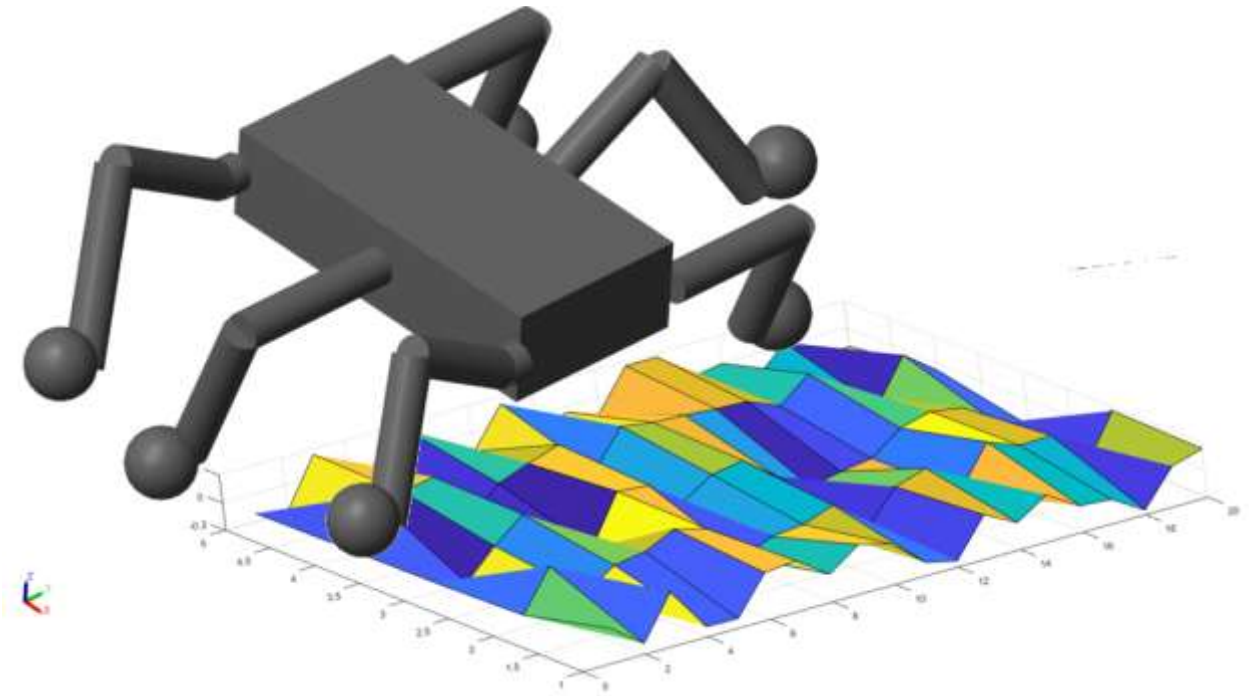

Figure 1: Front Leg-Pair and its Swath of Terrain

2.1. Acquisition of Training and Validation Data

The first step in the training process is to acquire the training/ validation data which is composed of leg-joint parameters and terrain data [15]. The terrain data is in the form of digital elevation models (DEM) of the terrain over which the hexapod is travelling. This composite data was acquired using a technique developed for this research called the Scan-Movement Cycle. This technique is composed of a sequence of operations grouped into numbered Timing Slots. The number value of a timing slot value determines which operations of the sequence are to be executed. The timing slot value begins from one (1) and ends at four (4). The timing slots and the operations associated with them are described in subsections 2.2 to 2.5.

2.2. Timing Slot 1

The first operation of timing slot 1 is to set the Reference Coordinate System (RCS) origin at the ground-contact point one of one of the legs of the leg-pair kinematic chain (the end to be initially grounded). Hence the kinematic chain is considered to begin from this ground-contact point of the initially grounded leg (called leg A), extend to the leg A's tibia, then its femur, then its coxa/trochanter, then extends from the coxa-trunk joint of that leg (A), across the breadth of the trunk to the coxa-trunk joint of the second leg of the pair (called leg B) on the opposite side of the trunk, then extend down to leg B's femur, then its tibia, and ends at the point with which leg B makes ground contact, thus coupling the pair of legs on opposite sides of the body into a single kinematic chain.

The next operation of timing slot 1 is to take a DEM (Digital Elevation Model) scan of the swath of terrain in front of the kinematic chain. The final operation in timing slot 1 is to capture the angles travelled by the joints of the kinematic chain in moving from its initial (home) configuration, to an intermediate configuration called the obstacle clearance configuration (OCC) that lifts leg B off the ground and makes it clear any obstacle in the swath of terrain as shown in Fig. 2.

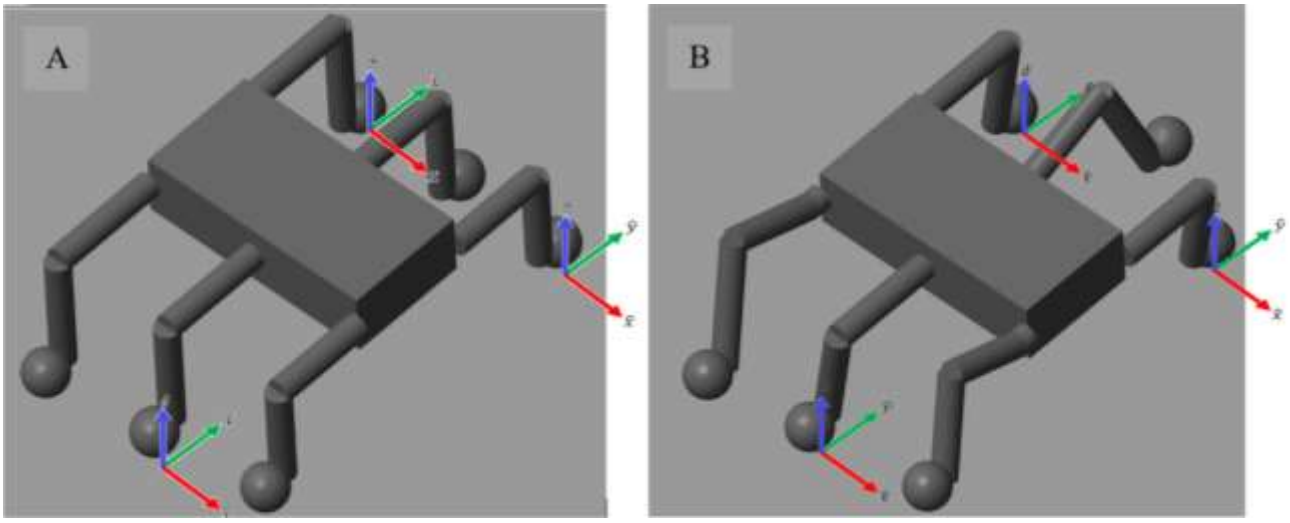

Figure 2: Transformation of the Leg-Pairs from the Home Configuration (A) to the OCC (B)

2.3. Timing Slot 2

The operation of timing slot to 2 is to capture the angles travelled by the joints of the kinematic chain in moving from the OCC to a configuration that places leg B in contact with the ground called the ground target configuration (GTC). This GTC is the home configuration (home configuration 2) for the subsequent movements (transformations) of the kinematic chain.

2.4. Timing Slot 3

The first operation of timing slot 3 is to switch the Reference Coordinate System (RCS) origin from the ground-contact point of leg A to the ground-contact point of leg B. The next operation in timing slot 3 is to capture the angles travelled by the joints of the kinematic chain in moving from home configuration 2 (GTC of timing slot 2) to an intermediate obstacle clearance configuration (OCC) that lifts leg A off the ground and makes it clear any obstacle in its swath of terrain.

2.5. Timing Slot 4

There is only one operation of Timing slot 4 which is to capture the joint angles travelled from the OCC to the ground target configuration (GTC) that places the foot of leg A in contact with the ground (within the scanned swath of terrain). This ends one iteration or cycle of the SCAN-MOVEMENT CYCLE.

Figure 3 presents the timing slot sequences. To execute another two-step movement cycle of the leg-body-leg kinematic chain, the sequence of operations from Timing Slot 1 to Timing Slot 4 is repeated.

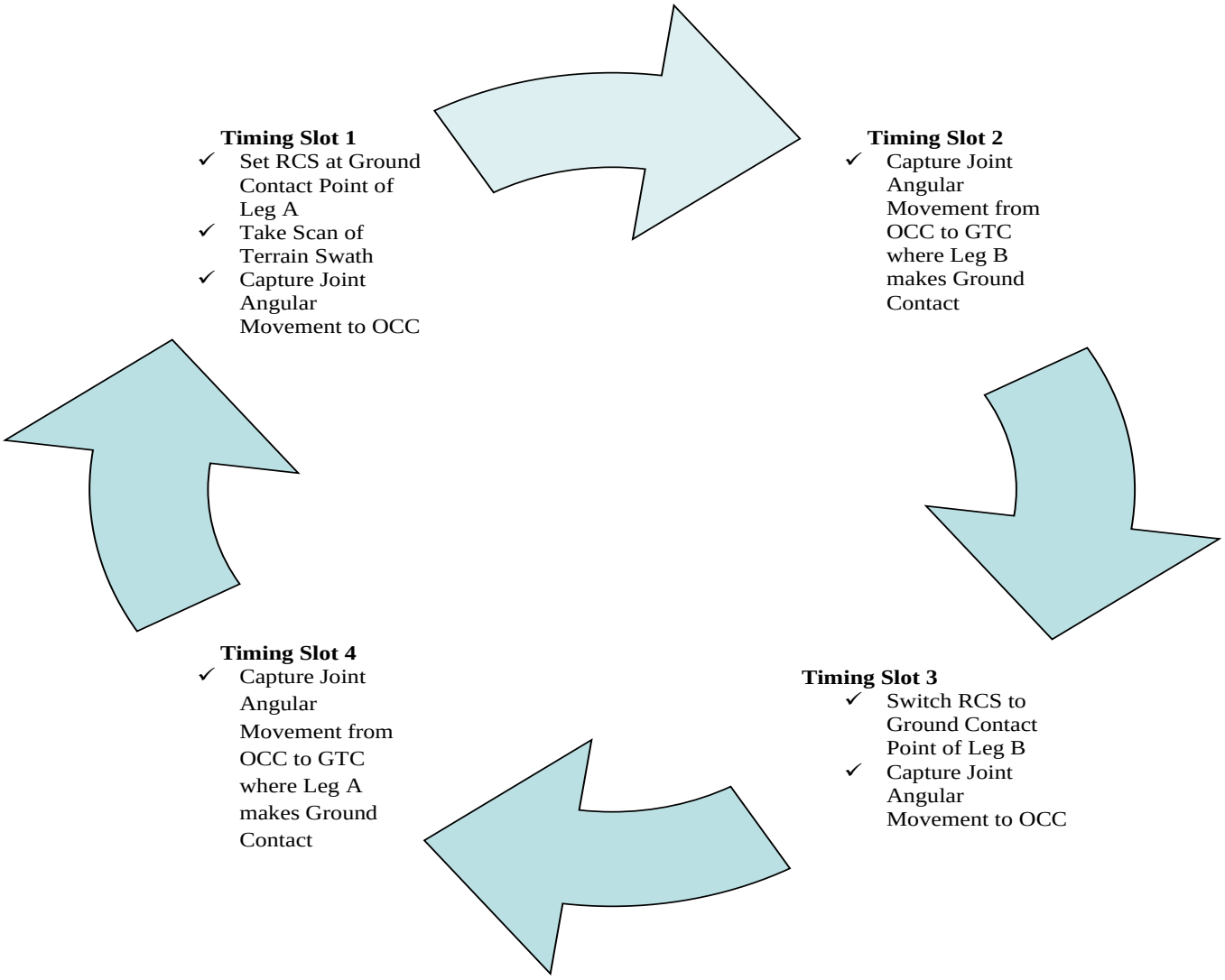

Figure 3: Four Timing-Step Iteration of SCAN-MOVEMENT CYCLE

2.6. Structuring the Acquired Training and Validation Data

The training and validation data composed of the timing data, terrain data (DEM) and the joint angle (travelled) data is structured to train the ML controller to execute a SCAN-MOVEMENT iteration composed of two steps of the kinematic chain.

At the highest level, the data is categorized into input data and output data. The input data is then further categorized into timing data and terrain data, while the output data is categorized into the angles travelled by each joint of the kinematic chain composed of six joints. The structure is as illustrated in Table 1.

Table 1: Structure of Training and Validation Data

Input Data		Output Data					
Timing Data	Terrain Data (DEM)	Angle Travelled by Joint 1	Angle Travelled by Joint 2	Angle Travelled by Joint 3	Angle Travelled by Joint 4	Angle Travelled by Joint 5	Angle Travelled by Joint 6

The timing input data is composed of a single timing number that indicates the current phase of the SCAN-MOVEMENT cycle for which the joint travel-angle outputs are appropriate. The DEM data is composed of a rectangular array of numbers which represent the topographical elevation of a grid of points in the swath of terrain within which the kinematic chain is to take two steps.
Hence the entire training training/validation data is structured into an array of rows and columns, in which each row is a sample that trains the controller to control the joints of the kinematic chain in response to timing data and DEM of the terrain.

2.7. Utilization of the structured data to train (or fit) Machine Learning (ML) models.
For each kinematic chain either a single algorithm could be trained to predict all six output variables of the output data set, using the input data set as shown in equation (1), or six algorithms could be created, with each algorithm dedicated to predicting only one output variable from the output data set as shown in equations (2) to (7).

$$[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = function(Timing\ Data, DEM\ Data) \quad (1)$$

$$[\theta_1] = function(Timing\ Data, DEM\ Data) \quad (2)$$

$$[\theta_2] = function(Timing\ Data, DEM\ Data) \quad (3)$$

$$[\theta_3] = function(Timing\ Data, DEM\ Data) \quad (4)$$

$$[\theta_4] = function(Timing\ Data, DEM\ Data) \quad (5)$$

$$[\theta_5] = function(Timing\ Data, DEM\ Data) \quad (6)$$

$$[\theta_6] = function(Timing\ Data, DEM\ Data) \quad (7)$$

where θ_x is the angle travelled by joint x of the kinematic chain, and 'function' is the name of the machine-learning function that computes the output data from the input data. Ten percent of the one hundred and one sample data set was set-aside to be used to validate the accuracy of the algorithms developed as shown in Fig 4. The Regression Learner Application of MATLAB/Simulink was used to create a Fine Tree algorithm to predict a single joint-travel output variable from the input variables.

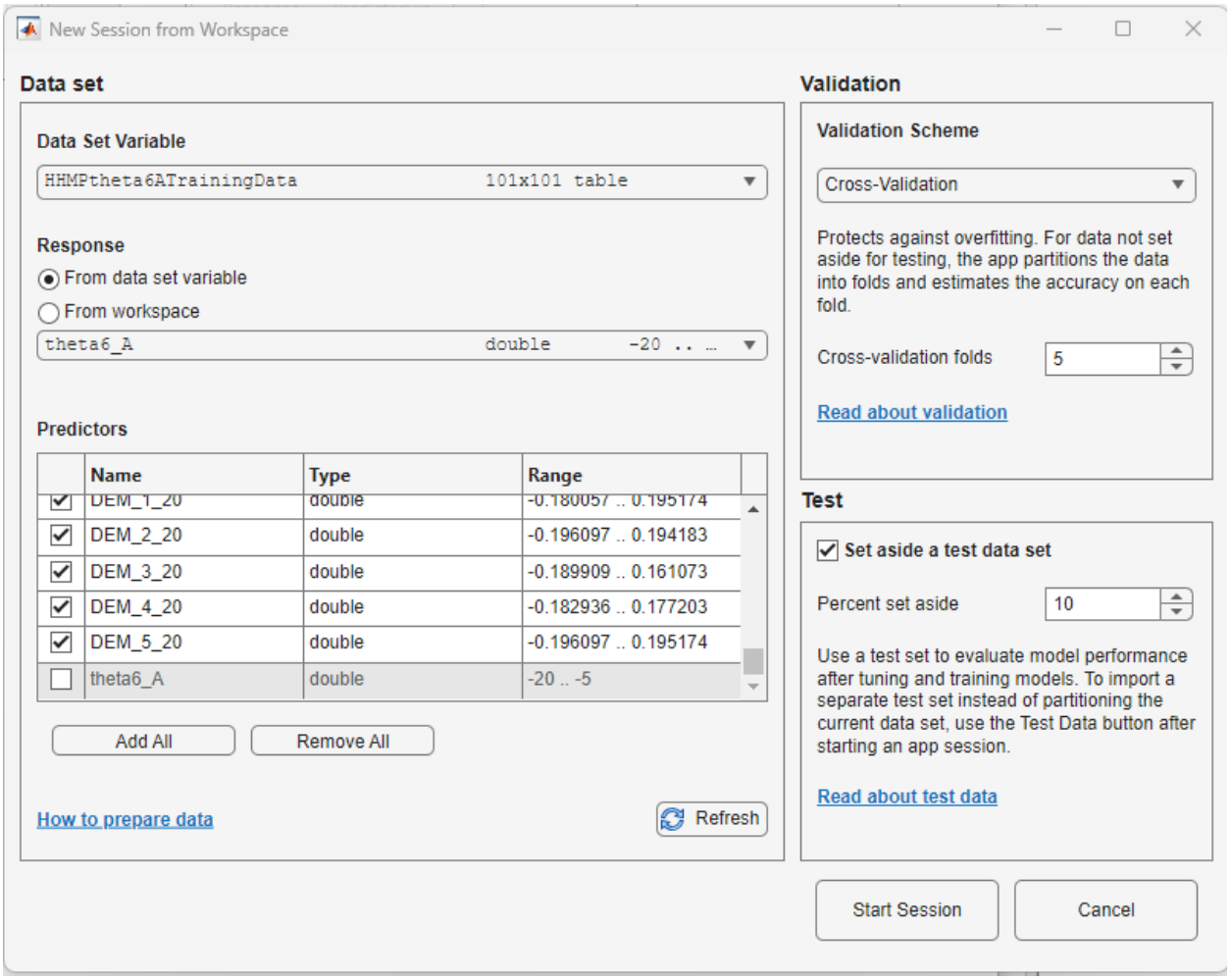

Figure 4: Configuring the Sample Data-Set for Training and Validation

3. Results and Discussion

After being trained for 23.004 seconds on 90% of the set of 101 input/output samples (setting-aside 10% of the data set for validation), the Fine Tree algorithm was able to predict the correct angle travelled by the joint with a RMSE (Root Mean Square Error) of 1.3406. Figure 5 presents the graph generated by the Regression Learner Application (of MATLAB) that shows the root-mean-square-error (RMSE) of all the twenty-six models (or algorithms) developed in this research. From Fig 5 the twenty-six algorithms developed and their RMSE values are presented in Table 2.

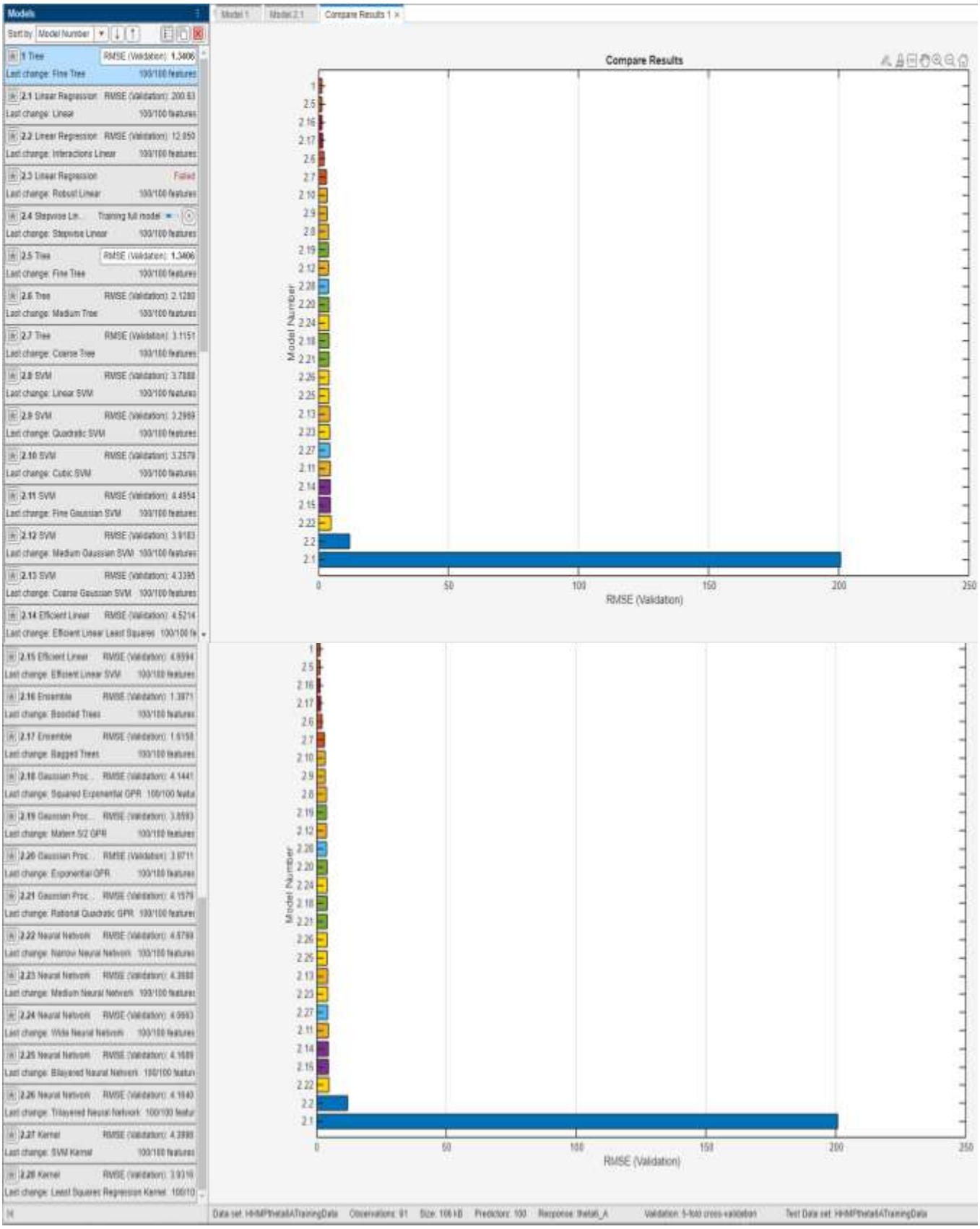

Figure 5:Result Summary and Graphs Showing the RMSE of all the Models Trained and Validated.

Table 2: Models Developed and their RMSE Values

S/N	Name of Model	RMSE
1.	Linear Regression	200.6300
2.	Interactions Linear Regression	12.0500
3.	Fine Tree	1.3406
4.	Medium Tree	2.1280
5.	Coarse Tree	3.1151
6.	Linear SVM	3.7888
7.	Quadratic SVM	3.2969
8.	Cubic SVM	3.2579
9.	Fine Gaussian SVM	4.4954
10.	Medium Gaussian SVM	3.9183
11.	Coarse Gaussian SVM	4.3395
12.	Efficient Linear Least Squares	4.5214
13.	Efficient Linear SVM	4.6594
14.	Boosted Trees Ensemble	1.3971
15.	Bagged Trees Ensemble	1.6158
16.	Squared Exponential GPR	3.9711
17.	Matern GPR	3.8593
18.	Exponential GPR	3.9711
19.	Rational Quadratic GPR	4.1579
20.	Narrow Neural Network	4.8769
21.	Medium Neural Network	4.3688
22.	Wide Neural Network	4.0663
23.	Bilayered Neural Network	4.1689
24.	Trilayered Neural Network	4.1640
25.	SVM Kernel	4.3998
26.	Least Squares Regression Kernel	3.9316

4. Conclusion

The results of this research indicate that the methodology proposed is viable and can be applied with slight modifications in training insect-mimicking hexapods with artificially intelligent locomotion controllers to closely mimic the gait patterns of the insects on which their designs are based.

References

1. Truitt, E. R., 2015. *Medieval Robots: Mechanism, Magic, Nature, and Art.* 1st ed. Philadelphia: University of Pennysylvania Press.
2. Bekey, G. A., 2005. *Autonomous Robots: From Biological Inspiration to Implementation and Control.* 1st ed. Cambridge: MIT Press.
3. Diamond, J., 1997. *Guns, Germs, and Steel: The Fates of Human Societies.* 1st ed. New York: W. W. Norton & Company.
4. Silva, M. F. and Machado, J. T., 2012. A literature review of the optimization of legged robots. *Journal of Vibration and Control,* 18(12), pp. 1753-1767.
5. Tedeschi, F. and Carbone, G., 2014. *Design issues for hexapod walking robots*. 1st ed. New York: Nova Science Publishers.
6. Wikipedia, 2024. Hexapod (robotics). [Online]. Available at: https://www.en.wikipedia.org
   [Accessed 3 May 2026].
7. Dupeyroux, J., Passault, G., Ruffier, F., Violet, S. and Serres, J., 2017. Hexapod: a small 3D-printed six-legged walking robot designed for desert ant-like navigation tasks. Toulouse, *IFAC World Congress 2017*.
8. Manoonpong, P., Pasemann, F. and Worgotter, F., 2013. Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots. *Frontiers in Neurorobotics, 7*(3).
9. Tejado, I., Torres, D., Perez, E. and Vinagre, B., 2021. C-Legged Hexapod Robot Design Guidelines Based on Energy Analysis. *Applied Sciences, 11*(6), p. 2513.
10. Cservenak, A. and Budai, C., 2022. Mechanical Design and a Novel Structural Optimization Approach for Hexapod Walking Robots. Machines, 10(6), p. 409.
11. Suzuki, S., Kano, T., Ijspeert, A. and Ishiguro, A., 2021. Simple analytical model reveals the functional role of embodied sensorimotor interaction in hexapod gaits. PLOS *Computational Biology, 17*(3), p. 2.
12. Liu, Y., Fan, X., Ding, L., Wang, J., Liu, T. and Gao, H., 2022. Adaptive Gait Generation for Hexapod Robots Based on Reinforcement Learning and Hierarchical Framework. *Machines, 10*(6), p. 417.
13. Iyekekpolo, E. N. and Orukpe, P. E., 2025. Conceptual Design and Modelling of a Hybrid Hexapod Mobility Platform. *Journal of Engineering for Development, 17, Special Issue: University of Benin, Faculty of Engineering First International Conference June 18 to 19, 2025*, pp. 77-86.
14. The MathWorks Inc., 2025. Statistics and Machine Learning Toolbox, [Online]. Available: https://www.mathworks.com. [Accessed 23 August 2025].
15. The MathWorks Inc., 2025 Supervised Learning Workflow and Algorithms, [Online]. Available: https://www.mathworks.com. [Accessed 23 August 2025].