

---

This item was submitted to [Loughborough's Research Repository](#) by the author.  
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

## Machine learning comparison for step decision making of a bipedal robot

PLEASE CITE THE PUBLISHED VERSION

<https://doi.org/10.1109/ICCRE.2018.8376427>

PUBLISHER

© IEEE

VERSION

AM (Accepted Manuscript)

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Kouppas, Christos, Qinggang Meng, Mark A. King, and Dennis Majoe. 2019. "Machine Learning Comparison for Step Decision Making of a Bipedal Robot". figshare. <https://hdl.handle.net/2134/35200>.

# Machine Learning Comparison for Step Decision Making of a Bipedal Robot

1<sup>st</sup> Christos Kouppas  
Computer Science  
Loughborough University  
Loughborough, UK  
C.Kouppas@lboro.ac.uk

2<sup>nd</sup> Qinggang Meng  
Computer Science  
Loughborough University  
Loughborough, UK  
Q.Meng@lboro.ac.uk

3<sup>rd</sup> Mark King  
Sports Biomechanics  
Loughborough University  
Loughborough, UK  
M.A.King@lboro.ac.uk

4<sup>th</sup> Dennis Majoe  
Computer Science  
ETH Zürich  
Zürich, Switzerland  
dennis.majoe@inf.ethz.ch

**Abstract**—This paper<sup>1</sup> presents the results of several machine learning techniques for step decision in a bipedal robot. The custom developed bipedal robot does not utilize electric motors as actuators and as a result has the disadvantage of imprecise movements. The robot is inherently unstable and maintain its stability by making steps. The classifiers had to learn when and which leg must be moved in order to maintain stability and locomotion. Methods like: Decision tree, Linear/Quadratic Discriminant, SVM, KNN and Neural Networks were trained. The results of their performance/accuracy are noted.

**Keywords** - Decision tree; Linear/Quadratic Discriminant; SVM; KNN; Neural Networks; Bipedal Robot; LSTM

## I. INTRODUCTION

Bipedal robots have been studied for decades, beginning with passive designs in early 80s [1] and later, with more power consuming models [2]. The stability of lower limbs is still not fully defined, in comparison with the upper part; arm manipulators are well defined due to extensive use in industry. During the current project, a novel bipedal robotic host was designed to be more efficient, both electromechanically and computationally. Electromechanically, the robot consists of a pair of legs and a small torso which will, in the future, accommodate manipulators. The recently accepted paper, “De-signing a novel bipedal Silent Agile Robust Autonomous Host (S.A.R.A.H)”, describes the main design characteristics of the robot [3]. Computationally, the robot must be able to react quickly and accurately but not precisely. Additionally, it will have the ability to learn during operation to improve its performance.

The robot, S.A.R.A.H. (Safe Agile Robust Autonomous Host), combines gait pattern generators and a “brain” which will decide when to make steps. A similar system with in humans [4], [5]. The actual decision making process must be taught by humans who have the experience of walking on two limbs. By transferring the human knowledge to a bipedal robot, the actions of the robot will be more natural. To transfer that knowledge and capture the useful information, a classifier must be chosen or designed.

Classifiers are used to categories data in different groups based on information that is available. That classification can be happened by using cluster, decision tree or more complex algorithms. The most common classifiers are described below:

- *Decision tree* classifiers are simple, very fast but inaccurate in complex problems. They are used in simple tasks like wheeled robots avoiding obstacles [6].

- *Linear or Quadratic Discriminant* classifiers are fast and accurate in a simple problem. They can handle more complex problems than the decision tree technique, e.g. detecting a fall of a bipedal robot [7].
- *Support Vector Machine (SVM)* classifier is more powerful in complex problems. However, the classifier is able to process the data that are currently present and does not have memory of previous states. One example, that is similar to the current project, is the classification of falling on a simulated bipedal lower limb robot in Open Dynamic Engine [8].
- *K-Nearest Neighbor (KNN)* classifier is more robust than SVM and can solve more complex problems by creating close-region clusters. Despite the flexibility, it has the same disadvantage as the SVM; it does not have memory and processes only the current information. KNN has been used for decision making of different walking path to walk over unknown slopes [9].
- *Neural Network (NN)* is a powerful multi-purpose tool. Simple configurations can define a classifier which is able to have multiple inputs/outputs. The NN offers the flexibility of “stacking” several NN on top of each other which means, the outputs of the classifier can connect as inputs to a second NN for controlling. There are a lot of examples of researchers extracting gait for bipedal robots [10], [11] or combining them with pattern generation for control [12].
- *Long Short-Term Memory (LSTM)* is an addition to NN to provide memory for the previous states of the system in NN’s calculation. It has all the benefits of a NN because inherently, it is a more sophisticated version of NN. LSTM can be used to learn features from tasks, like human’s gait [13] or behaviors/actions from a robot [14].



**Figure 1:** S.A.R.A.H. standing on a treadmill, supported by an elastic band.

<sup>1</sup> The project was partially funded from EPSRC and Innovate UK’s scheme “Emerging and Enabling Technologies”.

## II. EXPERIMENTAL PROCEDURE

S.A.R.A.H was designed to have 12 actuators, which are connected in antagonist pair. It also combines one 6 DoF IMU and 8 sliding potentiometers as the sensor system.

The actuators are connected to microcontrollers that contains gait pattern generators. The inputs that the microcontrollers require, are when and which leg to move.

The experimental procedure started with the treadmill moving backward with a slow speed, around 0.5 km/h (Figure 1). Then a user had to decide when and which leg the robot has to move. The length of the experiments were 10 - 15 minutes and 4 users were used to collect the data. This was done to reduce bias towards one specific user. The data that were captured were the 14 sensor inputs and, also, the input from the user (when and which leg is moved).

### A. Data Captured

The 14 inputs that were captured can be divided in two main groups, the high frequency (IMU data) and low frequency (sliders' data in roll of angular measurements). The data that were recorded are the raw values from the sensors, without filtering or bias removing. That was happened in order to have as much information is possible.

A sample of the acceleration and gyro rates of the IMU data can be seen in Figure 2. From the graphs, it can be clearly determined when a step is made, however, it is harder to determine with which leg the step was made. A sample of the low frequency data is demonstrated in Figure 3. In those data is clearer which leg is moving however, comparing them with the IMU data there is a delay in the response. This presents the need of both types of data to have an effective classification of the user's inputs. The last data that were captured, were the input of the user and a class was create with three labels, "Left Leg Key", "No Key Pressed" and "Right Leg Key".

## III. CLASSIFICATION

The data that were captured were used to train a classifier that will give the optimal performance without being too complex. The classifier will have 14 inputs and a single, three labelled, classification (user input). In order to make a predictor, the "Left Leg Key" and "Right Leg Key" labels were copied 10 times before the time that the actual key was pressed. Each label was assigned in a number; -1 to "Left Leg Key", 0 to "No Key Pressed", 1 to "Right Leg Key".

After the training, the classifier will be used in an on-board processor to make the decision online, alongside with the user. The processor on the robot is a low power Raspberry Pi 3, which means a complex classifier would take longer time to execute and make predictions. Additionally, the time between the predictions will include data capturing, data forming and data preprocessing that they may need. To minimize the prediction time, the raw data from the sensors were preferable, to eliminate the pre-processing time. The prediction must be quicker than humans' average reaction time and ideally half that time.

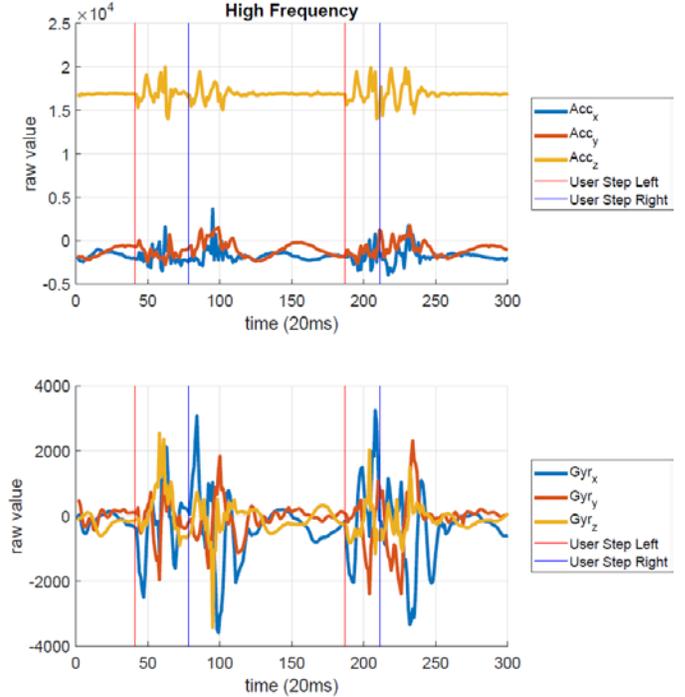


Figure 2: High Frequency Signals, IMU.

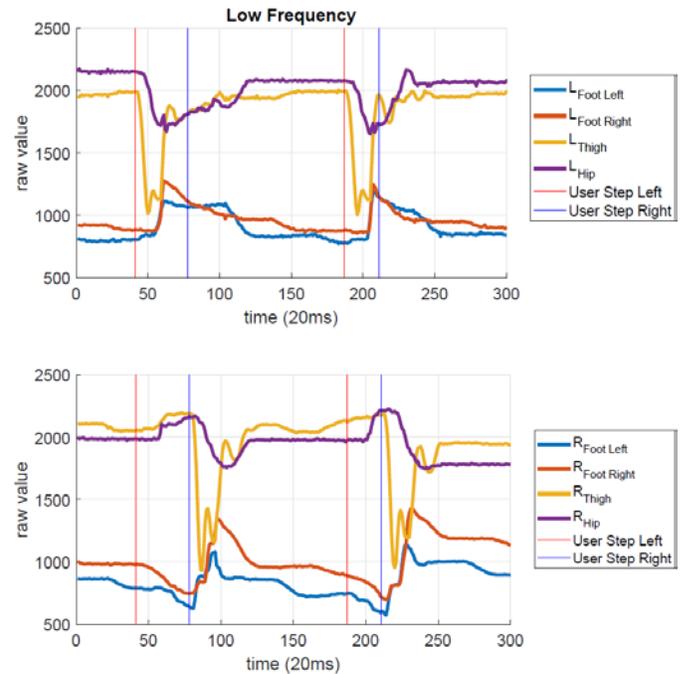


Figure 3: Low Frequency Signals, Sliders.

A total number of 20 classifiers were analyzed:

- Decision Tree classification was set up with three different settings. The models were Simple (4 Number of splits), Medium (20 Number of splits) and Complex (100 Number of splits) (All with Gini's diversity index, No surrogate decision splits).

- Discriminant classifier was split in two, the Linear and the Quadratic (Both with Full covariance structure).
- Support Vector Machine (SVM) classifier was divided in six different models: Linear (Auto kernel scale), Quadratic (Auto kernel scale), Cubic (Auto kernel scale), Coarse Gaussian (kernel scale = 15), Medium Gaussian (kernel scale = 3:7) and Fine Gaussian (kernel scale = 0:94) (All with Box constrain level = 1).
- K-Nearest Neighbor (KNN) classifier was divided also in six different models: Weighted (10 Number of Neighbors, Euclidean distance, Squared inverse weights), Cubic (10 Number of Neighbors, Minkowski distance, Equal weights), Cosine (10 Number of Neighbors, Cosine distance, Equal weights), Coarse Gaussian (100 Number of Neighbors, Euclidean distance, Equal weights), Medium Gaussian (10 Number of Neighbors, Euclidean distance, Equal weights) and Fine Gaussian (1 Number of Neighbors, Euclidean distance, Equal weights).
- Neural Network (NN) was categorized in Small (3 Layers, 50 neurons each), Medium (3 Layers, 100 neurons each) and Big (5 Layers, 100 neurons each). The training was completed with 500 inner epochs (do not change the cost value) and 100 external epochs (change the cost value). Loss function was set to categorical crossentropy and the optimization method to Adam.
- Long Short-Term Memory (LSTM) followed the same structure with NN, both in design and training, however every odd layer was replaced with an LSTM Layer with a time memory of 50. Thus, the categories were Small (LSTM-Normal-LSTM, 50 neurons each), Medium (LSTM-Normal-LSTM, 100 neurons each), Big (LSTM-Normal-LSTM-Normal-LSTM, 50 neurons each) and Deep (LSTM-Normal-LSTM-Normal-LSTM, 100 neurons each).

#### IV. RESULTS

Classifiers were trained with all raw sensor data (14 inputs - 14 features) as inputs and one class as the output; classification. Some of the classifiers used Principal Component Analysis (PCA) to reduce the dimensionality of the problem. The variances, that were used for PCA, were 90%, 95% and 99% which resulted in 6, 6 and 9 features, respectively. Those features were arbitrary and had not physical meaning.

Additionally, with post-processing and human heuristics, a set of features with physical meaning were extracted, e.g. left foot front, right foot front, torso lean front or back etc.

Each feature was representing an individual discrete physical position. Those features were used individually to train the same classifiers to predict the class. However, the results were at least 10% less than the ones with the raw data thus, they were omitted from the paper.

Measuring the performance was not straight forward because, if the classifier was classifying everything as “No key pressed” it was achieving an accuracy of 81.9%. To eliminate that, after the training all models (except NN) were checked that the mislabeled data were spread in all labels. The table I is summarizing the numerical results of the accuracy.

As it can be seen in table I, classifiers without memory cannot achieve more than 93% accuracy. Because a continues result of 0, will achieve an accuracy of 81.9%, an accuracy of 93% was actually 60%. The classifier had to predict the other 18% to achieve a 100%, by achieving 93% was performing an improvement of 11% out of 18%. A dynamic problem, like walking, cannot be described with models without memory and that is confirmed with those results. In the case of Neural Networks (NN), the simple stack of layers with neurons, performance was the worst because the pure postures could not translate in prediction of leg steps. However, replacing some layers with Long Short-Term Memory (LSTM) layers in NN improved the results and outperformed the other classifiers.

**Table 1: Results Of The Classifiers**

Classifiers		PCA No	PCA 90%	PCA 95%	PCA 99%
Decision Tree	Simple	86.1%	82%	82%	85.5%
	Medium	85.1%	82.5%	82.5%	84.9%
	Complex	85.1%	82.5%	82.5%	84.4%
Discriminant	Linear	54.6%	81.8%	81.8%	84.3%
	Quadratic	64.8%	61.7%	61.7%	64.3%
SVM	Linear	85.3%	81.9%	81.9%	84.7%
	Quadratic	88%	81.9%	81.9%	85.5%
	Cubic	89.2%	76.8%	77.5%	87%
	Coarse Gaussian	85.2%	81.9%	81.9%	84.9%
	Medium Gaussian	87.9%	82.3%	82.3%	85.7%
	Fine Gaussian	91.1%	85.2%	85.2%	88.7%
KNN	Weighted	92.8%	86.9%	86.9%	91.4%
	Cubic	91.1%	85.9%	85.9%	89.4%
	Cosine	90.8%	84.8%	84.8%	88.6%
	Coarse Gaussian	86.6%	84.3%	84.3%	86.3%
	Medium Gaussian	91.4%	85.7%	85.7%	89.5%
	Fine Gaussian	92.1%	85.3%	85.3%	90.9%
NN	Small	81.9%	-	-	-
	Medium	81.9%	-	-	-
	Big	81.9%	-	-	-
LSTM	Small	94.1%	-	-	-
	Medium	97.7%	-	-	-
	Big	94.2%	-	-	-
	Deep	98.3%	-	-	-

LSTM classifiers achieved an accuracy of 98.3% which is an improvement of 16.4% out of 18.1%; 90.6% actual improvement from a continuous 0 response. Analyzing the performance of LSTM, the results showed that a better performance can be achieved by increasing the number of neurons and not by increasing the number of layers. The number of parameters that were trained, were proportional to the performance increase. The best model was the Deep LSTM and had 227,303 parameters which is almost double of the second best model (Medium LSTM) that had 136,803 parameters. The Small LSTM and Big LSTM had 35,903 and 58,653 parameters, respectively.

Moving a step forward, the best classifier (LSTM) was implemented on a Raspberry Pi 3 that is hosted in the robot. Then the experiments were ran again but that time instead of capturing the data, a live prediction of the movements was printed on a screen. The results were impressive, with the predictor able to produce 20-30 predictions per second (with the Medium model) which is faster than humans' reaction time [15]–[17]. Additionally, the predictions were correct and most of them were slightly faster than the user input.

## V. CONCLUSION

In the current paper, the comparison between different, static and dynamic, classification methods are presented. The classifiers must learn the decision of when and which leg will move in a bipedal robot by using humans' experience. The algorithm must run on-board on a Raspberry Pi 3, thus low computational classifiers are preferred. The bipedal robot that was used, is described in our previous conference paper in UK - Robotics and Autonomous Systems conference [3].

Static classifiers do not have a good performance on the given problem, with Weighted KNN model performing the best out of all with a maximum accuracy of 92.8%, which is actual 60% on real predictions. Also, human cannot heuristically define features to improve the performance of the classifiers. Simple Neural Networks did not perform well since postures cannot describe the time of the leg move, they may give the information of which leg to move but that is not enough. The treadmill did not have a constant speed and that made the problem more dynamic with no measurable information, like walking speed and acceleration. That information require memory in the classifiers in order to be extracted.

Dynamic classifiers have memory that change with time. The dynamic classifier that was tested, was a neural network with LSTM layers. As it was measured, increase of the neurons in each layer improve the prediction more than the increase of layers. The highest accuracy that was achieved was 98.3% (90.6% actual accuracy) with the Deep model but the running time on board was 50% more than the time of the Medium model, with accuracy 97.7% (87.3% actual accuracy). Thus, the Medium model was preferred and moved on board, side by side with the user. The prediction rate, that was achieved, was 20 - 30 predictions per second and they were correct and sometimes faster than the user.

## VI. FUTURE WORK

The next steps is to let the classifier actually control the legs and walk with different speeds. To achieve that, more user sequences of 10-15 minutes, must train the networks in order to validate and make the classifier more robust. Additionally, dimensionality reduction of the inputs will be implemented to improve execute time on the on-board computer without compromising the accuracy.

## ACKNOWLEDGMENT

The project is partially funded from Innovate UK's scheme "Emerging and Enabling Technologies" and the "Engineering and Physical Sciences Research Council" (EPSRC) of UK. We thank, also, Motion Robotics LTD, a company based in Southampton, for the collaboration on the robot design and prototype.

## REFERENCES

- [1] T. Mita, T. Yamaguchi, T. Kashiwase, and T. Kawase, "Realization of a high speed biped using modern control theory," *International Journal of Control*, vol. 40, no. 1, pp. 107–119, 1984.
- [2] K. Akbari Hamed and J. W. Grizzle, "Reduced-order framework for exponential stabilization of periodic orbits on parameterized hybrid zero dynamics manifolds: Application to bipedal locomotion," *Nonlinear Analysis: Hybrid Systems*, vol. In review, 2016.
- [3] C. Kouppas, M. Rodosthenous, N. Sagyndyk, Q. Meng, M. King, and D. Majoe, "Designing a novel bipedal Silent Agile Robust Autonomous Host (S.A.R.A.H)," in *Robotics and Autonomous Systems: robots work-ing for and among us*, In press.
- [4] J. T. Choi and A. J. Bastian, "Adaptation reveals independent control networks for human walking," *Nature Neuroscience*, vol. 10, no. 8, pp. 1055–1062, aug 2007.
- [5] J. Nielsen, "How we Walk: Central Control of Muscle Activity during Human Walking," *The Neuroscientist*, vol. 9, no. 3, pp. 195–204, 2003.
- [6] Y. J. Li, W. C. Chou, C. Y. Chen, B. Y. Shih, L. T. Chen, and P. Y. Chung, "The development on obstacle avoidance design for a humanoid robot based on four ultrasonic sensors for the learning behavior and performance," *IIEEM2010 - IEEE International Conference on Industrial Engineering and Engineering Management*, no. Figures 1, pp. 376–379, 2010.
- [7] K. Ogata, K. T. K. Terada, and Y. K. Y. Kuniyoshi, "Falling motion control for humanoid robots while walking," *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pp. 306–311, 2007.
- [8] J. J. Kim, T. Y. Choi, and J. J. Lee, "Falling avoidance of biped robot using state classification," in *Proceedings of 2008 IEEE International Conference on Mechatronics and Automation, ICMA 2008*, 2008, pp. 72–76.
- [9] J. Nagasue, Y. Konishi, N. Araki, T. Sato, and H. Ishigaki, "Slope-walking of a biped robot with k nearest neighbor method," *ICIC Express Letters*, vol. 4, no. 3, pp. 893–898, 2010.
- [10] D. Lee and W. ElMaraghy, "A neural network solution for bipedal gait synthesis," in *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, vol. 2. IEEE, jun 2011, pp. 763–768.
- [11] F. Wang, Y. Zhang, S. Wen, and T. Ning, "An on-line gait generator for bipedal walking robot based on neural networks," *2011 6th IEEE Conference on Industrial Electronics and Applications*, pp. 2449–2453, 2011.
- [12] S. F. Rashidi, M. R. S. Noorani, M. Shoran, and A. Ghanbari, "Gait generation and transition for a five-link biped robot by Central Pattern Generator," *2014 2nd RSI/ISM International Conference on Robotics and Mechatronics, ICRoM 2014*, pp. 852–857, 2014.

- [13] Y. Feng, Y. Li, and J. Luo, "Learning effective Gait features using LSTM," *Proceedings - International Conference on Pattern Recognition*, pp. 325–330, 2017.
- [14] D. N. T. How, K. S. M. Sahari, Hu Yuhuang, and Loo Chu Kiong, "Multiple sequence behavior recognition on humanoid robot using long short-term memory (LSTM)," *2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pp. 109–114, 2014.
- [15] R. Abbasi-Kesbi, H. Memarzadeh-Tehran, and M. J. Deen, "Technique to estimate human reaction time based on visual perception," *Healthcare Technology Letters*, vol. 4, no. 2, pp. 73–77, 2017.
- [16] J. E. Birren and J. Botwinick, "Age Differences in finger, jaw, and foot reaction time to auditory stimuli," *Journal of Gerontology*, vol. 10, no. 4, pp. 429–432, 1955.
- [17] G. R. Grice, R. Nullmeyer, and V. A. Spiker, "Human reaction time: toward a general theory." *Journal of Experimental Psychology: General*, vol. 111, no. 1, pp. 135–153, 1982.