International Journal of Mechanical Engineering and Robotics Research Vol. 7, No. 4, July 2018

S.A.R.A.H.: The Bipedal Robot with Machine Learning Step Decision Making

Christos Kouppas and Qinggang Meng
Computer Science, Loughborough University, Loughborough, UK
Email: C.Kouppas@lboro.ac.uk, Q.Meng@lboro.ac.uk

Mark King
Sports Biomechanics, Loughborough University, Loughborough, UK
Email: M.A.King@lboro.ac.uk

Dennis Majoe
Computer Science, ETH Zürich, Zürich, Switzerland
Email: dennis.majoe@inf.ethz.ch

***Abstract*—Herein, we describe a custom-made bipedal robot that uses electromagnets for performing movements as opposed to conventional DC motors. The robot uses machine learning to stabilize its self by taking steps. The results of several machine learning techniques for step decision are described. The robot does not use electric motors as actuators. As a result, it makes imprecise movements and is inherently unstable. To maintain stability, it must take steps. Classifiers are required to learn from users about when and which leg to move to maintain stability and locomotion. Classifiers such as Decision tree, Linear/Quadratic Discriminant, Support Vector Machine, K-Nearest Neighbor, and Neural Networks are trained and compared. Their performance/accuracy is noted.**

***Index Terms*—Decision tree, Linear/Quadratic Discriminant, SVM, KNN, Neural Networks, Bipedal Robot, LSTM**

I. Introduction

Bipedal robots have been studied for decades, starting with passive designs in the early 1980s [1], leading to the development of more power consuming models later [2]. Despite the 35 years of study, lower limb stability is still not fully defined in comparison with that of the upper part; arm manipulators are well defined owing to their extensive use in industry. In the current study, a novel bipedal robotic host is designed to be efficient, both electromechanically and computationally. Electromechanically, the robot consists of a pair of legs and a small torso that will, in the future, accommodate manipulators. In a recently accepted paper entitled, "Designing a novel bipedal Silent Agile Robust Autonomous Host (S.A.R.A.H)," the main design characteristics of the robot were described [3]. Computationally, the robot must be able to react quickly and accurately but not precisely. Additionally, it must have the ability to learn during operation to improve its performance.

A. Bipedal Robot

The robot S.A.R.A.H. (Safe Agile Robust Autonomous Host) combines gait pattern generators and a "brain" that will decide when to take a step. The robot's design is inspired by the way in which flightless birds walk. In general, ostriches, consume less energy to walk than humans [4]. The structural characteristic of a central unit controlling gait pattern generators exists in humans and in animals [5], [6]. The actual decision-making process must be taught by humans, who have the experience of walking on two limbs. By transferring human knowledge to a bipedal robot, the robot can be enabled to move more naturally. To transfer that knowledge and capture useful information, a classifier must be selected or designed.

Mechanically, this involves combining the patent pending "bang-bang" actuators developed by Motion Robotics LTD [7] with hydraulics. The actuators use power from electromagnets to generate the torque required to rotate the robot's joints. The hydraulics is used for damping and braking. For control, five Atmel microchips [8] and two Raspberry Pi 3 (RPi3) units [9] are used hierarchically. One of the RPi3 units is responsible for controlling motion (written in C) and the other is responsible for learning and executing classifications (written in Python).

B. Classifiers

Classifiers are used to categorize data into different groups based on the available information. That classification can be achieved using cluster, decision tree, or more complex algorithms. The most common classifiers are as follows:

Manuscript received January 5, 2018; revised April 8, 2018.
This paper is an extend of the paper "*Machine Learning Comparison for Step Decision Making of a Bipedal Robot*" that was presented at International Conference on Control and Robotic Engineering (ICCRE2018).

doi: 10.18178/ijmerr.7.4.379-384

- *Decision tree* classifiers are simple and very fast, but they are inaccurate in terms of handling complex problems. They are to perform simple tasks such as obstacle avoidance in wheeled robots [10].
- *Linear or Quadratic Discriminant* classifiers are fast and accurate in solving simple problems. They can handle more complex problems than the decision tree technique, for example, fall detection in a bipedal robot [11].
- *Support Vector Machine (SVM)* classifier is more powerful in solving complex problems. However, the classifier can only process current data and does not remember previous states. An example that is similar to the current project is the classification of falling in the case of a simulated bipedal lower limb robot in the Open Dynamic Engine environment [12].
- *K-Nearest Neighbor (KNN)* classifier is more robust than SVM and can solve more complex problems by creating close-region clusters. Despite its flexibility, it is limited in the same way as SVM: it does not have memory and processes only current information. KNN has been used for selecting a walking path from a set of paths for walking over unknown gradients [13].
- *Neural Network (NN)* is a powerful multi-purpose tool. Simple configurations can be used to define a classifier that can have multiple inputs/outputs. NN offers the flexibility of "stacking" several NNs on top of each other, which means, the outputs of a classifier can be connected as inputs to a second NN for control. There are many examples of researchers extracting gaits of bipedal robots [14], [15] or combining them with pattern generation for control by using NNs [16].
- *Long Short-Term Memory (LSTM)* is an addition to NN to provide memory of the previous system states in the NN calculations. It offers 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, such as a human's gait [17] or behaviors/actions of a robot [18].

C. Locomotion

Robot locomotion is different in comparison to that of typical humanoids because it is important to increase efficiency at the cost of dexterity. S.A.R.A.H. has 12 Degrees of Freedom (DoFs), of which 6 are controllable and 6 are semi-controllable. Its locomotion is simple and can be described as the set of following actions in order (see Figure 2): (i) "knee" shortening, (ii) moving the leg in front by hip flexion, (iii) extending the "knee" to hit the floor, and (iv) moving the leg backward by extending the hip. The stability of the bipedal robot is based on an unstable system. The robot is stabilized when it moves in steps, not by controlling its upward position. Finally, the upper body is reserved for external modules that can be added in the future. The aim is to provide a generic host, flexible to meet the users' needs, and one that is able to stand, walk, and recover from pushes.

II. Experimental Procedure

S.A.R.A.H was designed with six controllable joints (three in each leg) and two semi-controllable joints (one in each foot). The controllable (Fig. 1, Green Dots) joints are controlled by two actuators that operate antagonistically and provide 6DoFs, one in each joint. In addition, the joints include one hydraulic mechanism that provides damping, spring force, and braking. The semi-controllable (Fig. 1, Red Dots) joints do not use an actuator but an additional hydraulic mechanism. The controllability that the two hydraulic mechanisms can achieve is based on the combination of leg trajectory, gravity, and timing of power down. The semi-controllable joints provides 6DoFs, three in each foot. The hydraulic mechanisms can be locked when they are not powered; thus, the robot can maintain its posture without consuming energy. This reduces power consumption, especially in the standby mode, because only the "brain" remains operational. The configuration is inspired by ostriches, which are more efficient movers than humans [4], and is a big part of the novelty of S.A.R.A.H.

The actuators are connected to microcontrollers that contain gait pattern generators. The inputs required by the microcontrollers to activate the actuators are when and which leg to move. The when/which information is provided by the RPI3, which runs the machine learning classifiers. The inputs of the classifiers are provided to a 6 DoF inertial measurement unit (IMU), which is attached to the main body, and 8 sliding potentiometers, which are attached onto the hydraulics.

The experimental procedure was started with a treadmill moving backward at a slow speed of around 0.5 km/h (Fig. 2). Then, a user decided when and which leg of the robot must be moved. The experiments lasted 10–15 min, and four users collected data to reduce the effect of bias of any one specific user. The captured data were the inputs to the 14 sensors and the user inputs (when and which leg must be moved).

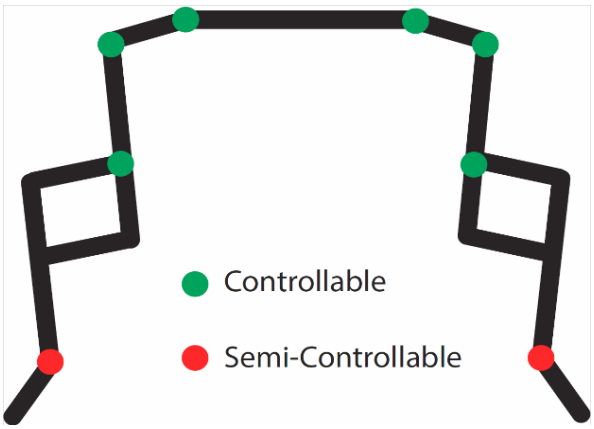

Figure 1. Skeleton of S.A.R.A.H. with degrees of freedom (DoF).

Figure 2. S.A.R.A.H. standing on a treadmill, supported by an elastic band.

A. Data Captured

The 14 inputs captured can be divided in two main groups, namely, high frequency (IMU data) and low frequency (sliders' data in roll of angular measurements). The recorded data are raw values from the sensors, without filtering or bias correction. This was done to capture as much information as possible.

A sample of the acceleration and gyro rates of the IMU data is shown in Fig. 3. From the graphs, it can be clearly determined when a step was made. However, it is harder to determine with which leg the step was made. A sample of the low-frequency data is shown in Figure 4. In these data, it is clearer which leg is moving. However, in comparison with the IMU data, these data show a delay in response. This presents the need for both types of data to achieve effective classification of the user's inputs. The last data that were captured were the user inputs, and a class was created with three labels, "Left Leg Key," "No Key Pressed," and "Right Leg Key."

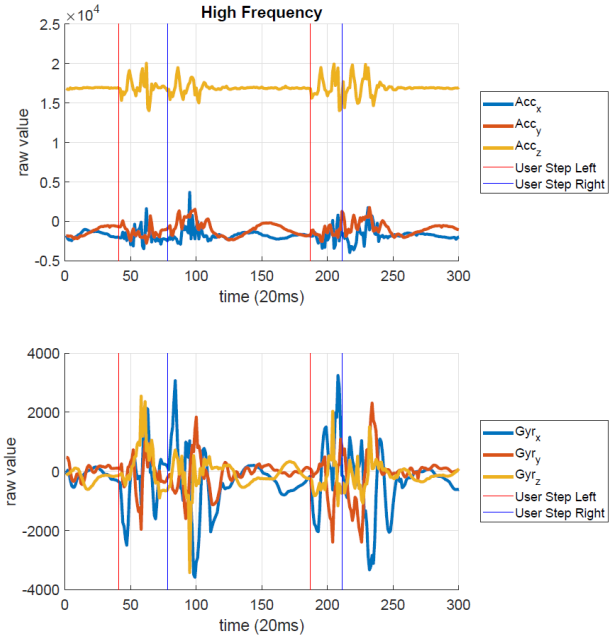

Figure 3. High-frequency signals, IMU.

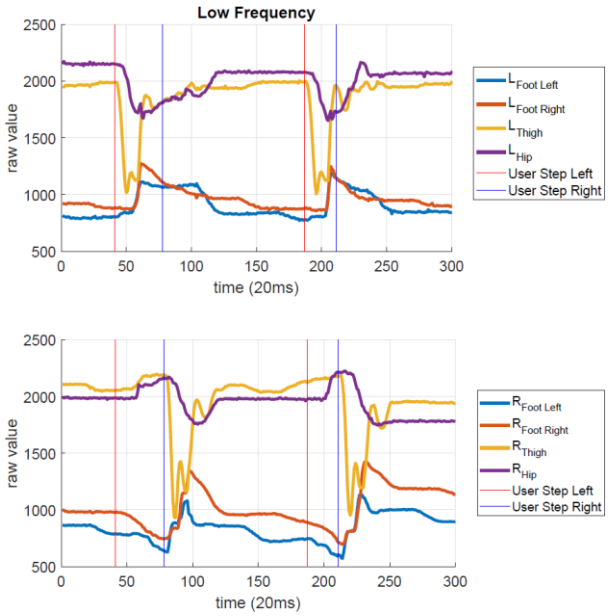

Figure 4. Low-frequency signals, sliders.

III. Classification

The captured data were used to train a classifier that achieved the optimal performance without being too complex. The classifier required 14 inputs and a single, three-labeled, classification (user input). To make a prediction, the "Left Leg Key" and "Right Leg Key" labels were copied 10 times before the actual key was pressed. Each label was assigned a number: -1 to "Left Leg Key," 0 to "No Key Pressed," and 1 to "Right Leg Key."

After training, the classifier was used in an on-board processor to make decisions online alongside the user. The processor on the robot was a low-power Raspberry Pi 3, which would take a long time to execute a complex classifier and make predictions. Additionally, the time between the predictions includes the time required for data capture, data forming, and data pre-processing, as may be required. To minimize the prediction time, the raw data from the sensors were preferable from the viewpoint of eliminating the pre-processing time. The prediction must be quicker than a human's average reaction time, and ideally, it must be half that time.

A total of 20 classifiers were analyzed:

- Decision Tree classification was set up with three different settings. The models were Simple (4 splits), Medium (20 splits), and Complex (100 splits) (All with Gini's diversity index, no surrogate decision splits).
- The discriminant classifier was split into two, Linear and Quadratic (Both with Full covariance structure).
- The SVM classifier was divided into 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 constraint level = 1).

- The KNN classifier was divided into six different models: Weighted (10 neighbors, Euclidean distance, squared inverse weights), Cubic (10 neighbors, Minkowski distance, equal weights), Cosine (10 Neighbors, cosine distance, equal weights), Coarse Gaussian (100 neighbors, Euclidean distance, equal weights), Medium Gaussian (10 neighbors, Euclidean distance, equal weights), and Fine Gaussian (1 neighbor, Euclidean distance, equal weights).
- The NN was categorized into 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 (with no change in cost value) and 100 external epochs (with change in the cost value). The loss function was set to categorical cross entropy and the optimization method to Adam.
- LSTM followed the same structure as that of NN, both in design and training, but 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

Mechanically, the robot weighed 45 kg, of which the batteries, hydraulics, and actuators accounted for 50%; skeleton, 25%; and outer shell, 25%. The maximum stride rate of the robot was 140 steps/min, which is comparable to that of humans [6]. However, the total stride length was just 5 cm which made the robot run at around 0.5 km/h. The speed can increase linearly with increasing stride length.

Computationally, the classifiers were trained with all raw sensor data (14 inputs - 14 features) as inputs and one class as the output: classification. A few 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 no physical meaning.

Additionally, with post-processing and human heuristics, a set of features with physical meaning was extracted, for example, left foot front, right foot front, and torso lean front or back.

Each feature represented an individual discrete physical position. These features were used individually to train the same classifiers for predicting class. However, the accuracy of the results was at least 10% less than the ones obtained using the raw data; thus, they were omitted from the paper.

Performance measurement was not straightforward because if the classifier classified everything as "No key pressed," it achieved an accuracy of 81.9%. To eliminate this, after training, all models (except NN) were cross-referenced in order to spread mislabeled data across all labels. The numerical results of classifiers' accuracy are summarized in Table I.

As can be inferred from Table I, the classifiers without memory could not achieve more than 93% accuracy. Because a continuous result of 0 will lead to an accuracy of 81.9%, the accuracy of 93% was actually 60%. The classifier had to predict the other 18% to achieve 100%, but the accuracy of 93% represented an improvement of 11% compared to 18%. A dynamic problem, such as walking, cannot be described using models without memory, and this is confirmed from the results. In the case of NN, the simple stack of layers with neurons performed the worst because the pure postures could not be translated into predictions of leg steps. However, replacing a few NN layers with LSTM layers improved the results, and the resulting classifier outperformed the other classifiers.

TABLE I. Results of 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%	-	-	-

The LTSM classifiers achieved an accuracy of 98.3%, which is an improvement of 16.4% out of 18.1%; 90.6% actual improvement relative to that achieved with a continuous 0 response. An analysis of the performance of

LSTM showed that 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 was proportional to the increase in performance. The best model was the Deep LSTM, and it comprised 227,303 parameters, almost double of number of parameters in the second-best model (Medium LSTM), which had 136,803 parameters. Small LSTM and Big LSTM had 35,903 and 58,653 parameters, respectively.

In terms of moving a step forward, the best classifier (LSTM) was implemented on one of the RPi3 units hosted in the robot. Then, the experiments were run again, but instead of capturing the data, live prediction of the movements was printed on a screen. The results were impressive, with the predictor being able to produce 20–30 predictions per second (with the Medium model), which is faster than a human's reaction time [19]–[21]. Additionally, the predictions were correct, and most of them were slightly faster than the user input.

V. Conclusion

In the current study, a comparison among various static and dynamic, classification methods of step decision was made for a custom novel, bipedal robot S.A.R.A.H. The robot combines a bioinspired mechanical design with pattern generators to maintain its stability. The robot is inherently unstable, and the classifiers serve as a novel alternative to the classical control theory. They are suitable for the two-stage on/off activation in the pattern generators. The classifiers must learn the decision of when and which leg to move in a bipedal robot by using humans' experience. The algorithm must be run on-board on a RPi3, thus necessitating the use of classifiers with low computational complexity. The bipedal robot used herein is described in our previous conference paper in UK's Robotics and Autonomous Systems Conference [3].

Static classifiers did not perform well on this problem, with the weighted KNN model performing the best out of all, yielding a maximum accuracy of 92.8%, which represents an actual accuracy of 60% in real predictions. Moreover, humans cannot heuristically define features to improve the performance of the classifiers. Simple NNs did not perform well because postures cannot describe the time of leg movement, albeit they may provide information about which leg to move, which is inadequate. The treadmill did not operate at a constant speed, which increased the dynamicity of the problem with no measurable information, such as walking speed and acceleration. Extraction of this information requires the classifiers to have memory of past states.

Dynamic classifiers have memory that changes with time. The tested dynamic classifier was a NN with LSTM layers. It was found that as the number of neurons in each layer, instead of the number of layers, increased, prediction performance increased. The highest accuracy achieved was 98.3% (90.6% actual accuracy) with the Deep model, but the running time on-board was 50% higher than that time of the Medium model, which yielded an accuracy of 97.7% (87.3% actual accuracy). Thus, the Medium model was preferred and implemented on-board, side by side with the user. The prediction rate achieved was 20–30 predictions per second; these predictions were correct and, sometimes, faster than the user.

VI. Future Work

The next steps are to let the classifier control the legs and walk at different speeds. To achieve this, a greater number of user sequences of 10–15 min must be used to train the networks to validate the classifier and make it more robust. Additionally, the dimensionality of the inputs will be reduced to improve execution time on the on-board computer without compromising accuracy.

Acknowledgment

The project is partially funded from Innovate UK's scheme "Emerging and Enabling Technologies" and Center of Doctoral Training of Embedded Intelligence (CDT-EI) funded from "Engineering and Physical Sciences Research Council" 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, pp. 107–119, 1984.

[2] K. A. 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. 25, pp. 227–245, 2017.

[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 Working for and Among Us*, In press.

[4] C. R. Taylor, N. C. Heglund, G. M. Maloiy, "Energetics and mechanics of terrestrial locomotion. I. Metabolic energy consumption as a function of speed and body size in birds and mammals," *The Journal of Experimental Biology*, vol. 97, pp. 1–21, 1982.

[5] J. T. Choi and A. J. Bastian, "Adaptation reveals independent control networks for human walking," *Nature Neuroscience*, vol. 10, pp. 1055–1062, 2007.

[6] J. Nielsen, "How we walk: Central control of muscle activity during human walking," *The Neuroscientist*, vol. 9, pp. 195–204, 2003.

[7] Motion Robotics LTD, https://www.motion-robotics.co.uk/

[8] Microchip, A.: SAM C21 Xplained Pro Evaluation Kit (2017), [Online]. Available:http://www.atm–el.com/tools/ATSAMC21-XPRO.aspx

[9] Raspberry Pi: Raspberry Pi 3 Model B (2017), https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[10] 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," *IEEM2010 - IEEE International Conference on Industrial Engineering and Engineering Management*, no. Figures 1, pp. 376–379, 2010.

[11] K. Ogata, K. T. K. Terada, and Y. K. Y. Kuniyoshi, "Falling motion control for humanoid robots while walking," *2007 7th IEEE-RAS Inter-national Conference on Humanoid Robots*, pp. 306–311, 2007.

[12] J. J. Kim, T. Y. Choi, and J. J. Lee, "Falling avoidance of biped robot using state classification," in *Proc. 2008 IEEE International Conference on Mechatronics and Automation*, ICMA 2008, pp. 72–76, 2008.

[13] 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, pp. 893–898, 2010.
[14] 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, pp. 763–768, 2011.
[15] 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.
[16] S. F. Rashidi, M. R. S. Noorani, M. Shoaran, 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.
[17] Y. Feng, Y. Li, and J. Luo, "Learning effective Gait features using LSTM," *Proceedings - International Conference on Pattern Recognition*, pp. 325–330, 2017.
[18] D. N. T. How, K. S. M. Sahari, H. Yuhuang, and L. C. 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.
[19] 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, pp. 73–77, 2017.
[20] J. E. Birren and J. Botwinick, "Age differences in finger, jaw, and foot reaction time to auditory stimuli," *Journal of Gerontology*, vol. 10, pp. 429–432, 1955.
[21] G. R. Grice, R. Nullmeyer, and V. A. Spiker, "Human reaction time: toward a general theory," *Journal of Experimental Psychology: General*, vol. 111, pp. 135–153, 1982.

Christos Kouppas is a PhD student in Computer Science at Loughborough University in United Kingdom. He obtained BEng in Mechanical Engineering from University of Cyprus, graduating first of his class in 2015. He also graduated from The University of Sheffield first of his class at the MSc Advanced Control & Systems Engineering in 2016. He has been awarded with Laverick Webster Hewitt Prize for his outstanding performance and The Eric Rose Prize for the best project, during his master studies. Additionally, he is a senior engineer in Motion Robotics Ltd, UK.

Dr Qinggang Meng is a Reader in robotics and autonomous systems at Loughborough University, UK. His main research interests and expertise include: cognitive robotics, multi-robot/UAV cooperation, AI, machine learning and computer vision, driverless vehicles, human-robot interaction, and ambient assisted living.

Dr Mark King is a Reader in Sports Biomechanics at Loughborough University, UK specializing in using subject-specific computer simulation models to understand optimum performance and injury risk in sport. Integral to this work is the role of muscle and technique on optimum performance and how force gets transmitted / energy dissipated through the body. He has been at Loughborough since 1990 graduating in Sport Science and Mathematics in 1993 and obtained his PhD in computer simulation of dynamic jumps in 1998.

Dr Dennis Majoe is Project Coordinator and Senior Scientist in the ETH Zurich Institute for Computer Systems. He is also CTO for Motion Robotics Ltd, U.K., His main area of research is robotics as applied to assistive living.