DOI 10.31256/UKRAS19.5

Controlling a Bipedal Robot with Pattern Generators Trained with Reinforcement Learning*

Christos Kouppas, Qinggang Meng, Mark King and Dennis Majoe

Abstract — **Herein, the use of reinforcement learning and pattern generators for balancing a bipedal robot, is described. SARAH (Silent Agile Robust Autonomous Host) is an underactuated robot designed by Motion Robotics LTD and aims to become an everyday bipedal robot that has fast, humanlike response. By utilizing V-Rep simulator, a simulated model of the robot was constructed and controlled with pattern generators. Then, those pattern generators were optimized by using reinforcement learning and a neutral advantage function agent. The training results are presented through graphs with respect to training steps, to show how the parameters converge to the optimum values.**

I. Introduction

Bipedal robots are becoming more sophisticated over the years however, their structural design remains the same. The majority of bipedal robots demonstrate a human-like mechanical structure which was established from the early 70s [1]. A full humanoid, like ASIMO of Honda [2] and ATLAS of Boston Dynamics [3], usually consists of at least 23 Degrees of Freedom (DoF) that can be categorized as: 3 for the head, 3 for each shoulder joint, 1 for each elbow joint, 2 for each hip joint, 1 for each knee joint and 3 for each foot joint. They can further be grouped into the upper part (head, shoulders, elbow – 11 DoFs) and the lower part (hip, knee, foot – 12 DoFs).

Our research focuses on the lower part of the bipedal robot because it is more challenging compared with the upper part which has similar structure as industrial robots and is well studied and optimized. In this paper, we designed an underactuated bipedal host which has 6 actuators and 10 DoFs [4]. The control of the robot was achieved by combining Central Pattern Generators (CPG) and Reinforced Learning (RL). As it was demonstrated, CPG are used by humans [5] and they are suitable for bipedal robots [6]. Reinforced Learning on the other hand, proved suitable for real-time applications [7].

The robot model was implemented in V-Rep Simulator [8] and the basic characteristics of the simulation were evaluated with the physical robot. Combining the simulator with reinforced learning, two set of parameters of the robot were optimized. The first set was to optimize to increase the number of steps per minute, in respect of the overall movement in the transverse plane. The second set of parameters was to adjust small movements in hip joints to modify flexion/extension and abduction/adduction during steps. These movements are aiming to keep a stable position, in the transverse plane, for as long as possible.

Figure 15: Ankle joints of SARAH with and without shoe.

II. Mechanical Structure

SARAH has four underactuated DoFs, two on each foot. These joints were responsible for the lateral movement of the forefoot and the hindfoot parts. They were free to rotate but limited by the shank, where they were touching on a pressure sensor, and a shoe was holding them from moving downwards thus, they could flex inside the shoe like a human foot. Figure 1 demonstrates the actual ankle joint with and without the shoe.

As shown in Figure 2, the physical model of the developed bipedal robot can be divided into three main groups. The first includes the main top compartment with the right and left hips, as well as four actuators for the adduction/abduction and flexion/extension of the robot. The other two main groups are the two legs, each one having one actuator ("knee") and two passive joints on the foot. The total weight of the robot was 30kg, with each group weighing 10kg. These specifications were also considered in the simulated model in order to achieve realistic dynamics.

III. Controlling Architecture

Balancing was provided from a CPG using information from an Inertial Measurement Unit (IMU) that was fixed in the

*Research was partially founded from EPSRC, funding from the Engineering and Physical Science Research Council Centre for Doctoral Training in Embedded Intelligence (grant no. EP/L014998/1).

Christos Kouppas is with Loughborough University, Loughborough, LE11 3TU, UK (corresponding author, e-mail: c.kouppas@lboro.ac.uk)

Qinggang Meng and Mark King are with Loughborough University, Loughborough, LE11 3TU, UK (e-mails: Q.Meng@lboro.ac.uk, M.A.King@lboro.ac.uk)

Dennis Majoe is with Motion Robotics LTD, Southampton, SO30 3DS, UK (e-mail: dennis.majoe@motion-robotics.co.uk)

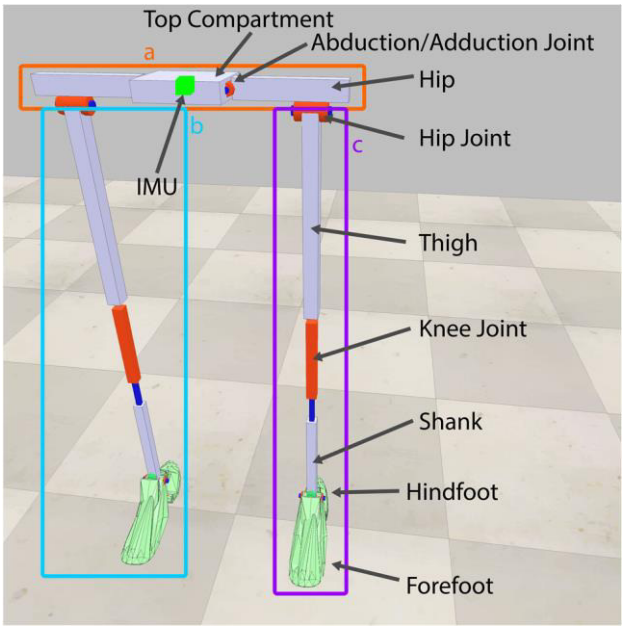

Figure 16: SARAH's simulated model in V-Rep with annotations. (a) top groups, (b) right leg, (c) left leg

center of the top compartment. From that, 6 axis IMU, only 4 pieces of information were used as the angular rate on the x and z axes were not necessary (for this stage of balancing). The Y-axis angular rate shows the speed at which the robot will jump from one leg to the other. From the x-y-z axes accelerations, a quantitively postural angle of the robot can be determined by dividing the correspondent (to the angle that was examined) axis with the total acceleration. Additionally, the four pressure sensors that are located on the ankle, were giving information if the forefoot and the hindfoot were on the ground. With the addition of the feedback from the actuators (six signals), the observations from the simulation sum up to 16. Figure 3 shows the complete CPG schematic with the equations for each condition and the actions that take place after each condition.

Additionally, in Figure 3 two sets of variables are noted, the P1-5 and the V1-4. These variables were extracted after two consecutive reinforced learning trainings using keras-rl library [9] and the normalized advantage function (NAF Agent) [10]. Under this agent, the calculation of the Q-value variant took place in a continuous form with experience replays. Model-free reinforcement learning with continuous outputs, uses raw inputs from the system (e.g. raw sensor data) and outputs a float number as a result in all the outputs. Those outputs can be used raw, in the inputs of the real/simulated system. The reason for two consecutive trainings and not one is based on the future exploration of the balancing problem. The first training was focusing more on the response speed than the stability and the second training was the opposite.

The models that were used as actor (mu_model), critic (V_model) and Q-maximizer model (L_model) were simple neural networks and the interaction between them achieved a complex non-linear result. The V_model had three layers of neurons with each layer having neurons equal to the square of the number of the observation signals (256). The L_model had four layers of 5 times the number of the observation signal (80) and the mu_model had four layers too, but with the number of observation signals (16). The layers had a sigmoid as an activation function and the output layer of each model had a linear activation to rectify the decisions in a continues space. Because, heuristically, it was observed that the parameters must be positive, the actor's rectification bias was initialized at 1, instead of 0.

The main difference between the reinforced learning and the supervised learning is the way that the data were collected, as the first one is collecting the data during training while the latter is using a pre-collected dataset. Another difference is that, reinforced learning needs to define a cost (if it is negative) or a reward (if it is positive) function based on the performance of the robot in the simulator or real world. This function will act as the "correct option" and the reinforced learning will maximize it. For the training of SARAH's model, the equation (2) was used as a reward function. Its value was calculated in every step of the simulation and their summation wes presented as a reward in the end of each simulation.

$$R = (0.5 - c_x) \cdot (0.5 - c_y) \cdot 2^{-(f-3)^4} \cdot c_z \qquad (2)$$

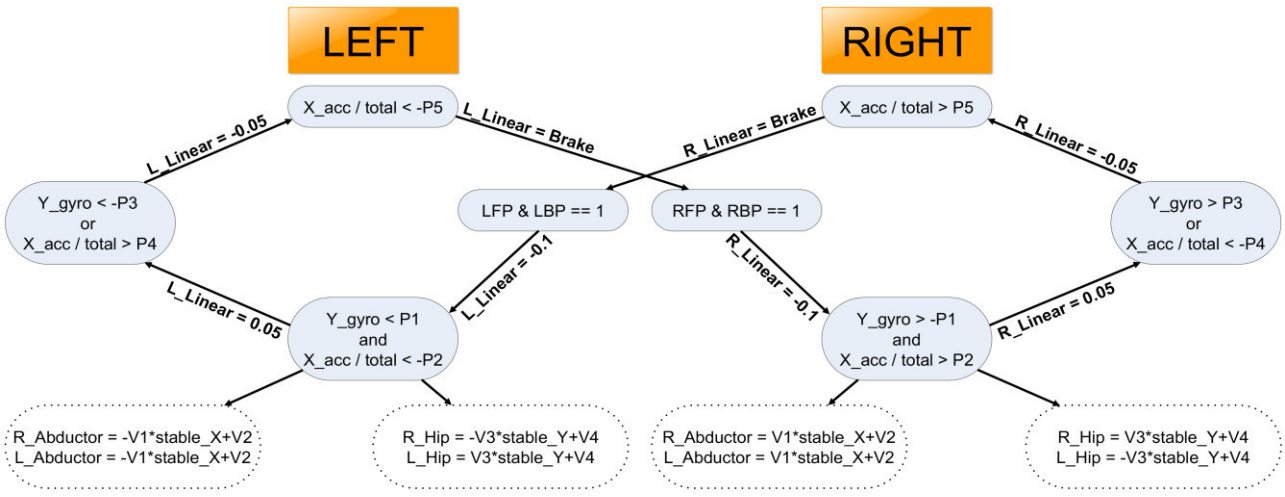

Figure 17: Central Pattern Generator for balancing the bipedal robot. P1-5 were the parameters that were trained during the first training. V1-4 were the parameters that were trained during the second training. Colored rectangles represent condition's statements and colorless (or having =) represents actions. LFP, LBP, RFP and RBP are the forefoot and hindfoot pressure sensors of the left and right foot, respectively. stable_Y and stable_X are the lowpass accelerations in y and x axes. total is the squared acceleration in the three axes.

where, *R* is the reward, *f* is the number of steps per second a*nd* $c_x/c_y/c_z$ is the position of the top compartment in 3 dimensions x-y-z, respectively.

First Training

During the first training the model was initialized standing upwards and stable. After one second, the CPG started moving the legs based on the parameters P1-5, which were the trained parameters. Meanwhile the movement of abduction/ adduction and extension/flexion was locked, requiring no training. The parameters were set in the beginning of each simulation, once by the outputs of the neural network and were not changed during the simulation. The aim of the first training was to find the best parameters for the CPG in order to have similar steps per second as a human (3-5 steps per second) but without sacrificing a lot of the planar stability.

The output of each model was 5, 1 and 15 for the actor, critic and Q-maximizer, respectively. The Q-maximizer outputs formed a 5x5 lower triangular matrix (L) and it was used for calculating the Q-value of the network based on the equations (3). The number of the outputs matches the number of parameters that must be trained.

$$Q_{(i)} = V_{(i)} - \frac{1}{2}(u - M_{(i)})(L \cdot L^T)(u - M_{(i)})^T \qquad (3)$$

where, Q(i) is the Q-value, V(i) is the output from V_model, u is the predicted actions with the addition of a random exploration value [1x5 matrix], M(i) is the predicted action [1x5 matrix] and L is the outputs of the L_model in a lower triangular matrix like equation (4).

$$L_{(i)} = \begin{matrix} L1 & 0 & 0 & 0 & 0 \\ L2 & L3 & 0 & 0 & 0 \\ L4 & L5 & L6 & 0 & 0 \\ L7 & L8 & L9 & L10 & 0 \\ L11 & L12 & L13 & L14 & L15 \end{matrix} \qquad (4)$$

Second Training

The second training was similar to the first one, as it was starting with the robot standing stable for one second. However, after that point, a random planar force was acted on the top compartment and had an amplitude up to 100 N. This force was displacing the robot by a few centimeters and during the steps, the robot was trained to return to its initial position by changing the parameters V1-4 in the CPG.

The network was the same as with the first training, except that the outputs of the actor and Q-maximizer were 4 and 10, respectively. The reward function stayed the same as the amount of step per second was not changing drastically from V1-4 parameters. Also, both trainings were stopped under three criterions. First, criterion was that the simulation will not stop after 30 second even if they performed well. The other criterion was a virtual 3D limit of movements by 25 cm. When those limits were reached, the simulation was stopped. Last criterion was if the parameters did not make the robot oscillate in the first 5 seconds, then again, the simulation was stopped. The criterions were implemented thus, the simulations that will produce low reward, will end sooner and reserve resources.

IV. Results And Discussion

First Training

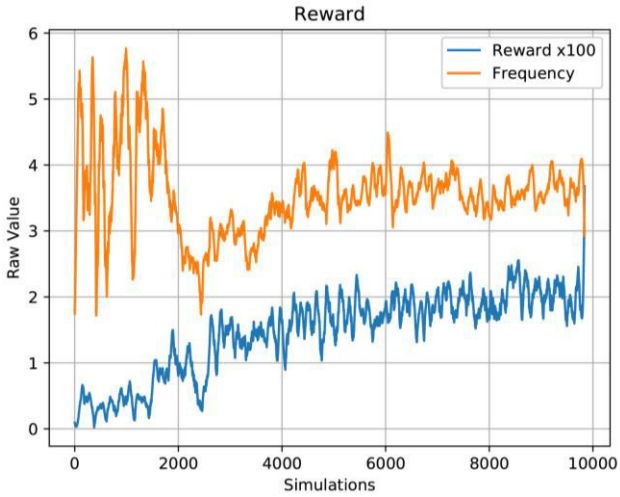

Figure 18: Reward and Frequency during First Training.

Figure 4 shows the performance of the NAF agent finding the parameters P1-5 to balance SARAH in V-Rep simulator. The objective of the training was to increase the number of steps per second without sacrificing the stability of the robot. It was observed that all parameters, P1-5, must be positive in order to have a stable and continues response. If the parameters were negative, small changes resulted in unexpected results.

Examining the parameters and how they changed in respect with the performance, the key parameters were determined. Figure 5 demonstrates the changes of parameters P1-P5 during training. Figure 3 shows that, the parameters P1-P3 were responsible mostly for the performance of the simulation, as the other two parameters were increasing rapidly but the reward was not. Those results were confirmed by manually varying those variables, after the training and showed that they do not change the reward/performance proportionally. Those parameters control the timing of each step so, if they were too big, the step cycle never finishes and if they were to small, the step finishes abnormally fast. Alternating the feet on the floor, was making the robot rotate in Y-axis and the faster the steps, the bigger the rotation speed. The parameters P1 and P3 were responsible to limit this rotation and they stabilized around 0.4

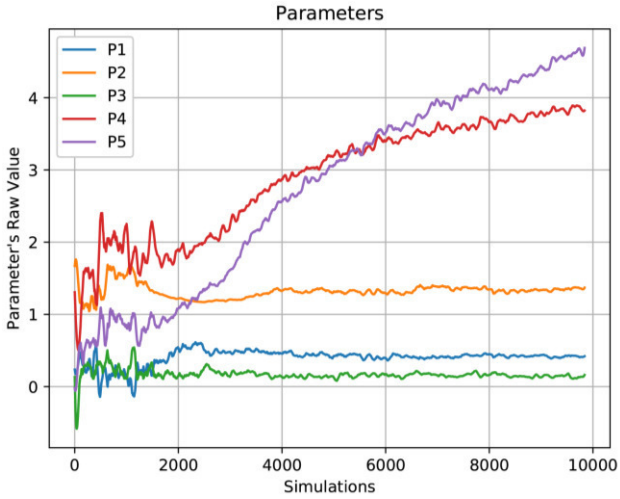

Figure 19: Parameter (P1-5) results during First Training.

and 0.1, respectively. The variable P2 was responsible for the amount of tilt, sidewise, and was stabilized at 1.3 which can be translated to ~7.5°.

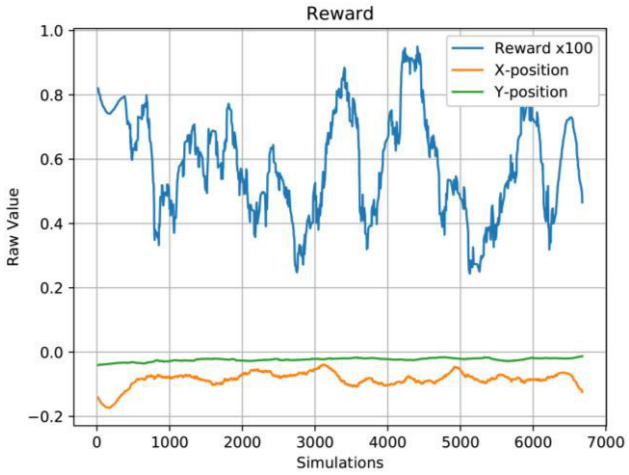

Figure 20: Reward and X-Y final position of the top compartment during Second Training.

Second Training

During the second training, the parameters V1-4 were trained to minimize the movements in the planar plane. Those variables were adjusting the angles for the abduction/adduction and flexion/extension based on X and Y axes accelerations, respectively. Figure 6 demonstrates the performance of the simulation and the movements in X-Y axes as the training got trained. The performance, as reward, was based on the same equation as before (equation (2)). However, because the steps per second were limited by the parameters P1-5, the reward was controlled mainly by the position of the top compartment.

Figure 7 presents the parameters V1-4 and it is noteworthy that, the variables V1 and V3 were more important than V2 and V4. The important variables were multiplying the acceleration of the X-axis and Y-axis, respectively, since they were changing the sensitivity of the respond as they were linear to the response. The variables V2 and V4, were responsible for the constant value that may be needed to keep the center of mass in the center of the robot. The randomness of the force added a great difficulty in the training algorithm. The variables

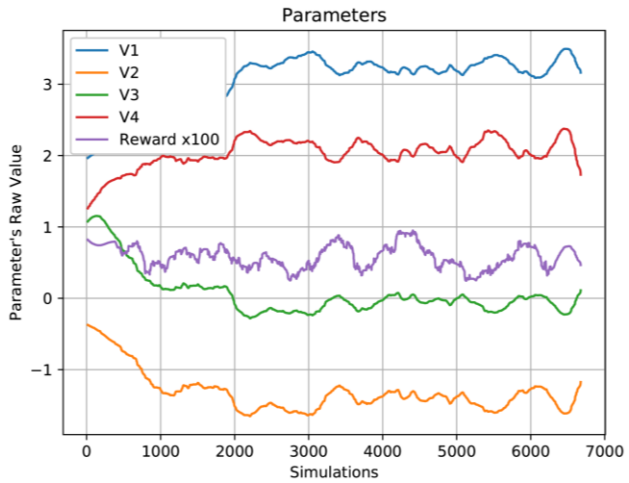

Figure 21: Parameter (V1-4) results during Second Training.

did not show a logical relation with the reward as V1 was following V4 trends and V2 was following V3 trends.

V. Conclusion

The use of neural networks with CPG in bipedal robots for balancing is not new [11], however the use of NN to optimize values of a CPG is novel. Utilizing reinforcement learning, the network is optimized to provide a parameter set solution that will replace certain variables in the CPG in order to balance SARAH. Reinforcement learning with NAF agent, offers exploration of the parameters and the ability of learning through experience.

The CPG used to drive the robot, works independently from the NN. The CPG increases the response rate of the system, as they can be executed faster than a neural network as each cycle is executed in few processor cycles but NN needs few cycles for each layer. However, the use of CPG with NN that we propose, includes the ability of learning from the robot as the NN can be trained offline from events while the robot was used. Afterwards, using NN trained classifiers, new CPG parameters can be pushed through and update movements appropriately based on real time sensors.

Future work of this project includes, the finalization of SARAH and to integrate the balance with this technique as well as to train it from its own experience, online. Finally, the algorithm will be tested in different terrains, with different slopes, friction or texture.

Acknowledgment

The authors would like to thank Motion Robotics LTD and all the employees for the assistant that they provided for this project. Also, the reviewers for their time and helpful feedback.

References

[1] S. Behnke and S. Behnke, "Humanoid Robots - From Fiction to Reality?," *KI-Zeitschrift*, vol. 4, no. December, pp. 5–9, 2008.

[2] American Honda Motor Co. Inc., "ASIMO Specifications | ASIMO Innovations by Honda," 2018. [Online]. Available: asimo.honda.com/asimo-specs/.

[3] Boston Dynamics, "ATLAS - The World's Most Dynamic Humanoid," 2018. [Online]. Available: www.bostondynamics.com/atlas. [Accessed: 20-Jun-2018].

[4] C. Kouppas, Q. Meng, M. King, and D. Majoe, "S.A.R.A.H.: The Bipedal Robot with Machine Learning Step Decision Making," *Int. J. Mech. Eng. Robot. Res.*, vol. 7, no. 4, 2018.

[5] J. B. Nielsen, "How we Walk: Central Control of Muscle Activity during Human Walking," *Neurosci.*, vol. 9, no. 3, pp. 195–204, Jun. 2003.

[6] S. Kolathaya and A. D. Ames, "Achieving bipedal locomotion on rough terrain through human-inspired control," *2012 IEEE Int. Symp. Safety, Secur. Rescue Robot. SSRR 2012*, 2012.

[7] K. Doya, "Reinforcement Learning In Continuous Time and Space," vol. 245, pp. 1–28, 1999.

[8] Coppelia Robotics GmbH, "V-Rep Pro Educational," 2018. [Online]. Available: http://www.coppeliarobotics.com/. [Accessed: 12-Jul-2018].

[9] M. Plappert, "keras-rl," *GitHub Repos.*, 2016.

[10] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous Deep Q-Learning with Model-based Acceleration," Mar. 2016.

[11] 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 Int. Conf. Robot. Mechatronics, ICRoM 2014*, pp. 852–857, 2014.