University of Hertfordshire UH

Biped robot walking – Reinforcement Learning

Literature Review

Roberto Figueiredo
SRN: 17054162

University of Hertfordshire UH

One of the most critical and challenging aspects of robotics is locomotion. In the past, robotics locomotion has been mainly focused on wheel-based, which can be limiting in challenging, unpredictable environments. Walking has come to attention in later years and is one of the biggest challenges in the RoboCup competition where the team Boldhearts competes. Our humanoid robots have regular updates on their structure due to improvements or frequent rule changes by the competition, making it hard to develop a stable walking algorithm. When compared to wheel-based, walking can be very challenging because of having to deal with naturally unstable systems. I will attempt to target this problem by applying reinforcement learning techniques to simplify future training. This literature review will cover the choice of algorithm, where I will compare two of the major algorithms by studying previous implementations and documentation. I will cover the simulation and training environment decision, the newly open to the public Mujoco. I will cover findings on engineering and action selection systems by analysing previous attempts, what difficulties were faced, and how they were overcome.

Gait

The first step to implementing a walking algorithm should be understanding its movement as it is one of the most complex movements performed by humans.
Biped robots are more agile and have better mobility than wheeled robots, this shows especially in uneven, rough terrain, the ability to overcome obstacles such as stairs.
"A gait is a cyclic, periodic motion of the joints of a legged robot, requiring the sequencing or coordination of the legs to obtain reliable locomotion. In other words, gait is the temporal and spatial relationship between all the moving parts of a legged robot. Gait generation is very important for legged robots because it determines the optimal position, velocity and acceleration for each Degree of Freedom (DOF) at any moment in time, and the gait pattern will directly affect the robot's dynamic stabilisation, harmony, energy dissipation and so on. Gait optimization determines a legged robot's quality of movement." (Gong & Zuo, 2010)

On a first draft, findings were pointing to the general stability of the body, leading to believing that maintaining the centre of mass stable would be crucial to developing this algorithm. Many attempts to keep the robot statically stable will lead to unnatural walking patterns, this is because humans walk dynamically; "when we walk, we assume that future movements of our body will prevent us from falling over" (Thompson & George, 2016). Humans will lean forward using gravity to maximize walking efficiency, instead of just relying on muscle movement. If there was a sudden stop to joint movement we would fall over, this leads to a lot of walking implementations following a less natural although more stable gait. These findings will lead to a different implementation of the reward function as the balance can still have an impact it shouldn't be one of the main features as it would impede the ability of the robot to learn to walk dynamically. Instead, we want to discourage instability. (Huang, et al., 2001)

Reinforcement learning algorithms

Roberto Figueiredo
SRN: 17054162

"Reinforcement learning has demonstrated remarkable progress, achieving high levels of performance in Atari games, 3D navigation tasks, and board games" (Brockman, et al., 2016) With new methods of reinforcement learning emerging and a lot of research starting to develop in the area of biped walking, it highlights as one of the best ways to approach this problem. There are multiple algorithms to choose from, as summarised by the following graph:

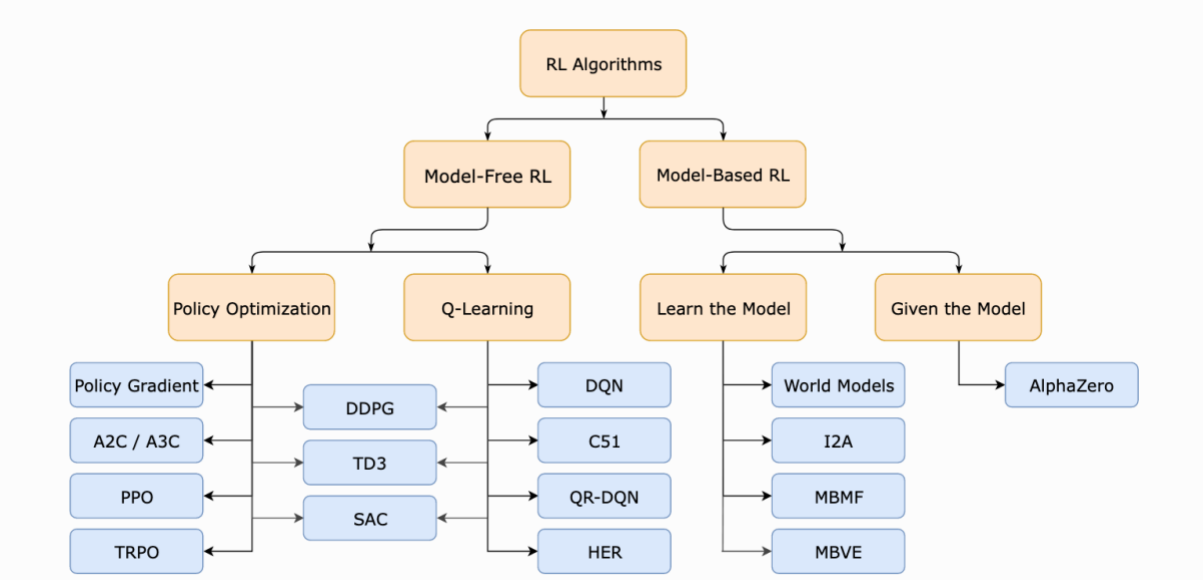

Figure 1: A non-exhaustive, but useful taxonomy of algorithms in modern RL (Openai, 2021)

The first decision would be between model-free and model-based, which most of the time is a straightforward decision as Model-learning is fundamentally hard, even with high resources, and time it can fail to produce good enough results. On Model-Free Reinforcement learning, there are two options, Policy Optimization and Q-Learning being the two most popular, Deep Q Networks and Policy Gradients.

The main objective of Q-learning is to find a state-action pair that will return the most reward, essentially bringing the agent closer to the goal, this is made by incorporating both past and present data. "Among reinforcement techniques, Q-learning (Watkins, 1989, Watkins and Dayan, 1992) has been especially well-studied and possesses a firm foundation in the theory of Markov decision processes" (Wang, 2020); Deep Q learning is an improved version of Q-Learning with a deep convolutional neural network. While both Deep Q learning and Policy gradients are driven on the Markov Decision Process, the main difference is that policy gradient methods target at modelling and optimising the policy directly, which can be stochastic while, Deep Q Learning learns a single deterministic combination of state-action pair.

Deep Q learning was preferred over Policy Gradient due to the larger amount of documentation and community, although, both will be tested to determine the better solution for the problem in study as both have trade-offs, "the primary strength of policy optimisation methods is that they are principled, in the sense that *you directly optimise for the thing you want.* This tends to make them stable and reliable. By contrast, Q-learning methods only *indirectly* optimise for agent performance by training to satisfy a self-consistency equation. There are many failure modes for this kind of learning, so it tends to be less stable.

But Q-learning methods gain the advantage of being substantially more sample efficient when they do work, because they can reuse data more effectively than policy optimisation techniques." (Openai, 2021) (Hu & Wellman, 2003)

Simulation-training solutions

The algorithm training will happen in a virtual environment; this decision is based on the fact that "current algorithms often require a large number of samples—on the order of tens of thousands of trials. "Moreover, such algorithms are often highly sensitive to hyperparameter settings and require considerable tuning, further increasing the overall sample complexity" (Haarnoia, et al., 2019) For this reason, many prior methods have studied the learning of locomotion gaits in simulation.

The simulation environment used by the team Boldhearts is Gazebo, although, I decided to use the recently opened to the public, and vastly used, Mujoco. (Tassa, et al., 2021) The main reason I decided to go with Mujoco, apart from being interested in developing knowledge on this tool, was due to its integration with OpenAI Gym, a toolkit used for developing reinforcement learning algorithms, "OpenAI Gym focuses on the episodic setting of reinforcement learning. In each episode, the agent's initial state is randomly sampled from a distribution, and the interaction proceeds until the environment reaches a terminal state. The goal in episodic reinforcement learning is to maximise the expectation of total reward per episode and to achieve a high level of performance in as few episodes as possible" (Brockman, et al., 2016) OpenAI is regularly used in conjunction with Mujoco, therefore, there is a vast amount of documentation available and a large community. (Arango, 2018)The Mujoco engine has also been praised for its simulation capabilities and realism, likely resulting in a smoother transition from simulation to real-world robots.

The algorithm will be trained using TensorFlow which offers multiple levels of abstraction giving me the ability to choose the most appropriate at each level of the project. TensorFlow also has the ability to use the high-level Keras API, which makes getting started with TensorFlow and machine learning easy.
(Tensorflow, 2021) (Heess, et al., 2017)

Reward system

To update the neural network, a reward must be calculated. From the research of past applications of similar techniques to other walking algorithms, I found that applicating a tiered reward system could be beneficial to give more importance over others at specific moments, therefore, creating a feature hierarchy. Reward system engineering is a very hard process. The first and most obvious features to impact the reward attribution would be the robot falling and distance moved. When looked further though, the structure of features stands as follows: In feature hierarchy, the robot would be penalised or rewarded depending on its angle towards the goal, this would make sure the first movement would make the robot align to the objective, afterwards we have a movement towards the goal feature. Some of the less obvious features are the centre of mass, which should have a low impact on rewarding to allow for dynamical walking, gradual and smoothness reward, which calculates the number of changes in directions from joints, preferring a smoother movement. Another feature with interest would be time, decreasing as time passes, encouraging the robot to choose a sequence of actions that brings it to the goal faster. (Thompson & George, 2016)

Action Selection

On a biped robot such as the ones used by the team Boldhearts, there are about 14 joints with relevance to the walking system, with 10 DOF per joint if we had decided to use discrete joint angles per action, it would lead to 10^{14} possible actions. A more reasonable approach is to use three different actions per joint, increase, decrease and maintain, both increase and decrease, changing the angle of the joint by a defined amount. In order to build a complete action, a good approach is seen in the development of the algorithm for the atlas robot by Alec T. and Nathan G. where, since the action, space is still very large, 3^{14}, we would not be able to access all the possible actions. The technique used in the paper instead passes to the neural network only 42 values, by evaluating the three possible actions one joint at a time.
An example of the process:

$$Joint\ 0: 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00 \rightarrow \begin{array}{ccc} 01 & \rightarrow & q = 0.2 \\ 00 & \rightarrow & q = -0.1 \\ 10 & \rightarrow & q = 0.3 \end{array}$$

$$Joint\ 1: 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 10 \rightarrow \begin{array}{ccc} 01\ 10 & \rightarrow & q = 0.25 \\ 00\ 10 & \rightarrow & q = -0.05 \\ 10\ 10 & \rightarrow & q = -0.15 \end{array}$$

$$\vdots$$

$$Final\ action: 00\ 01\ 01\ 00\ 00\ 10\ 10\ 01\ 10\ 00\ 00\ 10\ 01\ 10$$

Figure 2: Example of iterative construction of an action

Development structure

As seen from past developments, the best way to achieve good results is to split the development into different levels of complexity and iterate on the previous stage.

The first stage will cover the development of a simple agent simulating the well-known cart pole environment, this stage will use OpenAI but omit Mujoco for a simple 2d simulation. This will improve the knowledge and understanding of the process of developing a reinforcement learning algorithm. The second stage of development will apply a high-level approach to a simplified humanoid (full-bodied or partial) this will introduce Mujoco. The process of tunning and developing the reward system can actively take place at this stage; this will be an important step to understand the complexity of the problem and relevant issues such as the performance of training on such a large action space. The third stage would apply the same strategy developed on stage 2 to a realistic model of the humanoid used by the team Boldhearts and further iterations on the reward system. The desired final stage would be to successfully apply the learned algorithm to the real robot.

From the initial problem of developing a reinforcement learning algorithm for biped walking a lot of problems were raised, such as what algorithm to use, how to develop the reward and action selection system and how to run the simulation and train the algorithm. In this literature review, we briefly cover all the research findings answering the questions imposed. The research on algorithms and understanding the processes behind them such as the Markov-Decision Process and the Bellman equation are valuable in understanding the next steps of development and research. Another great insight from analysing past experiments with reinforcement learning algorithms is how unpredictable results can be and why the decision to run tests on both algorithms, Policy gradient and Deep Q Learning. Understanding past experiments, with great insights from the experiment on Atlas, was a great way to find tested features used for rewarding the agent, this has shown behaviours that contradicted the initial ideas for rewarding. The implementation of the iterative action selection system based on a discrete set of actions per joint has also brought great value to improve training performance.

Roberto Figueiredo
SRN: 17054162

University of Hertfordshire UH

References

Arango, M., 2018. *Deep Q-Learning Explained..* [Online]
[Accessed 13 November 2021].
Brockman, G. et al., 2016. *OpenAI Gym.* [Online]
Available at: https://arxiv.org/pdf/1606.01540.pdf
[Accessed 1 November 2021].
Gong, D. & Zuo, G., 2010. *A Review of Gait Optimization Based on Evolutionary Computation.* [Online]
Available at:
https://www.researchgate.net/publication/220449058_A_Review_of_Gait_Optimization_Based_on_Evolutionary_Computation
[Accessed 1 November 2021].
Haarnoia, T. et al., 2019. *Learning to Walk via Deep Reinforcement Learning.* [Online]
Available at: https://arxiv.org/pdf/1812.11103.pdf
[Accessed 1 November 2021].
Heess, N. et al., 2017. *Emergence of Locomotion Behaviours in Rich Environments.* [Online]
Available at: https://arxiv.org/abs/1707.02286
[Accessed 18 November 2021].
Huang, Q. et al., 2001. *Planning Walking Patterns for a Biped Robot.* [Online]
Available at: https://www.cs.cmu.edu/~cga/legs/kuff1a.pdf
[Accessed 1 November 2021].
Hu, J. & Wellman, 2003. *Nash Q-Learning for General-Sum Stochastic Games.* [Online]
Available at: https://www.jmlr.org/papers/volume4/temp/hu03a.pdf
[Accessed 9 November 2021].
Openai, 2021. *Kinds of RL Algorithms — Spinning Up documentation..* [Online]
Available at: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html
[Accessed 14 November 2021].
Tassa, Y. et al., 2021. *Opening up a physics simulator for robotics.* [Online]
Available at: https://deepmind.com/blog/announcements/mujoco
[Accessed 1 November 2021].
Tensorflow, 2021. *Why TensorFlow.* [Online]
Available at: https://www.tensorflow.org/about>
[Accessed 9 November 2021].
Thompson, A. & George, N., 2016. *Deep Q-Learning for Humanoid Walking.* [Online]
Available at: https://web.wpi.edu/Pubs/E-project/Available/E-project-042616-142036/unrestricted/Deep_Q-Learning_for_Humanoid_Walking.pdf
[Accessed 13 November 2021].
Wang, M., 2020. *Deep Q-Learning Tutorial: minDQN.* [Online]
Available at: https://towardsdatascience.com/deep-q-learning-tutorial-mindqn-2a4c855abffc
[Accessed 2021 November 2021].

Roberto Figueiredo
SRN: 17054162