# Informatics Project Proposal:
# Policy distillation for humanoid balancing control

Nick Robinson
s1784599@sms.ed.ac.uk

**Abstract**

We propose using reinforcement learning to train the Valkyrie humanoid robot to balance in 3D simulation. We frame humanoid balancing as a multi-task problem which requires balancing in both the sagittal and lateral planes. Our approach is to train in simulation two separate policies to each master one of these balancing tasks. and then distil the knowledge of both tasks into a single policy which can balance the humanoid robot in 3D. Reliable balancing and locomotion in humanoid robots usual requires solving optimisation problems that explicitly model the dynamics of the robot. Reinforcement learning is an alternative approach which has recently shown promising results in control problems, and offers the promise of both more efficiently utilising the dynamics of the robot and easily adapting to novel scenarios.

## 1 Overview

This aim is to use reinforcement learning (RL) to learn a policy that balances the Valkyrie humanoid robot in 3D simulation. The policy will map the robot's perception of its environment to torques on the joints in over to control its movements. We intend to separately learn policies for balancing in the lateral and sagittal planes, and use a policy distillation method to transfer behaviour from these policies into a single policy that can solve the full 3D balancing task. This is primarily an engineering project. It will require building a simulation environment, and successfully applying both RL algorithms and policy distillation methods. This work will contribute to the longer-term goal of having a policy that balances the real robot.

## 2 Valkyrie

Valkyrie is an electronically-actuated, torque-controlled humanoid robot developed by NASA (**?**). The robot is 180cm tall, weighs 125kg, and has a 32 degrees-of-freedom body. We will use a physics simulator and work with a 3D model of Valkyrie. There are many practical reasons to do this, not least because the robot is expensive and could easily be damaged in trial-and-error experiments, and reinforcement learning requires considerable experimentation. It is important that the simulation is accurate, so that the success of this project meaningfully contributes to the goal of balancing the real robot. Of course, simulations are not perfect. There is a significant gap between control in simulation and in reality, and that will be one of the challenges worth keeping in mind. Choosing which physics simulator to use will also be an important decision to make at the start of the project.

## 3 Balancing

The task of balancing a humanoid involves the regulation of roll, pitch and yaw of the attitude of the body, and the position of the centre of mass. We frame this as a multi-task problem, requiring balancing in both sagittal plane and lateral plane. Balancing in the sagittal plane predominantly requires controlling the pitch joints whereas balancing in the lateral plane mostly requires controlling the roll joints. In humanoid robots, balancing is usually achieved by solving optimization problems derived from the dynamics of the robot (**??**). For this is be feasible, the model of the dynamics usually involves simplifying assumptions, such as linear approximations, which are sufficient for the system to work but are introduce inefficiency, since the true dynamics are not fully exploited.

# 4 Reinforcement learning

Model-free reinforcement learning methods learn how to act from the raw experience of interacting with the environment. More specifically, an agent such as our model of Valkyrie learns a "policy" - a conditional distribution over actions given a perception of the environment - such that acting according to that policy maximises expected long-term rewards. In practice we will not find an optimal policy for balancing Valkyrie, so the goal is to achieve a policy which succeeds in reliably balancing Valkyrie. It would then be possible to compare the behaviour under this policy to alternative control methods. Designing an appropriate reward for the balancing tasks and "shaping" it to encourage learning the desired behaviour will clearly be a critical aspect of the project. The environment is the state of the physics simulation, and the agent's "perception" will be measurements that sensors on the real robot would have access to, such as angles and distance of different parts of the body from the floor. The actions will be be torques applied to the joints of the simulated robot.

With a simulation of the environment, an agent which can take actions and a reward signal, we have a reinforcement learning problem. To learn a policy, requires training with a RL algorithm which can handle the continuous action space and the complex dynamics of the environment. Recently, multiple RL methods using function approximators have been observed to find impressive solutions to continuous control problems, most commonly using deep neural networks and know as "deep RL" methods.

Broadly, RL algorithms can be classified as either on-policy or off-policy methods. On-policy methods, as their name suggests, utilise only simulation data collected under the current policy. This data inefficiency makes them impractical for complex tasks where a large amount of training data is required to achieved good performance, and is a reason to prefer off-policy methods which can improve performance using data collected under any policy, However, off-policy methods combined with function approximation is observed to be highly dependent on hyperparameter setting and even then outcomes can be high variance **?**. For continuous control tasks, many off-policy deep RL algorithms have been proposed, and will be reivewed in the final dissertation, including DDPG (**?**), PCL (**??**), PPO (**?**), soft Q-learning (**?**) and soft actor-critic (**?**), Q-prop (**?**) and interpolated policy gradients (**?**).

# 5 Policy distillation

Policy distillation methods consolidate multiple task-specific policies into a single policy that performs ably at all the tasks. Often this new policy is represented in some way that is also smaller or more efficient. Distillation of multiple experts has been observed to lead to perform better than training a single policy to master all tasks (**?**). Task-specific models can either be trained sequentially or they can be trained jointly while being constrained to stay close to the shared policy. The distilled model is then trained to be the centroid of all task policies. Distilling policies is challenging because reward values can be unbounded and on different scales depending on the task, and this instability in the consolidating of different tasks. We will build on recent work that has successfully applied policy distillation to continuous control tasks (**??**).

# 6 Challenges

There will be several challenges for the project to overcome. First, must choose a physics simulator to work with. From work on a related project, there is already a Valkyrie simulation environment using OpenAI Gym (**?**) and PyBullet[1]. Alternative simulators are MuJoCo[2] and ROS-gazebo, niether of which yet have a Valkyrie environmeent, but both can work with OpenAI Gym, just like PyBullet. Similarly, there is already code for 2D sagittal plane balancing, using the DDPG algorithm implemented in Python with Tensorflow.

Another challenge will be finding a stable algorithm for learning the task. The performance level of deep reinforcement learning methods is known to be highly dependent on hyperparameter settings and network architecture, but also the scale of the rewards, different runs, different random seeds, and even different library implementations of the same algorithm (**?**).

[1] pybullet.org

[2] mujoco.org

Finally, this project is part of a longer-term ambition to have a policy learned from scratch control the real robot. Recent work has used deep RL to control humanoids in simulation and to control fixed-based robotic arms, but not yet to control a floating-base robot, and the transferring performance from simulation to reality is an open challenge (?????).

## 7 Work plan

The project will begin on Monday 14th May and run for 14 weeks, with the final thesis submitted by Friday 17th August. Below I give a high-level description of each week of the project:

1. Complete literature review of deep RL algorithms, to be included in final report. Agree approach to be used.
2. Implement primary deep RL algorithm, and demonstrate on a simple control tasks such as cartpole task. Ensure previous sagittal balancing works.
3. Build a 2D simulation of Valkyrie in chosen environment, ready for training lateral balancing.
4. Build RL agent for training lateral balancing, and begin training. Verify simulation is working as expected.
5. Train lateral balancing.
6. Complete lateral balancing task and collect performance statistics over multiple runs and random seeds.
7. Write-up interim report document results on the 2d balancing tasks, due Friday 6th July.
8. Complete literature review of policy distillation methods, to be included in final report, and implement on a toy task.
9. Build a 3D simulation of Valkyrie.
10. Distillation.
11. Distillation.
12. Run experiments to collect performance statistics, and write-up results by Friday 3rd August.
13. Buffer time to decide how to wrap-up project.
14. Edit report. Complete handover of code. Submit by Friday 17th August