Black Sheep: Low-Cost Open Source Humanoid Robotics Research Platform

Daniel Barry [2] Merel Keijsers [1] Munir Shah [4] Humayun Khan [1] Banon Hopman [3] Simon Hoermann [1] [3]

[1] HITLab NZ [2] College of Engineering [3] School of Product Design [4] Stats NZ

University of Canterbury, Christchurch, New Zealand

***Abstract*—In this paper we present a low-cost entry-level 3D printed humanoid robotics platform for the kid-sized league of the RoboCup competition. We open source the platform under an MIT license and discuss the performance of the platform in the RoboCup World Cup competition in Sydney, Australia (2019). We offer source files for the chassis structure, a custom electronics solution for power delivery, as well as source code for the platform. The results of our work show that a low-cost humanoid platform is achievable but more work is necessary to be competitive in the domain.**

This project was funded by the University of Canterbury.[1]

Introduction

Embodied cognition suggests that cognition is dependent on the agent's body, meaning that the agent's ability to act and sense are important for developing intelligent behaviour [1]. If we extend this to artificial intelligence (AI), it could be strongly suggested that the development of human-like intelligence requires the agent to have a human-like body. Humanoid robotics research is quite an active and exciting field for exploring human-like behaviours, especially as the cost of hardware decreases and computational power increases.

In 1997 the concept of *RoboCup* was proposed as a standard AI problem, requiring advancements in multiple disciplines to solve [2]. The overall goal of the competition is that in 2050, the best robots compete against the best human football players in the world - and the robots win [3]. In the humanoid league in 2019, open problems include: walking on uneven artificial grass, seeing objects in a natural light environment, localizing in a highly symmetrical environment and cooperative agent behaviours. For the goal of testing the advancements of humanoid robotics and human-like artificial intelligence, RoboCup offers a suitably difficult problem space that is yet to be completely solved after more than 20 years.

The team *Electric Sheep* was created in 2017 and secured funding in 2018, where the team was able to qualify for the RoboCup World Cup in 2019 [4]. Electric Sheep is notable as the first major league competitor from New Zealand and is currently the only competitor from New Zealand actively competing. The ambitious goal of creating a platform completely from scratch was set to both meet the team's low-budget and to show that it is still possible for new teams to enter the competition. In the following we will discuss existing approaches, our approach, the results of the our approach and future work.

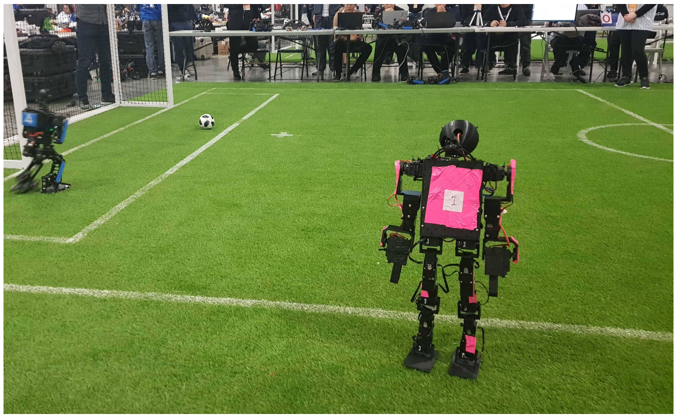


Fig. 1. Robots playing a game in controlled lighting.

Background

Many platforms suitable to the humanoid RoboCup domain exist. We will discuss a small selection that we considered whilst choosing whether to create a new platform. The Darwin OP3 platform, the 3rd version of the well tested Darwin OP series, costs $11k USD [5]. The Darwin OP3, whilst suitable for the competition, costs too much to purchase a full team. The next platform we considered was the Poppy Humanoid, which is an open source bio-inspired humanoid - but also costs a similar amount (due to an increased number of motors) [6]. Another consideration was the NimbRo-OP2 platform which both has a very pleasing design and is a well tested platform at the RoboCup competition [7]. Unfortunately it is both too large and too expensive to usable in a multi-agent team. Finally we considered the HamBot platform, again a platform that had

---

[1] Funding received from HITLab NZ, the College of Engineering and the School of Product Design.

visited RoboCup but unfortunately suffered some issues and was later abandoned [8].

The most expensive aspect of each of the platforms was the motors and all of them use some model of the Dynamixel series created by Robotis (as seen in the Darwin OP3). These motors are favoured for multiple reasons including: The ability to daisy chain power and communications via the RS-485 standard, their relatively high accuracy (4096 position encoding), their intelligent control and monitoring (e.g. temperature, angle, current, torque, voltage) and their ability to set PID values.

After reviewing the platforms, it was most obvious these motor characteristics were most desirable during walking, and not in the arms or head motors where little to no feedback is generally required. We therefore saw an opportunity to greatly reduce the cost of the platform by replacing upper body motors with cheaper, zero feedback motors. Because of the power of the motors we required, the size and shape of the lower-cost motors were greatly different, forcing us to re-think the kinematic structure of the robot. This in turn led to the decision to create a platform from scratch, as almost everything needed to be redesigned. When redesigning we decided to keep the degrees of the freedom the original Darwin OP platform had in order to ensure flexibility close to a well tested platform.

METHODOLOGY

The platform was designed and built in just two months before the qualification for RoboCup 2019 for Sydney, Australia [9]. Many revisions were made between the original qualification and the competition as a result of continuous testing and redesigns. In this section we will discuss the design and creation of the humanoid platform, specifically the electronics, chassis and software respectively.

Electronics

Our platform requires several high-current voltage conversions to ensure the relevant components are powered. Mainly we require 12V (smart motors), 5V (PWM motors) and 5V (controller boards). The reason for isolating the controllers supply and the motors supply at 5V is to ensure the voltage sensitive controller boards are always supplied with a clean 5V supply rail. Under high current draw, the voltage may drop as the DC step-down converter reaches saturation. Both of the DC step-downs are required to deliver 10A and the USB 5V is required to deliver 3A, so for a safe power margin the *power PCB* was required to handle 30A.

Where possible we used off-the-shelf components to reduce risk, such as the DC step-down converters and the USB 5V supply. Unfortunately there was no off-the-shelf solution we were able to find that allowed for switching between DC 17V and battery 16.8V (4 cell LiPo). Additionally we wanted the power supply to switch to DC when DC is available otherwise battery, alarm if the voltage drops too in the battery, ability to switch the output on/off, high current replaceable fuse and multiple output wires. Given our relatively unique requirements, we decided to design our own power circuit solution.

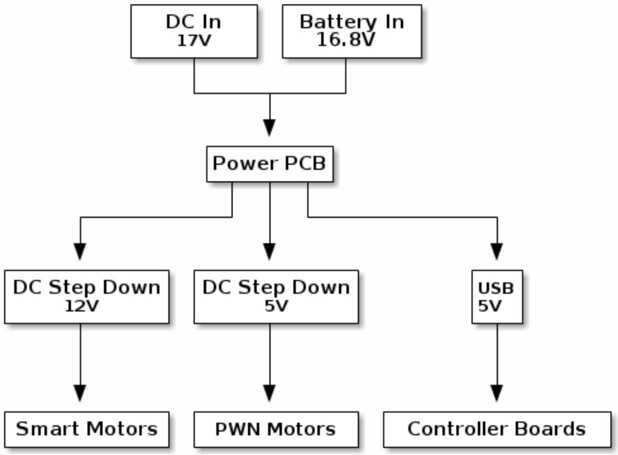


Fig. 2. Simplified input/output power diagram.

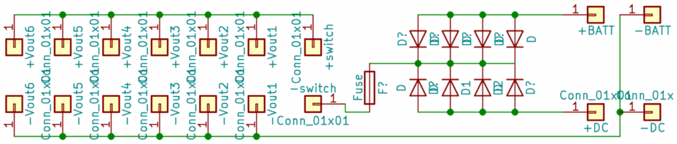


Fig. 3. The simple power PCB circuit design.

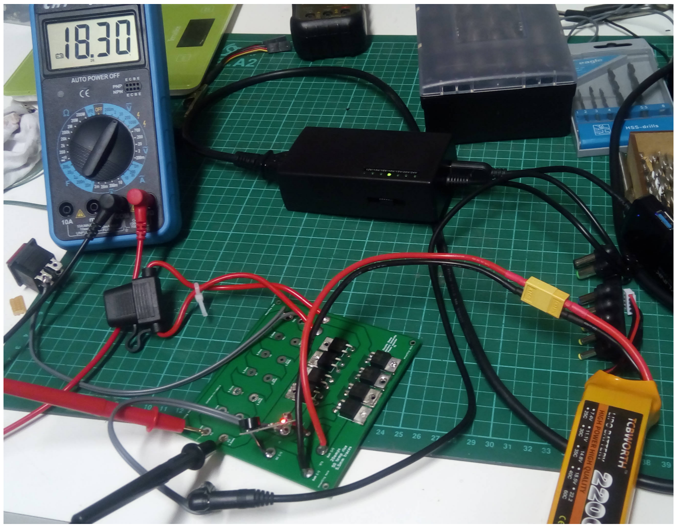
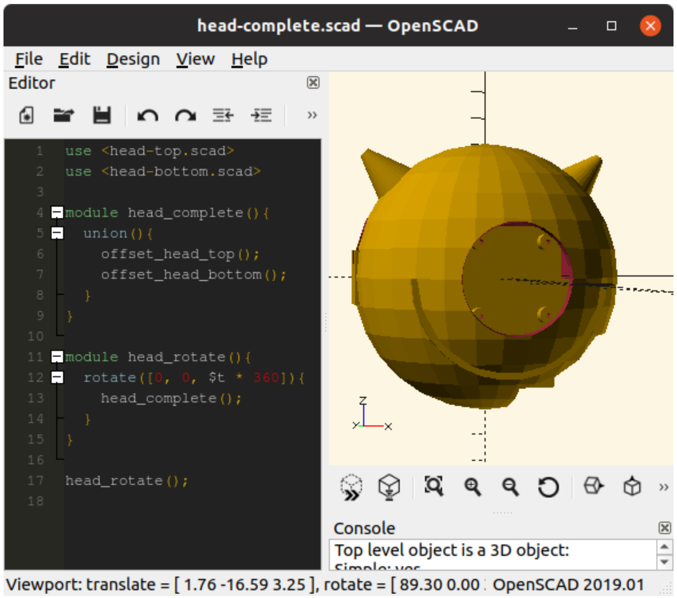


Fig. 4. Testing the power PCB design.

Chassis

The platform's chassis was designed in the parametric CAD software OpenSCAD to allow the team's mostly software based expertise to create 3D parts [10]. The platform was then printed in 1.75mm black PLA filament, chosen for it's reduced negative effects when breathed in and the PLA's ability to absorb impacts.



```
use <head-top.scad>
use <head-bottom.scad>

module head_complete(){
  union(){
    offset_head_top();
    offset_head_bottom();
  }
}

module head_rotate(){
  rotate([0, 0, $t * 360]){
    head_complete();
  }
}

head_rotate();
```

```
Top level object is a 3D object:
```

```
Viewport: translate = [ 1.76 -16.59 3.25 ], rotate = [ 89.30 0.00   OpenSCAD 2019.01
```

Fig. 5. Designing the platform's head in OpenSCAD.

Software

The software is custom and without a framework, simply for the fact that time was limited and we needed to be able to rapidly develop software around the evolving hardware. Despite this, the software has remained mostly modular and each component is relatively isolated. We will give an overview of the areas of interest:

Behaviour: This code is simple fall-through state logic, where state is recalculated per look based on entry-conditions. There is shared behaviour (such as falling down or listening to the game controller) and individual player behaviour.

Config: Our configuration system uses an open source C++ JSON library, allowing for a JSON template to define several overriding layers of configuration [11]. Configurations can be updated either via disk or debug server, with values able to be pulled in real-time over multiple threads. Values are cached once loaded in order to reduce conversion overheads.

Hardware: This section acts as an abstraction layer between hardware and software, where each piece of hardware usually expects to be able to block on the calling thread during an update. For this reason updates are performed on a separate hardware thread.

Localization: This is the start of an IMU based implementation of localization that attempts to track orientation. There is currently no drift correction, this simply shows how a future implementation may use an IMU to help track the robot's position.

Motion: Here we have a scheduler that balances between running motion scripts (a procedure of actions for the motors to be take) and Rhoban's inverse kinematics walk engine [12]. Scripts can be stacked, walking is merged together and each can be gracefully stopped.

Network: We have simple implementations for both the game controller (referee) [13] and open team communication [14].

Vision: We use the Darknet framework to implement a variation of the YOLO CNN [15]. Through our custom network we are able to achieve 10 frames per second on a 256x256 scaled image window.

WWW: We debug and read various states using an open source C++ HTTP web server [16], where we simply poll the agent to update data in the client browser. As security is not a concern, requesting specific data is achieved via URL paths which return JSON objects. Setting configuration values is achieved simply using HTTP POST.

DISCUSSION

The main contribution of the platform is a simple open-source humanoid robot that is able to pass the current qualification requirements of the kid-sized league whilst remaining low-cost (approximately $3k USD). As the size of the robots increase the cost also increases, meaning that teams will either have to seek greater funding or compete with less robots. This in turn makes the competition much more difficult for new teams to compete in as the cost to entry is increased. In this paper we show that a low-cost platform is still achievable.

Of course there are still issues to be addressed in the platform. One of the major flaws of the platform is the weight (5kg) and unreliability of the PWM motors, where the arm PWM motors essentially add 2kg of additional weight to the robot which it then struggles to lift. Because the PWM motors are so heavy, the centre of mass is very high and makes balance also quite difficult. In addition more strain is put on the motors, meaning more power consumption and heat. Yet another negative affect of the PWM motors is needing a significant amount of support circuitry for motor's high current 5V supply rail. In the next revision of the robot we will remove these motors entirely and replace them with smart motor equivalents, which are lighter, handle more torque, have a smaller profile, are more accurate and get less hot under load.

For the next version of the platform, research will be conducted in creating a brushless DC smart motor based on either RS-485 or the CAN protocol, as brushless motors and controllers have decreased in price since the popularization of multi-rotor flying craft. Another area of interest is an investigation of printing materials as PLA has shown warping at high motor temperatures, as well as breakage under high loads and shock forces.

In the spirit of the competition, version 2 will also be open source and the complete platform open sourced at the end of the annual competition in RoboCup 2020 in Bordeaux, France.

References

[1] Varela, Francisco and Thompson, Evan and Rosch, Eleanor, *The Embodied Mind: Cognitive Science and Human Experience*, 2nd ed. London, England: MIT Press, 2017.

[2] Kitano, Hiroaki and Asada, Minoru and Kuniyoshi, Yasuo and Noda, Itsuki and Osawa, Elichi, "RoboCup: The Robot World Cup Initiative," 1997, pp. 340–347.

[3] RoboCup Federation, "Objective." 2019.

[4] Barry, Daniel and Curtis-Black, Andrew and Keijsers, Merel and Shah, Munir and Young, Matthew and Khan, Humayun and Hopman, Banon, "Electric Sheep Team Description Paper Humanoid League Kid-Size 2019," University of Canterbury (NZ), 2019.

[5] Robotis, "Darwin OP 3." 2019.

[6] Poppy Project, "Poppy Humanoid." 2019.

[7] Ficht, Grzegorz and Allgeuer, Phillip and Faranzi, Hafez and Behnke, Sven, "NimbRo-OP2: Grown-up Printed Open Humanoid Platform for Research," 2017.

[8] Bestmann, Marc and Reichardt, Bente and Wasserfall, Florens, "Hambot: An Open SOurce Robot for RoboCup Soccer," 2015, vol. XIX, pp. 339–346.

[9] Barry, Daniel and Curtis-Black, Andrew and Keijsers, Merel and Shah, Munir and Young, Matthew and Khan, Humayun and Hopman, Banon, "Electric Sheep - Robot Specification," University of Canterbury (NZ), 2019.

[10] Kintel, Marius and Wolf, Clifford, "OpenSCAD." 2019.

[11] Lohmann, Niels, "JSON for Modern C++." 2018.

[12] Rhoban, "IKWalk." 2015.

[13] RoboCup Humanoid Technical Committee, "RoboCup SPL and Humanoid League GameController." 2019.

[14] RoboCup Humanoid Technical Committee, "Mixed Team Communication protocol." 2019.

[15] Redmond, Joseph and Farhadi, Ali, "YOLOv3: An Incremental Improvement," *arXiv*, 2018.

[16] Heller, Forrest, "EWS - Single .h File C Embeddable Web Server." 2019.