RoboCup 2024 Humanoid Soccer Competition Bold Hearts Software Description

Daniel Barry[†], Alessandra Rossi, Lewis Riches, Roberto Figueiredo, Karen Archer, Bente Riegler, Abel Moyo, Humayun Khan, and Daniel Polani

[†]db13acy@herts.ac.uk, Bold Hearts, https://robocup.herts.ac.uk
SPECS, University of Hertfordshire, Hatfield, AL10 9AB, UK

Abstract. In this document we describe the software used and the technological progress made by the Humanoid Kid-Size League team Bold Hearts in preparation for the RoboCup WorldCup 2024 competition. We specifically discuss the current state and intention of walking, vision, localization, team communication, behaviour and contributions made to the Scientific community.

1 The Robots

We joined the Humanoid League in 2013 with Darwin OP-1 robots. However, after some years these robots became outdated and we started to upgrade the platform each year to meet the league requirements that move us towards the 2050 RoboCup objective. These yearly changes brought us to create the Bold Bot as shown in Figure 1, where the Darwin OP structure has been almost entirely replaced.

We explore a new modular controller board to replace the original CM730, one of last remaining parts of the original Darwin OP-1 platform. Our objective is to upgrade the control board computation, IMU quality & update frequency, support larger motors & batteries, and non-linear motor control signals.

2 Walking & Motion

At the 2023 competition we had difficulty with all motion due to:

Fig. 1. The evolving BoldBot platform.

1. *~30ms motor position latency* – We had an ~30ms USB latency enforced by the dynamixel SDK and Linux environment, causing motor positions to be reported significantly delayed with respect to their current position. We also saw that as a result of this delay, the updates per second dropped from the previously achieved ~100Hz to ~30Hz, causing staggered motion.
2. *Motion interpolation* – In our motion scripts we would interpolate between key frames (e.g. a, b, c), for example: $a \rightarrow b \rightarrow c$. In reality, the motors would never quite reach their target angle and would instead achieve $a \rightarrow b' \rightarrow c'$. The longer the script, the larger the error is able to accumulate. With smaller humanoids the issue was not observed as the MX28s could overcome the weight of the robot. With the increasing size (and therefore weight), the error was significantly increased and caused motions to vary significantly based on factors such as surface friction. Given the possibility that motors may not reach their target position, we now treat each interpolation independently,

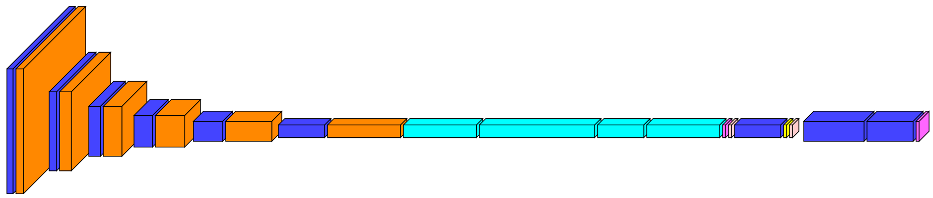

Fig. 2. The xYOLO CNN developed by Electric Sheep [3]. The cyan layer shows the XNOR 'backbone' on the network.

e.g. $a \rightarrow b'$ and $b \rightarrow c'$. With this change we found the scripts to run significantly more reliably.

3. *Motor initialisation voltage* – Our previous strategy to initialisation was to start the MX28s as quickly as possible, which was successful for many years. Since upgrading to the larger MX64s we have discovered that they draw more current and are more susceptible to low voltage conditions. As a result the MX64s need to ensure a stable power supply before being initialised to avoid unreliable bring-up.

We utilize the open-source, open-loop 2015 IKWalk Engine from Rhoban [1], with an appropriate ROS2 wrapper[1]. We tune the walk parameters and motor PID parameters in order to achieve stable walking.

The updates to the platform have also lead us to investigate using larger motors. We are aware that this will have an impact on open-loop walking as it continues to be difficult to tune for an increasingly complex environment. We have investigated utilizing model predictive control (MPC) [2] based on a learned model to adjust open-loop walk parameters. Several robust walks, such as zero moment point (ZMP), make assumptions regarding knowledge of centre of mass (CoM) or centre of pressure (CoP). We look to minimise reliance on such sensors as algorithms that rely on flat-ground with continuous ground contact are unlikely to yield the goal of running.

3 Vision

Bold Hearts previously used semantic segmentation as presented in 2019 to great success for detecting balls and goals [4]. One major challenge for the team was maintaining and compiling Tensor Flow. The xYOLO network developed by Electric Sheep[2] [3] proved to be simple to maintain[3] via a ROS2 wrapper maintained by the team. The network as shown in Figure 2 is upscaled for depth,

[1] IKWalk ROS2 node wrapper: https://gitlab.com/boldhearts/ros2_ik_walk
[2] Electric Sheep: https://electric-sheep-uc.github.io
[3] Fork of pjreddie's Darknet framework by AlexeyAB: https://github.com/AlexeyAB/darknet

Fig. 3. xYOLO network predictions, showing detections for the ball, goal, line crossings and a robot. Detections in a single frame may be missed or not accurate, but we aim to detections quickly and often.

with input size also slightly upscaled, resulting in a network capable of detecting balls, goals, cyan & magenta humanoids and line crossings in realtime. Figure 3 demonstrates our detection capability on a challenging natural light scenario.

We look to accelerate the network by leveraging JPEG DCT tables [5] provided by the camera frame, which appears to better preserve frequency information [6]. JPEG encode and decode has hardware acceleration in almost all modern CPUs and GPUs due the prevalence of the format. Further modifications to the Darknet framework will be requires to accept DCT coefficients.

4 Localization

In Figure 4 we show a localisation simulator in development for automating the testing of localisation algorithms against the same vision sensory data, as well as prevent regression during development. Specifically the simulator allows us to spawn multiple agents and we look to use the state of the ball and other agents in team communication to help de-localised players re-localise during active play.

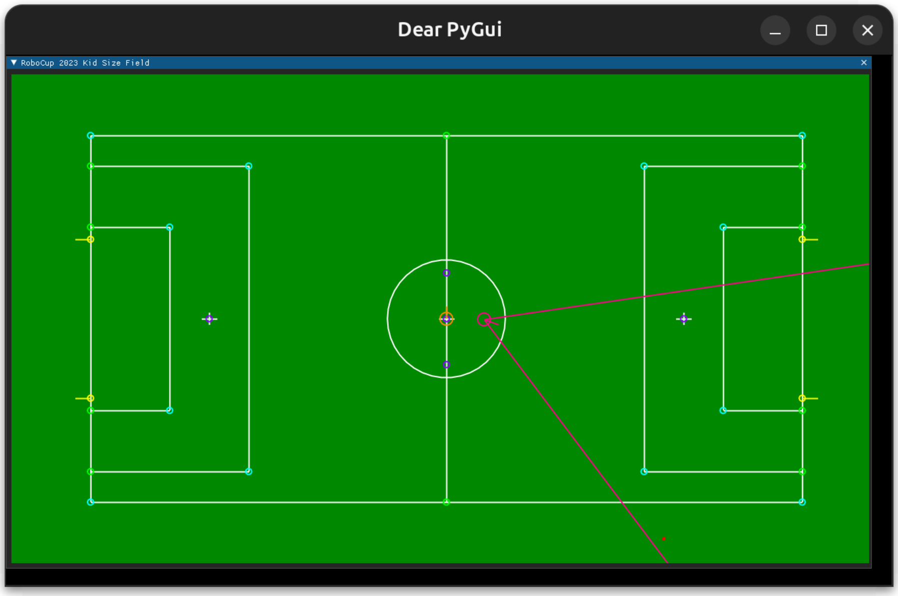

Fig. 4. Localisation simulator in development.

5 Team Communication

We have improved support for the game controller, including the transmission of heartbeat messages which previously was a source of confusion as to whether the robot could hear the game controller.

Team communication is currently based on the unmaintained `mitecom` project[4]. We are migrating to a more modern protocol[5].

6 Behaviour

Our framework is based on the ROS2 port effort started in 2019 [7]. We previously had great issue with the action server either crashing during a spin or stuck in an indefinite spin. This was realised and fixed upstream only, requiring custom patches.

The behaviour system works on the principle of a fall-through decision tree, where safety critical decisions are evaluated first, and then if all conditions are met the robot is allowed to play. No blocking behaviour is implemented in the decision tree and decisions are re-evaluated as new sensor data becomes available.

[4] Fumanoid & TC mitecom: https://github.com/RoboCup-Humanoid-TC/mitecom
[5] RobocupProtocol: https://github.com/RoboCup-Humanoid-TC/RobocupProtocol

7 Contributions

The *Robotics* module at the University of Hertfordshire successfully ran for the first time in 2023, which gave Bachelors students the opportunity to work with ROS2 and humanoid robots as part of an online module. The module was run and created by RoboCup members, to both give value back to the University and further expose RoboCup to students.

Two of our team's members also are, for several years, volunteering to be part of the RC-HL organisation committees for improving the league scientifically, organizationally and socially.

The team also is actively involved in editing and contributing to the unique journal project, titled "*The human in the loop: perspectives and challenges for robots' behaviours in RoboCup 2050*" which is in the final stages of publication.

Two years ago we started an annual RC hackathon event for students in both Schools of Engineering and Computer Science for robot design and software solutions. Team members are participating at demos, presentations for students and general public interaction events, and be one of the female faces at RC2023[6] for promoting RoboCup and related research.

We have several open-source ROS2 nodes to share with the community. We currently offer the CM730 node[7] (with the Dynamixel node likely to be open-sourced once we are more confident regarding stability), the Tensorflow node for running networks of various kinds[8], the IMU node[9] (currently used in some industry projects), a node for useful transforms[10], setting up the camera and publication of frames in a given format[11] (developed and maintained by an ex-RoboCupper, actively used in industry), a simple vision node[12], a description of a robot for simulation[13], simulation tools for virtual testing[14], a node for communicating with the game controller[15], and our legacy code base[16].

We have open-sourced some internally used tools, such as *Darknet Visual*[17], which allows for vector representations of neural networks to aid in publications (as seen in Figure 2). We also open-source our fixes to some legacy software, for

[6] Interview https://t.co/GXjej73za0
[7] CM730 node: https://gitlab.com/boldhearts/ros2_cm730
[8] Tensorflow node: https://gitlab.com/boldhearts/ros2_tflite
[9] IMU node: https://gitlab.com/boldhearts/ros2_imu_tools
[10] COM node: https://gitlab.com/boldhearts/ros2_humanoid_tf
[11] V4L2 node: https://gitlab.com/boldhearts/releases/ros2_v4l2_camera-release
[12] Vision node: https://gitlab.com/boldhearts/ros2_bold_vision
[13] Robot description node: https://gitlab.com/boldhearts/ros2_boldbot
[14] Simulation tools: https://gitlab.com/boldhearts/ros2_rcss3d
[15] Game controller node: https://gitlab.com/boldhearts/ros2_game_controller
[16] Legacy Bold Humanoid repository: https://gitlab.com/boldhearts/bold-humanoid
[17] Darknet Visual tool: https://gitlab.com/danbarry16/darknet-visual

example the `yolo-labeling-tool`[18] which allows for offline labelling of Darknet training data with very little resources.

8 Research

Several team members explore the use of intrinsic motivation through *empowerment* [8] as an approach to remove the need for hand-tuned motion scripts. This is currently explored with 2D humanoids. Some preliminary work was presented at the Rhoban Workshop in 2023, Bordeaux, France.

References

1. Rouxel, Q., Passault, G., Hofer, L., N'Guyen, S., Ly, O.: Rhoban hardware and software open source contributions for RoboCup humanoids. Presented at the (2015).
2. Shafiee-Ashtiani, M., Yousefi-Koma, A., Shariat-Panahi, M.: Robust bipedal locomotion control based on model predictive control and divergent component of motion. In: 2017 IEEE international conference on robotics and automation (ICRA). pp. 3505–3510 (2017). https://doi.org/10.1109/ICRA.2017.7989401.
3. Barry, D., Shah, M., Keijsers, M., Khan, H., Hopman, B.: xYOLO: A Model For Real-Time Object Detection In Humanoid Soccer On Low-End Hardware. In: IVCNZ. IEEE (2019).
4. Dijk, S.G. van, Scheunemann, M.M.: Deep learning for semantic segmentation on minimal hardware. In: Holz, D., Genter, K., Saad, M., and Stryk, O. von (eds.) RoboCup 2018: Robot world cup XXII. pp. 349–361. Springer International Publishing, Cham (2019).
5. Gueguen, L., Sergeev, A., Kadlec, B., Liu, R., Yosinski, J.: Faster neural networks straight from JPEG. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.) Advances in neural information processing systems. Curran Associates, Inc. (2018).
6. Xu, K., Qin, M., Sun, F., Wang, Y., Chen, Y.-K., Ren, F.: Learning in the frequency domain. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) (2020).
7. Scheunemann, M.M., Dijk, S.G. van: ROS 2 for RoboCup. In: Lecture notes in computer science. pp. 429–438. Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-35699-6_34.
8. Klyubin, A.S., Polani, D., Nehaniv, C.L.: Empowerment: A universal agent-centric measure of control. In: 2005 IEEE congress on evolutionary computation. pp. 128–135 Vol.1 (2005). https://doi.org/10.1109/CEC.2005.1554676.

[18] yolo-labeling-tool: https://gitlab.com/boldhearts/yolo-labeling-tool