Humanoid Adult Size Champion 2021 Sweaty

Klaus Dorer[1], Maximilian Giessler[1], Ulrich Hochberg[1], Manuel Scharffenberg[1], Rico Schillings[1], and Fabian Schnekenburger[1]

University of Applied Science Offenburg, Germany
{maximilian.giessler}@hs-offenburg.de

Abstract. Due to the Covid-19 pandemic, the RoboCup WorldCup 2021 was held completely remotely. For this competition the Webots simulator[1] was used, so all teams needed to transfer their robot to the simulation. This paper describes our experiences during this process as well as a genetic learning approach to improve our walk engine to allow a more stable and faster movement in the simulation. Therefore we used a docker setup to scale easily. The resulting movement was one of the outstanding features that finally led to the championship title.

Keywords: CMA-ES · Docker · Webots · ROS2.

1 Introduction

Due to the Covid-19 pandemic, in early 2021 the decision was taken to run the annual RoboCup competition as a worldwide virtual competition. The technical committee took the decision to partner with Cyberbotics in order to provide a platform (Webots) for a simulated humanoid robot challenge. Tremendous effort was spent to define and implement a specific communication protocol for such a competition as well as an automated referee. Also, the existing Webots simulation has been extended to provide support for joints with backlash and for the special turf. Despite the very narrow timeframe for this, this goal has been achieved an virtual competitions could be run.

Teams were asked to transform their real robots into simulated Webots robot models. In the end, three teams of humanoid adult size league accepted this challenge and successfully qualified for the 2021 RoboCup Worldwide.

2 Sweaty

Sweaty is a 1.70m tall humanoid robot that first participated in RoboCup 2014 in Brazil. 2019, Sweaty achieved the title of a vice-world champion in the humanoid adult size league. One of the major tasks for 2021 has been to make the robot available in the Webots simulation.

[1] https://cyberbotics.com/

2.1 Robot Model

Sweaty is special in that it uses linear actuation in its legs [3]. So the motors are not sitting in the rotation axes of the joint, but indirectly drive the joints. Apart from other advantages, this provides a considerably lower backlash compared to having e.g. Dynamixel servos sitting in the joint. This is because the force of the robot weight pushes on the mechanical joints.

In a first attempt, this design was transferred into simulation. However, the simulation speed dropped to unacceptable cycle times with only one robot of this kind on the field. Therefore, the technical committee decide to allow to replace the linear actuated motors with servo motors that have a comparable backlash as the linear motors had. This rendered the motor mapper component in our software not needed in the simulated version of Sweaty (see 2.2). An image of the simulated and the real Sweaty is shown in 1.

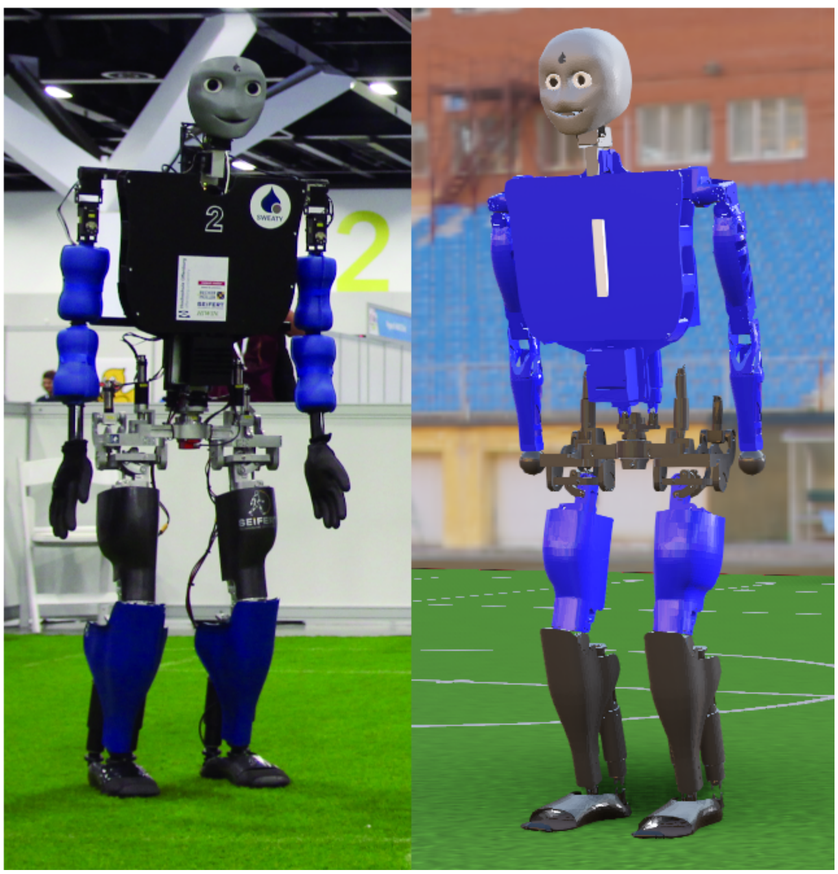

Fig. 1. Real Sweaty (left) and simulated (right).

2.2 Software Architecture

Sweaty uses ROS2 as its software architecture. Figure 2 shows the components. Similar to the previous design, all hardware components are incorporated into the ROS2 ecosystem via their corresponding drivers.

For the simulation, all hardware related nodes like imu/camera/motor joints and force sensors were replaced by an additional Webots-controller-node. We just had to make some minor changes in our walk engine to get the robot moving inside Webots, e.g. handle the divergent update rate from asynchronous hardware sensors to the simulation step size.

For decision making, our software uses an adapted version of our 3d soccer simulation team Magma as has been done in previous years. To perform the communication between our ROS2 ecosystem and the decision component written in java, a ROS bridge was used. This has been the only major change compared to the ROS1 setup used in previous years.

2.3 Vision

Our vision [4], used for the object detection and localization on the real robot, was already able to also detect landmarks and the ball from the rendered camera images of the Webots simulator. To improve the accuracy, we trained our model with the rendered images. Therefore, we have developed a tool for auto-labeling. It moves the camera randomly around the soccer field and extracts the landmark / object position from Webots to label the images automatically. This allowed us to generate 200-300 labeled images per minute on a single Webots instance to retrain our vision model. We used roughly 6000 such images to get a result good enough to localize and play well, but more would have been no problem to generate.

3 Learning

A major building block for our success has been a successful improvement of our walking gait by the use of genetic optimization. It is noteworthy that the gait of our real robot worked with very minor manual changes immediately in the Webots simulation at a speed of about 0.2 m/s. With the availability of having the robot in a simulation, we used the strength of simulations to allow for machine learning.

3.1 Algorithms

Based on our experiences in the RoboCup 3D simulation league [1], we already had a existing solution for the genetic algorithm CMA-ES [2]. CMA-ES is especially powerful when there is already a working solution to start with as it was the case here. Our implementation is based on the commons math implementation[2] but has been extended twofold: first, it allows to calculate the fitness of

[2] https://commons.apache.org/proper/commons-math/

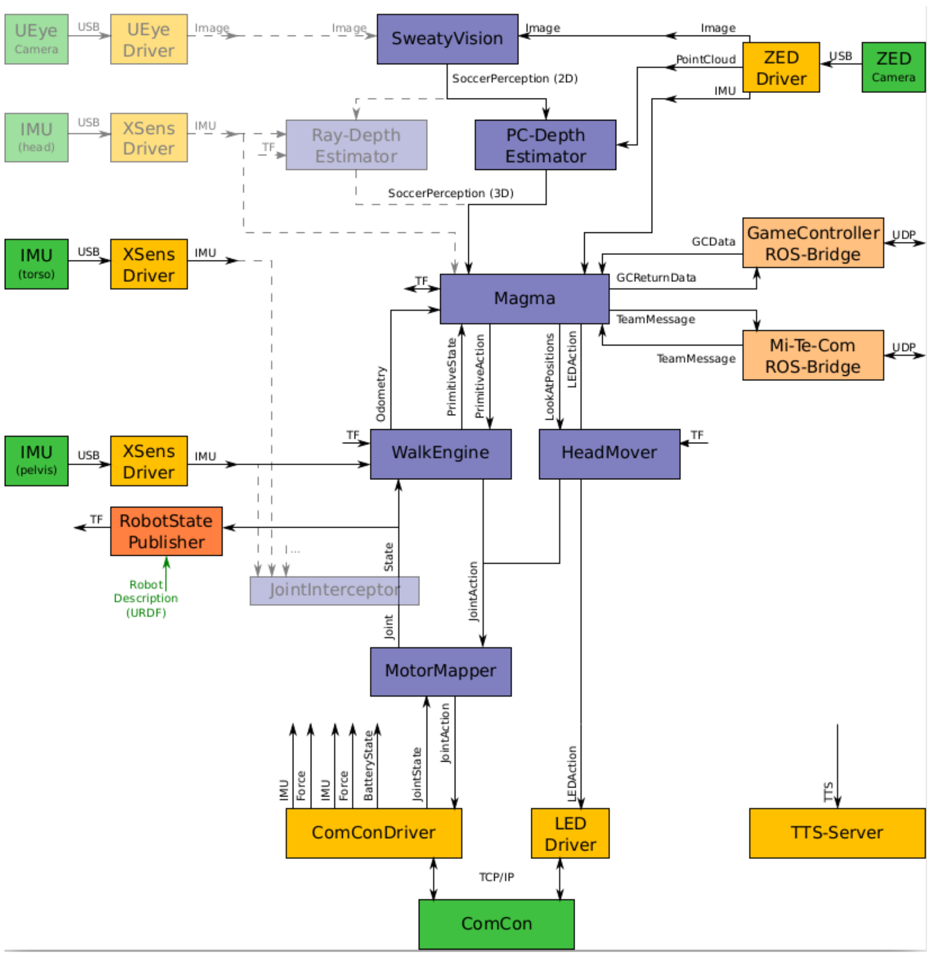

Fig. 2. Nodes and exchanged messages in ROS2.

each individual of a generation in parallel. Second, it allows to run oversampling runs of the same individual in parallel.

The learning architecture uses two separate docker containers in order to spread learning on multiple computers of the university. For the communication between those containers a REST API was implemented. The first container is the learning supervisor and runs the parallel CMA-ES algorithm and provides a ticket system by using the REST API. The other container includes a Webots simulator instance as also our robot specific code. This container then is able to send a GET request for a new ticket to the supervisor. The supervisor then responds with a new parameter set, generated from the CMA-ES. With this new parameter set, the walk engine of the robot is initialized and starts an attempt with those generated values. We specified a fixed duration Δ t to ensure each attempt was executed with the same conditions.

The reward function used the walked distance during the attempt to calculate the reward together with a heavy penalty for falling. This reward is then send back by a POST request to the supervisor to announce the completion of the previously requested ticket.

To start a learning run, docker compose was used. This allows to start the learner container with multiple instances easily by using the `--scale trainer=N` command. For large scaling across multiple server instances it was just necessary to specify the IP address of the instance where the learning supervisor was running. Figure 3 shows the described setup.

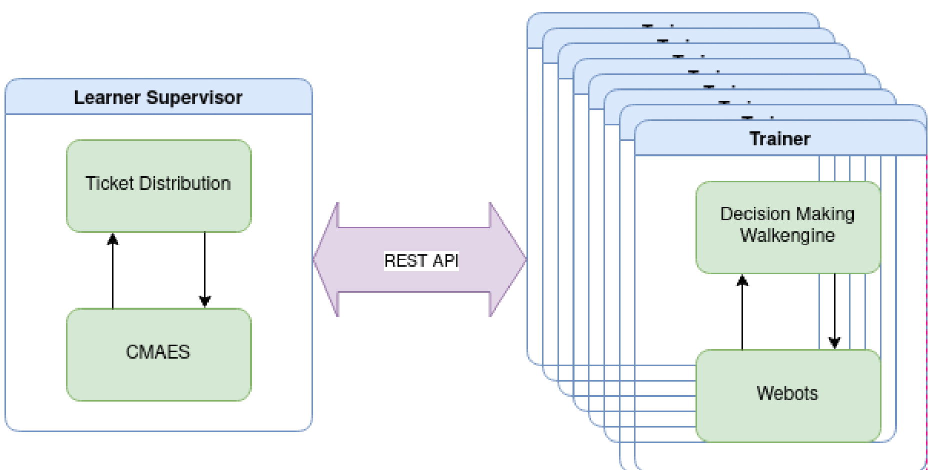

Fig. 3. Docker setup for the parallel CMA-ES

For this first approach we made some of the parameters (see table 1) used by the walk engine injectable to let CMA-ES optimize them. These parameters were stored in an external file to allow a quick exchange. In a further approach we attached a stability criteria to evaluate the performance of the gait. Therefor we integrated a calculation for the Imaginary Zero-Moment Point (IZMP) [6] and the support polygon. The support polygon is calculated using the geometry, the orientation of the feet and force-/torque-sensors mounted in both feet. We used the force in z-direction and the 3D pose to detect the contact state of the robot. To quantify the stability of the gait cycle we also introduced an algorithm, which evaluates on the one hand if the IZMP coordinates are located within the support polygon. On the other hand it evaluates the distance from the IZMP to each side of the support polygon. The integration into the reward function was realized through penalty for each cycle the IZMP was not located in a desired region of the support polygon.

Table 1. Used walk parameters to learn by CMA-ES.

Name	Description
max-step-size-turn	maximal stepsize while turning around
max-step-size-x (y)	maximal stepsize while moving on one axis direction
omni-dir-walk-dis	maximal destination distance to use omnidirectional walk
rate	update rate
control.hip (.pitch / .foot)	P/I/D values to control specific joints

3.2 Results

The learning runs were distributed on an increasing number of computers at our university with a peak of 500 docker containers running in parallel. Typical setups have been to use 50 individuals in a population and perform 30 oversampling runs for each individual to average out noise in each try.

Figure 4 shows the improvements over such a learning run. Typically, already the best individual of the first generation outperformed the manual solution used as a starting point. The best fitness of about 8.5 achieved in generation 81 translates into a speed of roughly 0.43 m/s. The speed of the walk gait doubled, while also the stability of the walk was improved by learning.

4 Conclusion

The required transfer of our real robot into the simulation was a profitable challenge. The competition was our first extensive use of ROS2, which runs very stable. The described approach for genetic optimization was also working very well an within a short timeframe of development. By using docker we can easily scale through available servers and take advantage of the resulting acceleration. The learned parameters already show a more stable movement and its speed could be increased by a factor of two.

Sweaty won the final game of the adult size competition 21:0. The overall number of goals was subject to an undesired behavior of the automated referee

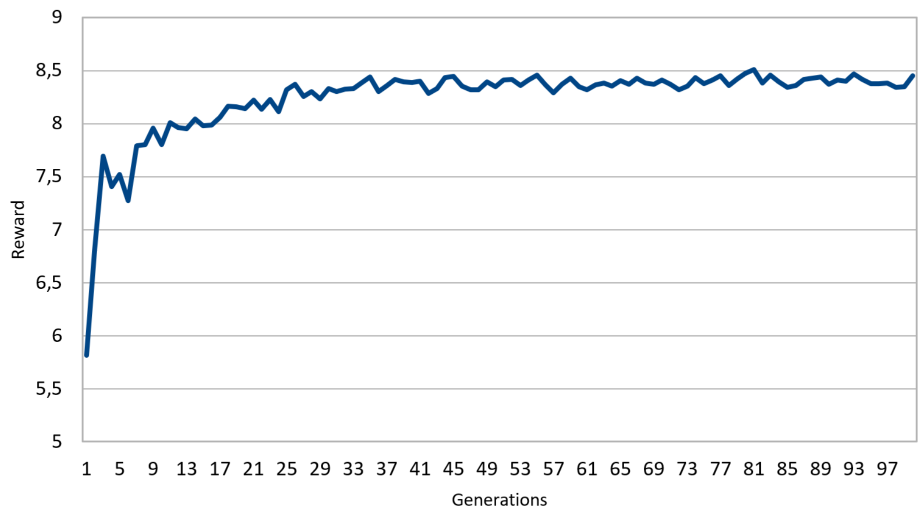

Fig. 4. Fitness of the best individual over the generations.

to extend the game for each goal scored. But the result shows that the Sweaty team succeeded best in transferring the work on the real robots into the Webots simulation. We attribute this to two main factors: first, our team has been the only team that finally had an overall technical system that was running stable enough throughout a game to keep two players on the field up and walking. This is mainly a matter of excellent programming skills in the tight schedule of the league. Second, machine learning turned out to improve the game play considerably. More important than the duplication of our walking speed - in humanoid adult size league, where robots are typically not able to get up after falling - has been the higher stability of the gait learned.

For the future we will try to transfer the learned parameters back to the real sweaty to observe if we also can use CMA-ES to optimize the behavior of our real robot. Furthermore we want to increase the parameter space to learn more parameters. At this time, only high level parameters like step length, step frequency and similar were subject to learning. In the future, this could be extended possibly also to the parameters that define the angular motions of joints directly as is already done for our simulated NAO robots of the 3D soccer simulation league [1]. Also we will try to learn further motion sequences, e.g. executing a multi-directional kick [5].

Acknowledgements

We highly acknowledge the tremendous effort that was taken by the technical committee of the humanoid league to get a competition running in the Webots simulator.

References

1. Dorer, K.: Learning to use toes in a humanoid robot. pp. 100 – 111. RoboCup 2017: Robot World Cup XXI, Springer, Heidelberg (2017)
2. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evol. Comput. **9**(2), 159–195 (2001)
3. Scharffenberg, M., Hochberg, U.E., Dorer, K., Friedrich, A., Wülker, M.: Improving torque and speed of joints by using rod-and-lever systems for electrically driven humanoid robots. 10th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots (2015)
4. Schnekenburger, F., Scharffenberg, M., Wülker, M., Hochberg, U., Dorer, K.: Detection and localization of features on a soccer field with feedforward fully convolutional neural networks (fcnn) for the adult-size humanoid robot sweaty. Proceedings of the 12th Workshop on Humanoid Soccer Robots, 17th IEEE-RAS International Conference on Humanoid Robots, IEEE, Birmingham (2017), https://nbn-resolving.org/urn:nbn:de:bsz:ofb1-opus4-26557
5. Spitznagel, M., Weiler, D., Dorer, K.: Deep reinforcement multi-directional kick-learning of a simulated robot with toes. In: 2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). pp. 104–110 (2021). https://doi.org/10.1109/ICARSC52212.2021.9429811
6. Vukobratovic, M., Borovac, B., Šurdilović, D.: Zero-moment point — proper interpretation and new applications. In: IEEE-RAS International Conference on Humanodi Robots. pp. 237–244. IEEE (2001)