ITAndroids Humanoid Team Description Paper for RoboCup 2020

Alexander Justa, Caroline Silva, Davi Herculano, Francisco Silva, Gustavo Misk, Gustavo Pereira, Igor Ribeiro, João Faria, Lucas Steuernagel, Marcos Máximo, Miguel Sampaio, Misael Cavalcanti, Otávio Ribas, Paulo de Castro, Rodrigo Aki, and Thiago Tonaco

Autonomous Computational Systems Lab (LAB-SCA)
Aeronautics Institute of Technology (ITA)
São José dos Campos, São Paulo, Brazil
{alexander.justa,herculanodavi,fbrunodr,gudmisk,gupereira97,
joaocdffilho,lucas.tnagel,miguelangelo.dss,misaellvcc,otaviohrguimaraes,
prfc.ubr,krodrigo71008,tonaco.k.t22}@gmail.com
{carol.c.duarte,ribeiro_igor}@hotmail.com
mmaximo@ita.br
http://www.itandroids.com.br

Abstract. ITAndroids is a robotics competition group associated to the Autonomous Computational Systems Lab (LAB-SCA) at Aeronautics Institute of Technology (ITA). ITAndroids is a reference team in Latin America, having won 42 trophies in robotics competitions in the last 8 years. In 2016, the group acquired a Robotis OP2 robot and material to build four more robots. In 2017, the team built four Chape robots and participated in RoboCup Humanoid KidSize for the first time. Since 2018 the team has been qualified for the quarterfinals. This paper describes our recent development efforts for RoboCup 2020.

1 Introduction

ITAndroids is a robotics research group at Aeronautics Institute of Technology. The group is multidisciplinary and contains about 50 students from different undergraduate and graduate courses. We are considered a reference team in Latin America, where we have won 42 awards in the last 8 years, including 8 in the Latin American Robotics Competition (LARC) 2020.

Regarding our humanoid team, we have been struggling with low cost robots since 2013, thus making it very hard to attain good results in competitions or even qualify for RoboCup. However, in 2016, we have received a Robotis OP2 robot and enough material to build 4 other robots. In 2017, we developed our own robot Chape and competed in RoboCup Humanoid KidSize, which was a great opportunity for learning. In LARC 2019, we placed 1st and 2nd in Humanoid Robot Racing with our two robot designs, and placed 1st in Humanoid KidSize.

This paper presents our recent efforts in developing a humanoid robot team to compete in RoboCup Humanoid KidSize 2020. The rest of the paper is organized as follows: sections 2 and 3 introduce the robot hardware. Sec. 4 presents

our software architecture and tools. Sec. 5 explains our computer vision techniques. Sec. 6 shows our localization approach. Sec. 7 discuss our motion control algorithms. Finally, Sec. 8 concludes and shares our expectations for the future.

2 Mechanics

The mechanic hardware project is based on the DARwIn-OP humanoid robot. Some changes were made in order to meet the team's needs. The robot's CAD was developed using Dassault Systèmes SOLIDWORKS [8].

We designed the robot's torso placing special emphasis on the mechanics and electronics integration. The torso should accommodate PCBs and cables while allowing easy access and assembly of the electronic hardware. For this purpose, we chose a drawer-like configuration. The design leaves room for the airflow and fans were placed on the upper back side, in order to avoid overheating.

The PLA covers of the torso have small vents, in order to allow the airflow, whereas prevent synthetic grass and small particles from entering in the torso and harming the electronics. To improve the lift movement after falls, new PLA gloves were designed. Due to the breaking of the shoulder servomotors, PLA shoulder pads were also designed.

To suit the competition's constraints, a new head was designed. This structure is formed by a 1.5 mm sheets of aluminium 5052-H34, which serves as a camera base, as well as a PLA cover, which is internally resealed with silicone. This project was tested in LARC 2019 and presented excellent results.

During 2019, we also intensively tested our new optimized leg design [19]. Thus, we are keeping this design for RoboCup 2020 and upgrading all our robots to use the new design instead of the old one which was based on DARwIn-OP. In Fig. 1, the current design of Chape 1st generation is shown.

We are also developing a 2nd generation of our robot which will be taller (about 65 cm) and feature XM-540 servos. The CAD of this new generation is now finished and we are beginning manufacturing of the prototype.

3 Electronics

The new version of Chape electronics is aimed at our Chape 2nd Generation robot and brings relevant improvements compared with the previous one. These changes are motivated by simplifying the design and improving the robot performance to allow operation of the physically increased robot. The design updates can be grouped as, integration of the electronics, improved leg servos networks, battery enhancement, a more powerful main computer, and a neural networks dedicated processor.

In order to simplify the electronics, the team worked in integrating PCBs. This brings many advantages: reducing of cabling, more reliable connections, and simplification of the mechanics. The main change is the integration of Control and Monitoring Board (CMB) with the Power Board (PWB) into an unique

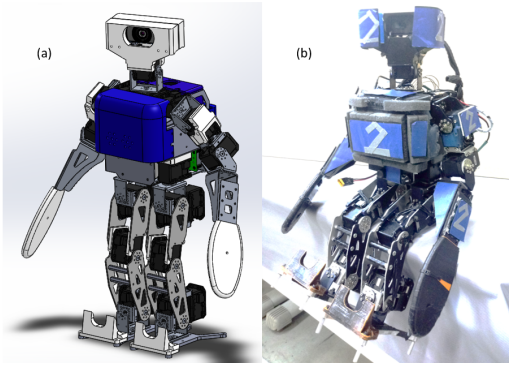

Fig. 1. (a) The current CAD Chape (1st gen.), (b) the current Chape (1st gen.).

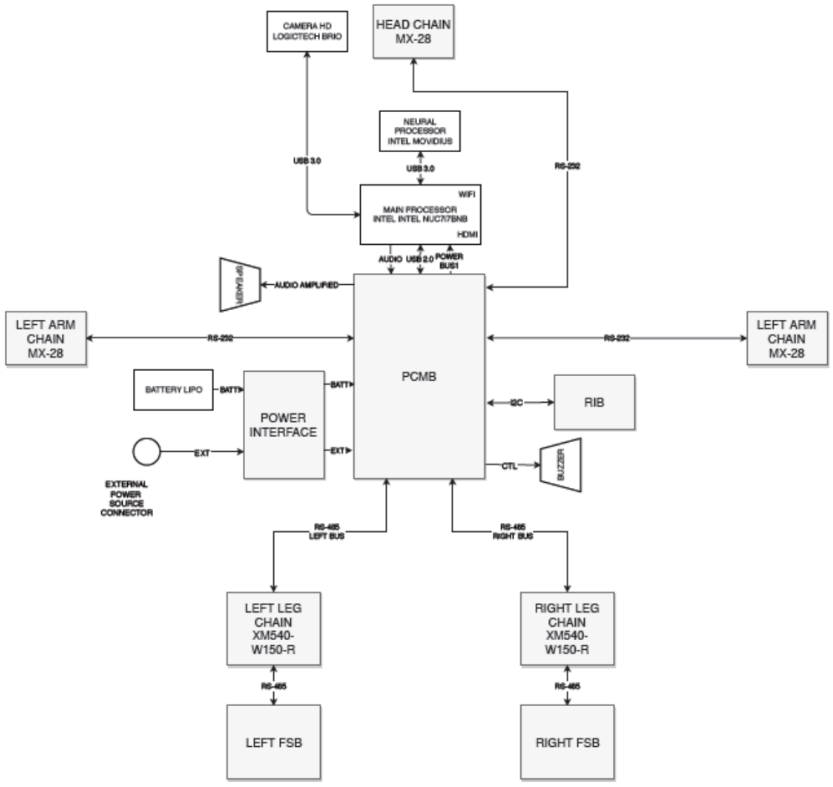

Fig. 2. Chape Electronics Architecture.

board Power Control and Monitoring Board, a 4 layer board located in the robot's torso. This modification increases robustness to intermittent failures caused by mechanical vibration and shocks.

The previous robot used an unique servos network of 20 servos communicating using a TTL interface. To improve the servos update rate, each leg now has its own differential RS-485 network ended by Foot Sensor Board (FSB). The leg servos of the new robot are XM-530 servos, which are more powerful than the MX-28 used in the current robot. The differential RS-485 is more robust to noise and adds higher throughput of communication. The FSB gives more stability to robot standing and movement. In order to reduce the weight and the cost of the robot, arms and neck are still using the MX-28 servos.

It is natural that a taller robot requires more energy to operate and a new LiPO battery with 3,000 mAh was included. The PCMB brings a dedicated battery monitoring circuit dedicated to monitor the battery state and alarm when critical conditions occur as very low voltages. Due the high current capacity, a fuse was included to protect the electronics from overcurrent events and short circuits.

The main computer was upgraded from a Intel NUC i5 4th generation to a Intel NUC i7 7th generation, which enhances the available processing power for the high level software. The new Chape also has a Intel Movidius 2, a dedicated stick processor from Intel for neural networks processing, which communicates with NUC via USB 3.0 bus.

4 Software Architecture and Tools

We use a module-based layered approach for code architecture. The main third-party libraries used are OpenCV (computer vision) [3], Eigen (linear algebra) [10], Boost (general purpose) [1], Tensorflow (machine learning) [2] and Keras (neural network) [5]. We heavily rely on the Robot Operating System (ROS) framework [16] and its related tools for testing and debugging purposes. We also use Qt [12] for graphical interface in our debugging tools. A full Humanoid Kid-Size simulation was also developed using Gazebo [4].

A new addition of our software for RoboCup 2020 is the game controller integration. Last year, we faced several problems to adapt it to our decision tree. As we started to plan a software refactoring, we decided to implement correctly the game controller to our decision making.

A telemetry system was developed for debugging and calibration purposes. It allows real-time feedback and control, as well as data logging and replaying. It uses rqt, the Qt-based framework for GUI development for ROS, and rviz, the 3D visualization tool for ROS, for easy user interfacing. This system has been extremely useful for vision and localization debugging. This telemetry system have been updated to allow the reception of debugging information of multiple robots during the game, when connected to the game controller network.

The simulation in Gazebo consists of a field with two goals, one ball and one Chape robot. There are a few significant differences from the real world, but it is accurate enough to test both the localization and the decision making algorithms.

5 Computer Vision

Our current approach on computer vision is based on Nimbro team's 2015 paper [9]. This approach uses a convex hull algorithm to detect the field and a hough line detector for the field lines. We do not detect the penalty cross due to the amount of false positives encontered. The ball and goalposts are detected using a convolutional neural network algorithm.

The first step on the vision pipeline is color segmentation on the image filtered by a Gaussian Blur. The color segmentation uses a color table generated by manually labeling pixels in various images and training a Neural Network classifier. To avoid the same color having two classes, for each labeled pixel the classes are given scores and the class with highest score is selected as the input of the neural network. Data augmentation is also used to improve performance.

To detect the white ball and the field's goalposts, we have a convolutional neueral network based in the model described in YOLO papers [17]. From the original FastYOLO architecture, we decreased the number of filters in each layer, for our robot has a limited processing power. Futhermore, we implemented the Deep Residual Learning for Image Recognition technique [11] to provide more accurate detections with little increase in computational complexity. As a result, our final convolutional network has a total of nine layers of 2D convolution, as shown in Fig. 3. The neural network was trained using Tensorflow [2] and Keras [5] libraries, and it uses Tensorflow [2] to run in real-time.

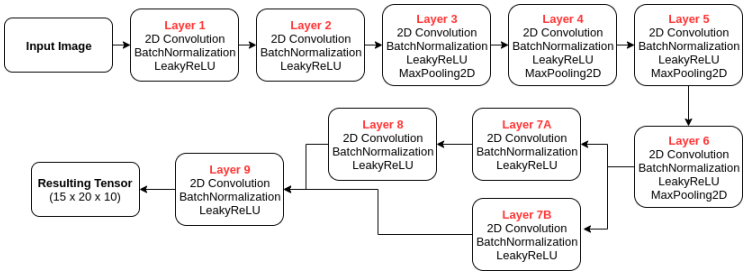

Fig. 3. Flow chart of the convolutional neural network we built. The image show also the shape of the output data for each layer. The names for each function used within layers are defined in Keras documentation [5].

Our algorithm reduces the size of the image from 640x480 to 160x120, then it creates a grid of 20x15 to return five values for each cell of this grid. The first value is the probability of the center of the object center's being located inside the cell, the second and third are the coordinates X and Y that represent the center of the ball according to the cell and the last two values are the height and width of the ball according to the whole image. Afterwards, we implemented a simple algorithm to find which cell contains the highest probability of having a ball, so as to know exactly its coordinates. For the detection of goalposts, our

vision algorithm has the same aforementioned logic, however instead of looking for the whole goalpost, we only detect the goalpost's vertical supports. Their location are essential to the robot's localization.

As an improvement from the RoboCup 2019 algorithm, we collected a new data set to train the neural network and expanded the use of data augmentation to increase the variety of images. Such a technique allowed us to decrease the number of false positives.

The transformation from the camera frame to the egocentric world frame proved to be a big challenge due to latencies associated with the system and the mechanical distortions. To solve this problem we consider the camera's timestamp as a reference for the image, and the joint and torso inclination information are synchronized with it. To achieve better results, we also compensate a fixed sensor delay in our torso observer model (2 ms), which was tuned manually.

The mechanical distortions are corrected by associating rotation offsets to the torso and camera, as well as translation offsets to the camera and torso height. These distoritions are calibrated using a routine consisted of the robot in stopped position moving its head detecting Aruco Codes known positions, as shown in Fig. 4. From last year's technique, we improved the result by Aruco codes instead of QR-Codes, because of better detection, and placing codes further away to improve precision at higher distances. Moreover, inspired by other teams, we reduced the number of offsets used for optimization.

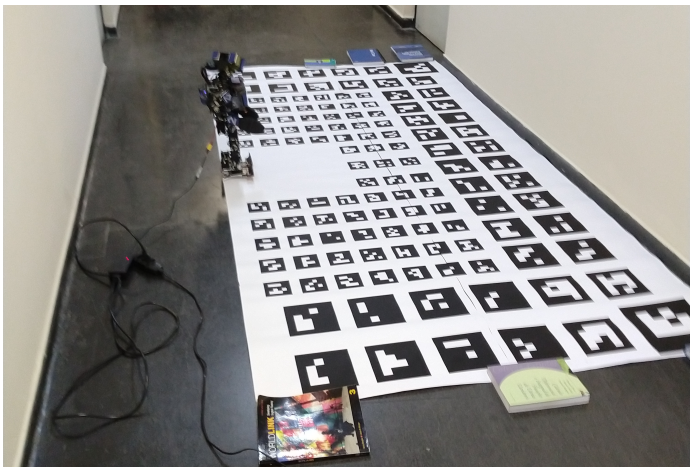

Fig. 4. Chape robot calibration using a Aruco code carpet.

6 Localization

To solve the global localization problem, we use a standard particle filter (Monte Carlo Localization), as described in [18, 15]. Localization is challenging in Hu-

manoid KidSize due to landmark ambiguity. The lack of unique features makes initializing the filter using a uniform distribution risky, since the filter may converge to the wrong side of the field. Therefore, at the beginning, our algorithm distributes the particles between the 4 possible starting poses in the soccer field, as stated in the rules [6]. Then, resampling is disabled while the head does a 180 degrees scan, accumulating information from the whole scan in the particles' weights before the first resampling. This proved very robust in initializing the filter to the correct robot pose.

Using the Gazebo simulator to guide our development, we made many optimizations to our algorithm throughout 2019: better head scan policy, parameter tunning (especially, higher odometry noise), making resampling less frequent etc. We currently have a working localization in simulation, but we still face issues regarding incorrect distance estimates in the real robot as commented in Sec. 5.

7 Motion Control

For walking and kicking, we use the ZMP-based algorithms described in [13]. We augment these algorithms with gravity compensation and a torso stabilization controller [14]. Since these techniques strongly rely on dynamics models, we developed a system identification technique to obtain a better mass distribution model experimentally through foot pressure sensors measurements [7] instead of using data from CAD files. However, we have not executed this process in the Chape robot yet.

The main innovation regarding motion control in 2019 was the development of a new kicking algorithm based on splines. In our previous motion, the kicking foot was constrained to be always parallel to the ground, which resulted in weak kicks. In the new kick, the x (forward), z (vertical) and θ (pitch) trajectories of the kicking foot are defined by points which specify the coordinate value at a given time. Then, the points are interpolated using separate cubic splines for each coordinate. The algorithm is only able to execute forward kicks for now, but we expect to generalize for other directions.

8 Conclusions and Future Work

This paper presented the recent efforts of ITAndroids in RoboCup Humanoid KidSize. The team evolved quickly since its first participation in RoboCup Humanoid KidSize in 2017. We currently have a robust hardware with 4 Chape robots. Our vision and motion control software also works well, but our localization system still lacks robustness. We intend to continue working on improving the system performance and robustness for RoboCup 2020. Moreover, we already designed a new prototype of a taller robot which we plan to build for the competition.

Acknowledgment

We thank our sponsors: Altium, Cenic, Intel, ITAEx, Mathworks, Metinjo, Micropress, Polimold, Rapid, Solidworks, ST Microelectronics, Wildlife Studios, and Virtual Pyxis. We also acknowledge Mathworks (MATLAB) and JetBrains (CLion) for providing access to high quality software through academic licenses.

References

1. *BOOST C++ Libraries*. `http://www.boost.org`.
2. Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
3. G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.
4. Dr. Philippe Capdepuy. DARwIn-OP Gazebo Simulation Model, 2016.
5. François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.
6. RoboCup Technical Commitee. RoboCup Soccer Humanoid League Laws of the Game 2018/2019, 2019.
7. Caroline C. D. da Silva, Daniela V. de Faria, Davi H. V. Barroso, Marcos R. O. A. Maximo, and Luiz C. S. Góes. Three-Dimensional Identification of a Humanoid Robot. In *Proceedings of the 2019 ABCM International Congress of Mechanical Engineering (COBEM)*, October 2019.
8. Dassault Systèmes. Solidworks, 2017.
9. H. Farazi, P. Allgeuer, and S. Behnke. A Monocular Vision System for Playing Soccer in Low Color Information Environments. In *Proceedings of the 10th Workshop on Humanoid Soccer Robots in conjunction with the 2015 IEEE-RAS International Conference on Humanoid Robots*, 2015.
10. Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.
11. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
12. The Qt Company Inc. Qt 5, 2017.
13. Marcos R. O. A. Maximo. Omnidirectional ZMP-Based Walking for a Humanoid Robot. Master's thesis, Aeronautics Institute of Technology, 2015.
14. Marcos R. O. A. Maximo. *Automatic Walking Step Duration through Model Predictive Control*. PhD thesis, Aeronautics Institute of Technology, 2017.
15. Alexandre Muzio, Luis Aguiar, Marcos Maximo, and Samuel Pinto. Monte Carlo Localization with Field Lines Observations for Simulated Humanoid Robotic Soccer. In: Latin American Robotics Symposium. In *Proceedings of the the 2016 Latin American Robotics Symposium (LARS)*, October 2016.
16. Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.
17. Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.
18. W. Burgard S. Thrun and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
19. T.R.O. Tonaco, D. Vacarini, C.C.D. Silva, M.R.O.A. Maximo, and M.A. Arbelo. Humanoid robot leg design. In *25th International Congress of Mechanical Engineering*, Uberlândia, MG, Brazil, 2019. COBEM.