ITAndroids Humanoid
Team Description Paper for RoboCup 2019

Alexander Justa, Arthur Azevedo, Davi Herculano, Daniela Vacarini, Francis Tanomaru, Igor Silva, João Filho, Lucas Steuernagel, Marcos Maximo, Miguel Ângelo, Misael Levi, Paulo Fonte, Rodrigo Aki, Thiago Tonaco, and Vinicius Araujo.

Autonomous Computational Systems Lab (LAB-SCA)
Aeronautics Institute of Technology (ITA)
São José dos Campos, São Paulo, Brazil
`{alexander.just,arthurazevedo41,herculanodavi,danivacarini,f.tanomaru, asilvaigor,joaocdffilho,lucas.tnagel,miguelangelo.dss,misaellvcc,prfc. ubr,krodrigo71008,tonaco.k.t22,viniciusdepadua6`
`mmaximo@ita.br`
`itandroids-humanoide@googlegroups.com`
`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 27 trophies in robotics competitions in the last 7 years. In 2017, the team participated in RoboCup Humanoid KidSize for the first time. Last year, the team placed top 8 in RoboCup KidSize after scoring a goal during the competition. This paper describes our recent developments for RoboCup 2019.

1 Introduction

ITAndroids is a robotics research group at Aeronautics Institute of Technology (ITA). The group is multidisciplinary and contains about 50 students from different undergraduate and graduate courses. We are considered a reference team in Brazil and Latin America, where we have won 27 trophies in the last 7 years.

Regarding our humanoid team, we participated in RoboCup KidSize for the first time in 2017 and placed top 16. In 2018, we placed top 8 in RoboCup KidSize after scoring a goal in our round of sixteen game against RoboFEI. Moreover, in the Latin American Robotics Competition 2018, we placed 1st in Humanoid KidSize, and placed 1st and 2nd in Humanoid Robot Racing using a Robotis OP2 and our custom made robot, respectively.

This paper presents our recent efforts in developing a humanoid robot team to compete in RoboCup 2019. The rest of the paper is organized as follows: sections 2 and 3 introduce the robot hardware. Sec. presents ideas for a new robot. Sec. 5 presents our software architecture and tools. Sec. 6 explains our computer vision system. Sec. 7 shows our localization approach. Sec. 8 discuss our motion control algorithms. Finally, Sec. 9 concludes and shares our expectations for the future.

2 Mechanics

The mechanical hardware is similar to the project presented last year [2], apart from some improvements that will be discussed in this section. Fig. 1(a) shows the robot's CAD, and Fig. 1(b) shows a picture of the manufactured robot.

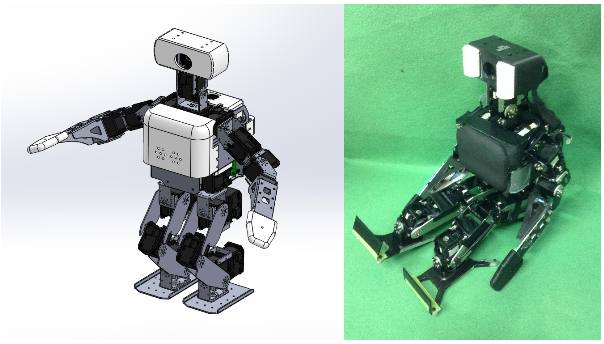

Fig. 1. Hardware of the Chape robot: a) CAD model of the robot; b) manufactured robot.

2.1 Head

As explained in [2], the head structure was not sufficiently resistant to withstand the falls fully and was slowly deforming. After some investigations, we concluded that the head position was causing it to be the first part of the body to hit the floor during a fall, due to the large camera offset.

As countermeasures, we reduced the camera offset, as showed in Fig. 2, and added a ABS cover with rubber foam attachments on the head and torso in order to lower the impact endured by the camera. This project was tested in the 2018 RoboCup Montreal and showed good results.

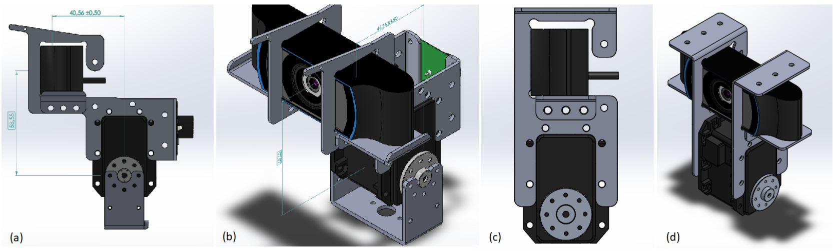

Fig. 2. Chape's head: (a) and (b) show RoboCup 2017 version with camera offset, (c) and (d) show the RoboCup 2018 version without camera offset

2.2 Legs

Until RoboCup 2018, the project was still adopting the leg design based on Robotis OP2 humanoid robot. Although it is a proven design, it requires precise metal folding to be manufactured. Since our team does not have the equipment required to fold metal sheets with such precision, the legs were misaligned. Another drawback was the thigh having two parts attached to each other by bolts, which would loose while the robot was walking, causing the leg to disassemble.

The team opt for a construction that would reduce the number of folds and the number of parts. This leg design was tested during the 2018 Latin American Robotics Competition (LARC) showing good results. We intend to continue the development of this design by optimizing the geometry of the parts. Fig. 1 shows the legs being developed.

2.3 Foot force sensor

The foot force sensor was designed to improve the robot walking stability. Some of the requirements were: (a) withstand forces of at least 100 N, (b) being capable of measuring forces up to 35 N, (c) easiness of manufacturing and assembling. We decided for a single part design, using 4 strain gauge (SG) half bridges. Fig. 3 shows the final design, which was obtained through topological optimization. More information can be found in [3].

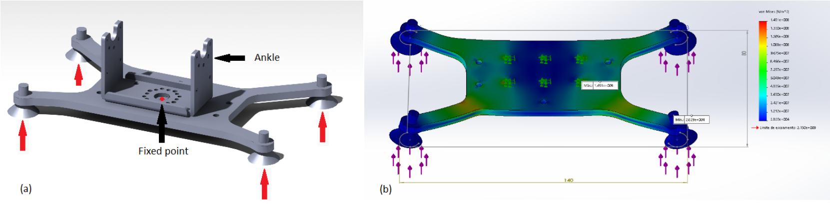

Fig. 3. Foot force sensor finite element analysis (FEA): (a) shows the constraints and applied forces (red arrows, 100 N); (b) FEA results.

3 Electronics

The electronic system of our Chape robot is in its third version. The architecture follows a modular approach, which is illustrated in Fig. 4.

The HPPU (High Performance Processing Unit) is an Intel NUC D54250WYK computer. The communication between HPPU and LPPU (Low Performance Processing Unit) is through USB 2.0. The custom made boards were developed using Altium and have the following functionalities:

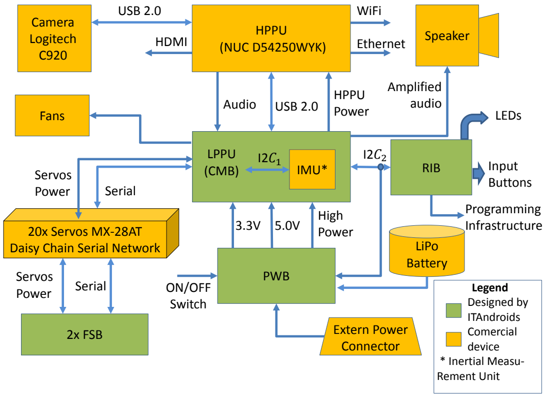

Fig. 4. Chape Electronics Architecture.

- CMB (Control and Monitoring Board): this board deals with low level hardware: servos communication, IMU data acquisition and interpretation, environment messages notification through the LEDs and buzzer, and servo power distribution control.
- PWB (PoWer Board): the battery and external power can be connected in this board. It measures power consumption and creates the regulated 3.3 V and 5 V power buses.
- RIB (Rear Interface Board): it is an interface with buttons to facilitate command input, with programming infrastructure and with diagnosis LEDs.
- FSB (Foot Sensor Board): board located in the robot's foot, equipped with strain gauges and an inertial unit to determine foot pressure and orientation. This board is being developed for RoboCup 2019.

In this version of Chape, the PWB and the CMB were redesigned with the goal of enhancing robustness. In PWB, the voltage conversion method was changed to a hybrid of switching and linear conversion, and had major changes in decoupling methods and electrical connection design. In CMB, the inertial unit was changed from AltIMU-10 v3 to MinIMU-9 v5, and the microcontroller was upgraded from the STM32 F103 to the STM32 F303 series. Also, USB connections and the programmer link connection were redesigned to avoid contact problems, and ESD protection was added to them.

With those upgrades, Chape now hardly presents malfunction due to electrical issues related to electrical contacts and missed sensor data communication. The next goals are to upgrade communication speed, adapt the firmware to the Dynamixel 2.0 Protocol for servos communication and fully implement the board monitoring, adding policies for issues like low battery voltage and overcurrent.

4 New Taller Robot

Our experience with our Chape robot taught us that we need at least two years of development of a new robot until reaching a robust hardware. Since the Humanoid League roadmap indicates that the minimum robot height in 2020 will be 60 cm, we plan to design and build a prototype of a new taller robot for RoboCup 2019, so we may have a robust hardware in RoboCup 2020.

We have already done a conceptual design by selecting components and defining the overall architecture. The height goal of the new robot is 65 cm. The robot will be drived by MX-28 servos at the neck and arms, and XM540-W150 servos at the legs. We intend to upgrade the main computer to an Intel NUC with a Core i7. Moreover, we expect to integrate the Intel Movidius neural compute stick to accelerate the processing of the convolutional neural network used in the computer vision. For the camera, we will use the Logitech BRIO. The electronic architecture will be similar to the one explained in Sec. 3, however we expect to divide the Dynamixel bus in three separate buses for faster servo communication and use RS-485 for the XM540-W150 servos.

5 Software Architecture and Tools

We use a module-based layered approach for code architecture. The main third-party libraries used are OpenCV (computer vision) [4], Eigen (linear algebra), Boost (general purpose), Qt (user interface), Tensorflow (machine learning) [1], and Keras (neural network) [5]. We heavily rely on the Robot Operating System (ROS) framework [12] and its related tools for testing and debugging purposes. A full Humanoid KidSize simulation was also developed using Gazebo [8], as shown in Fig. 5.

The simulation in Gazebo consists of a field with two goals, one ball and one DARwIn-OP 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. In the future, we plan to create a model of Chape for the simulator.

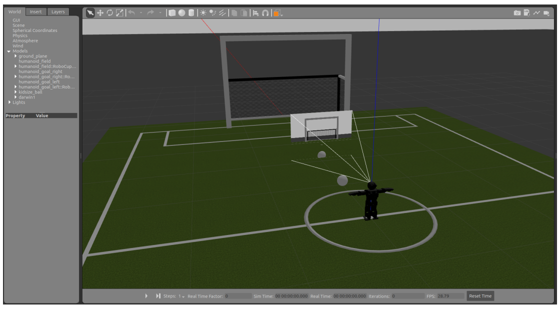

Fig. 5. A screenshot showing the robot in the humanoid field and the image captured by its camera.

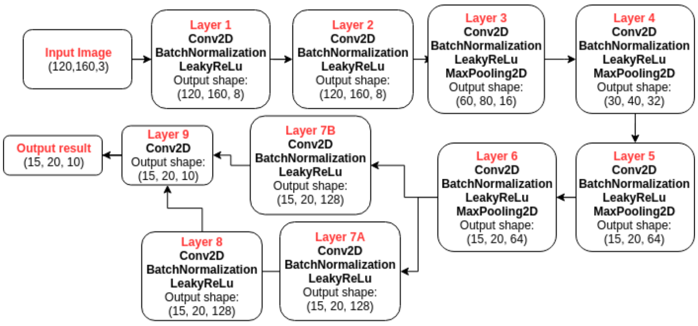

Fig. 6. Flow chart of the convolutional neural network we built. The image show also the shape of the output data for each layer. The names for each function used within layers are defined in Keras documentation [5].

6 Computer Vision

We have decided to abandon our previous algorithm which relied heavily on blob detection algorithms because of the difficulty in segmenting the grass field robustly. Our current approach is based mainly on a Nimbro team's 2015 paper [6]. We do not detect the penalty cross since we could not deal with false positives. The ball and goalposts are detected using a convolutional neural network.

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.

After the segmentation, green blobs are detected by merging horizontal run-lengths. The blobs that are bigger than a certain threshold have their run-lengths end points stored so the field boundary may be determined as the convex hull of these points. Then the field lines and field center are detected by performing canny edge on the V channel of the HSV image followed by a probabilistic hough line algorithm and filtering and merging algorithms. The filtering algorithms use the previously segmented image for color information. The lines intersections are also detected by checking for perpendicular angles between intersecting lines.

To detect the white ball and the field's goalposts, we made significant changes to our convolutional network. Nevertheless, it is still based in the model described in YOLO papers [13]. To deal with the detection of the goalposts (in addition to that of the ball), the new neural network has more layers and filters than the CNN we used in the RoboCup 2018. Futhermore, we implemented the Deep Residual Learning for Image Recognition technique [7] to provide more accurate detections with little increase in computational complexity. Our final convolutional network has a total of nine layers of 2D convolution, as Fig. 6 shows.

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 errors in the kinematic model of the robot due to mechanical imperfections. To solve the delay problem, we use the camera's timestamp to synchronize image and body transform. To mitigate errors in the kinematic model, we estimate the torso orientation by fusing the acceleration and angular velocity measured by the IMU through an Extended Kalman Filter (EKF).

We were still observing consistent estimate errors, then we decided to associate translation and rotation offsets to the torso and the camera. These offsets are calibrated using an experiment where the robot moves its head to detect QR-Codes in known positions. Then, the offsets are optimized with CMA-ES in order to obtain the values which result in the best fit to the known positions.

7 Localization

To solve the global localization problem, we use a standard particle filter (Monte Carlo Localization), as described in [14, 11]. Despite our team have already implemented localization in Soccer 3D [11], we have been struggling to make localization work robustly in our real humanoid robot. In 2018, we developed a localization simulator based on Qt so we could debug the algorithm faster. After many bug fixes, the algorithm itself is working correctly in this simulator. However, as commented in Sec. 6, we still faced many issues throughout 2018 due to incorrect distance estimates, which were making the algorithm diverge in the real robot. Our localization is now able to keep track of the correct robot's pose estimate if the robot is constantly seeing field features, but it usually loses this track if the robot focus on the ball for a long time. To countervent this problem, we are trying to improve our odometry model and our head scan behavior.

8 Motion Control

For walking and kicking, we use the ZMP-based algorithms described in [9]. We augmented these algorithms with gravity compensation, which employs the knowledge of the robot's mass model to compensate torques due to gravity in a feedforward manner. Since we used the CAD data for the mass model, we have considerable model mismatch. Therefore, we intend to use system identification based on feet pressure sensor measurements to obtain a better model.

In 2018, we also implemented a torso controller which greatly improved the robot stability when walking on artificial grass. This controller uses the torso orientation and angular velocity as feedback, and considers proportional-derivative controllers to drive the hip, knee, and ankle joints in order to keep the torso upright. To compute the gains, we modeled the robot as three-link and two-links manipulators in the sagittal and coronal planes, respectively. For more information, please refer to [10].

9 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 in improving the system performance and robustness for RoboCup 2019. Moreover, we expect to design and build a new prototype of a taller robot for the competition.

Acknowledgment

We thank our sponsors Altium, Intel Software, ITAEx, Metinjo, Micropress, Polimold, Rapid, Solidworks, ST Microelectronics, and Virtual Pyxis. We also acknowledge Mathworks (MATLAB), Atlassian (Bitbucket) and JetBrains (CLion) for providing access to high quality software.

References

1. Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
2. Arthur Azevedo, Davi Herculano, Daniela Vacarini, Gabriel Crestani, Igor Silva, Jo ao Filho, José Roberto Junior, Lucas Steuernagel, Marcos Maximo, Miguel Ângelo, Rodrigo Aki, and Samuel Pinto. ITAndroids Humanoid Team Description Paper for RoboCup 2018. 2018.
3. Arthur Azevedo, Daniela Vacarini, and Maurício Vicente Donadon. Sensoriamento dos Pés de um Robô Humanoide Estilo KidSize [in Portuguese]. In *ENCITA*, São José dos Campos, 2018.
4. G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.
5. François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.
6. 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.
7. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
8. N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.
9. Marcos R. O. A. Maximo. Omnidirectional ZMP-Based Walking for a Humanoid Robot. Master's thesis, Aeronautics Institute of Technology, 2015.
10. Marcos R. O. A. Maximo. *Automatic Walking Step Duration through Model Predictive Control.* PhD thesis, Aeronautics Institute of Technology, 2017.
11. 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.
12. 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.
13. 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.
14. W. Burgard S. Thrun and D. Fox. *Probabilistic Robotics.* MIT Press, 2005.