






MRL Champion Team Paper in Humanoid TeenSize League of RoboCup 2019

Hamed Mahmudi^(✉) , Amir Gholami , Mohammad Hossein Delavaran ,
Soheil Khatibi, Saeid Bazargan, Milad Moradi, Bita Alaei, Arash Rahmani,
Kazem Firouzmendi Bandpey, Peyman Fallahzadeh, and Meisam Teimouri^(✉) 

Mechatronics Research Lab, Department of Computer and Electrical Engineering,
Qazvin Islamic Azad University, Qazvin, Iran
{mahmudihamed,m.teimouri}@qiau.ac.ir
<http://hsl.mrl.ir>, <http://hsl.mrl.ir>

Abstract. RoboCup uses soccer competitions as a research area to promote robotics and artificial intelligence. The ultimate goal of the RoboCup is to develop a team of humanoid robots that can win against the human world champion soccer team in 2050. This paper presents the approach of the MRL TeenSize humanoid team to improve its hardware and software which leads to achieving first place in soccer competition and Drop-In games, second place in the technical challenge and third-best humanoid award.

Keywords: Champion team paper · RoboCup · Humanoid league · Bipedal locomotion

1 Introduction

The MRL project was started in 2003 in the Mechatronics Research Laboratory at Islamic Azad University, Qazvin branch looking onward to enhance the knowledge of robotics. The MRL humanoid soccer team is aimed to develop a humanoid platform for research and education. Our humanoid soccer-playing team has participated in the RoboCup Humanoid KidSize since 2011 [1–3]. As RoboCup uses soccer as a research area to develop a team of humanoid robots, we decided to participate in the TeenSize category from 2018 in order to move toward the goal of RoboCup and face new challenges. MRL earned first place in the TeenSize main league competition as well as Drop-In games and won second place in TeenSize Technical challenges and third-best humanoid award in the RoboCup 2019, Sydney, Australia.

The rest of this paper can be summarized in the following. In Sect. 2 an overview of the system is described. The electronic parts are discussed in Sect. 3. In Sect. 4 recent development of our software including methods of developing for visual perception and robot behavior are clarified. We will continue with solutions developed to overcome the technical challenges and advances made to increase the efficiency of robot kicking in Sect. 5. Finally, the paper is concluded in Sect. 6.

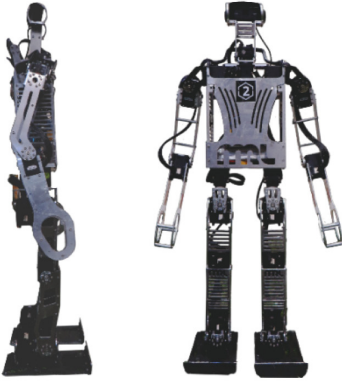


Fig. 1. Left: Amir humanoid TeenSize robot. Right: MRL-HSL team at RoboCup 2019 competition Sydney, Australia.

2 Overview of the System

Amir is our new humanoid robot with 20 degrees of freedom (Fig. 1). Our robots have a well-known 20 degree of freedom structure with a height of 83 cm and a weight of 6.6 kg. All joints are equipped with Robotis Dynamixel MX series actuators. We have used six Dynamixel MX-106 for each leg, three Dynamixel MX-64 for each arm and two Dynamixel MX-28 in neck and head. The robot is powered by a 3-Cell, 5000mAH LiPo battery. The main processing unit is an Intel NUC 7th generation and power management is done by our self-constructed controller board which will be described below.

Visual perception is done by a Logitech C920 normal webcam with a 78° field of view. All mechanical parts of the robot are made of aluminum alloy 6061. We manufactured robot components by utilizing CNC Milling in order to increase accuracy.

Our software architecture is based on the UPennalizers RoboCup released code [4]. The Vision, world model, and behavior modules are completely rewritten and the walk engine is enhanced to address disturbances more efficiently.

3 Electronic

3.1 Control Manager

Last year, we designed our control manager board (Fig. 2 left) that was responsible for power supply, actuator communication and attitude estimation of the robot. In MRL-CM1, we divided the actuators lane into three separate lanes to reduce the load. So the transmission speed between the mainboard and actuators located in different chains increased. However, we need to read sensors data with a higher rate to get smoother motion controllers. Reading actuators and sensors data through the same BUS makes the rate of reading constrained by the speed of actuators reading that is considerably lower than sensors reading. Moreover adding extra measuring instruments to the system makes it even slower. To cope with this problem, a new control manager board (Fig. 2 right) is designed that consists of two separated boards (actuators and sensors board).

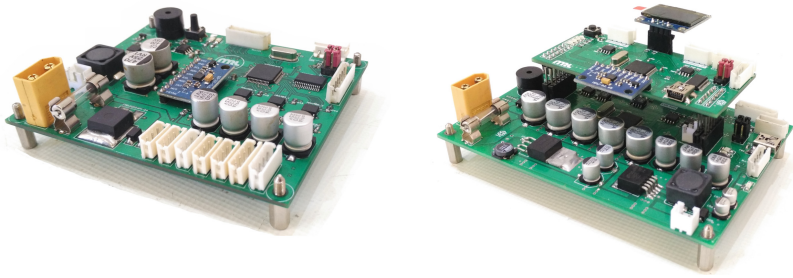


Fig. 2. Left: MRL-CM1. Right: New control manager board (MRL-CM2).

Each board is individually connected to the mainboard and transfers the data using the standard Dynamixel packets. The block diagram of the new system is demonstrated in Fig. 3.

Sensor Board. The processing unit of the actuators board is an ARM STM32F405VE microcontroller. The firmware is developed under standard HAL drivers [5]. This board is equipped with 2 inertial sensors (gyroscope and accelerometer). Also, it has a lane for communicating with strain gauges. Since its task is just to read data sensors and it has no communication with actuators, we can read data faster. An OLED display is attached for monitoring and debugging. In addition, it should be mentioned that another aim of this board is estimating roll, pitch, and yaw employing a complementary filter [6].

Actuator Board. The tasks of power management and actuators data streaming are accomplished by this board. The processing unit of the actuator board is the same as the sensor board. This board is designed to communicate with both TTL and RS-485 protocols simultaneously, but because of our available R series actuators of Dynamixel and efficiency of RS-485 over long distances between legs actuators, we used RS-485

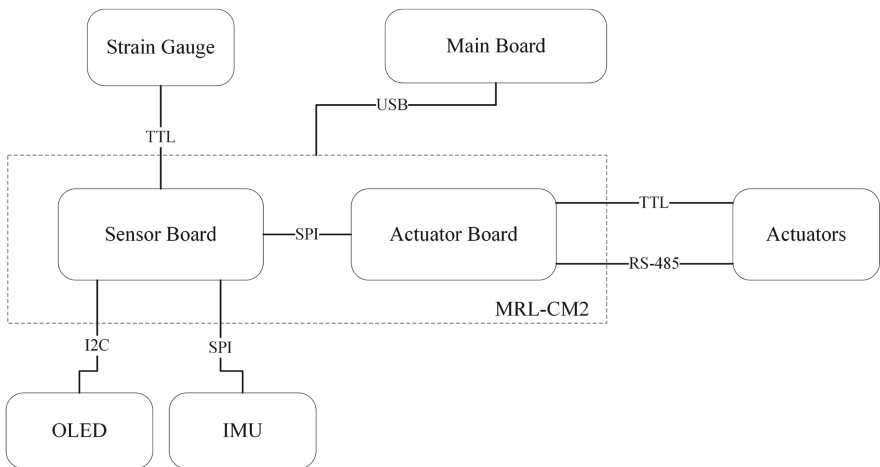


Fig. 3. MRL-CM2 Communications block diagram

protocol as our serial communication. Similar to MRL-CM1, to reduce the heavy load of data transfer, we have divided the communication lane into five independent lanes, each one has its own FIFO queue.

As well, the actuator board is able to supply the required power for the entire robot. We used a DC-DC boost converter in MRL-CM1 which provided 18V-5A at a frequency of 350 kHz, but there were some problems. It was used only to supply mainboard voltage and we got the actuator voltage directly from the battery and we had no way to change and control it. So, in this version of the board, we used a DC-DC buck converter to supply the voltage and current needed for the mainboard and actuators, as explained below.

DC-DC Buck Converter. The battery is the main input power supplier of the robot and its voltage drops over time. By implementing this system, we can keep motors voltage constant and apply desired voltage to the motor in different situations, such as higher voltage than usual when the robot is kicking or when the robot is standing up. Moreover, the process of motion calibration of the robot becomes simpler.

Buck converters are switched-mode step-down DC-DC converters; the output voltage of buck converters is a function of the PWM duty cycle which controls the MOSFETs status. The output signal is an average of the PWM square wave [7].

The architecture of the closed-loop DC-DC buck converter is depicted in Fig. 4. It is consisting of a power stage and a controller. The converter power stage consists of a second-order LC low-pass filter and it is prone to become unstable. In this design, we have to satisfy two important characteristics: the capability of rejecting disturbances and stability in all operating conditions. So a PID controller is implemented to enhance system stability and adjust the output voltage [8].

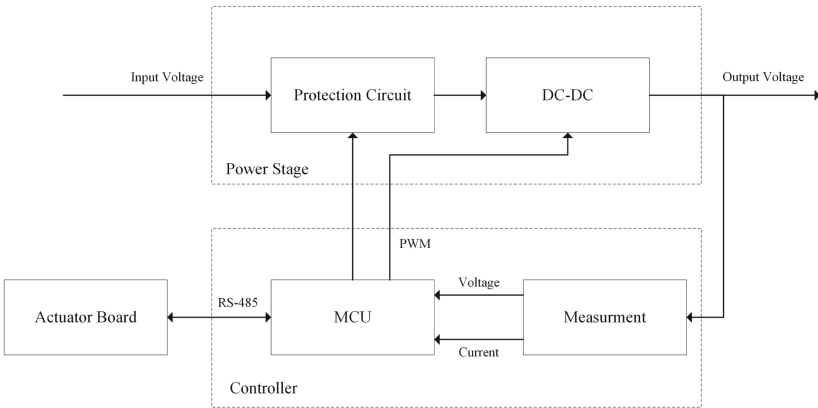


Fig. 4. Block diagram of the designed DC-DC buck converter.

4 Software

4.1 Visual Perception

Vision is the most important source of perception in a humanoid soccer playing robot. In this section, we briefly summaries the pipeline of our vision system and investigate the details of our segmentation module in the following subsection.

Like most of the participating teams, we use a monocular vision system. The captured image at first is fed to a segmentation module to perform semantic segmentation. Based on this segmented image the boundary of the field is determined. Then all object detection algorithms are applied only on the pixels located in the field boundary. The field lines and their intersections are detected using the Hough Transform [2]. For ball detection, first, some coarse regions of interest that may contain a ball are extracted using the segmentation map and camera matrix. Then each region is fed to a deep convolutional neural network to predict whether the region contains a ball or not and estimate the exact location of the ball in that region. The details of our ball detection approach are described in [9].

Semantic Segmentation. In RoboCup 2019 an important step toward outdoor conditions was taken by advancing to natural light in drop-in matches. Most humanoid teams used pixel-level image segmentation as the first step of vision pipeline, but due to the continuous changes of illumination in natural light, methods such as thresholding and lookup tables often yield unreliable and ineffective results. To overcome this challenge, we have used semantic segmentation [10] as the first step of our vision pipeline. This segmentation module uses a deep fully convolutional neural network which we will briefly describe in the following subsections.

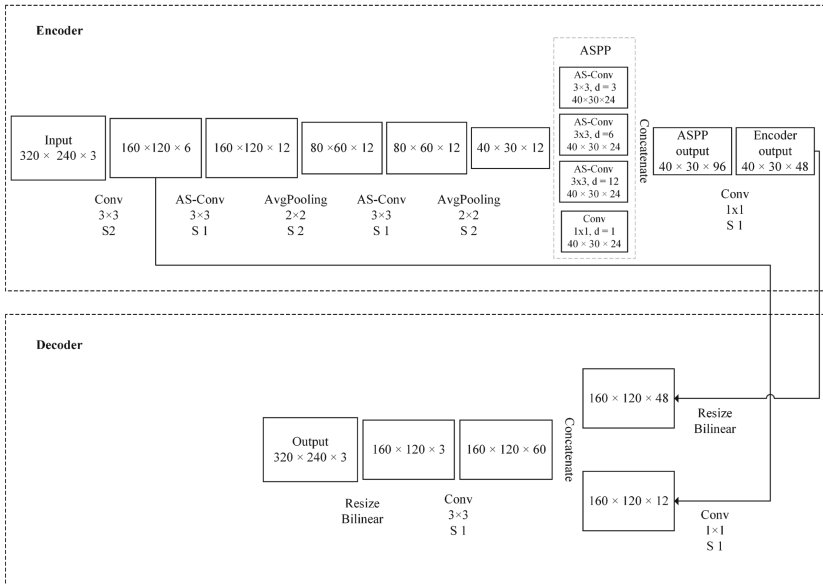


Fig. 5. Architecture of the segmentation network.

Model Architecture. Since most humanoid robots carry only a CPU as the processing units, semantic segmentation as a dense prediction task could be computationally expensive. By considering the limitations of our robots we have designed a segmentation model made an efficient trade off between accuracy and computational cost.

The designed model is highly inspired by DeepLab v3+ [10]. Following DeepLab our model has two parts: an encoder for downsampling and a decoder for upsampling. With the same scheme as DeepLab v3+, the encoder has a small base network followed by an Atrous Spatial Pyramid Pooling (ASPP) [10] module. In the decoder, low-level features have been pulled from the base network of the encoder and then concatenated with the upsampled output of the encoder. Finally, to produce the un-normalized class probabilities, a convolutional layer is applied. The output must be upsampled once more to match the input resolution. For the sake of reducing the computational cost of inference in most layers of the network, Atrous Separable Convolutions [10] (AS-Conv) has been used instead of the traditional convolutions. The details of the network and each layer is demonstrated in Fig. 5.

Data Set. Training of the described segmentation module requires samples with pixel-level labels and manual labeling of this data set is a challenge. Therefore, we have created a tool to ease up the labeling and save annotation time.

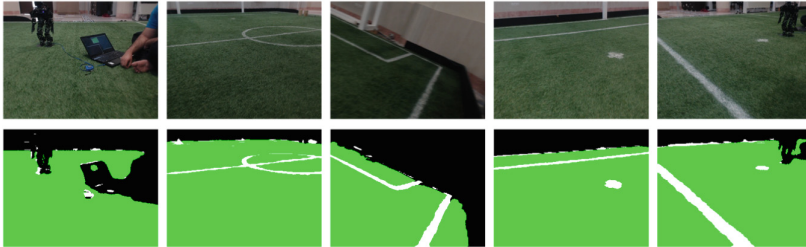


Fig. 6. The segmentation results on a few samples captured in MRL Labs at different times of a day.

Using this tool, we have labeled 700 samples collected in several RoboCup competitions and MRL labs with various light conditions. Labels of this dataset have three classes including grass, bright object (lines and white parts of the ball), and background. By massive augmentation of this data set including illumination, flipping, and cropping we were able to achieve more accurate results. But we are also hoping to include more samples and class labels in the future and increase the comprehensiveness of the data set.

Training. In training Adam optimizer has been used to minimize a SoftMax cross-entropy cost function. By empirical results, we have found $1e-3$ as the starting learning rate and used a piecewise constant schedule to decay it. As mentioned before massive augmentation of the data set was also necessary to achieve generalization.

Training of the network takes about 30 min on an Nvidia GTX 1070 TI GPU. Also, the entire training code was implemented using TensorFlow and will be available on the MRL-HSL GitHub page¹.

In RoboCup 2019 we used the trained model for all matches, and we achieved accurate results. Also, the trained model has been tested and used at various times of the day with different light conditions in our lab. As it is shown in Fig. 6 the segmentation results are accurate with fine edges.

The entire segmentation module can be processed as fast as 22 frames per second on an Intel Core i3 7th gen CPU alongside with other operations needed for a real game of play.

It is obvious that the work is not finished. During RoboCup 2019 we experienced some disturbances caused by the effect of direct sunlight on the grass. However, using multi-task learning other vision tasks of the robot can share the base network with the segmentation module. Furthermore, adding more classes to segmentation and making a benchmarking data set for the base of compression can be a good path toward the future of this work.

4.2 Behavior

Role Assignment. Our role set consists of four roles named Goalkeeper, Attacker, Defender, and Supporter which have their own responsibility description. The Goalkeeper is a static role which does not change while the others do according to the game situation. Hence the Goalkeeper stands in the middle of its goal and when the ball enters the goal area and no other teammates are close to it, the Goalkeeper comes to clear it. We assign a role to other robots in the field considering two factors. First, the chance of approaching the ball that is a combination of ball model probability and cost of reaching it facing the opponents goal. Second, distance to the own goal. The Attacker role is assigned to the robot which has the biggest chance to approach the ball. Then the robot that is not an Attacker and is closer to its own goal than others will be Defender. It should block the way that ball can be kicked toward our goal and be ready to chase the ball when needed. Finally, the last one will be Supporter. Supporter is a quite important role which can drastically improve the performance of each team by supporting both attacking and defending through apt positioning. It is quite dependent on accurate localization and error can cause serious problems like collision that impairs the team performance rather than improving it. In recent years we have been able to practice this role due to significant improvement in our localization accuracy. When our Attacker detects the ball the Supporter starts following it with a certain shift with respect to it in order to become Attacker in the case that the current Attacker loses the ball. This shift can vary according to different positions of the ball and the Attacker [11]. When the Attacker does not detect the ball (which means that none of the robots detect it), the Supporter plays like a Defender. It covers the other side that the Defender does not cover so that the goal area becomes safe. On the other hand, the Attacker can search for the ball not being concerned about conceding a goal.

¹ <https://github.com/mrl-hsl>.

Active Vision. Due to the limited field of view, our robot is not able to see every important observation in a specific head position. To avoid this problem, some teams have used cameras with a wider field of view and some others have used cameras with higher fps so that the robot can move its head fast and observe all important things around. But we have started to develop an active vision algorithm in which the robot goes through some predefined actions. The goal is to improve the models of self-localization and the ball. The uncertainty of a belief given by Shannon entropy can be a good benchmark to determine whether an action improves the models [12].

Calculating the entropy after doing an action consists of updating the models for all of the visible observations. The visibility of observations is determined with a polygon formed by projecting four corners of the image on the field. But we experienced serious problems while implementing this algorithm. Firstly, issues like losing the ball can occur because of projection and localization errors. Secondly, We do not have any feedback from the environment after taking action. For instance, If the robot can not detect a landmark due to the occlusion, there is no way to find it out. To solve these problems we are investigating a deep reinforcement learning network that takes a sequence of raw images as input and selects the best action to move the head through the right position and fix the error.

5 Motion Control

5.1 Push Recovery by Stepping

In the previous year, the hip-knee-ankle strategy for push recovery was applied using the PID controller for each joint [13]. Also, for more stability, the arm controller was added to the hip-knee-ankle strategy. This made the robot to resist more against pushes coming from the front, rear and sides of the robot [14].

For RoboCup 2019 our push recovery approach is extended by adding a simple stepping method. This method is activated against strong forces coming from the front, rear, and sides of the robot. To recover the stability by stepping method, an Inertial Measurement Unit (IMU) is required to get angle feedback of the upper body. When the humanoid robot is impacted by external forces, the ground projection of the CoM position starts to leave the support polygon, so there is a possibility for the robot to fall down. If the ground projection of the CoM position exceeds a predefined threshold, the robot increases the velocity of stepping in the direction of pushes and walks forward, backward or sides more quickly (Fig. 7). When CoM returns to the desired state, the velocity of stepping comes back to the normal condition.

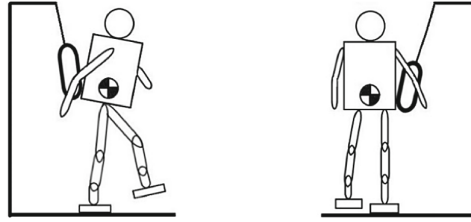


Fig. 7. Push recovery by stepping meth

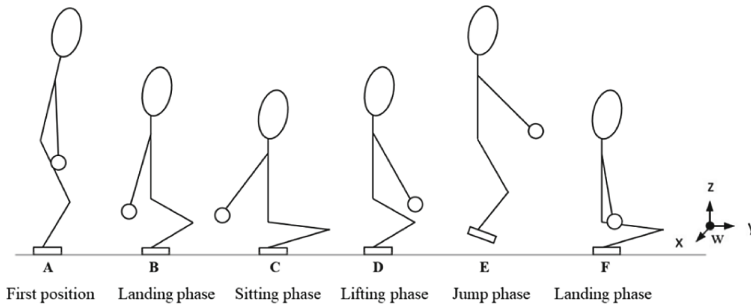


Fig. 8. Jump phases

5.2 High Jump

In this section, a motion planning method for the humanoid robot is proposed which can perform the vertical jump. A stable vertical jump is accomplished by maintaining the ground projection of the Center of Mass (CoM) in the support polygon. To satisfy the stability we have used arms. Therefore, the robot pulls back its arms while it is sitting down and subsequently pulls forward while it is jumping in the vertical direction. On the other hand, movement of the arms helps to make the robot do a higher jump [15].

The states of the robot in the jumping phases are shown in Fig. 8. The first phase of jumping (i.e. phase A) starts while the robot standing in an upright position. The robot moves downward by bending its knees and ankles (phase B). When it reaches to sitting position (phase C), stay in this position for 0.3 s. Take-off from the ground needs a fast response to reach the desired trajectory with overshoot. So the robot extends its lower body in a moment (phase D) and as a result, the robot takes off the ground (phase E). Landing without losing stability is remarkably important, hence the robot must be kept stable in this situation and the controller will help the robot to do this job correctly (phase F).

The designed trajectory is approximated from phase A to D using two linear equations and from phase D to F by the Fourier series as shown in Fig. 9. During the lifting phase, controlling the CoM position is the main problem and in order to solve this, a PD controller is designed to control the CoM position. The gains of the PD controller is tuned with Ziegler–Nichols method.

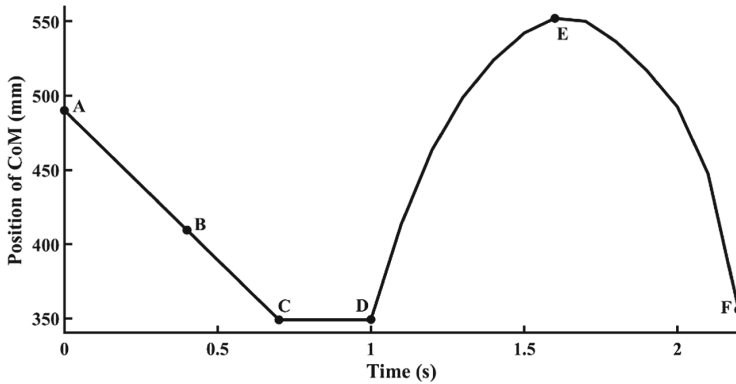


Fig. 9. The desired trajectory for jumping phases

5.3 Kicking

Kicking is a complex and vital motion in the humanoid robot league. Consequently, the MRL-HSL kicking trajectory has undergone some changes to achieve a higher displacement of the ball. The kicking trajectory of the robot is a combination of some simple motions. Thus the whole motion is divided into some simple phases such as lifting, kicking, landing and stabilizing the CoM of the robot. Furthermore, the upper body has the opposite movement compared to the swing foot which causes to provide more power to the ball (Fig. 10). The trajectory of kicking is designed by using [16].

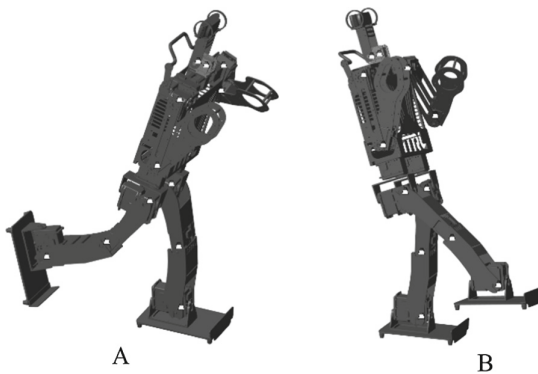


Fig. 10. Movement of body during kicking

6 Conclusion

In this paper, we have presented the specifications of the hardware and software of the MRL TeenSize humanoid robot system developed under the RoboCup 2019 rules.

The mentioned improvements including new control manager board, buck convertor, semantic segmentation, deep convolutional ball detector network, high jump motion and stepping push recovery strategy lead us to take four awards in Robocup 2019 competitions. In soccer competition we got first place by winning all matches. In drop-in games with 11 points we won the first place award and in the technical challenge by taking 14 points from two push recovery and high jump challenges second place is achieved. Overall we honored as the third-best humanoid team.

Also, We use our self-designed and self-constructed robots and we are working on this platform with some interested researchers and students optimizing vision, motion control, world modeling, behavior, and embedded control board.

References

1. Teimouri, M., et al.: MRL Team Description Paper for Humanoid KidSize League of RoboCup 2017. Mechatronics Research Lab, Department of Computer and Electrical Engineering, Qazvin Islamic Azad University, Qazvin, Iran (2017)
2. Teimouri, M., et al.: MRL Team Description Paper for Humanoid KidSize League of RoboCup 2018. Mechatronics Research Lab, Department of Computer and Electrical Engineering, Qazvin Islamic Azad University, Qazvin, Iran (2018)
3. Mahmoudi, H., et al.: MRL Team Description Paper for Humanoid KidSize League of RoboCup 2019. Mechatronics Research Lab, Department of Computer and Electrical Engineering, Qazvin Islamic Azad University, Qazvin, Iran (2019)
4. <https://github.com/UPenn-RoboCup/UPennalizers>
5. ST Microelectronics: RM0090 Reference manual STM32F405xx. STM32F407xx, STM32F415xx and STM32F417xx advanced ARM-based
6. Allgeuer, P., Behnke, S.: Robust sensor fusion for robot attitude estimation. In: 2014 IEEE-RAS International Conference on Humanoid Robots, pp. 218–224. IEEE (2014)
7. Kazimierzczuk, M.K.: Pulse-Width Modulated DC-DC Power Converters. Wiley, Hoboken (2015)
8. Guo, L.: Implementation of digital PID controllers for DC-DC converters using digital signal processors. In: 2007 IEEE International Conference on Electro/Information Technology, pp. 306–311. IEEE (2007)
9. Teimouri, M., Delavaran, M.H., Rezaei, M.: A real-time ball detection approach using convolutional neural networks. In: Proceedings of 23rd RoboCup International Symposium. Sydney, Australia (2019)
10. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818 (2018)
11. Petersen, K., Stoll, G., von Stryk, O.: A supporter behavior for soccer playing humanoid robots. In: Ruiz-del-Solar, J., Chown, E., Plöger, P.G. (eds.) RoboCup 2010. LNCS (LNAI), vol. 6556, pp. 386–396. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20217-9_33
12. Seekircher, A., Laue, T., Röfer, T.: Entropy-based active vision for a humanoid soccer robot. In: Ruiz-del-Solar, J., Chown, E., Plöger, P.G. (eds.) RoboCup 2010. LNCS (LNAI), vol. 6556, pp. 1–12. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20217-9_1
13. Teimouri, M., et al.: MRL Team Description Paper for Humanoid TeenSize League of RoboCup 2018. Mechatronics Research Lab, Department of Computer and Electrical Engineering, Qazvin Islamic Azad University, Qazvin, Iran (2018)

14. Mahmudi, H., et al.: MRL Team Description Paper for Humanoid TeenSize League of RoboCup 2019. Mechatronics Research Lab, Department of Computer and Electrical Engineering, Qazvin Islamic Azad University, Qazvin, Iran (2019)
15. Sakka, S., Yokoi, K.: Humanoid vertical jumping based on force feedback and inertial forces optimization. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 3752–3757. IEEE (2005)
16. Gholami, A., Moradi, M., Majidi, M.: A simulation platform design and kinematics analysis of MRL-HSL humanoid robot. In: Proceedings of 23rd RoboCup International Symposium, Sydney, Australia (2019)