Virtual RoboCup Asia-Pacific 2020

Team Description Paper

League Name:	Robocup junior humanoid soccer
Age Group:	Junior
Team Name:	Robokit
Team Website:	robokit.su
Participants Name:	Daniel Babaev, Matvey Ivashchenko, Korostinsky Gleb, Vladimir Tkach
Mentor Name:	Dmitriy Ustinov
Institution:	Phystech Lyceum
Region:	Dolgoprudny, Russia
Contact Person:	Dmitriy Ustinov
Contact Email:	dimaystinov@gmail.com
Date:	09.09.2020

Virtual RoboCup Asia-Pacific 2020

Team Description Paper

RCAP HUMANOID JUNIOR

Daniel Babaev, Matvey Ivashchenko, Korostinsky Gleb, Vladimir Tkach

Robokit, Phystech Lyceum, Russia

1. Abstract

Presentation of Team, Robot, Strategy, Records of ROBOKIT team from Phystech Lyceum, Moscow, Russia to Robocup Asia – Pacific Virtual competition.

2. Introduction

Our teams background is Phystech Lyceum in Moscow Region.

Kapitza Phystech-Lyceum (the official name is ANOO “Phystech-Lyceum” named after P.L. Kapitza) is a state owned comprehensive co-educational boarding and semi-boarding school located in Dolgoprudny (a town near Moscow, Russia), which incorporates primary, secondary and high school.

The Lyceum is aimed at giving students deep knowledge in the fields of mathematics, physics, astronomy, information technology and biology. Now the Lyceum is rated as the best school in Moscow Region and is included one of the best top ten schools of Russia. Our Lyceum students are winners of regional and all-Russia Olympiads in different subjects and some of them participate in international contests and conferences, with some of them becoming winners and prizers. Lyceum graduates enter best Russian universities and some of them continue their education in the universities all over the world.

The school started working in 2014, and now there are over 600 students in it. In September 2018 a new dormitory building for 200 students was opened on campus. The school development concept includes three notions: traditions, talents, technologies. The present day Lyceum was built on the basis of Dolgoprudny Lyceum №11 “Phystech-Lyceum” by a group of enthusiasts-lecturers and professors of Moscow Institute of Physics and Technology (MIPT) in 1991. The school engages best teaching staff including qualified school teachers and university lecturers; many of them become scientific advisors of the students’ projects and experimental works. The conferences “I am a Young Researcher” and “Start to Innovations” are held annually and all the students take part in them.

The school has well-equipped laboratories where the students carry out experiments in Physics and Chemistry; the laboratory of Chemistry has got devices for up-to-date molecular-biological manipulations, there is a laminar and a greenhouse for experiments in Biology and a telescope for the students keen on Astronomy.

Our team in humanoid robotics named Robokit were formed on 2019.

First lessons we were studied python with our trainer Dmitriy Ustinov. We have studied functions and capabilities of OpenMV like detecting ball by color and circumference. Later our robots arrived to us and we started to built them. We changed the robot servo for better one and also changed their plastic details. After assembly we taught robot come to the ball and kick it. Robots use their camera to detect the ball. Then, the built-in microcontroller Kondo was responsible for the movements of the robot. Also, for the convenience of programming and debugging algorithms, we began to use a program for simulating CoppeliaSim Edu. We created real models of robots in this program that actually see the surrounding space and analyze the picture using python. Using Coppelia we refine our program's and robots become much smarter. So you can watch our results on video here https://goo.su/2amD

We used Localization, Moving and Vision libraries that were written by MIPT students under mentorship of Azer Babaev in order to focus on game strategy and give our students the opportunity to implement their ideas. Student's decided their own tasks to improve robot's game.

Fig 1. Team Photo
Matvey Ivashchenko, Korostinsky Gleb, Azer Babaev, Daniel Babaev, Dmitriy Ustinov, Vladimir Tkach

Our website is http://robokit.su/

Korostinsky Gleb (15 years old) has designed ball approach algorithm. Robot must find the coordinate of the point to which it should go our robot to attack the ball. He have to write the module, that can calculate direction of walk and how much steps robot have to get to the point. For this program we used geometry.

He also did:

1) Measured parameters of court

2) Founded direction axles on the field

3) Started of axles placed in the center of the court

On the RoboCup Asia-Pacific 2019 competition our robots walked slowly and used really bad strategy of approach to the ball. Soccer found the ball and counted the distance to it, calculated the steps. Started his walk to the ball forward. After he passed half of his way he was calculating the remaining distance, adjusted its direction and continued to do so until the distance become less than the specified constant. After that, he would stop, adjust his course, and continue to approach the ball in small steps. In total, the Soccer walked strictly in straight lines.

The main factor, that was affecting the speed were little steps near the ball, the robot did not reach its destination for the first time, so it went on and on in small pieces.

Vladimir Tkach (15 years old) has designed the algorithm of choosing the type of walking and approaching to the ball. Now the robot must to configure his course to the ball, calculate the quantity of full and little(remaining) steps, firstly the Soccer walks full, and then little steps and at the end tunes to the angle, on which the robot must be turned near the ball. Thanks to that, the robot began to walk much faster, and the accuracy of the kicks didn’t suffer.

The main advantage of the new algorithm is that the player doesn’t make a large number of small steps near the ball.

Daniel Babaev (13 years old) has designed algorithm that finding ball on the field. Last algorithm had disadvantages, because robot saw in the form of a ball all the round objects of orange color. For example, orange thing or orange clothing outside the field. In new algorithm he added special rectangles that outline square and in case camera see in at least one of these rectangles green pigment, than considers that ball on the field. If not, then it takes this for a false object and ignore it.

Matvey Ivashchenko (15 years old) has designed goalkeeper game. Old strategy:

1) The goalkeeper searches for the ball on the field up to a certain distance

2) If robot finds the ball he immediately falls into the split for 8 seconds

3) Get up and continue looking for the ball

On a penalty kick, the robot did the same.

This was ineffective, since getting up from the split takes a lot of time, and during this time opponents could bypass the goalkeeper and score a goal.

Therefore, in his task, He must write the optimal actions of the goalkeeper in various situations on the field.

To do this, He divided part of the field into 8 sectors, and when the ball is in one of these sectors, the robot must perform special actions to protect the gate.

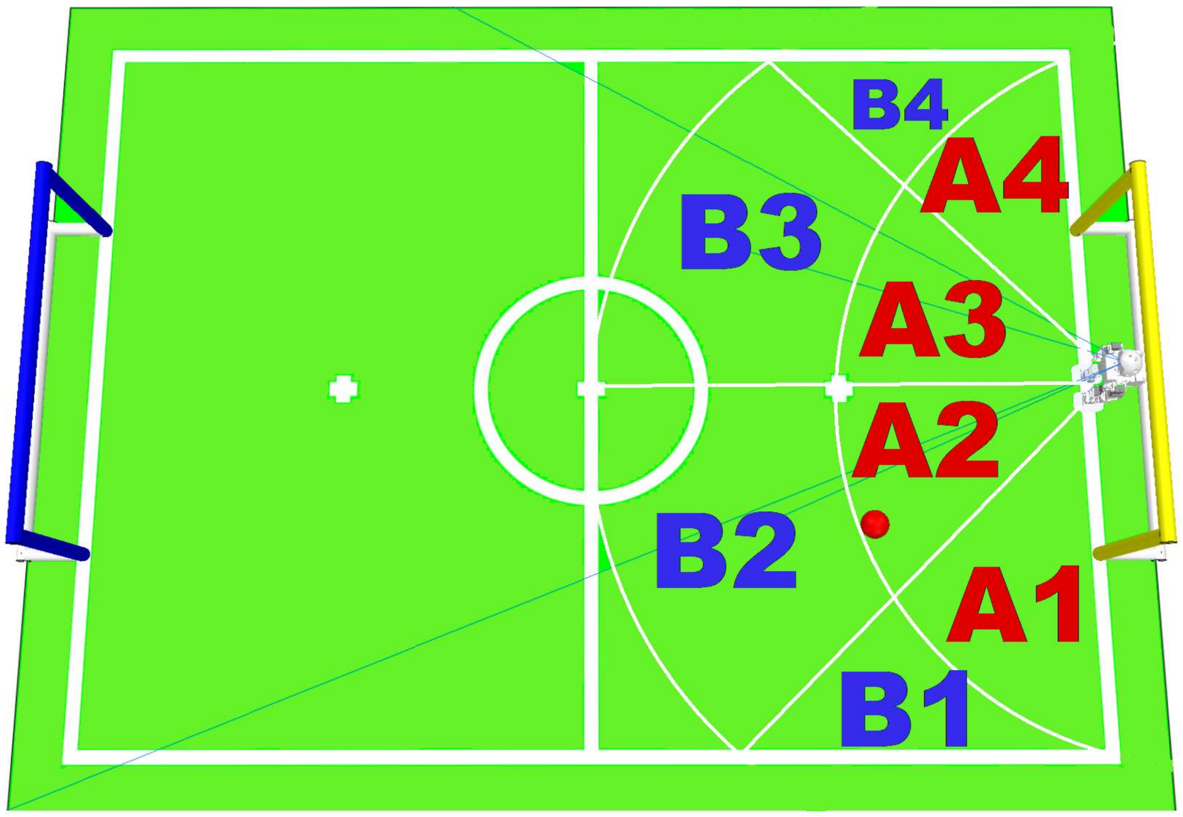


Fig. 2 Sectors of the field of view of the goalkeeper robot

If the ball moves towards the goal at high speed, the robot falls into the twine. Robot does the same when it spots the ball during a penalty kick. These actions are better, since the time spent on protecting the gate is much less than in the old program and the robot does not have to fall into the twine every time and the effectiveness of protection increases.

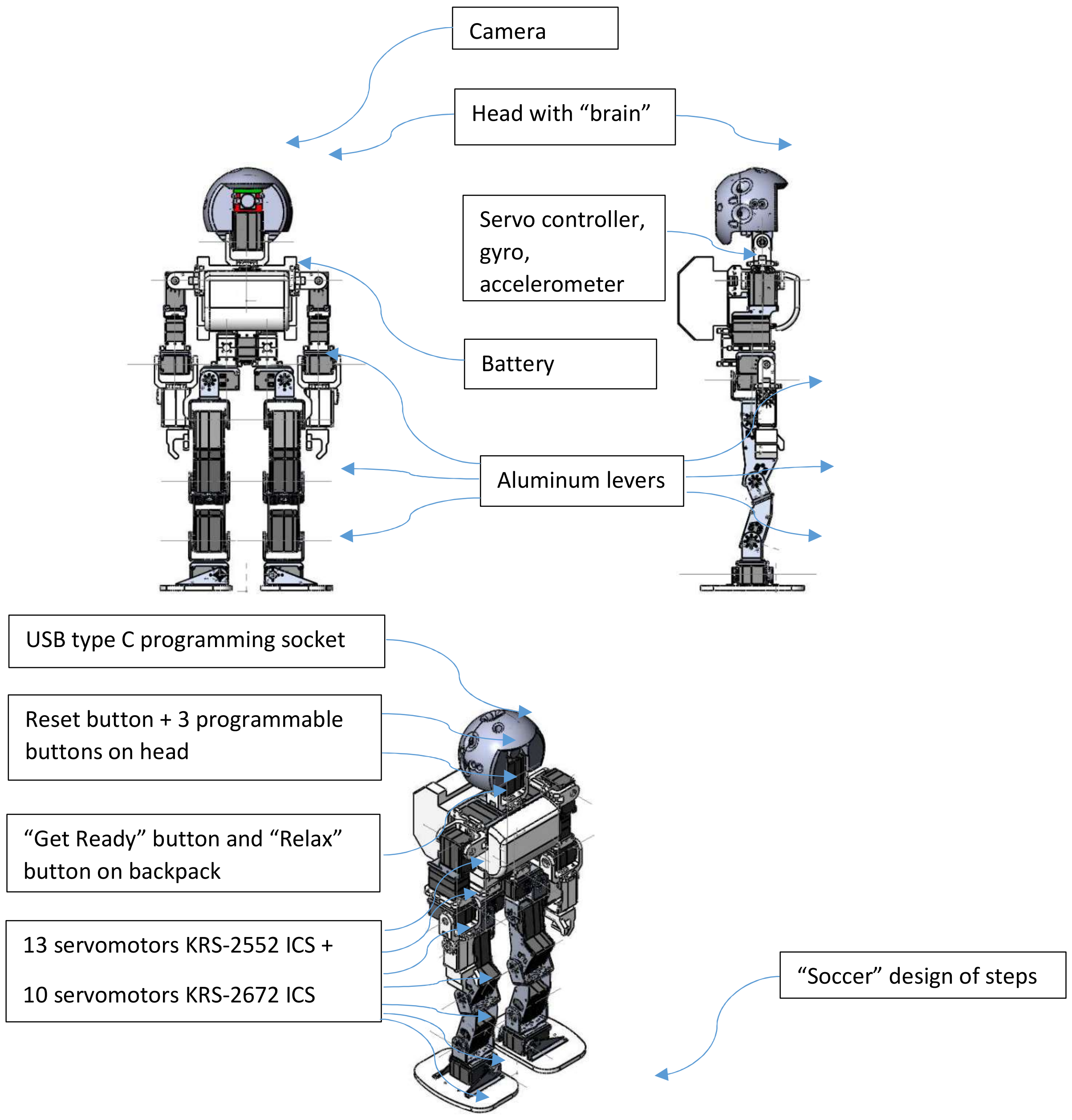
Together we worked out the strategy of the game.

Team member Daniel Babaev has awarded 2-nd place in FIRA 2019 Hurocup All-round with world record in Triple Jump.

On Robocup Asia Pacific 2019 competition our teams were awarded 1st and 3rd places. We haven’t done any other competitions yet.

3. Strategy

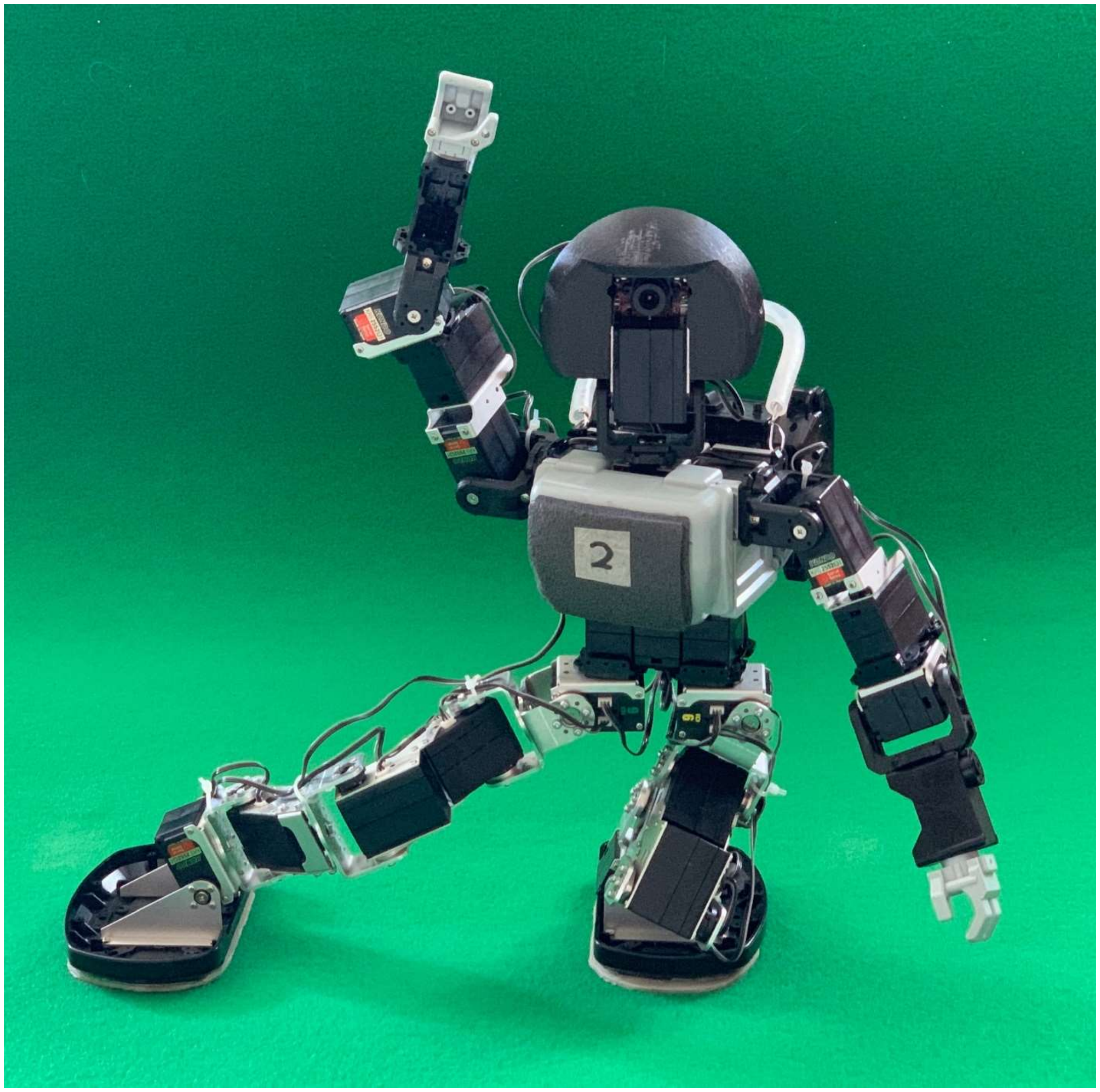
a. Robot design and structure


Camera
Head with “brain”
Servo controller, gyro, accelerometer
Battery
Aluminum levers
USB type C programming socket
Reset button + 3 programmable buttons on head
“Get Ready” button and “Relax” button on backpack
13 servomotors KRS-2552 ICS +
10 servomotors KRS-2672 ICS
“Soccer” design of steps

STARKIT Robot

STARKIT Robot Specification:

- Height 45 cm
- Weight 1.9 kg
- Battery voltage 12 V
- 23 DOF: 13 servomotors KRS-2552 ICS +10 servomotors KRS-2672 ICS
- Main controller: OpenMV H7 with 32-Bit Arm Cortex-M7 operating at 400MHz with 1Mb SRAM
- Motion controller: Kondo RCB-4HV
- Programming language: Micropython.
- Sensors: OV7725 640x480 camera, 6D digital IMU BNO055, 2D analogue Gyro, 3D analogue Accelerometer



Robot is made based on commercially available humanoid kit KONDO which is made in Japan.

From KONDO kit we have used following parts:

- servomotors,
- servo controller,
- general mechanical bipedal frame,
- analogue gyros,
- analogue accelerometers.

Following parts from original kit were modified:

- step sole,
- servomotor plastic levers are replaced by machined aluminum levers
- Battery and high current wires,
- arrangement of bipedal structure was re-designed to make robot to be capable for splits pose.

Following parts were designed and added to robot:

- Smart Camera OpenMV
- Digital IMU
- Robot Head(3D printed)
- Head pan and tilt servos

Modifications into factory bipedal structure and adding heavy head have led to cancellation of all factory supplied motions including walking engine. All motions and walking engine were re-designed.

b) Description of AI strategy.

Smart Camera OpenMV based on STM32 processor is capable for image still and machine vision processing with using Micropython programming language. Controller of servomotors KONDO is designed for low-level support for motion algorithms.

Basic 5 motion types like “Soccer_HomePosition”, “Get Up (stomach)”, “Get Up (Face Up)”, “Soccer Kick Forward”, “PenaltyDefence” and detection of vertical/horizontal pose were coded by HeartToHeart4 software utility provided by KONDO.

Omni walking motion, Fast Kick motion are designed to be performed in OpenMV cam with transfer to KONDO controller through UART communication.

Biggest part of software is designed on Micropython language.

Software parts critical to time of execution were designed on C language with integration to firmware of OpenMV. This part of work was performed by help of outsource engineer.

Important tip! All AI strategy was designed and tested using artificial physics simulation. This brings good acceleration to work.

General strategy of software is built in following way:

OpenMV part:

1) Camera is initialized and waits for button press. After button is pressed IMU reading is recorded in order to distinguish direction to attack.
2) Observation of surroundings. Robot moves head into 15 poses and takes 5 pictures in each pose. Each picture is processed in order to detect: ball, goal posts, linear marking, field border, penalty marks, obstacles. All data is processed in order to localize robot, ball and obstacles. Global coordinates of all above localized objects are used for dynamically planning of motion and for share with partners through wi-fi.
3) Following motions of robot differ depending on robot role in game. Following roles of player are defined: forward, goalkeeper, penaltyGoalkeeper, `penalty_Shooter, obstacle_runner`, etc.
   For each role following motions run in main cycle.
   For each main cycle sequence of following actions are combined:
- seek_Ball_In_Pose
- turn_To_Course(direction_To_Guest)
- far_distance_ball_approach
- near_distance_ball_approach_and_kick
- return to initial position.
4) As soon as robot falling situation is detected existing task is cancelled and observation of surroundings is launched.
5) Returning to initial position become possible because robot permanently performs self-localization and calculates it's global coordinate using following methods:
- Machine vision of goal posts
- Machine vision of field lines
- Machine vision of penalty marks
- Machine vision of green field border
- Retrieving of Euler yaw from IMU
- Step odometry

KONDO controller part:

- controller receives order from external source and distributes commands to single servos

- controller supports batch commands when several servos can move simultaneously.

- Minimum timestep for batch servo commands is 10ms.

- controller supports motion slots. A series of motions can be stored in motion slot for following playback.

- Stabilization of gait is performed by use of analogue gyro sensors with proportional loop to steps.
- Analogue accelerometers are used for detection of vertical/horizontal position of robot. In case if horizontal position is detected "Get Up (stomach)", "Get Up (Face Up)" motions are triggered depending on detected position.

Robot design tips:

- IMU sensor is allocated together with camera. This gives advantage in accuracy of measuring distance of robot from goal posts. Accuracy of IMU is higher than accuracy of pan servo. This gives opportunity to measure distance through retrieving of Euler Yaw for 2 visible posts. Computation of distance from 2 measured yaw angles and using fixed distance between posts yields more accurate distance data than direct triangulation.

c) Pseudo code.

Here is pseudocode of main cycle for goalkeeper main cycle:

```
def goalkeeper_main_cycle(self):
    near_distance_omni_motion(200, 0)   # get out from goal for 200 mm
    while (True):
        dist = -1.0
        if falling_Flag != 0:
            if falling_Flag == 3: break
            falling_Flag = 0
            turn_Face_To_Guest()
        while(dist < 0):
            a, dist, napravl, speed = seek_Ball_In_Pose()
            if abs(speed[0]) > 0.002 and dist < 1 : # if dangerous tangent ball speed
                if speed[0] > 0:
                    play_Motion_Slot(name ='PenaltyDefenceL') # fall to left side
                else:
                    play_Motion_Slot(name ='PenaltyDefenceR') # fall to right side
                continue
            if speed[1] < - 0.001 and dist < 1.5 :   # if dangerous front ball speed
                play_Motion_Slot(name = 'PanaltyDefenceReady_Fast')
                play_Motion_Slot(name = 'PenaltyDefenceF') # fall to splits
                play_Motion_Slot(name = 'Get_Up_From_Defence')
            if (dist == 0 and napravl == 0) or dist > 2.5:  # ball is too far
                position_limit_x1 = -landmarks['FIELD_LENGTH']/2 - 0.05
                position_limit_x2 = position_limit_x1 + 0.25
                if position_limit_x1 < coordinate_X < position_limit_x2 and
                   -0.05 < coordinate_Y < 0.05: break
                goto_Center()                      # go to center of goal
                break
            if (dist <= 0.7 and 0 <= napravl <= math.pi/4):          scenario_A1()
            if (dist <= 0.7 and math.pi/4 < napravl <= math.pi/2):   scenario_A2()
            if (dist <= 0.7 and 0 >= napravl >= -math.pi/4):         scenario_A3()
            if (dist <= 0.7 and -math.pi/4 > napravl >= -math.pi/2): scenario_A4()
            if ((0.7 < dist < landmarks['FIELD_LENGTH']/2) and
                (math.pi/18 <= napravl <= math.pi/4)):               scenario_B1()
            if ((0.7 < dist < landmarks['FIELD_LENGTH']/2) and
                (math.pi/4 < napravl <= math.pi/2)):                 scenario_B2()
            if ((0.7 < dist < landmarks['FIELD_LENGTH']/2) and
                (-math.pi/18 >= napravl >= -math.pi/4)):             scenario_B3()
            if ((0.7 < dist < landmarks['FIELD_LENGTH']/2) and
                (-math.pi/4 > napravl >= -math.pi/2)):               scenario_B4()

def scenario_A1():    #The robot knock out the ball to the side of the guest
    for i in range(10):
        if dist > 0.5 :         # if distance is more than 0.5m
            far_distance_plan_approach(direction_To_Guest)
        turn_Face_To_Guest()
        success_Code = near_distance_ball_approach_and_kick(direction_To_Guest)
        if success_Code == False and falling_Flag != 0: return
        if success_Code == False : break
        success_Code, napravl, dist, speed = seek_Ball_In_Pose()
        if dist > 1 : break
    turn_To_Course(course_to_home)
    goto_Center()
```

d. Describe and highlight innovative algorithms if any

We have developed following innovative unique algorithms:

- Fast detection of orange ball on green field: underneath orange blob rectangular Region Of Interest (ROI) is defined. Ball is on green field if green blob is detected in ROI. Method is 10 times faster than 2 color code blob detection.
- Calculation of approaching point for convenient position of ball attack. Convenient position is located on same line which starts from center of target goals, passes through position of ball and stops at distance 15 cm behind ball. Additionally, turning point's coordinate is calculated if player stands between ball and target goal gate.
- Algorithm of calculation of necessary steps for reaching desired coordinate on field. Feature of algorithm is to calculate optimum step size. In order to reach desired coordinate in shortest possible time traditionally robot uses biggest possible steps and last step is made with shorter size in order to adjust distance. Then robot has to brake motion before last step in order to avoid falling, last step follows after breaking main motion and consumes more time than each step in main motion. Our algorithm calculates optimized length of steps to provide all steps with same size therefore each step consumes same time and overall time appears less because we don't need adjustment step.
- Algorithm of fast choosing "best fit" objects between M number of detected candidates including noise candidates. Feature of algorithm is to choose randomly first candidate and calculate inlier objects number through given criteria, then choose randomly second candidate, and then repeat cycle N times. Best fit candidate is that one which has biggest number of inliers. In case if $N < M$ algorithm is speed effective. According to our experience $N = 10$ with $100 < M < 200$ yields satisfactory result which is nearly best candidate.

Problems encountered with design of robot and AI:

Problem 1. After all reasonable modifications into robot mechanics many factory-supplied motions especially walk engine appears useless. We forced to design new gait.

Solution 1. Many hints to design of new gait were studied in SPL league publications. Artificial physics simulation provided acceleration in design of new gait.

Problem 2. Original KONDO controller programming is possible by Python using library provided by vendor. Smart camera OpenMV which we attach to robot and use as AI is designed for coding by Micropython. Library designed for Python didn't run with Micropython environment.

Solution 2. Library was reworked in order to fit small memory limits. In final version mpy-cross frozen code compilation was used.

Problem 3. After restart of OpenMV firmware is loaded and camera consumes time for initial boot after that software stands by for pushing button. If handler pushes button earlier event is not detected. It is important to push button after boot is completed.

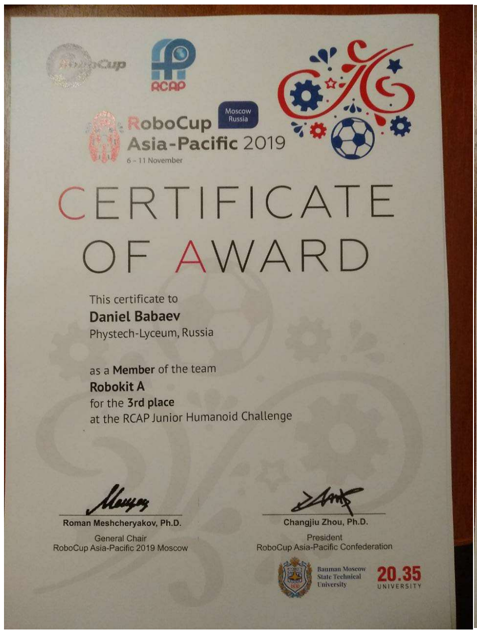
Solution 3. We have installed sound buzzer into digital port of controller in order to sound signal when software is ready to accept button push. Same buzzer is used to signal when battery runs low level.

Starting from 2020 team maintains ToDo list where all problems are recorded with solution statement. Here are several problems pending for solution:

- Currently algorithm of obstacle avoidance is designed to work effectively with 1 obstacle after which robot has to stop and plan new path. Avoidance of 2-3 obstacles by one path is planned. Possible solution: using A* heuristic.
- Currently goalkeeper make observation of surroundings in order to detect ball. We plan to re-design algorithm with aim to capture ball after first observation of surroundings and to keep ball under permanent watch by slight head turning.
- Intelligent ball detection by forward player after kick. Currently after kick forward player has to wait some time for ball stops motion, then detects ball by observation of surroundings and then starts chasing the ball. Intelligent ball detection supposes that player detects speed and direction of ball motion, calculate predicted point of ball stop and immediately move to target position, after arriving to target position player has to turn head in direction to predicted area where ball can be expected.

3. Track Record

a) Our team has awarded 1-st and 3-rd places in Robocup Asia-Pacific 2019.



RoboCup RCAP

RoboCup Asia-Pacific 2019 Moscow Russia 6 – 11 November

CERTIFICATE OF AWARD

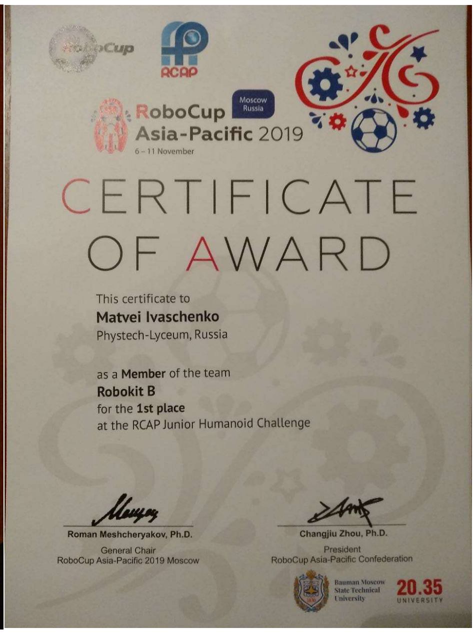
This certificate to
Daniel Babaev
Phystech-Lyceum, Russia

as a **Member** of the team
Robokit A
for the **3rd place**
at the RCAP Junior Humanoid Challenge

Roman Meshcheryakov, Ph.D.
General Chair
RoboCup Asia-Pacific 2019 Moscow

Changjiu Zhou, Ph.D.
President
RoboCup Asia-Pacific Confederation

Bauman Moscow State Technical University
20.35 UNIVERSITY



RoboCup RCAP

RoboCup Asia-Pacific 2019 Moscow Russia 6 – 11 November

CERTIFICATE OF AWARD

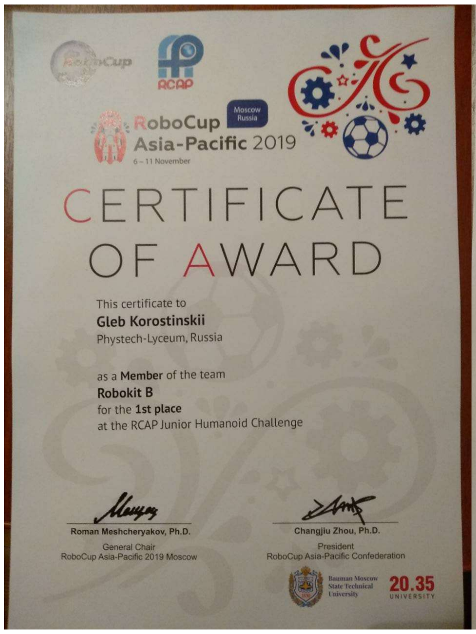
This certificate to
Matvei Ivaschenko
Phystech-Lyceum, Russia

as a **Member** of the team
Robokit B
for the **1st place**
at the RCAP Junior Humanoid Challenge

Roman Meshcheryakov, Ph.D.
General Chair
RoboCup Asia-Pacific 2019 Moscow

Changjiu Zhou, Ph.D.
President
RoboCup Asia-Pacific Confederation

Bauman Moscow State Technical University
20.35 UNIVERSITY



RoboCup RCAP

RoboCup Asia-Pacific 2019 Moscow Russia 6 – 11 November

CERTIFICATE OF AWARD

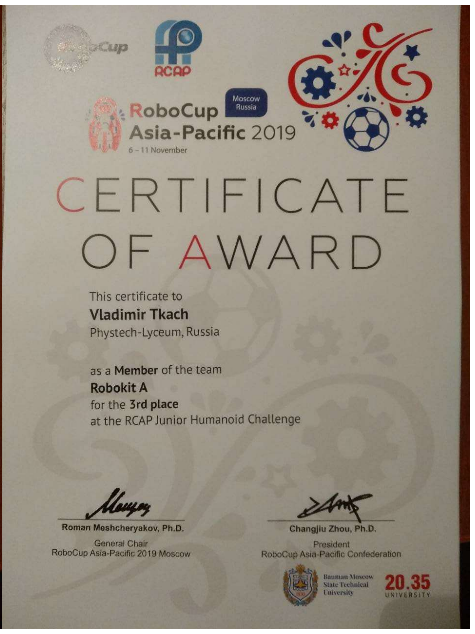
This certificate to
Gleb Korostinskii
Phystech-Lyceum, Russia

as a **Member** of the team
Robokit B
for the **1st place**
at the RCAP Junior Humanoid Challenge

Roman Meshcheryakov, Ph.D.
General Chair
RoboCup Asia-Pacific 2019 Moscow

Changjiu Zhou, Ph.D.
President
RoboCup Asia-Pacific Confederation

Bauman Moscow State Technical University
20.35 UNIVERSITY



RoboCup RCAP

RoboCup Asia-Pacific 2019 Moscow Russia 6 – 11 November

CERTIFICATE OF AWARD

This certificate to
Vladimir Tkach
Phystech-Lyceum, Russia

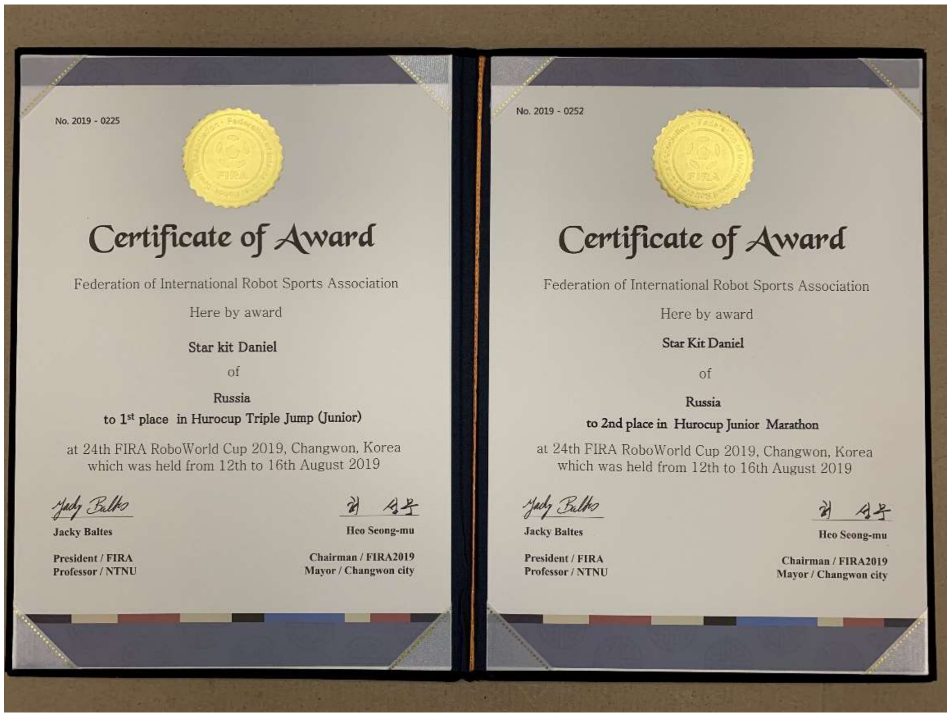
as a **Member** of the team
Robokit A
for the **3rd place**
at the RCAP Junior Humanoid Challenge

Roman Meshcheryakov, Ph.D.
General Chair
RoboCup Asia-Pacific 2019 Moscow

Changjiu Zhou, Ph.D.
President
RoboCup Asia-Pacific Confederation

Bauman Moscow State Technical University
20.35 UNIVERSITY

b) Our team member Daniel Babaev has awarded 2-nd place in FIRA 2019 Hurocup Junior All-round


No. 2019 - 0225

Certificate of Award

Federation of International Robot Sports Association

Here by award

Star kit Daniel

of

Russia

to 1st place in Hurocup Triple Jump (Junior)

at 24th FIRA RoboWorld Cup 2019, Changwon, Korea
which was held from 12th to 16th August 2019

Jacky Baltes
President / FIRA
Professor / NTNU

Heo Seong-mu
Chairman / FIRA2019
Mayor / Changwon city

No. 2019 - 0252

Certificate of Award

Federation of International Robot Sports Association

Here by award

Star Kit Daniel

of

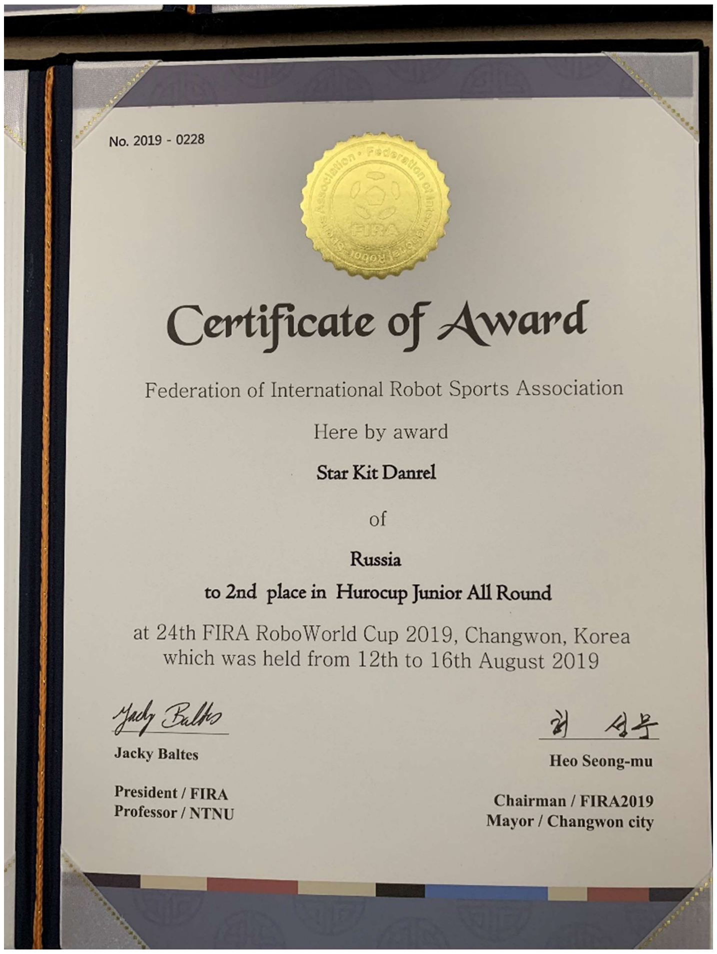
Russia

to 2nd place in Hurocup Junior Marathon

at 24th FIRA RoboWorld Cup 2019, Changwon, Korea
which was held from 12th to 16th August 2019

Jacky Baltes
President / FIRA
Professor / NTNU

Heo Seong-mu
Chairman / FIRA2019
Mayor / Changwon city


No. 2019 - 0228

Certificate of Award

Federation of International Robot Sports Association

Here by award

Star Kit Danrel

of

Russia

to 2nd place in Hurocup Junior All Round

at 24th FIRA RoboWorld Cup 2019, Changwon, Korea
which was held from 12th to 16th August 2019

Jacky Baltes
President / FIRA
Professor / NTNU

Heo Seong-mu
Chairman / FIRA2019
Mayor / Changwon city

5. Discussion and Conclusion

Our team is ready to share his experience with new teams in league. Our mutual task is to promote our League experience to World level. Hopefully we will be able to participate in future Robocup World competitions.

6. Acknowledgements

We would like to express our gratitude to STARKIT Team members from MIPT University especially to students Ilya Osokin for machine vision lessons, Ilya Ryakin for localization algorithms support.

7. References

1. https://www.coppeliarobotics.com/
2. An omni-directional kick engine for humanoid robots with parameter optimization. Pedro Pena, Joseph Masterjohn, Ubbo E Visser. University of Miami.
3. rUNSWift Walk2014 Report. Robocup Standard Platform League. Bernard Hengst.