RoboCup 2023 Submission Survey

Survey response 1

Software

Team Name

HERoEHS

Is your software fully or partially OpenSource. If so, where can it be found:

We are partially used open-source code.
We use the open-source Dynamixel SDK and Dynamixel Framework to communicate and control over the robot's actuator Dynamixel on ROS. Also, we use microstrain_3dm_gx5_45 open source code to get IMU data on ROS.
The lower link shows the name of open-source and GitHub address we used.

[Dynamixel SDK]
https://github.com/ROBOTIS-GIT/DynamixelSDK
[Dynamixel Framework]
https://github.com/ROBOTIS-GIT/ROBOTIS-Framework
[microstrain_3dm_gx5_45]
https://github.com/ros-drivers/microstrain_mips

Do you have a kinematic or dynamic model of your robot(s)? If so, how did you create it (e.g. measure physical robot, export from CAD model)?

We have a kinematic and dynamic model of ALICE.
The model was developed by extracting physical parameters like kinematic structure, shape of each part, excluding mass of components that make up ALICE from CAD model. The mass of the parts that make up the ALICE is measured and used as a parameter.

Are you using Inverse Kinematics? If so what solution (analytic, (pseudo)inverse jabcobian, etc...) are you using?

We're using inverse kinematics to control the ALICE. We use an analytical method-based inverse kinematic solution for real-time computational performance of inverse kinematic solutions.

Are you simulating your robot? If so what are you using simulation for?

We are simulating the Robot by Webots program. Humanoid Robots have sophisticated system as they have a lot of DOF. So, they can move unexpectedly, and there can be an accident. We are preventing these through simulation testing and saving time and resources.

What approach are you using to generate the robot walking motion?

Our Robot can balance and walk using ZMP. We are calculating the ZMP by simplify Robot as inverted pendulum. And adjust walking motion by force torque sensors in the Ankles.

What approach are you using to generate motions for standing up?

We are trying hardware and software to safely fall. After falling, the robot is standing up and balancing by making a support polygon with its feet, knees, and hand.

What approach are you using to generate kicking motions?

1. Divide the ALICE state based on which foot touches the ground. For example, if two feet are in contact with the ground, it can be expressed as a double-stance phase (DSP), if one foot is in contact with the ground, it can be expressed as a single-stance phase (SSP). The kicking action proceeds in the order in which the foot falls off the ground and touches it again. That is, it starts at the DSP and returns to the DSP after the SSP.
2. Set the position of the foot for each phase and use Inverse Kinematics to calculate the angle of each motor on the lower body of Alice. We generate the trajectory of the motor using the 5th Polynomial equation so that each motor can move at the speed and acceleration we intend.
3. Calibrate the target position of the motors mounted on the lower body of the ALICE, using data from the Force-Torque sensor and the IMU sensor to ensure that the ZMP is always located inside the Support Polygon to prevent the ALICE from losing its balance while performing the kick.

Do you use any other motions than the previously mentioned? If so, what approaches are you using to generate them?

Which datasets are you using in your research? If you are using your own datasets, are they public?

We are using various objects and areas of soccer stadiums as datasets for our research.
The number of data is approximately 10,000, and segmentation techniques are used to amplify the number of data.
Our dataset is not open to the public.

What approaches are you using in your robot's visual perception?

YOLOv4 enables the robot to recognize objects.
In addition, YOLOv4 is optimized and used through TensorRT.

Are you planning with objects in Cartesian or image space? If you are using Cartesian space, how do you transform between the image space and cartesian space?

We extract point clouds using ZED2i. And the bounding box coordinates of the object recognized through Pointcloud and YOLOv4 are calculated to extract the distance between the object and the robot.

How is your robot localizing?

Our localization algorithm is vision-based. Robot’s position is estimated by fusioning vision data, sensors data, and contorl data using various filters.

Is your robot planning a path for navigation? Is it avoiding obstacles? How is the plan executed by the robot (e.g. dynamic window approach)?

No, ALICE does not planning a path to move on. ALICE decides the next steps
in realtime. Also ALICE try not to crush on obstacles. When the ALICE faces
the obstacle, ALICE will try to kick the ball to take the ball.

How is the behavior of your robot's structured (e.g. Behavior Trees)? What additional approaches are you using?

I'm using behavior Tree to control the robot. Still there are some bugs to play
so I will going to make it better.

Do you have some form of active vision (i.e. moving the robots camera based on information known about the world)?

To ensure that the center of the ball is located in the middle of the ALICE field of view, Use the following control strategies:
In the vision recognition system, the center coordinates of the ball are expressed in pixels on a 2D image consisting of 1280*720 pixels. And the middle of the ALICE field of view is represented by coordinates (640, 360) based on pixels in a two-dimensional image.
Assuming that the position of the ball is expressed in coordinates (x, y) based on pixels, the ALICE must move by (640-x) on the x-axis and by (360-y) on the y-axis to keep the ball in the middle of the field of view. The direction to be moved should be in the positive direction if x is greater than 640, and in the negative direction if it is smaller. Similarly, if y is greater than 320, it should move The x-axis corresponds to the yaw of the ALICE head joint, and the y-axis corresponds to the pitch. The moving distance expressed in pixels is mapped to the motor angle of the ALICE and converted to the target position of the motor, and the target position is input to the PD position controller to drive.

Do you apply some form of filtering on the detected objects (e. g. Kalman filter for ball position)?

Our team applied Kalman filter to predidct where to lookl if the robot lost the ball.

Is your team performing team communication? Are you using the standard RoboCup Humanoid League protocol? If not, why (e.g. it is missing something you need)?

Yes we are using the Official Robocup Humanoid League protocol. It's work
properly. And I think it contains the every data that we need when play

Please list contributions your team has made to RoboCup

We advanced to the adult size quarter finals in Montreal, Canada in 2018. We finished fifth in the adult size round robin in Sydney, Australia in 2019. We finished third in the 2021 virtual adult size Competition. Finally, we finished second in the adult size competition in Bangkok, Thailand in 2022, the first time in Korea. We participated in the RoboCup Humanoid League every year since 2018. And we getting better as the years go by. We are leading Korea's robotics technology and promoting RoboCup through the media a lot.

Please list the scientific publications your team has made since the last application to RoboCup (or if not applicable in the last 2 years).

none

Please list the approaches, hardware designs, or code your team is using which were developed by other teams.

none

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?
ALICE3 has NUC and Xavier. NUC is running Ubuntu 20.04 and communicate with each other through ROS Noetic. NUC create and run the overall motions. In addition, Xavier sends and receives messages via ROS Melodic with Ubuntu 18.04. Xaiver performs the robot's vision mission.

Is there anything else you would like to share that did not fit to the previous questions?

If you have a description document of your software you would like to share, you may do so here.

[{ "title":"ALICE_Localization-Architecture","comment":"ALICE_Localization-Architecture","size":"127.0419921875","name":"1.ALICE-Localization-Architecture.pdf","filename":"fu_fgtn87gfmwzzp5t","ext":"pdf" },{ "title":"ALICE-Vision-Recognition-Diagram","comment":"ALICE-Vision-Recognition-Diagram","size":"45.2587890625","name":"2.ALICE-Vision-Recognition-Diagram.pdf","filename":"fu_z4ajvr8ui6cpku7","ext":"pdf" },{ "title":"ALICE-Walking-Architecture","comment":"ALICE-Walking-Architecture","size":"205.7509765625","name":"3.ALICE-Walking-Architecture.pdf","filename":"fu_aipw9w5h34cssis","ext":"pdf" }]

filecount - If you have a description document of your software you would like to share, you may do so here.
3