Survey response 21

Software

Team Name
HERoEHS

Is your software fully or partially OpenSource. If so, where can it be found:
Our source code is completely closed source. It is managed privately on GitHub.

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 create the kinematic and dynamic model of our robot by measuring from the CAD model. The mass of the components that make up the robot is obtained through the measurement of the actual parts, which is used for dynamic analysis.

Are you using Inverse Kinematics? If so what solution (analytic, (pseudo)inverse jabcobian, etc...) are you using?
We designed our hardware to facilitate Inverse Kinematics calculations through joint axis alignment, and we computed the Inverse Kinematics using an Analytic approach, which is based on mathematical formulas to accurately calculate the joint angles.

Are you simulating your robot? If so what are you using simulation for?
We are using a total of four simulation tools: Gazebo, Webots, Mujoco, and Unity. Gazebo is primarily used for checking simple robot motions and walking trajectories. Webots is mainly used for simulating decision-making in gameplay in conjunction with the RoboCup Game Controller. Mujoco has been developed with the intention of replacing Gazebo simulations in the future. It can be used to check robot motions and walking trajectories and is convenient for performing reinforcement learning. Unity is currently being developed to replace Webots simulations. It's being developed with the aim of reducing the resources of the simulation itself compared to Webots, to simplify the review of gameplay.

What approach are you using to generate the robot walking motion?
The FootStep Planner receives the robot's target movement point and generates each footstep point that can reach the target location. We have implemented preview control using the fifth-order polynomial function of the Robotis framework, and generate a walking trajectory for each footstep. The x, y, z, roll, pitch, yaw of the generated walking trajectory are converted into the radian values of each joint of the legs using inverse kinematics. Afterward, commands are given to the motors to perform walking.

What approach are you using to generate motions for standing up?
Using Inverse Kinematics and fifth-order polynomial functions, motions are pre-created, and upon receiving a standing up command, the respective motion is executed. Each step of the motion is divided into the motion execution time, allowing the sequence to progress accordingly.

What approach are you using to generate kicking motions?
Using Inverse Kinematics and fifth-order polynomial functions, motions are pre-created, and upon receiving a kicking command, the respective motion is performed. Each step of the motion is divided by the motion execution time, allowing the sequence to progress. The x, y, z, yaw, and kicking speed of the foot change dynamically during a kick, depending on the position of the ball and the direction it needs to be kicked. Additionally, the extent of waist bending, the degree of lifting the foot, and the angle of ankle pitch are configured as adjustable parameters, making the system easy to tune and develop.

Do you use any other motions than the previously mentioned? If so, what approaches are you using to generate them?
Our robot generates motion in the lower body through Inverse Kinematics. The upper body motion is created through position control. The torso calculates positional values to track according to the position of the ball and performs PID position control.

Which datasets are you using in your research? If you are using your own datasets, are they public?
We use a dataset that we created ourselves. It includes data on the ball, feature points of the field lines, the goal, goalposts, and opposing robots. Additionally, we have not made this dataset publicly available.

What approaches are you using in your robot's visual perception?
We perform Object Detection using Yolov4-tiny, and after optimizing the model with TensorRT, we applied it to the robot. It primarily recognizes the ball, feature points of the field lines (lpoint, cpoint, tpoint), the goal, goalposts, and opposing robots

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 use the Stereo ZED2i Camera, and extract the x, y, z coordinates of objects recognized through the camera's PointCloud. Additionally,
we use these extracted x, y, z coordinates after transforming them to the robot's pelvis center coordinates through a Transform matrix operation.

How is your robot localizing?

We have implemented vision-based localization, which is using points and lines on the soccer field from Vision System.
Also, our localization system can overcome even if a robot is moved by a handler, such as an untangled or a penalty situation.
Currently, we are developing our localization algorithm to reliably cope with variable situations, overcoming noise on sensors, control,
vision data caused by the robot's movement and implementing an algorithm about getting the starting position anywhere on the edge of the soccer field.

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)?

Our robot generates a path using Bezier curves to navigate to the target location. During the path generation process, if obstacles are detected, the robot creates waypoints to create a new path for avoiding these obstacles. Through this modified path, the robot can effectively avoid obstacles and reach its destination.

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

Our team implemented an action tree using an open source called 'BehaviourTree'. It is possible to effectively manage and develop complex robot behavior by organizing each situation into a condition branch and organizing the robot's behavior into action.

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

The camera is attached to the robot's head and its angle can be adjusted by the neck pitch, neck yaw, and waist yaw joints. The joints rotate to ensure that the estimated position of the ball is centered in the camera's field of view. The rotation amount of the joints is calculated using trigonometric functions, and to reduce computation, it has been standardized using regression analysis. An exponential function is used to progressively turn the waist yaw joint as the yaw angle increases. Each joint incorporates PID position control, and a limit on position change depending on the size of the P gain is set to prevent the camera from shaking or the joints from moving abruptly or vibrating.

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

A Low pass filter is used to reduce the noise in the x, y, z coordinates of the recognized object. Additionally,
Kalman filtering has been applied to continuously track the ball and estimate its approximate location when it goes out of the camera's view.

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)?

Our team communication was implemented as UDP communication according to the rules of the competition.
The information received by communication is information to easily solve the game, such as the position of the ball the robot is looking at and the position of the opponent robot.
We also implemented a GUI that can check the current state of the robot outside the field in real time.

Please list contributions your team has made to RoboCup

We have consistently participated in the RoboCup Humanoid AdultSize League for the past six years and have been recording increasingly better results. Our team leader, YounSeal Eum, has been involved in the Organizing Committee since 2023 and contributed to hosting RCAP2023 in Pyeongchang, South Korea.
Below is our record of participation in RoboCup.

RoboCup2023 Bordeaux RoboCupSoccer-Humanoid-AdultSize Soccer Competition reach second place.
RoboCup2023 Bordeaux RoboCupSoccer-Humanoid-AdultSize Technical Challenge reach first place.
RoboCup2022 Bangkok RoboCupSoccer-Humanoid-AdultSize Soccer Competition reach second place.
RoboCup2022 Bangkok RoboCupSoccer-Humanoid-AdultSize Technical Challenge reach second place.
RoboCup2021 Virtual RoboCupSoccer-Humanoid-AdultSize Soccer Competition reach third place.
RoboCup2019 Sydney RoboCupSoccer-Humanoid-AdultSize Soccer Competition reach fifth place.
RoboCup2018 Montreal RoboCupSoccer-Humanoid-AdultSize Soccer Competition reach the quarter finals.

Please list the scientific publications your team has made since the last application to RoboCup (or if not applicable in the last 2 years).
Swing foot pose control disturbance overcoming algorithm based on reference ZMP preview controller for improving humanoid walking stability (RoboCup Symposium 2023) Swing kick motion of humanoid soccer robot to successfully kick a moving ball (RoboCup Symposium 2023) Robust Line Detection in Soccer Fields Using Semantic Segmentation: An Approach towards Overcoming Traditional Computer Vision Limitations (RoboCup 2023 Software Challenge) Kang, G., Won, J., Ahn, M., & Han, J. (2023, June). The Development of the Linear Actuator with the Series Elastic System for Robot Linkage Mechanisms. In 2023 20th International Conference on Ubiquitous Robots (UR) (pp. 764-770). IEEE. Choi, J., Chun, Y., Min, I., Ahn, M. S., & Han, J. (2023, June). The Study on the Energy Efficiency of the Bipedal-Walking Pattern of a Humanoid Robot Using Four-Bar-Linkage Toe Joints. In 2023 20th International Conference on Ubiquitous Robots (UR) (pp. 108-114). IEEE. Chun, Y., Choi, J., Min, I., Ahn, M., & Han, J. (2023, January). DDPG Reinforcement Learning Experiment for Improving the Stability of Bipedal Walking of Humanoid Robots. In 2023 IEEE/SICE International Symposium on System Integration (SII) (pp. 1-7). IEEE.

Please list the approaches, hardware designs, or code your team is using which were developed by other teams.
We did not refer to other teams' approaches, hardware designs, or code.

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?
Our robot runs on the Ubuntu 20.04 operating system, and we are using ROS1 Noetic as the middleware. We plan to use Ubuntu 22.04 and ROS2 Humble in 2024, but it's not yet confirmed.

Is there anything else you would like to share that did not fit to the previous questions?
None.