Walking

Please give a brief summary of your walking algorithm (max. 1000 characters).

We use the end of two legs with 12DOFs as an end effector. Over time, create pattern that is a set of coordinates of the end effector, and then get the target angular position of each joint using inverse kinematics. We passively change parameters of the pattern using a tuning program and find the optimal value for stabilizing walking. We improved walking stability by detecting disturbance using IMU. We measure the ZMP from robot's COP using feet equipped with four load cells. The robot's ZMP doesn't deviate from the support polygon by the PID control using the measured ZMP value. If the estimated value of ZMP value and the angular position exceed the predefined it, the final compensation value is determined through the PID controller. And it helps restore the robot's torso angle and ZMP value to the normal range by controlling the joints of legs. Since we decided that this method would be easier to walk than to control with a predefined walking cycle, it plans to modify it so that the posture control is done step by step.

Vision

Please give a brief summary of your vision algorithm, i.e., how your robots detect balls, field borders, field lines, goalposts and other robots (max. 1000 characters).

We recognize the ball through deep learning using Yolo v4-tiny. By bringing the results of learning from an external PC, we succeeded in recognizing the intersection line(X-cross) of the field and the ball, and we plan to learn and recognize the penalty mark. Recognized results are used as data needed for localization. Convert to 3D world coordinates using a geometric method with the camera's focal length and vertex, height, and tilt. This is used to calculate the distance of the object of interest. In addition, it detects lines through image processing using binarization, canny edge detection, and Hough Transform. Using Edge points of the white line, we provide line information to the localization. And We will try to develop new vision system to recognize the ball and line through deep learning using semantic segmentation model.

Localization

Please give a brief summary of how your robots localize themselves on the field (max. 1000 characters).

Our team use localization system based on Monte-Carlo-Localization. Our localization model performs three-processes, Motion Model-Observation Model-Resampling. First, Motion model finds the direction and location of robot based on initial position using the IMU sensor and odometry model based on walking data of robot and spread particles throughout the map. Second, Observation model gives to weight to the particles for correction errors due to walk during playing and finds the right position. Our Observation model uses vision data about the distance of characteristic points like intersection lines(X-cross) or Penalty circles to gives weights to the particles compare to likelihood field that is a model stores weights values according to each location. Finally, Resampling process weed out particles has less weight value and correct location of robot using a particle has the highest weight. As a result, we can estimate the exact location of robot by these three processes.

Behavior

Please give a brief summary of your behavioral architecture and the decision processes of your robots (max. 1000 characters).

Our robot system consists of USB camera, a main controller (mini pc). Main controller executes overall algorithms and image processing. Sub controller (circuit with MCU) processes data from sensors (IMU, load cell), and controls motors (DOF). The data is transferred to main controller via RS-232. In our decision process, first we each move the robot's pan-tilt motor to find the ball and then we walk around their set areas and quickly search it in a wide field if we can't find it for a several times. If our team's robot finds the ball, we can share robot's position data on the field and ball position data using UDP method. So, to save time and efficiently play soccer, the robot

that is close to the ball approaches and kicks the ball, and the other robot moves forward or sideways depending on the position of the ball so that it doesn't get in the way of the robot kicking the ball. We change the path to a curve line, so that the kicker aligns with opponent’s goalpost to maintain a kickable distance quickly.