System Design Project, Group 11, Report 3

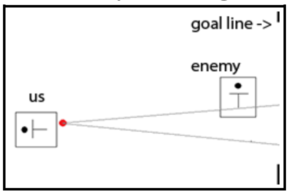
The following report describes our progress towards creating a Lego robot that can play robocup football. It also summarises our attainment of Milestone 3, which required our current robot to be able to navigate to a static ball on an empty pitch, dribble this ball while maintaining control of it, and kicking the ball into the goal. Our robot also required the ability to deal with obstacles appropriately. Reaching this milestone involved developing several aspects of our program, while making further improvements to our robot's design.

Progress Summary

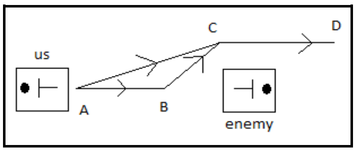
Area	Target Aim for Milestone 3	Status at Milestone 3	Assessment	Target Aim for Milestone 4	For Further Development
Vision	Modify the vision system to operate correctly with the second pitch.	Vision system operates on the second pitch correctly but not under all conditions.	On target	Convert first pitch thresholds to HSV colour space.	Refine vision with second pitch for friendlies. Improve performance from present ~17 frames per second.
Planning/ Strategy	Dribble the ball towards the goal.	Dribbling has been implemented, in a straight line.	On target	Calculate plans to intercept a moving ball.	Defend the goal appropriately. Optimise paths further.
Movement	Respond to the touch sensors.	Robot can now respond to the touch sensors.	On target	No changes required for this milestone.	Implement motion in arcs and reversing.
Robot Design	No changes required for this milestone.	No changes made to robot.	On target	No changes required for this milestone.	Significant changes planned, see dedicated section.
Integration	No changes required for this milestone.	The NXT brick has been flashed with the LeJOS operating system and integrated with our system.	Ahead of target	No changes required for this milestone.	Develop more unit tests to detect undesired behaviour in individual modules. Investigate measures to improve latency/ping time.

Programming

An improved kicking heuristic has been developed for the planner. An error value is applied to the angle the robot is facing, to determine a triangular area of rays from our robot within which the ball is likely to go once kicked. The probability of scoring is then determined by dividing the number of rays hitting the goal without intersecting the enemy by the total number of rays. If the probability of scoring is sufficient, then the robot always kicks. Testing showed that this heuristic improved our robocup player's goalscoring over our previous method.



Smoothing has been implemented in the pathfinder. It now removes intermediate points from the raw path generated by the A* algorithm, such as the example B in the diagram, if the direct path is an obstacle-free option based on our robot's size.



The planner code has been rewritten from Python to C++. When we tested our new planner, we found it to be around twenty-five times faster than the original, taking the planning time down from approximately 200ms to about 20ms, after our extra calculations are performed. This will free computing resources for other modules, while quicker planning will improve the competitiveness of our robocup player.

The vision system underwent significant changes to be useful on the second pitch. The lighting on this pitch varies depending on the time of day, so we found it difficult to use a simple background subtraction and also to choose a range of RGB threshold values. Our first attempt to counter this problem involved attempting to normalise the RGB colour space, by dividing each pixel's value by the sum of all pixel values, over all channels. This method worked in testing, but the per-pixel calculations were computationally expensive, taking approximately 400ms for a medium-size image. The slow performance made the normalisation method unsuitable. Instead, the vision system was converted from the RGB colour space to the HSV colour space. HSV has the advantage of storing hue, saturation, and value (for brightness) in a single range of threshold values, whereas using RGB requires multiple sets to achieve the same amount of detail. Using HSV countered the effects of changing light at much less computational cost. To further improve the vision system's accuracy, we altered it to use a background model incorporating multiple images instead of a single background image.

Our simulator has been enhanced so that it may play out a match between two opposing robots. This will improve our ability to test changes to planning, as we will be able to observe improvements and regressions in match situations. This feature will be useful in preparation for the friendlies.

We conducted some research into how our movement could be improved and came across LeJOS, a replacement firmware for the NXT brick. LeJOS can execute Java programs, as it includes a Java virtual machine. This allows versatile programs to be executed directly on the brick. LeJOS includes useful libraries for navigation, including the ability to smooth out movement, automatically adjust motor power depending on battery level, and keep track of the robot's own position by reading the motors' built in tachometers. After converting our movement code to suitable Java, we tested it running on the brick and found movement was much more responsive. Implementing movement this way eliminates the problem of lag from the vision system, where there is a wait for vision to catch up before the robot's position can be determined accurately. The movement module on the brick has been fully integrated with the rest of the system, and this will allow the internally-stored position of the robot to be corrected with vision information at appropriate times. The vision information would be used to improve the estimated internally-stored location, by measuring the delay between the frame time and the time the correction is applied on the brick. The Java program on the NXT brick also handles the robot's response to the activation of the touch sensors.

The versatility of our modular system has been demonstrated yet again, as our system now integrates modules written in Python, C++, and Java. Having the foresight to build our system this way from the ground up has allowed us to pick, in our opinion, the best language for each task.

Robot Design

We plan a significant overhaul of our robot's design. We will order an additional NXT motor, which shall be used to improve the force of our kicker. Two additional touch sensors will be incorporated at the rear of our robot, with a bar across them, so collisions can be detected during backwards movement. A similar physical connection between the two front-facing touch sensors will improve collision detection in this area of our robot. We considered installing side touch sensors, but these would serve no purpose since we do not move in this direction; they would only be triggered if our robot was fouled by an opponent.

Group Organisation

Overall, the group continues to work efficiently and productively. There have been no specific challenges in this area since the last milestone. The group co-ordinates through email and our Trac ticketing system. We continue to program to no particular paradigms, not having felt the need to adopt one. However, we have adopted a general process of building the simplest solution, then refining and adding complexity. This ensures we achieve the milestone before trying to get ahead in certain areas. We also investigate libraries that are available instead of 're-inventing the wheel'. All group members choose their tasks and work in areas they feel comfortable with, which currently works well, owing to our wide range of talents. We remain aware that working this way may not continue to be appropriate.

Milestone 3 Demonstration

We began our demonstration with a presentation of the improvements made to our simulator. The visualizer was used to display the vision information the simulator generated, while one of the simulated robots was moved using the remote-control module. The other simulated player followed adapting paths depending on where the remote-controlled player was 'driven.' The currently-calculated path was displayed on the visualizer so the adaptations could be observed. This section of the demo was a success.

The plan was then to have our robot navigate to the ball and score using the kicker. Unfortunately, despite thorough testing of the vision system, and running it live while other teams were demonstrating to verify its reliability, it failed to calculate the orientation of the robot during our demonstration. It did not recognise the black spot on the T-plate and so the robot was unable to determine which direction it was facing. Therefore, the robot was unable to move. This was extremely disappointing, because we had never experienced such a major failure during our testing.

The vision system was operating as expected during the demonstration of the team immediately before us. We have determined the cause of failure to be a change in lighting conditions as people left following the demonstration before ours, which made the table lighter and caused the vision system's threshold values to be inappropriate. The lighting on this table is inconsistent and this makes it difficult to determine a set of values that will work under all conditions. This failure was very frustrating for the team, who had put in a great number of hours of hard work and preparation only for it to be undone in such an unpredictable fashion. If our demonstration had occurred at any other time it is likely it would have been successful.

We did determine that, in a certain area of the table, the vision system would determine the robot's orientation. Therefore we were able to show that we could navigate to a static ball. We were also able to demonstrate the operation of the touch sensors, but we could not dribble or score a goal, owing to the vision system's problems, despite performing these abilities at the Milestone 2 demo.

When it became obvious that the vision system could not be adjusted in the time we had available during the demonstration, we went on to present our tools which can show representations of the vision system's calculations, to determine the locations of the robots and ball. We also briefly exhibited an unstable work-in-progress method to determine the orientation of the robots using only the T-shape on the plate and not the black spot.

Conclusion

Our project has met Milestone 3, though we were unable to demonstrate this. We have developed clear, achievable goals for the next two weeks. We have maintained our standing among the most competitive of the groups; therefore we believe we will perform well at the first friendlies in a week's time. The significant failure of the demonstration has demoralised the group, but we collectively appreciate the work goes towards the overall project and not just the milestone. The work we have done in the back-end of our system has improved performance and laid the foundations for observable improvements in weeks to come.