

RoboCup 2013

Humanoid Kidsize League Winner

Daniel D. Lee, Seung-Joon Yi, Stephen G. McGill, Yida Zhang,
Larry Vadakedathu, Samarth Brahmabhatt, Richa Agrawal,
and Vibhavari Dasagi

GRASP Lab, Engineering and Applied Science, Univ. of Pennsylvania, USA
`{ddlee,yiseung,smcgill13,yida,vlarry,samarthb,richaagr,dvib}@seas.upenn.edu`
<http://www.seas.upenn.edu/~robocup>

Abstract. The RoboCup Humanoid Kidsize League has rocketed into an exceptionally competitive league, supported by recent introduction of the open source hardware and software platforms. Teams must improve substantially their robot teams every year to stay competitive. In addition to the important skills of walking speed, kick stability, and visual acuity, the 2013 season introduced rule changes to test robot intelligence – and team adaptability. In this paper, Team DARwIn, the winning team of the Humanoid Kidsize League, presents its crucial improvements of its robotic platform and soccer system to mitigate risks from same colored goal posts, substitutions and noise in goalkeeper ball estimates.

1 Introduction

Team DARwIn from the University of Pennsylvania’s GRASP lab has been competing in the Humanoid kidsize league since 2010 with a DARwIn series of robots, and has been remaining champion since 2011. We also pushed forward to open source both the hardware and software platforms: the latest generation of DARwIn robots, the DARwIn-OP, is openly available in the form of blueprints and as a commercial product. The modular, easy-to-use UPenn humanoid robotic software platform has also been released as open source¹.

The open-source release of both software and hardware platforms made quite a big impact on the humanoid kidsize league this year. Two thirds of the teams used at least one DARwIn-OP robot or its variant, and more than half of the teams used our software framework [1] to varying degrees. For our own team, this competition was a time for focusing on localization to mitigate the complexity of having same goal post color with no landmarks, adding more flexible walk and kick transitions, and adapting our substitution execution. We will explain each in detail in the following sections.

¹ <http://seas.upenn.edu/~robocup>

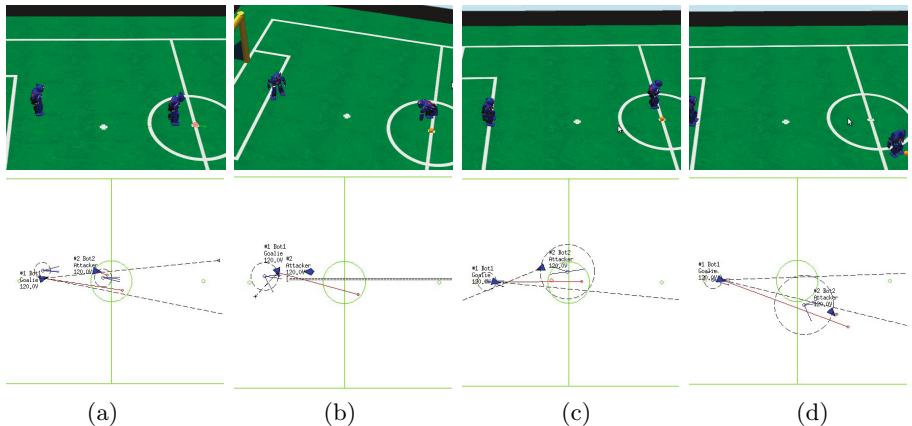


Fig. 1. The flip correction simulated in the Webots simulator. Each robot is represented by a blue triangle. The dotted circles show the error of the robot’s localization relative to its actual location, the red lines show the robot’s ball estimate while the dotted black lines show the robot’s estimate of the goalposts. (a) Correct localization on both robot (b) Field player falls down (c) Field player got flipped (d) Flip corrected by the ball information.

2 Localization in a Symmetric Field

The biggest challenge for RoboCup 2013 was the adoption of a fully symmetric field to make gameplay more compatible with human soccer. The goalposts on each side of the field are now required to have the same color; additionally, the unique half-field localization landmarks have been removed. This means that no unique landmark exists on the field, and the robots must keep track of their localization status using ambiguous information.

We have been using a particle filter based localization strategy which can easily be generalized for fully ambiguous landmarks. However, with the field being fully symmetric, the robot has two possible pose candidate given the observation: the actual pose and the ‘flipped’ pose. A flipped pose occurs when the pose particles get attracted to an orientation that is mirrored about the center of the field. If this flipped pose is believed by the robot, then it will attack and score in our own goal. We have found that these flipped poses can occur when the robot gets pushed around near the center of field to accumulate lots of error in pose estimation.

To fix the flipped localization status, we used the localization information of the goal-keeper. With our strategy on the field, the goal-keeper should never approach the center of the field; thus it can act as a marshaling agent for robots that are confused in the middle of the field. The goal-keeper helps to reverse the pose flipping player by using the soccer ball as a common reference point for both robots – only one ball occupies the field at a given time. Our assumption is that the goalkeeper’s ball estimation is fairly accurate. When the goal-keeper



Fig. 2. The ZMP preview control based kick during locomotion

estimates that the ball resides in its own half of the field and the confused robot can see the ball, the confused robot can properly disambiguate the goal posts.

However, we cannot use such a flip correction method if the ball is close to the center area. The goal keeper may easily misjudge the side of the field in which the ball resides, and the robot is able to score from the half field range. To prevent such mishaps, we assume that if the robot falls at the center field it instantly labels itself as confused. We do not allow the robot to kick directly to the goal until it is sure of its pose; instead, we make the robot move the ball to the side of the center field. In this way we can still keep the ball moving, while preventing the chance of scoring to our own goal. Additionally, it helps to disambiguate the localization based on the goalkeeper's observation.

3 Walk and Kick Controller Improvements

In spite of its relatively small size, the DARwIn-OP robot could walk with a speed up to 32cm/s with our ZMP-based walk controller. This was one of the fastest walk speed of the 2011 and 2012 competition, but this year many robots raised this bar and a few robots could walk with a speed exceeding 40cm/s. To remain a relatively speedy competitor, we increased the top walk speed of the robot to 40cm/s. Furthermore, we optimized the approach trajectory so that the robot walks in a curved trajectory to both minimize the traversal time and keep the robot stable. As a result, we could easily outrun larger and faster walking robots in spite of much shorter leg lengths.

Another improvement we have made this year is a new kick controller. Previously, we only used two kick controllers: a stationary one that is powerful but slow, and a dynamic one based on the locomotion engine that is very fast but relatively weak. This year, we debuted a new kick controller based on ZMP preview control. ZMP preview control is used by a few teams for locomotion and can

yield a more powerful kick than the analytic ZMP algorithm we use. However, it has a big disadvantage of requiring a preview period that makes the robot less reactive to a changing environment. Thus, we only use the ZMP preview control for kick generation since kicks do not require reactive movements. The analytic ZMP algorithm continues to control locomotion. The transition between these two ZMP controllers is achieved by including the boundary conditions of the analytic ZMP controller as additional cost terms in the optimization process for the ZMP preview controller [2].

For the competition, we tuned the new kick controller to have the same speed as the previous dynamic walk kick, yet the new controller improved kicking distance by roughly 50%. Table 1 shows a comparison of our three different kicks. In game terms, the robot needed to successfully kick twice in succession to score from center field with the analytic walk kick, and now it can score with just one kick. This improvement greatly helped us this year.

Table 1. Comparison of average distances and execution times of kicks

	Stationary	Walk-Kick	ZMP Preview
Kick Distance (cm)	411	188	292
Start Time (sec)	2.60	0.70	0.70
Total Time (sec)	4.30	0.95	0.97

4 Goalkeeper Behavior

The main role of the goalkeeper in RoboCup is to defend the goal post from incoming shots; one way of blocking them is by diving. Unfortunately, the goalkeeper cannot lie down on the field indefinitely, as laying down for more than 5 seconds is against the rules and leads to the goalkeeper being ejected off the field. On-demand diving against an opponents' kick requires the precise determination of the position of, and time when, the ball will cross the goal line. Visually determining these crucial variables using the ball's velocity is difficult, as camera images are inherently noisy.

These images are captured at approximately 30 frames per second; and we use our standard ball detection algorithm to get the current position of the ball. The simplest way to determine the velocity is to find the displacement of ball in two consecutive image frames, with respect to the time difference between these frames – this is called an averaging filter. Unfortunately, this approach is subjective to noise due to false positive ball detections and the noise in each position reading.

Intuitively, we would want to train our model to use all previous information instead of relying on the last two frames alone. We implemented a two dimensional linear Kalman filter, similar to [3], maintaining a state consisting of four variables: $s = [x, y, \dot{x}, \dot{y}]$. x and y are the relative ball positions from the robot frame in their respective axes.

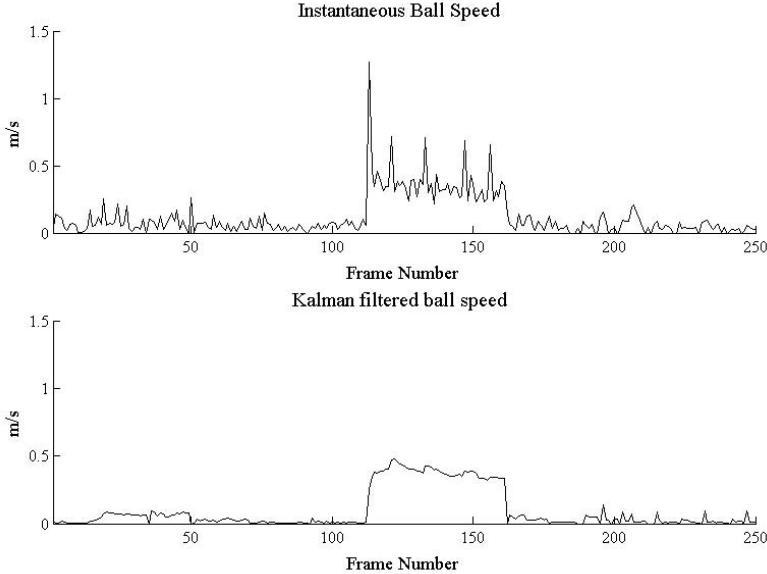


Fig. 3. When a ball is kicked, a standard frame by frame velocity filter is very noisy. In comparison, the Kalman filter eliminates much noise.

We assume a steady decrease in speed with the parameter α . α controls the velocity of the ball from a wrong estimation if the ball is occluded. If the velocity is not controlled, the Kalman filter would predict the next position from the previous velocity which might lead to erroneous predictions. No external input to the ball is modeled. Our Kalman process prediction is modeled with the following state transition formulation:

$$s_{t+1} = \begin{bmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & \alpha \end{bmatrix} s_t \quad (1)$$

This Kalman estimation provides smoother, more reliable ball estimation. Figure 3 shows the comparison between the estimated ball velocities using averaging filter and the Kalman filter. From the ball state, we extrapolate Δt seconds into the future to see if the ball is in a dangerous zone where the goalkeeper must dive: $x_f = \dot{x}\Delta t + x$, and similarly for y . In the real matches, the new ball velocity filter worked very well, triggering almost no false positive during the whole match.



Fig. 4. The team table with three robots ready for play and two substitute robots waiting to get their chance

5 Rapid Substitutions

Using custom hardware, teams in the humanoid kidsize league inherently encounter more hardware issues than their standard platform league counterparts. With these issues, substitutions play a critical role in winning a match. The current rules allow for a maximum of two substitutions per match, and those substitutions can target either the goalkeeper or a field player. The problem is that we don't know which robot will require a substitute in advance. We may need to substitute two field players, or the goalkeeper twice. Booting up the robot manually and setting up the roles during a high pressure match leads to mistakes. On the other hand, taking too much time to switch could mean that less than three players occupy the field – a highly detrimental situation.

Our solution is to implement a script where the robot handler can set the role of the robot via a button press right before placing the robot on the field. The robot handler receives visual and auditory feedback from the robot to reduce the chance of a mistake. Another feature of the script is that malfunctioning robots taken off the field can be easily be placed into waiting state by button press. In this mode, the robot cannot affect teamplay by sending team messages. Figure 4 shows the typical team table setup right before the match, with 3 robots ready to play and two robots waiting in a waiting state for possible substitution.

6 Conclusions

The 2013 tournament was an extremely stiff competition, full of very tight, nail-biting matches. This RoboCup pushed the overall flexibility of our robots to the

fore – being able to adjust unforeseen faulty localization, performing substitutions at the drop of the hat, and including more adaptable kicking engines. This flexibility required small but far reaching codebase adjustments, that hammer home the importance of modular design. Furthermore, with more avenues for failure, rigorous testing needed to be conducted on not just one robot at a time, but several – flip correction, for instance, requires both a goalie and field player for proper evaluation. With more changes upcoming in the RoboCup midsize league, the ability to adapt software very quickly will continue to be the number one priority. The same color goal post is just the first of many challenging changes to RoboCup software that must happen to achieve the 2050 goal of competing against the World Cup champions.

Acknowledgements. We acknowledge the support of the NSF PIRE program under contract OISE-0730206 and ONR SAFFIR program under contract N00014-11-1-0074. This work was also partially supported by the NRF grant of MEST (0421-20110032), the IT R&D program of MKE/KEIT (KI002138, MARS), and the ISTD program of MKE (10035348).

References

1. McGill, S.G., Brindza, J., Yi, S.-J., Lee, D.D.: Unified humanoid robotics software platform. In: The 5th Workshop on Humanoid Soccer Robots (2010)
2. Yi, S.-J., Hong, D., Lee, D.D.: A hybrid walk controller for resource-constrained humanoid robots. In: 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids) (October 2013)
3. Seekircher, A., Abeyruwan, S., Visser, U.: Accurate ball tracking with extended kalman filters as a prerequisite for a high-level behavior with reinforcement learning. In: The 6th Workshop on Humanoid Soccer Robots (2011)