RoboCup 2011 Humanoid League Winners

Daniel D. Lee[1], Seung-Joon Yi[1], Stephen McGill[1], Yida Zhang[1]
Sven Behnke[2], Marcell Missura[2], Hannes Schulz[2]
Dennis Hong[3], Jeakweon Han[3], and Michael Hopkins[3]

[1]GRASP Lab, Engineering and Applied Science, Univ. of Pennsylvania, USA
{ddlee,yiseung,smcgill3,yida}@seas.upenn.edu
http://www.seas.upenn.edu/~robocup

[2]Autonomous Intelligent Systems, Computer Science, Univ. of Bonn, Germany
{schulz,behnke,missura}@cs.uni-bonn.de
http://ais.uni-bonn.de

[3]RoMeLa, Mechanical Engineering, Virginia Tech, USA
{dhong,jk4robot,hopkns}@vt.edu
http://www.romela.org/robocup

Abstract. Over the past few years, soccer-playing humanoid robots advanced significantly. Elementary skills, such as bipedal walking, visual perception, and collision avoidance have matured enough to allow for dynamic and exciting games. In this paper, the three winning Humanoid League teams from the KidSize, TeenSize, and AdultSize class present their soccer systems. The KidSize winner team DARwIn used the recently introduced DARwIn-OP robot. The TeenSize winner NimbRo used their self-constructed robots Dynaped and Bodo. The AdultSize Louis Vuitton Best Humanoid Award winner CHARLI detail the technology behind the outstanding performance of its robot CHARLI-2.

1 Introduction

In the RoboCup Humanoid League, mostly self-constructed robots with a human-like body plan compete with each other on the soccer field. The league comprises three size classes: KidSize (<60 cm), TeenSize (100-120 cm) and AdultSize (>130 cm). While the KidSize robots are playing 3 vs. 3 soccer games, the TeenSize robots started to play 2 vs. 2 soccer games in 2010, and the AdultSize robots engage in 1 vs. 1 Dribble-and-Kick competitions. In addition, all three classes face technical challenges, like dribbling the ball through an obstacle course, double-passing, and throw-in of the ball. In this paper, the three winning teams of the RoboCup 2011 championship in Istanbul — DARwIn, NimbRo, and CHARLI — detail their hard- and software approaches to solve the problems of playing humanoid soccer. These include fast and flexible bipedal locomotion, controlling dynamic full-body motions, maintaining balance in the presence of disturbances, robust visual perception of the game situation, individual soccer skills, and team coordination.

2 KidSize Winner Team DARwIn

Team DARwIn is a joint team of the University of Pennsylvania's GRASP lab and Virginia Tech's RoMeLa lab. The DARwIn-OP proved to be a reliable design and became a commercialized product for general robotics research. Complementing Virginia Tech's tradition of humanoid hardware development, the University of Pennsylvania utilizes its long-time experience of RoboCup Standard Platform League participation. With Penn's open source release of its source code [1] and the open source DARwIn-OP, Team DARwIn based its performance on fully open source engineering.

2.1 DARwIn-OP Robot Hardware

We used the DARwIn-OP robot designed by RoMeLa lab as the robotic platform for RoboCup KidSize competition. It is 45 cm tall, weighs 2.8 kg, and has 20 degrees of freedom. It has a web camera for visual feedback, a 3-axis accelerometer and 3-axis gyroscope for inertial sensing. Position controlled Dynamixel servos are used for actuators, which are controlled by a custom microcontroller connected by an Intel Atom based embedded PC at a control frequency of 100Hz. One noticeable feature of our robotic platform is that after years of joint development, it has become reliable enough to be a commercial product produced by Robotis, co., Ltd. This gave us a big logistic advantage as we could work with a significant number of standardized robots.

2.2 Unified Humanoid Robotics Platform

We have been working with a number of the DARwIns for research-based tasks, in addition to robotic soccer. To exploit the commonalities of different platforms and tasks, and to reduce development time, we developed a flexible cross-robot software architecture, of which the main goals are modularity and portability. Every component of this architecture remains individually interchangeable, which ensures that we can easily port code between robots.For the robotic soccer task, we can use basically the same behavioral logic -the high level controller for interacting with the environment - regardless of which humanoid the platform interacts with. For this year's RoboCup, we used the same basic code for five different humanoid platforms: DARwIn-HP and DARwIn-OP for KidSize class, CHARLI for AdultSize class, Nao for standard league, and a Webots model of Nao for the Robostadium simulation league. The overall structure of our software architecture is shown in Figure 1. It consists of two subsystems: A motion subsystem and a vision subsystem, as well as a behavioral logic module which governs high-level behavior. The vision subsystem processes the video stream and extracts vision cues such as balls, goalposts and lines and passes them to the behavioral logic, which controls the high level behavior such as setting walk velocity or initiating special actions such as kicking or diving. Finally, the motion subsystem communicates with robot-specific actuators and sensors, and generates joint trajectories for numerous motions according to behavior

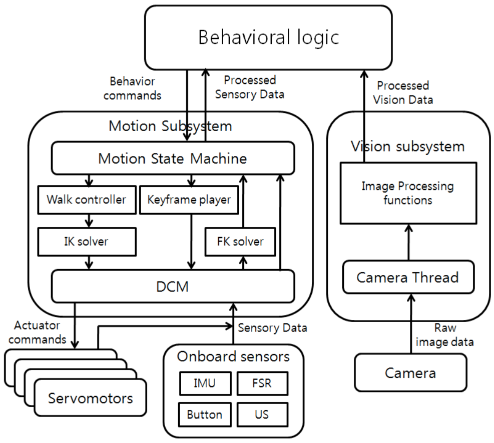

Fig. 1. Block Diagram of the Software Architecture.

commands from the behavioral logic. In addition to robotic soccer, we use this software platform for other research projects. An open source version of our platform can be downloaded at the UPennalizers website[1].

2.3 Walking

We use a zero moment point (ZMP) [2] based walk controller which uses the 3D linear inverted pendulum model (LIPM) to calculate the torso trajectory so that the actual ZMP lies inside the support foot. However, our walk engine has two notable differences. It is not periodic; instead we allow each step to have an arbitrary support foot, walk velocity and step duration. Additionally, we calculate the torso trajectory for each step period using an analytic solution of the ZMP equation assuming a piecewise linear ZMP trajectory. This induces a discontinuous torso velocity at each step transition, but as it occurs during the (most stable) double support phase, it does not hamper the stability much in practice. On the other hand, our approach enables high maneuverability and flexibility. We have achieved a maximum walk speed of 36 cm/s which is very high considering the relatively small size of our robot.

Foot trajectory generation: We divide the walking into a series of steps which can be generally defined as

$$STEP_i = \{SF, t_{STEP}, L_i, T_i, R_i, L_{1+1}, T_{i+1}, R_{i+1}\} \tag{1}$$

where SF denotes the support foot, WT denotes the type of the step, t_{STEP} is the duration of the step, and L_i, T_i, R_i, and $L_{i+1}, T_{i+1}, R_{i+1}$ are the initial

[1] https://fling.seas.upenn.edu/~robocup/wiki/

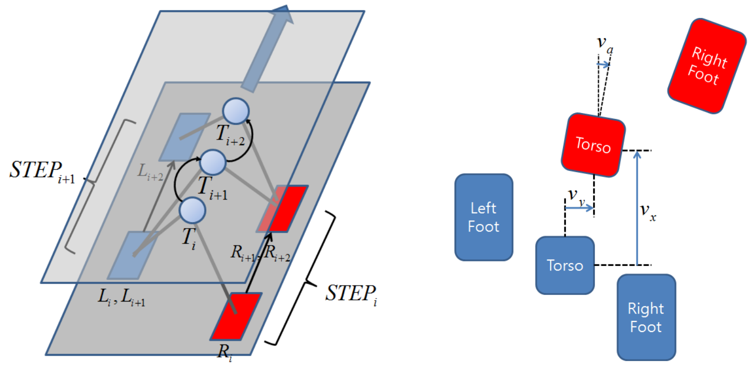

Fig. 2. Walking as a series of steps.

and final 2D poses of left foot, torso and right foot in (x, y, θ) coordinates. The L_i, T_i, R_i poses are determined by the final feet and torso poses from the last step, and L_{i+1}, R_{i+1} are calculated using the commanded walk velocity and current foot configuration to enable omnidirectional walking. Foot reachability and self-collision constraints are also taken into account when calculating target foot poses. We use a FIFO queue structure to handle special sequences of steps, such as dynamic kick. When the current step $STEP_k$ is over, a new step $STEP_{k+1}$ is determined, and foot trajectories for the new step are generated accordingly.

Torso trajectory generation: After generating foot trajectories, the torso trajectory should be generated such that the resulting ZMP lies inside the support polygon during the single support phase. In general, we need to solve an optimization problem. However, we opt to use an analytic torso trajectory solution with zero ZMP error assuming the following piecewise linear reference zero moment point (ZMP) trajectory $p_i(\phi)$ for the left support case

$$p_i(\phi) = \begin{cases} T_i(1 - \frac{\phi}{\phi_1}) + L_i \frac{\phi}{\phi_1} & 0 \le \phi < \phi_1 \\ L_i & \phi_1 \le \phi < \phi_2 \\ T_{i+1}(1 - \frac{1-\phi}{1-\phi_2}) + L_i \frac{1-\phi}{1-\phi_2} & \phi_2 \le \phi < 1 \end{cases} \quad (2)$$

where ϕ is the walk phase and ϕ_1, ϕ_2 are the timing parameters determining the transition between the single support and double support phases. This ZMP trajectory yields for following $x_i(\phi)$ solution with zero ZMP error during the step period:

$$x_i(\phi) = \begin{cases} p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} \\ \quad -\phi_{ZMP} m_i \sinh \frac{\phi - \phi_1}{\phi_{ZMP}} & 0 \le \phi < \phi_1 \\ p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} & \phi_1 \le \phi < \phi_2 \\ p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} \\ \quad -\phi_{ZMP} n_i \sinh \frac{\phi - \phi_2}{\phi_{ZMP}} & \phi_2 \le \phi < 1 \end{cases} \quad (3)$$

where $\phi_{ZMP} = t_{ZMP}/t_{STEP}$ and m_i, n_i are ZMP slopes which are defined as follows for the left support case

$$m_i = (L_i - T_i)/\phi_1 \tag{4}$$

$$n_i = -(L_i - T_{i+1})/(1 - \phi_2) \tag{5}$$

The parameters a_i^p and a_i^n can be uniquely determined by the boundary conditions $x_i(0) = T_i$ and $x_i(1) = T_{i+1}$.

Active Stabilization: The physical robot differs much from the ideal LIPM model, and external perturbations from various sources can make the open loop walk unstable. With the DARwIn-OP robot, we have 3 sources of sensory feedback: Filtered IMU angles, gyro rate readings and proprioception information based on joint encoders. We use this information to apply stabilizing torques at the ankle joints, called “ankle strategy.” In addition, we also implemented other human-inspired push recovery behaviors, hip and step strategies, and used machine learning to find an appropriate controller to reject disturbances using those strategies [3]. In spite of some success in controlled situations, we found the learned controller was not reliable enough for competition and we only used the ankle strategy controller. Overall, our robots were very stable during fast walking, but they fell down when colliding into other robots. Implementing a reliable push recovery controller that can prevent falling from collision remains a big challenge.

2.4 Kicking

Instead of using the typical key frame method for kicking, we use the walk engine to generate a set of parameterized kick motions. There are many advantages to this approach. Designing and tuning a new kick is much easier than making a key frame kick in joint space, the active stabilization can be used during kicking, and kicking can be seamlessly integrated with walking. Utilizing the walk engine also allows us to perform a dynamic kick. Typically, the robot would put its center of mass (COM) within its support polygon during kicking to be statically stable. Instead, the robot puts its ZMP within the support polygon to be dynamically stable during kicking.

Static Kick: To allow our step-based walk engine to execute static kicks, we first define a kick as a sequence of kick steps $KICK_i$

$$KICK_i = \{SF, t_{STEP}, L_i, T_i, R_i, L_{1+1}, T_{i+1}, R_{i+1}\} \tag{6}$$

which has the same format as $STEP_i$ but has 6D coordinate$(x, y, z, \psi, \theta, \phi)$ for L, T, R. Each kick step corresponds to an elementary action during the kick such as lifting, kicking and landing. With the help of this simple operation space definition for kicking, we can easily make and test a number of different kicks in a short time. The frontal kick we used for our matches consists of 7 kick steps and takes approximately 4 seconds to complete, and can kick balls up to 5 meters.

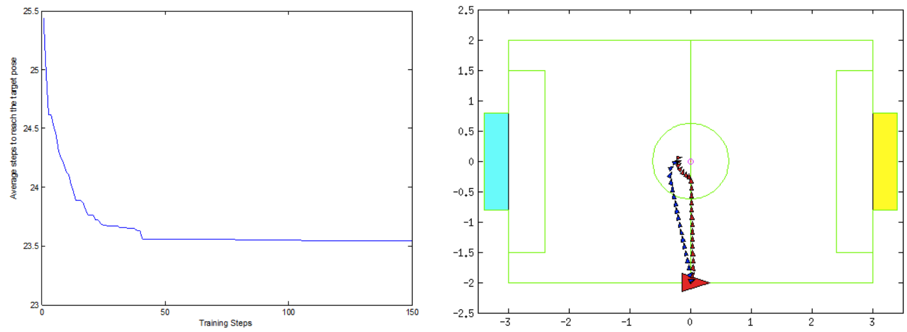

Fig. 3. Learning curve of the locomotion planning problem using PGRL and comparison of resulting approaching policy to rotate-chase-orbit strategy.

Dynamic Kick: Static kicks can be powerful, but the main disadvantage is that it requires a longer time to stabilize before beginning to walk again. However, when opponents are nearby, kicking fast is much more important than kicking strong. Thus, we need a dynamic kick, which can be much faster as it does not require a complete stop and static balancing. Dynamic front kick is implemented by putting two steps in the step queue, support step and kick step. For kick step we use longer step period and special foot trajectory so that it can maximize the foot velocity at hitting the ball. After the robot kicks the ball, the step queue is emptied and it resumes walking according to its commanded walk velocity without stopping. Similarly, the dynamic side kick consists of three steps including two normal steps and one special step.

We have found that as the body is also moving forward, the dynamic front kick has more range than its static kick counterpart. It can be executed very fast, too – it takes 3 steps in worst case which takes 0.75 sec. The main disadvantage is the weak kick strength: in most cases, the dynamic kick cannot shoot more than 1.5 meters. However, as the robot completes its kick way faster and it is moving forward during kicking, it can quickly catch up to the ball and kick again. We have tested several different tactics for choosing between dynamic and static kicks, and we have found fast dynamic kicks are much more effective against good teams. There is less probability of kicking out of bounds, and there is the unexpected side effect of deceiving enemy goalies.

2.5 Optimal Approaching

One challenge for robotic soccer is to arrive at the target pose in the shortest time possible while satisfying all the locomotion constraints. The basic strategy for arriving at the target pose is the rotate-chase-and-orbit strategy, which approaches the ball in straight path and orbits around the ball until it reaches the target pose. However, this is actually a motion planning problem, and can be formalized as a reinforcement learning problem with state S, action A and reward R:

$$\begin{aligned} &S = \{r, \theta_{ball}, \theta_{goal}\} \\ &A = \{v_x, v_y, v_a\} \\ &R = 100 \; at \; target \; pose, \; -1 \; otherwise \end{aligned} \tag{7}$$

We approximate the policy function by a heuristic controller with 5 continuous parameters and train it using a policy gradient RL algorithm. After 150 episodes of training, average steps to reach the target pose decreased by about 25% compared to baseline rotate-chase-orbit strategy. During the match, we found an unexpected side effect of this strategy – it tends to make the robot reach the kicking position first and then turn to face the goal, which effectively blocks opponents' kick towards our goal.

3 TeenSize Winner Team NimbRo

Team NimbRo has a long and successful history in RoboCup with overall nine wins in international Humanoid League competitions since 2005. In 2011, our team won the TeenSize class for the third time in a row. Our robots also won the 2011 technical challenges with a good performance in the Double Pass and the Obstacle Dribbling challenges. This year, the main rule change in the TeenSize class was an increase in size of the field to 9×6 m. We successfully adapted our system to the larger field size and repeated last year's reliable performance in the finals without a single fall and without the need for human intervention. In the remainder of this section, we describe the mechanical and electrical design of our robots, the visual perception of the game situation, and the generation of soccer behaviors in a hierarchical framework.

3.1 Mechatronic Design of NimbRo TeenSize Robots

Fig. 4 shows our two TeenSize robots: Dynaped and Bodo. Their mechanical design focused on simplicity, robustness, and weight reduction.

Fig. 4. RoboCup 2011 TeenSize finals: NimbRo vs KMutt Kickers. Our team played with the robots Dynaped (striker) and Bodo (goalie).

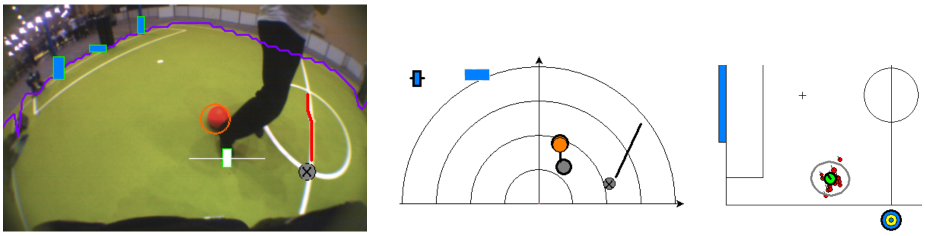

Fig. 5. NimbRo perception and localization. Left: TeenSize field with detected goal, ball, obstacle, X-crossing and center line. Center: Egocentric world view of the robot. Right: Localization given the perceived landmarks.

Dynaped is 105 cm tall and weighs 7.5 kg. The robot has 13 DOF: 5 DOF per leg, 1 DOF per arm, and one joint in the neck that pans the head. Its legs include parallel kinematics that prevents the robot from tilting in sagittal direction. Dynaped's leg joints are driven by master-slave pairs of Robotis Dynamixel EX-106 actuators. Bodo is 103 cm tall and has a weight of about 5 kg. The robot is driven by 14 Dynamixel actuators: six per leg and one in each arm.

The robot skeletons are constructed from rectangular milled aluminum tubes. The feet are made from sheets of composite carbon and heads are produced by 3D printing of polymer material. Both robots are protected against mechanical stress by a 'mechanical fuse' between the hip and the spine. This mechanism includes a pre-loaded spring that yields to large external forces. Together with foam protectors, it allows the robots to dive quickly to the ground as a goalie [4].

The robots are controlled by a pocket size PC, a Sony Vaio UX, which features an Intel 1.33 GHz ULV Core Solo Processor. This PC runs computer vision, behavior control, motion generation, and WLAN communication. The robots are also equipped with a HCS12X microcontroller board, which manages the detailed communication with all joints via an RS-485 bus. The microcontroller also reads in a dual-axis accelerometer and two gyroscopes. The robots are powered by high-current Lithium-polymer rechargeable batteries, which last for about 20 minutes of operation.

3.2 Proprioception, Visual Perception and Self-Localization

The perception of humanoid soccer robots can be divided into two categories: proprioception and computer vision. For proprioception, we use the joint angle feedback of the servos and apply it to the kinematic robot model using forward kinematics. Additionally, we fuse accelerometer and gyroscope measurements to estimate the tilt of the trunk in roll and pitch direction. Knowing the attitude of the trunk and the configuration of the kinematic chain, we rotate the entire model around the current support foot such that the attitude of the trunk matches the angle we measured with the IMU. This way, we obtain a robot pose approximation that can be used to extract the location and the velocity of the center of mass.

Temperatures and voltages are also monitored for notification of overheating or low batteries, respectively.

For visual perception of the game situation, we capture and process 752×480 YUV images from a IDS uEye camera with fish eye lens (Fig. 5 left). Pixels are color-classified using a look-up table. In down-sampled images of the individual colors, we detect the ball, goal-posts, poles, penalty markers, field lines, corners, T-junctions, X-crossings, obstacles, team mates, and opponents using size and shape information. We estimate distance and angle to each detected object by removing radial lens distortion and by inverting the projective mapping from field to image plane (Fig. 5 center). To account for camera pose changes during walking, we learned a direct mapping from the IMU readings to offsets in the image. We also determine the orientation of lines, corners and T-junctions relative to the robot.

We track a three-dimensional robot pose (x, y, θ) on the field using a particle filter [5] (Fig. 5 right). The particles are updated using a motion model which is a simple linear function of the gait velocity commanded to the robot. Its parameters are learned from motion capture data [6]. The weights of the particles are updated according to a probabilistic model of landmark observations (distance and angle) that accounts for measurement noise. To handle unknown data association of ambiguous landmarks, we sample the data association on a per-particle basis. The association of field line corner and T-junction observations is simplified using the orientation of these landmarks. Further details can be found in [7].

3.3 Hierarchical Reactive Behavior Control

We control our robots using a framework that supports a hierarchy of reactive behaviors [8]. This framework allows for structured behavior engineering. Multiple layers that run on different time scales contain behaviors at different abstraction levels. When moving up the hierarchy, the update frequency of sensors, behaviors, and actuators decreases. At the same time, they become more abstract. Raw sensor input from the lower layers is aggregated to slower, abstract sensors in the higher layers. Abstract actuators enable higher-level behaviors to configure lower layers in order to eventually influence the state of the world.

Currently, our implementation consists of three layers. The lowest, fastest layer is responsible for generating motions, such as walking, kicking and the goalie dive. Our omnidirectional gait [9] is based on rhythmic lateral weight shifting and coordinated swinging of the non-supporting leg in walking direction. This open-loop gait is self-stable when undisturbed. In order to reject larger disturbances, we recently extended our gait engine with a lateral capture step controller [10] that modifies the timing and the lateral location of the footsteps to maintain balance. This controller uses a linear inverted pendulum model to predict the motion of the robot's center of mass. For the goalie, we designed a motion sequence that accelerates the diving motion compared to passive sideways falling from an upright standing posture [4]. The goalie jump decision is based on a support vector machine that was trained with real ball observations.

At the next higher layer, we abstract from the complex kinematic chain and model the robot as a simple holonomic point mass that is controlled with a desired velocity in sagittal, lateral and rotational directions. We are using a cascade of simple reactive behaviors based on the force field method to generate ball approach trajectories, ball dribbling sequences, and to implement obstacle avoidance.

The topmost layer of our framework takes care of team behavior, game tactics and the implementation of the game states as commanded by the referee box.

4 AdultSize Winner Team CHARLI

Stemming from the success of team DARwIn in the KidSize class, team CHARLI has participated in the AdultSize division since its beginning at RoboCup 2010. Because the AdultSize class is still relatively new, the main focus of most teams in this division, including team CHARLI, has been on the development of a stable bipedal walking platform to serve as a basis for autonomy. At RoboCup 2010, we introduced our first adult-size platform, CHARLI-L (Cognitive Humanoid Autonomous Robot with Learning Intelligence — Lightweight), and at RoboCup 2011 we introduced a second version named CHARLI-2. Both robots share an emphasis on lightweight design in order to reduce costs, increase safety during use, and improve ease of handling. These characteristics are especially important for robotics competitions such as RoboCup. This paper presents some of the innovative mechanical design features of CHARLI-L and CHARLI-2, as well as an overview of the system architecture followed by our vision for future generations of the platform.

4.1 Mechanical System

CHARLI-L: One of the main goals of the mechanical design of CHARLI-L was to minimize the weight of the overall system by utilizing lightweight off-the-shelf actuators in the lower body. The challenge is that, when used in a conventional

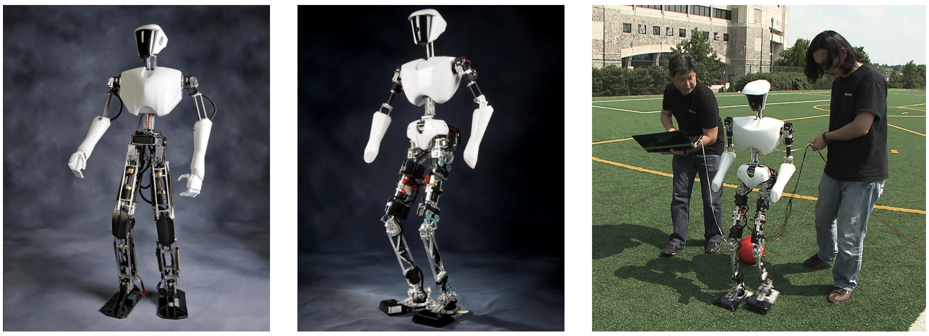

Fig. 6. CHARLI-L (left) and CHARLI-2 (middle and right, on a soccer field).

configuration, these actuators cannot produce enough torque for the application. Three features that enabled CHARLI-L to use such actuators were: a parallel four-bar linkage design for the legs, a synchronized actuator configuration, and tension springs to reduce the required actuator torque at the joints.

Fig. 7 shows the parallel four-bar linkage and orientation of the actuators. Unlike conventional adult-size humanoids, CHARLI-L does not use gear reduction mechanisms such as harmonic drives. Instead, multiple EX-106+ Dynamixel actuators are used in tandem to actuate each joint. Assuming the two actuators move simultaneously with identical torque, the overall torque is doubled. The parallel four-bar configuration makes the packaging of such a configuration easy to implement as two actuators can actuate a revolute joint each in the single kinematic chain. The design of the CHARLI-L's parallel four-bar linkage is such that each foot is constrained to be parallel to the ground at all times, enabling a walking gait with only 5 DOF. [11] [12] Eliminating one of the degrees of freedom of the leg resulted in further weight reduction, as fewer actuators were needed. Another advantage of the parallel four-bar approach was the ease of implementation of a tension spring to provide additional torque. The configurations of these springs are such that, when CHARLI-L's leg supports the upper body during walking, the resulting tensile force reduces the required torque of the actuators as shown in Fig. 7 (right). Using this approach, CHARLI-L was able to achieve stable walking using off-the-shelf components, while reaching an overall weight of only 12.7 kg.

CHARLI-2: Although the innovative mechanical design of CHARLI-L proved successful, a new design was chosen for the following version in order to investigate the benefits of a different approach. Thus, CHARLI-2 utilizes a more conventional serial chain configuration instead of the previous four-bar configuration. There were four main reasons behind this decision. First, although the four-bar configuration was thought to reduce the overall weight by eliminating a set of actuators, it turned out that the additional linkages outweighed the extra actuators. Second, the elimination of an active DOF limited the motion of the foot, constraining

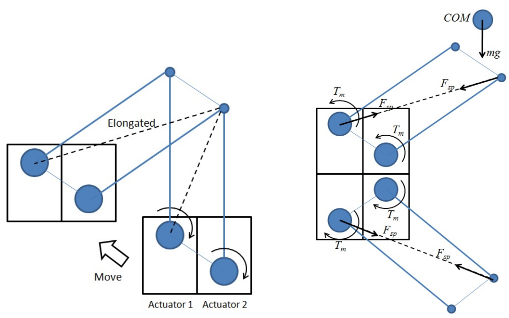

Fig. 7. Left: Concept of the spring assisted parallel four-bar linkage with synchronized actuation. Right: Diagram of CHARLI-L's leg.

the possible walking strategies. Third, although the tension springs did indeed help reduce the required torque at the joints, the nonlinear behavior of springs and hysteresis caused problems for the control algorithms. Lastly, a new walking approach under development required more torque than the previous design was able to provide. [13] [14]

To address these issues, CHARLI-2 was designed using a serial configuration with 6 active DOF per leg; furthermore, an additional gear train was added between the output of each pair of actuators and the output stage replacing the four-bar configuration. Fig. 8 shows the gear train of CHARLI-2's knee. The reduction ratio of this gear train is 3:1. Two identical EX-106+ actuators rotate in tandem to actuate the joint. Using this configuration, the maximum holding torque of the joint can reach 60 Nm — twice the required torque for normal walking. The hip roll and pitch joints are implemented using a similar gear system. For the other joints (i.e. the hip yaw, ankle roll and ankle pitch), a single EX-106+ actuator was used, as these joints require less torque (under 20 Nm). The result of this design change was a reduction in overall weight to 12.1 kg, and in increase in torque from 30 Nm to 60 Nm (for the knee). Additionally, the new design eliminated the control difficulties associated with the use of springs in the previous version. This led to a significant increase in performance and stability. Currently, CHARLI-2 can walk at a speed of up to 0.4 m/s, and with further optimization of the walking algorithm, we believe that this can be increased by 20%.

4.2 Electronic System

CHARLI-2 shares a common system architecture with our KidSize humanoid robot platform, DARwIn-OP. All high-level processing and control is performed on an Intel-based PC running GNU/Linux. A ROBOTIS Co. CM-730 sub-controller board acts as the communication relay between the Dynamixel actuators and PC, providing services for both sensor acquisition and actuator control. CHARLI-2's software architecture is based on the humanoid robotics framework employed by team DARwIn for the KidSize competition. (Please refer to the section regarding DARwIn-OP's software design.) Due to the differences in scale between the two robots, a specialized motion module was developed for CHARLI to allow stable ZMP-based walking.

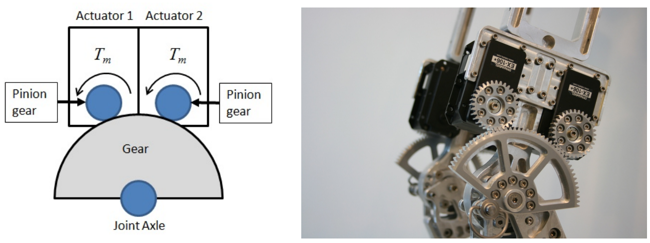

Fig. 8. Gear train of CHARLI-2's knee. Left: Design concept. Right: Real part

4.3 Future of CHARLI

The research of Virginia Tech's team CHARLI (and team DARwIn) is founded on innovative platform development. Against conventional wisdom, we have developed a brand new platform from scratch every year we have participated in RoboCup. This process produced two high performance platforms, leading to first place wins in both the AdultSize class (CHARLI-2) and KidSize class (DARwIn-OP). As a contribution to the robotics community, we have made the successful DARwIn-OP platform completely open source, both in hardware design and software, through sponsorship by the National Science Foundation and through collaboration with Purdue University, University of Pennsylvania, and ROBOTIS Co. We are considering doing the same for a future version of CHARLI, as we believe the open source approach is the quickest and most effective way to accelerate the development of robotics technology.

5 Conclusions

Robotic soccer, like human soccer, relies on a combination of the overall team strategy and individual skill sets. The 2011 competition showed notable progress in both areas.

In KidSize, for example, team DARwIn mainly focused on the individual skill of a striker, such as the ability to take possession of the ball and to move it as fast as possible. This strategy required developing algorithms such as quick omnidirectional walking, dynamic kicking and optimal approaching. As the KidSize league is inherently the most dynamic league among RoboCup leagues, DARwIn will keep focusing on extending the dynamic behavior of their robots.

From their experience with a broad range of humanoid robots, DARwIn developed an open-source, common software structure for humanoid robots. With its help, the team was able to save time preparing for different RoboCup leagues – and to win two championships in KidSize and Adult Size. The open-source release of their code base, in addition to the commercial release of the DARwIn-OP robot, will certainly help other RoboCup teams and stimulate general robotic research using small humanoid robots. A similar open platform is desirable for TeenSize and AdultSize.

In TeenSize, team NimbRo showed a very stable robot that played several matches without falling. For 2012, the team is developing a new robot that will be able to get up after a fall, and perform throw-ins. The team is working to improve robot balance after pushes and collisions and to integrate footstep planning into their behavior architecture.

In AdultSize, team CHARLI demonstrated reliable dribbling and kicking. Their performance was acknowledged by the team leaders with the Louis Vuitton Best Humanoid Award. Team CHARLI (and team DARwIn) are committed to continue the introduction of brand new platforms every year with innovative features and approaches which depart from the conventional. The team plans to introduce new effective walking strategies using compliant linear actuators with force control, and to demonstrate safe falling and recovery in the near future.

In the future, the Humanoid League will continue to raise the bar. Possible rule changes include enlarging the field, increasing the number of players, reducing color-coding of objects, soccer games for AdultSize robots, and new technical challenges, such as kicking the ball high. This will keep the competition challenging and will contribute towards the ultimate goal of humanoid robots playing soccer with humans.

References

1. S. G. McGill, J. Brindza, S.-J. Yi, and D. D. Lee. Unified humanoid robotics software platform. In *The 5th Workshop on Humanoid Soccer Robots*, 2010.
2. M. Vukobratovic and B. Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics (IJHR)*, pages 157–173, 2004.
3. S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee. Learning full body push recovery control for small humanoid robots. In *Proceeding of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2047 –2052, 2011.
4. M. Missura, T. Wilken, and S. Behnke. Designing effective humanoid soccer goalies. In *Proceedings of 14th International RoboCup Symposium*, 2010.
5. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2001.
6. A. Schmitz, M. Missura, and S. Behnke. Learning footstep prediction from motion capture. In *Proceedings of 14th International RoboCup Symposium*, 2010.
7. H. Schulz, W. Liu, J. Stückler, and S. Behnke. Utilizing the structure of field lines for efficient soccer robot localization. In *Proceedings of 14th International RoboCup Symposium*, 2010.
8. S. Behnke and J. Stückler. Hierarchical reactive control for humanoid soccer robots. *International Journal of Humanoid Robots (IJHR)*, 5:375–396, 2008.
9. S. Behnke. Online trajectory generation for omnidirectional biped walking. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2006.
10. M. Missura and S. Behnke. Lateral capture steps for bipedal walking. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2011.
11. A. Hamon and Y. Aoustin. Cross four–bar linkage for the knees of a planar bipedal robot. In *Proc. of 10th Int. Conf. on Humanoid Robots (Humanoids)*, 2010.
12. H. Choi and Y. Park. Development of a biped walking robot actuated by a closed-chain mechanism. *Robotica*, 24(1):31–37, 2006.
13. J. Han and D. Hong. Development of a full–sized bipedal humanoid robot utilizing spring assisted parallel four–bar linkages with synchronized actuation. In *IDETC/CIE Conference*, 2011.
14. S. Song, Y. Ryoo, and D. Hong. Development of an omni–directional walking engine for full–sized lightweight humanoid robots. In *IDETC/CIE Conference*, 2011.

Acknowledgements: The authors would like to recognize the National Science Foundation and the Office of Naval Research for partially supporting this work through grants CNS 0958406 and ONR 45006.