Team THORwIn

Team Description for Humanoid AdultSize League of RoboCup 2014

Stephen McGill, Seung-Joon Yi, Larry Vadakedathu, Qin He, Dennis Hong and Daniel Lee

GRASP Lab
School of Engineering and Applied Science, University of Pennsylvania, USA
Email: ddlee@seas.upenn.edu
URL: http://www.seas.upenn.edu/~robocup/

RoMeLa
Mechanical & Aerospace Engineering, UCLA, USA
Email: dennishong@ucla.edu

Abstract:

This paper details the hardware, software and electrical design of our new humanoid platform for the adult size league at RoboCup 2014 to be held later this year in João Pessoa, Brazil. THORwIn is a 49Kg bipedal robot that will show our experience from previous RoboCup encounters with Humanoid platforms like DARwIn-OP, CHARLI, and XOS in robot soccer autonomy since RoboCup 2007 [1,2].

1. Introduction:

THORwIn or Tactical Hazardous Operations Robot with Intelligence is a low cost, modular humanoid robot designed for testing various algorithms in scenarios varying from disaster response, firefighting on a ship and now for humanoid soccer. This robot shows modularity in both hardware and software. The hardware consists of standardized and general purpose actuators and structural components. University of Pennsylvania's modular software framework consists of a hybrid locomotion controller, a hierarchical arm controller and a platform independent operator interface. The ultimate goal of humanoid robots is to work in environments designed for humans and RoboCup aims to help achieve this goal by strictly improving challenges each year.

Standing not so tall at 1.47 m and weighing just 49 kg, THORwIn has 30 degrees-of-freedom powered by Dynamixel PRO servo actuators that use cycloidal reduction gears in them to have higher impact tolerance compared to harmonic drives.

RoboCup 2014 will be the first time in all our years of participation that we will be using such an advanced humanoid platform for soccer in the adult size league. Previously, we used CHARLI, DARwIn-OP, and XOS in the adult, kid and teen size leagues respectively with very successful results. Our software architecture is based off of our three time champion DARwIn-OP robot's codebase but is much cleaner and faster now with various upgrades.

Team THOR would like to commit to participate in the RoboCup 2014 Humanoid League Competition – Adult Size League. We are also able to have members of our team volunteer as referees as they have sufficient knowledge of rules, even the updated ones for this year's competition at João Pessoa.

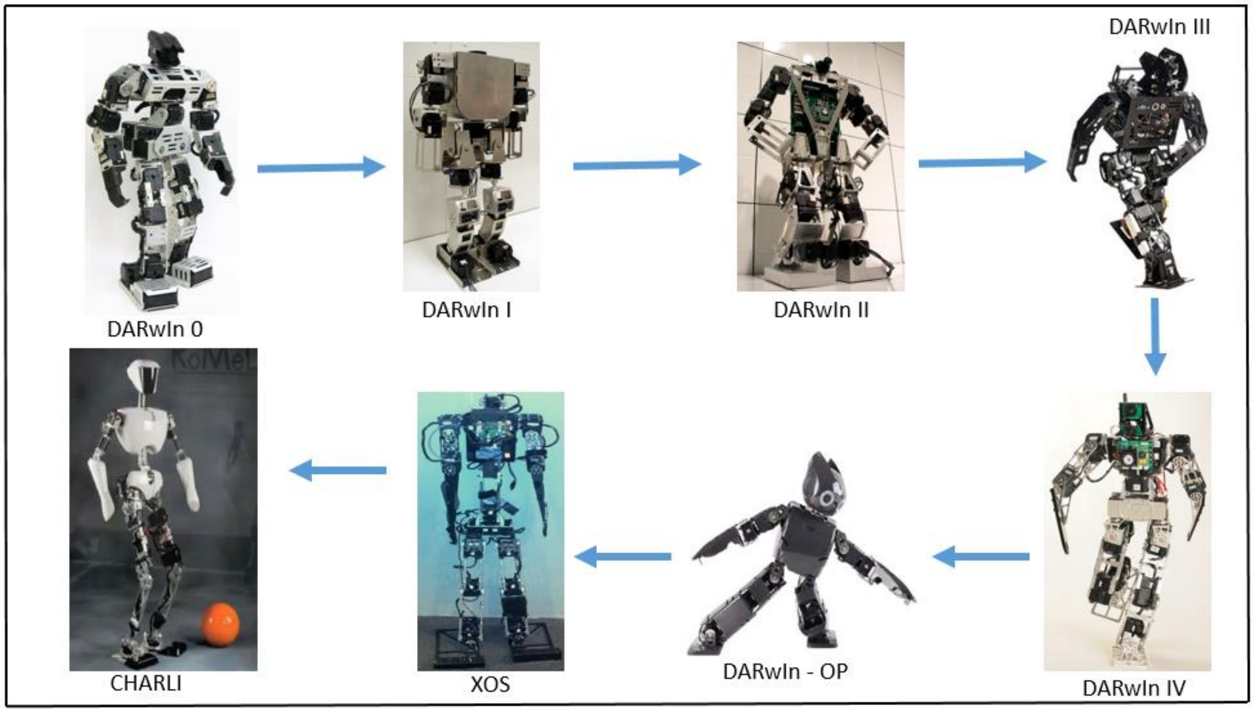

Fig 1: Lineage of humanoid robots from RoMeLa and University of Pennsylvania.

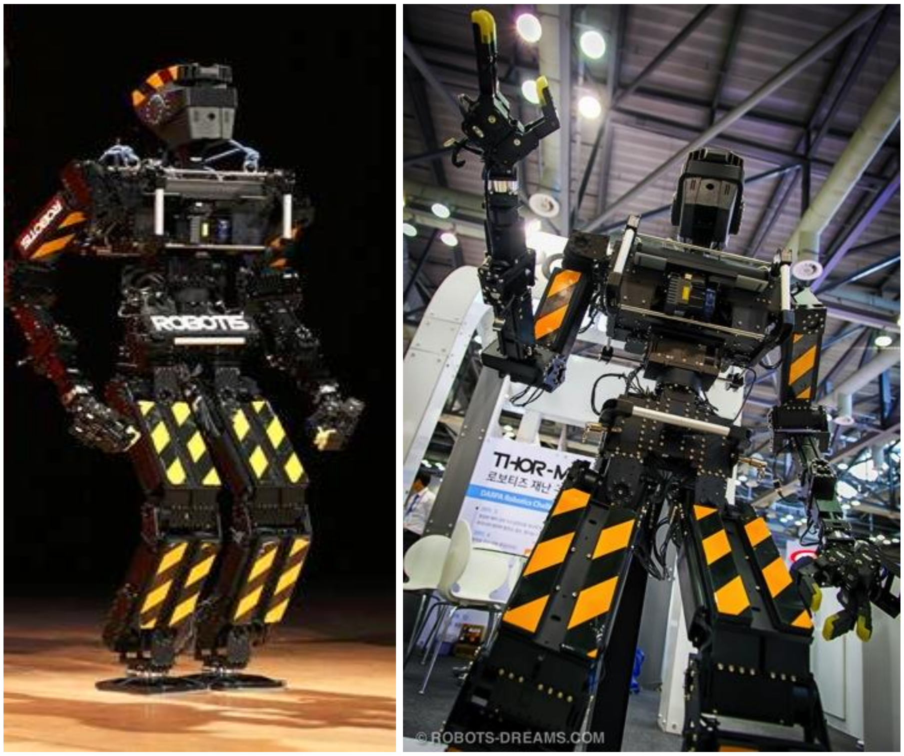

Fig 2: ROBOTIS's THORwIn/THOR-MANG

2. Research:

Team THORwIn has ample research experiences with successes in the humanoid robot arena, and more so with robot soccer. Our collaboration with UCLA’s (formerly Virginia Tech’s) RoMeLa over the past several years furthered humanoid research to very good levels not just at RoboCup but other projects from DARPA (Defense Advanced Research Projects Agency) and the ONR (Office of Naval Research).

Hybrid Walk Controller:
We use a hybrid locomotion controller [3] which can dynamically switch between a standard ZMP preview controller using linear quadratic optimization and a reactive ZMP-based controller which uses closed solution of the linear inverted pendulum equation. The main benefit of this approach is that it provides a less computationally expensive, latency-free locomotion.

Push Recovery:
To cope with unexpected perturbations, we have added two different types of push recovery control [4] that use the current state estimation from IMU and joint encoders.

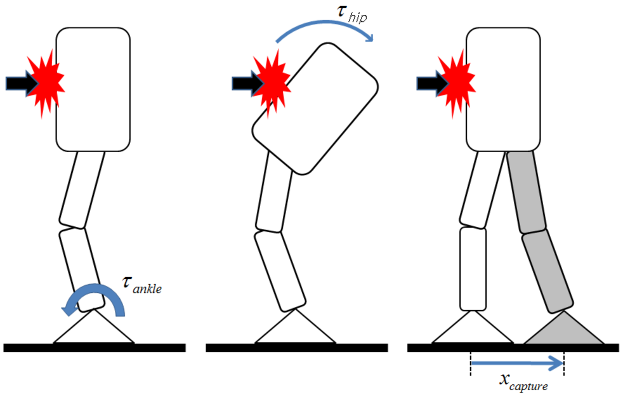

Fig 3: Push recovery

An ankle strategy is implemented by controlling the target joint angle of the position controlled ankle joint, which generates a control torque as a result, and the stop reflex strategy stops walking and lowers the center of mass to resist large perturbation.

Graphical User Interface:
Command line scripts lack an ability to display rich sensor information; to this end, we manufactured an HTML5 based graphical user interface (GUI). Browser support allowed for cross platform development and during the competition, we operated iOS, OSX, and Linux platforms simultaneously, with one master NodeJS [5] server to bridge ZeroMQ messages and raw UDP packets from the robot to WebSockets messages for the browser. The browser's JavaScript console helped to identify network conditions and overall debugging. We will use this UI to monitor the status of the robot (localization, vision, battery level etc.) during each two-and-half minute run at RoboCup 2014.

We have found that our software framework demands surprisingly little amount of computational power, averaging only 25% of the CPU time of a single onboard computer.

3. Hardware:

The robot consists of 30 actuators, 7 in each pair of arms, 6 in each pair of legs, 2 in the torso, 2 for the head.

The robot stands 1.47m tall, weighs 49kg and has a wingspan of approximately 1.95m. The arms and legs have shock absorbing padding to protect the robot in the event of a fall and there is a roll cage around the upper body both for protecting the sensors and computer and for ease of handling.

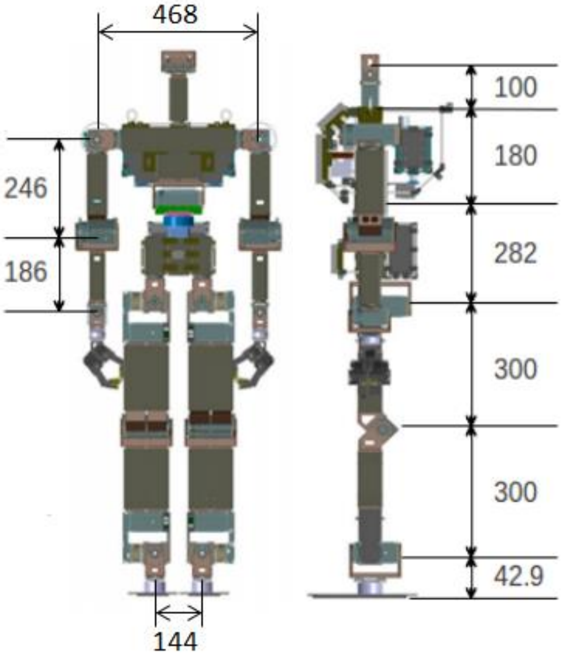

Fig 4: THORwIn Dimensions (in mm)

For actuators, the new series of Dynamixel PRO servomotors developed by Robotis, Co. Ltd are used at most joints. THORwIn uses three different motor types: H42-20-S300-R, H54-100-S500-R and H54-200-S500-R. They are rated at 20W, 100W and 200W respectively and can be fitted with a number of different reduction gear boxes. For the THORwIn platform, two different gearboxes are used, one with in-line output axle and one with parallel output axle. Both gearboxes use cycloidal reduction gears that have higher impact tolerance than common harmonic drives. Communication with the servos is done via four RS485 buses.

4. Electronics and Battery Power:

The overall system structure and interfacing on THORwIn is shown below. Two Axoimtek computers with a 1.65GHz dual core AMD G-series APU were available to perform high level computation, although only one was used in competition. Communication from these computers to the actuators was divided into four independent chains of RS485s over one USB to RS485 interface board. The robot operates on 24V for computation and actuators. The maximum current the robot has drawn at any time is 20A; for this reason, the two main computer reside on a separate supply to mitigate harmful current spike effects. With this separation, the required emergency stop switch kills only the servos when required, leaving the computer unaffected. The main sensors we use are a Logitech C920 HD head camera and MicroStrain 3DM-GX3-45 Inertial Measurement Unit.

THORwIn is capable of functioning completely under its own battery power. In fact, we use two 6-cell 11A LiPo batteries for the actuators and one 6 cell 3.25A LiPo battery for the computer. With these, the run time is close to 1 hour depending on the actions performed by the robot.

5. Software Architecture:

There are many advantages to modular software design. Each module performs logically discrete functionality, built separately from each other. When assembled together in proper hierarchy, they constitute the application program. This type of system is very reusable and extendable; in fact, we reused many modules from our own open-source code base for RoboCup [6] to save time and effort.

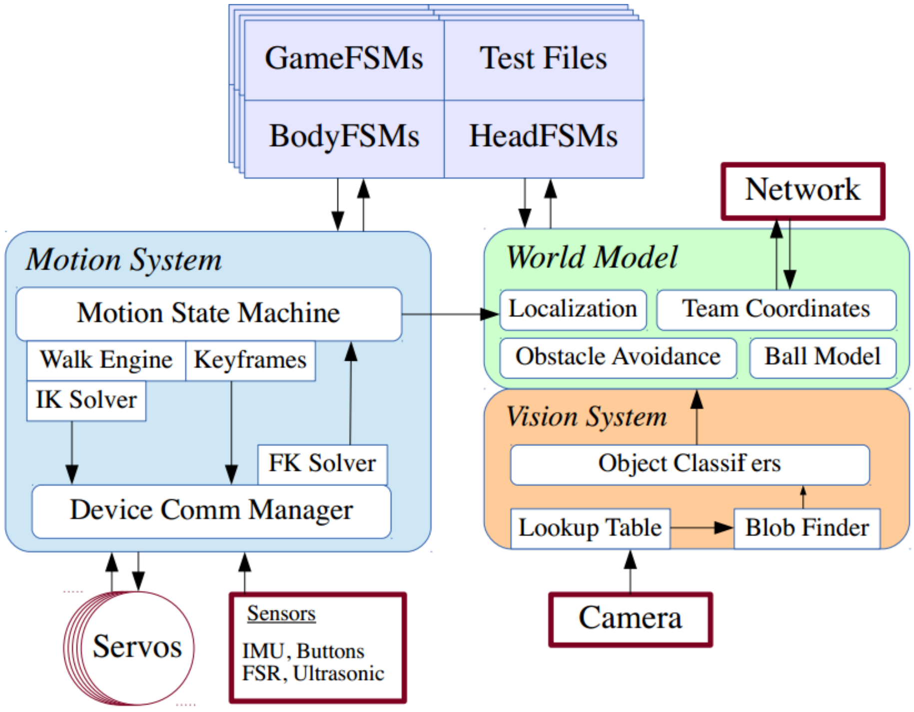

Fig 5: Software Architecture

We modularized our software architecture in several ways. First, we set aside processes to interface with each hardware device, including cameras, laser scanners, motors, and human interface devices. Each process had a twin sub-process in our simulation environment driver, such that no code modifications were needed to run the robot in simulation or on real hardware. Furthermore, we made a module, with a clear application programming interface (API), that translated generic robot commands and queries into robot specific kinematics and motor packets, with the intention that a new robot platform needs only a similar module conforming to the API. Next, we constructed state machine modules that ran independently in a behavior process -- ones for the overall body, the arms, the legs, the head, and the laser scanner. Finally, we formed operator modules to communicate data and commands with the robot, such that user interfaces could be mixed and matched.

Onboard the robot, communication channels included Boost shared memory segments [7] and ZeroMQ message queues [8]. While these inter-process channel assets would reside on one single computer, we leveraged ZeroMQ's request/reply method (and UDP fallbacks) to provide a remote operator with access. For instance, in cases where the inverse kinematics solver was not able to find solutions for arm trajectories due to odd arm configuration or limits on range of motion being reached, the operator performed high-level arm configuration and even direct joint angle overrides, stored in shared memory. For high bandwidth sensor feeds, the user could dynamically select between UDP and ZeroMQ's TCP PUB/SUB methods. All metadata was serialized via MessagePack [9].

6. Vision:

The eyes or eye of this adult size robot is a Logitech C920 HD camera. In our vision system, we use a look up table to first categorize raw camera pixels into one of 8 color labels. The goal of the colortable generating utility is to provide a supervised learning approach to classify raw camera pixels into a set of 8 color class labels - possibly green for the field, orange for the ball, etc. Typically, we receive a YCbCr color formatted pixel array from the camera, and convert this to RGB for displaying to the user. Figure 3 shows the colortable generating tool for a real robot camera frame, and how the generated lookup table performs in simulation.

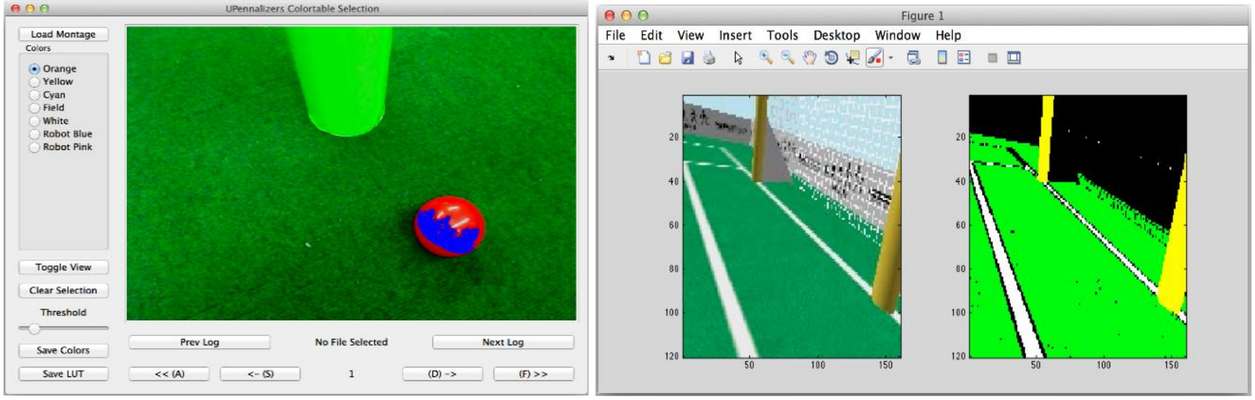

Fig 6: Colortables can be made with a QT user interface for classifying ball colors (left). The generated lookup table can be monitored in MATLAB (right).

The user clicks on pixels that belong to each of these labels to generate both positive and negative examples of the color. We then apply a Gaussian mixture model to generate color segments. For each mixture, we perform a threshold to find pixels with a high probability of belonging to that color class mixture. With a high enough score, we add that pixel to a lookup table mapping to the color class. This lookup table from the mixture model to color class saves on per pixel computation, since the probability does not need to be computed for each pixel on every incoming frame; down_sampling further increases speed. This labeled image is fed into high level blob detection routines and object classifiers.

7. Locomotion Control:

Overall, the locomotion of the robot can be controlled in three different modes. The *direct* mode, which uses the latency-free analytic ZMP controller, can control the velocity of the robot for every step. The next step position is calculated based on the current foot configuration, the target velocity and kinematic constraints. The *preview* mode uses the target pose of the robot to generate a number of optimal steps to reach the pose, and uses the preview controller for locomotion. Finally, the *special* mode is involved for special cases such as stepping over a block. For each mode, the start and end of locomotion is handled by a preview controller for a smooth transition between standing and walking states.

8. Prior Performance in RoboCup:

While Team THORwIn might be a new name, the members of the team are experienced members from Team DARwIn. In general, Team DARwIn has been competing since RoboCup 2007 led by Virginia Tech, and in recent years they partnered with the University of Pennsylvania. Team DARwIn with DARwIn-OPs (Dynamic Anthropomorphic Robot with Intelligence) is the reigning champion in the Kid-Size League since 2011. We are also 2 time Adult Size League champion in 2011 and 2012 with CHARLI (Cognitive Humanoid Robot with Learning Intelligence) from VT. This year, Team THORwIn is led by University of Pennsylvania in collaboration with University of California, Los Angeles (UCLA).

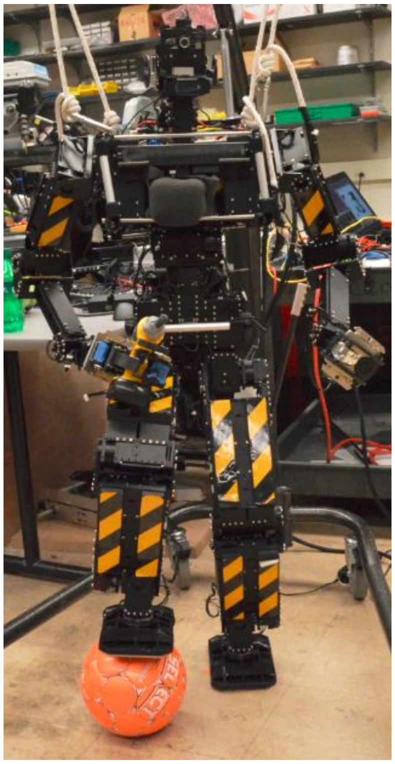

Fig 7: THORwIn gearing up for RoboCup at the GRASP Lab, University of Pennsylvania.

With ample RoboCup Humanoid League experience, we are confident to do well at RoboCup 2014 on this new robot platform which has improved hardware and software.

9. Conclusion

Having a proven background in developing successful humanoid robots, especially in front of international audience of several RoboCup championships, Team THORwIn from University of Pennsylvania vows to further humanoid robot research in locomotion, manipulation and localization by competing to its best abilities in João Pessoa, Brazil. We wish to be a contribution that leads to the ultimate goal of RoboCup which is for our humanoid robots to play against human beings at the highest level.

References

[1] D.Hong, Biologically inspired locomotion strategies: novel ground mobile robots at RoMeLa. *3rd International Conference on Ubiquitous Robots and Ambient Intelligence, Seoul, S.Korea, October 2006*

[2] K.Muecke and D.Hong. Darwin's evolution: Development of a humanoid robot. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pages 2574 – 2575, October – November 2007.*

[3] S.-J. Yi, D. Hong, and D. D. Lee, "A hybrid walk controller for resource-constrained humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots, 2013.*

[4] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, "Online learning of a full body push recovery controller for omnidirectional walking," in *IEEE-RAS International Conference on Humanoid Robots, 2011, pp.1–6.*

[5] Node.js webserver. [Online]. Available: http://www.nodejs.org

[6] 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*.

[7] Boost c++ libraries. [Online]. Available: http://www.boost.org

[8] P. Hintjens. (2010) ZeroMQ: The Guide. [Online]. Available: http://zguide.zeromq.org/page:all

[9] MessagePack serialization library. [Online]. Available: http://www.msgpack.org