UChile RoadRunners 2010 Team Description Paper

Javier Testart, Javier Ruiz-del-Solar, Rodrigo Palma-Amestoy, Isao Parra, Javier Moya, Rodrigo Schulz, Pablo Guerrero, Joaquín Figueroa

Department of Electrical Engineering, Universidad de Chile
Advanced Mining Technology Center, Universidad de Chile
jruizd@ing.uchile.cl
http://www.robocup.cl/humanoid.htm

Abstract. The UChile RoadRunners team is an effort of the Department of Electrical Engineering of the Universidad de Chile. The team participates in the Humanoid League since 2007. This year the team main novelties include: (i) the development of our new humanoid robot UCH H2, (ii) the use of a Atom Z530 1.6GHz based embedded PC running Linux as main processing unit, (iii) a new system for the detection of instability and the avoidance of falls, and (iv) a new real-time hybrid control architecture that incorporates active vision mechanisms.

1 Introduction

The UChile robotics team is an effort of the Department of Electrical Engineering of the Universidad de Chile in order to foster research in mobile robotics. The team is involved in RoboCup competitions since 2003 in different leagues: Four-legged 2003-2007, @Home in 2007-2009, Humanoid in 2007-2009, and Standard Platform League (SPL) in 2008-2009. UChile's team members have served RoboCup organization in many ways: Javier Ruiz-del-Solar was the organizing chair of the Four-Legged competition in 2007, TC member of the Four-Legged league in 2007, TC member of the @Home league in 2009, Exec Member of the @Home league since 2009, President of the RoboCup Chile committee since 2008, and co-chair of the RoboCup 2010 Symposium.

As a RoboCup research group, the team believes that its contribution to the RoboCup community is not restricted to the participation in the RoboCup competitions, but that it should also contribute with new ideas. In this context, the team has published a total of 20 papers in RoboCup Symposia (see table 1), in addition to many other publications about RoboCup related activities in inter-national journals and conferences (some of these works are available in [1]). It is worth mentioning [2][3], two articles that were included in the *Special Issue on Humanoid Soccer Robots* of the *Robotics and Autonomous Systems* journal. Among the main scientific achievements of the group are the obtaining of three important RoboCup awards: *RoboCup 2004 Engineering Challenge Award*, *RoboCup 2007 @Home Innovation Award*, and *RoboCup 2008 @Home Innovation Award*.

Table 1. UChile articles in RoboCup Symposia.

RoboCup Articles	*2003*	*2004*	*2005*	*2006*	*2007*	*2008*	*2009*
Oral	*1*	*2*	*1*	*1*	*2*	*3*	*2*
Poster	*1*	*1*	*1*	*-*	*3*	*2*	*-*

As a group having teams in both, the humanoid and the SPL league, the team can build bridges between both leagues, for transferring some of the developments in distributed control of networked robot from the SPL league to the humanoid league. One of the major ideas behind dividing RoboCup soccer among several leagues was to address specific problems in each league, which later could benefit all leagues. In this line of thought, the team intends to integrate some of the solutions already developed in the SPL league with the hardware and robot control developments of the humanoid league. As an example, the team has developed a real-time hybrid architecture for biped humanoids that was validated first in the Nao robots [4], and it is now in use in the team's robots of the Humanoid and SPL leagues. It is important to mention that UChile's SPL team is one of the 24 that already classified for the RoboCup 2010 world-competitions.

This year the team main novelties include: (i) the development of our new humanoid robot UCH H2, (ii) the use of a Atom Z530 1.6GHz based embedded PC running Linux as main processing unit, (iii) a new system for the detection of instability and the avoidance of falls, and (iv) a new real-time hybrid control architecture that incorporates active vision mechanisms.

2 Hardware

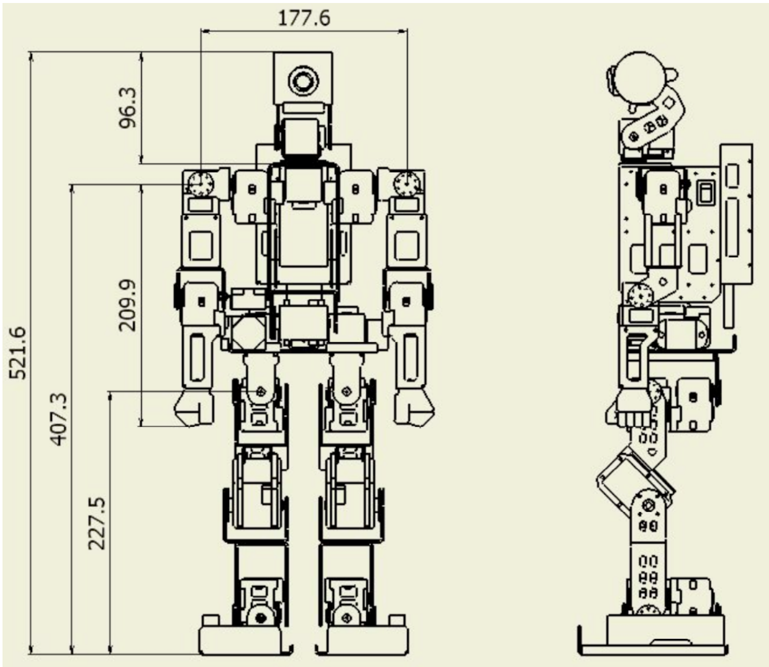
Our team uses two different robot platforms: modified Hajime HR18 robots as field players, and the UCH H2 robot as goalie player (see figure 1). We have made two main modifications to the HR18 robot: (i) we have replaced the original Hajime controller by a controller based on the TMS320F28335 DSP, and (ii) we have included a battery protection and balance circuitry. Both hardware modules have been completely designed and constructed by us. In addition, the UCH H2 robot has been designed and built in our lab during the last year, based on our experience with the UCH H1 robot.

This year we have replace the Fujitsu Siemens n560 Pocket PC by an embedded PC. We are using the Axiomtek PICO820VGA-Z530 embedded PC, which is based on a Intel® Atom processor Z530 1.6GHz, has 2 GB of RAM, and 4GB Flash memory, and runs a Debian Linux. The technical specifications of all these hardware components are shown in table 2.

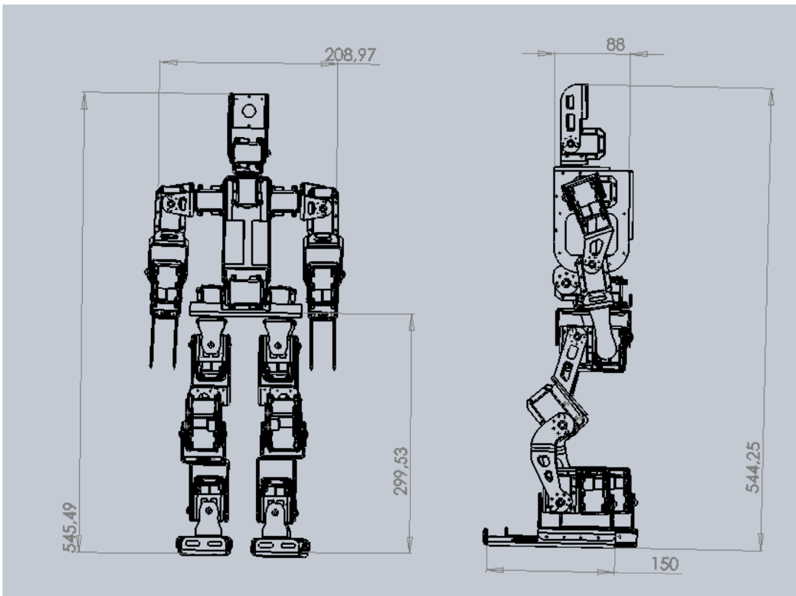
3 Software Architecture

The *UChileLib* robotics library provides several functionalities regarding computer processes, and their synchronization and communication. UChileLib makes use of the Boost Libraries [8], it is OS independent, and it can been compiled and executed in Windows or Linux.

In the library, a *module* is a part of the software with an encapsulated functionality that has a clear and explicit interface with the rest of the software. This interface is defined as the information the module shares with other modules. Each module is implemented as a thread, in order to allow them to work concurrently. Since they live in the same process, they can easily share memory to communicate with each other. Every module can have several connections to other modules. These connections could be used either for sending or receiving information.



(a)



(b)

Fig. 1. Frontal and lateral view of the (a) Hajime HR18 robot, and (b) UCH H2.

Table 2. Hardware specifications.

Player	Field Players (2)	Goalie Player (1)
Robot	Hajime HR18	UCH H2
Height	540 mm	560 mm
Weight	3.7 kg	3.8kg
Servo Motors	18x DX-117	18x RX-28
Dynamixel Robot Actuator	3x RX-64	3x RX-64
Degrees of freedom	21	21
	leg: 6x2	leg: 6x2
	arm: 3x2	arm: 3x2
	waist: 1	waist: 1
	neck: 2	neck: 2
Batteries	2x Li-Po 7.4V, 4000mAh	2x Li-Po 7.4V, 4000mAh
Internal Sensors		
Joint Angle Encoder	21	21
Accelerometer	ADXL320	ADXL320
Gyroscope	IDG-300	IDG-300
Camera	Philip SPC900NC	Philip SPC900NC
Sensor	CCD	CCD
Interpolated snapshot resolution	1.3 Mpixels	1.3 Mpixels
Max. frame rate	90 fps	90 fps
Lens	F2.2, D55°	F2.2, D55°
White balance	2600 – 7600 k	2600 – 7600 k
Min. illuminance	< 1 lux	< 1 lux
Colour depth	24 bit	24 bit
PC Link:	USB 1.1	USB 1.1
Control	Autonomous at frecuency 10 ms	Autonomous at frecuency 10 ms
Internal Controller	32bit DSP TMS320F28335 150MHz	32bit DSP TMS320F28335 150MHz
PC	Axiomtek PICO820VGA-Z530	Axiomtek PICO820VGA-Z530
Processor	Intel® Atom processor Z530 1.6GHz	Intel® Atom processor Z530 1.6GHz
Memory	System memory (RAM) 2 GB 667MHz	System memory (RAM) 2 GB 667MHz
	Compact Flash memory 4GB 133x	Compact Flash memory 4GB 133x
Wireless LAN	USB wireless adaptor, 802.11 b/g,	USB wireless adaptor, 802.11 b/g,
Operating system	Linux (Debian)	Linux (Debian)
Interfaces	USB 2.0 camera and wireless adaptor	USB 2.0 camera and wireless adaptor
	I/O buttons	I/O buttons

A *package* is defined as a group of data that will be transmitted as a whole from one module to another. Every time a module wants to send some information to another module, it generates a package. If the receiving module is ready to receive, the package is sent immediately. If not, the package is stored for a posterior delivery. When a new package needs to be sent, and there are others packages waiting to be sent, three possible courses of action can be taken, depending on the specific connection: (i) overwriting the existing package and keeping just the newest one, (ii) queuing all the packages in order to make sure that all of them will arrive to the listening module, or (iii) re-packaging the packages' information so that only one contains the information of all of them. This can be done when the information contained in the packages can be somehow "summed up", as for example in the odometry case.

Every time a module receives a package, a *callback* function in charge of processing the received information for that specific connection is executed. Generally, after processing this information, the receiving module sends a message back to the sending module to notify it is ready to receive a new package. A module that has incoming and outgoing communication connections has to handle two types of messages: (messages that notify the module that a new package has arrived from an incoming connection, and messages that notify the module that another module is ready to receive a new package through an outgoing connection. Since the module only has one thread to process these messages, they can be queued.

4 New developments of potential interest for the League

4.1. Fall Detection and Management in Biped Humanoid Robots

The appropriate management of fall situations – i.e. fast instability detection, avoidance of unintentional falls, falling without damaging the body, fast recovering of the standing position after a fall - is an essential ability of biped humanoid robots. This issue is especially important for humanoid robots carrying out demanding movements such as walking in irregular surfaces, running or practicing a given sport (e.g. soccer). Thus, fall management in biped humanoid robots should be tackle from an integrated point of view, considering the following elements: (i) *Robot design*. Robot bodies should passively help as much as possible to avoid fall damage. Passive techniques include padding and the use of protections; (ii) *Instability Detection*. Falls need to be detected as soon as possible to have sufficient time to manage them; (iii) *Fall Avoidance***.** After instability detection, a fall can be avoided by feedback control or by reflexes; and (iv) *Fall management*. In case a fall cannot be avoided or in case of an intentional fall, a falling sequence can be triggered with the purpose of reducing the body- damage, or even for fast recovering of the standing position after a fall.

In [2] we addressed the design of low-damage fall sequences, which can be activated/triggered by the robot in case of a detected unintentional fall or an intentional fall. Very recently we tackled the detection of instability and the avoidance of falls in biped humanoids, as well as the integration of all components in a single framework [5]. In this framework a fall can be avoided or a falling sequence can be triggered depending on the detected instability's degree. We focus on managing instabilities produced by external disturbances such as external forces produced by collisions with other robots or objects, foot slippage or irregular surfaces, instead of on the gait stabilization problem that usually deals with instabilities produced by the gait itself. The framework has been validated in real world-experiments with our humanoid robots, and we are doing its fine-tuning to use it in the next RoboCup.

4.2 A Real-Time Hybrid Architecture for Biped Humanoids with Active Vision Mechanisms

Mobile autonomous robots are becoming complex systems that are able to interact with humans and other robots in complex, challenging, and often dynamically changing environments. Robots operating in such challenging conditions need to be controlled appropriately. Deliberative, reactive and pure behavior-based control paradigms have been known for many years, and in today's applications combinations of them are often used. Robot control architectures should incorporate reactivity, planning capabilities, and a modular and hierarchical design [6]. Hybrid, or tiered control architectures incorporate dynamic, concurrent, and time-responsive control (reactive control), together with efficient decision making over long time scales (deliberative control) [7]. In addition, their clear interface definition has the advantage that the different layers can be developed and modified in parallel [6].

Taking all these elements into consideration, we have developed an hybrid control architecture for biped humanoid robots [4]. Although the architecture is designed to be general-purpose, the short-term goal is to use it in robot soccer applications. The architecture is modular and hierarchical. It organizes the main robot-control functionalities in four parallel modules: perception, actuation, world-modeling, and hybrid control. The hybrid control module is decomposed in three behavior-based hierarchical layers: planning layer, deliberative layer and reactive layer, which work in parallel and have very different response speeds and planning capabilities. The architecture allows (i) the coordination of multiple robots and the execution of group behaviors without disturbing the robot reactivity and responsivity, which is very relevant for biped humanoid robots whose gait control requires real-time processing, (ii) the straightforward management of the robot's resources using resource multiplexors, and (iii) the integration of active vision mechanisms in the reactive layer, under control of behavior-dependant value functions from the deliberative layer. This last feature adds flexibility in the implementation of complex functionalities, as the ones required for playing soccer in robot teams. The architecture was first validated in the humanoid robots of the SPL league [4], and now it is in use in the robots of our humanoid team.

Acknowledgements

This research was partially supported by FONDECYT (Chile) under Project Number 1090250.

References

[1] UChile Official Website: http://www.robocup.cl/

[2] Ruiz-del-Solar, J., Palma-Amestoy, R., Marchant, R., Parra-Tsunekawa, I., and Zegers, P. (2009). Learning to Fall: Designing Low Damage Fall Sequences for Humanoid Soccer Robots. *Robotics and Autonomous Systems* (*Special Issue on Humanoid Soccer Robots*), 57, pp. 796-807, July 2009.

[3] Zagal, J.C., Delpiano, J., and Ruiz-del-Solar, J. (2009). Self-Modeling in Humanoid Soccer Robots. *Robotics and Autonomous Systems* (*Special Issue on Humanoid Soccer Robots*), 57, pp. 819-827, July 2009.

[4] Testart, J., Ruiz-del-Solar, J., Schulz, R., Guerrero, R., and Palma-Amestoy, R. (2010). A Real-Time Hybrid Architecture for Biped Humanoids with Active Vision Mechanisms. *Robotics and Autonomous Systems* (*Special Issue on Hybrid Control of Autonomous Systems*) (submitted).

[5] Ruiz-del-Solar, J., Moya, J., Parra-Tsunekawa, I. (2010). Fall Detection and Management in Biped Humanoid Robots, *IEEE Int. Conf. on Robotics and Automation 2010 – ICRA 2010*, May 3-8, 2010, Anchorage, Alaska, USA (accepted).

[6] D. Korterkamp, and R. Simmons, Robotic Systems Architectures and Programming, Chapter A.8, in *Springer Handbook of Robotics*, B. Siciliano & O. Khatib (Eds.), pp. 187-206, Springer 2008.

[7] M. Mataric, and F. Michaud, Behavior-based systems, Chapter E.38, in *Springer Handbook of Robotics*, B. Siciliano & O. Khatib (Eds.), pp. 891-909, Springer 2008.

[8] Boost library official website: http://www.boost.org/