

The Rh-1 full-size humanoid robot: Control system design and Walking pattern generation

Mario Arbulú, Dmitry Kaynov and Carlos Balaguer
*Robotics lab, Department of Systems and Automation Engineering
University Carlos III of Madrid
Spain*

1. Introduction

Since industrial robots cannot be easily adapted to assist human activities in everyday environments such as in hospitals, homes, offices, there is a growing need for robots that can interact with a person in a human-like manner. Wheel robots sometimes cannot be used in such kinds of environments because of the obvious restrictions posed by the use of wheels. For example, it is impossible for this kind of robot to go downstairs and upstairs or to clear some obstacles on the floor. What is more, humanoid robots are expected to play a more important role in the future.

One of the most exciting challenges that has faced the engineering community in recent decades was obtaining a machine of a similar form, a humanoid robot, that could do the same activities as a human being in addition to walking in the same manner (such as HONDA robots, Hirai et al. 1998; HRP robots, Kaneko et al. 2002, 2008; Johnnie, Loeffler et al.; LOLA, Lohmeier et al. 2006.)

There are several reasons to construct a robot with these characteristics. Humanoid robots will work in a human atmosphere with greater effectiveness than any other type of robots because the great majority of environments where they would interact reciprocally with a human are constructed, taking into account the dimensions of the latter. If it is supposed that a machine should complete dangerous tasks or work in extreme conditions, in the ideal case its anthropometric measures must be as close as possible to the ones of its prototype. Inclusively, there are professionals who adduce that for a human being to interact naturally with a machine, it must look like him.

The main goal of this project is the development of a reduced weight human size robot which can be a reliable humanoid platform for implementing different control algorithms, human interaction, etc.

The main assumption for the mechanical design started with the weight of a 1.20 m person and the desired walking motion of the humanoid robot. With these, the requirements for each joint's torque were calculated and then, by dynamical analysis, the structure was designed and the dimensions of the motors were determined. It was an iterative process for

obtaining the optimal torques which allows the anthropometrical walking of a 1.45m human.

Nowadays, the development of humanoid robots has become a very active area. However, it is still limited by the very high cost of maintenance and development.

The main parts of the hardware of the humanoid robot are the "custom-built" components. The software also does not have any standardization or common rules for the humanoid robot's programming. It implies the growth of usage of some technologies from the industrial automation field in humanoid robotics because of their low cost and reliability.

The control system of the Rh-1 robot was designed using the conventional electronic components of the automation industry in order to reduce the development time and cost and to have a flexible and easily upgradeable hardware system.

While generating walking patterns we can compute joint angular speed, acceleration and torque ranges (Arbulu et al. 2005 and Stramiglioli et al. 2002). There are two methods for designing gait patterns: The distributed mass model and the concentrated mass model. In our case, the concentrated mass model is used because the humanoid dynamics is simplified significantly (Kajita et al. 2003, Gienger et al. 2001). In order to obtain a natural and stable gait, the 3D-LIPM method is used, where the pendulum mass motion is constrained to move along an arbitrarily defined plane. This analysis leads us to a simple linear dynamics: the Three dimensional Linear Inverted Pendulum Mode (3D-LIPM, Kajita et al. 2003). Furthermore, the sagittal and frontal motion can be studied in separate planes (Raibert 1986, Arbulu et al. 2006). The 3D-LIPM takes into account that the pendulum ball moves like a free ball in a plane following the inverted pendulum laws in the gravity field, so the ball motion has only three parameters: gravity, the plane position and the ball position. This model is applicable only during the single support phase. Another mass concentrated model is used, which is the Cart-table model (Kajita et al. 2003), implemented in order to improve the walking patterns because the ZMP position can be predicted and a closer relationship with the COG is obtained. Smooth patterns by optimizing jerk are obtained; this will be seen in successful experiments. In order to apply the obtained trajectory to the humanoid robot Rh-1, the ball or cart motion drives the middle of the hip link. The foot trajectories are computed by single splines, taking into account foot position and orientation and the landing speed of each foot to keep the humanoid from falling down. Several direction patterns will be computed. Some simulation experiments have been done using a 21 DOF VRML robot model.

Then, using the Lie-logic method, the inverse kinematics problem for the entire robot body was solved and the trajectory vectors for each joint were obtained (Paden et al. 1986). These trajectories are used as reference calculated motion patterns and denote feet, arms and the entire body trajectory of the robot.

An industrial robot usually operates in a well defined environment executing pre-programmed tasks or movement patterns. In the same way, the first approaches to make a humanoid robot walk (Shin et al. 1990) were based on the generation of stable offline patterns according to the ZMP (Zero Moment Point) concept (Vukobratovic et al. 1969). In contrast to industrial robots, a humanoid robot will interact with a person in a continuously changing workspace. Therefore, the use of only static motion patterns for the humanoid robot interaction is insufficient.

The other method is a real-time control, based on sensor information (Furusho et al. 1990 and Fujimoto et al. 1998). This approach requires a large amount of computing and

communication resources and sometimes it is not suitable for a humanoid robot with a high number of joints.

A humanoid robot can walk smoothly if it has previously defined walking patterns and an ability to react adequately to the disturbances caused by imperfections in its mechanical structure and irregular terrain properties. Previous works (Park et al. 2000 and Quang et al. 2000), rarely considered and published the detailed architecture for the online modification of dynamic motion patterns.

And finally, a new humanoid platform Rh-1 (Fig. 1) was constructed and successfully tested in a series of walking experiments.

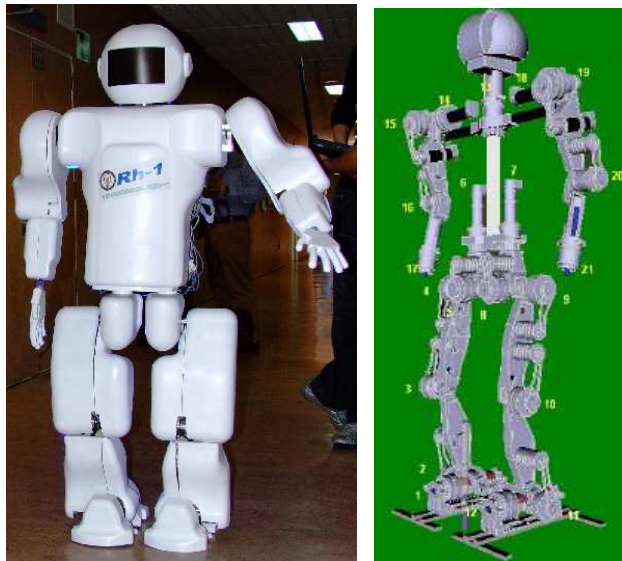


Fig. 1. Rh-1 humanoid robot

This paper presents the control architecture which combines use of the previously offline calculated motion patterns with online modifications for dynamic humanoid robot walking.

The main contributions of this work are:

- Validation of the novel approach for walking pattern generation: Local Axis Gait Algorithm (LAG). It permits walking in any direction and on an uneven surface (i.e. ramp, stairs).
- Implementation and validation of the use of screws provide a very geometric description of rigid motion, so the analysis of the mechanism is greatly simplified; furthermore, it is possible to carry out the same mathematical treatment for the different robot joints: revolute and prismatic.
- Development of hardware and software control architecture for the humanoid robot Rh-1. This allows us to obtain a more flexible and adaptable system capable of changing its properties according to user needs. Proposed hardware architecture is a novel solution for the area of humanoid robots that complies with modern tendencies in robotics. Software

architecture providing the robot with a standard functionality is easily upgraded and can use new one.

- Definition of purpose and validation of kinematics modelling of humanoid robots using screw theory and Paden-Kahan sub-problems, which have the following advantages:

They avoid singularities because they offer a global description of rigid body motion; we only need to define two frames (base and tool) and the rotation axis of each DOF, to analyze the kinematics in a closed way.

The Paden-Kahan sub problems allows for computing the inverse kinematics at position level.

There is a faster computation time of the inverse kinematics with respect to the inverse Jacobean method, Euler angles or D-H parameters, so it contributes to real-time applications.

- Implementation and validation of the inverted pendulum and cart-table-based walking patterns for any humanoid robot, under the LAG algorithm.

- Development of new efficient algorithms for joint motion control and stabilization of the humanoid robot gait. These algorithms provide simple solutions allowing for fast and reliable integral control of a robot.

The paper is organized as follows. Section 2 deals with the human biomechanics study. Sections 3, 4 and 5 consider hardware, software and communication infrastructures of the humanoid robot Rh-1. Then, the section 6 presents walking pattern generation and some simulations results. Section 7 considers the control architecture implemented in order to control the robot's motion and stability. Experimental results will be shown in section 8. And finally, Section 9 presents conclusions of this work.

2. Biomechanics

2.1 Outline

The humanoid design starts from its motion requirements, so dimensions, joint range motion, joint velocities, forces and wrench should be studied. After that, the link design can start. This first humanoid robot prototype deals with the study of locomotion, so human locomotion will be analyzed. First, human biomechanics anthropometry is studied; next, human walking motion is analyzed.

2.2 Kinematics

The term used for these descriptions of human movement is kinematics. Kinematics is not concerned with the forces, either internal or external, which cause the movement, but rather with the details of the movement itself. In order to keep track of all the kinematic variables, it is important to establish a convention system. Thus if we wish to analyze movement relative to the ground or the direction of gravity, we must establish a spatial reference system (Fig. 2).

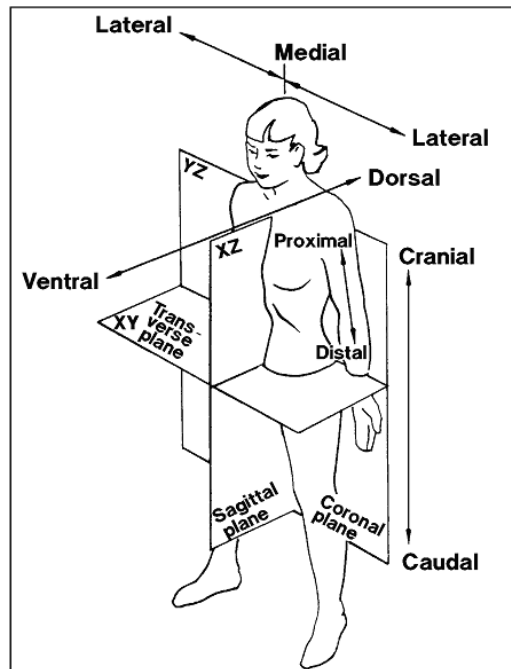


Fig. 2. Human motion planes, © NASA.

2.3 Human locomotion

For dividing the gait cycle in many stages or events, some considerations are taken into account such as the fact that the gait cycle is the period of time between any two identical events in the walking cycle (Ayyappa, E. 1997). As the gait cycle could be divided into events and the continuity between each of them must be maintained, any event could be selected as the starting of the gait cycle (that is in the ideal case because the terrain imperfections and human postures make gait cycle not periodic, see Fig. 3). So, the starting and finishing event are called the initial contact respectively. Otherwise, the gait stride is defined as the distance between two initial contacts of one foot.

The stance and swing are the events of the gait cycle. Stance is the event when the foot is in contact with the ground, (around 60 percent of the gait cycle). Swing is the event when the foot is in the air, (around 40 percent of the gait cycle).

2.4 Anthropomorphic human dimensions, volume and weight distribution

Human dimensions are taken into account as a reference because their proportions allow for stable walking and optimal distribution of the forces actuating while a human is walking. Biomechanics give us the relationship between human height and each link (Fig. 4, Winter, D. 1990), as well as in the same way as the mass.

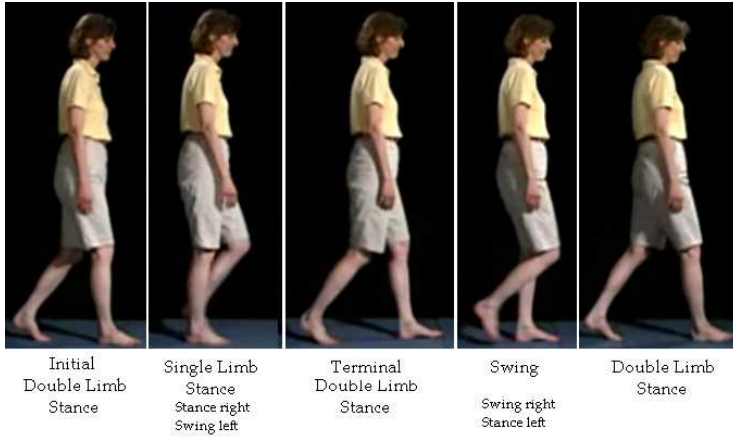


Fig. 3. The gait cycle has two phases: about 60-percent stance phase and about 40-percent swing phase with two periods of double support which occupy a total of 25 to 30 percent of the gait cycle.

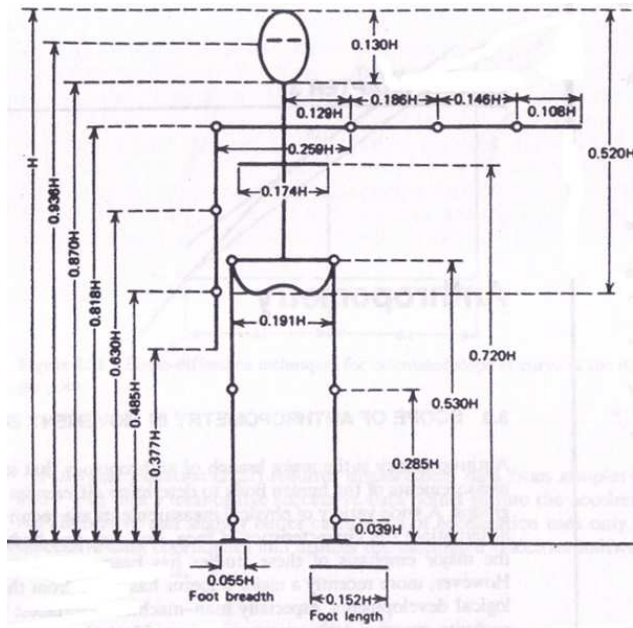


Fig. 4. Anthropomorphic human dimensions, © Winter, D.

2.5 Human walking trajectories

Human walking motion is studied in order to analyze the right motion of each link and joint during the step. The swing leg and hip motions must assure stable walking in any direction and speed.

The joint angular evolution during a walk should be measured with the appropriate devices, or by introducing the swing leg and hip trajectories as inputs of a kinematic model. For the humanoid robot, the joint angular evolution is the input for walking. The human swing foot normally falls to the ground when walking, while for a humanoid robot this must be avoided in order to protect the robot structure and force sensors of the soles. Thus, the adequate walking pattern should be generated for the COG and the swing foot. Normally, the human COG follows the laws of the inverted pendulum in the field of gravity during the walking motion, which is a hyperbolic orbit. It is suitable for making a smooth walking motion at the jerk level. However, the humanoid robot's swing foot motion should be faster than the human one in order not to fall while walking.

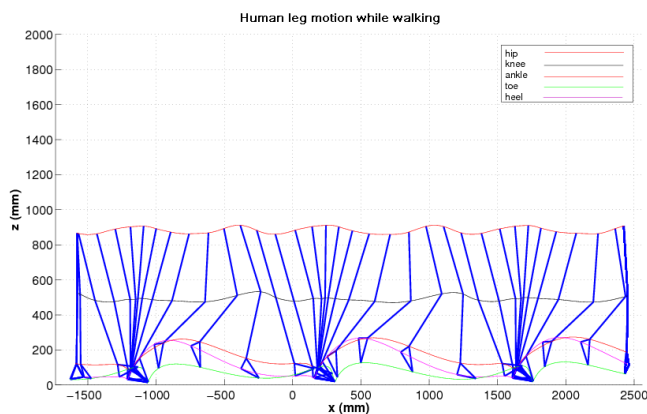


Fig. 5. Human leg motion, Sagittal view.

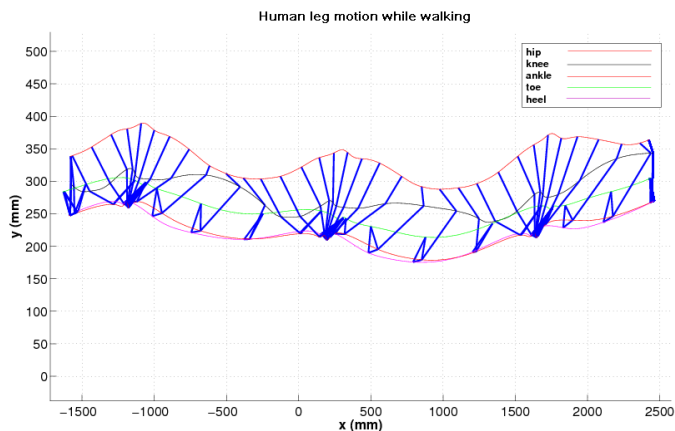


Fig. 6. Human leg motion, Top view.

Figures 5 and 6 show the leg motion and the hip, knee and feet trajectories (including the ankle, toe and heel). The hip trajectory is quite similar to the COG trajectory. In the sagittal view, that trajectory climbs and descends cyclically. The falling motion increases the sole

reaction force, so in the humanoid robot, it is better to have a motion on a horizontal plane; furthermore, the trajectory shape looks like the inverted pendulum motion (top view, Fig. 6), so we could approximate the humanoid robot in this way.

3. Hardware Architecture

The hardware architecture for the humanoid robot has some important restrictions imposed by the limited availability of space. In general, the basic requirements for hardware architecture of a humanoid robot are: scalability, modularity and standardized interfaces (Regenstein et. al. 2003). In the case of the Rh-1 robot with 21 DOF, which supposes the use of 21 DC motors in synchronized high-performance multi-axis application, it is first necessary to choose an appropriate control approach. The trend of modern control automation is toward distributed control. It is driven by one basic concept: by reducing wiring, costs can be lowered and reliability increased. Therefore, an electrical design of Rh-1 robot is based on a distributed motion control philosophy where each control node is an independent agent in the network. Figure 7 shows the physical distribution of the hardware inside the humanoid robot.

The architecture presented is provided with a large level of scalability and modularity by dividing the control task into Control, Device and Sensory levels (Fig. 8), (Kaynov et al. 2005).

The Control level is divided into 3 layers represented as a controller centered on its own tasks such as external communications, motion controller's network supervision or general control.

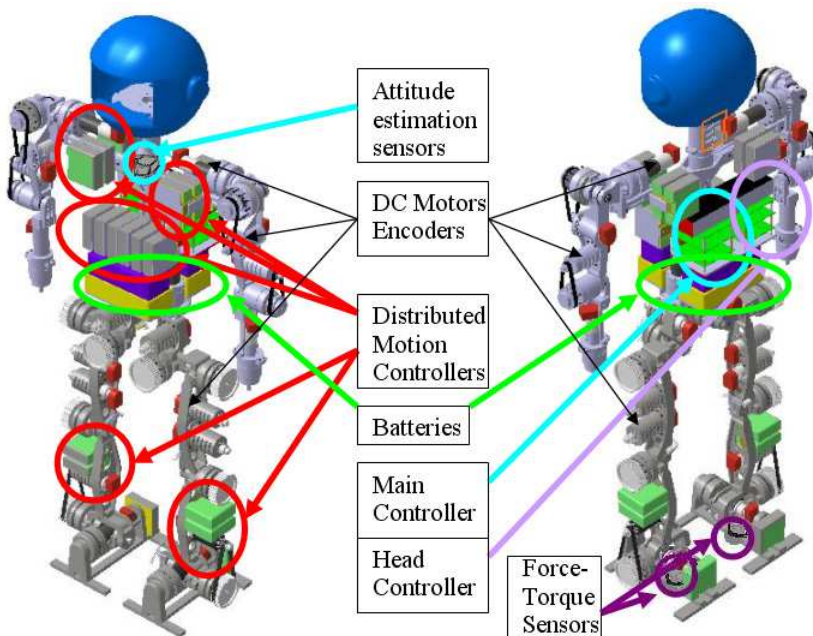


Fig. 7. Hardware distribution inside the humanoid robot.

In the Device level each servo drive not only closes the servo loop but calculates and performs trajectory online, synchronizes with other devices and can execute different movement programs located in its memory. These kinds of devices are located near the motors, thus benefiting from less wiring, which is one of the requirements for energy efficiency; they are lightweight and require less effort in cabling. Advanced and commercially available motion controllers were implemented in order to reduce development time and cost. Continuous evolution and improvements in electronics and computing have already made it possible to reduce the industrial controller's size to use it in the humanoid development project. Furthermore, it has the advantage of applying well supported and widely used devices from the industrial control field, and brings the commonly used and well supported standards into the humanoid robots development area. On the Control level, the Main controller is a commercial PC/104+ single board computer because of its small size and low energy consumption. It was used instead of a DSP controller because it has a different peripheral interface as is the Ethernet and RS-232, and an easy programming environment. In addition, there is a great variety of additional extension modules as the PC/104+ bus like CAN-bus, digital and analog input-output, and PCMCIA cards. Selection criteria were fast CPU speed, low consumption and availability of expansion interfaces. The Main Controller provides general synchronization, updates sensory data, calculates the trajectory and sends it to the servo controllers of each joint. It also supervises data transmission for extension boards such as Supervisory Controller and ZMP Estimation Controller via PC/104+ bus.

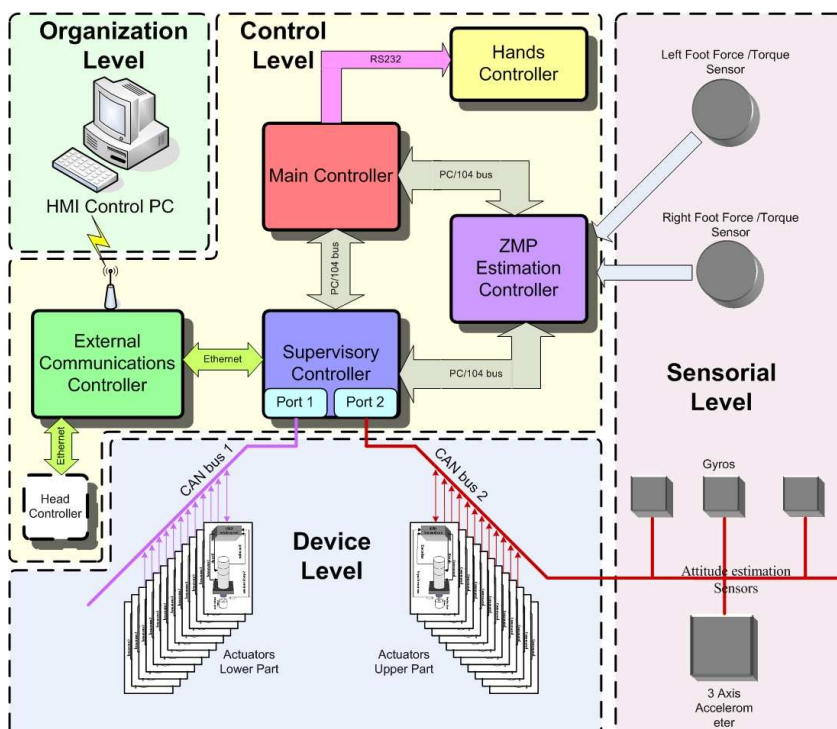


Fig. 8. Hardware architecture.

The Communication Supervisory Controller uses a network bus to reliably connect distributed intelligent motion controllers with the Main Controller.

The motion control domain is rather broad. As a consequence, communications standards to integrate motion control systems have proliferated. The available communication standards cover a wide range of capability and cost ranging from high-speed networked IO subsystems standards to distributed communications standards for integrating all machines on the shop floor into the wider company. The most appropriate solution to be implemented in the humanoid robot motion control system design seems to be the use of CAN-based standards. The CAN bus communication is used for the Sensory level and the CANOpen protocol on top of the CAN bus is used for the Device level of communications.

Thus, the control system adopted in the Rh-1 robot is a distributed architecture based on CAN bus. The CAN bus has also been chosen because of various characteristics, such as bandwidth up to 1 MBit/s that is of sufficient speed to control the axes of a humanoid robot, a large number of nodes (Rh-1 has 21 controllable DOF), differential data transmission, which is important for reducing the Electromagnetic Interference (EMI) effects caused by electric motors, and finally, the possibility for other devices such as sensors to reside in the same control network.

At the Device level, the Controller's network of the Rh-1 is divided into 2 independent CAN buses in order to reduce the load of the communication infrastructure. The Lower part bus controls 12 nodes of two legs and the Upper part bus controls 10 nodes of two arms and the trunk. To unify the data exchange inside the robot, the attitude estimation sensory system is also connected to the Upper part CAN bus. In this way, the communication speed of CAN bus used in Rh-1 is 1MBit/s. The synchronization of both parts is realized by the Supervisory Controller at the Control level of automation.

The External Communications module provides the Ethernet communication on the upper (Control) level of the automation with head electronics which comprise an independent vision and sound processing system. It also provides wireless communications with the Remote Client which sends operating commands for the humanoid robot. The proposed architecture complies with the industrial automation standards for the design of the motion control system.

4. Software Architecture

As mentioned above, a humanoid robot can be considered as a plant where the shop floor consists of a series of cells (intelligent motion controllers and sensors) managed by controllers (the Main Controller, Communication Supervisory Controller, etc.) In general, there are two basic control tasks for the control system of a humanoid robot. The first goal is to control all automation and supervise data transmission. Meanwhile, the second goal is to control and monitor the entire floor in order to detect failures as early as possible, and to report on performance indicators. In this context, the humanoid robot Rh-1 is provided with a software system allowing the implementation of the industry automation concepts (Kaynov et. al. 2007). The software architecture is based on the Server-Client model (Fig. 9).

For security reasons, the Control Server accepts the connections of other clients, such as the Head Client, responsible for the human-robot interaction, only if the Master Client allows it. If the connection is accepted, the Master Client only supervises the humanoid robot state and data transmission between the robot and other Client, but in the case of any conflict it always has top priority.

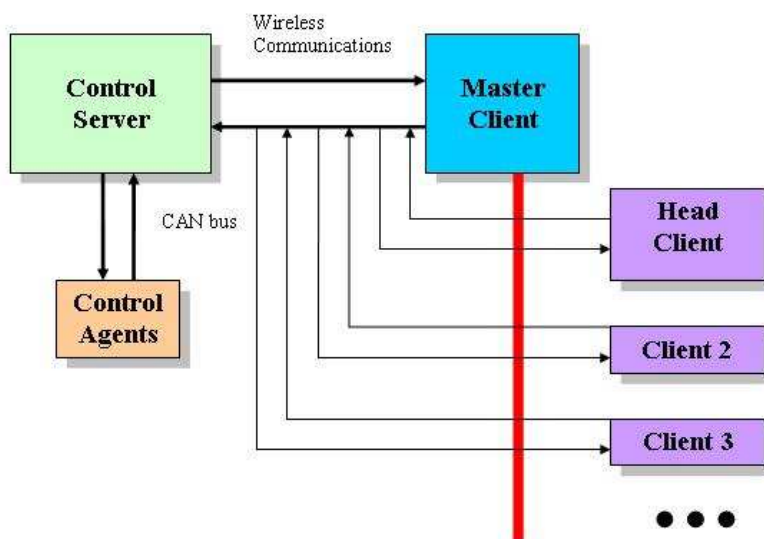


Fig. 9. System Architecture.

According to the Server-Client model, the humanoid robot is controlled by the passive Server, which waits for requests, and upon their receipt, processes them and then serves replies for the Client. On the other hand, the Server controls all Control Agents which reside in the CAN bus network. In that case, the Control Server is no longer a slave; it is a network master for Control Agents which performs their operations (motion control or sensing) and replies for the Server.

As a PLC in the automation industry, the Control Server is designed and programmed as finite state automata. Figure 10 shows the state diagram and Table II shows the state transition events of the humanoid robot Control Server functioning

Two basic types of incoming data are processed. A command is simple data, which can be executed by one Control Agent. The order is a complex command which needs the simultaneous action of many Control Agents and sensors which the humanoid robot possesses. After the connection of the Master Client, the humanoid robot stays in the Client Handling state waiting for an order or a simple command. The arrival of an order launches the User Program. The User Program is executed in the control area, the core of the humanoid robot Control Server software. It performs the data transmission between all Control Agents, sensory system and the Server. It performs trajectory execution at the synchronized multi-axis walking applications, controls the posture and ZMP errors at the dynamic walking mode, and reads the sensors' state, etc. The control area consists of different modules which provide the execution of motion control for stable biped locomotion of the humanoid robot. All tasks can be grouped by their time requirements.

The developed software provides the set of the C-based function to work with the robot and to generate the user's motions and control procedures that are not only for walking, but also for implementing different human-robot cooperation tasks. The code below shows the simple user program. The example in Figure 12 shows how the simple humanoid robot motion can be programmed. At the beginning, the synchronization procedure for every joint

is performed, and then the motion is started. The robot will change the gait (walking mode) according to user request.

Event	Event Description
E1	The Client is connected
E2	An Order has arrived
E3	A Command has arrived
E4	A Command is sent to the Control Agent
E5	Agent's reply has arrived
E6	An Answer is sent to the Client
E7	The User Program has successfully terminated or an Error Event has occurred
E8	Connection with the Client is lost
E9	The Robot is staying in the secure position
E10	All processes are terminated

Table 1. State Transition Events

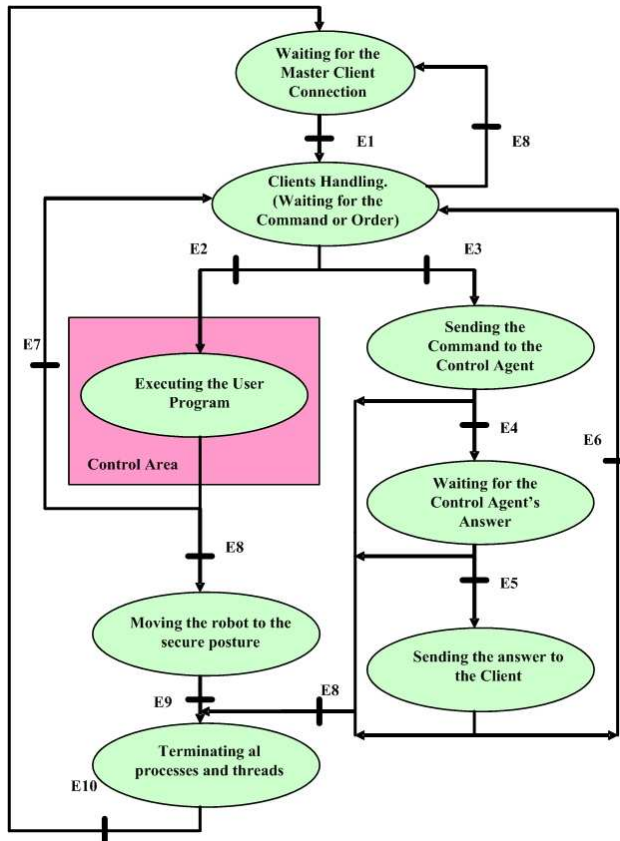


Fig. 10. Server functioning state diagram.

In the proposed software architecture, the Control Server is capable of accepting a large amount of clients' connections at the same time. It is evident that the Master Client, as the basic HMI of the humanoid robot, should provide and supervise the execution of the upper-level control tasks related with global motion planning, collision avoidance, and human-robot interaction. In general, these tasks are common for all mobile and walking robots and the design of these kinds of software systems is not considered in this paper. On the other hand, there are some bottom-level tasks that should be supervised such as sensory data acquisition, joint synchronization and walking stability control. In order to not overload the Master Client, which is more oriented to automation supervisory, these control tasks are processed with another client application. To provide the robot Rh-1 with bottom-level control, a SCADA system for the humanoid robot, called HRoSCoS (Humanoid Robot Supervisory Control System) Client was developed.

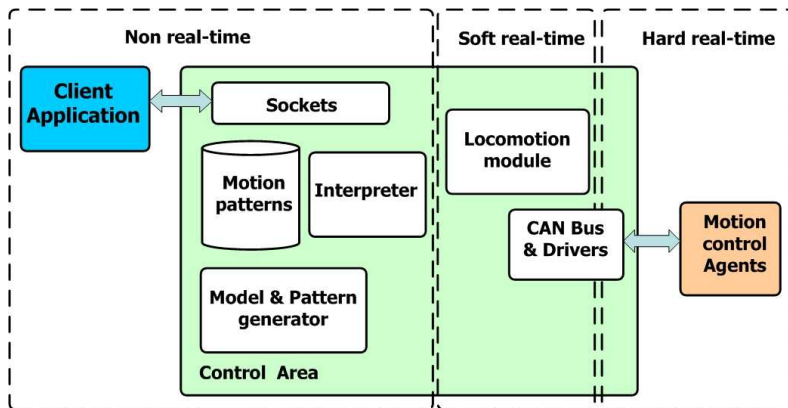


Fig. 11. Control area modules.

```

// Motion program example
#include "control.h"
#include "sensors.h"
#include "dynamic_gate.h"

main (void)
{
    EnableRobot();
    ReadJointsPosition();
    Sincronize();
    if ( Start() )
        do
        {
            PostureControl();
            DynamicGate(Gate1);
            if (ChangeGate)

            DynamicGate(Gate2);
        }
        while (!Stop());
    DisableRobot();
}

```

Fig. 12. Motion program example.

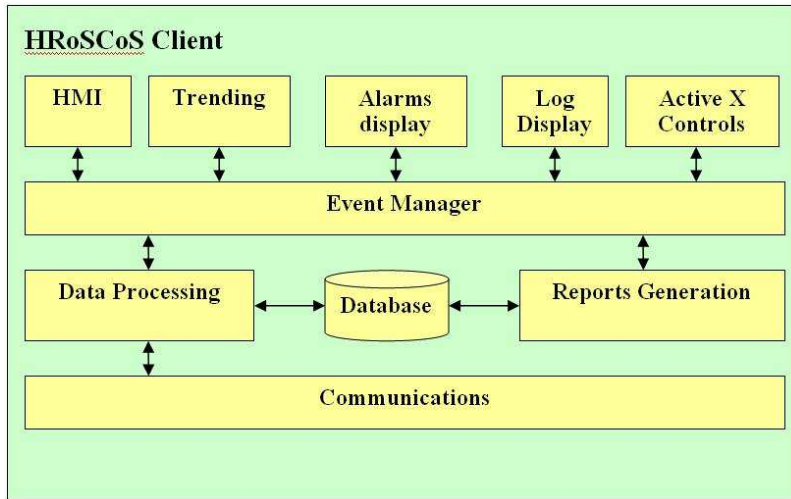


Fig. 13. HRoSCoS Client Architecture.

The developed software system is multi-tasking and the Control Server is also responsible for data acquisition and handling (e.g. polling motion controllers, alarm checking, calculations, logging and archiving) on a set of parameters when the HRoSCoS Client is connected. Figure 13 shows the HRoSCoS Client architecture.

The Client requires the data or changes control set points by sending commands. The arrival of a command launches its execution procedure (the right branch of the Server functioning State Diagram in Figure 10). It consists of the interpretation and transmission of the Command to a Control Agent. When the answer is received, it is converted and transmitted to the HRoSCoS Client to be processed and visualized.

The HRoSCoS Client provides the trending of different parameters of the robot, such as the joint velocities, accelerations, currents, body inclinations, forces and torques which appear during humanoid robot walking. Real-time and historical trending is possible, although generally not in the same chart. Alarm handling is based on limit and status checking and is performed in the Control Server (for example current limit or physical limit of the joint) and then the alarm reports are generated into the HRoSCoS Client application. More complicated expressions (using arithmetic or logical expressions) are developed by creating derived parameters on which status or limit checking is then performed. Logging of data is performed only if some value changes. Logged data can be transferred to an archive once the log is full. The logged data is time-stamped and can be filtered when viewed by a user. In addition, it is possible to generate different reports on the humanoid robot state at any time.

The HRoSCoS Client system presents the information to the operating personnel graphically. This means that the operator can observe a representation of the humanoid robot being controlled (Fig. 11).

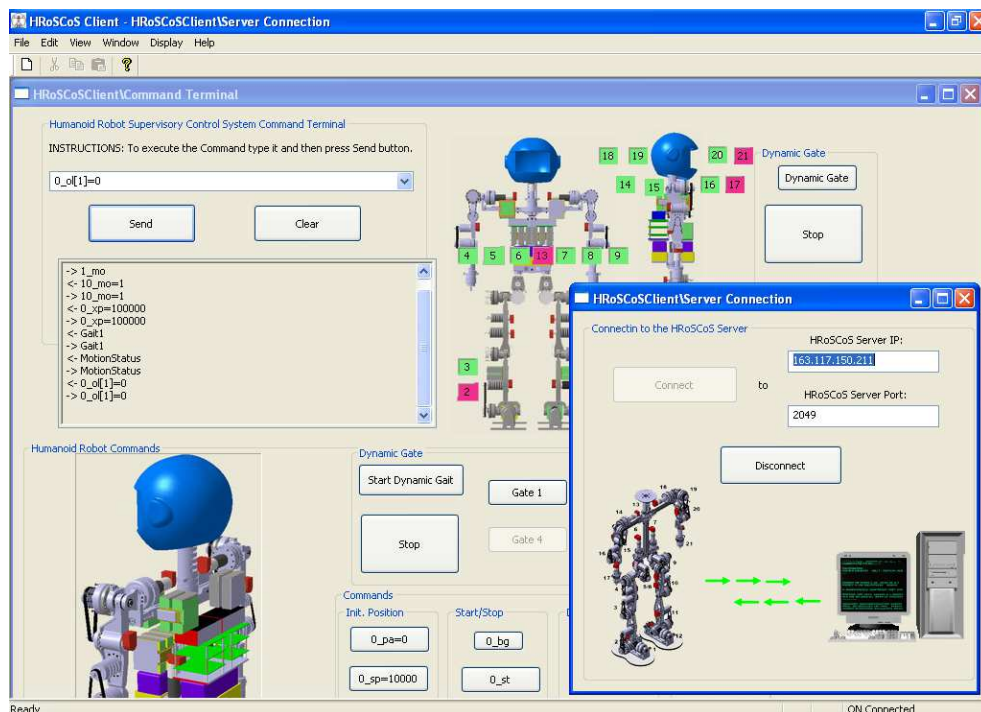


Fig. 14. HRoSCoS Client views.

The HMI supports multiple screens, which can contain combinations of synoptic diagrams and text. The whole humanoid robot is decomposed in "atomic" parameters (e.g. a battery current, its maximum value, its on/off status, etc.) to which a Tag-name is associated. The Tag-names are used to link graphical objects to devices. Standard windows editing facilities are provided: zooming, re-sizing, scrolling, etc. On-line configuration and customization of the HMI is possible for users with the appropriate privileges. Links are created between display pages to navigate from one view to another

5. Communication Infrastructure and Methods

When building automation applications, communication with the host is often a crucial part of the project. Nodes of the network always function as data servers because their primary role is to report information (status, acquired data, analyzed data, etc.) to the host at constant rates.

As shown in Figure 9, Hardware Architecture consists of three basic levels of automation which uses its own communication systems. The upper (Control) level uses a TCP/IP based communication protocol. Ethernet communication is one of the most common methods for sending data between computers. The TCP/IP protocol provides the technology for data sharing, but only the specific application implements the logic that optimizes performance and makes sense of the data exchange process. When data transmission begins, the sender should packetize each piece of data with an ID code that the receiver can use to look up the

decoding information. In this way, developed communication protocol hides the TCP implementation details and minimizes network traffic by sending data packages only when they are needed. When a data variable is transmitted by the sender, it is packetized with additional information so it can be received and decoded correctly on the receiving side. Before each data variable is transmitted, a packet is created that includes fields for Data Size, Data ID and the data itself. Figure 15 shows the packet format.

Data Size (8 bits)	Data ID (16 bits)	Data (32 bytes)
-----------------------	----------------------	--------------------

Fig. 15. The package format.

The Data ID field is populated with the index of the data array element corresponding to the specified variable. Since the receiving side also has a copy of the data array, it can index it to get the properties (name and type) of the incoming data package. This very effective mechanism is implemented to provide data exchange between the Control Server and different Clients on the Control level of automation of the humanoid robot.

Bottom-level (Sensory and Field) communications are realized using CAN and CanOpen protocols (Fig. 16).

These communication protocols provide data transmission in broadcast type of communication. A sender of information transmits to all devices on the bus. All receiving devices read the message and then decide if it is relevant to them. This guarantees data integrity as all devices in the system use the same information. The sensory system of the humanoid robot makes data exchange under lower CAN protocol and the intelligent motion controllers uses upper-level CANOpen protocol. The same physical layer of these protocols allows them to reside in the same physical network.

The communication implemented on the bottom level involves the integration of CANOpen (Drives and Motion Control Device Profile) and the introduction of new functionality which is not contained within the relevant device profiles for the sensory data processing.

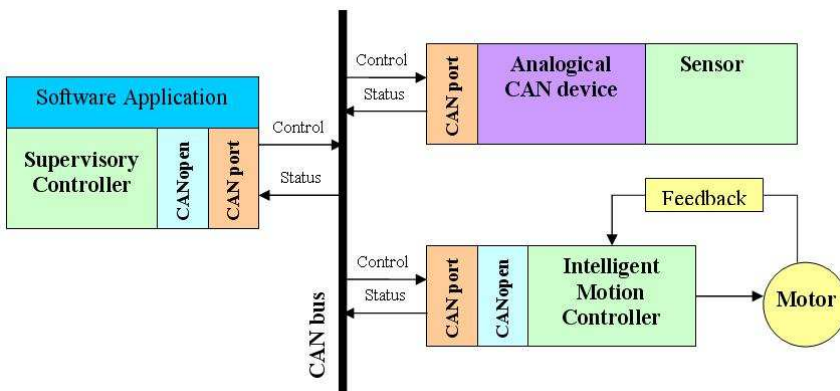


Fig. 16. CAN bus-based communication system.

6. Walking Pattern Generation

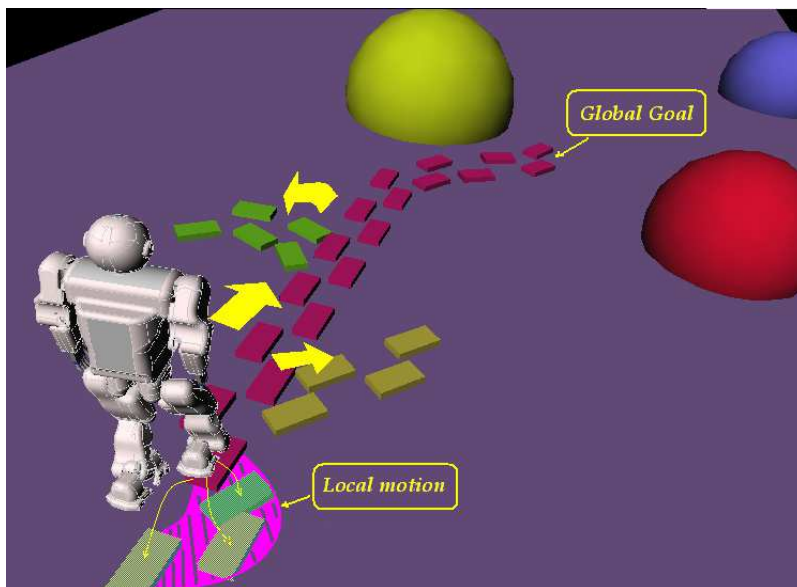


Fig. 17. Concept of gait generation method. For reaching the “Global goal”, a set of “Local motions” must be generated. Thus the local motion decide the better foot location for going ahead, go back, turn left, turn right, doing lateral step, climbing a stair or ramp.

There are many propositions for generating the walking patterns of humanoid robots, some of them a mass distributed based model, (Hirukawa et. al. 2007) and other ones a mass concentrated based model, (Kajita et. al. 2004 and Gienger et. al. 2001). The first approach describes the motion accurately, but it has a high computation cost, which is not suitable for real-time applications. On the other hand, the second approach saves computation time and performs the walking motion suitably. In this section, two kinds of mass concentrated models will be explained and discussed, that is: the inverted pendulum model and the cart-table model. Both models have been tested on the Rh-1 humanoid robot platform in order to generate stable walking patterns. At first, the 2D inverted pendulum model will be detailed, for introducing pendulum laws; next the 3D version is developed; after that, the Cart-table model will be introduced and its advantages with respect to the inverted pendulum are explained; next, the walking pattern strategy is proposed with the “Local Axis Gait” (Fig. 18) algorithm, (Arbulu et. al. 2007, 2008). Finally, in order to compute joint patterns the inverse kinematics model is proposed, by using the screw theory and Lie groups.

6.1 2D Inverted pendulum model

The gait pattern generation for a humanoid robot could be simplified as: studying the motion in the sagittal plane and concentrating all the body mass in the COG. In this way, it is possible to use the 2D Inverted pendulum model to obtain stable and smooth walking motion.

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

