

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# 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.

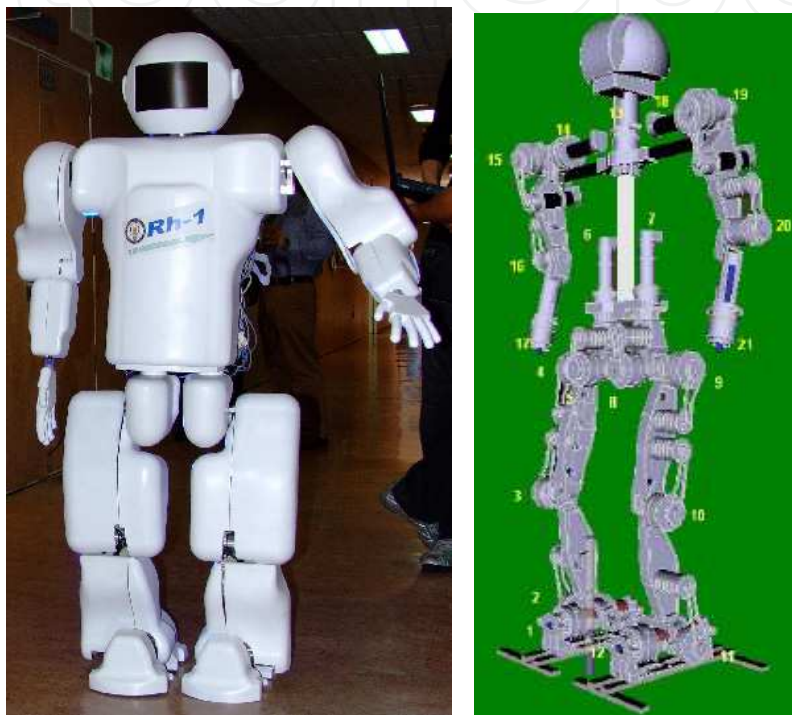


Fig. 1. Rh-1 humanoid robot

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).

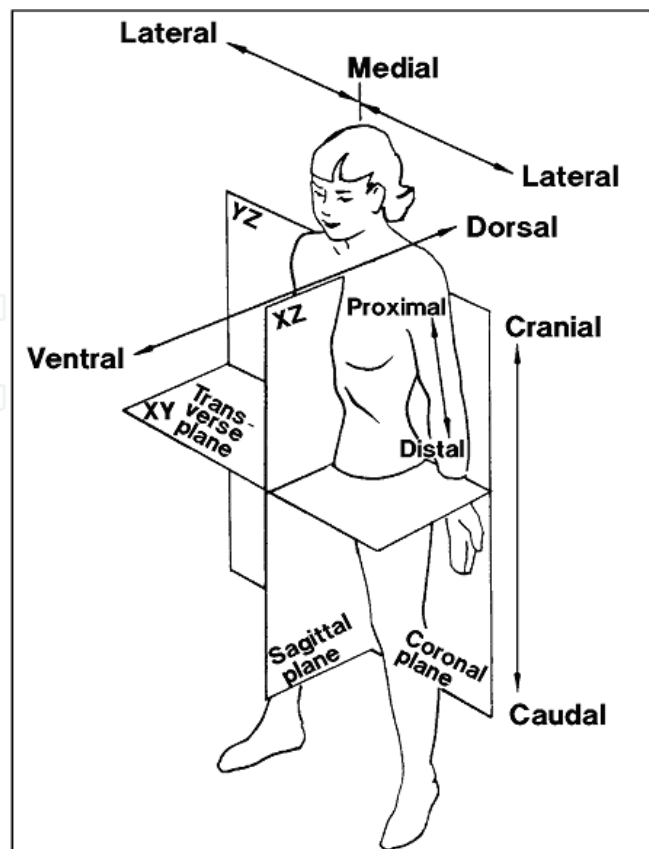


Fig. 2. Human motion planes, © NASA.

### 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.



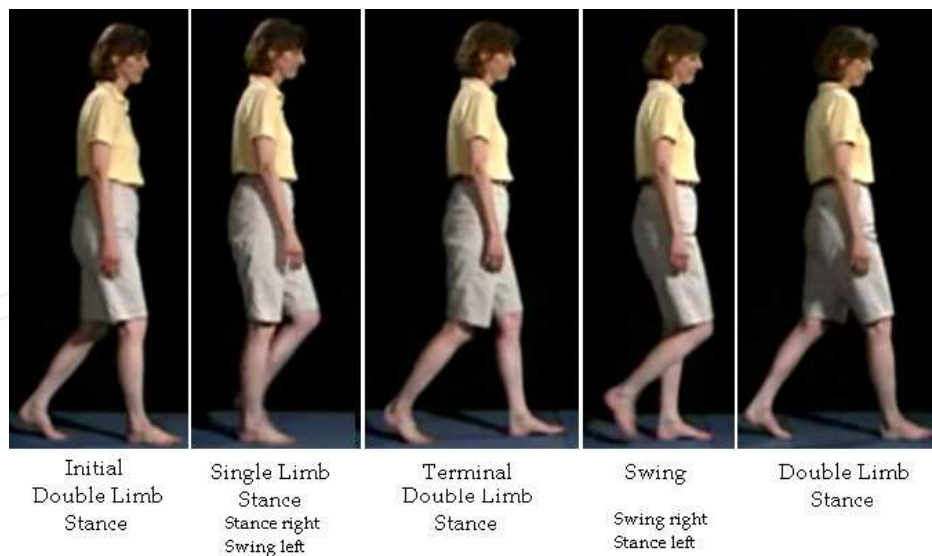


Fig. 3. The gait cycle has two phases: about 60-percent stance phase and about 40-percent swing phase with two periods of double support which occupy a total of 25 to 30 percent of the gait cycle.

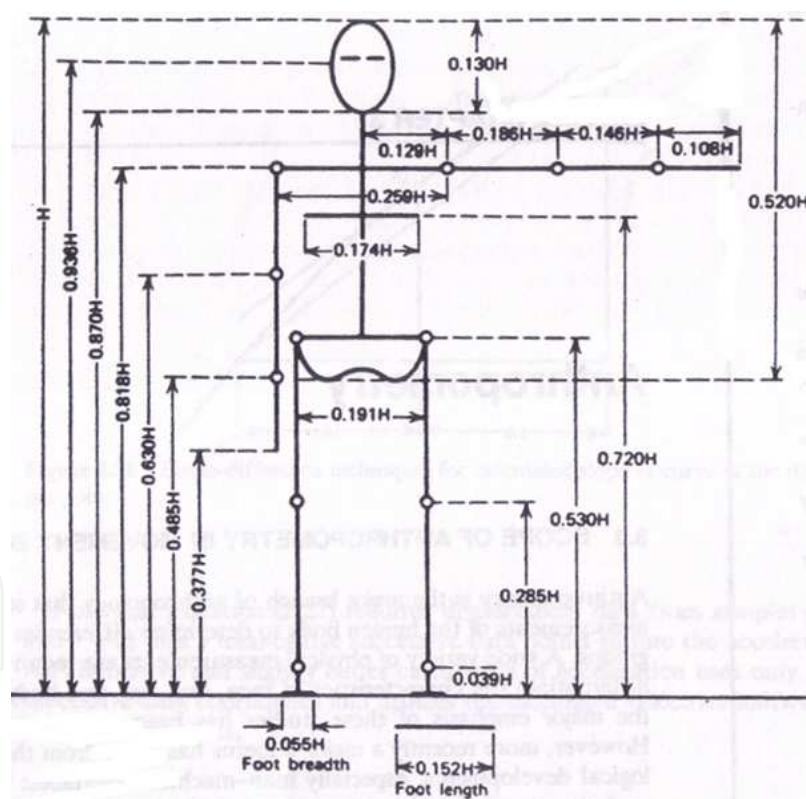


Fig. 4. Anthropomorphic human dimensions, © Winter, D.

## 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.

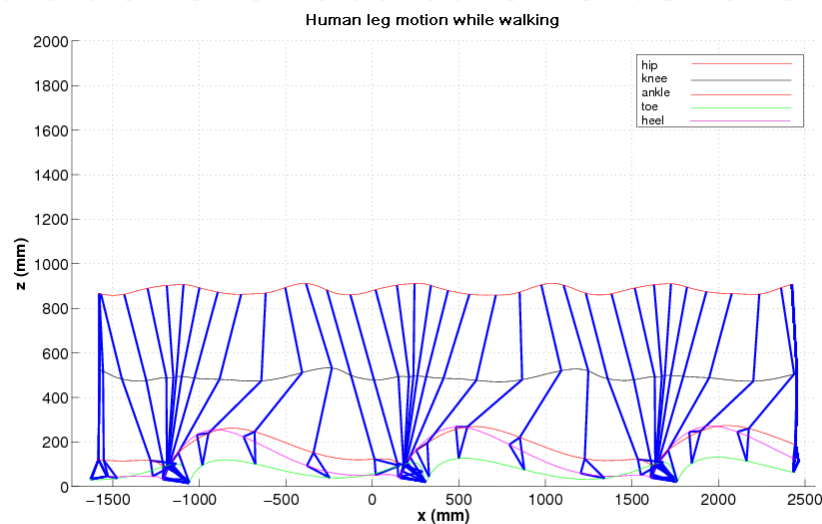


Fig. 5. Human leg motion, Sagittal view.

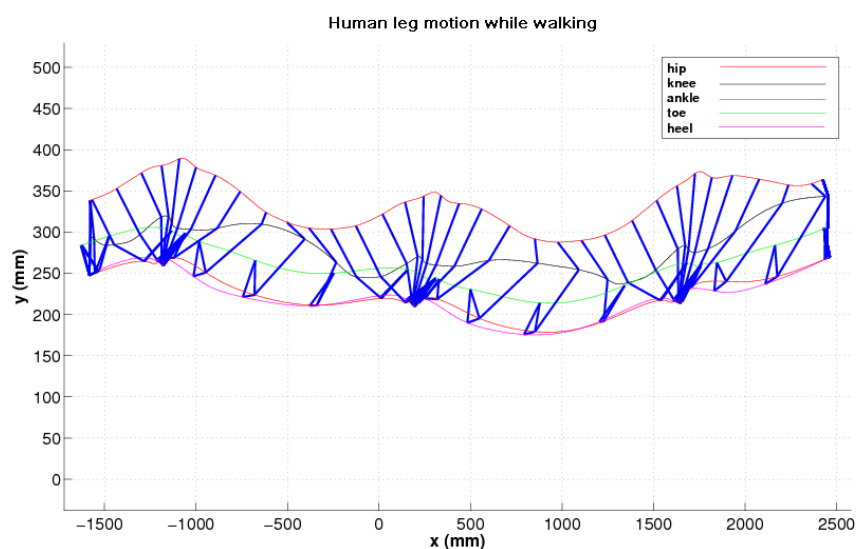


Fig. 6. Human leg motion, Top view.

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.

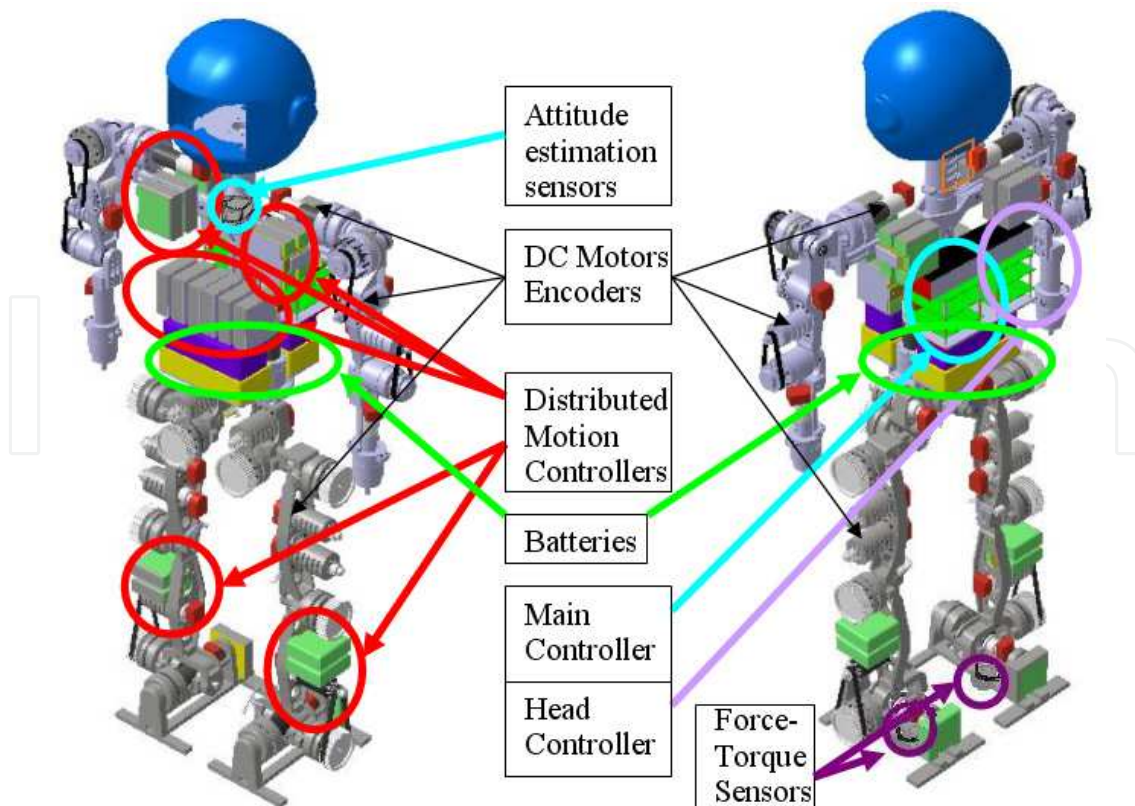


Fig. 7. Hardware distribution inside the humanoid robot.

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.

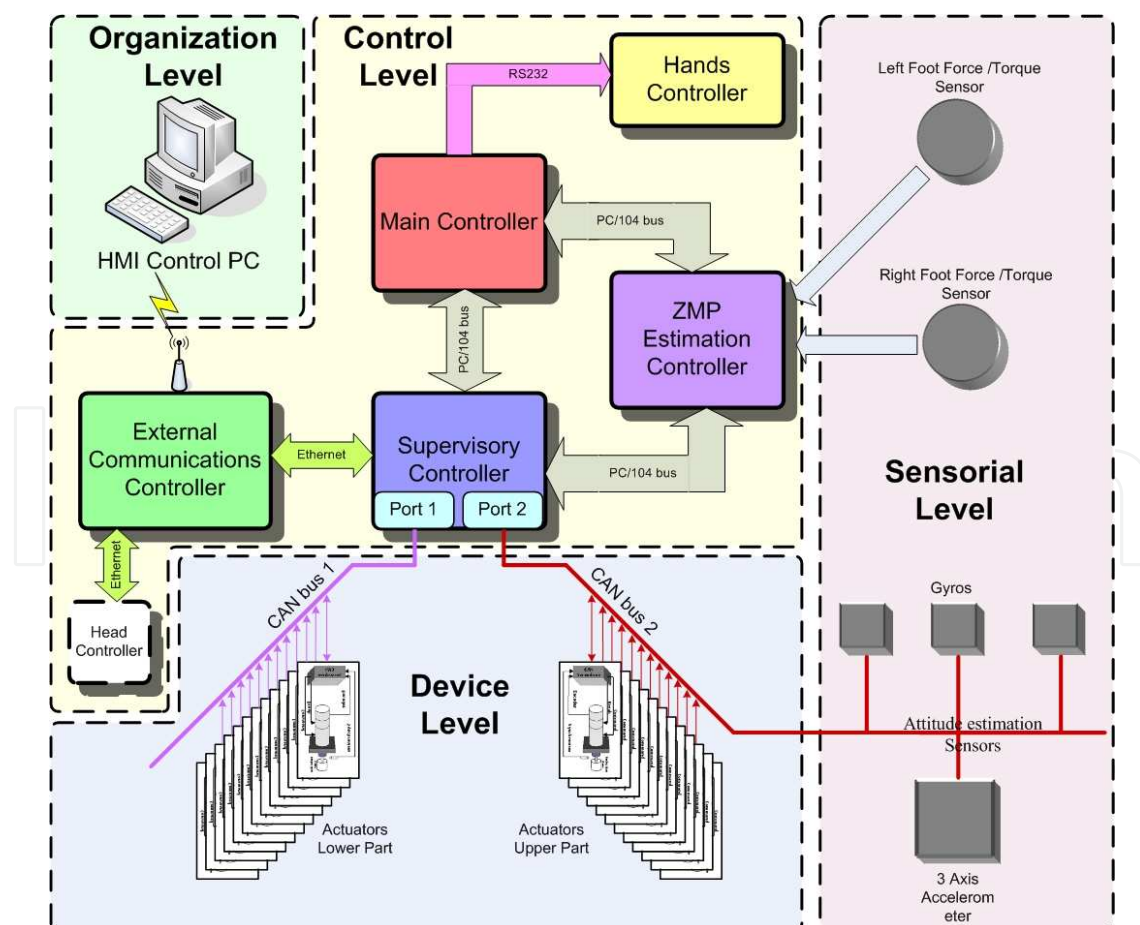


Fig. 8. Hardware architecture.

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.

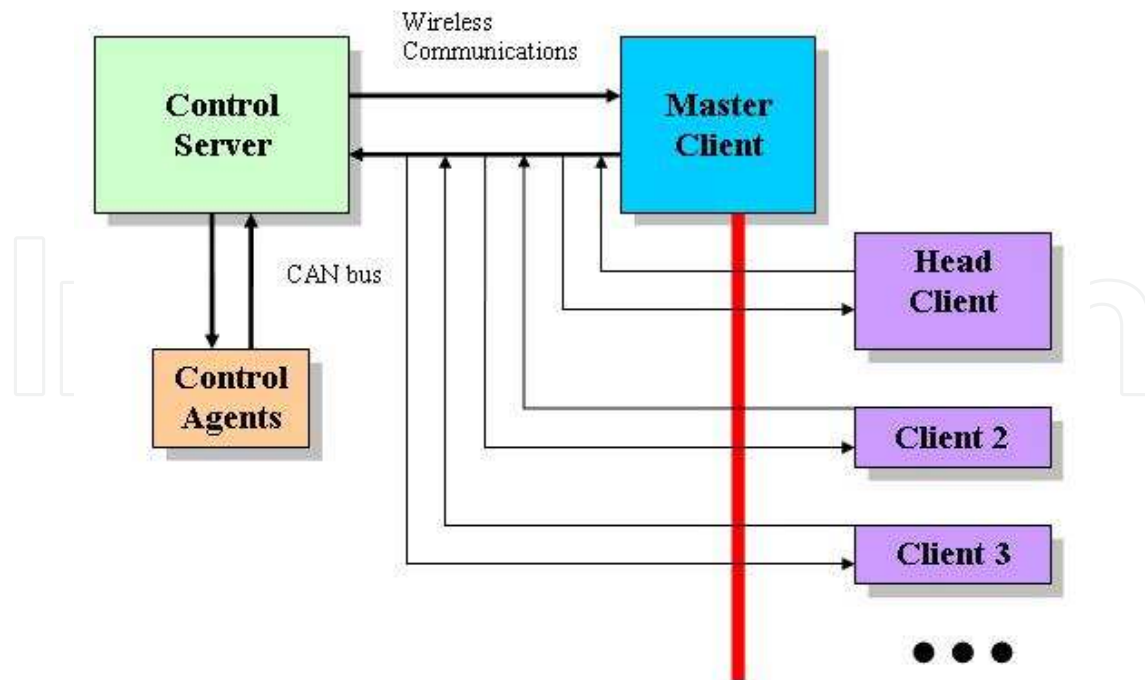


Fig. 9. System Architecture.

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

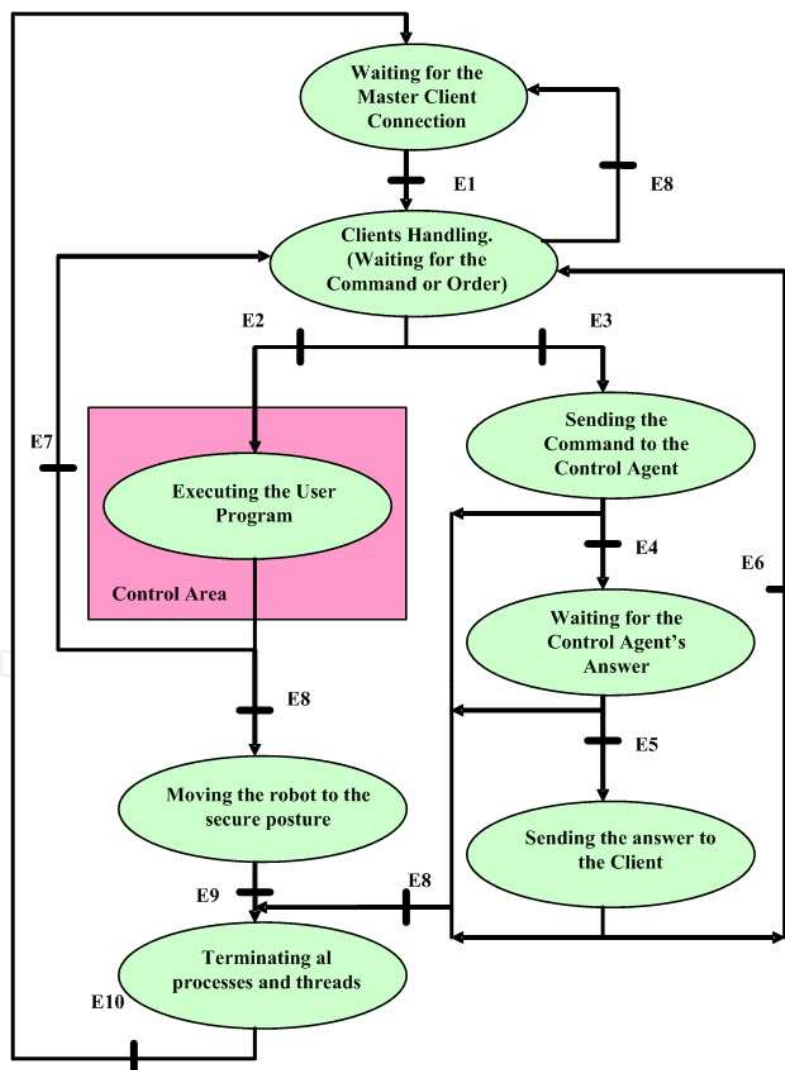


Fig. 10. Server functioning state diagram.



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.

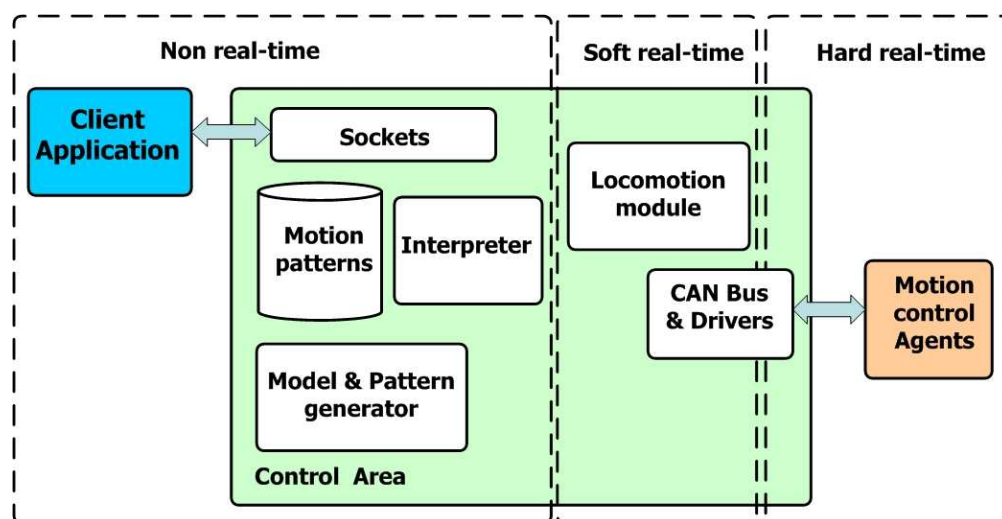


Fig. 11. Control area modules.

```
// 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.

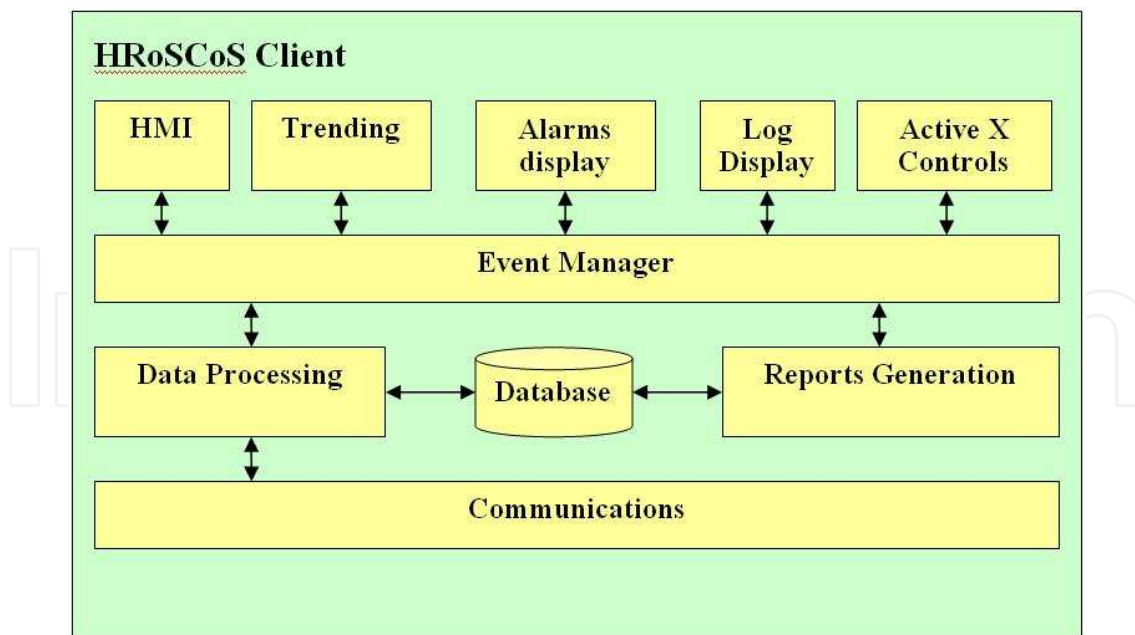


Fig. 13. HROSCoS Client Architecture.

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).

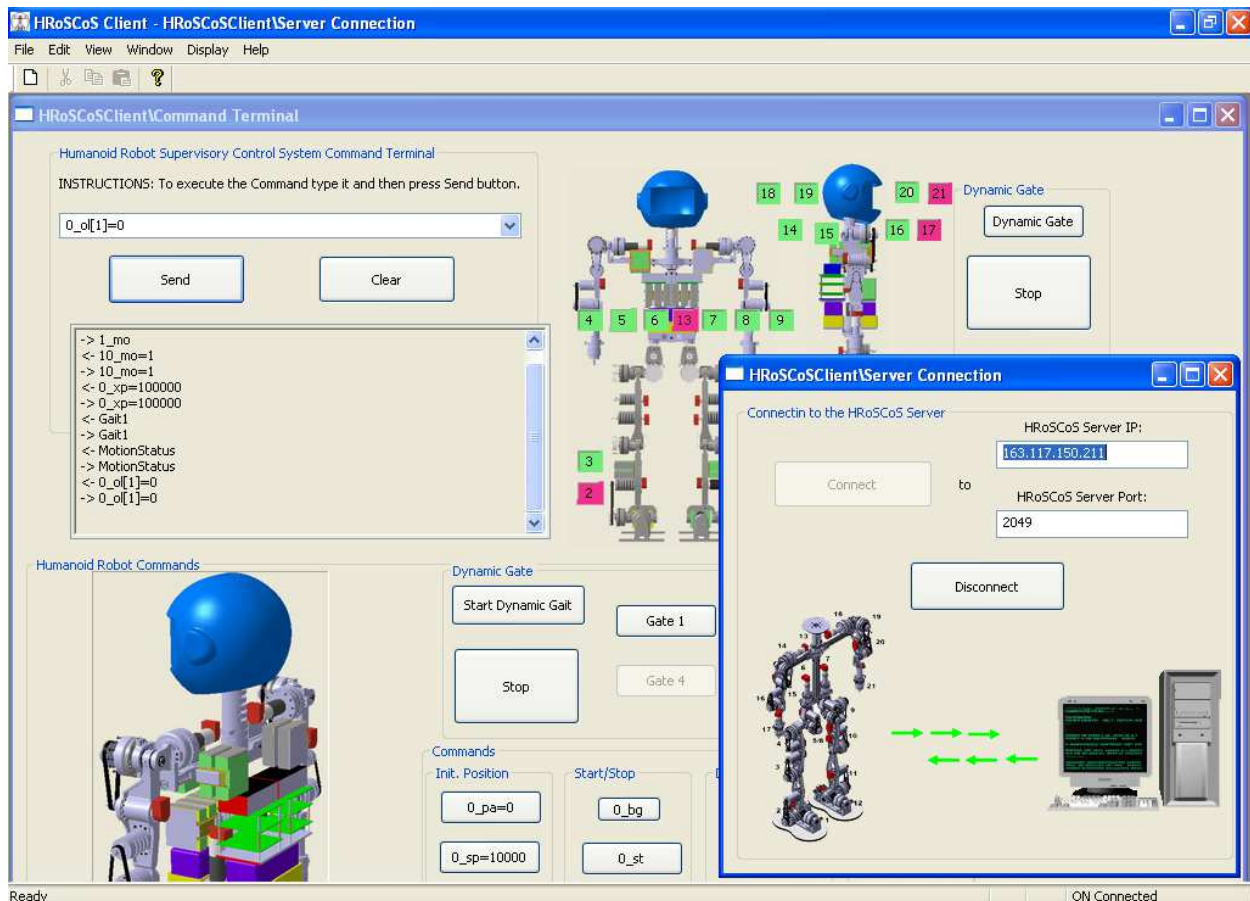


Fig. 14. HRoSCoS Client views.

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.

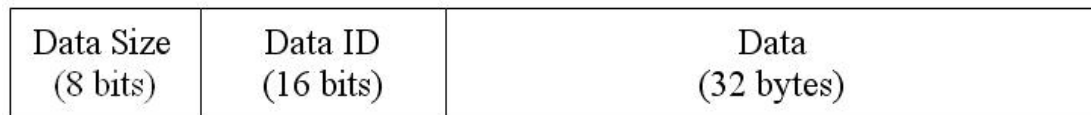


Fig. 15. The package format.

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.

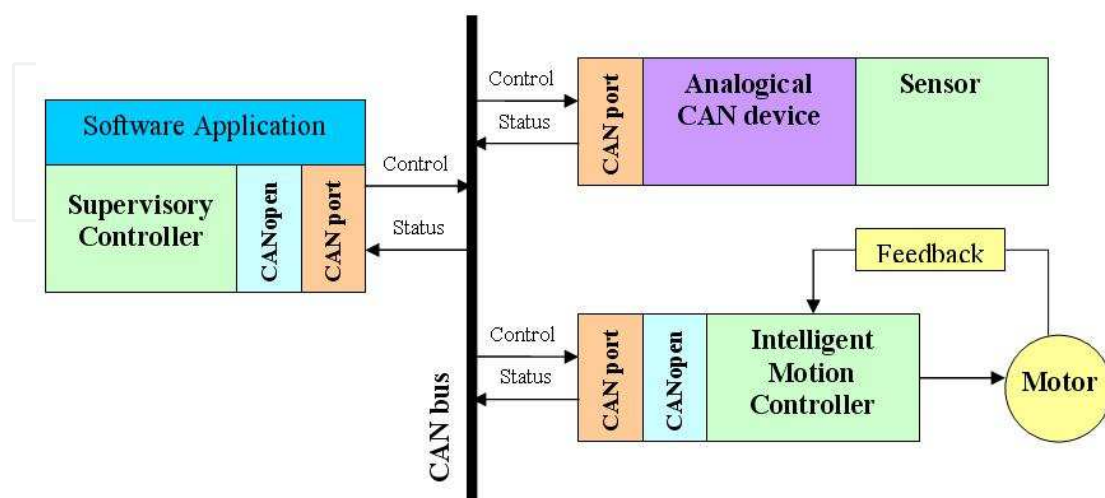


Fig. 16. CAN bus-based communication system.



## 6. Walking Pattern Generation

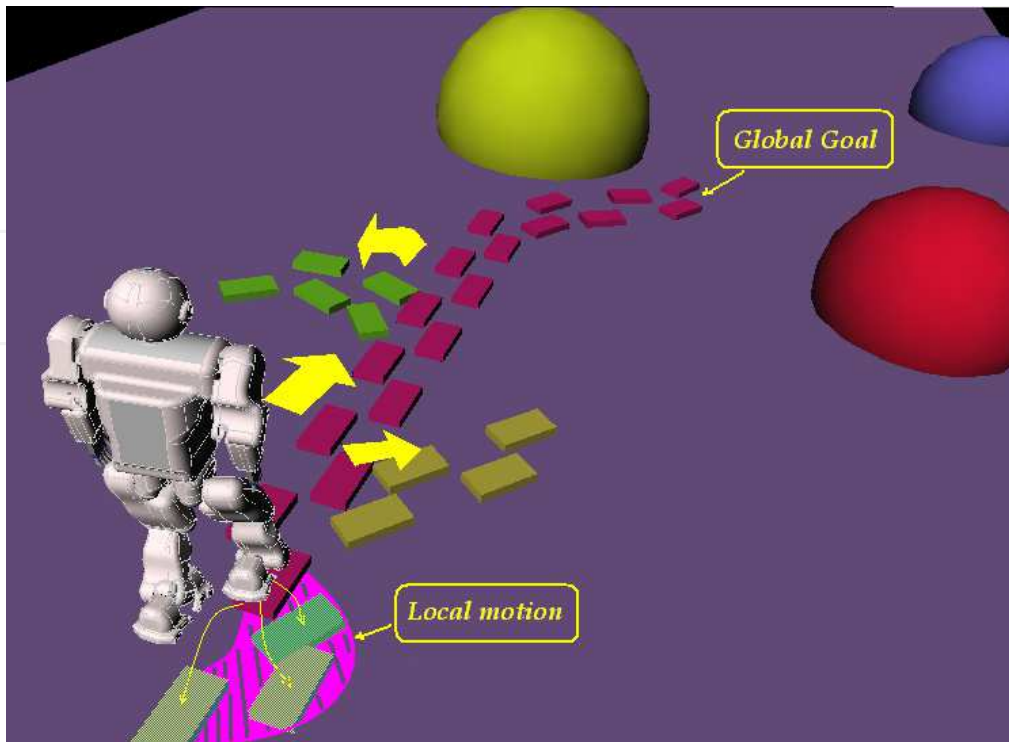


Fig. 17. Concept of gait generation method. For reaching the “Global goal”, a set of “Local motions” must be generated. Thus the local motion decide the better foot location for going ahead, go back, turn left, turn right, doing lateral step, climbing a stair or ramp.

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.



The 2D inverted pendulum model is composed of a mass and a telescopic leg without mass (Fig. 18).

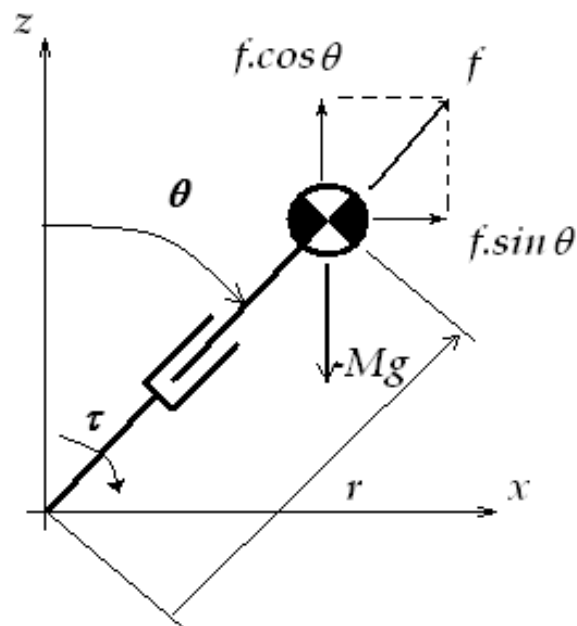


Fig. 18. The 2D Inverted Pendulum Model with motion in the x-z plane.

So, the model is described in the next state variables:

$r$  : Radius of Position vector  
(massless and telescopic leg)

$\theta$  : Pitch angle

$f$  : Reaction Force on pendulum

$\tau$  : External Pitch torque

From the free body diagram of the pendulum ball, the dynamics equations should be written as follows:

$$F_x = f \sin \theta + \frac{\tau}{r} \cos \theta \quad (1)$$

$$F_z = f \cos \theta - Mg - \frac{\tau}{r} \sin \theta \quad (2)$$

It is known that  $p=(x, z)$ , so the dynamics equations of the pendulum ball motion are:

$$F_x = M\dot{x} = f \sin \theta + \frac{\tau}{r} \cos \theta \quad (3)$$

$$F_z = M\dot{z} = f \cos \theta - Mg - \frac{\tau}{r} \sin \theta \quad (4)$$

There are several solutions for the ball pendulum motion from this complex dynamic model. In order to simplify the dynamic problem, some constraints could be taken:

- 1) Motion at constraint height:

$$z = z_c \quad (5)$$

$$\ddot{z} = 0 \quad (6)$$

- 2) It is possible to consider natural pendulum ball motion, so the input torque turns to zero:

$$\tau = 0 \quad (7)$$

From these constraints the dynamic equations (3) and (4) reduce the dynamic motion to a linear one:

$$F_x = M\dot{x} = f \sin \theta \quad (8)$$

$$F_z = 0 = f \cos \theta - Mg \quad (9)$$

By combining equations (8) and (9), the dynamic pendulum ball motion is obtained as:

$$\ddot{x} = g \frac{x}{z_c} \quad (10)$$

The natural motion of the pendulum ball depends on the potential gravity field ( $g$ ), position ( $x, z$ ) and distance from the pendulum base ( $z_c$ ). Thus, no linear dynamic motion equations are converted to linear ones; this way a single solution could be found and this kind of trajectory is applicable in real-time applications of walking locomotion.

In order to design walking patterns and to determine the spatial geometry of trajectories, the concept of orbital energy is introduced. Orbital energy evaluates the pendulum ball energy at the level of the motion plane. It is composed of the potential and kinetic energy of the pendulum ball. In this way, it is possible to determine if the pendulum motion is in a state of equilibrium, going forward, or if it never passes the zero position.

The mathematical expression of orbital energy could be developed by multiplying equation (10) by  $x$  and integrating it.

$$\dot{x} \left( \ddot{x} - g \frac{x}{z_c} \right) = 0 \quad (11)$$

$$\int \left( \ddot{x} - g \frac{x}{z_c} \right) dt = const \quad (12)$$

$$\frac{1}{2} \dot{x}^2 - \frac{1}{2} \frac{g}{z_c} x^2 = E \quad (13)$$

Equation (13) shows that a kind of energy is conserved, which is called orbital energy. The first term represents the kinetic energy per unit mass of the body, while the second one is the virtual energy caused by a force field that generates a force  $(g/z_c) \cdot x$  on the unit mass located at  $x$ . Furthermore (Fig. 19),  $E > 0$  means that the pendulum mass swings forward;  $E = 0$  represents: the equilibrium state, the pendulum mass swinging toward the equilibrium point or the pendulum mass swinging out from the equilibrium point; finally  $E < 0$  means that the body never passes point  $x = 0$ .

At this point, it is possible to generate two-dimensional stable natural walking patterns. This study is the basis for obtaining the solution of three-dimensional walking patterns, suitable for any humanoid robot. Human-like walking motion can be obtained because biomechanical studies demonstrate that COG human motion on the walking cycle could be approached by an inverted pendulum motion. The next section focuses on three-dimensional pendulum motion.

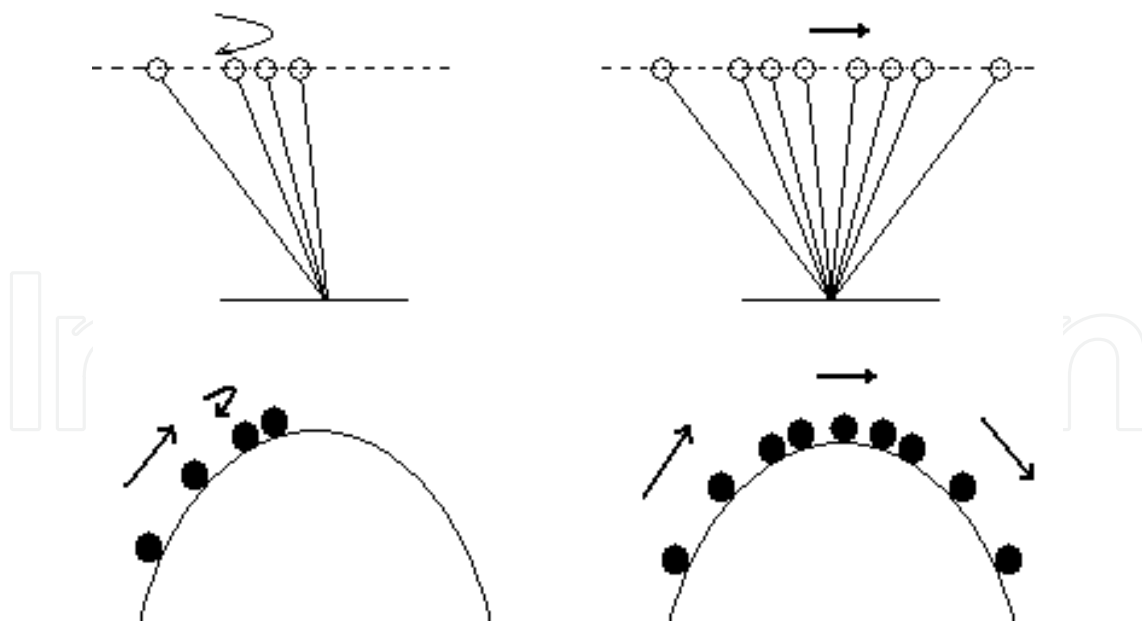


Fig. 19. Pendulum ball rolling in potentials.

### 6.2 Motion Laws of 3D-LIPM. Inverted Pendulum Model

In Figure 20, the three dimensional Inverted pendulum model is shown consisting of a point mass ( $p$ ) and a mass less telescopic leg, where  $p=(x, y, z)$  is the position of the mass  $M$ , which is uniquely specified by variables  $q= (\theta_r, \theta_p, r)$ , so:

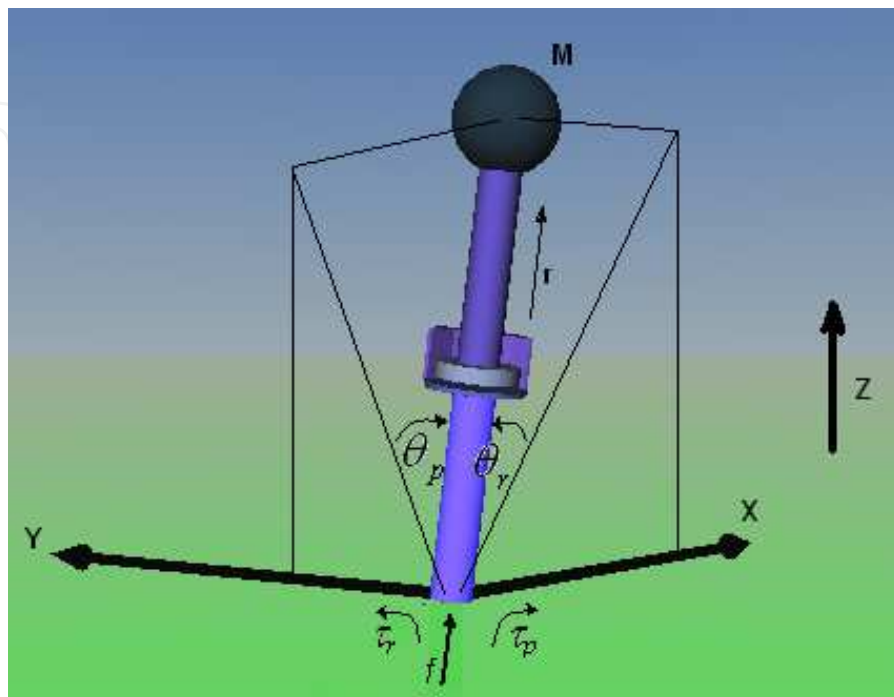


Fig. 20. Three dimensional Inverted Pendulum Model.

Following the right hand rule:

$$x = r \sin(\theta_p) \tag{14}$$

$$y = -r \sin(\theta_r) \tag{15}$$

$$z = r \sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2} \tag{16}$$

The motion equation of the inverted pendulum in Cartesian coordinates is:

$$\begin{pmatrix} \tau_r \\ \tau_p \\ f \end{pmatrix} = m \begin{pmatrix} 0 & -r \cos(\theta_r) & -\frac{r \cos(\theta_r) \sin(\theta_r)}{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}} \\ r \cos(\theta_p) & 0 & -\frac{r \cos(\theta_p) \sin(\theta_p)}{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}} \\ \sin(\theta_p) & -\sin(\theta_r) & \sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2} \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} + mg \begin{pmatrix} -\frac{r \cos(\theta_r) \sin(\theta_r)}{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}} \\ -\frac{r \cos(\theta_p) \sin(\theta_p)}{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}} \\ \sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2} \end{pmatrix} \tag{17}$$

So the dynamics along the  $x$ -axis is given by:

$$m(z\ddot{x} - x\ddot{z}) = \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_p)} \tau_p + mgx \quad (18)$$

And the equation for the dynamics along the y-axis is given by:

$$m(-z\ddot{y} + y\ddot{z}) = \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_r)} \tau_r - mgy \quad (19)$$

### 6.3 Natural three dimensional Linear Inverted Pendulum Mode (3D-LIPM)

In order to reduce the motion possibilities of the pendulum, we introduce some constraints to limit this motion. One constraint limits the motion in a plane, so:

$$z = k_x x + k_y y + z_c \quad (20)$$

Where  $z_c$  is the distance from the xy-plane to the pendulum mass. Replacing (20) and its second derivative into (18) and (19), we get:

$$\ddot{x} = \frac{g}{z_c} x + \frac{k_y}{z_c} (x\ddot{y} - \ddot{x}y) + \frac{1}{mz_c} \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_p)} \tau_p \quad (21)$$

$$\ddot{y} = \frac{g}{z_c} y - \frac{k_x}{z_c} (x\ddot{y} - \ddot{x}y) - \frac{1}{mz_c} \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_r)} \tau_r \quad (22)$$

The above equations allow pendulum motion in any plane and slope, if the motion is constrained to a flat plane ( $k_x=0$  and  $k_y=0$ ), so:

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{mz_c} \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_r)} \tau_r \quad (23)$$

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{mz_c} \frac{\sqrt{1 - \sin(\theta_r)^2 - \sin(\theta_p)^2}}{\cos(\theta_p)} \tau_p \quad (24)$$

Note that (23) and (24) are independent equations and no linear dynamics is simplified to a linear one.

The natural 3D-LIPM takes into account the trajectories of the Inverted Pendulum Model without input torques. Hence, the equations (23) and (24) are simplified to:

$$\ddot{x} = \frac{g}{z_c} x \quad (25)$$



$$\ddot{y} = \frac{g}{z_c} y \quad (26)$$

Solving those equations (25) and (26), a three dimensional pendulum ball motion is obtained in the gravity field. Figure 21 shows an example of the pendulum motion for a different support foot, i.e. blue pendulum motion for the left support foot at its local frame and red pendulum motion for right support foot at its local frame.

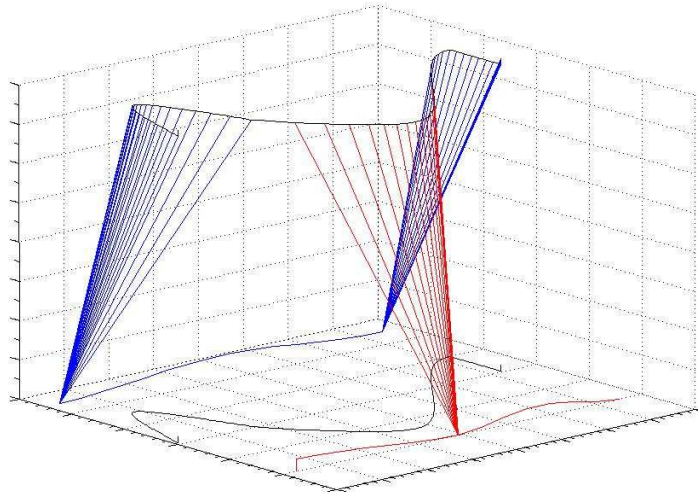


Fig. 21. Inverted Pendulum motion under natural 3D-LIPM.

#### 6.4 Geometry of trajectory

The pendulum spatial motion in the gravity field should be studied and analyzed in order to predict the stability and suitable three-dimensional local motion. So, describing the local motion (Fig. 22) in any rotational axis, it is possible to study the gravitational effects on natural pendulum motion, which is like potential energy acting on a space shuttle.

Orbital energy on each axis being:

$$E_x = \frac{1}{2} \dot{x}^2 - \frac{1}{2} \frac{g}{z_c} x^2 \quad (27)$$

$$E_y = \frac{1}{2} \dot{y}^2 - \frac{1}{2} \frac{g}{z_c} y^2 \quad (28)$$

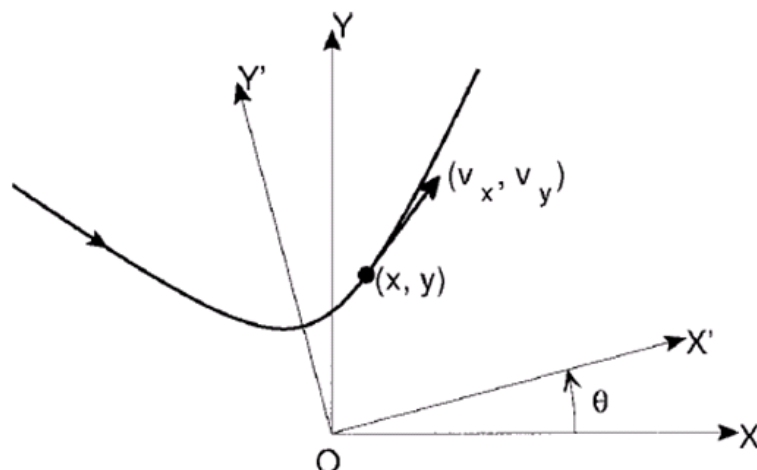


Fig. 22. 3D-LIPM projected onto the XY plane at the local axis.

The orbital energy on  $X'Y'$  axis is obtained as follows:

$$E'_x = \frac{1}{2} (\dot{x} \cos \theta + \dot{y} \sin \theta)^2 - \frac{1}{2} \frac{g}{z_c} (x \cos \theta + y \sin \theta)^2 \quad (29)$$

$$E'_y = \frac{1}{2} (-\dot{x} \sin \theta + \dot{y} \cos \theta)^2 - \frac{1}{2} \frac{g}{z_c} (-x \sin \theta + y \cos \theta)^2 \quad (30)$$

We can calculate the axis of symmetry by solving the variation of orbital energy with respect to the rotation angle, and in this way the maximum is found; the mathematical expression is developed as follows:

$$\frac{\partial E'_x}{\partial \theta} = A[(\sin \theta)^2 - (\cos \theta)^2] + B \sin \theta \cos \theta = 0 \quad (31)$$

$$A = \left( \frac{g}{z_c} \right) xy - \dot{x}\dot{y} \quad (32)$$

$$B = \left( \frac{g}{z_c} \right) (x^2 - y^2) - (\dot{x}^2 - \dot{y}^2) \quad (33)$$

Finding the symmetry axis from equation (31), by trigonometric identities:

$$\frac{A}{B} = - \frac{\sin \theta \cdot \cos \theta}{(\sin \theta)^2 - (\cos \theta)^2} \quad (34)$$

$$\theta = \frac{1}{2} \tan^{-1} \left( \frac{2A}{B} \right) \quad (35)$$

It is well known that the  $y$ -axis is the axis of symmetry for  $\theta = 0$ , so:

$$A = \left( \frac{g}{z_c} \right) xy - \dot{x}\dot{y} = 0 \quad (36)$$

$$\left( \frac{g}{z_c} \right) xy = \dot{x}\dot{y} \quad (37)$$

Eq. (37) could be used for computing the 3D-LIPM geometric shape with the orbital energy mathematical expressions from eq. (27) and (28):

$$\left( \frac{g}{z_c} \right)^2 x^2 y^2 = (2E_x + (g/z_c)x^2)(2E_y + (g/z_c)y^2) \quad (38)$$

By simplifying the last equation an interesting expression is found which describes the shape of the pendulum mass trajectory in the gravity field (eq. 39):

$$\frac{g}{2z_c E_x} x^2 + \frac{g}{2z_c E_y} y^2 = -1 \quad (39)$$

It is possible to deduce that  $E_x > 0$ , because the  $x$ -axis pendulum passes  $\mathbf{0}$  of the local frame and  $E_y < 0$ , due to the fact that the  $y$ -axis pendulum does not pass  $\mathbf{0}$  of the local frame (Fig. 19). These facts show us that the pendulum mass trajectory shape is a hyperbolic curve described by eq. (39). Furthermore, the natural pendulum mass motion in three dimensions gives us information about the motion range for several initial conditions, which could be applied to the single support phase of the humanoid body motion.

## 6.5 Temporal equations

With initial conditions  $(x_i, \dot{x}_i)$  and  $(y_i, \dot{y}_i)$  at time  $t_i$ , the mass trajectory is calculated by solving differential equations (25) and (26):

$$x(t) = x_i \cosh\left(\frac{t-t_i}{T_c}\right) + T_c \dot{x}_i \sinh\left(\frac{t-t_i}{T_c}\right) \quad (40)$$

$$\dot{x}(t) = \frac{x_i}{T_c} \sinh\left(\frac{t-t_i}{T_c}\right) + \dot{x}_i \cosh\left(\frac{t-t_i}{T_c}\right) \quad (41)$$

$$y(t) = y_i \cosh\left(\frac{t-t_i}{T_c}\right) + T_c \dot{y}_i \sinh\left(\frac{t-t_i}{T_c}\right) \quad (42)$$

$$\dot{y}(t) = \frac{y_i}{T_c} \sinh\left(\frac{t-t_i}{T_c}\right) + \dot{y}_i \cosh\left(\frac{t-t_i}{T_c}\right) \quad (43)$$

### 6.6 Cart-table model

In order to establish a relationship between the COG and ZMP motion, the cart-table model is proposed. This model, by controlling cart acceleration, gives us an interesting relationship between the ZMP and the COG.

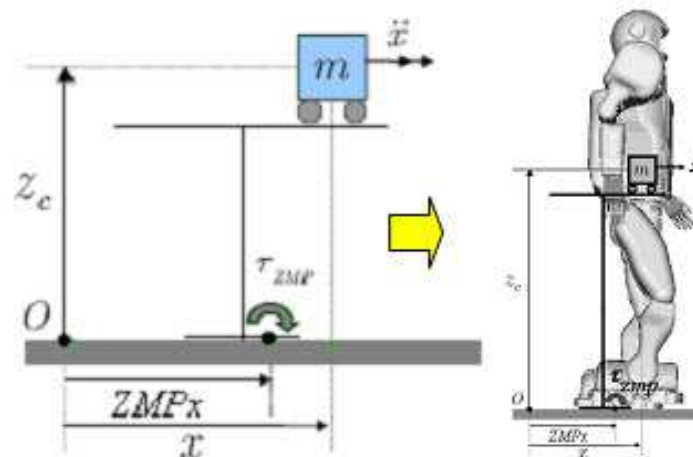


Fig. 23. Cart-table model.

By evaluating the torque in the ZMP (Fig. 23):

$$\tau_{zmp} = mg(x - ZMP_x) - m\ddot{x}Z_c \quad (44)$$

As we know the torque in the ZMP is zero, thus from the eq. (44),  $ZMP_x$  is being by:

$$ZMP_x = x - \frac{Z_c}{g}\ddot{x} \quad (45)$$

Note that eq. (45) is similar to the inverted pendulum (24), with the main difference being that  $ZMP_x$  is constrained to 0, while if we knew it, it would be fixed to any point on the local axis. In the y direction, a similar equation could be obtained. In order to get the COG motion as an inverse problem from the ZMP one, the solution of eq. (45) should be treated as a servo control problem.

### 6.7 Comparing pendulum and Cart-table models

In the inverted pendulum model the input reference is the ZMP and the output is the COG pattern. Note that the ZMP is always at the base of the pendulum (i.e. Fig. 24b). In the cart-table model, ZMP motion is instead generated by the COG as reference (i.e. Fig. 24a).

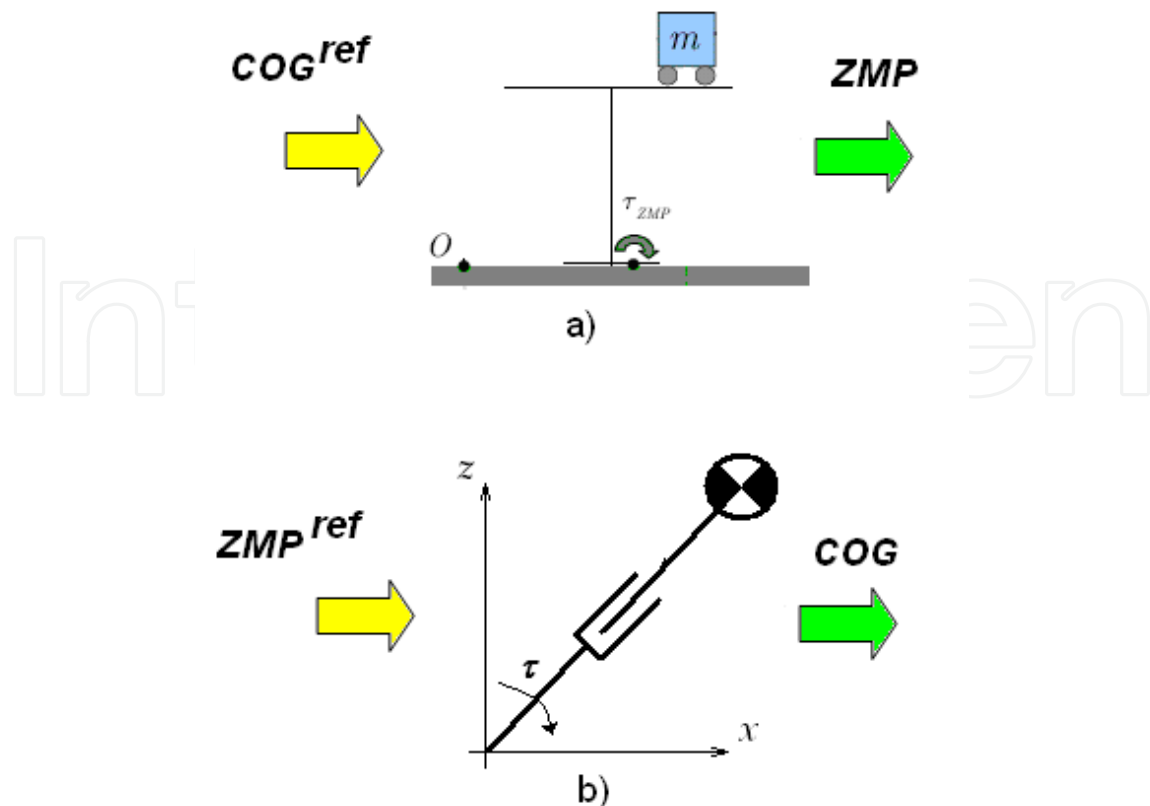


Fig. 24. Comparison between the: a) Cart-table, and b) Inverted pendulum.

Other facts are that there is a discontinuity in the change from a single support phase to a double one, so at high walking velocity, jerk is an important fact; it could be improved by using high-order splines. Thus, the cart-table model optimizes jerk and continuity is maintained at all times, no matter the change of phase, and in this way a high walking speed is possible.

### 6.8 Walking pattern strategy

Figure 25 show us the steps of the walking strategy, in the single support phase the pendulum ball follows 3D-LIPM laws (A to B, C to D and E to F); in the double support phase, the pendulum ball moves at a constant speed (B to C and D to E). This motion drives the COG of the humanoid robot. We could assume that the COG is in the middle of the hip joint. Foot trajectories are computed by single splines taking into account some constraints such as step length, maximum height of the foot, lateral foot motion, foot orientation and speed in order to avoid falling down and to reduce the impact force on the landing foot (Fig. 26).

### 6.9 Local Axis Gait algorithm (LAG)

In order to generalize the walking patterns of any direction and surface such as stairs or slopes; the "Local Axis Algorithm" (LAG) is proposed, (Arbulu et al. 2007) so as to plan the stable local walking motion. The LAG is divided into several stages: computation of the footprints; the decision of the ZMP limits around the footprints; the dynamic humanoid COG motion generation based on the mass concentrated model; and finally joining the



footprints of the swing foot by splines. In this way, it is possible to generate each step online, using the desired footprints as input.

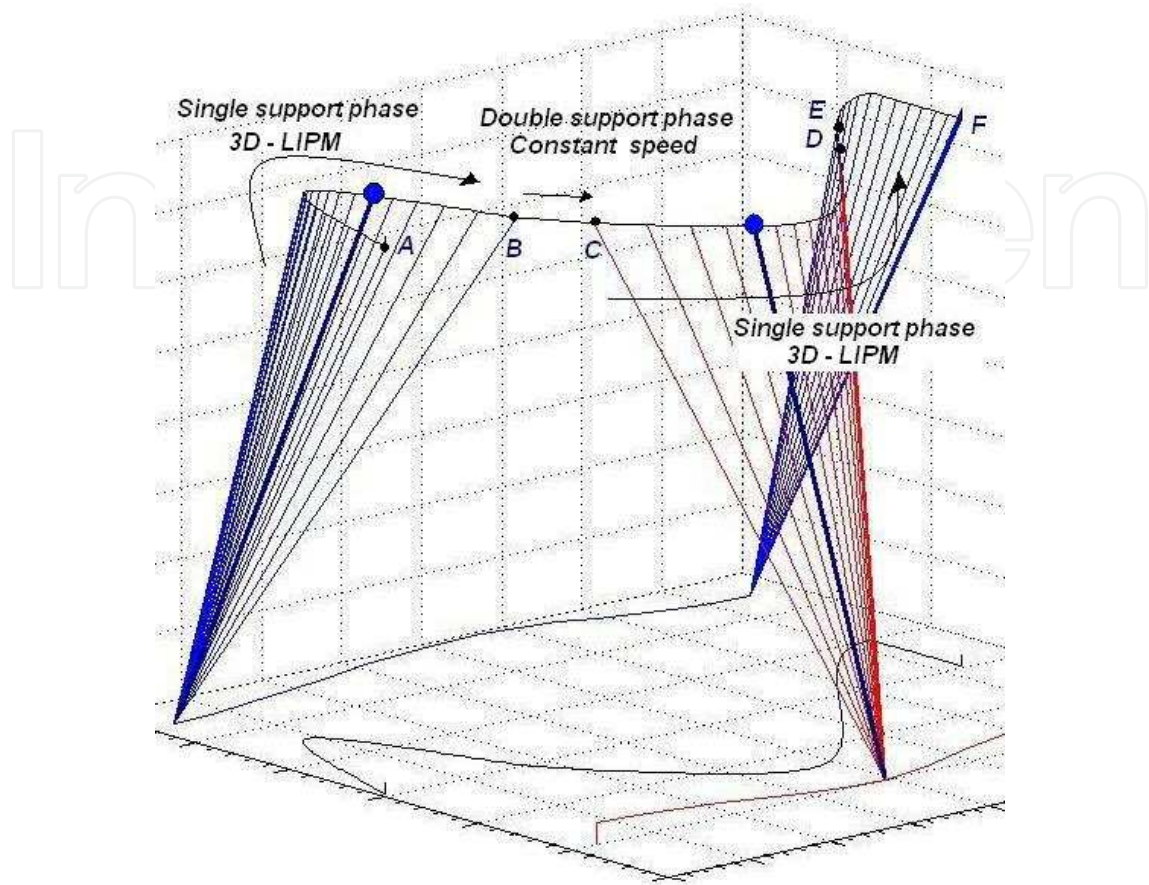


Fig. 25. Walking pattern strategy

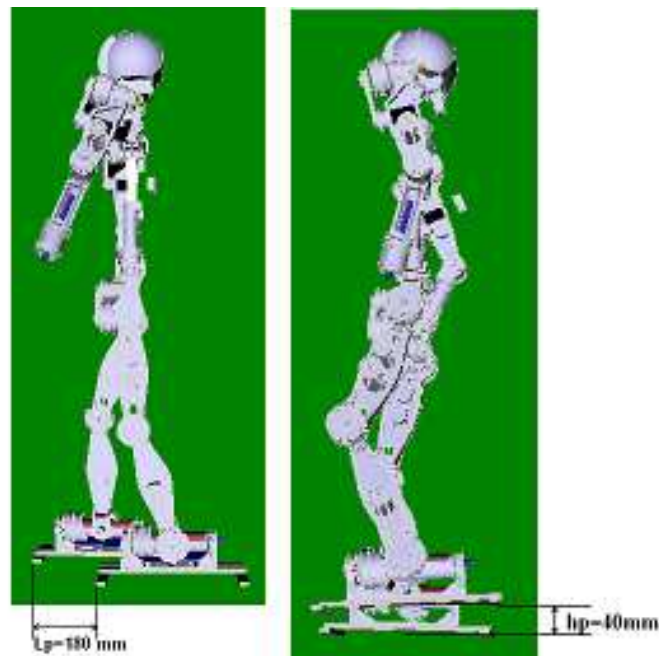


Fig. 26. Some foot trajectory constraints: max step length and max swing foot height.

The footprints (Fig. 27) for doing an n-th step can be computed as follows:

$$P^n = P^n + R(\theta_z^n)^T \cdot L^n \tag{46}$$

Where:

$$P^n = (p_x^n \quad p_y^n \quad p_z^n)^T,$$

$$L^n = (L_x^n \quad -(-1)^n L_y^n \quad L_z^n)^T$$

$\Sigma, \Sigma^n, \Sigma^{n-1}, \Sigma^{n+1}$  : World and feet frames

$P^n, P^{n-1}, P^{n+1}$  : feet position

$L_x^{n+1}, L_y^{n+1}, L_z^{n+1}$  : swing foot displacements

$\theta_x^{n+1}, \theta_y^{n+1}, \theta_z^{n+1}$  : rotations about world frame

The walking patterns developed are introduced into the inverse kinematics algorithm (Arbulu et al. 2005) to obtain the angular evolution of each joint; those are the reference patterns of the humanoid robot.

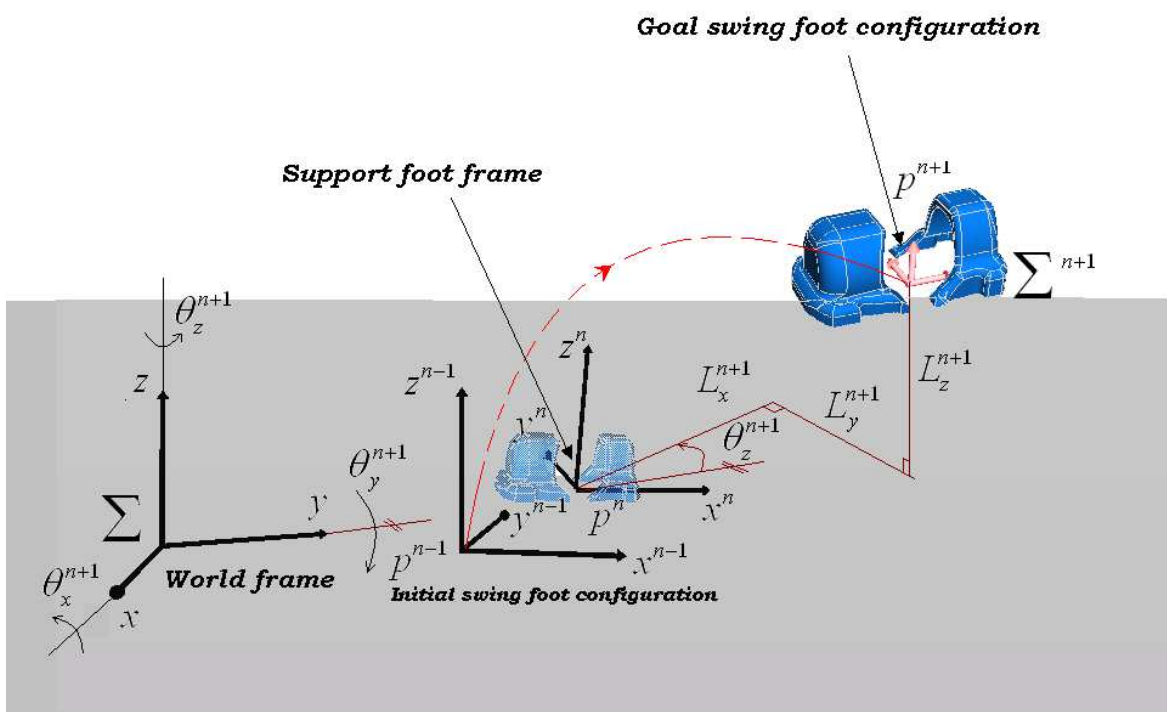


Fig. 27. Footprint location.

### 6.10 Inverse Kinematics model

In order to compute the robot's joint motion patterns some kinematics considerations must be made. Due to the fact that the kinematics control is based on screw theory and Lie logic techniques, it is also necessary to present a basic explanation.

#### *Lie logic background*

Lie groups are very important for mathematical analysis and geometry because they serve to describe the symmetry of analytical structures (Park et. al. 1985). A Lie group is an analytical manifold that is also a group. A Lie algebra is a vectorial space over a field that completely captures the structure of the corresponding Lie group. The homogeneous representation of a rigid motion belongs to the special Euclidean Lie group ( $SE(3)$ ) (Abraham et. al. 1999). The Lie algebra of  $SE(3)$ , denoted  $se(3)$ , can be identified with the matrices called twists " $\xi^\wedge$ ", (eq. 47), where the skew symmetric matrix " $\omega^\wedge$ ", (eq. 48) is the Lie algebra  $so(3)$  of the orthogonal special Lie group ( $SO(3)$ ), which represents all rotations in the three-dimensional space. A twist can be geometrically interpreted using screw theory (Paden 1986), as Charles's theorem proved that any rigid body motion could be produced by a translation along a line followed by a rotation around the same line; this is a screw motion, and the infinitesimal version of a screw motion is a twist.

$$\xi^\wedge = \begin{bmatrix} \omega^\wedge & v \\ 0 & 0 \end{bmatrix} \in se(3) / se(3) = \{(\omega^\wedge, v) : v \in \mathbb{R}^3, \omega^\wedge \in so(3)\} \in \mathbb{R}^{4 \times 4} \quad (47)$$

$$\omega^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_3 & \omega_1 & 0 \end{bmatrix} / \forall \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \wedge v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \Rightarrow \omega \times v = \omega^\wedge v \quad (48)$$

The main connection between  $SE(3)$  and  $se(3)$  is the exponential transformation (eq. 49). It is possible to generalize the forward kinematics map for an arbitrary "open-chain" manipulator with  $n$  DOF of magnitude  $g(\theta)$ , through the product of those exponentials, expressed as POE (eq. 50), where  $g(0)$  is the reference position for the coordinate system.

$$\left. \begin{aligned} e^{\xi^\wedge \theta} &= \begin{bmatrix} e^{\omega^\wedge \theta} & (I - e^{\omega^\wedge \theta})(\omega^\wedge v) + \omega^\wedge v \theta \\ 0 & 1 \end{bmatrix} \in SE(3); \omega \neq 0 \\ e^{\xi^\wedge \theta} &= \begin{bmatrix} I & v \theta \\ 0 & 1 \end{bmatrix} \in SE(3); \omega = 0 \\ e^{\xi^\wedge \theta} &= I + \omega^\wedge \sin \theta + \omega^\wedge{}^2 (1 - \cos \theta) \end{aligned} \right\} \quad (49)$$

$$g(\theta) = \prod_{i=1}^n e^{\xi_i^\wedge \theta_i} \cdot g(0) \quad (50)$$

A very important payoff for the POE formalism is that it provides an elegant formulation of a set of canonical problems, the Paden and Kahan sub-problems, (Pardos et. al. 2005, Arbulu et. al. 2005) among others, which have a geometric solution for their inverse kinematics. It is possible to obtain a close-form solution for the inverse kinematics problem of complex mechanical systems by reducing them into the appropriate canonical sub-problems.

The Paden and Kahan sub problems are introduced as following (Murray et al. 1994):

**Paden-Kahan 1: Rotation about a single axis**

Finding the rotation angle using “screw theory” and Lie groups, at first, the point rotation expression from “p” to “k” is expressed by (Fig. 28):

$$e^{\xi^{\wedge} \theta} p = k \tag{51}$$

The twist and projection vectors on the rotation plane are as follows:

$$\xi = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega \times r \\ \omega \end{bmatrix} \tag{52}$$

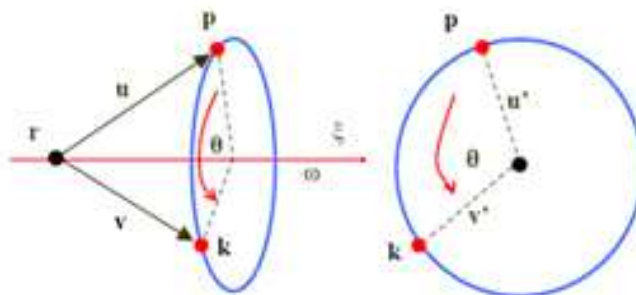


Fig. 28. Rotation on single axis “ω” from point “p” to point “k”.

$$u' = u - \omega \omega^T u \tag{53}$$

$$v' = v - \omega \omega^T v \tag{54}$$

Finally, the rotation angle is calculated with the following expression:

$$\theta = a \tan 2 \left[ \omega^T (u' \times v'), u'^T \cdot v' \right] \tag{55}$$

**Paden-Kahan 2: Rotation about two subsequent axes**

The rotation expression is the following (Fig. 29):

$$e^{\xi_1 \hat{\theta}_1} e^{\xi_2 \hat{\theta}_2} p = e^{\xi_1 \hat{\theta}_1} c = k \quad (56)$$

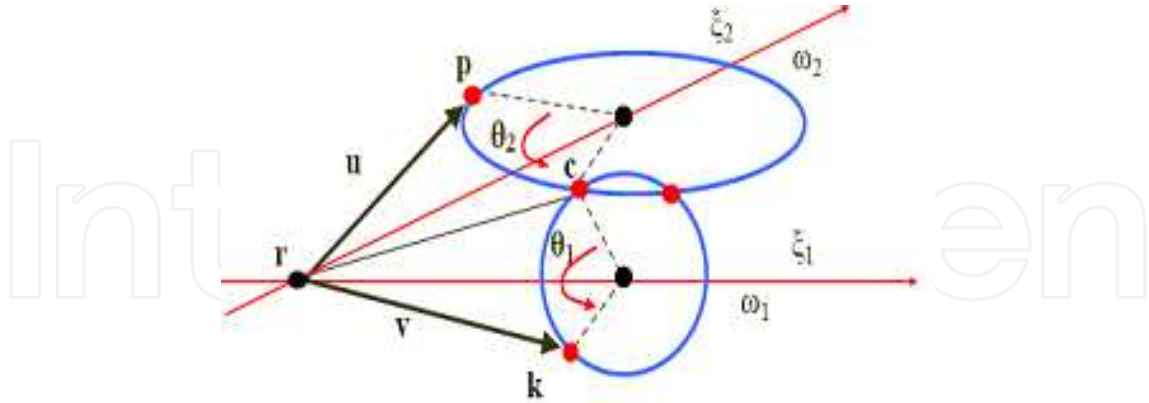


Fig. 29. Rotation on two subsequent axes " $\omega_1$ " and " $\omega_2$ " from " $p$ " to " $c$ " and from " $c$ " to " $k$ ".

The respective twists are described as follows:

$$\xi_1 = \begin{bmatrix} -\omega_1 \times r \\ \omega_1 \end{bmatrix} \wedge \xi_2 = \begin{bmatrix} -\omega_2 \times r \\ \omega_2 \end{bmatrix} \quad (57)$$

Some values are computed in order to obtain the point " $c$ " with the following expressions:

$$\alpha = \frac{(\omega_1^T \omega_2) \omega_2^T u - \omega_1^T v}{(\omega_1^T \omega_2)^2 - 1} \quad (58)$$

$$\beta = \frac{(\omega_1^T \omega_2) \omega_1^T v - \omega_2^T u}{(\omega_1^T \omega_2)^2 - 1} \quad (59)$$

$$\gamma^2 = \frac{\|u\|^2 - \alpha^2 - \beta^2 - 2\alpha\beta\omega_1^T \omega_2}{\|\omega_1 \times \omega_2\|^2} \quad (60)$$

Obtaining the point " $c$ ":

$$c = r + \alpha\omega_1 + \beta\omega_2 \pm \gamma(\omega_1 \times \omega_2) \quad (61)$$

Once we get " $c$ " for the second sub-problem, we can apply the first Paden-Kahan sub-problem to obtain the solutions for  $\theta_1$  and  $\theta_2$ . Note that there might be two solutions for " $c$ ", each of them giving a different solution for  $\theta_1$  and  $\theta_2$ .

***Paden-Kahan 3: Rotation to a given distance***

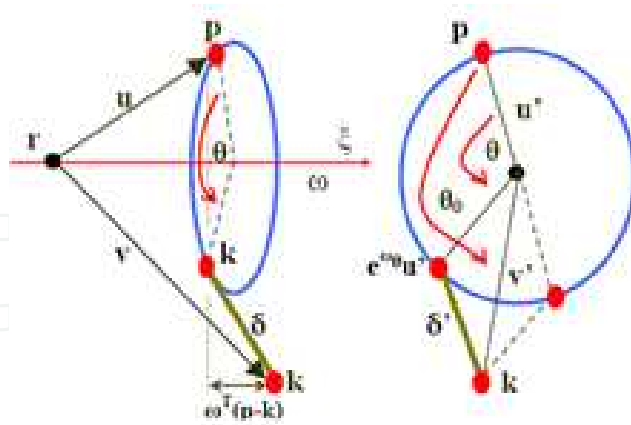


Fig. 30. Rotation of point “p” to “k” which is a distance “δ” from “q”.

The distance “δ” is shown as follows:

$$\|e^{\xi \wedge \theta} p - q\| = \delta \tag{62}$$

The associate twist and vectors projection in the perpendicular plane of rotation axis could be computed as:

$$\xi = \begin{bmatrix} \nu \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega \times r \\ \omega \end{bmatrix} \tag{63}$$

$$u' = u - \omega \omega^T u \tag{64}$$

$$v' = v - \omega \omega^T v \tag{65}$$

Projecting “δ” in “ω” direction:

$$\delta'^2 = \delta^2 - |\omega^T (p - q)|^2 \tag{66}$$

If we let “θ<sub>0</sub>” be the angle between the vectors “u” and “v”, we have:

$$\theta_0 = a \tan 2[\omega^T (u' \times v'), u'^T \cdot v'] \tag{67}$$

Finally, we obtain the rotation angle by:



$$\theta = \theta_0 \pm \cos^{-1} \left( \frac{\|u'\|^2 + \|v'\|^2 - \delta'^2}{2\|u'\|\|v'\|} \right) \quad (68)$$

The algorithm developed is called Sagittal Kinematics Division (SKD). It divides the robot body into two independent manipulators, one for the left and one for the right part of body (Fig. 31), subject to the following constraints at any time: keeping the balance of the humanoid ZMP and imposing the same position and orientation for the common parts (pelvis, thoracic, cervical) of the four humanoid manipulators.

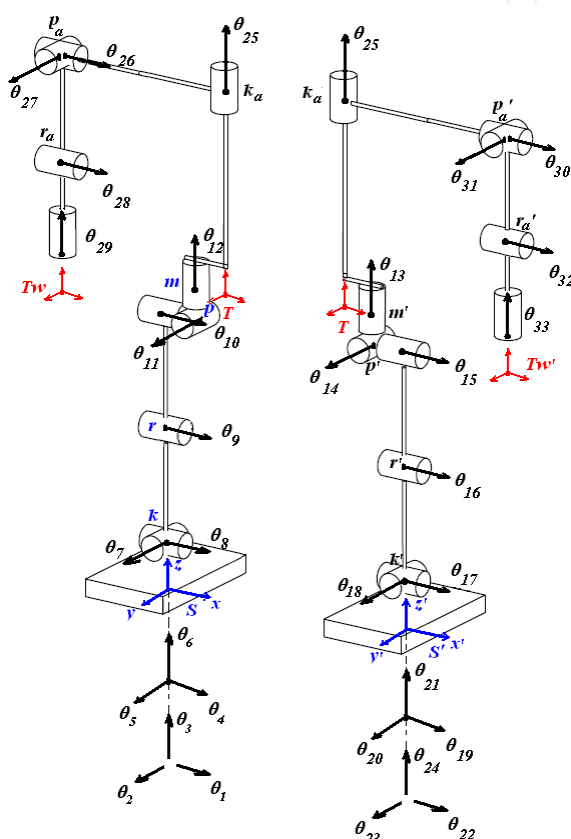


Fig. 31. Rh-1 Sagittal Kinematics Division (SKD).

### Solving the kinematics problem

It is possible to generalize the leg forward kinematics map with 12 DOF ( $\theta_1 \dots \theta_{12}$ ). The first six DOF correspond to the position ( $\theta_1, \theta_2, \theta_3$ ) and orientation ( $\theta_4, \theta_5, \theta_6$ ) of the foot. Note that these DOF do not correspond to any real joint and for that reason we call them “non-physical” DOF.

The other DOF are called “physical DOF” because they correspond to real motorized joints. These are:  $\theta_7$  for the hindfoot,  $\theta_8$  for the ankle,  $\theta_9$  for the knee,  $\theta_{10}$  for the hip on the  $x$  axis,  $\theta_{11}$  for the hip on the  $y$  axis and  $\theta_{12}$  for the hip on the  $z$  axis. Let  $S$  be a frame attached to the base system (support foot) and  $T$  be a frame attached to the humanoid hip.

The reference configuration of the manipulator is the one corresponding to  $\theta_i = 0$ , and  $g_{st}(0)$  that represents the rigid body transformation between  $T$  and  $S$  when the manipulator is at its reference configuration.

Then, the product of exponentials formula for the right and left legs forward kinematics is  $g_{st}(\theta)$  and  $g_{s't'}(\theta)$ , being  $\xi^{\wedge}$  the 4x4 matrices called "twists".

$$g_{st}(\theta) = e^{\xi_1^{\wedge} \theta_1} \cdot e^{\xi_2^{\wedge} \theta_2} \dots e^{\xi_{29}^{\wedge} \theta_{29}} \cdot g_{st}(0) \quad (69)$$

$$g_{s't'}(\theta) = e^{\xi_{24}^{\wedge} \theta_{24}} \cdot e^{\xi_{23}^{\wedge} \theta_{23}} \dots e^{\xi_{33}^{\wedge} \theta_{33}} \cdot g_{s't'}(0) \quad (70)$$

The inverse kinematics problem i.e. for the right leg (see Fig. 31) consists of finding the joint angles, that is, the six physical DOF ( $\theta_7 \dots \theta_{12}$ ), given the non-physical DOF ( $\theta_1 \dots \theta_6$ ) from the humanoid footstep planning, the hip orientation and position  $g_{st}(\theta)$ , which achieve the ZMP humanoid desired configuration. Using the PoE formula for the forward kinematics it is possible to develop a numerically stable geometric algorithm, to solve this problem, by using the Paden-Kahan (P-K) geometric sub-problems. It is straightforward to solve the inverse kinematics problem in an analytic, closed-form and geometrically meaningful way, with the following formulation.

At first, twist and reference configurations are computed:

$$g_{st}(0) = \begin{bmatrix} 1 & 0 & 0 & T_x - S_x \\ 0 & 1 & 0 & T_y - S_y \\ 0 & 0 & 1 & T_z - S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (71)$$

$$\nu_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \nu_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \nu_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \omega_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \omega_5 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \omega_6 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (72)$$

$$\omega_7 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \omega_8 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \omega_9 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \omega_{10} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \omega_{11} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \omega_{12} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (73)$$

$$\xi_1 = \begin{bmatrix} \nu_1 \\ 0 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} \nu_2 \\ 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} \nu_3 \\ 0 \end{bmatrix} \quad (74)$$

$$\xi_4 = \begin{bmatrix} -\omega_4 \times S \\ \omega_4 \end{bmatrix}; \xi_5 = \begin{bmatrix} -\omega_5 \times S \\ \omega_5 \end{bmatrix}; \xi_6 = \begin{bmatrix} -\omega_6 \times S \\ \omega_6 \end{bmatrix} \quad (75)$$

$$\xi_7 = \begin{bmatrix} -\omega_7 \times k \\ \omega_7 \end{bmatrix}; \xi_8 = \begin{bmatrix} -\omega_8 \times k \\ \omega_8 \end{bmatrix}; \xi_9 = \begin{bmatrix} -\omega_9 \times r \\ \omega_9 \end{bmatrix} \quad (76)$$

$$\xi_{10} = \begin{bmatrix} -\omega_{10} \times p \\ \omega_{10} \end{bmatrix}; \xi_{11} = \begin{bmatrix} -\omega_{11} \times p \\ \omega_{11} \end{bmatrix}; \xi_{12} = \begin{bmatrix} -\omega_{12} \times p \\ \omega_{12} \end{bmatrix} \quad (77)$$

Next, it is possible to compute the inverse kinematics as follows: angle  $\theta_9$  is obtained using the third *P-K* sub problem. We pass all known terms to the left side of the equation (69), apply both sides to point  $p$ , subtract point  $k$ , and apply the norm. We operate in such a way because the resulting equation (78) is only affected by  $\theta_9$ , and therefore we can rewrite the equation as (79), which is exactly the formulation of the Paden-Kahan canonical problem for a rotation to a given distance. Thus, we can geometrically obtain the two possible values for the variable  $\theta_9$ .

$$\left\| e^{-\xi_6 \hat{\theta}_6} \dots e^{-\xi_1 \hat{\theta}_1} g_{st}(\theta) g_{st}(0)^{-1} p - k \right\| = \left\| e^{\xi_7 \hat{\theta}_7} \dots e^{\xi_{12} \hat{\theta}_{12}} p - k \right\| \quad (78)$$

$$\delta = \left\| e^{\xi_9 \hat{\theta}_9} p - k \right\| \xrightarrow{P-K-3} \theta_9 \quad (79)$$

Next,  $\theta_7$  and  $\theta_8$  are obtained using the second *P-K* sub problem. We pass all possible terms to the left side of the equation (51) and apply both sides to point  $p$ . In so doing, the resulting equation (80) is only affected by  $\theta_7$ ,  $\theta_8$  and  $\theta_9$ , and therefore, we can rewrite the equation as (81), which is exactly the formulation of the Paden-Kahan canonical problem for two successive rotations.

Therefore, we can geometrically obtain the two possible values, for the pair of variables  $\theta_7$  and  $\theta_8$ .

$$e^{-\xi_6 \hat{\theta}_6} \dots e^{-\xi_1 \hat{\theta}_1} g_{st}(\theta) g_{st}(0)^{-1} p = e^{\xi_7 \hat{\theta}_7} \dots e^{\xi_{12} \hat{\theta}_{12}} p \quad (80)$$

$$q' = e^{\xi_7 \hat{\theta}_7} e^{\xi_8 \hat{\theta}_8} p' \xrightarrow{P-K-2} \theta_7, \theta_8 \quad (81)$$

After that,  $\theta_{10}$  and  $\theta_{11}$  are obtained using the second *P-K* sub problem. We pass all known terms to the left side of the equation (69) and apply both sides to point  $m$ . As a result of these operations, the transformed equation (82) is only affected by  $\theta_{10}$  and  $\theta_{11}$ , and we can rewrite the equation as (83), which is again the formulation of the Paden-Kahan canonical problem for two successive rotations around crossing axes. Hence, we can geometrically solve the two possible values for the pair of variables  $\theta_{10}$  and  $\theta_{11}$ .

$$e^{-\hat{\xi}_9 \theta_9} \dots e^{-\hat{\xi}_1 \theta_1} g_{st}(\theta) g_{st}(0)^{-1} m = e^{\hat{\xi}_{10} \theta_{10}} e^{\hat{\xi}_{11} \theta_{11}} e^{\hat{\xi}_{12} \theta_{12}} m \quad (82)$$

$$q'' = e^{\hat{\xi}_{10} \theta_{10}} \cdot e^{\hat{\xi}_{11} \theta_{11}} m \xrightarrow{P-K-2} \theta_{10}, \theta_{11} \quad (83)$$

Finally,  $\theta_{12}$  is obtained using the first  $P-K$  sub-problem. We pass all known terms to the left side of the equation (69) and apply both sides to point  $S$ . As a result, the equation is transformed into (84), which is obviously only affected by  $\theta_{12}$ , and we can rewrite it as (85), which is the formulation of the Paden-Kahan canonical problem for a rotation around an axis. Thus, we can geometrically obtain the single possible value for variable  $\theta_{12}$ .

$$e^{-\hat{\xi}_{11} \theta_{11}} \dots e^{-\hat{\xi}_1 \theta_1} g_{st}(\theta) g_{st}(0)^{-1} S = e^{\hat{\xi}_{12} \theta_{12}} S \quad (84)$$

$$q''' = e^{\hat{\xi}_{12} \theta_{12}} S \xrightarrow{P-K-1} \theta_{12} \quad (85)$$

The arm motion could be implemented as follows: i.e. the  $\theta_{25}$  to  $\theta_{29}$  solutions. The manipulator shoulder and wrist do not intervene in locomotion and therefore  $\theta_{25}$  and  $\theta_{29}$  are zero for the analyzed movement. The other arm DOF may or may not contribute to the locomotion, helping the balance control to keep the *COG* as close as possible to its initial geometric position; but to achieve this behavior, we must solve the arm inverse dynamics problem, which is beyond the scope of this paper. A very simple but effective practical arm kinematics solution takes advantage of the necessary body sagittal coordination (see the SKD model in Figure 31), and the right arm DOF is made equal or proportional to its complementary left leg DOF. Therefore, the values for the variables  $\theta_{25}$  to  $\theta_{29}$  are defined as (86).

$$\theta_{25} = 0; \quad \theta_{26} = \theta_{15}; \quad \theta_{27} = \theta_{14}; \quad \theta_{28} = \theta_{16}; \quad \theta_{29} = 0 \quad (86)$$

With these computations, the right manipulator inverse kinematics problem is solved in a geometric way, and what is more, we have not only one solution but the set of all possible solutions. For instance, the right leg has eight theoretical solutions, which are captured with the approach shown in this paper (87), if they exist.

$$Solutions = \theta_9 \text{Double} \times \theta_8 \theta_7 \text{Double} \times \theta_{11} \theta_{10} \text{Double} \times \theta_{12} \text{Single} = 8 \quad (87)$$

After repeating exactly the same technique for the left manipulator, the complete Rh-1 humanoid inverse kinematics problem is, in fact, totally resolved.

### 6.11 Simulation results

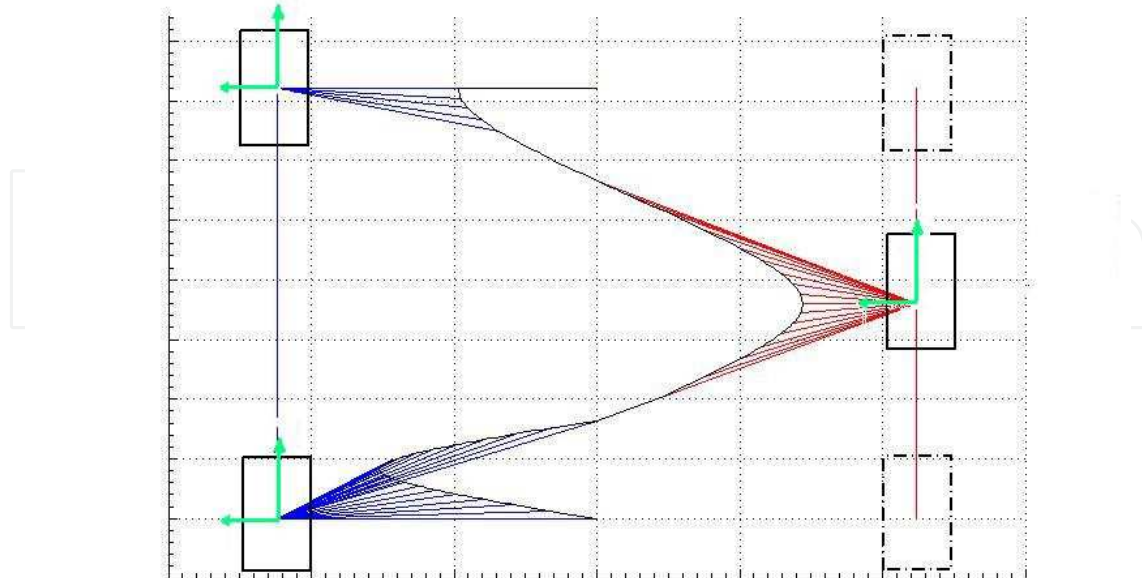


Fig. 32. *COG* hyperbolic trajectories in local axes (green)

For three steps, Figure 32 shows the spatial motion of the pendulum mass, and the local frame (green frames) of hyperbolic trajectories obtained in the single support phase; the trajectories shape looks like a hyperbolic curve as deduced above. It is a hyperbolic trajectory because the orbital energy in “*y-direction*” is negative (this is due to the fact that the pendulum frontal motion accelerates and decelerates without passing the equilibrium point, as shown in Figure 19), so the eq. 39 describes a hyperbole. The passive walkers have another walking principle, which is based on a limit cycle, when the gravity fields act on the device to achieve motion. In our case, we introduce the reference *COG* motion to make the robot walk, so we can pre-plane the stable walking pattern and introduce it to the humanoid robot. It is noted that the pendulum base is centered in the middle of the support foot and the natural ball pendulum motion follows a hyperbolic trajectory; the smooth pattern found drives the *COG* of the humanoid robot; natural and stable walking motion is obtained as will be demonstrated in several simulations and experimental results explained in the next paragraphs and sections. Figure 33 shows the temporal pendulum mass trajectories, in this walking pattern the single support phase takes 1.5 seconds and the double one 0.2 seconds. After computing the inverse kinematics at each local axis (Fig. 32), the joint patterns of the right humanoid leg and angular velocities obtained are shown in Figure 34. Those allow for checking the joint constraints in order to satisfy the actuator’s performances.

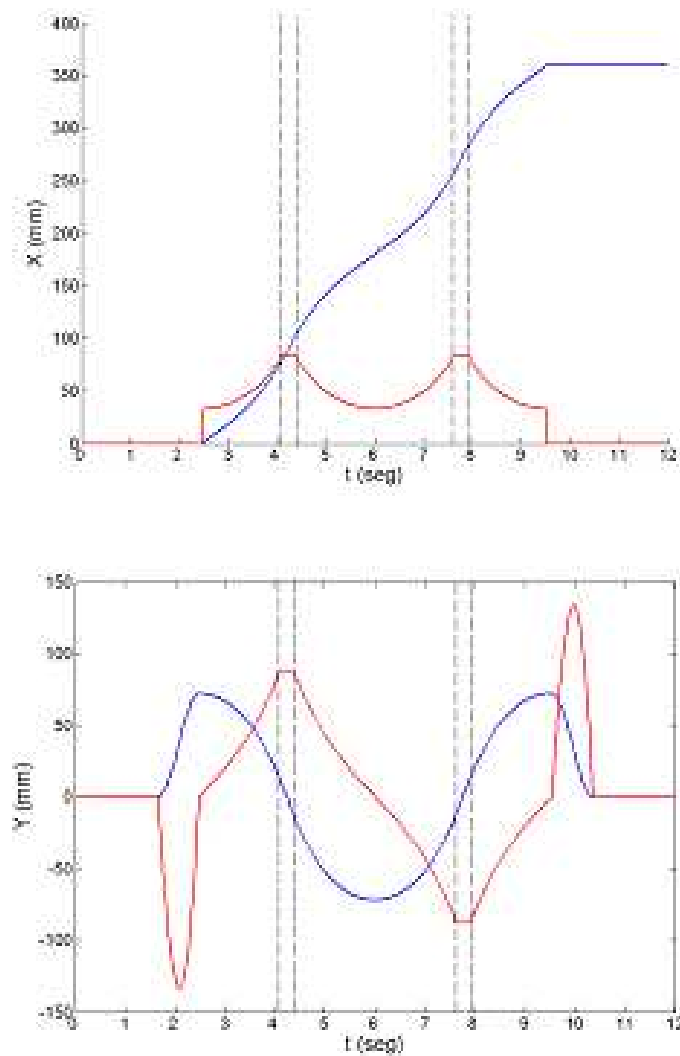


Fig. 33. *COG* temporal position (blue) and velocity (red) patterns for doing three steps. In the double support phase (between vertical dashed lines) constant speed maintains the trajectory's continuity.

Rh-1 simulator results are shown in Figure 35, from the VRML environment developed, which let us test the walking pattern previously so as to test it in the real humanoid robot. This environment lets us evaluate the angle motion range of each joint, avoid self-collision and obstacle collision, in order to obtain adequate walking patterns considering the robot's dimensions and mechanical limitations. It is verified by several simulation tests that smooth, fast and natural walking motion is obtained using the 3D-LIPM and foot motion patterns.



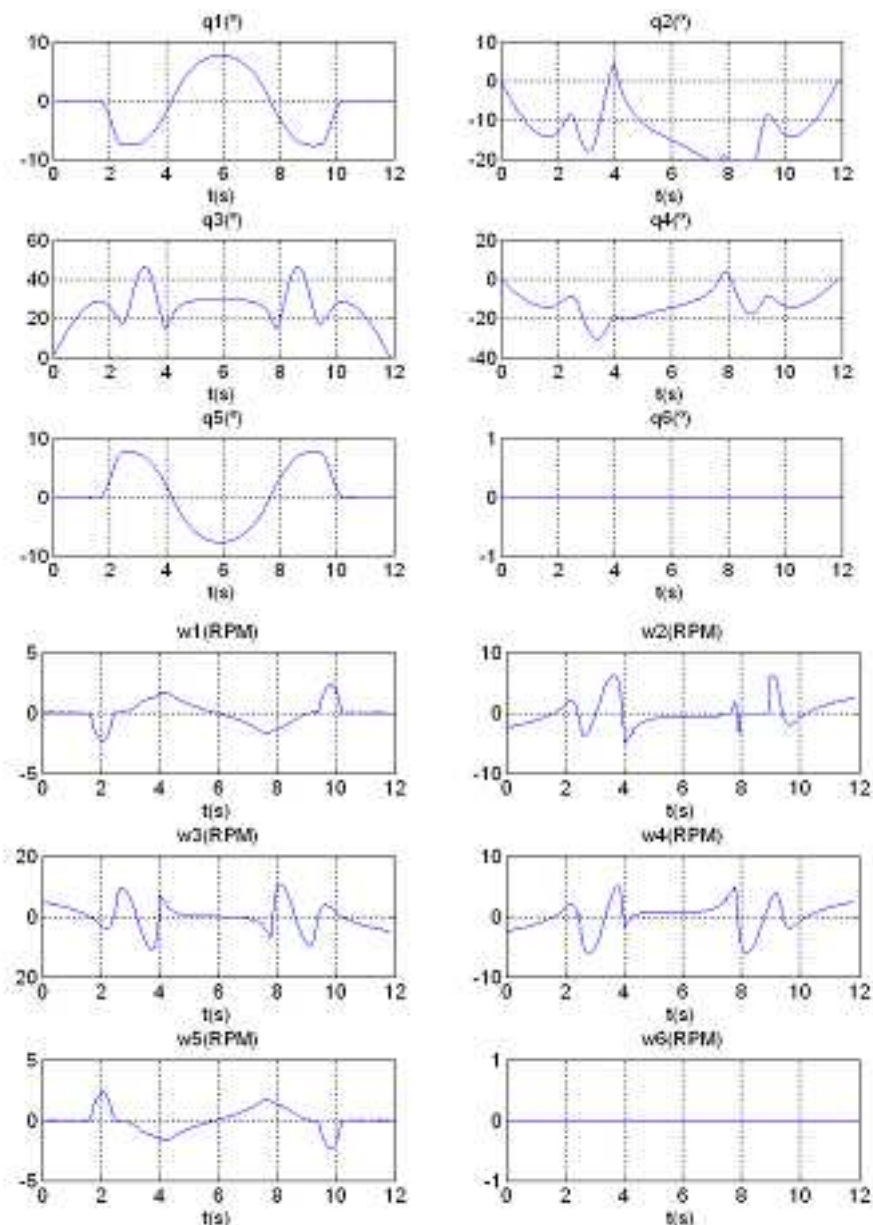


Fig. 34. Simulation of joint angle evolution and joint velocity evolution of right leg for three steps walking in a straight line (note that  $\theta_7$ , to  $\theta_{12}$  correspond to  $q_1$  to  $q_6$ )

In order to obtain global humanoid motion for avoiding an external obstacle (Yoshida et al. 2005), walking patterns in any direction could be developed using a rotation matrix around the  $z$ -axis of the local frame and the proper boundary conditions such as position and velocity at the local frame of the *COG*, and foot trajectories while changing the support foot in order to obtain smooth walking patterns. Figure 36 and Figure 37 shows us an example of planning walking motion with a change in direction. The walking pattern generated in each local frame maintains continuity with the previous and the next walking pattern. In addition, real-time walking pattern generation is possible, which changes direction, length and step width at any time using the information from sensors or by external command of the humanoid robot, according to the LAG algorithm.

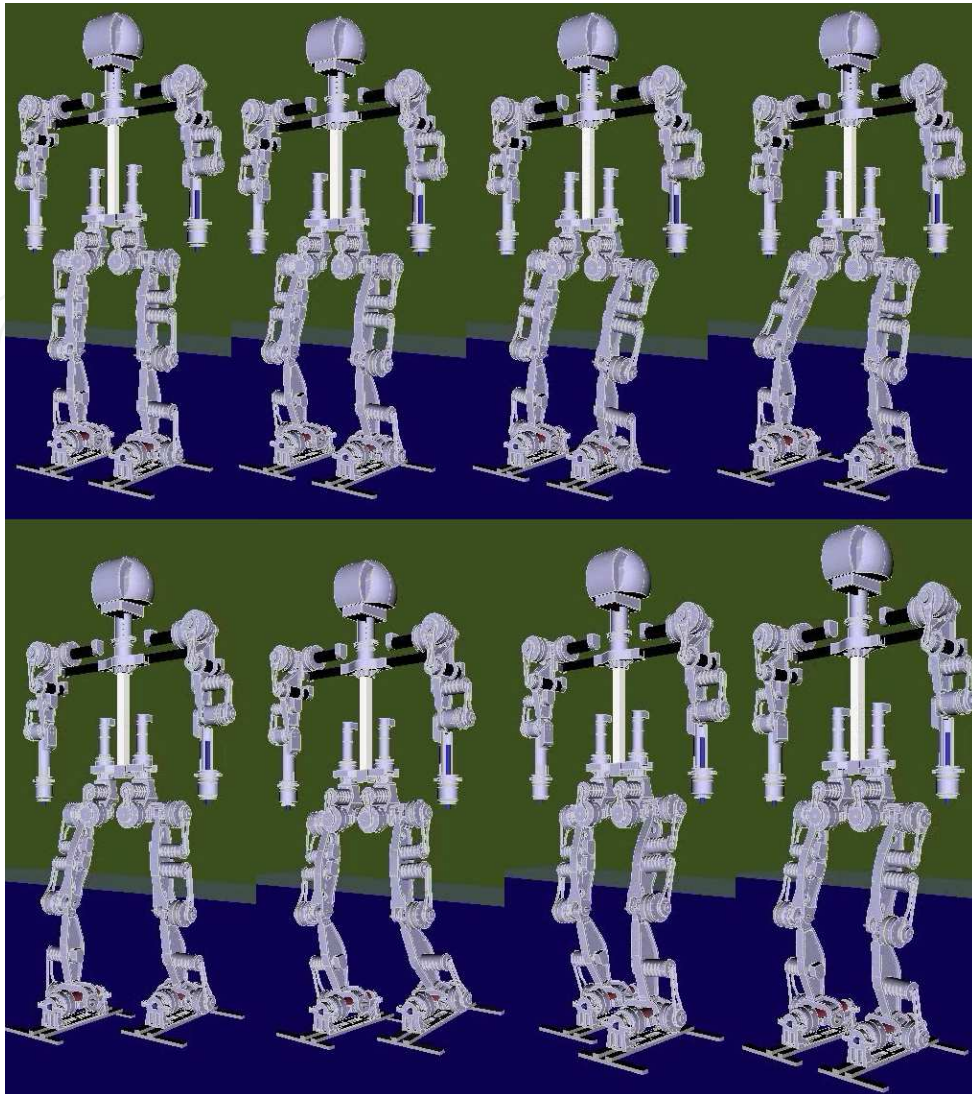


Fig. 35. Rh-1 simulator testing walking patterns.

In order to correct mechanical flexion and terrain irregularities, some joint patterns should be modified (i.e. ankle and hip joints), by offline and online control gait. The offline control reduces high joint acceleration at the beginning of the single support phase, because correction starts at this time. Online correction compensates for the actual environment changes and the mechanical imperfections. In this way, stable walking is obtained, by maintaining the body's orientation and ZMP in the right position.

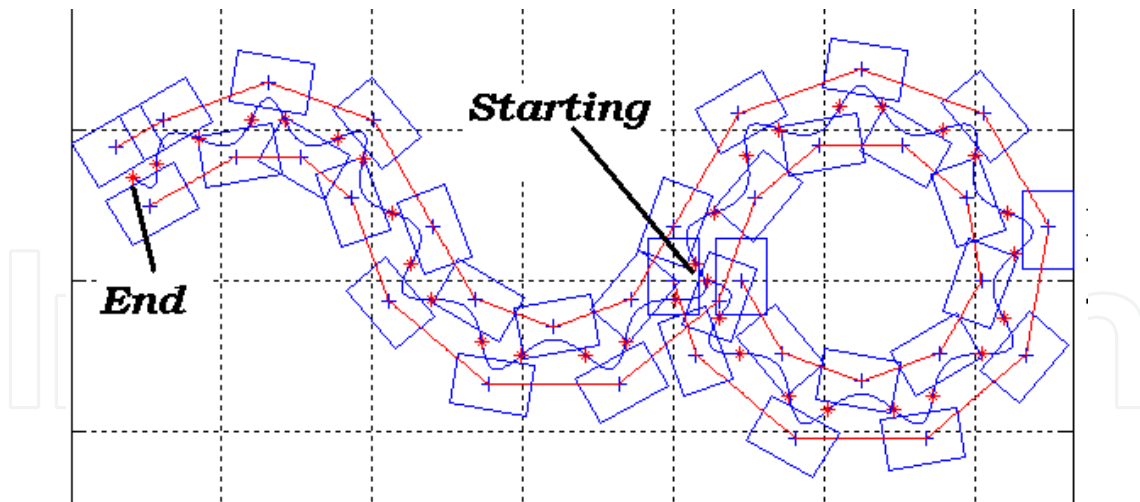


Fig. 36. Generating walking patterns in any direction.

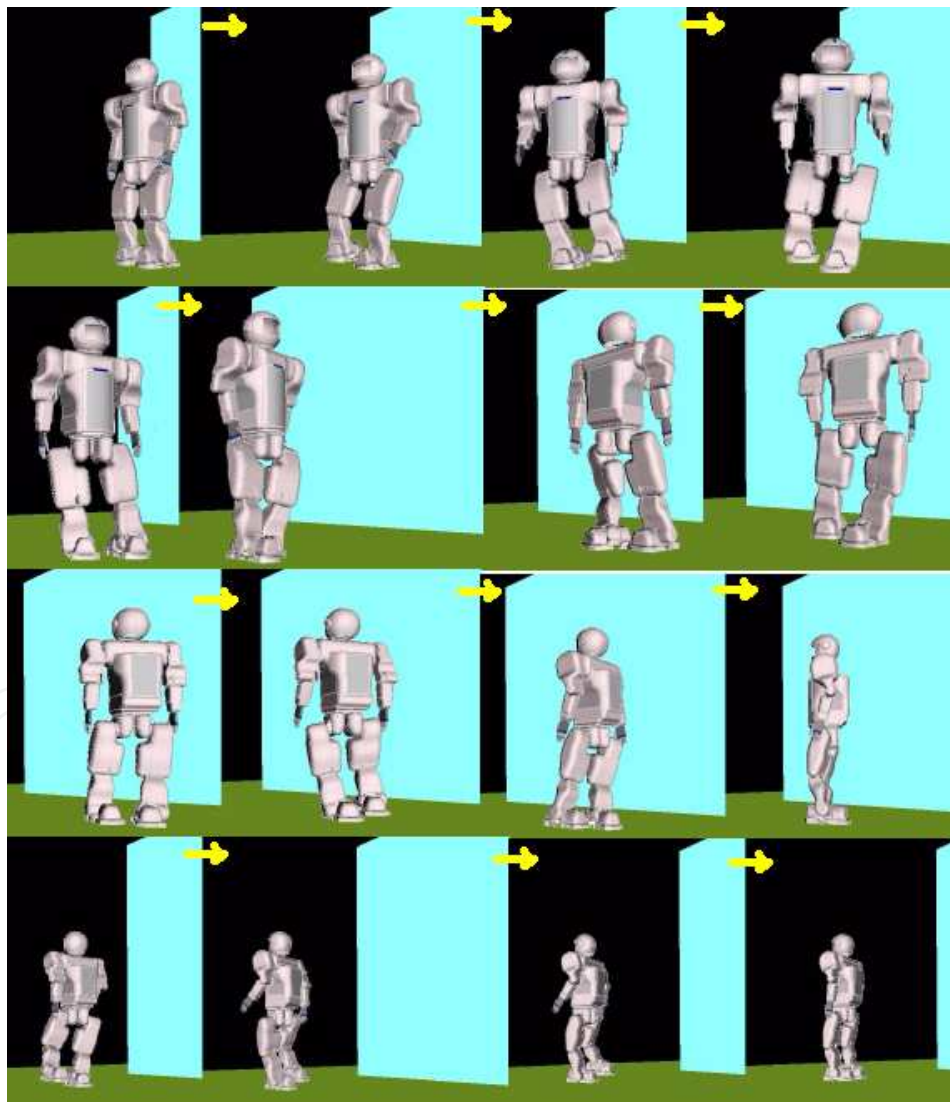


Fig. 37. Snapshots of walking patterns in any direction.

## 7. Control Architecture

### 7.1 Control overview

Humans usually walk using a specific walking pattern in normal environments. In cases of unexpected disturbances, this pattern can be changed immediately to another one in order to adapt to terrain requirements. The humanoid robot operates in the same space as the human and needs the same mechanism to adapt its walking to changing conditions.

Although planned motion patterns satisfy the stability constraints, some errors caused by the irregularity of the terrain or some external forces can cause the humanoid robot to fall down. Moreover, the flexion of the mechanical structure of the humanoid robot is the largest source of errors affecting its walking. To reduce the influence of these errors on walking stability, the control architecture (Fig. 38) for online motion patterns modification has been implemented.

The control architecture consists of two basic parts: Joint Control and Stabilization Control. The Joint Control is the core of the control scheme. When the offline calculated motion pattern is received and the motion is started, the Adaptive Control Algorithm adjusts the motion controller of each articulation to the diverse postures of the humanoid robot. If the sensorial system of the humanoid robot detects an error in the body position, Stabilization Control corrects it and then tries to recover each joint's trajectory in order to execute the previously calculated motion pattern.

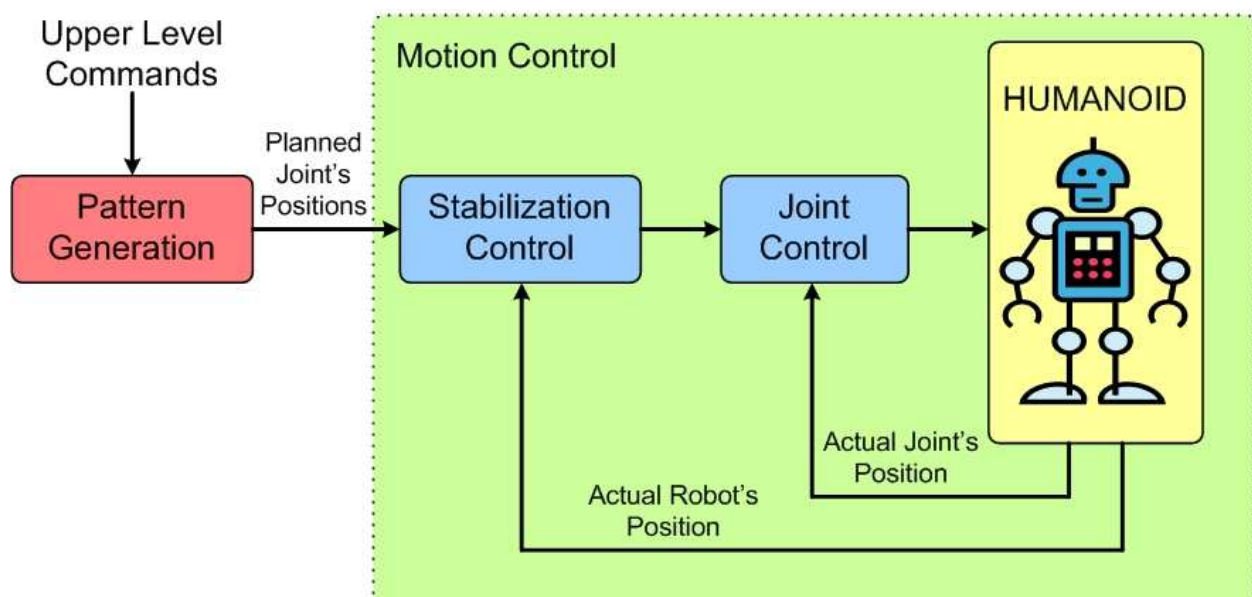


Fig. 38. Control architecture.

When the robot is working in the operational interaction mode (walking, object manipulation, etc.), there are several computing and communication tasks that need to be performed in a cyclical mode and be fast enough to avoid any possible loss of control. The periodic (with period  $T_s$ ) chain (Fig. 39) begins with the sensing task, taking the time interval  $t_{att}$  for attitude estimation gyros and accelerometers readings and  $t_{zmp}$  for ZMP force-torque sensor readings.



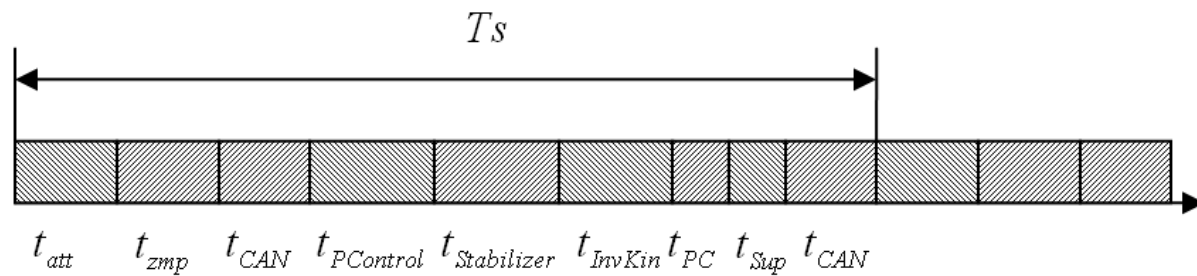


Fig. 39. Main computing and communication tasks for the Rh-1.

These tasks are followed by the tasks performing CAN bus communications, Posture Control and then Stabilizer and Inverse Kinematics Calculator computing, internal PC bus communications, Supervisory controller, and CAN bus transmission of new reference for each joint of the humanoid robot.

The period  $T_s$  should be small and compatible with the dynamics of the humanoid robot movement. On the one hand,  $T_s$  cannot be made arbitrarily small because the various computing and communication tasks with execution times  $t_{att}$ ,  $t_{zmp}$ ,  $t_{CAN}$ ,  $t_{PC}$ ,  $t_{Stabilizer}$ , etc., cannot themselves be made arbitrarily small. Also, a small value for  $T_s$  would generate too many messages in communication lines (PC bus, CAN bus) that would overload it. On the other hand,  $T_s$  cannot be made arbitrarily large because of the dynamics of the robot (Nyquist criterion).

Thus,

$$T_s > t_{att} + t_{zmp} + t_{CAN} + t_{PControl} + t_{Stabilizer} + t_{InvKin} + t_{PC} + t_{Sup} + t_{CAN} \quad (88)$$

And

$$T_s < \frac{1}{2 \cdot Fr} \quad (89)$$

Where  $Fr$  is the highest movement frequency on any robot link.

Assuming that the robot walks at about the same speed as a human walking normally,  $Fr = 2$  Hz.

Thus,

$$T_s < 250 \text{ ms} \quad (90)$$

This upper limit of the sample time is a strict real-time requirement considering the complexity of the computing and communication tasks to be performed within this time limit.

The following sections present the detailed design of Joint and Stabilization Controls.

## 7.2 Joint Control

Different postures of the humanoid robot radically change the motors' dynamics and require adaptive algorithms to improve the control of each joint. The main goal is to achieve the adequate systems response for the desired humanoid robot movements.

In the humanoid robot Rh-1, each motor is controlled by the motion controller. This motion controller includes the bottom part - three cascade control loops (Fig. 40): position loop (proportional), velocity loop (proportional - integral) and current loop (proportional - integral).

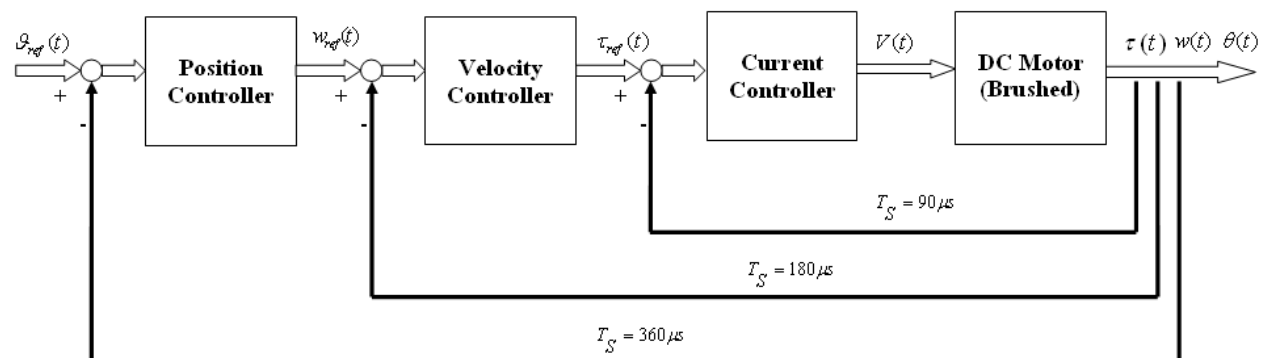


Fig. 40. Block diagram of the implemented motion controller.

This type of a control scheme has a reasonable advantage over a simple loop scheme. The internal, faster loop absorbs the major part of disturbances until these affect the more important external loop.

Moreover, the adaptive control algorithm is considered in order to maintain the real trajectory of each joint as similar as possible to the ideal one (motion pattern), obtained by solving the kinematics problem.

Equation (73) presents a classical dynamic model of a loaded DC motor which is usually used to move the joints of a humanoid robot,

$$\Omega(s) = \frac{Ks}{(D + s \cdot J)} \cdot I_a(s) - \frac{1}{(D + s \cdot J)} \cdot T_L(s) \quad (91)$$

where  $\Omega(s)$  is the velocity of rotor of the DC motor,  $I_a(s)$  - current,  $Ks$  - velocity constant,  $D$  - viscous friction or viscous damping inherent in the system,  $J$  - total moment of inertia of a joint, and  $T_L(s)$  - load torque. Evidently, when a humanoid robot moves, it assumes different postures characterized by different configurations of joints. This leads to continuous changing in the model of a DC motor. Thus, the adaptive algorithm, tuning the



controller of each joint when the model of a motor changes, is required. It carries out online parameter modification of the motion controller in compliance with the system's behaviors. The most suitable way to implement an adaptive control scheme is the gain-scheduling control. It does not need online identification. The offline software integrating identification and controller design algorithms estimate position and velocity controllers' gains for every possible configuration of the humanoid robot. The identification is made for real gaits (forward walking, backward walking and turning) in order to consider the influence of the posture of the robot on the system dynamics at every moment of the motion. In other words, every offline trajectory (gait) is divided into small parts. For each part, its own values of controller's gains are estimated and a multidimensional table of gains is created. Figure 41 shows the block diagram of the designed gain-scheduling motion controller.

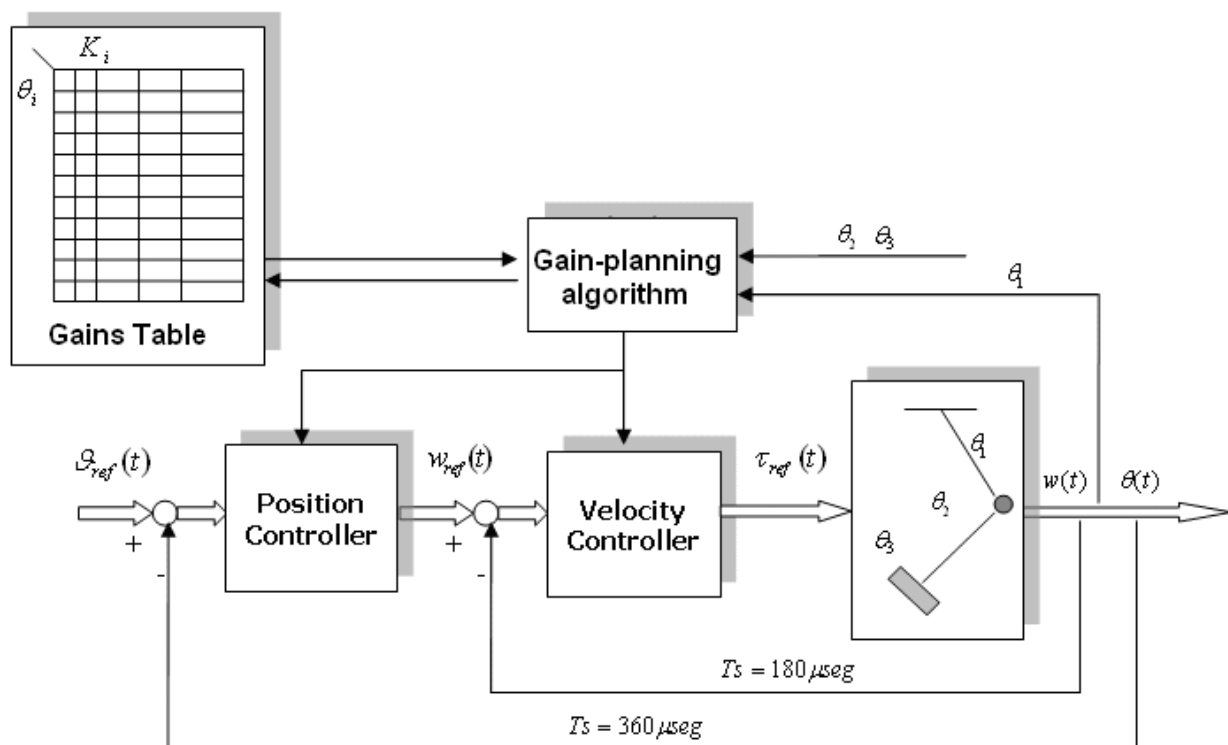


Fig. 41. Block diagram of the gain-scheduling motion controller.

The online part of the adaptive algorithm updates the controller's parameters following the offline calculated table. In general, it is necessary to choose adequate parameters according to the actual angular position of each joint and the posture of the whole robot.

This control approach has some advantages in comparison with the online adaptive control algorithms. It does not require a lot of online computational resources which is a critical restriction for implementing the controller for each of the 21 joints of the humanoid robot Rh-1.

### 7.3 Stabilization Control Joint Control

As was shown in the previous section, the humanoid robot's dynamics is governed by the ZMP equation 45. From these equations it can be concluded that there are two main variables to control:

$$e_{ZMP} = ZMP^d - ZMP^a \quad (92)$$

$$e_{COG} = COG^d - COG^a \quad (92)$$

where ZMP error is denoted as  $e_{ZMP}$  and COG position error is denoted as  $e_{COG}$ .  $ZMP^d$ ,  $COG^d$  are desired and  $ZMP^a$ ,  $COG^a$  are actual ZMP and COG positions. Equations (74) and (75) state that two different controls should be implemented in order to stabilize humanoid robot walking. Figure 42 presents the main structure of the stabilization controller for the humanoid robot.

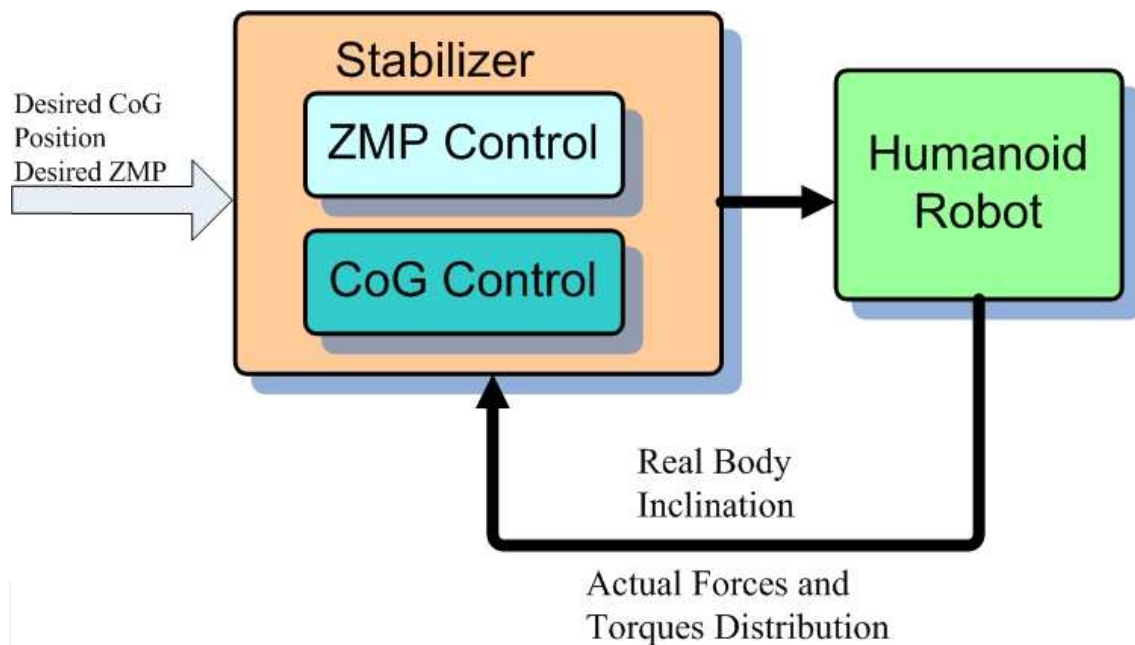


Fig. 42. Stabilizer structure.

Previously computed desired COG and ZMP positions generate joint trajectories which are sent for the humanoid motion. During the execution of motion patterns, the real ZMP and COG position should be measured. The Stabilizer compares it with the ideal ones and generates a new corrected motion pattern for the next moment. This control scheme provides a simple and effective way of controlling humanoid walking stability.

### 7.4 ZMP control

The implementation of the ZMP control should provide an easy and robust method to move the actual ZMP to the point where the target ZMP is located. The easiest way here is to neglect the influence of the part of the humanoid robot above its ankle joint. It can be made

by rotation of the upper part of the mechanism with respect to the ankle joint as shown in Figure 43.

Although the proposed compensational mechanism allows the actual ZMP to be moved to the point where the target ZMP is located, some problems appear. When the upper body is rotated with respect to the ankle joint, body angle errors in frontal  $\Delta q_{Bf}(t)$  and sagittal  $\Delta q_{Bs}(t)$  planes (Fig. 43) appear.

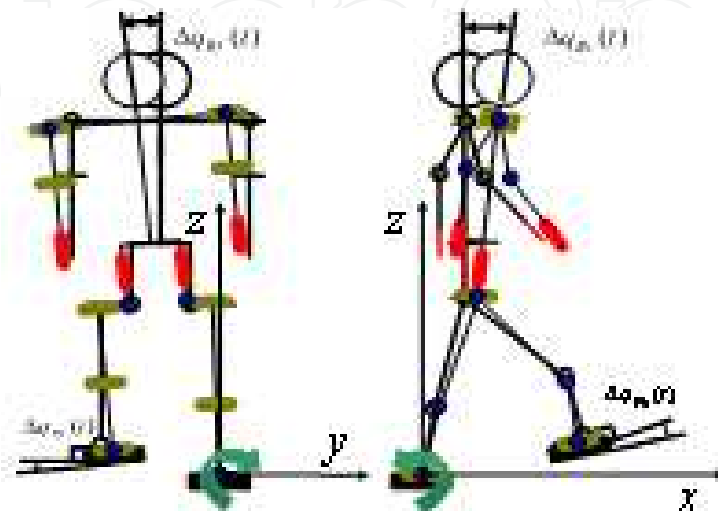


Fig. 43. ZMP error compensation.

These errors in frontal and sagittal planes cause the appearance of tilting torques and errors in the frontal  $\Delta q_{Ff}(t)$  and sagittal  $\Delta q_{Fs}(t)$  planes during the positioning of the hanging foot. It can overturn the robot or introduce very strong instability and vibrations when the foot is landing. Therefore, the mechanism for compensating for the body inclination errors in the frontal  $\Delta q_{Bf}(t)$  and sagittal  $\Delta q_{Bs}(t)$  planes should complement the ZMP control.

### 7.5 COG - Attitude control

Using a 3D-LIPM (or other method) for trajectory generation we assume that the humanoid's upper body does not have any influence on the dynamics of a model. In the real walking case it leads to unbounded errors in the COG position. It differs radically from the ideal position computed for a motion pattern. The tilting moment  $M_{CoG}$  compounded by  $M_x$  and  $M_y$  components denotes the robot's upper body dynamics (Fig. 44) and should be compensated by a control algorithm. The Attitude control is a special sort of a COG control which tries to maintain the trunk of the robot strictly vertical in every stage of its motion thus eliminating the tilting moment and body inclination errors.

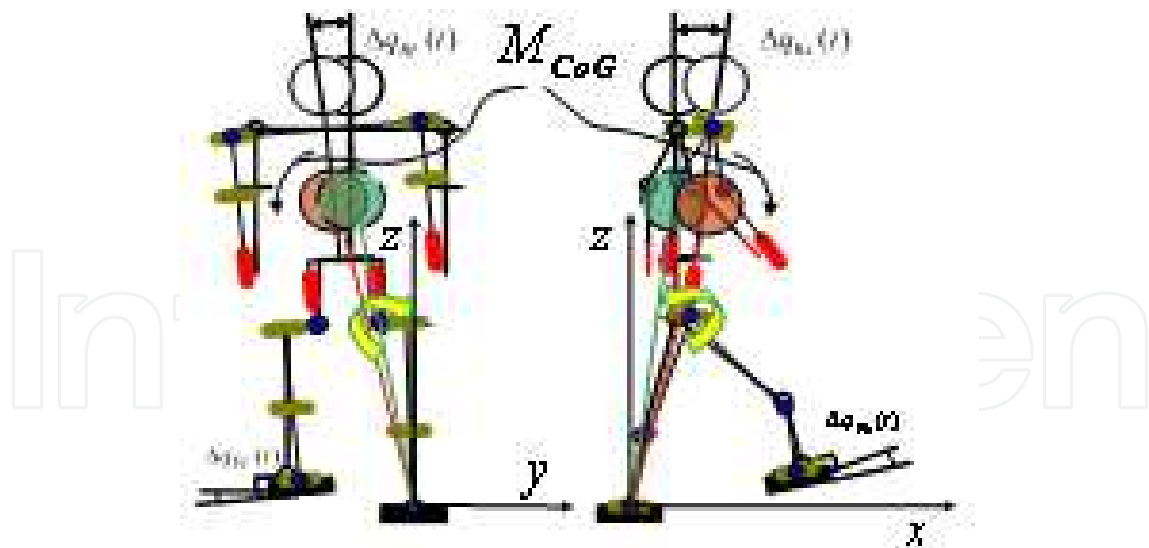


Fig. 44. COG error compensation – Attitude control.

The attitude of the robot is normally determined by a combination of gyroscopes and accelerometers placed in the waist, where the approximate center of the mass or the upper part of its backbone is located. As soon as the sensorial information is obtained, the most effective way to control the humanoid robot's body inclination is to maintain its backbone strictly vertical at every stage of the trajectory. In this case, it is enough to control only the hip joints in the frontal and sagittal planes

### 7.6 Double Inverted Pendulum

As was shown in the previous sections, to maintain the stability of a humanoid robot it is necessary to implement simultaneously both the ZMP and the Attitude controls acting on the ankle and hip joints. The humanoid robot in that case should be modeled as an inverted double pendulum shown in Figure 45(a).

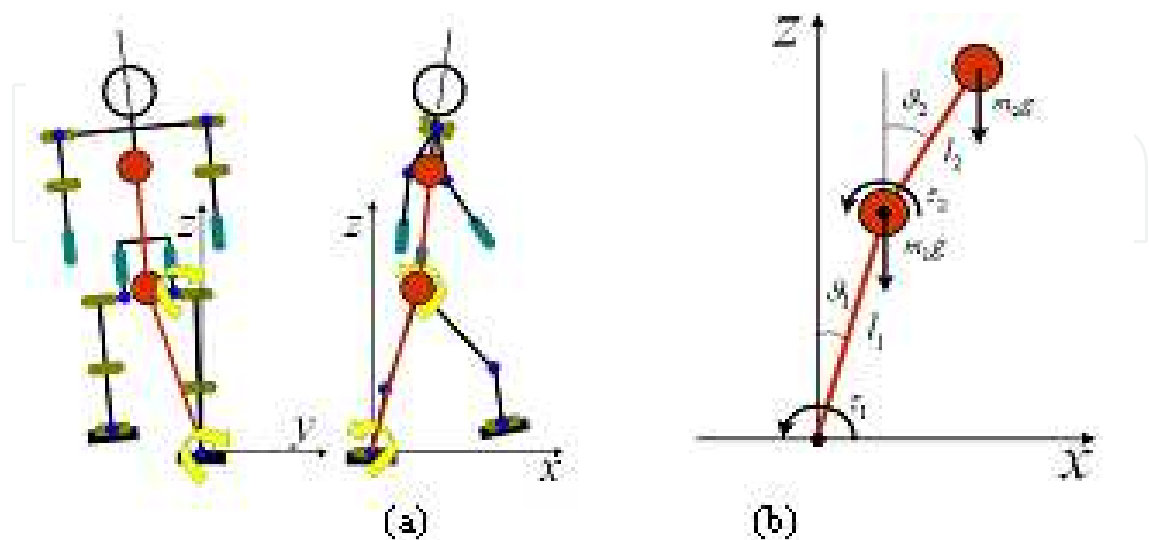


Fig. 45. a) Humanoid robot motion modelling. b) Double Inverted Pendulum.

A double pendulum consists of one pendulum attached to another. Consider a double bob pendulum with masses  $m_1$  and  $m_2$  ( $m_1 + m_2 = M$  - the total mass of the humanoid robot) attached by rigid mass less wires of lengths  $l_1$  and  $l_2$ . Furthermore, let the angles that the two wires make with the vertical be denoted  $\vartheta_1$  (ankle rotation) and  $\vartheta_2$  (hip rotation), as illustrated in Figure 45(b). The position of the center of mass of the two rods may be written in terms of these angles. The potential energy of the system is given by:

$$V = (m_1 + m_2)gl_1 \cos \vartheta_1 + m_2gl_2 \cos \vartheta_2 \quad (94)$$

The kinetic energy of the system is given by:

$$T = \frac{1}{2}m_1l_1^2\dot{\vartheta}_1^2 + \frac{1}{2}m_2[l_1^2\dot{\vartheta}_1^2 + l_2^2\dot{\vartheta}_2^2 + 2l_1l_2\dot{\vartheta}_1\dot{\vartheta}_2 \cos(\vartheta_1 - \vartheta_2)] \quad (95)$$

Writing the Lagrangian of a system using equations (94) and (95) for  $\vartheta_1$ ,  $\vartheta_2$  and simplifying we get:

$$(m_1 + m_2)l_1^2\ddot{\vartheta}_1 + m_2l_1l_2\ddot{\vartheta}_2 \cos(\vartheta_1 - \vartheta_2) + m_2l_1l_2\dot{\vartheta}_2^2 \sin(\vartheta_1 - \vartheta_2) - l_1g(m_1 + m_2)\sin \vartheta_1 = \tau_1 \quad (96)$$

$$m_2l_2^2\ddot{\vartheta}_2 + m_2l_1l_2\ddot{\vartheta}_1 \cos(\vartheta_1 - \vartheta_2) - m_2l_1l_2\dot{\vartheta}_1^2 \sin(\vartheta_1 - \vartheta_2) - l_2m_2g\sin \vartheta_2 = \tau_2 \quad (97)$$

The aim of the present research is to develop a simple control strategy allowing for the use of traditional controllers such as the linear quadratic regulator (LQR) or a simple PID controller to regulate the inverted pendulum about the upright equilibrium point. As the name may suggest, the LQR controller requires a linear system for which it will generate constant gains for full state feedback to make the equilibrium point globally asymptotically stable. However, the dynamics of double inverted pendulum systems are inherently nonlinear.

The approach chosen was to linearize the equations of motion regarding the operating point and define a domain of attraction within which the constant gain controller results in local asymptotic stability.

Take the approximation that perturbations in the problem are very small, and those terms of second order and higher are negligible. In this limit  $\cos \vartheta_i = 1$ ,  $\sin \vartheta_i = \vartheta_i$ ,  $\cos(\vartheta_1 - \vartheta_2) = 1$ .

Applying a small angle approximation we get the following linearization:

$$(m_1 + m_2)l_1^2\ddot{\vartheta}_1 + m_2l_1l_2\ddot{\vartheta}_2 - l_1g(m_1 + m_2)\vartheta_1 = \tau_1 \quad (98)$$

$$m_2l_2^2\ddot{\vartheta}_2 + m_2l_1l_2\ddot{\vartheta}_1 - l_2m_2g\vartheta_2 = \tau_2 \quad (99)$$

The state representation of the dynamical system in standard form is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g(m_1+m_2)}{m_1 l_1} & 0 & -\frac{m_2 g}{m_1 l_1} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{g(m_1+m_2)}{m_1 l_2} & 0 & \frac{g(m_1+m_2)}{m_1 l_2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{m_1 l_1^2} & -\frac{1}{m_1 l_1 l_2} \\ 0 & 0 \\ -\frac{1}{m_1 l_1 l_2} & \frac{(m_1+m_2)}{m_1 m_2 l_2^2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{100}$$

With the output equation:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{101}$$

where state variables  $x_1, x_2, x_3$  and  $x_4$  are defined as:  $x_1 = \vartheta_1, x_2 = \dot{\vartheta}_1, x_3 = \vartheta_2, x_4 = \dot{\vartheta}_2$ .

LQR controller design considers an optimal control problem that for given system equations (100) and (101) determines the matrix  $\mathbf{K}$  of the optimal control vector:

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \tag{102}$$

The block diagram showing the optimal controller configuration for the double inverted pendulum system is presented in Figure 46.

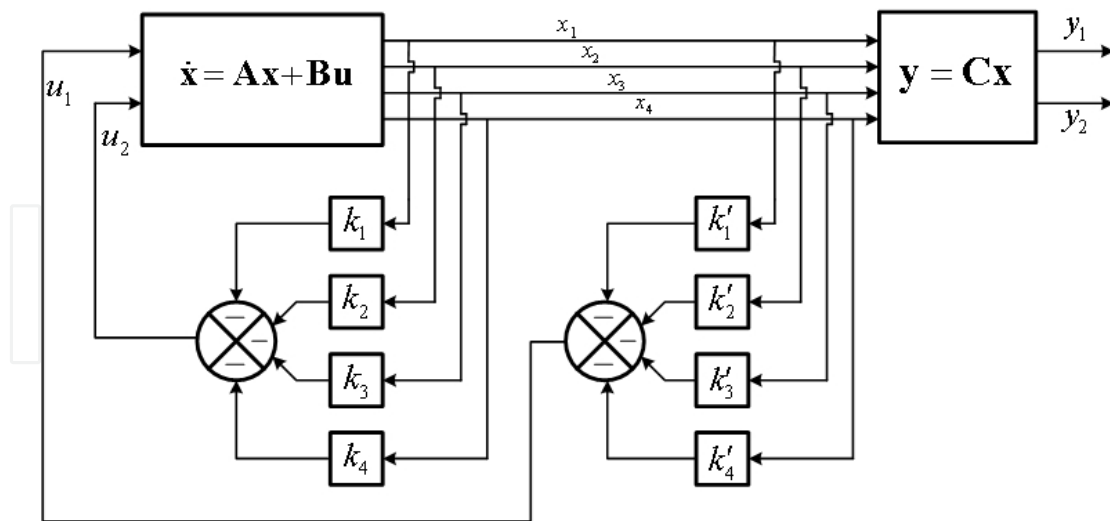


Fig. 46. Double inverted pendulum control system.

Note that the control system presented in Figure 46 is the regulator system. In this case the controller maintains desired angles  $\vartheta_1$  and  $\vartheta_2$  of the double pendulum close to zero.



### 7.7 Decoupled Control

Although the stabilization method for the humanoid robot based on the double inverted pendulum dynamics presented in the previous section seems to be the most appropriate for controlling the mechanism, in practice, it has many drawbacks. The main reason is that the double inverted pendulum considers control torques of both (hip and ankle) joints of the humanoid robot involved in the stabilization control. This means it is necessary use the torque controlled actuators to implement this control algorithm. Most contemporary humanoid robots are driven by DC motors with position control where torque control cannot be implemented. A decoupled approach considers that ZMP and Attitude can be implemented as totally independent and the influence of one on another will be negligibly small. To prove the appropriateness of this, let us examine the functioning of the already developed double inverted pendulum system in diverse initial positions of the pendulum (Figure 47), which correspond to different perturbations acting on the humanoid robot:

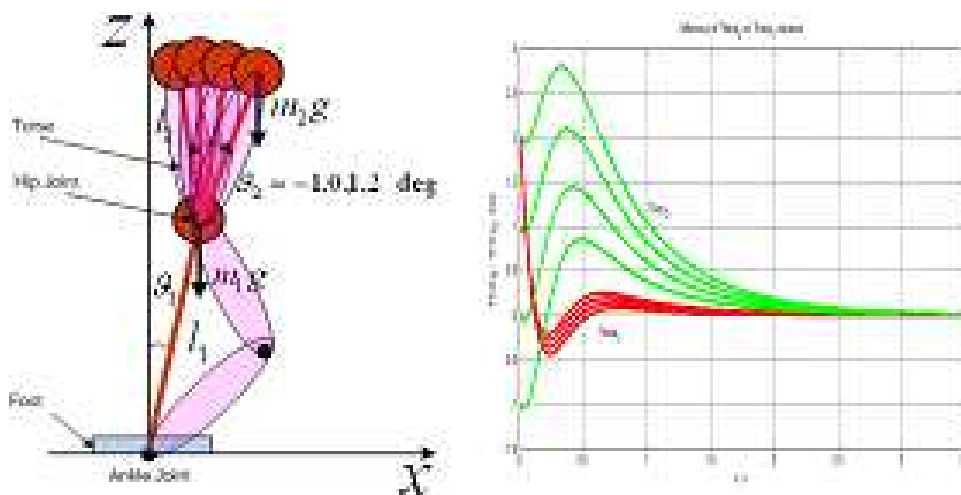


Fig. 47. a) Double Inverted Pendulum  $\theta_2$  variation b) Influence of the  $\theta_2$  on  $\theta_1$  variation.

From Figure 47(b) it can be observed that a large variation in  $\theta_2$  does not have much influence on  $\theta_1$ . Its variations stay practically at the same level in every experiment. Therefore, we take into consideration that the control of  $\theta_2$  will have more priority (or gain) and thus,  $\theta_2$  variation will be practically zero. On the other hand, variation of  $\theta_1$  sensed by  $\theta_2$  can be considered as an additional perturbation and will be compensated for by the controller. Moreover, it should be noted that the dynamics of the real robot will be attenuated by the reduction between its links. Thus, stabilizer control of the humanoid robot can be considered as a sum of two decoupled components related to Posture control ( $\theta_2$ ) and ZMP control ( $\theta_1$ ). In practice, decoupled ZMP and Attitude controllers were designed as LQR controllers.

After the foregoing detailed development of all the parts, we now turn to the detailed control architecture (Figure 48).

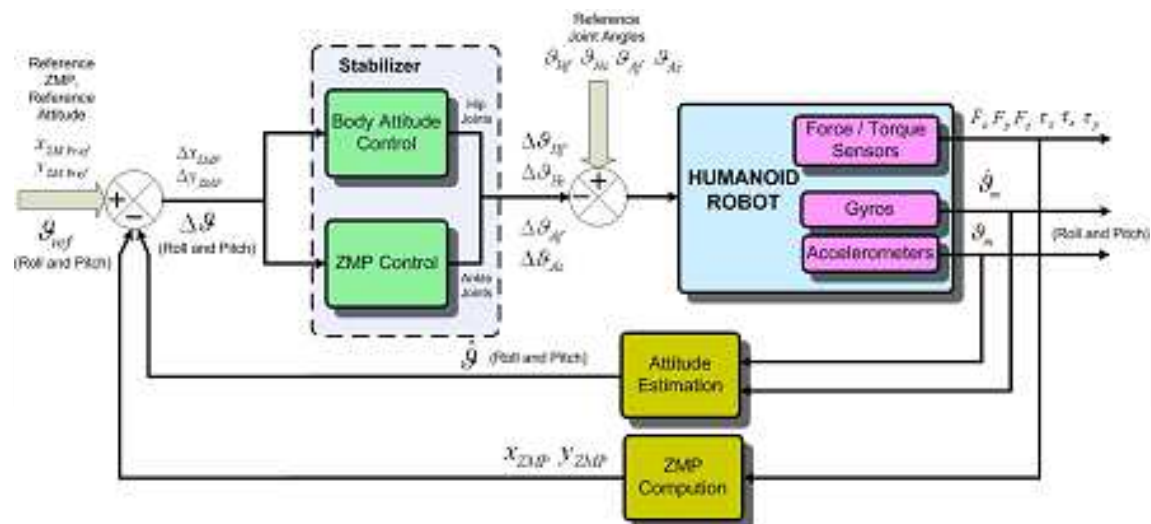


Fig. 48. Stabilizer architecture.

A sensorial system of the robot consisting of two six-axis force-torque should provide the controller with the real distribution of the forces and torques  $F_x$ ,  $F_y$ ,  $F_z$ ,  $\tau_x$ ,  $\tau_y$ ,  $\tau_z$  at the contact point of the foot with the ground. The 3-axis Gyro and Accelerometer provides the measurements of angular position  $\vartheta_m$  and angular velocity  $\dot{\vartheta}_m$  of the upper body (trunk) of the robot in the frontal and sagittal planes (Roll and Pitch), (Löffler et. Al., 2003 and Baerveldt et. al., 1997). After the actual ZMP position  $x_{ZMP}$ ,  $y_{ZMP}$  is computed by the ZMP Computational module and the real attitude is estimated in the Attitude Estimation module ZMP  $\Delta x_{ZMP}$ ,  $\Delta y_{ZMP}$  and Attitude  $\Delta \vartheta$ , errors can be estimated. These errors are the input data for the Stabilizer. The Stabilizer is designed as a decoupled controller. It controls error in ZMP and Attitude positioning of the humanoid robot by the motion of the ankle and hip joints as was discussed above. Finally, the compensational motion of the ankle  $\Delta \vartheta_{Af}$ ,  $\Delta \vartheta_{As}$  and hip  $\Delta \vartheta_{Hf}$ ,  $\Delta \vartheta_{Hs}$  joints in the frontal and sagittal planes should be superimposed with their reference trajectories  $\vartheta_{Af}$ ,  $\vartheta_{As}$ ,  $\vartheta_{Hf}$ ,  $\vartheta_{Hs}$  which are responsible for the entire walking process. By doing this, new motion patterns for ankle and hip joints are generated. The implementation of the decoupled stabilizer provides fast and easy control of the walking stability of the humanoid. All changes are applied to the ankle and hip joints eliminating the need for inverse kinematics computation.

## 8. Experimental Results

Once the design, construction and assembly stages were carried out, the robot's experimental results followed.

Experimental results applying the natural 3D-LIPM and Cart-table models with Body Posture Control algorithms on the Rh-1 humanoid robot platform are discussed in this section. At the beginning, snapshots of dynamic walking of the Rh-1 humanoid robot are shown in Figure 49. As shown in the VRML (Figure 35) environment, the same walking motion pattern is followed.

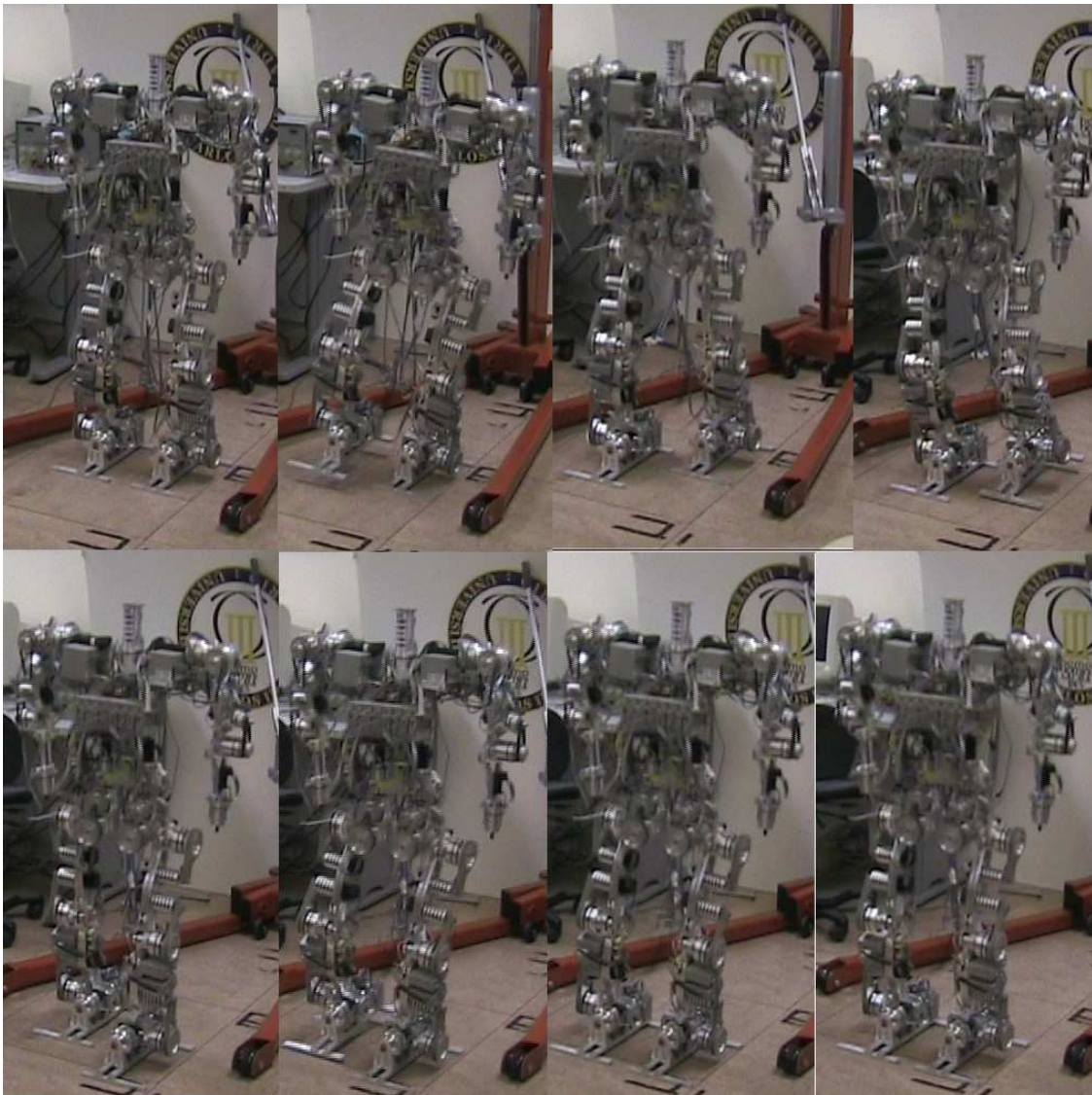


Fig. 49. Snapshots of the actual Rh-1 humanoid robot walking with dynamic gait.

The gait patterns proposed above allow for stable walking at 1 Km/h. Smooth and natural walking motions are obtained by introducing the adequate initial conditions on COG body motion and correct boundary conditions in foot planning. Some results are shown in Figure 50, where the frontal ankle joint's evolution can be seen in the chart, i.e.  $q_1$  is the frontal right ankle joint, the blue line is the pattern obtained from the inverse kinematics model using the COG and foot patterns as inputs, for doing three steps; the red line is the on-line compensated reference pattern which allows for frontal humanoid body reorientation. This compensation is necessary because the mechanical elasticity tilts the humanoid into an unstable zone. On the other hand,  $w_1$  is the frontal joint angular velocity of the right ankle, the blue line is the velocity pattern obtained without compensation, the red one is the compensated velocity; adequate interpolators have been developed in order to maintain the humanoid body in the stable zone. Compensated curves are delayed with respect to reference ones, because there exists a compensation time which starts in the single support phase. The  $q_{12}$  and  $w_{12}$  patterns correspond to the left ankle frontal joint. In the sagittal motion, a similar compensation had been made.

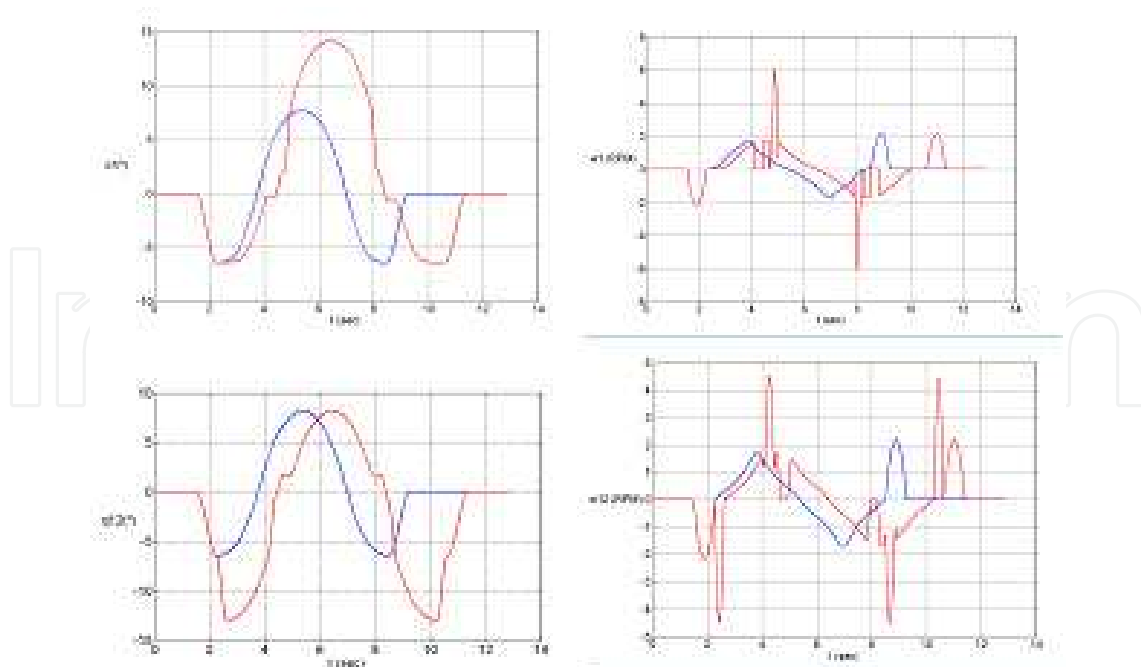


Fig. 50. The ankle joint's frontal motion (angular and velocity evolution), original evolution (blue line), compensated evolution (red line).

This compensation maintains the COG following pendulum laws, and thus, natural and stable walking is obtained. Posture compensation allows for a stable gait because the natural 3D-LIPM spatial trajectories are followed. Offline and online posture control compensates for several things: the gravitational effect on the humanoid structure in the sagittal and frontal motion planes, which tilts the humanoid to an unstable position; terrain irregularities and errors in the concentrated mass model that we have used for generating walking patterns. The offline compensated and real measured joint pattern references for both legs are shown in Figure 51 and Figure 52; furthermore the joints' current consumptions are presented in Figure 53, in order to check the actuators current limits. Next, snapshots following the Cart-table model are shown in Figure 54. Successful results of straight walking motion in the laboratory and hall have been obtained. The step length is about 180mm, max feet height 40mm and step time about 1.25 sec. The joint patterns and angular velocities of the right leg results are shown in Figure 55 and Figure 56. The reference, offline corrected and real measure patterns of each joint for taking three steps forward are overlapped. The main differences are overlapped between the offline compensation and measured joint evolution is because of the variations, friction on the surface and mechanical imperfections of the robot.

We test that the control system allows for a stable walking motion in a straight line. Further improvements on the mechanical structure, walking pattern generation and control (i.e. foot landing with compliance control, for reducing the impact forces) in order to compensate for whole body moments, will induce a smooth and natural walking motion.



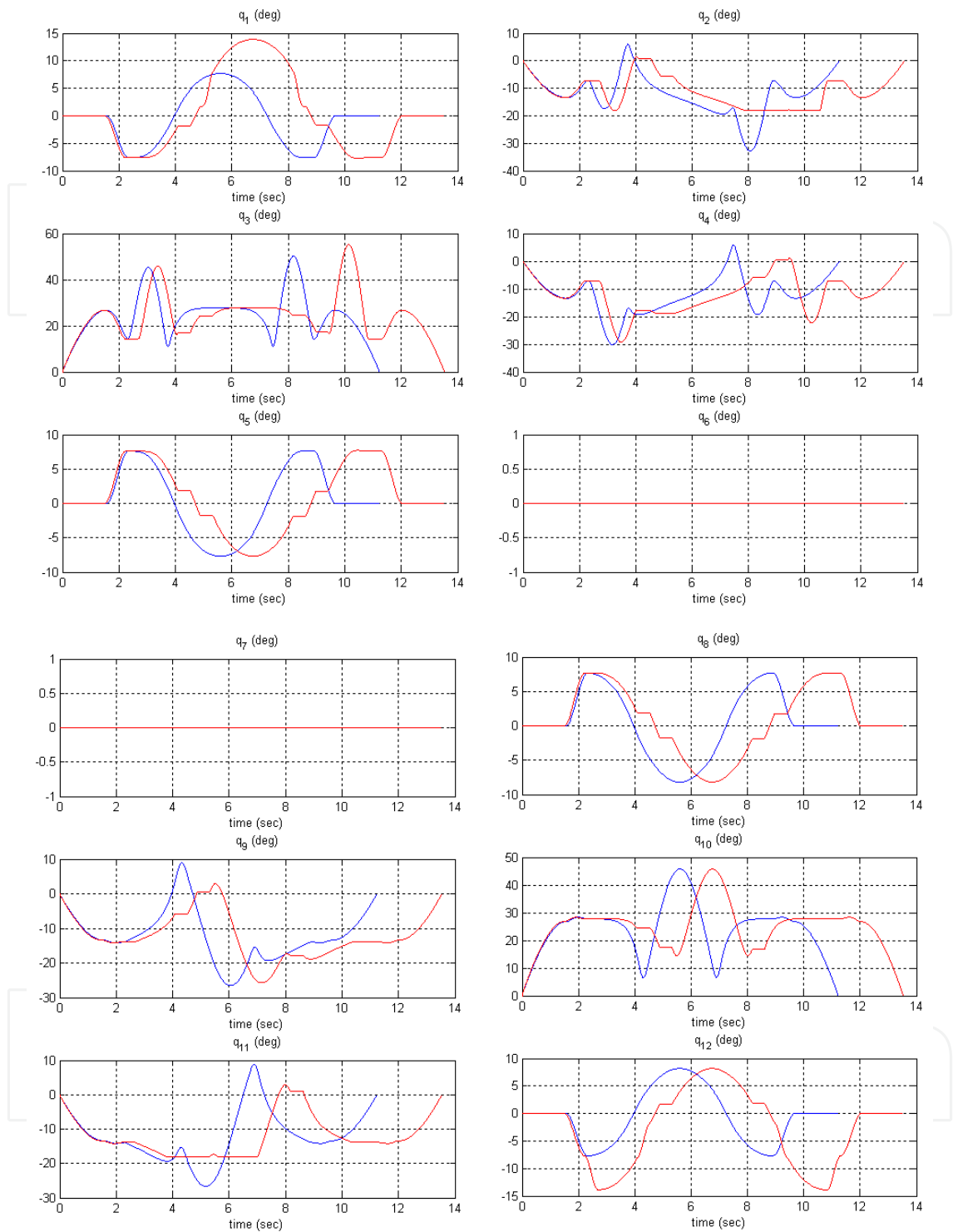


Fig. 51. Both leg joint patterns. Reference (blue) and offline corrected (red) overlapped.

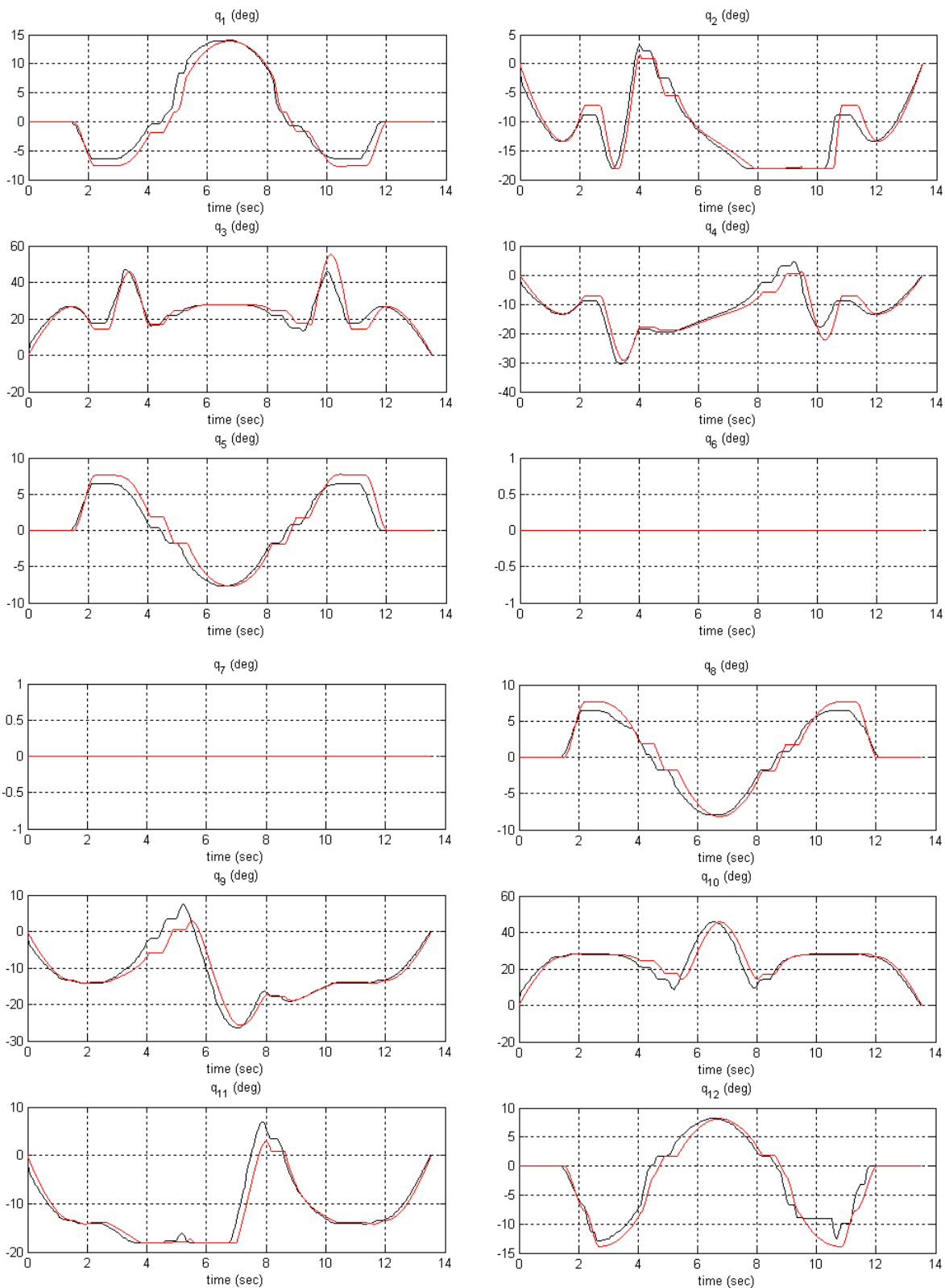


Fig. 52. Both leg joint patterns. a) Reference (blue) and offline corrected (red) overlapped; b) Offline corrected and measured (black) overlapped.



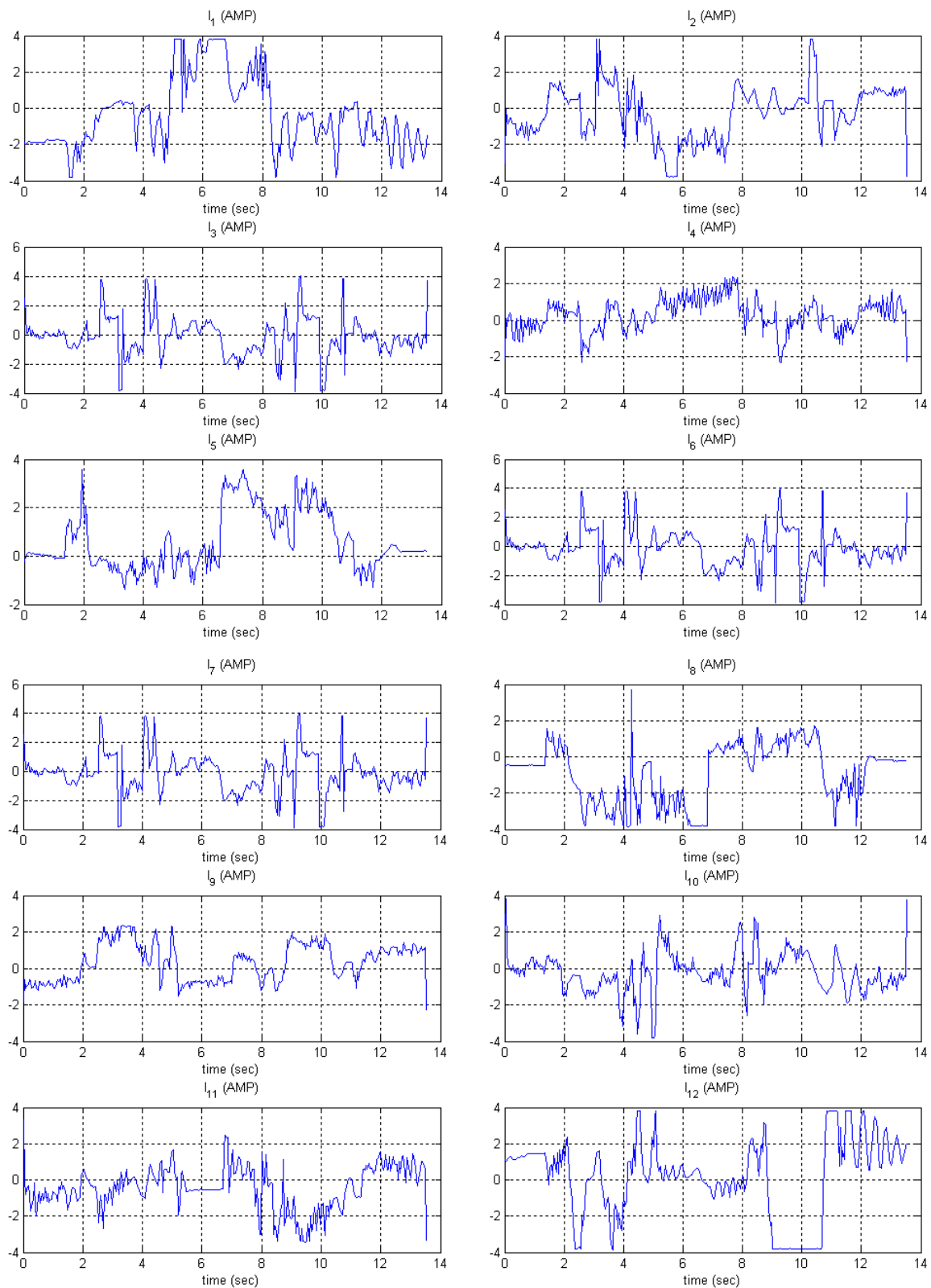


Fig. 53. Both legs' current consumption.

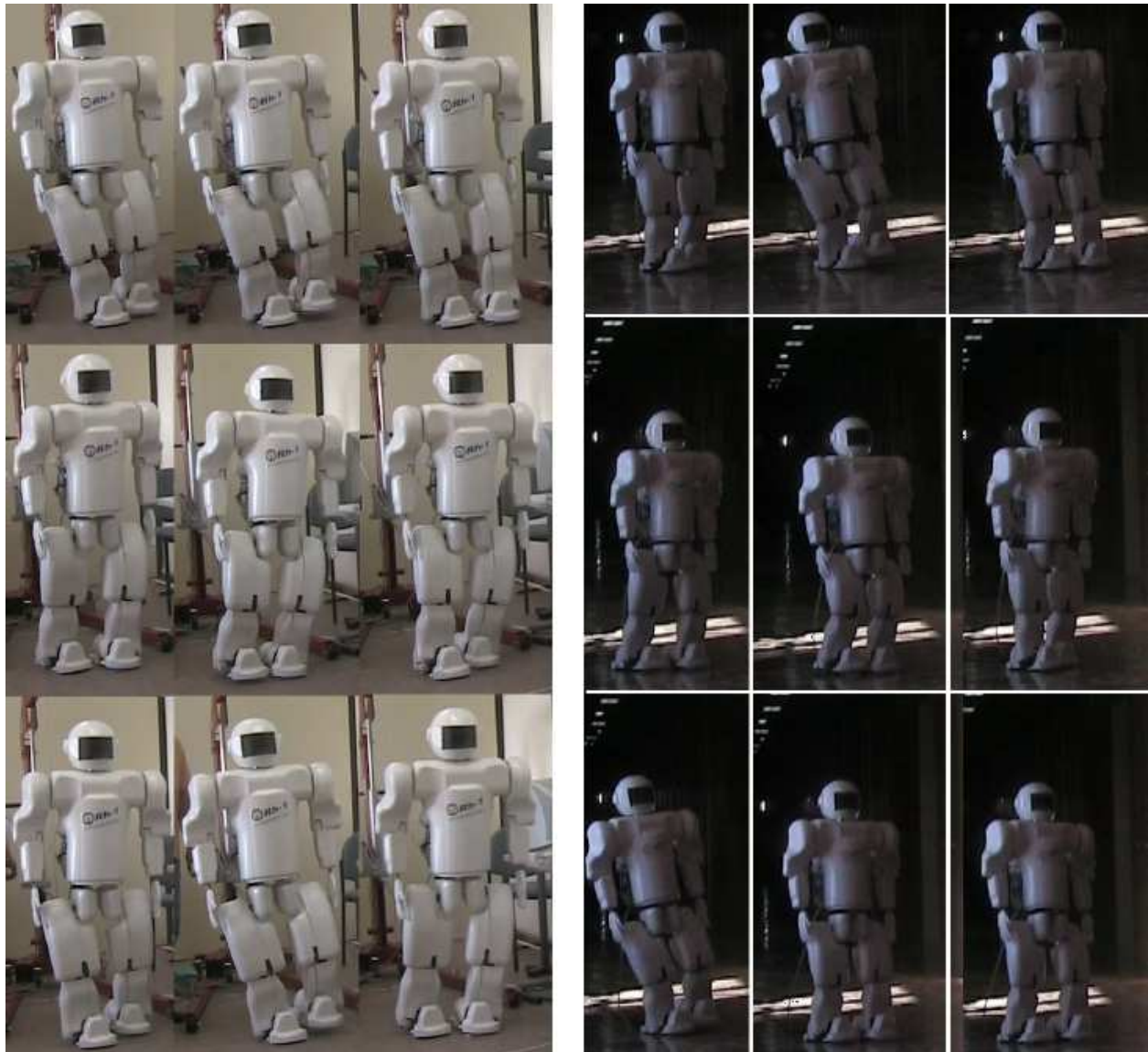


Fig. 54. Rh-1 snapshots walking forward.

## 9. Conclusions

Normal bipedal gait is achieved through a complex combination of automatic and volitional postural components. Normal walking requires stability to provide antigravity support of body weight, mobility of body segments and motor control to sequence multiple segments while transferring body weight from one limb to another. The result is energy-efficient forward progression. The human “gait cycle” has been analyzed in order to understand biped walking motion in its main phases, single support and double support phases and their properties: force reaction, cycle time, foot, knee, hip and body motion trajectories. In this way humanoid robot trajectories can be created on the order of human ones. It is demonstrated that the COG human motion follows the inverted pendulum laws at normal walking velocity, which is an important fact for maintaining stability while walking. Concerning the facts previously explained, it is possible to state that very satisfactory results were obtained, thus being a starting point for innumerable investigations in the future.

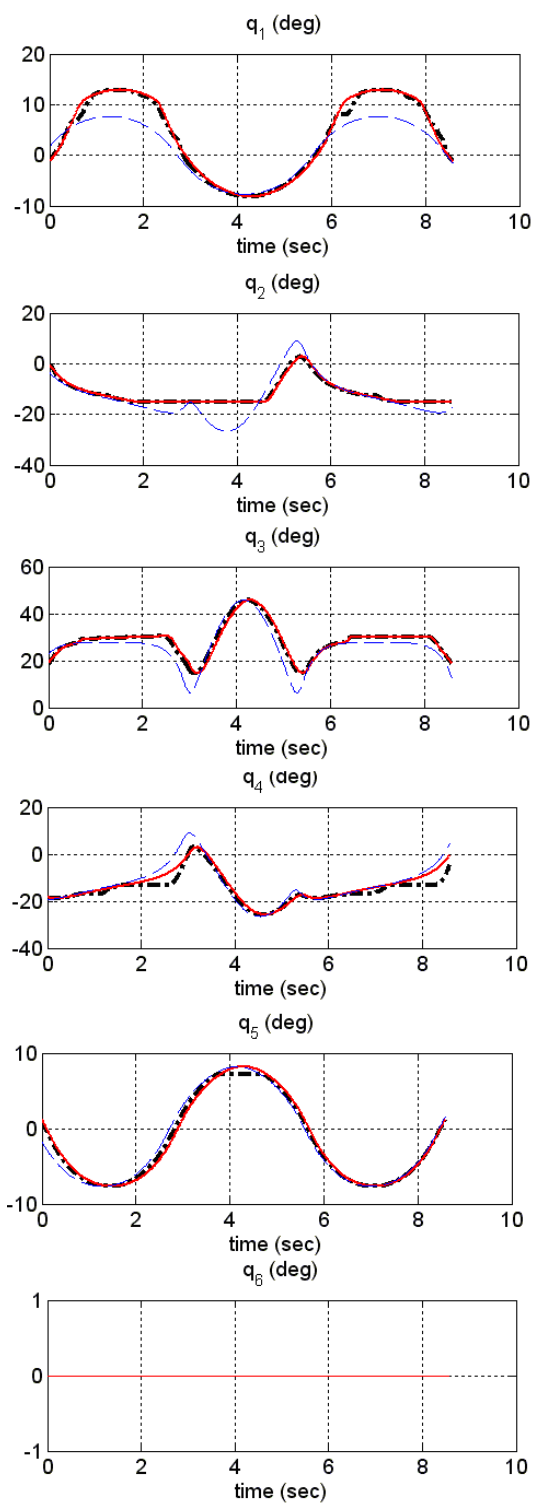


Fig. 55. Real joint angular evolution (dot dashed black line), offline compensated (continuous red line) and reference (blue dashed line). Right leg.

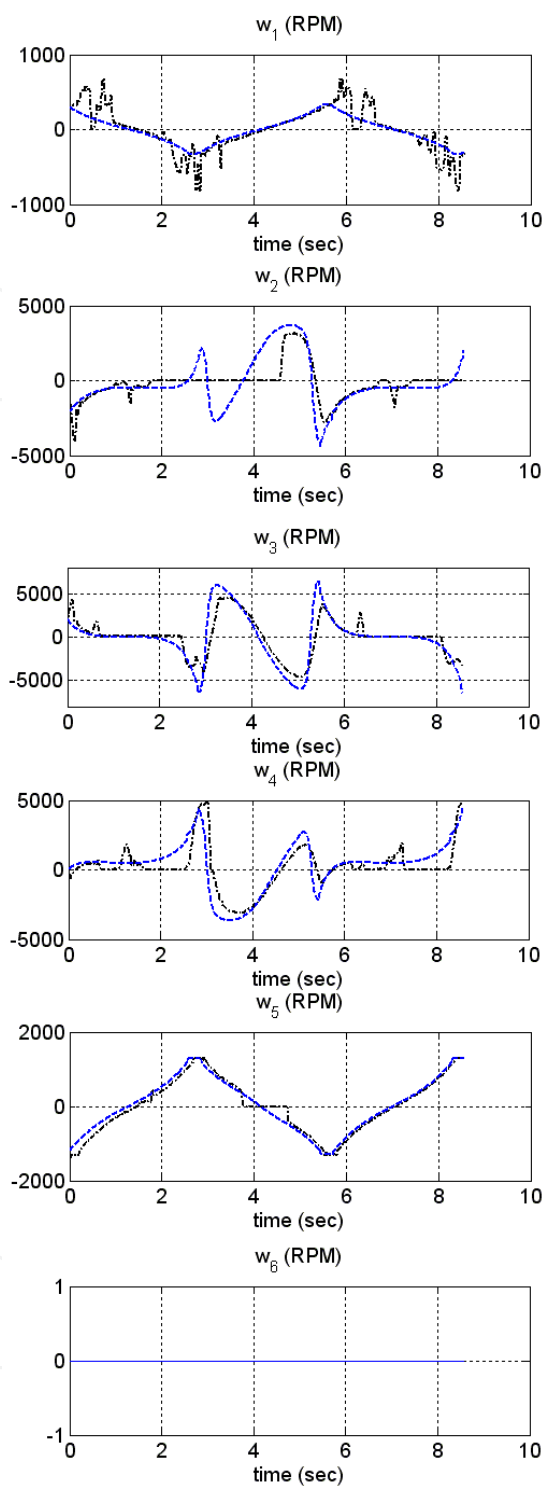


Fig. 56. Real (dot dashed black line) and reference (blue dashed line) motor angular velocity evolution. Right leg.

At the moment, many improvements and corrections are to be done to the mechanical parts. Due to the great amount of elements working together, some unwanted clearances and movements in the mechanical structure of the robot may appear. Furthermore, the robot is in its second evaluation stage and the number of elements that make up the robot must be decreased, either by redesigning the most critical ones or by fusing several of them into one. Compliance foot improvements will be implemented in order to reduce the efforts on each joint and overall structure.

Considering the hardware and software architecture of the Rh-1 robot, we must point out that this work makes an effort to show that there is a possibility of bringing some basic aspects of industrial automation and control to the other, more sophisticated fields of robotics, in order to extend further standardization and unification of the design processes. Moreover, the proposed approach allows for consideration of humanoid robot locomotion inside the global automation problem.

Dynamic walking was successfully implemented in the Rh-1 humanoid robot. It can walk smoothly and about twenty times faster than when using a static walking pattern, as was studied in previous works.

The SKD humanoid model makes it easy to solve the inverse kinematics problem using Lie groups math techniques, such as the POE. For bipedal locomotion, 3D-LIPM and Cart-table models of the COG motion can be computed in real time and be dynamically stable. The algorithms introduced have closed-form solutions with clear geometric meaning, and therefore can be useful for developing robust real-time applications.

It was demonstrated that offline compensation of the body orientation contributes to online control, reducing high joint accelerations. As a result, a stable motion was obtained.

The Stabilizer was designed as a decoupled controller. It controls the error in ZMP and Attitude positioning of the humanoid robot by the motion of the ankle and hip joints.

The humanoid robot Rh-1 provided with the developed control architecture is able to walk stably on a plain surface and to absorb some external disturbances. Future work will be focused on adding other elements to the proposed control architecture such as a foot landing control in order to correct for structural and walking surface imperfections, and to reduce the mechanical landing impact on the humanoid structure, which are the essential conditions for achieving stable humanoid robot walking on irregular terrain. Also further improvements on existing mechanical, hardware and software architecture will be continued.

## 10. References

- K. Hirai, M. Hirose, Y. Hikawa and T. Takanaka, The development of Honda humanoid robot, IEEE International Conference on Robotics and Automation ICRA 1998) Leuven (Belgium)

- K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota and T. Isozumi, "Design of prototype humanoid robotics platform for HRP", Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp. 2431-2436, 2002.
- J.M. Pardos; C.Balaguer, Rh-0 Humanoid Robot Bipedal Locomotion and Navigation Using Lie Groups and Geometric Algorithm. International Conference on Intelligent Robots and Systems (IROS'2005). Edmonton. Canada. Aug, 2005
- M. Arbulú, J.M. Pardos, L.M. Cabas, P. Staroverov, D. Kaynov, C. Pérez, M.A. Rodríguez; C. Balaguer, Rh-0 humanoid full size robot's control strategy based on the Lie logic technique, IEEE-RAS International Conference on Humanoid Robots (Humanoids'2005). Tsukuba. Japan. Dec, 2005
- S. Stramigioli, B. Mashke, C. Bidard, On the geometry of rigid body motions: the relation between Lie groups and screws, Journal of Mechanical Engineering Science, Vol. 216, n. C1, pp 13-23, 2002.
- M. Arbulú, F. Prieto, L. Cabas, P. Staroverov, D. Kaynov, C. Balaguer, ZMP Human Measure System. 8th International Conference on Climbing and Walking Robots (Clawar'2005). London. United Kingdom. Sep, 2005
- J. Yamaguchi, E. Soga, S. Inoue A. and Takanishi, Development of a bipedal humanoid robot control method of whole body cooperative dynamic bipedal walking, IEEE International Conference on Robotics and Automation (ICRA' 1999), Detroit, (USA)
- S. Kajita, F. Kaneiro, K. Kaneko, K. Fujiwara, K. Yokoi. and H. Hirukawa, Biped walking pattern generation by a simple 3D inverted pendulum model, Autonomous Robots, vol 17, n<sup>a</sup>2, 2003
- M.H. Raibert, Legged robots that balance, MIT Press:Cambridge, 1986
- M. Arbulú; L.M. Cabas; P. Staroverov; D. Kaynov; C. Pérez; C. Balaguer. On-line walking patterns generation for Rh-1 Humanoid Robot using a simple three-dimensional inverted pendulum model. 9th Internacional Conference on Climbing and Walking Robots (Clawar 2006). Brussels. Belgium. Sep, 2006.
- C.L. Shin, Y.Z. Li, S.Churng, T.T. Lee and W.A. Cruver. Trajectory Synthesis and Physical Admissibility for a Biped Robot During the Single-Support Phase, Proc. of IEEE International Conference on Robotics and Automation, pp. 1646-1652, 1990
- M. Vukobratovic, D. Juricic. Contribution to the Synthesis of Biped Gait. IEEE Tran. On Bio-Medical Engineering, Vol. 16, No. 1, pp. 1-6, 1969
- J. Furusho and A. Sano, Sensor-Based Control of a Nine-Link Biped, Int. J. on Robotics Research, Vol.. 9, No. 2, pp. 83-98, 1990
- Y. Fujimoto, S. Obata and A. Kawamura. Robust Biped Walking with Active Interaction Control between Foot and Ground, Proc. of IEEE International Conference on Robotics and Automation, pp. 2030-2035, 1998
- J. H. Park and H. C. Cho. An On-line Trajectory Modifier for the Base Link of Biped Robots to Enhance Locomotion Stability, Proc. of the IEEE ICRA2000, pp. 3353-3358, 2000.
- Q. Huang; K. Kaneko; K. Yokoi; S. Kajita; T. Kotoku; N. Koyachi; H. Arai; N. Imamura; K. Komoriya; K. Tanie. Balance Control of a Biped Robot Combining Off-line Pattern with Real-time Modification, Proc. of IEEE International Conference on Robotics and Automation, 2000.
- L. Cabas, S. de Torre, I. Prieto, M. Arbulu, C. Balaguer, Development of the lightweight human size humanoid robot RH-0. CLAWAR 2004, Madrid September 2004.



- L.M. Cabas; R. Cabas; P. Staroverov; M. Arbulú; D. Kaynov; C. Pérez; C. Balaguer. Challenges in the design of the humanoid robot RH-1. 9th Internacional Conference on Climbing and Walking Robots (Clawar 2006). Brussels. Belgium. Sep, 2006.
- A. Bicchi, G. Tonietti, and R. Schiavi. Safe and Fast Actuators for Machines Interacting with Humans. In Proc. of the 1st Technical Exhibition Based Conference on Robotics and Automation, TExCRA2004, November 18-19, TEPIA, Tokyo, Japan, 2004.
- L.M. Cabas; R. Cabas; P. Staroverov; M. Arbulú; D. Kaynov; C. Pérez; C. Balaguer. Mechanical Calculations on a Humanoid Robot. 9th Internacional Conference on Climbing and Walking Robots (Clawar 2006). Brussels. Belgium. Sep, 2006.
- K. Regenstein and Rudiger Dillmann, Design of an open hardware architecture for the humanoid robot ARMAR, Proc. of IEEE Int. Conference on Humanoid Robots, 2003.
- D. Kaynov; M.A. Rodríguez; M. Arbulú; P. Staroverov; L.M. Cabas; C. Balaguer. Advanced motion control system for the humanoid robot Rh-0. 8th International Conference on Climbing and Walking Robots (Clawar 2005), 2005.
- D. Kaynov, C. Balaguer. Industrial automation based approach to design control system of the humanoid robot. Application to the Rh-1 humanoid robot. Accepted for IEEE International Symposium on Industrial Electronics (ISIE2007)
- E. Yoshida, I. Belousov, C. Esteves and J. P. Laumond. Humanoid Motion Planning for Dynamic Tasks, Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids 2005), pp. 1-6, 2005.
- Löffler, M. Gienger and F. Pfeifer. Sensors and Control Design of a Dynamically Stable Biped Robot, Proc. of IEEE Int. Conference on Robotics and Automation, pp. 484-490, 2003
- M. Gienger, K. Löffler, and F. Pfeifer, "Towards the design of biped jogging robot", Proc. of IEEE Int. Conference on Robotics and Automation, pp. 4140-4145, 2001.
- A.-J. Baerveldt, R. Klang. A low cost and Low-weight Attitude Estimation System for an Autonomous Helicopter. Proc. of IEEE International Conference on Intelligent Engineering Systems, pp. 391-391, 1997.
- H. Hirukawa, S. Hattori, S. Kajita, K. Harada, K. Kaneko, F. Kanehiro, M. Morisawa, and S. Nakaoka, A pattern generator of humanoid robots walking on a rough terrain, in IEEE International Conference on Robotics and Automation, Roma and Italy, April 10-14 2007, pp. 2781- 2187.
- S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, Biped walking pattern generation by using preview control of zero-moment point, in IEEE International Conference on Robotics Automation, Taipei and Taiwan, September 14-19 2003, pp. 162-1626.
- M. Arbulu and C. Balaguer, Real-time gait planning for Rh-1 humanoid robot, using local axis gait algorithm, in 7th IEEE-RAS International Conference on Humanoid Robots, Pittsburgh, USA, Nov. 29-Dec. 2 2007.
- F.C. Park, J.E. Bobrow, and S.R. Ploen, "A Lie group formulation of robot dynamics," Int. J. Robotics Research. Vol. 14, No. 6, pp. 609-618, 1995.
- R.A. Abraham, and J.E. Marsden, Foundations of Mechanics. Perseus Publishing, 1999.
- B. Paden. Kinematics and Control Robot Manipulators. PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1986.



- S. Torre; L.M. Cabas; M. Arbulú; C. Balaguer. Inverse Dynamics of Humanoid Robot by Balanced Mass Distribution Method. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2004). Sendai. Japan. Sep, 2004.
- M. Arbulu and C. Balaguer, Real-time gait planning for Rh-1 humanoid robot, using local axis gait algorithm, in International Journal of Humanoid Robotics. Print ISSN: 0219-8436. Online ISSN: 1793-6942. Vol. 6. No. 1. pp.71-91. 2009
- R. M. Murray, Z. Li, and S. S. Sastry. Mathematical Introduction To Robotic Manipulation. CRC Press, 1994.
- E. Ayyappa. Normal human locomotion, part 1: Basic concepts and terminology. Journal of Prosthetics and Orthotics, pages 10-17, 1997.
- D. A. Winter. Biomechanics And Motor Control of Human Movement. A Wiley-Interscience Publication, 1990.
- K. Loeffler, M. Gienger, F. Pfeiffer, and H. Ulbrich. Sensors and control concept of a biped robot. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 51:1-9, 2004.
- S. Lohmeier, T. Buschmann, H. Ulbrich, F. Pfeiffer: Modular joint design for performance enhanced humanoid robot LOLA. In: Proc. IEEE Int. Conf. Rob. Aut. (ICRA), pp. 88-93 (2006)
- K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, K. Akachi, Humanoid Robot HRP-3, IEEE-RAS International Conference on Humanoid Robots (Humanoids'2008). Nice. France.

IntechOpen



## **Climbing and Walking Robots**

Edited by Behnam Miripour

ISBN 978-953-307-030-8

Hard cover, 508 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

Nowadays robotics is one of the most dynamic fields of scientific researches. The shift of robotics researches from manufacturing to services applications is clear. During the last decades interest in studying climbing and walking robots has been increased. This increasing interest has been in many areas that most important ones of them are: mechanics, electronics, medical engineering, cybernetics, controls, and computers. Today's climbing and walking robots are a combination of manipulative, perceptive, communicative, and cognitive abilities and they are capable of performing many tasks in industrial and non- industrial environments. Surveillance, planetary exploration, emergence rescue operations, reconnaissance, petrochemical applications, construction, entertainment, personal services, intervention in severe environments, transportation, medical and etc are some applications from a very diverse application fields of climbing and walking robots. By great progress in this area of robotics it is anticipated that next generation climbing and walking robots will enhance lives and will change the way the human works, thinks and makes decisions. This book presents the state of the art achievements, recent developments, applications and future challenges of climbing and walking robots. These are presented in 24 chapters by authors throughout the world. The book serves as a reference especially for the researchers who are interested in mobile robots. It also is useful for industrial engineers and graduate students in advanced study.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mario Arbulu, Dmitry Kaynov and Carlos Balaguer (2010). The Rh-1 Full-Size Humanoid Robot: Control System Design and Walking Pattern Generation, Climbing and Walking Robots, Behnam Miripour (Ed.), ISBN: 978-953-307-030-8, InTech, Available from: <http://www.intechopen.com/books/climbing-and-walking-robots/the-rh-1-full-size-humanoid-robot-control-system-design-and-walking-pattern-generation>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

[www.intechopen.com](http://www.intechopen.com)

[www.intechopen.com](http://www.intechopen.com)

IntechOpen

IntechOpen

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen