Design and Low-Level Control of a Humanoid Robot Using a Distributed Architecture Approach

Vítor M. F. Santos

Department of Mechanical Engineering, TEMA, University of Aveiro, 3810-193 Aveiro, Portugal

Filipe M. T. Silva[1]

Department of Electronics, Telecommunications and Informatics, IEETA, University of Aveiro, 3810-193 Aveiro, Portugal

Abstract: ***This paper describes the methods and strategies to develop a humanoid robot using a distributed architecture approach where centralized and local control co-exist and concur to provide robust full monitoring and efficient control on a complex system with 22 DOF. A description of the hardware is given before introducing the architecture since that influences greatly the methods implemented for the control and helps understanding the general decisions. The platform is still in improvement, but the results are very promising, mainly because many approaches and research issues suddenly opened and will provide opportunities to test distributed control systems with possibilities that go far beyond the classical control of robots. Some practical issues on servomotor control are also given since that turned out necessary before entering higher levels of control. That is addressed in the last part the paper which shows an example being developed to demonstrate the possibility of keeping a humanoid robot in upright balance position only by local control after reaction forces on the ground.***

Keywords: *Humanoid robots; Biped locomotion; Modular architectures; Distributed control.*

[1] *Corresponding author. Tel.: +351-234-370531; fax: +351-234-370545; E-mail: fsilva@det.ua.pt*

1. Introduction

The field of humanoid robotics has been attracting the attention of a growing community, both from the industry and academia. On the one hand, several companies have unveiled walking robots with impressive designs and skills, as the well-known Honda's ASIMO (Sakagami, Y., *et al.*, 2002) and Sony's QRIO (Nagasaka, K., *et al.*, 2004). On the other hand, the continuous progress in robotics technology opens up new possibilities for academic research on low-cost and easy-to-design humanoids, such as PINO (Yamasaki, F., *et al.*, 2000), ESYS (Furuta, T., *et al.*, 2001) and HanSaRam (Kim, J.-H., *et al.*, 2004), and others. Some platforms, however, show up limitations due to centralized control approaches and lack of modularity, making them difficult for others to replicate.

The main scope of the project beneath this paper has been the development of a humanoid platform to carry out research on control, navigation and perception. In particular, this paper focuses on the design and implementation of a distributed architecture for a humanoid robot where centralized and local controls co-exist and concur to provide a robust and versatile operation. The paper begins by presenting the design concepts and the technological solutions to build a small-size humanoid robot at reduced costs using off-the-shelf technologies, but still aiming at a fully autonomous platform for research. Afterwards, the software development and the integration techniques in building the proposed control architecture are described.

One most relevant feature of this implementation is the distributed architecture, supported on a CAN bus, in which independent and self-contained control units may allow either a cooperative or a standalone operation. The integration in these simpler control units of sensing, processing and acting capabilities play a key role to allow for localized control based on feedback from several sensors, ranging from joint position monitoring to force sensors. Moreover, the reprogrammable modules conduce to the central question of a true autonomy, i.e., the ability of self-control in which the robot may evolve over time. At the same time, the main advantage of the modular system is the possibility of reusing specific modules, in terms of both hardware and software, with no major efforts. Comparing with other architectures (Albero, M., *et al.*, 2005; Tomokuni, N., *et al.*, 2005), even other based on CAN bus (Cho, Y.-J., *et al.*, 1999), stands out the high versatility of implementation and the easy expansibility at the system's level and reusable hardware, as described further.

2. The Humanoid Platform

2.1. Mechanical design

The platform has 22 degrees of freedom with 12 of them dedicated to the legs, which represent the most challenging part both for designing and control. Structure is made essentially of aluminum and steel for axles and other small components, weights about 6 kg and is about 60 cm tall. Fig. 1 shows a CAD model and a current stage of development.

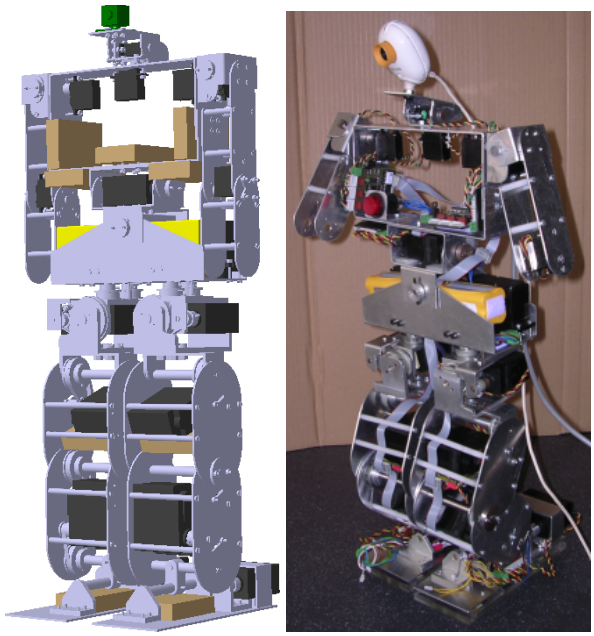


Fig. 1 - 3D model of the humanoid robot and current stage of implementation

2.2. Actuators

Currently, the system is conceived with 22 actuators of three different types according to torque requirements of the several joints: more power on legs and less power on neck and arms. For the dimensions involved, off-the-shelf actuation technologies do not offer significant alternatives other than the small RC servomotors, such as those from FUTABA, HITEC and similar. There are several general characteristics that have made them actuators of choice in a large number of other applications: small, compact and relatively inexpensive. In fact, the servomotor itself has a built-in motor, gearbox, position feedback mechanism and controlling electronics. These features appear much more advantageous than assembling from scratch a DC motor plus gear mechanisms and its control unit; the result would certainly be not as compact as the RC servomotor. The drawback of these commercially common servomotors is that they accept only position control by means of a PWM signal at about 50 Hz and duty-cycles around 1-2 ms and offer no velocity or torque control!

This hard limitation had to be overcome by software emulation of velocity control by means of dynamic PWM generation with real feedback from the motor internal potentiometer. Several software techniques had to be developed (Ruas, M., *et al*., 2006) in order to perform velocity control and also evaluate joint torque so torque control may be a real possibility soon. The servomotors had to be deeply studied and several experimental analyses were carried out to develop appropriate controllers to ensure proper velocity and trajectory control.

Since these servos, although the most powerful among their counterparts, offer torques not much higher than 2 Nm, gear transmissions had to be implemented in the mechanical structure.

2.3. General perception and sensors

Perception assumes a major role in an autonomous robot, therefore it must be reliable or abundant, or both if possible! For this platform the following perception was planned:

- Joint position (reading servo own potentiometer)

- Joint motor current (related to torque)
- Force sensors on the feet (ground reaction forces)
- Inclination of some links (using accelerometers)
- Angular velocity of some links (using a gyro)
- Vision unit (located on top)

Up to now, only vision has not yet been specifically approached in the context of this project. The remainder sensors were addressed with different levels of accuracy, but all potentially usable with current hardware.

2.3.1 Servo potentiometer and motor current

Joint position is currently read directly from the servomotor potentiometer. This was not as easy as initially expected due the complexity of the servos internal control unit which appears not be completely disclosed to the general public. Indeed, the position reading only makes sense when duly synchronized with the PWM generation because doing otherwise will conflict with the servo own integrated controller (Fig. 2).

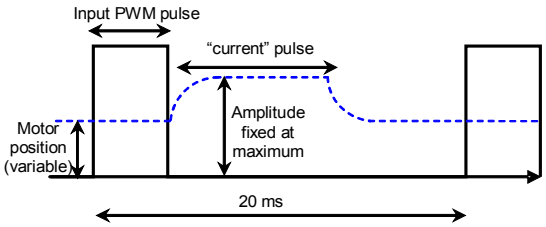


Fig. 2 - PWM for motor control and position feedback potentiometer reading

Having solved this initial difficulty, the need for an additional external potentiometer or encoder is now postponed *sine dia*. Related to this phenomenon is the electric current consumption which was initially expected to be measured indirectly by the voltage on a resistor (0.47 Ω) in series with the servo (Fig. 10). Fortunately, after studying the servos potentiometer during operation as observed in Fig. 2, current reading may be extracted from the potentiometer voltage level itself. All this has required elaborated low-level software development since PWM generation and sensor reading should be synchronized and tuned with resolution of up to 1 μs for three simultaneous servos (Fig. 9). The 0.47 Ω resistors became not needed and were later short-circuited.

2.3.2 Foot force sensors

The foot sensors are intended to measure the force distribution on each foot to further assist during locomotion or simply keeping upright. Four sensors on each foot allow evaluating balance and a behavior as the one illustrated in Fig. 3 is expected.

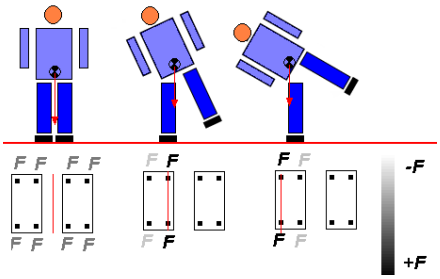


Fig. 3 - Force sensors and balancing

Commercial force sensors are expensive, so it was decided to develop a system based on strain gauges and amplify the deformation of a stiff material. The result is a kind of foot whose details can be viewed in Fig. 4 and is based on 4 acrylic beams located on the four corners of each foot that deform according to the robot posture. A simple Wheatstone bridge and an instrumentation amplifier complete the measuring setup (Fig. 5). The electronics hardware lays on a piggy-back board mounted on the local control unit, as can be seen in the lower part of Fig. 10.

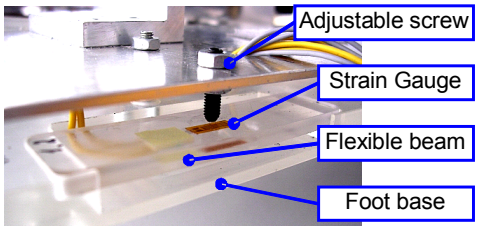


Fig. 4 - Foot sensor details

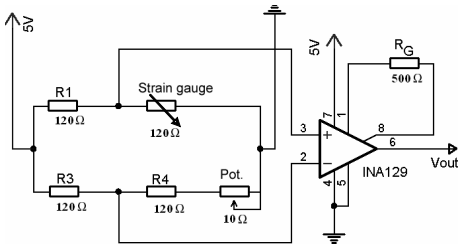


Fig. 5 - Circuit to measure force on the feet

2.3.3 Inertial devices

Inertial perception is also a relevant source of information for dynamic and also static locomotion and balancing. Accelerometers and gyroscopes furnish information on acceleration and angular velocity.

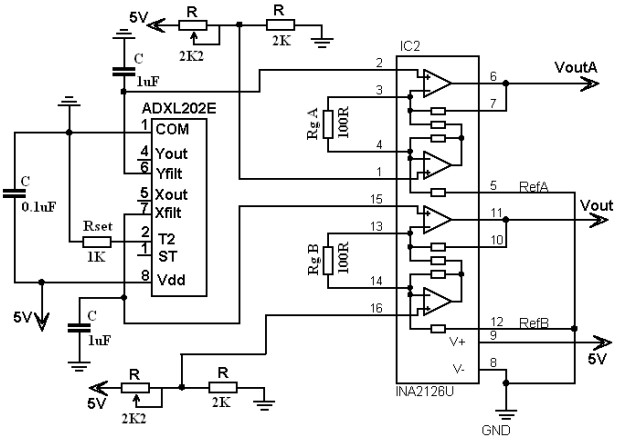


Fig. 6- Dual accelerometer electrical circuit

The accelerometers can be used to measure the acceleration of gravity, or better said, its component aligned with some axis. In other words, they can be used to measure inclination. That is what has been done by using the ADXL202E from Analog Devices. This very small MEMS device has two accelerometers in orthogonal axis that can be used to monitor tilt and roll angles of the platform. The

system was mounted on a small PCB as shown in Fig. 7. Measuring inclination can only be done in static or nearly static operation since motion inertia will be coupled with system accelerations.

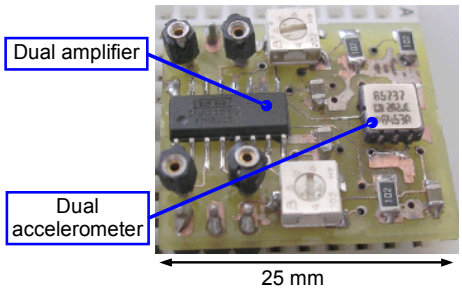


Fig. 7- Piggy-back board with two accelerometers

Finally, in what concerns sensors, a gyroscope unit has also been developed. The GYROSTAR ENJ03JA from MURATA has been selected due to several advantages and ease of interfacing. Up to now, although built and tested, the unit has not yet been used in the developed platform but its circuit is simply adapted from the vendor datasheet and using a INA129 amplifier.

3. Distributed architecture and software development

3.1. Principles of the approach

When the project begun, emerged the concern that controlling a machine with so many degrees of freedom would be very demanding if done on a centralized control unit. Further, different levels of control would coexist making it difficult to develop new software and would also make debugging difficult. Last, but not the least, the endless web of wires and plugs connecting motors, sensors and central unit would make it not an easy task when assembling or reassembling components!

Besides these practical concerns, it made much more sense to enable the platform with simpler control units responsible for fewer tasks and therefore more robust to failure. All units would be interconnected by a local network so information could be exchanged among them in case it was necessary.

All this led to the conception of a distributed system. Without loss of generality of the approach, some motion joints have been grouped by vicinity criteria and are controlled locally by a dedicated board based on a PIC microcontroller. A CAN bus relays all units in a slave configuration. The PIC controller chosen was the PIC18F258 simply because it contains CAN bus managing integrated hardware, and that was the main reason why to use the 18F series; other possibilities such as the PIC18F458 would be acceptable without any changes on the software.

Slave control units (SCU) generate PWM waves to control up to three actuators instead of only one. This rationalizes controllers in a practical implementation and also allows fast communication among three close joints by not depending on the bus. This represents an intermediate level of distributed system: not all actuators have fully independent hardware controllers. This concept was extended to perception. Once again, it would not be practical to have each of the several sensors with its own board hung up in the CAN bus. So, each slave unit has attached some nearby sensors that are somehow related to the actuators being controlled.

To collect data from the several SCUs, a special unit was conceived. Its role is to query all slaves for data and also give them directives or instructions for their behaviour. It was named the Master Control Unit

since it accesses in first place the CAN bus and relays all SCUs. Its functions are simply to dispatch and collect data from the bus. Nonetheless, and being made from the same hardware as are the slaves, it could also control actuators and monitor sensors (those roles are refrained for now by a matter of principle!).

To end the architecture description a final element is necessary: the main or general control unit. In a first stage, this unit is the interface for the programmer with the remainder blocks and allows accessing data and issuing orders to actuators but, in the future, its role will be the general control directives, interface with remote systems and, very importantly, to process high rate data such as vision. It will also be able to change the SCU firmware and control programs, but progressively it is expected not to do so since the purpose is to proceed to a largely distributed control system, even with the possibility of self-learning. Self-learning is for now a utopia, but with local sensors and control, and an adequate software application, the local controllers may be able to tune parameters and store the new values since the used hardware (Flash PIC controllers) has on-board EEPROM that can be used for dynamic storage. Fig. 8 illustrates a simplified representation of the architecture control units where only the 2 legs, one arm and the two joints for the trunk are shown making only 17 DOF visible; 3 more DOF exit for the other arm and 2 more for the neck/head, making a total of 5 additional DOF completing the set of 22 actuators.

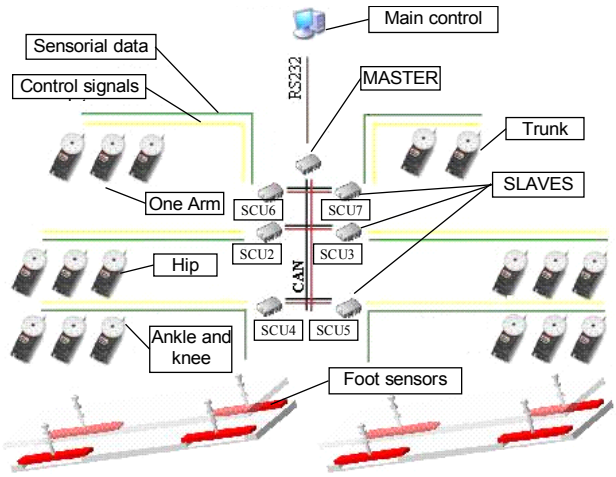


Fig. 8 - Schematic of architecture layout

3.2. Hardware for the units in the architecture

As mentioned, master and slave units are based on a PIC microcontroller. Slaves are all alike and can be distinguished by a configurable address. Slaves can drive up to 3 servomotors, and can monitor their angular positions and electrical current consumption. Concerning additional sensors, each slave unit has the possibility of accepting a piggy-back board where additional circuit can lay to interface to other sensors. Some examples of the developed piggy-back boards include force-sensors, accelerometers and gyroscope.

Fig. 9 shows a generic diagram of a slave unit. There, the main internal blocks can be seen, such as power supply regulation, CAN interface, the PIC controller, the multiplexer for sensor interfacing, PWM lines, CAN address switches and also lines prepared for RS232 communication. This kind of layout allows high versatility both on hardware and software approaches.

Being all similar, the construction of the boards is easier, along with software development (the same base code for all units). The master unit is different since it is not expected to drive motors neither to

acquire many sensorial data. Furthermore, it communicates both by CAN and serial RS232 to the upstream controller. Hence, its piggy-back module was used to interface electrically the RS232 communications by installing a MAX232 circuit instead of sensor acquisition.

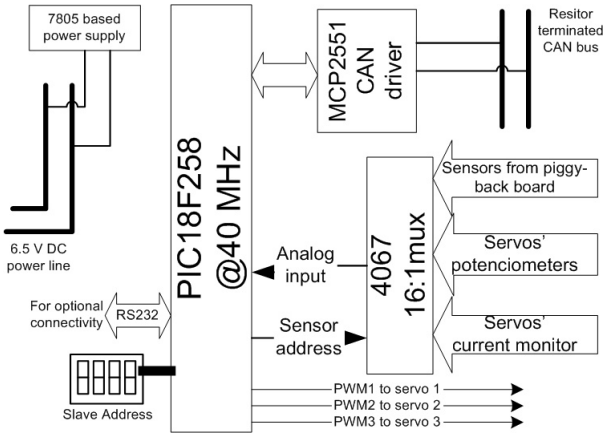


Fig. 9 - Block diagram of generic slave unit

Slaves will be able to perform local control when adequate algorithms will be developed. In the slave units, three PWMs are generated for the three servomotors with resolutions of few micro-seconds according to directives received from the CAN bus, but local algorithms may decide better how to control the motors instead of relying on central control. Still at the slaves, the sensorial data is currently acquired with 8 bits, but 10 bits are possible in case it becomes necessary and adequate signal filtering and conditioning is provided. Fig. 10 shows a slave unit PCB with the main components and also includes a piggy-back board for signal electric conditioning. The RS232 plug is only used in the master to communicate with main control unit; slaves do not use it for now.

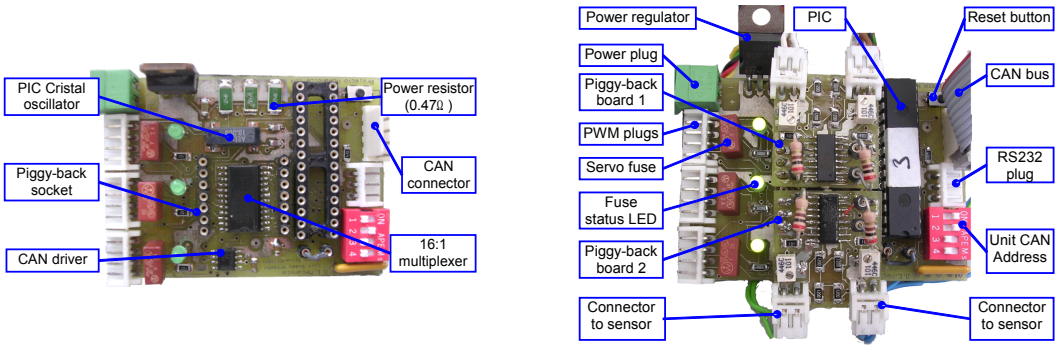


Fig. 10 - The slave processing board without most components (right) and in full mounting with two piggy-back boards for force sensors (left).

3.3. Communications: CAN and RS232 messages

In the current stage of development, on power-up or reset, each slave checks its address and starts monitoring the CAN bus. While no messages arrive, the slave unit will drive its joints to a home position at a reduced speed and starts monitoring local sensors at a given rate (pre-programmed). When messages from the CAN bus arrive, the slave unit will process them; messages are of two kinds: imposing new desired position and speeds to each of the three motors, or query for sensorial data. These requests come from the master at a variable rate depending on the number of slaves and the amount of data exchanged between master and slaves (currently, 8 units are used); the CAN bus is driven at nearly 1 Mbit/s.

CAN messages contain a data field with 8 bytes, enough to exchange (in one message only) orders for three servos (3 positions and 3 speeds). On the other hand, to gather data from the slaves, more than one message may be necessary. Indeed from 3 motors 6 variables are required (3 positions and 3 current levels), and additional sensor values (such as force or inertial) must go on other messages. The master keeps a current status of the full system and delivers that data to the main control unit, when requested, using the RS232 link. Currently, this protocol defines a 4-byte message to the master and a 6-byte message from master to control unit, but if necessary this may change for matters of efficiency.

Although the servos need not to be updated at rates higher than 50 Hz, the reading of some sensors may be higher than that for finer tracking, or during the development phase when all information is precious. Any way, a single calculation can be done: if 8 slave units are used and admitting for each board 2 messages towards slaves and 2 messages back to the master, this results in a full cycle of 8×4=32 CAN messages. If we assume 50 Hz as the maximal useful refresh rate, this results in 1280 CAN messages exchange per second that, at about 80 bits per message, yields ca. 102 kbits/s, which is far below the 1 Mbit/s that CAN provides. This means that more slaves may be added or additional information may be exchanged between master and slaves.

One of the strengths of this architecture is that the main base code is the same for all slaves. Different behaviors are now decided by each SCU own address (dip-switch adjustable). Base code consists of generating PWM for motors obeying position and velocity requirements. Torque control (or something similar to that) appears possible since instant motor current can be monitored. Next developments will enable slaves to accept control directives and perform their own control accordingly.

4. Low-level controller

Among the major challenges in building low-cost and easy-to-reproduce humanoid robots, the performance of their control architectures and the constraints on actuator systems assume a special importance. From the control point of view, the computational and sensing abilities of the robotic system impose additional limitations on the robot's performance. In general, the control problem consists of (1) providing the adequate computational resources and (2) using control laws and strategies to achieve the desired system response and performance. The first part of the problem has been extensively discussed in Section 3. Here, we concentrate on the second part with the emphasis being placed on the implementation of the low-level controllers to achieve an improved performance.

The humanoid robot combines a number of special-purpose pieces of hardware, deals with inputs from a variety of sensors with different time scales, and responds to external events in real time while attempting to meet several goals. The distributed set of interacting microcontroller units is the key element towards a control system where centralized and local controls co-exist and concur to provide robust full monitoring and easier expandability. A further advantage in generating the control signals locally at each controller unit is the contained computational overhead on the controlling software.

The basic idea is to select measurable parameters that can be used to anticipate the influence of the disturbances on the process variables. Then, feedback is provided to introduce suitable compensation control actions via the closure of an outer position control loop. In this work, procedures are described on how an external microcontroller can read the shaft position in order to evaluate intrinsic velocity by the

motor. The experimental results obtained can be seen as a basis for further position and velocity control improvements.

4.1. Servomotor limitations

The selected servomotor is practical and robust because the control input is based on a digital signal, whose pulse width indicates the required position to be reached by the device. Its internal controller decodes this input pulse and tries to drive the motor up to the required position. However, the controller is not aware of the motor load and its velocity varies with the load. By design, servos drive to their commanded position fairly rapidly depending on the load, usually faster if the difference in position is larger. Additionally, which may be critical, as the load increases a steady-state error occurs, turning the device into a highly non-linear actuator upon variable loads on the shaft.

4.1.1 Experimental setup

An entire system was set up to evaluate the actuator performance required for the servomotor advanced control; that includes a master and a slave unit controlling a servomotor properly fixed and loaded as described ahead. On the one hand, the master unit is connected to a computer through a RS-232 link, using MatLab software as the user's interface. On the other hand, the slave unit is connected to the servo mechanism in two ways: by sending the desired servo position command and by reading the potentiometer feedback signal. The experimental apparatus comprises several loads that will be applied to the servo shaft through a linkage with 10 cm long. The servo is fixed in a mechanical lathe such that its zero position corresponds to the perpendicular between the link and the gravity vector. Finally, and this was the sole hardware intervention on the servomotor unit, in order to measure the servo position feedback signal, an extra output wire was connected to the servo internal potentiometer. Fig. 11 shows photos from this experimental arrangement where a calibrated weight is being lifted up.

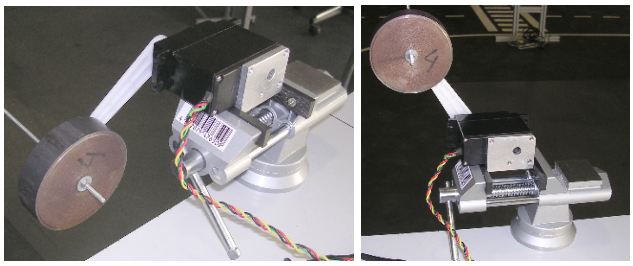


Fig. 11 - Experimental evaluation of actuators' response and velocity control using a HITEC HS805BB servomotor

4.1.2 Open-loop performance

All the control issues mentioned in this section report to the application of a given pulse train with a specific width. Therefore, the servo will be always presented with a Heavyside step in position. The first experiments are performed with "large" steps (equivalent to 90º) for several loads and, then, smaller steps (few degrees each) are used in order to simulate some kind of linear input and launching the basis for velocity control.

After applying a step from -45º to +45º, the first notorious observation is the presence of steady-sate errors. For a low mass, the steady state error is negligible, but for the larger load (1129g) about 8º error remains after the transient phase (Fig. 12).

Another observed anomaly in Fig. 12 is the unstable dynamic behavior on position reading, which shows at the beginning a sudden jump to a position below -45º and some oscillations during the path up to the final set point. The interesting part of this observation is that the motor shaft, physically, did not show this behavior; a continuous and a fast motion to the final position were observed without speed inversions or oscillations.

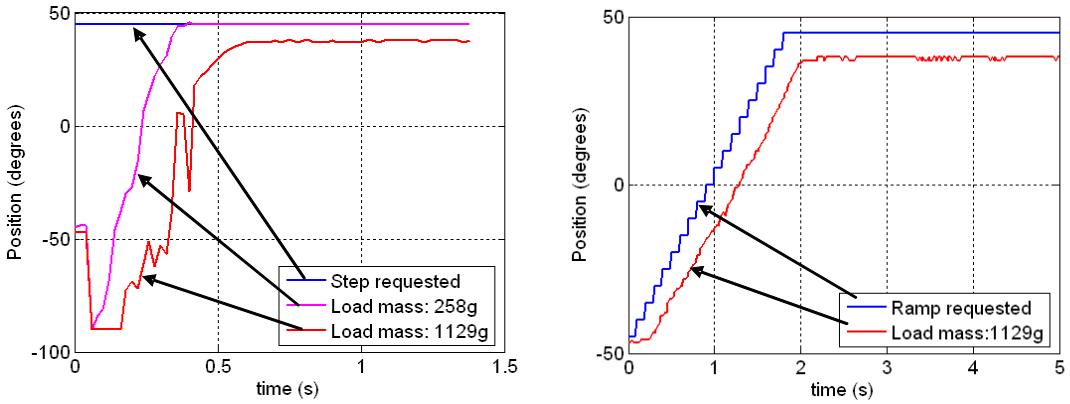


Fig. 12 - Step response for two loads from -45º to +45º (left) and response to a slope input (right)

In order to implement some sort of velocity control, some experiments were then carried out in a manner that a variable position would be successively requested to the servo. The rate at which each new position was imposed settled some kind of velocity. Nonetheless, the only way is still to give (smaller) position steps to the servo controller; only their magnitude and rate will dictate some desired "average velocity". This approach will generate an approximately linear increase (slope) for the position, which is to say, some constant velocity.

This way, the current demands will only practically depend on the load torque because of the speed limitation introduced by the ramp input (the levels of current will be lower). In addition, beyond the position control, velocity control is introduced by the definition of the ramp length. In Fig. 12 it can be seen that, although the transient response has a very improved behavior, the steady state error still exists.

4.2. Servo control enhancement

In all experiments reported above, the servo's own controller is the only responsible for the resulting performance. To overcome the mentioned limitations, an external PWM controller should be applied to the servo, and adequate parameterization has to be formulated.

4.2.1 Highlight of the approach

On the basis of the above limitations, two kinds of possible solutions could be devised. On one hand, the trend followed by several authors has been to emphasize on the embedded hardware by changing the motor internals. The price to pay, however, is often the replacement of the electronics unit of the motor package by dedicated control boards. On the other hand, it is expected that enhanced performance can also be achieved by software compensation, provided that position and/or torque measurements are available. In such cases, an effective strategy to improve the servo's operation is using an external controller, where an outer position control loop is closed around each slave unit. Fig. 13 illustrates the block diagram of the proposed servo controller.

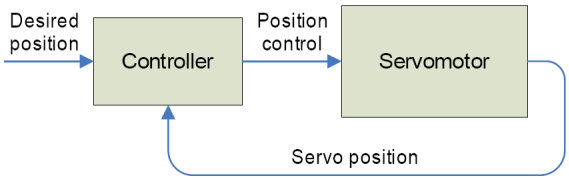


Fig. 13 - Servo controller diagram

The servo circuit has a very narrow input control range and it is difficult to control accurately, though it has adequate speed and torque characteristics. The outer position control loop is proposed as an effective tool to achieve good performance in terms of steady-state behavior and enhanced trajectory tracking capabilities. That is achieved by a variable PWM throughout the full excursion o a joint. The algorithm is based on dynamic PWM tracking using the servo own potentiometer for feedback. In other words, the software tracks motor position with time and adjusts the PWM in order to accelerate or decelerate the motor motion.

For that purpose, several control algorithms can be derived. The simplest approach that can be followed is to consider a digital PID-controller (or a particular combination of P, I and D cases). In this line of thought, this section focuses on the control and planning algorithms to generate smooth and stable motions, without requiring any modification of the servo internals. In order to validate these principles, the control schemes proposed are tested in a number of experiments using the same setup as described before. All control algorithms are implemented in discrete time at 20 ms sampling interval.

4.2.2 Incremental algorithm

In the case of interest, the system to control is formed by a single joint axis driven by an actuator with pulse-width control. To guide the selection of the control structure, it is also important to note that an effective rejection of the steady-state errors is ensured by the presence of an integral action so as to cancel the effect of the gravitational component on the output. These requisites suggest that the control problem can be solved by an incremental algorithm in which the output of the controller represents the increments of the control signal. Hence, the block diagram in Fig. 14 illustrates, in the z-domain, the proposed control scheme whose control law is described by the following equation:

$$U(z) = \frac{\left[K_I + K_P \cdot \left(1 - z^{-1}\right)\right] \cdot E(z) - K_D \cdot \left(1 - z^{-1}\right)^2 \cdot Y(z)}{1 - z^{-1}} \tag{1}$$

where $K_I = k_i \cdot T_S, K_P = k_p, K_D = k_d / T_s$ are constant positive gains, $Y(z)$ is the system position output and $E(z)$ is the error (difference between desired input and current output affected by a small dead-zone).

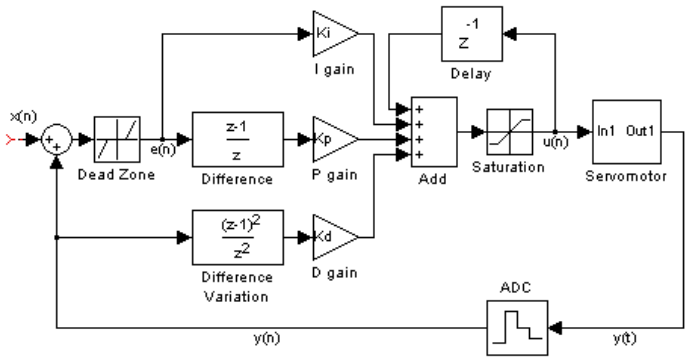


Fig. 14 - Implementation of the incremental algorithm

Several experiments were carried out in order to make a comparison between variations of the control scheme. The first experiment is aimed at verify the effectiveness of the integral action. It is required to move the joint angle from an initial value $q_i = -45^o$ to a final value $q_f = 45^o$ in a given time $t_f = 2s$, for a load of 924 g. Once again, the determination of the specific trajectory is given by position steps successively updated.

The results are presented in Figure 16 in terms of the desired and the measured angular positions. It can be observed significant differences occurring in the performance of the open-loop and the closed-loop system: the steady state error is eliminated and the delay time is reduced when applying this compensator. The additional curve (controller output) represents the real pulse-width control signal necessary to guarantee the effective conformity between input signal and output shaft position.

In the second experiment the proportional action is introduced in order to obtain a PI-controller that leads to improved speed response and damping. In this case, it is chosen a more demanding specification for the desired slope. Each new step position is update at the maximum rate of 50 Hz (corresponds to the PWM period) with an amplitude of 5 degrees. Let the desired initial and final angular positions of the joint to be -90 and +50 degrees, respectively, with time duration of 1.12 seconds.

Fig. 15 demonstrates the effect of increasing K_I for a fixed proportional term (K_P = 0.04). As expected, increasing K_I reduces the steady-state error due to the gravitational disturbances, but at the expense of overshoot. Although the PI control eliminates the steady-state error, it can be recognized that path tracking accuracy is poor during execution. Fig. 15 shows the time lag of the response behind the command input, thereby producing a large tracking error.

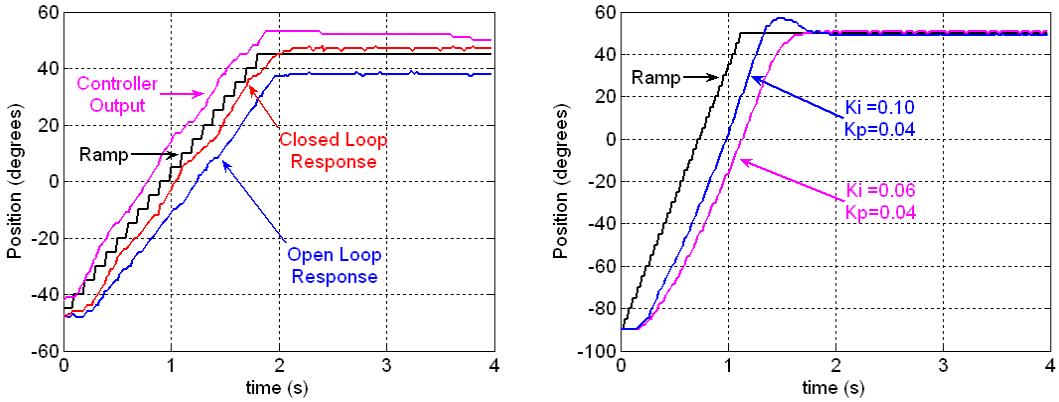


Fig. 15 - Response to a slope input for integral control with K_I = 0.2 (left) and for proportional plus integral control (right)

4.2.3 Trajectory planning

Improvement of the position tracking accuracy might be achieved by increasing the position gain constant K_P; however, this would give rise to larger overshoot and establishment times. Therefore, a better tracking performance is not expected in view of lack of a derivative term.

To this purpose, a third experiment is conducted such that the control algorithm is rewritten aimed to include the proportional, integral and derivative terms. However, a planning algorithm is used to generate smooth trajectories that not violate the saturation limits and do not excite resonant modes of the system. In general, it is required that the time sequence of joint variables satisfy some constraints, such as continuity of joint positions and velocities. A common method is to generate a time sequence of values

attained by a polynomial function interpolating the desired trajectory. The choice of a third-order polynomial function to generate the joint trajectory represents a valid solution. The velocity has a parabolic profile, while the acceleration has a linear profile with initial and final discontinuities.

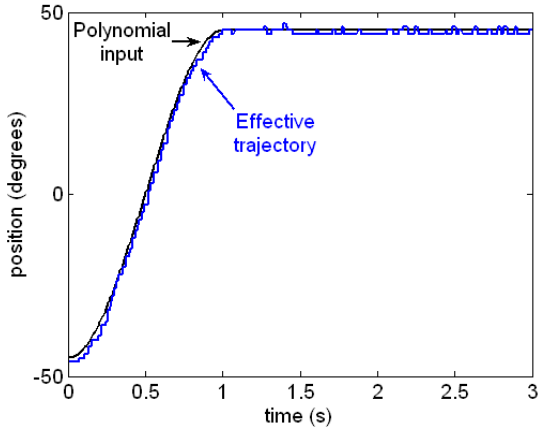


Fig. 16 - Response to a slope input for PID control (K_P = 1.46, K_I = 0.39, K_D = 0.15)

Fig. 16 illustrates the time evolution obtained with the following data: q_i = -45º, q_f = +45º, t_f = 1.12 s. The gains of the outer control loop have been tuned to limit the tracking errors. Significant improvements in the system's performance can be observed: zero steady-state error with no overshoot and limited tracking errors. The main drawback of the PID controller is that the load seen by the actuator can vary rapidly and substantially. As the control task becomes more demanding, involving high-speed movements or large loads, the performance of the PID controller begins to deteriorate.

4.3. Humanoid robot control

It is now desirable to extend the previous results from the single-axis system to the complete humanoid robot. Although the next development phase was facilitated by the reduction of performance demands and also smaller joint excursions, the interpretation of the last results deserves attention given the influence of the driving system. On the one hand, the single-axis system was actuated with direct drive for high demands and, therefore, the weight of nonlinearities becomes relevant. Instead, the humanoid system is actuated by servomotors with reduction gears of low ratios for typically reduced joint velocities. The presence of gears tends to decouple the joints in view of the reduction of nonlinearity effects. The price to pay is the occurrence of joint friction, elasticity and backlash that may limit the system's performance.

At the lower level in the control system hierarchy lay the local controllers connected by a CAN bus to a master controller. These slave control units generate PWM waves to control three motors grouped by vicinity criteria (entire foot up to knee and hip joints) and monitor the joint angular positions by reading the servo own potentiometer. In order to verify the effectiveness of the control scheme, a large number of experimental trials were carried out with the humanoid platform. The primary step is to demonstrate the behavior of a single-leg when performing some basic movements. More concretely, the desired movements to be performed consist of: (1) a vertical motion from an upright posture; and (2) a lateral motion in which the leg leans sideways (±27 degrees). In both cases, an additional load of 2.1 kg is attached to the upper part of the leg to emulate the mass of other segments.

There are two servo loops for each joint control: the inner loop consists of the servo's internal controller as sold by the vendor; and the outer loop which provides position error information and is updated by the microprocessor every 20 ms. We now compare the robotic system's behavior when only the inner loop is present (hereinafter "open-loop control") and when the extra feedback loop is added (hereinafter "closed-loop control"). In the later case, the outer servo loop gains are constant and tuned to perform a well-damped behavior at a predefined velocity. Further, the joint trajectories along the path are generated according to a third-order interpolating polynomial with null initial and final velocities.

The experimental results in Fig. 17 show the significant differences occurring in performance of the two control schemes (open-loop, and the cascading close-loop controller). The first observation is the usually poor performance of the open-loop control, particularly for steady-state conditions, which restricts the scope of its application. As a consequence of the imposed vertical motion, the limitations of the open-loop scheme are more evident when observing the temporal evolution of the ankle (foot) joint. On the other hand, an improved performance is successfully achieved with the proposed outer control loop, both in terms of steady-state behavior and enhanced trajectory tracking.

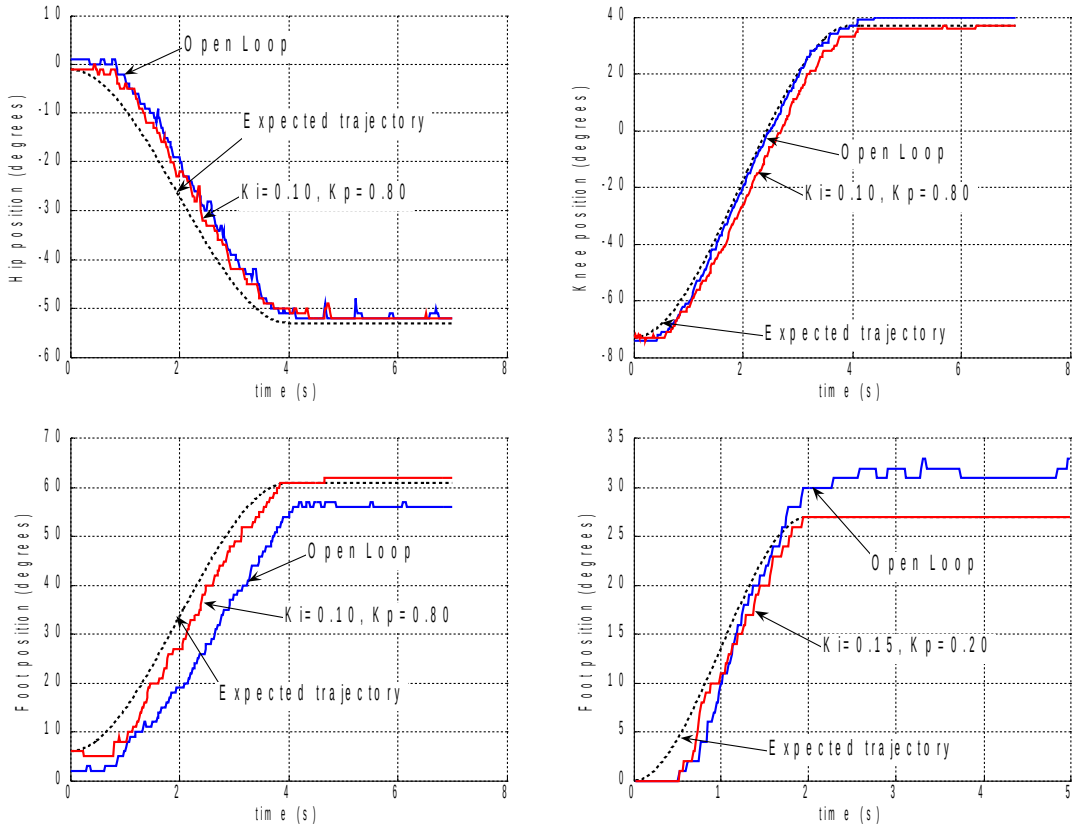


Fig. 17 - Response to slope inputs for a PI controller. Top and left-bottom charts: behaviours of the 3 involved joints during up-down motion of legs. Bottom-right: behaviours of foot joint in lateral motion.

Finally, the closed-loop control scheme was successfully extended to demonstrate that the robot could perform the same basic tasks while synchronizing and coordinating both legs. In order to simplify the experimental arrangement, the upper structure was disconnected and replaced by the 2.1 kg load. Fig. 18 shows the snapshots of the humanoid robot performing the desired vertical and lateral movements. Bearing this in mind, it can be asserted that the proposed control structure represents a contribution to

achieve an enhanced position-velocity control, which can in turn challenge and open new directions of research towards more advanced control algorithms.

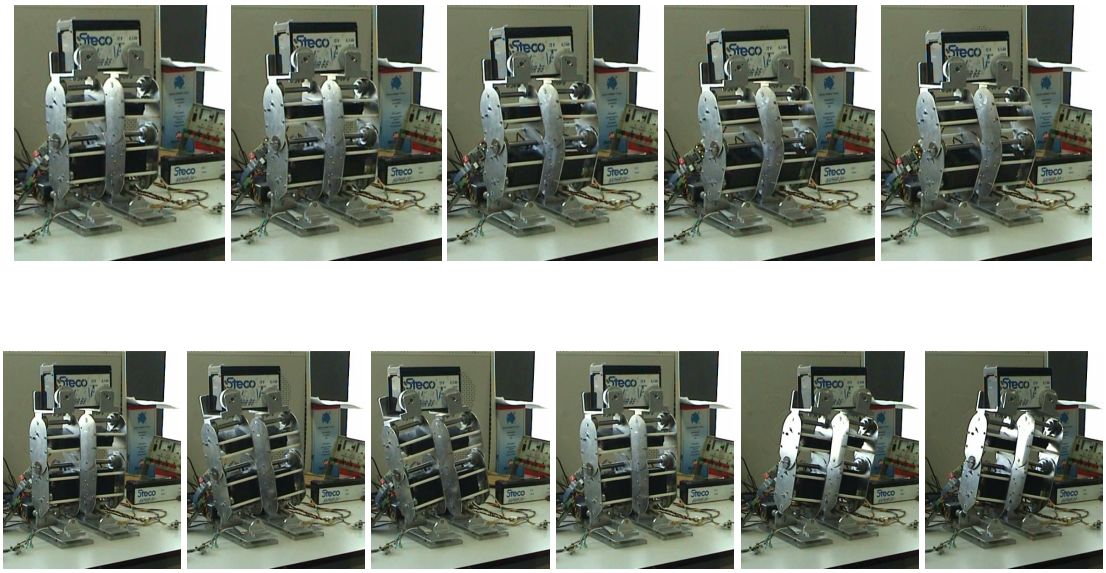


Fig. 18 - Legs performing vertical motion lifting (upper sequence) and lateral motion supporting (lower sequence) a 2.1 kg load in complete leg synchronization

5. Example of local control approach

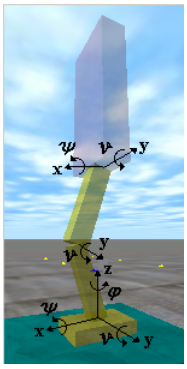
The general desire of providing the robotic system with enhanced capabilities motivated the appearance of different lines of thought. One of the most promising directions of research points out to the emergence of proper locomotion skills from the physical interaction between the machine and the environment itself (Fujimoto, Y., *et al.*, 1998; Park, J., 2001). It should be no surprise that robots with force, touch, distance and visual feedback are expected to autonomously operate in deterministic, but unknown terrains. This section shows an example that is being developed to demonstrate the possibility of achieving the motion of the upper body using a local control approach after considering the reaction forces on the ground. The main goal is to exploit and evaluate basic principles and control strategies based on the interaction forces between the robot and the environment. The realization of a force control scheme can be entrusted to the closure of an outer force control loop generating or modifying the reference input to the motion control scheme the robot is usually endowed with.

The open challenge is to allow local controllers to perform actuator control based on sensor feedback and possibly a general directive. For instance, supposing that the global order is to keep balance in an upright position, although all actuators can intervene, the ankle and knee joints have a relevant role to keep an adequate force balance on each foot. Here, we emphasize the role of a local controller, grouping the entire foot and knee joints, aiming to conciliate two imperatives: mobility and stability. The next subsections describe the control strategies applied to a simplified model in a dynamic simulation environment.

5.1. System and task description

The control algorithms presented in this paper are applied to a simulated 3-D robot model with 5-DOF and 4-links (foot, shank, thigh and trunk). The ODE simulation library (Russell, 2004) is used along with an interactive graphical user interface. The contact of the foot with the constraint surface is modeled

through linear spring-damper systems in the horizontal and vertical directions. Fig. 19 illustrates the articulated system, while the detailed parameters of this model are summarized in Table I.



Link	Mass (kg)	Dimensions (m)			Spring-damper model	
		lx_i	ly_i	lz_i	K_z (N/m)	B_z (Ns/m)
Trunk	4.0	0.06	0.15	0.30	50.0×10^3	1000.0
Thigh	0.5	0.04	0.04	0.15	Friction model	
Shank	0.5	0.03	0.03	0.15	μ^s	μ^d
Foot	0.1	0.12	0.08	0.02	1.20	2.50

Table I - Robot and environment parameters

Fig. 19 - Three-dimensional 4-link model

The tasks to be performed include a variety of motion goals specified in the intuitive Cartesian space, as well as in the joint space. More specifically, the desired tasks to be performed consist of: (1) movement of crouch from standing and then thrust the body upwards to assume again an upright static posture; and (2) voluntary trunk movements such as side bending. A useful means to assess balance skill and gain insight into postural control is by applying external perturbations and recording reactions. One typical disturbance experienced by a service robot is a perturbation due to external forces applied while the system is moving (Huang, Q., *et al.*, 2005).

5.2. The control algorithm

Our hypothesis is that the ground reaction forces are the key element through which new control strategies may be proposed to provide the required level of compliance, adaptation and dynamic stability (Popovic, M., *et al.*, 2005). In this line of thought, a hierarchical control structure, based on the interaction forces between the foot and the ground, is investigated. A block diagram of the general controller is sketched in Fig. 20, revealing the parallel operation of a force control loop and a position control loop. This parallel composition of control actions has been applied to exploit the redundancy of the system: a given actuator can be utilized to meet more than one task requirement (Puga, J., *et al.*, 2006). In view of the local control approach, we restrict our attention to a simpler implementation, while maintaining the two control loops. More specifically, the actuators of the ankle and knee will contribute to attain the motion directives (specified in the Cartesian space) based on a force control loop; the actuators of the hip will contribute to regulate the desired trunk orientation based on a position control loop.

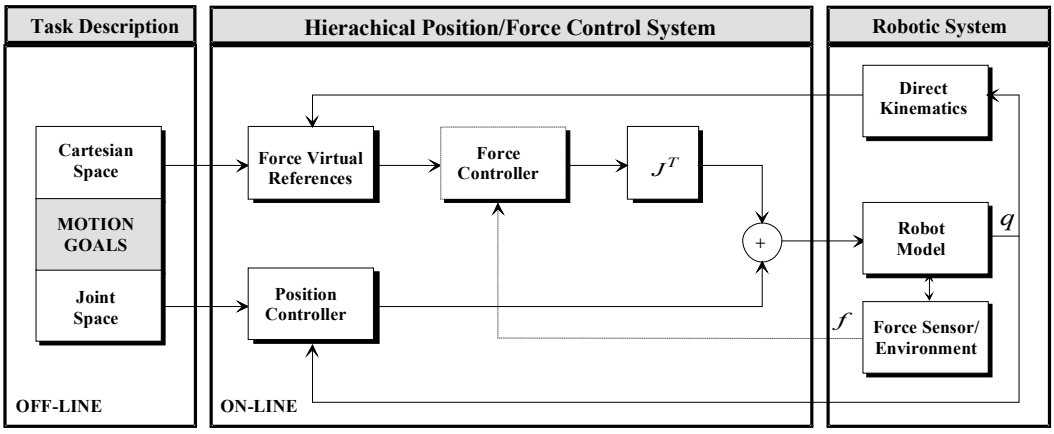


Fig. 20 - Blocks diagram of the hierarchical control scheme

As far as the force control is concerned, the tasks to be performed depend on motion goals defined in the Cartesian space (e.g., hip coordinates, Centre of Gravity - COG). On the other hand, the variables to be controlled are the reaction forces distributed along the foot's corners. In order to ensure the proper behavior through the execution of an interaction task, the reference variables must be generated online in result of the demands imposed to the system. These are the variables that some force control law must follow. For the present purposes, the reference forces are calculated through suitable actions on the position errors in both horizontal and vertical directions. The resultant normal reaction force is calculated from the errors measured in the vertical coordinate (z-axis) using a linear control law:

$$f_n^{ref} = BW + \left[K_P^f (z^{ref} - z) + K_v^f (\dot{z}^{ref} - \dot{z}) \right] \tag{2}$$

Here, f_n^{ref} is the reference normal force, BW is the total system's weight, z^{ref} and z are the desired and real vertical coordinates, $\dot{z}^{ref}$ and $\dot{z}$ are the corresponding velocities, and K_p^f and K_v^f are appropriated constant feedback gains. On the other hand, the desired location of the centre of pressure (COP) is calculated from the errors measured in the horizontal coordinates (x and y axis), as follows:

$$COP^{ref} = K_p^{COP} (x^{ref} - x) + K_v^{COP} (\dot{x}^{ref} - \dot{x}) \tag{3}$$

where COP^{ref} is the reference centre of pressure, x^{ref} and x are the desired and real horizontal coordinates, $\dot{x}^{ref}$ and $\dot{x}$ are the corresponding velocities, K_p^{COP} and K_v^{COP} are the position and velocity feedback gains.

Having defined the reference forces, there are many different ways to implement the compliance control. A relevant feature of the proposed method is the possibility of performing both indirect and direct force control. The former is obtained via motion control and without explicit closure of a force feedback loop (solid line). The later, instead, offers the possibility of controlling the contact force to a desired value, thanks to the closure of a force feedback loop (dashed line). This paper contributes with one strategy that considers only the indirect force control. In spite of the enhanced disturbance rejection provided by an inner force control loop, a compliant behavior can be successfully achieved with the proposed solution. This paper adopts one strategy that involves two steps. In the first step, the reference COP is actively used to calculate the distribution of the total reaction force along the extremities of the support foot. In the

second step, the signal forces obtained for each corner of the foot are transformed into joint torques by using the transpose of the Jacobian matrix:

$$\tau = \sum_{i=1}^{4} J_i^T f_n^i \tag{4}$$

where J_i is the Jacobian matrix which transforms the differential variation in the joint space into the differential variation of the end-effector's frame *i* (each foot corner) with respect to the reference frame (located at the hip). The subscript ^{T} denotes the transpose of a matrix.

In other words, the stance leg "feels" the forces while the controller distributes them as driving torques in order to regulate the desired high-level directive. Equation (4) requires lower computation than inverse kinematics or dynamics equations and it is well-behaved since, for a given force vector, a corresponding torque vector can always be obtained even if the robot is in a singular configuration. Although this strategy could be generalized to all degrees of freedom, only the ankle and knee joints torques are considered. It is interesting to note that, after some analytical simplifications, a computationally simpler control law can be derived, as follows:

$$\begin{aligned} \tau_{ankle\_\nu} &= -COP_x^{ref} f_n^{ref} \\ \tau_{ankle\_\psi} &= COP_y^{ref} \cos(\theta_1) f_n^{ref} \\ \tau_{knee} &= \left[COP_y^{ref} \sin(\theta_1)\sin(\theta_2) + l_{z2}\sin(\theta_1) - COP_x^{ref}\cos(\theta_2) \right] f_n^{ref} \end{aligned} \tag{5}$$

where θ_1 and θ_2 are the ankle angles pitch ν and roll ψ, respectively; the constant l_{z2} is the length of the shank.

5.3. Simulation results

Motivated by applications in biped locomotion, this subsection presents two simulation analyses. The first analysis illustrates the properties of the hierarchical control scheme based on local actions, as well as the systems behavior under perturbation. The second analysis illustrates the influence of voluntary trunk movements and how they are reflected in the ground reaction forces.

5.3.1 Robustness to perturbations

The first analysis illustrates the properties of the proposed control scheme when the system is on the level ground subjected to unpredictable perturbations. The results displayed below are based on the following desired path: the system is standing, moves down and up again to the initial posture in 4.5 seconds, while the trunk maintains a vertical posture. The initial state is set to $z_{hip} = 0.29m$ and the desired centre of gravity (x_G, y_G) should be zero along the motion. The motion planning is accomplished by prescribing the desired trajectories using sinusoidal-based functions. The controller's performance is evaluated by applying two unpredicted perturbations. These perturbations correspond to horizontal forces of $F_x = \pm 10N$ applied to the hip section at pre-defined moments of time and sustained for 20 ms.

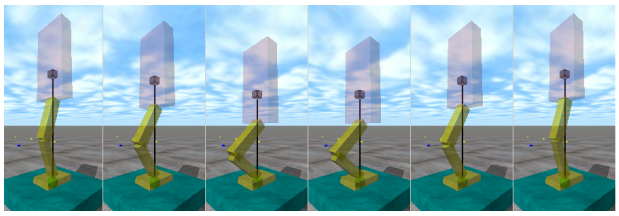


Fig. 21 - Movement sequence with the robot subject to unpredictable perturbations

The simulation results are shown in Fig. 21 up to Fig. 24. It is observed that the actual hip height profile was well-achieved, and the system makes the necessary postural adjustments. The system is only displaced by a few millimeters and it has stabilized shortly after the push. The control method is able to minimize the sway by generating a shear force quickly at the ground to counteract the perturbation. It depends on the latency at which it starts to resist the push and the rate at which this force can be increased. The last graphs show the temporal evolution of the computed joint torques. Given the proposed task, it is required a knee torque value that is significantly greater than the ankle pitch torque, while the lateral joints require almost no torque.

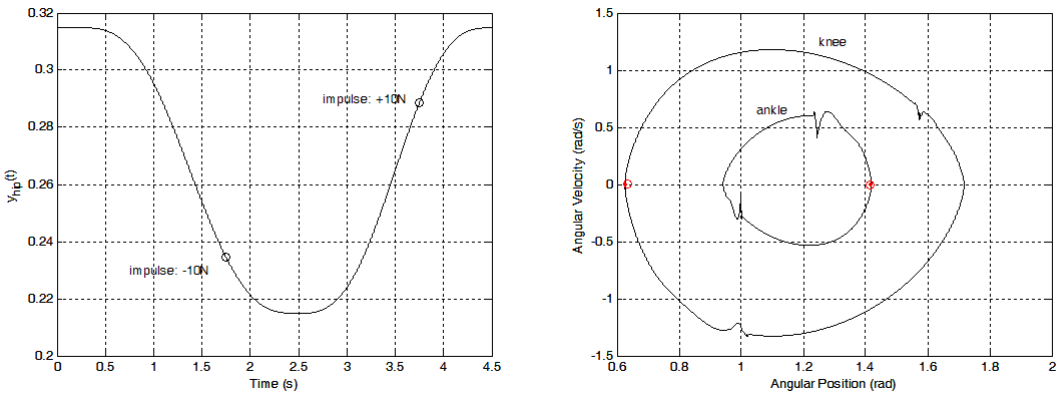


Fig. 22 - Temporal evolution of the desired vertical hip motion (left) and phase plane for the ankle and knee joints (right)

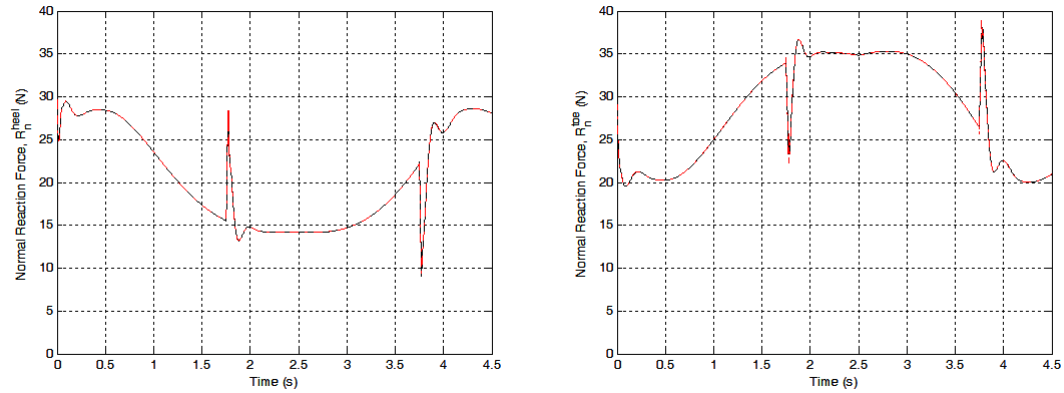


Fig. 23 - Temporal evolution of the real *vs*. reference normal ground reaction forces in the heel (left) and on the toes (right)

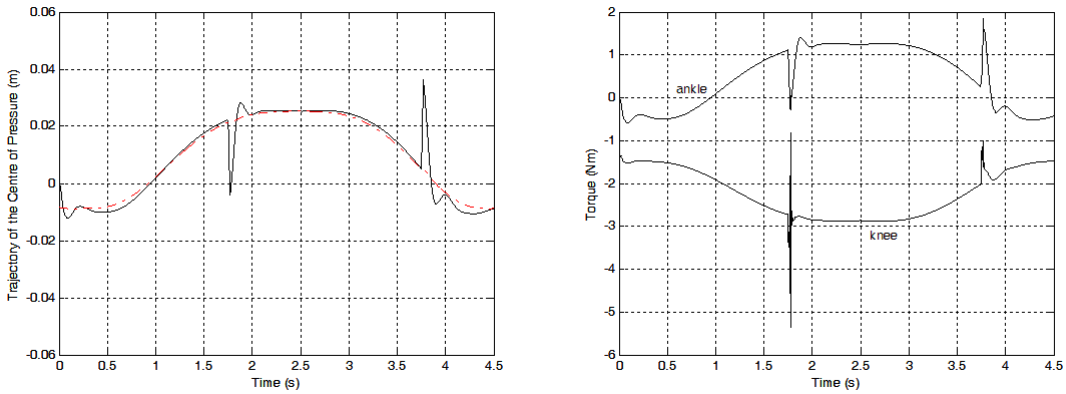


Fig. 24 - Temporal evolution of the centre of pressure along the *x*-axis (left) and joint torques (right)

5.3.2 Voluntary trunk movements

A control algorithm very similar to the one in the previous subsection was constructed for this analysis. Here, the objective is to prescribe a pre-defined oscillatory movement to the trunk, while the leg maintains a desired height in the sagital plane (the desired hip height is set to $z_{hip} = 0.29m$) and the desired (x_G, y_G) should be zero along the motion. It is assumed that the goal is to slant the trunk sideways.

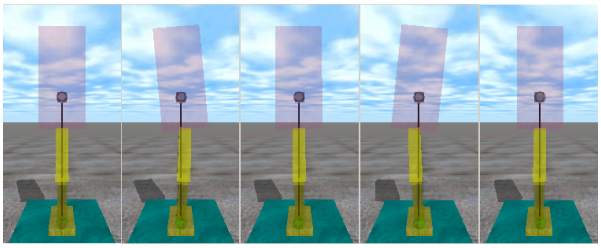


Fig. 25 - Movement sequence for voluntary trunk movements

The simulation results are shown in Fig. 25 and Fig. 26. The temporal evolution resembles in large measure what could be expected from the human behavior. The first graph shows the change induced in the hip's position that helps to counterbalance the trunk's movement. The second graph shows that the system was closely able to achieve the desired projection of the COG.

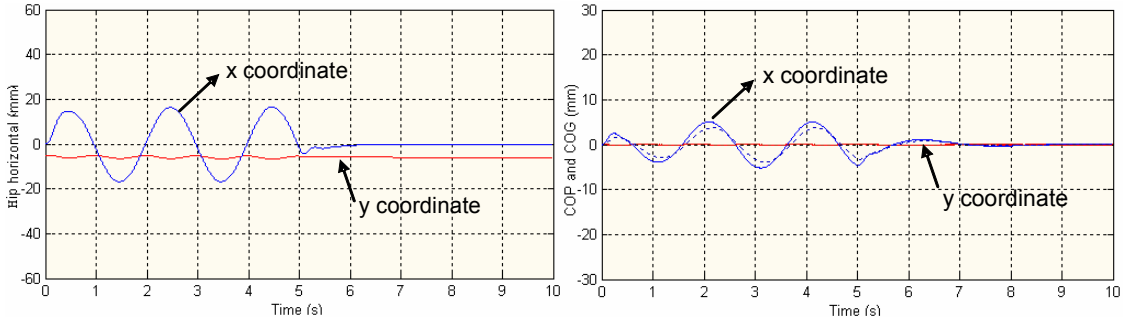


Fig. 26 - Simulation data for trunk movement. Left chart: hip's horizontal position along the *x* and *y*-axis. Right chart: location of the COP (solid line) and projected COG (dotted line) along the *x*- and *y*-axis.

6. Conclusion and perspectives

This paper described the development and integration of hardware and software components to build a humanoid robot based on off-the-shelf technologies. The design considerations that governed this project

are based on modular and reusable principles. The main features are the distributed control architecture and the relevance given to the sensorial information in order to achieve a better adaptive behavior. The distributed set of microcontroller units is the key element towards a control system that compensates for large changes in reflected inertia and providing variable velocity control.

Particular attention was given to the low-level control of RC servomotors as a relevant and abundant component of the humanoid system. Results with a closed-loop controller implemented with software show that motors' low-level velocity control has been made possible. The humanoid system reached a point where intermediate and high level control can now flourish. A simulation example has been given for a kind of intermediate level control implemented as a local controller based on force sensing.

Ongoing developments on the humanoid platform cover the remainder hardware components, namely the inclusion of vision and its processing, possibly with a system based on PC104 or similar. The future research, which has already started, will cover distributed control, alternative control laws and also deal with issues related to navigation of humanoids and, hopefully, cooperation. Force control techniques and more advanced algorithms such as adaptive and learning strategies will certainly be a key issue for the developments in periods to come in the near future.

7. Acknowledgments

The authors would like to thank the following former undergraduate students at the University of Aveiro for their contributions in the humanoid hardware and software implementation: Milton Ruas, David Gameiro, Filipe Carvalho, Luis Rego, Renato Barbosa, Mauro Silva, Luis Gomes, Ângelo Cardoso and Nuno Pereira.

8. References

Albero, M., Blanes, F., Benet, G., Pérez, P. Simó, J., Coronel, J., 2005, "Distributed Real Time Architecture for Small Biped Robot YABIRO", in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Espoo, Finland, June 27-30.

Cho, Y.-J., *et al.*, 1999, "A Compact/Open Network-Based Controller Incorporating Modular Software Architecture for a Humanoid Robot", *Journal of Intelligent and Robotic Systems* **25**, 341-355.

Fujimoto, Y., Kawamura, A., 1998, "Robust Biped Walking with Active Interaction between the Foot and Ground", in *Proceedings of the International Conference on Robotics and Automation*, Leuven, Belgium, May 16-20, pp. 2030-2035.

Furuta, T., Tawara, T., Okumura, Y., Shimizu, M., Tomiyama, K., 2001, "Design and Construction of a Series of Compact Humanoid Robots and Development of Biped Walk Control Strategies", *Robotics and Autonomous Systems* **37**, pp. 81-100.

Höhn, O., Gačnik, J., Gerth, W., 2005, "Detection and Classification of Posture Instabilities of Bipedal Robots", in *Proceedings of the 8th International Conference on Climbing and Walking Robots*, London, UK, September 13-15, pp. 409-416.

Huang, Q., Nakamura, Y., 2005, "Sensory Reflex Control for Humanoid Walking", *IEEE Transactions on Robotics* **21(5)**, pp. 977-984.

Kim, J-H. , *et al.*, 2004, "Humanoid Robot HanSaRam: Recent Progress and Developments", *Journal of Computational Intelligence* **8(1)**, pp. 45-55.

Nagasaka, K., *et al.*, 2004, "Integrated Motion Control for Walking, Jumping and Running on a Small Bipedal Entertainment Robot", in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, USA, April 26 - May 1, pp. 3189-3194.

Park, J., 2001, "Impedance Control for Biped Robot Locomotion", *IEEE Transactions on Robotics and Automation* **17(6)**, pp. 870-882.

Popovic, M., Goswami, A., Herr, H., 2005, "Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications", *The International Journal of Robotics Research* **24(12)**, pp. 1013-1032.

Puga, J., Silva, F., Cunha, B., 2006, "Online Motion Control Algorithms Towards Biped Walking", in *Knowledge and Decision Technologies*, Z. Vale, C. Ramos and L. Faria, eds., Polytechnic Institute of Porto, Porto, pp. 219-226.

Ruas, M., Silva, F., Santos, V., 2006, "Parameter Measurement for Speed and Torque Control of RC Servomotors on a Small-Size Humanoid Robot", in *Proceedings of the Encontro Científico do Festival Nacional de Robótica*, ROBOTICA2006, ISBN 978-972-98603-2-4. pp. 71-79, Guimarães, Portugal.

Russell, S., 2004, "Open Dynamics Engine v0.5", in http://www.ode.org, May 2004.

Sakagami, Y., *et al.*, 2002, "The Intelligent ASIMO: System Overview and Integration", in *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, September 30 - October 5, pp. 2478-2483.

Tomokuni, N., Saiga, M., Yabuta, T., 2005, "Distributed Controller for Cooperative Joint Torque Control of Compact Humanoid Robots", in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan, December 5-7, pp. 259-264.

Yamasaki, F., Matsui, T., Miyashita, T., Kitano, H., 2000, "PINO - the Humanoid that Walk", in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Boston, Massachusetts, USA, September 7-8.

List of Figures