

Development of Control Algorithms for a Humanoid Robot

Milton Ruas, Filipe M. T. Silva, Vítor M. F. Santos¹

¹ Department of Mechanical Engineering, TEMA
University of Aveiro, 3810-193 Aveiro, Portugal

Resumo - Este artigo descreve o trabalho desenvolvido ao longo de um projecto final de curso, visando o desenvolvimento de algoritmos de controlo para um robot humanoíde de pequenas dimensões usando tecnologias comerciais. O destaque principal é dado à implementação das estratégias de controlo de baixo nível com vista à realização de tarefas básicas de locomoção e equilíbrio. As características dominantes desta implementação são o controlo distribuído e a relevância atribuída à informação sensorial. Por serem fundamentais para a especificação dos níveis mais elevados de controlo, são descritos alguns aspectos práticos relacionados com os actuadores utilizados (servomotores). No que concerne ao melhoramento do seu desempenho recorre-se a um método de compensação por software. A arquitectura distribuída utilizada nas unidades controladoras representa o elemento chave para um sistema de controlo que permite mecanismos de compensação face a variações de inércia, proporcionado controlo de posição e velocidade. Além disso, foram implementados controladores locais de médio nível, baseados em sensores de força, dotando o sistema de um comportamento adaptativo em resposta a variações na inclinação da superfície de suporte.

Abstract - This paper describes the work developed in a final course project, aiming at the development of control algorithms for a small-size humanoid robot using off-the-shelf technologies. The main focus is placed on the implementation of low-level control strategies for both locomotion and balancing tasks. The main features of this implementation are the distributed control approach and the relevance given to the sensorial information. Some practical issues on servomotor control are given since that turned out necessary before entering higher levels of control. Particular attention is given to the low-level control of RC servomotors and the enhanced performance achieved by software compensation. The distributed set of controller units is the key element towards a control system that compensates for large changes in reflected inertia and providing position and velocity control. Furthermore, an intermediate local-level controller is implemented based on force sensing, providing robust and adaptive behaviour to changes in a slope surface.

I. INTRODUCTION

The field of humanoid robotics has stimulated an increasing interest in a wide community of researchers. As

consequence, the number of research and development projects aimed at building bipedal and humanoid robots abound. The great success of Honda's robots [1,2] has inspired others to replicate the impressive design and skills [3]. On the other side, humanoid projects from academia, operating on a much smaller budget, aims at research on low-cost and easy-to-design robots, such as PINO [4], ESYS [5] and HanSaRam [6]. As robotics technology continues to progress, there will be a need for software and algorithms useful to improving the usability and autonomy of these complex machines.

The goal of this paper is to present the technical, technological and innovative controls aspects in building a small-size humanoid robot at reduced costs using off-the-shelf technologies. The work reported here is based on a final course project performed at the Department of the Electronics, Telecommunications, and Informatics (DETI), with the collaboration of the Department of Mechanical Engineering (DEM), which provided a humanoid platform still in a prototype phase, but offering already capabilities of implementing control algorithms using a network of microcontrollers connected to several actuators and sensors. In this way, this project has an experimental basis, where all proposed ideas and particular control algorithms have been tested and verified on this real robot to form a critical hypothesis-and-test loop. Fig. 1 shows the humanoid robot at the current stage of development.

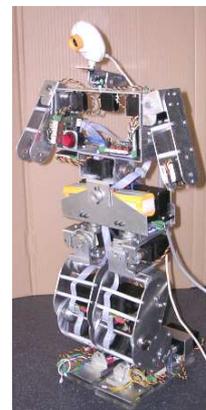


Fig. 1 - Biped humanoid robot with 22 DOFs.

The paper begins by presenting the engineering solutions and the research results with potential of practical

application, including the performance of servomotors, practical issues on their control and the test of the distributed control approach. Afterwards, it is reported the development of the low-level control structure which is obtained by the addition of an outer position feedback loop to the servo's internal controller. The most relevant feature of this implementation is the distributed architecture where centralized and local control may co-exist to provide robust full monitoring and efficient control. The integration in simpler local control units of a sensorimotor command layer that encapsulates useful combinations of sensing and action play a key role to allow for more advanced algorithms that go far beyond the classical control of robots. Finally, the paper describes the development of force-driven actuation and control modules, successfully applied to demonstrate the possibility of keeping the humanoid robot in upright balance position using the ground reaction forces.

This remainder of the paper is organised as follows. Section 2 describes the main technological aspects, including the actuators and the sensorial requirements. Section 3 presents an overview of the actuators performance and its limitations in the wide scenarios they will face with. Section 4 describes the implementation of the low-level locomotion controllers using a dynamic PWM generation with real feedback from the motor internal potentiometer. Section 5 gives an explanation with examples of the intermediate level control implemented as a local controller based on force sensing. Section 6 concludes the paper and outlines the perspectives towards future research.

II. ROBOTIC SYSTEM'S DESCRIPTION

The main scope of the project beneath this paper has been the development of a humanoid robot to carry out research on control, navigation and perception. The ultimate goal is to build a prototype capable of participating in the RoboCup humanoid league where a wide range of technologies need to be integrated and evaluated. In order to provide adequate mobility, the each leg consists of six rotary joints: two joints at the ankle whose axes are orthogonal (pitch and roll), one at the knee (pitch) and three at the hip (pitch, roll and yaw).

The most relevant achievements of this implementation include the distributed control architecture, based on a CAN bus, and the modularity at the systems level [7][8].

A. Distributed Control Approach

From the very beginning of the project, one major concern has been the development of a flexible control system to allow for short and possibly longer term developments. The key concept for the control architecture is the distributed approach, in which independent and self-contained tasks may allow a standalone operation. The platform was given a network of controllers connected by a CAN bus in a master-multi slave arrangement. Master

and slave units are based on a PIC microcontroller. Fig. 2 shows a generic diagram of the controlling units.

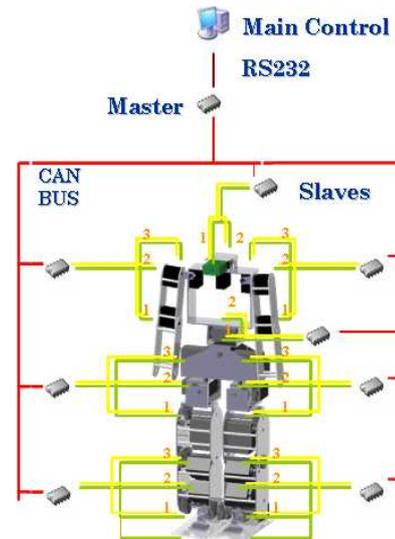


Fig. 2 - General architecture layout.

Slaves (Fig. 3) can drive up to three servomotors, monitor their angular positions and electrical current consumption. The system joints have been grouped by vicinity criteria and are controlled by a dedicated board. Concerning additional sensors, each slave unit has the possibility of accepting a piggy-back board where additional circuit can lay to interface force-sensors, accelerometers and gyroscope. A complete description of the control architecture can be found elsewhere [8].

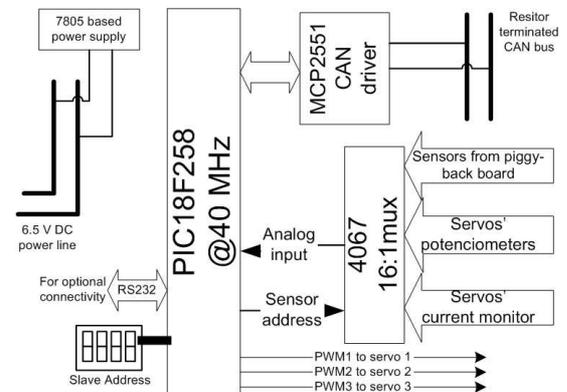


Fig. 3 - Block diagram of generic master/slave units.

B. Actuators

The complete 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 HITEC. The servomotor itself has a built-in motor, gearbox, position feedback mechanism and controlling electronics. 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.

This type of devices is so practical to use and robust, that their widespread use made them affordable to a large community. Currently, many brands of servos exist and roughly since the early 1990s some uniformity and standards have been established among brands for pins, sizes, and other technical issues [9]. The selected servomotors are 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.

Fig. 4 shows one of the HITEC servomotors used in this project. Table 1 presents their main specifications according to common information on flyers and datasheets spread over internet sites and vendors [10]. It should be noted, however, that not all specs were actually verified as announced. Most people working with servos rely on sparse or informal knowledge available in websites [11].

An entire system was set up to evaluate the actuator's performance that includes a master and a slave unit controlling a servomotor properly fixed and loaded as described above (Fig. 5). On the one hand, the master unit is connected to a computer through an 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.



Fig. 4 - HITEC HS805BB servomotor.

Spec	Values
Control system	Pulse Width Control 1.5 ms neutral
Voltage range	4.8V to 6.0V
Teat voltage	@ 4.8V @ 6.0V
Speed (no load)	60°/0.19 s 60°/0.14 s
Stall torque	1.94 Nm 2.42 Nm
Operating angle	45° /one side pulse traveling 400µs clockwise/pulse traveling 1.5 to 1.9
Direction	ms
Current drain	8mA (idle); 700mA (no load running)

Dead bandwidth	8 µs
Dimensions	66 x 30 x 57.6 mm
Weight	152g

Table 1 - HITEC HS805BB servomotor specifications.

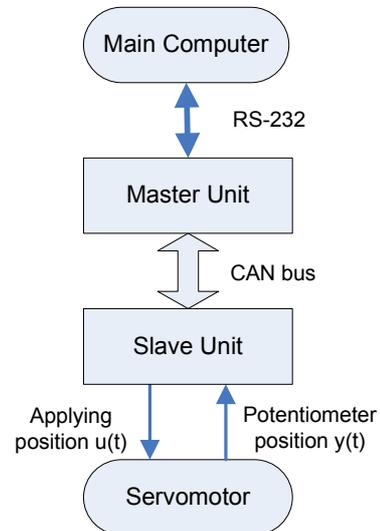


Fig. 5 - Block diagram of the experimental setup.

The experimental apparatus comprises several loads that will be applied to the servo shaft through a linkage 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. Fig. 6 shows this experimental arrangement where a calibrated weight is being lifted up. The complete set of individual loads used in the experiments and their combinations are indicated in Table 2. 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. 6 - Experimental evaluation of the actuator's response using a HITEC HS805BB servomotor.

Load	Mass (g)	Torque (N.m)
0	9	0.009
1	258	0.253
2	463	0.454

Load	Mass (g)	Torque (N.m)
3	675	0.662
1+3	924	0.906
2+3	1129	1.108
1+2+3	1378	1.352

Table 2 - Different loads used during experiments and maximum static torque required.

C. Programming Issues Related to Servos

To be able to control this device, we need to generate a PWM pulse with a 20ms period (50Hz) and a variable duty-cycle from 1 to 2ms. Concerning the sensorial information, additional requirements are imposed in order to read the shaft motor position and the current consumption. A PIC18F258 microcontroller from Microchip was chosen to perform these tasks.

The PWM generation requires two software interrupts used for the rise up (timer 1 at 20 ms) and the fall down voltage level (timer 2). The PIC has embedded PWM generators but they could not be used since their lowest frequency was too high for this application. To be able to control successfully the duty-cycle for each motor, a periodical high frequency interrupt is generated within the PWM impulse (1 to 2 ms) with a resolution corresponding to 1° (Fig. 7). In the PIC program, each servo has an associated variable for the impulse duration and, for each step, it is verified whether the PWM signal will fall down or not. This way, it is possible to control any number of motors using only two timers, as far as there is enough CPU bandwidth to execute the necessary code: for a 10 MIPS microprocessor, we can use up to 55 assembly instructions for each step of 1°.

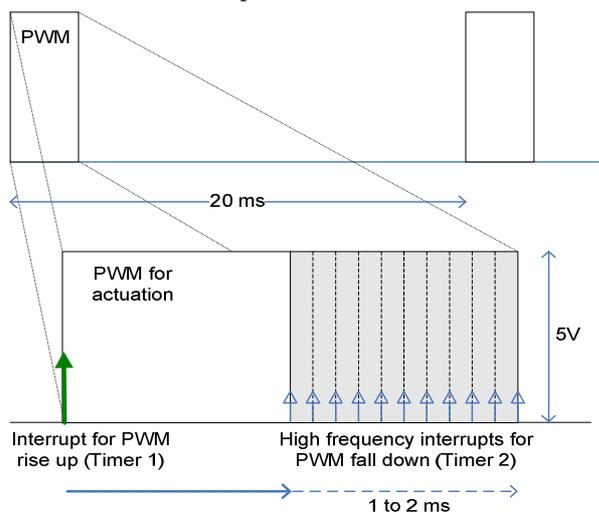


Fig. 7 - PWM generation.

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. Indeed, the position reading only makes sense when duly synchronized with the PWM generation; doing otherwise will conflict with the servo own integrated controller. Having solved this initial difficulty, the need for an additional external potentiometer or encoder is now postponed *sine die*.

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. Fortunately, after studying the servos potentiometer during operation, as observed in Fig. 8, 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 are synchronized and tuned with resolution of up to 1 μs for three simultaneous servos.

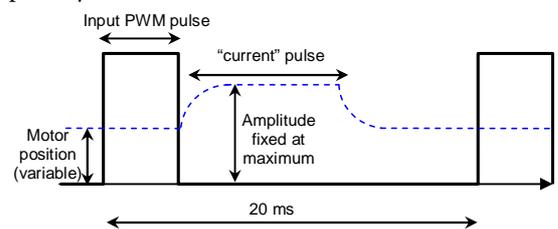


Fig. 8 - PWM for motor control (black) and position feedback potentiometer reading (blue).

D. 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 force sensors on each foot allow evaluating balance. 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 based on 4 acrylic beams located on the four corners of each foot that deform according to the robot posture (the details can be viewed in Fig. 9). A Wheatstone bridge and an instrumentation amplifier complete the measuring setup (Fig. 10). The electronics hardware lays on a piggy-back board mounted on the local control unit.

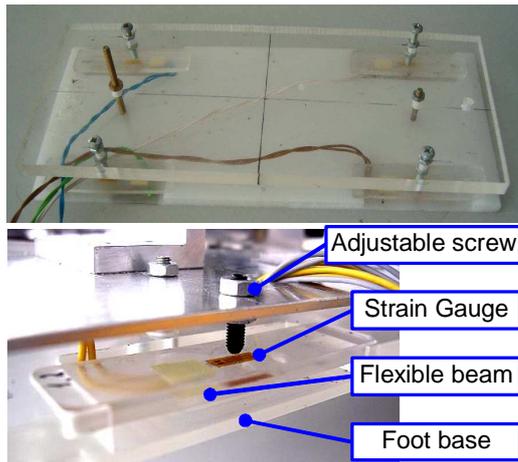


Fig. 9 - Foot details with the four force sensors.

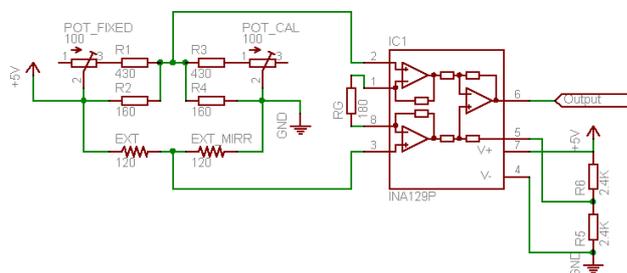


Fig. 10 - Strain gauges outputs amplifier.

Due to the very small amplitude of the signals involved, additional care in building the circuit for force measurement has been taken. Namely, to compensate for temperature variations and noise asymmetries, the bridges have been made with full symmetric components: (i) two strain gauges were used; one subjected to the external force (EXT) and a second one in the opposite arm of the bridge that does not suffer any strain, but ensures similar temperature and noise responses (EXT_MIRR); and (ii) similar adjusting potentiometers have been installed on the other bridge arms to ensure the proper behaviour for temperature and noise (POT_FIXED e POT_CAL). In general, results are quite satisfactory, showing a large stability even for small forces.

III. SERVOMOTOR'S PERFORMANCE

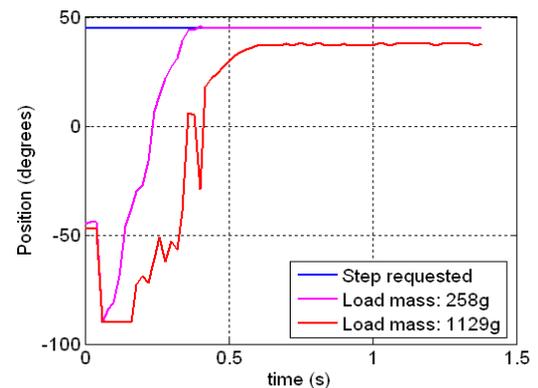
All control mentioned in this section reports to the application of a given pulse train with a specific width. Therefore, the servo will be always presented with a heavyside step in position. The first experiments are performed with "large" steps (equivalent to 90°) for several loads and, then, smaller steps (few degrees each) are used in order to simulate some kind of linear input and launching the basis for velocity control. The experimental setup used is the same as described in Section 0.

In all experiments reported in this section only the servo's own controller will be responsible for the resulting performance. The only feedback available, besides the visual input for humans, is the shaft position read at the

servo's potentiometer as described in the previous section; further, it shall be noted that this process also requires care since apparently the internal controller interferes with the voltage drop on this potentiometer and that can affect external readings of the shaft position.

A. Response to one large input step

After applying a step from -45° to $+45^\circ$, the first notorious observation is the presence of steady-state errors. For a low mass, the steady state error is negligible, but for the larger load (1129g) about 8° error remains after the transient phase (Fig. 11).


 Fig. 11 - Step response for two loads from -45° to $+45^\circ$.

Another observed anomaly in Fig. 11 is the unstable dynamic behaviour on position reading, which shows at the beginning a sudden jump to a position below -45° 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 behaviour; a continuous and fast motion to the final position was observed without speed inversions or any kind of oscillations.

B. Response to a "slope" input

In order to implement some sort of velocity control, several 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 due to 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 can be seen that, although the transient response has a very improved behaviour, the steady state error still exists. Table 3 shows the results of an

experiment carried out to stress this effect: the servo is requested to successively move a given weight to some positions; for each position, after motion completion, the potentiometer is sampled to obtain the real position that the servo achieved.

Relating the positional error with the static torque exerted in the joint, $\tau = mgL \cos \theta$, a direct conclusion can be drawn: the higher the torque, the higher is the steady state error. Here, θ is the angular position, g the acceleration of gravity, m and L are the load mass and the linkage length, respectively.

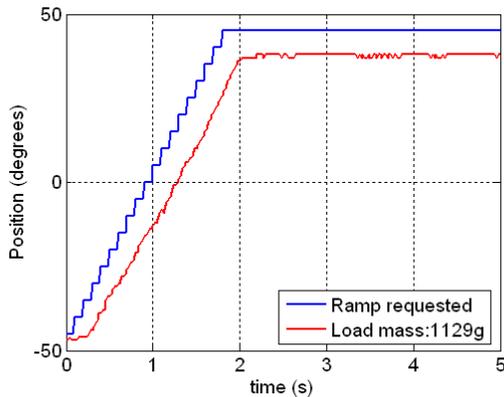


Fig. 12 - Response to a slope input for the highest load

Requested position °	measured position °	Error °	Torque (Nm)
-80	-80	0	0.198
-60	-62	2	0.569
-40	-45	5	0.872
-20	-28	8	1.069
0	-9	9	1.138
+20	+11	9	1.069
+40	+33	7	0.872
+60	+55	5	0.569
+80	+80	0	0.197

Table 3 - Steady state error and physical torque exerted in the motor for a fixed set of positions using a 1138g load.

To correct these deviations, an external controller could be devised which resorts to proper selection of measured parameters and output feedback.

C. Parameter Measurement

The last results show that, when carrying out long trajectories at the maximum speed (such as in response to large step inputs), the potentiometer output tends to become unstable, especially when increasing the load mass. This is due to the way the potentiometer voltage is read: different grounds for external measurement and internal controller. Therefore, the potentiometer output is only consistent with the real servo position when a low

current is being drained! For high loads (or fast motions) the servo increases the current demands and adds an “extra” output pulse above the position voltage on the potentiometer. This occurs perfectly synchronized with the input PWM signal (Fig. 13). The previous observations yield clues on the servomotor internal controller principles. After receiving each input pulse, the internal controller will provide current to the motor in the extent needed to accomplish the required angular displacement.

With the assumption that the internal controller uses a power bridge and an internal PWM generator to drive the motor, it is reasonable to accept that the “extra” pulse in the potentiometer level is due to the fact that instantaneous power is applied during that period to the motor by the internal controller. It was also observed that the “extra” pulse on the potentiometer level has a fixed top value.

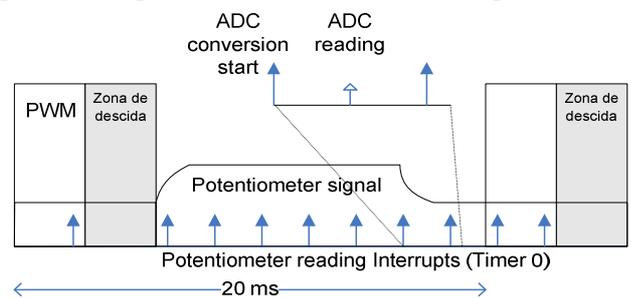


Fig. 13 - Temporal organization of the interrupts.

All things add up to conclude that the average power being transferred to the motor during a cycle of PWM (20 ms in these servos) is proportional to the width of this “extra” pulse. In other words, admitting a fixed voltage applied by means of an internal bridge, the electric current being required by the motor is proportional to the width of the pulse, that we shall now onwards refer to as the “current pulse”.

Two main issues were concluded by these observations: care must be taken when reading the potentiometer value, and a means to measure electric current (in periods of 20 ms) was found! Only the accuracy in measuring the “current” pulse width will limit the accuracy in reading the current.

Strategies were then devised to measure motor position (i.e., potentiometer voltage). The simplest way is to sample the potentiometer several times, during the full period of one PWM pulse, and take the minimal value. Nevertheless, for high loads and high current demands, the current “pulse” can last the entire PWM free period, inhibiting position reading! Such a situation occurred in the experiments reported in Fig. 11: low peaks in the transient behaviour correspond to high peaks in the voltage caused by the current pulse interference. This can be minimized by reading position exactly during the active part of the input PWM pulse. Note that the current pulse has its ascension at the PWM fall down, so, even for high demands, the referred phase usually is not affected by current pulses. Nevertheless it is important to avoid

excessive current draining due that the current pulse can last all PWM period making impossible position reading.

In this line of thought, a 200 μ s periodical interrupt (timer 0) is used for the entire PWM period excepting the fall down zone in order to provide maximum CPU bandwidth for the fall down task (Fig. 13). Thus, during 19ms the potentiometer voltage is read, updating continuously its minimum value and counting the number of cycles (of 200 μ s) in which the output is above a certain level of the previous PWM period minimal voltage. This level is defined as being half the current pulse amplitude. The accuracy of the results is related to the cycle duration.

Nevertheless, this cycle duration has some constraints: the CPU bandwidth to execute the code for one measurement, the acquisition time necessary so that the ADC have a stable voltage on its input, and the conversion time for the digitalization of the input voltage. Additionally, reading the three servos' positions is made through a multiplexer whose selected input must be switched, requiring an extra delay fundamental to assure a stable output. Considering 10 μ s for the associated code execution, 20 μ s for the acquisition time and 40 μ s for the conversion time, 150 μ s remain for multiplexer output stabilization, which is far enough to ensure a viable result. In order to full exploit the free CPU bandwidth, all the events are generated by interrupts, and servo readings are multiplexed in time resulting in 600 μ s to read all three sensors. The complete sequence is illustrated in Fig. 14.

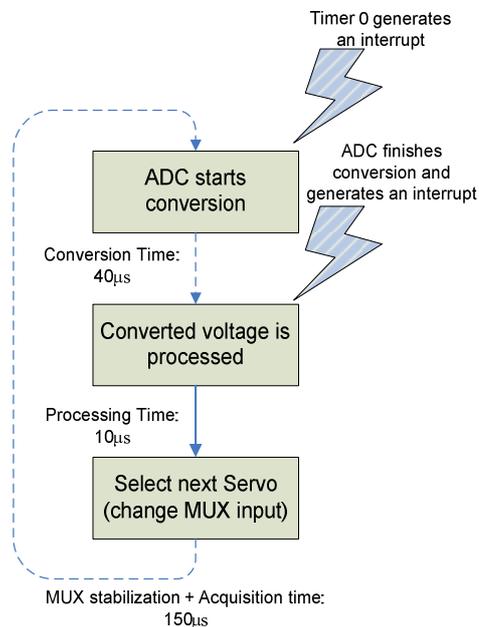


Fig. 14 - Sensors reading sequence.

In order to ensure that the sensor reading interrupts do not interfere with the PWM generation, a simple verification is made before the conversion starts: if there is enough time to execute all steps before a PWM interrupt, the process begins; otherwise, nothing is done and the routine waits for the next timer 0 interrupt.

IV LOW-LEVEL CONTROL

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. 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 Sections 0 e 0.

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 basic idea is 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.

It is expected that enhanced performance can 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. 15 illustrates the block diagram of the proposed servo controller.

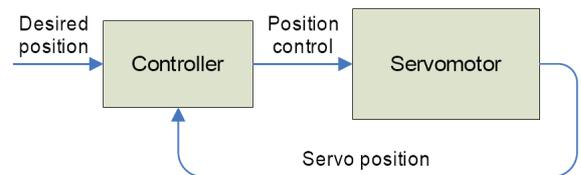


Fig. 15 - Servo controller diagram.

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 behaviour and enhanced trajectory tracking capabilities. That is achieved by a variable PWM throughout the full excursion of 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 case). 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.

A. 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. 16 illustrates, in the z-domain, the proposed control scheme whose control law is described by the following equation:

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

where $K_I = k_i \cdot T_S$, $K_P = k_p$, $K_D = k_d$ are constant positive gains. The resulting control structure is based on the error between the desired joint position $x(n)$ and the measured output position $y(n)$ converted by the ADC.

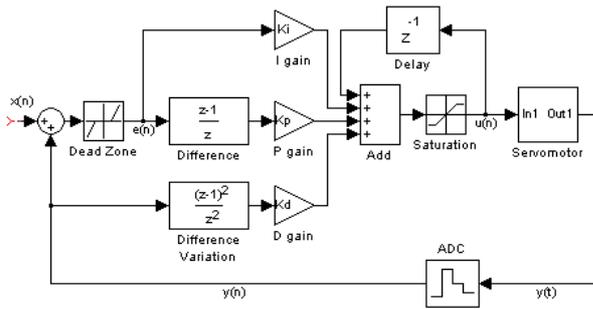


Fig. 16 - Implementation of the incremental algorithm.

B. Integral control

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 $q_i = -45^\circ$ to a final value $q_f = +45^\circ$ 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 Fig. 17 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 represents the output control signal that commands the servo mechanism. This represents the real pulse-width control signal necessary to guarantee the effective conformity between input signal and output shaft position. Fig. 18 compares the trajectory errors of the open and closed-loop control systems.

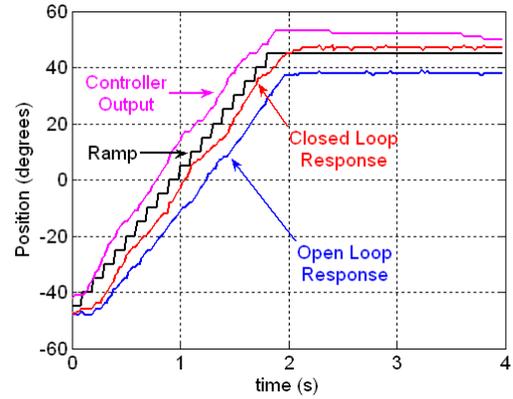


Fig. 17 - Response to a slope input for integral control ($K_I = 0.2$).

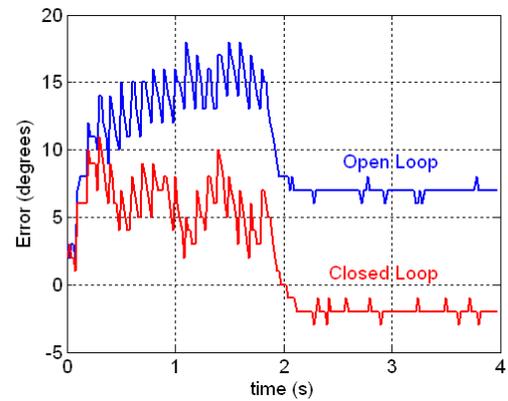


Fig. 18 - Trajectory errors of the open and closed loop control systems.

C. Proportional plus integral control

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. 19 demonstrates the effect of increasing K_I for a fixed proportional term ($K_P = 0.04$). As expected, increasing K_I reduces the lag time improving tracking accuracy, but at the expense of overshoot. Changing K_P to a higher value ($K_P = 0.30$) overshoot is minimized maintaining the lag time for $K_I = 0.10$. From these observations the role of each component can be deduced:

- Integral action reduces time lag at the expense of an increased overshoot;
- Proportional action reduces overshoot, deteriorating the establishment time for very high gains.

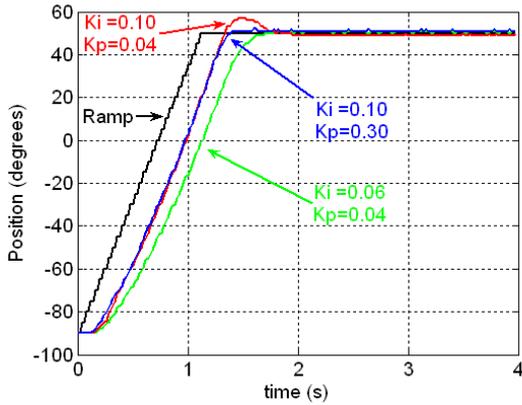


Fig. 19 - Response to a slope input for proportional plus integral control.

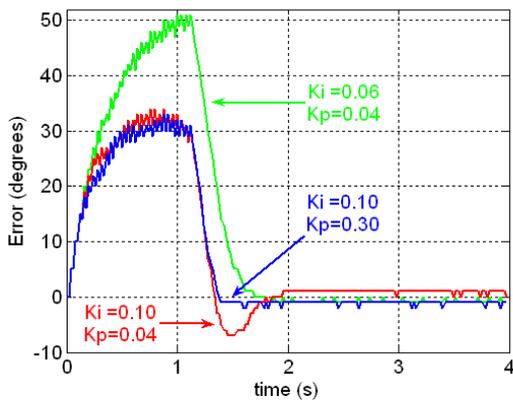


Fig. 20 - Trajectory errors for PI control.

D. PID control

Improvement of the position tracking accuracy might be achieved by increasing the position gain constant K_I controlling the overshoot effects by adjusting K_P . However, for high demands in terms of lag time, compensation tuning becomes very hard due to the presence of unstable oscillations during transient response.

To this purpose, a third experiment is conducted such that the control algorithm is rewritten aimed to include the proportional, integral and derivative terms in order to improve transient response. 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. 21 illustrates the time evolution obtained with the following data: $q_i = 45^\circ$, $q_f = 45^\circ$, $t_f = 1.12$ s. As regards the gains of the outer control loop, these have been

optimized in such a way to limit tracking errors. It can be observed significant improvements in the system's performance: zero steady-state error with no overshoot and limited tracking errors.

E. Double-support phase control

It is now desirable to extend the previous results from the single-axis system to the 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. The humanoid system is actuated by servomotors with reduction gears of low ratios for typically reduced joint velocities.

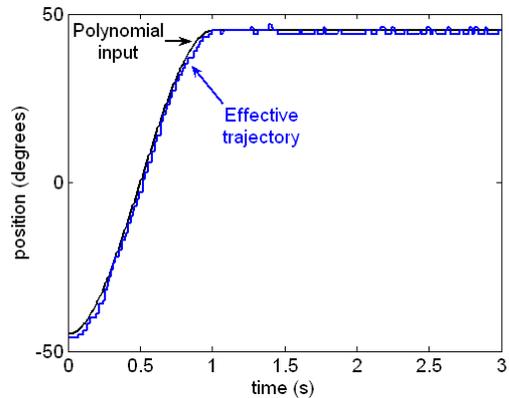


Fig. 21 - Response to a slope input for PID control ($K_P=1.46$, $K_I=0.39$, $K_D=0.15$).

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 (Fig. 22). 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 first trial was to demonstrate the behaviour of the legs during the double-support phase, when performing some basic movements (Fig. 23). 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.

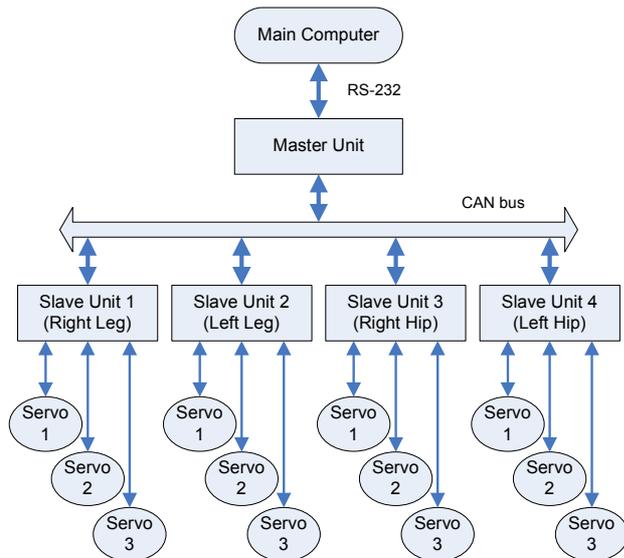


Fig. 22 - Blocks diagram of the overall layout.

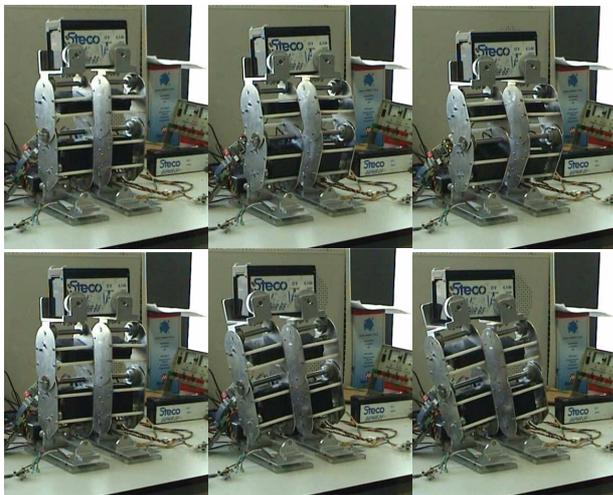


Fig. 23 - Vertical and lateral motions with a 2.1 Kg load attached at the hip section.

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 behaviour 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 behaviour 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. 24 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 behaviour and enhanced trajectory tracking.

F. Additional Improvements

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. At first sight, an improvement of the control performance could be achieved by using a torque control scheme. To this purpose, it is significant to estimate the electrical current which flows in the servomotor.

In order to gain insight into the current consumption evolution during the execution of a given task, several experiments were carried out. The main results are presented for three different cases: static posture, open-loop and closed-loop control system (Fig. 25, Fig. 26 and Fig. 27 respectively). It can be recognized that the time history of the estimated average current shows appreciable variation. At this point, the potential of this measured parameter needs to be exploited. In view of the difficulties concerned with position-velocity control, the problem of torque control will deserve special attention.

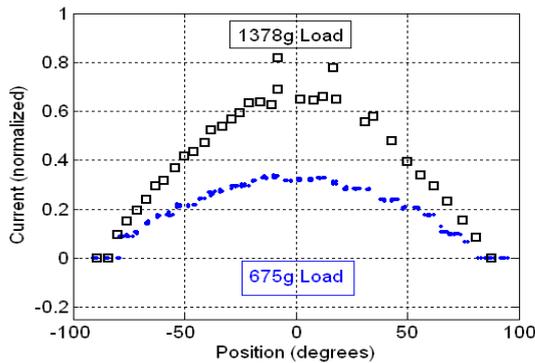


Fig. 25 - Current measurement: static case.

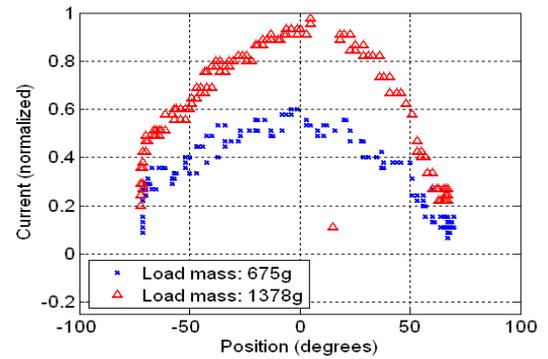


Fig. 26 - Current measurement: open-loop control system.

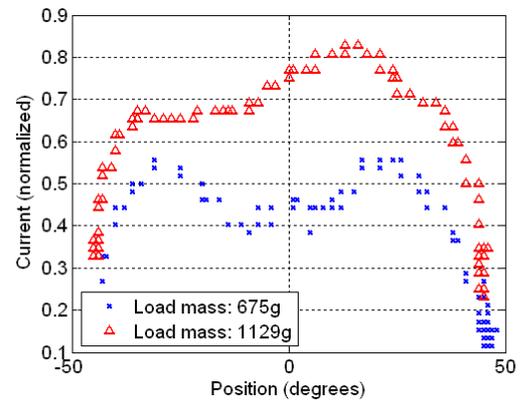


Fig. 27 - Current measurement: closed-loop control system.

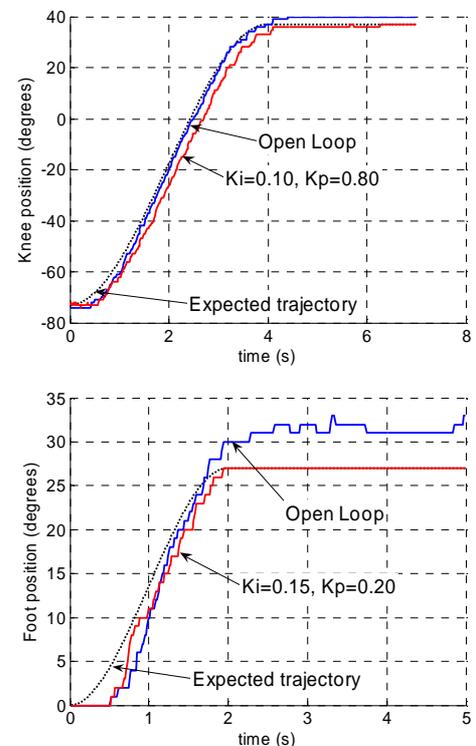
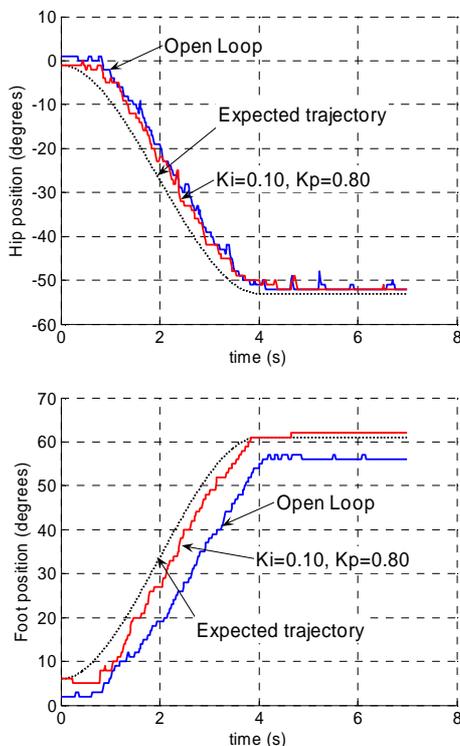


Fig. 24 - Response to slope inputs for a PI controller. Top and left-bottom charts: behaviours of the 3 involved joints during up-down motion of legs. Bottom-right: behaviours of foot joint in lateral motion.

V. FORCE-DRIVEN LOCAL CONTROL

The major problems associated with human-like walking results from the high centre of gravity (COG) with a small contact area to the ground. With other words, balance maintenance is a central concern in order to engage useful tasks, from standing upright posture to motion goals. In what concerns control, the difficulty lies in the uncertainty of the environment and the limitations of the contact between the robot and the environment.

This section shows an example that is being developed to demonstrate the possibility of achieving proper humanoid leg balancing using a local control approach. To this purpose, it is considered feedback control from several sensors, including angular position in each joint and four force sensors inserted into the foot corners. The sensors in the feet provide information about the ground reaction forces and the location of the centre of pressure (COP), as well as about the full contact of the foot with the ground. This opens up new avenues and possibilities for distributed architecture approaches where centralised and local control co-exist and concur to provide robust full monitoring and efficient operation of such a complex system.

A. Humanoid leg balancing

The ability to balance in single support, while standing on one leg, is an important requirement for walking and other locomotion tasks. In the previous section, the approach to balance control assumed the presence of explicitly specified joint reference trajectories and calculations based on static configurations to derive the necessary PWM input signal. The goal of this section is to present the developed control algorithm that provides enhanced robustness in the control of balancing by accounting for the ground reaction forces. Thus, the system is able to stand on an uneven surface or one whose slope suddenly changes (Fig. 28). In a similar way, the control system could sense that it has been pushed, using the force sensors in the soles of its foot, and acts to maintain the postural stability.

The open challenge is to allow local controllers to perform control based on sensor feedback and possibly a general directive. Here, the global order is to keep balance in a desired COP location and, although all actuators can intervene, the ankle joints have the relevant role to keep an adequate force balance on each foot. The controller presents the following key features. First, the force sensors are used to measure the actual COP coordinates, instead of calculating other related variables, such as the centre of mass location. Second, the control system commands the joint actuators by relating the joint velocities (\dot{q}) to the error (e) between the desired and the current position of the COP. The choice of the relationship between \dot{q} and e allows finding algorithms with different performances. The simplest method is the straightforward application of

a proportional law: $\dot{q} = Ke$. The controller is independent of the robot's model or any nominal joint trajectory. This approach has the main advantage of its simplicity: each component of the error vector relates directly and in an independent way to the ankle joints (pitch and roll joints), due to their orthogonal relations.

Alternatively, by interpreting a small displacement in the joint vector as a torque and the error vector as a force suggests the following update law: $\dot{q} = J^T Ke$. Here, J^T is the transpose of the COG Jacobian matrix which transforms the differential variation in the joint space into the differential variation of the COG's position. Suitable values for diagonal gain matrix K can be chosen to ensure convergence and to weight the units of one joint relative to the others (ankle and knee joints). At the same time, a joint velocity saturation function is used to avoid abrupt motions, while satisfying dynamic balance constraints.



Fig. 28 - Humanoid leg and moving platform where balance experiments based on force were done.

B. Experimental results

The following analysis illustrates the emergence of an appropriate behaviour when the system stands on a moving platform. The desired goal is to stand in an initial posture, while the control system relies on the reaction force data to estimate slope changes in the support surface. As stated before, the emphasis in this work is on procedures that allow the robot to calibrate itself with minimal human involvement. Thus, after an initial procedure in which the humanoid leg is displaced to the desired posture, the control system generates online the necessary joint adjustments in accordance with the pre-provided goal. The joint velocity values are computed in real time to modify dynamically the corresponding PWM signal.

Fig. 29 to Fig. 32 illustrate the time evolution of the COP and the ankle joint angle obtained, while the humanoid leg adapts to unpredictable slope changes accordingly to the two control laws presented above. Fig. 29 and Fig. 30 correspond to a COP variation along the xx -axis (sagittal plane), using the proportional and Jacobian-based laws respectively. Fig. 30 and Fig. 32 report to the yy -axis variation (lateral plane). In both cases, the use of the

proposed control algorithm gives rise to a tracking error which is bounded and tends to zero at steady state. This indicates that the posture was adjusted and the differences on the ground reaction forces become small.

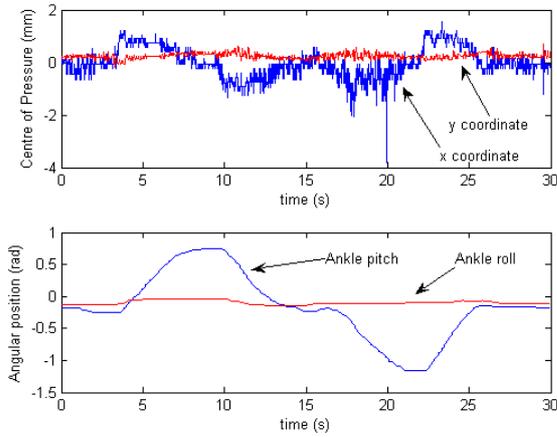


Fig. 29 - Ankle pitch joint predominance: temporal evolution of the centre of pressure (up) and joint angular positions (down) using the proportional law.

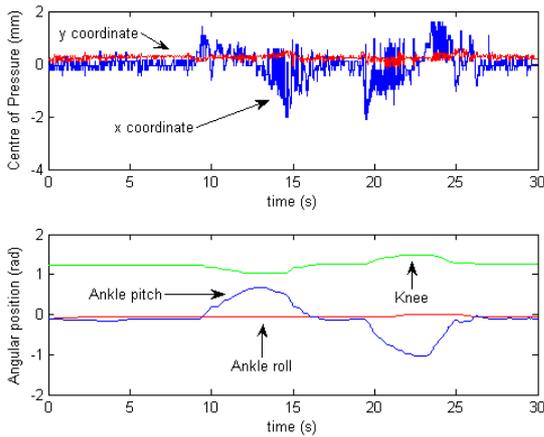


Fig. 30 - Ankle pitch joint predominance: temporal evolution of the centre of pressure (up) and joint angular positions (down) using the Jacobian-based method.

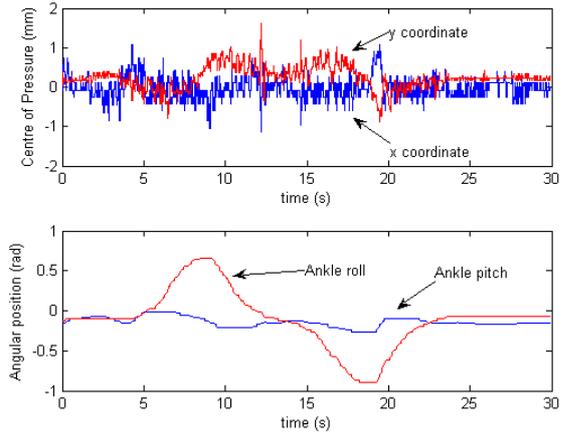


Fig. 31 - Ankle roll joint predominance: temporal evolution of the centre of pressure (up) and joint angular positions (down) using the proportional law.

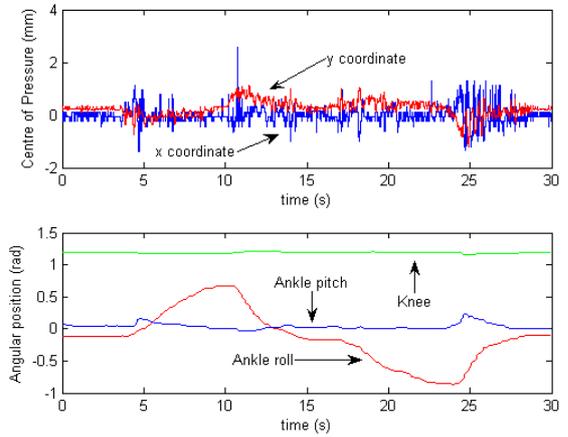


Fig. 32 - Ankle pitch joint predominance: temporal evolution of the centre of pressure (up) and joint angular positions (down) using the Jacobian-based method.

VI. 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 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. An example has been given for a kind of intermediate level control implemented as a local controller based on force sensing.

Most of the final platform hardware has been built and the results are very promising, mainly because many

approaches and research issues suddenly opened and will provide opportunities to test distributed control systems.

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.

REFERENCES

- [1] K. Hirai et al., *The Development of Honda Humanoid Robot*, Proc. IEEE Int. Conf. on R&A, pp. 1321-1326, 1998.
- [2] Y. Sakagami et al., *The Intelligent ASIMO: System Overview and Integration*, Proc. IEEE Int. Conf. Intelligent Robots & Systems, pp. 2478-2483, 2002.
- [3] K. Nagasaka et al., *Integrated Motion Control for Walking, Jumping and Running on a Small Bipedal Entertainment Robot*, Proceedings of the IEEE Int. Conf. on R&A, pp. 3189-3194, 2004.
- [4] F. Yamasaki, T. Miyashita, T. Matsui, H. Kitano, *PINO the Humanoid: A Basic Architecture*, Proc. Int. Workshop on RoboCup, Australia, 2000.
- [5] T. Furuta, et al., *Design and Construction of a Series of Compact Humanoid Robots and Development of Biped Walk Control Strategies*, Robotics and Automation Systems, Vol. 37, pp. 81-100, 2001.
- [6] J.-H. Kim et al. – *Humanoid Robot HanSaRam: Recent Progress and Developments*, J. of Comp. Intelligence, Vol 8, nº1, pp.45-55, 2004.
- [7] V. Santos, F. Silva, *Development of a Low Cost Humanoid Robot: Components and Technological Solutions*, in Proc. 8th International Conference on Climbing and Walking Robots, CLAWAR'2005, London, UK, 2005.
- [8] V. Santos, F. Silva, *Engineering Solutions to Build an Inexpensive Humanoid Robot Based on a Distributed Control Architecture*, in Proc. IEEE/RSJ International Conference on Humanoid Robots, December 5-7, Tsukuba, Japan, 2005.
- [9] Fatlion (2005), The Gigantic Servo Chart, <http://www.fatlion.com/sailplanes/servochart.html>, seen in Feb 2006.
- [10] HITEC (2005), Analog Servos, <http://www.hitecrc.de/store/home.php?cat=309>, seen in Feb 2006.
- [11] ePanorama (2006), Motor Controlling, <http://documents.epanorama.net/links/motorcontrol.html>, seen in Feb 2006.