Parameter Measurement for Speed and Torque Control of RC Servomotors on a Small-Size Humanoid Robot

Milton Ruas[†], Filipe M. Silva[†], Vítor M. Santos[*]
a21824@alunos.det.ua.pt, fsilva@det.ua.pt, vsantos@mec.ua.pt
[†]Department of Electronics and Telecommunications
[*]Department of Mechanical Engineering
University of Aveiro, 3810-193 Aveiro, Portugal

Abstract: Building affordable robots with many degrees of freedom, such as a humanoid, frequently encounters restrictions when actuators are concerned, and most off-the-shelf solutions converge to the servomotors used in radio-controlled models adopted by hobbyists. Being simply a position controlled device with no kind of velocity regulation, these motors require great concerns if other types of control are pursued with minimal intervention on the package. In this paper, procedures are described on how an external microcontroller can read the shaft position and current consumption in order to evaluate intrinsic velocity and torque developed by the motor. As detailed datasheets of these servomotors are not available from the suppliers, extensive experimental behaviour analysis was done, also based on some theoretical models and assumptions. The results show how to measure velocities and current, hence a measure of the torque, after simply pulling out a wire connected to an internal potentiometer. Results also include smooth path planning with controlled velocity, and open a new front for velocity and torque control, which is very important for a humanoid robot based on 22 of such servomotors.

1. Introduction

The design and control of humanoid robots represent one of the more challenging topics in the field of robotics. Progress in the direction of maximizing mobility, speed and efficiency has been difficult due to the high dimensionality, the inherent nature of locomotion and the constraints on actuator systems. Despite the difficulties, several companies have unveiled walking robots with impressive designs and skills, as represented by Honda's ASIMO [1] and Sony's QRIO [2]. At the same time, the continuous progress in robotics technology and the advances in computing hardware have promoted research on low-cost and easy-to-design humanoids, such as PINO [3], ESYS [4] and HanSaRam [5]. The major challenge here is to provide good performance of the control architecture and modularity at the system's level.

In this paper, we describe parts of the control system architecture for a small-size 22 degrees of freedom (DOF) humanoid robot. The research focuses on the distributed control architecture, with the emphasis being placed on how actuators are driven to achieve an improved performance. For the dimensions involved, off-the-shelf actuation technologies do not offer significant alternatives other than small 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 built-in motor, gearbox, position feedback mechanism and controlling electronics.

However, this common method of driving a robotic joint deeply influences the system's performance. First, it is well known that the control of the individual joints of the humanoid robot involves variation of the load inertia. Most certainly, such variations should be taken into account when trying to determine the proper control action; otherwise a decrease in performance will occur. A second problem concerning the mentioned servomotors is that they do not offer directly velocity control. Those servos can be controlled to move to any position just by using simple pulse width modulation (PWM). By design, servos drive to their commanded position fairly rapidly depending on the load, usually faster if the difference in position is larger.

A 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. A further advantage in generating the control signals in each controller unit is represented by 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 and current consumption in order to evaluate intrinsic velocity and torque developed by the motor. The experimental results obtained can be seen as a basis for further speed and torque control improvements.

This paper is organised as follows. Section 2 describes the distributed control approach and the experimental setup. Section 3 presents the main features of the RC servomotors with reference to their limitations. Section 4 analyzes the control problem to enlighten the possibility of achieving a better performance. The paper concludes with a discussion of our results and a proposal for future work.

2. Framework

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. A complete humanoid model and a view of the current stage of development are illustrated in Fig. 1.

The most relevant achievements of this implementation include the distributed control architecture, based on a CAN bus, and the modularity at the systems level [6] [7]. This section is aimed at presenting the features of the control system already implemented and the experimental setup used in the present study.

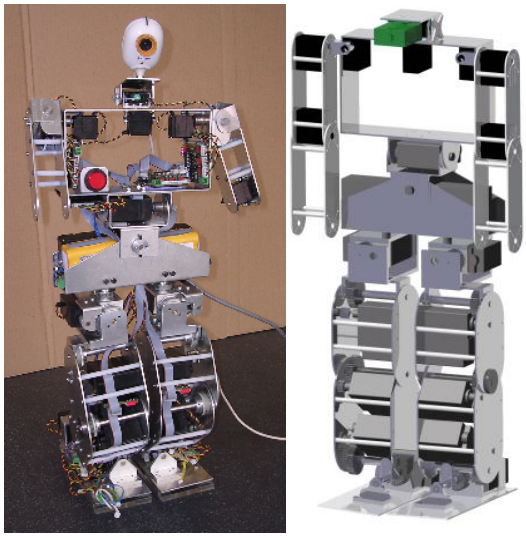

Fig. 1 – The humanoid robot with 22 DOFs

2.1 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 salve arrangement. Master and slave units are based on a PIC microcontroller. Fig. 2 shows a generic diagram of controlling units.

Slaves 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 [7].

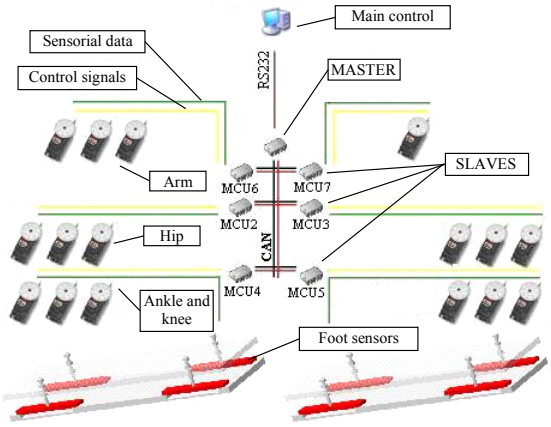

Fig. 2 - General architecture layout

2.2 Experimental Setup

To evaluate the measurements required for the study of the servomotor advanced control, a entire system was set up; 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. Fig. 3 shows a generic diagram of the slave unit. 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 changes.

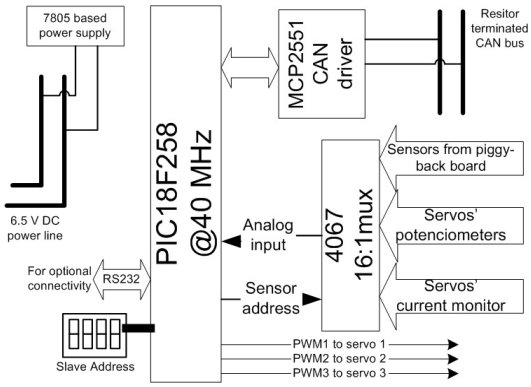

Fig. 3 - Block diagram of generic slave unit

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. Fig. 4 shows photos from 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 1.

Load	Mass (g)	Torque (N.m)
0	9	0.009
1	258	0.253
2	463	0.454
3	675	0.662
1+3	924	0.906
2+3	1129	1.108
1+2+3	1378	1.352

Table 1: Different loads used during experiments and maximum static torque required.

Although the manufacturer indicates a maximum stall torque of 2.42 N.m at 6.0V voltage supply, for loads higher than 1.5 kg (1.47 N.m) the servomotor was unable to perform a complete path between the extreme angular positions (i.e., ±90 degrees).

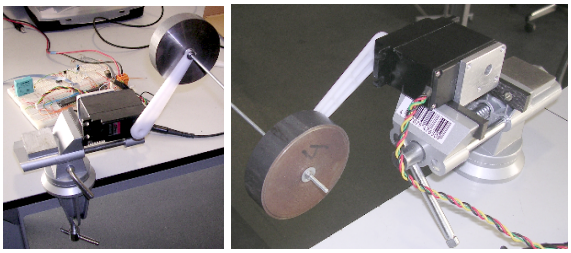

Fig. 4 - Servomotor lifting a calibrated load

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.

2.3 Programming Issues

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 1º. 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 1º.

3. Servomotors and Their Control

3.1 Basics on RC servomotors

For several years now, hobbyists of radio-controlled (RC) models have been using compact motors in packages which consist of the actuator itself (motor), gears, and the unit responsible for position closed loop control; the unit includes a potentiometer and electronic circuits to perform that control.

This type of devices, so-called *RC servomotors* or, for short, servomotors or, so simply, servos, 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 [8].

The servo 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. The internal controller decodes this input pulse and tries to drive the motor up to the required position. However, the controller isn't aware of the motor load and its velocity varies with the load; 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. 5 shows one of the HITEC servomotors used in this project. Table 2 presents its main specifications according to some common information on flyers and datasheet spread over internet sites and vendors [9]. 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 many websites [10].

Fig. 5 - 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)	60º/0.19 s	60º/0.14 s
Stall torque	1.94 Nm	2.42 Nm
Operating angle	45º /one side pulse traveling 400µs	
Direction	clockwise/pulse traveling 1.5 to 1.9 ms	
Current drain	8mA (idle); 700mA (no load running)	
Dead bandwidth	8 µs	
Dimensions	66 x 30 x 57.6 mm	
Weight	152g	

Table 2 - HITEC HS805BB servomotor specifications

3.2 Open-Loop 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 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.

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; it shall be noted further that this process, too, 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.

3.2.1 Response to one large input step

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. 6).

Another observed anomaly in Fig. 6 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 behaviour; a continuous and a fast motion to the final position were observed without speed inversions or oscillations.

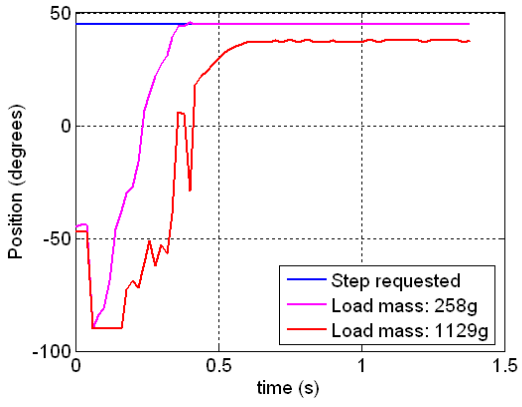

Fig. 6 - Step response for two loads from – 45º to +45º

3.2.2 Response to a "slope" input

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.

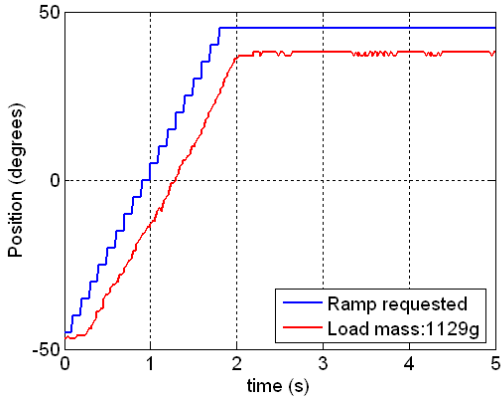

Fig. 7 - Response to a slope input for the highest load

In Fig. 7 it can be seen that, although the transient response has a very improved behavior, 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 where the servo is. Relating the positioning error with the static torque exerted in the joint, $\tau = mg\cos\theta$, where θ is the angular position, m the weight mass and g the gravity acceleration, a direct conclusion can be drawn: the higher the torque, the higher is the steady state error.

Requested position º	measured position º	Error º	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.

3.3 Parameter Measurement

The last results tell us 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 related to the way the potentiometer voltage is read: the issue is related to different grounds for external measurement and internal controller. Therefore, the potentiometer output is consistent with the real servo position only when a very low current is being drained! For high loads (or fast motion) the servo increases the current demands and strangely adds an "extra" output pulse above the position voltage on the potentiometer. This occurs perfectly synchronized with the input PWM signal (Fig. 8). 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.

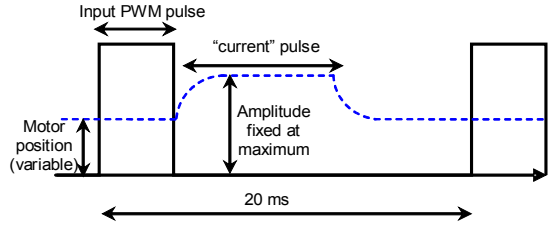

Fig. 8 - Motor position at the potentiometer

With the assumption that the internal controller uses a power bridge and internal PWM synthesis to push the motor, it is reasonable to accept that the "extra" pulse in the potentiometer level is due 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 potentiometer level has a fixed top value. 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, or 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. 6: the low peaks in the transient behaviour correspond to high peaks in 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 120µs 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. Thus, during 19ms the potentiometer voltage is read, updating continuously its minimum value and counting the number of cycles (of 120µs) 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 position is made through a multiplexer whose selected input must be switched, requiring therefore an extra delay fundamental to assure a stable output. Considering 10µs for the associated code execution, 20µs for the acquisition time and 40µs for the conversion time, 50µs 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 360µs to read all three sensors. The sequence is illustrated in Fig. 9.

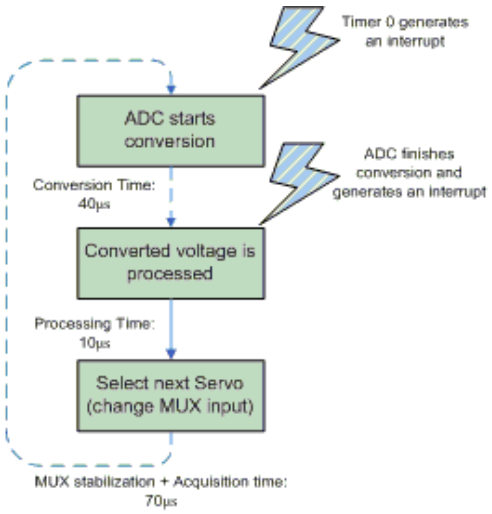

Fig. 9 - Sensors reading sequence.

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. Fig. 10 shows the temporal organization of the interrupts.

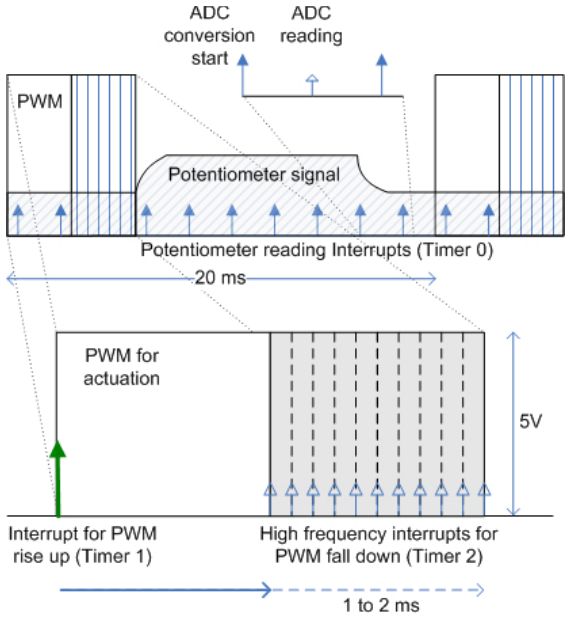

Fig. 10 - Temporal organization of the interrupts.

4. Servo Control Approach

On the basis of the above limitations, two kinds of possible solutions could be devised. On one hand, the tendency 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. 9 illustrates the block diagram of the proposed servo controller.

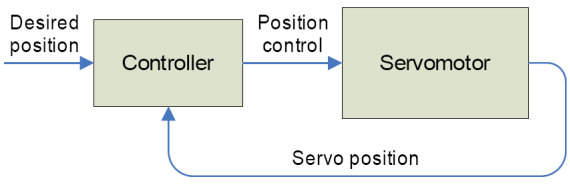

Fig. 11 - Servo controller diagram

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 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 pause 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.

4.1 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. 12 illustrates, in the z-domain, the proposed control scheme whose control law is described by the following equation:

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

here $K_I = k_i \cdot T_S$, $K_P = k_p$, $K_D = k_d$ 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.

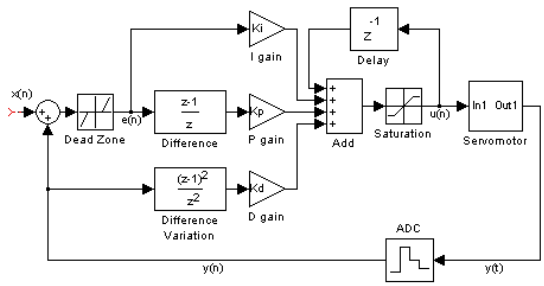

Fig. 12 - Implementation of the incremental algorithm

4.1.1 *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 $q_i = -45^o$ 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 Fig. 13 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. 14 compares the trajectory errors of the open and closed-loop control systems.

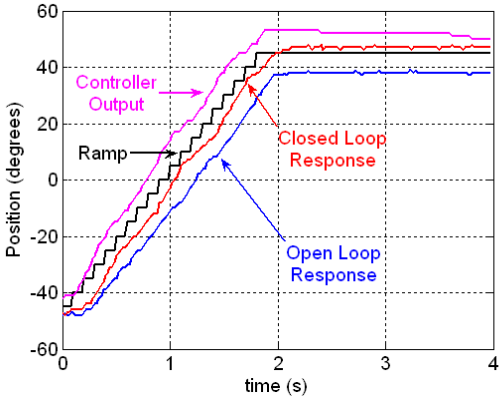

Fig. 13 - Response to a slope input for integral control only ($K_I = 0.2$)

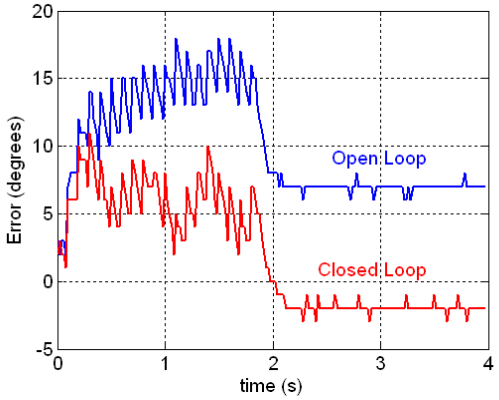

Fig. 14 - Trajectory errors of the open and closed loop control systems

4.1.2 *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. 15 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 (Fig. 16). 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.

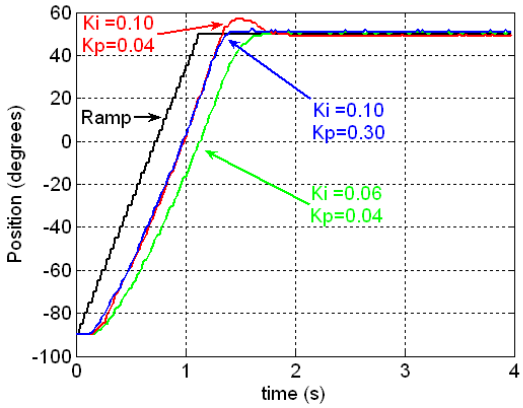

Fig. 15 - Response to a slope input for proportional plus integral control

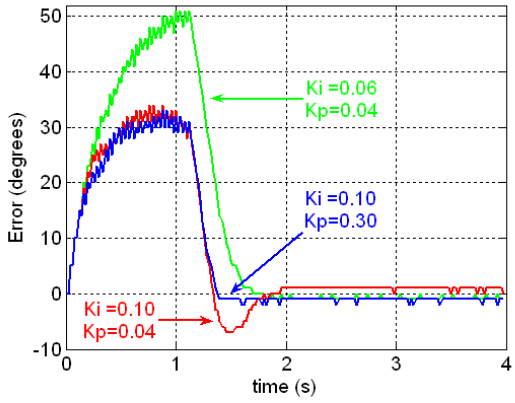

Fig. 16 - Trajectory errors for PI control

4.1.3 *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.

Figure 15 illustrates the time evolution obtained with the following data: $q_i = 45º$, $q_f = 45º$, $t_f = 1.12$ 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.

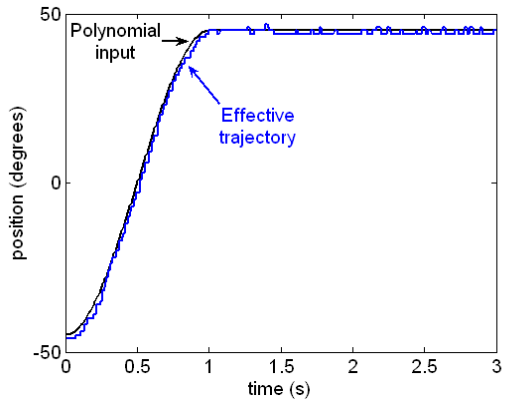

Fig. 17 - Response to a slope input for PID control ($K_P = 1.46$, $K_I = 0.39$, $K_D = 0.15$)

4.2 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 (Figs. 18, 19 and 20, 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.

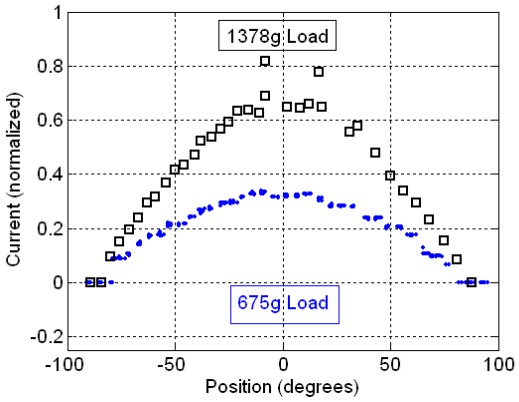

Fig. 18 - Current measurement: static case

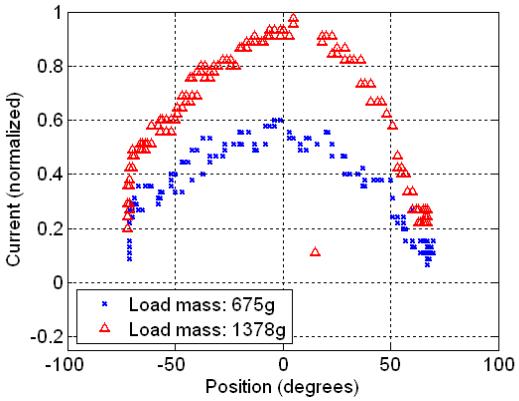

Fig. 19 - Current measurement: open-loop control system

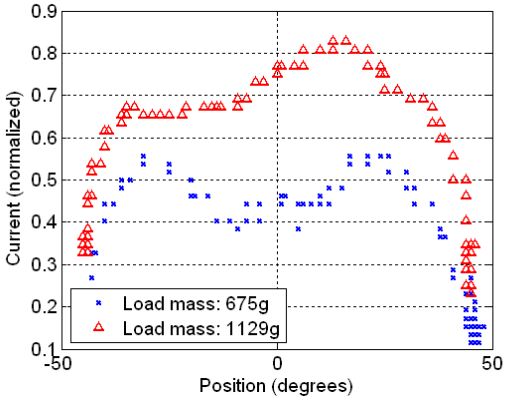

Fig. 20 - Current measurement: closed-loop control system

5. Conclusions

This paper described methods to perform position, velocity and current measurements in RC servomotors without hardware modifications. These parameters allow for more advanced motor control which is required for complex robots, such as the one in the current case: a 22-DOF humanoid. The results showed that velocity control and also trajectory planning is possible with such devices.

Further, by having a process to evaluate average current, which was also described, and relying also on velocity measurement, a door opens to perform some kind of torque control on these servomotors. This will allow highly robust control strategies of complex robots by means of simple RC servomotors. Therefore, issues like automatic gain, gain scheduling, continuous adaptation and variable structure control are being currently challenged.

References

[1] Y. Sakagami et al., "The Intelligent ASIMO: System Overview and Integration", Proc. IEEE Int. Conf. Intelligent Robots & Systems, pp. 2478-2483, 2002.

[2] 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.

[3] F. Yamasaki, T. Miyashita, T. Matsui, H. Kitano, "PINO the Humanoid: A Basic Architecture", Proc. Int. Workshop on RoboCup, Australia, 2000.

[4] 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.

[5] J.-H. Kim et al. –"Humanoid Robot HanSaRam: Recent Progress and Developments", J. of Comp. Intelligence, Vol 8, nº1, pp.45-55, 2004.

[6] 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.

[7] 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.

[8] Fatlion (2005), The Gigantic Servo Chart, http://www.fatlion.com/sailplanes/servochart.html, seen in Feb 2006.

[9] HITEC (2005), Analog Servos, http://www.hitecrc.de/store/home.php?cat=309, seen in Feb 2006.

[10] ePanorama (2006), Motor Controlling, http://documents.epanorama.net/links/motorcontrol.html, seen in Feb 2006.