Variable Speed Control … for a Compact Humanoid Robot

Milton Ruas[†], Filipe M. Silva[†], Vítor M. Santos[*]
`a23570@alunos.det.ua.pt, vsantos@mec.ua.pt, fsilva@det.ua.pt`
[†]Department of Electronics and Telecommunications
[*]Department of Mechanical Engineering
University of Aveiro, 3810-193 Aveiro, Portugal

Abstract: no fim ☺

1. Introduction

In recent years, the field of humanoid robotics has attracted the attention of a growing community, both from the industry and academia. It becomes increasingly evident the dichotomy in the styles used to design compact humanoid robots. On the one hand, several companies have unveiled walking robots with impressive designs and skills, as represented by Honda's ASIMO [1] and Sony's QRIO [2]. On the other hand, 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 [4]. Here, the major challenge 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 a desired 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 can deeply influence 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 a key element to implement an adaptive scheme that compensates for the large changes in reflected inertia and providing variable velocity control.

Instead of changing the motor internals, as some other authors do, it was decided to improve its operation by software. 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 pause motor motion (the loop is closed back to the controller). Further, generating the control signals in each controller unit will reduce significantly the overhead on the controlling software.

2. Project Overview and Framework

Comentário [vs1]: Esta secção 2 é provisória e poderá ser substituida ou ser rescrita.

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, and also to offer opportunities for under and pos-graduate students to apply engineering methods and techniques in such ambitious and overwhelming endeavour. Purchasing a commercial platform carries prohibitive costs and it would reduce the involvement at the lowest levels of machine design, which was posed as a relevant pursuit for the desired engineering approach.

The ultimate goal of the project 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, giving added value for project-oriented education.

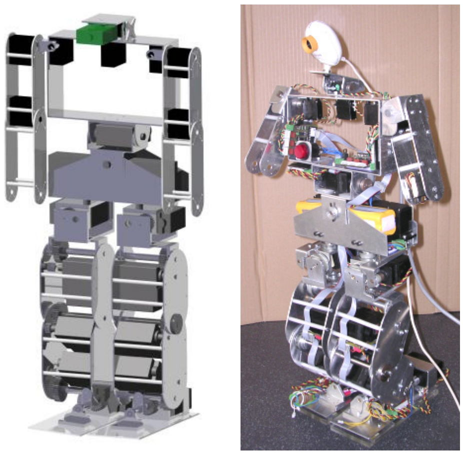

Fig. 1 - Model of the humanoid robot and current stage of implementation.

The most relevant achievements of this implementation include the distributed control architecture, based on a CAN bus, and the modularity at the system's level.

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. shows a generic diagram of controlling units.

The master unit relays all slave units by dispatching medium and high level orders and by collecting sensorial data to be exchanged with the central unit. The central unit is currently an off-board computer but will be migrated to a local controller based on a PC104+ board with image processing capability.

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 to other sensors (e.g., force-sensors, accelerometers and gyroscope).

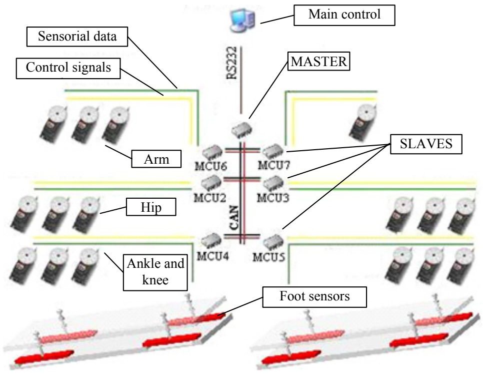

Fig. 2 - General architecture layout.

2.2 Experimental Setup

Now, we will describe the setup components and its most relevant aspects in order to understand the conditions present in the experimental procedures.

The main entity of the experimental setup is the servomotor since they will be used to control the robotic joints motion. RC model servos are fairly sophisticated devices that incorporate both position and speed feedback with a goal to provide precise position control. In normal use they compare the 1-2 ms, 50 hertz (50 Hz, 5-10% duty cycle) input pulse signal with an internal linear pulse generator controlled by the feedback servo position potentiometer (pot.) and by the motor back-EMF (voltage generated between power pulses) which is used as a speed sensor (Fig.3). The difference in pulse width, known as the error signal, is then "amplified" with a pulse stretcher. This circuit provides the servo control gain. The pulse stretcher output drives the servomotor through an H-bridge circuit to close the servo loop. The speed sensor feedback is normally used only to stabilize the position of the servo so the servo drives quickly to its commanded position with minimal overshoot. Fig.3 shows a Futaba S3003 circuit diagram along with the circuit functions believed to be in the BA6688. Although it is not the model used in our experimental setup (HITEC HS805BB) we believe that its main functioning aspects are the same due to its similar behaviour and product specifications.

An aspect that is worth to point is the PD controller type used to manipulate the servo motion. The proportional module uses the potentiometer error position to control the servo final position, and the speed sensor constitutes the derivative module in order to minimize overshoot. Due to the absence of the integrative component it is to expect a steady state error when moving high loads.

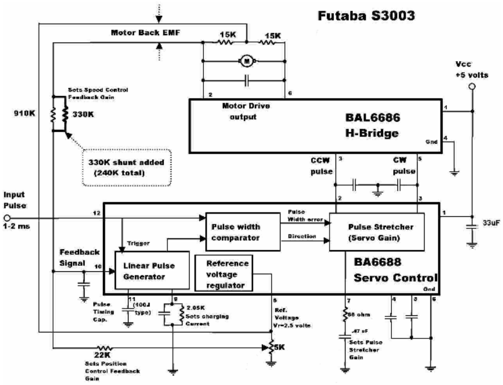

Fig. 3 - Futaba servomotor schematics.

In order to use the servo position feedback signal in a potential external controller, an extra output wire was connected to the potentiometer output whose signal will be used by a slave microcontroller unit to correct undesired deviations from the ideal response. In contrast with the Futaba S3003 servo, HITEC's potentiometer doesn't possess the same voltage reference (ground) as the servo itself, and so too, the microcontroller ADC converter used to measure its output voltage. Had to this, the potentiometer measured signal doesn't always correspond to servo position, introducing some offset deviations. By experimental observations we concluded that these deviations were related to the consumed current. As the H-bridge functioning uses an impulse input, its consumed current will follow the same behavior: a digital impulse using only a section of the PWM period. Because of that, potentiometer output, according to servo reference voltage, will present an offset deviation during current consumption, with the standard voltage (related to the servo position) in the remaining time (see Fig.5). An advantage can be taken from this behavior: as servo position can be measured from potentiometer output minimal voltage, current can also be known from the same output by its impulse length measure.

In the experimental procedures a mass load is applied indirectly to the servo shaft through an arm of 10 cm length recreating the torque appliances in the humanoid joints. To estimate the torque exerted, the servo was fixed in a mechanical lathe in which its neutral position (predefined to be the zero degrees position) corresponds to the maximum applied torque (gravity vector and link perpendicular between each other). On the other hand, the extreme locations (±90 degrees position) correspond to the null torque appliance. To test a wide range of torques, a set of weights were used, whose mass is indicated below:

- Mass 0: 0g (no load)
- Mass 1: 249g
- Mass 2: 454g;
- Mass 3: 714g
- Mass 4: 666g

Considering that the connection screw between the load and the arm link weights about 9g, the tested loads combines the weights above in the following way:

Weights combination	Weights mass (g)	Effective load (+9g)	Maximum torq (N.m)
Mass 0	0	9	0.009
Mass 1	249	258	0.253
Mass 2	454	463	0.454
Mass 4	666	675	0.662
Mass 4+1	915	924	0.906
Mass 4+2	1120	1129	1.108
Mass 4+2+1	1369	1378	1.352

Table 1: Set of loads used during experiments.

Although the manufacturer indicates a maximum stall torque of 2.42 N.m at 6.0V voltage supply, in practice for loads higher than about 1.5 Kg (1.47 N.m), the servomotor was disabled to realize all traject between position extremes (±90 degrees). That is why the tested maximum load torque is only 1.352 N.m.

To recreate the same distributed architecture between microcontroller units in the humanoid robot, a master-slave system, connected themselves through a CAN bus, was used (Fig.4), in which the slave unit is connected to the servo mechanism, using as output the servo control position signal and as input the potentiometer feedback signal, and the master unit is connected to a computer, through a RS-232 link, using MatLab software as interface. Fig.5 shows a photo of the experimental setup used to test the servomechanism.

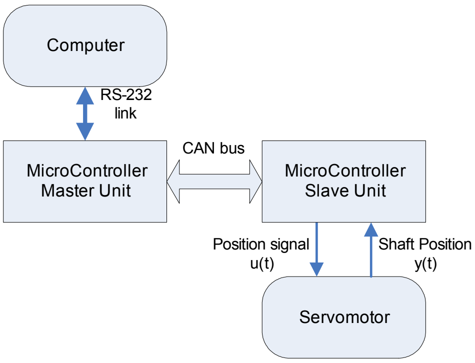

Fig. 4 - Experimental setup diagram.

Comentário [MRS2]: Adicionar referência: http://www.seattlerobotics.org/encoder/200009/S3003C.html

3. Servomotors and Their Control

3.1 Open-Loop Performance

- Load variation (step response)
- Set point control (table and graphs)
- Speed and current versus torque

We will now study the servo behaviour using only its internal controller to reach the set point requested. When we refer open loop state, we refer in the servo outside point of view, without introducing any external controller. We will start to study its behaviour to step inputs.

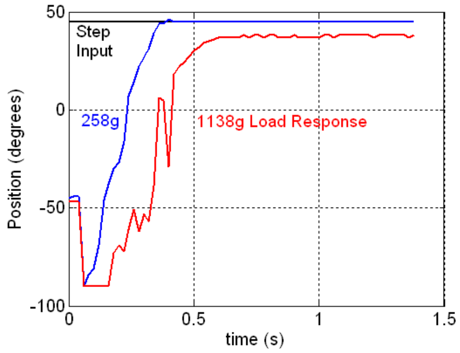

Fig. 5 - Step response for two loads from the position -45° to +45°.

Applying a step from -45° to +45° position for different loads, the first distinctive difference is the steady state errors. For the low mass the steady state error is practically negligible, but for the 1129g load we can measure a 8° error. Another characteristic observed is the unstable dynamic behavior with a jump to a position below of -45° and some oscillations during travel to the final set point. Nevertheless these conditions were not observed in reality showing a continuous and a fast motion to the final position without speed inversions nor oscillations. This tell us that when carrying out long trajectories at the maximum speed (characteristic in step responses) the potentiometer output tend to be unstable when increasing the load mass. This is related to the way this sensor is read: due to the not common ground in potentiometer measure on the servo side, its output signal only is consistent with the real servo position only when a very low current is being consumed. For high loads or fast movements the servo increases its current consumption and introduces an output impulse of amplitude above its position voltage and synchronized periodically with the PWM signal, whose length is directly related to the current consumed (Fig.4).

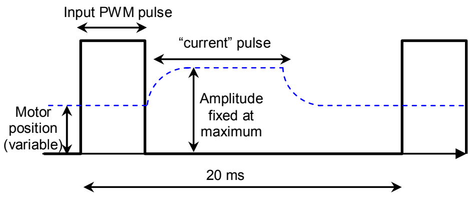

Fig. 6 - Potentiometer voltage behavior in a PWM cycle.

To avoid measure interference, the potentiometer is sampled several times in each PWM period and only the minimal value is taken into account. Nevertheless for high loads the current impulse can cover all PWM period incapacitating position reading. That is what happens in Fig.3: the low peaks in the transient behavior correspond to high peaks in voltage caused by the current impulse interference. To avoid this kind of problems and guarantee the occurrence of finite current impulses so as to be able to measure the real potentiometer output, from now on we will only study the ramp response, a succession of steps of constant amplitude increase over time, until final position is reached. This way the current consumption will only practically depend on the load torque exerted, because of the speed limitation introduced by the ramp input, and therefore the levels of current consumed will be lower. In addition, beyond the position control, velocity control is introduced by the definition of the ramp length.

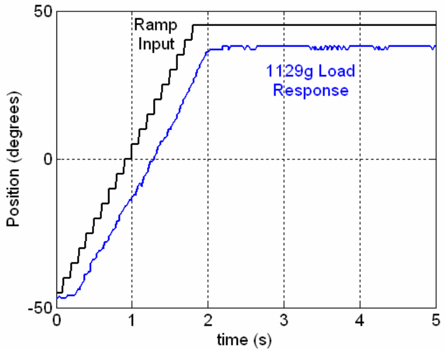

Fig. 7 - Ramp response for the highest load.

Although the transient response has a very improved behavior, the steady state error continues present. Table 1 shows the results of an experiment realized to study this effect: a weight is requested to move to a set of defined positions, and for each one, we wait the completion of the movement and, afterwards, the potentiometer is sampled to obtain the real position where the servo is. Relating the positioning error with the tork exerted in the joint – $\cos(\theta)*m*g$, whith θ being the weight angular position, m its mass and g the gravity acceleration – we find a notorious relation between them: as higher is the torq, higher is the steady state error. This lead us to think that the servo internal controller does not possess an integrator module which would bring the steady state error to zero, independently of the existing torque.

Requested position	Effective position	Error	Torq exerted
-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 2: Steady state error and torq exerted in the joint for a fixed set of positions using a 1138g load.

To correct these deviations from the ideal behavior, namely null steady state error and a low delay time in transient response, an external controller becomes necessary to improve these characteristics (Fig.6).

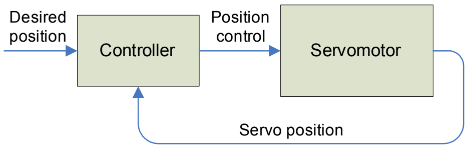

Fig. 8 - Servo controller diagram.

Adaptive Controller

As we can see, the servo behavior differs with the load angular position which makes it very hard to control having to considerer the beginning and final position to define a proper controller to adapt to each situation. Doing a theoretical analysis to our system we can understand where these complications are owing to. Let's considerer a weight of mass *M* connected to a movable joint through a link of length *L*.

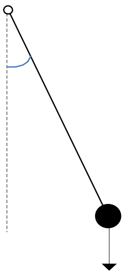

Fig. 9 - System weight plus link connected to a movable joint.

The system dynamics is given by the following equation where *c* is the viscous friction coefficient, *g* is the gravity acceleration and τ is the binary applied to the joint:

$$\tau = \ddot{\theta} + \frac{c}{M \cdot L^2}\dot{\theta} + \frac{g}{L}\sin(\theta)$$

As can be seen, this system has a non linear dynamics owing to the sin(θ) factor, making unappropriated the use of linear controllers. This is why the servo internal controller is inefficient to handle to all possible situations: its internal controller is based on a PD type which is only efficient to deal with linear dynamics. In some circumstances, the factor sin(θ) can be approximated to θ, making suitable the use of linear controllers, nevertheless such approximation would limit θ variation to some degrees around an established reference angle, which is not the case.

As a result, as far as the mass approximates the maximum binary point (link and gravity force vector perpendicular) the response delay time and steady state error tend to increase deteriorating the controller's response. In the same way, these parameters are also related to the mass applied rising as far as we increase its weight.

3.2 Position Feedback Control Loop

In order to minimize the steady state error and improve the transient response by the reduction of the delay time, an external controller was added with the desired and the servo feedback position as inputs and the servo control position as output (Fig.6). Its model is described in Fig.8 and is based in a digital PI controller, whose proportional control is represented by a simple K_P gain and the integrator by the error sum module.

Comentário [MRS3]: O controlador usado é inteiramente equivalente ao apresentado, pelo que, por motivos de fácil compreensão, acho que é melhor apresentar o formato tradicional.

A partir daqui todas as referências a K_P mudam para K_I e os K_D mudam para K_P.

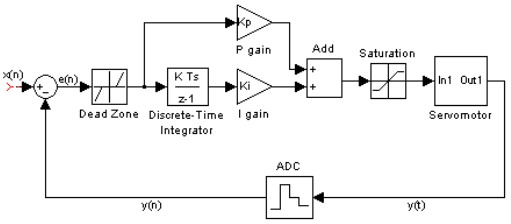

Fig. 10 - External controller diagram blocks.

The desired position $x(n)$ and the measured position $y(n)$ (converted by the ADC) are compared and the result constitutes the error signal $e(n)$. To this signal is applied an integrative compensation to clear the steady state error and a proportional gain fundamental to improve the transient response. The following differential equation describes the controller's functioning:

$$u(n) = K_I \cdot \sum e(n) + K_P \cdot e(n)$$

Note that this equation represents a digital PI whose linear nature doesn't solve the non linearity problems. However it offers the possibility to adapt its behavior to each specific scenario by direct modification of its parameters, something not possible with the servo internal controller.

3.3 Adaptation to Load Variation

- Current measurement
- Adaptive algorithm added to 4.1
- Step response (overshoot, steady-state error!)

Several tests were made to prove the viability of this external controller in terms of improvement of response. Fig.9 compares open loop with closed loop response for a high load using only the proportional component.

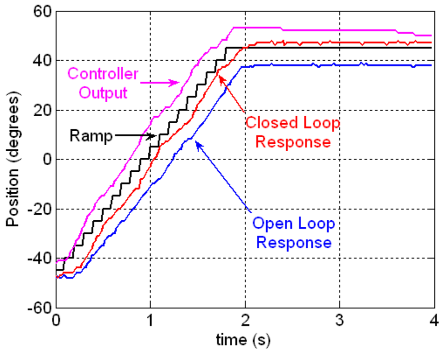

Fig. 11 – Open and closed loop ramp response for K_I=0.20 (924g load).

As can be observed, the steady state error is eliminated and the delay time is reduced when applying a pure integrative compensator. The green curve shows the controller output signal that enters in the servo mechanism with a higher position requisition than the desired in order to guarantee the effective position corresponds to the requested one.

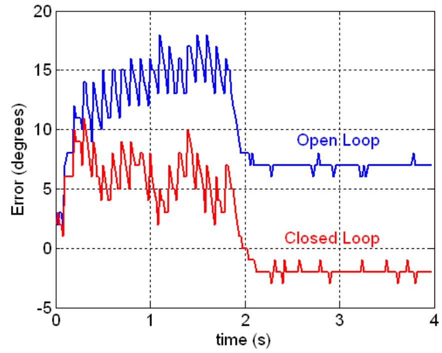

Fig. 12 – Open and closed loop error signal for K_I=0.20 (924g load).

Fig.10 shows the difference between the error signal in open and in closed loop confirming the proposed idea. The next figures introduce the proportional component as a reinforcement to improve even more the response quality. To demand a more exigent input we shrink the ramp step time update to the minimum – 20ms (PWM period) – with an amplitude update of 5 degrees. Defining a trajectory variation of 140 degrees (-90º to +50º) we will request the completion of the traject in only 1.12 seconds.

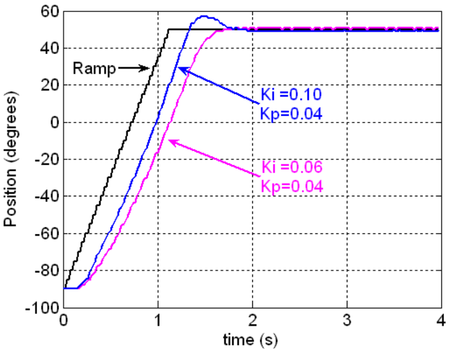

Fig. 13 – Closed loop ramp response for different K_P parameters (924g load).

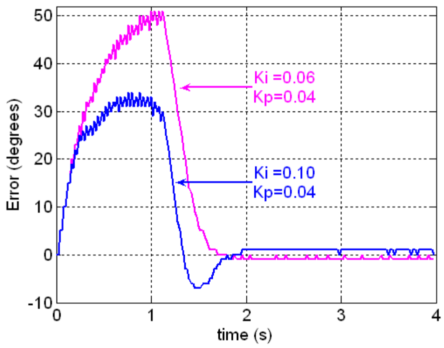

Fig. 14 – Closed loop error signal of Fig.11.

Sampling the response for different K_I parameters maintaining fixed K_P can be seen that K_I affects the response delay time being so much closed to the requested trajectory as higher it is. Nevertheless it has to be limited to avoid instability which is reflected in overshoot and higher establishment times. Fig.11 and Fig.12 shows such situation when we increased K_I from 0.06 to 0.10.

However, manipulating K_P we can eliminate the overshoot and reduce the establishment time by its increase. Fig.13 confirms this idea: increasing the K_P parameter the overshoot verified in Fig.12 were successfully controlled.

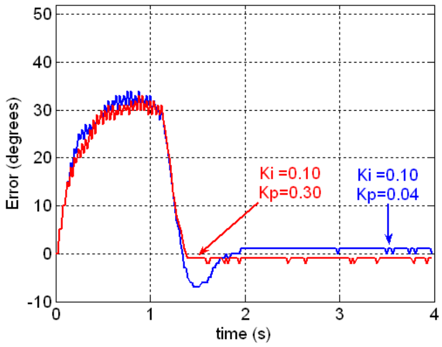

Fig. 15 – Closed loop error signal for an increased K_I parameter.

In respect to the steady state error, the previous graphs proved its absence observing a maximum error of 3 degrees, derived to the “dead zone” block (Fig.8) necessary to prevent oscillation transmission to the controller blocks verified on the potentiometer reading. Note that the controller parameters used were adapted to this specific scenario having to be updated if such scenario is modified, due to the non linear dynamics.

3.4 Variable Velocity Control

An important subject in humanoid locomotion is trajectory planning, the ability to generate a trajectory from a collection of set points. Although we already were implementing trajectories using a ramp configuration (Fig.14) with a final position and mean velocity as set points, a huge disadvantage was present in this configuration: at the start and at the end of the trajectory a Delta of Dirac is expected in the acceleration curve, something not possible on reality, which contributes to increase the servo response delay time and the overshoot.

To prevent this effect, instead of a pure ramp, we can generate a third polynomial configuration trajectory (Fig.15) which differentiates, by approximation, three stages: a first with a low velocity, a second stage of greater and practically constant velocity, and a final one of deacceleration until the servo stops. In this configuration there is no Deltas of Dirac in the acceleration curve and a soft locomotion is verified during trajectory execution, with the extras of low response delay and establishment times.

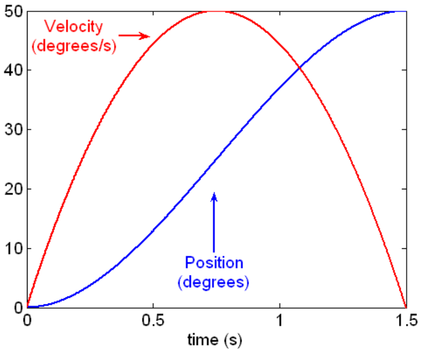

Fig. 16 - Third order polynomial configuration trajectory.

Fig.15 shows the results to the same experiment of the previous section, but applying a 3^{rd} order polynomial trajectory using the same set points (position extremes and execution period).

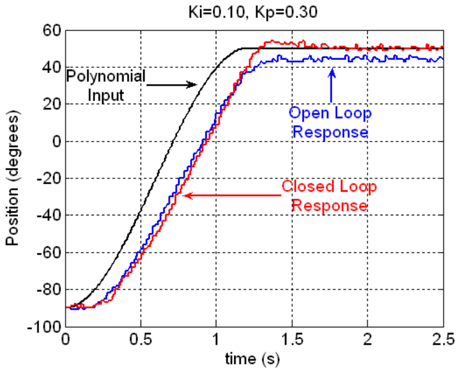

Fig. 17 - Polynomial traject execution of Fig.9.

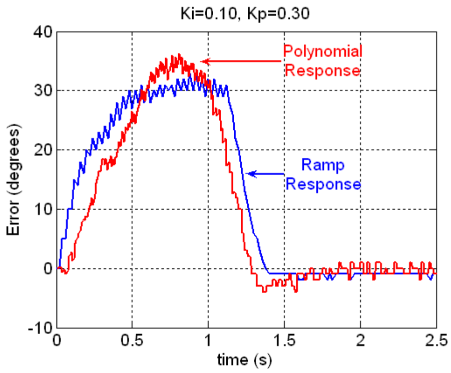

Fig. 18 - Closed loop error comparison between traject configurations for a KI=0.10 and KP=0.04.

Comparing the experimental results relatively to the ramp case (Fig.16 versus Fig.13) we verify a decreased delay time at the trajectory beginning and at the end motivated by the low velocity value, however at the middle the delay time surpasses the ramp value. Note that the trajectory execution period is equal for both configurations resulting in a superior velocity for the polynomial response during intermediate stage, which contributes to increase the delay time. Even so, we get a lower establishment time for the polynomial response with K_P properly tuned. Although we have to increase a little bit the traject period relatively to ramp configuration, due to the velocity reached in the intermediate stage, the lower general delay and establishment times allied to the soft locomotion, makes this trajectory configuration preferable for our needs.

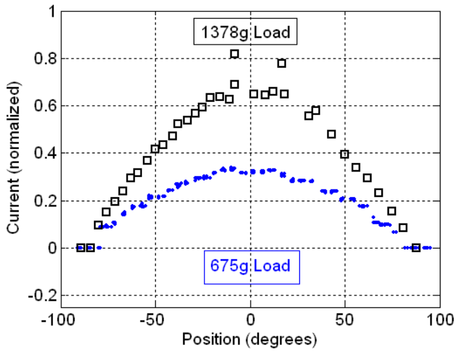

Fig. 19 - Current consumption static measure (upwards direction).

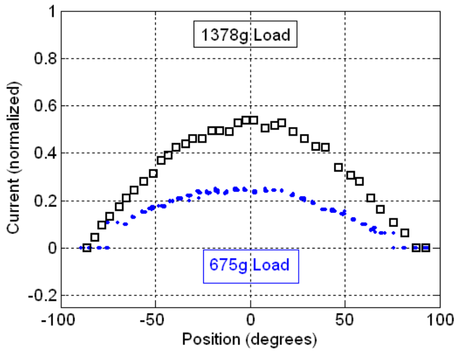

Fig. 20 - Current consumption static measure (downwards direction).

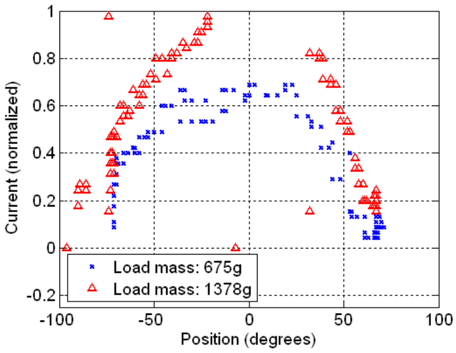

Fig. 21 - Current in open loop when performing a polynomial trajectory from -70º to +70º in 1.0 seconds.

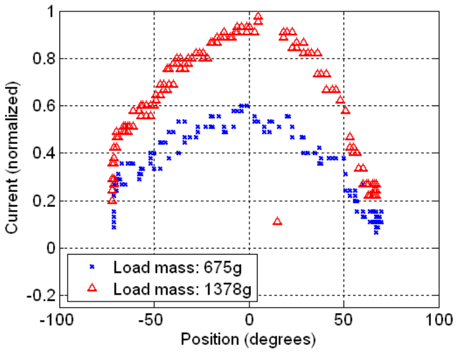

Fig. 22 - Current in open loop when performing a polynomial trajectory from -70º to +70º in 1.5 seconds.

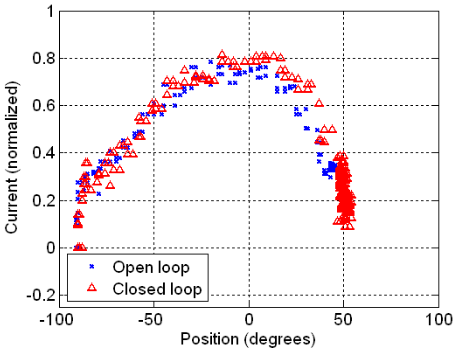

Fig. 23 - Current consumption for the situation of Fig.15 (1129g load).

4. Conclusions

Referir a capcidade de controlo de velocidade a partir do controlo de posição.

Referir também o reenforço de estabilidade no controlo com a aplicação de trajectórias polinomias de 3ª ordem.