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

Milton Ruas[1]
Filipe M. T. Silva[1]
Vítor M. F. Santos[2]

[1] Department of Electronics and Telecommunications
[2] Department of Mechanical Engineering
University of Aveiro, PORTUGAL

Centre for Mechanical Technology and Automation ▪ TEMA
Institute of Electronics Engineering and Telematics ▪ IEETA

➢ http://www.mec.ua.pt/robotics

Overview

- Introduction
- Humanoid Platform: Overview

- Actuation Element: the Servomotor
- Experimental Setup
- Speed Control
- Open and Closed Loop Performance
- Torque Measurement

- Conclusions and Open Issues

Introduction

Project 's Motivation

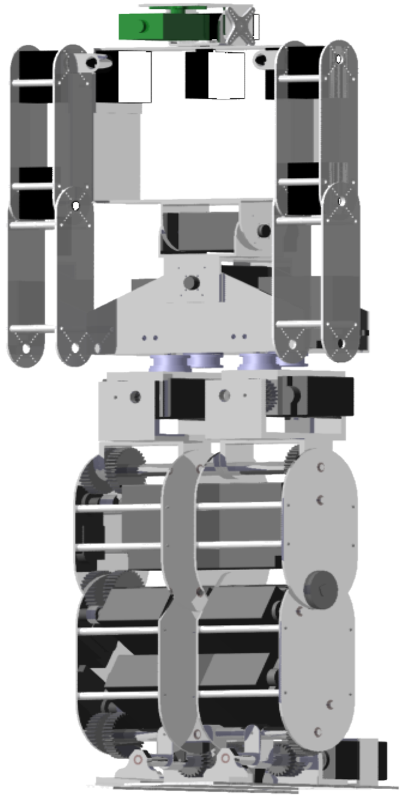
- Develop a humanoid platform for research on control, navigation and perception
- Offer opportunities for under & pos-graduate students to apply engineering methods and techniques
- The utopia of Man to develop an artificial being with some of its own capabilities…

Objectives

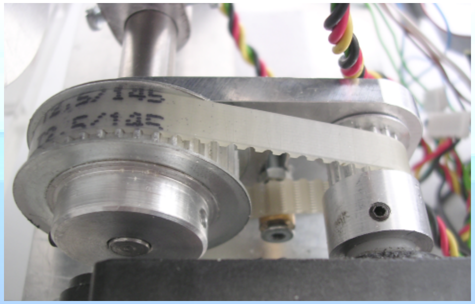
- Describe how an external microprocessor can read the shaft position and current consumption of RC servomotors
- Improve servo's performance towards a control system that provides variable velocity and compensates for load variations
- All this with minimal hardware intervention:
  - Feedback is provided to introduce suitable compensation control actions via the closure of an outer control loop

Humanoid Platform

- Complete humanoid model
  - 22 degrees of freedom
  - Weight - 5 *kg*
  - Height - 60 *cm*
  - Max. width - 25 *cm*
  - Foot print - 20 × 8 (cm^2)

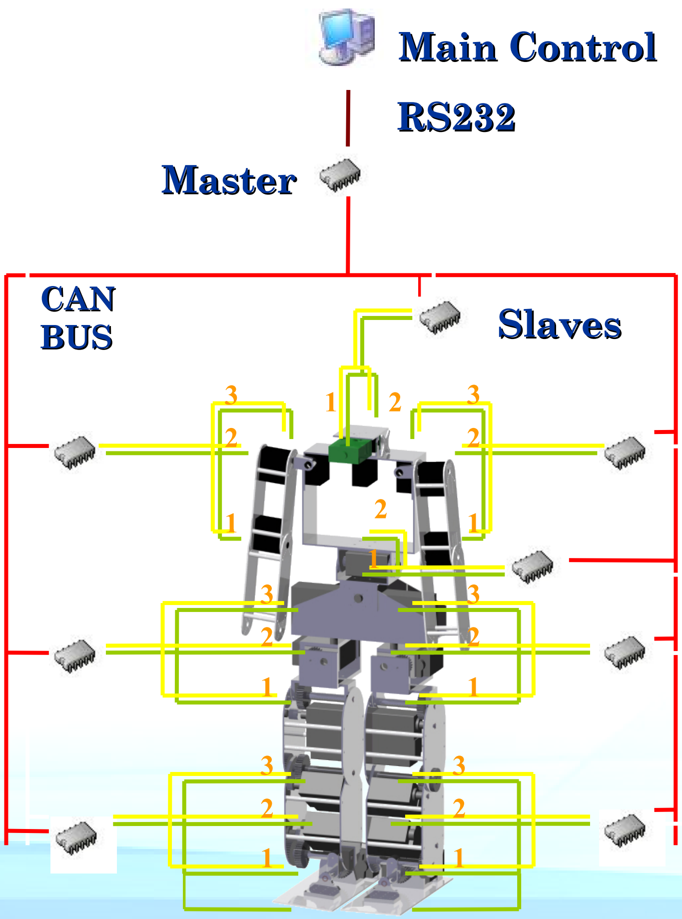


- Actuation
  - Servomotors with transmission belts
- Sensors
  - Servos' internal potentiometers
  - Sensitive foot
  - Accelerometers
  - Gyroscopes



Control System Architecture

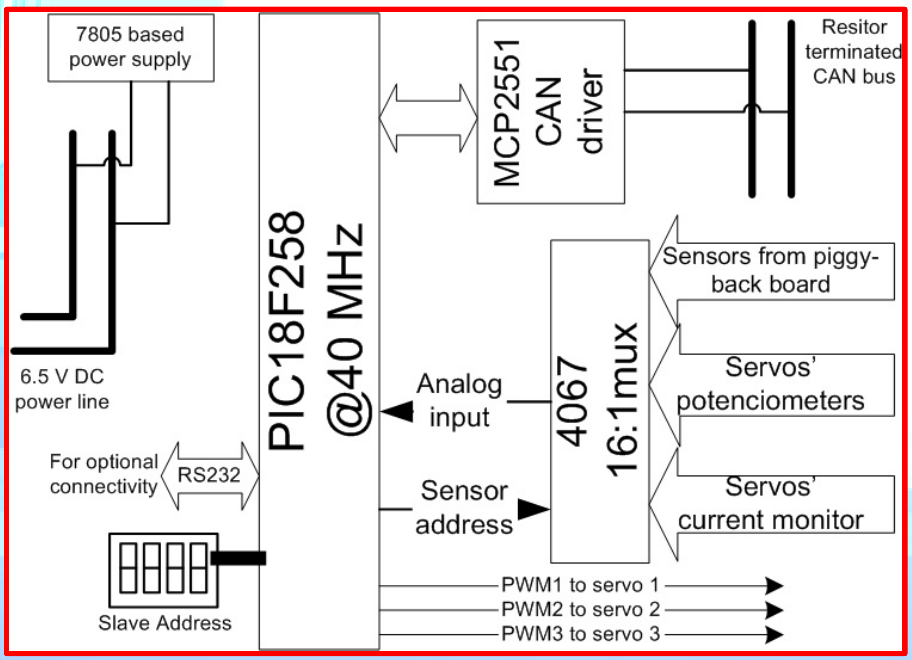
- **Distributed control system**
  - A network of controllers connected by a CAN bus
  - Master/multi-slave arrangement:
    - 8 slave units for joints actuation and sensors reading
    - 1 master unit for interface between main unit and slaves

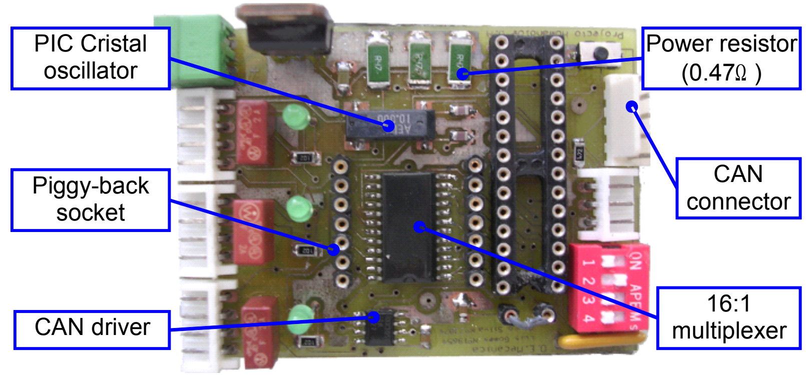


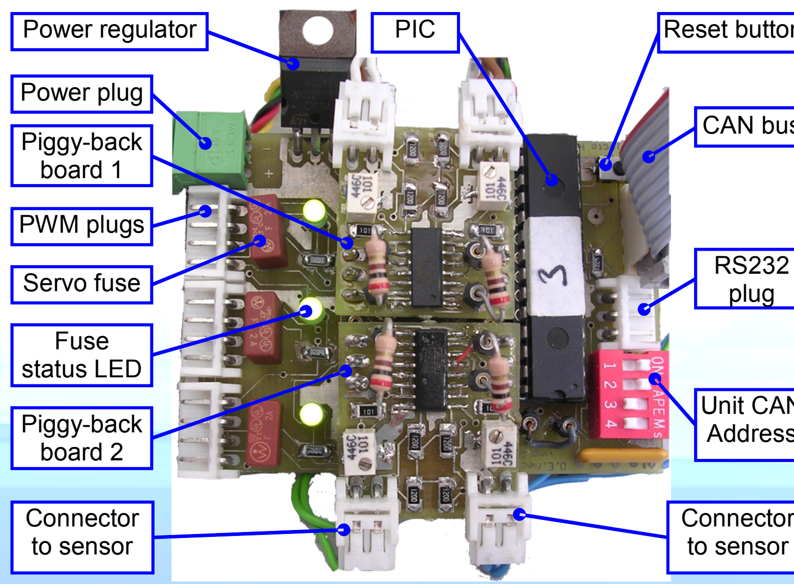
- **Asynchronous communications**
  - Between master and slaves: CAN bus at 833.3 Kbit/s
  - Between master and high level controller: currently serial RS232 at 115200 baud

Local Control

- Each slave controller is made of a PIC 18F258 device with I/O interfacing
- All slave units:
  - Connect up to 3 servomotors
  - Have a common base (a piggy-back unit can add I/O sensors)

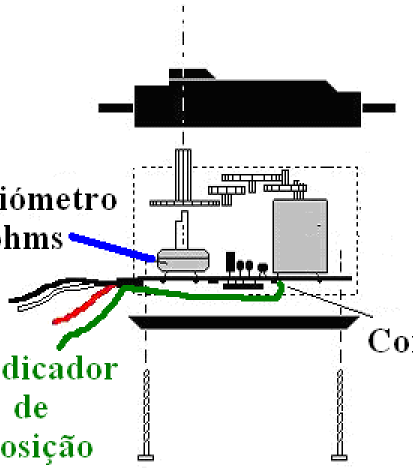




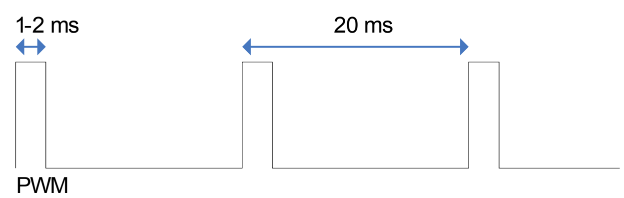


The Servomotor

- **Why servomotors?**
  - Small and compact
  - Relatively inexpensive
  - Position control included



- **Servomotor parts:**
  - DC motor
  - Gearbox
  - Controlling electronics
  - Position feedback mechanism



- **Characteristics**
  - Motion excursion: 180º
  - Position control: digital signal, PWM

- PWM:
  - Period: 20ms (50 Hz)
  - Duty cycle: 1-2 ms

- **Some constraints:**
  - Doesn't offer velocity control
  - Doesn't consider the external load

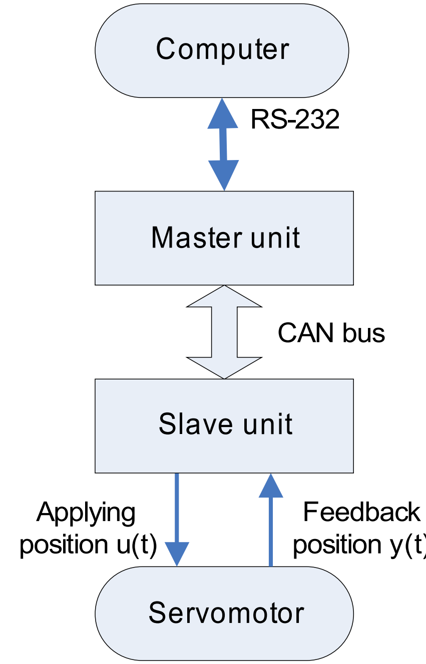
Experimental Setup

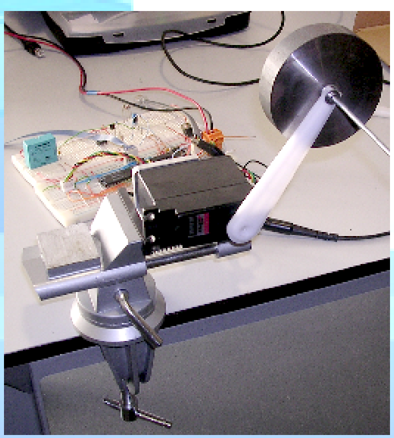
❑ Main goals

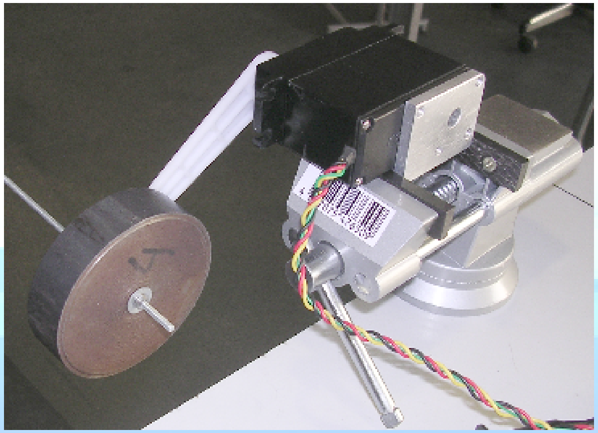
- ➢ To study the servomotor’s performance with high loads and/or velocities
- ➢ Improve the system’s behaviour by software compensation

❑ Only one physical intervention:

- ➢ Connection of an extra output wire to the servo internal potentiometer



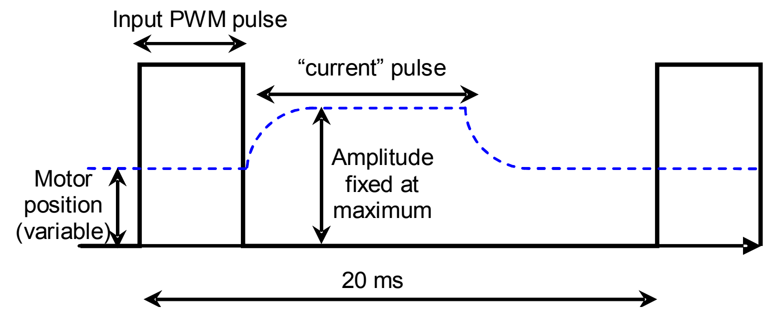


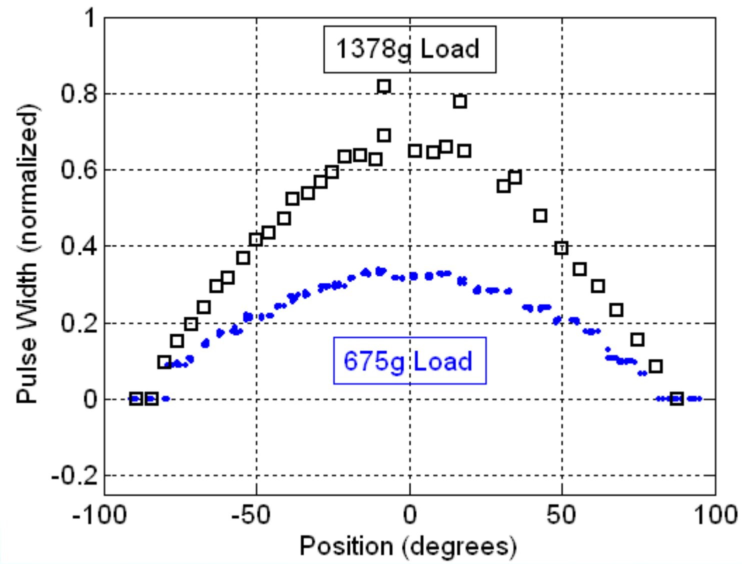


Load	Mass (g)	Torque (N.m)
0	9	0.009
1	258	0.253
2	463	0.454
3	675	0.662
1+3	924	0.906
2+3	1129	1.108
1+2+3	1378	1.352

Parameter Measurement

- **Potentiometer outputs shaft position (feedback signal)**
- **Different voltage "grounds" for external measurement and internal controller can produce measuring fluctuations**
- **For high loads/fast motions a pulse above the position voltage is added:**
  - Fixed amplitude
  - Synchronized to PWM
  - Pulse width related with current consumption!
- **Position and current measured from the same output:**
  - Position: minimal value
  - Current: pulse width
  - For high current draining, position reading can be inhibited!



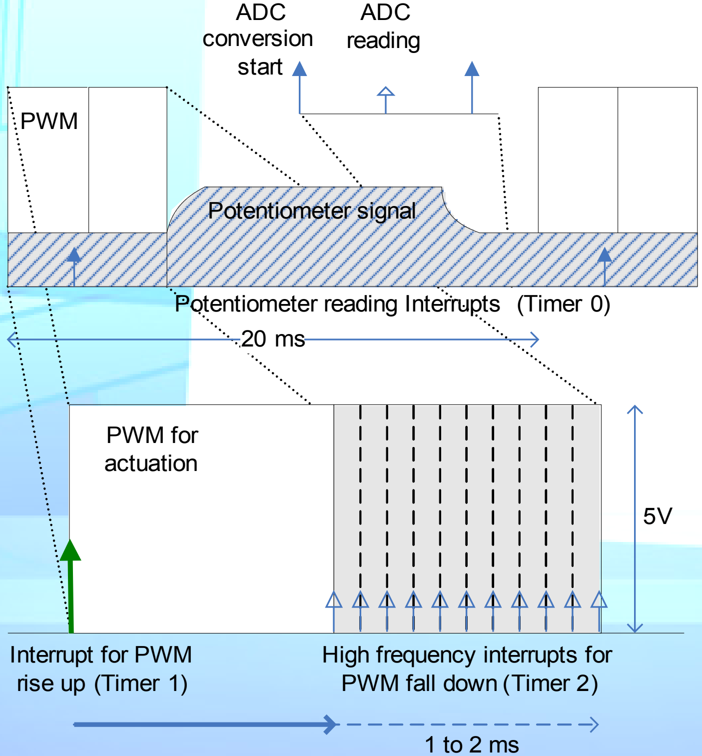


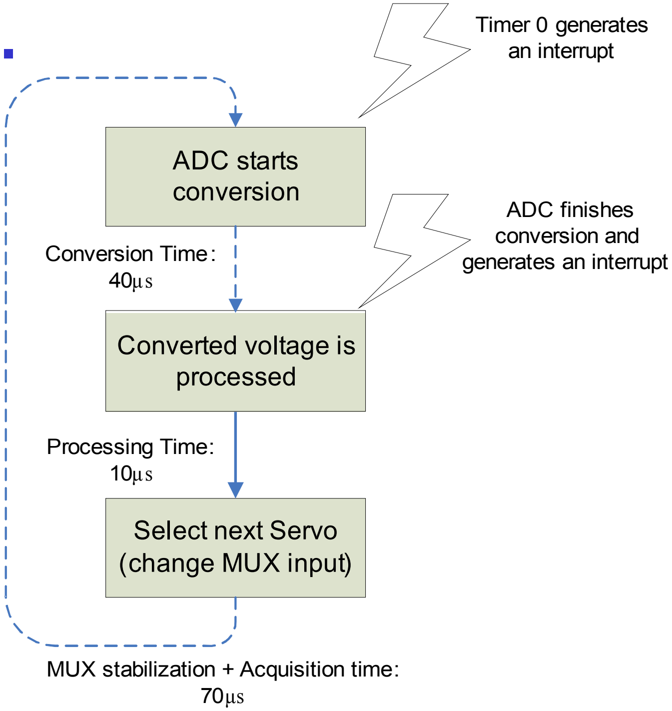
Pulse width proportional to torque/current

$\tau_g = Ka.I = m.g.L.\cos(\Theta)$

Programming Issues

- **Each slave unit controls up to 3 servos.**
- **Actuation:**
  - Each servo has associated to it a variable to store its PWM width
  - µC Timer2 interrupts compares a counter to each variable to apply PWM fall-down

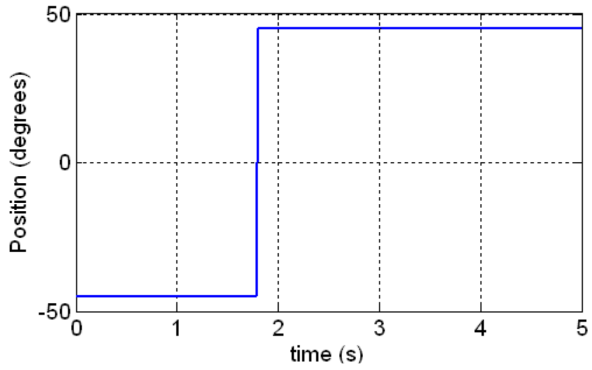


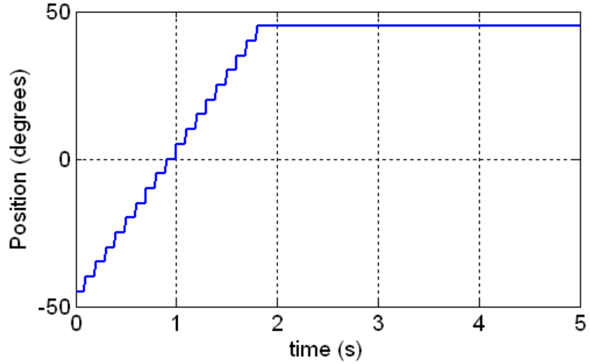


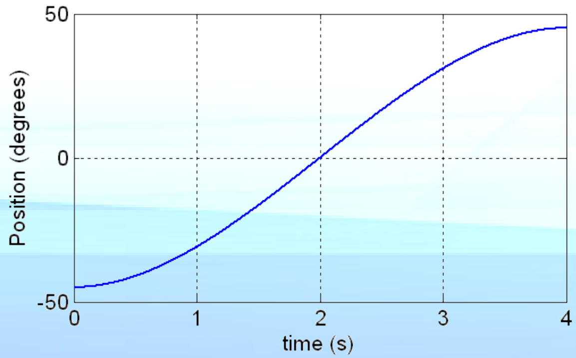
- Sensing
  - Timer0 interrupts used to measure feedback data
  - Servo output sensing is multiplexed in time (120µs*3).

Velocity Control

- **Application of a position step**
  - Servo drives to the commanded position at maximum speed
  - User cannot directly control velocity!
- **How to control velocity with only position control?**
  - Trajectory planning!
- **Slope input**
  - Application of successive "small" step
  - Amplitude and time delay of each step defines the average speed
  - Existence of speed discontinuities!
- **3ª order polynomial**
  - $x(t)=a_0+a_1t+a_2t^2+a_3t^3$
  - Total period defines speed
  - Null initial and final speed
  - Finite initial and final acceleration

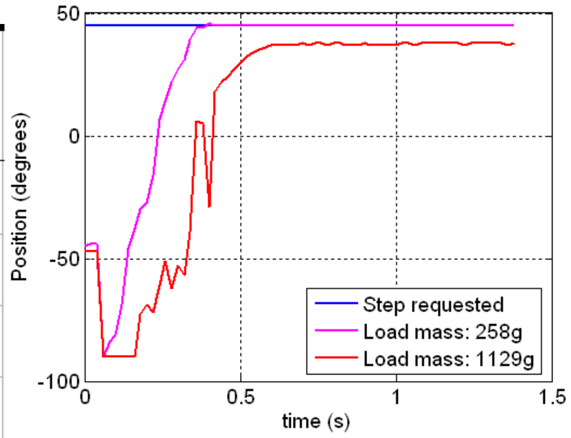


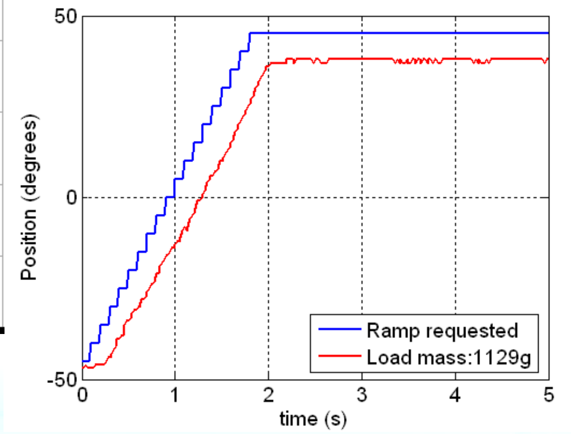




Open Loop Performance

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



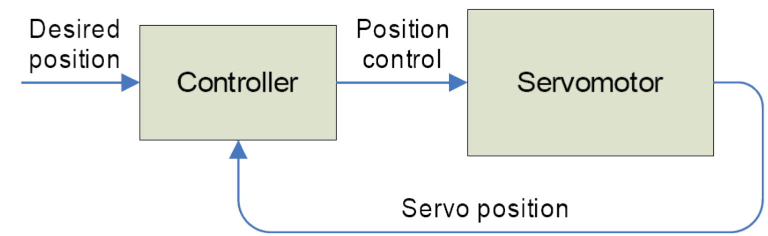


- **Static analysis**
  - Steady state error depends of gravitational terms

External Control

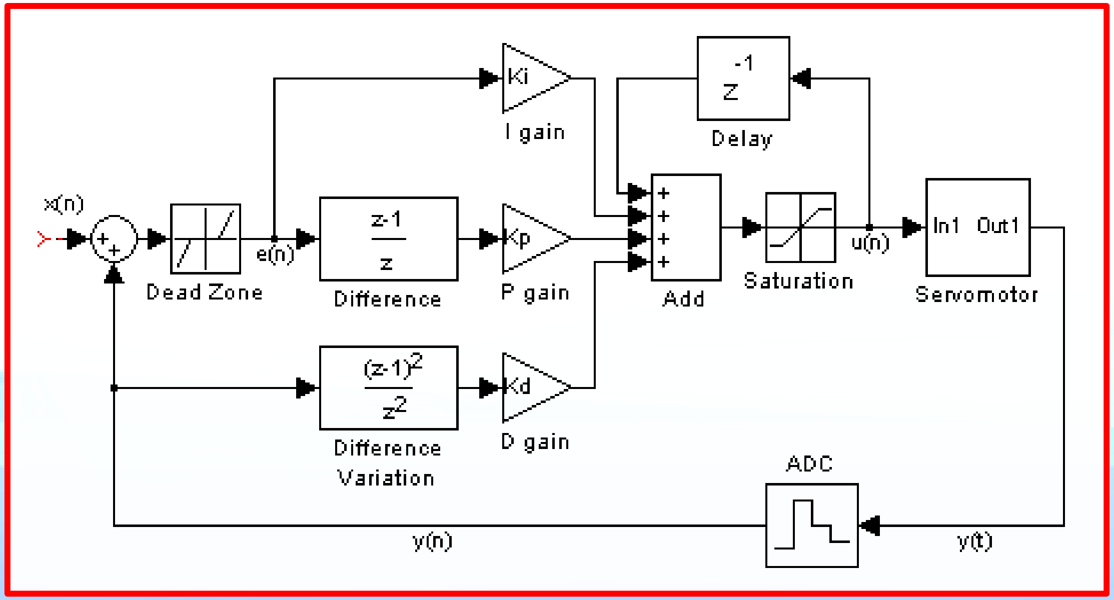
❑ Objectives

- Eliminate steady state error
- Reduce response time lag
- … using software solutions



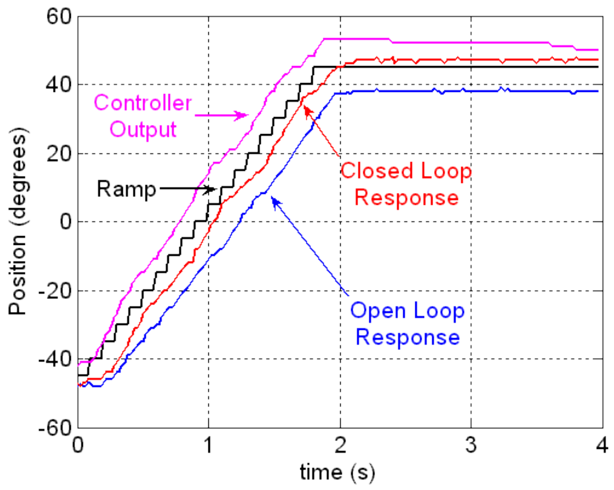
❑ An external controller is implemented using:

- Servo’s own potentiometer for feedback
- Incremental algorithm of a digital PID controller

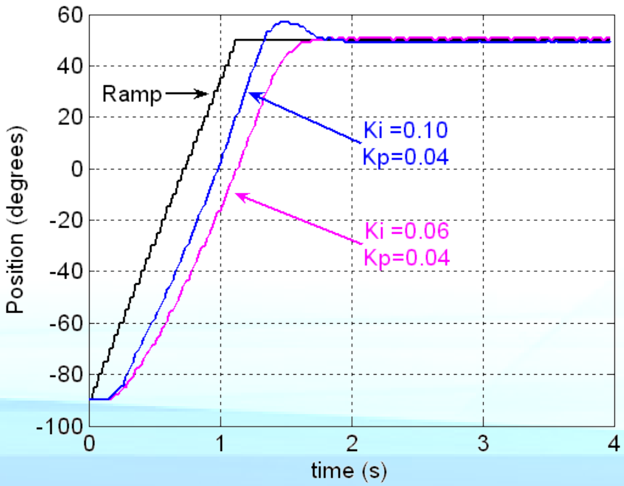


Integral Control

- **Ramp:**
  - -45º → +45º
  - Δt=100ms, Δx=5º, T=1.8s
- **Open loop *vs*. closed loop (K_I=0.2)**
  - Steady state error eliminated
  - Time lag reduced

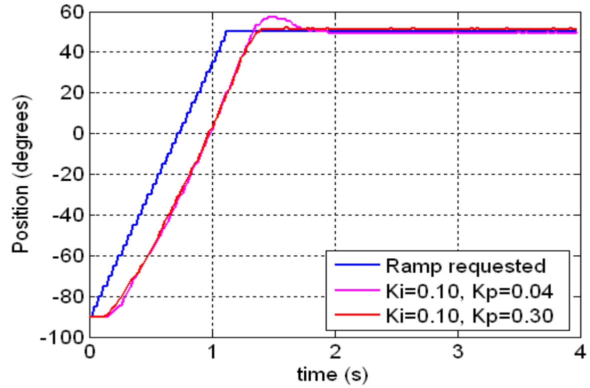


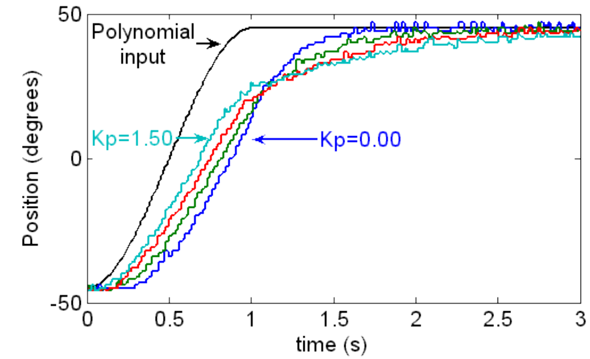
- **Ramp:**
  - -90º → +50º
  - Δt=40ms, Δx=5º, T=1.12s
- **K_I=0.06 *vs*. K_I=0.10 (K_P=0.04)**
  - Time delay reduced
  - Overshoot arises

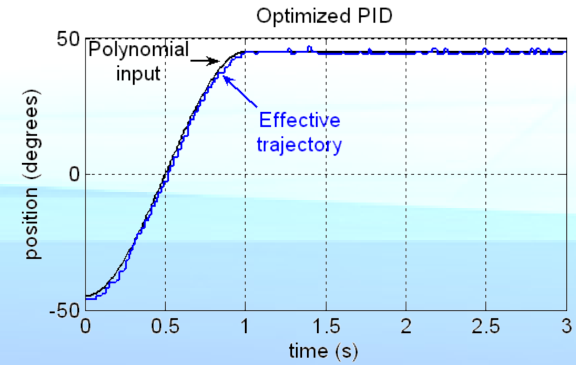


PID Control

- **Ramp response**
  - -90º → +50º
  - Δt=40ms, Δx=5º, T=1.12s
- **K_P=0.04 VS K_P=0.30 (K_I=0.10)**
  - Overshoot eliminated
  - No interference with time delay
- **Polynomial response (T=1s): Increasing K_P…**
  - Overshoot is reduced
  - Time delay diminishes
  - Establishment time increases
- **Adding Derivative component:**
  - Transient state smoothed
  - Overshoot and time delay remains unchanged
  - Sensitive to noise







P+I+D Terms

- **Integrator term**
  - Eliminates steady state error
  - Improves time lag
  - Deteriorates overshoot

- **Proportional term**
  - Reduces overshoot
  - Improves time delay
  - Deteriorates establishment time

- **Derivative term**
  - Smoothes transient response
  - Very sensitive to noise

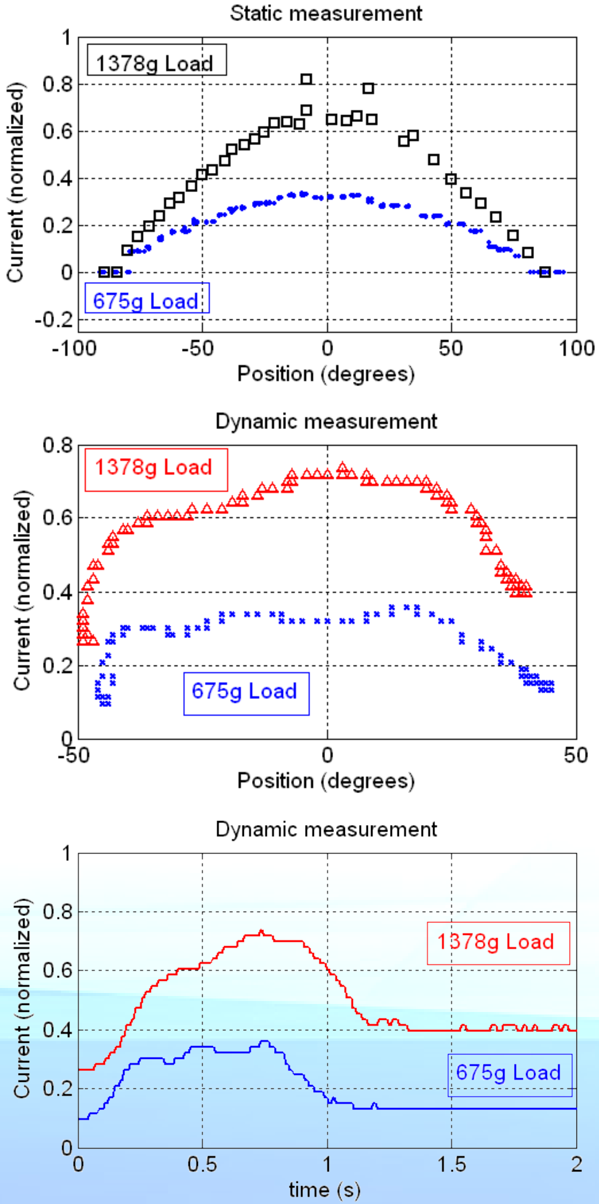
Torque Measurement

- **For a Humanoid platform, the load seen by each actuator can vary rapidly and substantially**

- **For each task it is required to apply an adequate set of control parameters to ensure good performance**

- **Two approaches can be followed:**
  - Parameters determination and manual application from the main unit for each task
  - Local parameter automatic adaptation, through torque analysis

- **Torque estimation can be done from current measurement (τ=K*I)**

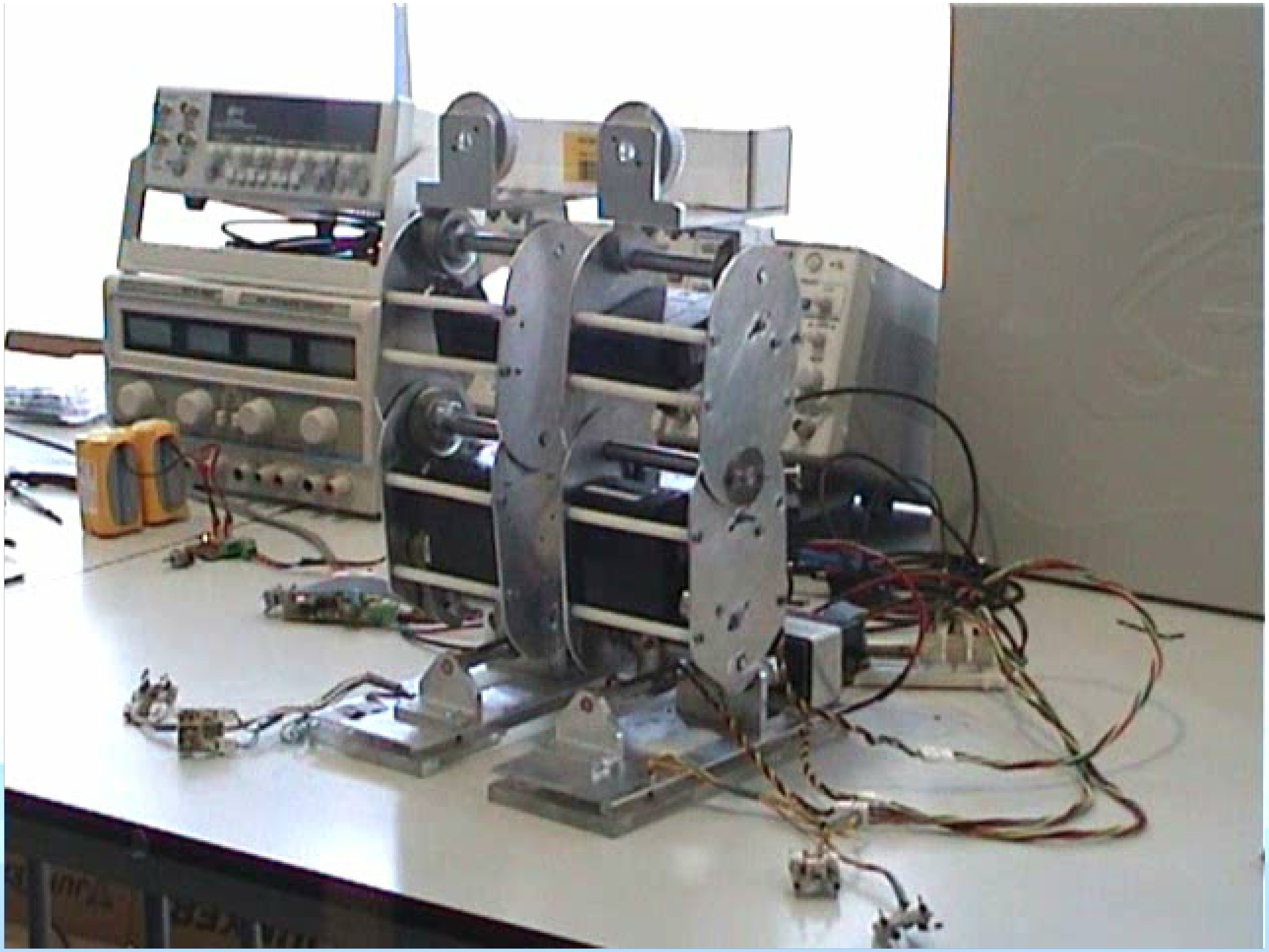


Conclusions

- **Procedures were presented to measure servos'…**
  - shaft position
  - current consumption (torque)
- **With a servomotor by itself…**
  - without velocity control
  - whose speed and steady-state error was dependent of motor load
- **…methods were described on how…**
  - to introduce velocity control through trajectory planning
  - to correct time lag and steady-state error through an external controller using position reading as feedback
- **…all this without hardware interventions!**
- **Nevertheless, controller needs to be updated to maintain efficiency:**
  - To build an adaptive PID controller whose parameters are based on torque estimation is the next objective to pursue

Humanoid Motions

- The robot is able to stand, lean on sides, for/backward



Humanoid Motions (With Loads)

- **Load weight: ≈2Kg**