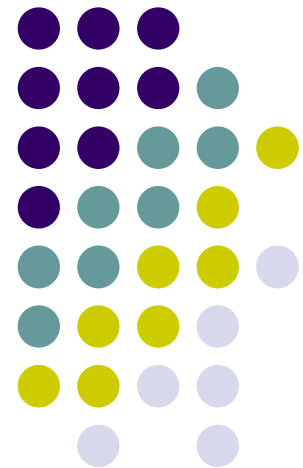
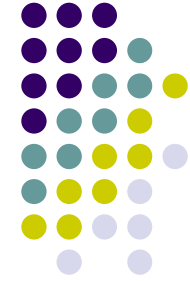


Unidade de Processamento e Sistema de Visão para um Robô Humanóide

Prof. Dr. Filipe Miguel Teixeira Pereira da Silva
Prof. Dr. Vítor Manuel Ferreira dos Santos
Mauro André Moreira Rodrigues





Resumo da Apresentação

- Objectivos
- Plataforma Humanóide
- Unidade Central de Processamento
- Ambiente de Desenvolvimento
- Sistema de Visão
- Resultados
- Conclusões
- Trabalho Futuro





Objectivos

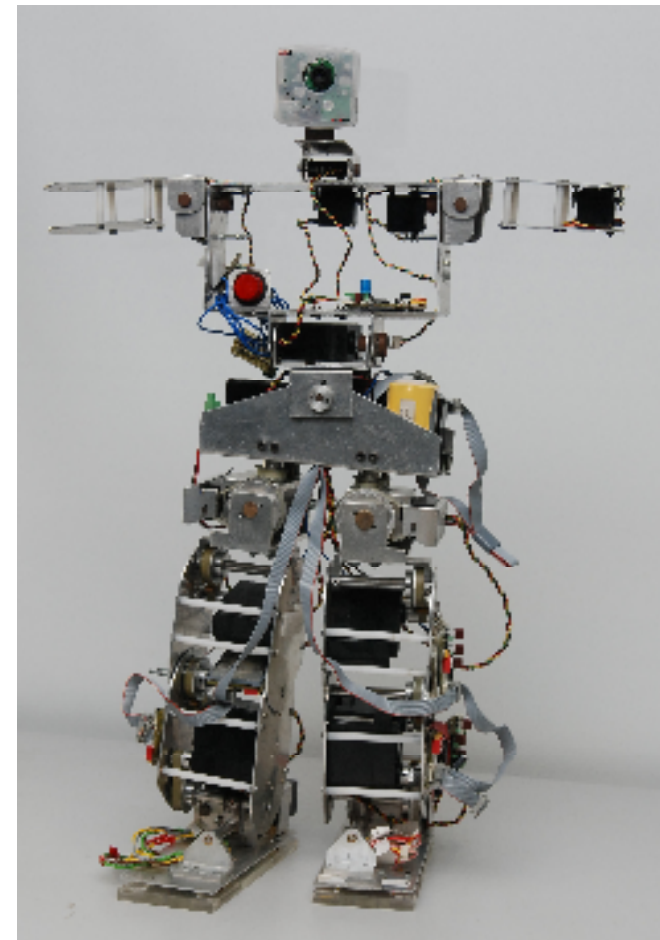
- Integração da Unidade Central de Processamento
- Implementação de um Ambiente de Desenvolvimento
- Desenvolvimento do Sistema de Visão





Plataforma Humanóide

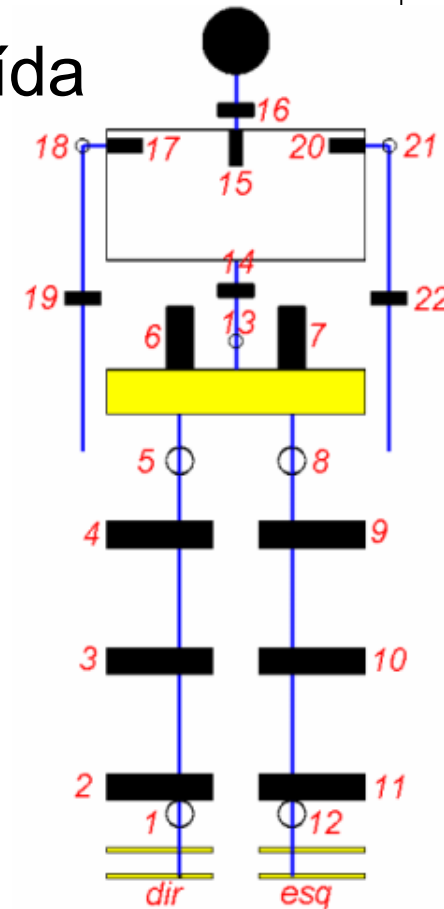
- Robô Humanóide desenvolvido na Universidade de Aveiro.
- A ambição é a participação no *RoboCup*.





Plataforma Humanóide

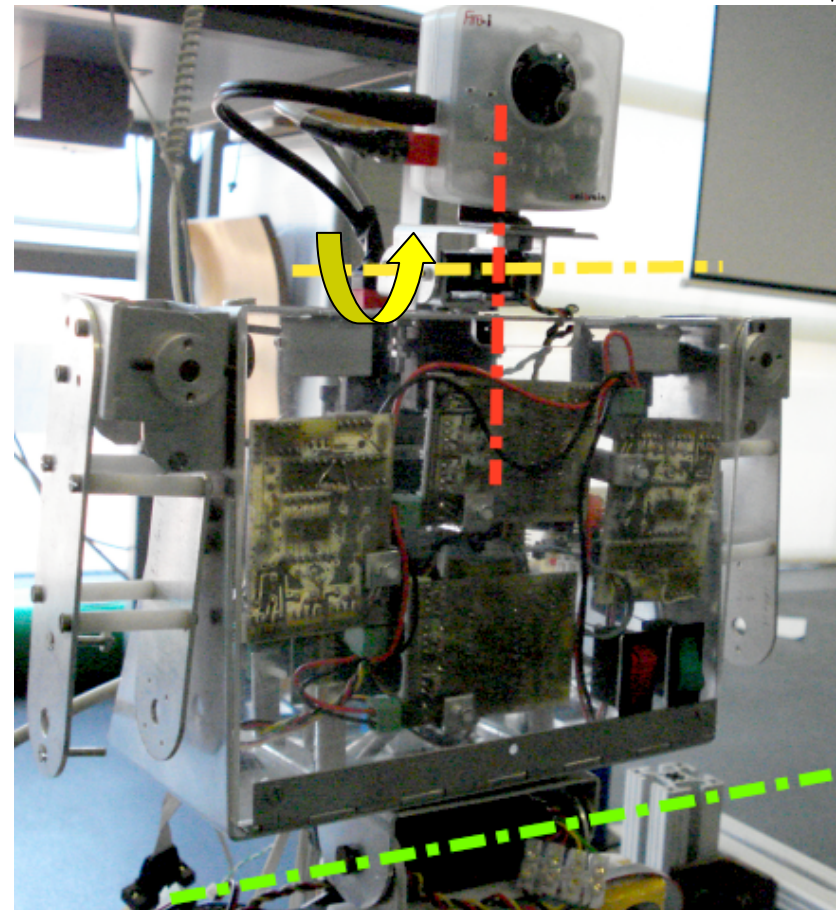
- A plataforma humanóide é constituída por 22 graus de liberdade:
 - 2 em cada pé (2x2);
 - 1 em cada joelho (1x2);
 - 3 em cada anca (3x2);
 - 2 no tronco (2x1);
 - 3 em cada braço (3x2);
 - 2 na cabeça (PTU) (2x1).
- Atinge os 70 cm de altura e uma massa total de 7 Kg.

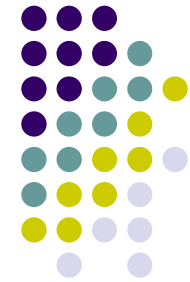




Plataforma Humanóide

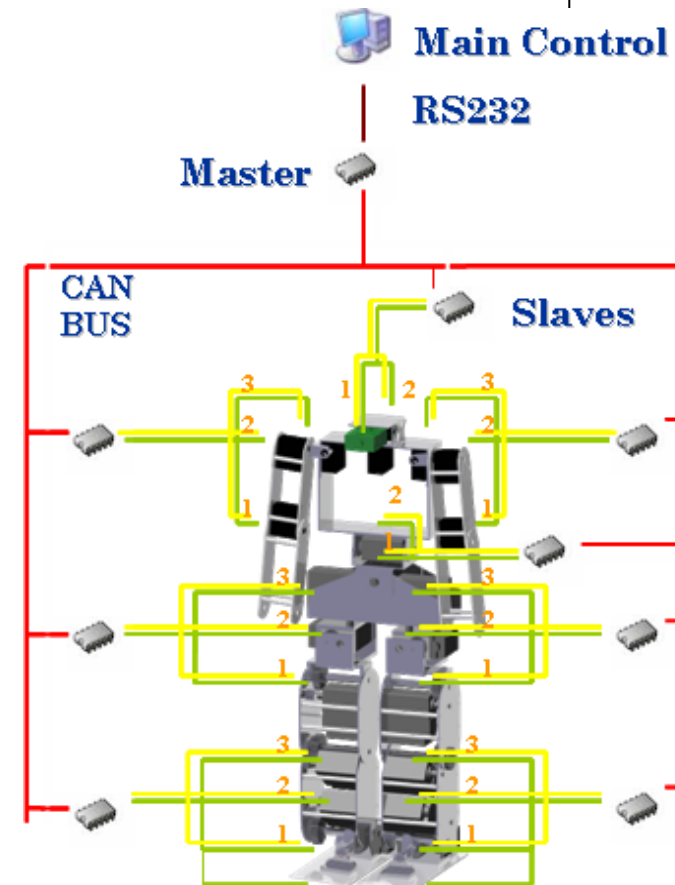
- Controlo Visual
 - Unidade *Pan&Tilt* (PTU)
 - Inclinação do tronco

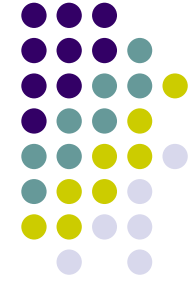




Plataforma Humanóide

- Arquitectura de controlo distribuída
- Três tipo de unidades, formando uma rede de controladores.
- Interligação por barramento *CAN* em configuração *Master/Multi-Salve*.





Plataforma Humanóide

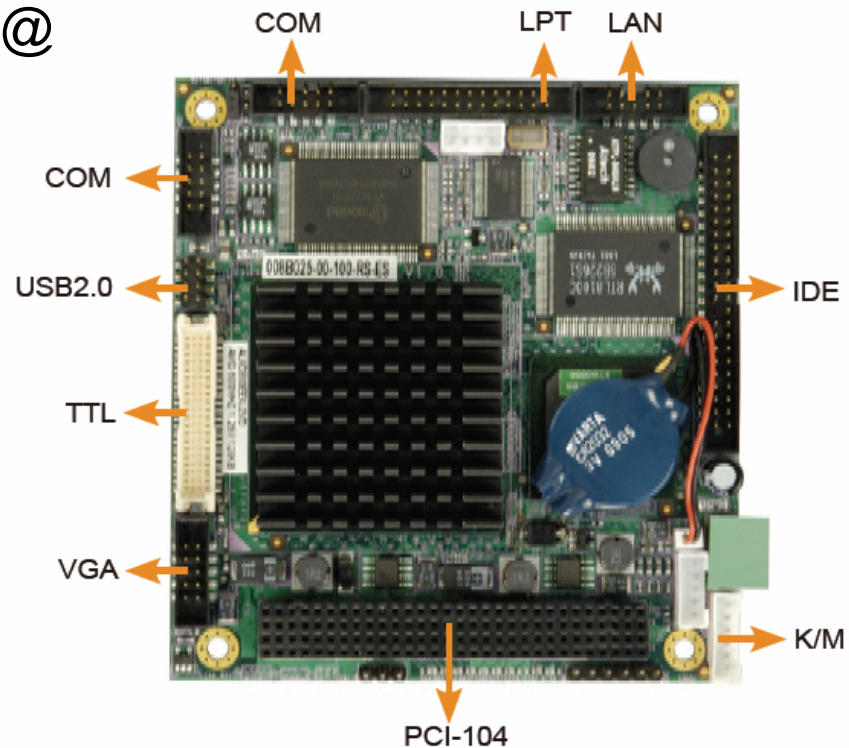
- **Unidade Central de Processamento:** responsável pela gestão global dos procedimentos:
 - Interacção com computador externo para monitorização, *debug* ou tele-operação;
 - Cálculo das configurações que as juntas devem adoptar com base em directivas de alto nível;
 - Processamento de imagem;
 - Controlo Visual.





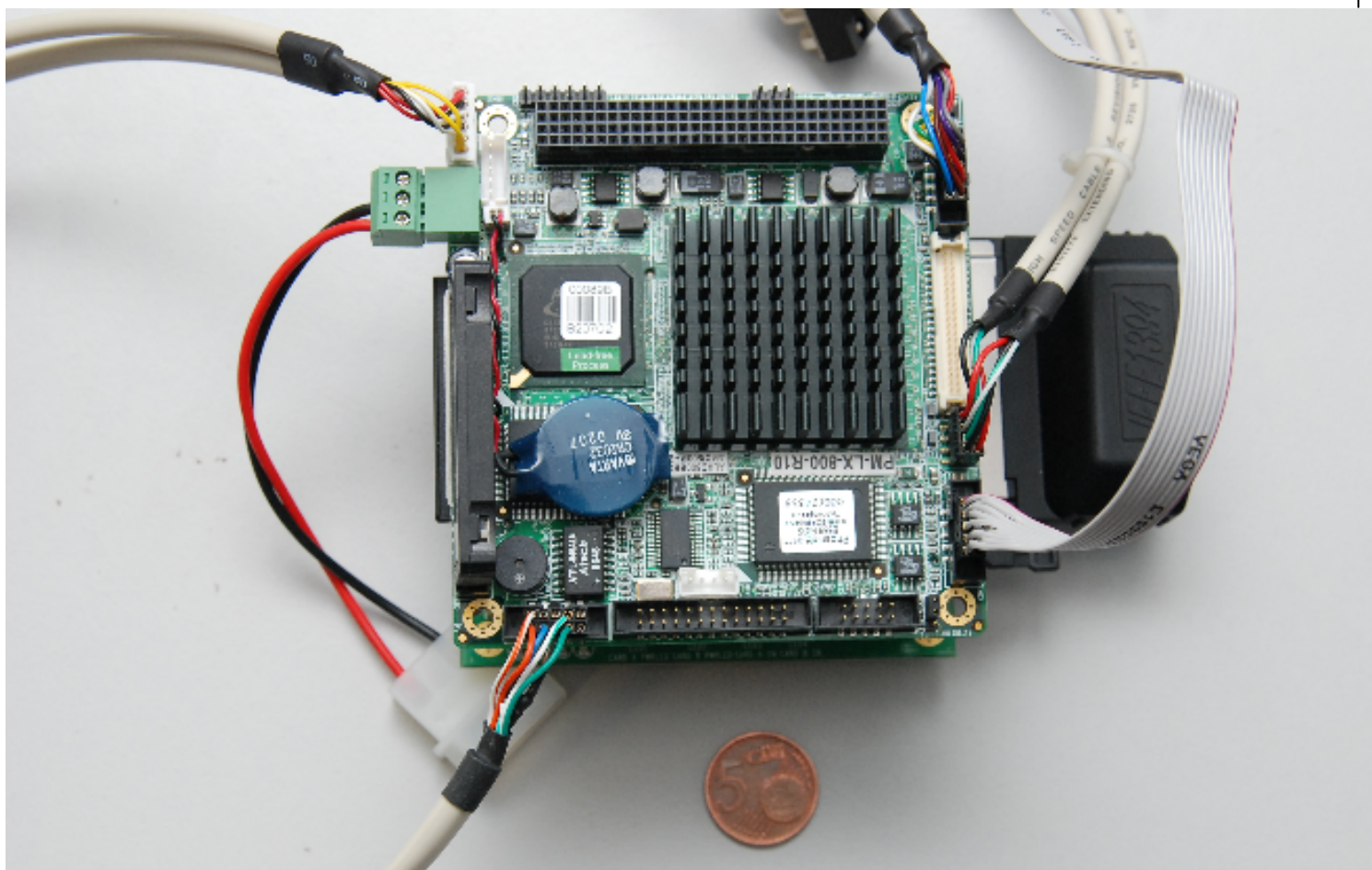
Unidade Central de Processamento

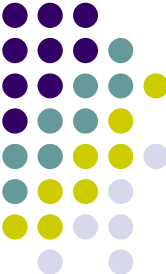
- CPU *standard* PCI-104
 - AMD Geode LX-800 @ 500MHz
 - 512Mb RAM
 - SSD 1Gb



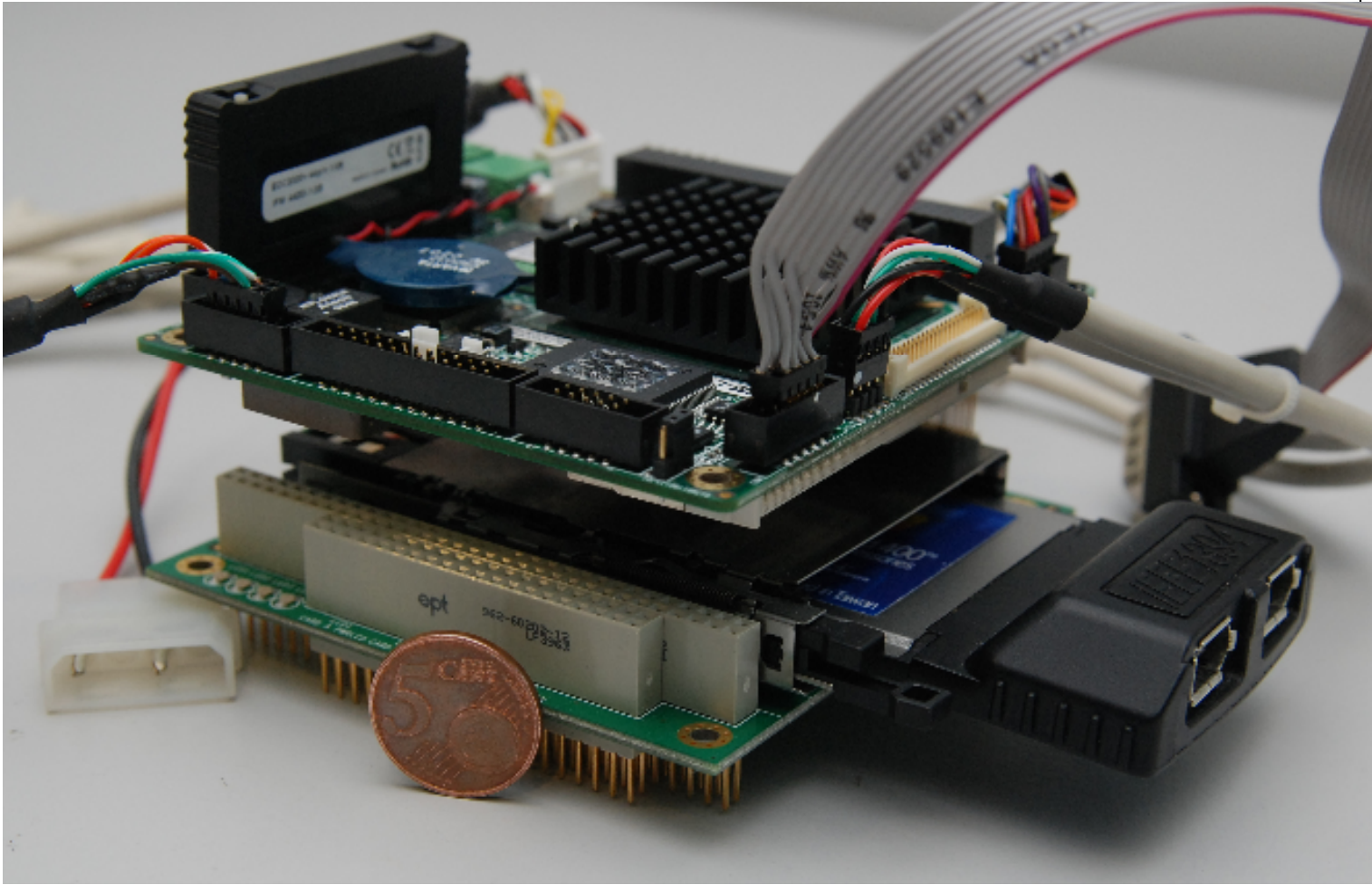


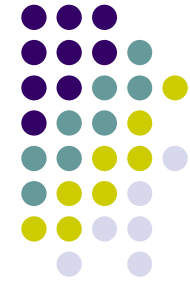
Unidade Central de Processamento





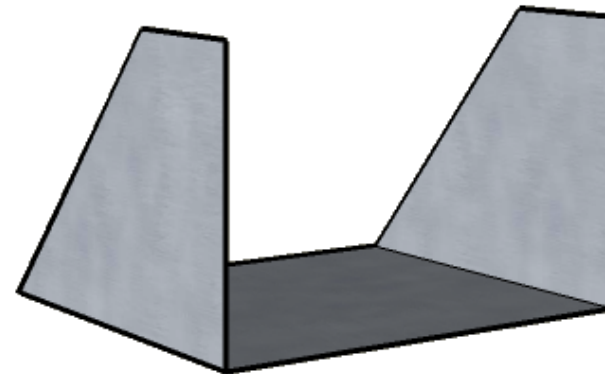
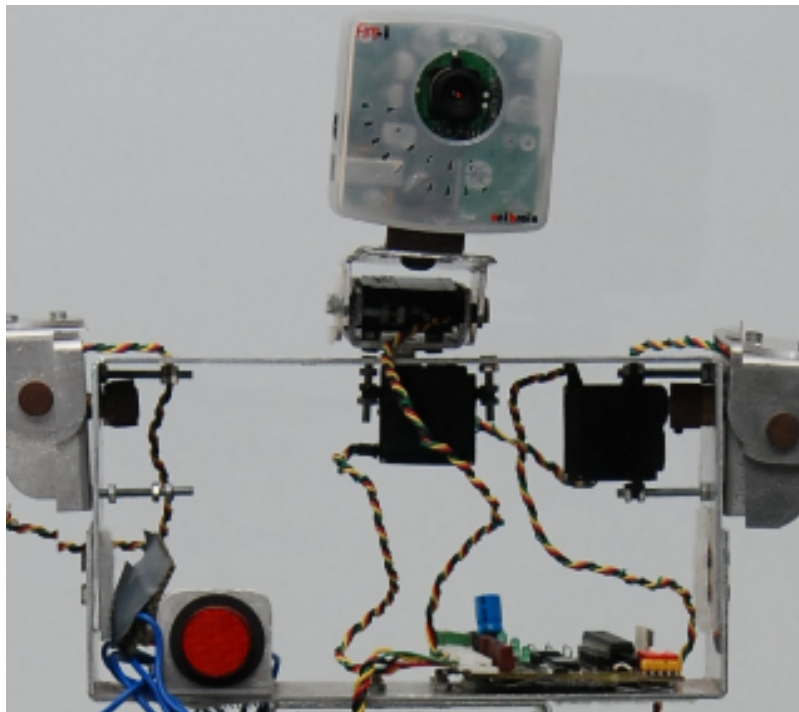
Unidade Central de Processamento

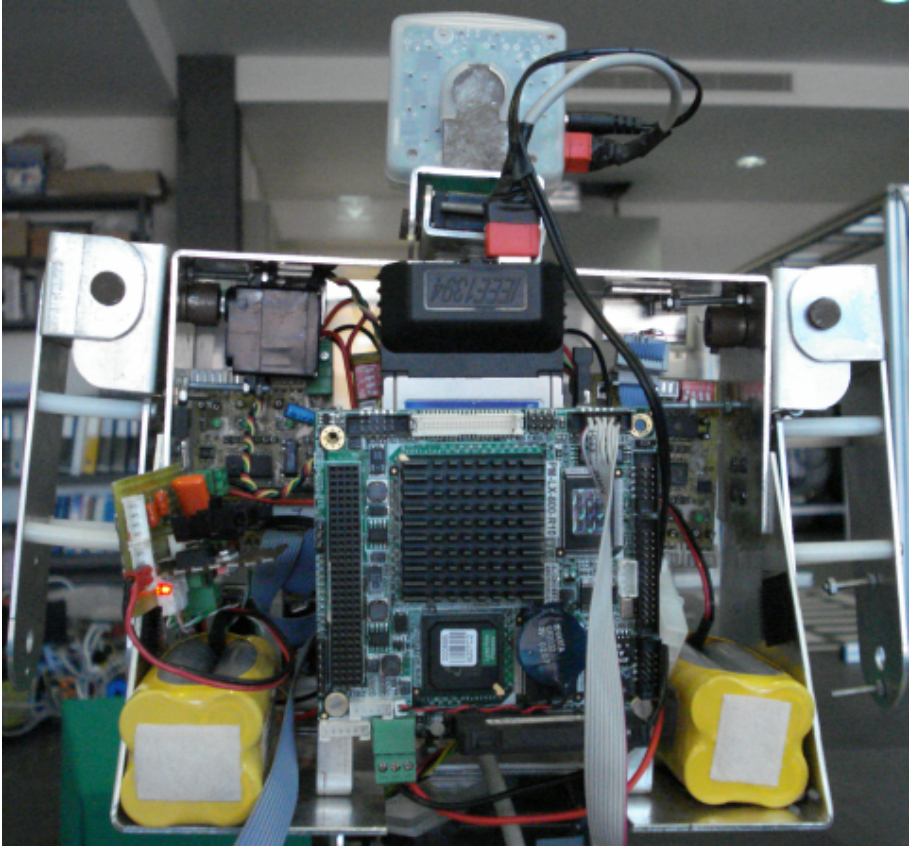
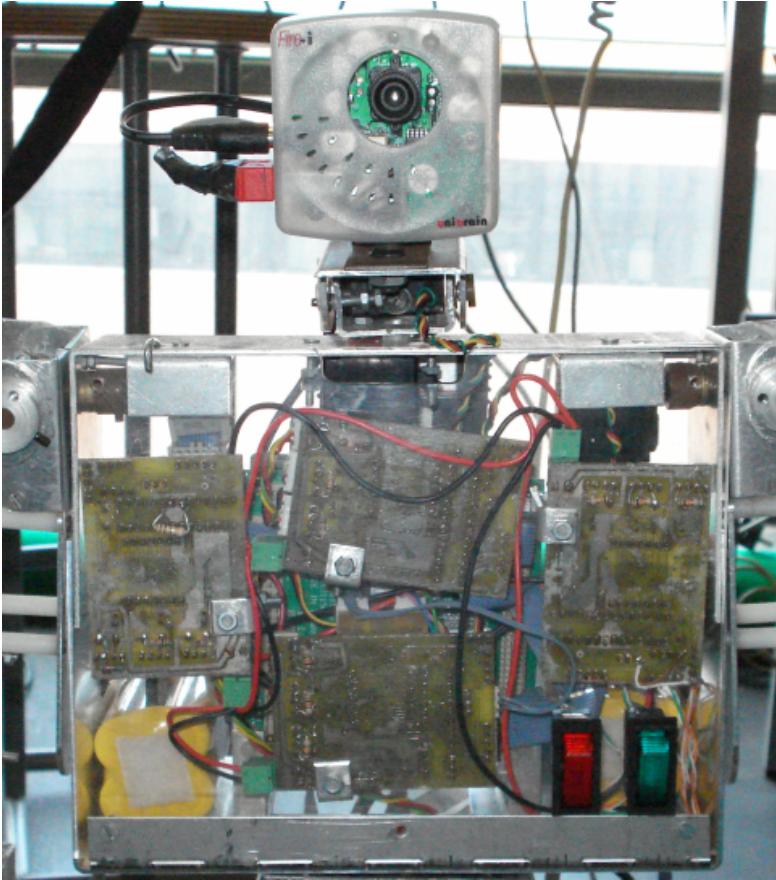


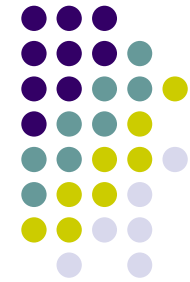


Plataforma Humanóide

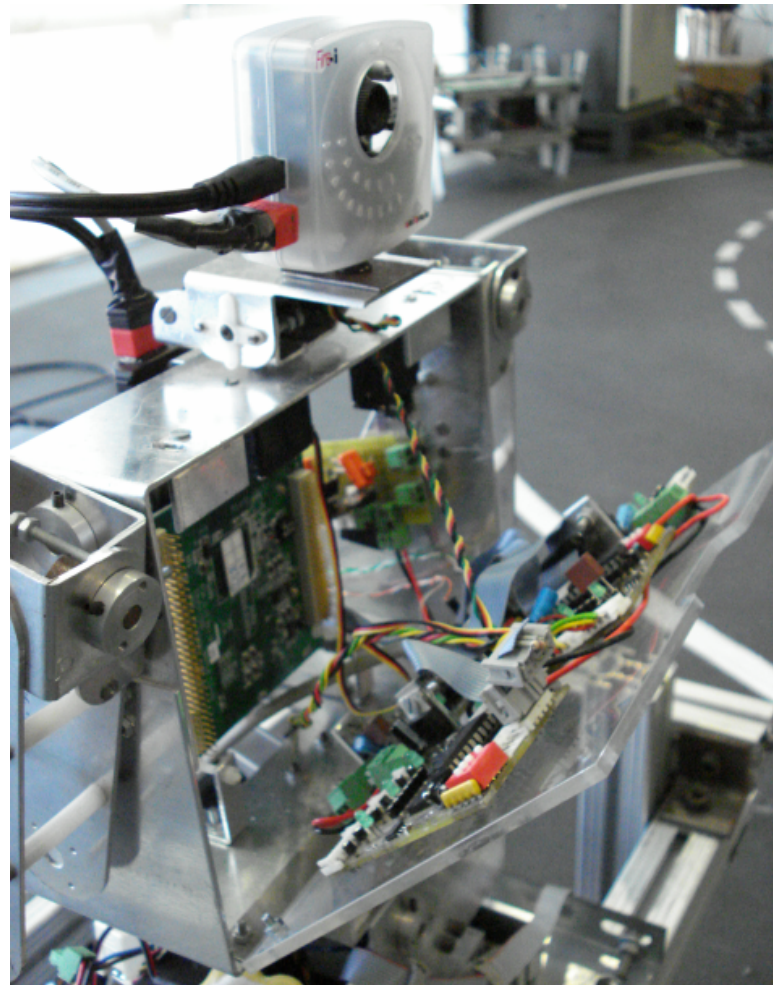
- Alterações à estrutura







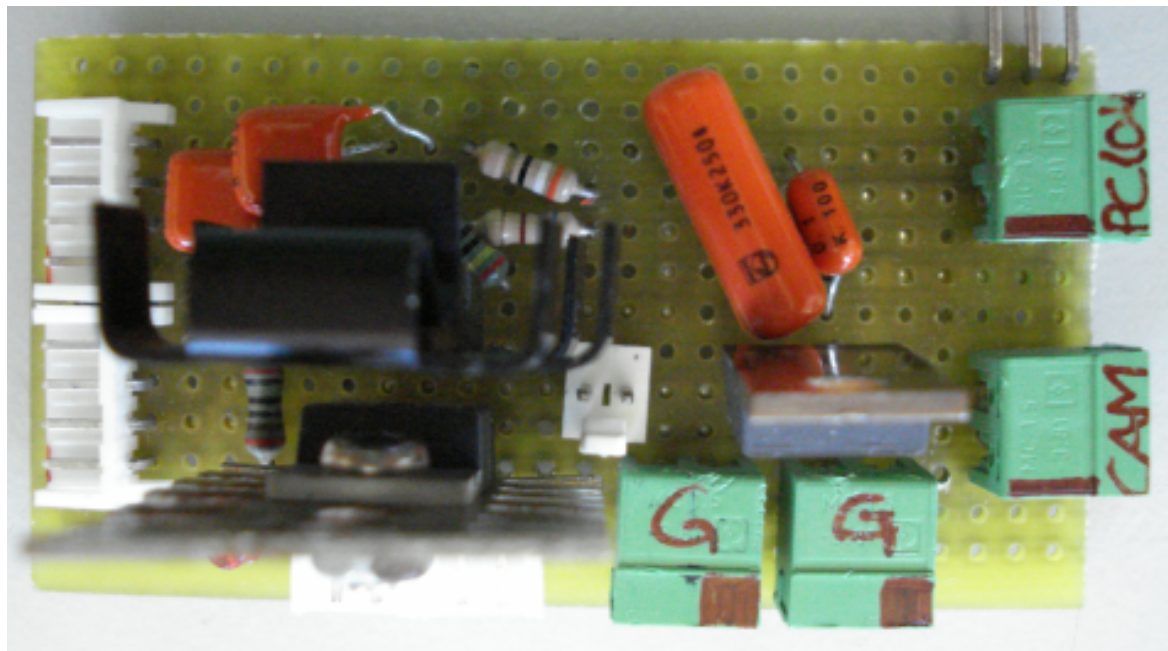
Plataforma Humanóide





Plataforma Humanóide

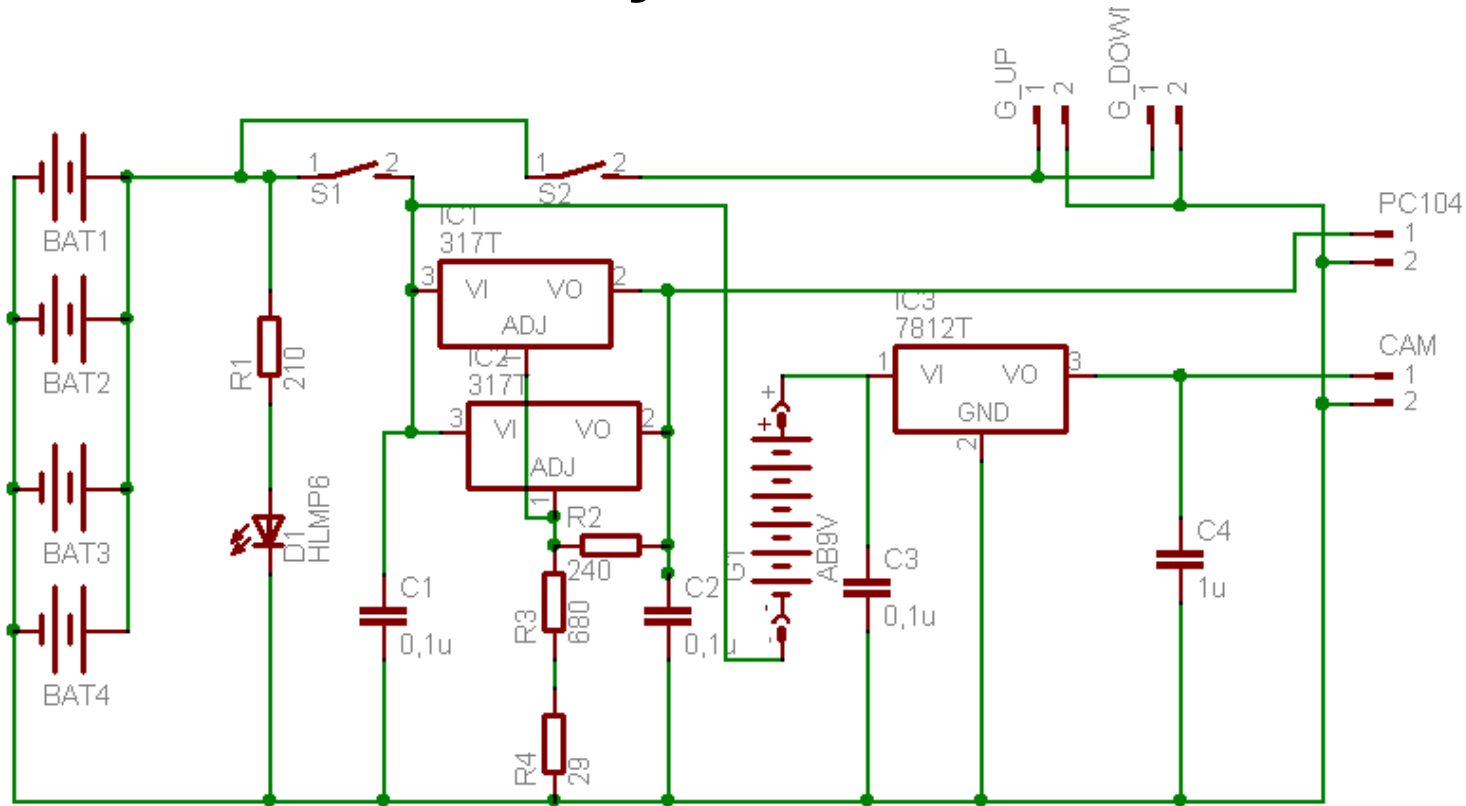
- Placa de Alimentação





Plataforma Humanóide

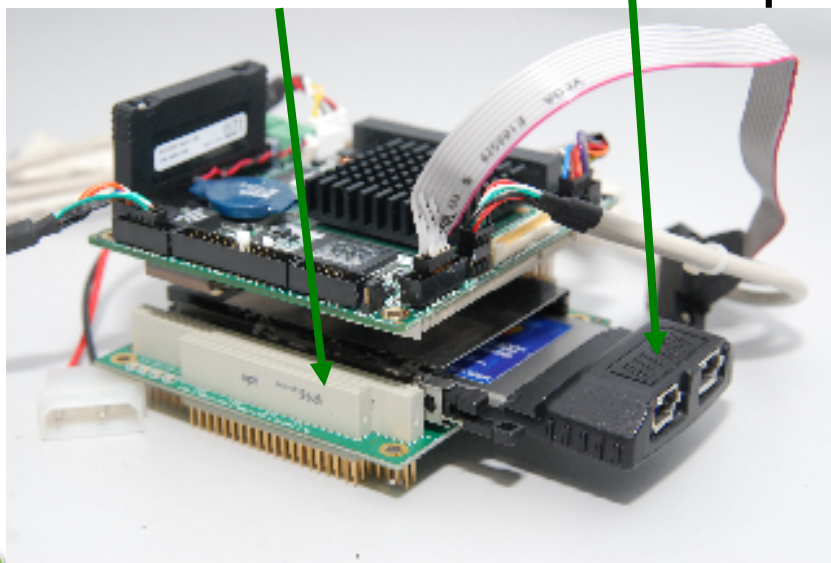
- Placa de Alimentação





Unidade Central de Processamento

- Captura de Sinal Vídeo
 - Câmara *UniBrain Fire-i @ 30fps (640x480)*
 - Placa *PCMCIA FireWire*
 - Módulo *Dual PCMCIA p/ PC104*





Unidade Central de Processamento

- Sistema Operativo
 - GNU/Linux *Debian 40r0 i386 Net Install*
 - Versão mínima, ≈200Mb.

- Gestão de Memória
 - Ciclos de escrita limitados
 - Sem Área de *Swap*
 - Utilização de *Ramdisk*





Ambiente de Desenvolvimento

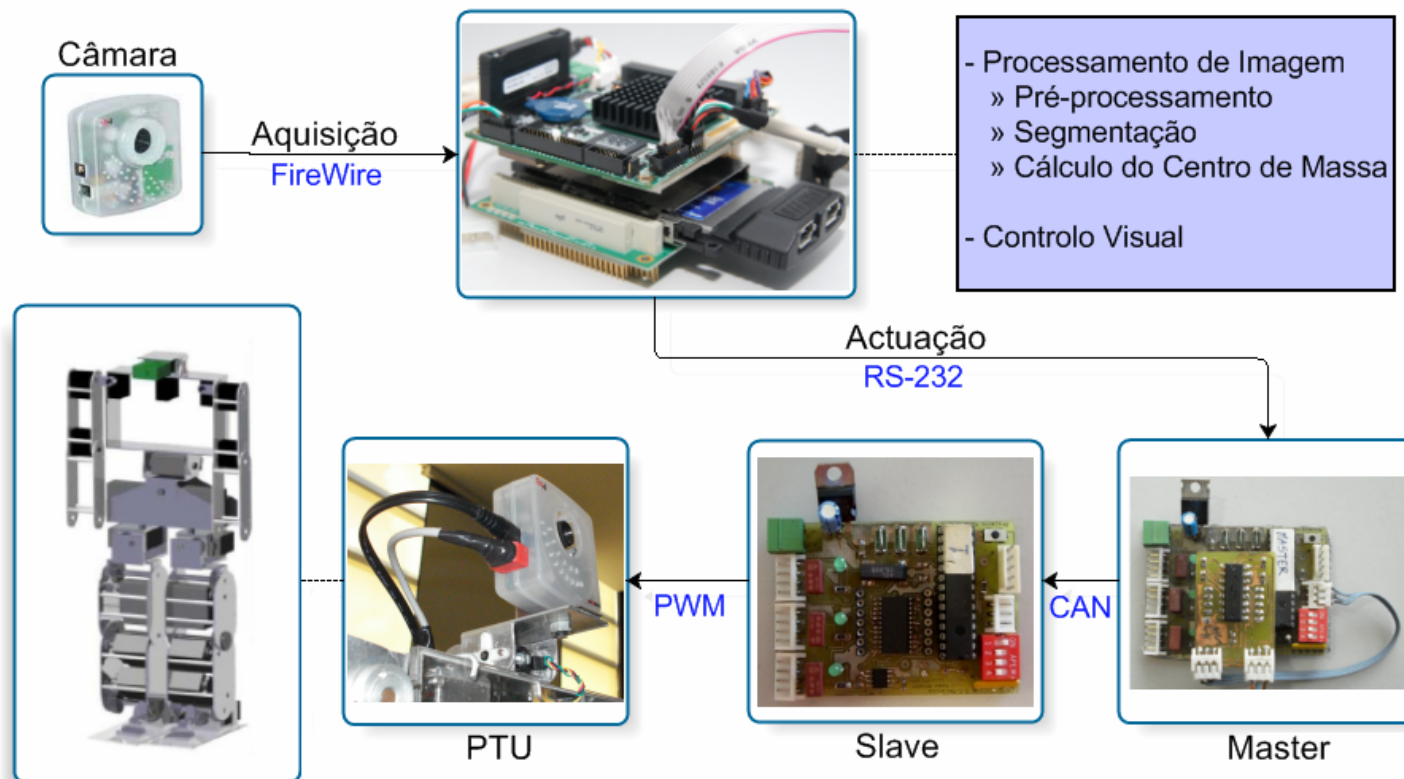
- Ferramentas de Sistema
 - Ambiente Gráfico
 - Xfce
 - Gestão de Câmaras IEEE1394
 - Coriander
- Bibliotecas
 - OpenCV
- Ferramentas de Programação
 - Editor
 - Vim
 - Compilador
 - GCC
 - Make
 - IDE
 - Kdevelop





Sistema de Visão

- Arquitectura do Sistema de Visão





Sistema de Visão

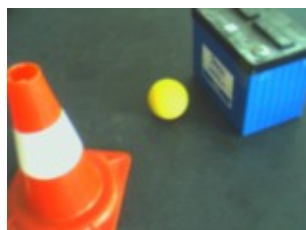
Aquisição



Segmentação - Componentes H, S e V



Pré-processamento



Máscara



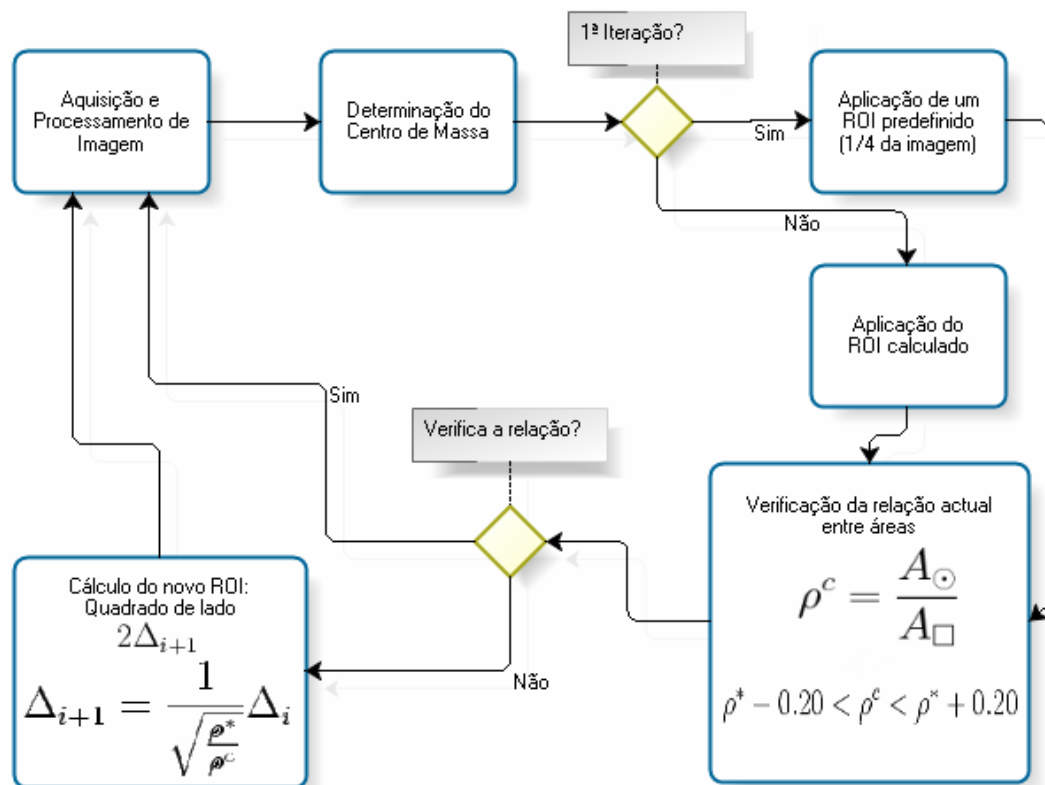
Localização do Objecto

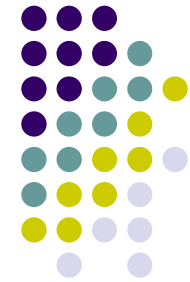




Sistema de Visão

- Região de Interesse (ROI)





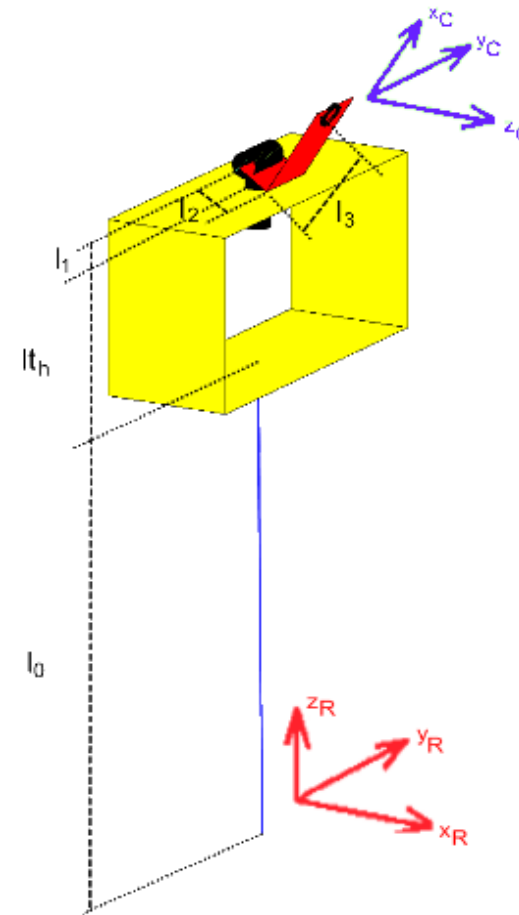
Sistema de Visão

● Análise Cinemática

$$T_C = \left[\begin{array}{ccc|c} -S_1 & C_1 S_2 & C_1 C_2 & l_2 C_1 C_2 + l_3 C_1 S_2 \\ C_1 & S_1 S_2 & S_1 C_2 & l_2 S_1 C_2 + l_3 S_1 S_2 \\ 0 & C_2 & -S_2 & -l_2 S_2 + l_3 C_2 + l_0 + l_{tH} + l_1 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

$$\hat{z}_C = \begin{pmatrix} C_1 C_2 \\ S_1 C_2 \\ -S_2 \end{pmatrix}$$

$$\begin{cases} \alpha = -\frac{z_C}{\hat{z}_C^z} \\ x_B = x_C + \alpha \hat{z}_C^x \\ y_B = y_C + \alpha \hat{z}_C^y \\ z_B = 0 \end{cases} \Rightarrow \begin{cases} x_B = \frac{C_1}{S_2}(l_3 + L C_2) \\ y_B = \frac{S_1}{S_2}(l_3 + L C_2) \end{cases}, L = l_0 + l_{tH} + l_1$$





Sistema de Visão

- Algoritmo baseado em Imagem
 - Lei proporcional, $\dot{q} = Ke$
 - \dot{q} , é o vector de velocidade das juntas
 - K , matriz de ganhos constantes
 - $e = [C_x, C_y]^T$, é o vector de erro definido pelo desvio da bola
 - Aproximação Jacobiana

$$\dot{q} = \left[\frac{K_p C_x}{\sin(q_t)}, \frac{K_t C_y}{\cos(q_t)} \right]$$





Resultados

Tabela 1. aquisição OpenCV a 640x480 - tempos (ms)				
	<i>max</i>	<i>min</i>	<i>medio</i>	<i>desvio</i>
<i>aquisição</i>	31,0230	11,8310	20,2630	4,6492
<i>pyr down</i>	83,6740	47,1840	47,6000	3,2164
<i>processamento</i>	21,9830	9,5830	9,9274	1,0795
<i>centro de massa</i>	3,1960	3,1050	3,1327	0,0136
<i>algoritmo</i>	0,0170	0,0140	0,0148	0,0062
<i>actuação</i>	52,8640	3,0700	43,1530	12,5260
<i>total</i>	156,0990	80,1240	124,0900	13,3760

Tabela 2. aquisição directa a 160x120 - tempos (ms)				
	<i>max</i>	<i>min</i>	<i>medio</i>	<i>desvio</i>
<i>aquisição</i>	6,4660	2,8860	3,1088	1,9602
<i>pyr down</i>				
<i>processamento</i>	42,1950	9,7330	10,3318	2,7850
<i>centro de massa</i>	18,8870	2,7850	2,9364	1,0359
<i>algoritmo</i>	0,2530	0,0140	0,0168	0,0161
<i>actuação</i>	3637,2710	2,4100	11,7372	12,5260
<i>total</i>	3653,0450	18,6400	28,1310	162,6566

- Tempos de execução
 - Aquisição lenta pela biblioteca OpenCV
 - Recurso a biblioteca proprietária melhora tempo de aquisição

- Aquisição
 - Caso 1. ~8Hz
 - Caso 2. 30Hz





Resultados

Tabela 3. aquisição directa a 320x240 c/pyr down - tempos (ms)

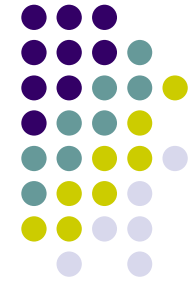
	<i>max</i>	<i>min</i>	<i>medio</i>	<i>desvio</i>
aquisição	33,6860	12,0350	13,8940	2,1626
pyr down	1783,1240	9,4760	18,6820	125,7200
processamento	169,9590	9,7050	10,9220	11,4220
centro de massa	11,4430	3,0680	3,4089	0,8132
algoritmo	0,2350	0,0140	0,0164	0,0157
actuação	90,4820	2,0250	4,5410	6,3831
total	1816,1850	39,4350	51,4640	126,6200

Tabela 4. ROI - tempos (ms)

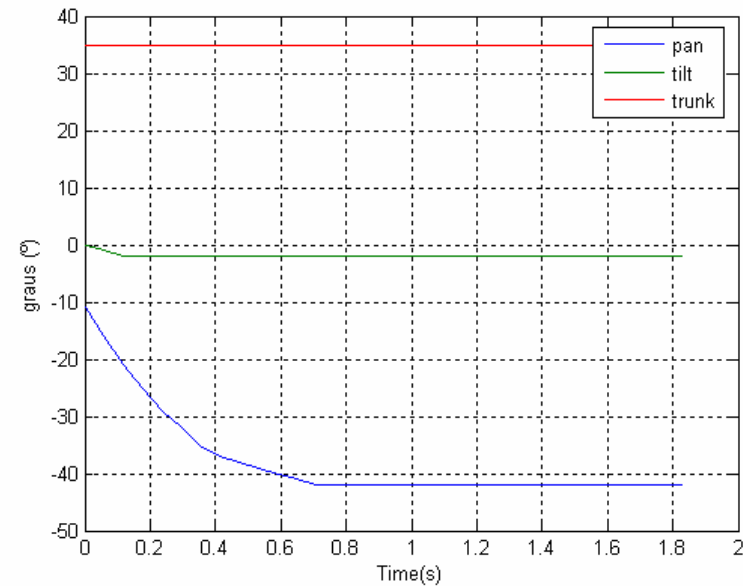
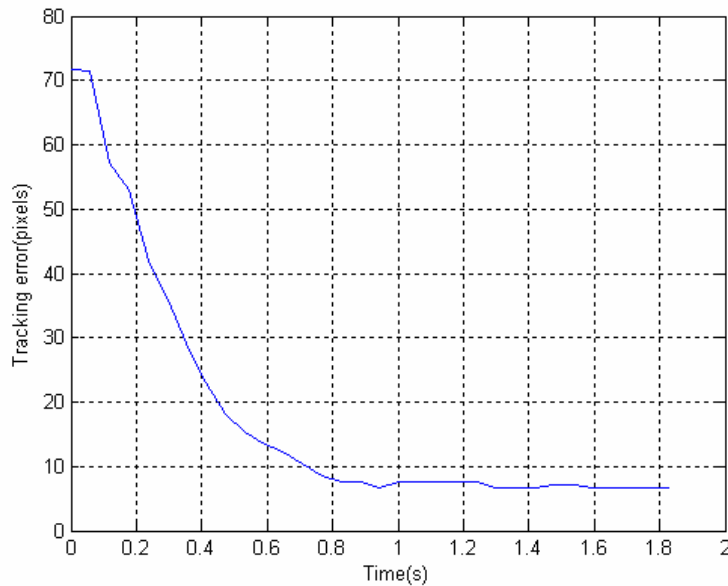
	<i>max</i>	<i>min</i>	<i>medio</i>	<i>desvio</i>
aquisição	32,4020	11,8780	13,6820	2,0275
pyr down	25,9050	9,4730	9,8432	1,6330
processamento	41,6030	9,3320	9,8456	2,4185
centro de massa	3,2550	0,3970	1,3079	0,4478
algoritmo	0,1590	0,0140	0,0154	0,0093
actuação	37,2460	2,1670	4,4850	2,6913
total	118,6600	35,9060	39,1520	7,1468

- Tempos de execução
 - Aquisição e redução de resolução
 - Inclusão do ROI
- Aquisição
 - Caso 3. ~19Hz
 - Caso 4. ~25Hz



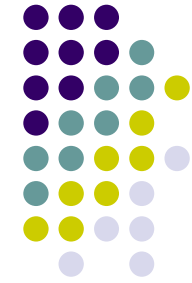


Resultados

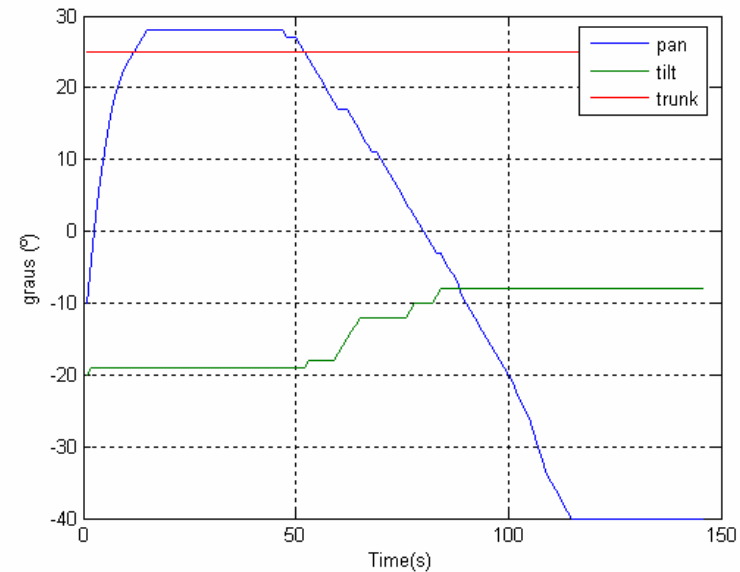
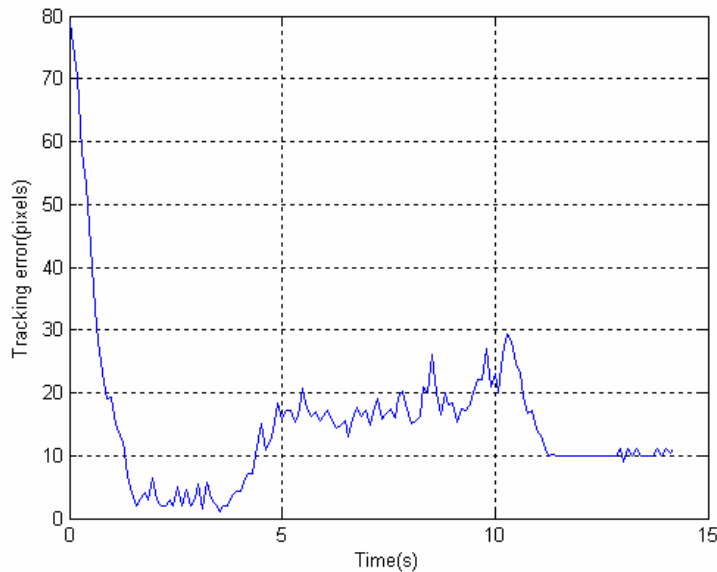


- Alinhamento com a bola
 - Rápido (~1s)
 - Erro em regime estacionário (~7 pixels)



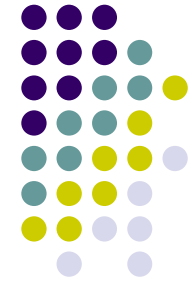


Resultados

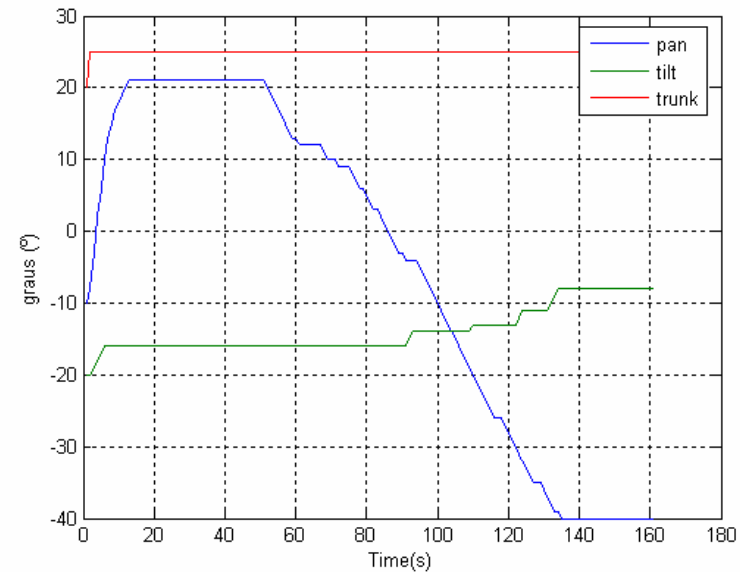
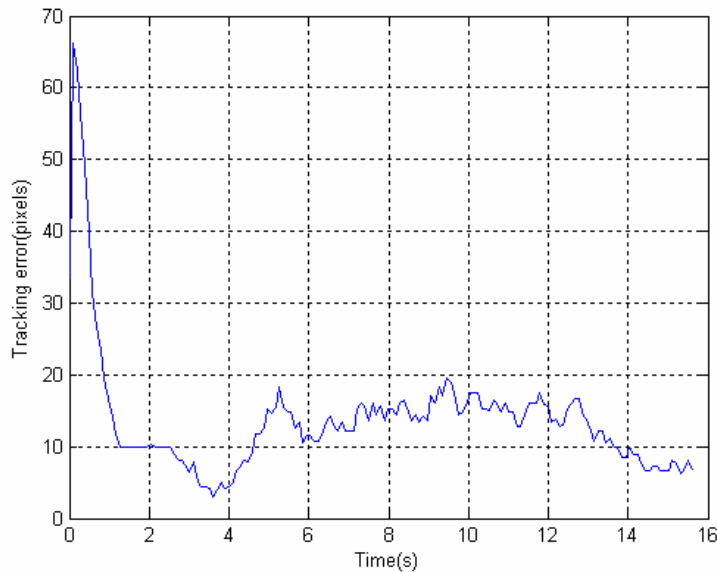


- Seguimento em *Pan* sem aproximação Jacobiana
 - Erro elevado na zona frontal ao robô
 - Algoritmo não compensa a proximidade



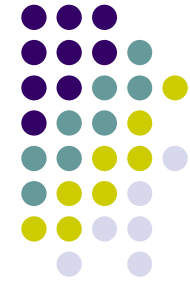


Resultados



- Seguimento em *Pan* com Aproximação Jacobiana
 - Redução do erro na zona frontal
 - Variação da velocidade relativa com a proximidade compensada





Conclusões

- Controlador PC/104 baseado em Linux revelou-se um bom compromisso entre capacidade de processamento, versatilidade, consumo e tamanho.
- É possível efectuar todo o processamento na plataforma, com a integração da Unidade Central, na arquitectura distribuída de controlo já existente.
- Foi criado um ambiente de desenvolvimento versátil que permite desenvolvimento directo na plataforma ou remoto.
- A arquitectura implementada separa o processamento de visão alto-nível, do controlo baixo-nível, em malha fechada dos actuadores. Como consequência, foi possível criar dois métodos para o controlo do seguimento global.
- O sistema de visão atingiu um nível apreciável de funcionamento com a taxa média de processamento do ciclo de controlo a ser aproximadamente 25Hz, um número suficiente para permitir estímulos rápidos e outros tipos de entradas visuais com variação rápida.
- O processo de localização e alinhamento quando a bola se encontra numa posição fixa exhibe valores de erro relativamente baixos, e rápida convergência.





Trabalho Futuro

- Para obter o melhor partido da arquitectura distribuída, é necessário substituir as unidades Master/Slaves por versões mais robustas, compatíveis com os actuais requisitos do sistema.
- Avançar para a detecção de forma para validar a detecção da bola.
- Inferir acerca do papel do o controlador de força local para a compensação do movimento do tronco.
- Estudar a influência do movimento do robô na informação visual e no desempenho do sistema de seguimento. Por exemplo, verificar se o sistema de visão distingue movimento do alvo e o seu próprio movimento.





- Muito obrigado pela atenção.

