

# PMR2440 - Programação para Automação

# Sistemas Embarcados



# Motivações para o curso

- Projetos de sistemas mecatrônicos requerem sistemas embarcados (Embedded systems): Hardware + Software.
- Exemplo de sistemas mecatrônicos:
  - Robôs submarinos,
  - PIGs (Sistemas de inspeção de oleodutos),
  - Máquinas CNC. (Na disciplina PMR2450 Projeto de Máquinas será projetado e construído uma máquina CNC incluindo hardware mecânico, hardware analógico, hardware digital e software).

# Projeto VSOR

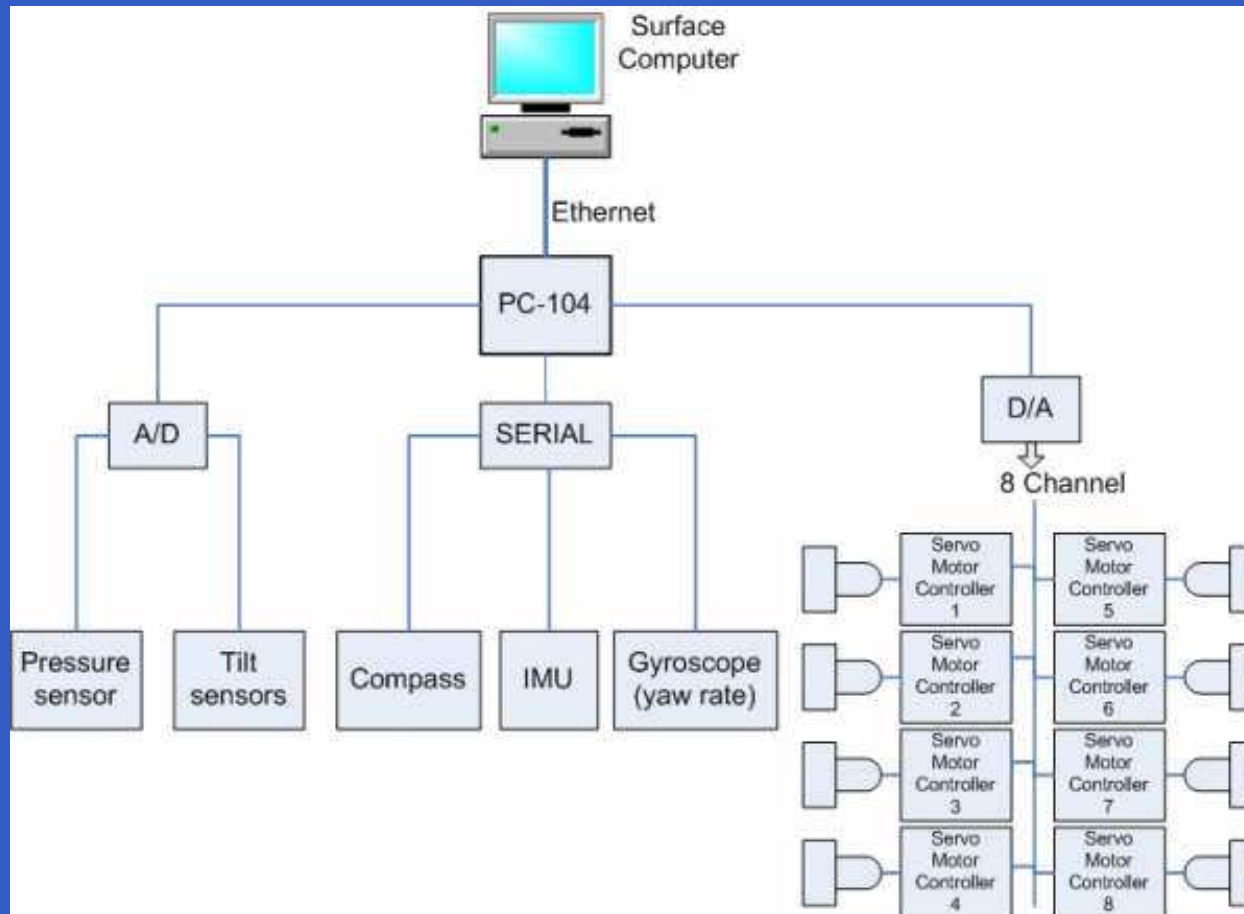
Veículo submarino sendo desenvolvido no Laboratório de Sensores e Atuadores e Laboratório de Sistemas Embarcados



# Sistema sensorial

Variable	Sensor (Manufacturer)	Precision
Heading	Compass TCM2 (PNI)	$\pm 1^{\circ}$
Roll and Pitch	Tilt Series 757 (Applied Geomechanics)	$\pm 2^{\circ}$
Depth	Pressure Sensors, MPX5100DP (Motorola)	3.5cm
Yaw Rate	Fiber Optic Gyro, E-Core 2000 (KVH)	Bias $< 2^{\circ}/h$
Height	Altimeter, PA-200 (Tritech)	1 mm
Linear Acceleration	Inertial Measurement Unit, VG700A (Crossbow)	Bias $< \pm 12mg$
Angular Velocity	Inertial Measurement Unit, VG700A (Crossbow)	Bias $< \pm 20^{\circ}/h$

# Arquitetura de controle atual



# PIGs

## Sistemas de ultra-som para inspeção de oleodutos





# PIGs

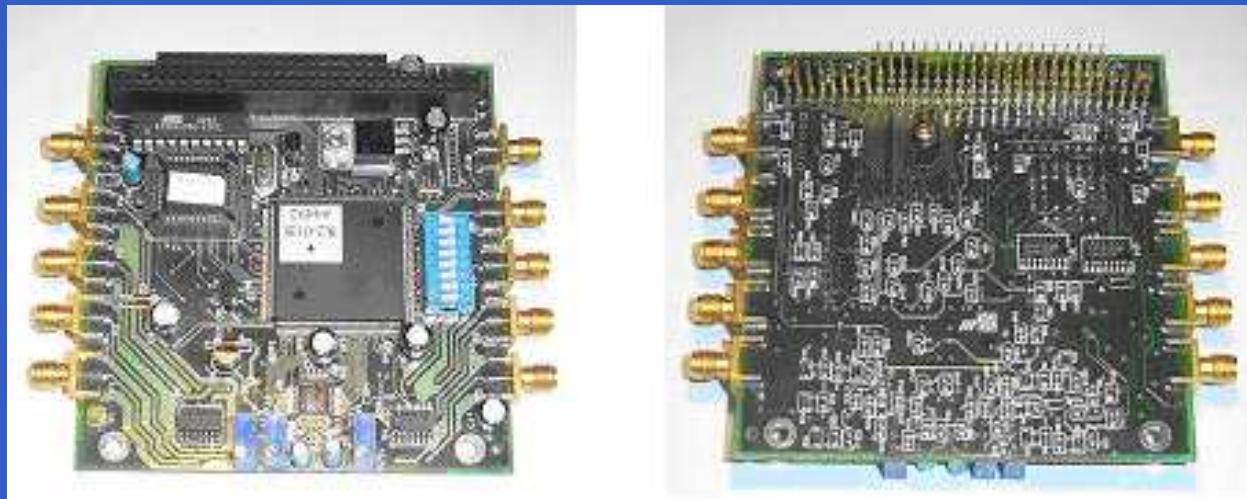
Sistemas de ultra-som para inspeção de oleodutos





# PIGs

O LSA e o LSE desenvolvem eletrônica embarcada para PIGs ultra-sônicos. Na foto projeto desenvolvido para o CENPES/Petrobras.



## Projeto em conjunto com a Empresa Pipeway

- O LSA e o LSE estão desenvolvendo para a Empresa Pipeway eletrônica embarcada para PIGs ultra-sônicos para aplicação em oleodutos e tubulação de gases



# PIG inercial

- O LSA já desenvolveu um estudo para a Petrobras sobre a utilização de sistemas de navegação inercial dentro de PIGs.



# Sistemas embarcados

- Uma possível definição: conjunto *hardware + software* embarcado para o controle de algum sistema.
- Atualmente praticamente todos os equipamentos possuem sistemas embarcados.
- Exemplo: carros (vários microcontroladores), eletrodomésticos em geral (geladeiras, dvd players, tvs, ar condicionados, celulares, controle remotos, etc.)
- Algumas vezes *embedded systems* é traduzido como sistemas embutidos.

# Características

Podemos caracterizar os sistemas embarcados através das seguintes características (não necessariamente simultâneas):

- Dedicados a tarefas específicas,
- Utilizam uma vasta gama de microprocessadores (Intel Pentium IV, Freescale Power PC, AMD, etc.) e microcontroladores (PIC, Intel 8051, AVR, Freescale 6805, 6808, 6811, ARM 7, 9, 11, etc.),
- Restrições de custo,
- Alta confiabilidade,
- Ambiente de operação não controlado,

# Características ...

- Restrições de recursos (memória, tempo de execução, etc.),
- Ferramentas de desenvolvimento especializadas,
- Linguagens de programação, em geral: Assembly, C/C++, Java, ADA (para sistemas muito sofisticados, aviãoica por exmplo),
- Sistemas operacionais: Windows CE, RT-Linux, QNX, VxWorks.



# Característica principal

- Um sistema embarcado efetivamente controla um sistema.
- O controle em geral engloba dois aspectos:
  - Controle de malha fechada: execução cíclica de controladores com período de tempo definido (um controlador PID por exemplo). Em geral representado por equações de diferença.
  - Controle de eventos discretos: resposta do controlador quando da ocorrência de eventos internos ou externos. Pode ser representado por autômatos ou redes de Petri por exemplo.

# Sistemas não críticos

Alguns sistemas não possuem caráter crítico (*non safety critical systems*), ou seja, se o sistema falhar não existem grandes conseqüências. Por exemplo:

- DVD player,
- Trava e alarme eletrônico dos carros.

# *Safety critical systems*

- Alguns sistemas são ditos críticos devido ao elevado custo e/ou devido ao risco que expõe a vidas humanas.
- Exemplos: aviões, foguetes, mísseis, equipamentos de radioterapia, etc.

# Sistemas operacionais

- O sistema operacional intermedia as ações entre o usuário e o hardware.
- Permite a utilização do *hardware* (incluindo todos os periféricos) de maneira eficiente.
- Permite executar vários programas simultaneamente.
- Sistemas embarcados simples (Ex: DVD player) não requerem sistema operacional.

# Sistemas operacionais ...

- Exemplos de sistemas operacionais: Windows (Arrrghhh !!!!), Unix System V, OS X, Linux, Windows-CE, RT-Linux, QNX, Vxworks, etc.
- Sistemas embarcados necessitam de sistemas operacionais com *small footprint* (código reduzido), já que em geral não há HDs, vídeo, floppy, etc.
- A principal atividade neste caso é o escalonamento de processos, ou seja, um algoritmo que decide quais programas, quando e por quanto tempo eles executam.

## Sistemas Operacionais Multi-tasking

- Em geral *tasks*, processos, e programas se referem as entidades que são executadas simultaneamente. Threads são como processos dentro de processos.
- Somente um processo pode rodar no sistema de cada vez. O escalonador decide como chavear a execução entre os processos, inclusive quando ele mesmo é executado.
- O sistema operacional é o gerenciador do sistema, mas criador e criatura se revezam no microprocessador e microcontrolador.



# SO de Tempo Real

- Sistema embarcados em geral requerem sistemas operacionais de tempo real.
- Estes possuem mecanismos para garantir que deadlines de tarefas sejam respeitados.
- Exemplo: QNX, RT-Linux, Vxworks.

## Hard Real Time versus Soft Real Time

- Hard real time: tarefas (tasks) devem ser terminadas em horizontes de tempo (*deadline*) bastante rígidos,
- soft real time: tarefas podem infringir *deadlines* sem que haja degradação significativa de performance

# Falha de software 1: Therac-25

- Entre 1985 e 1987 uma máquina de radioterapia denominada Therac-25 projetada pela empresa Atomic Energy Canada Limited (AECL) causou seis acidentes envolvendo doses maciças de radiação. Cinco pessoas morreram devido a este problema.
- Uma situação de *racing condition* entre várias atividades concorrentes no programa de controle permitia que o canhão de Raio-X operasse sem que o paciente estivesse devidamente posicionado.

## Falha de software 2: Primeiro vôo Ariane 5

- Em 4 de Junho de 1995, um erro de software levou à destruição do foguete Ariane 5, cerca de 40 segundos após o lançamento.
- Uma conversão de ponto flutuante para inteiro de 16 bits com sinal resultou em overflow (o número resultou maior que +32767) e um erro em tempo de execução fez com que tanto o computador principal quanto o reserva parassem (*shutdown*).

## Falha de software 2: Primeiro vôo Ariane 5 ...

- Sistema de controle de trajetória não conseguiu mais atuar e os controladores decidiram explodir o foguete.
- Tempo de desenvolvimento do foguete, 10 anos, custo 7 bilhões de dólares. Carga: 4 satélites de pesquisa, custo 500 milhões de dólares.

- 
- 
- 

## Falha de software 3: Problemas marcianos ...

### The Mars Pathfinder Mission

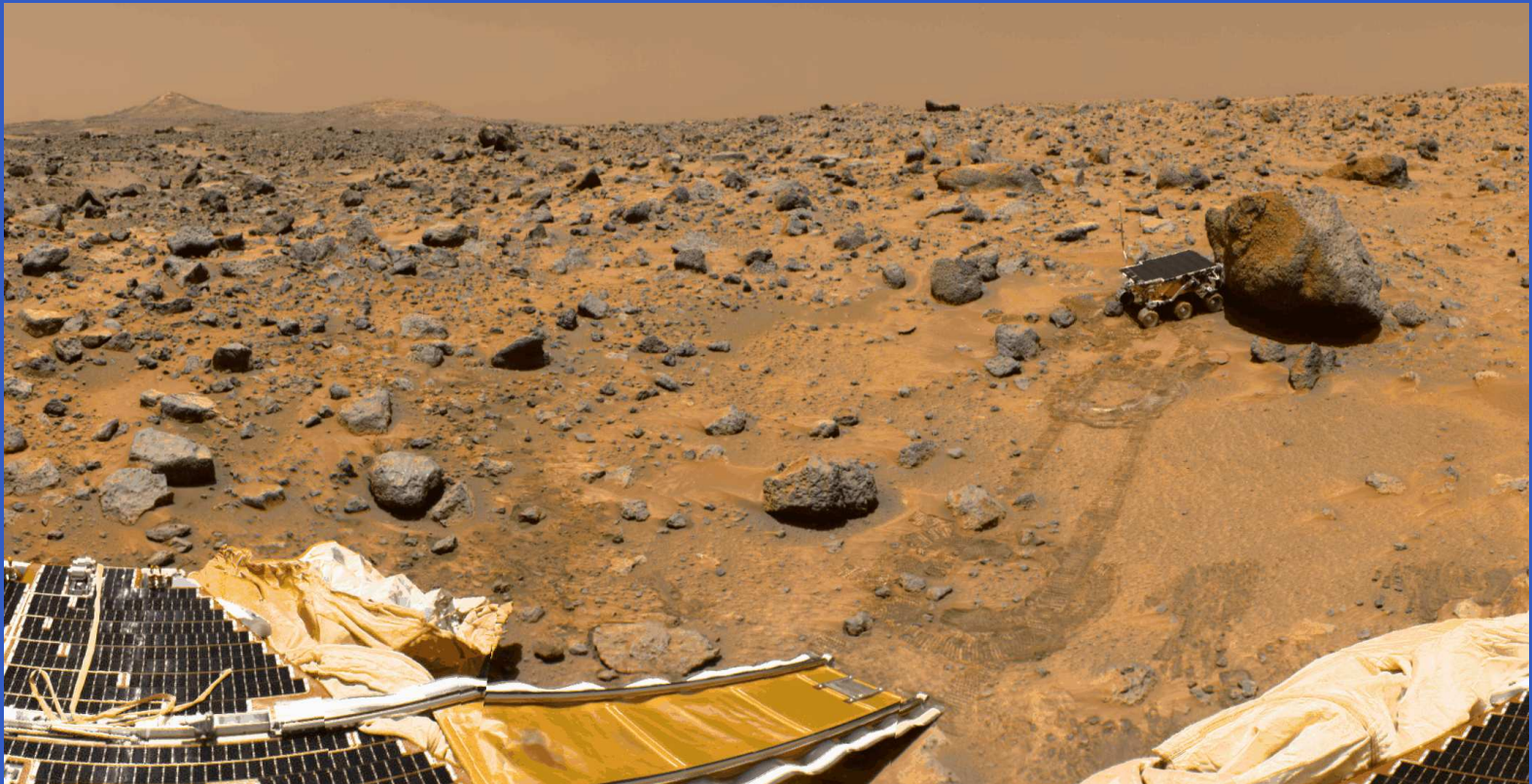




- 
- 
- 

## Falha de software 3: Problemas marcianos ...

### The Mars Pathfinder Mission



## Falha de software 3: Problemas marcianos ...

- A missão Mars Pathfinder consistindo de uma nave espacial desceu no solo de Marte em 04 de Julho de 1997. A nave ainda liberou no solo Marciano o robô Sojourner Rover.
- Um erro no software provocava um reset do sistema e perda total dos dados.

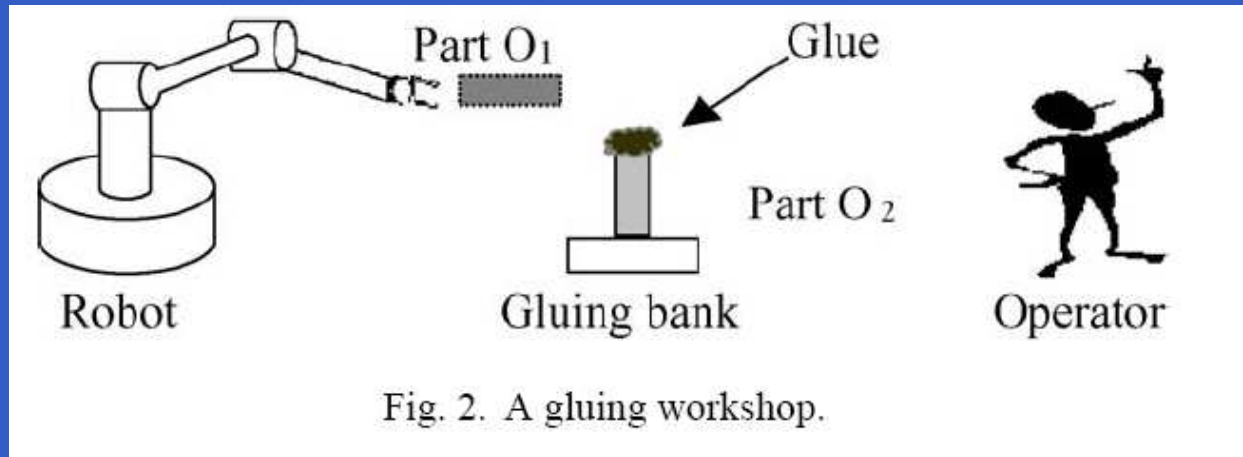
## Falha de software 3: Problemas marcianos ...

- A nave utilizava o sistema operacional de tempo real Vxworks que utiliza escalonamento preemptivo com prioridades (oops preemptivo ????).
- Eventualmente, quando um processo de média prioridade estava sendo executado e um processo de maior prioridade estava na fila para ser executado. Um outro processo (*watchdog*) monitorava se o processo de maior prioridade estava sendo executado ou não e resetava todo o sistema caso o tempo de espera era excessivo.

## Falha de software 3: Problemas marcianos ...

- Em terra, os engenheiros da NASA conseguiram reproduzir o erro e graças a características especiais do SO Vxworks que produz o logging de todos os processos, foi possível determinar que se tratava de um fenômeno denominado *Inversão de Prioridade*
- Foi possível corrigir o problema, e fazer um *upload* da nova versão do programa.

# Um exemplo de requisitos - Colagem industrial



## Colagem industrial

- Objetivo: colar  $O_1$  em  $O_2$ ,
- $O_1$ :
  - transportada e colocada sobre  $O_2$  pelo robô,
  - robô precisa de 4-5 unidades de tempo para transportar  $O_1$ ,
  - robô deve iniciar sua tarefa dentro de 2 unidades de tempo desde o início do processo.

# Colagem industrial ...

- $O_2$ 
  - Preparada pelo operador (passa cola),
  - Operador precisa de 1 unidade de tempo para passar a cola.

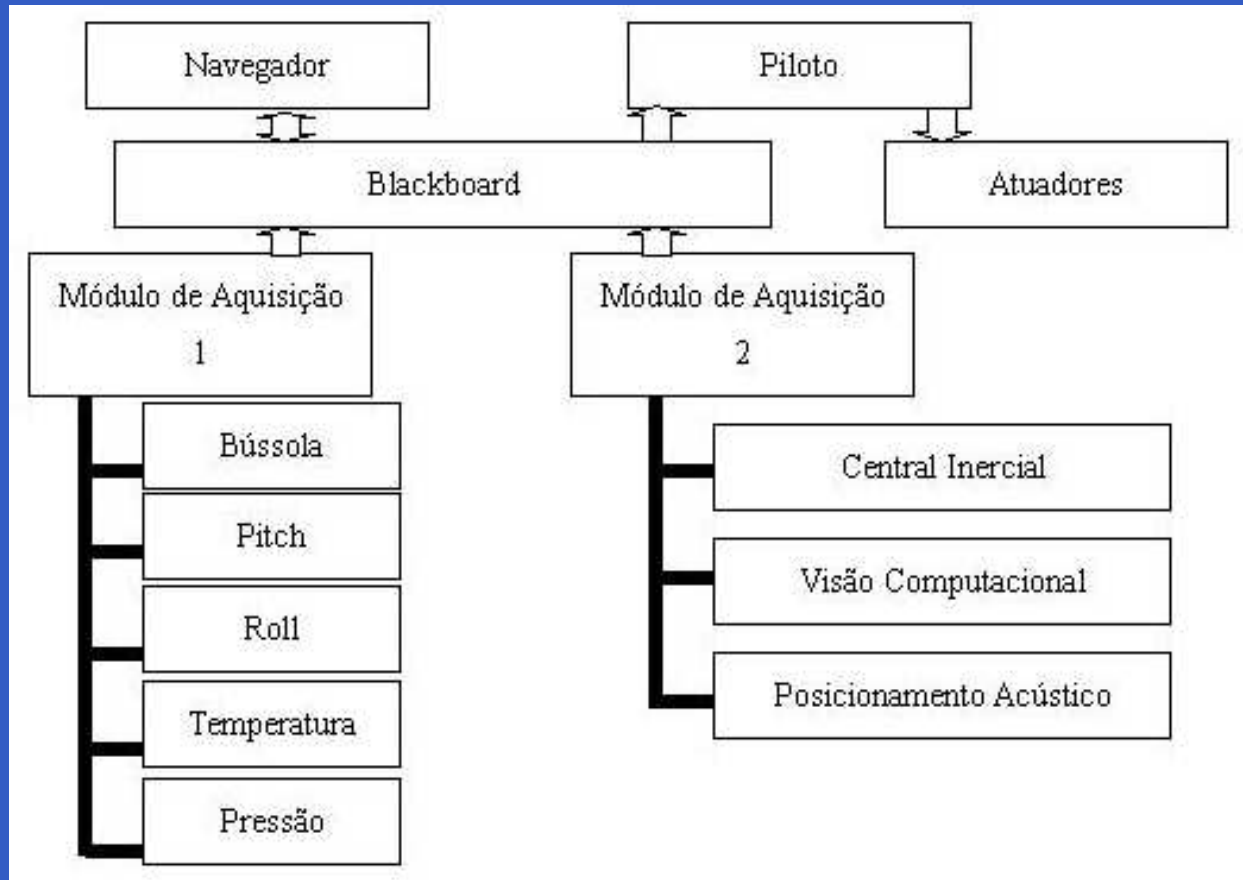
# Colagem industrial ...

- Colagem:
  - Cola ficar pronta para colagem em 3 unidades de tempo (desde o início de sua colocação),
  - Permanece boa para colagem durante 3,5 unidades de tempo.

A solução para este problema é bastante complexa. Verifique o artigo correspondente no site da disciplina.



# Software embarcado para robôs móveis



## Software embarcado para robôs móveis ...

Tabela 1: Módulo de aquisição 1.

Sensor	Tempo de Processamento (ms)	Periodicidade (ms)
Bússola	10.0	25.0
Pitch	8.0	25.0
Roll	5.0	50.0
Temperatura	4.0	50.0
Pressão	2.0	100.0

Tabela 2.a: Módulo de aquisição 2 – Modo de Operação 1.

Sensor	Tempo de Processamento (ms)	Periodicidade (ms)
Central Inercial	32.0	80.0
Visão Computacional	5.0	40.0
Posicionamento Acústico	4.0	16.0

## Software embarcado para robôs móveis ...

Tabela 2.b: Módulo de aquisição 2 – Modo de Operação 2.

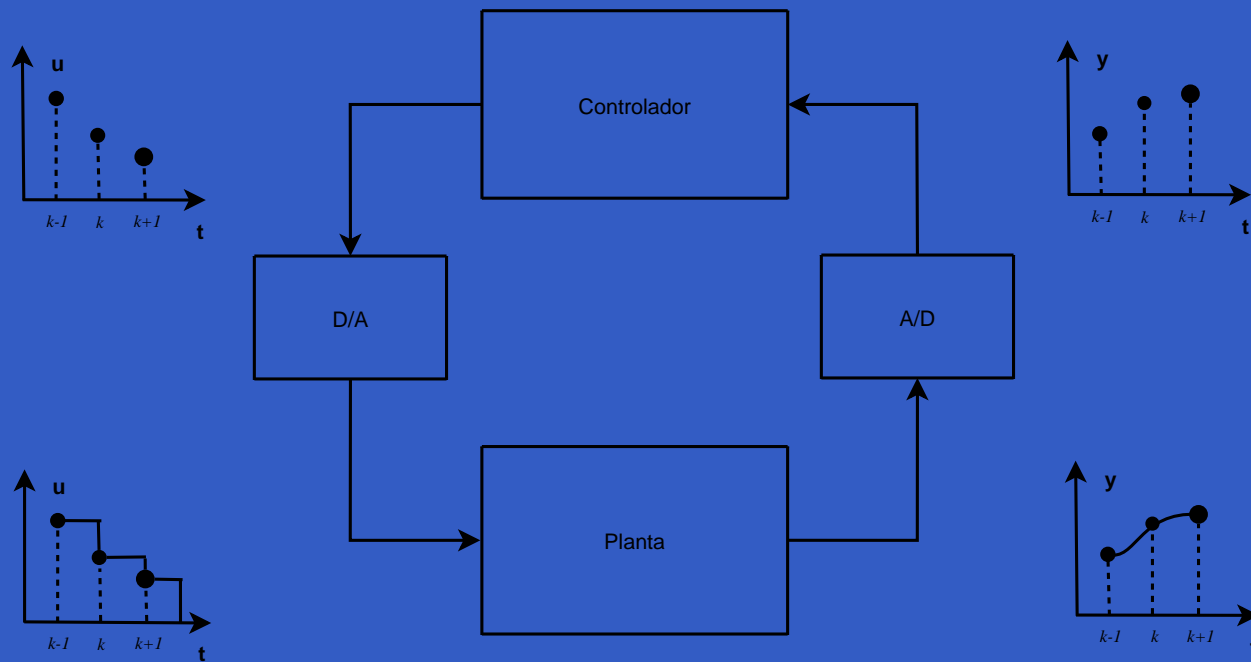
Sensor	Tempo de Processamento (ms)	Periodicidade (ms)
Central Inercial	12.0	50.0
Visão Computacional	10.0	40.0
Posicionamento Acústico	10.0	30.0

Tabela 2.c: Módulo de aquisição 2 – Modo de Operação 3.

Sensor	Tempo de Processamento (ms)	Periodicidade (ms)
Central Inercial	40.0	80.0
Visão Computacional	10.0	40.0
Posicionamento Acústico	5.0	20.0

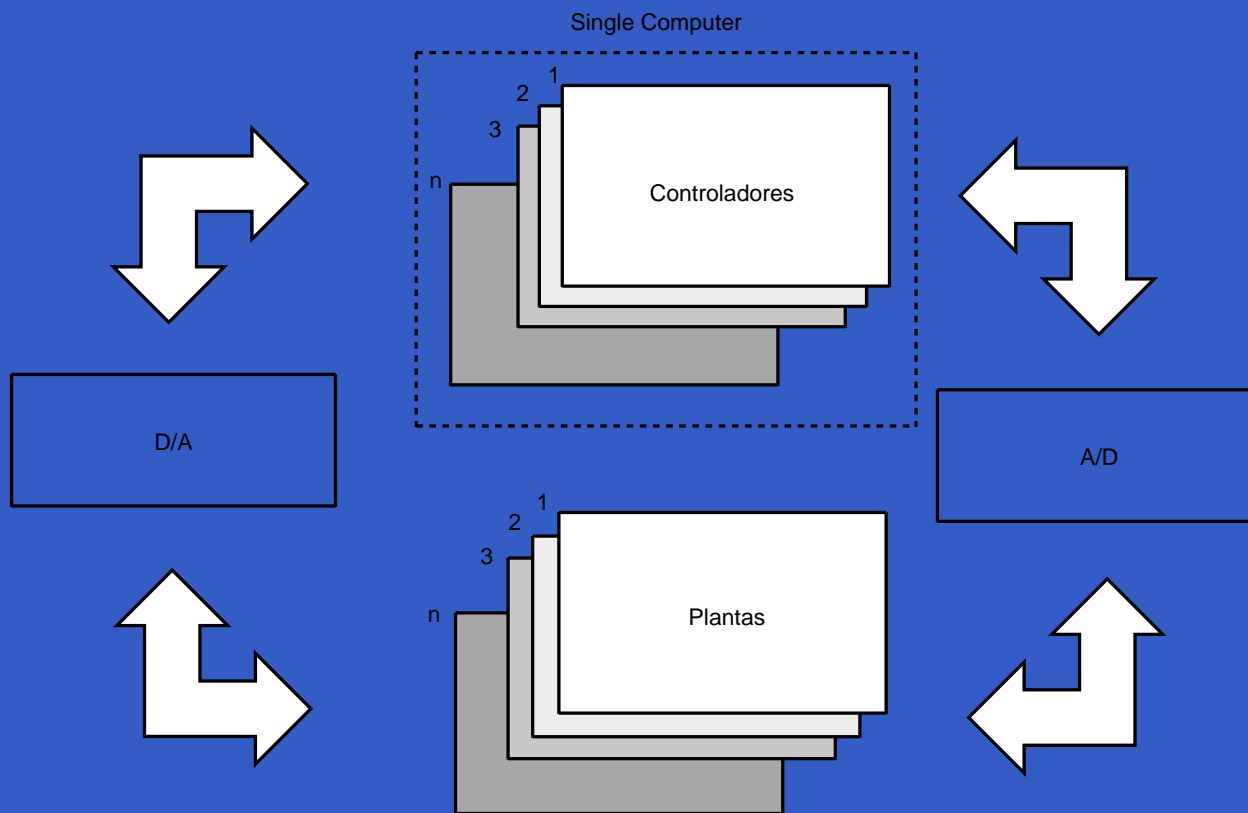
# Sistemas de controle distribuídos em redes

## A visão tradicional de sistemas de controle



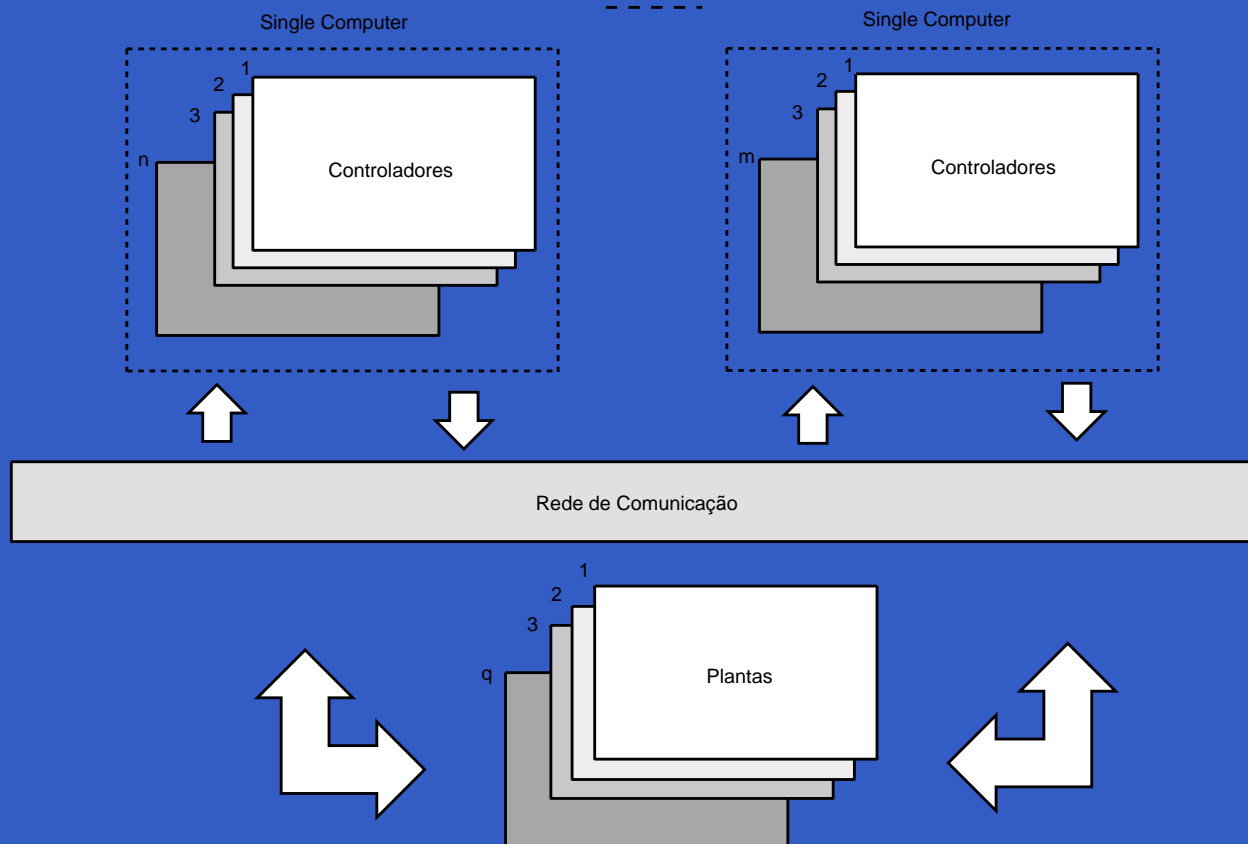
# Sistemas de controle distribuídos em redes ...

## Múltiplos processos, processador único



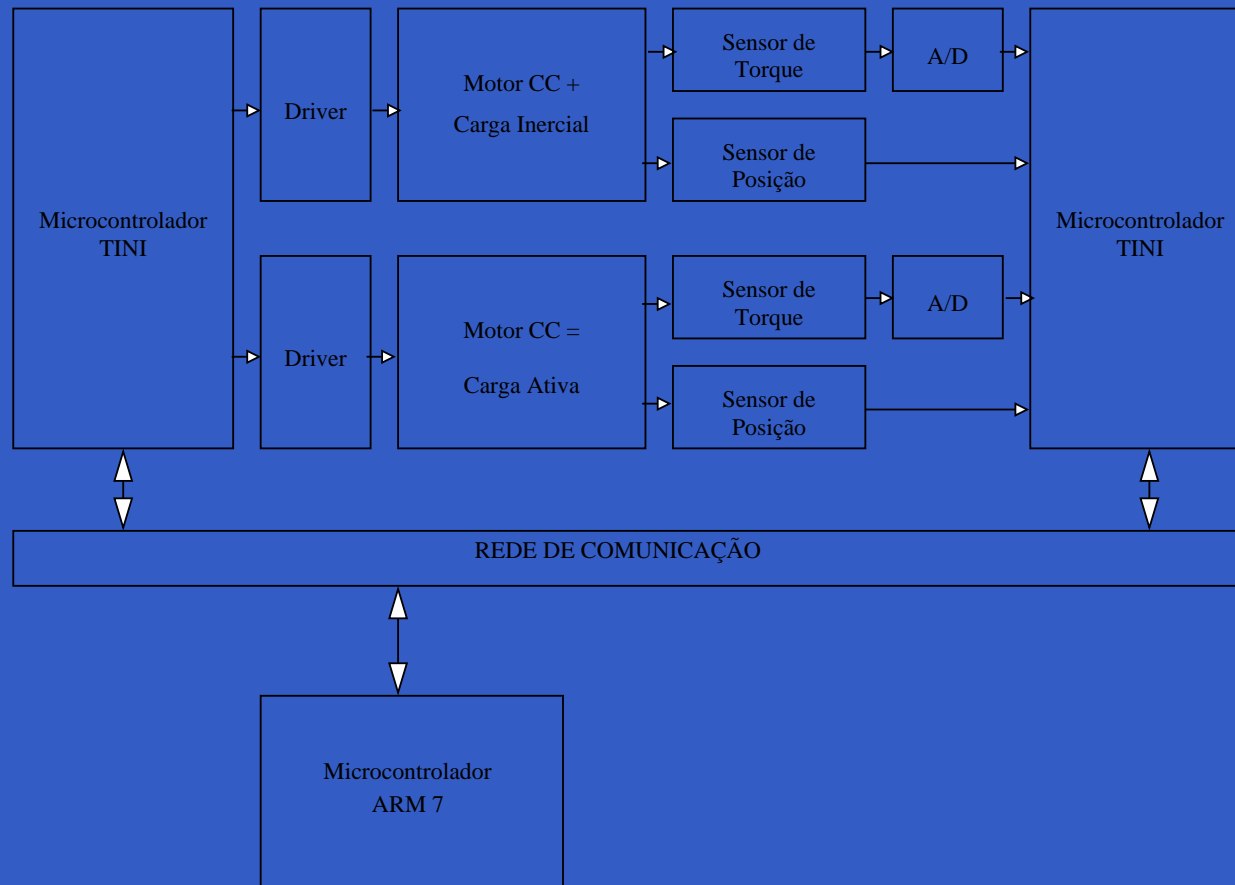
# Sistemas de controle distribuídos em redes ...

## Múltiplos processos, múltiplos processadores



# Sistemas de controle distribuídos em redes ...

## Proposta de implementação



# O futuro do VSOR

## Proposta da nova arquitetura de controle do VSOR

