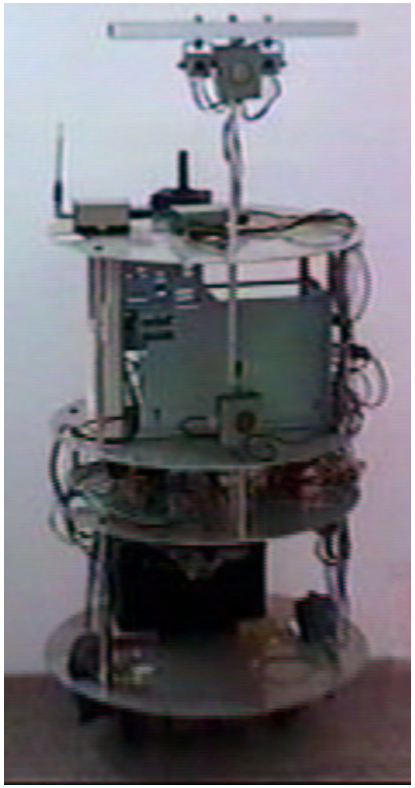
Retrofitting de Robôs

Walter Fetter Lages
Universidade Federal do Rio Grande do Sul
Departamento de Engenharia Elétrica
fetter@eletro.ufrgs.br

sicompi 2003

Introdução

- Robôs
  - Manipuladores
    - Robôs Industriais
  - Móveis
    - Rodas
    - Pernas
    - Esteiras



Introdução

- Pequenas e médias empresas não tem acesso à tecnologia de robótica devido aos altos custos
  - Custos dos periféricos como módulos de I/O e de comunicação
  - Custos da programação em linguagens dedicadas
- Grandes empresas estão trocando robôs velhos
  - A mecânica dos robôs não tem mudado muito
  - A eletrônica e o *software* pode ser atualizado para a tecnologia atual

Retrofitting de Robôs

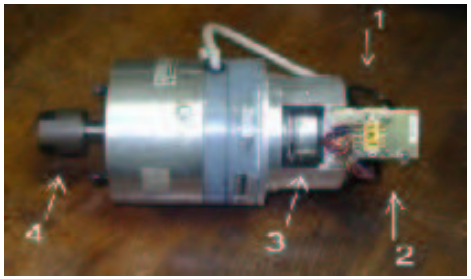
- Revisão Mecânica
- Atualização Elétrica
- Substituição do Controlador
  - Arquitetura Aberta
  - Baseada em sistema distribuído
- Substituição do *Software*
  - C++ e biblioteca de classes adequada
  - Programação *Off−line*

Mecânica do ASEA IRB6

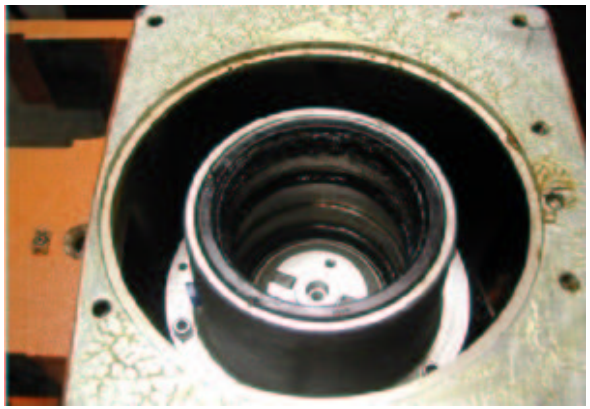
- Construído em 1977
- 5 juntas
- Totalmente desmontado
  - Identificação das peças
  - Limpeza
  - Lubrificação
  - Algumas peças substituídas

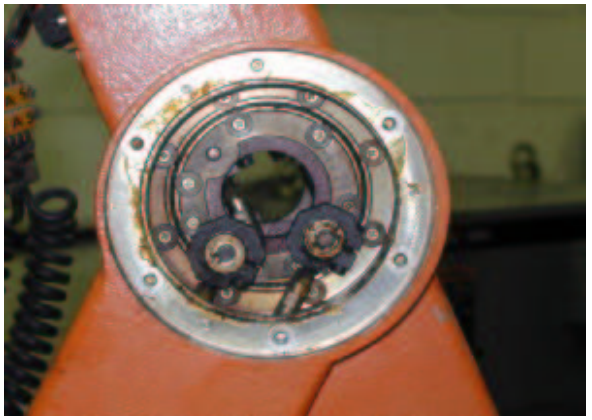
Desmontagem do Robô

- Instrumentação das juntas
  - Motor D.C.
  - Resolver
  - Tacômetro
  - Chave de fim−de−curso
  - Frio eletromecânico



Desmontagem





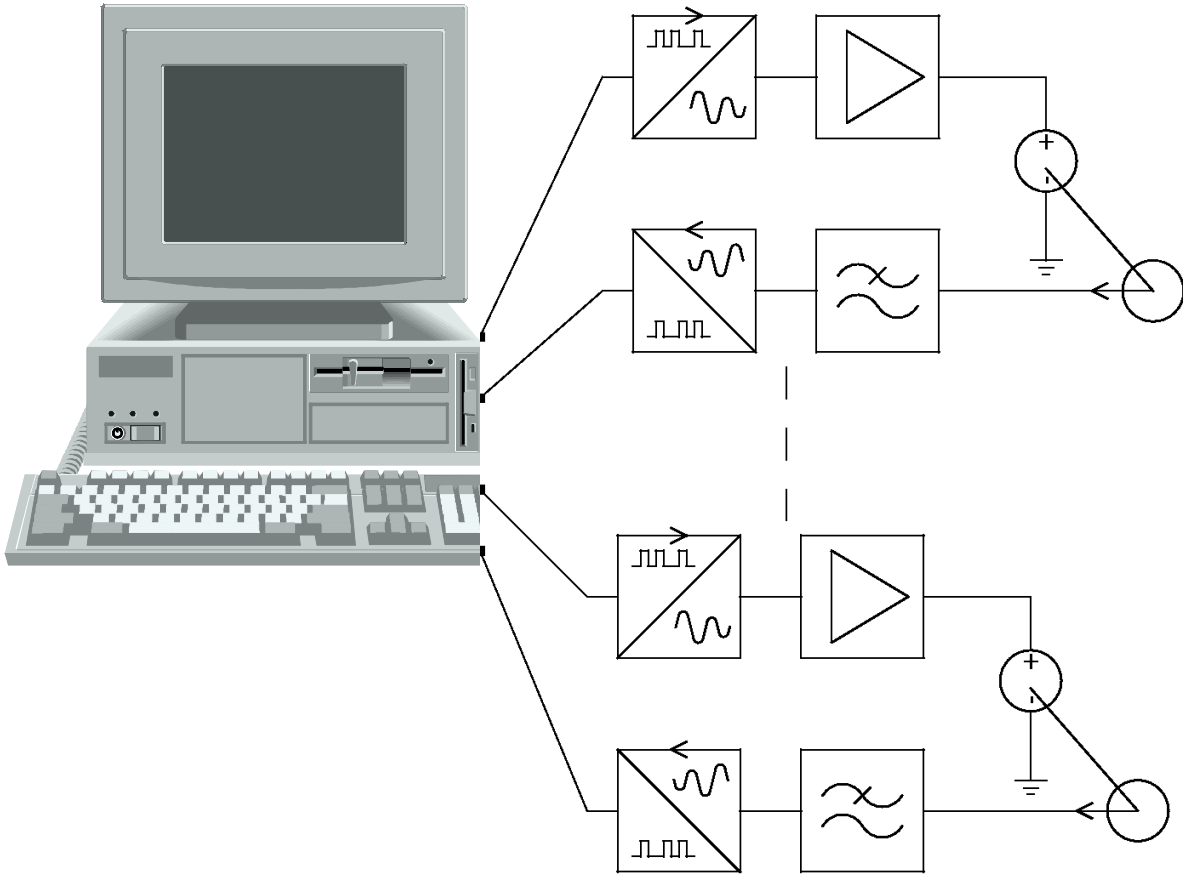
Atualização Elétrica

- Controlador não operante
  - Baseado em eletrônica analógica antiquadacs
  - Esquemáticos não disponíveis
  - Fonte reprojetada
  - Resolvers e tacômetros substituídos por encoders incrementais

Arquitetura de Controle

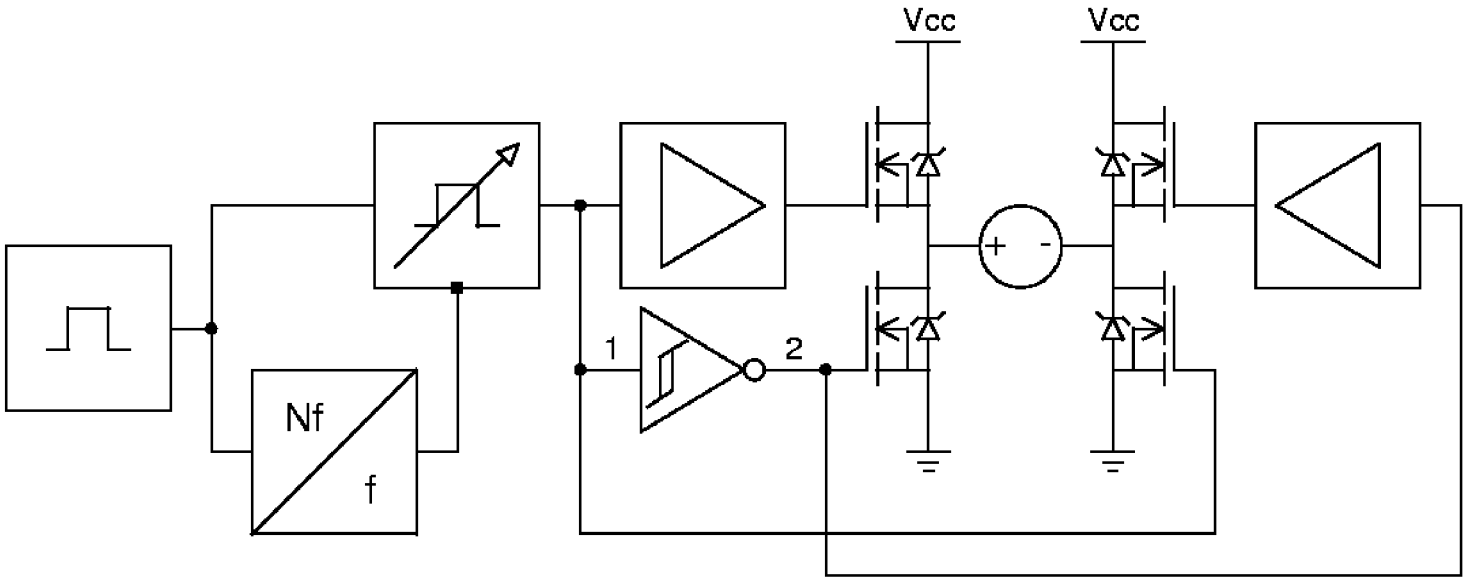
- Arquitetura de controle desenvolvida pra o robô Janus
  - 2 braços e cabeça de visão stereo
  - 8 juntas/braço
- Nenhuma adaptação foi necessária para utilizar no ASEA IRB6

Arquitetura de Hardware Típica

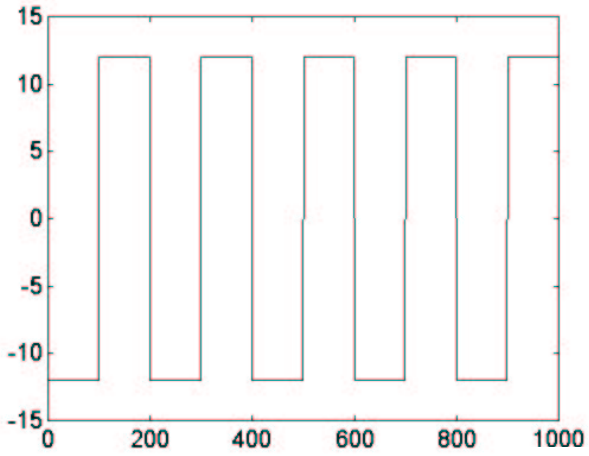


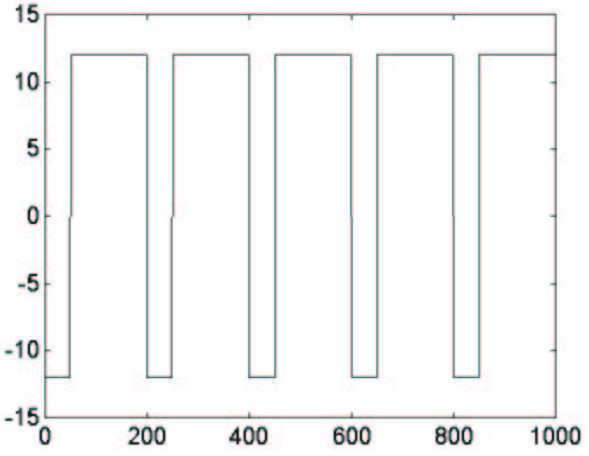
Acionamento por PWM Digital

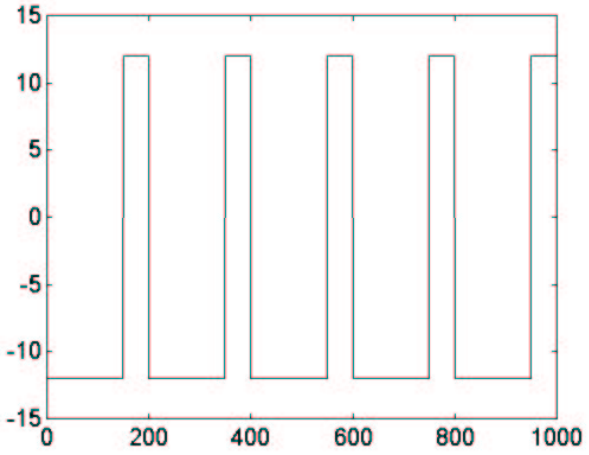
- Acionamento totalmente digital
- Freqüência do PWM pode ser programada



Pulse Width Modulation

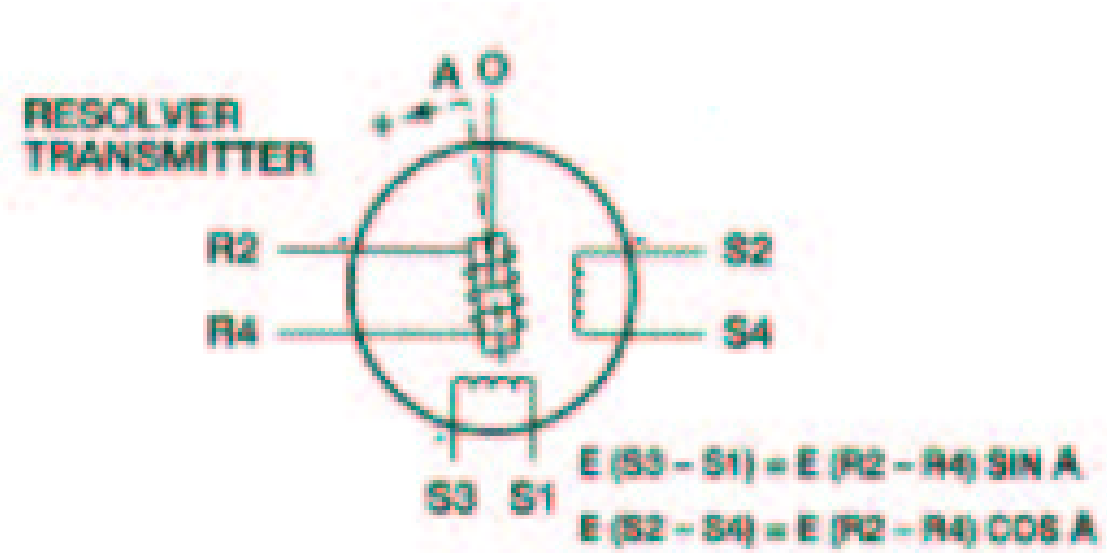




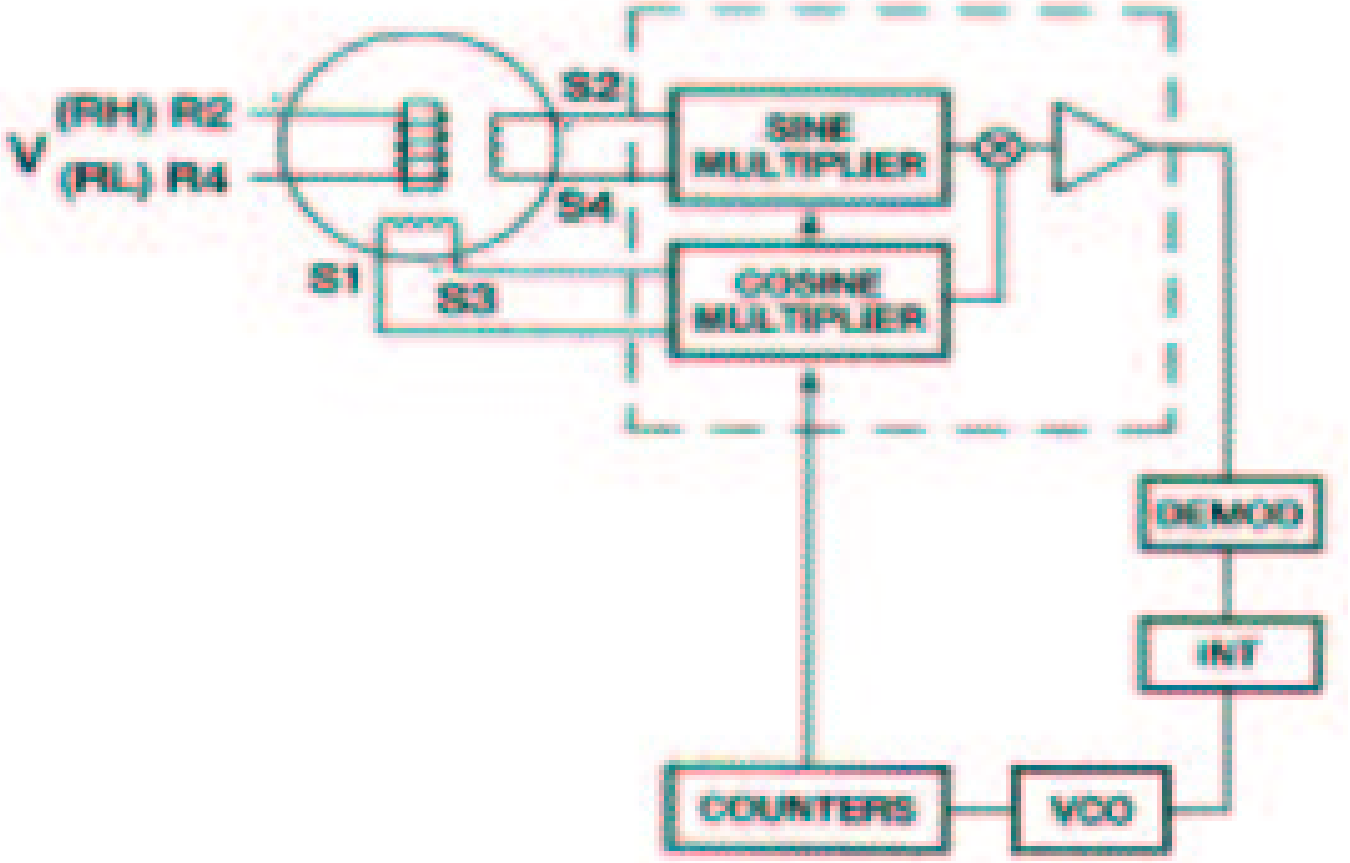


Resolvers

- Sinal de excitação de 400Hz
- V1= Vsen(wt)senA
- V2= Vsen(wt)cosA
- Processamento
  - por demodulação
  - Por amostragem

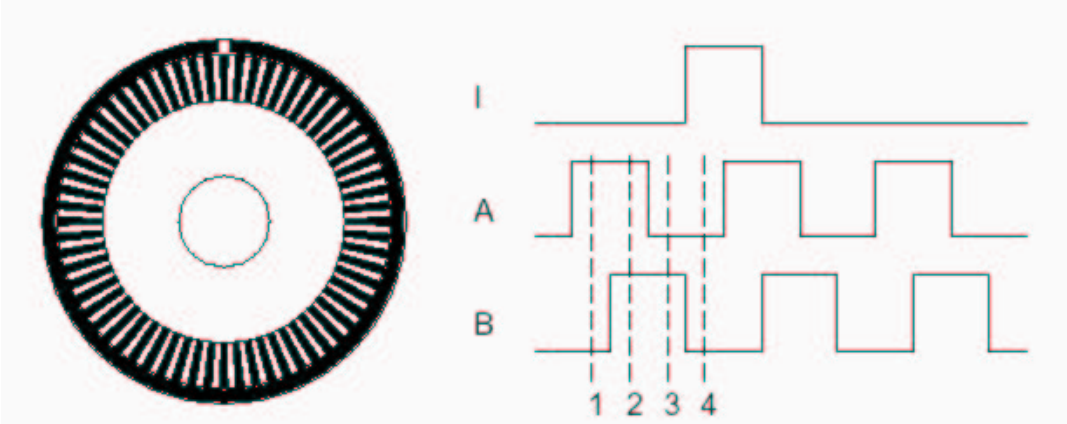


Demodulação do Resolver

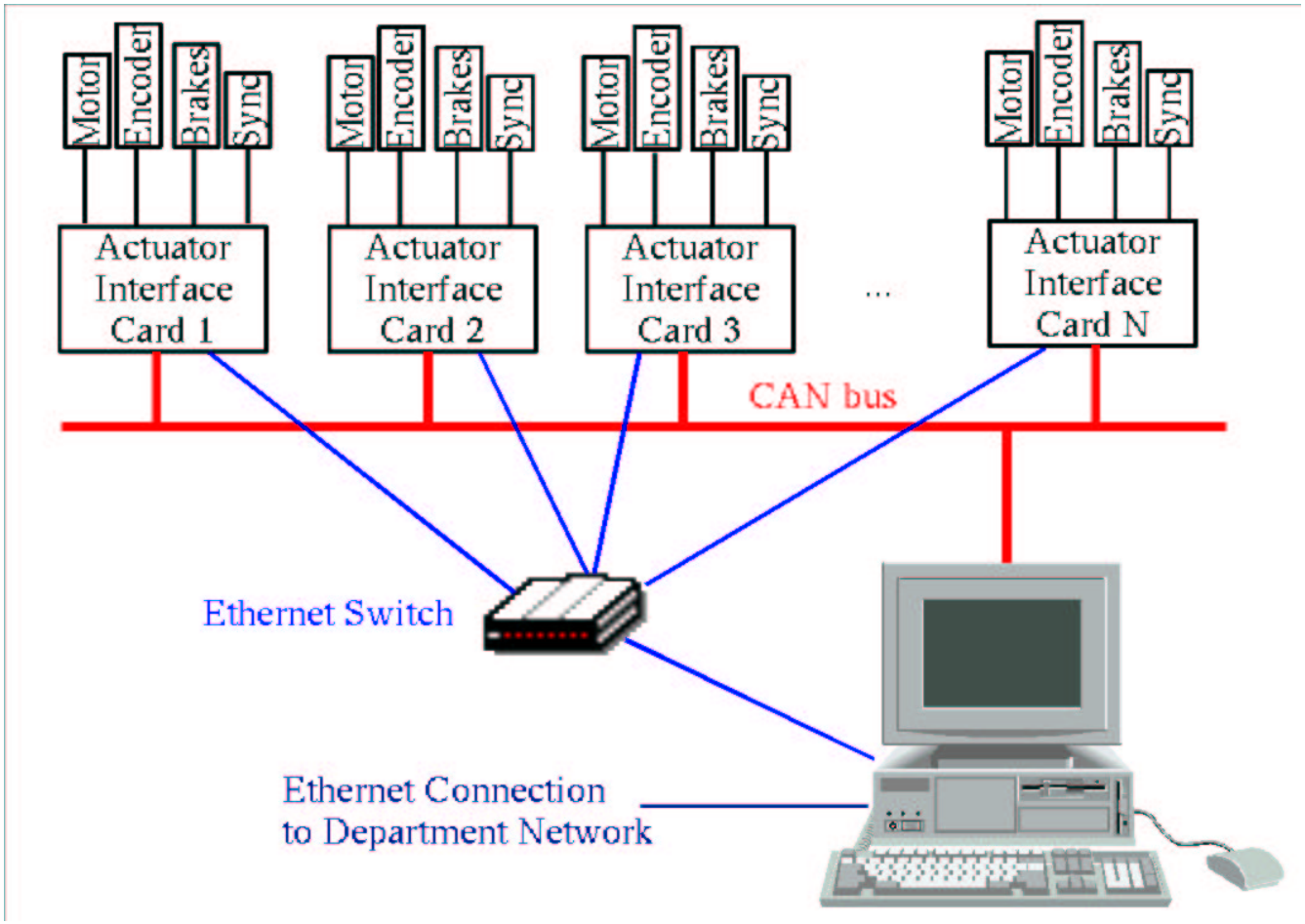


Encoder Óptico Incremental

- Permite altas resoluções
- Requer sensor de índice
- Decodificação em quadratura permite multiplicar por 4 a resolução do disco
- Contagem e decodificação deve ser feita por hardware



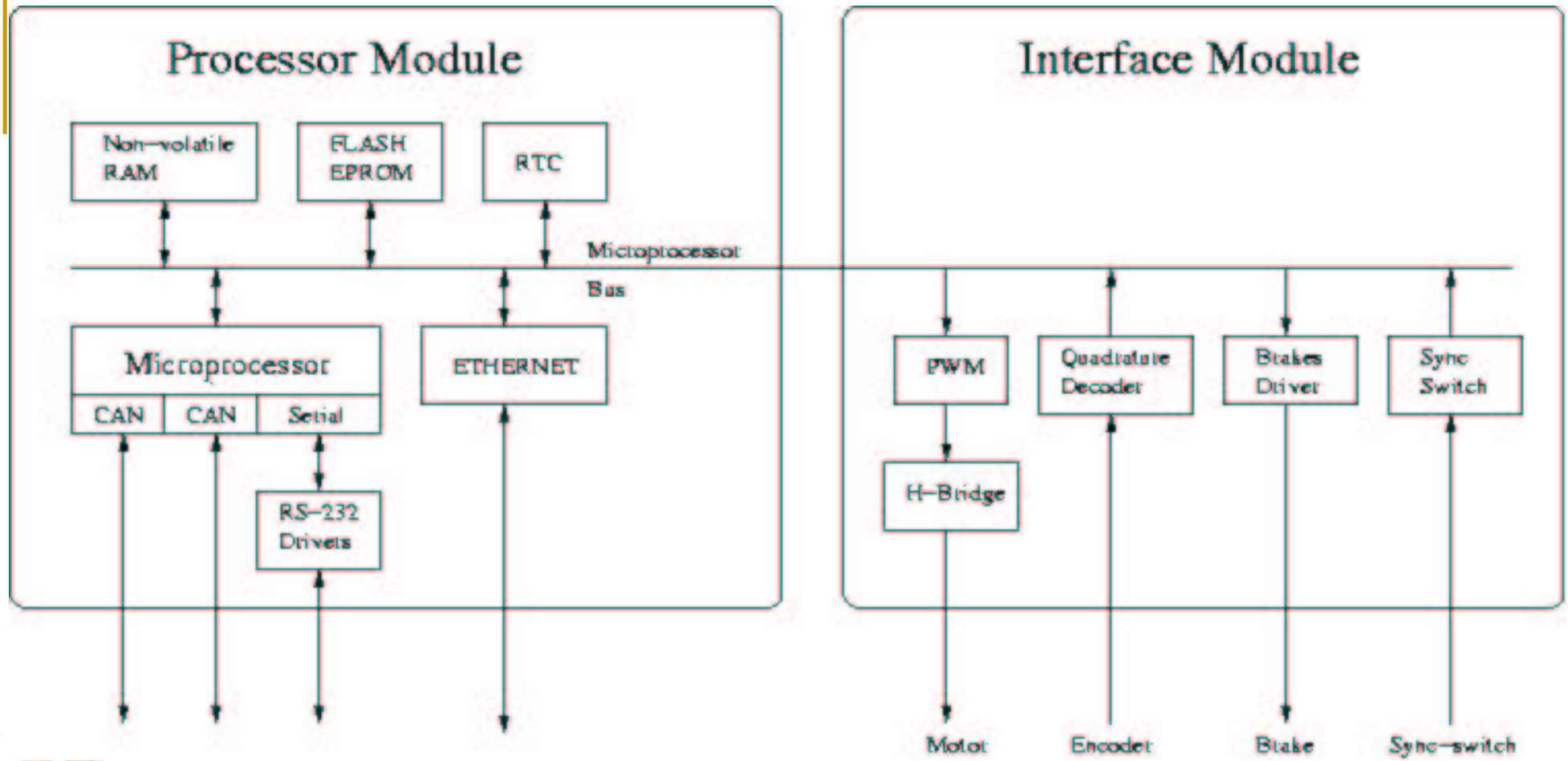
Arquitetura de Controle



Características da Arquitetura

- CAN utilizado para dados em tempo−real
- Ethernet utilizada para dados de supervisão
- Interface como usuário executa em um PC
- A função das AIC é flexível
  - Deve ler os sensores e acionar o motor
    - Pode funcionar como um controlador local de juntas (PID)
    - Pode funcionar como um processador de I/O

Actuator Interface Card (AIC)



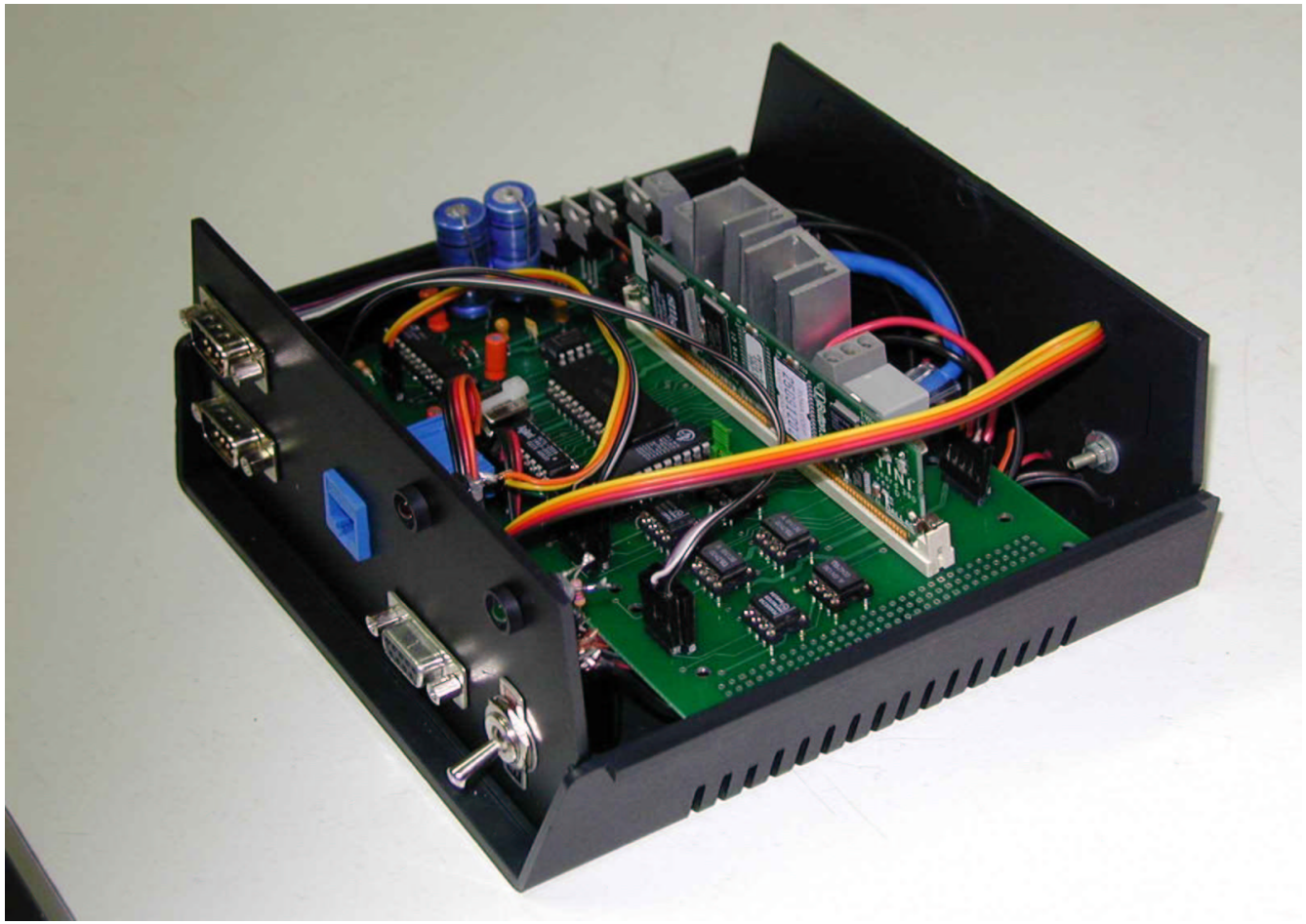
Módulo Processador

- Módulo TINI da Dallas Semiconductor
  - Processador
  - Real−time clock
  - Interfaces
    - CAN
    - Ethernet
    - RS −232
  - SIMM72
  - Pode ser substituída mantendo−se o módulo de interface

Módulo de Interface

- Soquete SIMM72 para o módulo processador
  - PWM com freqüência programável
  - Ponte H com MOSFETs
  - Decodificador em Quadratura para encoder
    - Mutiplica por 4 a resolução do encoder
    - Contador de 16 bits para acumular os pulsos entre leituras
  - Interface para chave de fim−de−curso
  - Interface para freio

AIC



Software

- AIC executa um sistema operacional multitarefa
  - Gerenciamento de memória e I/O
  - Sistema de arquivos
  - Pilha TCP/IP
  - Máquina virtual Java
  - Shell Unix−like
  - Servidores de TELNET, FTP e console serial
  - Cliente DHCP

Pacote Java AIC

- br/ufrgs/eletro/AIC
  - Classes para modelar os dispositivos conectados à AIC
    - PWM, Motor, Encoder, Brake, Index
    - Métodos públicos para suportar as operações possíveis
      - Brake.apply(), Brake.release()
      - Motor.on(), Motor.off(), Motor.set(double voltage)
  - A classe Host abstrai a comunicação
    - HostCAN, HostUDP
  - Métodos críticos implementados em Assembly como métodos nativos

Daemon AIC

- Usa o pacote AIC e implementa os serviços da AIC
  - Controlador local de junta
  - Processador de I/O
- Carregado na AIC por FTP
- Disparado pelos scripts de inicialização
- AIC possui memória não volátil, portanto o daemon está pronto assim que a a AIC é ligada

AIC Daemon Utilizado no ASEA IRB6

- Implementa um processador de I/O
- Programa Java Multi−threaded
- A thread principal trata os argumentos e dispara duas outras threads
  - SendSensors
    - Envia a leitura dos sensores a cada 10ms
  - GetCommand
    - Recebe comandos do host e executa−os

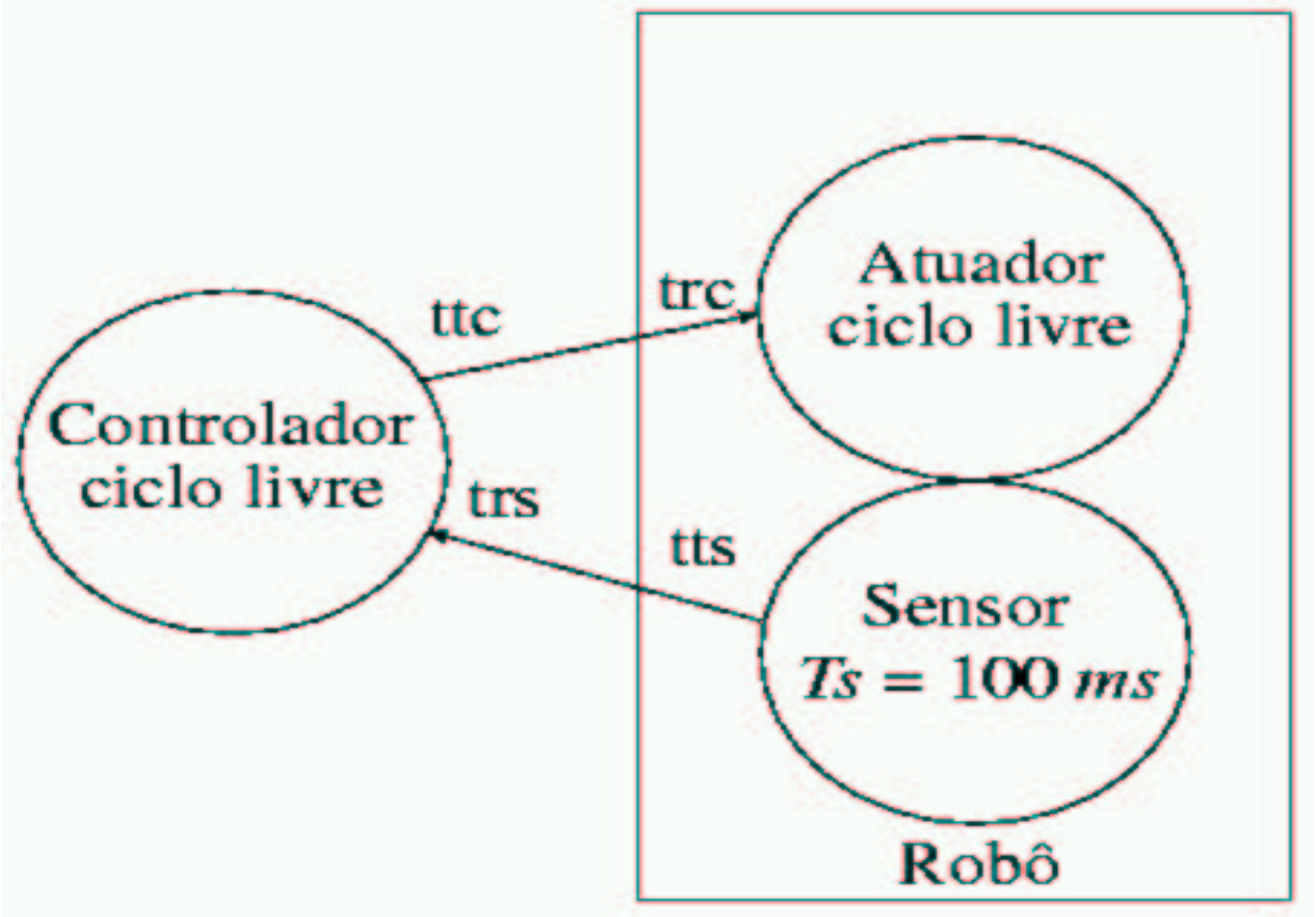
Software do Host

- Biblioteca AIC em C++, semelhante em estrutura ao pacote Java AIC
  - Classes para modelar Motor, Encoder, Freio e Índice
  - Classe AIC derivada para AIC_CAN e AIC_UDP
    - Detalhes de comunicação encapsulados
  - Programador não necessita estar consciente da arquitetura distribuída
  - Classes e métodos são similares aos existentes nas AICs mas implementados em C++ ao invés de Java

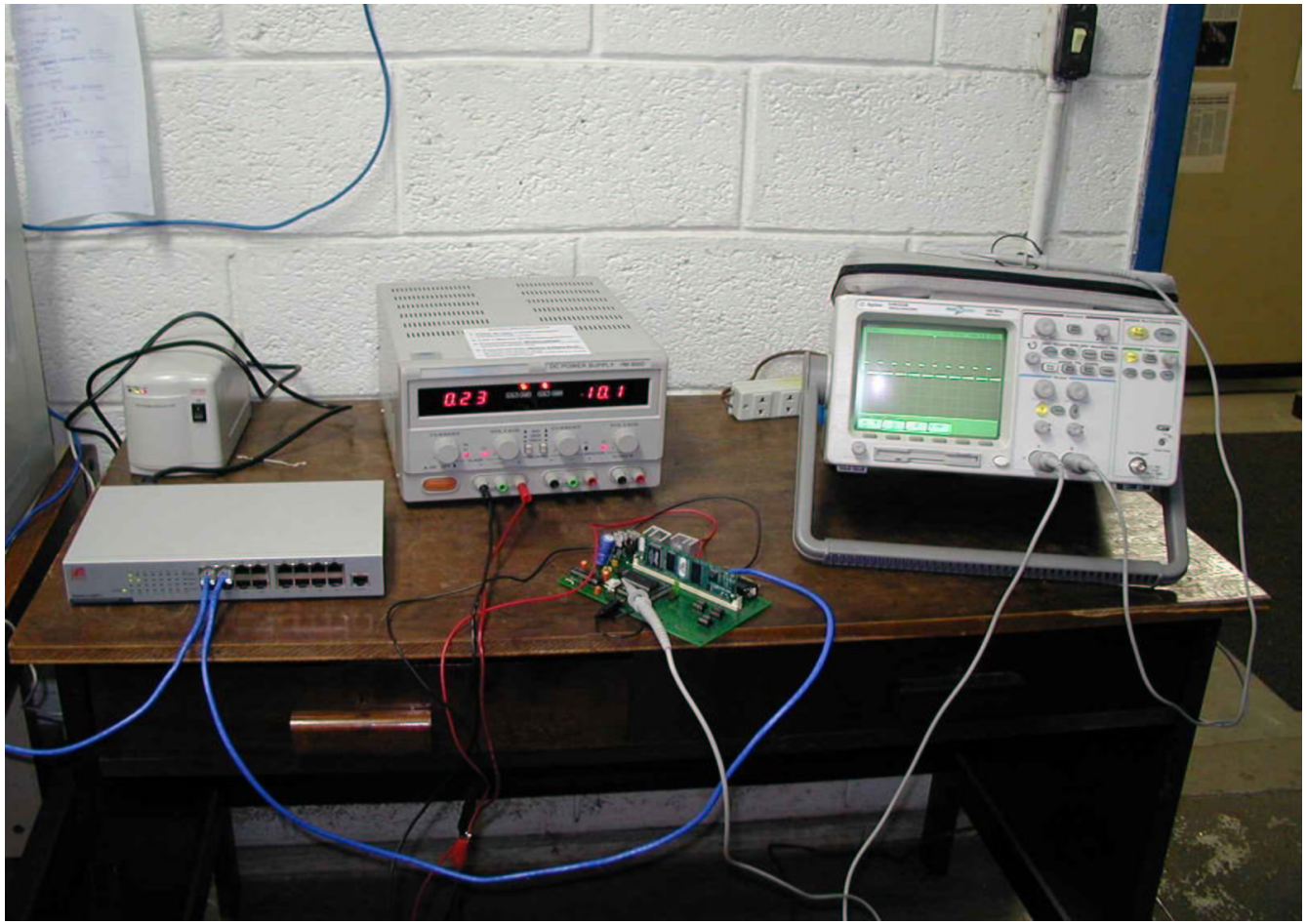
Software do Host

- C++ é melhor para controle avançado do que Java
  - A teoria de controle moderno e o controle de robôs em particular é baseado em álgebra matricial
  - C++ supporta sobrecarga operadores
  - Bibliotecas de manipulação de matrizes podem ser construidas de forma bastante conveniente
- O Host executa RTAI
  - Não há suporte para execução de Java em tempo real
  - Real−time java não é tão madura quanto POSIX

Esquema de Controle

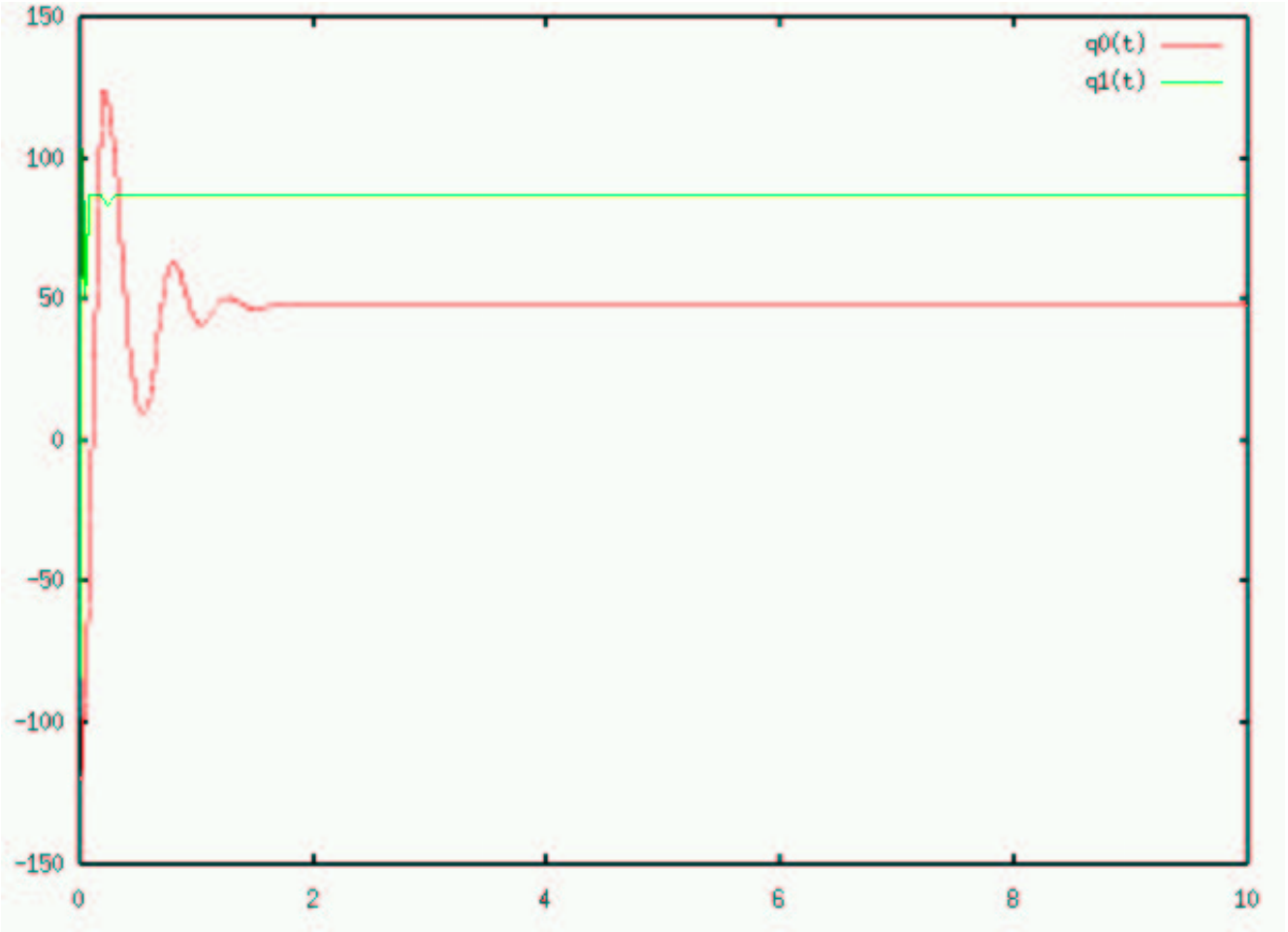


Experimentos

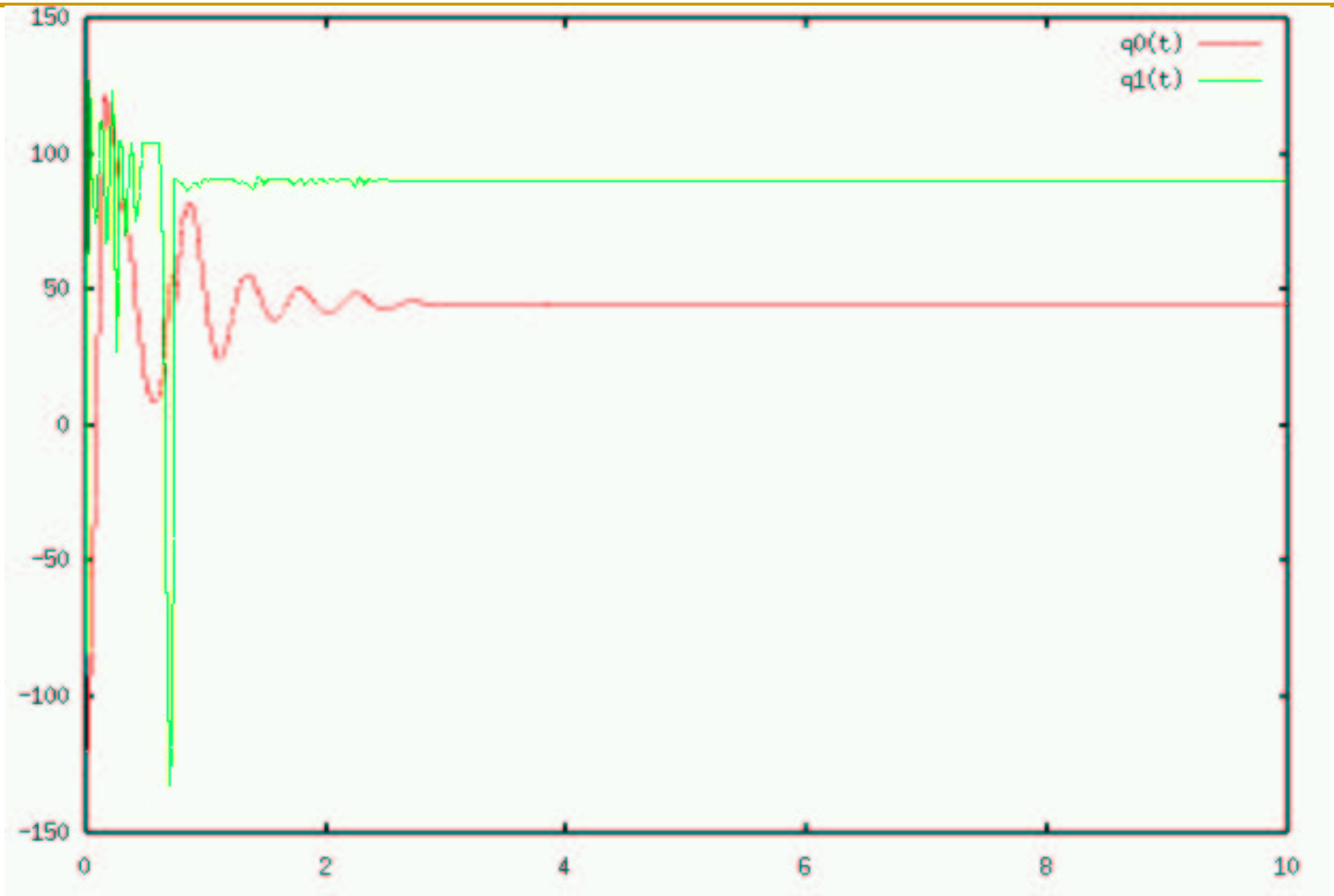


Experimentos

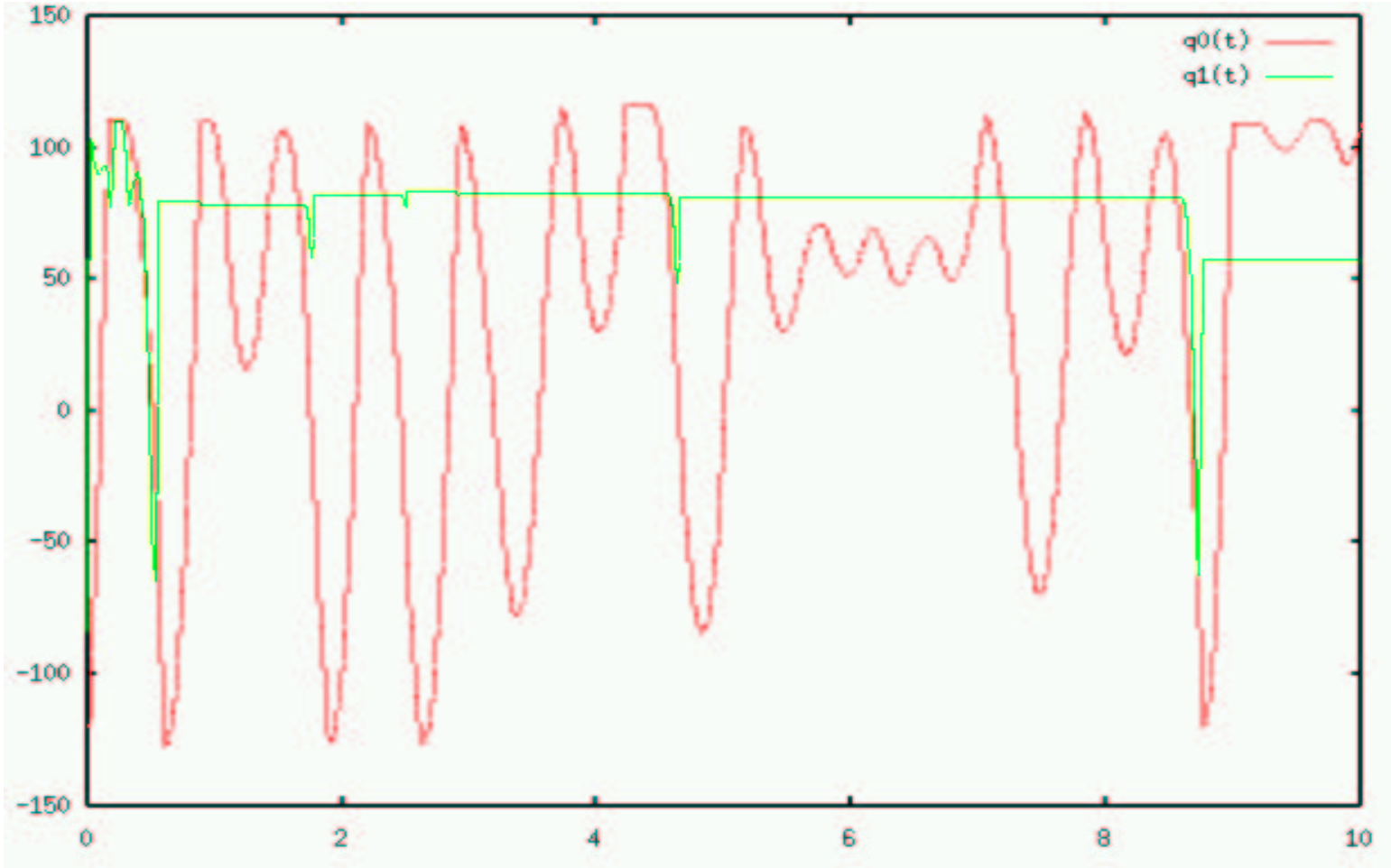
Resposta do Controlador



Resposta do Controlador com Baixo Tráfego



Resposta do Controlador com Alto Tráfego



Conclusão

- Retrofitting de robôs antigos pode ser realizado com excelentes resultados
  - Um robô de arquitetura abertas pode ser obtido por uma fração do custo de um robô novo
- A aquitetura proposta é bastante flexível
  - Possíveis extensões
    - Sensores de força/torque (em desenvolvimento)
    - Motores AC
- Programação de robôs em C++ é mais barata e mais conveniente do que em linguagens dedicadas

Vídeos

- AICs
- J13
- J45
- J1345

Agradecimentos

- Este trabalho foi financiado em parte por:
  - FAPERGS
    - Processo 01/05670
  - FAPEMIG
    - Processo TEC 2471/98
  - RECOPE/MANET