

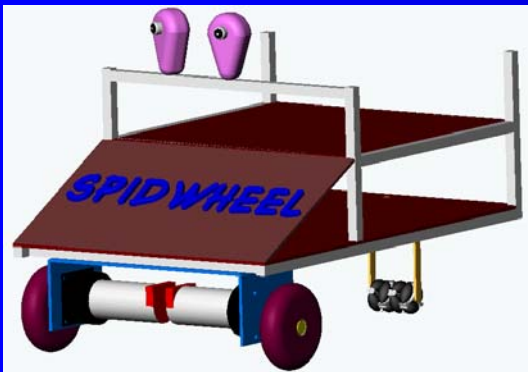
isep

instituto
superior de
engenharia do
porto



**robótica
2004**
festival nacional de robótica

Sistema de Controlo e Gestão de Missões para um Veículo Autónomo



Realização: Eduardo José Cruz Silva

Orientadores: Eng.º José Miguel Almeida

Eng.º Alfredo Martins

Eng.º Eduardo Silva

Data de realização: 13 de Setembro de 2004

LSA
I S E P

Objectivos

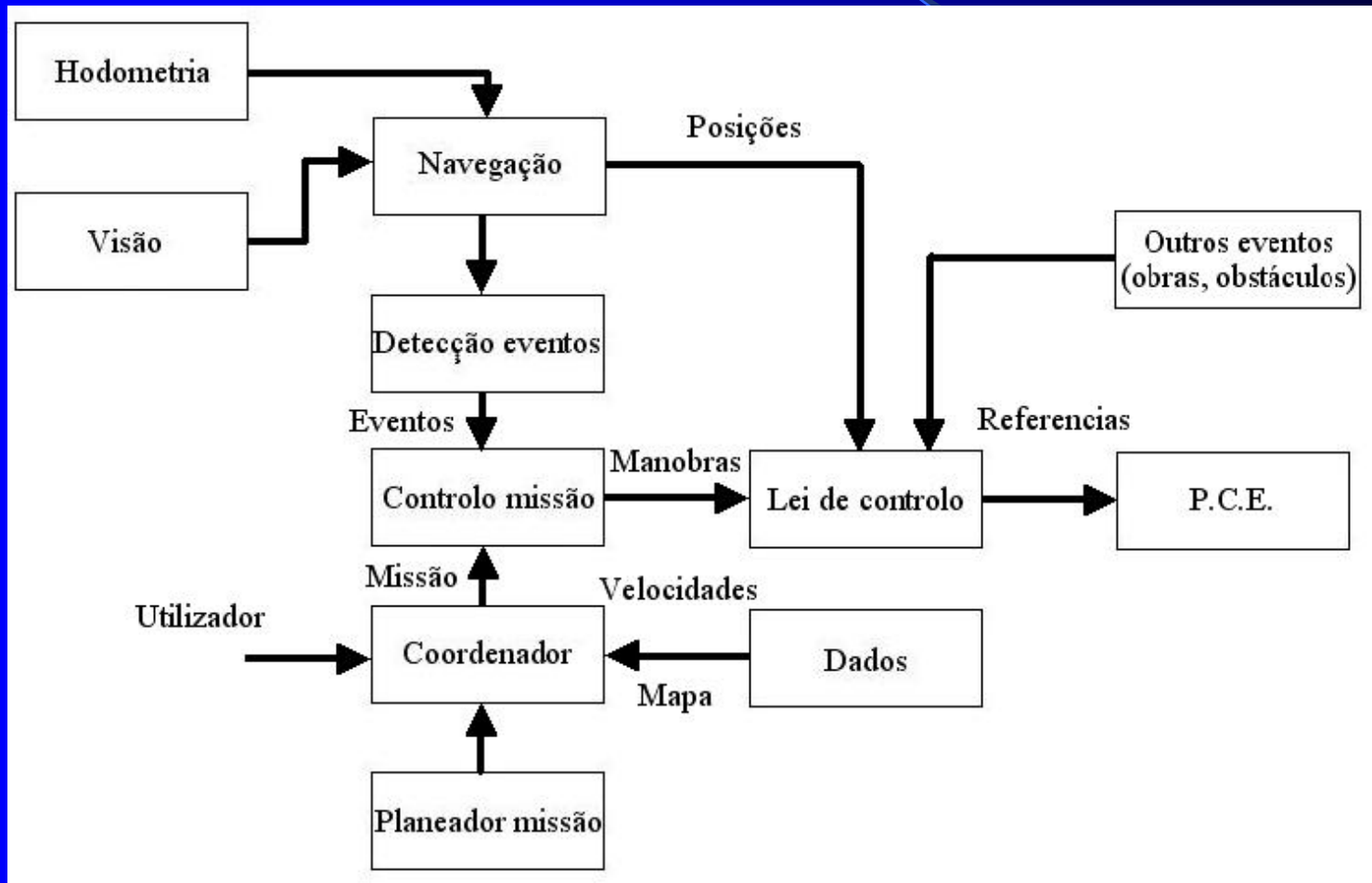
Desenvolvimento de um:

- Sistema de controlo
- Gestor de missões com planeamento de trajectórias

Cenários possíveis para este tipo de plataformas móveis

- Transporte de mercadorias seguindo pessoas ou linhas
- Tarefas de limpeza varrendo áreas
- Estradas como meio de transporte de pessoas sendo orientado por mapas pré definidos ou GPS
- Serviços de vigilância em edifícios monitorizando movimentos de pessoas
- Vigilância em florestas como prevenção de fogos
- Eventos desportivos como demonstração das capacidades do veículo

Arquitectura do sistema



Seguimento de trajectórias

● Rectas

- Orientação da recta:

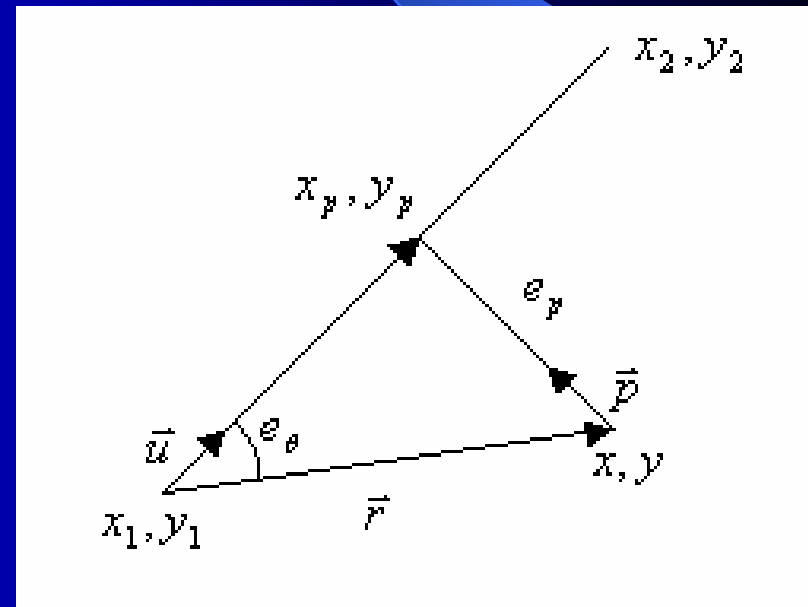
$$\theta_{ref} = \arctan(y_2 - y_1, x_2 - x_1)$$

- Erro da trajectória:

$$\begin{aligned} e_p &= \vec{p} \cdot \vec{r} \\ &= -\sin(\theta_{ref}) \cdot (x - x_1) + \cos(\theta_{ref}) \cdot (y - y_1) \end{aligned}$$

- Erro em orientação:

$$e_\theta = \theta_{ref} - \theta$$



Seguimento de trajectórias

● Curvas

- Os erros e_p e e_θ

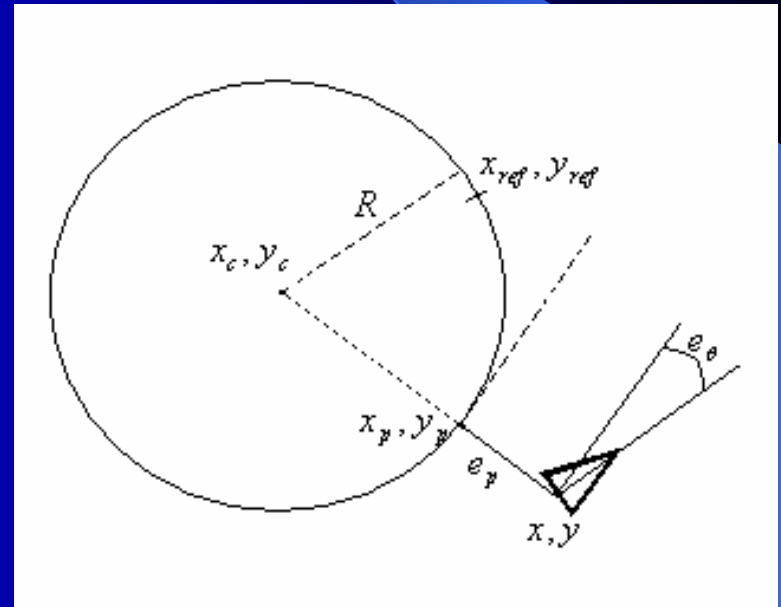
$d = 1$ se o sentido for directo

$d = -1$ se o sentido for inverso

- Os erros serão dados por:

$$e_\theta = \arctan(-(x - x_c), (y - y_c)) - \theta$$

$$e_p = d \cdot (\sqrt{(x - x_c)^2 + (y - y_c)^2} - R)$$



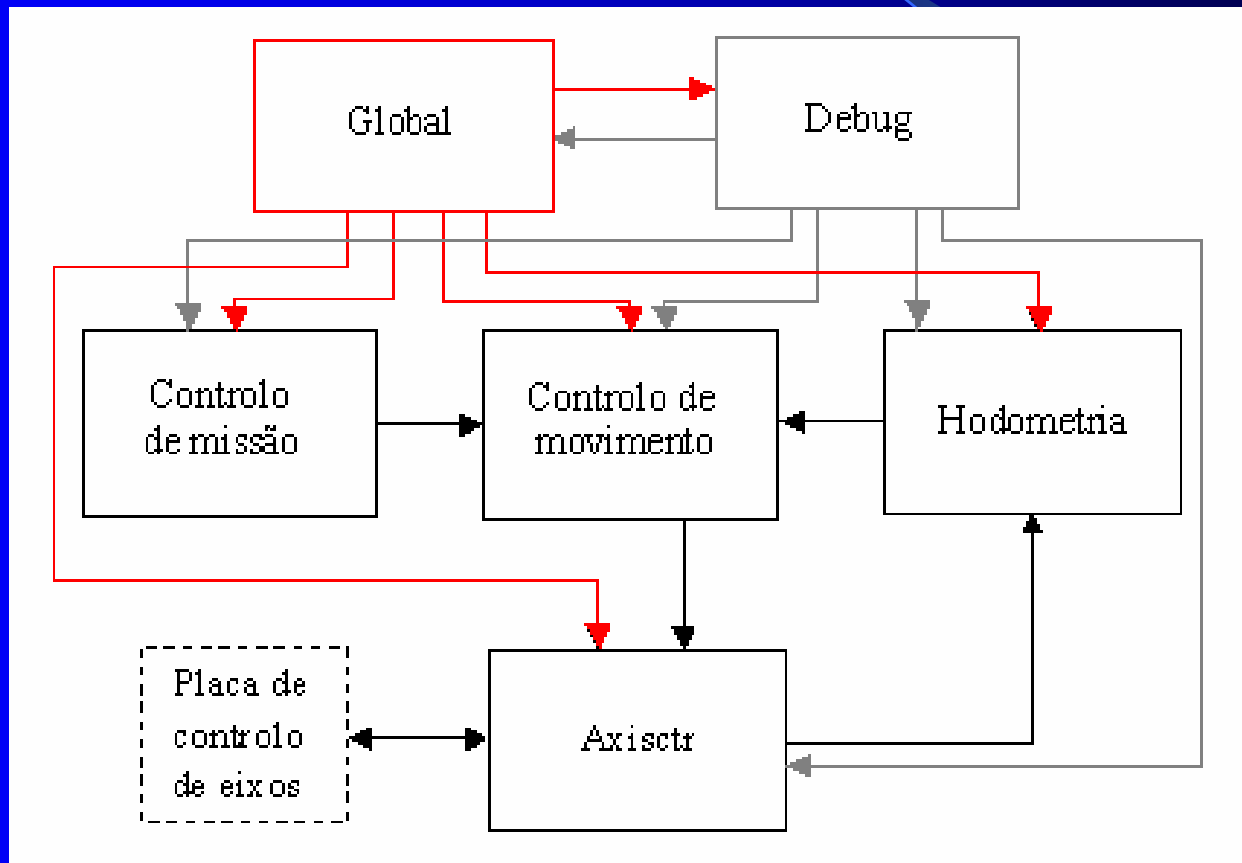
Controlo de velocidade

- Controlador PID

- Permite implementar três acções de controlo (proporcional, derivativa e integrativa)

- Não necessita à priori do conhecimento preciso da dinâmica do robot.

Arquitectura do software



Global

- Start.c
 - Função “*main*”
- Global.c
 - Função de conversão de referências para o PID

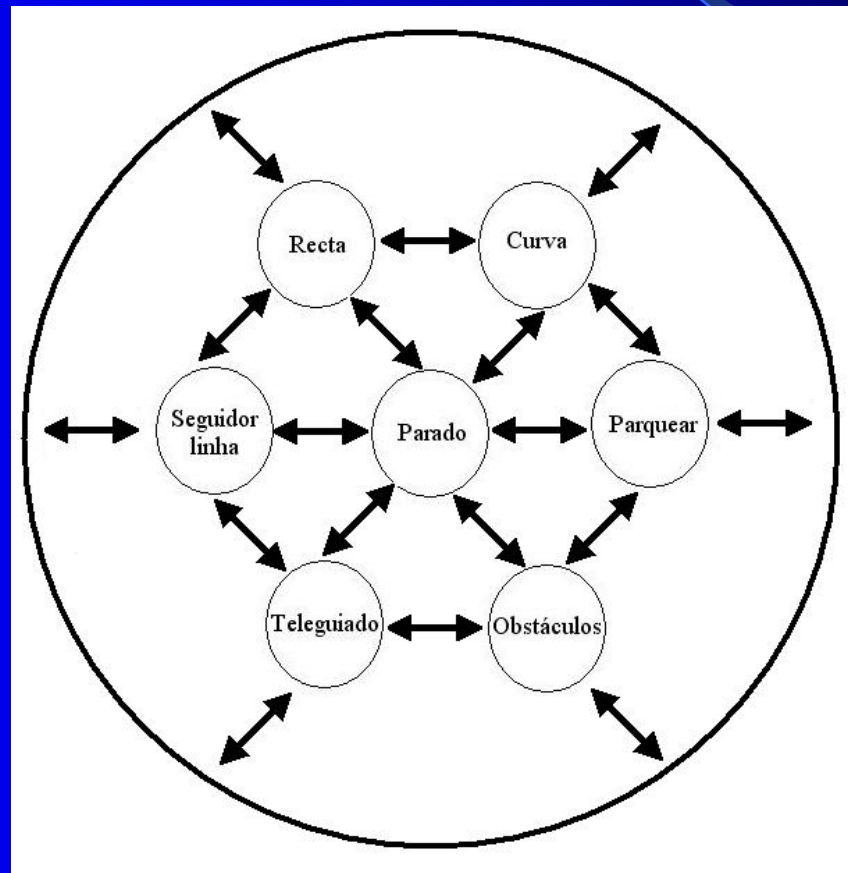
Hodometria

- Leitura dos contadores na placa de I/O
- Informação em posição (x, y) e angulo (θ)

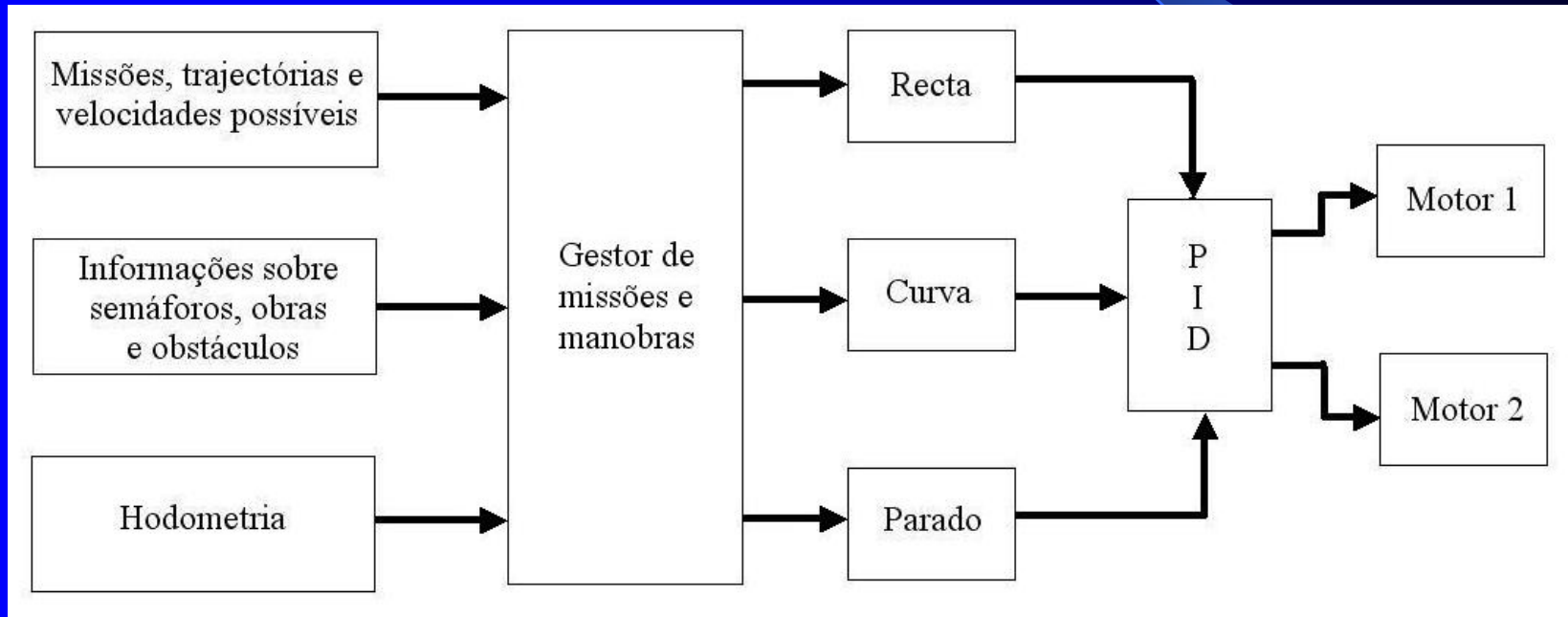
Debug

- Accionamento de funções
- Visualização de alguns “*prints*”

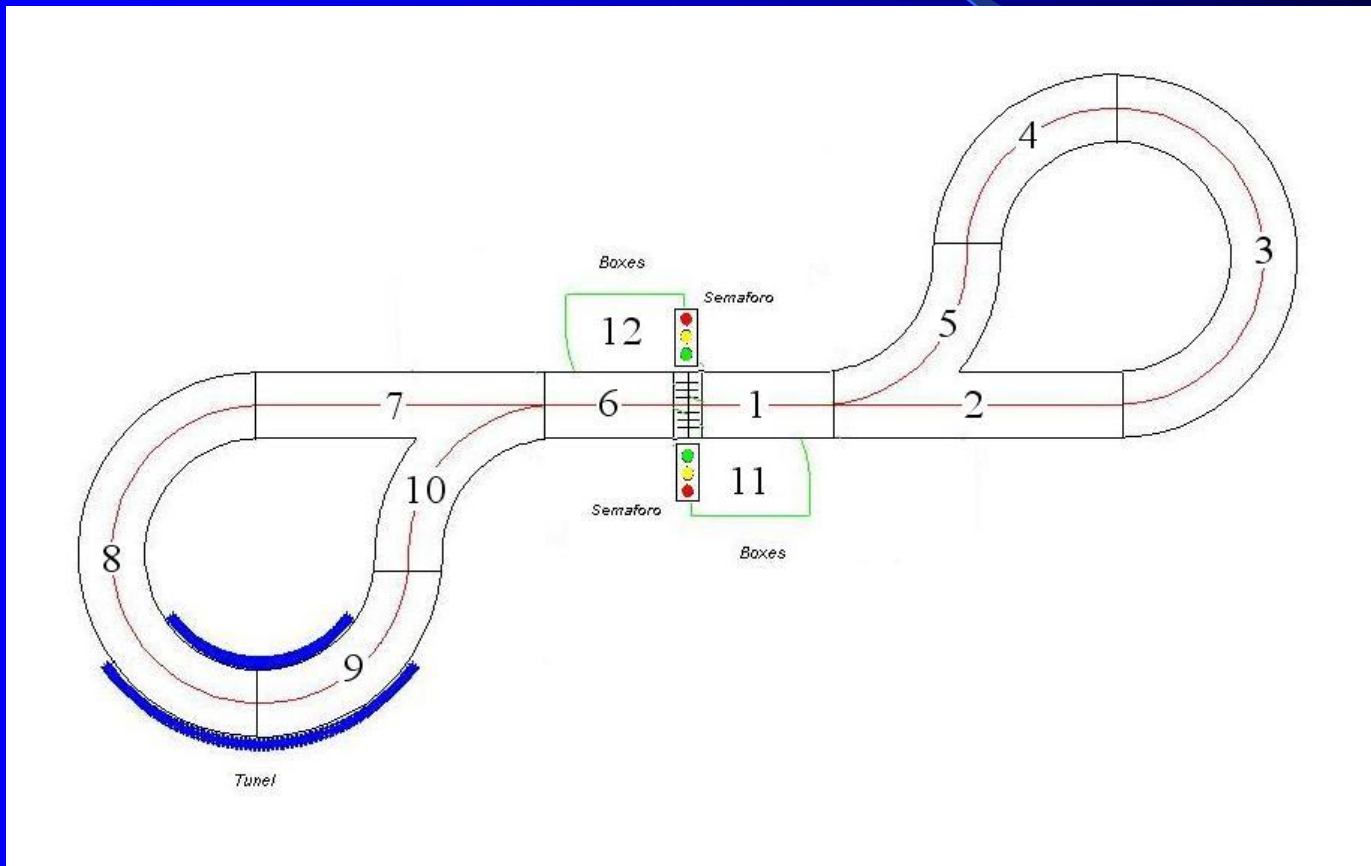
Controlo de movimento



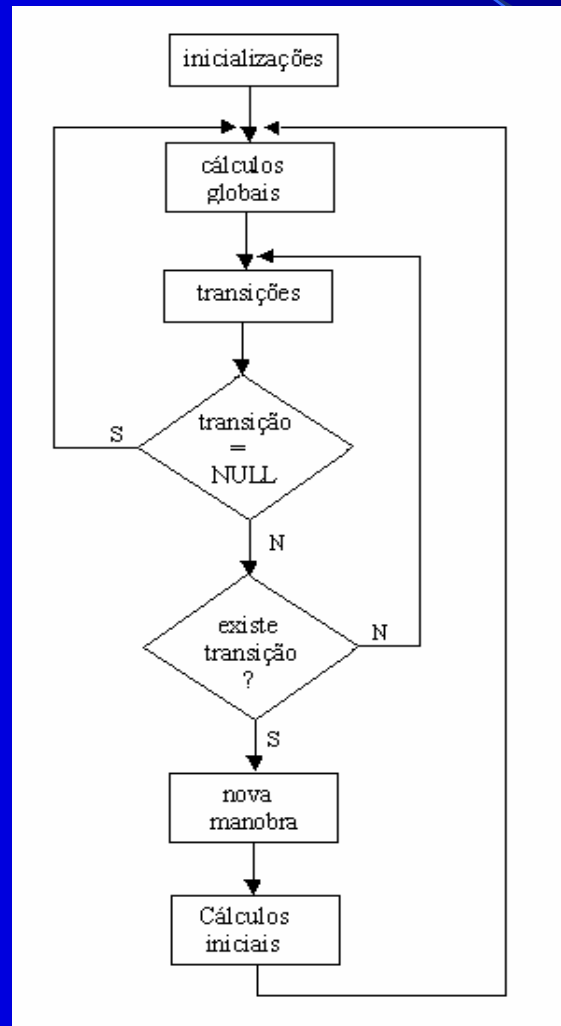
Controlo de missão



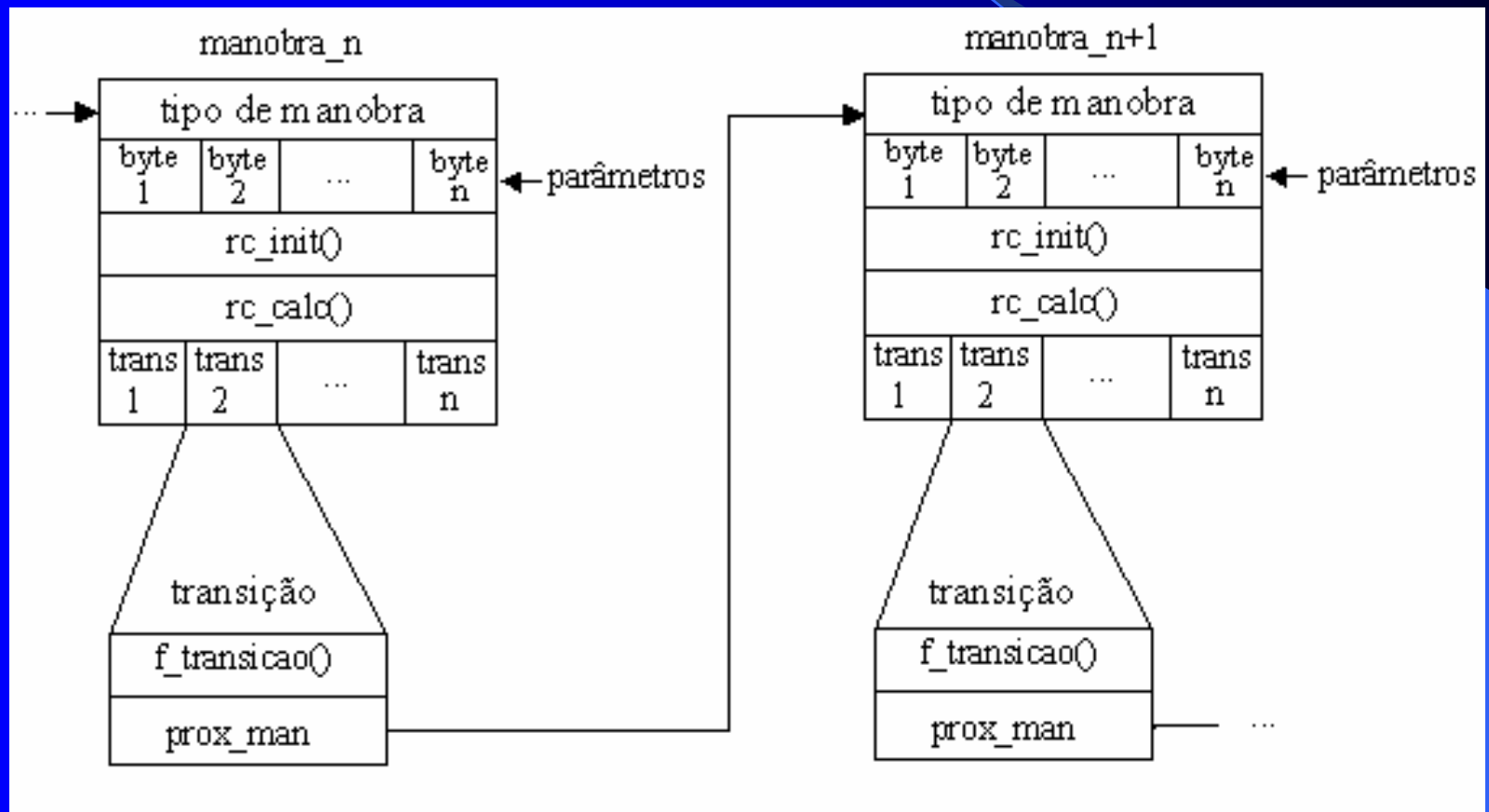
Mapa da pista



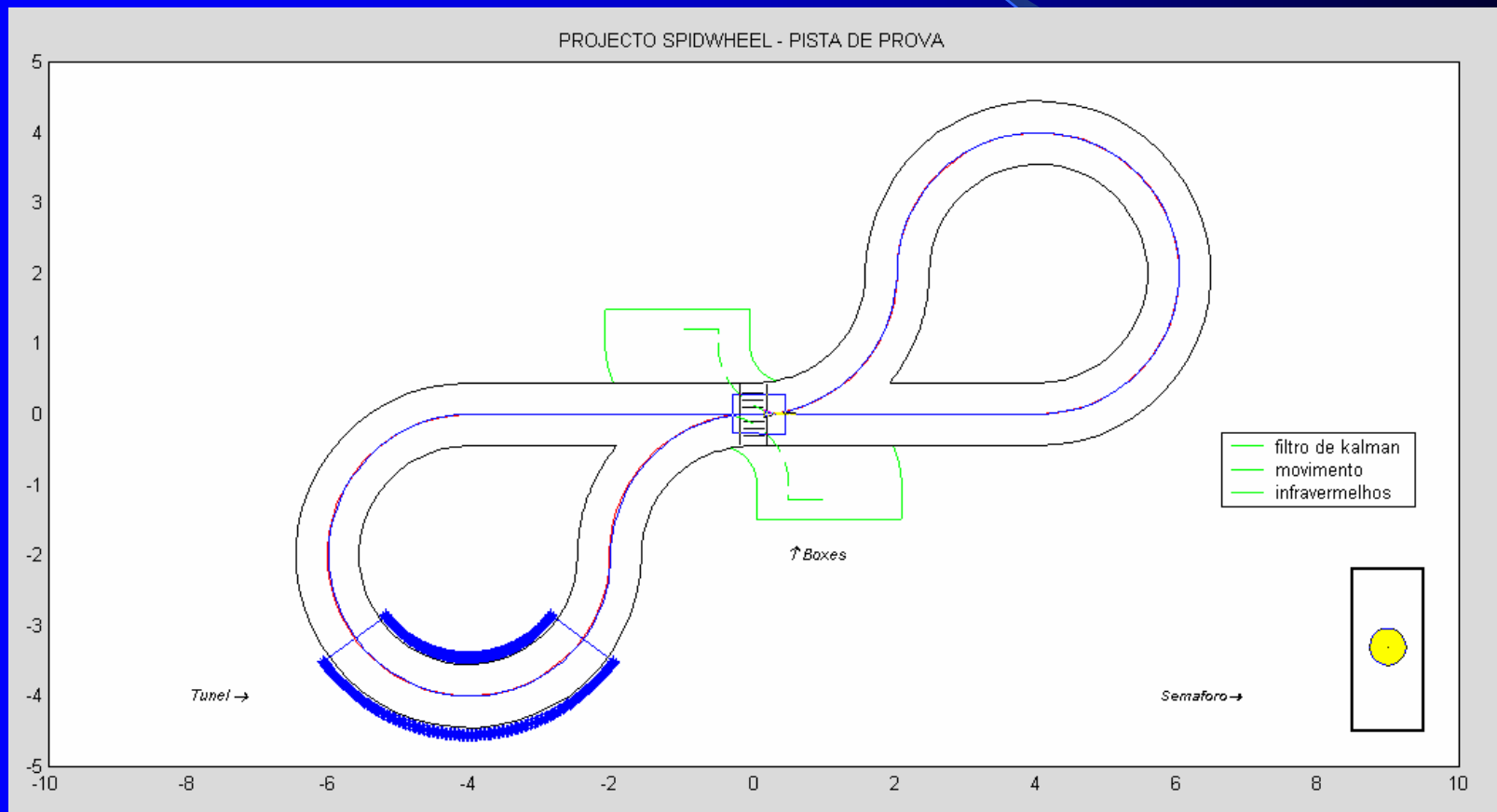
Executor de missões



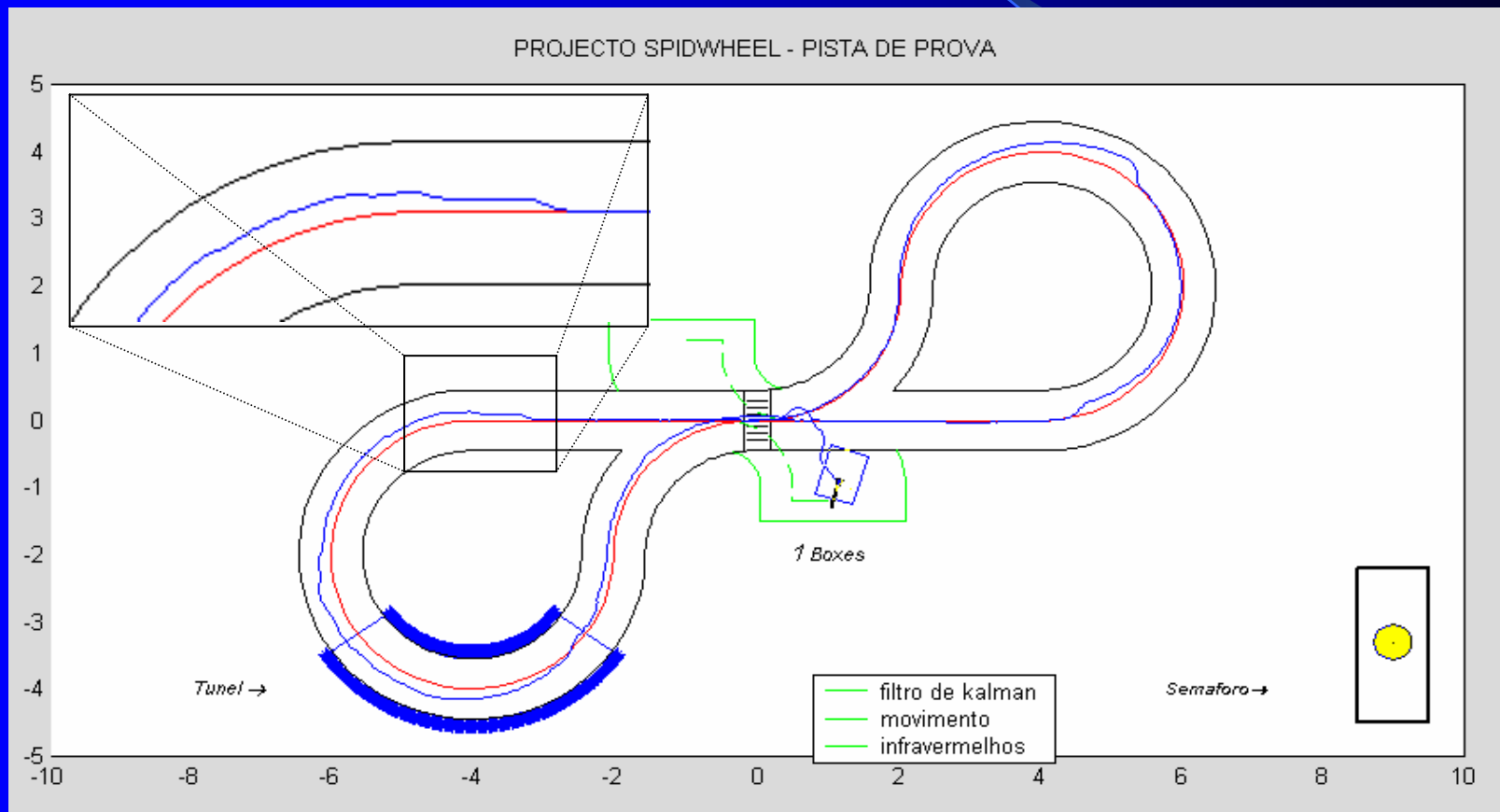
Transições de manobras



Simulação de trajetórias



Simulação de trajetórias



Robótica 2004

Spidwheel



Runner



Conclusões

- O robot consegue percorrer diferentes tipos de manobras
- É possível planear diversas missões pré definidas, que o gestor de tarefas decide a missão a percorrer e o tipo de manobras
- O robot é capaz de detectar e corrigir trajectórias
- O tipo de referência (seguimento de trajectórias) usado neste sistema de controlo, é bastante adequado ao problema proposto
- Não necessitarmos de fechar a aplicação, sempre que for necessário fazer alterações de parâmetros

Evolução futura

- “SLAM” (Simultaneous Localisation and Map Building).
- Reconhecimento de obstáculos, arranjando automaticamente caminhos alternativos
- Andar de marcha-a-trás
- Capacidade do robot rodar sobre si próprio
- Monitorização dos níveis de energia, e se necessário auto carregar-se.

Fim