



Universidade de São Paulo - São Carlos, SP

**Instituto de Ciências Matemáticas e de Computação**

USP – ICMC – SSC

### **SSC0714 - Robôs Móveis Autônomos**

**Professor responsável:** *Fernando Santos Osório*

**Semestre:** 2010/1

**Horário:** Quarta 9h20

**Local:** Campus II (Bloco Didático - Sala1)

**E-mail:** fosorio .at. icmc.usp.br

fosorio .at. gmail.com

Estagiário PAE: Maurício Acconcia Dias

**Web:** <http://www.icmc.usp.br/~fosorio/>

## ***TRABALHO FINAL DE ROBÓTICA - TF*** ***Trabalho em Grupo: Projeto Prático***

**[Data desta Definição: 05/05/2010 – Ver. 0.1]**

#### ***\* Especificação do Trabalho:***

- Data limite para definição do Projeto Prático: **19/05/2010 (Definição Grupo/Projeto)**
- Data da Apresentação: **09/06/2010 ou 23/06/2010 (Seminário apresentando o Projeto Prático)**
- Data da Entrega do Trabalho Prático: **23/06/2010 (Documentação/Implementação)**

#### - Definição dos Grupos:

- + Serão formados grupos com **2 alunos em cada grupo** (escolha livre entre alunos).
- + Em casos especiais (trabalhos cujo nível de complexidade seja muito grande), onde seja mesmo necessário um número maior de alunos, então serão aceitos grupos de no **máximo 03 alunos**. Estes casos precisam ser **DISCUTIDOS** e **APROVADOS** previamente pelo professor.

#### - Detalhes sobre o Material a ser entregue:

- + **Trabalho Escrito:** Documento descrevendo o trabalho (objetivos, métodos usados, problema tratado, implementação, instruções de uso e resultados).  
Deverá ser entregue ao professor em formato digital (.doc ou .pdf) no dia da apresentação.
- + **Apresentação:** prever uma rápida apresentação do trabalho de 10 a 15 minutos.  
Os alunos devem apresentar de modo direto e sucinto o trabalho implementado.
- + **Implementação:** o projeto prático deve envolver uma implementação, seja ela um programa desenvolvido completamente pelo grupo, ou também poderá ser, um módulo de um programa a ser usado junto a uma outra ferramenta (e.g. um controlador para uso com o player-stage). O tema da implementação tem que ser definido previamente em acordo com o professor – ver a seguir as sugestões e detalhes sobre os projetos.

#### - Detalhes sobre a Apresentação:

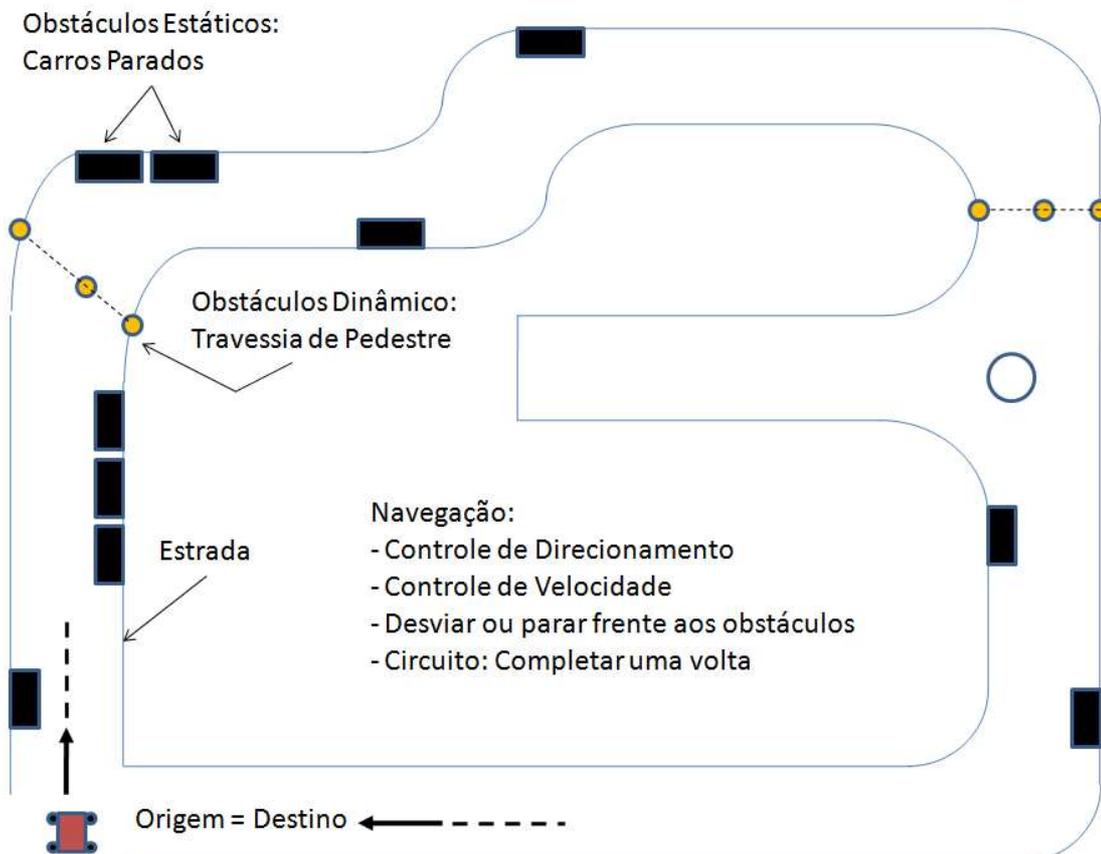
- + Projeto a ser apresentado em aula para o professor e os colegas.
- + Preferencialmente com uma demonstração prática do projeto desenvolvido.
- + Duração máxima do seminário: 10 a 15 MINUTOS

**\* Detalhamento sobre o TRABALHO PRÁTICO:**

- ⇒ O trabalho é prático, portanto requer que seja implementado um programa ou módulo de um programa maior.
- ⇒ O trabalho pode ser implementado em diferentes plataformas e linguagens (C, C++, Java, Python, MatLab para Linux ou Windows).
- ⇒ O trabalho pode ser implementado integralmente pelos alunos (sem uso de ferramentas ou bibliotecas externas), ou se quiser, pode ser implementado com o uso de uma plataforma de software (por exemplo: Player-Stage-C/C++, RoboCode-Java, OpenCV-C/C++, etc).
- ⇒ Se o aluno fizer uso de uma plataforma de software ou mesmo de trechos de código “recuperados” da Internet, deve OBRIGATORIAMENTE fazer referência de modo explícito na documentação ao uso das plataformas e programas externos (com endereço de onde foi baixado o código).

**\* TEMAS SUGERIDOS E EXEMPLOS DE PROPOSTAS DOS PROJETOS PRÁTICOS:**

1. Simulação de um veículo autônomo para navegação em estradas usando o player-stage. Assume-se que o mapa será da forma de uma estrada, com obstáculos estáticos (carros estacionados) e/ou dinâmicos (pessoas), onde o veículo deve seguir o trajeto da origem até o seu destino. Exemplo:



2. Implementar um sistema de navegação baseado em mapas métricos. Considerar que dado o mapa do ambiente e a posição inicial e de destino do robô junto a este mapa:
  - i) O robô deverá ser capaz de estimar aproximadamente sua posição no mapa;
  - ii) O robô deverá ser capaz de desviar de obstáculos que estão presentes no ambiente mas que não foram descritos no mapa (cadeiras, mesas);
  - iii) O robô deverá ser capaz de realizar um deslocamento de sua posição inicial até a posição final (aproximada), desviando dos obstáculos descritos no mapa ou não.
3. Implementar um sistema de navegação baseada em mapa topológico para ser usado junto ao Player-Stage. O usuário deve fornecer apenas um mapa topológico (não métrico, apenas de conectividade), indicando o ponto de origem e de destino. O robô móvel deve ser capaz de traçar sua rota, se deslocando da posição origem até a posição de destino, seguindo as informações do mapa topológico. Neste ambiente todos os obstáculos são estáticos e o mapa topológico reflete a estrutura de um ambiente composto por corredores e salas, sem obstáculos que possam dificultar sua passagem. O mapa topológico deve poder ser facilmente descrito através de uma representação de alto nível (arquivo texto simples).
4. Implementar um sistema de visão do tipo “follow-me”, baseado no uso do OpenCV ou do Player-Stage (Gazebo). Um robô deve seguir (acompanhar) uma marca visual/perceptiva que irá se deslocando na sua frente. A marca visual/perceptiva pode ser definida pelo grupo, podendo ser uma ou mais marcas que irão identificar o objeto móvel que está sendo seguido. O sistema de acompanhamento do tipo follow-me deve ser robusto para seguir o objeto mesmo se ele perder momentaneamente o contato visual com o objeto alvo (no caso de uma curva fechada).
5. Implementar um algoritmo de planejamento de trajetórias baseado no A\* para múltiplos robôs. Os robôs devem “cercar” o alvo usando a estratégia de não repetir exatamente a mesma trajetória.
6. Implementar um sistema inteligente (agente móvel inteligente) capaz de controlar um carro virtual (TORCS). Esta proposta é inspirada nos moldes da competição do Computational Intelligence in Games Conference - Car Racing (CIG 2009, CEC 2009). Esta opção de trabalho serve também para outras variantes:
  - Controlar um agente móvel autônomo do tipo PacMan
  - Controlar um agente móvel autônomo do tipo um avião simulado
7. Implementar: (detalhar melhor a técnica a ser adotada e as ferramentas usadas)
  - Algoritmo para mapeamento de ambientes;
  - Algoritmo para auto-localização baseado em sistema de partículas;
  - Algoritmo para visão computacional associada a uma aplicação de robótica;
  - Algoritmo para fusão sensorial (GPS, Bússola, Laser, IMU);
  - Algoritmo para sistema multi-robôs de ação coordenada;
  - Algoritmo de aprendizado de máquina para robôs (aprender a controlar o robô).