Aula 5 – Introdução à Robótica Móvel

Navegação

Prof. Assoc. Marcelo Becker

SEM - EESC - USP

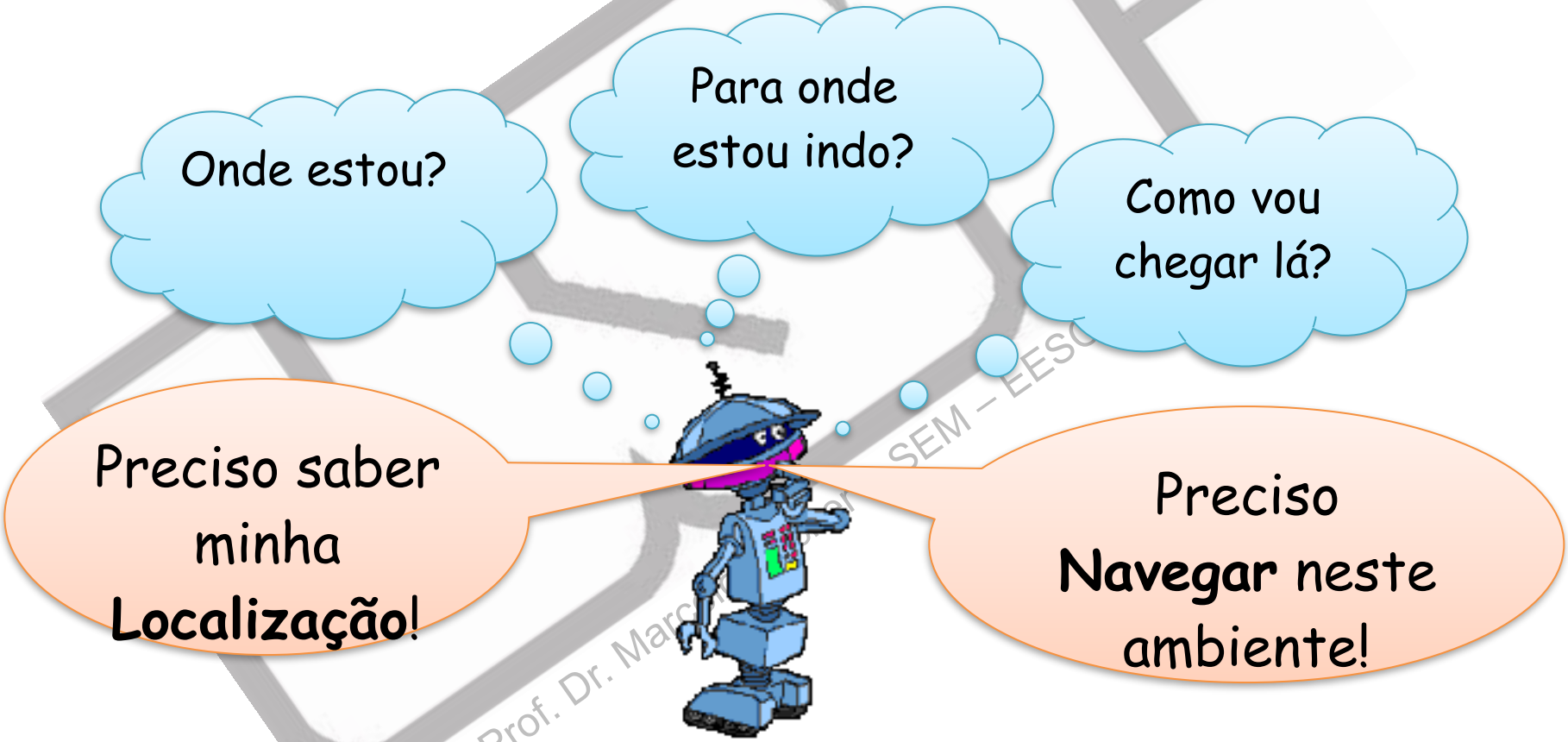
Laboratório de Robótica Móvel

LabRoM

Prof. Dr. Marcelo Becker _ SEM – EESC – USP

Introdução

- Problemas Básicos em Robótica Móvel:

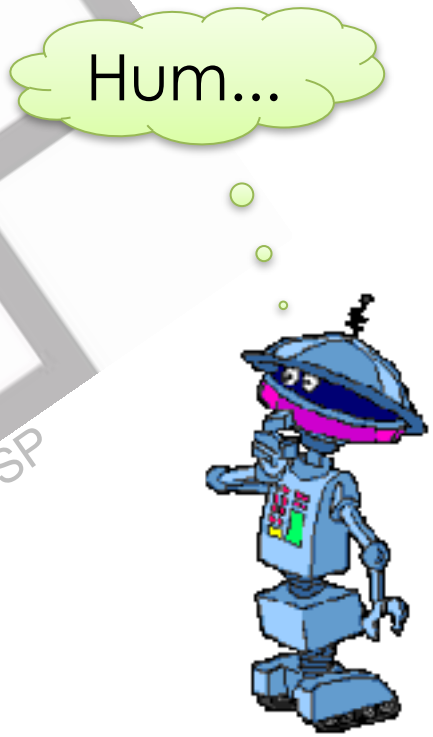


(Leonard and Durrant-Whyte, 1992)

© M. Becker 2017

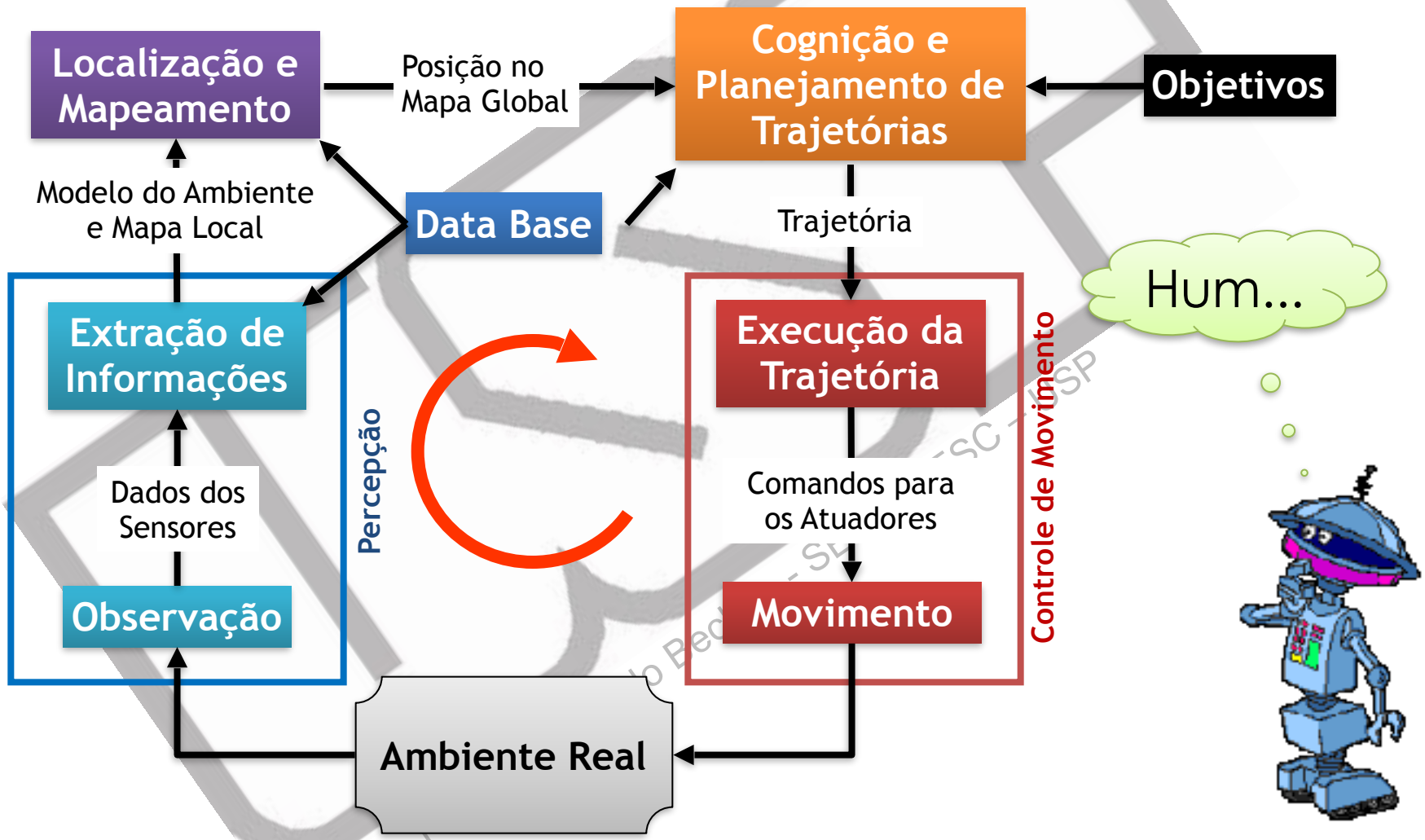
Introdução

- 3 Perguntas fundamentais...
  - Onde estou?
  - Para onde vou?
  - Como faço isso?
- Para respondê-las, o robô deve:
  - ✓(Ob)Ter um modelo do ambiente
  - ✓Observar e analisar o ambiente
  - ✓Encontrar sua posição no ambiente
  - ✓Planejar e executar uma trajetória



Prof. Dr. Marcelo Becker – SEM – EESC – USP

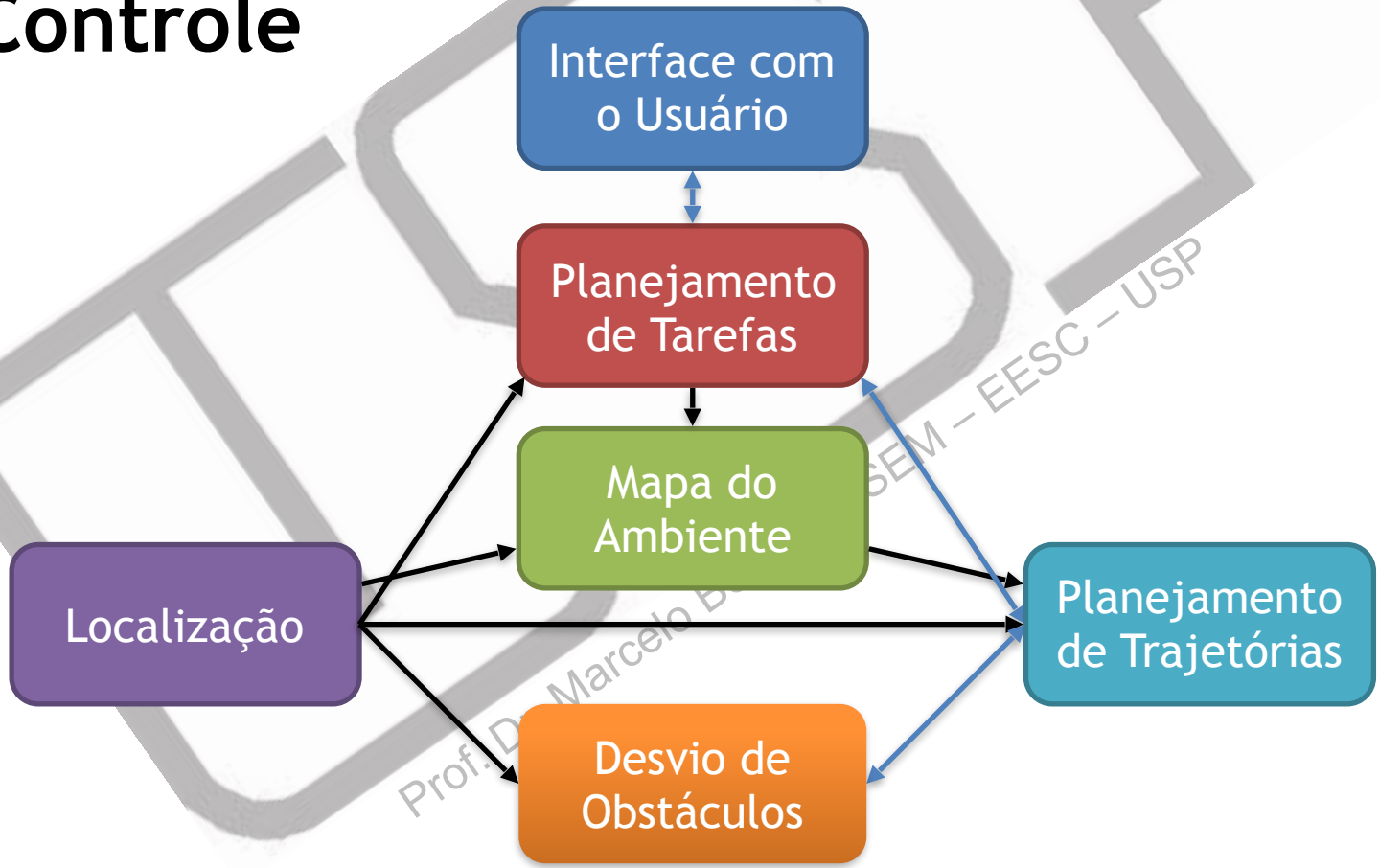
Introdução



© M. Becker 2017

Introdução

- **Arquitetura Geral do Sistema de Controle**



Prof. Dr. Marcelo Becker – SEM – EESC – USP

© M. Becker 2017

Introdução

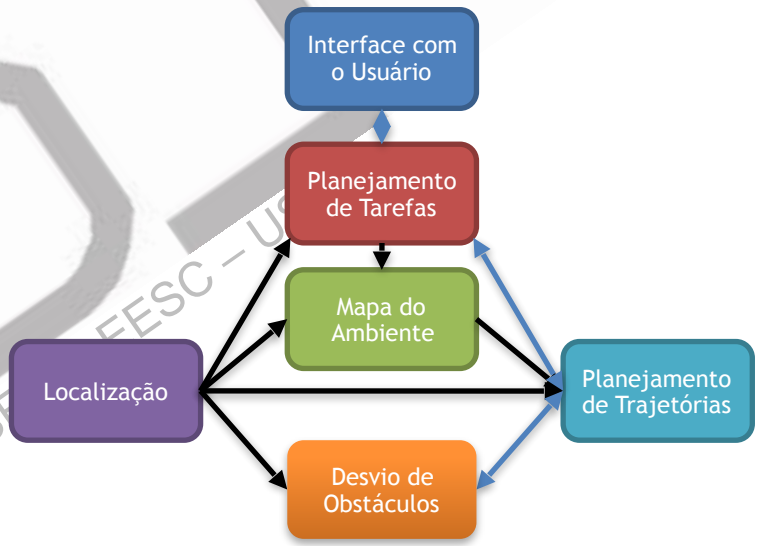
- ***Definição para Planejamento de Trajetórias***: encontrar uma trajetória no ambiente a partir da posição inicial até a posição desejada **evitando qualquer colisão com obstáculos**.

- Tem-se assim, que pode-se atuar no Planejamento de Trajetórias de modo:
  - ***Global***
  - ***Local***

Prof. Dr. Marcelo Becker - SEM - EESC – USP

Introdução

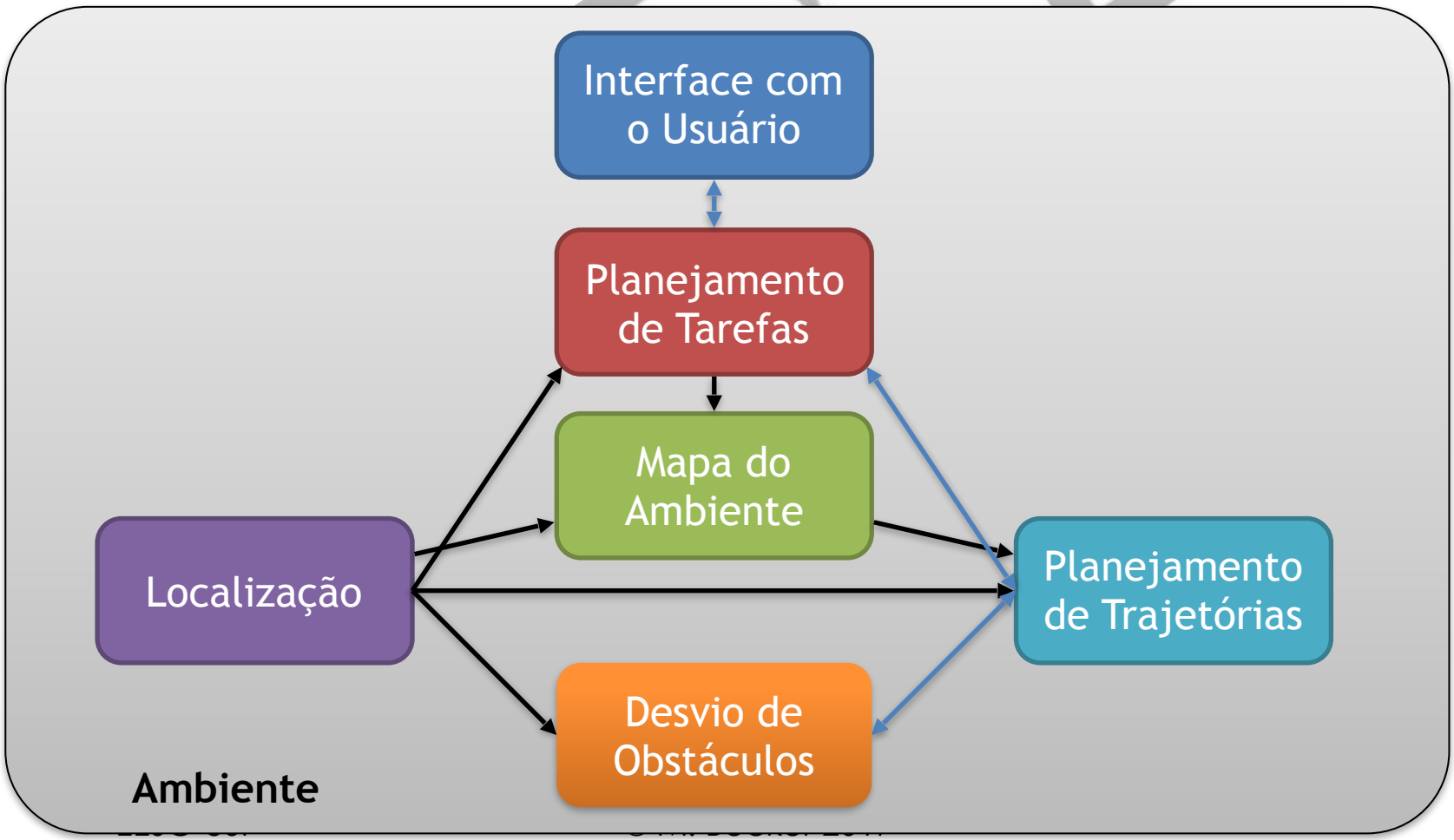
- ***Problema:*** Conflito de Comportamento
  - Ir à Posição Desejada vs. Desviar de Obstáculos
- Técnicas Separadas
  - Separam os comportamentos
  - Atuam em paralelo
  - Mais simples e rápidas!
- Técnicas Integradas
  - Unificam os comportamentos
  - Atuam em conjunto em busca de uma trajetória ótima
  - Mais complexas...



Prof. Dr. Marcelo Becker - SEM - EESC – USP

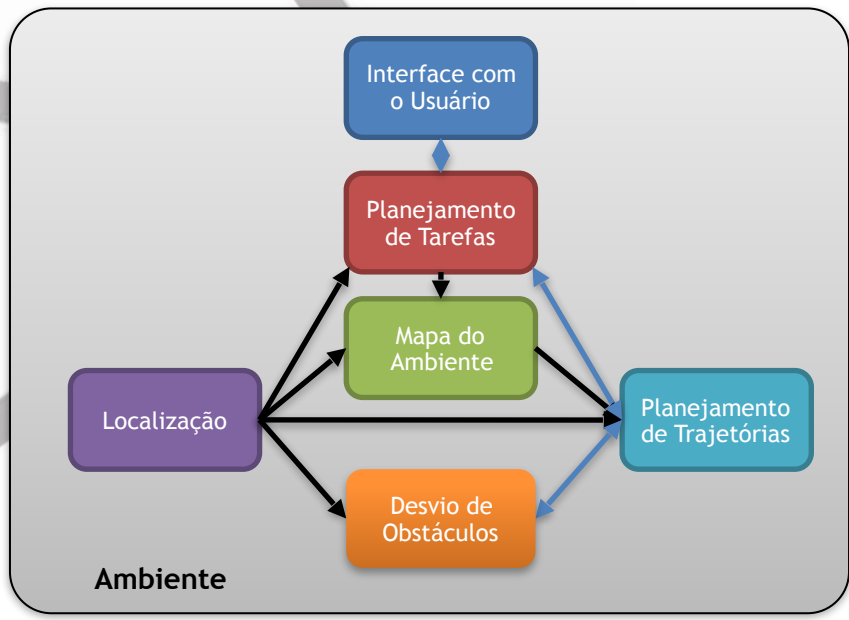
Introdução

- Observando novamente a Arquitetura de Controle...



Introdução

- ***Problema:*** Tipo de Ambiente
  - Estático vs. Dinâmico
  - Presença de Humanos
  - ...
- Respostas em Tempo Real
- Tipos de sensores
- Quantidade de Sensores



Prof. Dr. Marcelo Becker

© M. Becker 2017

Introdução

- **Técnicas que serão apresentadas**
  - Visibility Graph
  - Diagrama de Voronoi
  - Cell Decomposition
  - NF1
  - Breadth-First Search e Depth-First Search
  - Algoritmo A*
  - Algoritmo de Dijkstra
  - Campos Potencias e variações

Planejamento Global da Trajetória

Prof. Dr. Marcelo Becker - SEM - EESC – USP

© M. Becker 2017

Introdução

- **Técnicas que serão apresentadas**
  - Bug 1 e Bug 2
  - VFH, VFH$^+$ e VFH*
  - Bubble Band
  - CVM e LCV
  - Dynamic Window Approach
  - Occupancy Grid
  - Velocity Obstacle e variação
  - Probabilistic Occupancy Grid

Planejamento Local da Trajetória

Prof. Dr. Marcelo Becker - SEM – EESC – USP

© M. Becker 2017

Planejamento Global da Trajetória

Prof. Dr. Marcelo Becker - SEM – EESC – USP

© M. Becker 2017

Visibility Graph

(Latombe, 1991)

- ***Idéia Básica***:
  - O robô móvel é representado como um ponto.
  - As dimensões dos obstáculos são acrescidas do raio do robô.
  - O ambiente, bem como os obstáculos (**poligonais**) existentes nesse ambiente, podem ser bem representados em um espaço R^2 (2D) – Mapa Topológico.
- ***Deseja-se***:
  - Construir uma trajetória através de segmentos de reta livres de obstáculos que conectam a posição inicial à posição final.

Prof. Dr. Marcelo Becker – SEM – EESC – USP

Visibility Graph

Método:

- <u>Nós</u>: $p_{inicial}$, p_{final} e vértices dos obstáculos
- <u>Arcos</u>: 2 nós conectados por um segmento de reta

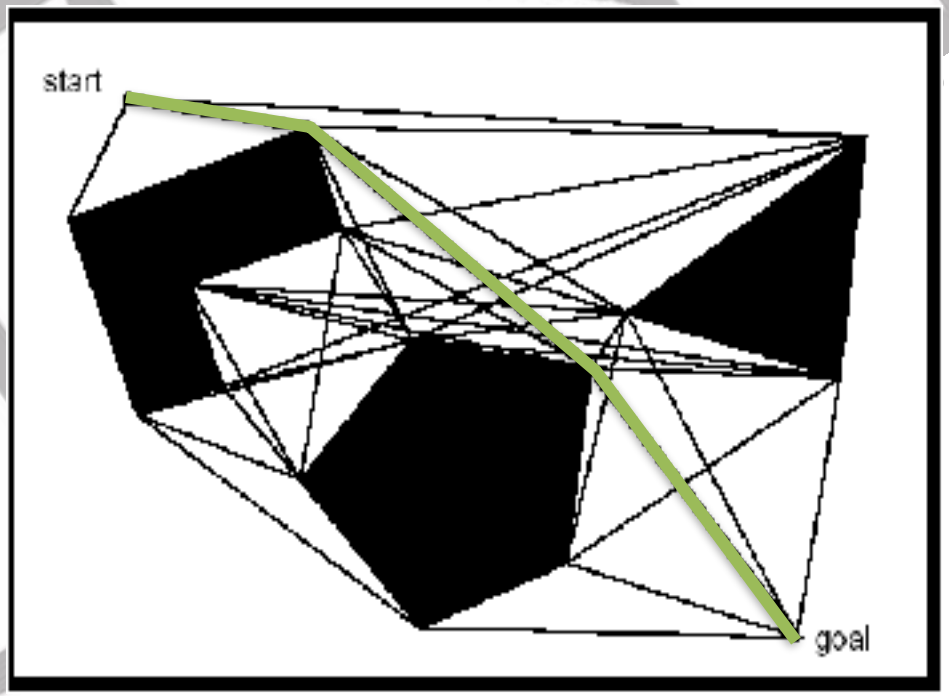
Algoritmo:

1. Construir o *Visibility Graph*
2. Procurar uma trajetória de $p_{inicial}$ a p_{final}
3. **SE** a trajetória for encontrada, retorná-la **SENÃO**, retornar FALHA na obtenção de trajetória

Prof. Dr. Marcelo Becker – SEM – EESC – USP

Visibility Graph

- Adequado para obstáculos semelhantes a polígonos
- A trajetória obtida é dita *Shortest Path*
- As dimensões dos obstáculos são aumentadas para evitar colisões



© M. Becker 2017

Visibility Graph

- ***Prós***
  - A trajetória encontrada é dita ótima pois ela é a trajetória de menor comprimento (*Shortest Path*).
  - Implementação simples parra obstáculos poligonais.

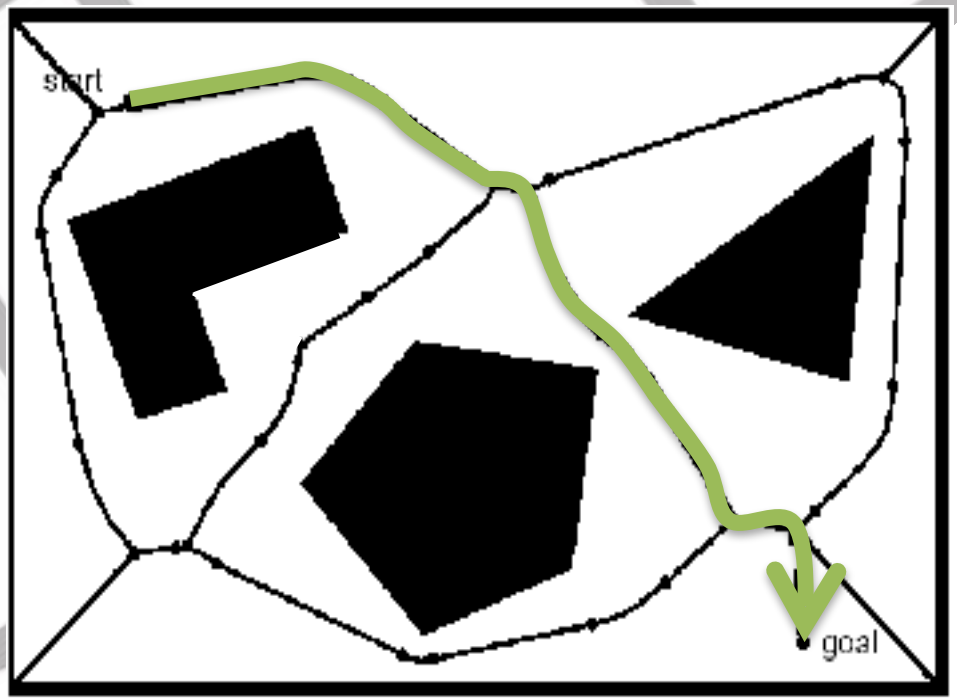
- ***Contras***
  - A solução dada pelo *Visibility Graph* tende a fazer com que o robô movimente-se o mais próximo possível aos obstáculos ☞ uma solução comum é utilizar um fator de segurança no aumento das dimensões dos obstáculos (o aumento é maior que o raio do robô para aumentar a segurança).
  - O número de nós aumenta proporcionalmente ao número de polígonos e ao número de vértices dos polígonos.
  - Pode ser ineficiente em ambientes repletos de obstáculos...

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Diagrama de Voronoi

(Latombe, 1991)

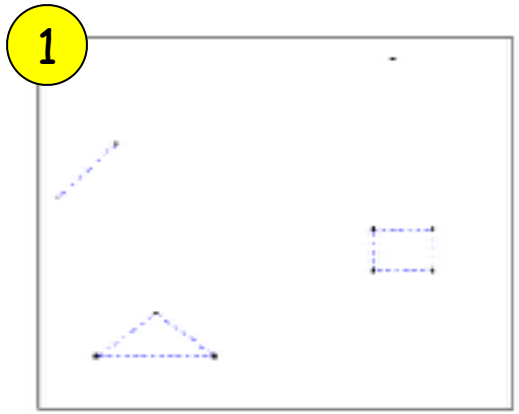
- Ao contrário do *Visibility Graph* o Diagrama de Voronoi tende a maximizar a distância entre o robô e os obstáculos
- Fácil de implementar e executar

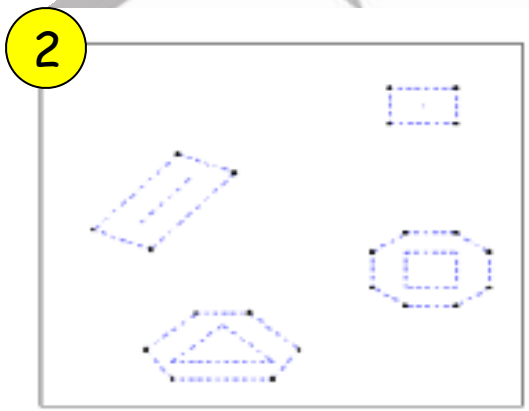


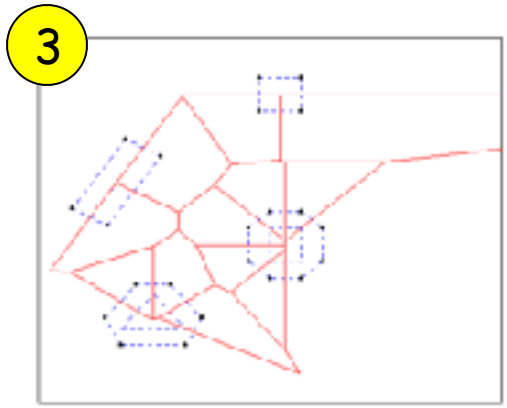
© M. Becker 2017

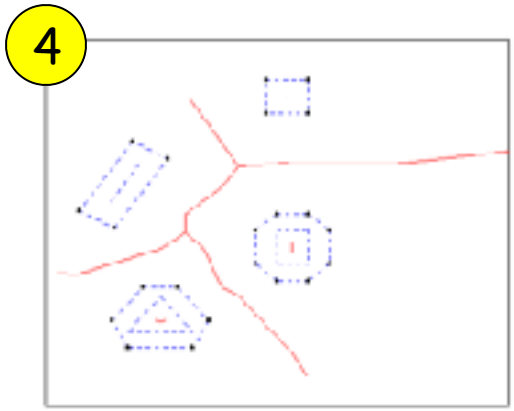
Diagrama de Voronoi

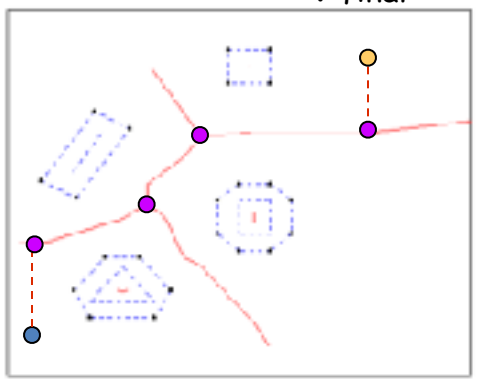
- Exemplo de Aplicação











© M. Becker 2017

Diagrama de Voronoi

- ***Prós***
  - Usando sensores de distância como laser ou sonar, um robô pode navegar utilizando o Diagrama de Voronoi empregando regras de controle simples.

- ***Contras***
  - Como o Diagrama de Voronoi tenta manter o robô o máximo possível longe de obstáculos, o emprego de sensores de distância de pequeno alcance não é recomendável.

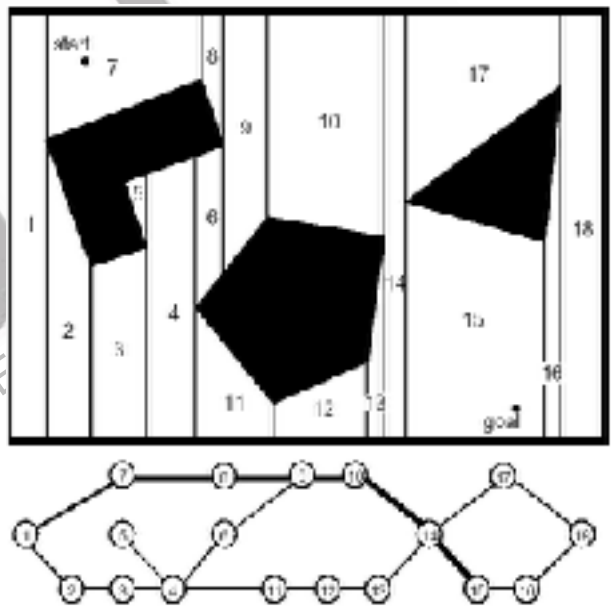
- ***Peculiaridades***
  - Quando os obstáculos são poligonais, o Diagrama de Voronoi é formado exclusivamente por segmentos de reta e parabólicos.

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Cell Decomposition

(Russell and Norvig, 1995)

- ***Idéia Básica***:
  - Dividir o ambiente em regiões simples conectadas chamadas de células (*cells*).
  - Construir um grafo de conectividade das células.
  - Encontrar as células em que as posições inicial e final estão e procurar no grafo um caminho entre elas.



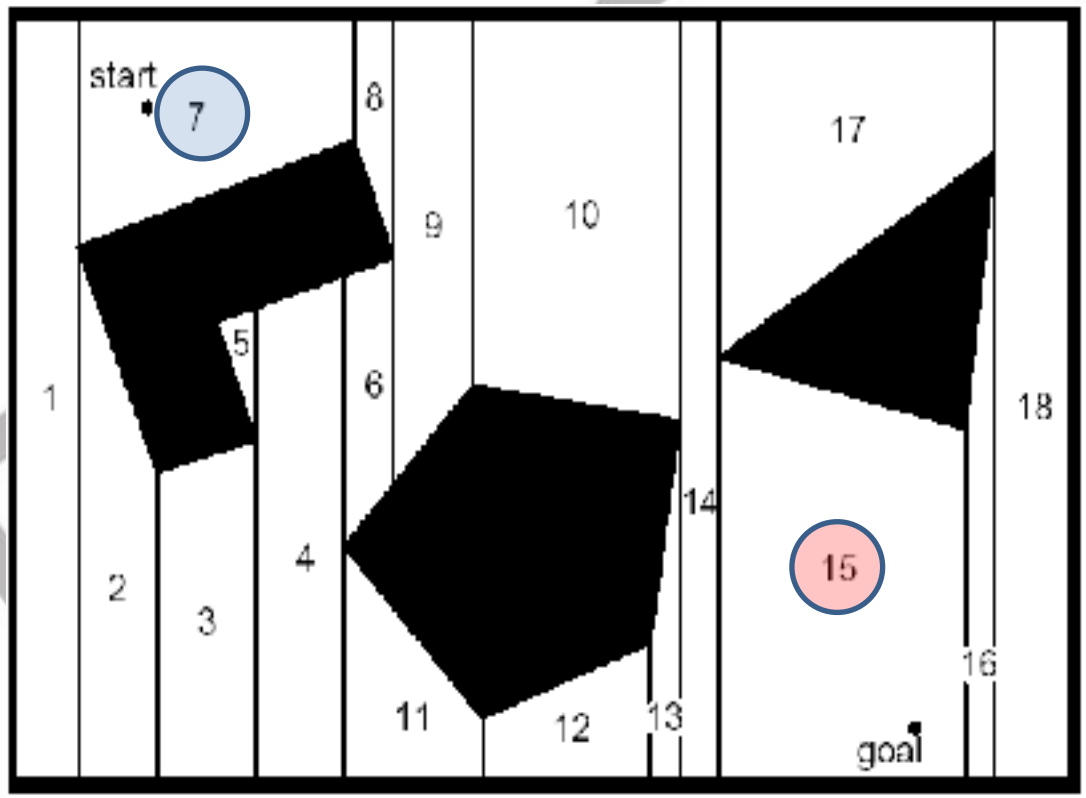
Prof. Dr. Marcelo Becker - SEM – E

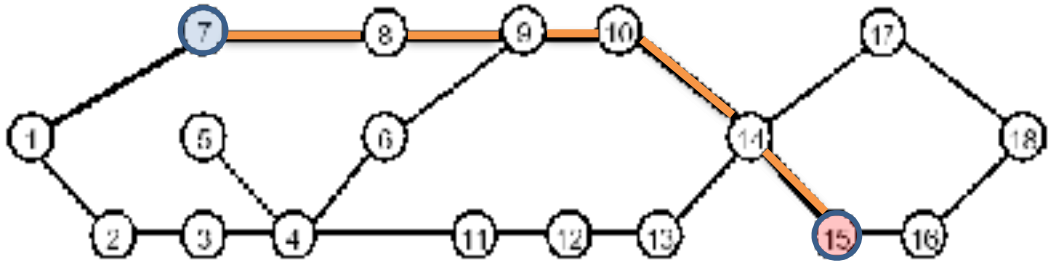
Cell Decomposition

- ***Idéia Básica***:
  - A partir da seqüência de células encontrada no grafo, utilizar um algoritmo adequado para planejar a trajetória do robô dentro da células.
    - Por exemplo: margear as bordas das células, ir em direção ao centro das células, etc.
  - Variações do *Cell Decomposition*:
    - *Exact cell decomposition*
    - *Approximate cell decomposition*:
      - *Fixed cell decomposition*
      - *Adaptive cell decomposition*

Prof. Dr. Marcelo Becker - SEM - EESC – USP

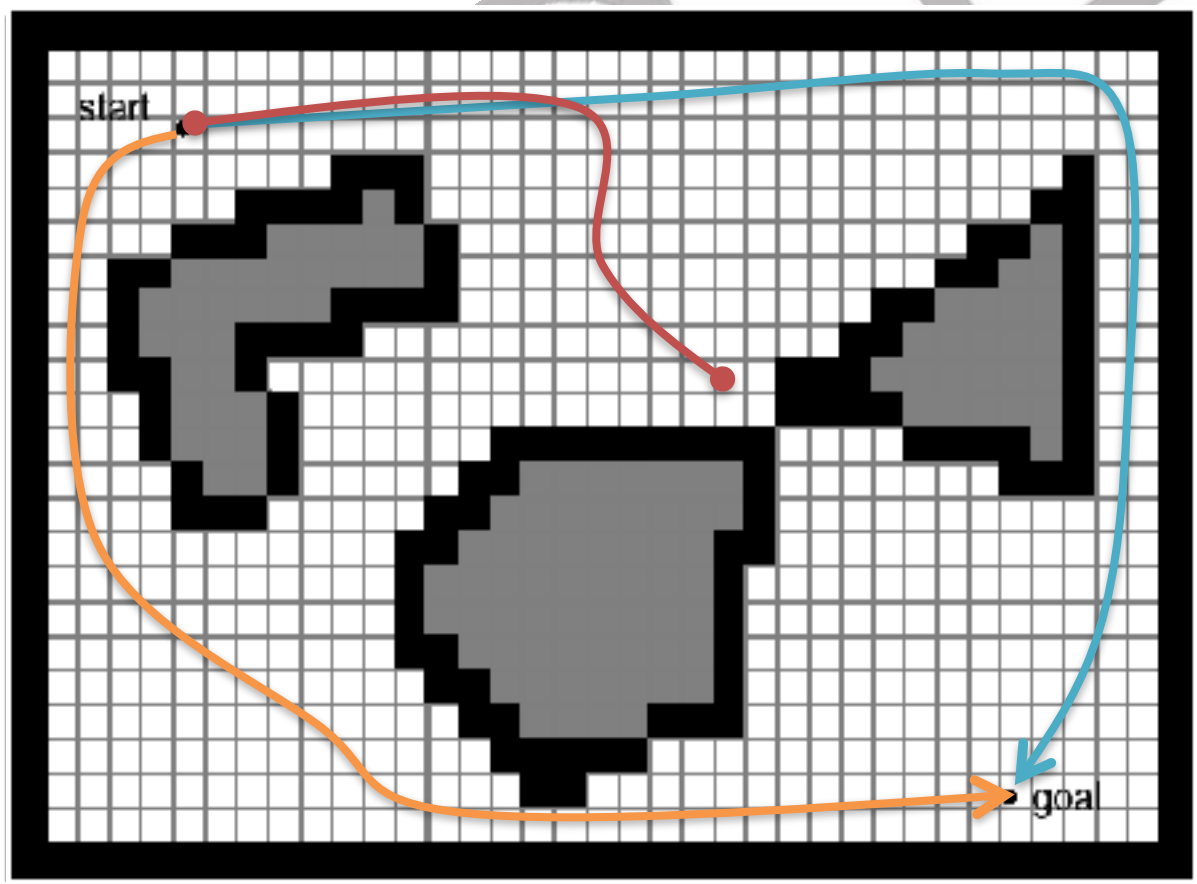
Exact Cell Decomposition





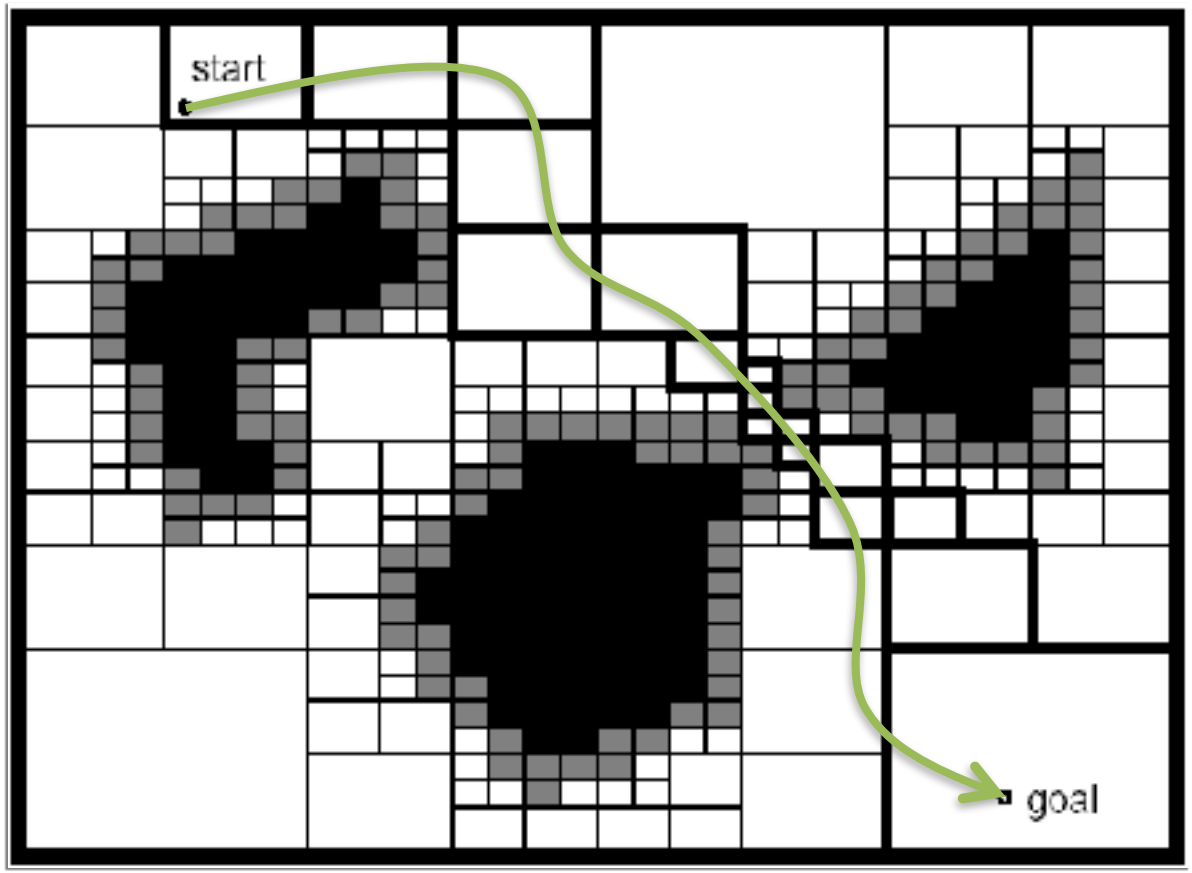
© M. Becker 2017

Fixed Cell Decomposition



© M. Becker 2017

Adaptive Cell Decomposition



© M. Becker 2017

Wavefront Expansion NF1

(Siegwart and Nourbakhsh, 2004)

- Baseada em Grafos

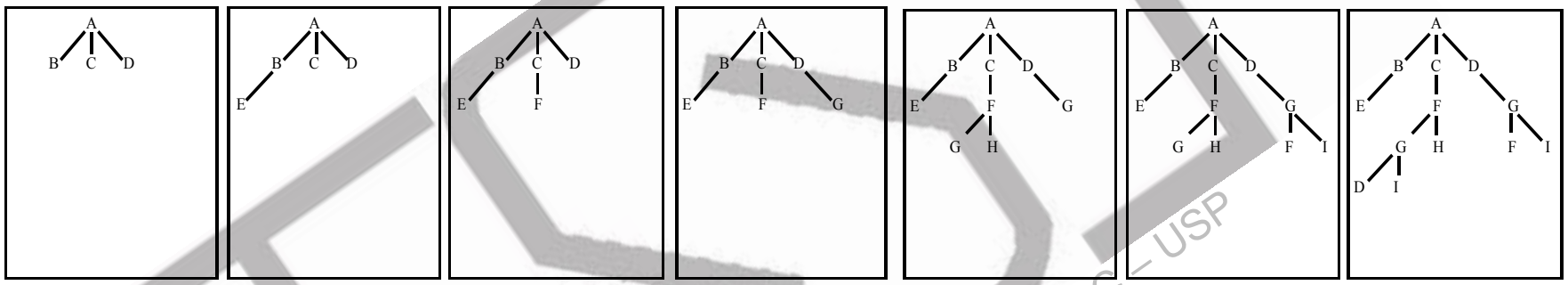
Prof. Dr. Marcelo Becker - SEM – EESC – USP

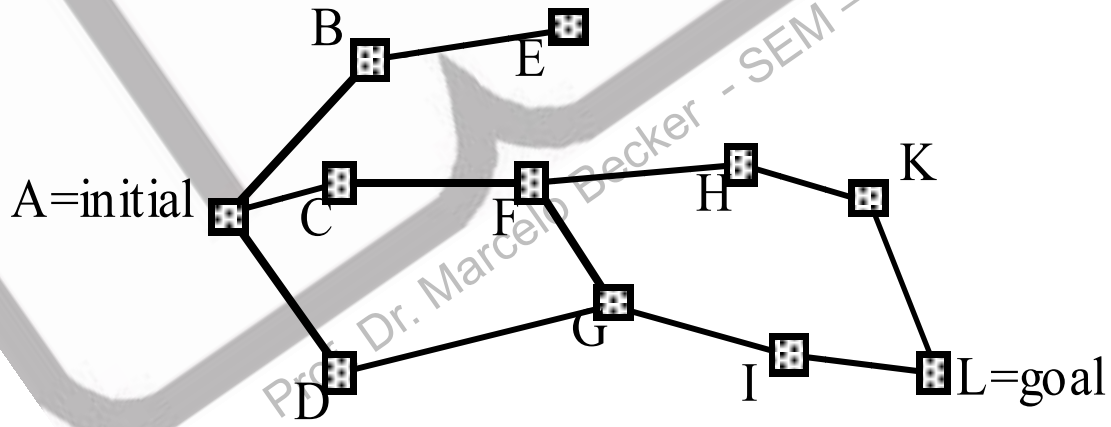
© M. Becker 2017

Breadth-First Search

(Knuth, 1997)

- Baseada em busca em Grafos



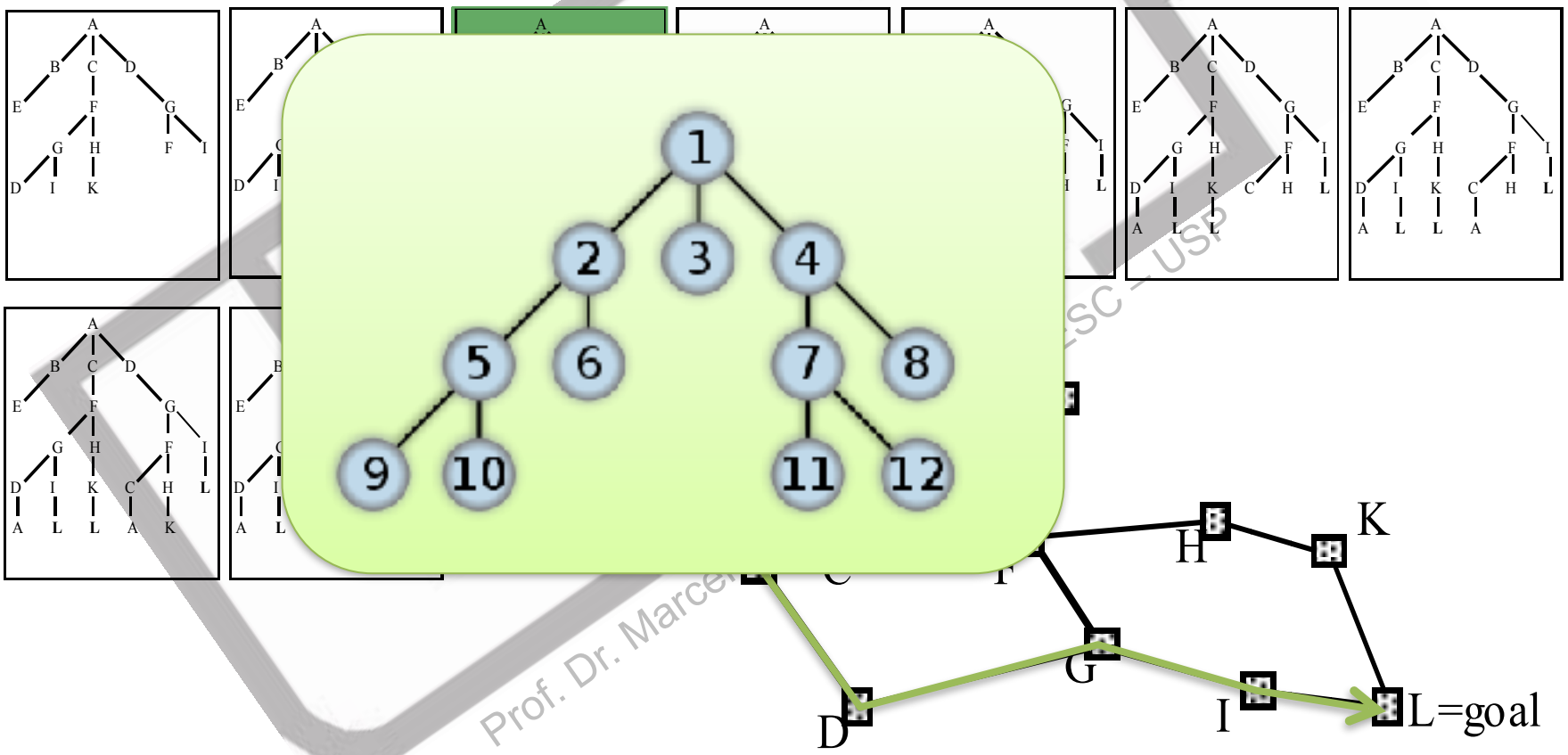


Prof. Dr. Marcelo Becker - SEM – EESC – USP

© M. Becker 2017

Breadth-First Search

- Continuando...



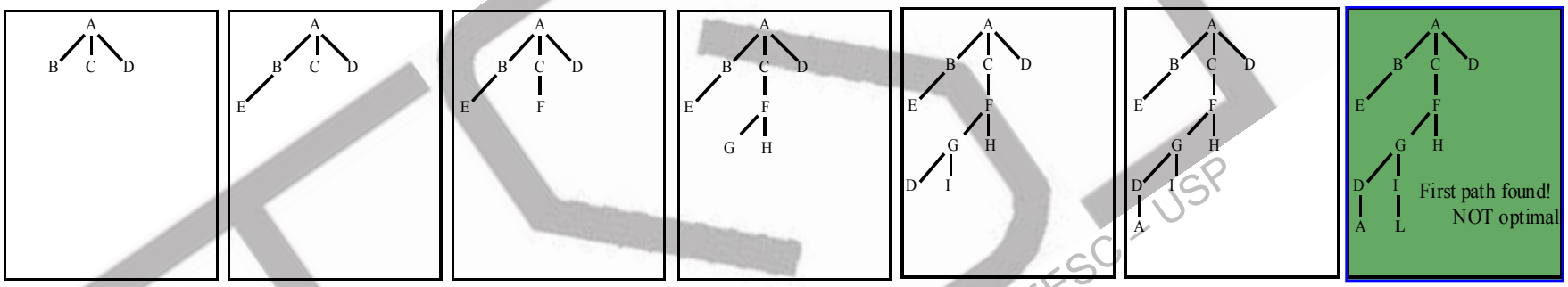
Prof. Dr. Marcel

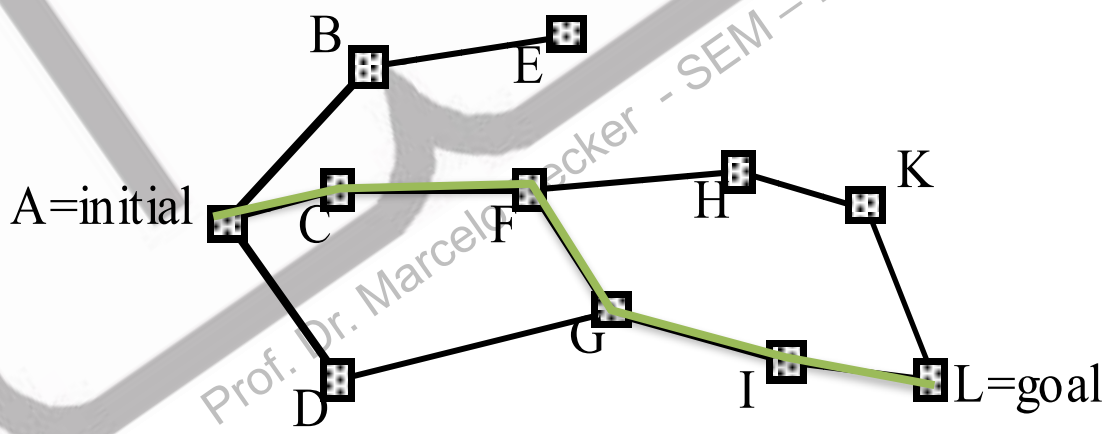
© M. Becker 2017

Depth-First Search

(Cormen et al., 2001)

- Também baseada em busca em Grafos



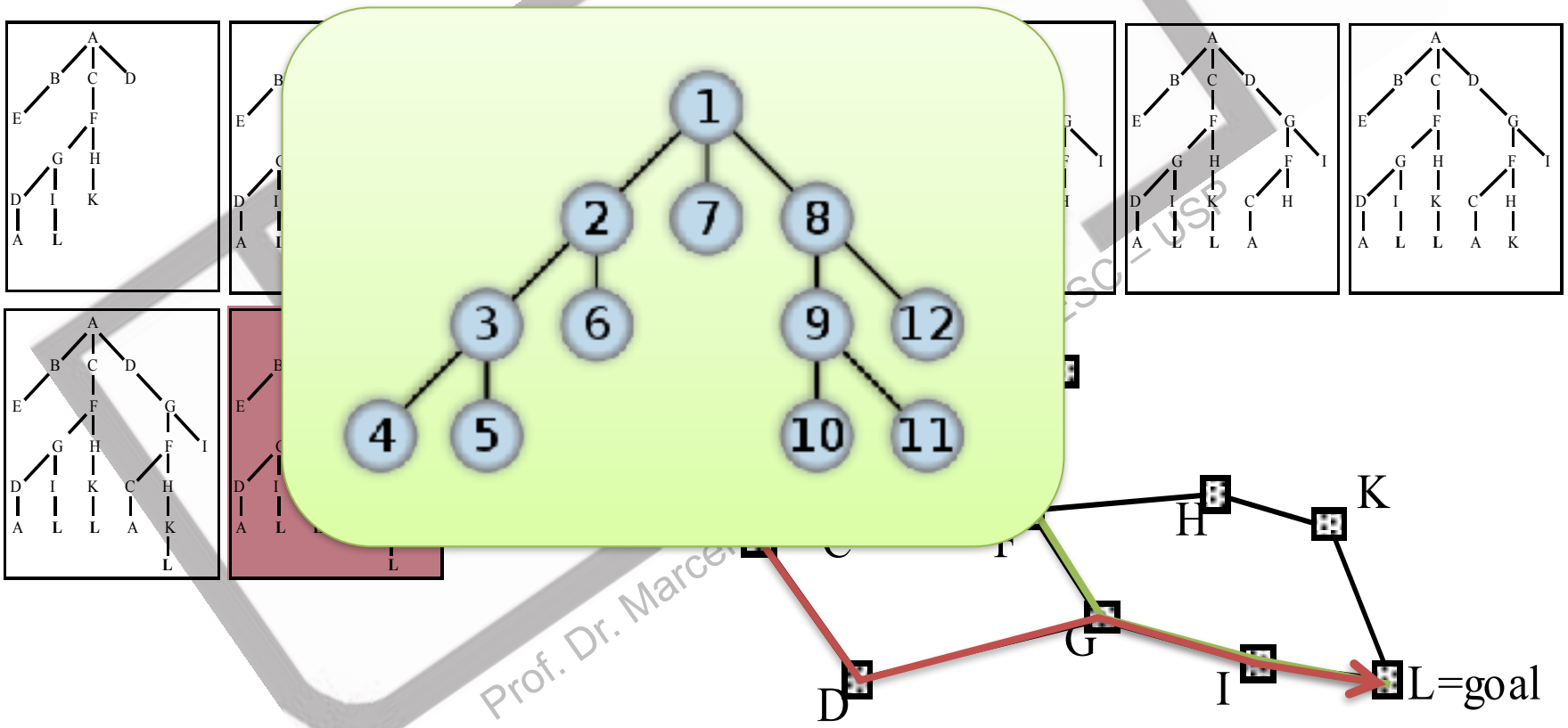


Prof. Dr. Marcelo Becker - SEM – EESC - USP

© M. Becker 2017

Depth-First Search

- Continuando...



Prof. Dr. Marcel

© M. Becker 2017

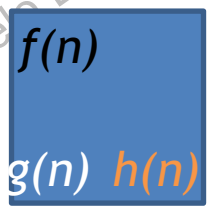
Algoritmo A*

(Pearl, 1984)

- Algoritmo baseado em grafo que busca minimizar uma função de custo:

$$f(n) = g(n) + h(n)$$

  – Usa-se uma função de custo *f(n)* que é igual à soma dos custos da trajetória *g(n)* do nó de início ao n-ésimo nó e da distância *h(n)* em linha reta do n-ésimo à posição desejada (*goal*).



Prof. Dr. Marcelo Becker - SEM - EESC - USP

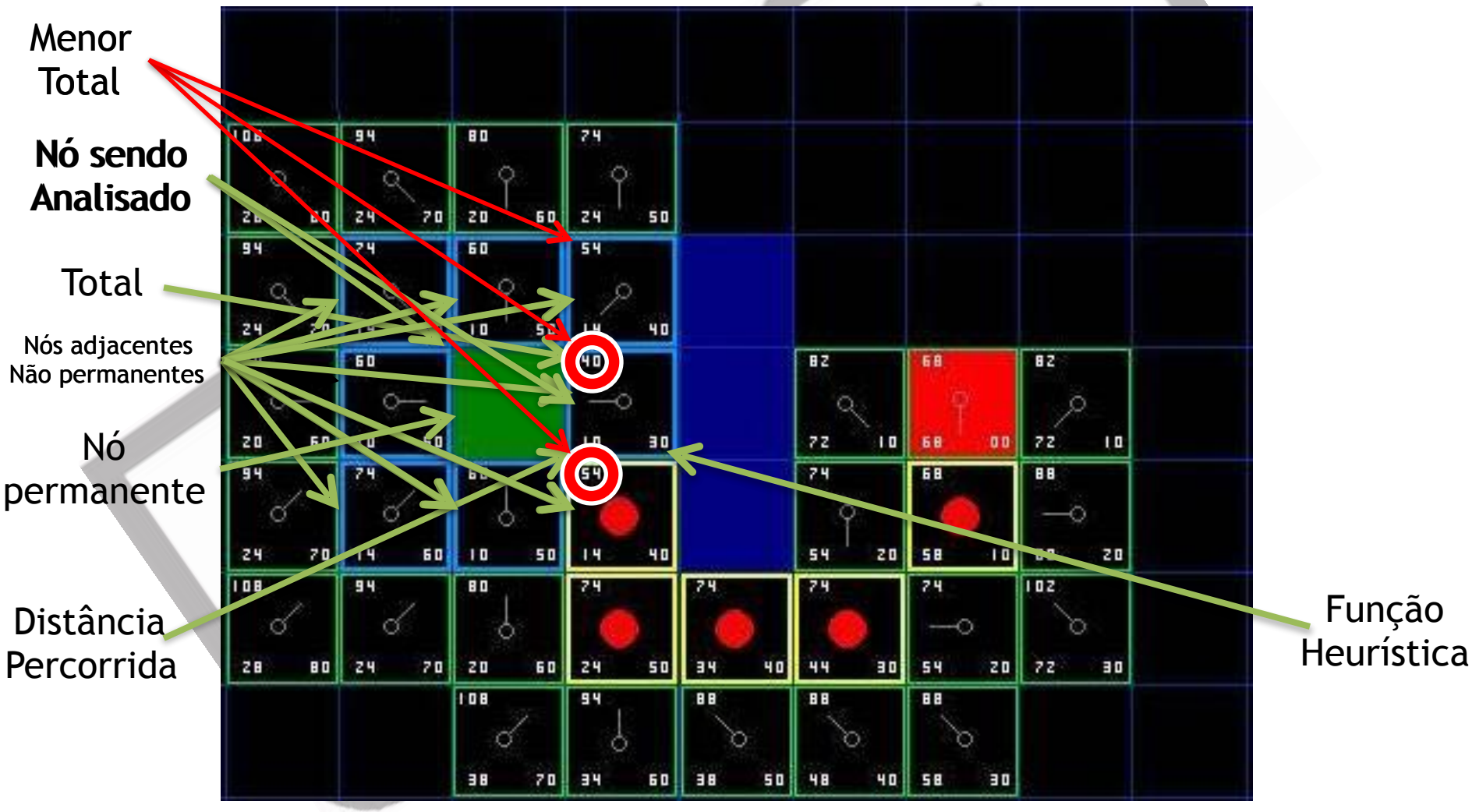
Algoritmo A*

- Requisitos do A*
  - Uma raiz da busca
  - Arestas com pesos não negativos
- ***Prós***:
  - Menor custo computacional
  - Resposta rápida
- ***Contras***:
  - Não fornece necessariamente o melhor caminho

Prof. Dr. Marcelo Becker - SEM – EESC – USP

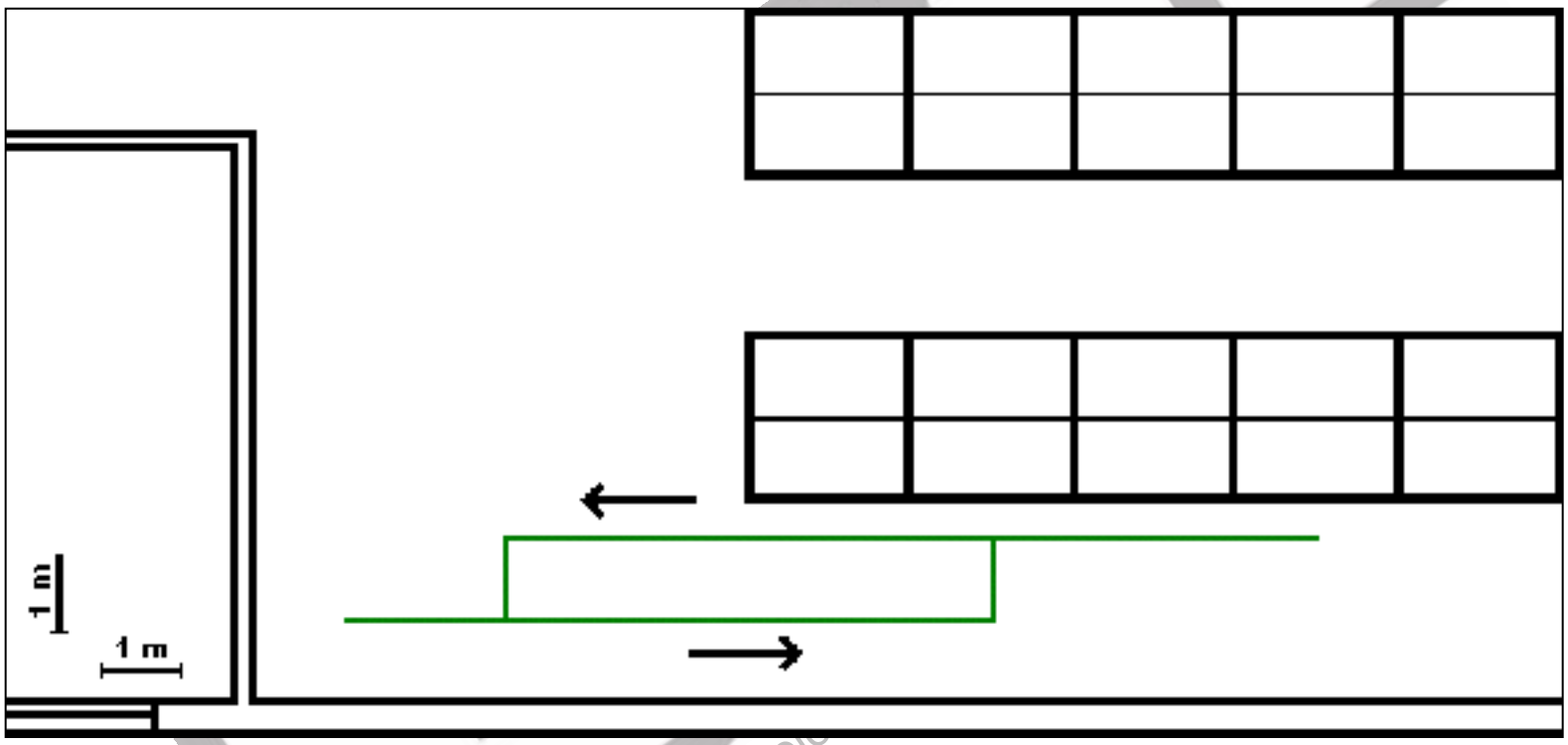
© M. Becker 2017

Algoritmo A*



© M. Becker 2017

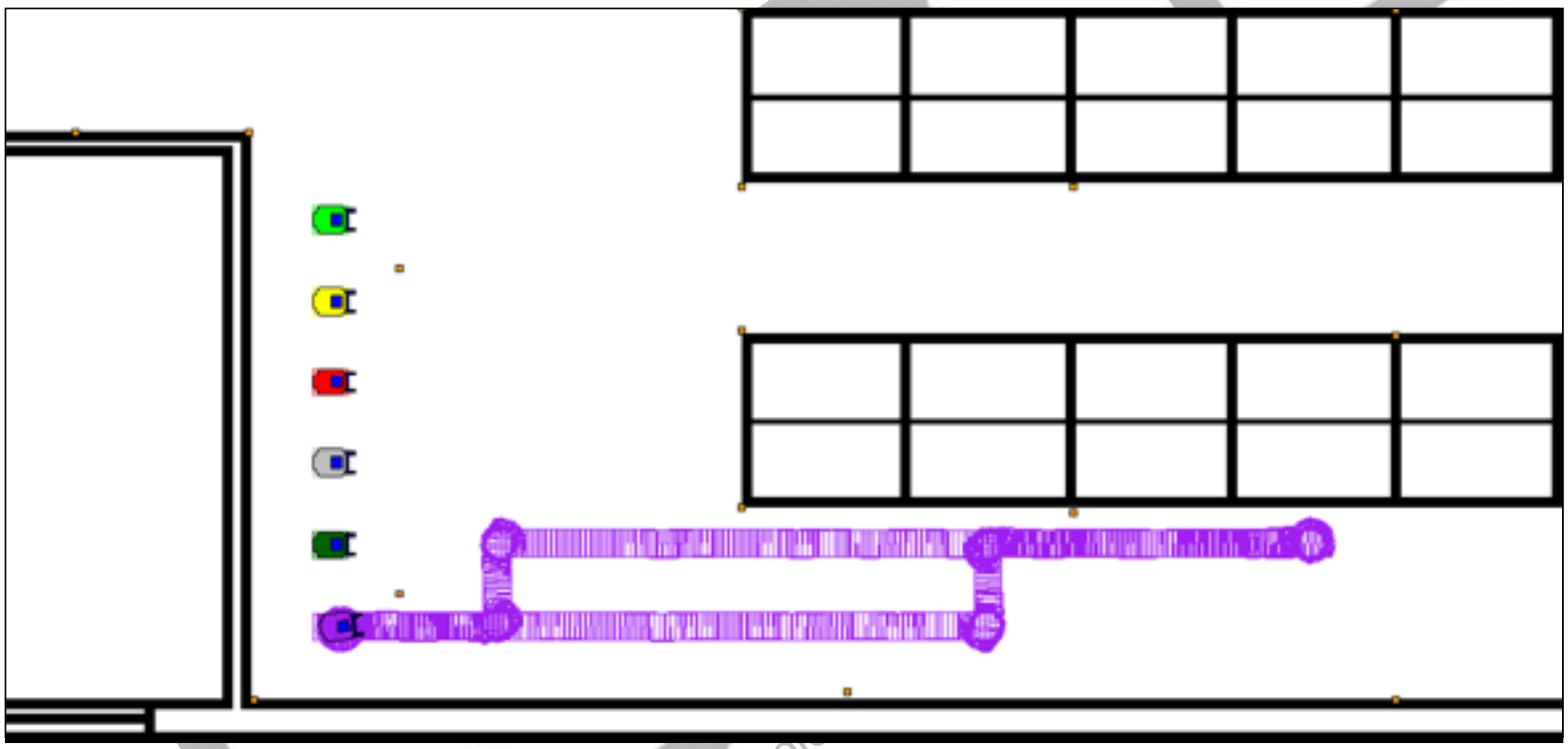
Algoritmo A*



Prof. Dr. Marcelo

© M. Becker 2017

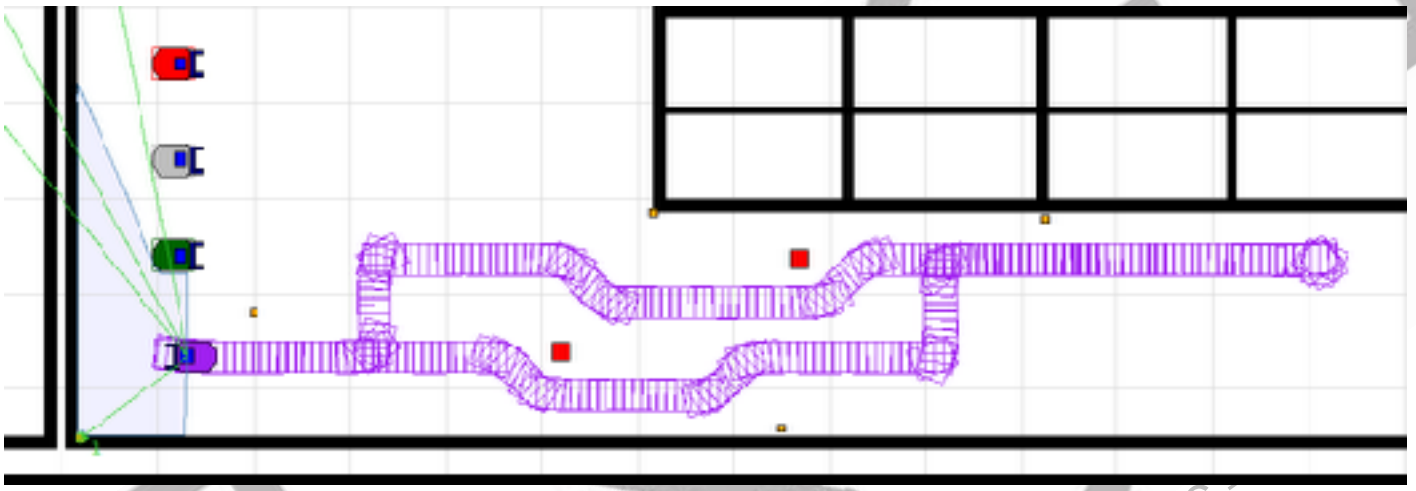
Algoritmo A*



Prof. Dr. Marcel

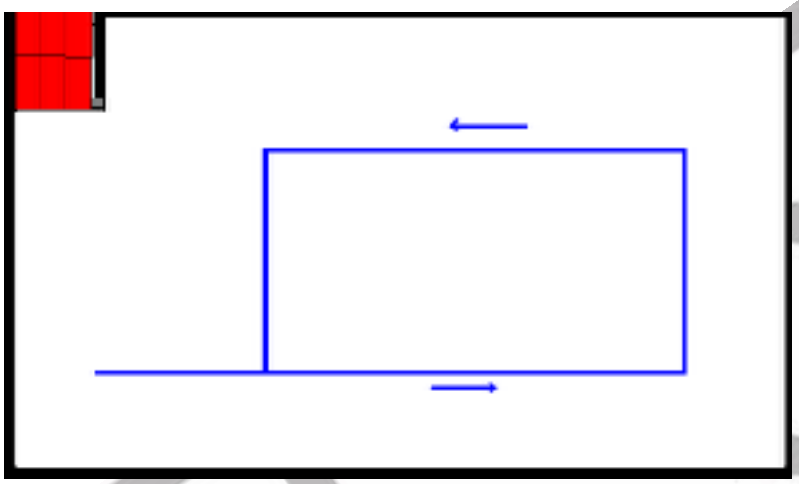
© M. Becker 2017

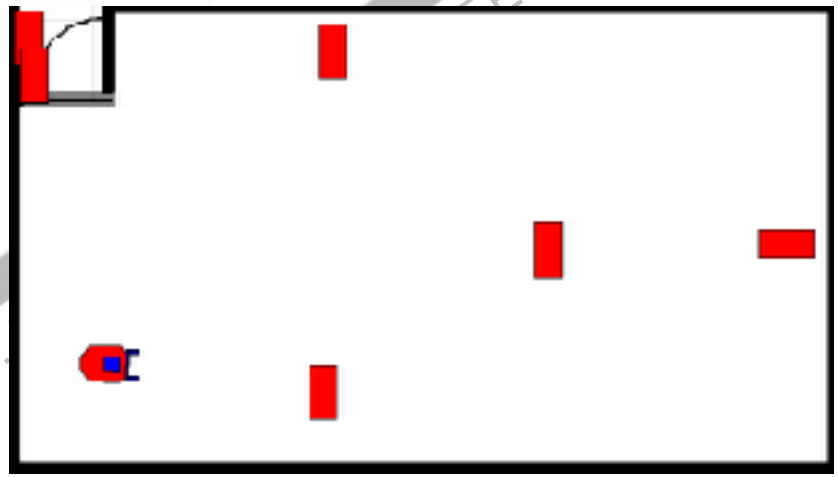
Algoritmo A*



© M. Becker 2017

Algoritmo A*

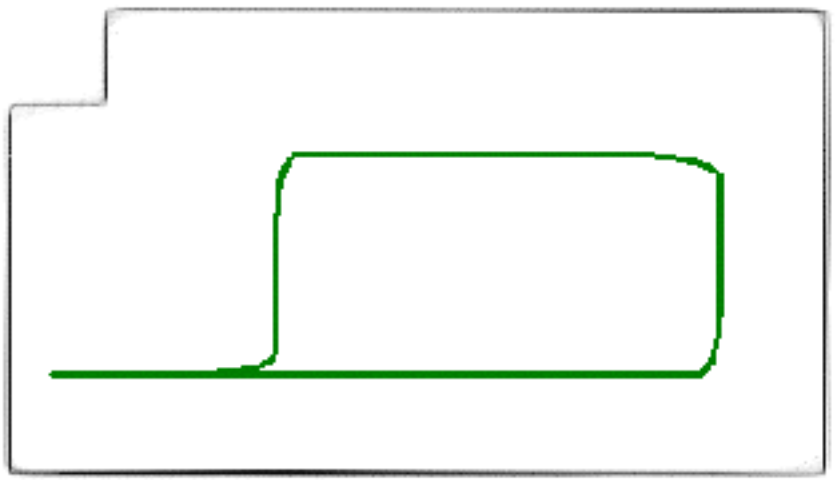


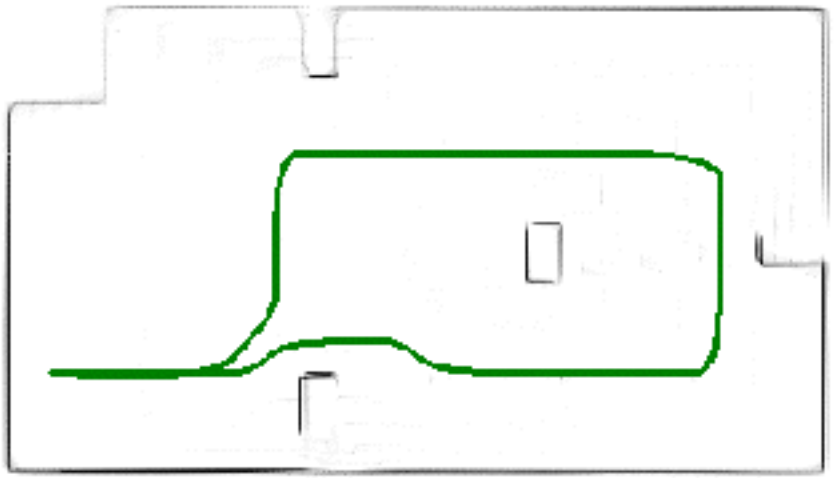


Prof. Dr. ... EESC – USP

© M. Becker 2017

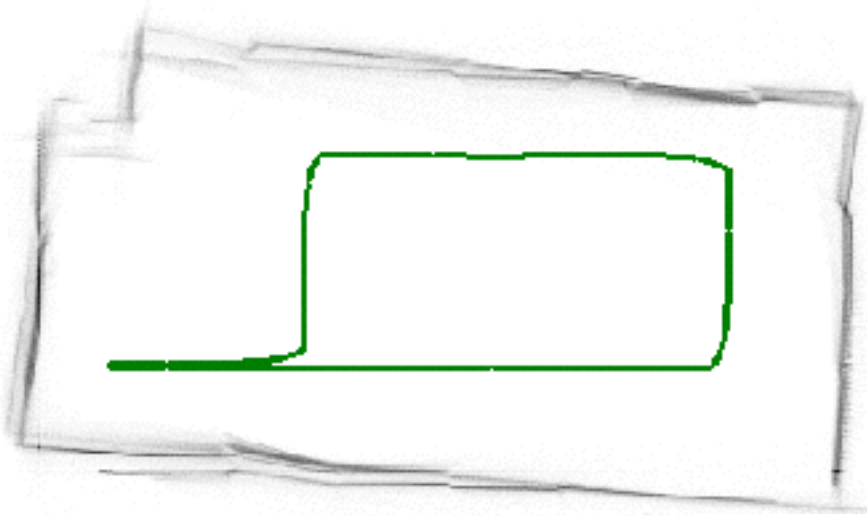
Algoritmo A*

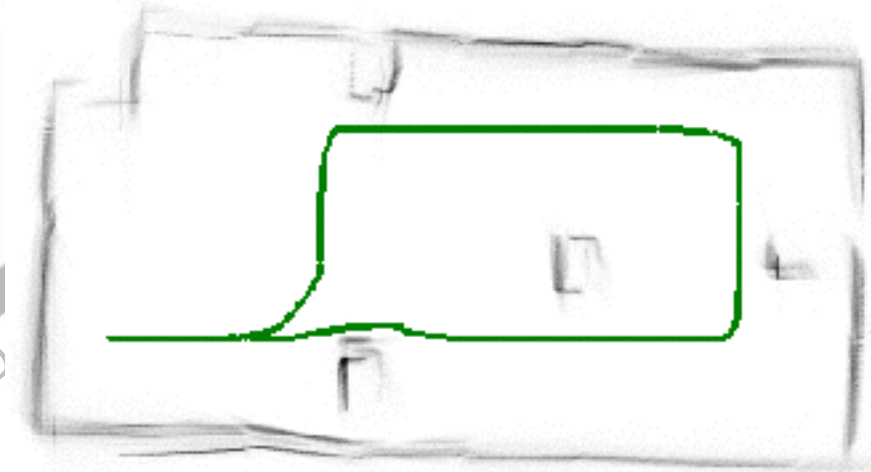




© M. Becker 2017

Algoritmo A*





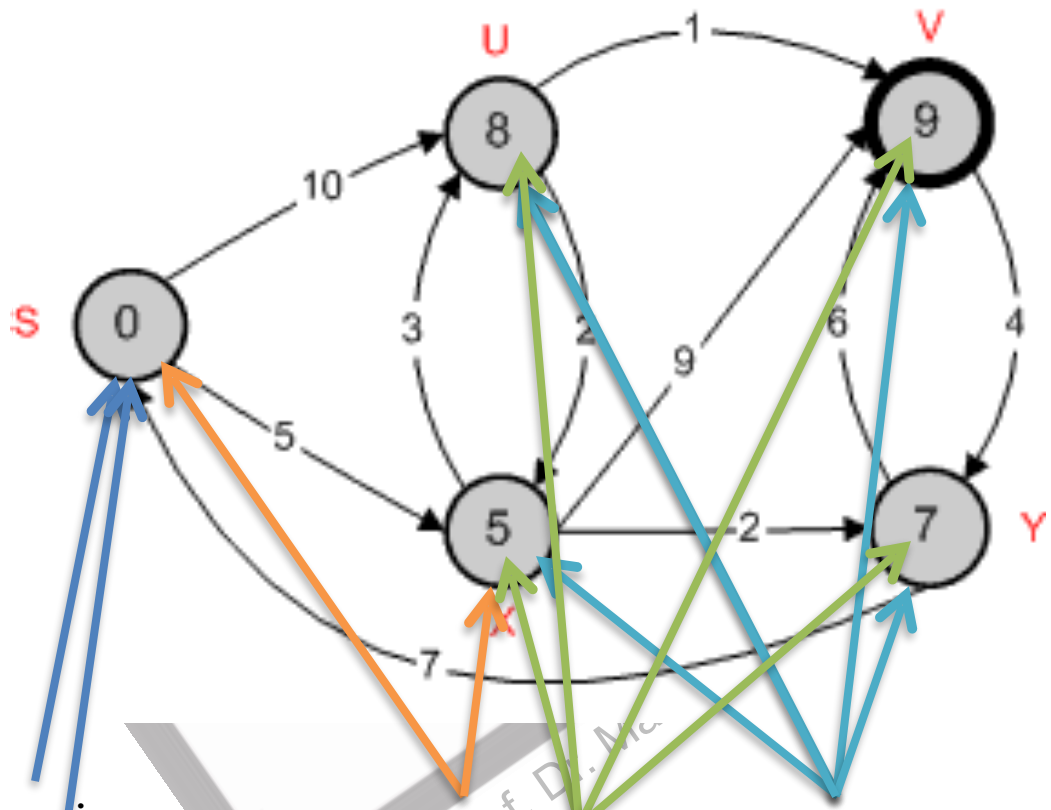
EESC – USP
Prof. D

© M. Becker 2017

Algoritmo de Dijkstra

(Dijkstra, 1959)

- Busca baseada em Grafos



vértice	Perm	Dist	Path
s	sim	0	-
u	sim	10	s
x	sim	5	s
v	sim	9	u
y	sim	7	x

Nó de origem
(Raiz da busca)
Nó permanente

Nó sendo
analisado

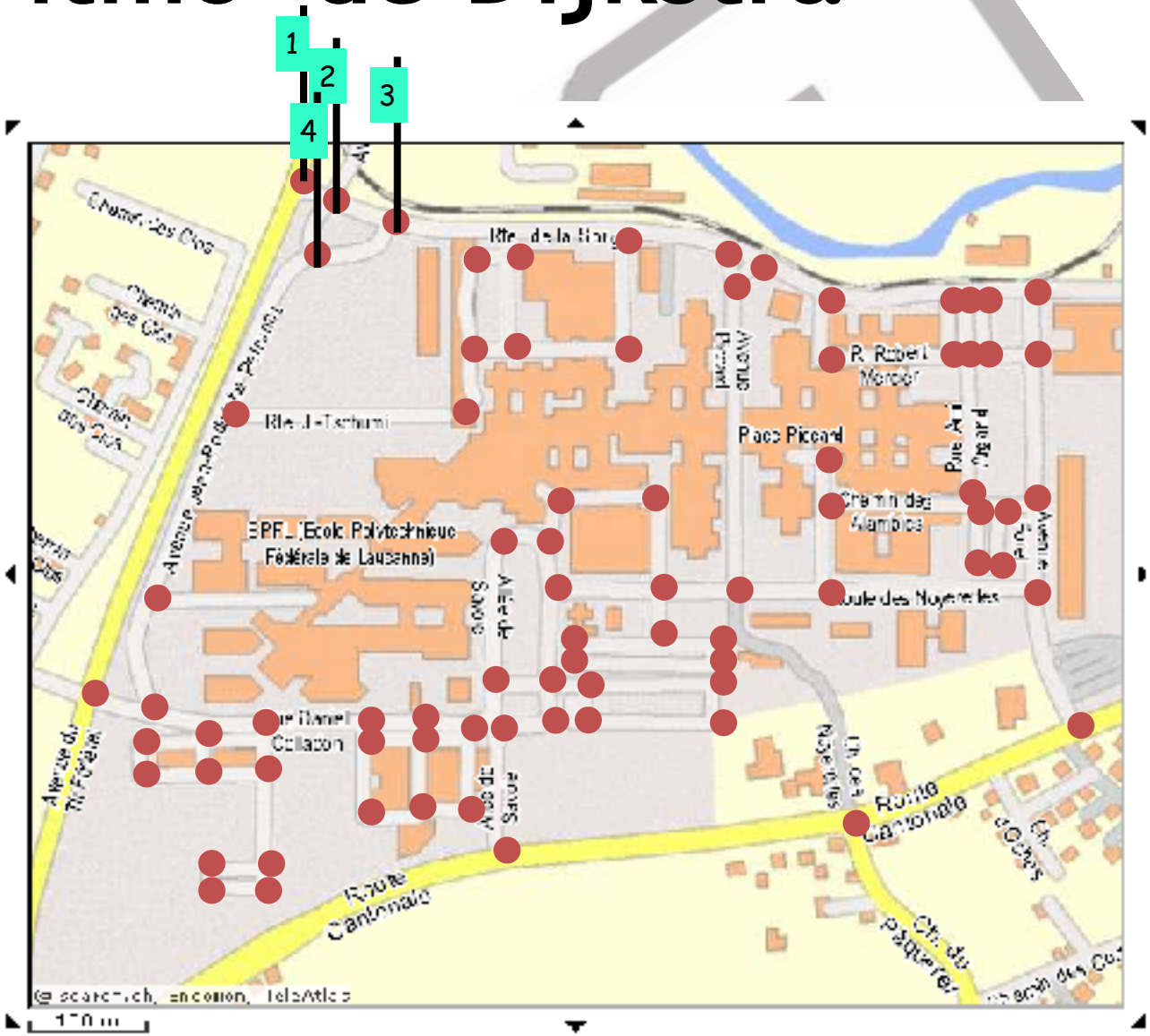
Custo da origem até o
nó

Nós adjacentes
Não permanentes

© M. Becker 2017

Algoritmo de Dijkstra

~ 80 nós



© M. Becker 2017

Algoritmo de Dijkstra

Matriz de Conexão

$$
\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 1 & 0 & 1 & 0 \\
2 & 0 & 1 & 0 & 1 \\
3 & 1 & 0 & 1 & 1 \\
4 & 0 & 1 & 1 & 1
\end{array}
$$

Matriz de Distâncias

$$
\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & \infty & 88 & \infty \\
2 & \infty & 0 & \infty & 30 \\
3 & 88 & \infty & 0 & 75 \\
4 & \infty & 30 & 75 & 0
\end{array}
$$

EESC – USP

© M. Becker 2017

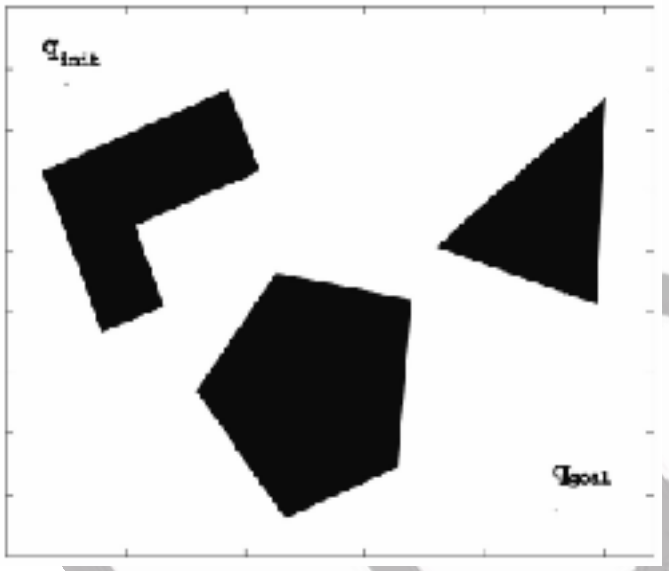
Campos Potenciais

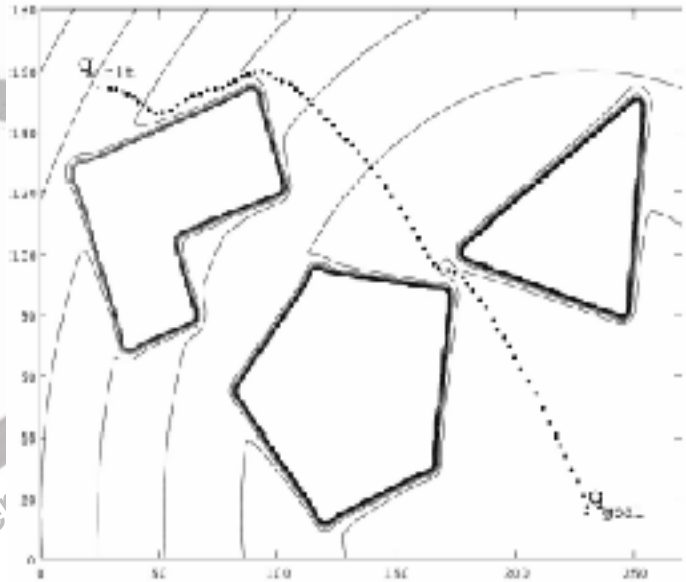
(Latombe, 1991)

- O robô é tratado como um ponto material sob a influência de um campo potencial artificial.
  - O movimento do robô assemelha-se ao de uma bola rolando em direção ao vale
  - A posição desejada (*Goal*) gera forças de atração.
  - Os Obstáculos, forças de repulsão.

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Campos Potenciais





Prof. Dr. Marcelo Becker

© M. Becker 2017

Campos Potenciais

- Geração do Campo Potencial: função *U(q)*:
  - Campos de atração (*Goal*) e repulsão (obstáculos).
  - Somatória dos Campos.
  - Funções devem ser diferenciáveis.
- Geração do Campos de Forças *F(q)*:

$$F(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) = \begin{bmatrix} \dfrac{\partial U}{\partial x} \\ \dfrac{\partial U}{\partial y} \end{bmatrix}$$

- Fazer com que a velocidade do robô (v_x, v_y) seja proporcional à força *F(q)* gerada pelo Campo.

Prof. Dr. Marcelo Becker

© M. Becker 2017

Campos Potenciais

- Forças de Atração:
  - Funções Parabólicas relacionadas à distância Euclidiana ao *Goal*: $\rho_{goal} = \left\| q - q_{goal} \right\|$

$$
\begin{aligned}
U_{att}(q) &= \frac{1}{2} k_{att} \cdot \rho_{goal}^2(q) \\
&= \frac{1}{2} k_{att} \cdot (q - q_{goal})^2
\end{aligned}
$$

  - A força de atração converge linearmente a zero (*Goal*)

$$
\begin{aligned}
F_{att}(q) &= -\nabla U_{att}(q) \\
&= k_{att} \cdot (q - q_{goal})
\end{aligned}
$$

© M. Becker 2017

Campos Potenciais

- Forças de Repulsão: devem gerar uma barreira entorno dos obstáculos:
  - A força é forte se o robô está próximo ao obstáculo e é desprezível, se o robô está longe do obstáculo.

$$U_{rep}(q) = \begin{cases} \frac{1}{2} k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right)^2 & \text{se } \rho(q) \leq \rho_0 \\ 0 & \text{se } \rho(q) \geq \rho_0 \end{cases}$$

  - $\rho(q)$: mínima distância ao obstáculo
  - O campo é positivo ou nulo e tende ao infinito quando q aproxima-se do obstáculo

Prof. Dr. Marcelo Becker

© M. Becker 2017

Campos Potenciais

- Forças de Repulsão:

$$F_{rep}(q) = -\nabla U_{rep}(q)$$

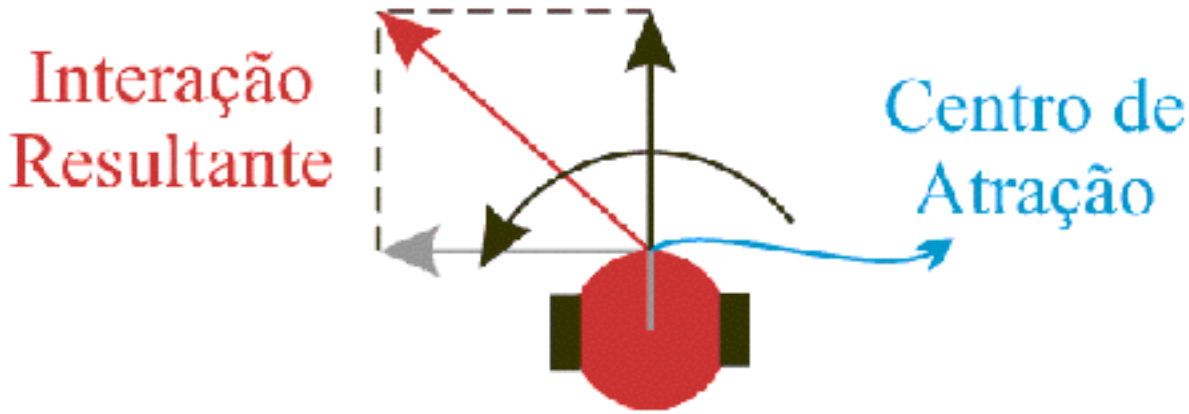
$$= \begin{cases} k_{rep}\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2(q)}\frac{q - q_{obst}}{\rho(q)} & \text{if } \rho(q) \leq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases}$$

Prof. Dr. Marcelo Becker

© M. Becker 2017

Campos Potenciais

- Definição da Direção de Movimento



Prof. Dr. Marc

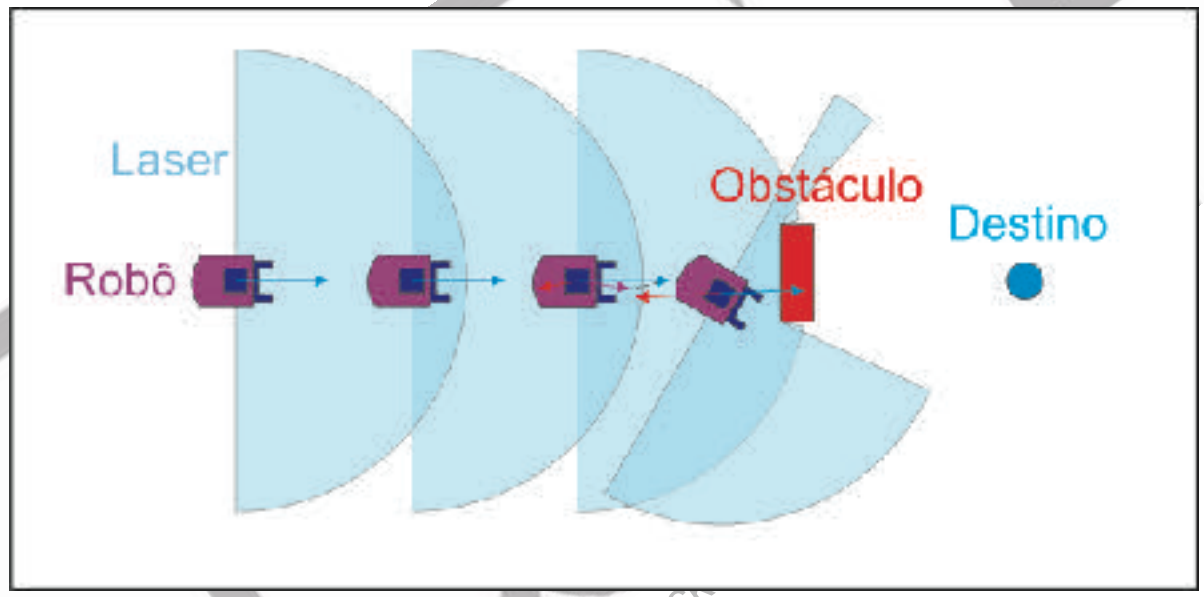
© M. Becker 2017

Campos Potenciais

- ***Observações***:
  - Presença de Mínimos Locais
  - O problema fica mais complexo se o robô não for considerado como ponto material
  - Caso os obstáculos tenham formato convexo, há situações onde ocorre a presença de várias distâncias mínimas $\rightarrow$ oscilações no movimento do robô

Prof. Dr. Marcelo Becker - SEM - EESC - USP

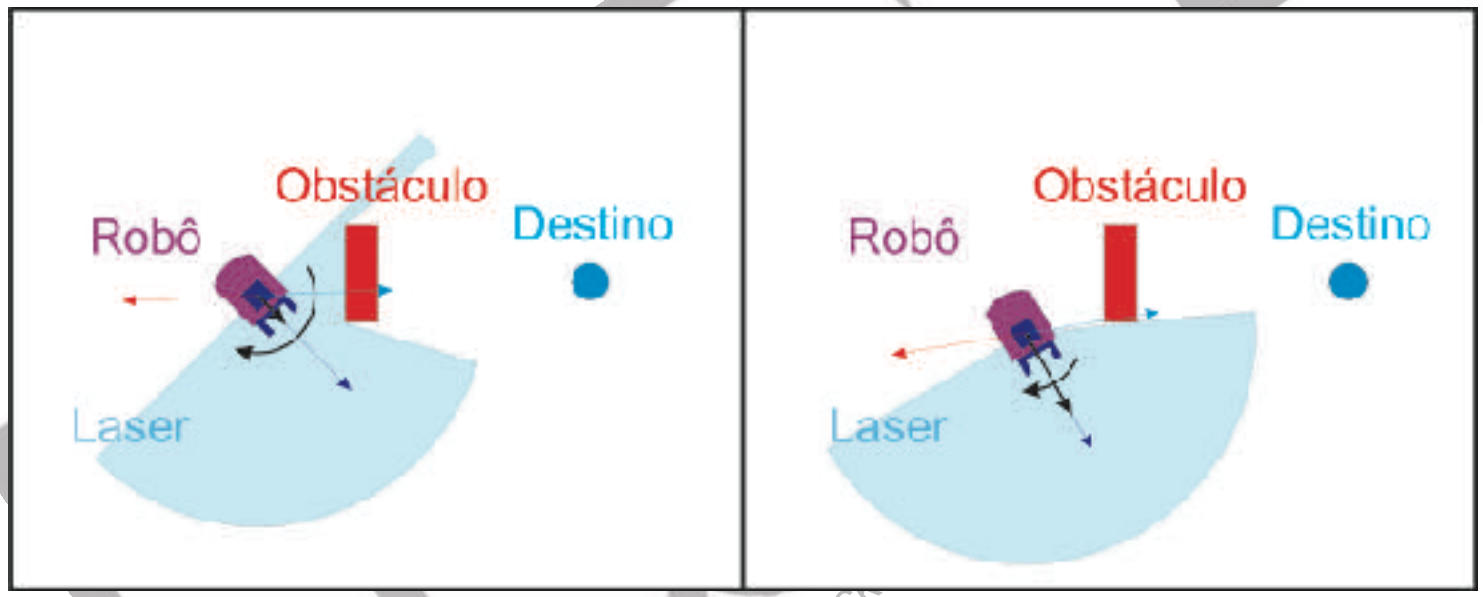
Campos Potenciais



Prof. Dr. Marcelo Beck

© M. Becker 2017

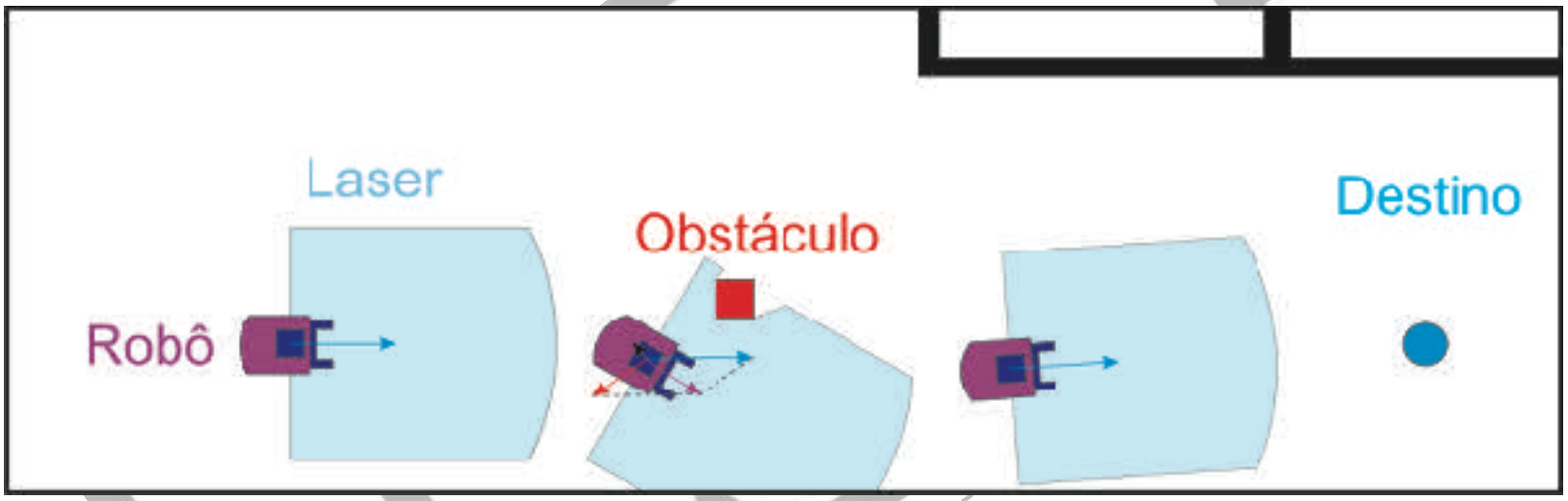
Campos Potenciais



Prof. Dr. Marcelo Beck

© M. Becker 2017

Campos Potenciais

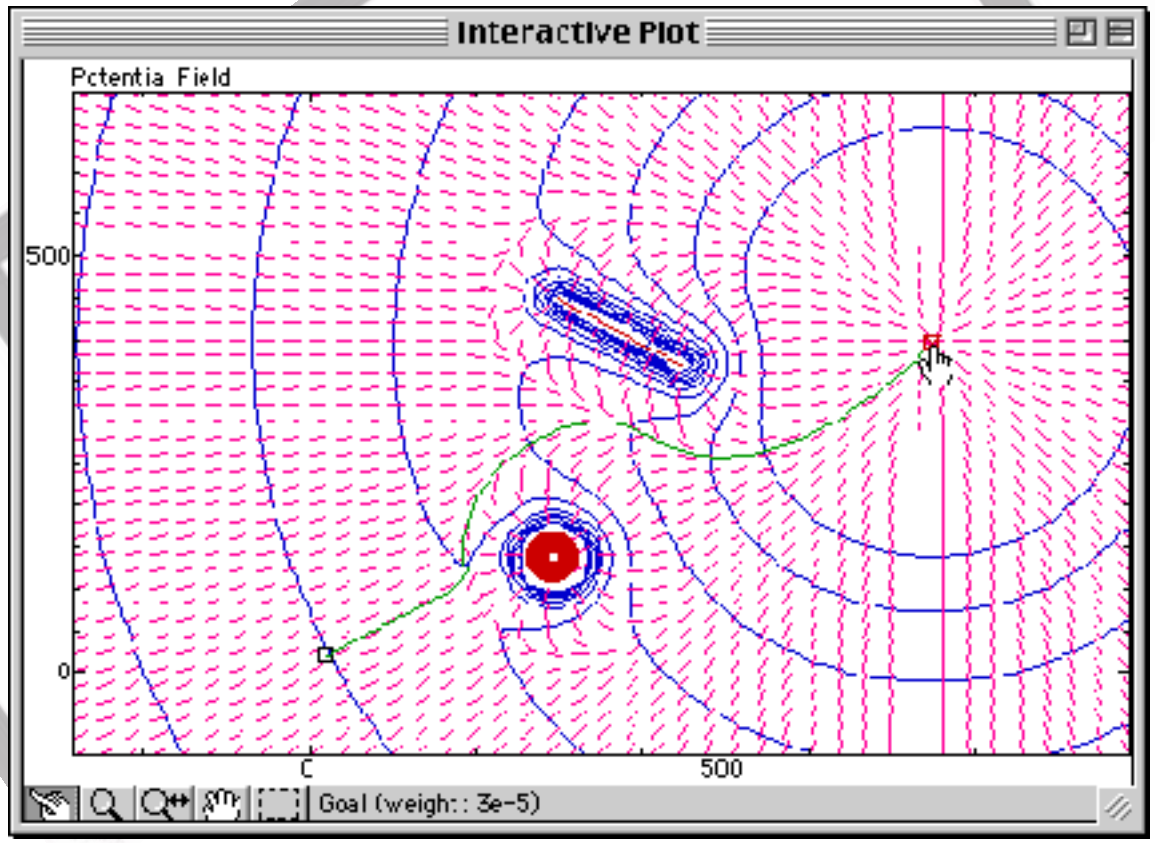


Prof. Dr. Marcelo Becker

© M. Becker 2017

Campos Potenciais

- ***Demo***: Software livre SysQuake LE (www.k-team.com)



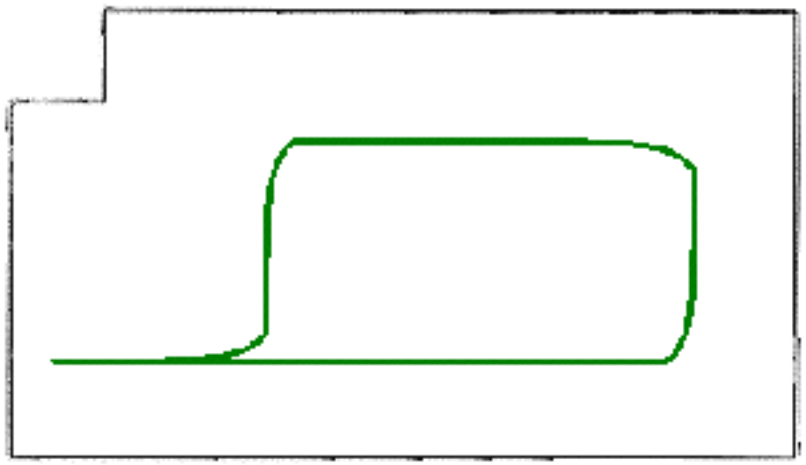
© M. Becker 2017

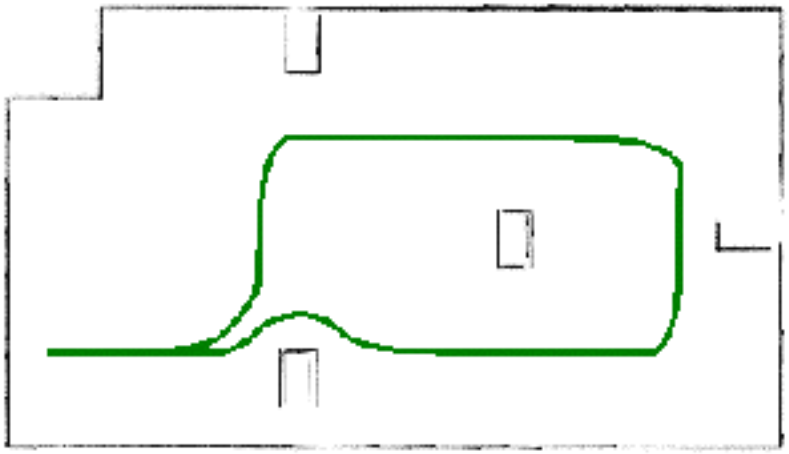
Campos Potenciais

Prof. Dr. Marce

© M. Becker 2017

Campos Potenciais

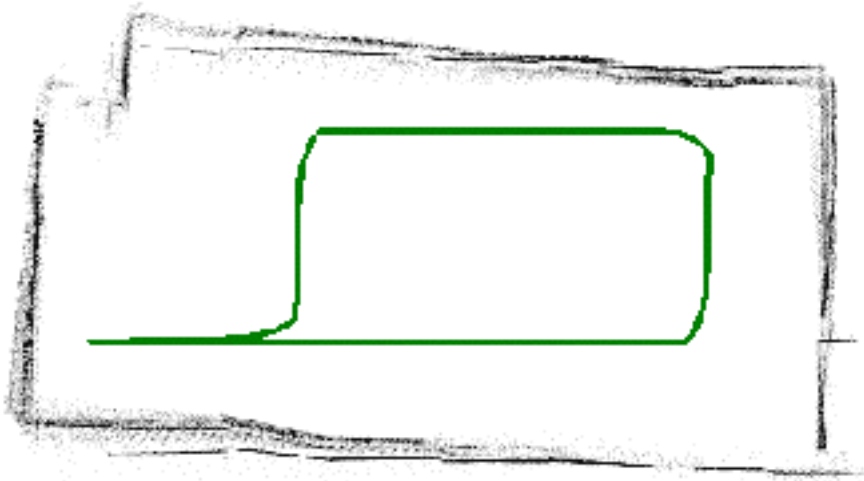


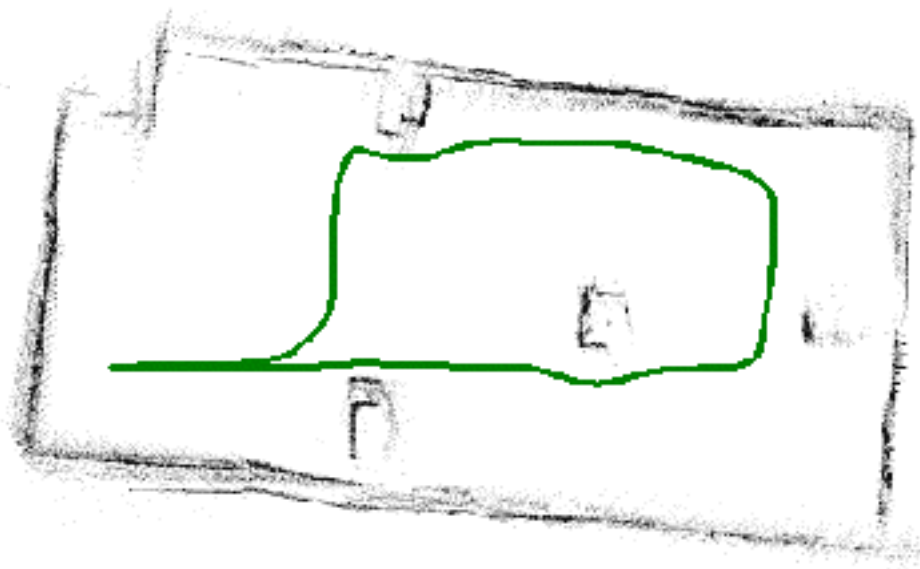


Prof. Dr. Marcelo
SC – USP

© M. Becker 2017

Campos Potenciais





Prof. Dr. Marcelo

© M. Becker 2017

Campos Potenciais Estendido

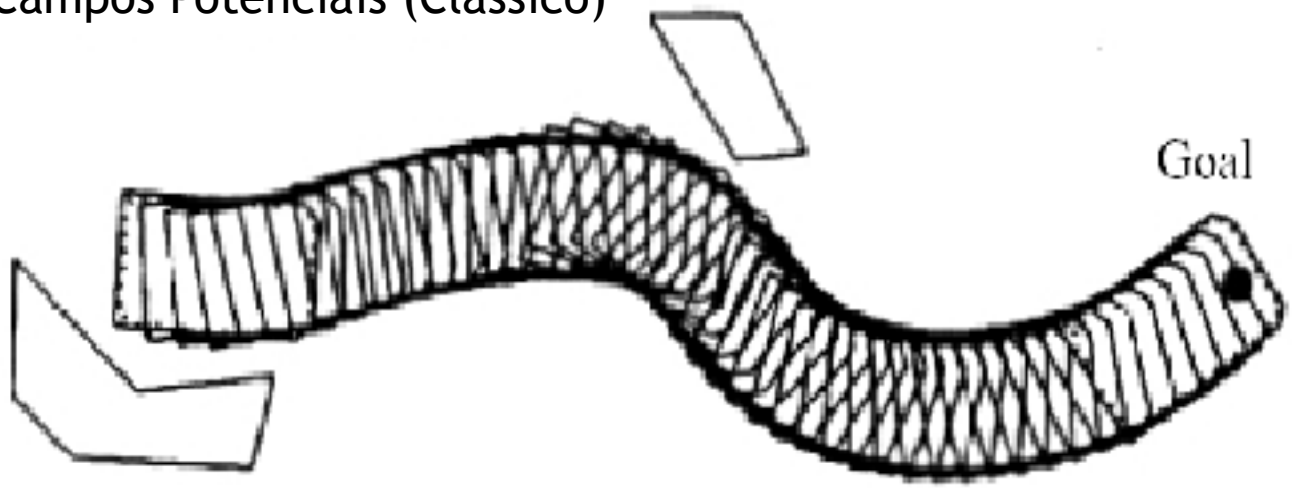
(Khatib and Chatila,)

- ***Rotation Potential Field***
  - A força é tb. uma função do ângulo de orientação do robô com relação aos obstáculos.
  - Introduz-se um fator de ganho (β) que reduz a força repulsiva quando o obstáculo está na direção paralela ao sentido de movimento do robô.
- ***Task Potential Field***
  - Obstáculos que não vão influenciar na movimentação do robô são filtrados.
  - Apenas os obstáculos na região em frente ao robô são considerados na geração dos campos potenciais

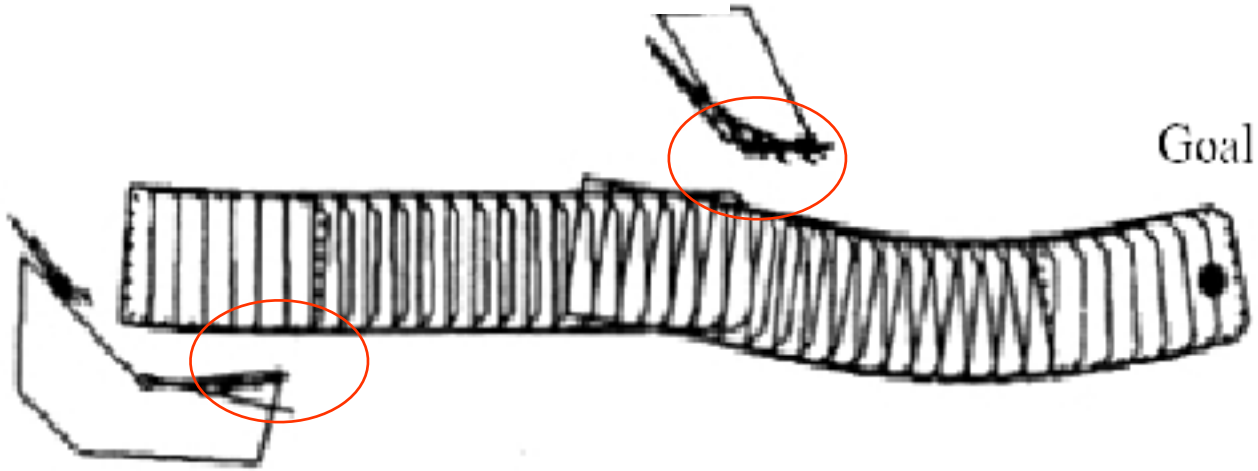
Prof. Dr. Marcelo Becker - SEM – EESC - USP

Campos Potenciais Estendido

Campos Potenciais (Clássico)



Rotation Potential Field (ganho β)



© M. Becker 2017

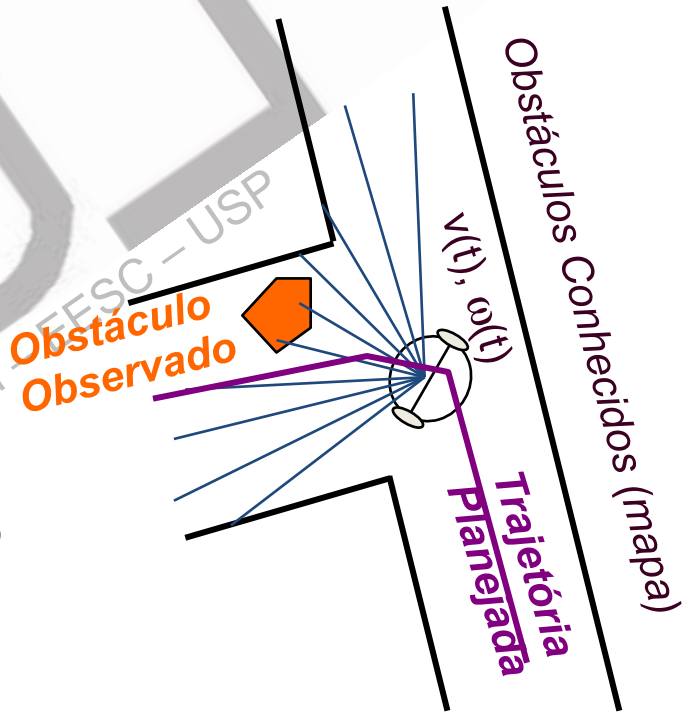
Planejamento Local da Trajetória

Prof. Dr. Marcelo Becker

© M. Becker 2017

Planejamento Local da Trajetória

- Baseia-se em um mapa local (em geral)
- Em geral, é implementado como uma tarefa independente
- O desvio de obstáculos deve levar em conta:
  - A posição desejada (*Goal*)
  - A velocidade e a dinâmica do robô
  - Os sensores embarcados
  - O risco atual e futuro de colisões



Prof. Dr. Marcelo Becker - SEM - EESC – USP

© M. Becker 2017

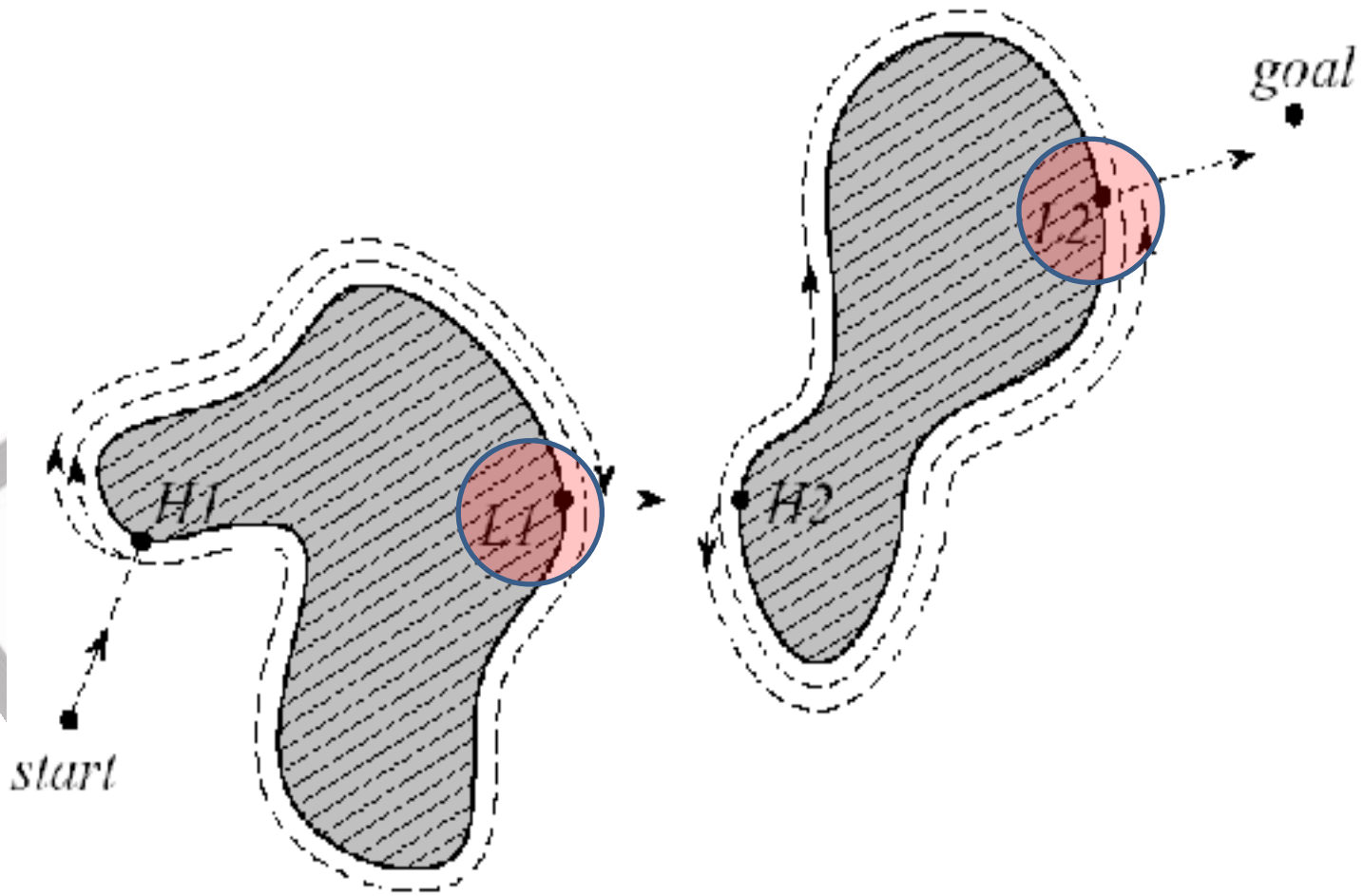
Bug 1

(Lumelsky and Stepanov, 1990)

- Comportamento primitivo: quando um obstáculo for encontrado ao logo da trajetória desejada, deve-se segui-lo, contornando-o completamente.
- Após isso, encontra-se o ponto mais próximo ao objetivo desejado (*Goal*) e retoma-se a trajetória.

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Bug 1



© M. Becker 2017

Bug 2

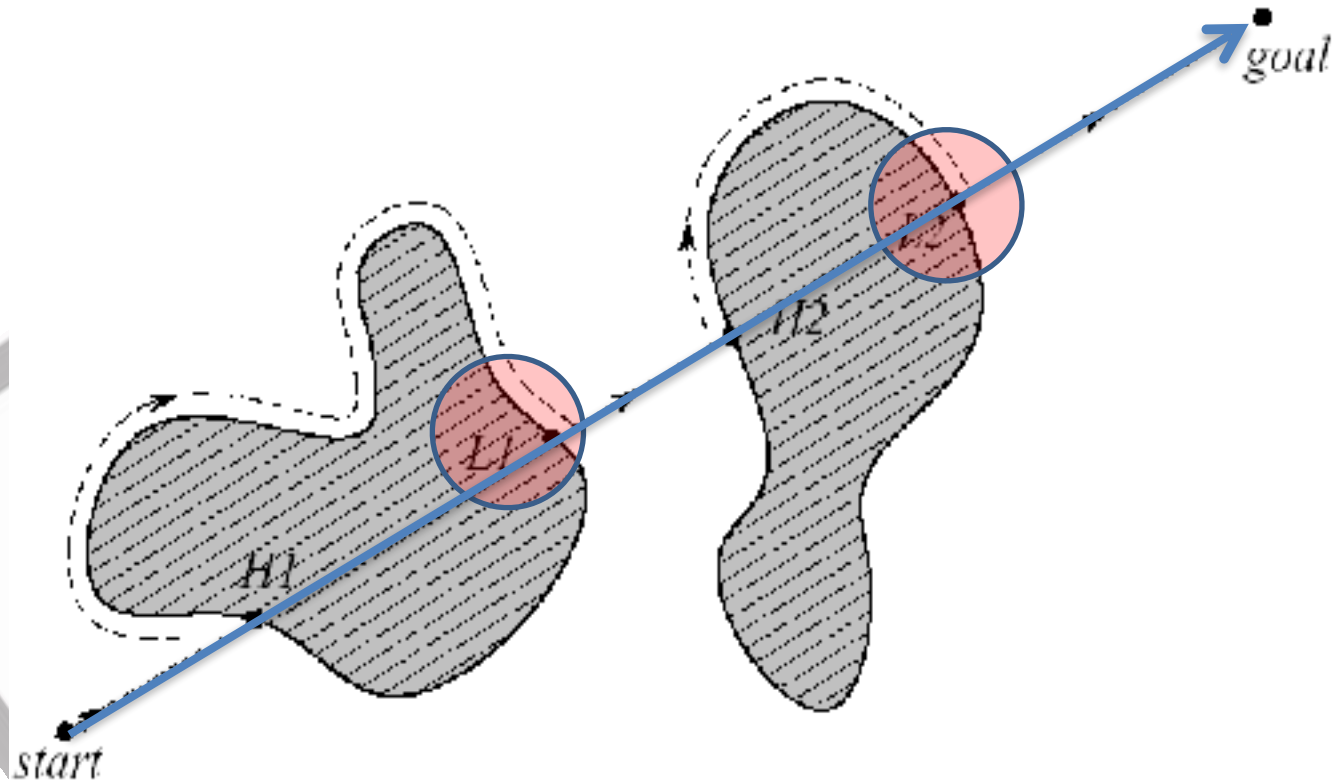
(Lumelsky and Stepanov, 1990)

- O obstáculo é contornado seguindo uma heurística (S.H. ou S.A.H.).

- Assim que possível retomar uma trajetória em direção ao *Goal*, o robô pára de contornar o obstáculo.

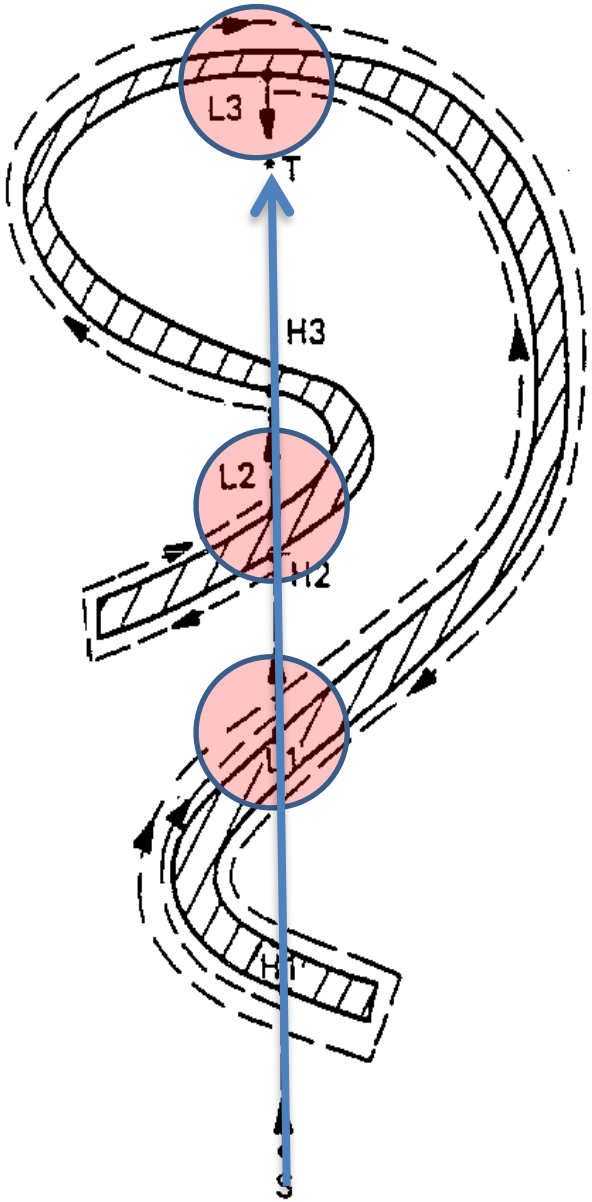
Prof. Dr. Marcelo Becker - SEM – EESC – USP

Bug 2



© M. Becker 2017

Bug 2

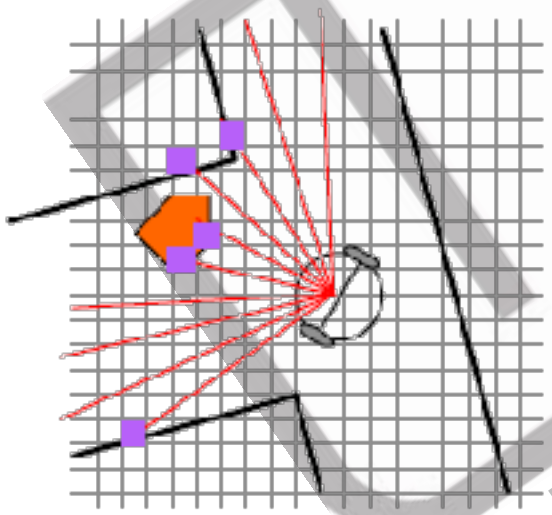


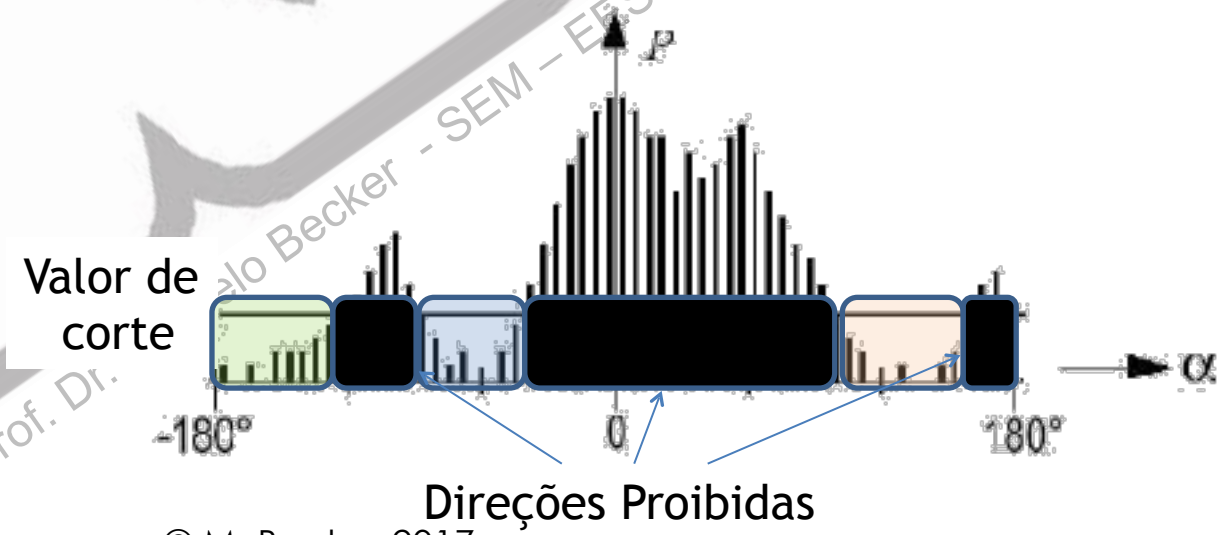
© M. Becker 2017

Vector Field Histogram - VFH

(Borenstein and Koren, 1991)

- O ambiente é representado por um *Grid* de Células (2D)
  - Cada Célula possui um valor relativo à probabilidade de estar ocupada.
  - Obtém-se um histograma polar que mostra as direções de esterçamento que evitam colisões (problema em 1D).





Prof. Dr. ...alo Becker - SEM – EESC – USP

© M. Becker 2017

Vector Field Histogram - VFH

- Passos para a determinação da direção ideal de esterçamento :
  1. Determina-se todas as direções em que o robô pode passar.
  2. Seleciona-se a de menor função custo *G*

$$G = a.\theta_{target} + b.\theta_{wheel} + c.\theta_{robô}(t-1)$$

θ_{target} é o ângulo entre a orientação do robô e a trajetória para o *Goal*

θ_{wheel} é a diferença entre a nova direção de esterçamento das rodas do robô e a direção atual de esterçamento

$\theta_{robô}(t-1)$ é a diferença entre a direção de esterçamento previamente selecionada e a nova direção de esterçamento

Prof. Dr. Marcelo Becker - EESC – USP

Vector Field Histogram - VFH

- Limitações:
  - Áreas estreitas (portas) e ambientes dinâmicos
  - Mínimos Locais
  - Não garante-se chegar ao objetivo (*Goal*)
  - Não se considera a dinâmica do robô

© M. Becker 2017

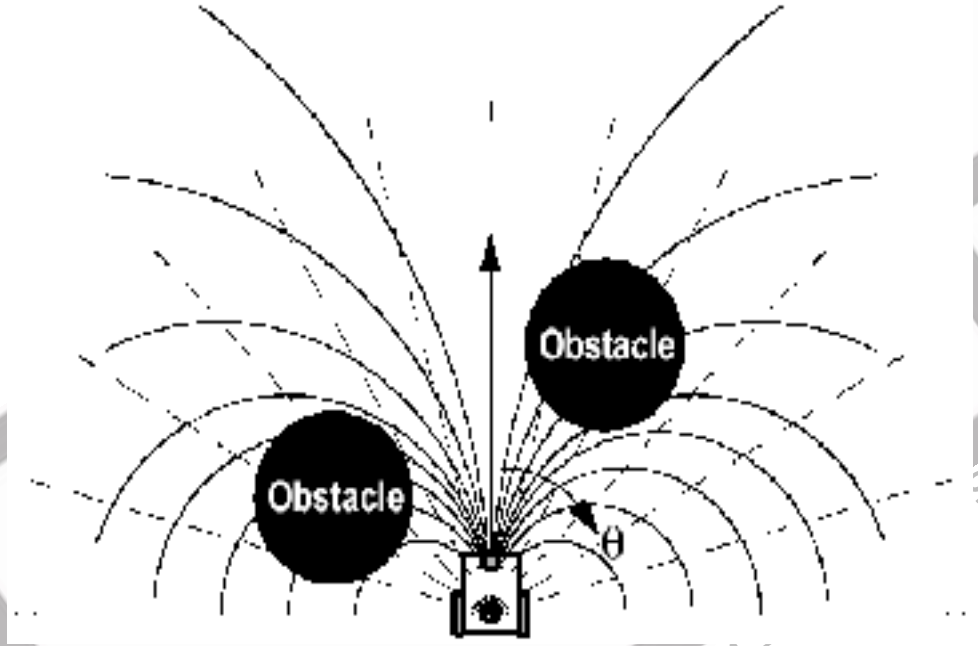
Vector Field Histogram$^+$ - VFH$^+$

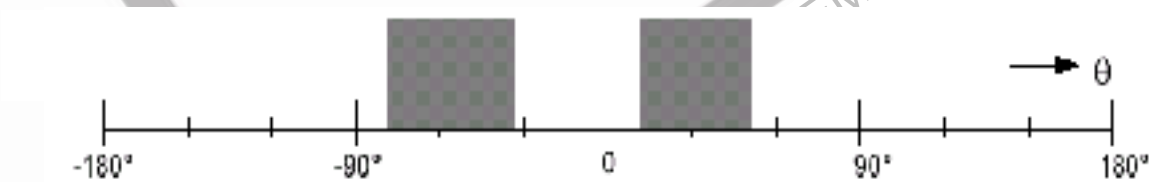
(Ulrich and Borenstein, 1998)

- O ambiente é representado por um *Grid* de Células
  - Cada Célula possui um valor relativo à probabilidade de estar ocupada
  - O robô move-se em arcos ou linhas retas
  - Todas as trajetórias em uma certa região são descartadas se essa possuir direções bloqueadas por obstáculos
    - ✓ A região de direções proibidas é aumentada para evitar que o robô assuma um ângulo de esterçamento impróprio.

Prof. Dr. Marcelo Becker - SEM-EESC-USP

Vector Field Histogram+ - VFH+



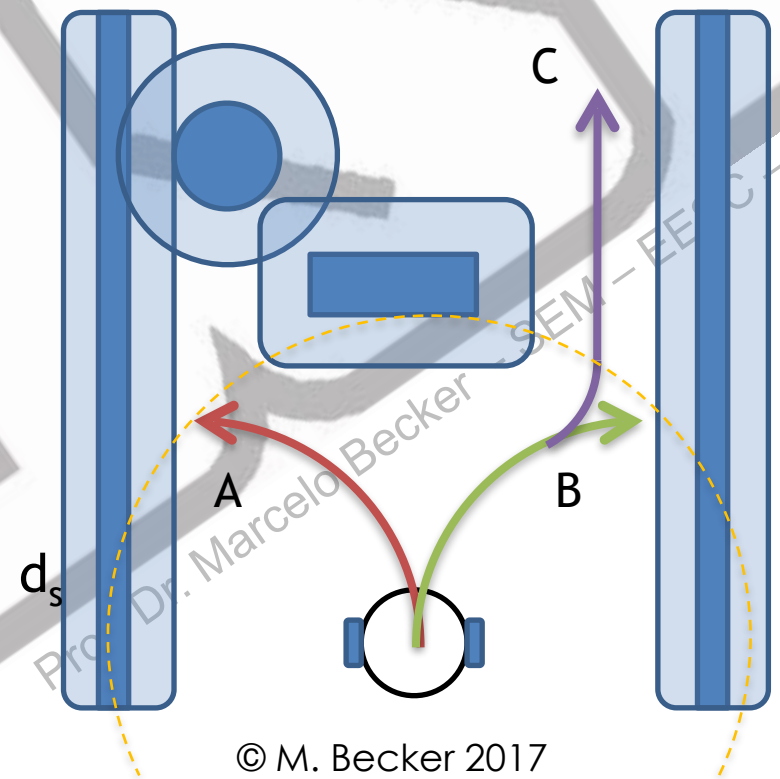


© M. Becker 2017

Vector Field Histogram* - VFH*

(Ulrich and Borenstein, 2000)

- Problema quando se aplica um planejamento meramente local...



d_s: Distância Projetada para o passo

Prof. Dr. Marcelo Becker - SEM – EESC – USP

© M. Becker 2017

Vector Field Histogram* - VFH*

- Evolução do VFH+
  - O ambiente é representado por um Grid de Células.
  - Cada Célula possui um valor relativo à probabilidade de estar ocupada.
  - O robô é modelado como ponto material.
  - O movimento restringe-se a arcos ou linhas retas.

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Vector Field Histogram* - VFH*

- Basicamente:
  - Utiliza-se o VFH+ para selecionar direções candidatas.
  - As direções candidatas são analisadas (verifica-se as conseqüências para n_g passos - *goal depth*).
    - Estrutura de Grafo $\rightarrow$ Algoritmo A* (fç heurística = G)
    - Parâmetros chave:
      - ✓ d_t: Distância Projetada Total
      - ✓ d_s: Distância Projetada para o Passo
      - ✓ n_g: Número de Passos

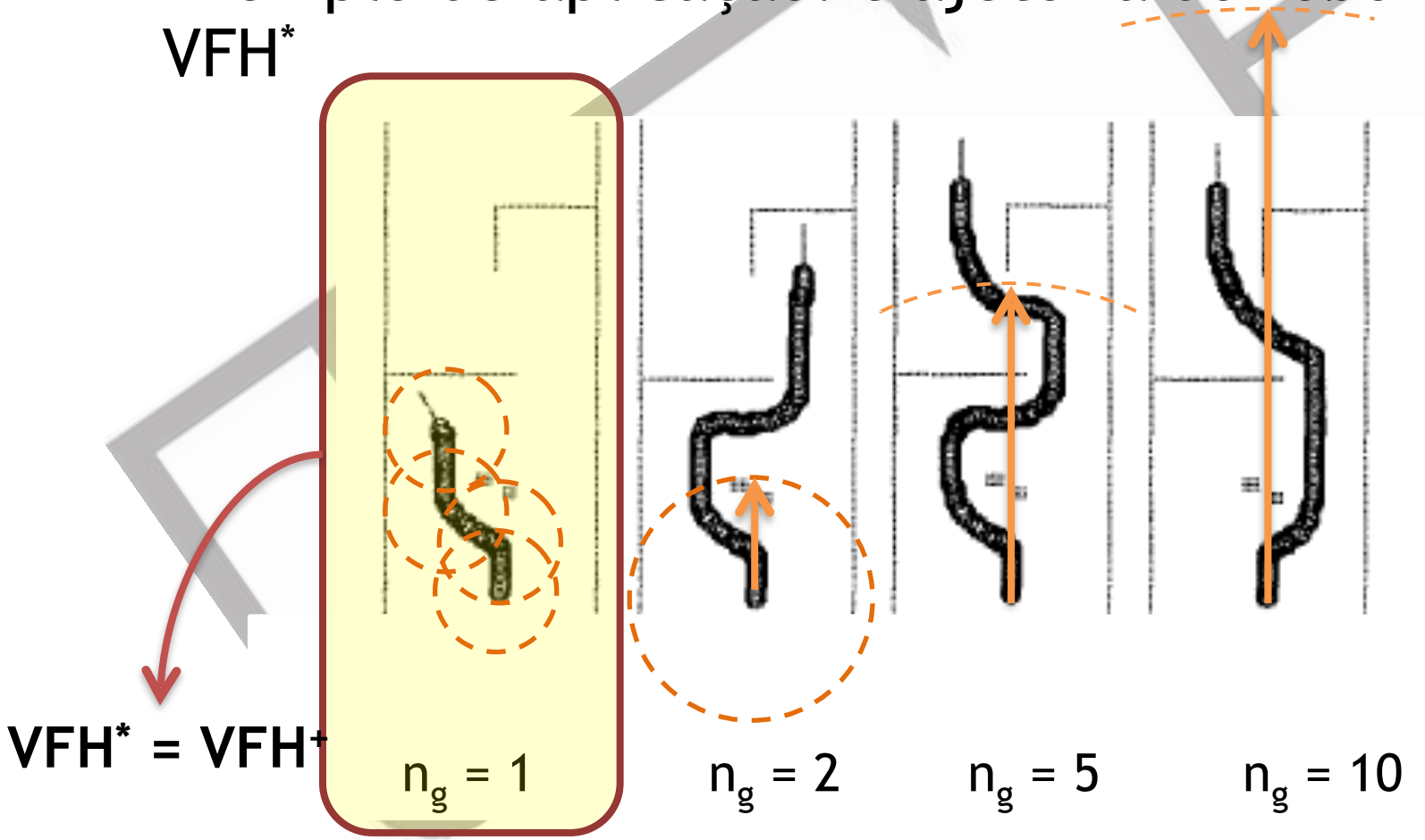
$$d_t = n_g \cdot d_s$$

Prof. Dr. Marcelo Becker - SEM – EESC – USP

© M. Becker 2017

Vector Field Histogram* - VFH*

- Exemplo de aplicação: trajetória do robô com VFH*



© M. Becker 2017

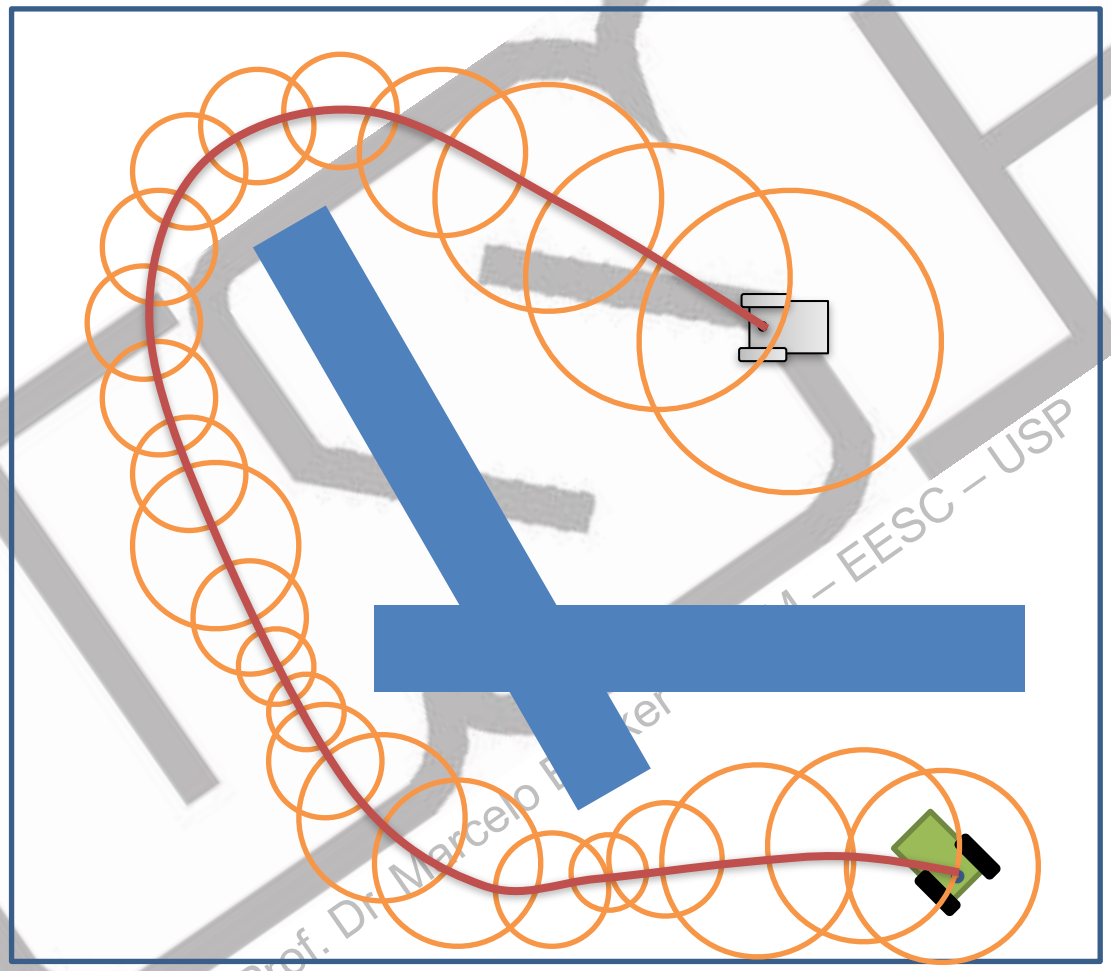
Bubble Band

(Khatib and Quinlan, 1993)

- ***Proposta inicial**: Elastic Bands*
- *Bubble* é o máximo espaço livre que pode ser alcançado sem que haja risco de colisão:
  - É gerado utilizando a distância ao obstáculo e um modelo simplificado do robô.
  - Utiliza-se um grupo de *bubbles* conectadas que ligam a posição inicial e a desejada (*Goal*) para definir a trajetória a ser seguida

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Bubble Band



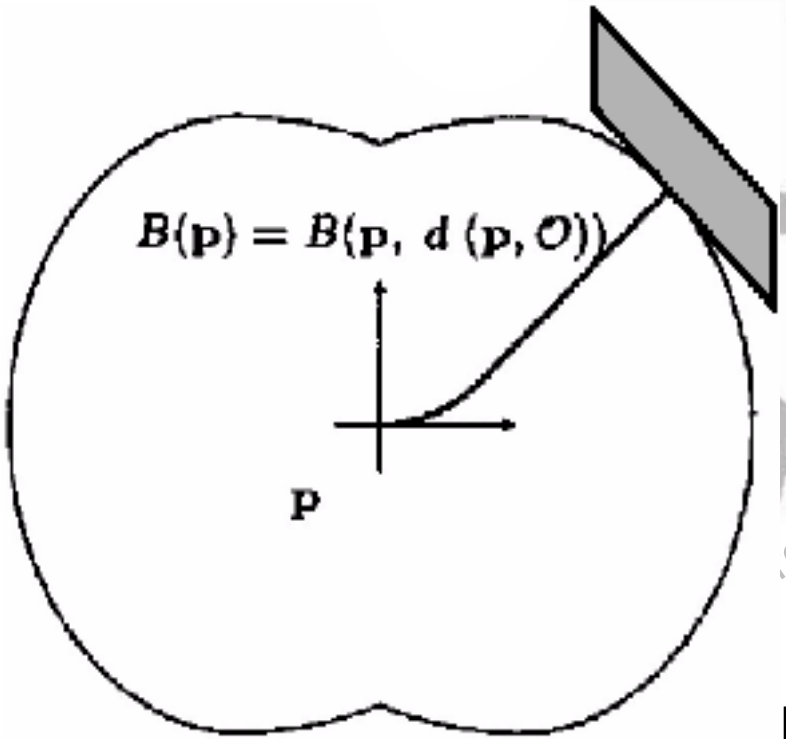
Exemplo de aplicação em Robô Omni-direcional

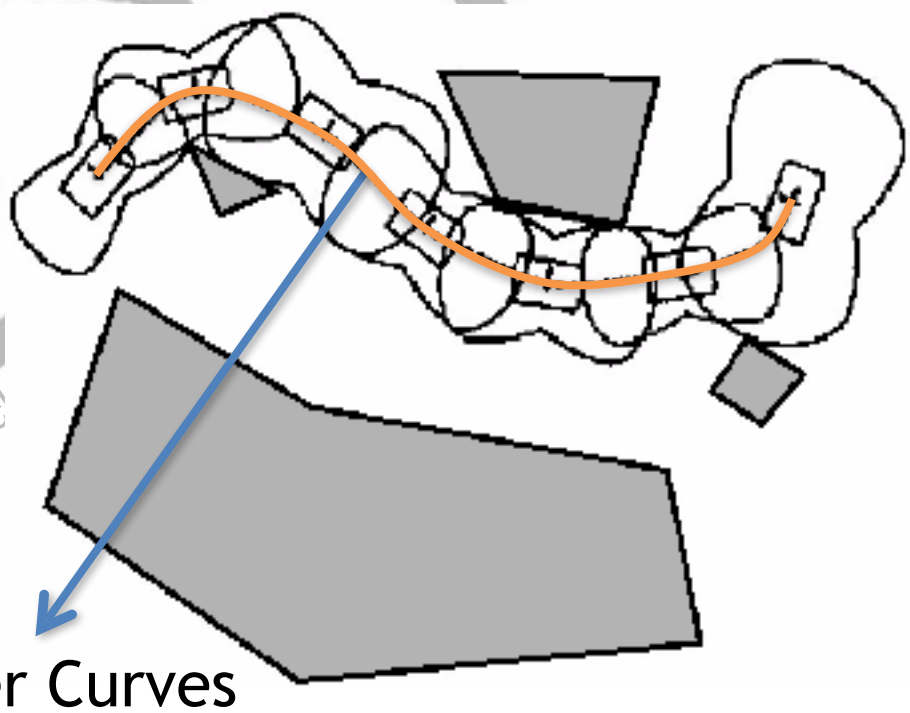
© M. Becker 2017

Bubble Band

(Khatib et al., 1997)

- ***Evolução**: Nonholonomic Bubbles*





© M. Becker 2017

Basic Curvature Velocity

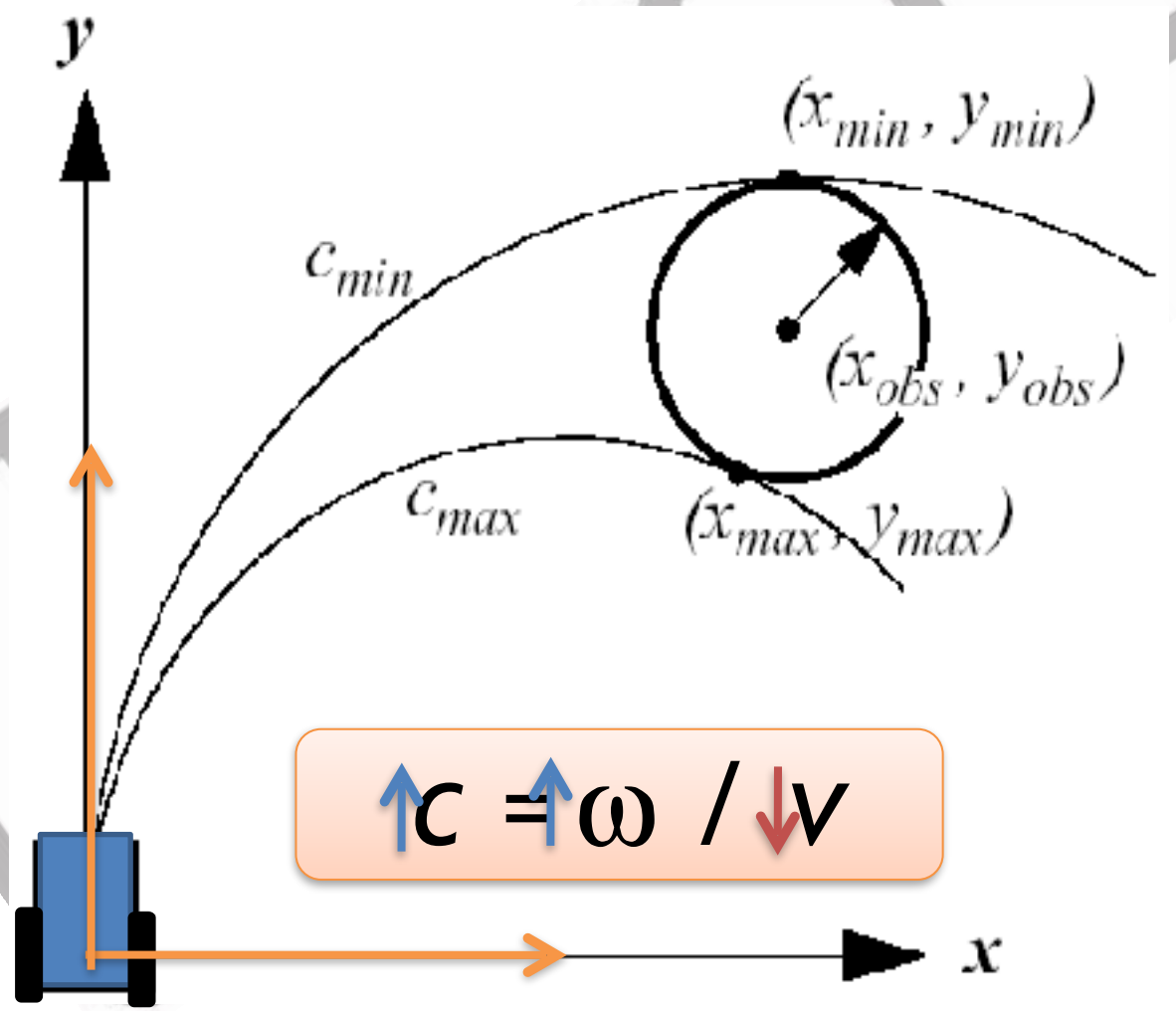
(Simmons, 1996)

- Adiciona-se as restrições físicas do robô e do ambiente no espaço de velocidades do robô (v, ω):
  - O robô desloca-se através de movimentos em forma de arcos ($c = \omega / v$)
  - Restrições de aceleração:
    - $v_{min} < v < v_{max}$ e $\omega_{min} < \omega < \omega_{max}$
  - As restrições devido aos obstáculos são transformadas para o espaço de velocidades
  - Uma Função Objetivo é empregada para selecionar e velocidade ótima

Prof. Dr. Marcelo Becker - SEM - EESC – USP

© M. Becker 2017

Basic Curvature Velocity

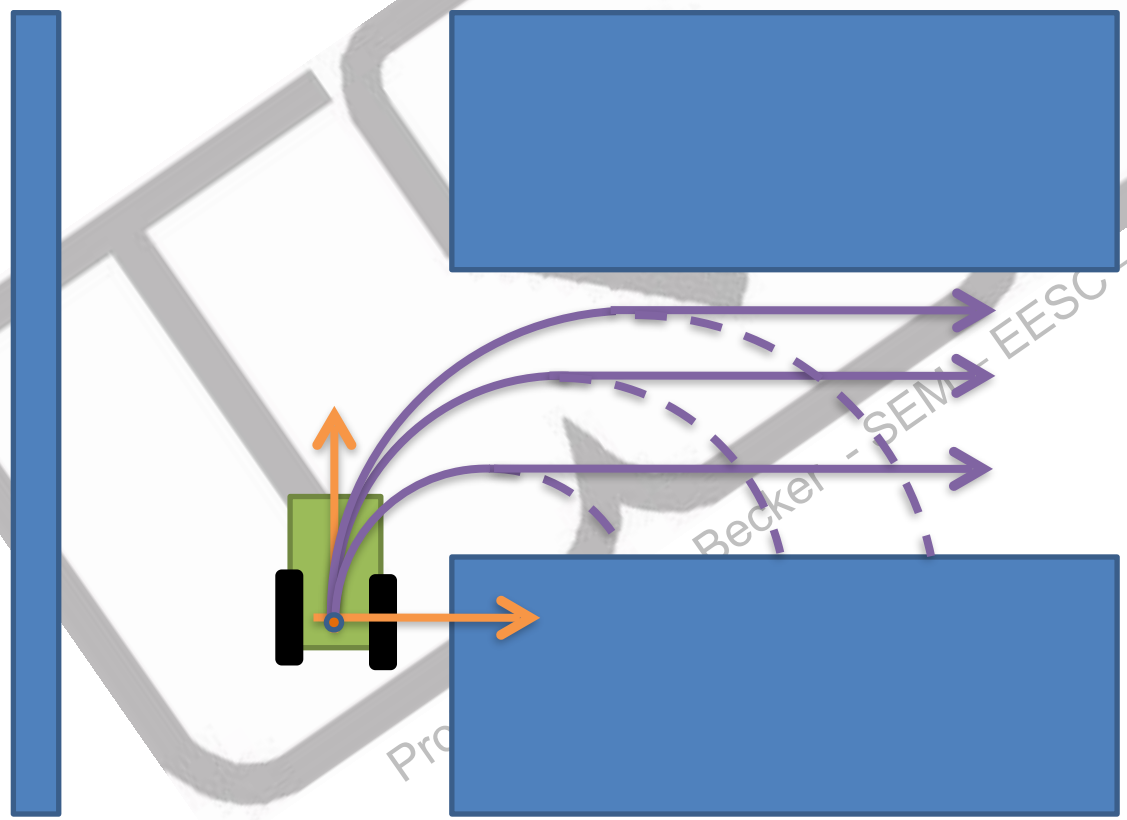


$$\uparrow c = \uparrow \omega \; / \downarrow v$$

© M. Becker 2017

Lane Curvature Velocity

- Problema do *Basic Curvature Velocity*
  - São considerados apenas arcos



© M. Becker 2017

Lane Curvature Velocity

(Ko and Simmons, 1998)

- Melhoria do *Basic Curvature Velocity*
  - Não são considerados apenas arcos, mas tb. faixas de navegação.
  - Os comprimentos e larguras das faixas são calculadas levando-se em consideração os obstáculos mais próximos.
  - A faixa com as melhores características é escolhida utilizando-se uma função objetivo.
- Observação: Produz melhores resultados em áreas estreitas (p.e.: corredores e portas), mas ainda apresenta problemas com mínimos locais.

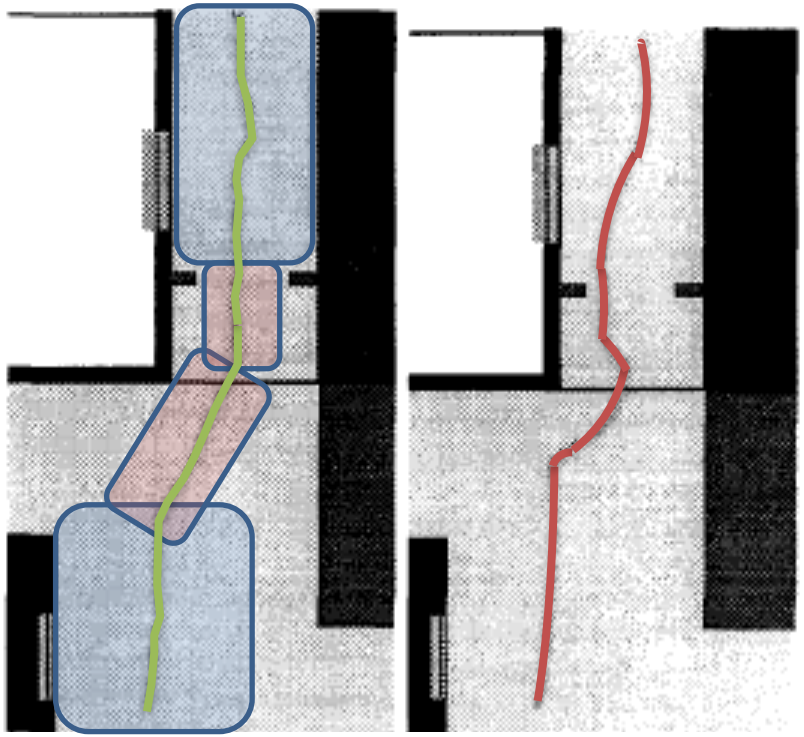
Prof. Dr. Marcelo Becker - SEM - EESC - USP

© M. Becker 2017

Lane Curvature Velocity

- Exemplo de aplicação

Robô Xavier
(CMU)



Lane Curvature Velocity Method

Basic Curvature Velocity Method

© M. Becker 2017

Dynamic Window Approach

(Fox, Burgard, and Thrun, 1997)

- A dinâmica do robô é levada em conta no espaço de busca empregado:
  - Trajetórias Circulares: definidas pelo par (v, ω) que representa as velocidades translacional e rotacional.
  - Velocidades Admissíveis: um par (v, ω) é considerado admissível, se o robô é capaz de parar antes que atinja o obstáculo mais próximo na trajetória circular.
  - *Dynamic window*: restringe as velocidades admissíveis a aquelas que podem ser atingidas tendo em vista o tempo de resposta do robô e os seus limites de aceleração

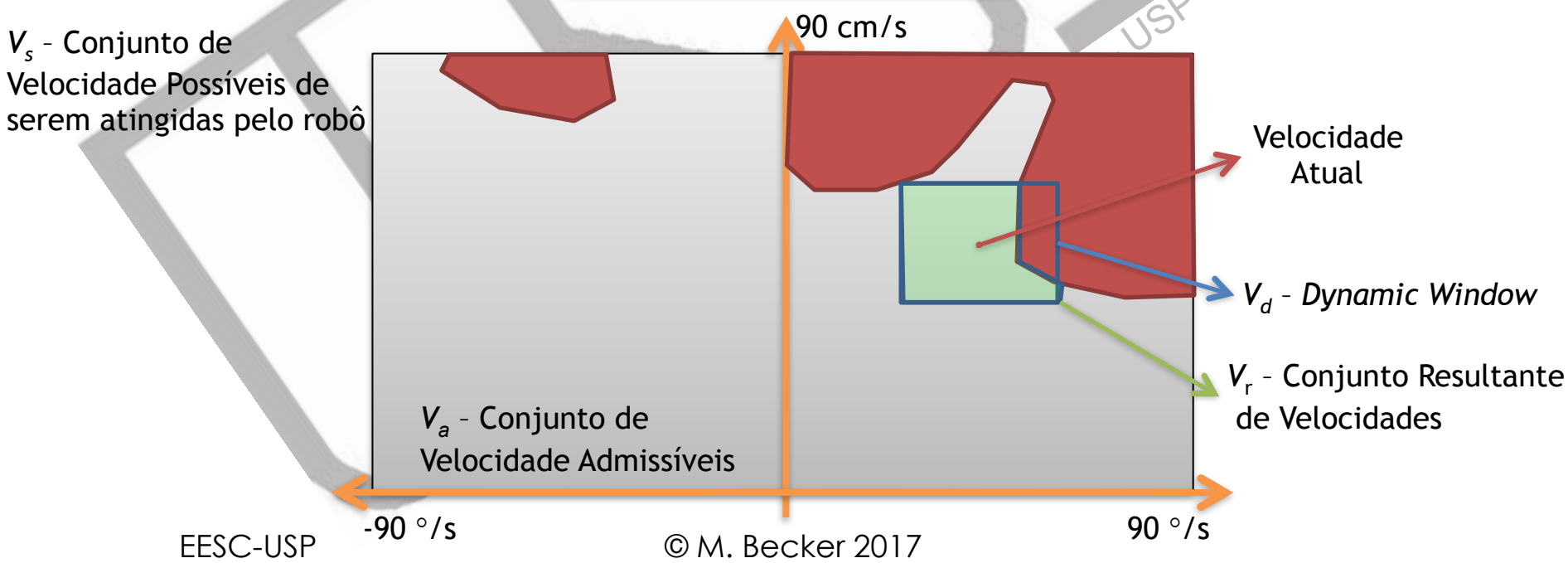
$$V_d = \left\{ (v,\omega) \middle| v \in \left[v_a - \dot{v} \cdot t, v_a + \dot{v} \cdot t \right] \wedge \omega \in \left[\omega_a - \dot{\omega} \cdot t, \omega_a + \dot{\omega} \cdot t \right] \right\}$$

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Dynamic Window Approach

- Espaço de Buscas resultante:
  - A área V_r (Conjunto Resultante de Velocidades) é definida como a intersecção das demais áreas:

$$V_r = V_s \cap V_a \cap V_d$$



© M. Becker 2017

Dynamic Window Approach

- Maximizando a função objetivo:
  - Para levar em consideração no critério de busca de velocidade desejada a direção do *Goal*, calcula-se o máximo da função objetivo, $G(v, \omega)$, em V_r:

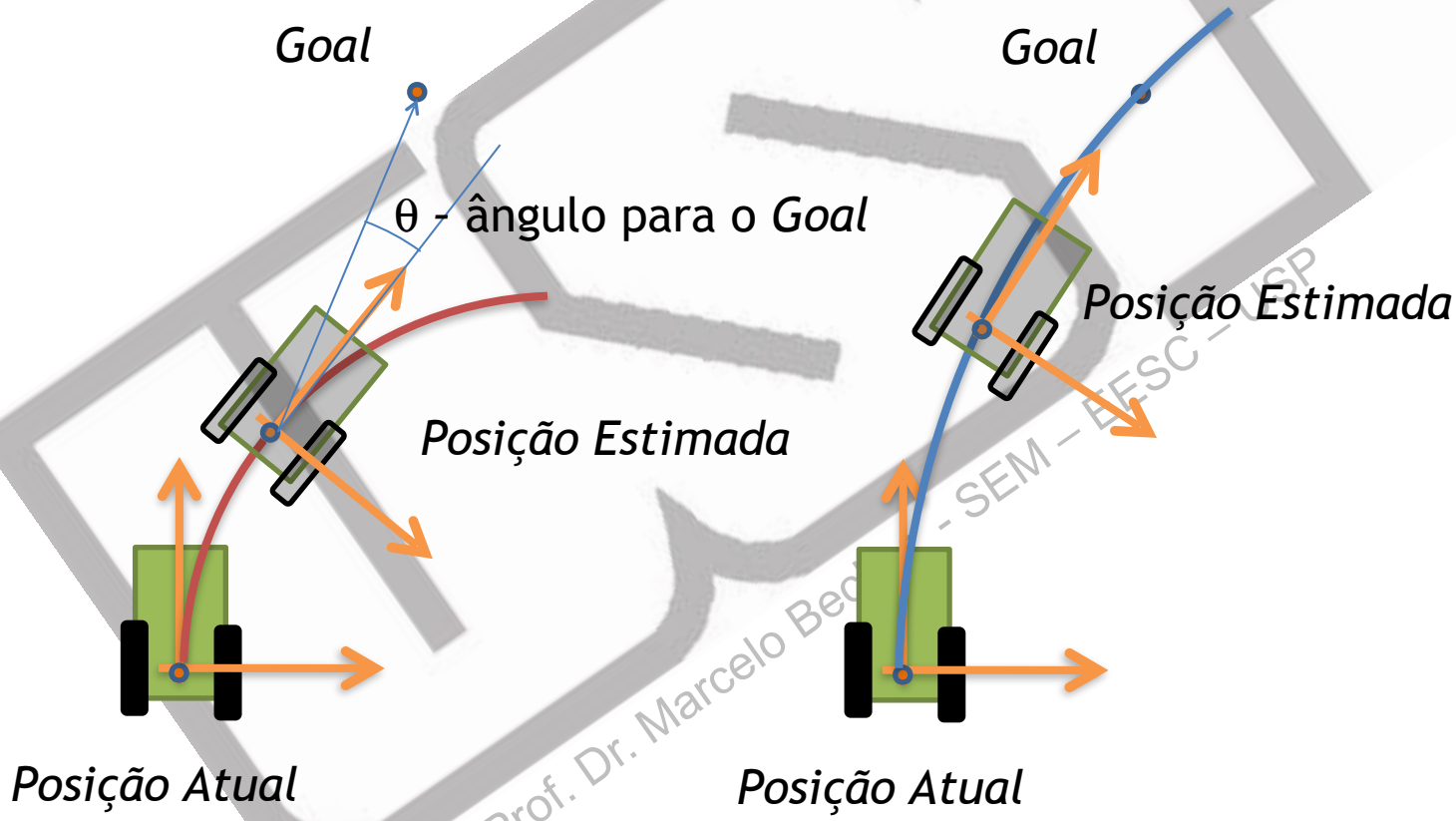
$$G(v,\omega) = \sigma\left(\alpha \cdot heading(v,\omega) + \beta \cdot dist(v,\omega) + \gamma \cdot velocity(v,\omega)\right)$$

heading = Medida do progresso do ângulo de orientação para atingir o *Goal*
dist = Distância ao obstáculo mais próximo na trajetória
velocity = Velocidade do robô, considerando-se movimentos rápidos

Prof. Dr. Marcelo Becker - SEM - USP

Dynamic Window Approach

A medida do Progresso é fazer com que $\theta \rightarrow 0$

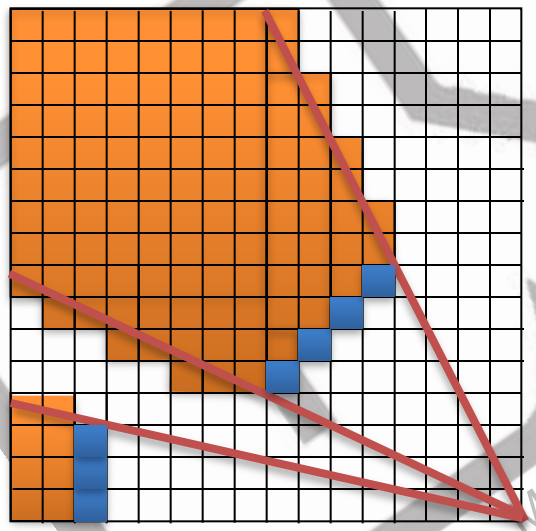


Prof. Dr. Marcelo Becker - SEM – EESC – USP

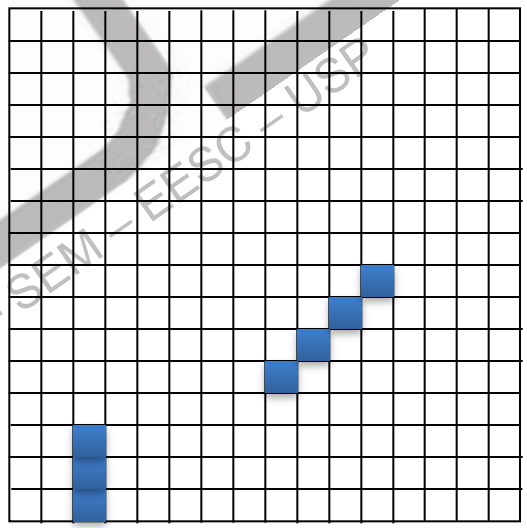
© M. Becker 2017

Occupancy Grid

- Representação do ambiente entorno do robô por um *Grid* de Células (2D):
  - Proposta por Elfes (1989)
  - Evolução: *Time Stamp Map* - TSMs (Fiorini and Shiller, 1993)



Occupancy Grid Representation



Time Stamp Map

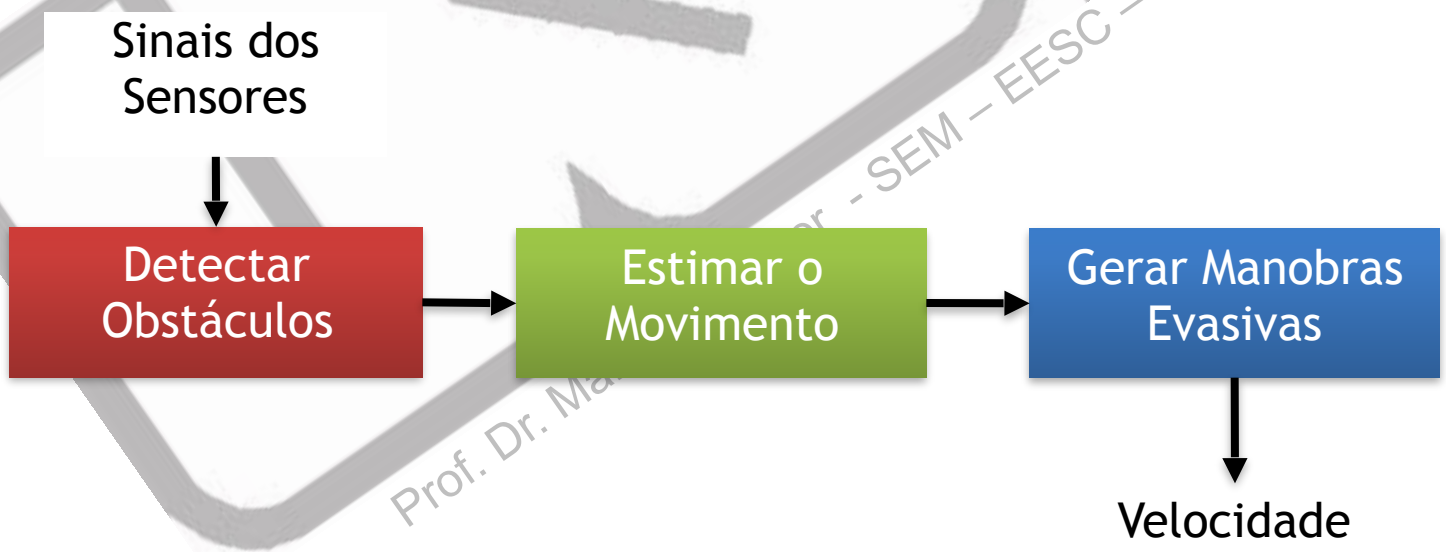
Prof. Dr. Marcelo Becker - SEM - EESC - USP

© M. Becker 2017

Velocity Obstacle

(Fiorioni and Shiller, 1993)

- Consiste basicamente em 3 algoritmos:
  - ✓ Detecção e Classificação de Obstáculos
  - ✓ Monitoramento
  - ✓ Geração de Manobras Evasivas

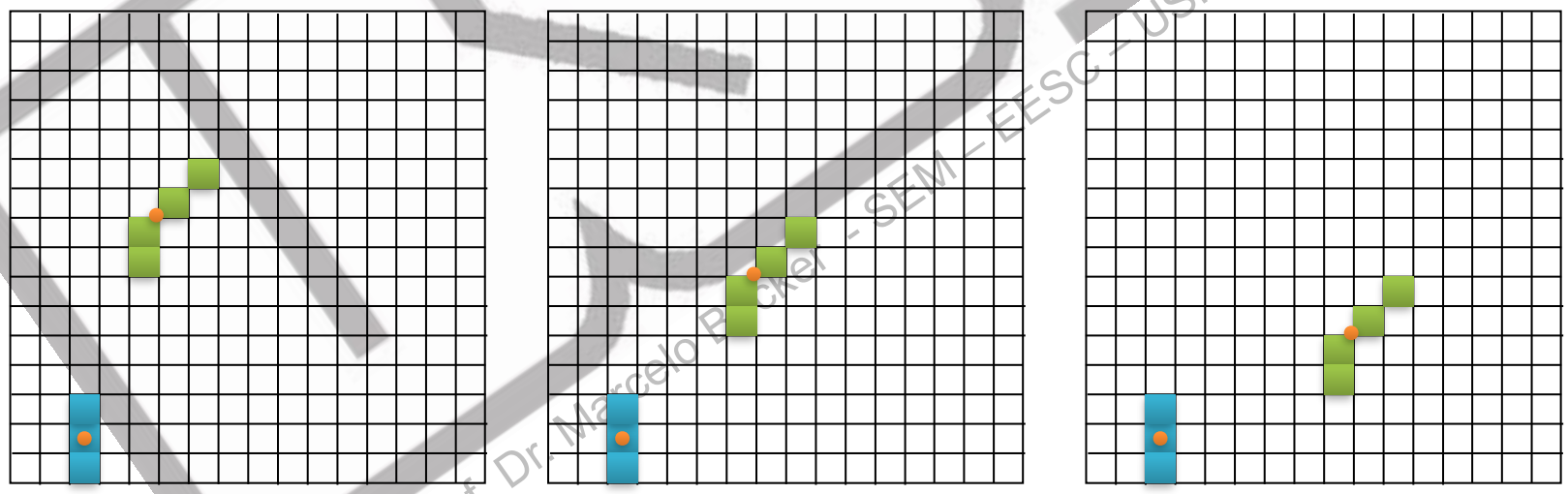


Prof. Dr. Ma... - SEM – EESC – USP

© M. Becker 2017

Velocity Obstacle

- Na implementação utiliza o *Time Stamp Map*:
  - ✓ Une células próximas em grupos.
  - ✓ Calcula os CGs dos grupos
  - ✓ Alterações na posição dos CGs indicam a possibilidade de movimento



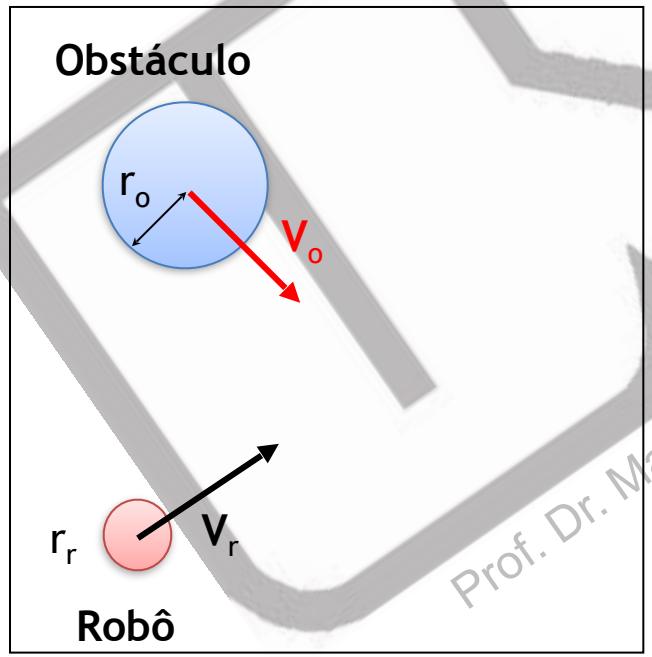
t-2 ***t-1*** ***t***

Prof. Dr. Marcelo Becker - SEM – EESC – USP

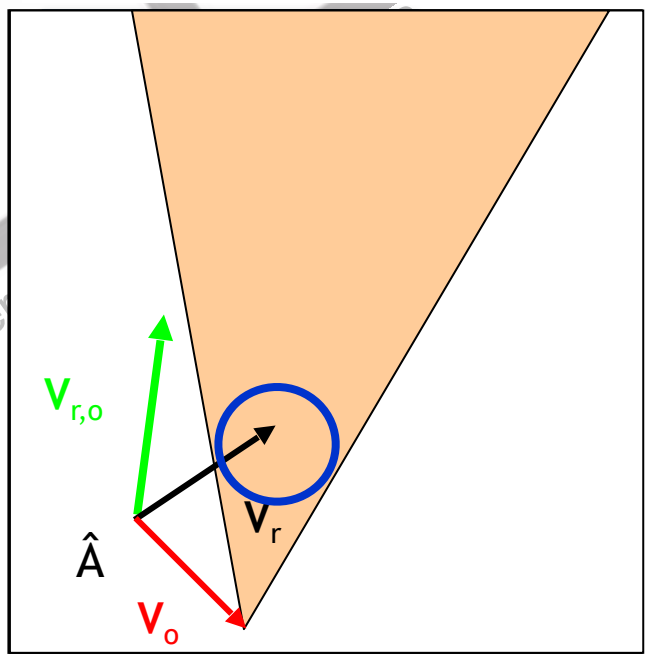
© M. Becker 2017

Velocity Obstacle

- Gerar Manobras evasivas onde o robô margeia os obstáculos
- Baseia-se em um “Cone de Colisão” entre robô e obstáculos



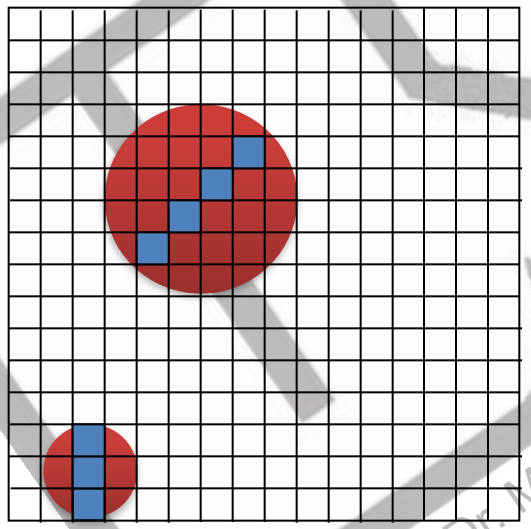
Cone *velocity obstacle*



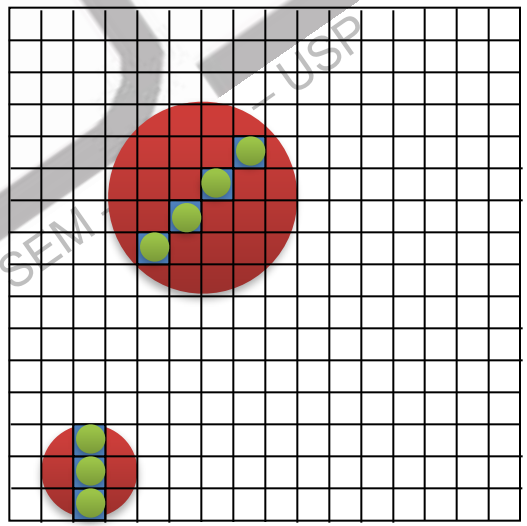
© M. Becker 2017

Velocity Obstacle

- Cuidados com circunscrição dos obstáculos:
  - ✓ Empregar as células ocupadas de modo isolado para considerar uma maior área para manobras...



Conjunto completo



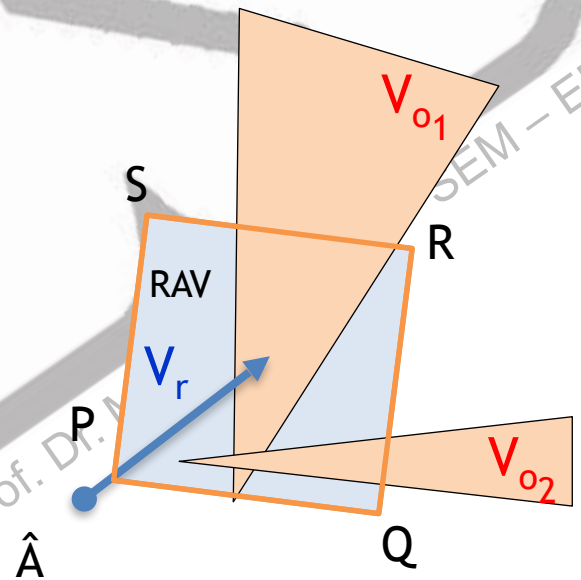
Células isoladas

Prof. Dr. Marcelo Becker - SEM - EESC - USP

© M. Becker 2017

Velocity Obstacle

- Determinação das possíveis velocidades:
  - ✓ Os limites dinâmicos do robô são empregados para gerar os vértices PQRS.
  - ✓ Obtém-se o conjunto de velocidades livres de colisão (*Reachable Avoidance Velocities* - RAV).

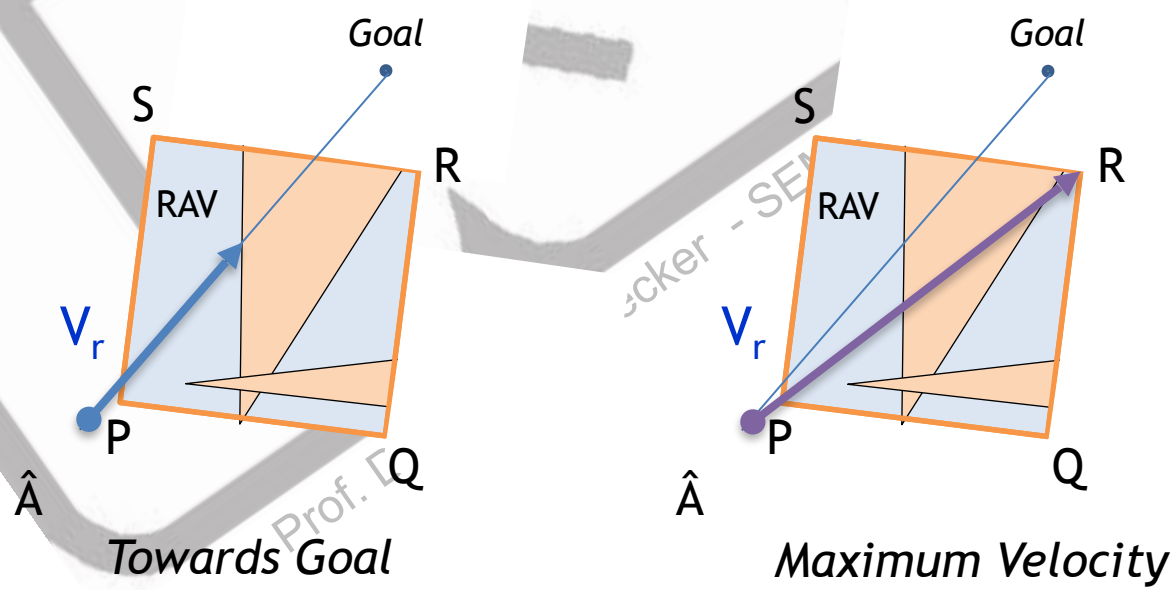


Prof. Dr. M SEM – EESC – USP

© M. Becker 2017

Velocity Obstacle

- Estratégias para a seleção do vetor velocidade baseadas em heurística:
  - ✓ Ir na direção da posição desejada.
  - ✓ Mover-se com a maior velocidade possível.

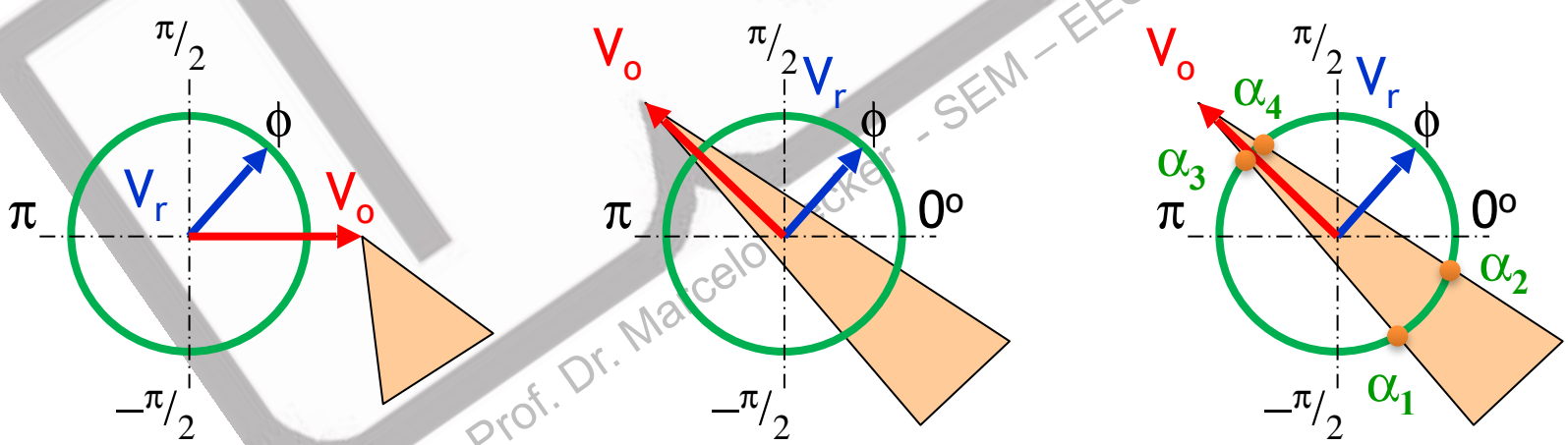


© M. Becker 2017

Velocity Obstacle Modificado

(Becker, 2000)

- Posicionamento do Vetor Velocidade
  - ✓ Velocidade de Cruzeiro
  - ✓ Limites de Aceleração e Velocidade do robô
  - ✓ Parada de emergência

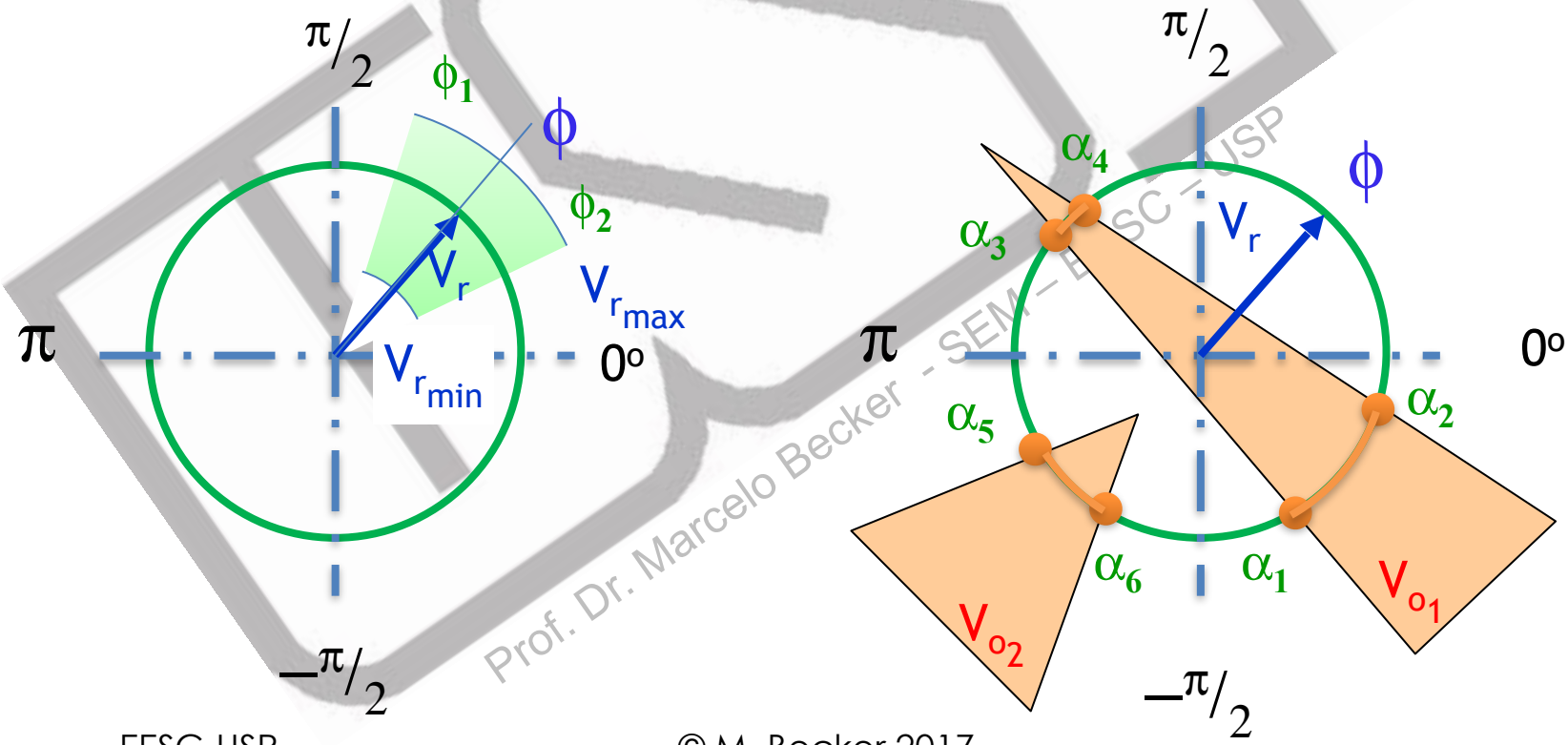


Prof. Dr. Marcelo Becker - SEM – EESC – USP

© M. Becker 2017

Velocity Obstacle Modificado

- Limites dinâmicos do robô e presença de múltiplos obstáculos...

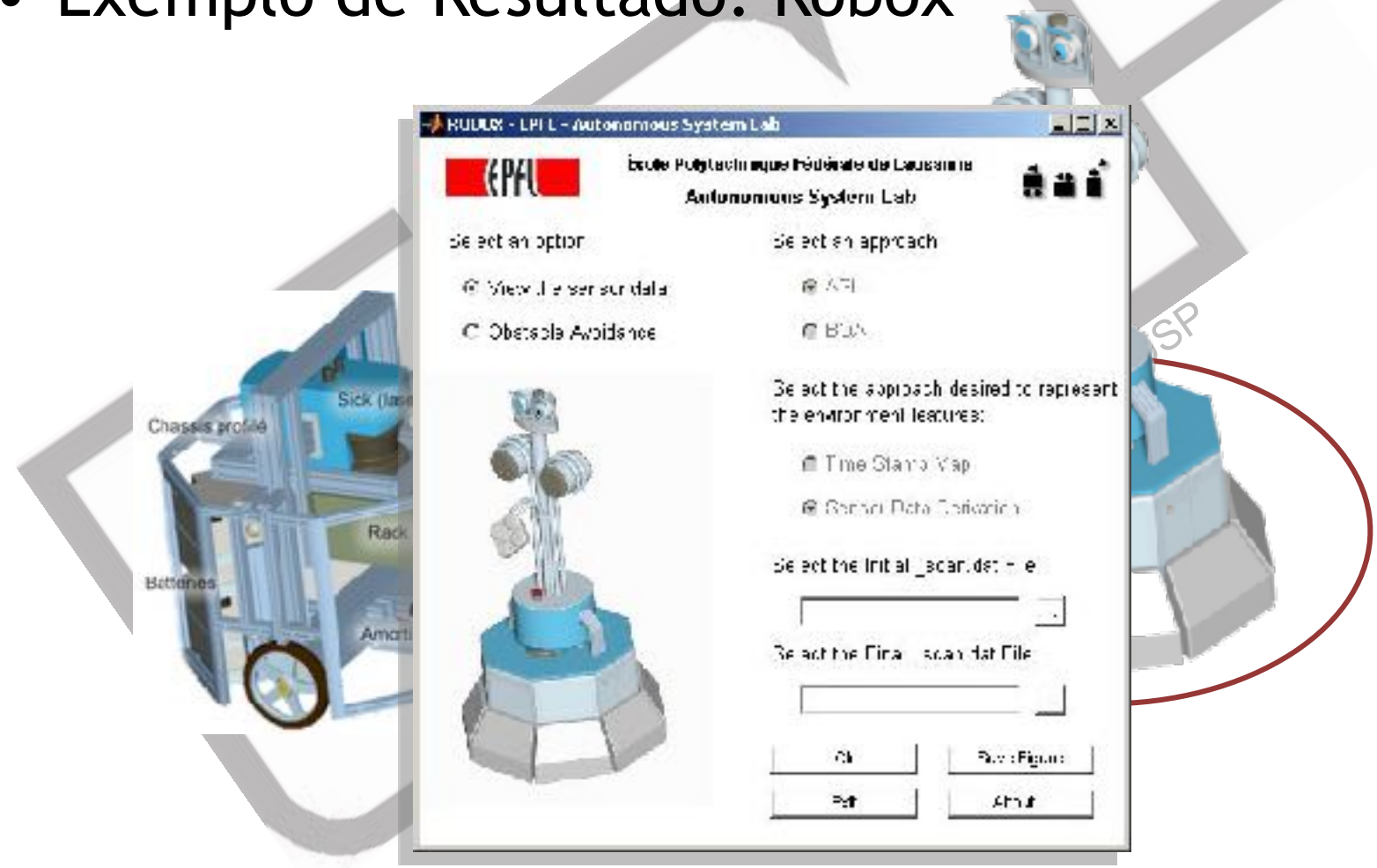


Prof. Dr. Marcelo Becker - SEM – EESC – USP

© M. Becker 2017

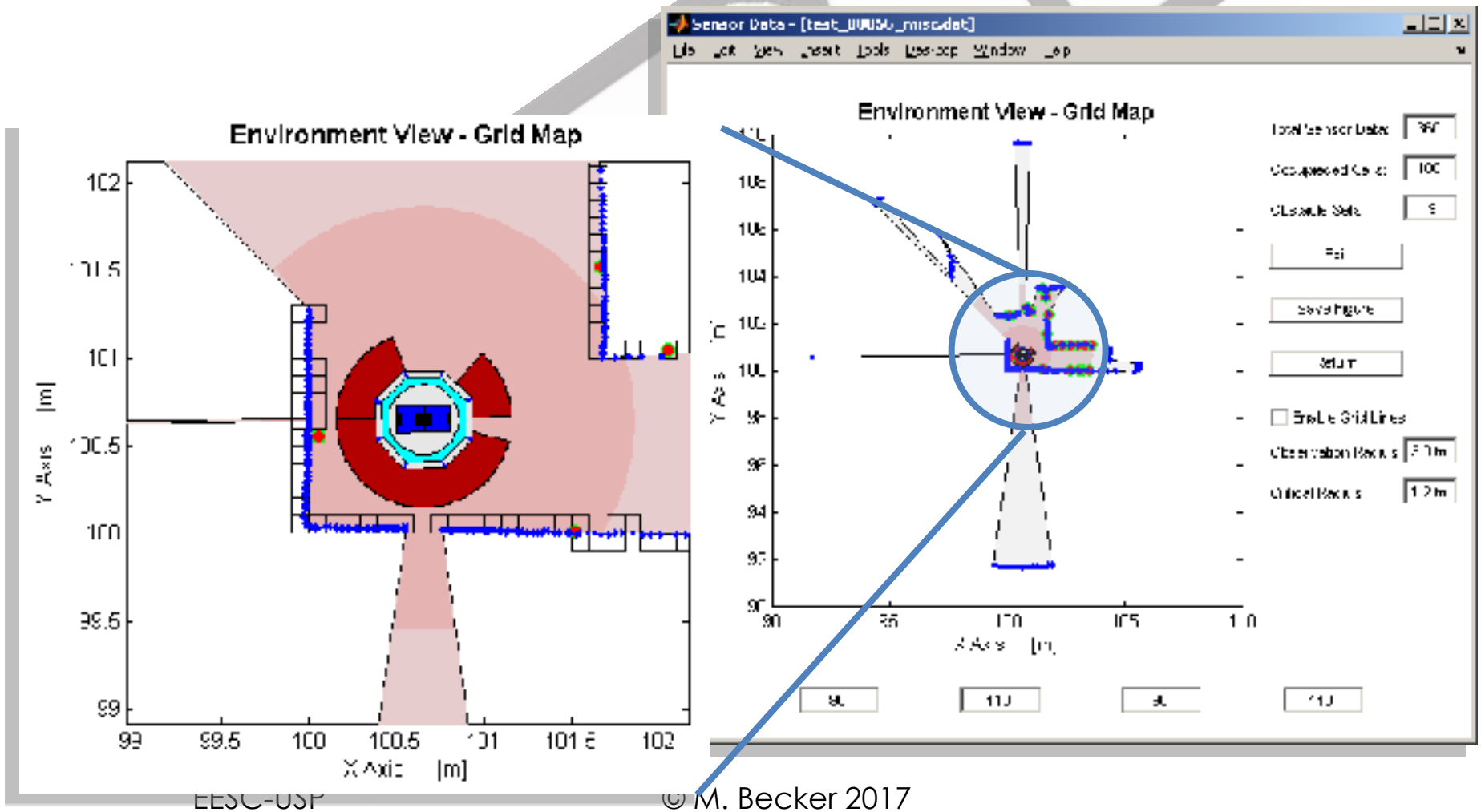
Velocity Obstacle Modificado

- Exemplo de Resultado: Robox



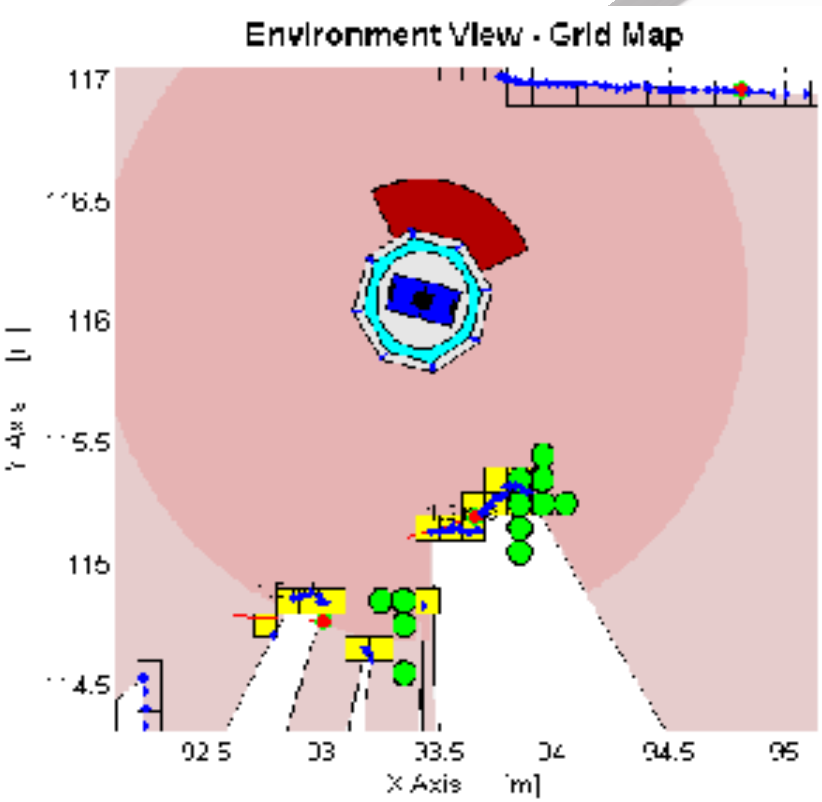
© M. Becker 2017

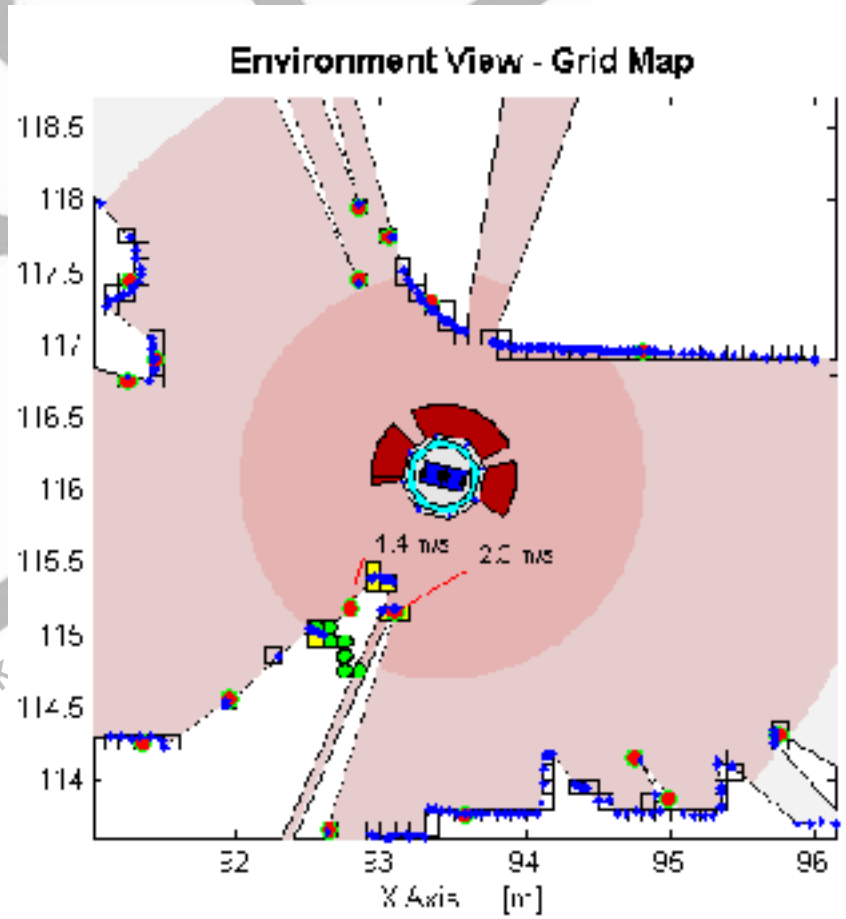
Velocity Obstacle Modificado



© M. Becker 2017

Velocity Obstacle Modificado





© M. Becker 2017

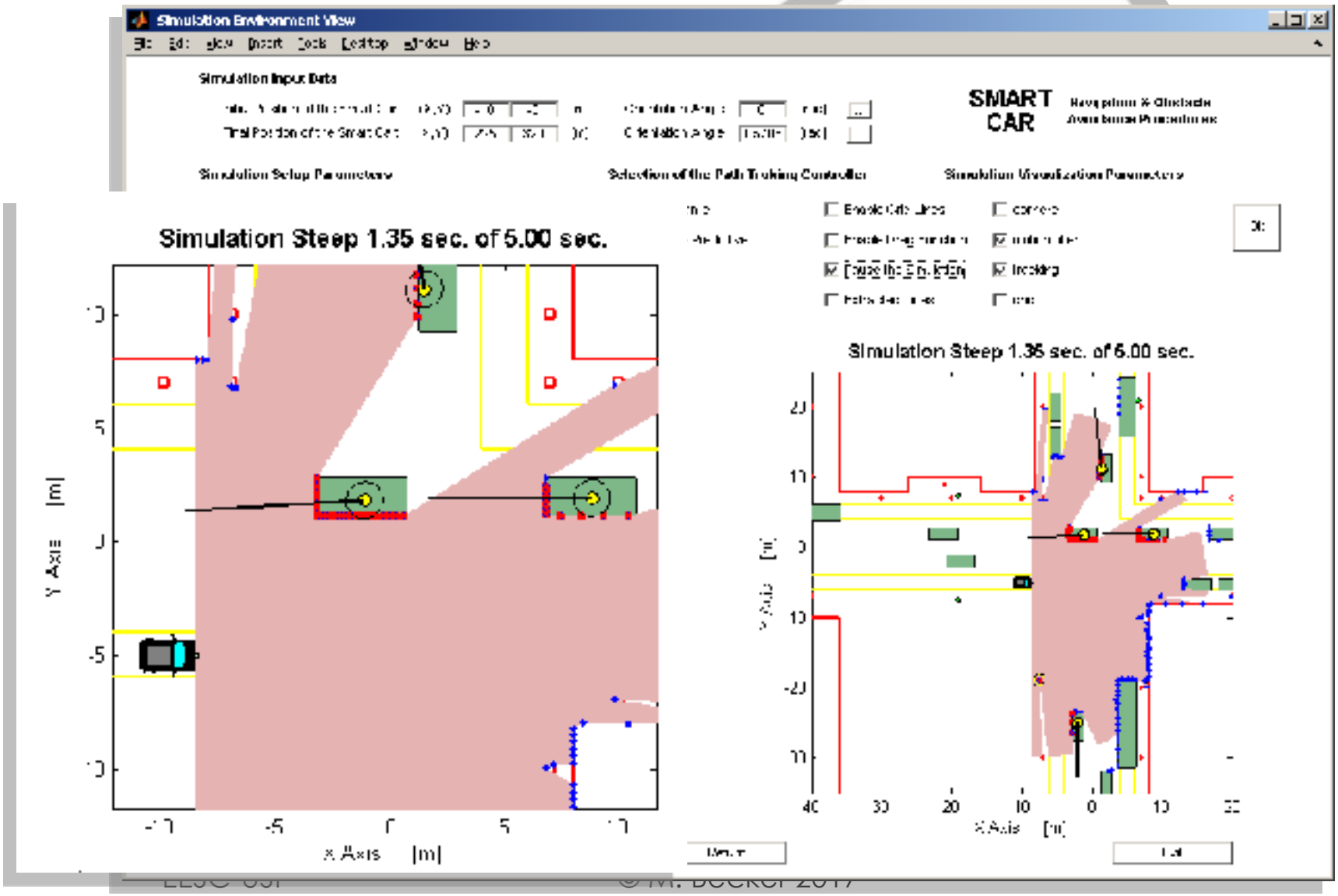
Probabilistic Occupancy Grid

(Becker et al., 2009)

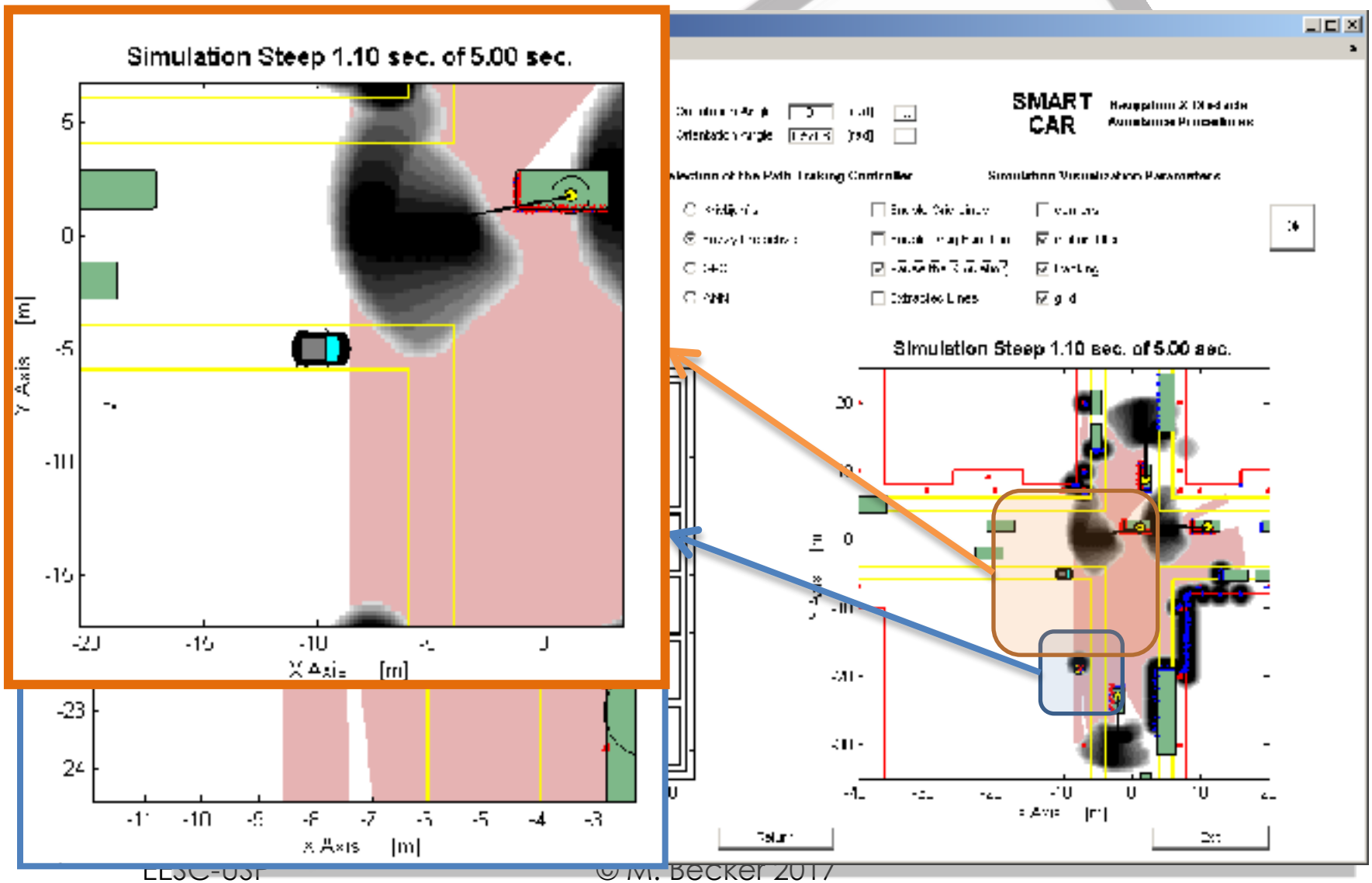
- Representação Probabilística do ambiente entorno do robô por um *Grid* de Células (2D):
  - Auxilia no planejamento de trajetórias para uma janela de tempo
  - Emprega:
    - *Trackers*
    - Filtros Probabilísticos (KF, EKF, UKF, etc.)

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Probabilistic Occupancy Grid



Probabilistic Occupancy Grid



© M. Becker 2017

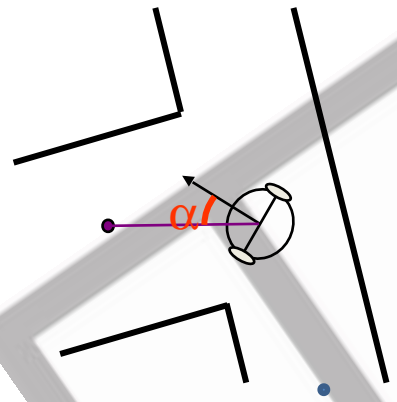
ASL Approach

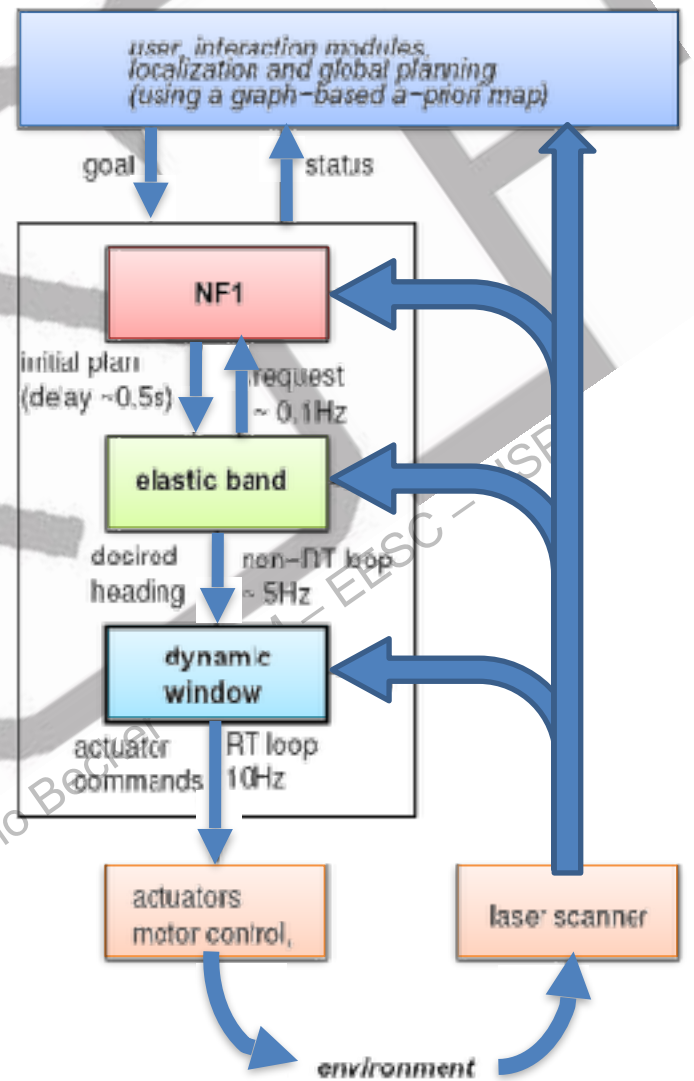
(Philippsen and Siegwart, 2003)

- *Dynamic Window* combinado com uma estratégia global de planejamento de trajetórias
  - O Planejamento Global é feito *a priori*.
  - A trajetória inicial é adaptada se forem encontrados obstáculos obstruindo-a.
  - *Dynamic Window* considera a forma do robô.
  - Na seleção da velocidade, utiliza-se o critério de máxima velocidade.

Prof. Dr. Marcelo Becker - SEM - EESC - USP

ASL Approach

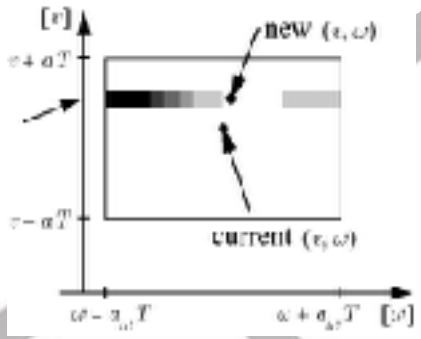




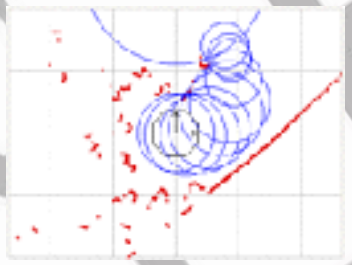
Prof. Dr. Marcelo Becker – EESC – USP

© M. Becker 2017

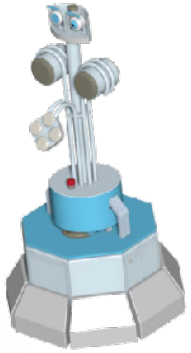
ASL Approach



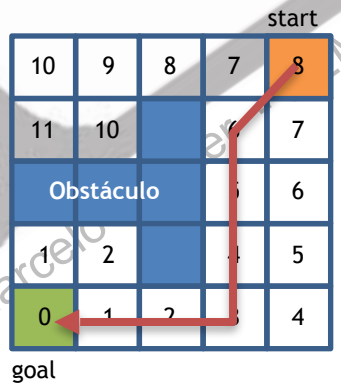
Dynamic Window



Elastic Band



EXPO 2002

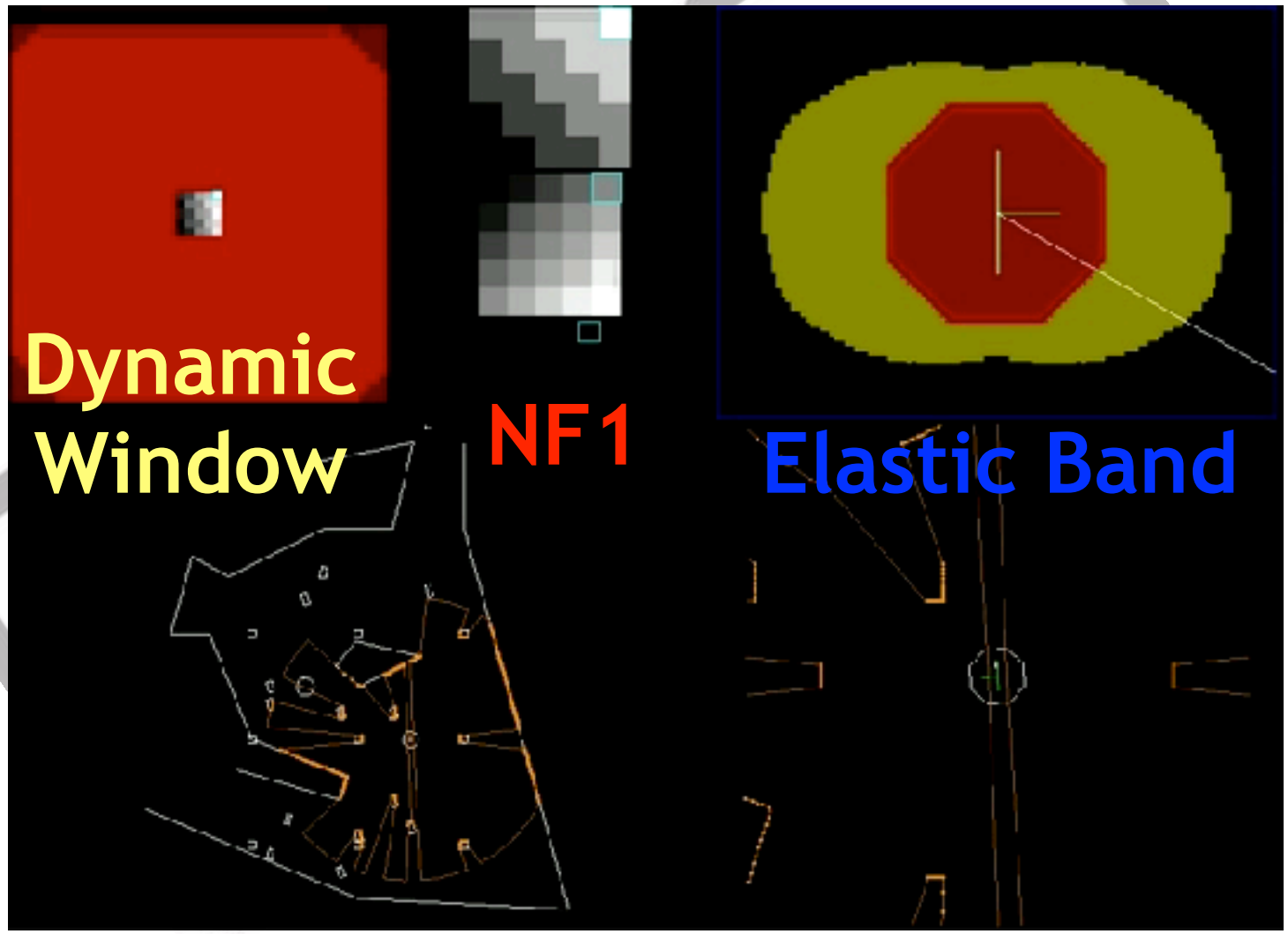


NF1

Prof. Dr. Marcelo Becker – SEM – EESC – USP

© M. Becker 2017

ASL Approach



© M. Becker 2017

Outras Técnicas

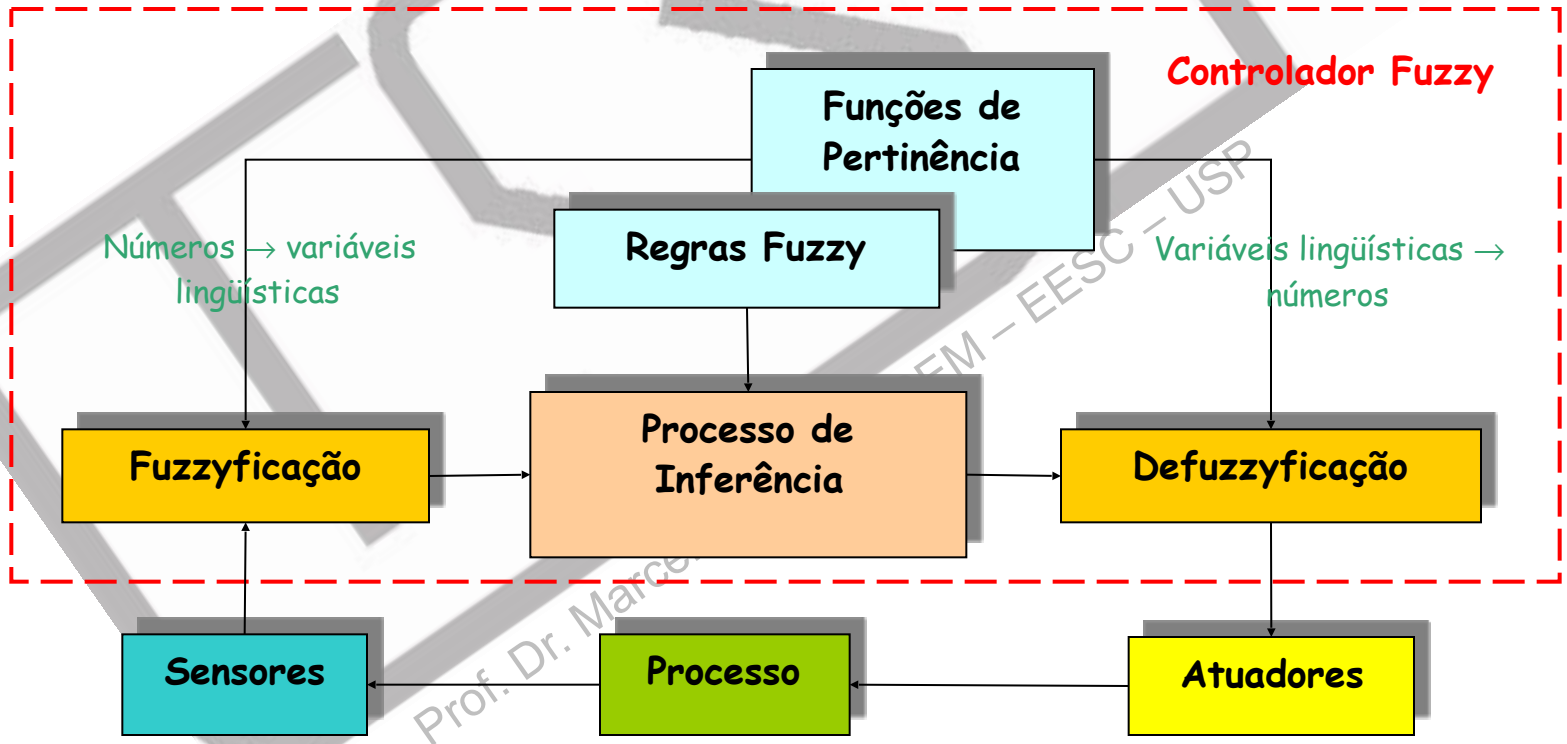
- Baseadas em Comportamento:
  - Dificilmente produzem trajetórias ótimas.
  - Nem sempre atingem o objetivo (*Goal*).

- *Fuzzy*, *Neuro-Fuzzy*:
  - Necessidade de aprendizagem.
  - Difíceis de serem generalizadas.

Prof. Dr. Marcelo Becker - SEM – EESC – USP

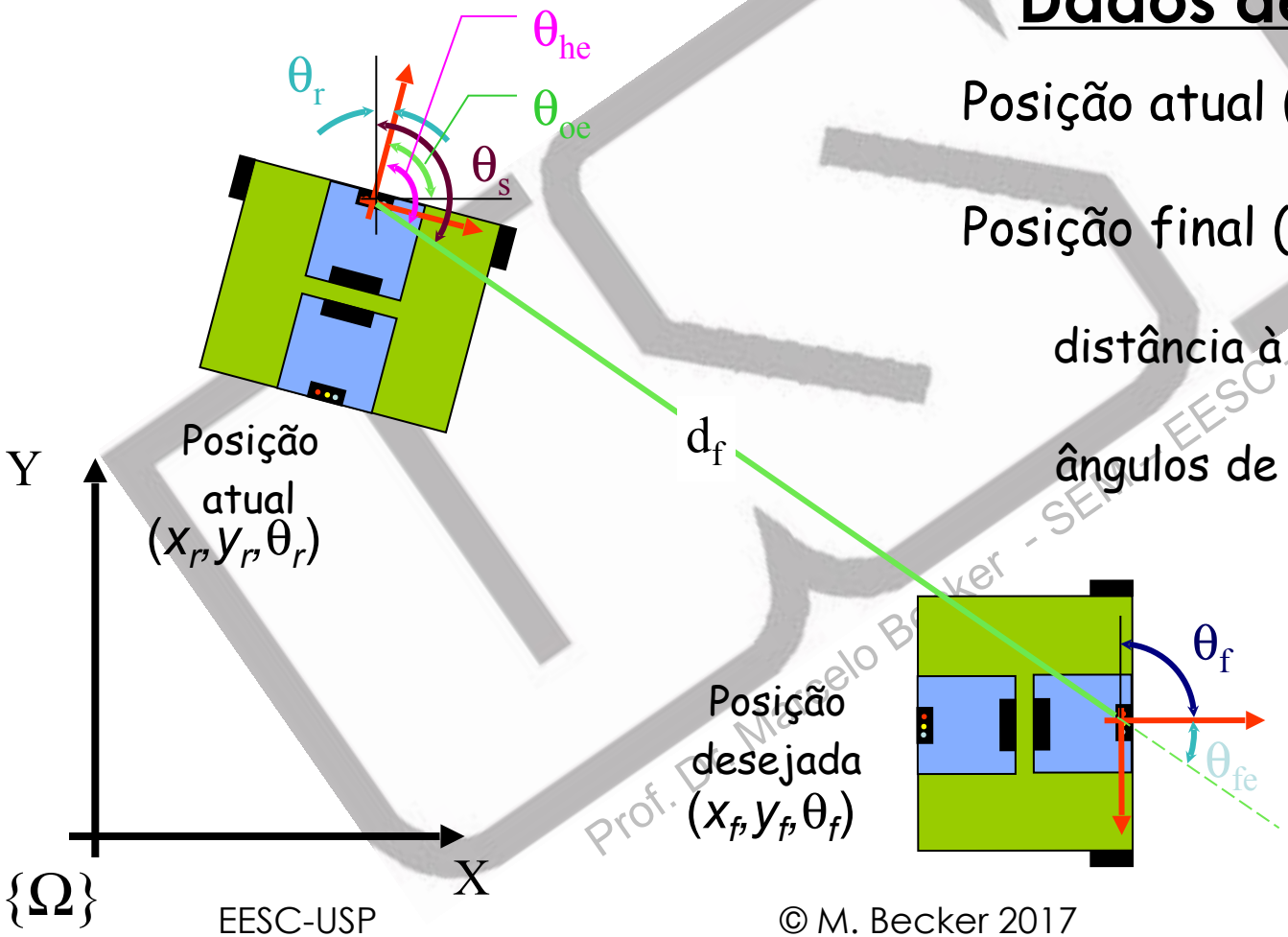
Controlador Fuzzy de Navegação

- Vantagem: robusto na presença de incertezas



Prof. Dr. Marcelo Becker – SEM – EESC – USP

Controlador Fuzzy de Navegação



Dados de entrada

Posição atual (x_r, y_r, θ_r)

Posição final (x_f, y_f, θ_f)

distância à posição desejada (d_f)

ângulos de erro $(\theta_{fe}, \theta_{oe}, \theta_{he})$

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Controlador Fuzzy de Navegação

- Banco *fuzzy* de regras
  - 4 variáveis lingüísticas para $d_f \rightarrow$ 4 matrizes
  - cada uma 15 x 9 = 135 regras, no total: <u>540 regras</u>!
  - posição final $\rightarrow (x_f, y_f, \theta_f)$

d_f (ZE)

θ_{oe} \ θ_{fe}	PA	PB	PP	PL	PM	PS	PZ	ZE	NZ	NS	NM	NL	NP	NB	NA
PA	NB	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	NB
PP	ZE	ZE	ZE	ZE	ZE	ZE	NM	NB	NB	NB	NB	ZE	ZE	ZE	ZE
PM	PS	ZE	ZE	PS	PS	PS	ZE	NS	NM	NM	NB	NM	ZE	ZE	PS
PS	PM	ZE	ZE	PM	PS	PS	ZE	NS	ZE	NS	NM	NB	NM	ZE	PM
ZE	PB	ZE	PB	PM	PS	PS	ZE	ZE	ZE	NS	NS	NM	NB	ZE	PB
NS	PB	ZE	PB	PB	PM	PS	ZE	PS	ZE	NS	NS	NM	ZE	ZE	PB
NM	PB	ZE	ZE	PB	PB	PM	PM	PM	ZE	NS	NS	NS	ZE	ZE	PB
NP	PB	ZE	ZE	ZE	ZE	ZE	ZE	PB	ZE	ZE	ZE	ZE	ZE	ZE	PB
NA	PB	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	PB

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Desvio de Obstáculos

- Controle Reflexivo
  - Similar aos reflexos humanos
- Controle Reativo
  - Similar ao modo humano de tomar decisões
  - Mapeamento
  - Planejamento de Trajetórias

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Desvio de Obstáculos

Controle Reflexivo

- Reage baseado em dados de sensores
  - Ultrassom (sonares)
  - Lasers
  - Infra-vermelho
- Obstáculo → Alterar Direção
- Não emprega Mapa → Baixo uso de Memória
- Rápido
- Reage a obstáculos dinâmicos
- Não é muito Inteligente

Prof. Dr. Marcelo Becker – SEM – EESC – USP

Desvio de Obstáculos

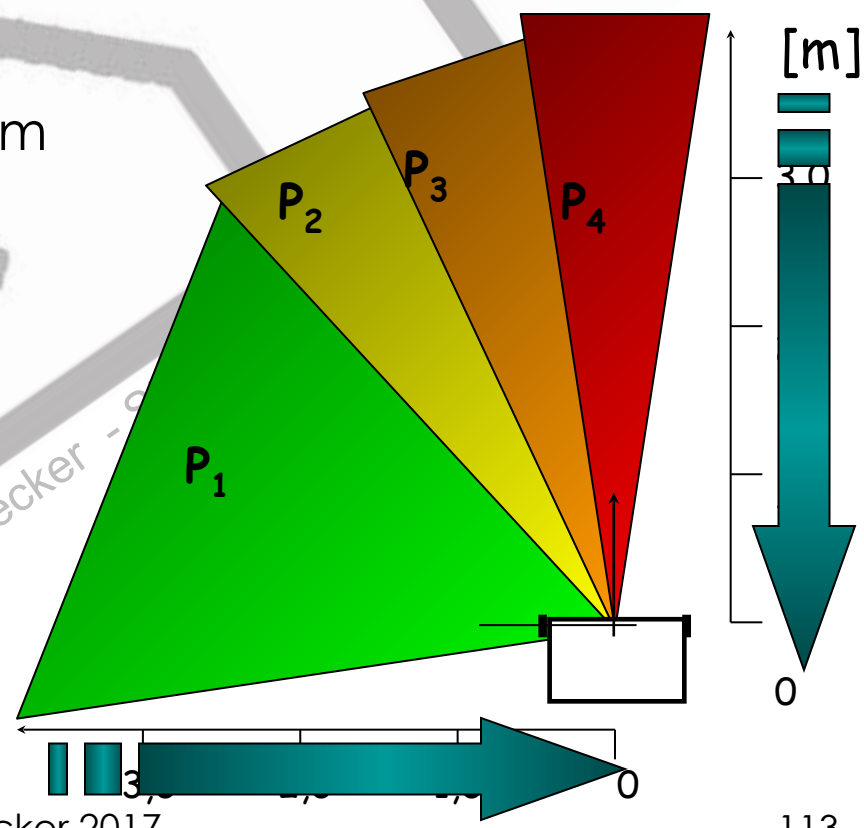
Controle Reativo

- Reações baseadas em informações prévias do ambiente (mapa)
- Ao mesmo tempo armazenam novas informações obtidas pelos sensores → atualização do mapa...
- Planejamento de Trajetória (localmente ou globalmente)
- Tem um "gasto" computacional maior
- Mais lento que o controle reflexivo

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Regras Inibitivas

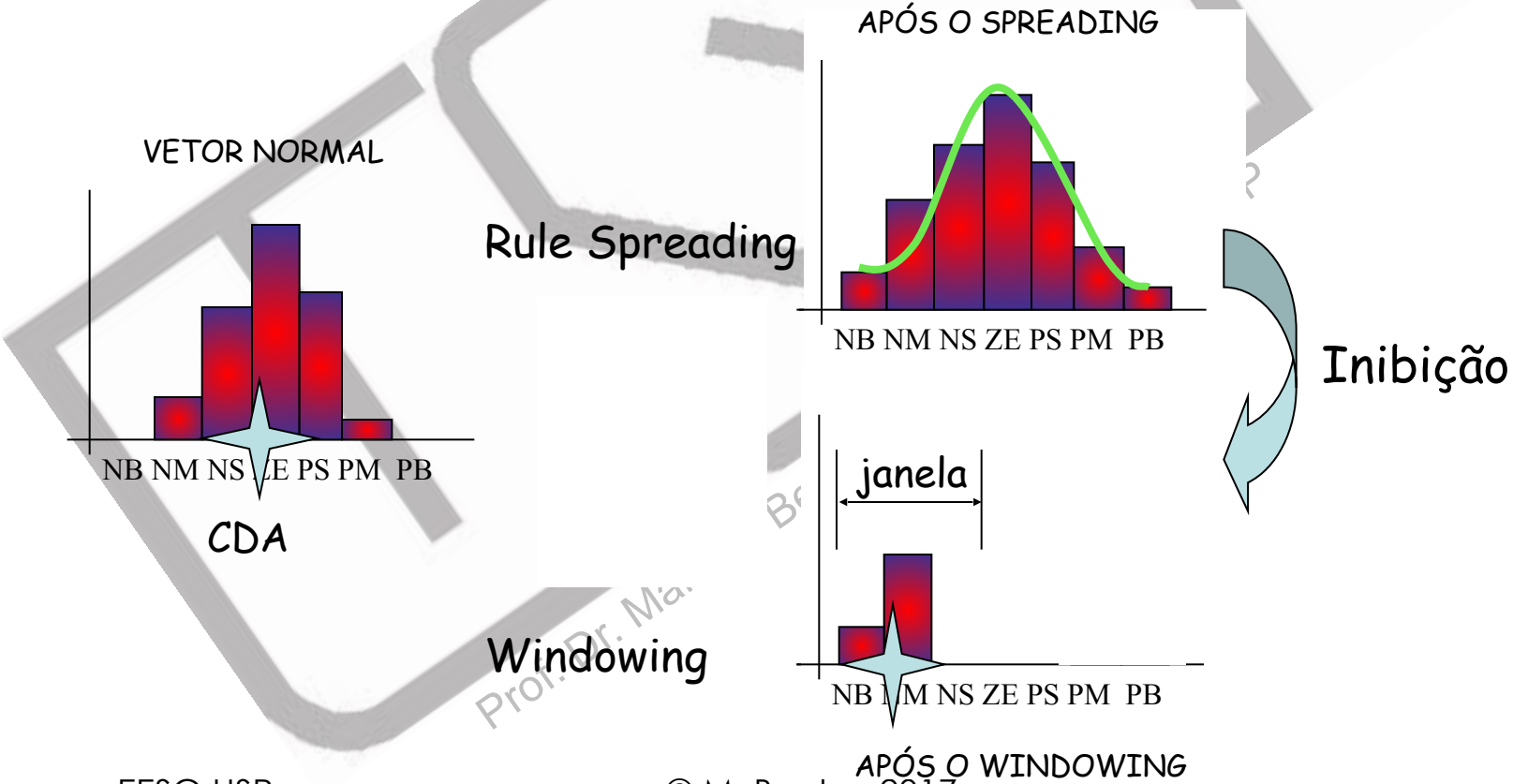
- Adicionadas ao controlador de navegação
- *"SE o obstáculo está À ESQUERDA então NÃO esterce ZE, NS, NM, etc..."*
- Grau de ativação é reduzido em função da proximidade do obstáculo (1 a 0)
- Vetor máscara



Prof. Dr. Marcelo Becker

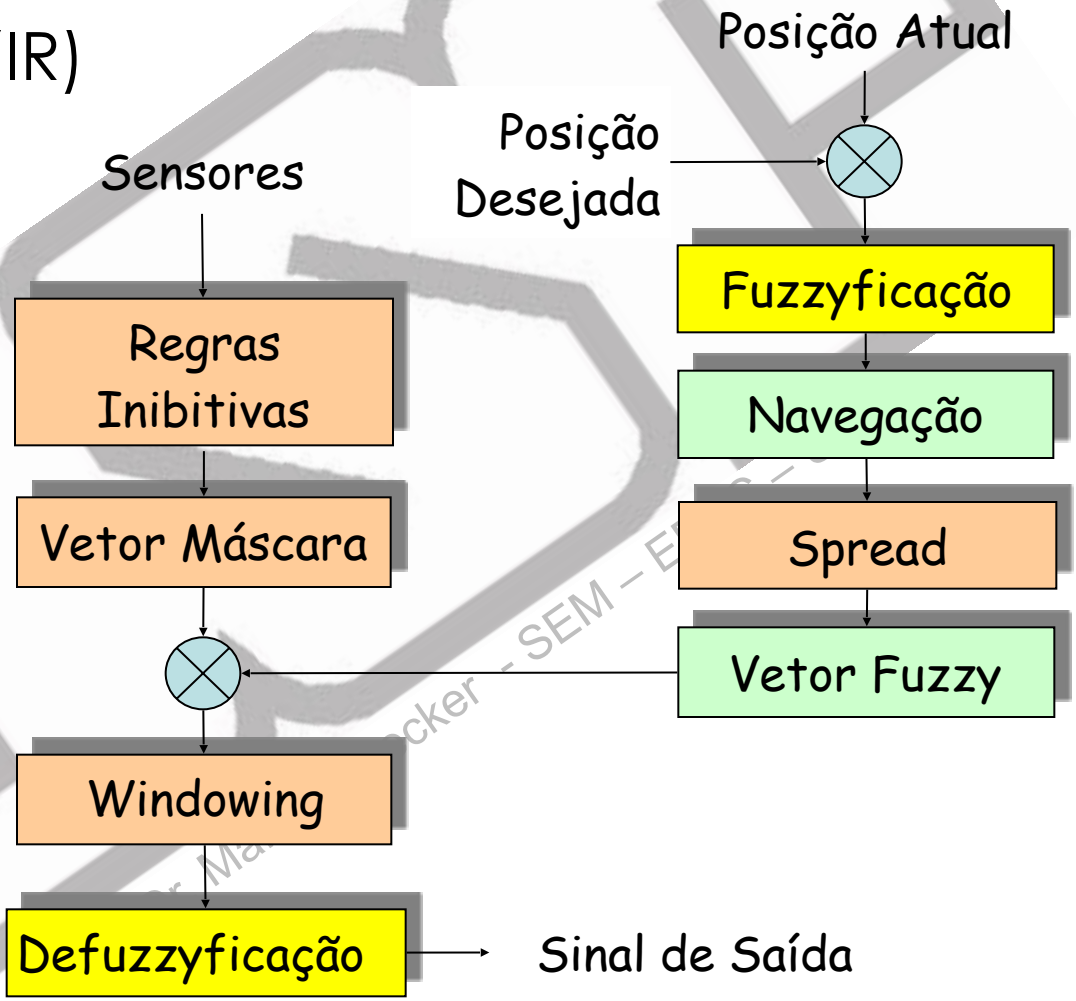
Regras Inibitivas

- Problemas:
  - regras inibitivas e *defuzzyficação* por CDA



Regras Inibitivas

- Fluxograma (IR)



Seletor Proporcional de Comportamento

- Controlador *fuzzy* específico
- O seletor combina os comportamentos:
  - ir à posição desejada vs. desviar de obstáculos
  - a distância ao obstáculo e à posição final

$$\alpha_c = d_{obst} \cdot e^{0.1(d_{obst} - 15)}$$

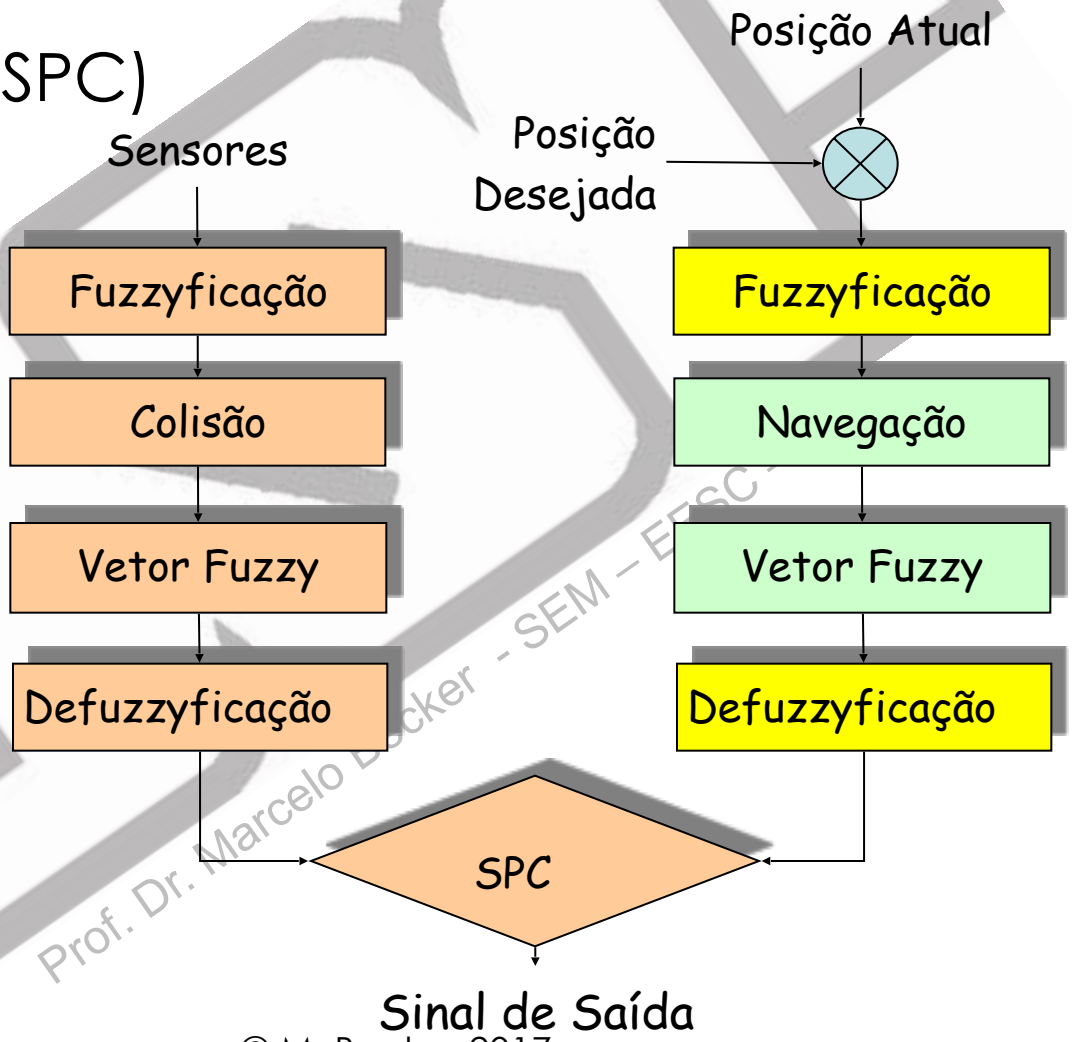
$$u = u_c \frac{\alpha_c}{(\alpha_c + \alpha_n)} + u_n \frac{\alpha_n}{(\alpha_c + \alpha_n)}$$

$$\alpha_n = d_f \cdot e^{0.1(d_f - 15)}$$

Prof. Dr. Marcelo Becker - SEM - EESC – USP

Seletor Proporcional de Comportamento

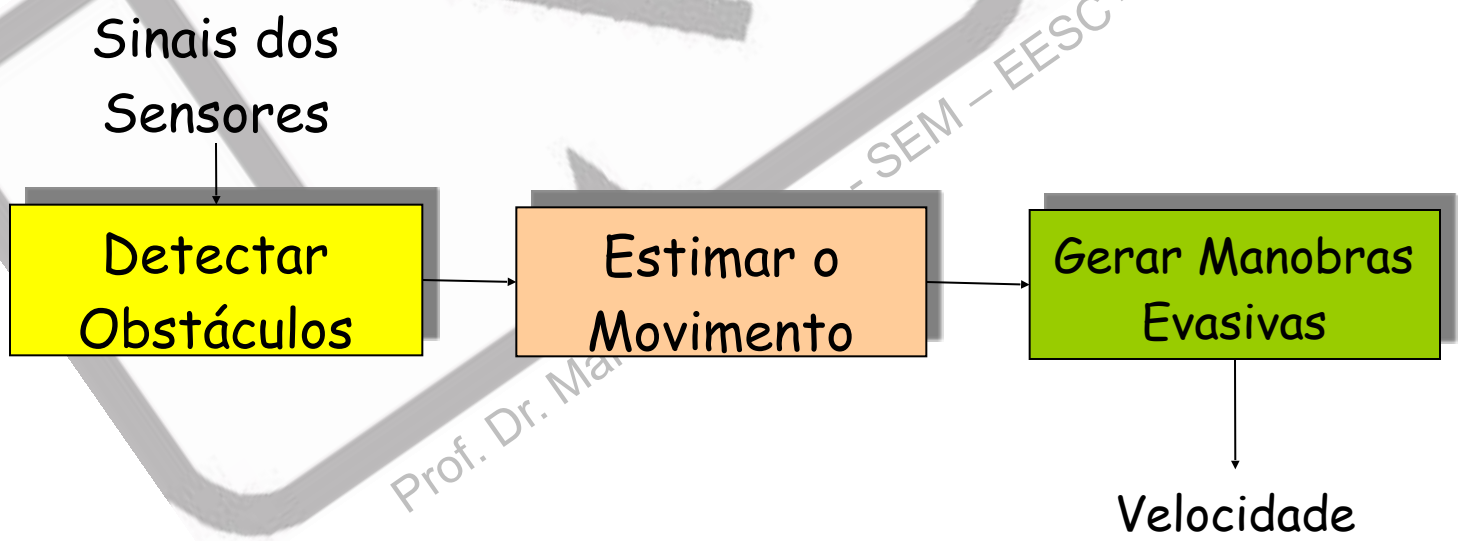
- Fluxograma (SPC)



Prof. Dr. Marcelo Becker - SEM – EESC

Velocity Obstacle Modificado

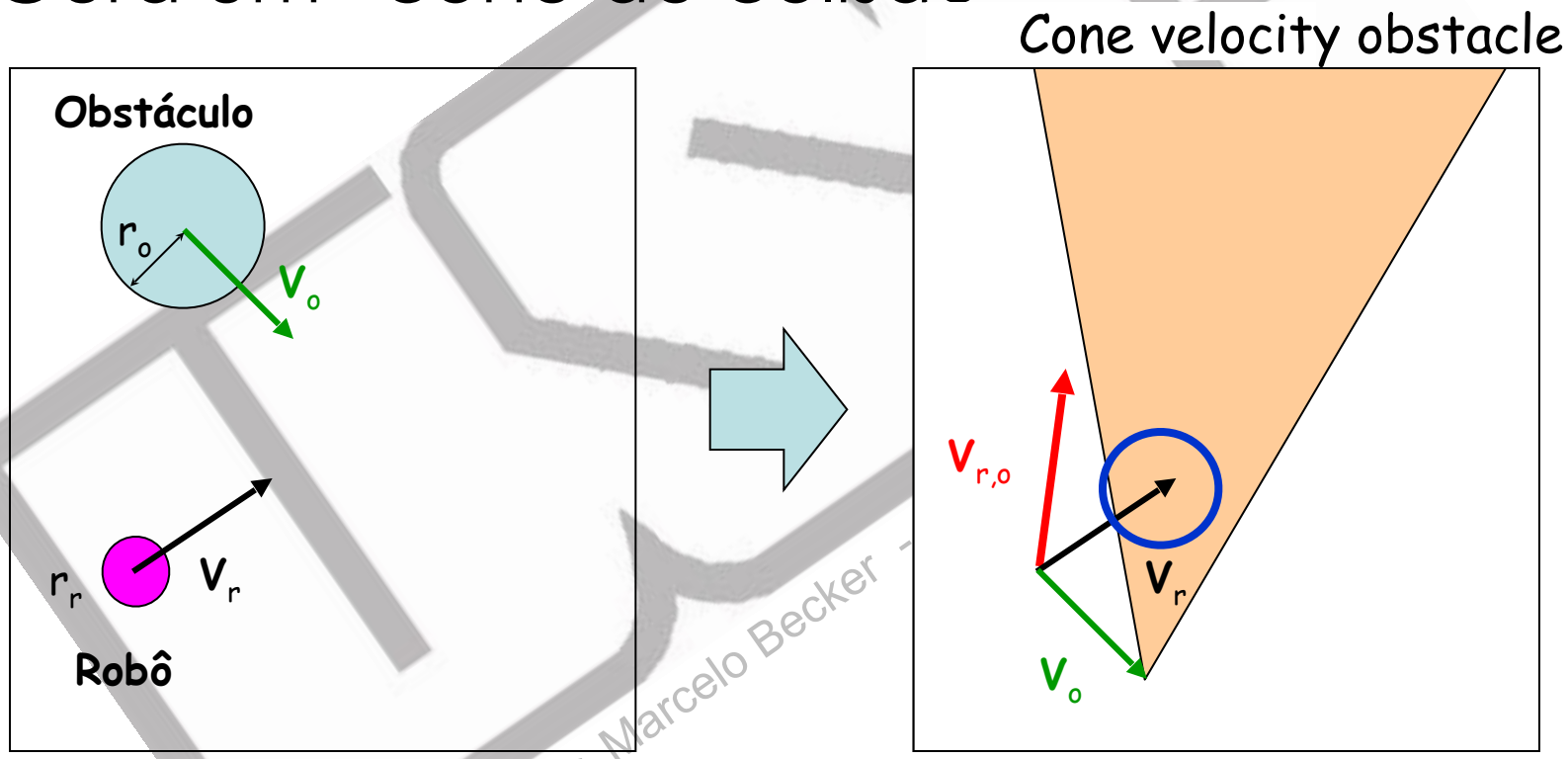
- Consiste de 3 algoritmos:
  - detecção e classificação de obstáculos
  - monitorar o movimento
  - gerar manobras evasivas



Prof. Dr. Mar... SEM – EESC – USP

Velocity Obstacle Modificado

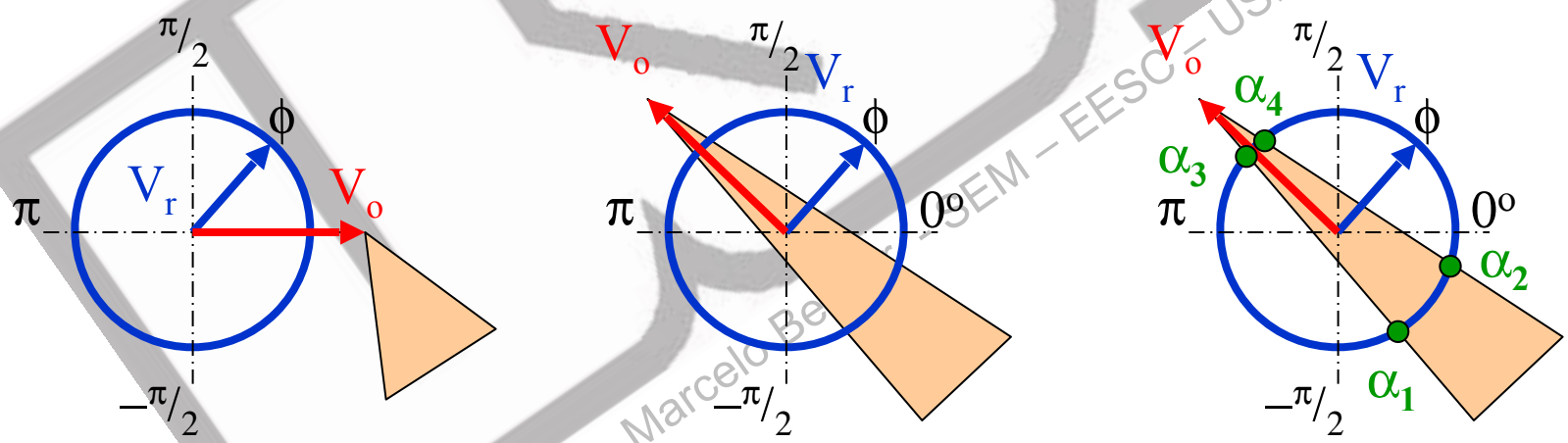
- Gera um "cone de colisão"



Prof. Dr. Marcelo Becker

Velocity Obstacle Modificado

- "Posicionamento" do vetor velocidade
  - velocidade de cruzeiro
  - acelerar ou frear o veículo
  - parada de emergência

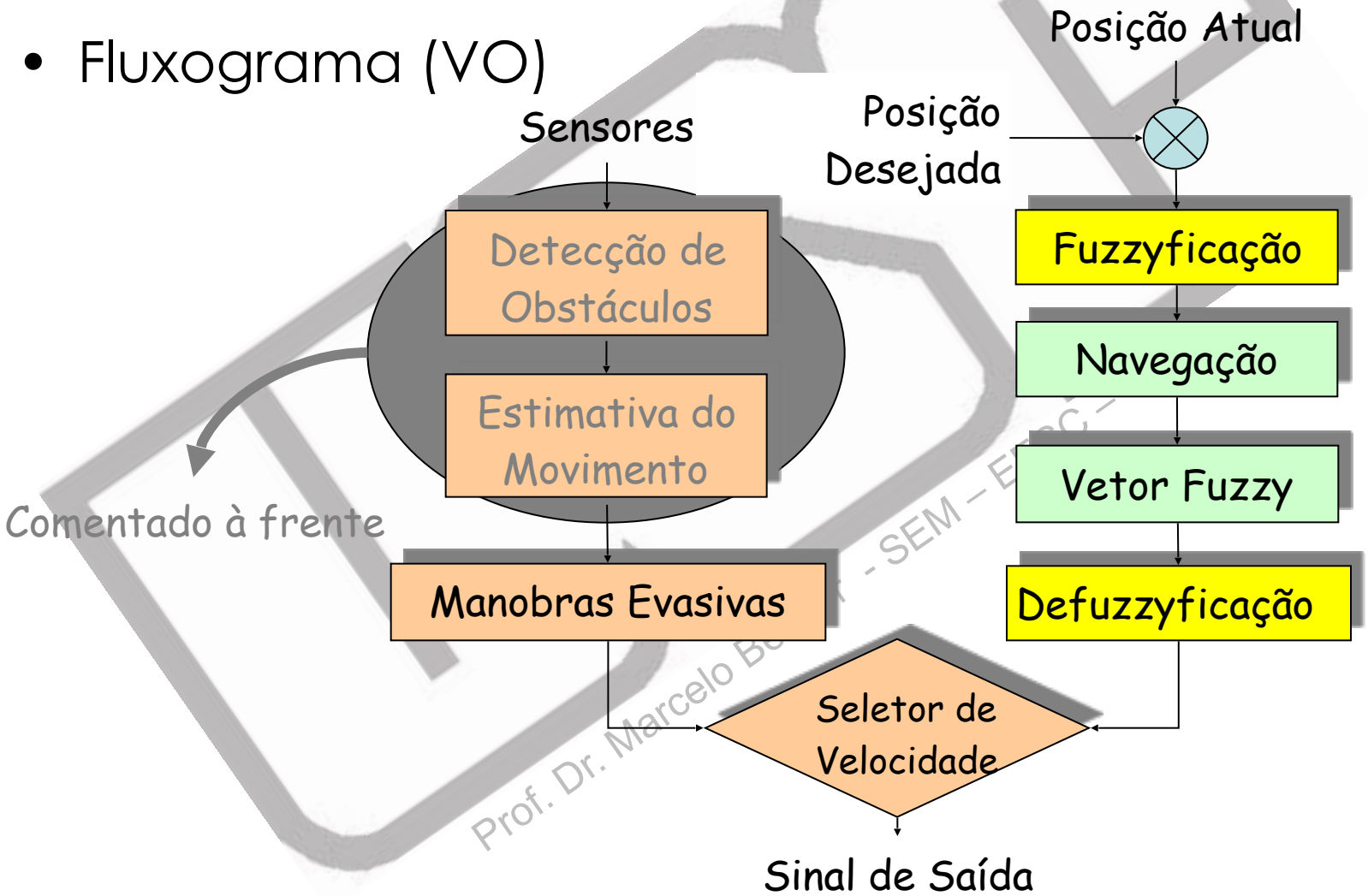


- Margear obstáculos

Prof. Dr. Marcelo Becker – SEM – EESC – USP

Velocity Obstacle Modificado

- Fluxograma (VO)



Controlador de Velocidade

- Controlador *fuzzy*
- Dados de entrada:
  - ângulo de correção ($\theta_{correção}$)
  - distância à posição desejada (d_f)
- Regras:

SE $|\theta_{correção}|$ é *grande* ENTÃO → Vel é devagar

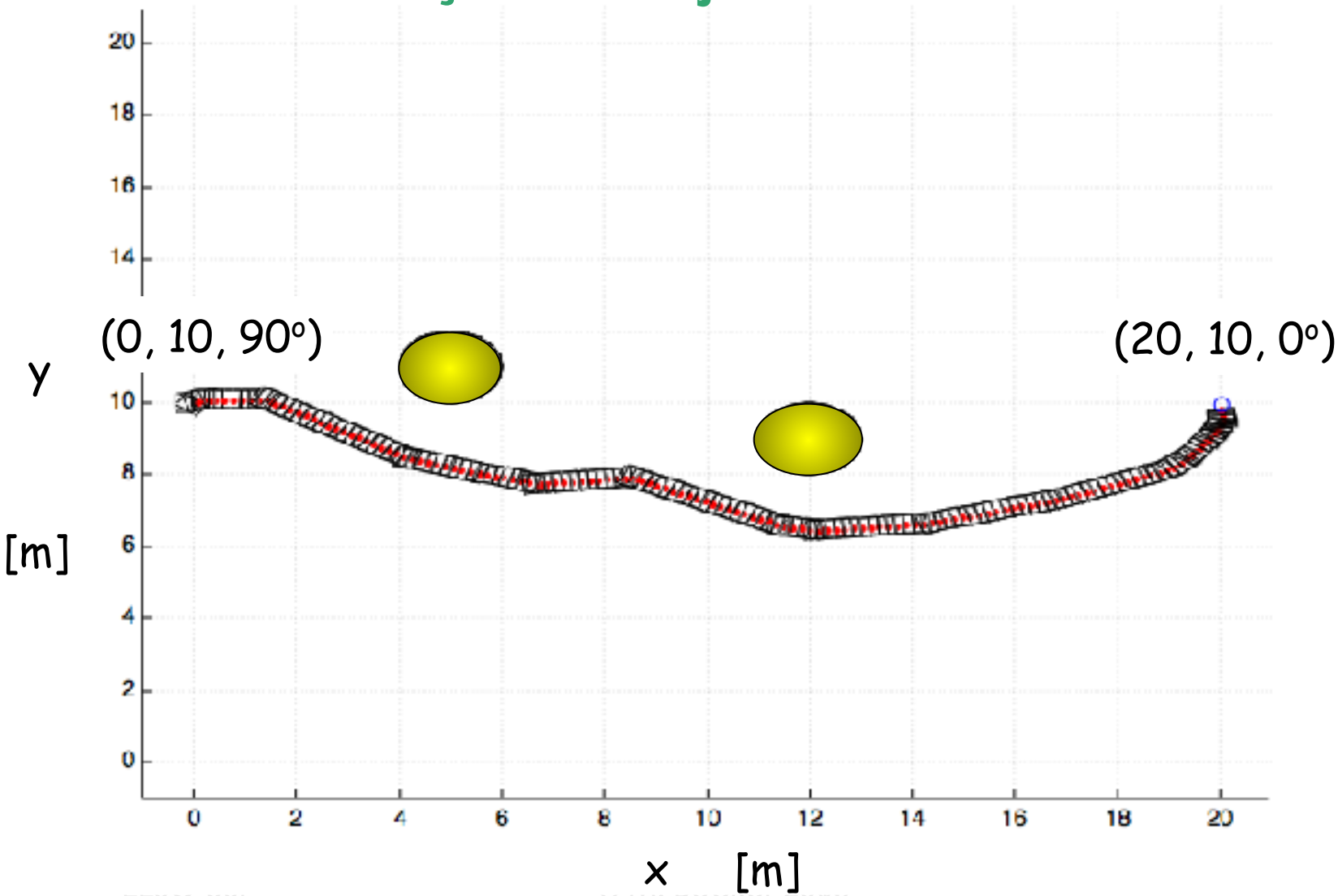
SE $|\Delta\theta_{correção}|$ é *grande* ENTÃO → Vel devagar

SE $|\Delta\theta_{correção}|$ é *pequeno* e d_f *não é pequena* ENTÃO → Vel rápida

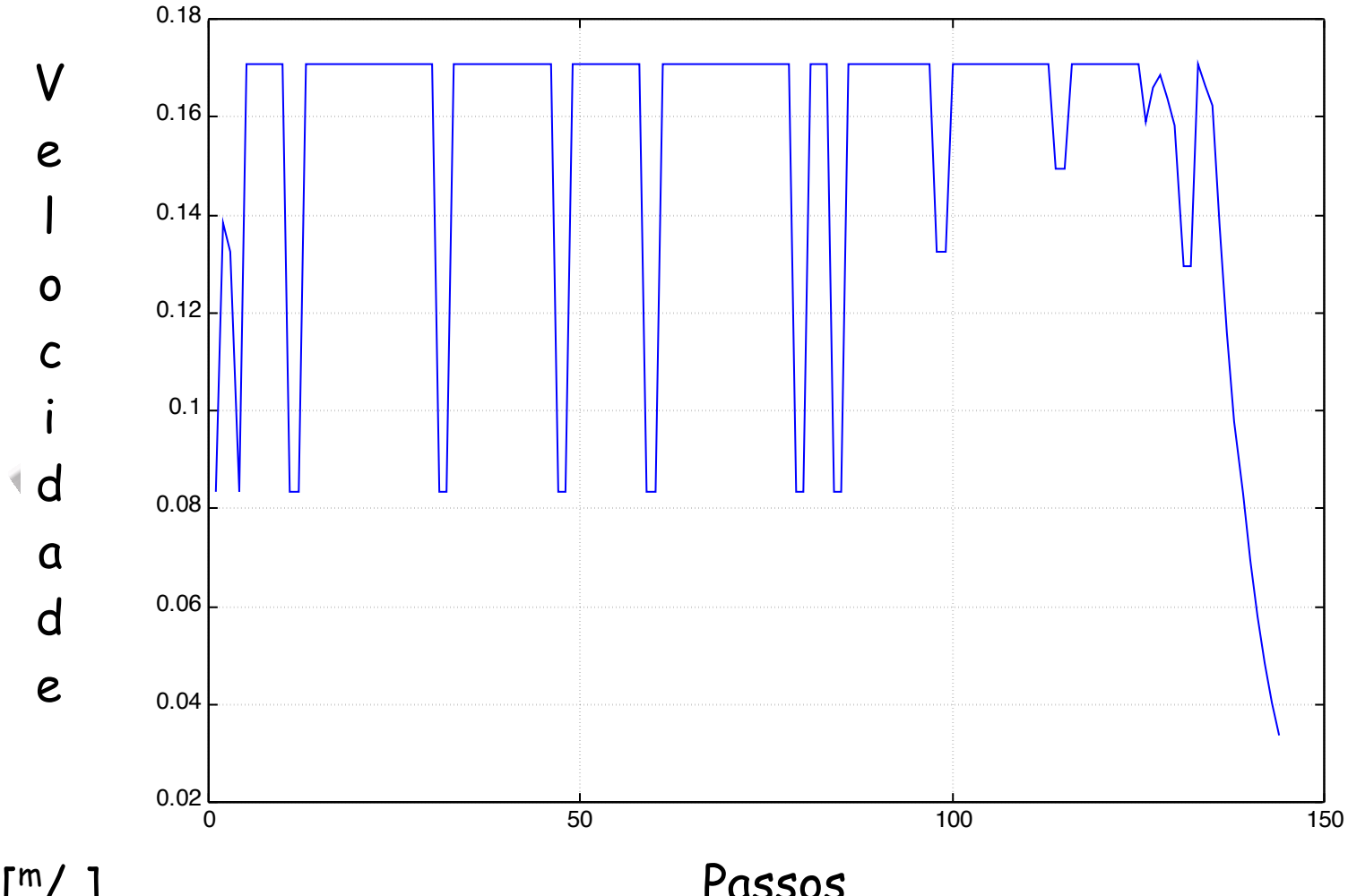
SE d_f é *pequena* ENTÃO → Vel devagar

Planejamento de Trajetórias:

Simulações - Trajetória Simulada



Planejamento de Trajetórias:
Simulações - Velocidade



© M. Becker 2017

Controlador de Velocidade

- Controlador *fuzzy*
- Dados de entrada:
  - ângulo de correção ($\theta_{correção}$)
  - distância à posição desejada (d_f)
- Regras:

~~SE $|\theta_{correção}|$ é *grande* ENTÃO → Vel é devagar~~

~~SE $|\Delta\theta_{correção}|$ é *grande* ENTÃO → Vel devagar~~

~~SE $|\Delta\theta_{correção}|$ é *pequeno* e d_f *não é pequena* ENTÃO → Vel rápida~~

SE d_f é *pequena* ENTÃO → Vel devagar

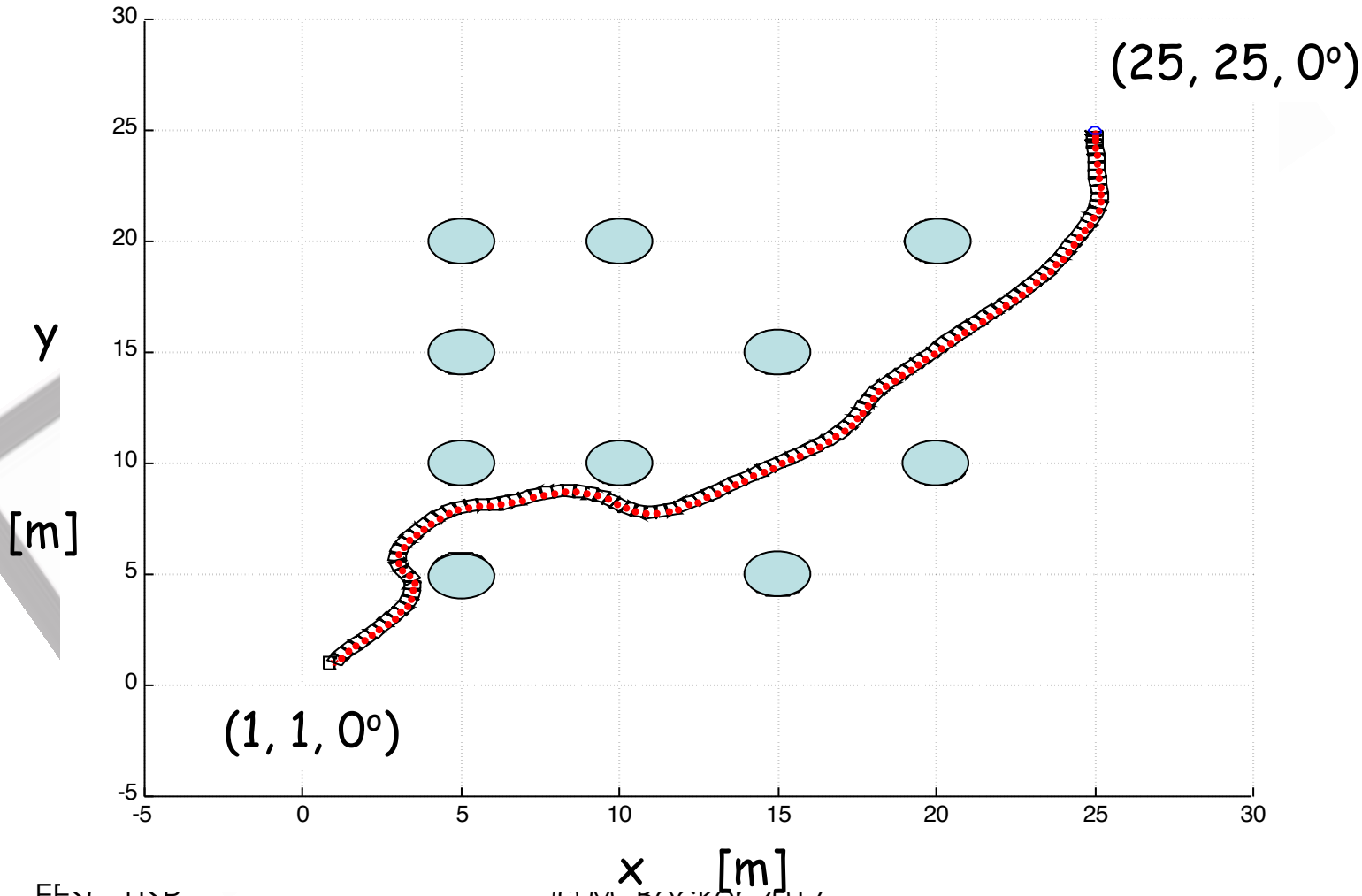
APROXIMAÇÃO FINAL

Planejamento de Trajetórias:
Simulações - Desvio de Obstáculos

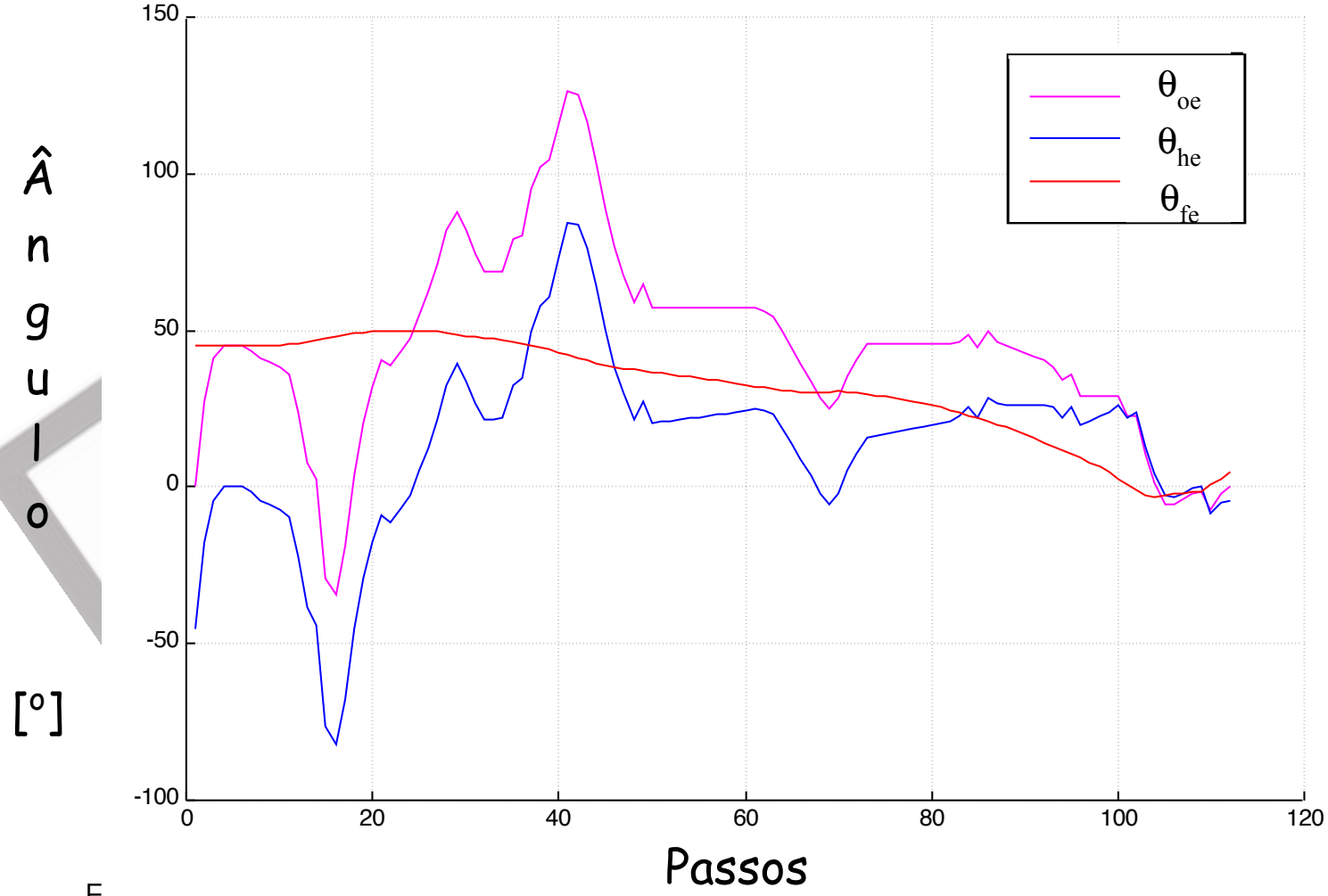
- MatLab™
  - ambiente com 10 obstáculos fixos
  - controlador *fuzzy* de navegação
  - regras inibitivas (RI) vs. seletor proporcional de comportamento (SPC)
  - velocidade de cruzeiro
- 3 gráficos

Prof. Dr. Marcelo Becker - SEM – EESC – USP

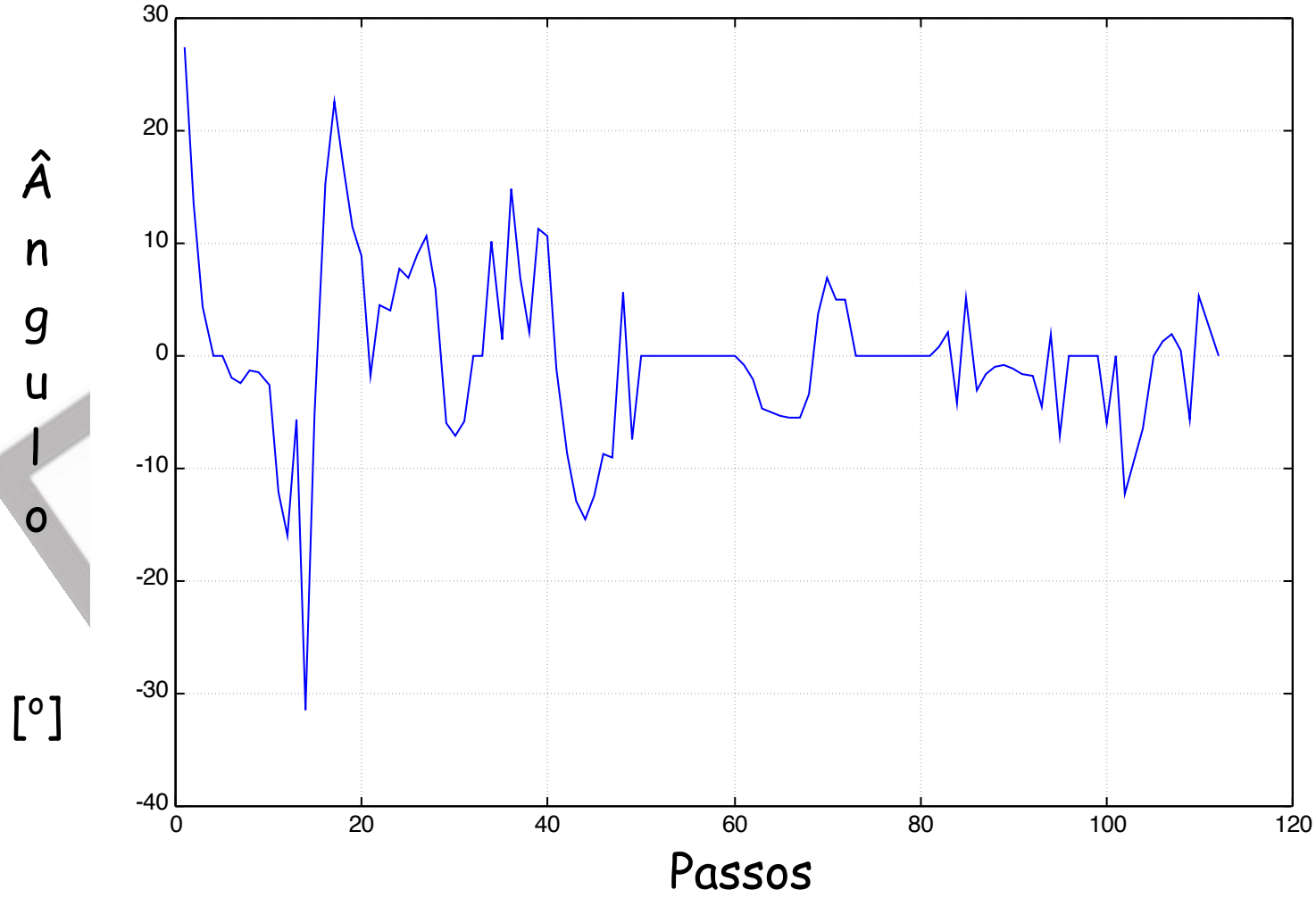
Planejamento de Trajetórias:
Simulações - Trajetórias Simuladas



Planejamento de Trajetórias:
Simulações - Ângulos de Erro



Planejamento de Trajetórias:
Simulações - Ângulos de Saída ($\theta_{correção}$)



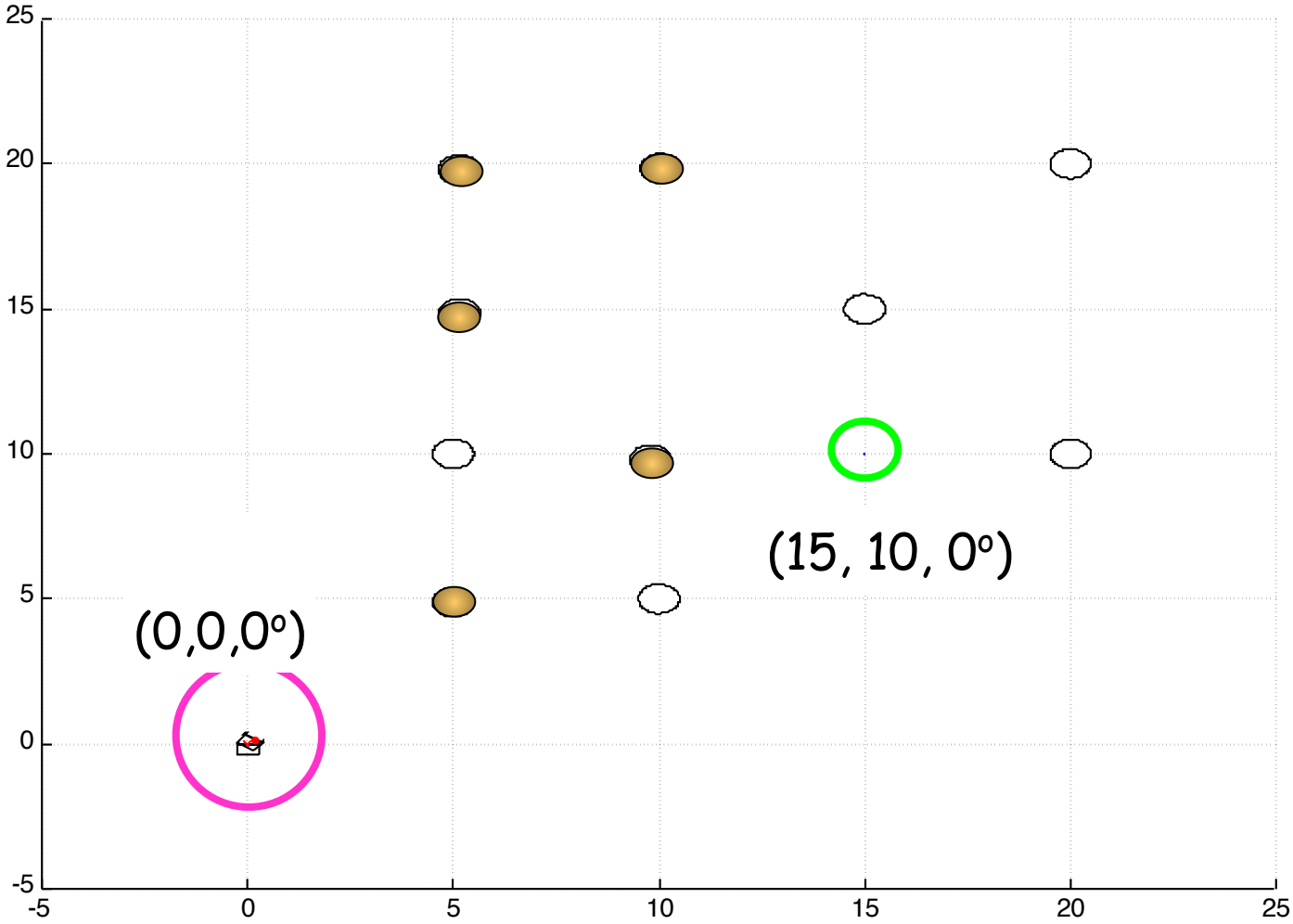
Planejamento de Trajetórias:
Simulações - Desvio de Obstáculos

- MatLab™
  - ambiente com 10 obstáculos (5 fixos e 5 móveis)
  - controlador *fuzzy* de navegação
  - regras inibitivas (RI) vs. *velocity obstacle* modificado (VO)
  - velocidade de cruzeiro
- 3 gráficos

Prof. Dr. Marcelo Becker - SEM – EESC – USP

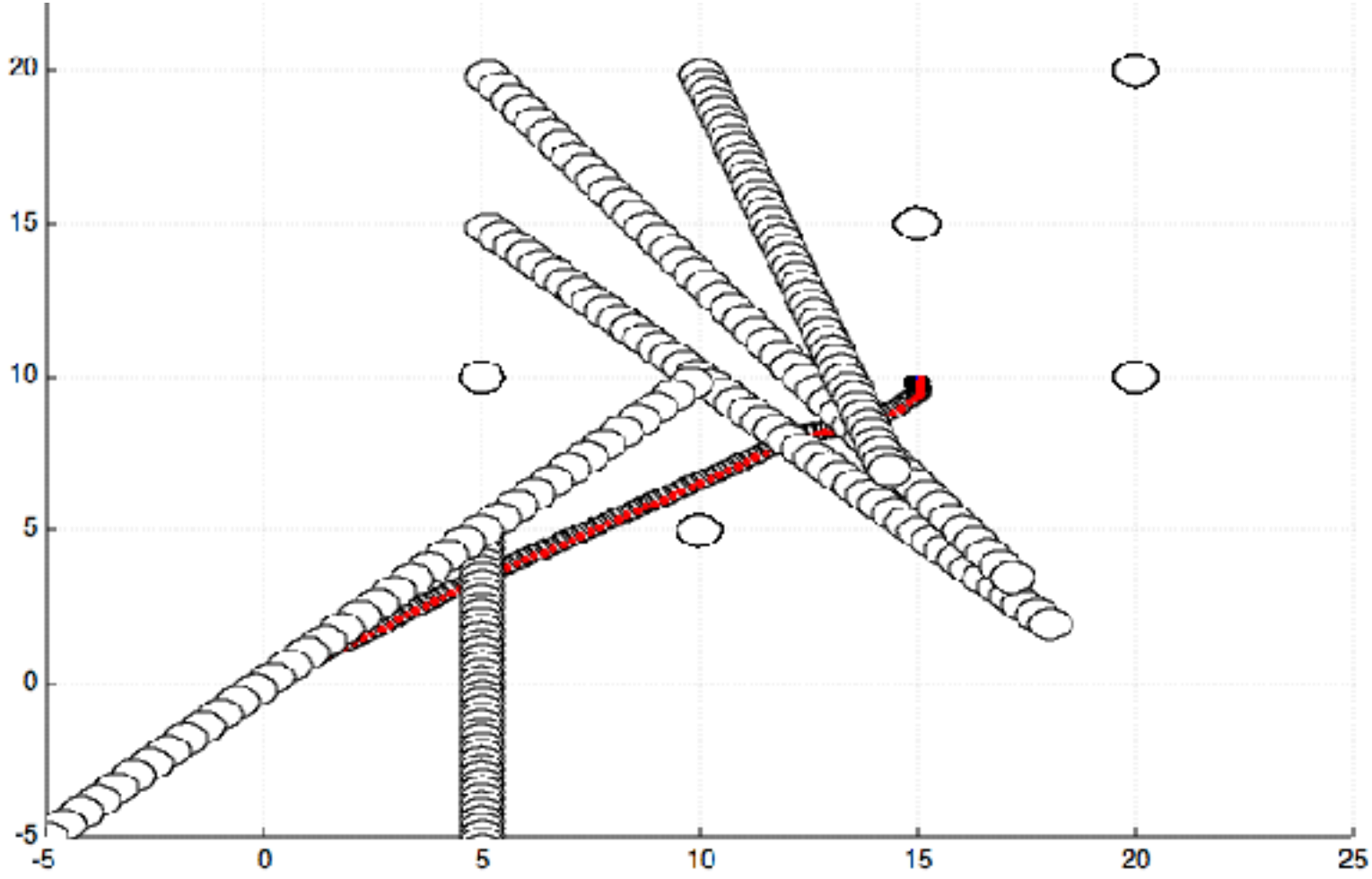
Planejamento de Trajetórias:

Simulações - Ambiente Dinâmico



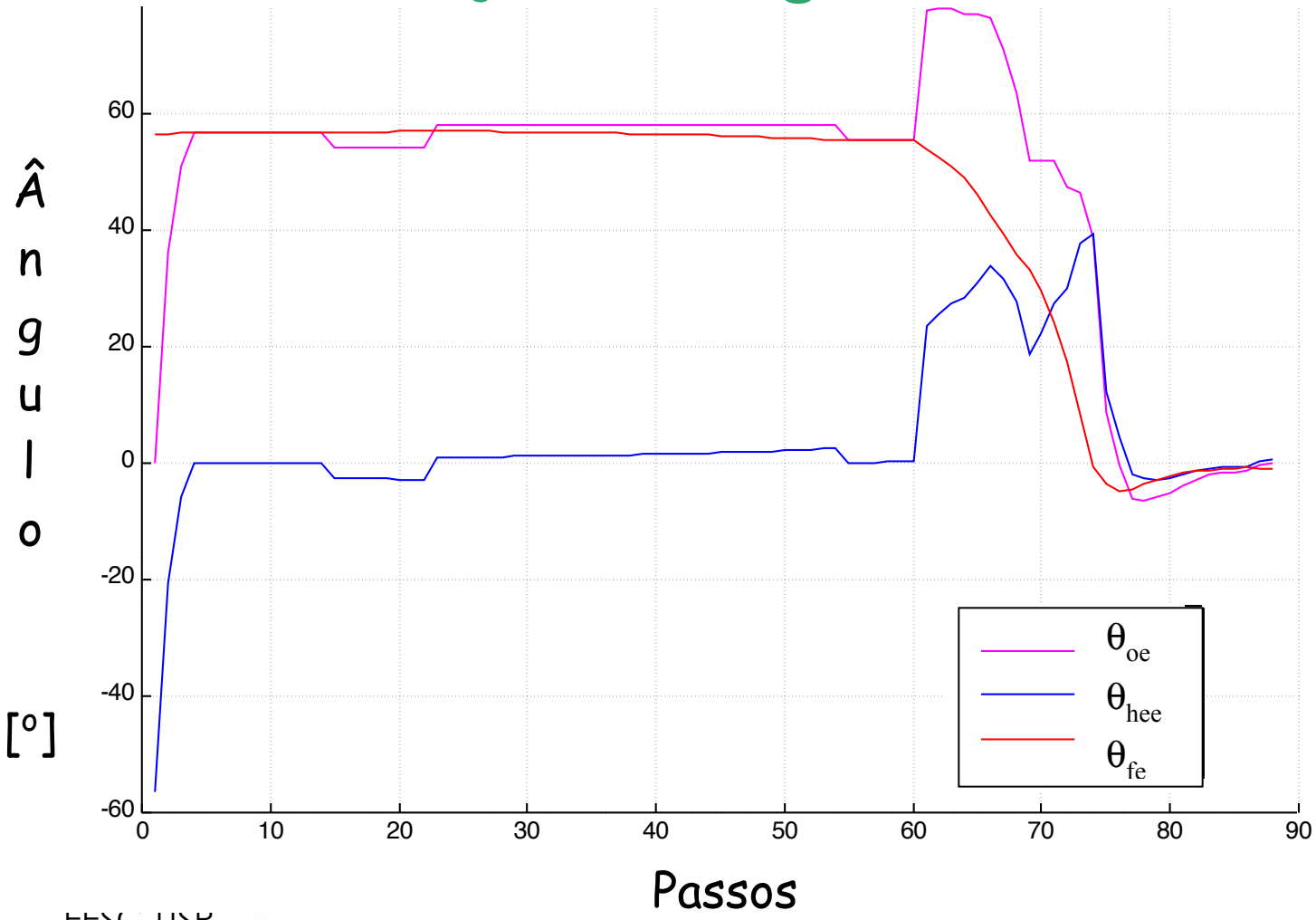
Planejamento de Trajetórias:

Simulações - Trajetórias



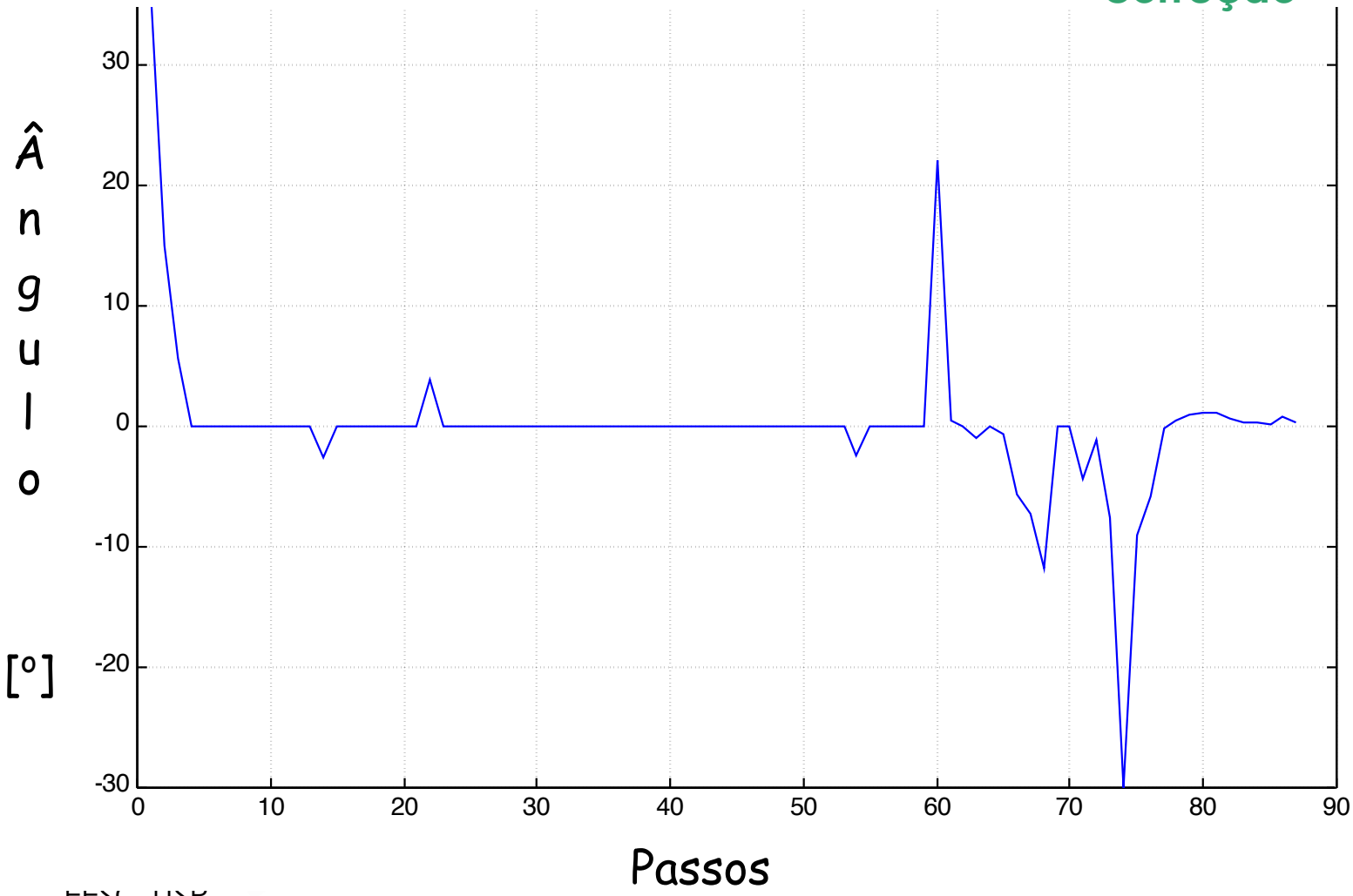
Planejamento de Trajetórias:

Simulações - Ângulos de Erro



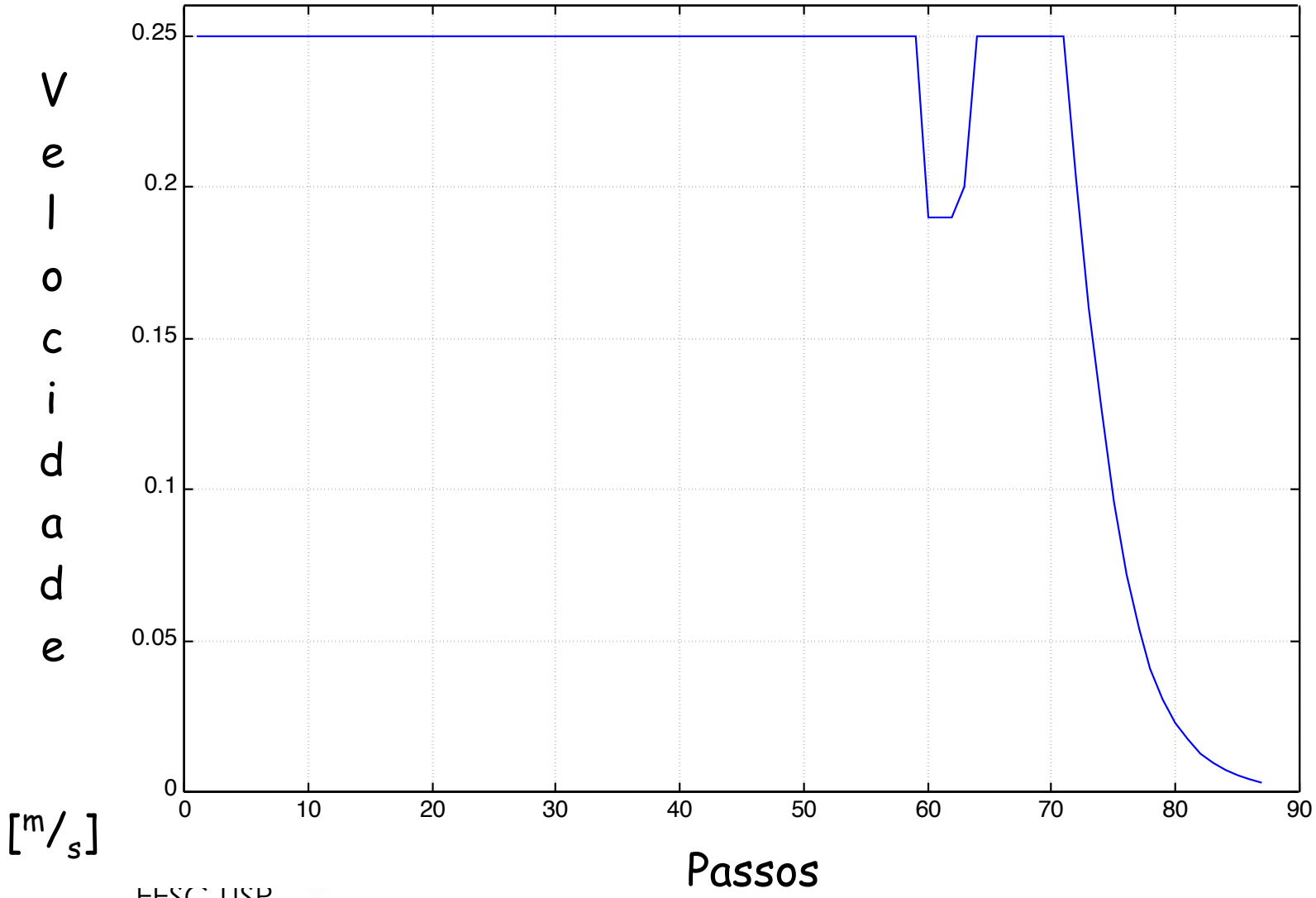
Planejamento de Trajetórias:

Simulações - Ângulo de Saída ($\theta_{correção}$)



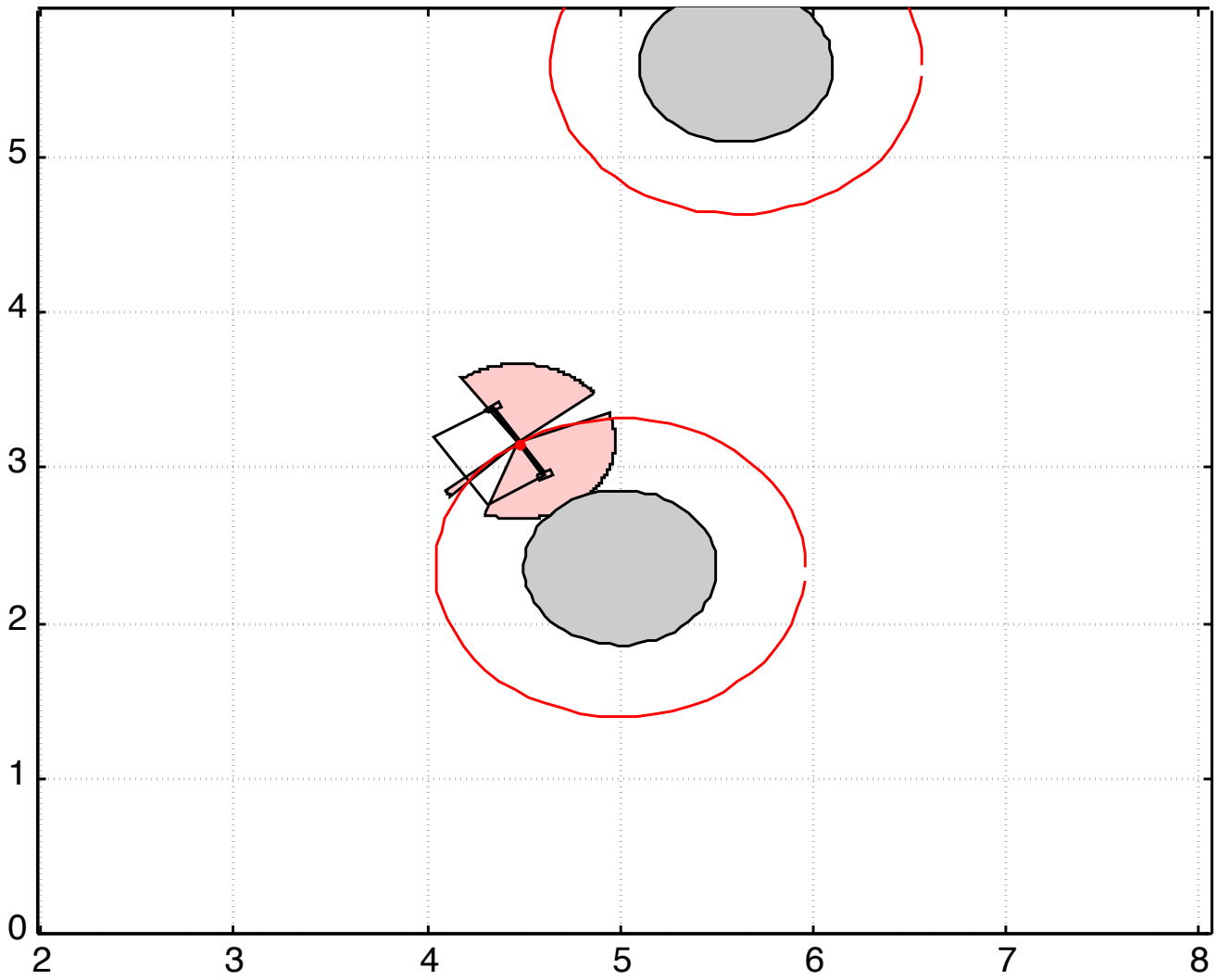
Planejamento de Trajetórias:

Simulações - Velocidade



Planejamento de Trajetórias:

Simulações - Detalhe dos Conjuntos de Ângulos "Proibidos"



Velocity Obstacle Modificado

Detecção e Monitoramento de Obstáculos

- O algoritmo deve:
  - detectar obstáculos
  - classificá-los em móveis ou estáticos
  - monitorar o movimento dos obstáculos móveis

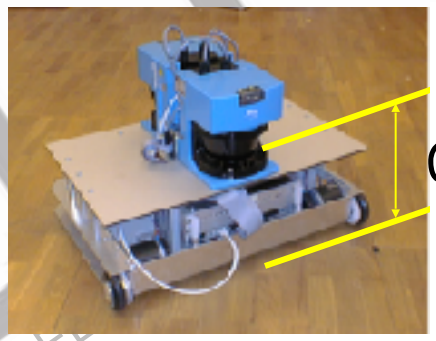
- Dificuldade:
  - número de dados/seg
  - tempo real
  - monitorar pessoas

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Velocity Obstacle Modificado

Detecção e Monitoramento de Obstáculos

- Implementação:
  - sensores laser SICK LMS200
    - 3 *scans*/seg
  - SmartROB-2
    - XOberon

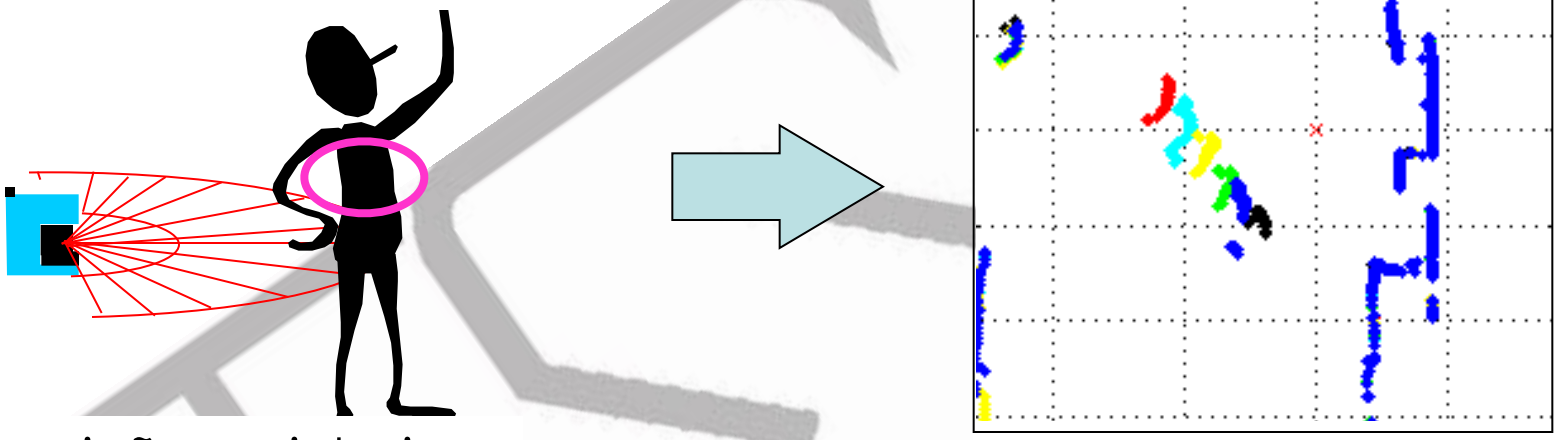


- Dificuldade:
  - posição dos sensores próxima ao solo

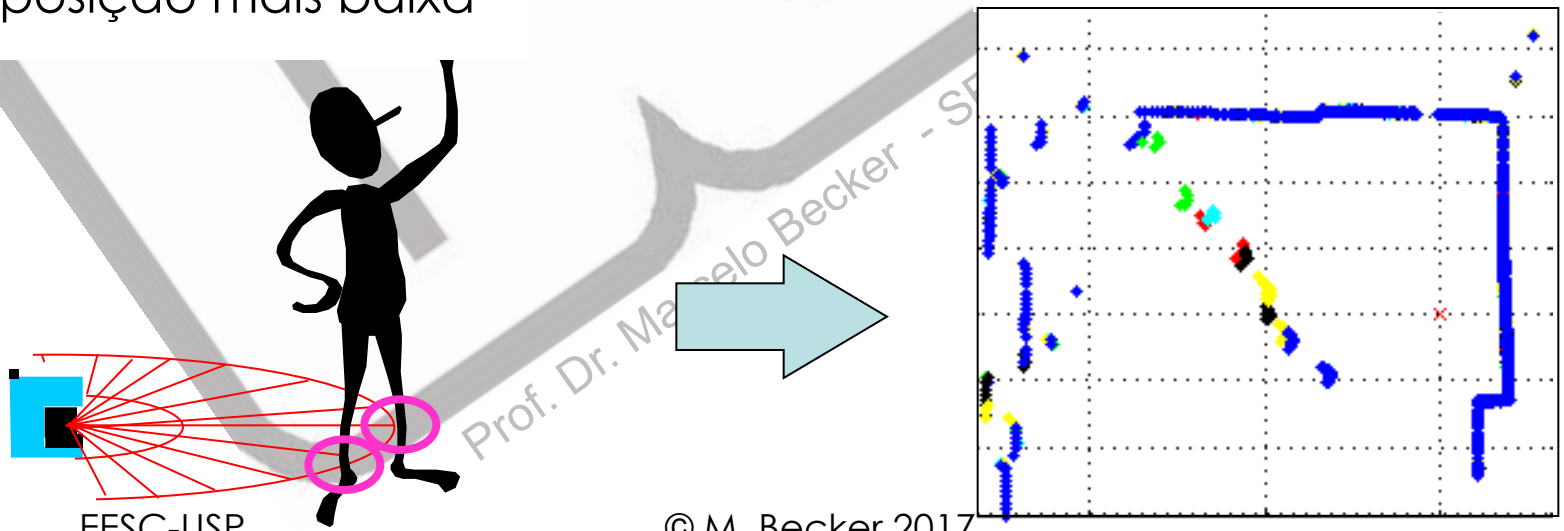
Prof. Dr. Marcelo Becker - SEM – EESC

Velocity Obstacle Modificado

- posição mais alta



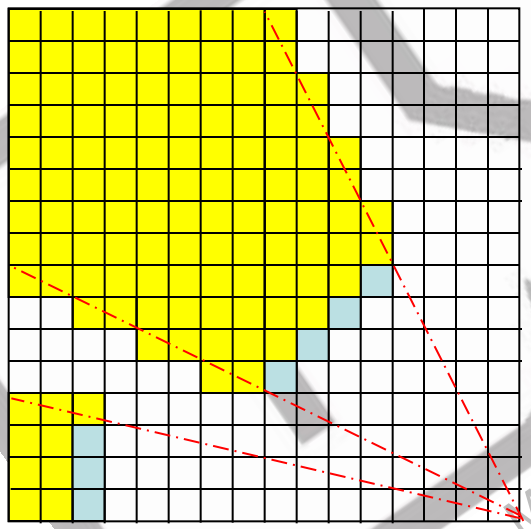
- posição mais baixa



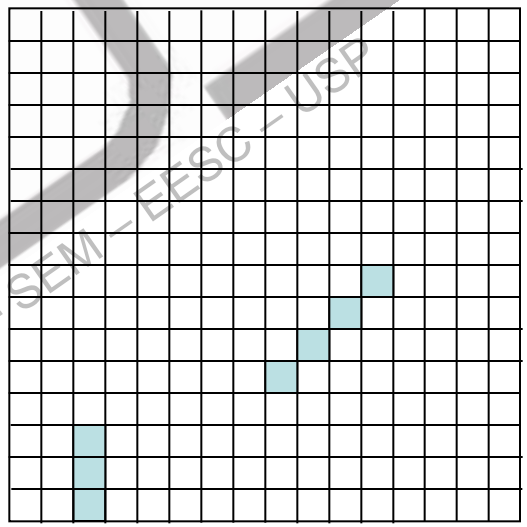
Prof. Dr. Marcelo Becker - SE

Velocity Obstacle Modificado

- Implementação I:
  - representação dos sinais em um *Grid* [Elfes 89]
  - *Time Stamp Map* (TSMs) [Fiorini e Shiller 98]



Occupancy Grid Representation

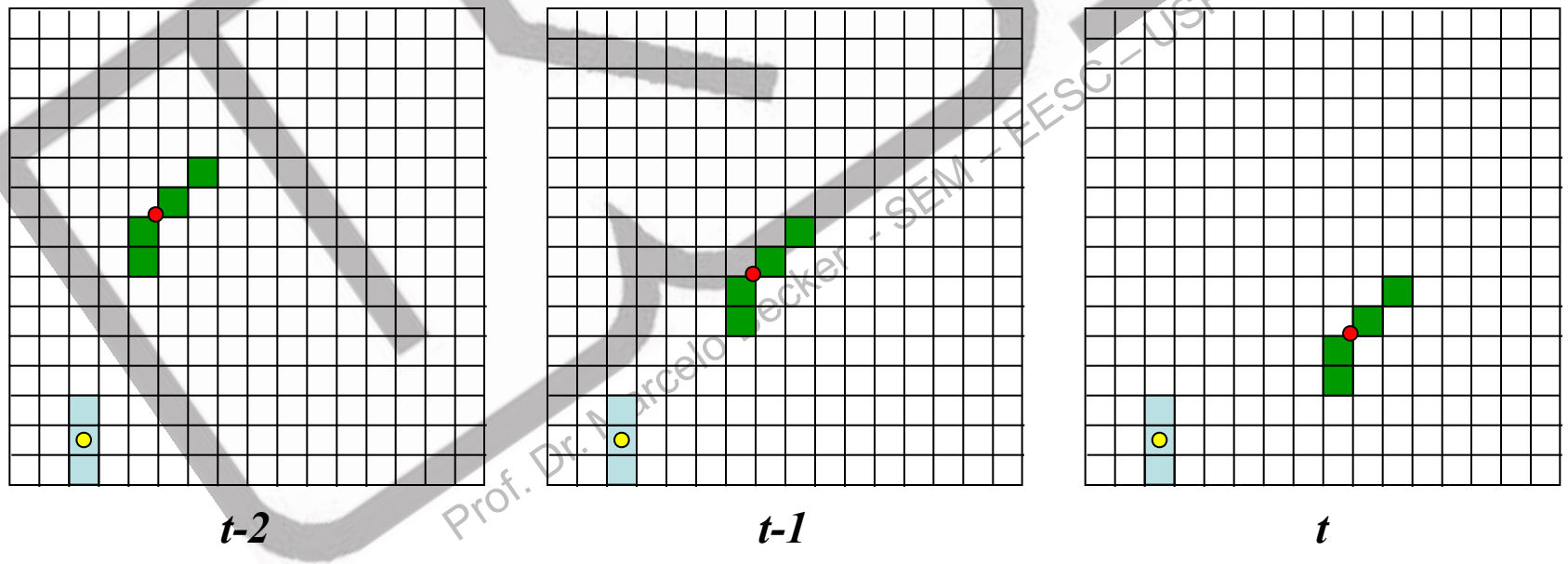


Time Stamp Map

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Velocity Obstacle Modificado

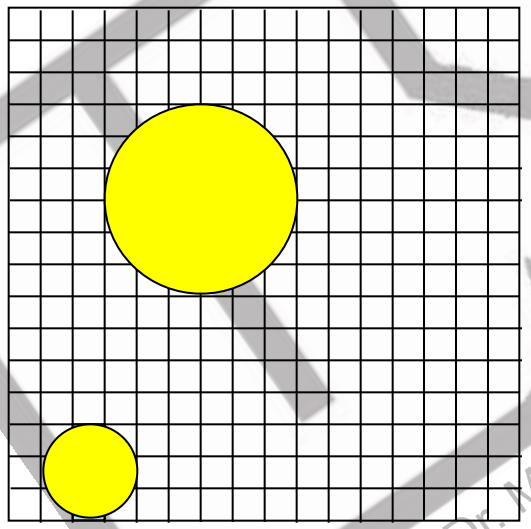
- Implementação II:
  - unir células próximas em conjuntos
  - calcular os CGs dos conjuntos
  - alterações nas posições dos CGs podem indicar movimento...



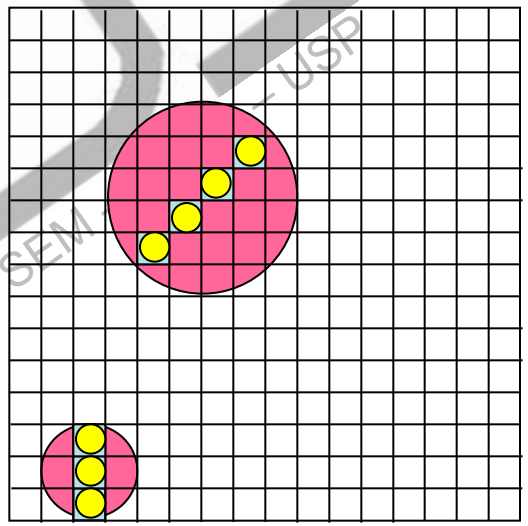
Prof. Dr. Marcelo Becker - SEM - EESC – USP

Velocity Obstacle Modificado

- Implementação III:
  - estima-se a velocidade dos obstáculos ($V_{média}$)
  - para gerar as manobras evasivas → <u>células isoladas</u>



Conjunto completo

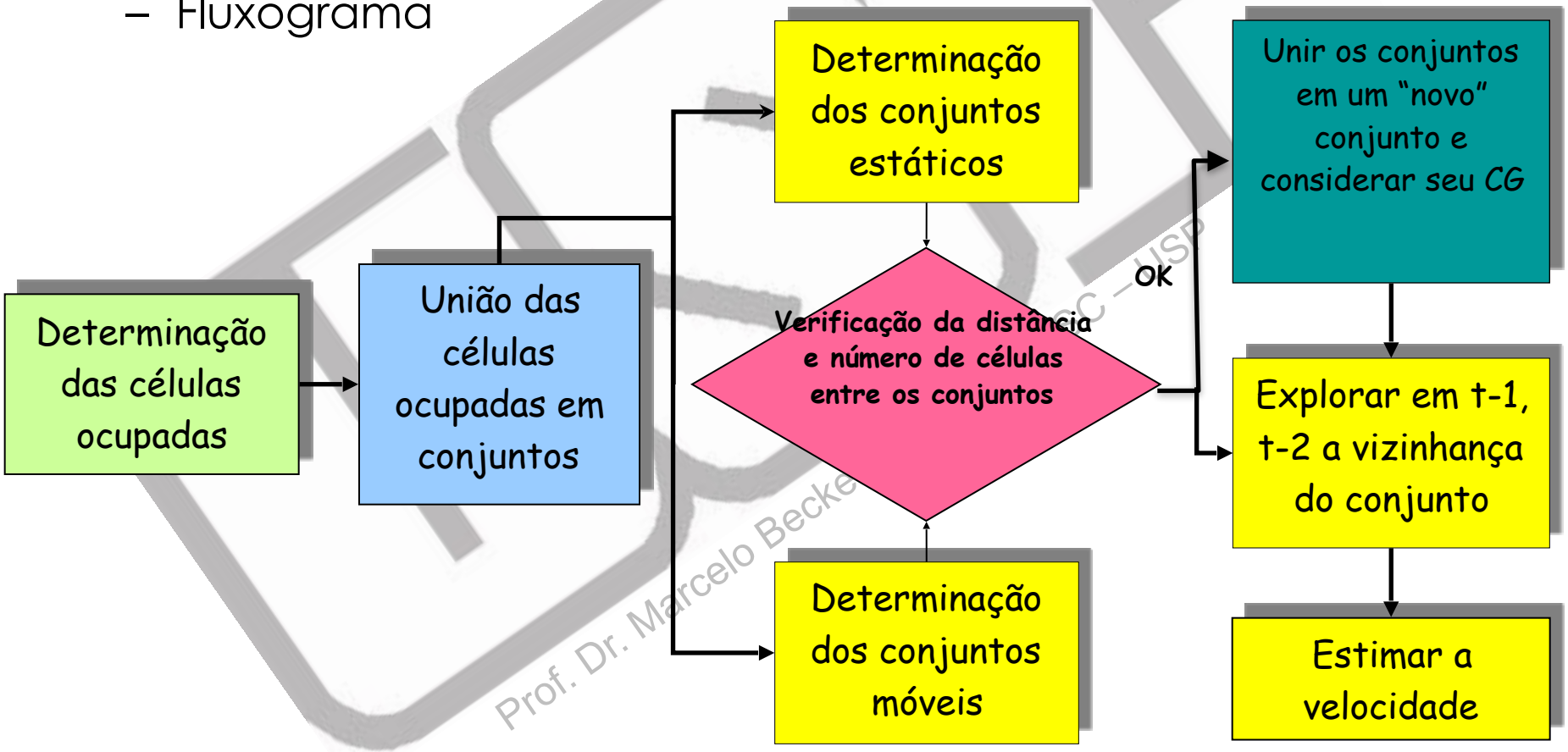


Células isoladas

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Velocity Obstacle Modificado

- Implementação V - *sensores próximos ao solo*:
  - Fluxograma



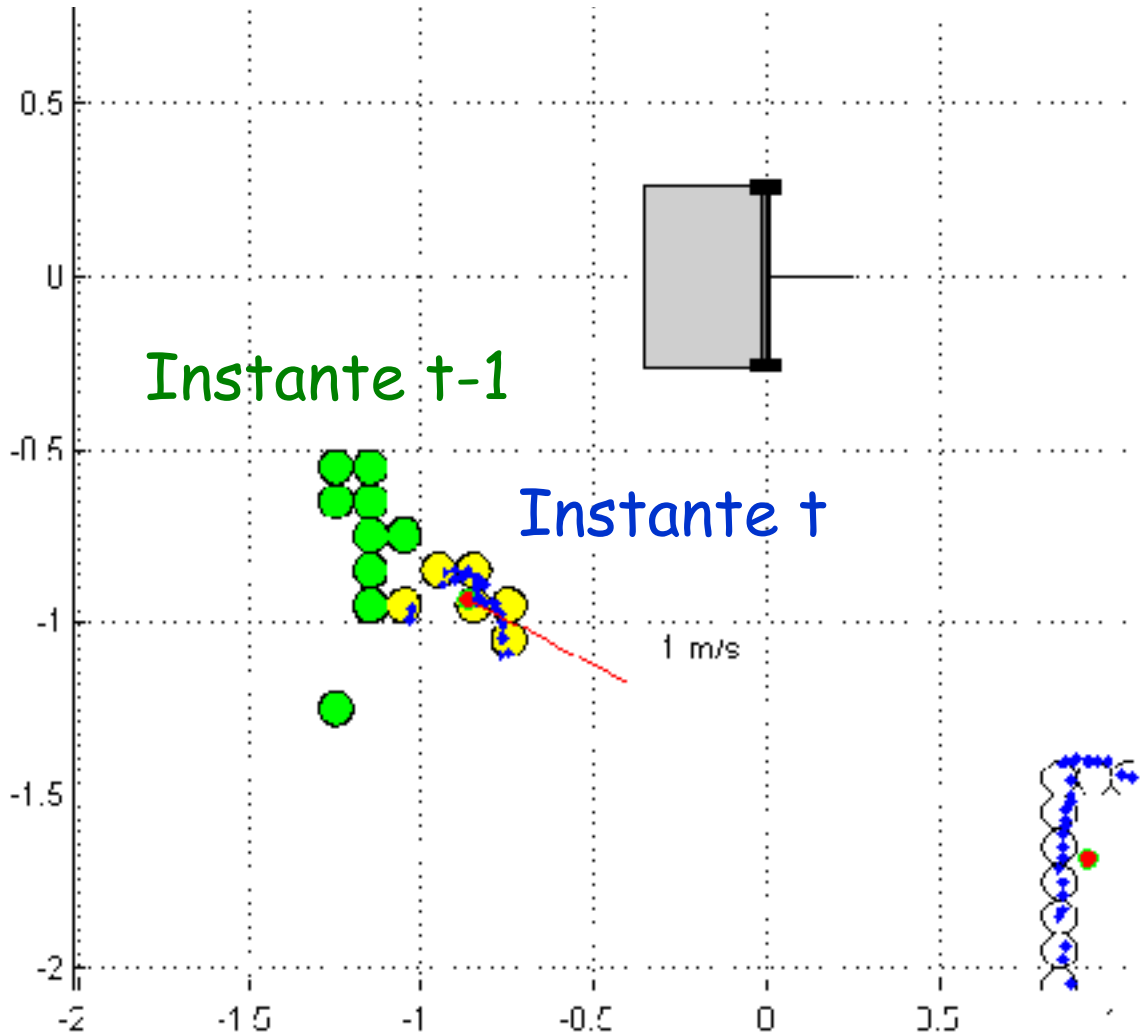
Prof. Dr. Marcelo Becker EESC – USP

Detecção e Monitoramento de Obstáculos

Resultados Experimentais

- robô parado no laboratório
  - sensor em posição elevada

Prof. Dr. Marcelo Becker - SEM – EESC – USP

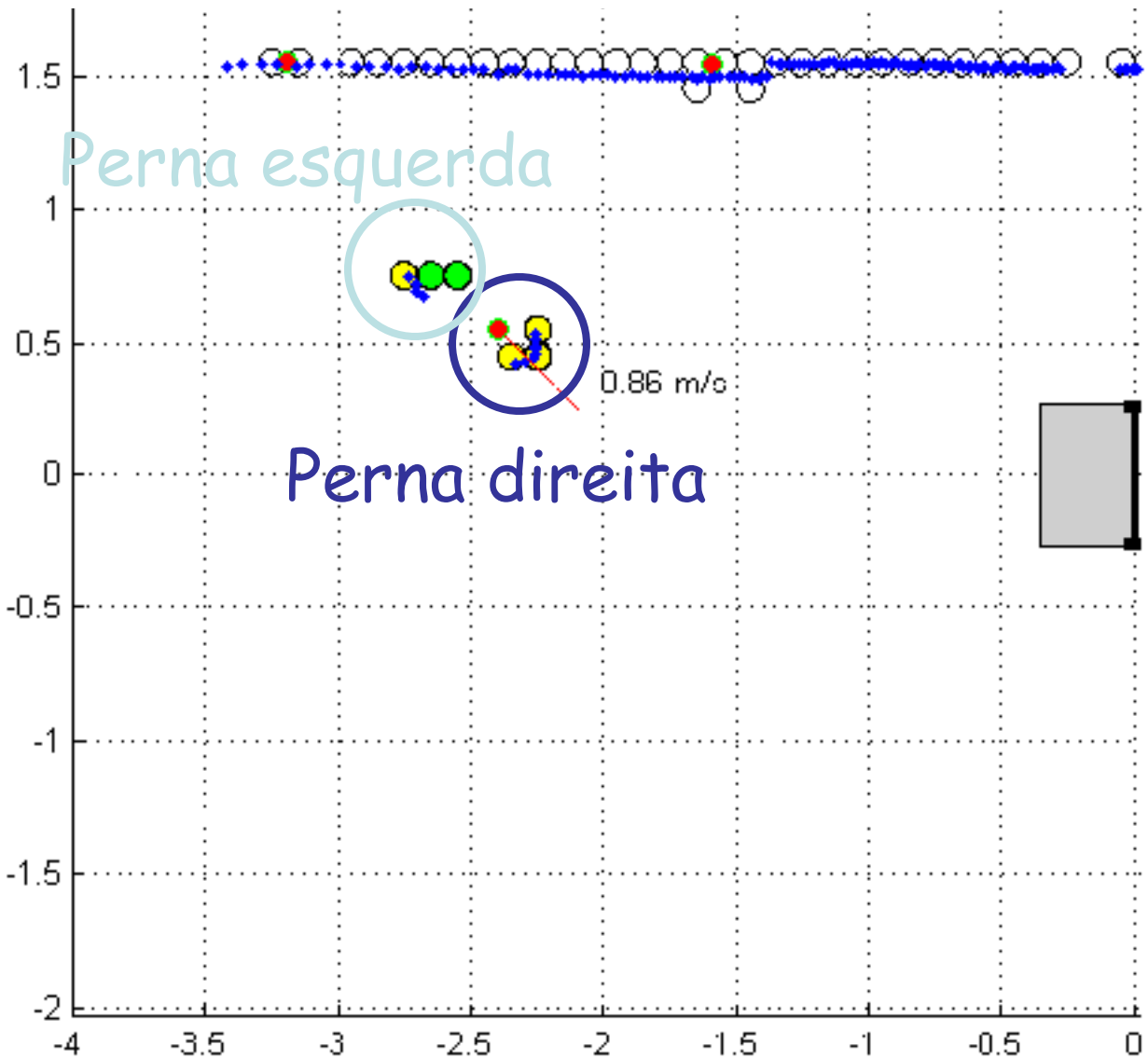

Instante t-1
Instante t
1 m/s
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5

Detecção e Monitoramento de Obstáculos

Resultados Experimentais

- robô parado no laboratório
  - sensor em posição próxima ao solo

Prof. Dr. Marcelo Becker – SEM – EESC – USP

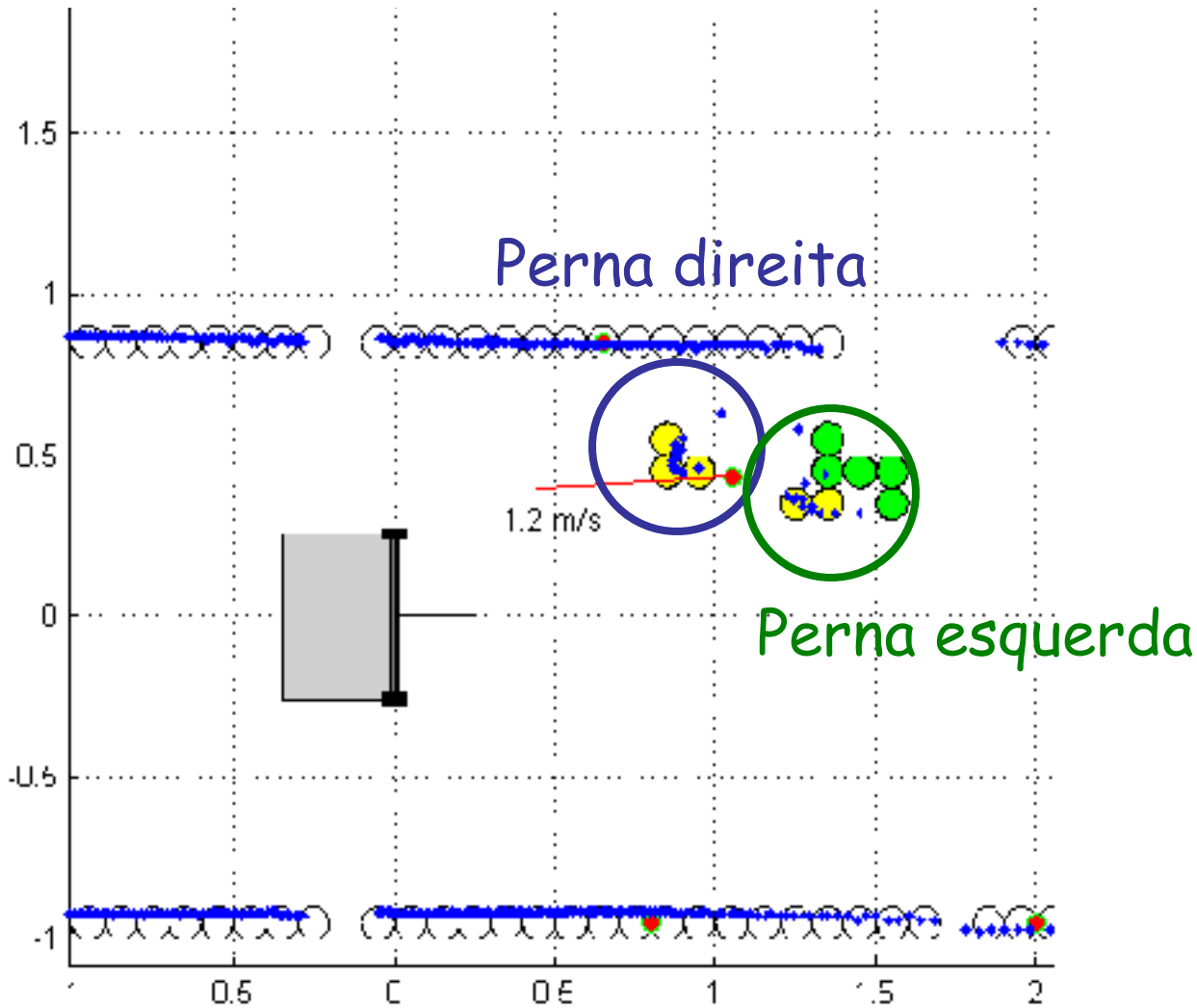

Perna esquerda
Perna direita
0.86 m/s
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0

Detecção e Monitoramento de Obstáculos

Resultados Experimentais

- robô em movimento em corredor estreito
  - sensor em posição próxima ao solo

Prof. Dr. Marcelo Becker - SEM - EESC - USP


Perna direita
Perna esquerda
1.2 m/s
1.5
1
0.5
0
-0.5
-1
0.5
0
0.5
1
1.5
2

ASL #1

Robox

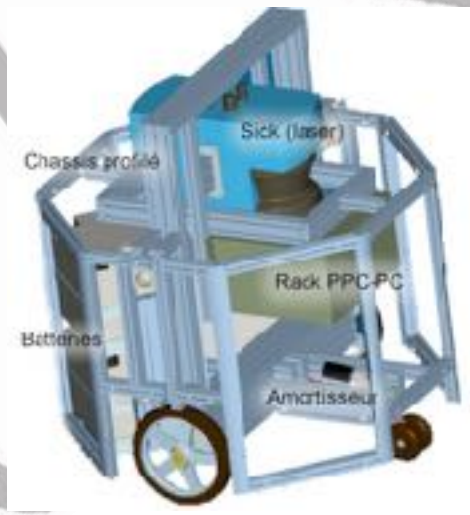
Agosto 2005 – Julho 2006

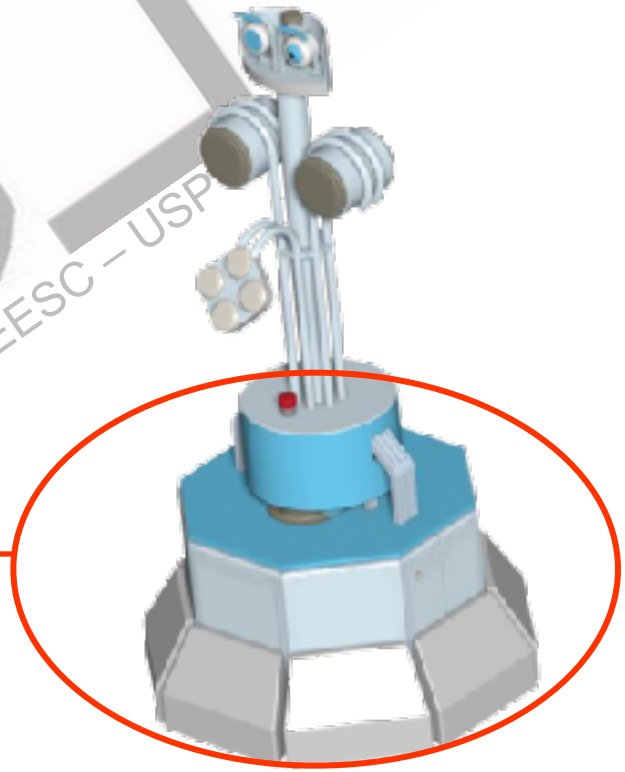
ASL - EPFL

Prof. Dr. Marcelo Becker - SEM – EESC – USP

ASL - EPFL

Obstacle Avoidance Procedures

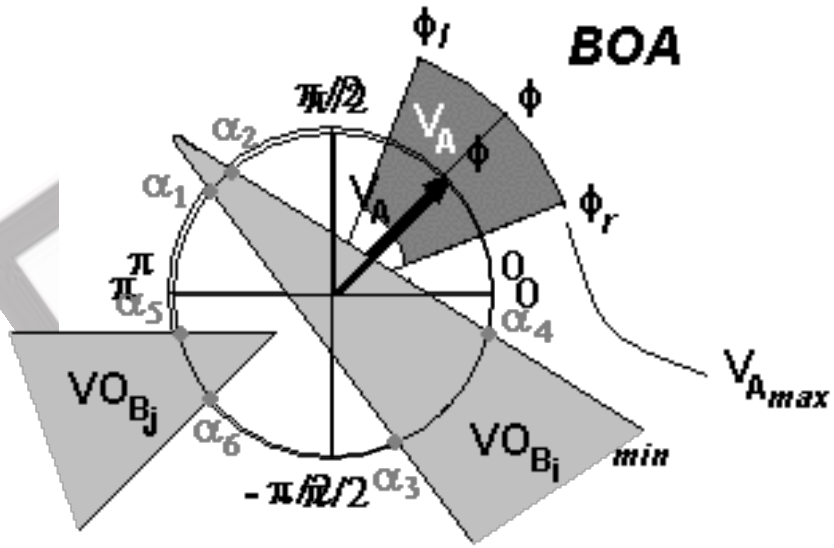


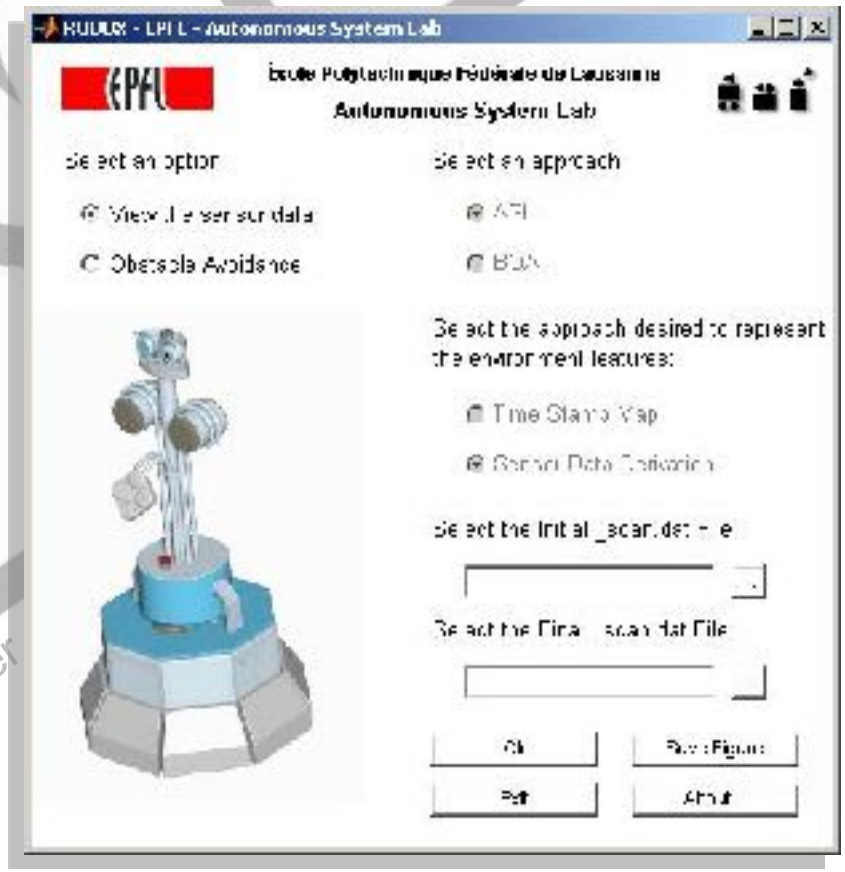


Becker - SEM – EESC – USP

ASL - EPFL

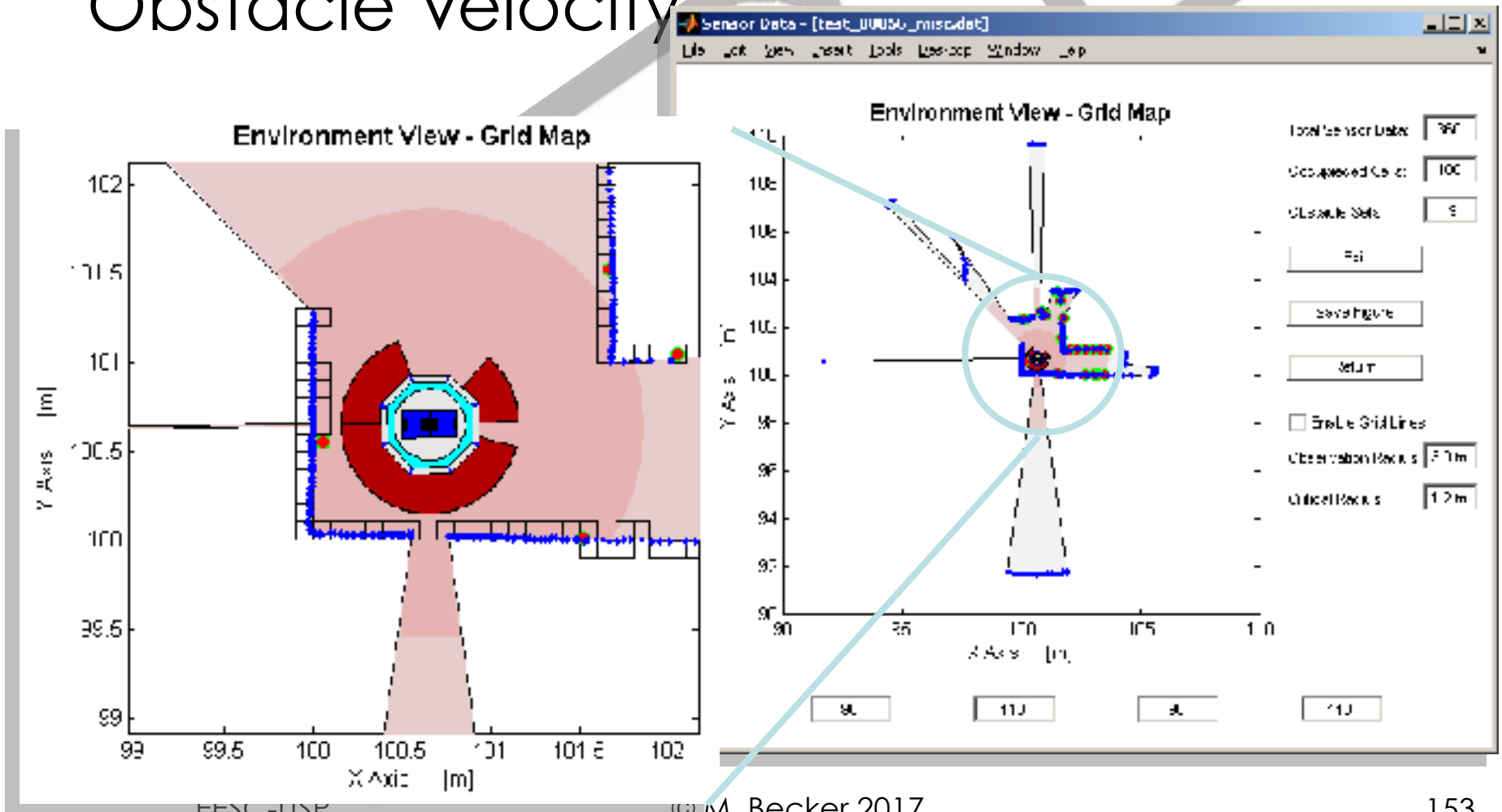
Obstacle Velocity





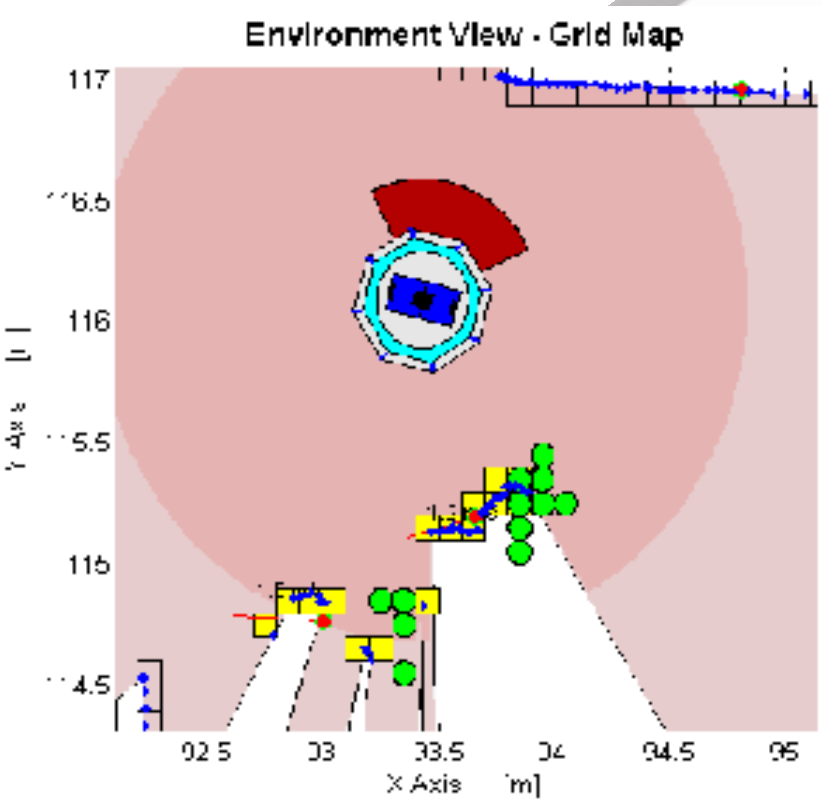
ASL - EPFL

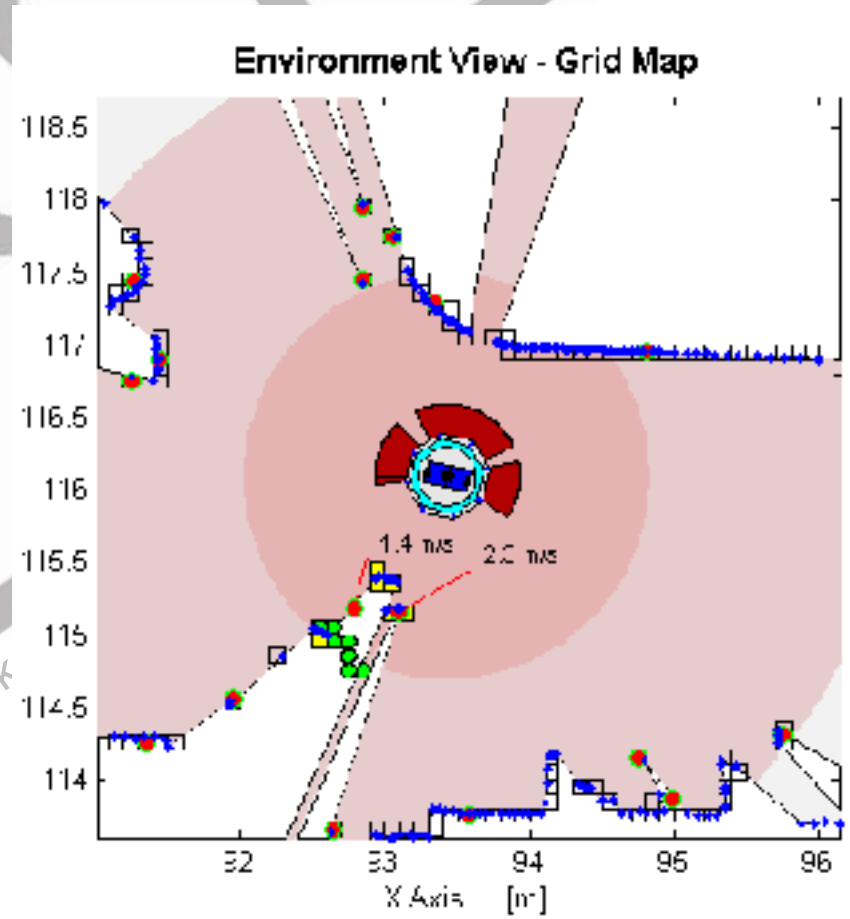
Obstacle Velocity



ASL - EPFL

Obstacle Velocity





© M. Becker 2017

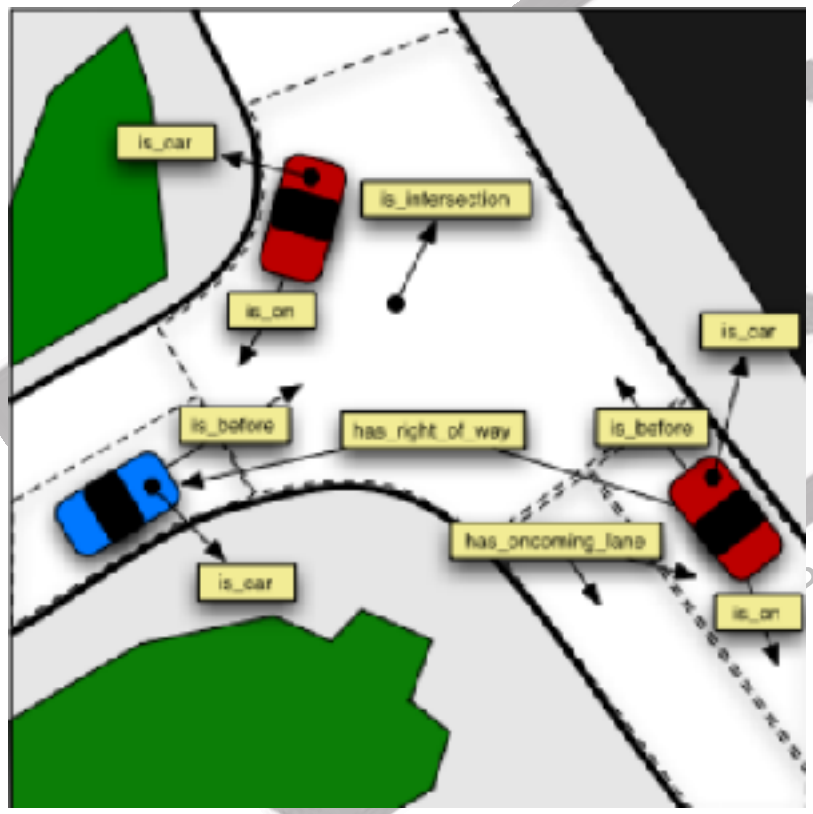
Desafios...

- Garantia de Solução Ótima
  - Busca Determinística
- Planejamento Global e Local
  - *Framework* unificado: um único planejador
- Tempo de Execução Baixo – *Real Time*
  - i.e.: < 0,1[s]
- Ambientes onde a interação com seres humanos é muito próxima...

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Desafios...

- Ambientes urbanos...



© M. Becker 2017

Referências

•Borenstein, J. and Koren, Y., 1991, “The vector field histogram – fast obstacle avoidance for mobile robots”, IEEE Journal of Robotics and Automation, vol. 7, n. 3, pp. 278-288.
•Brock, O. and Khatib, O., 1999, “High speed navigation using the global dynamic window approach”, Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, MI, pp. 341-346.
•Fox, D., Burgard, W. and Thrun, S., 1997, “The Dynamic Window Approach to Collision Avoidance”, IEEE Robotics & Automation Magazine, March 1997, pp. 23-33.
•Fiorioni, P. and Shiller, Z., 1993, “Motion Planning in Dynamic Environments using the Relative Velocity Paradigm”, Proceedings of the 1993 IEEE International Conference on Robotics & Automation, Atlanta.
•Movarec, H.P. and Elfes, A., 1985, “High Resolution Maps from Wide Angle Sonar”, IEEE Conference on Robotics and Automation, Washington D.C., pp. 116-121.
•Prassler, E., Scholz, J., Strobel, M. and Fiorini, P., 1999, “An Intelligent (Semi-) Autonomous Passenger Transportation System”, IEEE, ISBN 0-7803-4975-X, pp. 374-379.
•Siegwart, R. and Nourbakhsh, I.R., 2004, “Introduction to Autonomous Mobile Robots”, ISBN 0-262-19502-X, 331 pp.
•Dijkstra, E.W., A note on two problems in connexion with graphs. In: Numerische Mathematik, 1 (1959), pp. 269-271, 1959.

Prof. Dr. Marcelo Becker SEM – EESC – USP

Referências

•Ulrich, I. and Borenstein, J., 1998, “VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots”, Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Leuven, Belgium, May 1998, pp. 1572-1577.

•Ulrich, I. and Borenstein, J., 2000, “VFH*: Local Obstacle Avoidance with look-ahead Verification”, Proceedings of the 2000 IEEE International Conference on Robotics & Automation, San Francisco, CA, April 2000.

•Russell, S., Norvig, P., Artificial Intelligence, a Modern Approach. Prentice Hall International, 1995.

•Latombe, J.-C., Robot Motion Planning. Norwood, MA, Kluwer Academic Publishers, 1991.

•Philippsen, R., Siegwart, R., “Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003), Taipei, Taiwan, 2003.

•Simmons, R., “The Curvature Velocity Method for Local Obstacle Avoidance,” in Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, April 1996.

•Ko, N.Y., Simmons, R., “The Lane-Curvature Method for Local Obstacle Avoidance,” in Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’98), Victoria, B.C., Canada, October 1998.

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Referências

•Lumelsky, V., Stepanov, A., “Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape,” in Cox, I.J., Wilfong, G.T. (editors), Autonomous Robot Vehicles. New York, Spinger-Verlag, 1990.

•Khatib, M., Chatila, R., “An Extended Potential Field Approach for Mobile Robot Sensor-Based Motions,” in Proceedings of the Intelligent Autonomous Systems IAS-4, IOS Press, Karlsruhe, Germany, March 1995, pp. 490–496.

•Becker, Marcelo ; HALL, Richard ; KOLSKI, Sascha ; Macek, Kristijan ; SIEGWART, Roland ; Jensen, Björn . 2D laser-based probabilistic motion tracking in urban-like environments. Journal of the Brazilian Society of Mechanical Sciences and Engineering (Impresso), v. 31, p. 83-96, 2009.

•Becker, Marcelo. Aplicação de Técnicas de Controle em Cadeiras de Rodas Inteligentes. PhD. Thesis, FEM – Unicamp, 150p., 2000.

•J. J. Leonard and H. F. Durrant-Whyte. Directed Sonar Sensing for Mobile Robot Navigation. Boston: Kluwer Academic Publishers, 1992.

•Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., and Stein, Clifford. *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 22.3: Depth-first search, pp. 540–549.

•Knuth, Donald E., *The Art Of Computer Programming Vol 1. 3rd ed.*, Boston: Addison-Wesley, ISBN 0-201-89683-4, 1997.

Prof. Dr. Marcelo Becker - SEM - EESC - USP

Referências

SITES

- http://www.mobilerobots.org
- http://www.sri.com/news/storykits/
- http://groups.csail.mit.edu/lbr/genghis/
- http://www.stanford.edu/~learnest/cart.htm
- http://www.asl.ethz.ch
- http://www.euron.org/
- http://www.bluebotics.com/
- http://www.inuktun.com/
- http://www.service-robots.org
- http://world.honda.com/ASIMO/history/
- http://world.honda.com/HDTV/ASIMO/
- http://robotics.usc.edu/~avatar/

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Referências

SITES

- http://www.e-puck.org/
- http://www.swarm-bots.org
- http://brl.ee.washington.edu/Research_Past/Biologically_Based/Biologically_Based_Index.html
- http://www.esa.int/home-ind/ESA-Article-fullArticle_par-17_1054042928063.html
- http://www.neurotechnology.neu.edu/latmaneuvering.html
- http://eap.jpl.nasa.gov
- http://www.ias.uwe.ac.uk/energy-autonomy.htm

Prof. Dr. Marcelo Becker – SEM – EESC – USP

Referências

SITES

- http://web.mit.edu/spotlight/archives/troody.html
- http://omconsults.net/Bot/Mecanum.htm
- http://www.airtrax.com/vehicles/sidewinder.html
- http://www.msl.ri.cmu.edu/projects/ballbot
- http://www.ai.mit.edu/projects/leglab/robots/3D_hopper/3D_hopper.html

Prof. Dr. Marcelo Becker - SEM – EESC – USP

Referências

NOTAS DE AULA

- **Siegwart, R.** (ETHZ - Suíça):
http://www.mobilerobots.org

- **Simões, A. S.** (UNESP - Brasil):
http://www.sorocaba.unesp.br/professor/assimoes/rm/index.html

- **Zufferey, J-C.** (EPFL - Suíça):
http://moodle.epfl.ch/course/view.php?id=261

Prof. Dr. Marcelo Becker - SEM - EESC – USP