

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO

Localização e Planejamento de Caminhos para
um Robô Humanoide e um Robô Escravo com
Redes.

André Mendes Santana

Orientador: Prof. Dr. Adelar do Adalino Dantas de Macedo

NATAL-RN, JULHO DE 2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Localização e Planejamento de Caminhos para um Robô Humanóide e um Robô Escravo com Rodas.

André Macêdo Santana

Orientador: Prof. Dr. Adelardo Adelino Dantas de Medeiros

Dissertação de Mestrado apresentada ao Programa de Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal-RN, julho de 2007

Divisão de Serviços Técnicos
Catalogação da Publicação na Fonte. UFRN / Biblioteca Central Zila Mamede

Santana, André Macedo.

Localização e planejamento de caminhos para um Robô humanóide e um Robô escravo com rodas / André Macedo Santana. – Natal, RN, 2007.
67 f. il.

Orientador : Adelardo Adelino Dantas de Medeiros.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica.

1. Robôs industriais - Dissertação. 2. Fusão sensorial – Dissertação. 3. Sistema robótico – Planejamento de caminho – Dissertação. I. Medeiros, Adelardo Adelino Dantas de. II. Título.

RN/UF/BCZM

CDU 621.865.8(043.3)

**Localização e Planejamento de Caminhos para
um Robô Humanóide e um Robô Escravo com
rodas.
Programa de Pós-Graduação em Engenharia
Elétrica da UFRN.**

André Macêdo Santana

Dissertação de Mestrado aprovada em 10 de julho de 2007 pela banca examinadora composta pelos seguintes membros:

Dr. Adelardo Adelino Dantas de Medeiros (orientador) DCA/UFRN

Dr^a Silvia Silva da Costa Botelho (examinadora externa) FURG

Dr. Adrião Duarte Dória Neto (examinador interno) DCA/UFRN

Dr. Diogo Pinheiro Fernandes Pedrosa (examinador interno) DCA/UFRN

Dr. Pablo Javier Alsina (examinador interno) DCA/UFRN

*A vida só pode ser compreendida
olhando-se para trás; mas só pode
ser vivida olhando-se para a frente.*

Agradecimentos

O primeiro gesto é de agradecimento a Deus - do qual todas as coisas dependem - e que, com a sua infinita bondade, concedeu-me a oportunidade de, na aspérrima caminhada, experimentar a inefável sensação de galgar um patamar a mais.

Depois, o agradecimento aos meus pais que, na segurança de um lar bem estruturado, plasmaram a minha personalidade e fincaram o balizamento do meu caráter.

Logo a seguir, acode-me a imagem querida da minha doce companheira, Carol, sempre acreditando em mim, impelindo-me a constantes buscas de aperfeiçoamento, inculcando-me o salutar desejo de superar as minhas próprias e humanas limitações.

Minhas irmãs, Janaína e Sâmia também são peças fundamentais para esta conquista.

É impossível não agradecer a pessoa do professor Adelardo Medeiros, meu orientador, pelos conselhos, ensinamentos (em especial a capacidade de abstrair problemas complexos em coisas simplórias) e confiança no meu trabalho.

Muitas outras pessoas também merecem meus agradecimentos: meus amigos do convívio diário Kelson Aires, Vinicius Machado, Raimundo Moura, Érico Leão; professor Pablo Alsina que costumo dizer que foi meu "co-orientador"; professor Affonso Guedes pelos incentivos e longos "papos". Enfim, a todos os colegas do "LAR" que contribuíram direta ou indiretamente para esta conquista.

Obrigado a todos!!

Resumo

Esse trabalho apresentará os sistemas de localização e planejamento de caminho para um sistema robótico formado por um humanóide não instrumentado e um robô escravo com rodas. O objetivo do sistema é efetuar a navegação do humanóide, que não possui sensores mas que pode ser remotamente controlado por infra-vermelhos, utilizando um robô escravo com rodas. O robô com rodas deverá se posicionar atrás do humanóide e, através da imagem, estabelecer o posicionamento relativo do humanóide em relação a ele.

A localização do robô com rodas será obtida fundindo informações de odometria e detecção de marcos utilizando o Filtro de Kalman Extendido. A posição relativa do humanóide será encontrada a partir da fusão da pose do robô com rodas juntamente com as características do marco fixado nas costas do humanóide utilizando outro Filtro de Kalman. Sabendo a posição do robô com rodas e a posição relativa do humanóide em relação a ele tem-se a posição absoluta do humanóide.

O planejador de caminho foi desenvolvido de forma a proporcionar a movimentação cooperativa dos dois robôs incorporando as restrições de visibilidade existente do sistema robótico.

Palavras-chave: *Localização, Planejamento de Caminho, Fusão Sensorial.*

Abstract

This work presents the localization and path planning systems for two robots: a non-instrumented humanoid and a slave wheeled robot.

The localization of wheeled robot is made using odometry information and landmark detection. These informations are fused using a Extended Kalman Filter. The relative position of humanoid is acquired fusing (using another Kalman Filter) the wheeled robot pose with the characteristics of the landmark on the back of humanoid. Knowing the wheeled robot position and the humanoid relative position in relation to it, we acquired the absolute position of humanoid.

The path planning system was developed to provide the cooperative movement of the two robots, incorporating the visibility restrictions of the robotic system.

Keywords: *Localization, Path Planning, Sensor Fusion.*

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Algoritmos	v
1 Introdução	1
1.1 Motivação	3
1.2 Caracterização do problema: visão geral	3
1.3 Objetivos	4
1.4 Trabalhos Relacionados	4
1.5 Organização do trabalho	7
2 Localização de Robôs Móveis	9
2.1 Fundamentação teórica	9
2.1.1 Localização baseada em Odometria	9
2.1.2 Localização baseada em Marcos	11
2.1.3 Filtro de Kalman	12
2.2 Caracterização do problema	15
2.2.1 Localização absoluta do robô com rodas	15
2.2.2 Localização relativa do humanóide	16
2.3 Solução proposta	17
2.3.1 Localização do robô com rodas	17
2.3.2 Localização relativa do humanóide	23
2.4 Considerações	30
3 Planejamento de Caminho	31
3.1 Técnicas tradicionais de planejamento de caminho	31
3.1.1 Espaço de Configuração	31
3.1.2 Mapa de rotas - <i>roadmaps</i>	32
3.1.3 Campo de Potencial	34
3.1.4 Decomposição em Células	35
3.2 Caracterização do problema	37
3.3 Solução proposta	40

4	Resultados	45
4.1	Localização absoluta do robô com rodas	45
4.2	Localização relativa do humanoíde	49
4.3	Planejamento de caminho	52
4.3.1	Considerações iniciais	52
4.3.2	Cenário 1	53
4.3.3	Cenário 2	53
5	Conclusões e Trabalhos Futuros	57
	Referências bibliográficas	59

Lista de Figuras

1.1	Arquitetura de navegação de um sistema robótico.	2
1.2	Robôs utilizados.	3
2.1	Odômetro de Leonardo Da Vinci	10
2.2	Deslocamento de um robô medido por odometria.	10
2.3	Regiões de incerteza durante o movimento de um robô.	11
2.4	Localização usando marcos.	12
2.5	Robô com rodas.	15
2.6	Correção do erro de odometria.	16
2.7	Robô humanóide.	16
2.8	Modelagem do sistema de localização absoluta para o robô com rodas. . .	17
2.9	Variáveis do modelo geométrico.	18
2.10	Parâmetros de Hough.	22
2.11	Sistema de coordenadas.	22
2.12	Modelagem do sistema de localização relativa do humanóide.	24
2.13	Sistemas de Coordenadas dos robôs.	26
2.14	Informações sensoriais utilizadas.	26
2.15	Modelagem do sensor.	27
2.16	Transformações de eixos.	28
3.1	(a)Configuração do robô (b)Espaço de Configuração.	32
3.2	Modificações incorporadas às dimensões originais do obstáculo.	33
3.3	Grafo de Visibilidade.	33
3.4	Diagrama de Voronoi.	34
3.5	Campo de Potencial.	34
3.6	Decomposicao em células exatas.	35
3.7	Decomposicao em células aproximadas.	36
3.8	Sistema Robótico.	37
3.9	Imagens do robô humanóide.	38
3.10	Imagem do robô humanóide.	39
3.11	Imagem do robô humanóide.	39
3.12	Imagens do robô humanóide.	40
3.13	Restrições do Sistema.	40
3.14	Restrições do Sistema.	41
3.15	Exemplo do planejador.	43
3.16	Exemplo do planejador.	44

4.1	Resultados dos sistemas de localização implementados.	46
4.2	Gráfico comparativo do sistema de localização por correção geométrica com o sistema de localização utilizando filtro de Kalman.	47
4.3	Efeitos da perturbação nos sistemas de localização.	48
4.4	Localização Relativa com a câmera parada.	49
4.5	Localização Relativa com a câmera em movimento.	50
4.6	Comportamento do erro nos sistemas de localização relativa.	51
4.7	Mapa utilizado no planejamento de caminho.	53
4.8	Caminho planejado: cenário 1.	54
4.9	Caminho planejado: cenário 1.	54
4.10	Caminho planejado: cenário 2.	55
4.11	Caminho planejado: cenário 2.	55

Lista de Algoritmos

1	<i>Filtro de Kalman Discreto.</i>	13
2	<i>Filtro de Kalman Extendido.</i>	14

Capítulo 1

Introdução

O desenvolvimento tecnológico vem proporcionando um significativo aumento de qualidade de vida para os humanos. Um exemplo clássico desses avanços são os robôs. Estas máquinas estão sendo cada vez mais empregadas para auxiliar ou substituir o homem em diversas tarefas, tais como transporte de cargas, limpeza e manutenção de ambientes, segurança, diversão, exploração de ambientes perigosos ou nocivos ao ser humano, em tarefas repetitivas e em tarefas de precisão [Silva 2003].

De uma forma mais ampla, os robôs podem ser definidos como máquinas que, além de serem capazes de reproduzir tarefas associadas aos seres humanos e movimentos implícitos em sua construção, complementam a parte mecânica com dispositivos eletrônicos inteligentes de suporte, os quais lhes dão um certo grau de autonomia [Medeiros 1998a].

Segundo Craig (2004), os robôs são classificados em duas grandes classes: robôs manipuladores e robôs móveis. Os manipuladores robóticos têm forte atuação nas indústrias, sobretudo na automobilística. A classe de robôs móveis é constituída por robôs que tem a capacidade de locomoção no espaço tridimensional (robôs aereoespaciais e robôs subaquáticos) ou no espaço planar (robôs terrestres). Os robôs terrestres, por sua vez, possuem variações: robôs com pernas (*legged robots*), destaque para os humanóides, e robôs com rodas (*wheeled robots*).

As aplicações envolvendo robôs móveis vêm crescendo significativamente nos últimos anos devido a uma classe de tarefas que esses podem realizar, já que, ao contrário dos manipuladores, possuem capacidade de se locomover livremente pelo espaço de trabalho, limitados apenas por eventuais obstáculos.

Comparando robôs humanóides e robôs com rodas percebe-se que os robôs móveis com rodas são mais simples de serem construídos e fáceis de controlar. As rodas permitem uma maior praticidade de locomoção em ambientes planos e dão um maior suporte estático ao robô e o simples acionamento dos motores faz com que ele se mova. Já para robôs humanóides, a movimentação das pernas deve ser feita de forma sincronizada com a preocupação adicional de não fazê-lo perder o equilíbrio e cair.

Um fator importante para o bom desempenho de um sistema robótico é torná-lo autônomo. Segundo Medeiros (1998b), robôs autônomos devem apresentar reações às mudanças do ambiente, comportamento inteligente, integração entre vários sensores, resolução de múltiplas tarefas, robustez, operar sem falhas, programabilidade, modularidade, flexibilidade, expansibilidade, adaptabilidade e raciocínio global.

No contexto da execução de tarefas de forma autônoma por um robô destaca-se o problema da navegação. Ela pode ser definida como os procedimentos necessários para permitir que um sistema robótico saia de uma configuração inicial e alcance uma configuração final percorrendo um determinado caminho planejado, desviando de obstáculos quando existirem, obedecendo a restrição temporal imposta para o cumprimento dessa tarefa.

Alsina et al. (2002) dividem o processo de navegação em cinco níveis hierárquicos: modelagem do ambiente, localização, geração de caminho, geração de trajetória e execução da trajetória. A figura 1.1 ilustra a arquitetura.



Figura 1.1: Arquitetura de navegação de um sistema robótico.

- **Modelagem do ambiente:** neste nível o robô utiliza sensores para coletar informações e criar um modelo com as principais características estruturais do ambiente;
- **Localização:** é determinada a posição e orientação do robô no espaço, com base no modelo obtido no nível de modelagem do ambiente;
- **Geração de caminho:** o robô calcula a curva geométrica que o leva de sua localização inicial a sua localização final, desviando dos obstáculos, quando existirem;
- **Geração de trajetória:** o caminho calculado no nível anterior é adaptado às restrições temporais impostas e a partir daí são calculadas as velocidades que devem ser aplicadas ao mesmo para que possa percorrer o caminho no intervalo de tempo especificado;
- **Execução de trajetória:** neste nível, os atuadores do robô são controlados de forma que a trajetória percorrida por ele seja a mais próxima possível da calculada no nível anterior.

Dentre os cinco níveis hierárquicos que compõem o processo de navegação este trabalho tratará especificamente de dois: Localização e Geração de caminho.

1.1 Motivação

A pesquisa em robôs humanóides é de muito interesse para a comunidade científica, pois esses robôs podem ser facilmente utilizados nos ambientes humanos. Muitos laboratórios do mundo que desenvolvem pesquisas em robótica não têm acesso a robôs humanóides reais. Isso se deve, principalmente, ao fato de que sua construção é muito difícil, demorada e possui um custo elevado.

A motivação principal deste trabalho consiste em tentar operar de forma cooperativa um robô humanóide e um robô com rodas de forma a permitir utilizar, para fins de pesquisa, um robô humanóide não instrumentado. Isto será obtido utilizando-se um robô com rodas para complementar o sensoramento necessário para a navegação do humanóide.

O sistema desenvolvido é baseado em duas plataformas: um robô móvel com rodas, denominado *Karel*, que funciona como uma plataforma móvel para um notebook e possui dois motores DC com *encoders* acoplados a seus eixos, além de câmera, e um robô humanóide *Robosapiens* da empresa *Wow Wee* [Anônimo 2007]. Este robô é acionado por controle remoto e possui quatro funções básicas de movimentação: andar para frente, andar para trás, girar para a direita e girar para a esquerda. A figura 1.2 mostra os dois robôs que compõem o sistema robótico utilizado neste trabalho.

(a) *Karel*(b) *Robosapien*

Figura 1.2: Robôs utilizados.

1.2 Caracterização do problema: visão geral

Como o apresentado no início deste capítulo, este trabalho aborda as etapas de Localização e Planejamento de caminho no contexto de navegação robótica.

O sistema robótico utilizado é formado por um robô humanóide não instrumentado mas que pode ser remotamente controlado por um elo de comunicação de raios infravermelhos e um robô escravo com rodas dotado de *encoders* e câmera. Como o humanóide não possui nenhum tipo de sensoramento que possa fornecer sua pose (posição e orientação) fez-se necessário a utilização de técnicas de localização relativa para calcular sua pose em relação a uma câmera. Câmera essa que encontra-se acoplada na estrutura do robô escravo com rodas.

Durante a movimentação do conjunto é necessário que o robô com rodas se posicione atrás do humanóide e, através da imagem capturada por uma câmera embarcada na sua estrutura, estabeleça o posicionamento relativo do humanóide em relação a ele. Sendo conhecida sua própria posição e a posição relativa do humanóide, tem-se a posição absoluta do humanóide. Conhecida a posição do humanóide e os seus movimentos pode-se planejar um caminho que possa ser executado pelo conjunto de forma que o robô com rodas não perca de vista o rótulo existente no humanóide para sua identificação.

É importante destacar que para executar a movimentação conjunta dos robôs muitos problemas intermediários devem ser considerados. Dentre eles podem-se destacar a localização do robô com rodas, a localização relativa do humanóide em relação ao robô com rodas e o planejamento de caminho cooperativo.

1.3 Objetivos

Objetivo geral:

O objetivo deste trabalho é desenvolver, no contexto de navegação, os módulos de Localização e Planejamento de caminho para um sistema robótico formado por um humanóide não instrumentado e um robô escravo com rodas.

Objetivos específicos:

- desenvolver um sistema de localização eficiente para o robô com rodas utilizando as medidas de sua posição relativa (obtida por odometria) e de sua posição absoluta (obtida pela detecção de marcos naturais) e fundi-las utilizando Filtro de Kalman Estendido para que a pose encontrada seja a mais precisa possível;
- desenvolver um sistema de localização relativa para o robô humanóide utilizando a câmera embarcada no robô com rodas. Para tal, a pose do robô com rodas (já filtrada) juntamente com as características extraídas do rótulo são fundidas em um filtro de Kalman Estendido a fim de obter a pose relativa do humanóide;
- desenvolver um planejador de caminho para o sistema humanóide e robô com rodas de forma que o robô com rodas sempre se posicione atrás do humanóide e seja capaz de capturar a imagem do rótulo existente no humanóide.

1.4 Trabalhos Relacionados

A literatura mostra muitos trabalhos relacionados ao problema da localização de robôs e planejamento de caminhos.

Kelly (2001) descreveu analiticamente como os erros de odometria se propagam em determinados modelos cinemáticos. Ele propõe a discretização e linearização do conjunto de equações diferenciais não-lineares do modelo de odometria e representa analiticamente

a relação existente entre o distúrbio ocorrido nas variáveis de entrada (velocidades medidas pelos sensores) e o distúrbio nas variáveis de saída (posição e orientação do veículo).

Jang et al. (2002) utilizaram como marco natural a figura de um quadrado dividido horizontalmente em dois retângulos coloridos. Após a detecção do marco na imagem, estando o mesmo em uma posição conhecida no ambiente, a posição do robô era determinada com base na posição da câmera com relação ao marco. De forma semelhante, Fairfield (2002) utilizou, como marcos naturais, pequenos anéis de plástico dispostos pelo ambiente de trabalho do robô e Launay et al. (2002) usaram as lâmpadas do teto de um corredor para localizar o robô.

Utilizar uma câmera externa e determinar, na imagem, a forma geométrica do robô é o trabalho apresentado por Jim et al. (2003). Bezerra (2004) utiliza como marcos naturais as linhas do chão que compõem o ambiente e identifica-as utilizando transformada de Hough. Ele propõe reduzir o custo da transformada aplicando-a somente a parte da imagem que possui maior probabilidade de conter o marco.

Odakura et al. (2004) executam a localização do robô utilizando filtro de Kalman com observações parciais. A cada etapa em que são lidas as informações dos sensores elas são avaliadas e é verificado a correspondência da medida com a coordenada da pose do robô que ela se relaciona e esta medida é utilizada para corrigir somente a coordenada com a qual se relaciona. Kiriy & Buehler (2002) mostram no seu trabalho a localização de um robô móvel para seguir diversos marcadores dispostos no ambiente também utilizando filtro de Kalman.

Como trabalhos mais recentes temos o de Borges et al. (2005), que usa uma câmera e um sensor de profundidade baseado em laser para resolver o problema de localização e mapeamento local, o de Saedan et al. (2006) que utiliza visão omni-direcional para localizar o robô e o de Xia et al. (2006) que corrige a localização de um veículo autônomo observando as faixas de delimitação em ruas.

Sobre planejamento de caminho. Trabalhos mais antigos como os de Choset & Burdick (1995) e Díaz de León & Sossa (1998) utilizam o Diagrama de Voronói. Soares & Alsina (2000) apresentam resultados obtidos em um estudo de caso do campo de potencial em ambientes de futebol de robôs. Já Chen & Zalzal (1997) utilizam como método de planejamento de caminho a técnica de decomposição em células.

Pedrosa (2001) propôs uma estratégia de geração de caminho baseada em polinômios paramétricos de terceiro grau em x e y , e a orientação θ do robô é obtida através da relação entre y e x de modo que os caminhos gerados respeitem as restrições holonômicas do robô.

Pimenta et al. (2004) apresentam uma proposta de geração de trajetória para robôs móveis utilizando métodos de elementos finitos. Nele, O espaço de trabalho do robô é traduzido em condições de contorno para a equação de Laplace. A equação de Laplace é resolvida utilizando métodos de elementos finitos e resulta num valor de potencial para cada nó da malha. A partir desses potenciais dos calcula-se um valor de gradiente (campo potencial) para cada elemento e utiliza-os como uma função de potencial para navegação de robôs.

Baseado em mapa de rotas probabilístico Adôrno et al. (2005) apresentam uma estratégia de geração de trajetória usando um interpolador baseado em curvas de Bezier.

No tocante a planejamento de caminho para humanóides, Hoff et al. (2000) descrevem uma técnica de movimentação, baseada no diagrama de Voronói, considerando três tipos distintos de movimento ao percorrer um caminho: caminhar normal, caminhar girando o torso quando houver obstáculos à parte superior do corpo e caminhar de lado em passagens estreitas.

Shao & Terzopoulos (2005) propõem um sistema de controle de pedestres para permitir que grandes ambientes virtuais sejam povoados com humanóides autônomos. Este sistema é baseado em uma estrutura de dados que agrega informações de navegação a informação geométrica do ambiente.

Durante a pesquisa literária áreas como coordenação de robôs, formação, frota, seguimento de líder e trailer foram avaliadas [Lamiroux et al. 2006, Ghabcheloo et al. 2005, Fujimori et al. 2005].

Em Gava et al. (2006) e de la Cruz et al. (2006) é proposta uma estratégia de cooperação centralizada baseada em um sistema de visão omini-direcional, onde o líder é responsável pela geração do caminho e coordenação das tarefas.

Em Carelli et al. (2006), a tarefa de seguimento de líder é realizada por meio de um sistema de visão artificial com uma câmera perspectiva, com a qual o robô seguidor obtém as informações de posição e orientação do robô líder.

Divelbiss & Wen (1992) apresentam a solução para a navegação de um trailer calculando *a priori* o caminho a ser executado pelo móvel e posteriormente recalcula-o de forma a incorporar as informações não-holonômicas. Posteriormente, os mesmos Divelbiss & Wen (1997) propõem uma estratégia para estacionar um trailer em uma garagem. Ainda na mesma época, Lamiroux & Laumond (1998) mostram uma estratégia que combina laços de controle de forma a conduzir um robô com reboque a uma configuração final.

Han et al. (2004) analisam as trajetórias quando o sistema robô-reboque segue alguns tipos representativos dos trajetos e propõe o conceito "tamanho-global" a ser utilizado na representação dos obstáculos no espaço de trabalho. Baseado neste conceito, um método que modela o ambiente é proposto. Han et al. (2004) afirmam que o algoritmo melhora o método computacional de expandir o tamanho dos obstáculos e reduz a perda do trajeto praticável no espaço livre.

Baseadas no método de Lyapunov, Yuan, Huang & Han (2004) desenvolveram uma estratégia simples de controle para robôs com rodas. O algoritmo proposto tem por idéia chave controlar primeiro o reboque para seguir o trajeto desejado a fim manter-se próximo do mestre proporcionando assim que o sistema inteiro diminua a probabilidade de colisão.

Woodley & Acar (2004) implementaram uma estratégia de controle hierarquizada em cinco níveis para controlar um robô com reboque. Os níveis são tratados como agentes inteligentes. O primeiro é responsável por calcular um caminho inicial, o segundo por ajustar o trajeto (desvio de obstáculos), o terceiro analisa as restrições do sistema e o último é responsável por calcular os movimentos.

Yuan, Huang & Sun (2004) tratam do projeto estrutural-físico para uma estrutura variável TTMR (*tractor-trailer mobile robot*) baseados nas características das trajetórias de-

envolvidas pelo robô e pelo reboque.

Liu et al. (2006) planejam o caminho utilizando como métricas o comprimento e a largura do trajeto. Para tal, baseiam-se em um mapa de circuitos e a corrente é o resultado do trajeto planejado.

Um trabalho interessante é apresentado em Lamiroux et al. (2005). Este trabalho trata de um acordo entre a empresa Airbus e o governo francês a fim de transportar, via terrestre, os componentes básicos do avião Airbus 380 (asas, a parte traseira, central, fuselagens dianteiras, e o plano da cauda horizontal). Para tal, dois tipos de caminhão reboque foram projetados para o transporte das peças.

A equipe de pesquisadores do LAAS-CNRS ficou responsável por gerar o planejamento do movimento dos caminhões, baseado em técnicas de navegação robótica. A estratégia utilizada por eles consiste em gerar, *a priori*, uma trajetória padrão e em seguida "deformá-la", usando funções de potencial, de modo a incorporar as restrições não holonômicas do movimento dos caminhões e garantir que o trajeto seja o mais seguro possível.

1.5 Organização do trabalho

Este trabalho é composto de mais quatro capítulos:

- O Capítulo 2, que detalha os sistemas de localização absoluta (robô com rodas) e relativa (humanóide);
- O Capítulo 3, que descreve o planejador de caminho proposto;
- O Capítulo 4 que apresenta os resultados obtido com o sistema proposto;
- O Capítulo 5, que mostra as conclusões e propostas de trabalhos futuros.

Capítulo 2

Localização de Robôs Móveis

Uma importante etapa no processo de navegação de robôs móveis autônomos é a tarefa de localização. A partir de um conhecimento acumulado sobre o ambiente e utilizando as leituras atuais dos sensores, o robô deve ser capaz de determinar e manter atualizada a sua posição e orientação em relação a este ambiente, mesmo que os sensores apresentem erro e/ou ruído. Em outras palavras, localizar um robô consiste em determinar sua pose (posição e orientação) no espaço de trabalho em um determinado instante de tempo.

Borenstein & Feng (1996) classificam os métodos de localização em duas grandes categorias: métodos de localização relativa, que dão a pose do robô em relação a uma pose inicial e métodos de localização absoluta, que dão a pose global do robô e não necessitam das posições calculadas anteriormente.

No caso de robôs com rodas é comum o uso de *encoders* acoplado aos eixos de rotação das rodas, na técnica de localização relativa denominada odometria [Borenstein & Feng 1996].

As técnicas de localização absoluta utilizam marcos para localizar o robô. Estes marcos podem ser artificiais, quando introduzidos no ambiente com a finalidade de auxiliar na localização do robô, ou naturais, quando já se encontram no ambiente. É importante ressaltar que mesmo as técnicas de localização absolutas são imprecisas devido aos ruídos oriundos dos sensores utilizados.

Com o intuito de obter a pose do robô com o menor erro possível é interessante filtrar as informações provindas dos seus sensores. Uma ferramenta matemática eficiente para realizar essa tarefa é o filtro de Kalman [Kalman 1960].

A estrutura deste capítulo foi organizada da seguinte forma: a Seção 2.1 mostra a fundamentação teórica básica das ferramentas utilizadas neste trabalho, a Seção 2.2 caracteriza o problema, a Seção 2.3 apresenta os sistemas de localização propostos e, por fim, a Seção 2.4 traz algumas considerações sobre os sistemas.

2.1 Fundamentação teórica

2.1.1 Localização baseada em Odometria

O conceito de odometria foi descrito por Vitruvius no século I ac. Leonardo da Vinci por volta do ano de 1500 construiu um artefato que possuía um conector cheio de pedras

e com um orifício na parte inferior fixo à estrutura de um veículo. A rotação de uma das rodas faz rodar um tambor em torno do conector. O tambor tem um orifício do mesmo tamanho do existente no contentor. Depois de um certo número de rotações da roda, os dois orifícios coincidem e uma pedra cai, sendo recolhida numa caixa. O número de pedras recolhidas na caixa, num intervalo de tempo, permite estimar a distância percorrida. A figura 2.1 mostra o odômetro de Da Vinci.

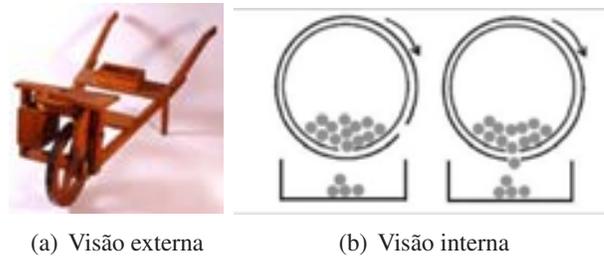


Figura 2.1: Odômetro de Leonardo Da Vinci

O método de odometria consiste na determinação da posição e orientação do robô através da integração dos deslocamentos incrementais de suas rodas, medidos a partir de um referencial fixo. Para exemplificar esse método, considere um robô se deslocando no seu ambiente seguindo uma trajetória reta, como mostrado na figura 2.2.

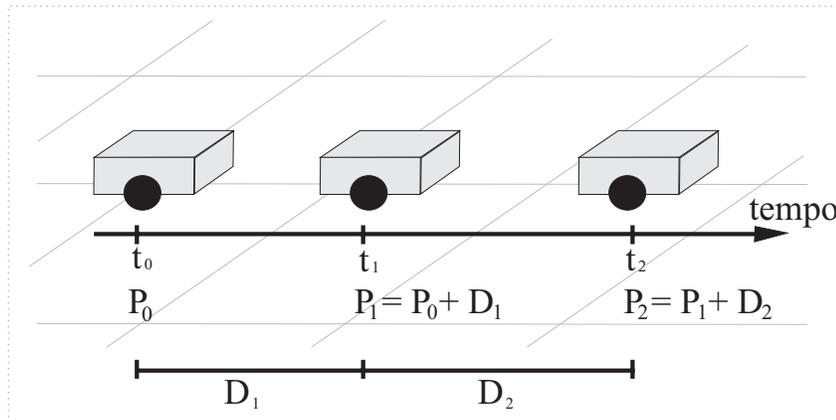


Figura 2.2: Deslocamento de um robô medido por odometria.

Percebe-se que a posição atual do robô pode ser calculada com base no acúmulo dos deslocamentos efetuados por ele, com relação a sua posição inicial.

Para medir o deslocamento do robô é necessário a utilização de um sensor que possa medir a rotação das suas rodas. Em aplicações envolvendo robôs móveis, um tipo de sensor de odometria bastante utilizado é o *encoder* óptico. Esse tipo de sensor funciona com base na transmissão e recepção de luz através de um disco perfurado, que gira acoplado ao eixo que movimenta a roda do robô.

A técnica de odometria está sujeita a erros que fazem com que ela forneça apenas uma estimativa da localização exata do robô em um dado instante de tempo [Latombe 1991].

Como a odometria determina a localização do robô com base no acúmulo das informações obtidas nos instantes anteriores, um erro em um determinado instante também compromete as medições dos instantes seguintes.

Os erros de odometria podem ser classificados como erros sistemáticos e não-sistemáticos. Os erros sistemáticos são aqueles causados por incertezas nos parâmetros do modelo cinemático do robô (comprimento do eixo diferente do nominal, desalinhamento das rodas, etc.). Já os erros não sistemáticos são imprevisíveis, causados por situações que surgem inesperadamente, como o escorregamento das rodas, imperfeições no solo e fenômenos similares.

Tonouchi et al. (1994) representam os erros de odometria por uma região elíptica ao redor da posição atual do robô. A figura 2.3 ilustra as regiões de incerteza.

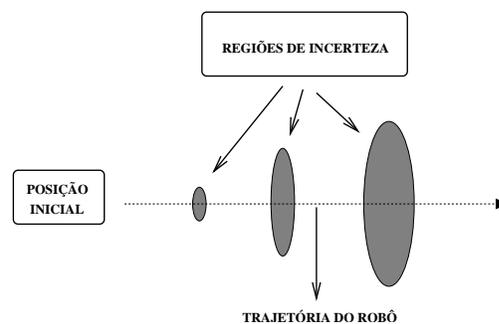


Figura 2.3: Regiões de incerteza durante o movimento de um robô.

Para corrigir os erros de odometria é possível a utilização de outros tipos de sensores, em conjunto com os odométricos, a fim de zerar periodicamente os erros odométricos. A estratégia abordada neste trabalho consiste em utilizar marcos naturais para corrigir os erros.

2.1.2 Localização baseada em Marcos

Borenstein & Feng (1996) definem marcos como elementos distintos que o robô é capaz de reconhecer através de seus sensores. Geralmente os marcos estão em posições fixas e conhecidas em relação ao referencial do ambiente. As informações dos sensores são utilizadas para reconhecer os marcos e determinar as suas posições com relação ao referencial do robô. Conhecendo a posição dos marcos com relação ao referencial do ambiente e com relação ao referencial do robô, é possível determinar a posição do robô com relação ao referencial do ambiente.

Além dos sensores, um sistema de localização baseado em marcos deve possuir um método para identificar os marcos no ambiente e um método para calcular a pose do robô a partir desses marcos. Diferentes métodos têm sido utilizados para realizar essas duas tarefas, que variam substancialmente dependendo das características do ambiente, dos sensores e dos marcos utilizados.

O problema da localização do robô através de marcos pode ser resumido através do relacionamento entre os referenciais do ambiente $\{W\}$, do robô $\{R\}$, da câmera $\{C\}$ e do

marco $\{M\}$. Esse relacionamento é ilustrado na figura 2.4. O objetivo é obter a localização do robô com relação ao ambiente ${}^W_R T$. A localização da câmera em relação ao robô ${}^R_C T$ é conhecida. A câmera é então utilizada para capturar uma imagem do marco, a partir da qual é determinada a localização da câmera em relação ao marco ${}^C_M T$. Sendo conhecida a localização do marco com relação ao ambiente, ${}^W_M T$ determina-se ${}^W_R T$ como sendo ${}^W_R T = {}^W_M T \cdot {}^M_C T \cdot {}^C_R T$.

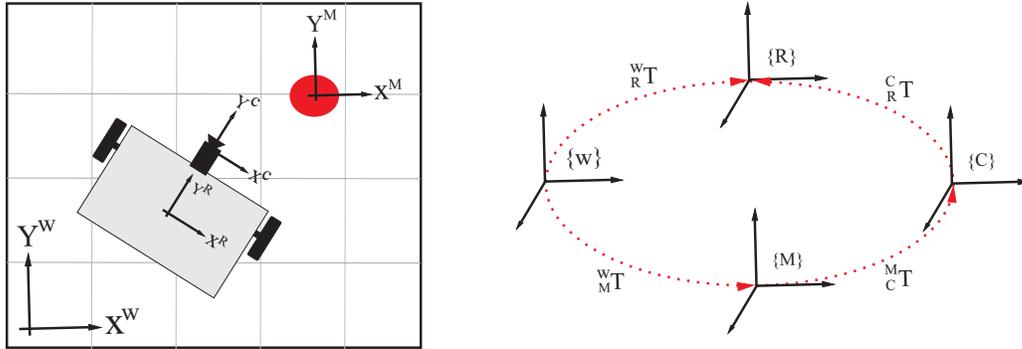


Figura 2.4: Localização usando marcos.

2.1.3 Filtro de Kalman

Em 1960, Rudolph Emil Kalman publicou um famoso artigo descrevendo um processo recursivo para solucionar problemas lineares relacionados à filtragem de dados discretos [Kalman 1960]. Sua pesquisa proporcionou contribuições relevantes ajudando a estabelecer bases teóricas sólidas em várias áreas da engenharia de sistemas.

Com o avanço computacional, o filtro de Kalman e suas extensões a problemas não lineares representam um produto largamente utilizado dentro das modernas teorias da engenharia. A seguir será descrito, de forma resumida, o filtro de Kalman aplicado a sistemas com modelo linear e não-linear.

Filtro de Kalman Discreto - FKD

Aiube et al. (2006) definem filtro de Kalman como um conjunto de equações matemáticas que constitui um processo recursivo eficiente de estimação, uma vez que o erro quadrático de estimação é minimizado. Através da observação da variável denominada “variável de observação” outra variável, não observável, denominada “variável de estado” pode ser estimada eficientemente.

A modelagem do Filtro de Kalman Discreto pressupõe que o sistema seja linear e descrito pelo modelo de equações do sistema 2.1:

$$\begin{cases} \mathbf{s}_t &= \mathbf{A}_t \mathbf{s}_{t-1} + \mathbf{B}_t \mathbf{u}_{t-1} + \gamma_t \\ \mathbf{z}_t &= \mathbf{C}_t \mathbf{s}_t + \delta_t \end{cases} \quad (2.1)$$

onde $\mathbf{s} \in R^n$ é o vetor de estados; $\mathbf{u} \in R^l$ é o vetor das entradas de controle; $\mathbf{z} \in R^m$ é o vetor de medições; a matriz $n \times n$, \mathbf{A} , é a matriz de transição de estados; \mathbf{B} , $n \times l$, é a matriz de coeficientes de entrada; a matriz \mathbf{C} , $m \times n$, é a matriz de observação; $\boldsymbol{\gamma} \in R^n$ representa o vetor de ruídos do processo e $\boldsymbol{\delta} \in R^m$ o vetor de erros de medição. Os índices t e $t - 1$ representam os instantes de tempo atual e anterior respectivamente.

O filtro opera em modo de predição-atualização levando em consideração as propriedades estatísticas do ruído. Um modelo interno do sistema é usado para atualização e um esquema de realimentação realiza as medições. As etapas de predição e atualização para o FGD podem ser descritas pelos sistemas 2.2 e 2.3 respectivamente.

$$\begin{cases} \bar{\boldsymbol{\mu}}_t &= \mathbf{A}_t \boldsymbol{\mu}_{t-1} + \mathbf{B}_t \mathbf{u}_{t-1} \\ \bar{\boldsymbol{\Sigma}}_t &= \mathbf{A}_t \boldsymbol{\Sigma}_{t-1} \mathbf{A}_t^T + \mathbf{R}_t \end{cases} \quad (2.2)$$

$$\begin{cases} \mathbf{K}_t &= \bar{\boldsymbol{\Sigma}}_t \mathbf{C}_t^T (\mathbf{C}_t \bar{\boldsymbol{\Sigma}}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1} \\ \boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t (\mathbf{z}_t - \mathbf{C}_t \bar{\boldsymbol{\mu}}_t) \\ \boldsymbol{\Sigma}_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\boldsymbol{\Sigma}}_t \end{cases} \quad (2.3)$$

O filtro de Kalman representa o vetor de estados \mathbf{s}_t no tempo t por sua média $\boldsymbol{\mu}_t$ e covariância $\boldsymbol{\Sigma}_t$. As matrizes \mathbf{R} , $n \times n$, e \mathbf{Q} , $l \times l$, são as matrizes de covariância dos ruídos de processo ($\boldsymbol{\gamma}$) e medição ($\boldsymbol{\delta}$) respectivamente e a matriz \mathbf{K} , $n \times m$, representa o ganho do sistema.

A partir da especificação matemática apresentada, um algoritmo para o FGD é apresentado a seguir no quadro 1. Maior detalhamento sobre o algoritmo bem como seu desenvolvimento matemático pode ser encontrado em Thrun et al. (2005).

Algoritmo 1 *Filtro de Kalman Discreto.*

- 01: FGD($\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, u_t, z_t$)
 - 02: $\bar{\boldsymbol{\mu}}_t = \mathbf{A}_t \boldsymbol{\mu}_{t-1} + \mathbf{B}_t \mathbf{u}_{t-1}$
 - 03: $\bar{\boldsymbol{\Sigma}}_t = \mathbf{A}_t \boldsymbol{\Sigma}_{t-1} \mathbf{A}_t^T + \mathbf{R}_t$
 - 04: $\mathbf{K}_t = \bar{\boldsymbol{\Sigma}}_t \mathbf{C}_t^T (\mathbf{C}_t \bar{\boldsymbol{\Sigma}}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1}$
 - 05: $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t (\mathbf{z}_t - \mathbf{C}_t \bar{\boldsymbol{\mu}}_t)$
 - 06: $\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\boldsymbol{\Sigma}}_t$
 - 07: *return*($\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t$)
-

Uma das derivações do filtro de Kalman aplicado a sistemas não lineares é o Filtro de Kalman Estendido.

Filtro de Kalman Estendido - FKE

A idéia do FKE é linearizar as funções em torno da estimação corrente usando as derivadas parciais do processo e das funções de medição para calcular as estimações, mesmo em face a relações não-lineares.

O modelo do sistema para o FKE é dado pelo sistema 2.4:

$$\begin{cases} \mathbf{s}_t &= g(\mathbf{u}_{t-1}, \mathbf{s}_{t-1}) + \gamma_t \\ \mathbf{z}_t &= h(\mathbf{s}_t) + \delta_t \end{cases} \quad (2.4)$$

onde $g(\mathbf{u}_{t-1}, \mathbf{s}_{t-1})$ é uma função não-linear que representa o modelo do sistema, e $h(\mathbf{s}_t)$ é uma função não-linear que representa o modelo das medições. Suas etapas de predição e atualização podem ser obtidas pelos sistemas de equações 2.5 e 2.6 respectivamente.

$$\begin{cases} \bar{\mu}_t &= g(\mathbf{u}_{t-1}, \mu_{t-1}) \\ \bar{\Sigma}_t &= \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{R}_t \end{cases} \quad (2.5)$$

$$\begin{cases} \mathbf{K}_t &= \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t &= \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\mu}_t)) \\ \Sigma_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t \end{cases} \quad (2.6)$$

A matriz \mathbf{G} , $n \times n$, é o jacobiano que lineariza o modelo, e \mathbf{H} , $l \times n$, o jacobiano que lineariza o vetor de medições. Tais matrizes são definidas pelas equações 2.7 e 2.8.

$$\mathbf{G}_t = \frac{\partial g(\mathbf{u}_{t-1}, \mathbf{s}_{t-1})}{\partial \mathbf{s}_{t-1}} \quad (2.7)$$

$$\mathbf{H}_t = \frac{\partial h(\mathbf{s}_t)}{\partial \mathbf{s}_t} \quad (2.8)$$

Um algoritmo para o FKE é apresentado quadro 2.

Algoritmo 2 *Filtro de Kalman Extendido.*

- 01: FKE($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
 - 02: $\bar{\mu}_t = g(\mathbf{u}_{t-1}, \mu_{t-1})$
 - 03: $\mathbf{G}_t = \frac{\partial g(\mathbf{u}_{t-1}, \mathbf{s}_{t-1})}{\partial \mathbf{s}_{t-1}}$
 - 04: $\bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{R}_t$
 - 05: $\mathbf{H}_t = \frac{\partial h(\mathbf{s}_t)}{\partial \mathbf{s}_t}$
 - 06: $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1}$
 - 07: $\mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\mu}_t))$
 - 08: $\Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t$
 - 09: *return*(μ_t, Σ_t)
-

Comparando os algoritmos apresentados nos quadros 1 e 2 percebe-se duas diferenças:

- linha 3 do quadro 1 com linha 4 do quadro 2;
- linha 4 do quadro 1 com linha 6 do quadro 2;

Isso se deve ao fato de o FKE utilizar generalizações não-lineares das matrizes utilizadas no FKD. No filtro estendido o jacobiano \mathbf{G}_t corresponde as matrizes \mathbf{A}_t e \mathbf{B}_t e o jacobiano \mathbf{H}_t corresponde a matriz \mathbf{C}_t [Thrun et al. 2005].

2.2 Caracterização do problema

Foi definido no capítulo 1, que o sistema robótico utilizado neste trabalho necessita de dois sistemas de localização: um para a localização absoluta do robô com rodas e outro para localização relativa do humanóide. A seguir serão detalhados cada um deles.

2.2.1 Localização absoluta do robô com rodas

Para decidir uma estratégia de localização de um robô é necessário conhecer os sensores que ele possui. O robô utilizado possui dois *encoders* acoplados aos eixos dos motores e uma câmera dedicada para o sistema de localização absoluta. A figura 2.5 mostra o robô.



Figura 2.5: Robô com rodas.

Utilizar somente os *encoders* significa fazer uma localização baseada em odometria. Porém, como já foi mencionado neste capítulo, a técnica de odometria está sujeita a erros cumulativos que comprometem a sua utilização. Para corrigir os erros de odometria uma estratégia amplamente utilizada na robótica consiste em corrigir os erros via detecção de marcos. Uma vez detectado um marco, a pose do robô é corrigida zerando os erros de odometria. A figura 2.6 ilustra o procedimento de correção.

É importante ressaltar que mesmo as técnicas de localização absolutas, como a utilização de marcos, são imprecisas devido aos ruídos oriundos dos sensores utilizados. Portanto, é interessante a utilização de um filtro para obter as informações sensoriais com a maior precisão possível. Um técnica que pode ser utilizada é o Filtro de Kalman.

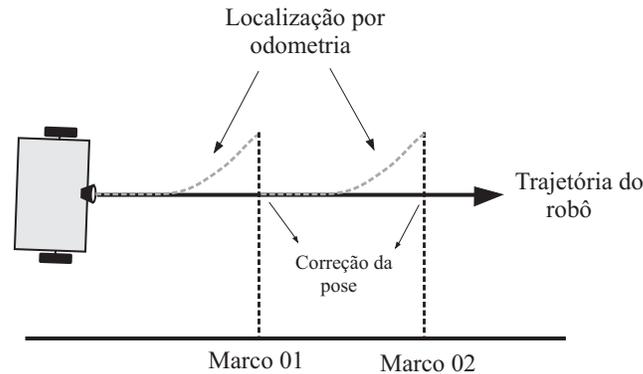


Figura 2.6: Correção do erro de odometria.

O problema abordado neste trabalho, no tocante à localização do robô com rodas, consiste em desenvolver um sistema capaz de localizá-lo em um ambiente onde as linhas do chão formam uma grade bidimensional. Para tal, as linhas são identificadas como marcos naturais e suas características, juntamente com o modelo de odometria do robô, são incorporados em um filtro de Kalman a fim de obter a pose corrigida do robô.

2.2.2 Localização relativa do humanóide

A localização relativa do humanóide faz-se necessária porque ele é não instrumentado. Para permitir o cálculo da sua pose foi inserido um marco nas suas costas, como mostra a figura 2.7.

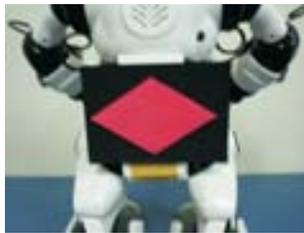


Figura 2.7: Robô humanóide.

A partir de imagens do marco obtidas por uma câmera localizada no robô com rodas e utilizando algum método de estimação de pose pode-se calcular a posição do humanóide em relação ao robô com rodas. Supondo então conhecida a posição absoluta do robô com rodas pode-se calcular a pose absoluta do humanóide.

A identificação do marco é feita utilizando a transformada de Hough de forma a encontrar cinco pontos na imagem: os quatro vértices do losango e mais o ponto de intersecção entre as suas diagonais.

As medidas das diagonais (maior e menor) do losango, o ponto de intersecção entre elas e a pose do robô com rodas funcionam como entrada para um filtro de Kalman que tem como saída a pose relativa do humanóide em relação ao robô com rodas.

2.3 Solução proposta

Seguindo a idéia apresentada na seção 2.2, as soluções propostas para o sistema de localização do robô com rodas e o sistema de localização relativa do humanóide serão apresentadas de forma separada.

2.3.1 Localização do robô com rodas

Como o mencionado na secção 2.2 o problema, no tocante a localização do robô com rodas, consiste em desenvolver um sistema capaz de localizá-lo em um ambiente onde as linhas do chão formam uma grade bidimensional. Para tal, as linhas são identificadas como marcos naturais e suas características, juntamente com o modelo de odometria do robô, são incorporados em um filtro de Kalman a fim de obter sua pose corrigida.

O esquema apresentado na figura 2.8 representa a modelagem do problema.

A seguir serão detalhadas as etapas de predição e atualização do filtro de Kalman bem como definidas as matrizes que serão utilizadas.

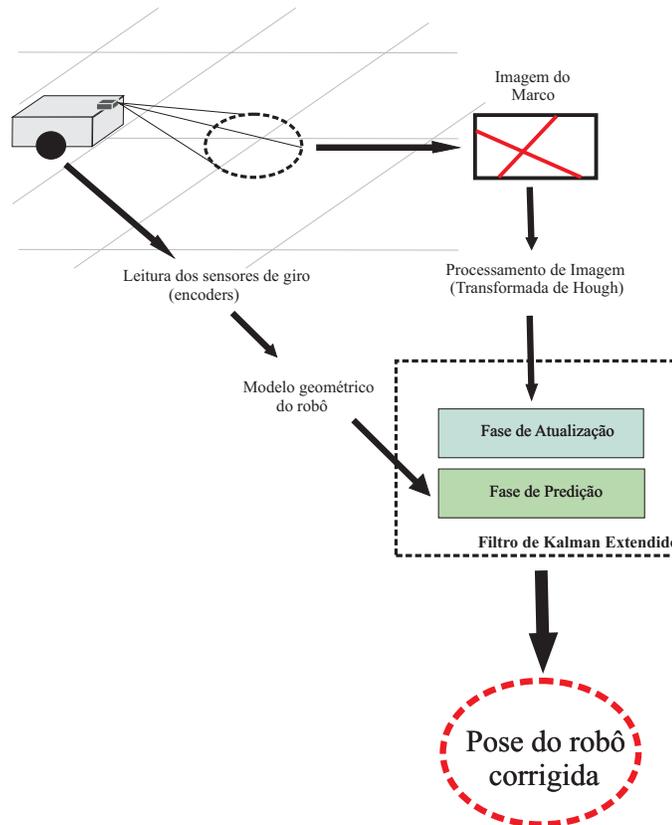


Figura 2.8: Modelagem do sistema de localização absoluta para o robô com rodas.

Fase de Predição: Modelo da Odometria do Movimento do Robô

Tradicionalmente, o comportamento do movimento de um robô é descrito por seu modelo dinâmico. A modelagem deste tipo de sistema é bastante complexa pois existem muitas variáveis envolvidas (massas e momentos de inércia, atritos, atuadores, etc.). Mesmo os sistemas mais elaborados não conseguem retratar fidedignamente o comportamento do movimento dos robôs.

Um método clássico utilizado para calcular a pose de um robô é a odometria. Este método utiliza sensores, *encoders* óticos por exemplo, que medem as rotações das rodas do robô. Usando o modelo cinemático do robô sua pose é calculada através da integração dos seus movimentos a partir de um referencial fixo.

Thrun et al. (2005) propõem que as informações odométricas não funcionem como medidas sensoriais, mas sim, que elas sejam incorporadas ao modelo do robô. Este modelo consiste em utilizar as leituras dos sensores odométricos (*encoders* no nosso caso) funcionando como sinais de controle aplicados ao modelo do robô na fase de predição do filtro de Kalman.

Considere um robô com acionamento diferencial em que os sinais de controle aplicados a seus atuadores não sejam tensão mas sim, deslocamentos angulares, conforme a Figura 2.9. De acordo com esta idéia, e supondo que as velocidades são constantes no período de amostragem, pode-se determinar o modelo geométrico do movimento do robô (sistema 2.9).

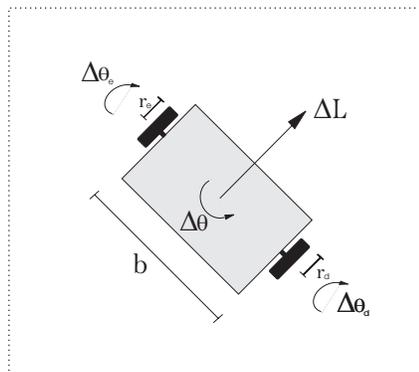


Figura 2.9: Variáveis do modelo geométrico.

- ΔL : deslocamento linear do robô;
- $\Delta\theta$: deslocamento angular do robô;
- b : comprimento do eixo do robô;
- $\Delta\theta_d$: deslocamento angular da roda direita do robô;
- $\Delta\theta_e$: deslocamento angular da roda esquerda do robô;
- r_d : raio da roda direita do robô;
- r_e : raio da roda esquerda do robô;

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta L}{\Delta \theta} [\sin(\theta_{t-1} + \Delta \theta) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} - \frac{\Delta L}{\Delta \theta} [\cos(\theta_{t-1} + \Delta \theta) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta \theta \end{cases} \quad (2.9)$$

Para facilitar a legibilidade do sistema 2.9, que representa o modelo de odometria do robô, foram utilizadas duas variáveis auxiliares ΔL e $\Delta \theta$

$$\begin{cases} \Delta L = (\Delta \theta_d r_d + \Delta \theta_e r_e) / 2 \\ \Delta \theta = (\Delta \theta_d r_d - \Delta \theta_e r_e) / b \end{cases} \quad (2.10)$$

onde $\Delta \theta_d$ é a leitura do *encoder* direito e funciona para o robô como o deslocamento angular aplicado à roda direita; $\Delta \theta_e$ é a leitura do *encoder* esquerdo e funciona como o deslocamento aplicado à roda esquerda; b representa a distância de roda a roda do robô, e r_d e r_e são os raios de suas rodas direita e esquerda respectivamente.

É importante ressaltar que em aplicações reais o deslocamento angular efetivamente realizado pela roda difere do medido pelo *encoder*. Além disso, a suposição de que as velocidades são constantes no período de amostragem, que foi utilizada para obter o modelo 2.9, nem sempre é verdade. Daí, existem diferenças entre os deslocamentos angulares “reais” das rodas ($\Delta \hat{\theta}_d$ e $\Delta \hat{\theta}_e$) e os medidos pelos *encoders* ($\Delta \theta_d$ e $\Delta \theta_e$). Esta diferença será modelada por um ruído gaussiano aditivo, conforme o sistema 2.11.

$$\begin{cases} \Delta \hat{\theta}_d = \Delta \theta_d + \varepsilon_d \\ \Delta \hat{\theta}_e = \Delta \theta_e + \varepsilon_e \end{cases} \quad (2.11)$$

Sabe-se que a odometria possui erro acumulativo, portanto, os ruídos ε_d e ε_e não possuem variância constante. Assume-se que estes ruídos têm desvio padrão proporcional ao módulo do deslocamento medido.

Com essas novas considerações o sistema 2.9 passa a ser representado pelo sistema 2.12:

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta \hat{L}}{\Delta \hat{\theta}} [\sin(\theta_{t-1} + \Delta \hat{\theta}) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} - \frac{\Delta \hat{L}}{\Delta \hat{\theta}} [\cos(\theta_{t-1} + \Delta \hat{\theta}) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta \hat{\theta} \end{cases} \quad (2.12)$$

onde

$$\begin{cases} \Delta \hat{L} = (\Delta \hat{\theta}_d r_d + \Delta \hat{\theta}_e r_e) / 2 \\ \Delta \hat{\theta} = (\Delta \hat{\theta}_d r_d - \Delta \hat{\theta}_e r_e) / b \end{cases} \quad (2.13)$$

Observa-se que este modelo não pode ser utilizado quando $\Delta\hat{\theta} = 0$. Ocorrendo isto, utiliza-se um modelo odométrico mais simples do robô (sistema 2.14), obtido a partir do limite do sistema 2.12 quando $\Delta\hat{\theta} \rightarrow 0$.

$$\begin{cases} x_t = x_{t-1} + \Delta\hat{L}\cos(\theta_{t-1}) \\ y_t = y_{t-1} + \Delta\hat{L}\sin(\theta_{t-1}) \\ \theta_t = \theta_{t-1} \end{cases} \quad (2.14)$$

Utilizar a idéia de Thrun implica em uma modificação na formulação original do FKE, pois o ruído não é mais aditivo e sim incorporado à função que lineariza o modelo como mostra o sistema 2.15, a ser comparado com o sistema 2.4:

$$\begin{cases} \mathbf{s}_t = p(\mathbf{u}_{t-1}, \mathbf{s}_{t-1}, \boldsymbol{\varepsilon}_t) \\ \mathbf{z}_t = h(\mathbf{s}_t) + \boldsymbol{\delta}_t \end{cases} \quad (2.15)$$

onde $\boldsymbol{\varepsilon}_t = [\varepsilon_d \ \varepsilon_e]^T$ é o vetor de ruído associado à odometria.

É necessário, portanto, fazer uma modificação na fase de predição do FKE tradicional. O próprio Thrun propõe que a nova fase de predição seja dada pelo sistema 2.16:

$$\begin{cases} \bar{\boldsymbol{\mu}}_t = \boldsymbol{\mu}_{t-1} + p(\mathbf{u}_{t-1}, \boldsymbol{\mu}_{t-1}, 0) \\ \bar{\boldsymbol{\Sigma}}_t = \mathbf{G}_t \boldsymbol{\Sigma}_{t-1} \mathbf{G}_t^T + \mathbf{V}_t \mathbf{M}_t \mathbf{V}_t^T \end{cases} \quad (2.16)$$

onde a matriz \mathbf{M} , $l \times l$, é a matriz de covariância dos ruídos dos sensores ($\boldsymbol{\varepsilon}$) e \mathbf{V} , $n \times m$, é o jacobiano que mapeia o ruído dos sensores para o espaço de estado. A matriz \mathbf{V} é definida pela equação 2.17.

$$\mathbf{V}_t = \frac{\partial p(\mathbf{u}_{t-1}, \mathbf{s}_{t-1}, 0)}{\partial \mathbf{u}_{t-1}} \quad (2.17)$$

Utilizando o modelo de odometria do robô descrito nesta seção e as definições das matrizes utilizadas pelo filtro de Kalman, tem-se:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & \frac{\Delta\hat{L}}{\Delta\hat{\theta}} [\cos(\theta_{t-1} + \Delta\hat{\theta}) - \cos(\theta_{t-1})] \\ 0 & 1 & \frac{\Delta\hat{L}}{\Delta\hat{\theta}} [\sin(\theta_{t-1} + \Delta\hat{\theta}) - \sin(\theta_{t-1})] \\ 0 & 0 & 1 \end{pmatrix} \quad (2.18)$$

$$\mathbf{V} = \begin{pmatrix} k1 \cos(k2) - k3[\sin(k2) - \sin(\theta_{t-1})] & -k1 \cos(k2) + k3[\sin(k2) - \sin(\theta_{t-1})] \\ k1 \sin(k2) - k3[-\cos(k2) + \cos(\theta_{t-1})] & -k1 \sin(k2) + k3[-\cos(k2) + \cos(\theta_{t-1})] \\ r_d/b & -r_e/b \end{pmatrix} \quad (2.19)$$

$$\mathbf{G}^* = \begin{pmatrix} 1 & 0 & -\Delta\hat{\theta} \sin(\theta_{t-1}) \\ 0 & 1 & \Delta\hat{\theta} \cos(\theta_{t-1}) \\ 0 & 0 & 1 \end{pmatrix} \quad (2.20)$$

$$\mathbf{V}^* = \begin{pmatrix} \cos(\theta_{t-1}) & 0 \\ \sin(\theta_{t-1}) & 0 \\ r_d/b & -r_e/b \end{pmatrix} \quad (2.21)$$

As matrizes \mathbf{G} e \mathbf{H} são utilizadas para o modelo completo (equação 2.12) e as matrizes \mathbf{G}^* e \mathbf{H}^* para o modelo simplificado (equação 2.14).

A matriz \mathbf{M} é definida por:

$$\mathbf{M} = \begin{pmatrix} (\alpha_1 |\Delta\hat{\theta}_d|)^2 & 0 \\ 0 & (\alpha_2 |\Delta\hat{\theta}_e|)^2 \end{pmatrix} \quad (2.22)$$

onde os valores dos elementos m_{11} e m_{22} na equação 2.22 representam o fato dos desvios padrões de ε_d e ε_e serem proporcionais ao módulo do deslocamento angular. As variáveis k_1 , k_2 e k_3 são dadas pelo sistema 2.23, considerando $r_d = r_e = r$.

$$\begin{cases} k_1 = \frac{r(\Delta\hat{\theta}_d + \Delta\hat{\theta}_e)}{b(\Delta\hat{\theta}_d - \Delta\hat{\theta}_e)} \\ k_2 = \theta_{t-1} + \frac{r(\Delta\hat{\theta}_d - \Delta\hat{\theta}_e)}{b} \\ k_3 = \frac{b\Delta\hat{\theta}_e}{2(r(\Delta\hat{\theta}_d - \Delta\hat{\theta}_e)/b)^2} \end{cases} \quad (2.23)$$

Fase de Atualização: Modelo do sensor para a Detecção dos Marcos Naturais

Os marcos naturais utilizados são um conjunto de linhas retas formadas pelas arestas do piso do ambiente onde o robô irá navegar. A escolha dos marcos deu-se devido a já existirem no ambiente, e além disso representam uma característica muito comum no mundo real.

Escolhida as retas como marcos, a técnica adotada para identificá-los é a transformada de Hough [Hughes 1986]. Esta transformada é um método para detectar em uma imagem digital, uma classe de formas geométricas que pode ser representada por uma curva paramétrica [Gonzales 2000]. No caso das retas, é feito um mapeamento entre o espaço cartesiano (X, Y) e o espaço de parâmetros (ρ, α) em que a reta é definida.

Hough define uma reta utilizando sua representação normal como mostra a equação 2.24, onde o parâmetro ρ representa o comprimento do vetor que passa pela origem e é perpendicular à reta e α o ângulo que esse vetor forma com o eixo X.

A Figura 2.10 mostra a representação geométrica desses parâmetros.

$$\rho = x \cos(\alpha) + y \sin(\alpha) \quad (2.24)$$

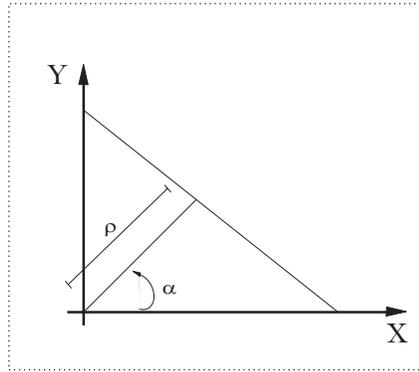


Figura 2.10: Parâmetros de Hough.

O robô navega em um ambiente onde a posição das retas no mundo é conhecida e a cada passo identifica os descritores das retas contidas na imagem α^I e ρ^I . Esses descritores são mapeados para o plano de um sistema de coordenadas móvel obtendo-se α^M e ρ^M . Tal transformação é fácil e depende exclusivamente da calibração correta dos parâmetros extrínsecos da câmera. A figura 2.11 ilustra os sistemas de coordenadas utilizados na dedução matemática do modelo do sensor.

Como nomenclaturas adotadas tem-se que (X^F, Y^F) é um sistema de coordenadas fixo no mundo; (X^M, Y^M) é um sistemas de coordenadas móvel; (x_M^F, y_M^F) é a coordenada da origem do sistema móvel mapeado no sistema de coordenadas fixo e a variável θ_M representa o ângulo de rotação do sistema de coordenadas móvel.

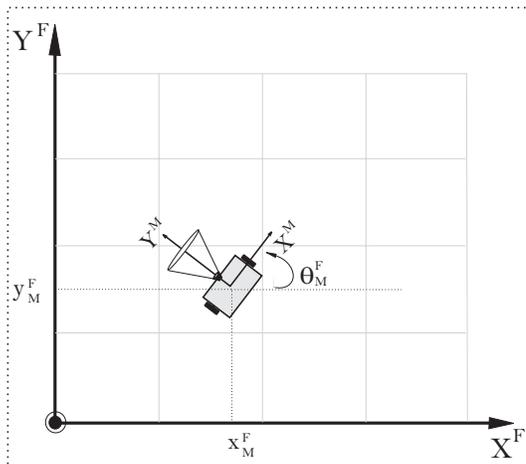


Figura 2.11: Sistema de coordenadas.

Parte-se de uma transformação simples que mapeia um ponto no sistema de coordenadas móvel {M} para o sistema de coordenadas fixo {F} como mostra o sistema 2.25.

$$\begin{cases} x^F = \cos(\theta_F^M)x^M - \sin(\theta_F^M)y^M + x_M^F \\ y^F = \sin(\theta_F^M)x^M + \cos(\theta_F^M)y^M + y_M^F \end{cases} \quad (2.25)$$

Da equação 2.24 tem-se que:

$$\begin{cases} \rho^F = x^F \cos(\alpha^F) + y^F \sin(\alpha^F) \\ \rho^M = x^M \cos(\alpha^M) + y^M \sin(\alpha^M) \end{cases} \quad (2.26)$$

Substituindo (2.25) em (2.26) obtém-se o sistema 2.27 que representa o modelo do sensor a ser utilizado no filtro.

$$\begin{cases} \alpha^M = \alpha^F - \theta_M^F \\ \rho^M = \rho^F - x_M^F \cos(\alpha^F) - y_M^F \sin(\alpha^F) \end{cases} \quad (2.27)$$

Neste sistema, α^F e ρ^F são dados, pois representam a descrição do marco no mapa, que é suposto conhecido. As equações expressam a relação entre a informação retornada pelo sensor (α^M, ρ^M) e as grandezas que se quer estimar (x_F^M, y_F^M, θ_F^M).

Note-se que há uma relação direta entre estas variáveis (x_F^M, y_F^M, θ_F^M) e a pose do robô (x_R, y_R, θ_R) que é dada pelo sistema 2.28

$$\begin{cases} x_R = x_F^M \\ y_R = y_F^M \\ \theta_R = \theta_F^M + \frac{\pi}{2} \end{cases} \quad (2.28)$$

O modelo do sistema 2.27 é incorporado ao filtro de Kalman através da matriz H (equação 2.8) que é dada pela equação 2.29.

$$H = \begin{pmatrix} -\cos(\theta_M^F) & -\sin(\theta_M^F) & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (2.29)$$

2.3.2 Localização relativa do humanóide

Foram apresentados na secção 2.2 as características do sistema de localização relativa necessárias para a identificação e cálculo da pose do robô humanóide.

A idéia apresentada consiste em identificar em um losango inserido nas costa do humanóide, as dimensões das suas diagonais e o ponto de interseção entre elas e utilizar essas informações em conjunto com a pose do robô com rodas em um filtro de Kalman a fim de obter a pose relativa do humanóide. A figura 2.12 apresenta o esquema da modelagem do problema.

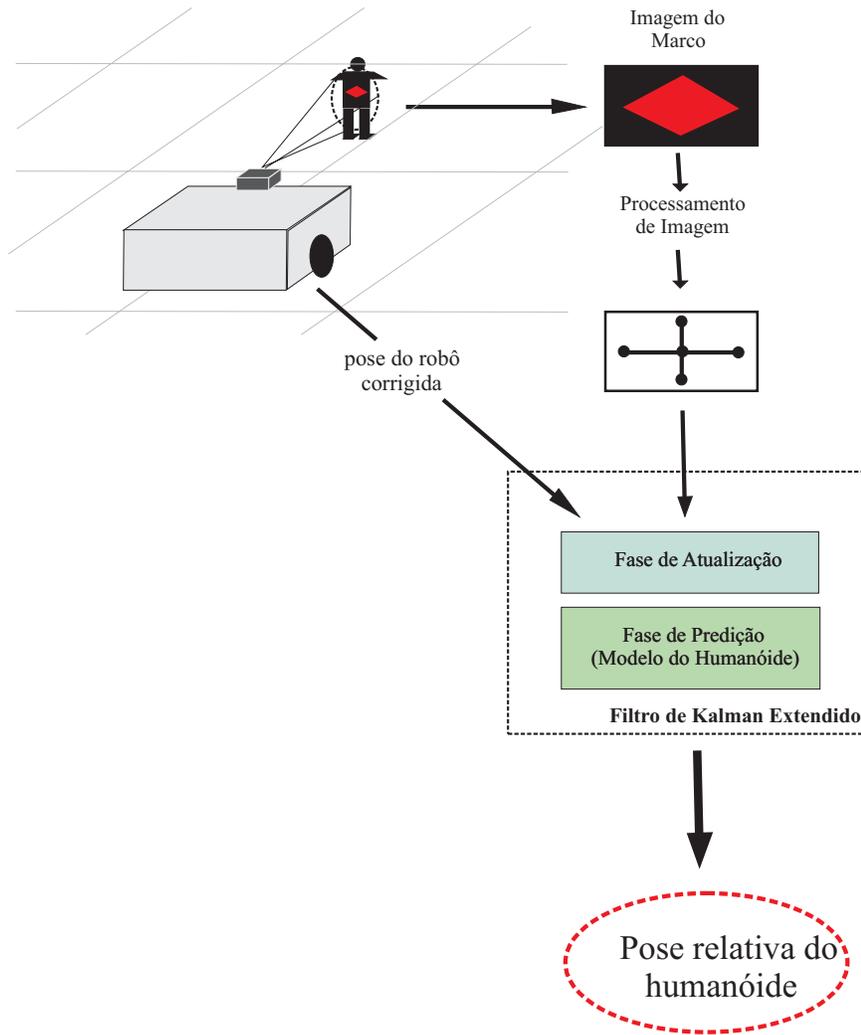


Figura 2.12: Modelagem do sistema de localização relativa do humanóide.

A seguir será detalhado as etapas de predição e atualização do filtro de Kalman bem como definidas as matrizes que serão utilizadas.

Fase de Predição: Modelo do Humanóide

O robô humanoide utilizado possui quatro funções básicas em relação à sua movimentação: andar para frente, andar para trás, girar para a esquerda e girar para a direita. Sua movimentação é regida por um tipo de marcha denominada marcha estática. Neste tipo de marcha, o robô deve realizar movimentos de tal forma que o seu centro de massa permaneça sempre interior ao perímetro do seu pé de apoio. Dessa forma, o robô estará sempre estável. Contudo, este tipo de locomoção restringe os movimentos do robô a movimentos com velocidades constantes e lentas, para não introduzir componentes dinâmicas [Nogueira 2005].

Baseado nos comentário de Nogueira (2005), o humanóide foi modelado de forma simplista, ou seja, não foi incorporado ao seu modelo nenhum tipo de dinâmica. A equação 2.30 apresenta o modelo do robô onde v_t e w_t representam os sinais de controle.

$$\begin{cases} x_t = x_{t-1} + v_t \cdot \cos(\theta_{t-1}) \cdot \Delta t \\ y_t = y_{t-1} + v_t \cdot \sin(\theta_{t-1}) \cdot \Delta t \\ \theta_t = \theta_{t-1} + w_t \cdot \Delta t \end{cases} \quad (2.30)$$

Sabe-se que em aplicações reais o efeito das velocidades comandadas diferem-se das efetivamente aplicadas. Para incorporar esse efeito, um ruído branco gaussiano aditivo foi inserido a v e w .

$$\begin{cases} \hat{v} = v + \tau_1 \\ \hat{w} = w + \tau_2 \end{cases} \quad (2.31)$$

Com a modificação feita o modelo de movimento do humanóide passa a ser dado pelo sistema 2.32.

$$\begin{cases} x_t = x_{t-1} + \hat{v}_t \cdot \cos(\theta_{t-1}) \cdot \Delta t \\ y_t = y_{t-1} + \hat{v}_t \cdot \sin(\theta_{t-1}) \cdot \Delta t \\ \theta_t = \theta_{t-1} + \hat{w}_t \cdot \Delta t \end{cases} \quad (2.32)$$

Observe que a alteração feita implica que o sistema modificado é semelhante ao definido na equação 2.15. Portanto, a fase de predição utilizada obedece o sistema de equações 2.16.

Usando equações 2.7, 2.17, as matrizes \mathbf{G} , \mathbf{V} são definidas por:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & -\hat{v}_t \cdot \sin(\theta_{t-1}) \cdot \Delta t \\ 0 & 1 & \hat{v}_t \cdot \cos(\theta_{t-1}) \cdot \Delta t \\ 0 & 0 & \hat{w}_t \cdot \Delta t \end{pmatrix} \quad (2.33)$$

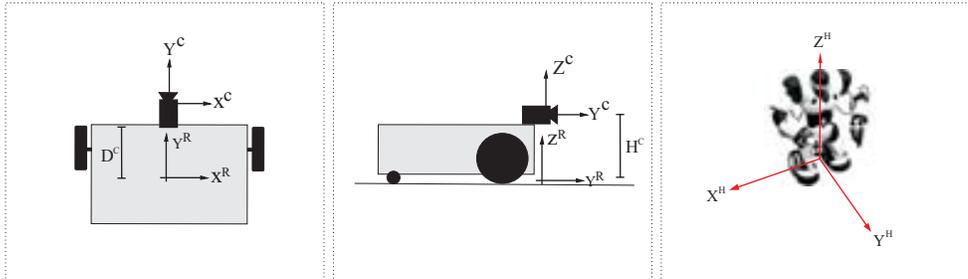
$$\mathbf{V} = \begin{pmatrix} \cos(\theta_{t-1}) \cdot \Delta t & 0 \\ \sin(\theta_{t-1}) \cdot \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \quad (2.34)$$

A matriz \mathbf{M} , que representa a covariância dos sensores, é dada por:

$$\mathbf{M} = \begin{pmatrix} (\tau_1)^2 & 0 \\ 0 & (\tau_2)^2 \end{pmatrix} \quad (2.35)$$

Fase de Atualização: Modelagem do sensor para a detecção do Marco

Para entender a modelagem feita é necessário conhecer a representação dos sistemas de coordenadas dos robôs. A figura 2.14 ilustra os sistemas.



(a) Sistema de coordenadas do robô com rodas, vista de cima. (b) Sistema de coordenadas do robô com rodas, vista lateral. (c) Sistema de coordenadas do humanóide.

Figura 2.13: Sistemas de Coordenadas dos robôs.

- (X^C, Y^C, Z^C) : sistema de coordenadas da câmera;
- (X^R, Y^R, Z^R) : sistema de coordenadas do robô com rodas;
- (X^H, Y^H, Z^H) : sistema de coordenadas do humanóide;
- D^C : distância da câmera a origem do sistema de coordenadas do robô móvel;
- H^C : distância da câmera para o chão.

O marco utilizado para identificar o humanóide foi um losango colocado nas suas costas. Deste polígono extraem-se cinco pontos: os quatro vértices e o ponto de interseção das suas diagonais. A partir dos vértices calcula-se o tamanho das diagonais que juntamente com a informação do ponto de interseção representam as informações sensoriais utilizadas.

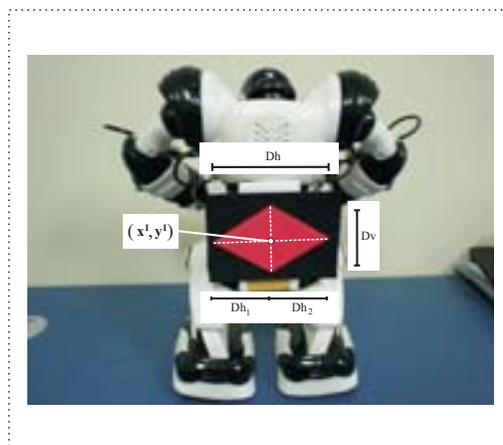


Figura 2.14: Informações sensoriais utilizadas.

Na figura 2.14 Dh representa a medida da diagonal horizontal, Dv a medida da diagonal vertical e (x^I, y^I) o ponto de interseção das diagonais.

A escolha de representar as informações sensoriais por Dv, Dh e (x^I, y^I) deu-se devido conterem informações suficientes para estimação da pose do humanóide e além disso poderem ser obtidas facilmente em uma imagem.

Sendo $(x_C^H, y_C^H, \theta_C^H)$ a pose relativa do humanóide em relação a câmera existente no robô com rodas, é fácil constatar as relações entre $(x_C^H$ e $x^I)$, $(y_C^H$ e $Dv)$ e $(\theta_C^H$ e $Dh)$. Se houver uma mudança na coordenada x^I do ponto de interseção das diagonais houve um desalinhamento dos robôs, ou seja, x_C^H muda de valor. A relação entre Dv e y_C^H pode ser observada quando o robô se distancia da câmera ocasionando uma diminuição do comprimento de Dv na imagem. A última relação θ_C^H e Dh pode ser notada quando o humanóide executa um movimento de giro. O giro faz com que Dh apareça menor na imagem.

Definidas as informações sensoriais é necessário agora formalizá-las matematicamente. Analisando a figura 2.15 e considerando um modelo *pin-hole* da câmera tem-se o sistema 2.36.

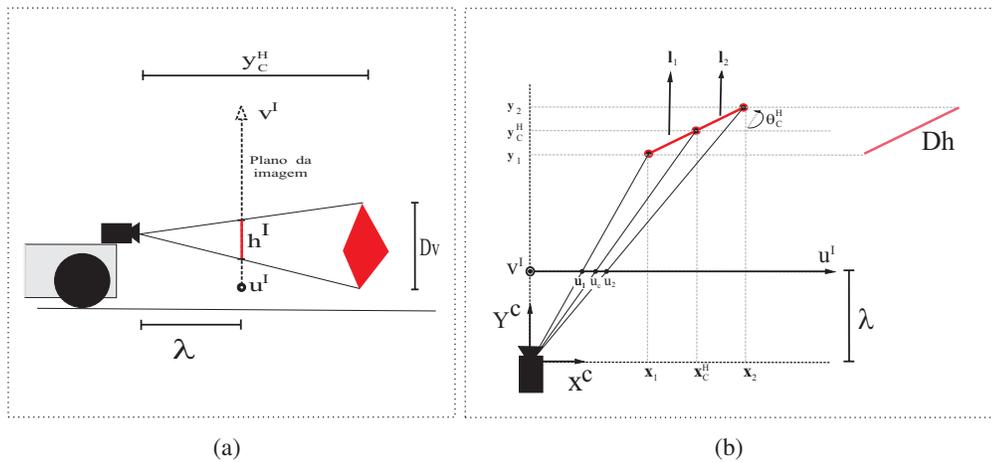


Figura 2.15: Modelagem do sensor.

- $(x_C^H, y_C^H, \theta_C^H)$: pose relativa do humanóide em relação a câmera;
- (u^I, v^I) : sistema de coordenadas da imagem;
- Dv : tamanho real da diagonal vertical do marco;
- Dh : tamanho real da diagonal horizontal do marco;
- λ : distância focal;
- h^I : tamanho com que a diagonal vertical aparece na imagem;
- u_C^I : coordenada x do ponto de interseção das diagonais projetada no plano da imagem;
- l^I : tamanho que a diagonal horizontal aparece na imagem ($l^I = l_1 + l_2$);

$$\left\{ \begin{array}{l} u_C^I = \frac{x_C^H \cdot \lambda}{y_C^H} \\ h^I = \frac{\lambda \cdot Dv}{y_C^H} \\ l_1 = \frac{\lambda \cdot (Dh/2) \cdot (y_C^H \cdot \cos(\theta_C^H) - x_C^H \cdot \sin(\theta_C^H))}{(y_C^H)^2 - (Dh/2) \cdot (y_C^H) \cdot \sin(\theta_C^H)} \\ l_2 = \frac{\lambda \cdot (Dh/2) \cdot (y_C^H \cdot \cos(\theta_C^H) - x_C^H \cdot \sin(\theta_C^H))}{(y_C^H)^2 + (Dh/2) \cdot (y_C^H) \cdot \sin(\theta_C^H)} \end{array} \right. \quad (2.36)$$

Por simplificação matemática não foram utilizadas diretamente as equações apresentadas no sistema 2.36 e sim as definidas no conjunto de equações 2.37.

$$\left\{ \begin{array}{l} (h^I) = \frac{\lambda \cdot Dv}{y_C^H} \\ (u_C^I/h^I) = \frac{x_C^H}{Dv} \\ (l_1/l^I) = \frac{y_C^H + (Dh/2) \cdot \sin(\theta_C^H)}{2 \cdot y_C^H} \end{array} \right. \quad (2.37)$$

É necessário que a modelagem do sensor seja expressa em função das variáveis de estado do humanóide. Considerando a figura 2.16 e as devidas transformações de eixos obtém-se o sistema 2.38.

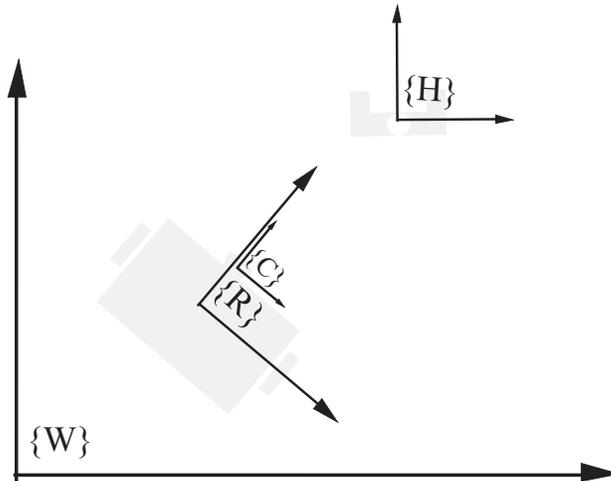


Figura 2.16: Transformações de eixos.

$$\begin{cases} x_C^H = x_W^H \cdot \cos(\theta_W^R) + y_W^H \cdot \sin(\theta_W^R) - x_W^R \cdot \cos(\theta_W^R) - y_W^R \cdot \cos(\theta_W^R) \\ y_C^H = -x_W^H \cdot \sin(\theta_W^R) + y_W^H \cdot \cos(\theta_W^R) + x_W^R \cdot \sin(\theta_W^R) - y_W^R \cdot \cos(\theta_W^R) - D^C \\ \theta_C^H = \theta_W^H - (\theta_W^R - \frac{\pi}{2}) \end{cases} \quad (2.38)$$

onde $(x_W^H, y_W^H, \theta_W^H)$ e $(x_W^R, y_W^R, \theta_W^R)$ são as respectivas poses do humanóide e do robô com rodas em relação ao mundo.

Substituindo as equações do sistema 2.38 em 2.37 e utilizando a definição apresentada em 2.8 pode-se definir \mathbf{H} , que é a matriz utilizada no filtro de Kalman para incorporar as informações dos sensores.

Para a localização relativa do humanóide a matriz \mathbf{H} é dada por:

$$H = \begin{pmatrix} h_{11} & h_{12} & 0 \\ h_{21} & h_{22} & 0 \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \quad (2.39)$$

onde,

$$h_{11} = \frac{\lambda \cdot Dv \cdot \sin(\theta_W^R)}{(x_W^H \cdot \sin(\theta_W^R) + y_W^H \cdot \cos(\theta_W^R) + x_W^R \cdot \sin(\theta_W^R) - y_W^R \cdot \cos(\theta_W^R) - D^C)^2}$$

$$h_{12} = \frac{-\lambda \cdot Dv \cdot \cos(\theta_W^R)}{(x_W^H \cdot \sin(\theta_W^R) + y_W^H \cdot \cos(\theta_W^R) + x_W^R \cdot \sin(\theta_W^R) - y_W^R \cdot \cos(\theta_W^R) - D^C)^2}$$

$$h_{21} = \frac{\cos(\theta_W^R)}{Dv}$$

$$h_{22} = \frac{\sin(\theta_W^R)}{Dv}$$

$$h_{31} = \frac{Dh \cdot \sin(\theta_W^R) \cdot \sin(\theta_W^H - \theta_W^R)}{4 \cdot (x_W^H \cdot \sin(\theta_W^R) + y_W^H \cdot \cos(\theta_W^R) + x_W^R \cdot \sin(\theta_W^R) - y_W^R \cdot \cos(\theta_W^R) - D^C)^2}$$

$$h_{32} = \frac{-Dh \cdot \cos(\theta_W^R) \cdot \sin(\theta_W^H - \theta_W^R)}{4 \cdot (x_W^H \cdot \sin(\theta_W^R) + y_W^H \cdot \cos(\theta_W^R) + x_W^R \cdot \sin(\theta_W^R) - y_W^R \cdot \cos(\theta_W^R) - D^C)^2}$$

$$h_{33} = \frac{Dh \cdot \cos(\theta_W^H - \theta_W^R)}{4 \cdot (x_W^H \cdot \sin(\theta_W^R) + y_W^H \cdot \cos(\theta_W^R) + x_W^R \cdot \sin(\theta_W^R) - y_W^R \cdot \cos(\theta_W^R) - D^C)^2}$$

2.4 Considerações

Tradicionalmente os sistemas de localização usam as informações dos sensores para calcular a pose do robô ou sua distância a um referencial fixo.

O diferencial dos sistemas de localização propostos neste trabalho está na modelagem do sensor ótico.

A modelagem foi feita de tal forma a proporcionar a utilização dos parâmetros obtidos no processamento de imagem diretamente nas equações do filtro de Kalman, sem passar por nenhuma etapa intermediária de cálculo.

A seguir, no Capítulo 3, será apresentado o planejador de caminho proposto neste trabalho. Os resultados experimentais comparando os sistemas de localização propostos com sistemas de localização utilizando odometria e correção por geometria serão apresentados no Capítulo 4.

Capítulo 3

Planejamento de Caminho

Para decidir qual o caminho seguir a fim de realizar uma tarefa faz-se necessário algum tipo de planejamento. Decidir baseado em critério de otimalidade (caminho mais curto, caminho mais seguro, caminho que forneça o menor consumo de energia, etc.) é tratado na robótica como o problema de planejamento de caminhos (*path planning*).

Em robótica móvel, planejar um caminho refere-se principalmente a competência que um robô móvel possui de escolher, dentre várias possibilidades, a rota que, segundo critérios de otimalidade, pretende percorrer. A fase de planejamento necessita, para sua execução, de uma configuração inicial e de uma configuração final do robô definidas em um mesmo sistema de referência. Além disso, normalmente pressupõe o conhecimento de um mapa baseado no qual irá avaliar as opções, a posição corrente por onde está se movendo e obstáculos existentes. A partir dessas informações o planejador deve ser capaz de gerar uma curva geométrica que conduza o robô de forma segura, desviando de obstáculos quando existirem, entre as duas configurações.

A seguir será apresentado na Seção 3.1 um resumo das técnicas de planejamento de caminho tradicionalmente utilizadas na robótica, na Seção 3.2 a caracterização do problema e, por fim, na Seção 3.3 a solução proposta.

3.1 Técnicas tradicionais de planejamento de caminho

Segundo Choset et al. (2005) e Lavelle (2006) as abordagens atuais de planejamento de caminho podem ser divididas basicamente em três grupos: métodos que utilizam mapas de rotas (*roadmaps*), métodos que utilizam funções de potencial (*potential fields*) e métodos de decomposição em células (*cell decomposition*). A seguir serão descritos, de forma simplificada, cada um destes métodos.

3.1.1 Espaço de Configuração

Antes de explorar as técnicas tradicionais de planejamento de caminho é interessante um breve comentário sobre espaço de configurações.

Sabe-se que os robôs móveis são construídos de diversos tamanhos e com diversas formas. A fim de simplificar suas representações uma estratégia interessante consiste em

formalizar o robô como uma entidade pontual contida em um espaço de configuração através do qual ele possa navegar.

O espaço de trabalho, W , pode ser definido como o espaço físico por onde o robô se move. No caso de robôs móveis que se deslocam apenas no plano, o espaço de trabalho é bidimensional, ou seja, definido no R^2 .

A configuração, q , de um robô pode ser definida como a especificação completa da posição de todos os pontos do robô relativa ao sistema de coordenadas fixo no ambiente. O espaço de configurações, C , é definido como sendo o conjunto de todas as possíveis configurações do robô. Neste espaço de configurações, o robô é considerado um ponto, q . Estes conceitos foram introduzidos na área de planejamento de caminho primeiramente por Udupa (1977) e popularizada posteriormente por Perez (1983). A figura 3.1 ilustra tais definições.

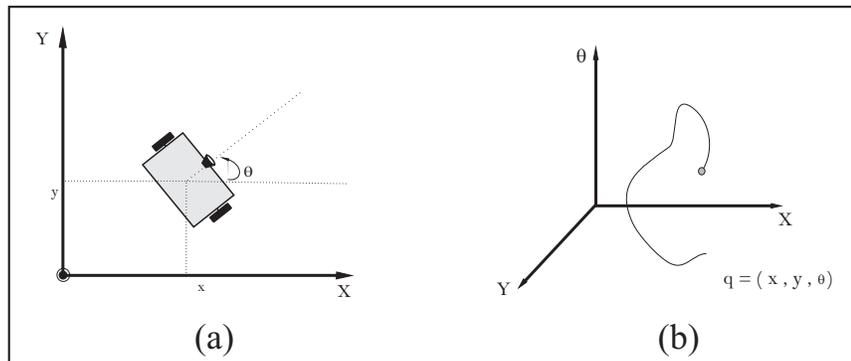


Figura 3.1: (a)Configuração do robô (b)Espaço de Configuração.

Dentro do contexto de espaço de configurações, pode-se definir dois conjuntos que auxiliam a formulação de problemas de planejamento de movimento: espaço de configuração ocupado por obstáculos, C_{obs} , e espaço de configurações livre, C_{free} . O espaço de configurações ocupado por obstáculos, C_{obs} , é definido como sendo o conjunto de configurações para as quais existe intersecção entre o robô e os obstáculos. Já o espaço de configurações livre, C_{free} , é definido como sendo o espaço em que não ocorre intersecção do robô com obstáculos.

Conforme mencionado anteriormente, no espaço de configurações o robô é considerado uma unidade pontual. Para tal, é necessário incorporar aos obstáculos as dimensões do robô. Para exemplificar, considere um robô móvel com dimensão retangular e configuração $q = \{x, y, \theta\}$ movendo-se no espaço de trabalho R^2 . A figura 3.2 apresenta as modificações incorporadas às dimensões originais do obstáculo no espaço de trabalho e no espaço de configurações.

3.1.2 Mapa de rotas - *roadmaps*

Uma *roadmap* é uma rede de curvas unidimensionais que captura a conectividade do espaço de configurações livres [Junior Valdir 2006]. Em outras palavras, os segmentos de

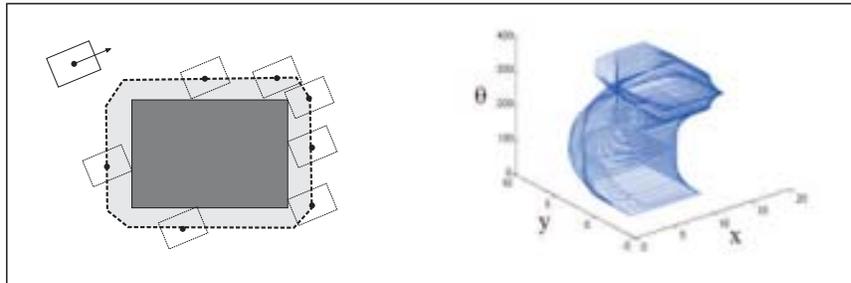


Figura 3.2: Modificações incorporadas às dimensões originais do obstáculo.

reta que conectam o vértice de um obstáculo ao vértice de outro sem interceptar o interior de qualquer outro obstáculo, formam o conjunto de caminhos denominado *roadmap*.

Latombe (1991) e Choset et al. (2005) apresentam vários métodos que utilizam *roadmap*. Destes, pode-se citar o método do grafo de visibilidade, método da retração (diagrama de Voronoi generalizado) e o método da silhueta. Cada um desses métodos propõe uma maneira diferente de se construir a *roadmap*.

O método do grafo de visibilidade, proposto inicialmente por Nilsson (1969) pode ser aplicado para espaços de configuração bidimensionais e com obstáculos poligonais. Neste método, um nó do grafo é conectado a outro se estes estão no campo de visão um do outro. Além disso, cada ponto do espaço de configurações livres deve estar no campo de visão de pelo menos um nó do grafo de visibilidade. A criação de um grafo de visibilidade é feita quando se conecta com uma linha reta dois pares de vértices, desde que esta linha não atravesse o interior de nenhum obstáculo. A figura 3.3 apresenta um exemplo da utilização do grafo de visibilidade para obter um caminho a ser percorrido pelo robô.

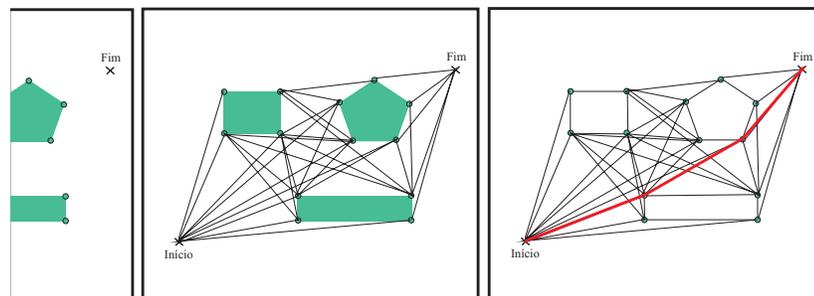


Figura 3.3: Grafo de Visibilidade.

O método da retração, em particular o diagrama de Voronoi generalizado é definido como sendo um conjunto unidimensional de espaços de configuração livres que maximiza a distância entre o robô e os obstáculos [Choset et al. 2005]. A figura 3.4 exemplifica o método.

O método da silhueta proposto por Canny (1988) constrói uma *roadmap* a partir da silhueta do espaço de configurações livres. No entanto, esta silhueta é formada por segmentos de curvas desconexas. O método então adiciona alguns segmentos para uni-los.

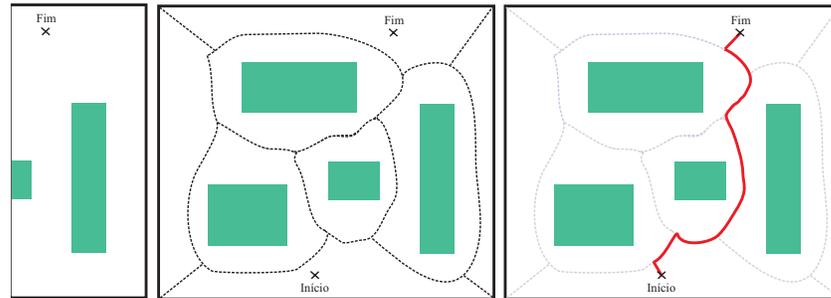


Figura 3.4: Diagrama de Voronoi.

A conexão ocorre em pontos críticos da silhueta, definidos de um modo geral como sendo os pontos onde ocorre uma mudança na conectividade da *roadmap*. Uma definição mais detalhada do método pode ser encontrada em Choset et al. (2005) e Lavelle (2006) .

3.1.3 Campo de Potencial

O método de campos de potenciais foi desenvolvido inicialmente por Khatib (1985) como um método para navegação e desvio de obstáculos quando não se tem conhecimento *a priori* do mapa do ambiente, e portanto os obstáculos são percebidos à medida que o robô navega.

Os métodos que utilizam funções de potencial trabalham de maneira diferente dos de *roadmap*. A sua implementação mais básica consiste em sair da origem e ir ao destino em pequenos passos direcionados sob a ação de forças. O destino tem uma força de atração e os obstáculos possuem forças de repulsão. A figura 3.5 ilustra o método.

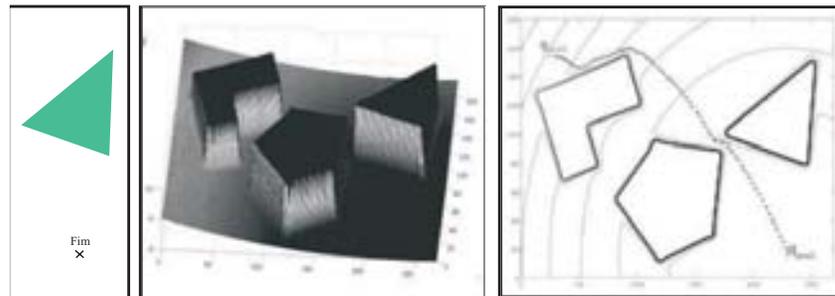


Figura 3.5: Campo de Potencial.

O principal problema desta abordagem são os mínimos locais onde as diferentes forças possuem o somatório igual a zero, resultando na parada do robô. Apesar disso, Latombe (1991) apresenta algumas técnicas *ad hoc* para lidar com essas situações e tentar fazer com que o robô saia destes mínimos locais.

3.1.4 Decomposição em Células

O método de decomposição em células consiste em dividir o espaço livre do robô em regiões convexas simples (células), de forma que um caminho entre quaisquer duas configurações em uma mesma célula possa ser facilmente obtido. Quando duas células possuem uma fronteira em comum, elas são adjacentes. A relação de adjacência ou conectividade entre as células é representada por um grafo não direcional. Este grafo é denominado grafo de conectividade. Os vértices que compõem este grafo são as células extraídas do espaço de configurações livre do robô. Há uma aresta entre dois vértices se e somente se as células correspondentes a eles são adjacentes. O resultado da busca efetuada é uma seqüência de células denominada canal. Um caminho contínuo pode ser computado a partir do canal.

Os métodos de decomposição em células podem ser divididos em métodos de decomposição exata e métodos de decomposição aproximada.

Na decomposição exata, a união das células corresponde exatamente ao espaço de configurações livre. A divisão de células é feita em pontos críticos do espaço, ou seja, pontos em que ocorre a divisão do espaço livre por um obstáculo.

Na decomposição aproximada, o espaço de configurações livre é decomposto em células de um formato específico, por exemplo quadrados de dimensões fixas. A união destas células não corresponde exatamente ao espaço livre, mas está incluído nele. As figuras 3.6 e 3.7 apresentam possíveis soluções utilizando decomposição exata e aproximada respectivamente.

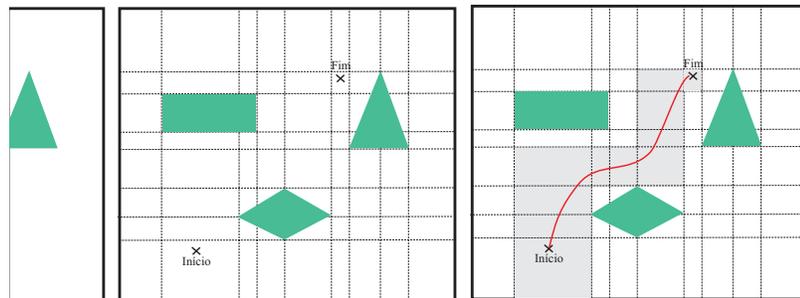


Figura 3.6: Decomposicao em células exatas.

Junior Valdir (2006) ressalta que os métodos de decomposição exata são considerados métodos de planejamento completo, ou seja, o método é capaz de encontrar uma solução para o problema do planejamento se ela existir.

Foi mencionado no início desta seção que a obtenção do canal é feita através de uma busca no grafo de conectividade. A escolha da técnica de busca deve ser feita com base em diversas características, como eficiência, garantia de encontrar uma solução ótima, sempre que possível, e facilidade de implementação.

Um técnica muito utilizada para buscas complexas é o algoritmo A^* [Russell & Norvig 2004]. No algoritmo A^* o espaço de busca é reduzido excluindo-se múltiplos caminhos para um determinado sub-nó e deixando somente o caminho de menor custo. A escolha do caminho de menor custo para expansão em cada estágio é melho-

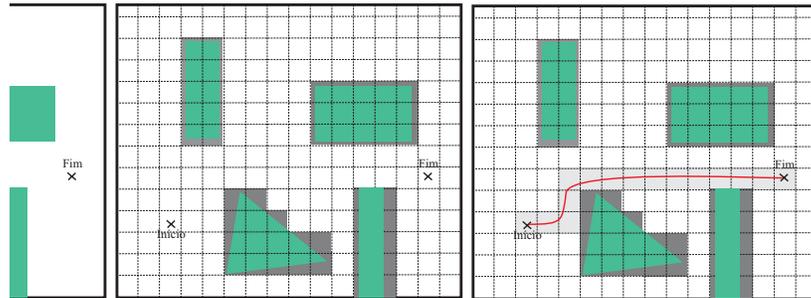


Figura 3.7: Decomposicao em células aproximadas.

rada adicionando-se ao custo atual um custo estimado para os caminhos restantes. Se o custo estimado futuro for sempre menor que o custo efetivo futuro, a busca A^* produzirá uma solução ótima. Uma função típica utilizada na estimativa desse custo no planejamento de trajetória é a distância euclidiana entre a posição representada pelo nodo atual e a posição objetivo [Russell & Norvig 2004].

Algoritmo A^*

O algoritmo A^* encontra o caminho de menor custo de um vértice a outro em um grafo examinando apenas os vizinhos mais promissores do vértice atual da busca.

O algoritmo mantém duas listas de vértices, Abertos e Fechados, para vértices a explorar e explorados, respectivamente. No início, Fechados está vazia e Abertos contém apenas o vértice de origem. A cada iteração, o melhor vértice é removido da lista de Abertos para ser explorado. Para cada vizinho (ou sucessor) do vértice explorado, são feitas as seguintes operações:

- Se está em Fechados, é ignorado;
- Se está em Abertos, suas informações são atualizadas quando o custo do caminho atual é menor do que o custo calculado anteriormente para ele;
- Se não está em nenhuma das duas listas, é um novo vértice e portanto é colocado na lista Abertos.

O vértice removido da lista Abertos é colocado na lista Fechados após todos os seus vizinhos serem processados. A cada iteração, o vértice da lista de Abertos com o menor custo estimado será escolhido pelo algoritmo. O critério para escolher o melhor vértice a ser examinado é essencialmente uma estimativa da distância entre cada vértice e o vértice de destino. O custo estimado do caminho que passa por um determinado vértice é dado pela seguinte equação 3.1:

$$f(v) = g(v) + h(v) \quad (3.1)$$

onde, $f(v)$ é custo total do caminho do vértice de origem ao de destino passando pelo vértice v ; $g(v)$ é o custo calculado do caminho do vértice de origem ao vértice v e $h(v)$ é o custo estimado do caminho do vértice v ao vértice de destino.

Em particular, o componente $h(v)$ da equação 3.1 merece destaque. Ele representa uma estimativa heurística da distância entre um determinado vértice e o vértice destino. Essa estimativa pode obedecer a qualquer função que subestime o real valor da distância entre o vértice e o destino, chamada de função admissível. Nilsson (1998) prova que, se $h(v)$ é uma sub-estimativa desse valor, então o algoritmo A^* garante encontrar o caminho mínimo entre a origem e o destino. A complexidade teórica do algoritmo A^* é $O(n^2)$ onde n é o número de vértices do grafo [Russell & Norvig 2004].

3.2 Caracterização do problema

O problema abordado neste trabalho consiste em planejar um caminho cooperativo, supondo uma mapa conhecido *a priori*, pra um sistema robótico formado por um robô humanóide não instrumentado e um robô móvel com rodas. A identificação do robô humanóide é feita através de uma marca inserida na sua parte anterior e todo o processamento da informação é executado no robô móvel que contém um notebook acoplado à sua estrutura. A figura 3.8 ilustra o sistema em questão.



Figura 3.8: Sistema Robótico.

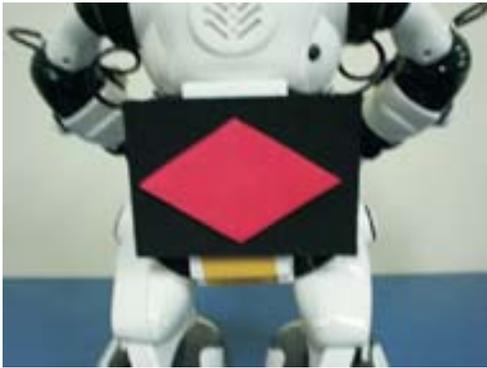
A movimentação do robô com rodas depende totalmente da movimentação do humanóide, pois este deve acompanhar o humanóide e garantir sempre uma boa imagem do mesmo. No posicionamento do robô com rodas o principal problema não é garantir que ele atinja uma posição final desejada, mas sim que ele se mantenha sempre atrás do humanóide e com uma boa visada do rótulo.

Devido à necessidade do robô com rodas posicionar-se sempre atrás do humanóide e manter uma boa imagem do mesmo, as restrições de visibilidade do marco foram levadas em consideração no planejamento do caminho.

Três problemas relacionado a perda de visibilidade do marco foram tratados:

- distância máxima e mínima entre os robôs;
- ângulo de visão da câmera.
- ângulo relativo entre os robôs;

O quesito distância é importante porque está relacionado diretamente com o tamanho que o marco aparece na imagem. Quanto mais distante estiverem os robôs menor será o tamanho da marca dificultando sua detecção. Em antítese, ou seja, quando os robôs estiverem muito próximos problemas relacionados a movimentos bruscos do humanóide podem influir na detecção do marco. Para ilustrar, a figura 3.9 mostra imagens do robô humanóide capturadas a distâncias de 20cm e 40cm respectivamente e imagem capturada com o humanóide em movimento.



(a) Imagem a 20cm



(b) Imagem a 40cm



(c) Imagem com o humanóide em movimento

Figura 3.9: Imagens do robô humanóide.

Outro quesito tratado foi o ângulo de visão da câmera. Para a identificação efetiva do marco é necessário que ele esteja contido totalmente na imagem capturada pela câmera.

A figura 3.10 apresenta uma situação em que o marco do humanóide está parcialmente contido na imagem capturada.

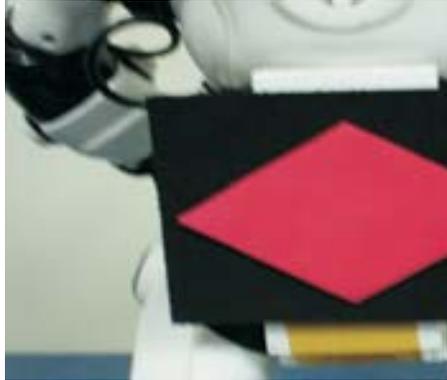


Figura 3.10: Imagem do robô humanóide.

Como mostra a figura 3.11 o ângulo relativo (α_{rel}) entre os robôs pode ser dado pela equação 3.2:

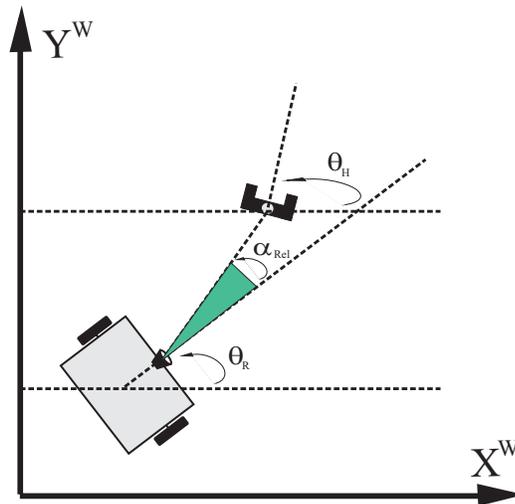


Figura 3.11: Imagem do robô humanóide.

$$\alpha_{rel} = \theta_H - \theta_R \quad (3.2)$$

onde θ_H e θ_R são os ângulos de orientação do humanóide e do robô com rodas respectivamente.

O aumento do módulo do ângulo relativo entre os robôs representa problema para a identificação do marco devido a projeção da marca no plano 2D tornar-se cada vez menor. Para caracterizar este problema a figura 3.12 mostra imagens do humanóide rotacionado de 45° nos sentidos horário e anti-horário.

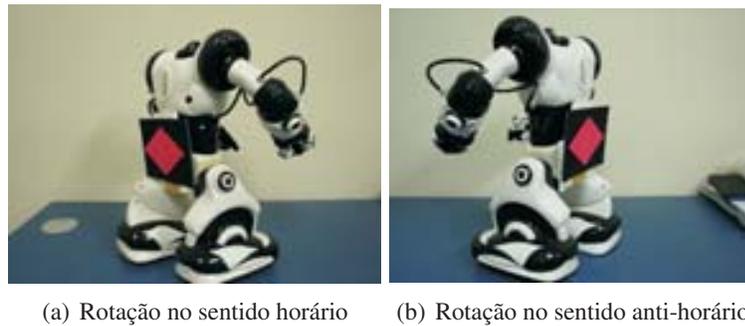


Figura 3.12: Imagens do robô humanóide.

As técnicas tradicionais de planejamento calculam caminhos sem levar em consideração as restrições não-holonômicas dos robôs. Normalmente o caminho é calculado e posteriormente é adaptado de forma a incorporar tais restrições.

A aplicação destas técnicas para o sistema robótico utilizado neste trabalho torna-se inviável devido ao fato do sistema ser formado por dois robôs heterogêneos.

A proposta de um planejador que calcula caminho respeitando as restrições não-holonômicas dos dois robôs e as restrições de visibilidade do sistema é apresentado na seção seguinte.

3.3 Solução proposta

A fim de evitar os problemas elencados na seção anterior, o planejador desenvolvido considera o sistema (humanóide e robô com rodas) como uma estrutura variável sujeita a restrições de visão. A figura 3.13 mostra como o sistema foi modelado bem como apresenta as restrições impostas.

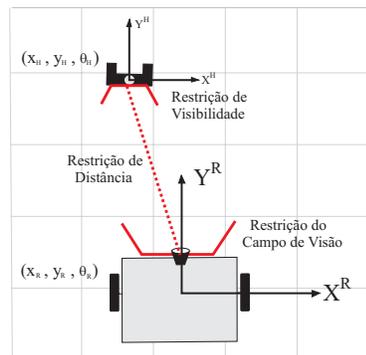


Figura 3.13: Restrições do Sistema.

Fazendo uma equivalência com a seção anterior a restrição de distância refere-se ao problema das distâncias máxima e mínima. A restrição do campo de visão refere-se ao

problema do ângulo de visão da câmera e a restrição de visibilidade ao problema do ângulo relativo entre os robôs.

O planejador desenvolvido monta um grafo onde cada nó possui, dentre outras informações, a pose do humanóide (x_H, y_H, θ_H) e a pose do robô com rodas (x_R, y_R, θ_R) . A fase de expansão do grafo é feita considerando que o conjunto de movimentos dos robôs é um conjunto de oito elementos. A figura 3.14 ilustra o comentado.

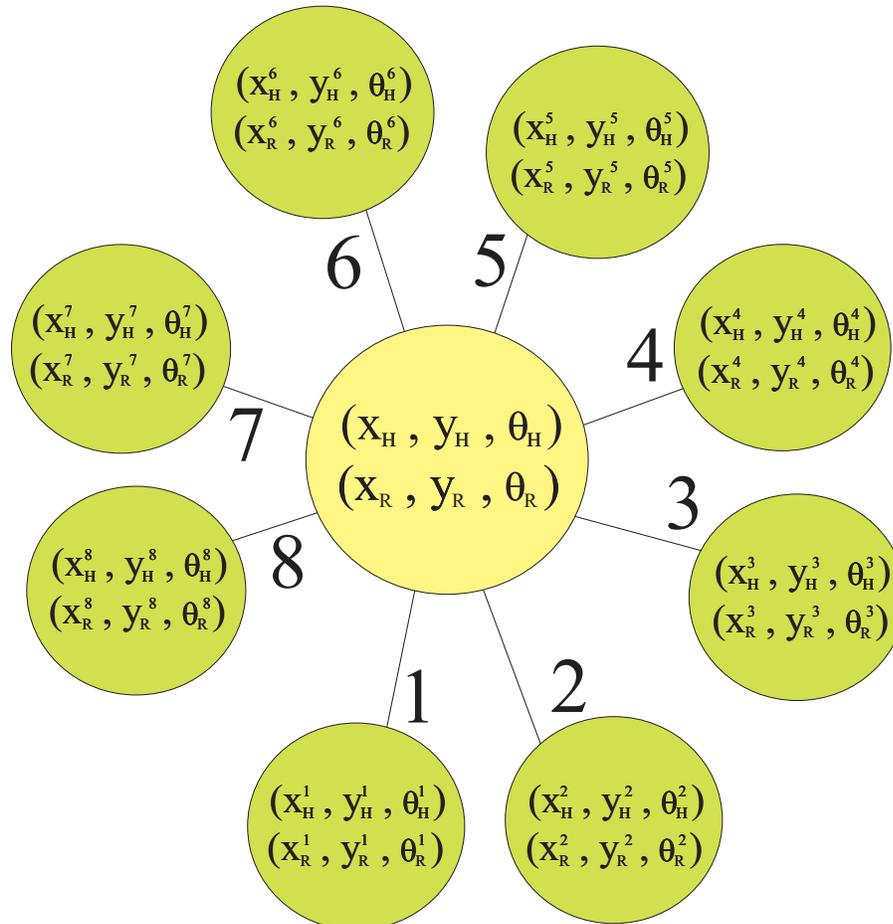


Figura 3.14: Restrições do Sistema.

Na figura 3.14 considere:

- (1): Movimento para frente do humanóide;
- (2): Movimento para frente do robô com rodas;
- (3): Movimento de giro do humanóide no sentido horário;
- (4): Movimento de giro do robô com rodas no sentido horário;
- (5): Movimento para trás do humanóide;
- (6): Movimento para trás do robô com rodas;
- (7): Movimento de giro do humanóide no sentido anti-horário;
- (8): Movimento de giro do robô com rodas no sentido anti-horário;

As restrições de visão do sistema foram incorporadas na etapa de validação do novo nó gerado. Tradicionalmente nesta etapa é realizado o teste que avalia junto ao mapa se o novo nó é um obstáculo, caso seja, este nó não é incorporado ao grafo. No planejador proposto a validação do nó é feita testando se a pose do humanoíde ou a pose do robô com rodas está dentro de um obstáculo e se obedece as seguintes condições:

$$|\alpha_{rel} - \theta_H| < \alpha_{max}^{cam} \quad (3.3)$$

$$|\alpha_{rel} - \theta_H| < \alpha_{max}^{mar} \quad (3.4)$$

$$\sqrt{(x_H - x_R)^2 + (y_H - y_R)^2} > D_{min} \quad (3.5)$$

$$\sqrt{(x_H - x_R)^2 + (y_H - y_R)^2} < D_{max} \quad (3.6)$$

- D_{max} e D_{min} : distâncias máxima e mínima entre os robôs respectivamente;
- α_{rel} : ângulo relativo entre os dois robôs;
- α_{max}^{cam} : ângulo máximo de visibilidade da câmera;
- α_{max}^{mar} : ângulo máximo que permite a marca ser identificada;

Caso o nó expandido não obedeça alguma das condições apresentadas ele é considerado um obstáculo virtual não é incorporado ao grafo.

Fazendo uma analogia com as restrições definidas na seção anterior a equação 3.3 refere-se à restrição do campo de visão, a equação 3.4 refere-se à restrição de visibilidade e as equações 3.5 e 3.6 referem-se à restrição de distância.

Uma vez que o grafo foi montado a busca pelo caminho é realizada utilizando o algoritmo A^* .

A heurística implementada no algoritmo é baseada em distância euclidiana (equação 3.8). Seguindo as definições apresentadas para compor a equação 3.1 e o conjunto de movimentos dos robôs, as funções $h(v)$ e $g(v)$ são definidas pelas equações 3.7 e 3.8 respectivamente.

$$h(v) = \sqrt{(x_H^{fim} - x_H^{atu})^2 + (y_H^{atu} - y_H^{ini})^2} \quad (3.7)$$

$$g(v) = \begin{cases} g(v)^* + K_1 \cdot \Delta L_{Hum} & se, Humanoide Frente \\ g(v)^* + K_2 \cdot \Delta L_{Hum} & se, Humanoide Tras \\ g(v)^* + K_3 \cdot \Delta \theta_{Hum} \cdot (L_{Hum}/2) & se, Humanoide Horario \\ g(v)^* + K_3 \cdot \Delta \theta_{Hum} \cdot (L_{Hum}/2) & se, Humanoide Anti-Horario \\ g(v)^* + K_4 \cdot \Delta L_{Rob} & se, RobRodas Frente \\ g(v)^* + K_5 \cdot \Delta L_{Rob} & se, RobRodas Tras \\ g(v)^* + K_6 \cdot \Delta \theta_{Rob} \cdot (L_{Rob}/2) & se, RobRodas Horario \\ g(v)^* + K_6 \cdot \Delta \theta_{Rob} \cdot (L_{Rob}/2) & se, RobRodas Anti-Horario \end{cases} \quad (3.8)$$

Onde:

- $g(v)^*$: custo acumulado;
- ΔL_{Hum} : passo de movimento linear do humanóide;
- $\Delta \theta_{Hum}$: passo de movimento angular do humanóide;
- L_{Hum} : largura do humanóide;
- ΔL_{Rob} : passo de movimento linear do robô com rodas;
- $\Delta \theta_{Rob}$: passo de movimento angular do robô com rodas;
- L_{Rob} : largura do robô com rodas;
- K : constantes para dimensionar os pesos dos movimentos;

As constantes K_i ($i = 1, 2, \dots, 6$) foram definidas a fim de ponderar os movimentos. Por exemplo, para priorizar o movimento para frente do humanóide em relação ao seu movimento para trás basta fazer $K_1 > K_2$.

É necessário ressaltar que na implementação foram feitas duas discretizações: uma para tornar finita a quantidade de sucessores de um nó e outra para tornar o grafo finito.

Para gerar um conjunto finito de sucessores foram quantizados os movimentos dos robôs em um conjunto de oito elementos. E a fim de tornar o grafo finito foi considerado que duas poses são iguais se a distância euclidiana entre elas for menor que um limiar.

Para ilustrar o planejador proposto considere o exemplo hipotético apresentado na figura 3.15 que contém dois robôs. Um humanóide que pode mover-se para frente, para trás, para a direita e para a esquerda e um robô com rodas que se move exclusivamente para frente e para trás.

Considere também que distância máxima que permita visada da marca seja de 45cm e que o passo de movimento linear dos robôs seja de 10cm . Intuitivamente a escolha do próximo movimento a ser executado seria mover o humanóide para frente. Porém, o planejador não considera este nó porque ele infringiu a restrição de distância máxima. A escolha, então, dá-se por mover o robô com rodas para frente (figura 3.16).

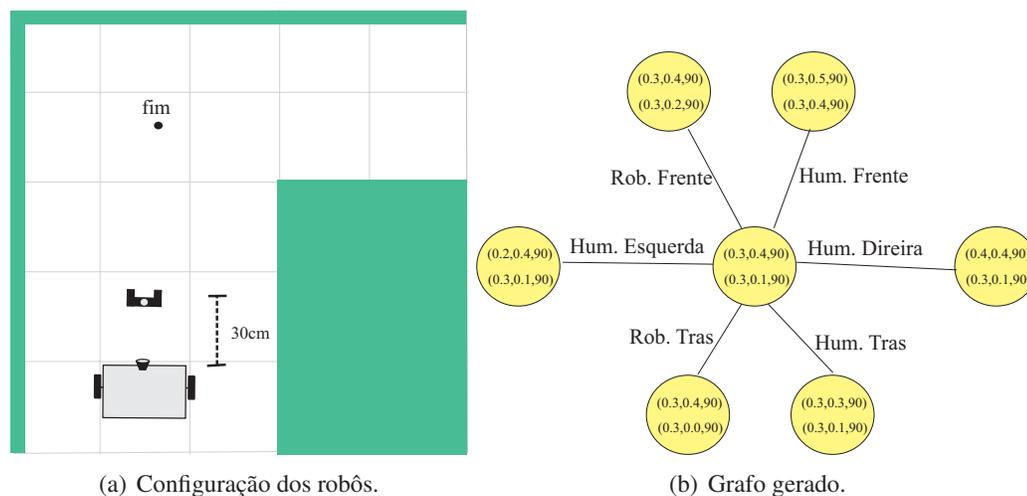


Figura 3.15: Exemplo do planejador.

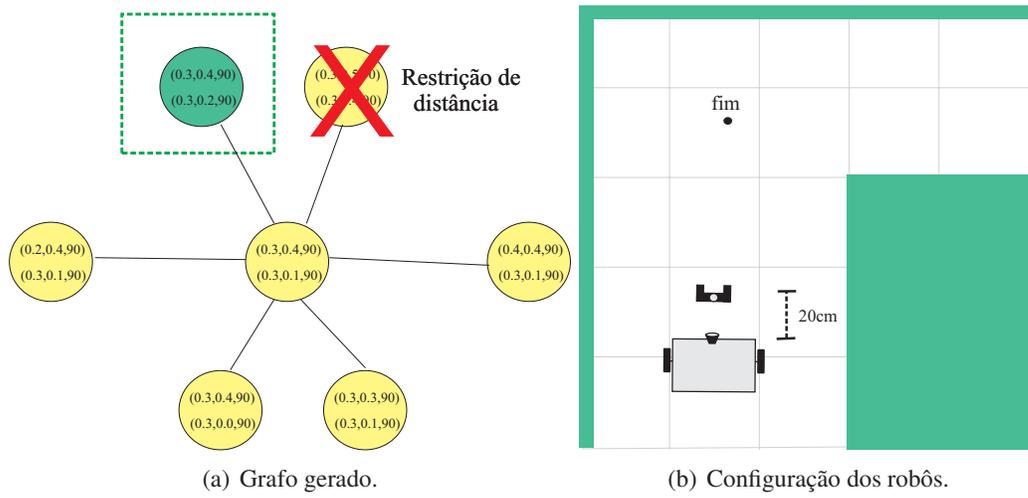


Figura 3.16: Exemplo do planejador.

Capítulo 4

Resultados

Este capítulo apresenta na seção 4.1 os resultados do sistema de localização absoluta do robô com rodas, na seção 4.2 os resultados da localização relativa do humanoíde e, por fim, na seção 4.3 os resultados obtidos do planejamento de caminho.

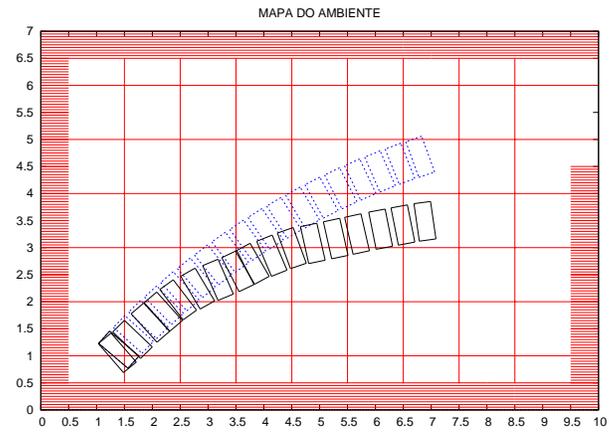
4.1 Localização absoluta do robô com rodas

As situações que serão apresentadas nesta seção foram obtidas por simulação. Procurou-se utilizar a medida dos ruídos dos sensores condizentes com a realidade. Para tal, nos *encoders* foi implementado um ruído branco gaussiano com desvio padrão proporcional à quantidade de pulsos lidos, e para a identificação dos parâmetros das retas ρ e α o desvio padrão do ruído também obedece uma proporção que é regida pelo tamanho que a reta aparece na imagem.

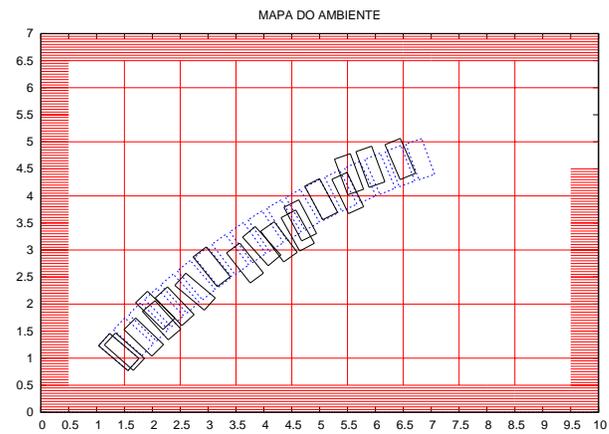
Neste trabalho foram implementados três sistemas de localização:

- Sistema de Localização por Odometria: neste sistema são utilizados somente as informações oriundas dos *encoders* para localizar o robô;
- Sistema de Localização por correção geométrica: neste sistema, a cada passo as linhas são identificadas e usadas para calcular a pose do robô utilizando trigonometria. Quando não há retas identificadas a correção é feita por odometria;
- Sistema de Localização utilizando Filtro de Kalman: neste sistema as informações dos odômetros são fundidas com as características das retas identificadas a fim de obter a pose do robô com a maior confiabilidade possível;

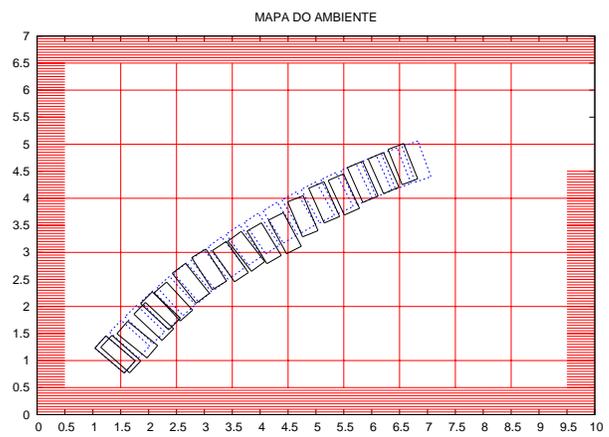
A figura 4.1 apresenta o resultado dos três sistemas de localização. Para as figuras, o retângulo tracejado representa a pose real do robô e o retângulo contínuo a pose calculada.



(a) Localização por Odometria.



(b) Localização por Odometria e Correção Geométrica.



(c) Localização utilizando Filtro de Kalman, fundindo odometria e detecção de marcos.

Figura 4.1: Resultados dos sistemas de localização implementados.

Para comparar o sistema de localização por correção geométrica com o sistema de localização utilizando Kalman foi plotado o gráfico do erro seguindo a média euclidiana apresentada na equação 4.1:

$$Erro = \sqrt{(x_{Real} - x_{Calc})^2 + (y_{Real} - y_{Calc})^2} \quad (4.1)$$

onde, x_{Real} e y_{Real} representam a localização (x, y) real do robô e x_{Calc} e y_{Calc} a localização calculada pelo sistema de localização.

A figura 4.2 apresenta o gráfico do erro. A linha tracejada representa o erro calculado utilizando o sistema de localização por correção geométrica e a linha cheia o erro calculado utilizando o sistema de localização com Kalman.

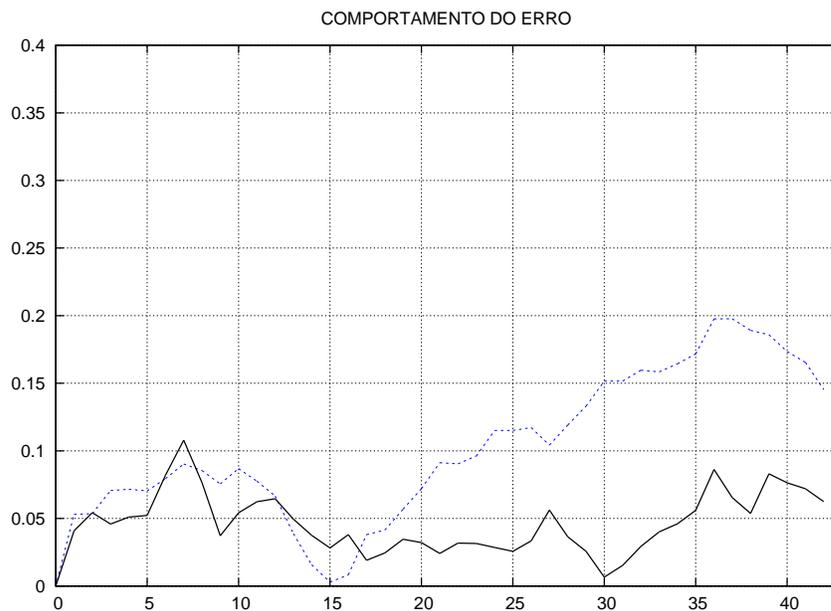
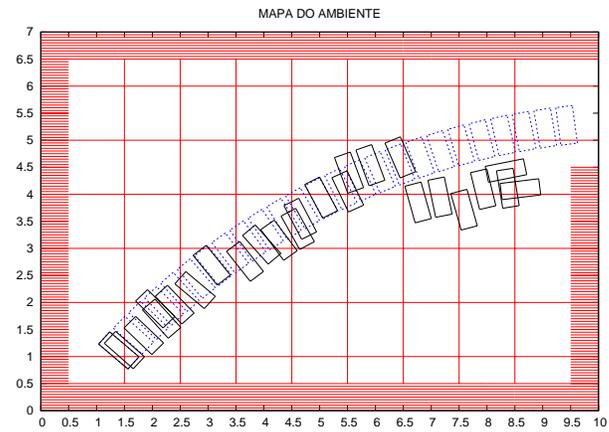


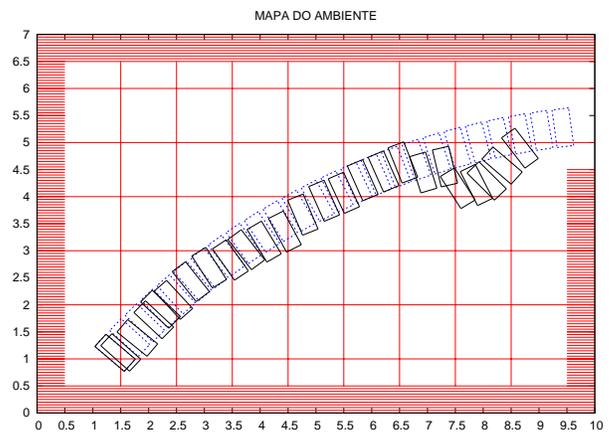
Figura 4.2: Gráfico comparativo do sistema de localização por correção geométrica com o sistema de localização utilizando filtro de Kalman.

Uma situação em particular foi implementada para testar a robustez dos sistemas de localização. Para tal, uma perturbação (ruído branco gaussiano de variância dez vezes maior que a utilizada) foi adicionada ao sistema quando o robô se aproxima da posição (6.5,4.5). Os resultados encontrados utilizando geometria e Kalman são exibidos na figura 4.3.

Analisando a figura 4.1 percebe-se que o comportamento do sistema de localização utilizando o filtro de Kalman mostrou-se mais satisfatório que os utilizando odometria e correções geométricas. No tocante a rejeições de perturbações (figura 4.3), o sistema baseado no filtro também mostrou-se eficiente pois tende a retornar à pose real, enquanto que o sistema baseado em correção geométrica não conseguiu o mesmo resultado.



(a) Efeito da perturbação na Localização por Geometria.



(b) Efeito da perturbação na Localização utilizando Filtro de Kalman

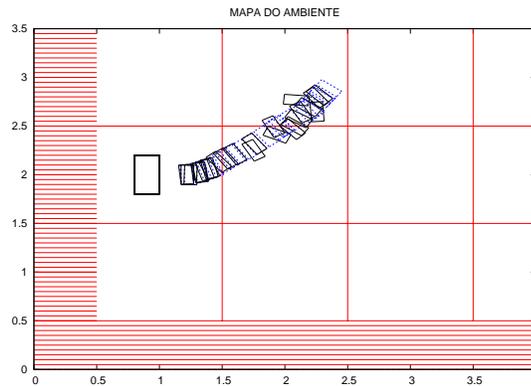
Figura 4.3: Efeitos da perturbação nos sistemas de localização.

4.2 Localização relativa do humanóide

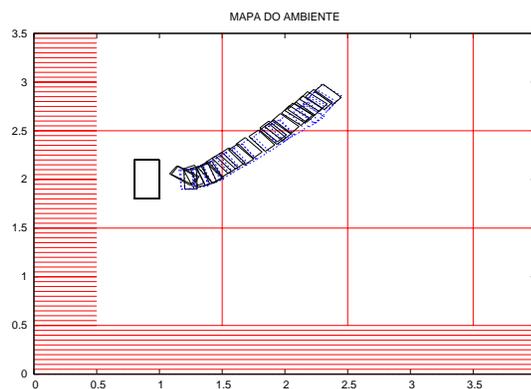
As situações que serão apresentadas nesta seção foram obtidas por simulação. Procurou-se utilizar a medida dos ruídos dos sensores condizentes com a realidade. Para tal, na medida das diagonais foi implementado um ruído com desvio padrão proporcional ao tamanho com que elas aparecem na imagem.

Neste trabalho foram implementados dois sistemas de localização relativa, um utilizando correções geométricas e outro fundindo a pose do robô com rodas com as informações sensoriais em um filtro de Kalman.

As figuras 4.4 e 4.5 apresentam respectivamente os resultados dos dois sistemas de localização considerando o robô com rodas parado (câmera fixa) e em movimento (câmera móvel). Para as figuras o retângulo maior representa o robô com rodas, o retângulo menor tracejado a pose real do humanóide, o retângulo menor contínuo a pose do humanóide calculada.

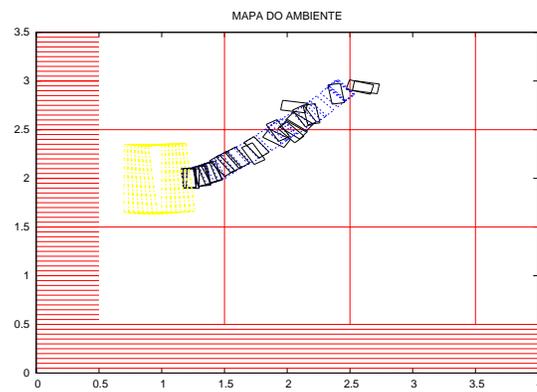


(a) Localização Relativa por Odometria e Correção Geométrica.

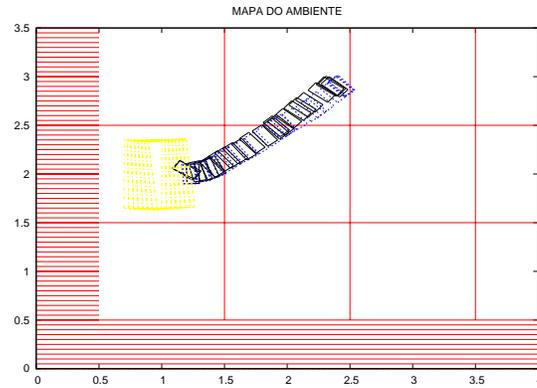


(b) Localização Relativa utilizando filtro de Kalman fundindo as informações sensoriais com a pose do robô com rodas.

Figura 4.4: Localização Relativa com a câmera parada.



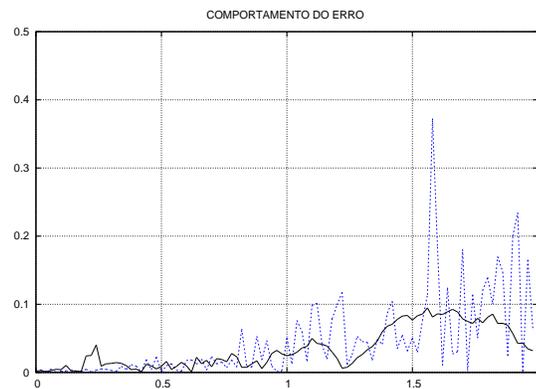
(a) Localização Relativa por Odometria e Correção Geométrica.



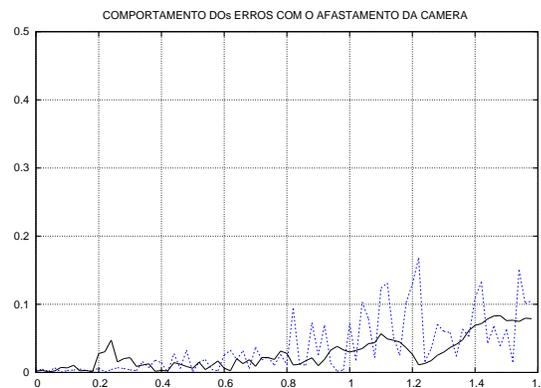
(b) Localização Relativa utilizando filtro de Kalman fundindo as informações sensoriais com a pose do robô com rodas.

Figura 4.5: Localização Relativa com a câmera em movimento.

De acordo com a equação 4.1, a figura 4.6 apresenta o comportamento do erro para as duas situações apresentadas nas figuras 4.4 e 4.5. A linha tracejada representa o erro calculado utilizando o sistema de localização por correção geométrica e a linha cheia o erro calculado utilizando o sistema de localização com filtro de Kalman.



(a) Erro: robô com rodas parado.



(b) Erro: robô com rodas em movimento.

Figura 4.6: Comportamento do erro nos sistemas de localização relativa.

Analisando as figuras percebe-se que o sistema de localização utilizando a filtragem de Kalman comportou-se de forma mais satisfatória que o sistema utilizando correção geométrica. O comportamento do erro, como esperado, aumenta à medida que a distância entre os robôs é maior. Porém, mesmo com este aumento, o sistema de localização utilizando filtro de Kalman comportou-se de forma mais satisfatória que o sistema de localização utilizando correção geométrica.

4.3 Planejamento de caminho

4.3.1 Considerações iniciais

Os coeficientes definidos na modelagem do planejador de caminho foram dimensionados da seguinte forma:

- L_{Hum} foi encontrado considerando o humanóide como circunferência e medindo seu diâmetro;
- ΔL_{Hum} e $\Delta \theta_{Hum}$ foram obtidos através de dois experimentos, um que consistia em mover várias vezes para frente o humanóide e a partir das medidas extrai-se a média do valor distância percorrida por unidade de tempo, e o outro que fazia o humanóide girar várias vezes em um mesmo sentido e calculava-se, em média, o valor do deslocamento angular por unidade de tempo;
- Os valores de L_{Rob} , ΔL_{Rob} e $\Delta \theta_{Rob}$ foram obtidos de forma similar aos do humanóide porém utilizando o robô com rodas;
- α_{max}^{mar} , D_{max} foram estimados com base no trabalho de Nogueira (2005) e
- α_{max}^{cam} e D_{min} foram medidos por experimentação utilizando o sistema de visão que está sendo desenvolvido para o robô com rodas.

Os valor dos coeficientes utilizados foram:

- $L_{Hum} = 0.4m$;
- $L_{Rob} = 0.6m$;
- $\Delta L_{Hum} = 0.02m$;
- $\Delta L_{Rob} = 0.02m$;
- $\Delta \theta_{Hum} = 30^\circ$;
- $\Delta \theta_{Rob} = 30^\circ$;
- $\alpha_{max}^{cam} = 45^\circ$;
- $\alpha_{max}^{mar} = 45^\circ$;
- $D_{max} = 0.6m$;
- $D_{min} = 0.2m$;

Ainda sobre o planejador, é necessário comentar que foi utilizado um mapa, conhecido *a priori*, de dimensões $7.0m$ de largura por $6.0m$ de comprimento contendo três obstáculos fixos como mostra a figura 4.7.

A seguir serão apresentados os resultados do planejamento de caminho utilizando a idéia de cenários para caracterizar o problema.

Um cenário é descrito pela tarefa de calcular um caminho para o humanóide de uma pose inicial a uma pose final levando em consideração a pose inicial do robô com rodas e o valor das constantes de ponderação.

Dois cenários foram utilizados: o Cenário 1 para apresentar o problema de mínimos locais e o Cenário 2 para apresentar os robôs movendo-se perto de uma região de quina.

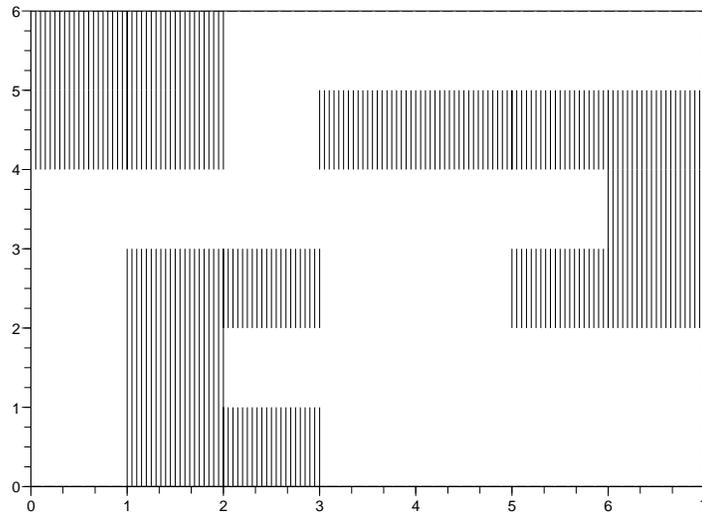


Figura 4.7: Mapa utilizado no planejamento de caminho.

4.3.2 Cenário 1

- Pose inicial do humanóide: $(0.2, 0.5, 90)$;
- Pose final do humanóide: $(5.0, 1.0, -90)$;
- Pose inicial do robô com rodas: $(0.2, 0.2, 90)$;
- Constantes utilizadas: $K_1 = K_2 = K_3 = K_4 = K_5 = K_6 = 1$.

O resultado do cenário 1 é exibido na figura 4.8. Nela, o caminho apresentado na forma de pontos é o do humanóide e o caminho apresentado de forma tracejada é o caminho calculado para o robô com rodas.

A figura 4.9 apresenta um trecho resultado do cenário 1 plotado de forma que os pontos calculados para o humanóide e para o robô com rodas são ligados por um vetor. A origem do vetor é um ponto calculado para o robô com rodas e a extremidade um ponto calculado para o humanóide.

4.3.3 Cenário 2

- Pose inicial do humanóide: $(0.2, 0.5, 90)$;
- Pose final do humanóide: $(7.0, 6.5, 0)$;
- Pose inicial do robô com rodas: $(0.2, 0.2, 90)$;
- Constantes utilizadas: $K_1 = K_3 = K_4 = K_6 = 1$ e $K_2 = K_5 = 3$.

O resultado do cenário 2 é exibido na figura 4.10. Como legenda para a figura adote que o caminho apresentado na forma de pontos é o do humanóide e o caminho apresentado de forma tracejada é o caminho calculado para o robô com rodas.

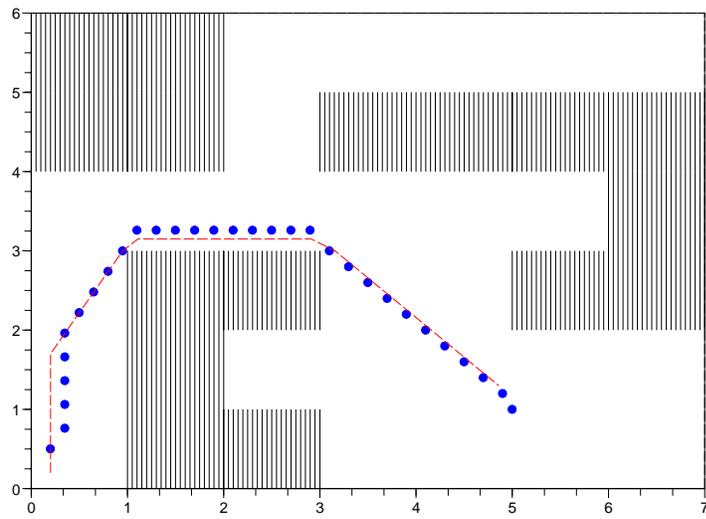


Figura 4.8: Caminho planejado: cenário 1.

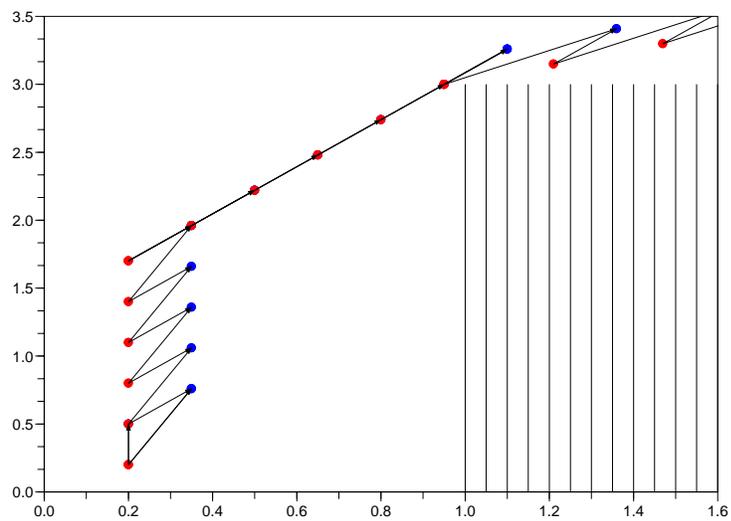


Figura 4.9: Caminho planejado: cenário 1.

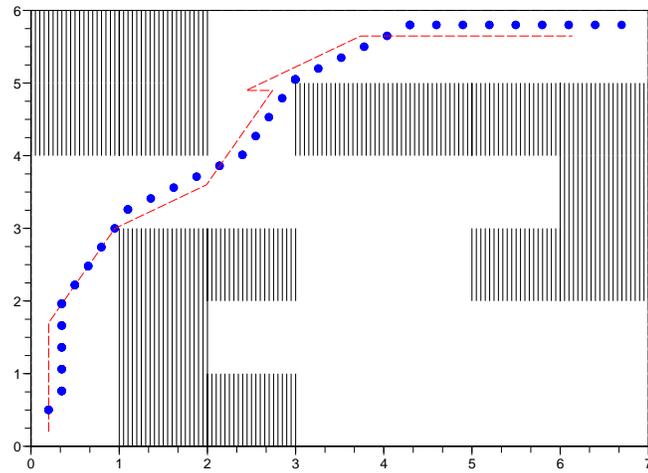


Figura 4.10: Caminho planejado: cenário 2.

A figura 4.11 apresenta um trecho resultado do cenário 2 plotado de forma que os pontos calculados para o humanóide e para o robô com rodas são ligados por um vetor. Analisando a figura, mais especificamente quando o humanóide encontra-se próximo do ponto (3.0,5.0), percebe-se que o robô com rodas fez um movimento para trás, mesmo com o aumento no custo final que este tipo de movimento acumula, por ser a única possibilidade de respeitar as restrições visuais.

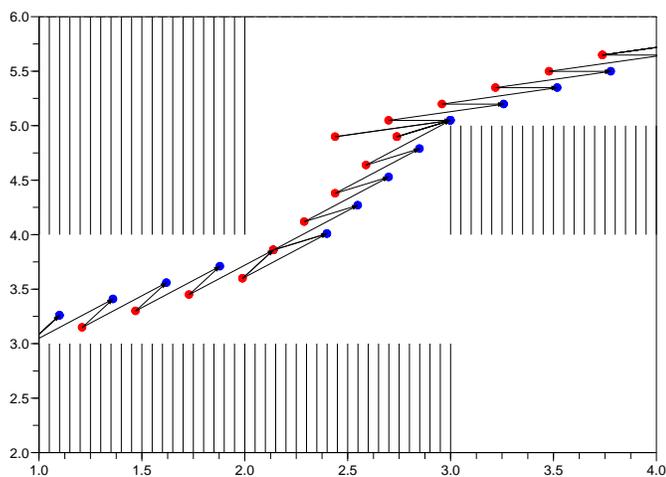


Figura 4.11: Caminho planejado: cenário 2.

Capítulo 5

Conclusões e Trabalhos Futuros

Esse trabalho apresentou os sistemas de localização e planejamento de caminho para um sistema robótico formado por um humanóide não instrumentado e um robô escravo com rodas.

Como principais contribuições tem-se: a modelagem do sensor ótico feita de forma a proporcionar a utilização dos parâmetros nativos do processamento de imagem diretamente na formulação de Kalman e um planejador de caminhos que respeita as restrições não holonômicas dos dois robôs e incorpora as restrições de visibilidade existentes no sistema robótico utilizado.

Dois foram os sistemas de localização implementados: um de localização absoluta para o robô com rodas e outro de localização relativa para o humanóide.

A localização do robô com rodas foi feita utilizando seu modelo de odometria juntamente com a detecção de marcos naturais (linhas que compõem o chão do ambiente) fundidos em um Filtro de Kalman Extendido. Com fim comparativo, outros dois sistemas de localização absoluta também foram implementados: localização por odometria e localização por correção geométrica. A comparação entre esses sistemas foi feita e foi verificado a melhor eficiência do sistema que utiliza filtro de Kalman.

A implementação do sistema de localização absoluta do humanóide utiliza as características de um marco existente nas suas costas (um losango) juntamente com a pose (já corrigida) do robô com rodas fundidas em um outro Filtro de Kalman Extendido. Um sistema de localização absoluta utilizando correção geométrica foi implementado para que o sistema usando Kalman pudesse ser comparado. Os resultados obtidos com o método proposto foram semelhantes aos do sistema de localização absoluta, ou seja o sistema de localização utilizando filtro de Kalman mostrou-se mais eficiente.

No tocante ao planejamento de caminho foi apresentado um planejador capaz de calcular caminhos para o sistema robótico obedecendo as restrições não holonômicas dos dois robôs, bem como às restrições de visibilidade. Os resultados obtidos foram satisfatórios porém o planejador mostrou ser custoso computacionalmente.

Como trabalhos futuros para os sistemas desenvolvidos pretende-se:

Localização

- implementar outras formulações do Filtro de Kalman, por exemplo, o Filtro de Kalman com Observações Parciais;
- substituir o filtro de Kalman por um Filtro de Partículas, pois este incorpora mais facilmente as não-linearidades do problema além de tratar com ruídos não gaussianos;
- evoluir esta estratégia de localização para uma proposta de SLAM (*Simultaneous Localization and Mapping*), de forma que o robô seja capaz de fazer sua localização sem um conhecimento prévio do mapa e simultaneamente mapear o ambiente onde navega.

Planejamento de caminho

- melhorar a heurística utilizada;
- investigar outras técnicas de busca em grafo que possam proporcionar a implementação do planejador em tempo real;
- expandir a idéia para ambientes dinâmicos;

Referências Bibliográficas

- Adôrno, B. V., C. S. R. Aguiar & G. A. Borges (2005), 'Planejamento de trajetória para o robô omni utilizando o algoritmo mapa de rotas probabilístico', *Simpósio Brasileiro de Automação Inteligente* .
- Aiube, F. A. L., T. K. N. Baidya & E. A. H. Tito (2006), 'Processos estocásticos dos preços das commodities: uma abordagem através do filtro de partículas', *Rev. Bras. Econ.* vol.60 no.3 Rio de Janeiro.
- Alsina, Pablo Javier, Luiz Marcos Garcia Gonçalves, Adelar Adelino Dantas Medeiros, Diogo Pinheiro Fernandes Pedrosa & Frederico Carvalho Vieira (2002), Navegação e controle de robôs móveis, *em* 'Mini Curso - XIV Congresso Brasileiro de Automação'.
- Anônimo (2007), 'Robosapiens', <http://www.robosapien.com> - Visitada em Julho de 2006.
- Bezerra, C. G. (2004), Localização de um robô móvel usando odometria e marcos naturais, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.
- Borenstein, J. & L. Feng (1996), 'Measurement and correction of systematic odometry errors in mobile robots', *IEEE Transactions on Robotics and Automation* .
- Borges, G. A., M. J. Aldon, V. H. Alcalde & L. R. Suarez (2005), 'Local map building for mobile robots by fusing laser range finder and monocular video images', *Simpósio Brasileiro de Automação Inteligente* .
- Canny, J. F. (1988), *The complexity of Robot Motion Planning*, MIT Press.
- Carelli, R., F. R. Roberti, R. F. Vassallo & T. F. Bastos (2006), 'Estrategia de control de formaciones estable para robots móviles', *XX Congreso Argentino de Control Automático* .
- Chen, M. W. & A. M. S. Zalzal (1997), 'Dynamic modelling and genetic-based trajectory generation for non-holonomic mobile manipulators', *Control Engineering Practice* .
- Choset, H. & J. Burdick (1995), 'Sensor based planning: Incremental construction of the generalized voronoi graph', *Proceedings of IEEE International Conference of Robotics and Automation* .

- Choset, H., K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki & S. Thrun (2005), *Principles of Robot Motion: Theory, Algorithms and Implementations*, MIT Press.
- Craig, J. J. (2004), *Introduction to Robotics: Mechanics and Control*, Pearson Education.
- Díaz de León, J. L. & J. H. Sossa (1998), 'Automatic path planning for a mobile robot among obstacles of arbitrary shape', *Proceedings of IEEE Transactions on Systems* .
- de la Cruz, C., R. Carelli & C. C. Gava (2006), 'Control centralizado de formacion usando una cámara omnidireccional', *Jornadas Argentinas de Robótica* .
- Divelbiss, A. W. & J. Wen (1992), 'A global approach to nonholonomic motion planning', *Proceedings of the 31 st Conference on Decision and Control* .
- Divelbiss, A. W. & J. Wen (1997), 'A path space approach to nonholonomic motion planning in the presence of obstacles', *IEEE Transaction on Robotics and Automation* .
- Fairfield, N. (2002), 'Simple landmark localization on a three-layer mobile robot'.
- Fujimori, A., T. Fujimoto & G. Bohats (2005), 'Distributed leader-follower navigation of mobile robots', *Intemational Conference on Control and Automation* .
- Gava, C. C., R. F. Vassallo, R. Carelli & T. F. Bastos (2006), 'A nonlinear control applied to team formation based on omnidirectional vision', *Internacional Symposium on Industrial Electronics* .
- Ghabcheloo, R., A. Pascoal, C. Silvestre & I. Kaminerz (2005), 'Coordinated path following control of multiple wheeled robots using linearization techniques', *International Journal of Systems Science* (6), 399–414.
- Gonzales, R. C. (2000), *Processamento de Imagens Digitais*, Edgard Blucher.
- Han, Q., Y. Huang, J. Yuan & Y. Kang (2004), 'A method of path planning for tractor-trailer mobile robot based on the concept of global-width', *Proceedings of the 5 World Congress on Intelligent Control and Automation* .
- Hoff, K. E., T. Culver, J. Keyser, M. C. Lin & D. Manocha (2000), 'Interactive motion planning using hardware-accelerated computation of generalized voronoi diagrams', *Proceedings of IEEE International Conference of Robotics and Automation* pp. 2931–2937.
- Hughes, R. C. (1986), 'Enhanced single camera photogrammetry algorithms for real-time control applications', *In Proc. Int. Soc. Photogrammetry and remote sensing* .
- Jang, G., S. Kim, Lee, W. & I. Kweon (2002), 'lor landmark based self-localization for indoor mobile robots', *International Conference on Robotics and Automation* .

- Jim, T., S. Park & J. Lee (2003), 'A study on position determination for mobile robot navigation in an indoor environment', *Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation* .
- Junior Valdir, G. (2006), *Arquitetura Híbrida para Robôs Móveis Baseada em Funções de Navegação com Interação Humana*, Tese de doutorado, Escola Politécnica da Universidade de São Paulo.
- Kalman, R. E. (1960), 'A new approach to linear filtering and predictive problems', *Transactions ASME, Journal of basic engineering* .
- Kelly, A. (2001), 'Linearized systematic error propagation in vehicle odometry', *International Conference on Intelligent Robotics and Systems* .
- Khatib, O. (1985), 'A real-time obstacle avoidance for manipulators and mobile robots', *Proceedings of IEEE International Conference of Robotics and Automation* pp. 500–505.
- Kiryi, E. & M. Buehler (2002), 'Three-state extended kalman filter for mobile robot localization', *Report Centre for Intelligent Machines* .
- Lamiroux, F., J. Laumond, C. Van Geem & G. Raust (2005), 'Trailer-truck trajectory optimization', *IEEE Robotics Automation Magazine* .
- Lamiroux, F., J. Laumond, C. Van Geem & G. Raust (2006), 'Experiments in multirobot coordination', *Robotics and Autonomous Systems* (54), 265–275.
- Lamiroux, F. & J. P. Laumond (1998), 'A practical approach to feedback control for a mobile robot with trailer', *IEEE Transaction on Robotics and Automation* .
- Latombe, J. C. (1991), *Robot Motion Planning*, Boston: Kluwer.
- Launay, F., A. Ohya & S. Yuta (2002), 'A corridors lights based navigation system including path definition using a topologically corrected map for indor mobile robots', *Proceedings IEEE International Conference on Robotics and Automation* .
- Lavalle, S. M. (2006), *Planning Algorithms*, Cambridge University Press.
- Liu, Z., J. Gao, P. Yang, D. Wang & L. Xing (2006), 'Path planning of tractor-trailer mobile robot in multi-grid circuit map', *Proceedings of the 6th World Congress on Intelligent Control and Automation* .
- Medeiros, Adelardo Adelino Dantas (1998a), Introdução à robótica, em 'XVII Encontro Nacional de Automática', Natal, RN, pp. 56–65.
- Medeiros, Adelardo Adelino Dantas (1998b), A survey of control architectures for autonomous mobile robots, em 'Journal of the Brazilian Computer Society - JBCS'.
- Nilsson, J. N. (1998), *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann.

- Nilsson, N. J. (1969), 'A mobile automation: An application of artificial intelligence techniques', *Proceedings of the 1st International Joint Conference on Artificial Intelligence* pp. 509–520.
- Nogueira, M. B. (2005), Posicionamento e movimentação de um robô humanóide utilizando imagens de uma câmera móvel externa, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, UFRN.
- Odakura, V., A. Costa & P. Lima (2004), 'Localização de robôs móveis utilizando observações parciais', *Simpósio da Sociedade Brasileira de Computação* .
- Pedrosa, D. P. F. (2001), Sistema de navegação para robôs móveis autônomos, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, UFRN.
- Perez, T. L. (1983), 'Spatial planning: A configuration space approach', *IEEE Transactions on Computers* **2**, 108–120.
- Pimenta, L. C. A., G. A. S. Pereira, R. C. Mesquita, W. M. Caminhasy & M. F. M. Campos (2004), 'Elementos finitos na navegação de robôs móveis', *Congresso Brasileiro de Automática* .
- Russell, S. & P. Norvig (2004), *Inteligência Artificial*, Editora Campus - Tradução da 2ª Edição.
- Saedan, M., C. W. Lim & M. H. Ang (2006), 'Vision-based localization using a central catadioptric vision system', *Sym. on Experimental Robotics* .
- Shao, W. & D. Terzopoulos (2005), 'Interactive motion planning using hardware-accelerated computation of generalized voronoi diagrams', *Eurographics symposium on Computer Animation* pp. 19–28.
- Silva, L. R. (2003), Análise e programação de robôs móveis autônomos da plataforma eyebot, Dissertação de mestrado, Universidade Federal de Santa Catarina - UFSC.
- Soares, A. A. F. & P. J. Alsina (2000), 'Mobile robot obstacle avoidance based on potential field: a case study', *Anais do Congresso Nacional de Engenharia Mecânica* .
- Thrun, S., W. Burgard & Fox. D. (2005), *Probabilistic Robotics*, MIT Press.
- Tonouchi, Y., T. Tsubouchi & S. Arimoto (1994), 'Fusion of dead-reckoning positions with a workspace model for a mobile robot by bayesian inference', *International Conference on Intelligent Robots and Systems* .
- Udupa, S. (1977), Collision Detection and Avoidance in Computer Controlled Manipulator, Tese de doutorado, Department of Electrical Engineering, California Institute of Technology.

- Woodley, R. & L. Acar (2004), 'Autonomous control of a scale model of a trailer-truck using an obstacle- avoidance path-planning hierarchy', *Proceeding of the American Control Conference* .
- Xia, T. K., M. Yang & R. Q. Yang (2006), 'Vision based global localization for intelligent vehicles', *Proc. of the IEEE Intelligent Vehicles* .
- Yuan, J., Y. Huang & F. Sun (2004), 'Design for physical structure of tractor-trailer mobile robot', *IEEE International Conference on Robotics and Biomimetics* .
- Yuan, J., Y. Huang & Q. Han (2004), 'A strategy of path following control for wheeled mobile robots', *Proceedings of the 5 World Congress on Intelligent Control and Automation* .

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)