

UNIVERSIDADE ESTADUAL DO MARANHÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO E
SISTEMAS
MESTRADO EM ENGENHARIA DA COMPUTAÇÃO

EDUARDO HENRIQUE CASTRO MENDES

**Medição de posição, velocidade e aceleração utilizando
visão computacional em robôs móveis.**

São Luís, MA

2015

EDUARDO HENRIQUE CASTRO MENDES

MEDIÇÃO DE POSIÇÃO, VELOCIDADE E ACELERAÇÃO UTILIZANDO VISÃO COMPUTACIONAL EM ROBÔS MÓVEIS.

Dissertação apresentada ao Programa de Pós Graduação em Engenharia de Computação e Sistemas da Universidade Estadual do Maranhão, como requisito parcial para a obtenção do título de Mestre em Engenharia de Computação e Sistemas.

Aprovada em / /

BANCA EXAMINADORA

Prof. **Areolino de Almeida Neto**(Orientador)

Doutor em Engenharia Aeronáutica e Mecânica

Universidade Federal do Maranhão

Prof. **Luciano Buonocore**

Doutor em Engenharia Eletrônica e Computação

Universidade Federal do Maranhão

Prof^a. **Mauro Sérgio Silva Pinto**

Doutor em Engenharia Elétrica

Universidade Federal do Maranhão

Prof. **José Pinheiro de Moura**

Mestre em Engenharia Elétrica

Universidade Estadual do Maranhão

Mendes, Eduardo Henrique Castro

Medição de posição, velocidade e aceleração utilizando visão computacional em robôs móveis/ Eduardo Henrique Castro Mendes– São Luís, 2015.

74 f.

Impresso por computador (fotocópia).

Orientador: Areolino de Almeida Neto.

Dissertação (Mestrado) – Universidade Estadual do Maranhão, Programa de Pós-Graduação em Engenharia de Computação e Sistemas, 2015.

1. Medição– Robôs Móveis. 2. Localização. I. Título.

CDU 010.101.01: 010.010.1

Dedicatória

A Jesus Cristo, nosso senhor, por me amparar nos momentos difíceis, me dar força interior para superar as dificuldades, mostrar os caminhos nas horas incertas e me completar em todas as minhas necessidades.

À minha amada esposa, Glaciane, pelo amor, companheirismo, carinho, apoio incondicional, compreensão e ânimo.

Aos meus dois filhos, Vinicius e Thomaz, adolescentes, que trazem constantes motivação e incentivos para minha vida. Em especial, a ajuda técnica do Vinicius.

Ao meu orientador, Prof. Areolino, pela confiança, paciência, incentivo, amizade e orientações.

Aos professores participantes da banca examinadora que compartilharam deste momento importante: Prof. Luciano Buonocore, Prof. José Pinheiro e Prof. Reinaldo.

Agradecimentos

Agradeço a todos os professores do programa PECS - Programa de Pós-Graduação em Engenharia de Computação e Sistemas da Universidade Estadual do Maranhão, em especial ao Prof.^o. Henrique Mariano, fundador e incentivador do programa.

Resumo

Sistemas para navegação autônoma utilizando visão computacional referem-se à maneira como um robô pode movimentar-se e encontrar o seu caminho a partir da captura e processamento de imagens digitais. A maioria das aplicações relativas ao sistema de locomoção de um robô móvel envolve um sistema de medição e estimação da localização, da velocidade e da aceleração. Sua principal finalidade é obtenção de dados reais do ambiente e do robô, afim de garantir sua estimativa de estado. Portanto, este trabalho objetiva o desenvolvimento de um sistema para localização e medições de um robô móvel baseada em técnicas de visão computacional utilizando a biblioteca OpenCV. O sistema captura imagens de um robô móvel em um plano a partir de uma câmera montada no teto do ambiente onde irá navegar e após o processamento dessas imagens é realizada a computação da localização e medições das velocidades e acelerações. Assim, a solução proposta neste trabalho visa sua viabilidade como uma ferramenta de navegação robótica autônoma em ambientes estruturados.

Palavras-chaves: visão computacional, robô móvel, localização e medição de grandezas cinemáticas.

Abstract

Autonomous Navigation Systems using computer vision refers to the way a robot can move around and find its way from the digital images capture and processing. In many applications of mobile robotics there is the need to verify the location, velocity and acceleration of the displacement of a robot for obtaining dynamic results. Although it is fundamental, it is not an easy task to measure these quantities, as it requires a project for the trajectory control system. Therefore, this paper intends to develop a localization and measurement system of a mobile robot based on computer vision techniques using the OpenCV library. The system captures images of a mobile robot from a camera mounted on the ceiling and after processing these images it is performed the computation of localization and measurement of velocities and accelerations. As a consequence, the solution proposed in this paper aims its viability as an autonomous robotic navigation tool in structured environments.

Key-Words: computer vision, mobile robot, localization, measurement.

Lista de Figuras

Figura 1. Robô Curiosity (NASA).....	3
Figura 2. Robôs terrestres, aéreos, aquáticos e espaciais.....	8
Figura 3. Roda Fixa - 1 grau de liberdade.....	9
Figura 4. Roda com orientação no centro e 2 graus de liberdade.....	10
Figura 5. Roda castor wheel (rodinha de apoio).....	10
Figura 6. Roda sueca: três graus de liberdade.....	11
Figura 7. Robô omnidirecional.....	12
Figura 8. Robô Uniciclo Pioneer.....	12
Figura 9. Robô Triciclo Olin.....	13
Figura 10. Robô Quadriciclo.....	14
Figura 11. Sensor ótico por retro reflexão.....	16
Figura 12. Atuadores hidráulicos, pneumáticos e eletromagnéticos.....	17
Figura 13. Encoder Óptico.....	20
Figura 14. Sensores Inercial de Orientação (Acelerômetro e Inclinação).....	21
Figura 15. Módulo (GPS).....	22
Figura 16. Estrutura de um sistema de navegação baseada em visão.....	23
Figura 17. Deslocamento em uma dimensão.....	24
Figura 18. Velocidade média geométrica.....	24
Figura 19. Localização de um ponto em plano de duas dimensões.....	25
Figura 20. Movimento de uma partícula no plano xy.....	26
Figura 21. Centro de massa (m) em robôs com três rodas.....	27
Figura 22. Cinemática do robô móvel e disposição das rodas.....	28
Figura 23. Robô móvel Scisoccer Xbot.....	31
Figura 24. Diagrama de blocos da unidade móvel.....	32
Figura 25. Rádio base USB.....	32
Figura 26. Tranceiver RF Nordic.....	33
Figura 27. Pulso do fotodetector e indicação da largura de pulso (LP).....	33
Figura 28. Chassi do robô com três conjuntos motor-roda-encoders.....	34
Figura 29. Baterias e recarregador da plataforma robótica.....	34
Figura 30. Moto redutor e rodas do robô Scisoccer.....	35
Figura 31. Placa Mãe Robô.....	36

Figura 32. Placa Driver.....	36
Figura 33. Módulos acoplados na base móvel	36
Figura 34. <i>Software</i> Futebol robôs Sci-Soccer	37
Figura 35. Mensagens (PC –Rádio Base-Robô)	39
Figura 36. Imagem Digital	40
Figura 37. Níveis de intensidade de cinza.....	41
Figura 38. Imagem ampliada (5x5 pixels) em níveis de cinza	41
Figura 39. Níveis de intensidade das bandas RGB	41
Figura 40. Imagem das cores R, G, B e RGB	42
Figura 41. Representação de cores padrão HSI	42
Figura 42. Imagem binarizada.....	43
Figura 43. Fluxograma dos elementos de um (SVA).....	43
Figura 44. Estrutura básica da biblioteca OpenCV (BRADISK; KAEHLER, 2008)	47
Figura 45. Estrutura geral do sistema	49
Figura 46. Robôs acionados a partir do aplicativo desenvolvido	50
Figura 47. Exemplo de calibração por observação	51
Figura 48. Calibração sistema de visão em diferentes posições.....	52
Figura 49. Calibração e correção de distorção	52
Figura 50. Posicionamento do robô no ambiente	53
Figura 51. Sistema de calibração	53
Figura 52. Processamento dos parâmetros: (Δ_q) e (Δ_t)	55
Figura 53. Fluxograma do algoritmo de medição	56
Figura 54. Estrutura cvSmooth.....	57
Figura 55. Função cvInRangeS.....	58
Figura 56. Função cvMoments	58
Figura 57. Localização das coordenadas x e y do robô	59
Figura 58. Deslocamento em duas dimensões	59
Figura 59. Projeções do vetor velocidade	60
Figura 60. Localização x e y do robô no ambiente	62
Figura 61. Localização x e y do robô na imagem	62
Figura 62. Variação do deslocamento do robô no eixo x	63
Figura 63. Deslocamento do robô ao longo do eixo y	63
Figura 64. Velocidades do robô no ambiente	64
Figura 65. Acelerações do robô no ambiente.....	64

Lista de Tabelas

Tabela 1 - Exemplos de sensores e suas classificações	29
Tabela 2 - Resultado do sistema de calibração usando software imageJ.....	68
Tabela 3 - Parâmetros do sistema.....	87

Lista de Abreviaturas e Siglas

2D	Duas Dimensões
3D	Três Dimensões
API	Application Programming Interface
C/C++	Linguagem de programação
CCD	Charge Coupled Device
DLL	Dynamic Linked Library
Ghz	Giga Hertz
GFSK	Gaussian Frequency Shift Keying
Gb	Gigabyte
GPS	Global Positioning System
HD	Hard Disk
HSI	Hue, Saturation e Intensity
IHM	Interação Humano Computador
InovTec	Laboratório de Inovações Tecnológicas
LP	Largura de Pulso
LED	Light Emitting Diode
mA/h	Miliampere-hora
MSP430	Microcontrolador Texas Instruments
MATLAB	MATrixLABoratory
Ni-Mh	Níquel Metal Hidreto
OpenCV	Open Source Computer Vision Library
OV	Odometria Visual
PC	Personal Computer
Robocup	Liga internacional de futebol de robôs
RGB	Red, Green and Blue
RAM	Random Access Memory
RF	Radio Frequency
SmallSize/ F180	Categoria de futebol de robôs
SVA	Sistema de Visão Artificial
UFMA	Universidade Federal do Maranhão
USB	Universal Serial Bus
Xbot	Empresa especializa em soluções educacionais para robótica

Sumário

1	INTRODUÇÃO	1
1.1	Objetivos	2
1.2	Motivação	2
1.3	Revisão Bibliográfica	3
1.4	Organização da Dissertação	5
2	ROBÓTICA MÓVEL	7
2.1	Tipos de Rodas	9
2.1.1	Rodas Convencionais	9
2.1.2	Rodas Omnidirecionais	11
2.1.3	Disposição de Direção	11
2.2	Sensores	14
2.2.1	Tipos de Sensores	14
2.3	Atuadores	16
2.4	Sistemas de Medição	18
2.4.1	Medições Relativas	20
a)	Medição Baseada em Velocidade	20
b)	Medição Baseada em Odometria	20
c)	Medição Baseada em Navegação Inercial	21
2.4.2	Medições Absolutas	21
2.4.3	Medição Baseada em Visão	22
2.5	Cinemática de Robôs Móveis	23
2.5.1	Movimento em Uma Dimensão	23
a)	Deslocamento	24
b)	Velocidade Média	24
c)	Aceleração Média	25

2.5.2	Movimento em Duas Dimensões	25
a)	Deslocamento.....	25
b)	Velocidade Média	26
c)	Aceleração Média.....	26
2.5.3	Estabilidade	26
2.5.4	Modelagem Cinemática	27
3	PLATAFORMA ROBÓTICA.....	31
3.1	Robô Móvel Scisoccer.....	31
3.2	Rádio Base.....	32
3.3	Transceiver	33
3.4	Módulo Encoder	33
3.5	Alimentação	34
3.6	Locomoção.....	35
3.7	Placa Mãe e Placa Driver	35
3.8	<i>Software</i> para Futebol robôs Scisoccer.....	37
3.9	Protocolo de comunicação	38
4	VISÃO COMPUTACIONAL	40
4.1	Imagem Digital	40
4.2	Processamento de Imagens Digitais.....	40
4.3	Estrutura de um Sistema de Visão Artificial	43
4.3.1	Aquisição da imagem.....	43
4.3.2	Pré-processamento.....	44
4.3.3	Segmentação.....	44
4.3.4	Seleção de Características	45
4.3.5	Reconhecimento e Interpretação	45
4.4	Biblioteca OpenCV	45
4.4.1	Estrutura e conteúdo da biblioteca OpenCV	46

5	EXPERIMENTOS E RESULTADOS	48
5.1	Requisitos de Hardware do Sistema	48
5.2	Funcionamento e Execução do Sistema	48
5.2.1	API de acionamento do robô Scisoccer	48
5.2.2	Operação <i>off-line</i>	51
a)	Aquisição e calibração do sistema de visão	51
b)	Computação dos parâmetros: (Δq) e (Δt)	54
5.2.3	Processamento de imagem com sistema de medição	55
5.3	Algoritmo de Medição Proposto	56
5.3.1	Aquisição de imagens	56
5.3.2	Correções e Filtros	56
5.3.3	Segmentação	57
5.3.4	Extração e identificação de atributos	58
5.3.5	Cálculo: deslocamento, velocidades e aceleração	59
5.4	Experimentos	61
5.5	Resultados	61
6	CONCLUSÃO	65
6.1	Trabalhos Futuros	66

1 INTRODUÇÃO

O campo da robótica é um tema de pesquisa fascinante. Os robôs vêm evoluindo rapidamente de robustos mecanismos industriais para máquinas cada vez mais complexas, sofisticadas e capazes de realizar tarefas desafiadoras do nosso dia a dia.

Muitas definições têm sido sugeridas para o que possa ser chamado de robô. Segundo (MURPHY, 2000), robôs são um conjunto de tecnologias usadas para auxiliar ou substituir o homem em tarefas repetitivas que demandem esforços físicos ou que ofereçam risco a vida com capacidade de interação com o mundo de forma dinâmica a partir de dispositivos eletromecânicos com certo grau de autonomia e inteligência.

Existem algumas características essenciais que um robô deve ter e isso ajuda na decisão de quais recursos o projetista precisará para construir estes tipos de máquinas. Sensoriar informações tanto do ambiente como de si mesmo, ter certa autonomia com base em suas próprias decisões e agir no mundo físico são características conhecidas na robótica como “Sentir”, “Pensar” e “Agir”.

A área da robótica exige aplicação simultânea de várias disciplinas de pesquisa e demanda do pesquisador um pensamento lateral e multidisciplinar. Áreas como engenharia mecânica, engenharia elétrica, engenharia de computação, inteligência artificial, biologia, entre outras interagem em perfeita harmonia na projeção dessas formidáveis máquinas autônomas. Assim, conforme os avanços tecnológicos na área os robôs se tornam realidade no dia a dia e cada vez possuem habilidades (mesmo que limitadas) de ouvir, tomar decisões, falar, ver e interagir com objetos e seres humanos.

Com relação às capacidades mencionadas a habilidade relativa à visão robótica ou visão computacional tem sido alvo de diversos trabalhos de pesquisas na área de computação nos últimos tempos. Diversas ferramentas têm sido desenvolvidas pela comunidade que simplifica a criação de aplicações a partir da disponibilidade de algoritmos e bibliotecas específicas para visão computacional. Uma destas ferramentas é a biblioteca criada pela Intel OpenCV, que conta com funções de processamento de imagens, detecção de movimento, rastreamento, reconhecimento de padrões em imagens, aprendizagem de máquina, etc.

Outra linha tem sido o estudo da capacidade de movimentação e interação do robô em um ambiente estruturado de maneira que a forma e as condições de como este navegará no ambiente também deve ser considerada. Nesse aspecto esse dispositivo necessita de *softwares* que viabilizem e possibilitem alcançar algum tipo de objetivo. Neste trabalho o robô objeto de estudos é o Scisoccer.

Assim este trabalho visa desenvolver um sistema para medição do valor da posição, da velocidade e da aceleração de um robô móvel a partir do processamento de imagens digitais afim de contribuir com a melhoria das pesquisas relacionadas à navegação e da visão computacional em robótica móvel.

1.1 Objetivos

Este trabalho tem como objetivo desenvolver um aplicativo para medição do valor da posição, da velocidade e da aceleração de um robô móvel através de visão computacional utilizando a biblioteca OpenCV em ambiente fechado.

Os objetivos específicos são:

- Desenvolver um sistema de visão computacional usando a biblioteca OpenCV para localização do robô;
- Desenvolver um aplicativo de medição por visão computacional para determinar a velocidade e a aceleração de um robô móvel.

1.2 Motivação

O Scisoccer, disponibilizado para pesquisas e elaboração deste trabalho no Laboratório de Inovações Tecnológicas (InovTec) da UFMA, é uma plataforma robótica projetada para realizar partidas de futebol de robôs da classe SmallSize, conhecida como F180. Apesar de a plataforma possuir uma variedade de recursos, a falta de sensores no robô restringe sua utilização. Dessa forma, torna-se necessária a criação de um sistema onde a plataforma possa realizar outras tarefas, mesmo com sua limitação sensorial. Assim, a falta de um sistema de informação da localização do robô levou a realização do presente trabalho.

1.3 Revisão Bibliográfica

Um robô é um sistema autônomo que existe no mundo real, capaz de "sentir" ou realizar sensoriamento e "agir" sob um ambiente afim de alcançar um determinado objetivo. Já a robótica é o estudo dos robôs, ou seja, o estudo da sua capacidade de sentir e agir no mundo físico de forma autônoma e intencional (MATARIC, 2007).

Projetar um mecanismo robótico com capacidade de encontrar sua posição no ambiente, reconhecer e responder à situações que envolvam sua locomoção, são dificuldades fundamentais no estudo da robótica móvel autônoma (DUDEK; JENKIM, 2000).

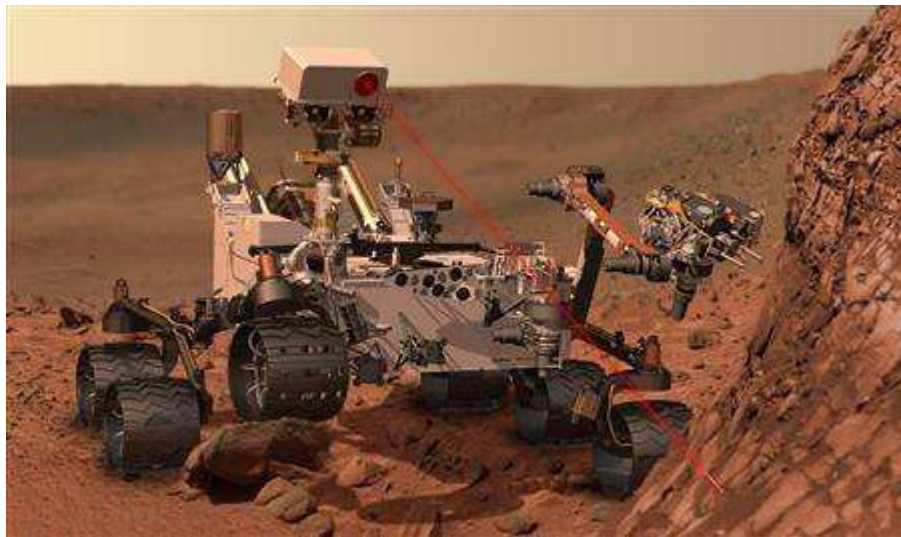


Figura 1. Robô Curiosity (NASA)

A aquisição de características a partir de imagens digitais vem sendo amplamente utilizada na robótica móvel como uma estratégia de navegação. Essa técnica é conhecida como odometria visual (OV). O termo foi escolhido pela sua semelhança com a odometria de roda, a qual calcula de forma incremental o movimento de um veículo integrando o número de voltas de suas rodas ao longo do tempo. A odometria visual opera de forma similar, incrementando as mudanças entre a movimentação das imagens digitais captadas pela câmera para estimar a posição de um veículo (SCARAMUZZA *et al.*, 2011). Já existem vários exemplos de aplicações robóticas onde o uso desta técnica se faz presente, dentre elas podem-se destacar: robôs de exploração espacial, exploração submarina, vigilância, navegação de veículos não tripulados, manufatura, entre outros.

Um estudo mais detalhado desta técnica é relatado no trabalho de (NADARAJAH, 2013) onde é feita uma análise sobre local de posicionamento da câmera. O referido trabalho discute se a câmera deve ser posicionada no teto ou de lado, e sobre a aplicação de visão global ou de visão local, nesse último caso com as câmeras montadas em cima do robô, podendo ser divididas também em três tipos de posicionamento: omnidirecional, visão estéreo e visão monocular. Também é discutido alguns modelos de algoritmos de processamento de imagem que mais se adequam a estes tipos de aplicações.

No trabalho de (SE *et al.*, 2005) foi desenvolvido um sistema para a localização de robô móvel utilizando visão global e o algoritmo da Hough no processamento de imagem. Nesse trabalho as características do *frame* atual são comparadas com o *frame* anterior e a partir destes dados são calculadas as posições do robô. Em (ANDRADE *et al.*, 2006) foi descrito um sistema onde as características do ambiente são obtidas através do processamento de imagens extraídas a partir de duas câmeras mantadas no teto. Os dados são suficientes para a realização da locomoção e controle de um robô móvel em um ambiente dinâmico. Outra abordagem similar está no trabalho de (DE BEM; COSTA, 2006) onde é apresentada uma metodologia para o rastreamento visual de múltiplos alvos em sequências de imagens. Os alvos são descritos e reconhecidos por seus contornos que mantêm uma boa estimativa da posição de seus centroides. Já o sistema desenvolvido por (ORQUEDA, 2007) mostra-se uma arquitetura muito interessante, os robôs recebem marcações, as imagens são obtidas a partir de câmeras móveis e um filtro de Kalman é desenvolvido para a estimação da pose, do deslocamento e da velocidade.

(PENA, 2013) desenvolveu um aplicativo que utiliza técnicas de visão computacional com análise do fluxo de pixels em vídeos para localização e rastreamento de padrões de imagens. O CVMob foi desenvolvido no Instituto de Física da Universidade Federal da Bahia em linguagem C++, utilizando o *framework* QT4 e a biblioteca de visão computacional OpenCV para geração de dados sobre o rastreamento de um pendulo simples. (PINHEIRO, 2008) realizou um estudo das técnicas de medição por visão computacional no movimento humano. Ele desenvolveu um sistema capaz de capturar e construir a trajetória do movimento no espaço. O sistema desenvolvido utiliza câmeras de vídeo para registrar o movimento e um programa processa as imagens extraídas dos vídeos de forma a estimar a

trajetória do movimento e calcula as suas variáveis cinemáticas. Ao final é feita uma avaliação experimental da precisão e exatidão do sistema proposto e alguns estudos de casos que tiveram como objetivo demonstrar o potencial do seu sistema.

(JOSÉ *et al.*,2008) desenvolveram um sistema de estimação de posição e orientação baseado em técnicas de visão computacional usando correspondência de imagens. O sistema se baseia na comparação de imagens adquiridas em diferentes instantes para determinar os deslocamentos angular e linear entre elas e, assim, estimar a trajetória do robô. A comparação das imagens é realizada usando uma técnica de casamento de pontos baseada nas distâncias entre os pontos de interesse. Neste sentido, as distâncias entre os pontos de interesse são agrupadas em conjuntos de distâncias e as distribuições de distâncias destes conjuntos são usadas para realizar a correspondência das imagens. Sistema desenvolvido foi avaliado a partir de imagens obtidas por uma câmera de baixa resolução fixada em uma plataforma móvel.

(KLASER,2014) apresentou um sistema de navegação autônoma para veículos terrestres em ambientes não estruturados usando visão computacional como sistema de percepção que utiliza uma câmera estéreo em um veículo Ackerman. Já o trabalho de (DO NASCIMENTO, 2014) e (SCHUEROFF *et al.*, 2014) mostraram uma proposta muito propícia para a exploração de ambientes usando visão computacional em locais *indoor* e *outdoor* em tempo real.

1.4 Organização da Dissertação

Este trabalho está dividido em seis capítulos, além desse de introdução, da seguinte forma:

Capítulo 2: Apresenta as principais plataformas e topologias de rodas, sensores, atuadores e sistemas de medição utilizados na robótica móvel, assim como as classificações e parâmetros referentes à cinemática de posições, velocidade e aceleração.

Capítulo 3: apresenta as características gerais de *hardware* e *software* da plataforma robótica Scisoccer.

Capítulo 4: São abordados os conceitos sobre visão computacional e processamento digital de imagens, assim como as características da biblioteca OpenCV.

Capítulo 5: Tem-se os detalhes sobre as técnicas e metodologias desenvolvidas no presente trabalho, assim como as etapas do algoritmo de processamento de imagem referentes à localização, ao deslocamento, a velocidade e a aceleração do robô. Nesta seção, também são abordados os experimentos e os resultados obtidos.

Capítulo 6: Apresenta as conclusões obtidas e as possíveis propostas para trabalhos futuros.

2 ROBÓTICA MÓVEL

A robótica é o estudo dos robôs e suas características mecânicas, eletromecânicas, eletrônicas e computacionais. Segundo (SECCHI, 2008) um robô móvel é uma estrutura eletromecânica montado em uma base não fixa, com capacidade de “sentir” e “agir” seu ambiente de trabalho. É dotado por um sistema de locomoção que lhe garante a capacidade e navegação com certo grau de autonomia e objetivo. Suas aplicações são muito variadas, vão desde tarefas simples, educacionais e domésticas a aplicações industriais, militares, nocivas e arriscadas a vida humana.

Um robô móvel é considerado autônomo quando ele tem a capacidade de se locomover e tomar decisões sem a intervenção humana. Para que isto seja concebido alguns aspectos devem ser considerados e dentre estes pontos destacam-se:

- Capacidade de percepção - através do uso de sensores capazes de obter informações sobre o ambiente onde o robô se encontra no momento;
- Robustez e inteligência - capacidade de lidar com as mais diversas situações de modo a resolver e executar tarefas por mais complexas que sejam (WOLF et. al., 2009);
- Capacidade de agir- através do uso de atuadores e motores aptos a realizar ações como por exemplo o deslocamento do robô no ambiente.

Os robôs móveis são classificados de acordo com seu sistema de locomoção (DUDEK & JENKIN, 2010), conforme podem ser observados na figura 2.

- Robôs terrestres: Geralmente possuem rodas, esteiras, pernas ou patas bio-inspiradas, propulsão modular de corpo (*snake robot*);
- Robôs aéreos: Classe de robôs onde são referenciados geralmente como veículos aéreo não tripulados, quadricópteros ou drones;

- Robôs aquáticos: São adequados para ambientes marinhos, rios e piscinas.
- Robôs espaciais: São desenvolvidos para trabalharem em micro gravidades, específicos para exploração espacial.



Figura 2. Robôs terrestres, aéreos, aquáticos e espaciais

Outro critério utilizado na classificação dos robôs móveis está na maneira como ele interage com outros robôs. Robôs móveis podem ser projetados para

atuarem de forma individual ou em grupos, por exemplo, nas competições de futebol de robôs (BRAND *et al.*, 2002).

2.1 Tipos de Rodas

Os robôs terrestres com rodas podem ser constituídos a partir de diversas topologias, podendo ou não ser holonômicos (ARKIN,1998). A característica holonômica ocorre quando o robô é capaz de atingir todas as posições imediatamente adjacentes a sua posição atual, em apenas um movimento (MATARIC,2007). A operacionalização dos robôs móveis com rodas é consequência de dois pontos essenciais: o tipo das rodas que possui e a sua localização na estrutura física.

Os tipos de rodas utilizadas na robótica móvel são basicamente de dois tipos: convencional e omnidirecional (também conhecida por roda sueca).

2.1.1 Rodas Convencionais

Para estes tipos de rodas, não existe deslizamento entre o ponto de contato da roda e o chão, satisfazendo a condição de rolamento puro.

As rodas convencionais classificam-se em três tipos:

- **Rodas fixas:** O centro da roda é um ponto fixo no chassi do robô, conforme ilustrado na figura 3. Tem rotação apenas em torno do eixo. Em geral, este tipo está associado ao sistema de tração do robô.

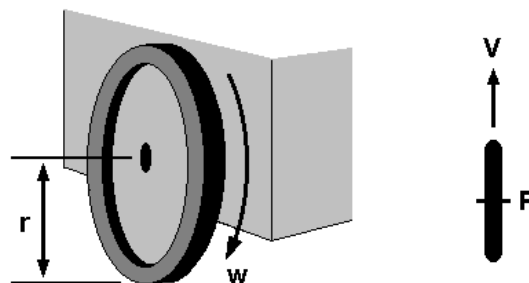


Figura 3. Roda Fixa - 1 grau de liberdade

Fonte: (MATIAS, 2015)

- **Rodas orientadas no centro:** Tem dois tipos de movimentos, em relação ao um eixo vertical de rotação e outro passando através do seu centro. A construção é a mesma das rodas fixas, porém o ângulo de giro é alterado ao longo do tempo. Em geral é usada como roda de direção, conforme figura 4.

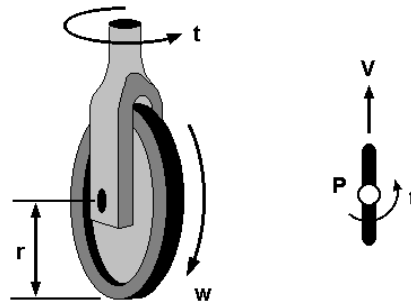


Figura 4. Roda com orientação no centro e 2 graus de liberdade

Fonte: (MATIAS, 2015)

- **Rodas orientadas fora do centro (Castor Wheel):** Possuem orientação em relação ao chassi do robô, rotação se dá em torno do eixo vertical que não passa pelo seu centro e em torno do ponto de contato, conforme a figura 5. A sua principal função é estabilizar a estrutura mecânica do robô.

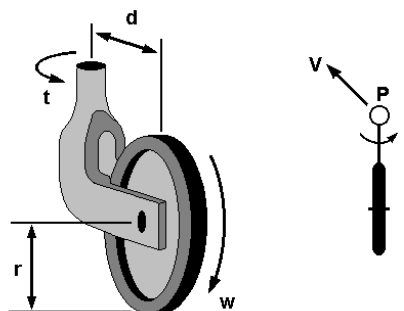


Figura 5. Roda castor wheel (rodinha de apoio)

Fonte: (MATIAS, 2015)

2.1.2 Rodas Omnidirecionais

Nas rodas omnidirecionais supõe-se que apenas uma das componentes da velocidade do ponto de contato da roda com o terreno é igual a zero, diferentemente das rodas convencionais onde todas as componentes da velocidade do ponto de contato são consideradas iguais a zero ao longo do movimento (SECCHI,2008).

Ela é responsável pela tração na direção perpendicular ao eixo do motor e permite deslizar na direção do seu eixo, conforme ilustrado na figura 6. Para assegurar as características relacionadas à omnidirecionalidade é preciso que as rodas possuam o mínimo de atrito na direção do eixo do motor, facilitando a sua movimentação.

Durante o deslocamento do robô podem ser associados movimentos de translação e rotação assegurando ao robô a chegada em seu destino com o ângulo desejado.

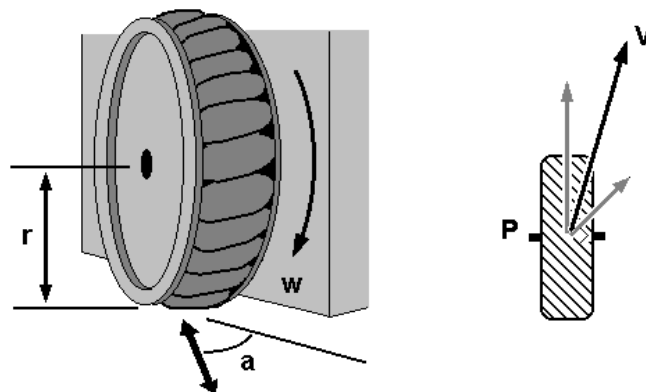


Figura 6. Roda sueca: três graus de liberdade.

Fonte: (MATIAS, 2015)

2.1.3 Disposição de Direção

Como já mencionado, a dirigibilidade de um robô móvel é o resultado da junção dos tipos de rodas que utiliza e de sua disposição na estrutura. Com relação as opções disponíveis, as direções são classificadas em quatro tipos: Omnidirecionais, Uniciclo, Triciclos e Quadricículos.

- **Robôs Omnidirecionais:**

Os robôs omnidirecionais têm total grau de liberdade no plano, ou seja, movimentem-se em qualquer direção sem necessidade de se reorientarem. A figura 7 apresenta o robô youbot desenvolvido pela empresa Kuka. De acordo com a rotação de cada uma das suas rodas, o robô pode andar em linha reta, girar ou deslocar-se lateralmente, neste último caso sem necessidade de mudar a sua orientação.

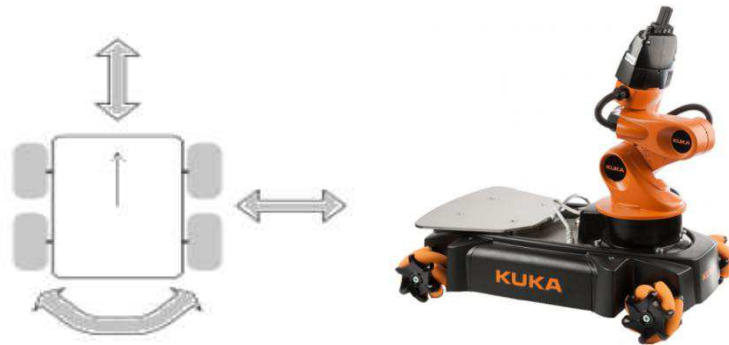


Figura 7. Robô omnidirecional

Fonte: <http://www.youbot-store.com>

- **Robôs Uniciclo:**

Os robôs uniciclos são plataformas de simples construção, possuindo certa facilidade na modelagem e entendimento de sua cinemática e controle. É a opção mais comum e normalmente mais usada pelos pesquisadores para bases de testes de novas estratégias de controle e navegação (ROMERO,2014).

Conforme demonstra a figura 8, sua estrutura é composta por duas rodas fixas convencionais dispostas sobre o mesmo eixo e controladas de forma independente por uma ou mais rodas livres que lhe garantem estabilidade.

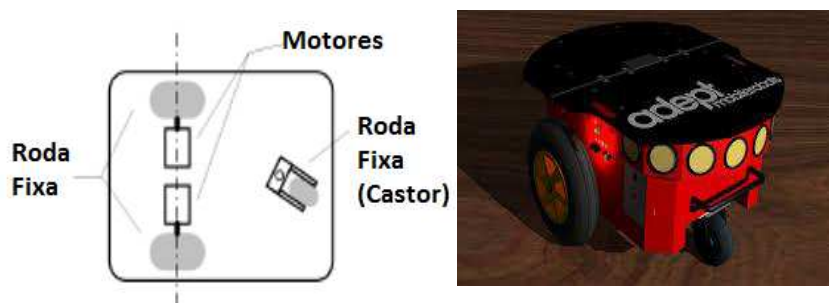


Figura 8. Robô Uniciclo Pioneer.

Fonte: www.cyberbotics.com

- **Robôs Triciclos:**

O sistema de movimentação deste tipo de base robótica, conforme a figura 9, consiste em duas rodas convencionais fixas sobre um mesmo eixo e uma roda convencional, centrada e orientável que reúne as funções de tração e direção (COUTINHO,2014). Assim como no robô Uniciclo sua cinemática é bastante simples tornando-o adequado para diversas pesquisas na área. Contudo, apresenta limitações quanto à sua utilização, pois seu centro de gravidade posiciona-se em alguns casos, nos limites da superfície de equilíbrio definida pelas três rodas, gerando perda momentânea de tração e erros na determinação da posição do robô, limitando-o para aplicações que exijam grande precisão.



Figura 9. Robô Triciclo Olin

Fonte: www.deploy.virtual-labs.ac.in

- **Robôs Quadricículos(ou Ackerman):**

Os robôs quadricículos também conhecido por Ackerman, surgiram a partir da necessidade de solução do problema do erro na determinação do posicionamento dos robôs triciclos (COUTINHO,2014).

Conforme figura 10, neste robô, os eixos das duas rodas dianteiras interceptam-se em um ponto (C na imagem) pertencente ao eixo comum das rodas traseiras. Esta estrutura contribui para uma maior estabilidade e evita o deslizamento das rodas, reduzindo os erros de posicionamento (odometria).

Apesar da maior complexidade esta configuração é bastante recomendada e utilizada em robôs que transitem em terrenos irregulares.

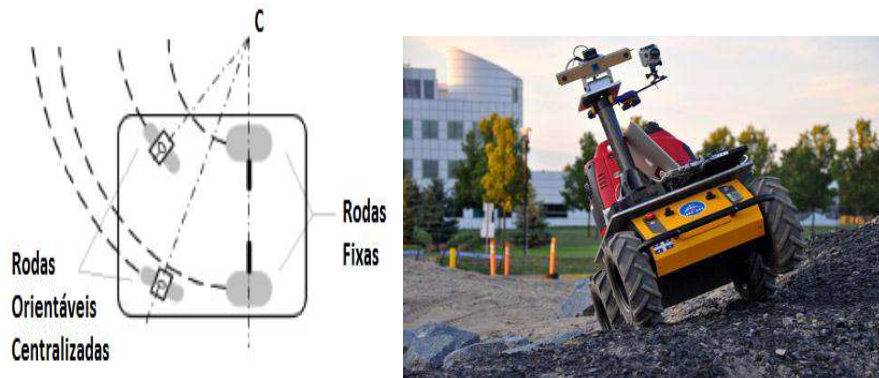


Figura 10. Robô Quadriciclo.

Fonte: <http://spectrum.ieee.org/automaton/robotics/diy>

2.2 Sensores

Os robôs móveis caracterizam-se pela capacidade de se movimentarem de forma autônoma em um ambiente parcial ou totalmente desconhecido.

Para isso o robô precisa ser dotado de um sistema sensorial que lhe permita coletar dados do meio por onde circula para depois extrair os aspectos mais relevantes do ambiente e reagir conforme os eventos e imprevistos durante a sua navegação.

A qualidade da reação do robô é proporcional à quantidade e precisão das informações que são fornecidas pelos sensores, assim, surge a necessidade de equipar um robô com sensores cada vez mais variados e complexos. Idealmente, um robô deveria ter capacidades sensoriais semelhantes ao ser humano, recebendo várias informações do mundo exterior, baseando as suas reações em apenas um tipo ou a partir da fusão de vários sensores. Em vários trabalhos são relatados características e propriedades desses elementos como em (BEKEY,2005), (BRAUNL,2006) e (WOLF et. al., 2009).

2.2.1 Tipos de Sensores

Há uma grande diversidade de sensores utilizados em robótica móvel. Alguns são empregados simplesmente para obter valores como a temperatura ou a velocidade de rotação dos motores, enquanto outros, mais complexos, podem ser

usados para obtenção de informações de distancias aos obstáculos do ambiente ou para determinar diretamente a sua posição global.

Comumente, os sensores são classificados segundo as suas características em dois grupos funcionais (ROMERO, 2014).

- Proprioceptivos (internos) ou Exteroceptivos (externos)
- Passivos ou Ativos

O grupo Proprioceptivos/Exteroceptivos refere-se à origem das informações sensoriais obtidas. Proprioceptivos são sensores que medem valores internos do robô como a velocidade do motor, tensão da bateria, etc. Já os sensores Exteroceptivos coletam os dados do meio envolvente ao robô tais como: distâncias, intensidade de luz ou amplitude de som, etc.

Com relação aos Passivos/Ativos, estes correspondem à forma como é feita a aquisição dos dados existentes no ambiente. Sensores passivos captam a energia existente no próprio ambiente sem necessidade de circuitos eletroeletrônicos complementares. Sondas de temperatura, microfones e sensores CCD das câmeras, são exemplos desses dispositivos.

Os sensores ativos dispõem de circuitos complementares e emitem energia para o ambiente, sendo a resposta desta energia a informação a ser captada. Têm uma performance superior, mas, são bastante suscetíveis a interferências do meio. Exemplos destes sensores são encoders, sensores ultrassônicos ou lasers.

Na tabela 1, tem-se alguns exemplos de sensores utilizados na robótica bem como a sua classificação segundo as características apresentadas. Além das características comuns há outras considerações a serem verificadas em aplicações em robótica móvel tais como: resolução, precisão, alcance, imunidade a variações de condições ambientais, robustez a vibrações, tamanho, consumo de energia, desgaste e segurança de funcionamento.

Um exemplo de aplicação dos sensores ativos está na técnica de medição de distâncias por meio da triangulação de sensor infravermelhos. Conforme a figura 11, tem-se a medição da distância através do envio e recebimento por reflexão de um sinal infravermelho emitido em direção ao alvo. A variação de luminosidade que entra no receptor é então convertida em uma variação de tensão.

Tabela1: Classificação de sensores em robótica móvel

Classificação Geral	Tipo de Sensor	Origem	Aquisição da Informação
Sensores de Táteis	Barreiras Óticas	Exteroceptivo	Ativo
Sensores de Motores/Rodas	Encoders Óticos	Proprioceptivo	Ativo
Sensores de Orientação	Bússola Digital	Exteroceptivo	Passivo
Sensores de Referência	GPS	Exteroceptivo	Ativo
Sensores de Medição de Distâncias	Sensores de Ultrassons	Exteroceptivo	Ativo
Sensores de Movimento/Velocidade	Radar Doppler	Exteroceptivo	Ativo
Sensor de Visão Artificial	CCD/ Câmeras	Exteroceptivo	Passivo

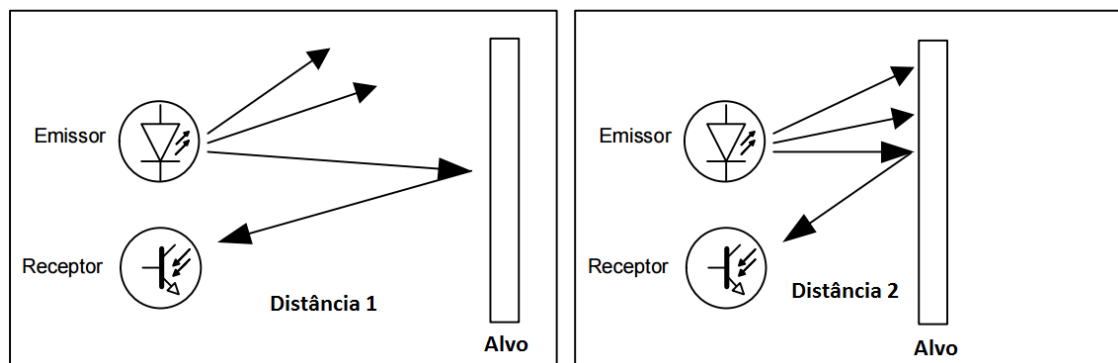


Figura 11. Sensor ótico por retro reflexão

2.3 Atuadores

Os atuadores são dispositivos que executam a conversão da energia elétrica, hidráulica ou pneumática em energia mecânica. A potência mecânica gerada pelos atuadores é enviada as partes móveis através dos sistemas de transmissão para que os robôs se movimentem.

Segundo (FELIZARDO; BRACARENSE, 2013) os atuadores classificam-se de acordo com o tipo de energia que utiliza, conforme apresentados na figura 12:

- Atuadores Hidráulicos: utilizam um fluido sob pressão para movimentar um braço manipulador. Geralmente operam grandes cargas onde é exigido grande potência e velocidade mas oferecem baixa precisão;
- Atuadores Pneumáticos: utilizam um gás sob pressão para movimentar um braço manipulador. São mais baratos que os hidráulicos e oferecem baixa precisão ficando limitados a operações do tipo pega e coloca (do inglês, *pick and place*);
- Atuadores Eletromagnéticos: São motores elétricos (de passo, servos, corrente contínua ou corrente alternada) ou músculos artificiais usados em robôs de pequeno e médio porte e muito utilizado na robótica móvel.

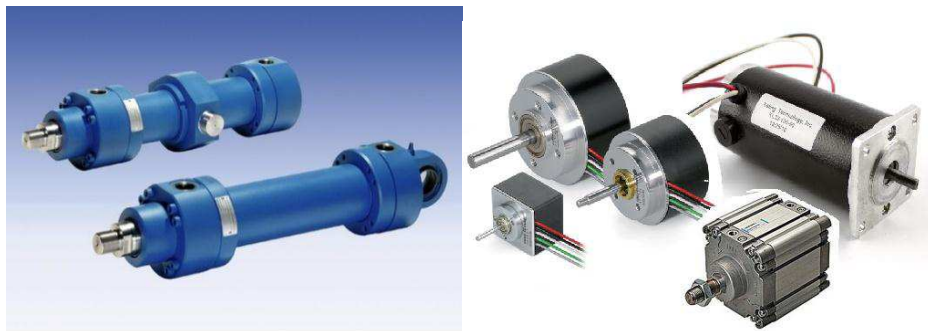


Figura 12. Atuadores hidráulicos, pneumáticos e eletromagnéticos

Fonte: www.logismarket.ind.br

Conforme demonstrado o desenvolvimento de um robô móvel envolve uma gama de variáveis a serem consideradas, tais como: problemas com carga de bateria, ruídos nas leituras de sensores, movimentação de atuadores, problemas de programação, erros de posição e orientação, etc. Assim, para que um projeto de um robô móvel tenha êxito em sua execução é necessário levar em consideração os seguintes pontos (HIRANO, 2006):

- Reconhecimento: o robô deve ter capacidade de reconhecer de alguma forma o ambiente onde está situado;
- Navegação: capacidade que permite o robô de criar uma forma de representação do ambiente. Esta modelagem de mundo deve ser

criada de forma refinada até que erros de sensoriamento sejam mínimos e desprezíveis para a navegação do robô.

- **Aprendizado:** deve ser considerado algoritmos que permitam ao robô calibrar seus sensores em função das modificações do ambiente e respectivas ações externas.
- **Cognição:** criar técnicas para armazenamento de conhecimento e utilizar este conhecimento para a tomada de decisão, tornando o robô um elemento com habilidades cognitivas de alto nível que o permitam total autonomia com relação ao meio.

2.4 Sistemas de Medição

A medição é essencial para a vida humana e consequentemente para os sistemas robóticos. Nos seres vivos a medição é realizada pelos sentidos para avaliar perigos, reprodução, alimentação, etc. O sistema de medição também está associado com a aprendizagem, sendo que este não é possível sem o primeiro. Na robótica medições são utilizadas para obtenção de informações sobre grandezas físicas relativas ao robô e sobre o ambiente. Sendo muito útil para a monitoração, locomoção e determinação da arquitetura de controle: a) reativa (baseada em comportamento), b) deliberativa (centralizada) e c) híbrida (reativa-deliberativa) (NAKHAEINIA, 2011). Estas arquiteturas podem ser retratadas por três princípios básicos da robótica: Sentir, Planejar e Agir (MATARIC,2007).

Assim, a finalidade de um sistema de medição é, então, a de prover informações sobre uma determinada grandeza com máxima qualidade de acordo com os requisitos da aplicação. O processo de medição consiste em mapear um objeto desconhecido do mundo real em uma imagem abstrata, usando um conjunto de símbolos afim dos resultados serem utilizados para tomada de decisões, controle, entre outras.

Os sistemas de medição podem ser realizados de cinco formas diferentes, apresentadas conforme a sua ordem de complexidade (Instrumentação...,2015):

- **Nominal:** Nesta forma de medição o resultado é fornecido de forma qualitativa. As medidas são classificadas apenas se pertencem ou não a uma determinada classe, ou seja, se são verdadeiros ou falsos.

- Ordinal: É o sistema de medição quantitativo onde duas medidas são comparadas entre si e o resultado é da forma maior que, igual ou menor que.
- Intervalar: Nesse tipo de medição é estimado se uma determinada medida pertence a um intervalo definido ou se podem comparar quantitativamente dois intervalos de medição. São exemplos de medição intervalar: medições entre intervalos de tempo e distâncias (sem o valor zero ou origem)
- Racional: Na medição racional o resultado indica a razão entre dois objetos de medição, ou entre um objeto de medição e uma referência interna. Esse resultado deve ser sempre positivo. Nesse tipo de medição a origem é fixa. Medições percentuais são um bom exemplo desta categoria, fixando a referência interna como o valor máximo possível dentro da faixa de medição.
- Cardinal: Este é o tipo mais complexo de medição. Os resultados são fornecidos numa escala padrão de medição, como em metros (m) ou segundos (s) no SI. Os resultados da medição são dados na forma: $a = mU$, onde m é um número real e U é a unidade no sistema padrão utilizado. Na medição cardinal o sistema de medição tem que ser aferido de acordo com uma escala padrão.

A percepção do ambiente pelo robô é conseguida pela utilização de sensores que disponibilizam ao sistema de processamento medidas absolutas e relativas. As medidas absolutas são medições informadas ao robô referente ao ambiente em que se encontra, a qualquer momento e sem conhecimento prévio de medidas anteriores. Geralmente são realizadas por meio de sensores exteroceptivos. Já as medidas relativas, precisam do conhecimento das medidas anteriores para a geração da sua posição atual. São obtidas através de sensores proprioceptivos, como é o caso do odômetro (MATARIC,2007).

2.4.1 Medições Relativas

Os principais métodos e medidas utilizadas para localização relativa são: medição baseada em velocidade, medição baseada em odometria e medição baseada em navegação inercial (JOSÉ *et al.*, 2008).

a) Medição Baseada em Velocidade

É usada quando a velocidade das rodas não é considerada para efeito de cálculo da posição. Tem como base a medição das velocidades translacional e rotacional do robô em sua posição anterior e atual, em um determinado instante de tempo. A partir da medição da velocidade e tempo de percurso, estima-se o deslocamento nesse período, que somado a posição inicial, estima-se a posição atual.

b) Medição Baseada em Odometria

Para a obtenção da localização o método utiliza a integração ao longo do tempo da velocidade das rodas do robô móvel obtida por encoder, figura 13. Estes dispositivos eletromecânicos são capazes de contar ou reproduzir pulsos elétricos a partir do movimento rotacional de seu eixo. Podem gerar pulsos para um determinado incremento de rotação do eixo (encoder rotativo) ou um pulso para uma determinada distância linear percorrida (encoder linear). Embora seja mais utilizado no controle de posição também é utilizado para medir velocidade, uma vez que medindo a distância total percorrida (através da contagem dos pulsos em sua saída) e o tempo necessário para esta distância ser percorrida, consegue-se estimar a velocidade e aceleração.



Figura 13. Encoder Óptico

Fonte: www.directindustry.com

c) Medição Baseada em Navegação Inercial

Os sensores inerciais, ilustrado na figura 14, são bastantes utilizados na localização de robôs móveis. Nos métodos inerciais os dados sobre velocidade angular e aceleração linear dos robôs em relação a um referencial são obtidos a partir de sensores como giroscópios, acelerômetros e inclinômetros (BORENSTEIN,1997).

Essas informações são integradas afim de se obter o deslocamento angular e após dupla integração, obtêm-se o deslocamento linear em cada eixo. Neste método há um grande acúmulo dos erros, tornando-o um método impreciso (WALDMANN et al.,2000).

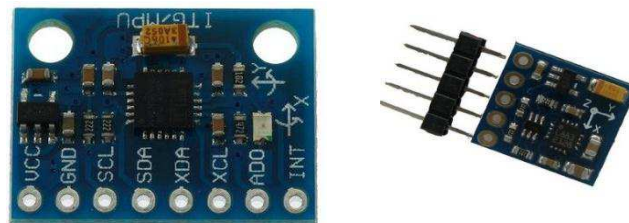


Figura 14. Sensores Inercial de Orientação (Acelerômetro e Inclinômetro)

Fonte: <http://loja.vidadesilicio.com.br>

2.4.2 Medições Absolutas

A medição absoluta da posição, velocidade e aceleração de robôs móveis consiste na obtenção dos dados do robô independente de leituras em instantes anteriores. Geralmente são obtidos a partir de sensores como bússolas eletrônicas ou receptores *GPS* (*Global Positioning System*), (BORENSTEIN,1997).

Através das bússolas eletrônicas é possível obter a orientação do robô em relação ao norte magnético da terra e, com o *GPS*, demonstrado na figura 15, uma rede de satélites artificiais em órbita da Terra fornece as coordenadas de qualquer ponto do planeta favorecendo a localização e cálculo de velocidade e aceleração do robô móvel.

2.4.3 Medição Baseada em Visão

A visão é sem dúvida um dos sentidos mais importantes do ser humano, sendo dotado de habilidades capazes de realizar inúmeras percepções com pouco esforço de consciência. Em contrapartida, lidar de forma inteligente com o grande volume de dados fornecidos por um sistema de visão computacional é um dos maiores desafios para a robótica.



Figura 15. Módulo (GPS)

Fonte: https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf

A visão computacional é uma ciência com enormes ramificações e de difícil definição. De modo geral, pode-se dizer que a mesma contempla um conjunto de técnicas, área de concentração, etc., responsáveis por fazer uma máquina "enxergar" através de imagens do ambiente em sua volta, recuperando das mesmas informações, dados significativos que possibilitem a máquina executar tarefas inteligentes, aproximando-se da inteligência do homem (MILANO; HONORATO, 2010).

Segundo (BORENSTEIN, 1997), (MILANO; HONORATO, 2010) a navegação baseada em visão, descrita na figura 16, pode ser dividida nas seguintes fases:

- Aquisição de dados: é a parte do sistema responsável pela aquisição e digitalização das imagens captadas de uma ou mais câmeras.

- Segmentação de objetos: a partir de uma imagem digital tem-se a identificação e o processamento de imagens para identificação de regiões de interesse(contornos) com base em níveis de cinza, cor, profundidade ou movimento.
- Reconhecimento de objetos: Nesta etapa o sistema tenta realizar a identificação dos pontos de interesse analisados comparando-os com referências previamente armazenadas.
- Cálculo da Posição: Após identificação dos pontos de interesse a localização é processada com relação a um sistema de referência conhecido.

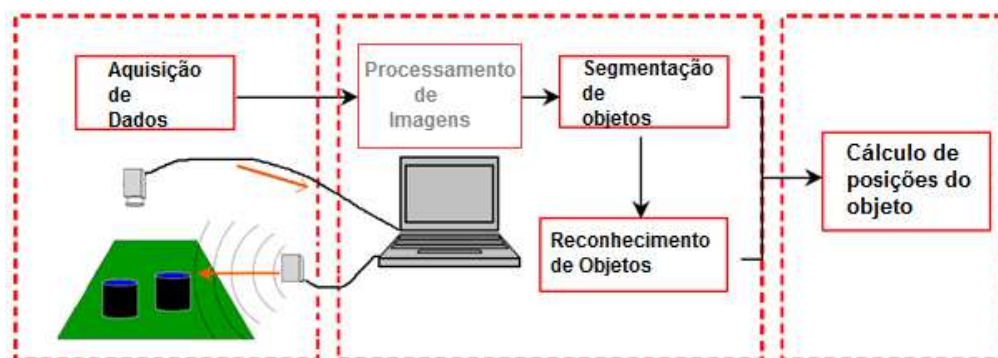


Figura 16. Estrutura de um sistema de navegação baseada em visão

2.5 Cinemática de Robôs Móveis

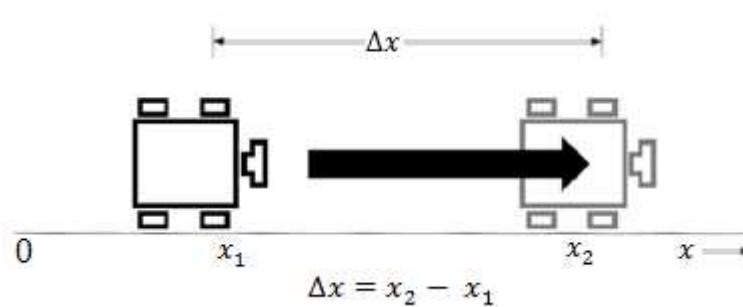
A cinemática é a ciência mais elementar do comportamento de um sistema mecânico. Ela descreve os movimentos de um corpo, procurando determinar sua posição, velocidade e aceleração em cada instante (NICOLAU *et al.*, 2007). Na robótica móvel é essencial e elementar entender estes comportamentos afim de embasar a elaboração do projeto e tornar o sistema adequado à movimentação do robô.

2.5.1 Movimento em Uma Dimensão

Qualquer corpo pode ser considerado como um ponto material, desde que suas dimensões sejam desprezíveis em relação ao movimento considerado.

a) Deslocamento

A figura 17 demonstra um robô na posição (x_1) no instante (t_1) e na posição (x_2) no instante (t_2). A modificação da posição da plataforma móvel, ou seja, o deslocamento (Δx) é dado pela diferença entre as posições ($x_2 - x_1$), logo:

$$\Delta x = x_2 - x_1 \quad (1)$$


O diagrama mostra um robô representado por um retângulo com rodas, deslocando-se ao longo de um eixo horizontal rotulado 'x'. O robô está inicialmente na posição x_1 e depois na posição x_2 . Uma seta grossa indica a direção do movimento. O deslocamento Δx é a distância entre x_1 e x_2 , indicada por uma linha tracejada com setas nas extremidades. A equação $\Delta x = x_2 - x_1$ é mostrada duas vezes: uma acima do robô e outra abaixo dele.

Figura 17. Deslocamento em uma dimensão

b) Velocidade Média

Define-se a velocidade média como a razão entre o deslocamento (Δx) e o intervalo de tempo ($\Delta t = t_2 - t_1$). Também é interpretada como o coeficiente angular (inclinação) da reta que passa pontos (t_1, x_1) e (t_2, x_2) , conforme a figura 18.

$$v = \frac{\Delta x}{\Delta t} = \text{inclinação} \quad (2)$$

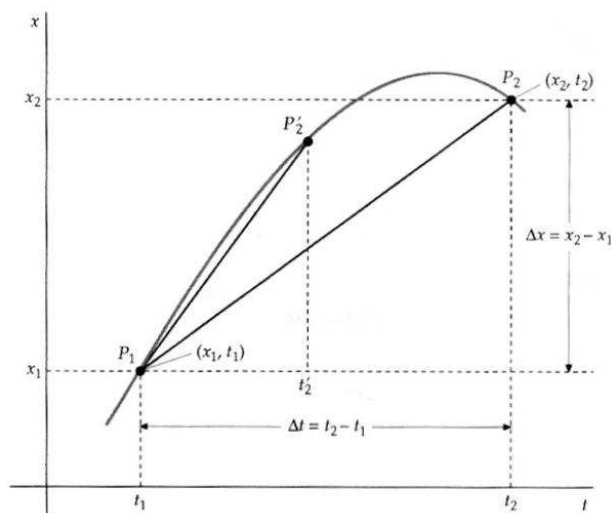


Figura 18. Velocidade média geométrica

c) Aceleração Média

A aceleração média é a taxa temporal de variação da velocidade, sendo expressa pela equação:

$$a = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{t_2 - t_1} \quad (3)$$

2.5.2 Movimento em Duas Dimensões

A localização de uma partícula (p) no plano 2D, ilustrado na figura 19, é dado pela sua coordenada x e y em relação a um sistema de referência, que tem seu vetor posição dado por:

$$r = x(i) + y(j) \quad (4)$$

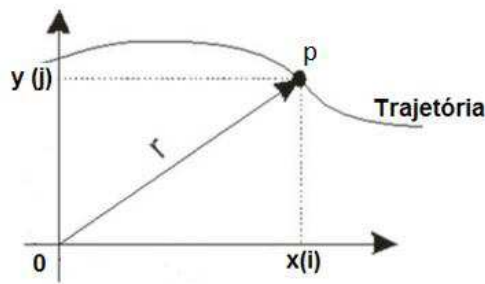


Figura 19. Localização de um ponto em plano de duas dimensões

a) Deslocamento

O movimento de uma partícula no plano xy é dada pela diferença entre os vetores de posição. Conforme a figura 20, o deslocamento da partícula do ponto A para o ponto B, é dado pelo intervalo de tempo entre os dois pontos considerados. O deslocamento é a diferença vetorial ($\Delta \vec{r} = \vec{r}_f - \vec{r}_i$) que é guiado da posição inicial da partícula para sua posição final.

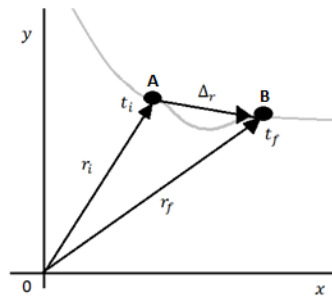


Figura 20. Movimento de uma partícula no plano xy

b) Velocidade Média

É a razão entre o vetor deslocamento e o intervalo de tempo. A velocidade média vetorial não depende da trajetória, apenas do vetor posição inicial e final. Sua direção será igual ao vetor trajetória.

Assim, o vetor velocidade média é calculado pela equação:

$$\vec{v} = \frac{\Delta \vec{r}}{\Delta t} \quad (5)$$

c) Aceleração Média

A aceleração escalar de uma partícula que se move de um ponto para outro, é definido como sendo a razão entre a variação do vetor velocidade instantânea e o tempo de variação.

$$\vec{a} = \frac{\vec{v}_f - \vec{v}_i}{t_f - t_i} = \frac{\Delta \vec{v}}{\Delta t} \quad (6)$$

2.5.3 Estabilidade

São necessárias pelo menos duas rodas para garantir a estabilidade de um robô. Contudo, robôs diferenciais (que contam com duas rodas padrão fixa), têm a garantia de estabilidade quando seu centro de massa estiver abaixo do eixo das rodas. Nos robôs com três rodas, demonstrado na figura 21, esta estabilidade ocorre devido ao centro de massa do robô estar localizado dentro do triângulo formado por

elas. Esta estabilidade favorece a navegação em terrenos irregulares ou regulariza problemas devido possíveis deformações nas partes do robô (MATARIC,2007).

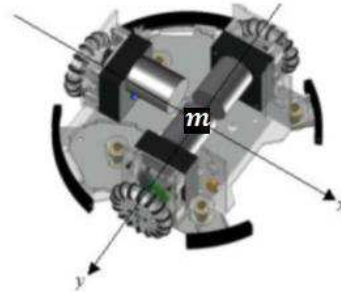


Figura 21. Centro de massa (m) em robôs com três rodas

2.5.4 Modelagem Cinemática

Para qualquer tipo de robô móvel é necessário um estudo detalhado dos diversos tipos de sistemas de coordenadas que o compõe. A marcação de coordenadas em torno de seu eixo central ou centro de massa, faz parte de um sistema local. Já um sistema global, é formado por um referencial inercial conhecido como sistema de referência do mundo. A partir destas representações, pode-se projetar as posições e orientações do robô através de um tratamento geométrico que simplifica e deduz a tarefa de reproduzir as características de todos os movimentos possíveis (BAEDE,2006).

Para tornar os parâmetros da velocidade e posições mais genérico, devem-se adotar para as análises os seguintes critérios: desconsidera-se as forças e as massas; mantem-se constante a potência dos motores; admite-se que a plataforma robótica seja constituída por um corpo rígido, cujas rodas não sofram deformações; que as limitações físicas dos atuadores (LINS BARRETO *et al.*, 2014), incertezas e situações de não linearidade, não afetam seu desempenho e que sua movimentação se dá apenas no plano horizontal. Conforme a figura 22, o desenvolvimento da plataforma se dará apenas no tratamento geométrico do sistema físico e embasado nas as obras de (Y. LIU *et al.*, 2008), (BENAVIDEZ; JAMSHIDI, 2011) e (CONCEIÇÃO,2012).

Os seguintes símbolos serão usados na modelagem:

$\omega_1, \omega_2, \omega_3 \left[\frac{rad}{s} \right]$ -> Velocidade angular das rodas

$v_1, v_2, v_3 \left[\frac{m}{s} \right]$ -> Velocidades linear das rodas

$roda_1$ -> Roda da frente

$roda_2$ -> Roda esquerda traseira

$roda_3$ -> Roda direita traseira

$d[m]$ -> Distância entre ponto central do robô e a roda

$x_r, y_r[m]$ -> Posição do robô no plano

$v, v_n \left[\frac{m}{s} \right]$ -> Velocidade linear do robô

$\omega \left[\frac{rad}{s} \right]$ -> Velocidade angular do robô (sobre seu próprio eixo)

$\theta[rad]$ -> Posição angular do robô

$v_x, v_y \left[\frac{m}{s} \right]$ -> Velocidade linear do robô segundo as referências (x_r, y_r)

A partir destas representações, pode-se projetar as posições e orientações do robô através de relações geométricas que simplifica e deduz a tarefa de reproduzir as características de todos os movimentos possíveis.

O objetivo da análise da cinemática dos robôs é obter uma descrição dos seus movimentos sem preocupar-se com as suas causas. Trata-se, portanto, da forma mais simples de analisar o comportamento de sistemas mecânicos.

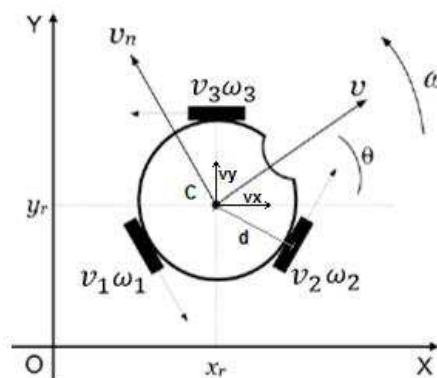


Figura 22. Cinemática do robô móvel e disposição das rodas

Desta forma, a análise da cinemática direta fornece uma descrição do movimento do corpo do robô em função das variáveis das rodas enquanto que com

a análise cinemática inversa obtém-se a relação contrária, ou seja, a descrição dos movimentos das rodas em função das variáveis do corpo do robô.

A partir das posições (x_r, y_r, θ) ao longo do tempo, pode-se calcular as velocidades (v_x, v_y, ω) a partir da derivada, conforme as equações:

$$v_x(t) = \frac{dx(t)}{dt} \quad (7)$$

$$v_y(t) = \frac{dy(t)}{dt} \quad (8)$$

$$\omega(t) = \frac{d\theta(t)}{dt} \quad (9)$$

Com base nas velocidades lineares v_x e v_y , pode-se calcular as velocidades lineares v e v_n , pela equação da cinemática.

$$\begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix} \quad (10)$$

Com isso, a partir da cinemática inversa é possível calcular as velocidades v_x e v_y , conforme a equação:

$$\begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) & 0 \\ \sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix} \quad (11)$$

A relação cinemática entre as velocidades das rodas defasadas (v_1, v_2, v_3) e as velocidades do robô (v, v_n, ω) são descritas conforme a equação:

$$\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} -\sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) & d \\ 0 & -1 & d \\ \sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) & d \end{bmatrix} * \begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} \quad (12)$$

Também aplicando a cinemática inversa, é possível obter as equações que determinam as velocidades do robô em função das velocidades das rodas. Para isto basta inverter a matriz apresentada em (V), resultando na seguinte equação:

$$\begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{2 \cdot \sin(\frac{\pi}{3})} & 0 & \frac{1}{2 \cdot \sin(\frac{\pi}{3})} \\ \frac{1}{2 \cdot (\cos(\frac{\pi}{3}) + 1)} & -\frac{1}{(\cos(\frac{\pi}{3}) + 1)} & \frac{1}{2 \cdot (\cos(\frac{\pi}{3}) + 1)} \\ \frac{1}{2 \cdot d \cdot (\cos(\frac{\pi}{3}) + 1)} & \frac{\cos(\frac{\pi}{3})}{d \cdot (\cos(\frac{\pi}{3}) + 1)} & \frac{1}{2 \cdot (\cos(\frac{\pi}{3}) + 1)} \end{bmatrix} * \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix} \quad (13)$$

Simplificando, tem-se:

$$\begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \\ \frac{1}{3} & -\frac{2}{3} & \frac{1}{3} \\ \frac{1}{3 \cdot d} & \frac{1}{3 \cdot d} & \frac{1}{3 \cdot d} \end{bmatrix} * \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \quad (14)$$

Assim, as equações das velocidades do robô em função das velocidades lineares das rodas são dadas por:

$$v(t) = \frac{\sqrt{3}}{3} \cdot (v_3(t) - v_1(t)) \quad (15)$$

$$v_n(t) = \frac{1}{3} \cdot (v_3(t) + v_1(t)) - \frac{2}{3} \cdot v_2(t) \quad (16)$$

$$\omega(t) = \frac{1}{3 \cdot d} \cdot (v_1(t) + v_2(t) + v_3(t)) \quad (17)$$

3 PLATAFORMA ROBÓTICA

Neste capítulo são abordadas algumas características específicas tanto de *Hardware* quando de *Software* da plataforma Xbot–Scisoccer. Todas as características e funcionalidades levantadas foram embasadas nas documentações e ficheiros do fabricante(Xbot,2015). Os referidos arquivos encontram-se disponíveis no site e em material impresso no Laboratório de Inovações Tecnológicas – Inovtec/UFMA.

3.1 Robô Móvel Scisoccer

O robô móvel Scisoccer é um dispositivo automático capaz de se movimentar e interagir em ambiente estruturado. Desenvolvido pela empresa Cientistas Associados Desenvolvimento Tecnológico Ltda, fabricado e comercializado pela empresa XBot, foi projetado para realizar partidas de futebol de robôs autônomos - classe SmallSize, conhecida como F-180 da Robocup (*Robot World Cup*) (Xbot,2015).

O robô Scisoccer apresentado na Figura 23 possui três rodas suecas igualmente espaçadas, formando um ângulo de 120° entre elas, montada sobre uma base móvel ao qual é acoplado o *hardware* de todo o sistema. Cada um dos três motores está interligado a um driver de potência com limitação na corrente e o espaço entre eles servem para acomodação do sistema de baterias. Aos motores são acoplados caixas redutoras que oferecem um torque maior ao eixo. Sua estrutura básica está descrita na figura 24.



Figura 23. Robô móvel Scisoccer Xbot

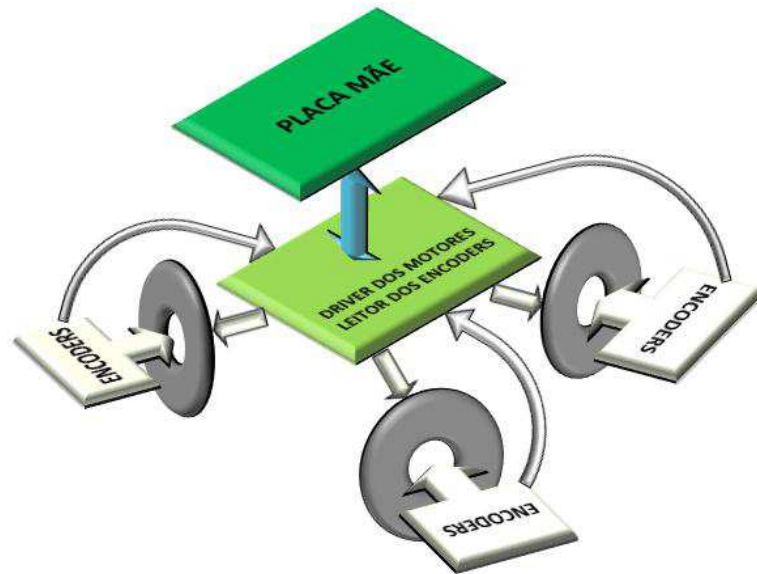


Figura 24. Diagrama de blocos da unidade móvel

3.2 Rádio Base

Este módulo é conectado ao PC pela porta USB por meio da interface serial universal de um microcontrolador MSP430 e configurada para trabalhar em modo assíncrono. A mesma porta do microcontrolador então retransmite o fragmento para o transceptor Nordic nRF2401 (em modo síncrono) que se encarrega do envio do fragmento via rádio para o robô destinatário. A partir deste momento, dependendo do comando enviado na mensagem, o robô poderá enviar alguma informação ao servidor.

O rádio base é um periférico essencial para que a comunicação entre os *softwares* de controle e o robô seja estabelecida. É um rádio wireless que trabalha na frequência de 2.4Ghz digitais (modulação GFSK – *Gaussian Frequency Shift Keying*) e possui dois LEDs de indicação quando ocorre o envio e o recebimento de informações. Na figura 25 temos sua ilustração (Xbot,2015).



Figura 25. Rádio base USB

3.3 Transceiver

O robô Scisoccer possui dois transceivers (dispositivo que combina funções de transmissor e receptor em um único aparelho) da empresa Nordic, ilustrado na figura 26. Um localizado no robô e outro no rádio base, é responsável pela troca de informação entre a placa mãe da unidade móvel e o rádio base conectado ao PC. Segundo o fabricante o raio de alcance máximo em uma transmissão sem obstáculos entre as antenas é a de até 100 metros (Xbot,2015).



Figura 26. Tranceiver RF Nordic

3.4 Módulo Encoder

O controle de velocidade e o torque do robô é efetuado utilizando um modulo *encoder* constituído de um disco perfurado acoplado ao eixo de saída do motor e um circuito fotodetector, desta forma se gera um trem de pulsos que é enviado ao microcontrolador central, demonstrado na figura 27.

O processamento do sinal do fotodetector é realizado utilizando técnica de medição por largura de pulso. Esta técnica consiste em disparar um contador referência após a detecção de uma borda do sinal, parando-o após detecção da borda sucessiva.

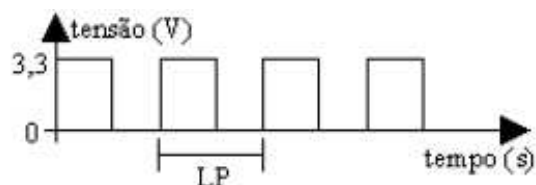


Figura 27. Pulso do fotodetector e indicação da largura de pulso (LP)

Conhecendo-se a base de tempo do contador é possível obter a largura do pulso, dada em unidades de segundos, apenas com a distância entre o valor capturado inicial do contador e o valor capturado final. O parâmetro n indica o número de furos do disco do encoder e com esse dado obtém-se a velocidade (v), em rotações por minuto (rpm), conforme a equação 18. Na figura 28 temos a ilustração do conjunto motor, roda e encoder da plataforma.

$$v = \frac{60}{(LP * n)} \quad (18)$$

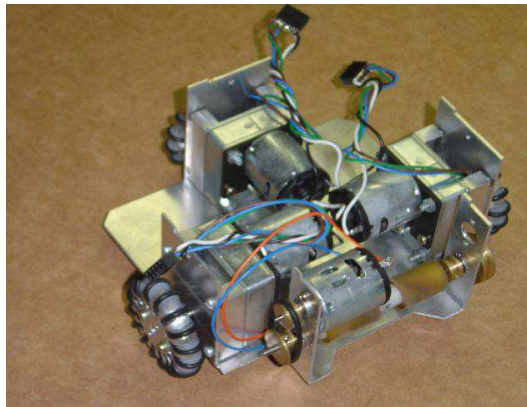


Figura 28. Chassi do robô com três conjuntos motor-roda-encoders

3.5 Alimentação

O sistema de alimentação é composto por duas baterias, Figura 29, de Ni-Mh que fornecem uma tensão de 14.4V e corrente nominal de 2500mA/h. As baterias são em um total de duas para cada robô. São posicionadas na parte inferior da base e formada por 12 pilhas de 1.2V recarregáveis em série.



Figura 29. Baterias e recarregador da plataforma robótica

3.6 Locomoção

Conforme ilustrado na figura 30, o sistema de locomoção do robô consiste em 3 motoredutores e 3 rodas omnidirecionais. Dependendo da combinação do acionamento entre elas, bem como o sentido de rotação, é possível executar qualquer tipo de movimento.

Seu principal componente é a roda omnidirecional, que possui roletes em toda a sua volta, fazendo com que ela possa se deslocar na direção perpendicular ao eixo do motor, sem arrasto.



Figura 30. Moto redutor e rodas do robô Scisoccer

3.7 Placa Mãe e Placa Driver

É a responsável pela centralização e controle da unidade móvel, possuindo um microcontrolador MSP430 que gerencia todas as execuções do *hardware*. Conforme a figura 31, a placa faz a troca de informações com o transceiver, placa de acionamento dos motoredutores (placa driver) e realiza leitura dos encoders.

A placa driver ilustrada na figura 32 é responsável pelo acionamento de potência dos motoredutores e também pelo recebimento das informações dos encoders. Na figura 33, temos o conjunto placa mãe e placa drive devidamente acoplado no chassi do robô.

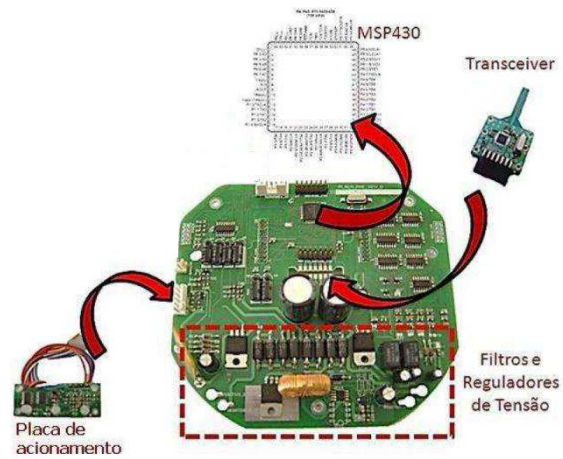


Figura 31. Placa Mãe Robô

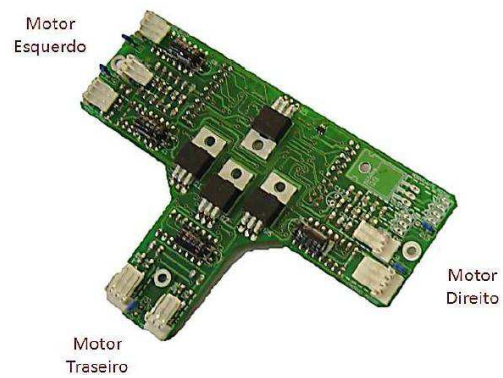


Figura 32. Placa Driver

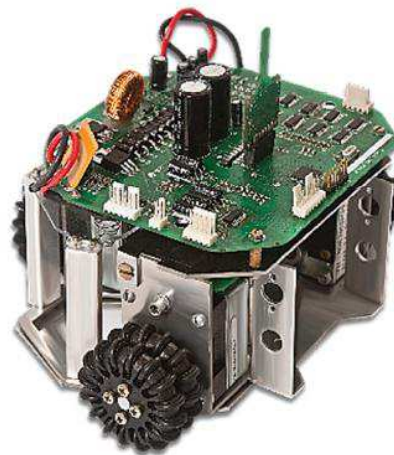


Figura 33. Módulos acoplados na base móvel

3.8 Software para Futebol robôs Scisoccer

O aplicativo de futebol de robôs desenvolvido pela empresa Xbot, denominado Sci-Soccer é uma interface gráfica que envolve quatro módulos: Módulo de captura de imagens, módulo de rastreamento, módulo de comunicação e módulo de estratégia de jogo.

O sistema tem as seguintes características:

- Captura das imagens da câmera para realizar o rastreamento dos robôs e da bola;
- Processamento das imagens recebidas para determinar a posição dos mesmos;
- Carregamento da estratégia de jogo;
- Comunicação entre o robô e o computador.

O *software* é opensource, possibilita que o usuário crie e execute novas estratégias de jogo através das bibliotecas para desenvolvimento de novos códigos. O sistema ilustrado na figura 34 possibilita o usuário jogar as partidas de futebol com as estratégias já pré-programadas pelo fabricante ou criar suas próprias rotinas, favorecendo a criatividade do usuário. O sistema Sci-Soccer foi desenvolvido para ser rodado nos sistemas operacionais Debian 6 ou Ubuntu 10.10.



Figura 34. Software Futebol robôs Sci-Soccer

3.9 Protocolo de comunicação

O protocolo de comunicação da plataforma é baseado em pacotes com um total de 30 octetos (tamanho fixo) no seguinte formato:

- Endereço do dispositivo fonte (5 octetos)
- Endereço do dispositivo destino (5 octetos)
- Canal de rádio utilizado (1 octeto)
- Tamanho dos dados úteis (1 octeto)
- Comando (1 octeto)
- Argumentos do comando (0 a 14 octetos)
- Dígito verificador de integridade – *checksum* (1 octeto)
- *Package-ID* (1 octeto)
- Demais octetos: preenchidos com zero, caso necessário para compor o tamanho fixo de 30 octetos.

A figura 35 ilustra a troca de mensagens entre o computador e o robô. O campo *checksum* (verificação de erros), possibilita que o receptor do pacote verifique se os dados recebidos conferem com os que foram transmitidos. Tanto os pacotes de solicitação quanto os de resposta seguem o mesmo padrão acima descrito. Uma resposta quando comparada a uma solicitação, tem os campos destino e fonte trocados entre si e o comando e *package-ID* permanecem o mesmo.

O campo *package-ID* é um identificador sequencial do pacote. Tem a função de auxiliar caso ocorra falhas na comunicação e em casos de retransmissão de dados

A retransmissão ocorre por *timeout*, ou seja, se após um tempo, o PC não obtiver uma resposta bem-sucedida, ele retransmite o pacote. Após algumas falhas sucessivas, o PC indica o estado de “falha de comunicação”. O tempo de *timeout* e o número de falhas consecutivas é configurável no driver do PC, sendo 50ms para o *timeout* e 3 falhas sucessivas.

Motores, rodas, chassis, dispositivos eletromecânicos e comunicação *wireless* presentes no robô Scisoccer foram desenvolvidos considerando todas as normas da Robocup, categoria SmallSize (BIANCHI *et al.*, 2006).

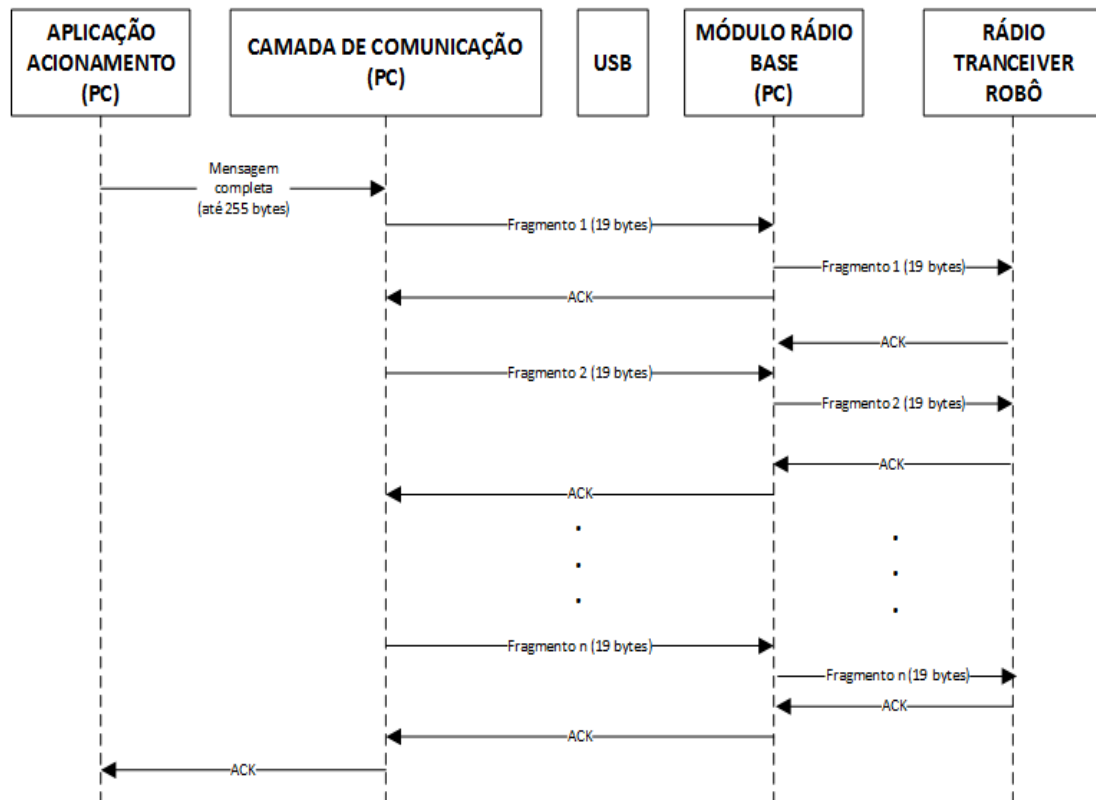


Figura 35. Mensagens (PC –Rádio Base-Robô)

4 VISÃO COMPUTACIONAL

4.1 Imagem Digital

Uma imagem digital é vista computacionalmente como uma matriz no qual os índices de linhas e colunas representam um ponto na imagem(pixel), sendo a intensidade (brilho) o respectivo valor do elemento dessa matriz. Na figura 36 tem-se um exemplo de uma imagem digital.

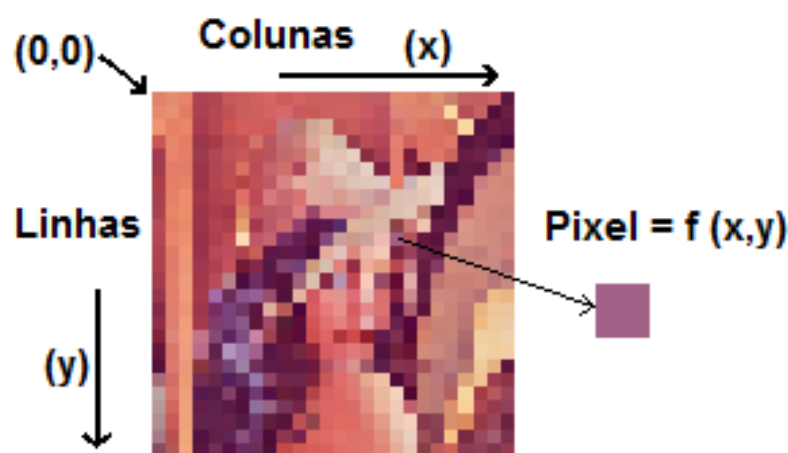


Figura 36. Imagem Digital

4.2 Processamento de Imagens Digitais

Quando uma imagem é digitalizada ela é guardada na memória do computador conforme uma matriz de dados onde cada elemento é definido como um pixel da imagem. Segundo (GONZALEZ; WOODS 2000) uma imagem pode ser definida como uma função $f(x, y)$, onde o valor das coordenadas x e y correspondem ao brilho (intensidade) da imagem nessa coordenada. Esta matriz numérica é, então, processada pelo computador para extrair as informações desejadas da imagem (ORTH *et al.*, 2001).

No processamento digital de imagens há basicamente 3 tipos de imagens: em escala de cinza (1 banda), colorida (3 bandas) e Imagem binária (dois valores possíveis para indicar cor branca e preta). No caso da imagem em escala de cinza, figuras 37 e 38, a intensidade dos pontos da imagem, ou seja, o valor do pixel, varia

em uma escala entre 0 e 255, onde 0 (zero) indica a cor preta e 255 a cor branca, tendo nesse intervalo a intensidade do cinza.



Figura 37. Níveis de intensidade de cinza

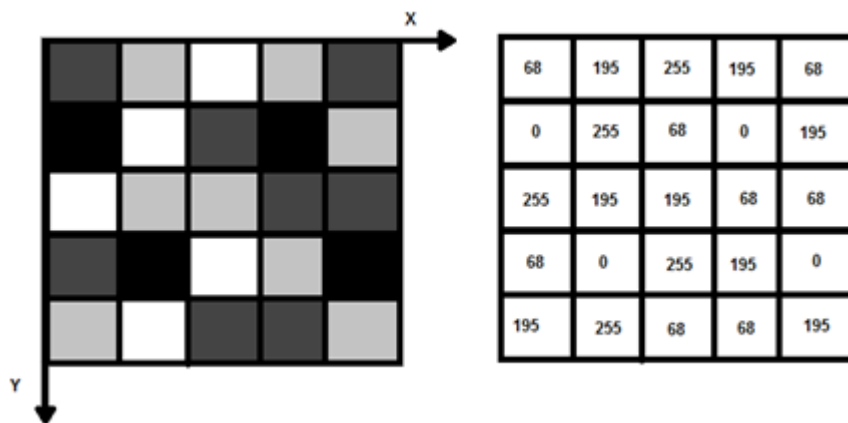


Figura 38. Imagem ampliada (5x5 pixels) em níveis de cinza

Já nas imagens coloridas, ou seja, aquelas que possuem mais de uma banda de frequência ilustrado na figura 39, cada banda possui sua própria função de intensidade de brilho. No padrão RGB, figura 40, a imagem possui três bandas que compõem as três cores primárias, *RED*(Vermelho), *GREEN*(Verde) e *BLUE*(Azul).

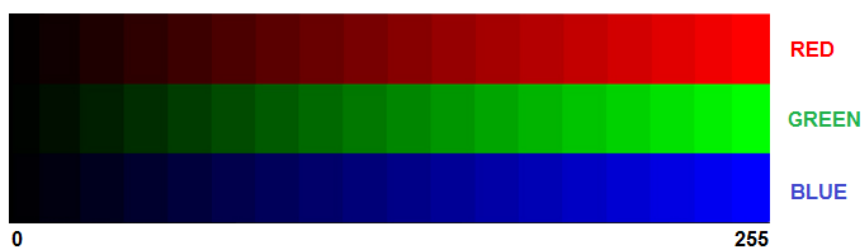


Figura 39. Níveis de intensidade das bandas RGB

Analisando, pode-se considerar que uma imagem colorida é a composição de três imagens monocromáticas, denominadas, respectivamente, de banda vermelha (ou banda R), banda verde (ou banda G), e banda azul (ou banda B) e que a sobreposição destas três imagens, que individualmente são monocromáticas, compõe uma imagem colorida. E com a sobreposição/mistura de cada uma dessas cores e suas respectivas intensidades obtêm-se a formação de outras cores, conforme figura 40.

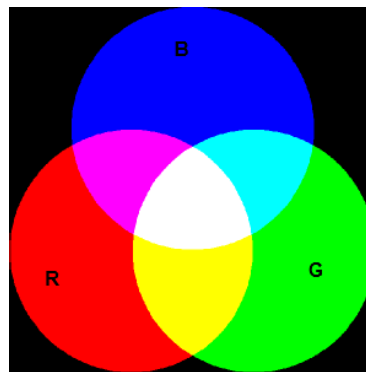


Figura 40. Imagem das cores R, G, B e RGB

Outro padrão de imagem colorida é o HSI (*Hue*, *Saturation* e *Intensity*), ilustrado na figura 41. O mesmo divide as cores *Hue* (Matiz), *Saturation* (Saturação) e *Intensity* (Intensidade), sendo a matiz a cor propriamente dita, a saturação identifica o quanto a cor é forte e a intensidade luminância da cor (PENHARBEL,2004).

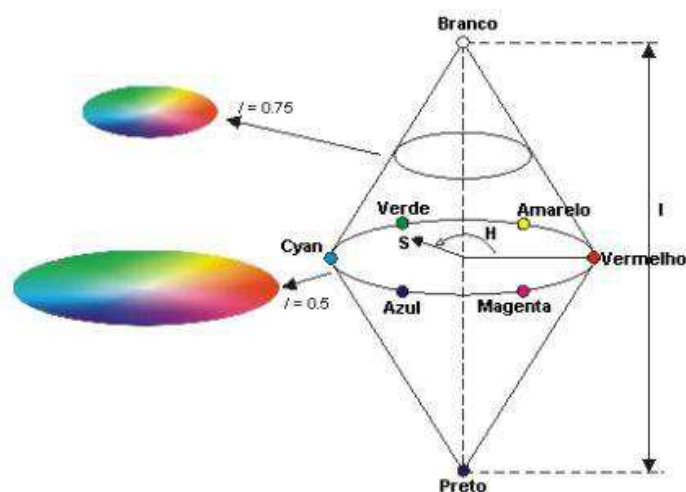


Figura 41. Representação de cores padrão HSI

Conforme a imagem 42, nas imagens binárias, tem-se apenas duas opções de intensidade do pixel, o preto ou branco. Dependendo da imagem e do problema a ser processado, o preto pode representar o fundo da imagem e o branco os objetos ou vice-versa.



Figura 42. Imagem binarizada

4.3 Estrutura de um Sistema de Visão Artificial

Conforme (MARQUES FILHO; VEIRA NETO ,1999) um Sistema de Visão Artificial (SVA) é um sistema computacional capaz de adquirir, processar e interpretar imagens correspondentes a cenas reais. A figura 43 mostra o diagrama em blocos dos elementos que compõe um SVA e suas principais etapas.



Figura 43. Fluxograma dos elementos de um (SVA)

4.3.1 Aquisição da imagem

(CONCI *et al*, 2008) explica que no momento da aquisição de uma cena real (tridimensional) ela é convertida através de um dispositivo de aquisição CCD

(*Charge Couple Device*) em uma representação 2D levando-se em consideração uma função $f(x,y)$, que descreve a energia eletromagnética luminosa em uma matriz de dados digitais de coordenadas espaciais (x,y) . Um conjunto de prismas e filtros decompõe a imagem em componentes de R (*red*), G (*green*) e B (*blue*), gerando o sistema de cor RGB, muito utilizado em câmeras e monitores de vídeo.

4.3.2 Pré-processamento

Conforme (GONZALEZ; WOODS, 2000), nesta etapa ocorre uma melhoria da imagem digitalizada vinda do processo anterior. Técnicas como aumento de contraste, remoção e redução de ruídos, realce e normalização, são aplicadas na imagem afim de corrigir tais imperfeições e assim dar prosseguimento para as próximas etapas.

Esta fase também é conhecida por etapa de baixo nível, pois trabalha diretamente manipulando os valores dos pixels, sem nenhum conhecimento prévio da imagem adquirida. O resultado desta etapa é uma imagem digitalizada de melhor qualidade que a original em função da aplicação das técnicas de suavização (filtros), realce e detecção de bordas.

4.3.3 Segmentação

Em geral a etapa da segmentação é a divisão de uma imagem de entrada em suas unidades de interesse, ou seja, nos alvos de interesse que a compõem. Apesar da simples descrição, a segmentação constitui um dos elementos de maior dificuldade de implementação do processamento de imagem.

A segmentação é realizada por diversos algoritmos, os quais são baseados em duas características básicas: descontinuidade e similaridade. Na descontinuidade, a divisão da imagem é fundamentada no subconjunto de pontos de um objeto que o segrega da totalização da imagem. Já na similaridade, a segmentação é com base nas técnicas de limiarização (utilizada neste trabalho), crescimento por regiões, união e divisão de regiões (GONZALEZ; WOODS, 2000).

4.3.4 Seleção de Características

Segundo (PEDRINI; SCHWARTZ, 2008) esta etapa busca extrair características convenientes das imagens resultantes da segmentação. A saída da segmentação tem como característica um conjunto de dados (*pixels*) ainda inconsistentes. Assim, este estágio é necessário para haja uma representação mais apropriada dos dados, garantindo o processamento pelo computador.

A descrição dos elementos de uma imagem deve ser representada por uma estrutura de dados adequada ao algoritmo de reconhecimento. É importante frisar que nesta etapa a entrada ainda é uma imagem, mas a saída é um conjunto de dados relacionados àquela imagem. Resulta na extração de características que diferenciem uma classe de objetos de uma outra.

4.3.5 Reconhecimento e Interpretação

Após a extração das características de um objeto da imagem, a próxima etapa é identificar, reconhecer o que cada uma dessas classes representa e a partir desse ponto, atribuir um determinado valor, um rótulo baseado em suas características. A interpretação baseia-se na atribuição de um determinado significado para as características de um objeto reconhecido pelo sistema de visão artificial (PEDRINI; SCHWARTZ, 2008).

4.4 Biblioteca OpenCV

OpenCV é uma biblioteca (*open-source*) para programação de visão computacional (OPENCV, 2015). Desenvolvida pela Intel Corporation, projetada em linguagem C e C++ e idealizada para tornar a programação de visão computacional mais acessível a usuários e desenvolvedores de áreas como: IHM (interação humano-computador) em tempo real e robótica.

A biblioteca permite ligação automática a uma DLL (*Dynamic Linked Library*) e já vem customizada para ser executada por diversos tipos de processadores. São funções de alto-nível que possuem uma gama de ferramentas que permitem a realização de interpretação de imagens, indo de um simples filtro de ruído, até operações complexas, como a análise de movimentos, reconhecimento de padrões

e aprendizagem de máquina tornando mais fácil a resolução de problemas relacionados a visão computacional.

A biblioteca possui mais de 2500 algoritmos otimizados. Esses algoritmos podem ser usados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, movimento de objetos em uma pista, extrair modelos 3D de objetos, encontrar imagens similares a partir de um banco de imagens, remover olhos vermelhos de imagens tiradas com flash, reconhecer cenários e estabelecer marcadores com a realidade aumentada, etc. A biblioteca OpenCV possui mais de 47 mil usuários e comunidades e o número de downloads já ultrapassa os 7 milhões (OPENCV,2015). A biblioteca é amplamente utilizada por empresas, grupos de pesquisas, entusiastas e órgãos governamentais. Empresas como a Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda e Toyota já empregam o uso da ferramenta. Ela tem interfaces para C, C++, Python, Java e MATLAB e está disponibilizado para os sistemas operacionais Windows, Linux, Android e Mac OS.

O download encontra-se disponível em <http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.4.9/opencv-2.4.9.exe/download>.

4.4.1 Estrutura e conteúdo da biblioteca OpenCV

Em geral, a estrutura da biblioteca OpenCV é dividida em 4 módulos, conforme a Figura 44:

- **CV:** Contém funções para processamento de imagens, análise de estrutura de imagens, funções para detecção de movimento e rastreamento, padrões de reconhecimento e calibração de câmera;
- **MLL:** Possui funções para classificações estatísticas e ferramentas para *clustering* (vários PCs em rede trabalhando em paralelo);
- **HighGUI:** Bloco destinado para geração de funções de controle de interface e dispositivos de entrada e saída;

- **CXCORE:** Módulo que contém as estruturas de dados, funções para transformações de dados, persistência de objetos, gerenciamento de memória, manipulação de erros, e também funções para desenho, texto e operações básicas de cálculos matemáticos.

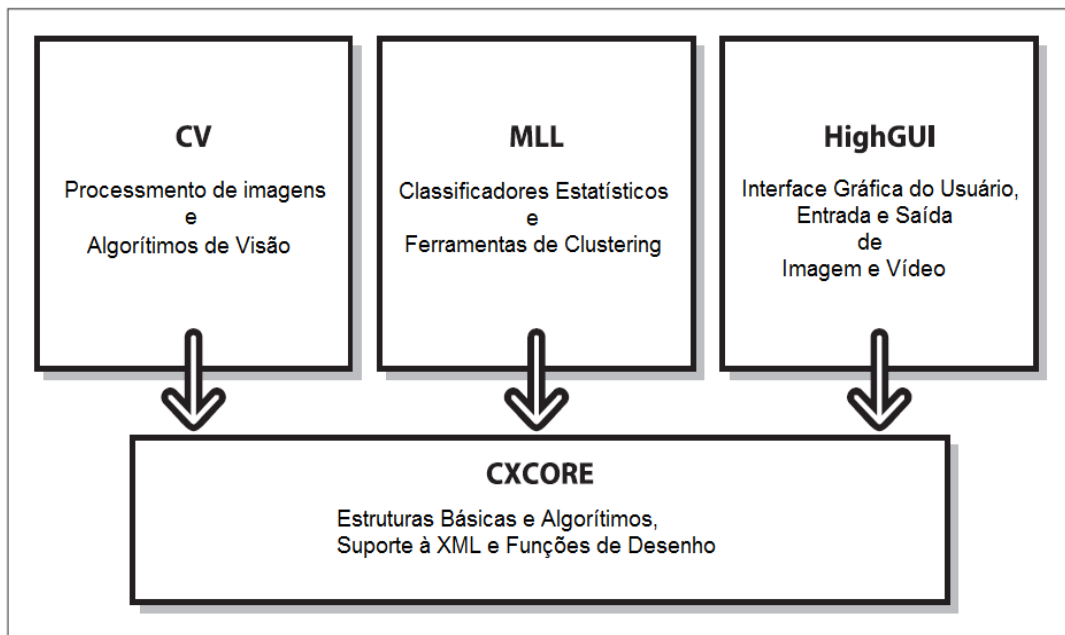


Figura 44. Estrutura básica da biblioteca OpenCV (BRADISK; KAEHLER, 2008)

5 EXPERIMENTOS E RESULTADOS

Neste capítulo será apresentada a síntese dos resultados bem como a descrição dos algoritmos de medição, seu desempenho e as suas características gerais de funcionamento.

5.1 Requisitos de Hardware do Sistema

Para que o sistema realize as medições foram utilizados os seguintes elementos de *hardware*:

- Uma câmera USB (*webcam*) colorida, posicionada sobre o campo a 3,3m de altura;
- Um cabo extensor de 10m *USB* com amplificação de sinal;
- Plataforma robótica móvel Scisoccer;
- Netbook com CPU Intel Atom 1 Ghz, 2 GB RAM e HD de 500Gb

Em conjunto com o *hardware* são necessários os softwares: biblioteca de acionamento curumim (kit do ambiente de programação da plataforma robótica educacional desenvolvida pela empresa xbot), Codeblocks13.1 (*Integrated Development Environment*, IDE para linguagem de programação C/C++), OpenCV 2.4.2 e ImageJ (software baseado em java para processamento de imagem).

5.2 Funcionamento e Execução do Sistema

Conforme ilustrado na figura 45, o funcionamento geral da proposta é realizado em três etapas: sistema de acionamento, operação *off-line* e processamento de imagem com sistema de medição.

5.2.1 API de acionamento do robô Scisoccer

Dotar um robô com a capacidade de executar movimentos não é uma tarefa simples. O acionamento da plataforma é um requisito fundamental para o desenvolvimento de projetos em robótica móvel. A plataforma Sci-soccer foi desenvolvida para realização de partidas de futebol de robôs e, dessa forma, para acioná-los de forma independente foi desenvolvido um aplicativo específico para a

realização de tarefas elementares como mover-se para frente, para trás, virar, entre outras ações.



Figura 45. Estrutura geral do sistema

Para o desenvolvimento da API foi importante a leitura de documentos, manuais e bibliotecas do *software* Sci-Soccer, principalmente a biblioteca de comunicação. Os referidos documentos podem ser encontrados para download no site da empresa XBot (www.xbot.com.br). O aplicativo foi implementado em duas versões. Em object pascal (Delphi) e C++. A aplicação em Delphi pode ser classificada como um (*fork*, uma ramificação) da giga de testes cedida para estudo pela empresa Xbot. Do aplicativo de exemplo da Xbot, foram derivadas as especificações, cabeçalhos de funções, tipificação, e acrescentadas funcionalidades e melhorias, que não existem na giga de testes. O projeto modificado permite criar um *dataset* (conjunto de dados) com os parâmetros de trajetória e carregar esse *dataset* para posterior execução do percurso. Outra nova funcionalidade é que é possível simplesmente digitar uma ou mais trajetórias, e depois executar cada linha de parâmetros em pilha.

Devido as similaridades entre as plataformas robóticas *curumim* e Scisoccer, a implementação da API de acionamento em C++ do robô Scisoccer foi desenvolvida com base na biblioteca de comunicação do *curumim*. Assim, todas as rotinas existentes para manipulação das variáveis, parâmetros e lista de comandos da biblioteca de comunicação e acionamento do robô *curumim* estão disponíveis para ambas plataformas.

Para cada comando listado se encontram, entre parênteses, os tipos de cada parâmetro solicitado e/ou tipo de seu retorno. Como exemplo, na figura 46 é ilustrado os robôs sendo acionados e movimentando-se a partir do comando “move”, cuja assinatura é:

`move (<endRobo>, <endRbase>, <distDiant>, <distTras>, <velocidade>)`,
onde:

- `<endRobo>`: endereço do robô (inteiro 8 bits sem sinal);
- `<endRbase>`: endereço do rádio base (inteiro 8 bits sem sinal);
- `<distDiant>`: distância a ser percorrida pelos dois motores dianteiros, em milímetros;
- `<distTras>`: distância a ser percorrida pelo motor traseiro, em milímetros;
- `<velocidade>`: velocidade a ser assumida pelo(s) motor(es) que é (são) utilizado(s), em rpm.



Figura 46. Robôs acionados a partir do aplicativo desenvolvido

5.2.2 Operação *off-line*

O processo *off-line* é composto por duas etapas conforme descrição a seguir:

a) Aquisição e calibração do sistema de visão

A captação é a aquisição de imagens de um cena real tridimensional convertida por um dispositivo de aquisição em uma representação 2D, levando-se em conta uma função $f(x,y)$ que descreve a energia luminosa emitida em uma coordenada espacial (x,y) .

Já a calibração de sistemas de visão é o processo de determinação dos parâmetros e elementos que determinam a relação entre a imagem bidimensional que a câmera reconhece e a informação tridimensional da imagem real do objeto (MILESKI, 2014). A ideia por trás da calibração é a determinação do relacionando entre as coordenadas conhecidas de um conjunto de pontos 3D e suas projeções, afim de determinar os parâmetros da câmera. Esses parâmetros podem ser subdivido em parâmetros intrínsecos(internos) e extrínsecos(externos) da câmera. Este trabalho adota uma solução de calibração por observação. Este processo é baseado na observância da posição detectada pela câmera para o alvo em diversas posições no ambiente, a partir da calibração da imagem resultante distorcida. Além destes fatores, também adotou-se que o CCD é uma matriz regular $f(x,y)$, sendo (x,y) as coordenadas em pixels da imagem na qual baseia-se a observância captada pela câmera, conforme as figuras 47 e 48.

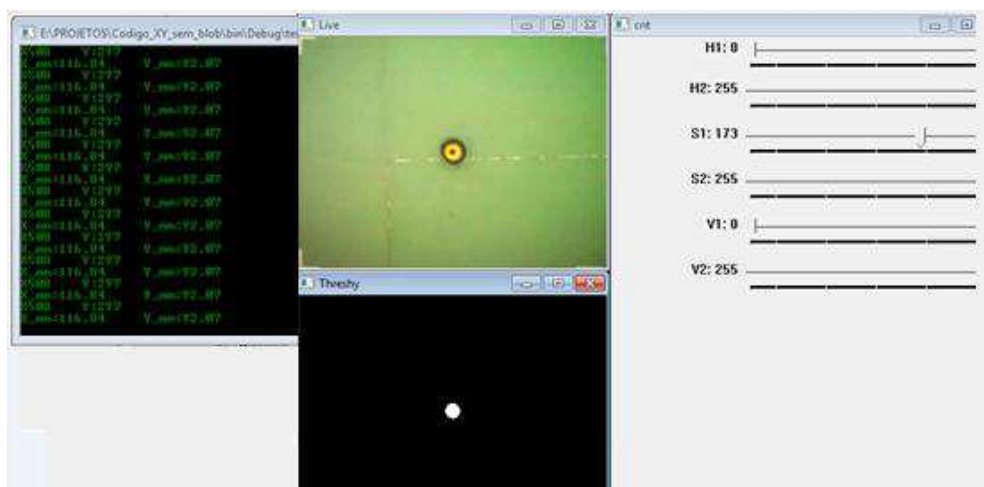


Figura 47. Exemplo de calibração por observação

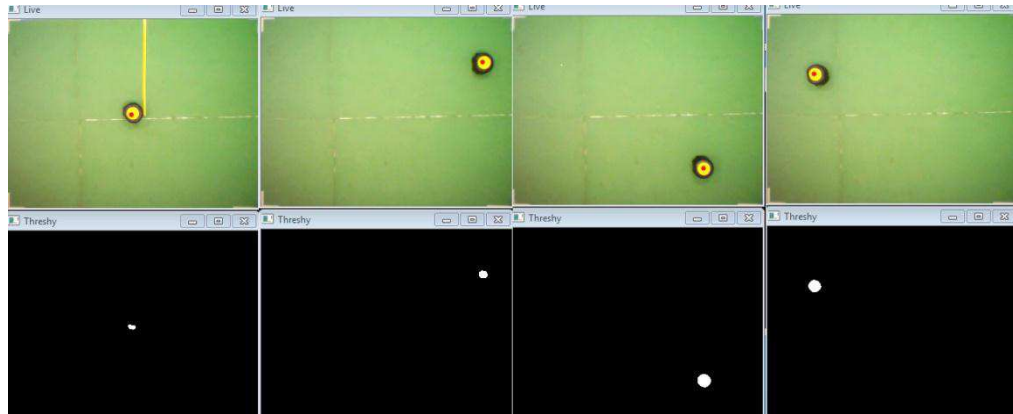


Figura 48. Calibração sistema de visão em diferentes posições

O sistema de calibração por observação foi desenvolvido para retornar as coordenadas x e y em pixels da área central da imagem. Assim, a partir da medição física das coordenadas centrais informadas pelo sistema obtém-se o dimensionamento de toda área da imagem, além da obtenção da relação pixels/milímetros.

A maioria das lentes das câmeras de vídeo distorcem a imagem (*barrel distortion*), ou seja, em vez de bordas retas, surgem nas extremidades bordas arredondadas. Antes do processamento da imagem é necessário a correção desta distorção. Para determinação da imagem sem distorções o OpenCV fornece um algoritmo capaz de obter os parâmetros de calibração da câmera e uma imagem resultante sem distorções a partir de um padrão de pontos do tabuleiro de xadrez (OPENCV,2015), conforme ilustrado na figura 49. Uma vez estimados estes parâmetros, os mesmos não necessitam mais ser reprocessados, desde que a câmera não tenha sua posição alterada.

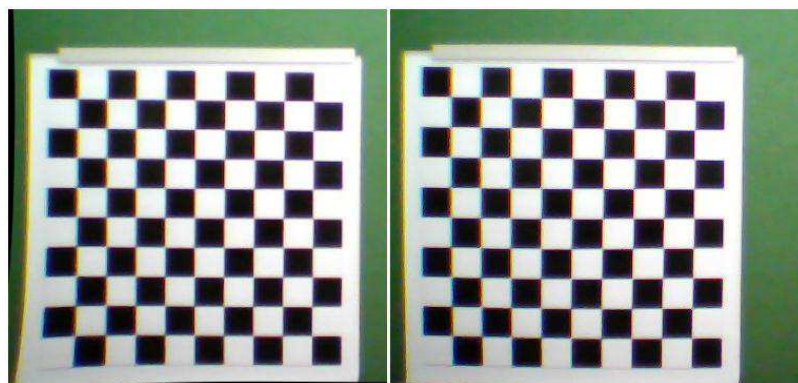


Figura 49. Calibração e correção de distorção

Para complementar o sistema de calibração, também foi utilizado o *software* livre *ImageJ*. Esse sistema foi desenvolvido para análise e processamento de imagens digitais, possui vários recursos e é totalmente *open-source*. Pode ser aplicado em várias áreas de conhecimento da computação gráfica e está totalmente disponível no site <http://rsb.info.nih.gov/ij/>. Nessa etapa, utilizou-se o dimensionamento da quantidade de *pixels* nos eixos x e y da imagem do ambiente, figura 50, além do fator de escala (*pixels/ cm*), ilustrado na figura 51.



Figura 50. Posicionamento do robô no ambiente

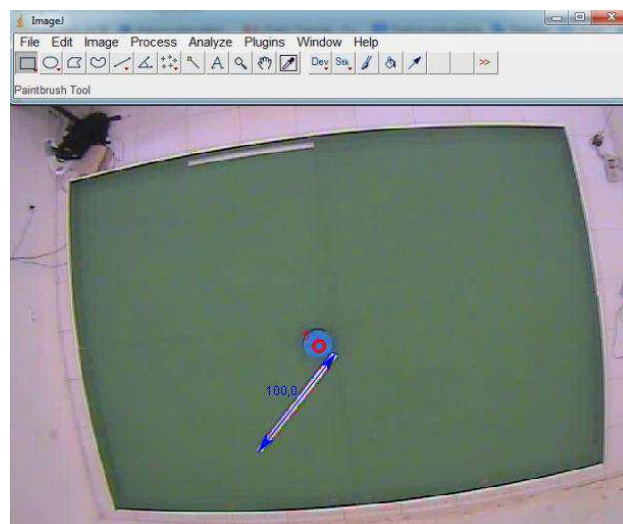


Figura 51. Sistema de calibração

A tabela 2 apresenta os dados da calibração obtidos em 10 diferentes medidas referentes as coordenadas do robô no ambiente nas unidades pixels e centímetros. A altura da câmera posicionada no teto é de 3,3m, a resolução da câmera é de 640X480, a resolução da imagem (monitor de vídeo) é de 1024X768 e os fatores da conversão em x e y são:

$$e_x = \frac{\text{pixels}_x}{\text{cm}} = 4,49 \text{ px/cm}$$

$$e_y = \frac{\text{pixels}_y}{\text{cm}} = 3,56 \text{ py/cm}$$

Tabela 2: Resultados coletados após sistema de calibração

Posição	Imagem		Medição Imagem		Medição Ambiente		Erro		%Erro
	X(px)	Y(px)	X(cm)	Y(cm)	X(cm)	Y(cm)	X(cm)	Y(cm)	
1	508	297	113,14	83,43	108,00	90,00	-5,14	6,57	4,76%
2	902	140	200,89	39,33	190	45,00	-10,89	5,67	5,73%
3	780	485	173,72	136,24	165	148,00	-8,72	11,76	5,28%
4	188	187	41,87	52,53	37	63,00	-4,87	10,47	13,16%
5	443	360	98,66	101,12	96	104,00	-2,66	2,88	2,77%
6	153	326	34,08	91,57	28,5	99,00	-5,58	7,43	19,56%
7	100	542	22,27	152,25	21	155,00	-1,27	2,75	6,06%
8	962	524	214,25	147,19	204	157,50	-10,25	10,31	5,03%
9	845	304	188,20	85,39	200	89,00	11,80	3,61	-5,90%
10	238	187	53,01	52,53	60	61,00	6,99	8,47	-11,66%
Média							-3,06	6,993	4,48%
Erro Max							11,8	11,76	
Erro Min							-10,9	2,753	

b) Computação dos parâmetros: (Δ_q) e (Δ_t)

Para o desenvolvimento desta etapa foram estimados dois parâmetros essenciais para o algoritmo de estimação: a quantidade de quadros executados por segundo (Δ_q) e tempo levado na execução do processamento da imagem (Δ_t).

O aplicativo foi desenvolvido utilizando a biblioteca OpenCV. Inicialmente adotou-se como marcação inicial e final de tempo, as variáveis <start> e <end> respectivamente, ambas do tipo <time_t>, da biblioteca <time.h> que contém os

protótipos padrões para manipulação de tempo e data na linguagem C/C++. Para o contador de quadros, utilizou-se um incremento da variável <counter>, conforme a figura 52. Após a captura do primeiro frame, as variáveis passam a contabilizar respectivamente as operações da diferença de tempo entre captura e processamento de imagem (Δ_t), e quantidade de *frames* capturados por segundo (Δ_q).

Durante este procedimento, foram adequadas a iluminação do ambiente, desinstalados todos os programas utilitários, como antivírus e outros componentes desnecessários do Windows, pois foi constatado que em casos de mudança de luminosidade do ambiente, execução de algum programa do Windows ou outro aplicativo rodando em paralelo, o tempo de computação (Δ_t) e (Δ_q) aumentou.



Figura 52. Processamento dos parâmetros: (Δ_q) e (Δ_t)

5.2.3 Processamento de imagem com sistema de medição

A execução deste algoritmo é rodada a partir da finalização e ajustes dos parâmetros realizados na operação *off-line*. Assim, o sistema de processamento e medição também usa a biblioteca padrão OpenCV para reconhecimento e estimação da localização, cálculo das velocidades e aceleração do robô.

5.3 Algoritmo de Medição Proposto

Após as etapas e ajustes no processamento *off-line*, o passo seguinte é a execução do algoritmo de medição visual. O sistema foi desenvolvido conforme o fluxograma demonstrado na figura 53.

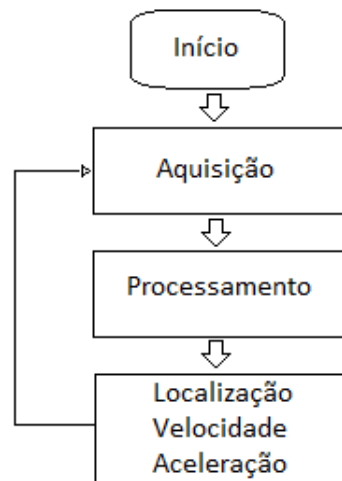


Figura 53. Fluxograma do algoritmo de medição

5.3.1 Aquisição de imagens

O processo de captura e aquisição de uma imagem usando o OpenCV é efetuado usando a estrutura `<CvCapture>` que está declarada nos cabeçalhos da biblioteca. Os métodos desta estrutura podem ser utilizados para captura de vídeo de uma câmera em tempo real ou para captura de imagem de um vídeo pré-gravado. Após a criação da estrutura, pode-se realizar as solicitações de um *frame* atual de vídeo capturado.

5.3.2 Correções e Filtros

A suavização é aplicada para correção e remoção de ruídos nas imagens digitais. No OpenCV, diferentes algoritmos de filtro passa baixa são implementados a partir de uma única função, mudando apenas seus parâmetros de entrada. De acordo com a figura 54, a função `<cvSmooth>` possui os seguintes parâmetros (*src*, *dst*, *type*, *param1*, *param2*, *param3*, *param4*), onde temos:

1. *src* - imagem de entrada ou de origem
2. *dst* - imagem de saída ou de destino
3. *type* - algoritmo de filtragem
4. *param1* - tamanho da janela de filtragem
5. *param2* - desvio padrão nas colunas
6. *param3* - desvio padrão nas linhas
7. *param4* - tipo de cálculo feito nas bordas da imagem

```
72 | cvSmooth(p_imgOriginal, smooth, CV_GAUSSIAN, 3, 3, 0, 0);
73 |
```

Figura 54. Estrutura cvSmooth

Outro parâmetro implementado foi a utilização do espaço de cor HSV, em vez do espaço RGB. Em HSV, para cada "tom" de cor é dado um número específico (Tonalidade). Isso torna mais fácil a escolha de uma cor com base em sua tonalidade.

No OpenCV, tem-se a função <cvCvtColor> que converte os espaços de cores das imagens, conforme a assinatura: cvCvtColor (origem, destino, codigo_conversor).

5.3.3 Segmentação

Existem várias funções para a segmentação de imagens disponíveis no OpenCV. Dentre elas, destaca-se a função utilizada neste trabalho <cvInRangeS>.

Conforme a figura 55, a função aplica uma limiarização retornando como pixel igual a 255 os pixels que estiverem entre dois limiares. O parâmetro <p_imgHSV> é a imagem convertida do padrão RGB para HSV, <cvScalar> é uma função que representa a menor e a maior intensidade da cor e <p_imgProcessed_robo> é a imagem binarizada em 8 bits, que receberá a limiarização.

```

102 | cvInRangeS(p_imgHSV,
103 |           cvScalar(H_MIN_robô, S_MIN_robô, V_MIN_robô),
104 |           cvScalar(H_MAX_robô, S_MAX_robô, V_MAX_robô),
105 |           p_imgProcessed_robô);
---
```

Figura 55. Função cvInRangeS

5.3.4 Extração e identificação de atributos

Para extrair informações de imagens segmentadas, a biblioteca OpenCV possui várias estruturas de análise e descrição de formas. Neste trabalho foi usado a função <cvMoments>, que calcula todos os momentos da imagem ou momentos estatísticos até a terceira ordem de um polígono. Através desses momentos é possível achar o centro de massa x e y de um objeto, representados pelos momentos $\frac{M_{10}}{M_{00}}$ e $\frac{M_{01}}{M_{00}}$, respectivamente, onde M_{00} é o momento que representa a área do objeto. Na figura 56, linha 123, temos o extrato de código onde inicialmente foi alocado memória para armazenar a estrutura de momentos e posterior cálculo dos momentos da imagem segmentada.

```

123 | CvMoments *moments = (CvMoments*)malloc(sizeof(CvMoments));
124 |
125 | cvMoments(p_imgProcessed_robô, moments, 1);
```

Figura 56. Função cvMoments

O passo seguinte é encontrar as coordenadas x e y dos momentos calculados do robô. Conforme ilustrado na figura 57, para posição x, basta realizar a divisão da variável de primeira ordem (moment10) pela variável de segunda ordem (área) e para posição y, o resultado da divisão da variável de primeira ordem (moment01) pela variável de segunda ordem (área).

Após a captura das posições das coordenadas x e y, é necessário armazená-las em variáveis estáticas, de modo que seus valores possam ser armazenados e mostrados de forma contínua.

```

137 static int x1,y1;//Armazena as ultimas e atuais posições
138
139 x1 = moment10/area;//Calculo da posição x
140 y1 = moment01/area;//Calculo da posição y
141
142
143 // Imprimindo as posições . . .
144 printf("Posicao (X,Y)do Robo: (%d, %d)\n", x1, y1);

```

Figura 57. Localização das coordenadas x e y do robô

5.3.5 Cálculo: deslocamento, velocidades e aceleração

Conforme apresentado no capítulo 3 e ilustrado na figura 58, a localização de um objeto no plano 2D é dado pela sua coordenada x e y, que tem seu vetor posição dado por: $r = x(i) + y(j)$.

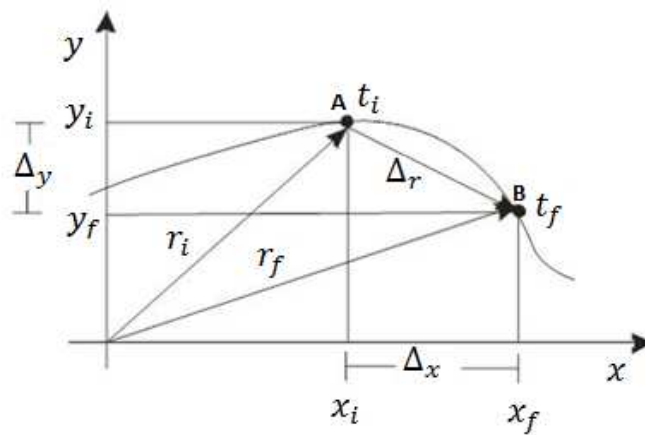


Figura 58. Deslocamento em duas dimensões

O deslocamento é dado pela variação da posição $\Delta \mathbf{r}$, assim:

$$\Delta \mathbf{r} = \mathbf{r}_f - \mathbf{r}_i \quad (19)$$

$$\Delta \mathbf{r} = (x_f - y_f) - (x_i - y_i) \quad (20)$$

$$\Delta \mathbf{r} = (x_f - x_i) + (y_f - y_i) \quad (21)$$

Simplificando, tem-se:

$$\Delta r = \Delta x(i) - \Delta y(j) \quad (22)$$

Assim, sua velocidade é dada em função dos deslocamentos nos dois eixos do plano em um dado intervalo de tempo Δt .

$$\vec{v} = \frac{\Delta r}{\Delta t} = \frac{\Delta x \cdot i}{\Delta t} + \frac{\Delta y \cdot j}{\Delta t} = \bar{v}_x i + \bar{v}_y j \quad (23)$$

As projeções do vetor velocidade nos dizem o valor absoluto da velocidade nos eixos x e y, ou seja, sua cinemática direta, sendo:

$$\vec{v}_x = \vec{v} \cdot \cos(\theta) \quad (24)$$

$$\vec{v}_y = \vec{v} \cdot \sin(\theta) \quad (25)$$

Assim, a velocidade resultante é definida pela equação 26 conforme as projeções $\vec{v}_x$ e $\vec{v}_y$ ilustrada na figura 59.

$$\vec{v} = \sqrt{(\vec{v}_x)^2 + (\vec{v}_y)^2} \quad (26)$$

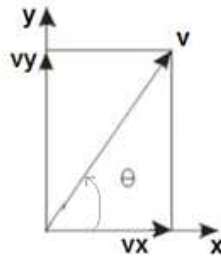


Figura 59. Projeções do vetor velocidade

Para a aceleração tem-se:

$$\vec{a} = \overrightarrow{ax}(i) + \overrightarrow{ay}(j) \quad (27)$$

$$\overrightarrow{ax} = \frac{\Delta v_x}{\Delta t} = \frac{vx_f - vx_i}{\Delta t} \quad (28)$$

$$\overrightarrow{ay} = \frac{\Delta v_y}{\Delta t} = \frac{vy_f - vy_i}{\Delta t} \quad (29)$$

$$\vec{a} = \sqrt{(\overrightarrow{ax})^2 + (\overrightarrow{ay})^2} \quad (30)$$

A partir desta análise, a proposta para a estimação da velocidade é baseada nas distâncias euclidianas entre dois pontos de interesse da imagem, ou seja, a diferença entre suas coordenadas x e y em pixels, em um intervalo de tempo. Desta forma, obtêm-se a relação:

$$v = \frac{\Delta q}{\Delta t} \quad (31)$$

Onde:

v -> Velocidade de um determinado ponto da imagem

Δq -> Deslocamento desse ponto na imagem, executadas dentro da quantidade de quadros captados por segundos

Δt -> Intervalo de tempo entre aquisição e processamento de imagem

Assim, a definição do algoritmo para estimação da posição, velocidade e aceleração é:

1. Localizar as coordenadas x e y do robô colorido
2. Rastrear sua movimentação no ambiente
3. Determinar a distância percorrida pelo robô nos primeiros 15 *frames*
4. Calcular sua velocidade e aceleração

5.4 Experimentos

Na etapa de experimentação são realizados os testes e a verificação do sistema de medição visual. Assim, a proposta foi sistematizada nas seguintes etapas:

- Acionamento da plataforma robótica Scisoccer através dos sistemas desenvolvidos;
- Calibração do sistema;
- Execução do sistema de medição e armazenamento dos dados.

5.5 Resultados

Para avaliação dos resultados do sistema foram realizadas 10 medições de uma mesma trajetória. Em função do tempo disponível, não foi possível ampliar a quantidade do número de testes e trajetórias disponíveis no ambiente no sentido de

avaliar a qualidade e a confiabilidade das medições em várias posições. Foi registrado e computado para efeito de análise dos resultados apenas a média dos deslocamentos, velocidades e acelerações em uma única trajetória retilínea. Em todas as etapas das medições os resultados foram satisfatórios e dentro dos parâmetros esperados (erro médio até 10%). A tabela 3 é exibida a lista com todos os parâmetros finais obtidos pelo sistema de medição e nas figuras 60 e 61 apresentam as respectivamente localizações x e y do robô no ambiente e na imagem.

Tabela 3: Parâmetros finais do sistema

MEDIDA	VALORES
Tamanho da imagem capturada	640X480 px
Tamanho da imagem processada	320X240 px
Taxa de aquisição de imagens da câmera	30 quadros por segundo
Taxa de aquisição de imagens do sistema	15 quadros por segundo
Área do ambiente	2,25m X 1,67m
Tempo de execução entre captura e processamento	0.587 segundos
Fator de conversão - pixels/cm	4,49px/cm – 3,56py/cm

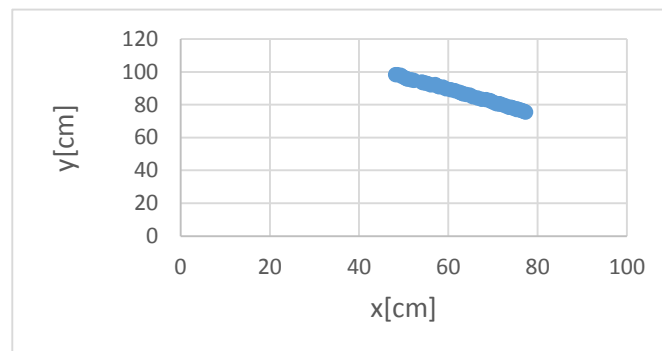


Figura 60. Localização x e y do robô no ambiente

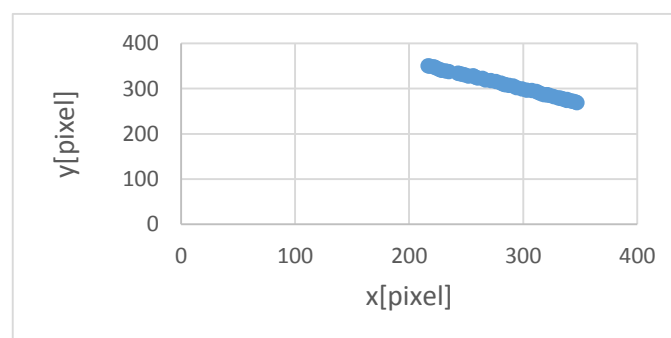


Figura 61. Localização x e y do robô na imagem

Na figura 62 são apresentadas as curvas de deslocamento do robô ao longo do eixo x no ambiente (em cm) e na imagem (em pixels).

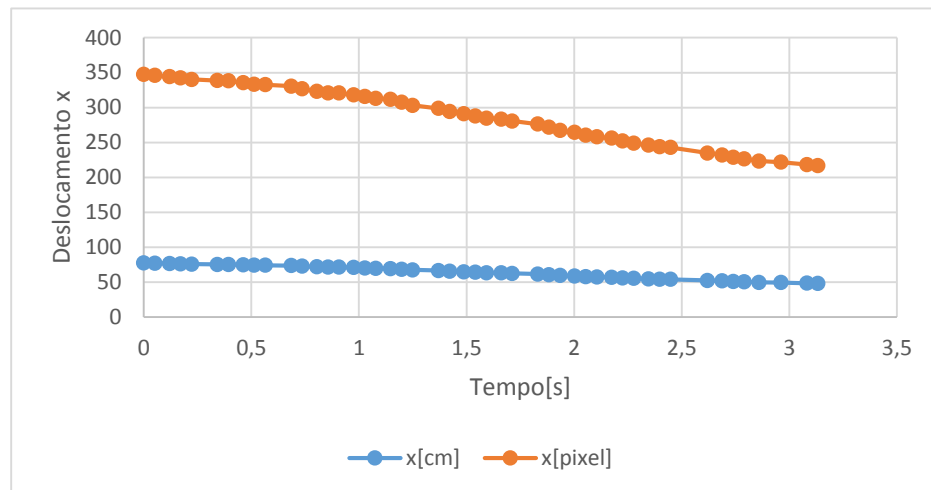


Figura 62. Variação do deslocamento do robô no eixo x

Na figura 63 é mostrado as curvas de deslocamento do robô ao longo do eixo y no ambiente (em cm) e na imagem (em pixels).

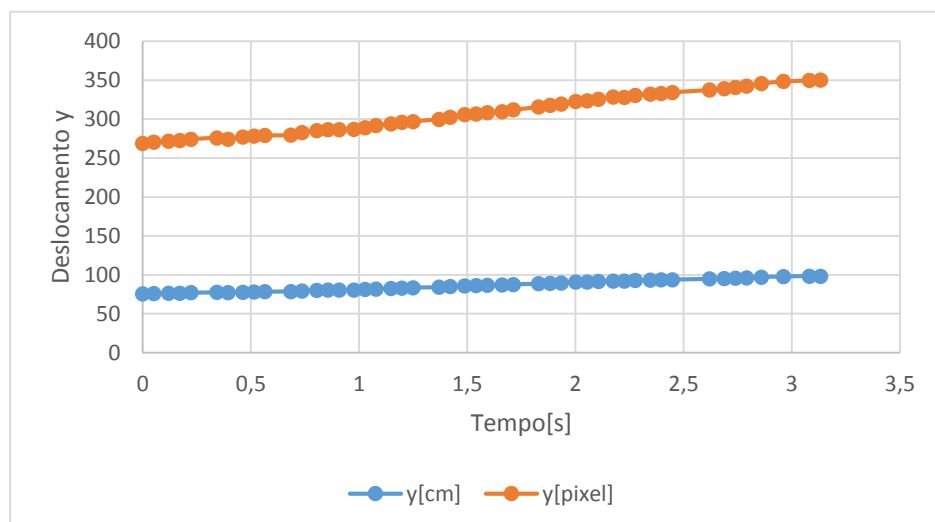


Figura 63. Deslocamento do robô ao longo do eixo y

Na figura 64 são mostrados as curvas das velocidades do robô ao longo do eixo x e y no ambiente. As velocidades médias são: $v_{med} = 36,94 \text{ cm/s}$, $v_{med x} = 1,62 \text{ cm/s}$ e $v_{med y} = 2,52 \text{ cm/s}$.

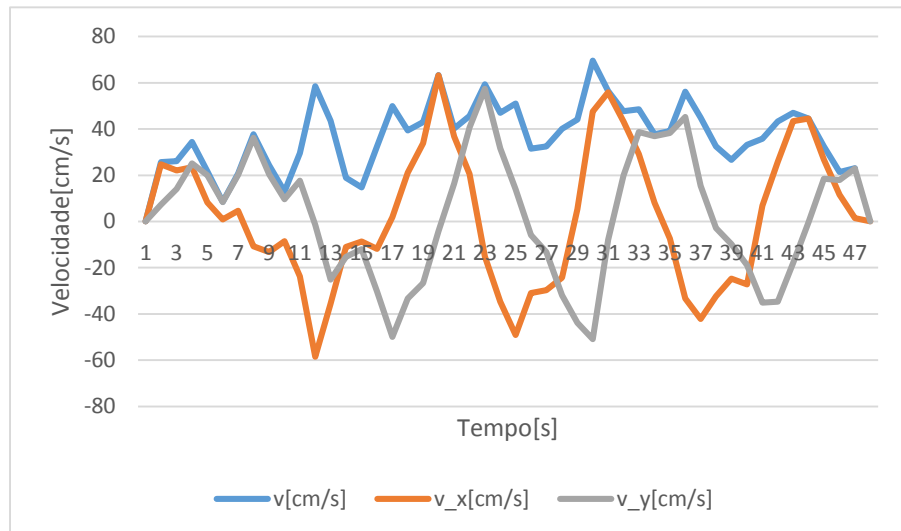


Figura 64. Velocidades do robô no ambiente

Na figura 65 são demonstrados as curvas das acelerações do robô ao longo do eixo x e y no ambiente.

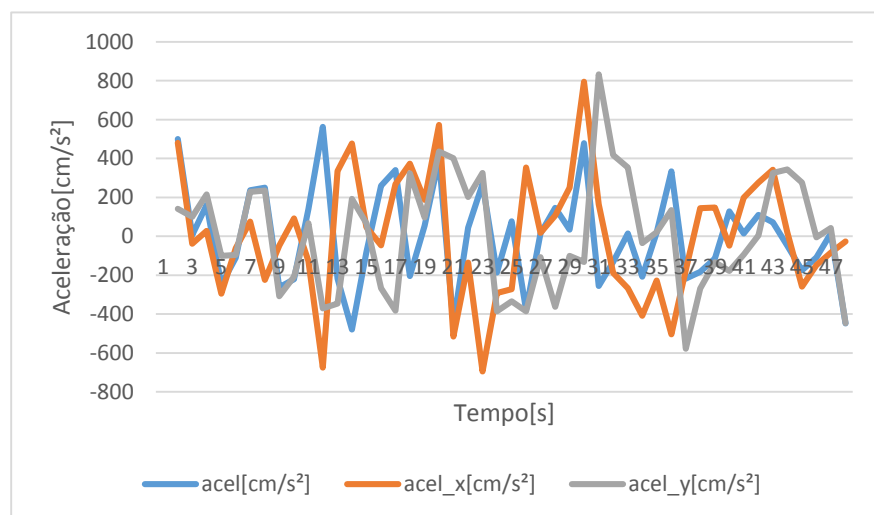


Figura 65. Acelerações do robô no ambiente

6 CONCLUSÃO

Esta dissertação propõe um sistema para o acionamento e medição de posição, velocidade e aceleração por meio de visão computacional em robôs móveis. A aplicação permite que o robô móvel da plataforma Scisoccer se locomova para pontos de interesse e no mesmo instante um sistema de medição visual passa a registrar suas coordenadas, velocidade e aceleração com base em sua movimentação no ambiente.

Sistemas de visão computacional vêm sendo amplamente utilizados em plataformas robóticas. Pode-se constatar que as bibliotecas são ótimas ferramentas para o desenvolvimento de projetos na área de robótica móvel e visão computacional. O algoritmo implementado apesar da sua simplicidade se mostrou eficaz, se apresentando como ferramenta interessante para sistemas de visão robótica.

De maneira geral, o trabalho conseguiu atingir os objetivos definidos nesta proposta. Os algoritmos implementados no acionamento e na medição através de imagens, apresentaram resultados satisfatórios, tornando-se uma ferramenta válida com capacidade real de aplicabilidade em situações que automatize o trabalho humano em suas tarefas domésticas, comerciais ou industriais, por exemplo, possibilitando a realização destas atividades com maior autonomia, precisão e eficiência, aumentando significativamente o bem-estar, a produtividade e a garantia da qualidade de produtos e serviços.

6.1 Trabalhos Futuros

Com o objetivo de obter resultados melhores, este trabalho gerou novas ideias que não puderam ser implementadas devido à falta de tempo hábil. Assim, são propostas como trabalhos futuros para tornar a metodologia mais completa:

- A primeira é um estudo das incertezas em cada fase. Poderiam ser mensuradas para que algoritmos específicos fossem executados e aplicados em cada etapa.
- Outra oportunidade é a quantificação dos erros existentes nas medições com a implementação de algoritmos para otimização desses erros.
- Outra concepção é a implementação de uma interface gráfica amigável que integre todo o sistema com a plotagem de gráficos.
- Outra ideia é a otimização do algoritmo implementado a partir da biblioteca OpenCV para implementação em sistemas de baixo custo computacional.
- As medições realizadas pelo sistema foram obtidas a partir de trajetórias retilíneas. Como sugestão, além da movimentação em linha reta, o robô poderia deslocar-se fazendo curvas e movimentos circulares.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, Beatriz Trinchão; MATOS, Leonardo Nogueira; FREIRE, Eduardo Oliveira. Um Sistema de Visão Computacional para Robôs Móveis. In: **XXVI Congresso da SBC**, Campo Grande, MS. 2006.

ARKIN, Ronald C. **Behavior- based robotics**. MIT press, 1998.

BAEDE, T. A. **Motion control of an omnidirectional mobile robot**. 2006.

BRADISK, Gary; KAEHLER, Adrian. **Learning OpenCV – Computer Vision WithOpenCV Library**. 1ªed. O´Reilly Media, 2008

BRAND, V. C. et al. Um Algoritmo Eficiente para Rastreamento de Objetos em Futebol de Robôs. **EVINCI UFPR**, Curitiba, PR, 2002.

BRAUNL, T. **Embedded robotics: mobile robot design and applications with embedded systems**. Berlin: New York: Springer, 2006.

BEKEY, G. A. **Autonomous robots: from biological inspiration to implementation and control**. Cambridge: MIT Press, 2005.

BENAVIDEZ, Patrick; JAMSHIDI, Mo. Mobile robot navigation and target tracking system. In: **System of Systems Engineering (SoSE), 2011 6th International Conference on**. IEEE, 2011. p. 299-304.

BORENSTEIN, Johann *et al.* **Mobile robot positioning-sensors and techniques**. NAVAL COMMAND CONTROL AND OCEAN SURVEILLANCE CENTER RDT AND E DIV SAN DIEGO CA, 1997.

BIANCHI, Andrea G. Campos; Barbosa, Luís F.; Calin, Gabriel; Deniponte, Juliana G.; Nascimento, Sandro do; Sánchez, Dámian R.; Junior, Carlos R. Mingoto; Bertone, Osvaldo Hugo. Scisoccer – Desenvolvimento de uma Plataforma de Robôs

Móveis voltada ao Futebol de Robôs. In: **XXVI Congresso da SBC**, Campo Grande, MS. 2006.

CONCI, A., AZEVEDO, E., LETA, F. R. **Computação Gráfica: Teoria e Prática – volume 2**. Ed. Elsevier. Rio de Janeiro, 2008.

COUTINHO, César Valentino Ribeiro. **Robótica Móvel–Sistema de Condução Autônoma**. 2014. Tese de Doutorado. INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA.

DUDEK, G.; JENKIN, M. (2010). **Computational Principles of Mobile Robotics**. Cambridge, MA: Cambridge University Press.

Diferença entre sensores CMOS e CCD. Disponível em: <<http://thegadgetsquare.com>>. Acesso em: 02 jun. 2015.

DE ARAÚJO, Sidnei Alves; LIBRANTZ, André Felipe Henriques. Navegação autônoma de robôs. **Exacta**, v. 4, n. Esp, p. 81-83, 2006.

Desafio de robótica. Disponível em: < <http://www.xbot.com.br/desafio/>>. Acesso em: 19 jun. 2015.

DE BEM, Rodrigo Andrade; COSTA, Anna Helena Realí. **Rastreamento visual de múltiplos objetos baseado em contornos aplicado ao futebol de robôs**. 2006.

DO NASCIMENTO, Rafaella Cristianne Alves. **Localização de Robôs Móveis em Ambientes Fechados em Tempo Real Utilizando Câmeras Montadas no Teto**. 2014. Tese de Doutorado. Universidade Federal do Rio Grande do Norte.

FELIZARDO, Ivanilza Felizardo; BRACARENSE, Alexandre Queiroz. *Processos de Soldagem - Processos Mecanizados e Automatizados. Apostila. 2013*. Disponível em: <http://ivanilzafe.dominiotemporario.com/soldagem_20.html>. Acessado em 23 de setembro de 2015.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**, 2000. Editora: Edgard Blücher LTDA.

GN, JOSÉ *et al.* **Estimação de Posição e Orientação para Robôs Móveis**.2008.

HIRANO, Laos Alexandre; **Trajetórias caóticas para o problema da exploração de terrenos com metas de busca ou vigilância usando robôs móveis**. In: Monografia – Graduação em Engenharia de Controle e Automação- Universidade Federal de Ouro Preto, Ouro Preto, 2006.

HELFRICK, A.D., COOPER, W.D. **Instrumentação Eletrônica Moderna e Técnicas de Medição**. Rio de Janeiro. Prentice Hall do Brasil, 1994.

Instrumentação e Técnicas de Medidas Conceitos. Apostila. 2013. Disponível em: <http://http://www.peb.ufrj.br/cursos/eel710/EEL710_Modulo01.pdf>. Acessado em 13 de Novembro de 2015.

KLASER, Rafael Luiz. **Navegação de veículos autônomos em ambientes externos não estruturados baseada em visão computacional**. 2014. Tese de Doutorado. Universidade de São Paulo.

LIU, Yong et al. Omni-directional mobile robot controller based on trajectory linearization. **Robotics and Autonomous Systems**, v. 56, n. 5, p. 461-479, 2008.

LINS BARRETO, S. et al. Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot. Mechatronics, **IEEE/ASME Transactions on**, v. 19, n. 2, p. 467-476, 2014.

MILANO, Danilo de; HONORATO, Luciano Barrozo. **Visão computacional**. Universidade Estadual de Campinas-(Unicamp), Faculdade de Tecnologia, 2010.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento Digital de Imagens**, Rio de Janeiro: Brasport, 1999.

MATIAS, Mariana C . Bernades. **Cinemática de Robôs: Tipos de rodas**. Notas de aula. Disponível em: <<http://pt.slideshare.net/FilipeRibeiro/aula-12-44676413?related=1>>. Acessado em 23 de Julho de 2015.

MATARIC, Maja J. **The Robotics Primer**. MIT Press, 2007.

MILESKI, Y.R et al. Calibração de sistemas ópticos para uso em visão computacional. **XXV salão de iniciação científica – UFRGS**, 2014.

NICOLAU, G. F. et. al. **Os Fundamentos da Física**, volume único – São Paulo, Moderna, 2007.

NAKHAEINIA, D. *et al.* A review of control architectures for autonomous navigation of mobile robots. **International Journal of the Physical Sciences**, v. 6, n. 2, p. 169-174, 2011.

OPENCV. Disponível em: < <http://opencv.org/>>. Acesso em: 20 Jun. 2015.

ORQUEDA, Omar AA; FIERRO, Rafael. Visual tracking of mobile robots in formation. In: **American Control Conference, 2007. ACC'07**. IEEE, 2007. p. 5940-5945.

ORTH, Alexandre et al. **Desenvolvimento de um sistema de visão para medir o desgaste de flanco de ferramentas de corte**. 2001.

PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.

Plataforma móvel omnidirecional Kuka youbot. Disponível em:< <http://www.youbot-store.com> >. Acesso em: 12 jul. 2015.

Pioneer 3Dx. Disponível em:< <https://www.cyberbotics.com/item?id=29>>. Acesso em: 01jun. 2015.

PENHARBEL, Eder A. et al. Filtro de imagem baseado em matriz RGB de cores-padrão para futebol de robôs. **Submetido ao I Encontro de Robótica Inteligente**, 2004.

PENA, Norberto *et al.* Instrumento livre para medidas de movimento. **Revista Brasileira de Ensino de Física**, v. 35, n. 3, p. 3505, 2013.

PINHEIRO, Alan Petrônio. **Desenvolvimento de um sistema de captura e análise de movimentos baseado em técnicas de visão computacional**. 2008.

ROMERO, Roseli Aparecida Francelin et (org). **Robótica Móvel**. LTC, 07/2014. Vital Book file.

Robots. Disponível em:< <http://www.deploy.virtual-labs.ac.in/>>. Acesso em: 01jun. 2015.

Robô Huskies. Disponível em: < <http://spectrum.ieee.org/automaton/robotics/diy>>. Acesso em: 02 jun. 2015.

Radar Doppler. Disponível em: < www.gmheng.com >. Acesso em: 02 jun. 2015.

Robocup. Disponível em: <<http://www.robocup.org>>. Acesso em: 12 jun. 2015.

SE, Stephen; LOWE, David G.; LITTLE, James J. Vision-based global localization and mapping for mobile robots. **Robotics, IEEE Transactions on**, v. 21, n. 3, p. 364-375, 2005.

SCARAMUZZA, Davide; FRAUNDORFER, Friedrich. *Visual odometry [tutorial]*. **Robotics & Automation Magazine, IEEE**, v. 18, n. 4, p. 80-92, 2011.

Sensores. Disponível em: <<http://loja.vidadesilicio.com.br/sensores-ct-5d3a4>>. Acesso em: 02 jun. 2015.

Sensores GPS. Disponível em: < www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf>. Acesso em: 02 jun. 2015.

Sensores Medição de Distancias. Disponível em: < produto.mercadolivre.com.br>. Acesso em: 02 jun. 2015.

SECCHI, H. A: **Uma introdução aos robôs móveis.** 2008.

SCHUEROFF, Joildo *et al.* Desenvolvimento de um Sistema de Visão Artificial para um Robô Explorador. **Anais do Computer on the Beach**, p. 214-222, 2014.

WOLF, Denis Fernando et al. Robótica móvel inteligente: Da simulação às aplicações no mundo real. In: **Minicurso: Jornada de Atualização em Informática (JAI), Congresso da SBC.** 2009.

WALDMANN, Jacques; CERÁVOLO, Carlos EP. SIMNAVIN: avaliação do impacto da qualidade de sensores inerciais montados em plataforma solidária sobre os erros de navegação. In: **CONGRESSO BRASILEIRO DE AUTOMÁTICA.** 2000.

Xbot. Disponível em:< <http://xbot.com.br/externo/scisoccer.php>>. Acesso em: 09 Nov. 2015.