

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ENGENHARIA DE COMPUTAÇÃO
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

LEANDRO VINICIUS SILVA FORNECK
MAIKO MIN IAN LIE
RAFAEL BARCELLOS LEMOS DOS SANTOS

**PLATAFORMA PARA DESENVOLVIMENTO DE TELERROBÓTICA
ANALÍTICA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2016

LEANDRO VINICIUS SILVA FORNECK
MAIKO MIN IAN LIE
RAFAEL BARCELLOS LEMOS DOS SANTOS

PLATAFORMA PARA DESENVOLVIMENTO DE TELERROBÓTICA ANALÍTICA

Trabalho de Conclusão do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná – Câmpus Curitiba, apresentado como requisito parcial para a obtenção do grau de Engenheiro de Computação.

Orientador:

Prof. Dr. Hugo Vieira Neto

CURITIBA

2016

LEANDRO VINICIUS SILVA FORNECK
MAIKO MIN IAN LIE
RAFAEL BARCELLOS LEMOS DOS SANTOS

**PLATAFORMA PARA DESENVOLVIMENTO DE TELERROBÓTICA
ANALÍTICA**

Este trabalho de conclusão de curso foi julgado e aprovado como requisito parcial para obtenção do grau de Engenheiro de Computação pela Universidade Tecnológica Federal do Paraná.

Curitiba, 4 de abril de 2016.

Prof. Dr. Hugo Vieira Neto

Prof. Orientador

Prof. Dr. Gustavo Benvenuto Borba

Prof. Dr. Ricardo Lüders

AGRADECIMENTOS

Ao prof. Gustavo Benvenuti Borba, pela tolerância em relação à minha ausência do LABIEM durante o desenvolvimento deste trabalho. A Fernando Padilha, Bruno Meneguele e Vinicius Arcanjo, pelo projeto de robótica móvel que me inspirou a arriscar na área. A Samuel Pelegrinello, pelo apoio sem o qual a minha graduação teria sido truncada já nos primeiros semestres e minha participação neste trabalho não seria possível.

M. M. I. Lie

A Leandro Schmitz, pelo auxílio no projeto de hardware; Daniel Matias, pela soldagem do protótipo da placa de sensoriamento de corrente; Tiago Silva pelo auxílio na interpretação dos dados de giroscópio e acelerômetro; Gerson Soek, pelo corte do protótipo da placa de locomoção.

R. B. L. dos Santos

Ao orientador, prof. Hugo Vieira Neto, pelo respeito, paciência, apoio e — não menos importante — inspiração, através do seu próprio trabalho, cuja influência esperamos que esteja aparente nos pontos fortes deste trabalho.

À Universidade Tecnológica Federal do Paraná, pelo auxílio financeiro através da Bolsa de Apoio a Trabalhos de Conclusão de Curso (Edital 013/2015 — PROGRAD), e pela excelente formação que tornou este trabalho possível.

L. S. Forneck

M. M. I. Lie

R. B. L. dos Santos

What the old man did might have been alright for a university but he had no idea at all about industrial production. (...) It was young Rossum who had the idea of making robots that would be a living and intelligent workforce. (...) The old atheist had no idea about industry and commerce, and the young man ended up shutting him up in his laboratory where he could play around with his great failures while he got on with the real job himself in a proper scientific way.

— R.U.R., Karel Čapek

O que o velho fez pode ter sido razoável para uma universidade mas ele não tinha a mínima ideia sobre produção industrial. (...) Foi o jovem Rossum quem teve a ideia de construir robôs que seriam uma força de trabalho viva e inteligente. (...) O velho ateu não tinha a mínima ideia sobre indústria e comércio, e o jovem rapaz acabou por trancafiá-lo no seu laboratório onde poderia brincar com seus grandes fracassos enquanto ele prosseguia com o verdadeiro trabalho por si mesmo de maneira adequadamente científica.

— R.U.R., Karel Čapek

RESUMO

FORNECK, Leandro Vinicius Silva. LIE, Maiko Min Ian. SANTOS, Rafael Barcellos Lemos dos. Plataforma para desenvolvimento de telerrobótica analítica. 2016. 67 f. Trabalho de conclusão de curso (Graduação em Engenharia de Computação) – Departamento Acadêmico de Informática, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

A robótica móvel tem impressionado pela sofisticação dos seus resultados em diversas aplicações nas últimas décadas, seja na automação industrial, indústria militar, exploração espacial ou entretenimento. O aumento gradativo do acesso a plataformas de desenvolvimento embarcado, sensores, atuadores, e outros dispositivos eletrônicos, sugere que a tendência é de resultados ainda mais impressionantes nos próximos anos. Entretanto, constata-se que ainda existe grande dependência na abordagem de tentativa e erro no desenvolvimento de robôs móveis. Esta abordagem leva a sistemas pouco robustos, que caracterizam apenas uma prova de existência da funcionalidade do robô em determinadas condições. A Robótica Analítica, isto é, a aplicação do método científico no desenvolvimento de sistemas de robótica, sugere métodos para a melhor compreensão e modelagem do comportamento de robôs e sua interação com o seu ambiente. Considerando isso, este trabalho descreve o desenvolvimento de uma plataforma de telerrobótica móvel, baseada em um robô hexápode, adequada ao desenvolvimento da Robótica Analítica. Isso é atingido por meio de duas funcionalidades principais: a descrição quantitativa do estado do robô e a possibilidade de reprodução de experimentos com o mesmo. A primeira é implementada pelo monitoramento de sensores adequados (i.e. corrente consumida pelos servomotores, distância em quatro direções, velocidade angular, aceleração e visão), a segunda pela operação remota e simultânea, que permite a reprodução de condições experimentais pela operação de um mesmo robô em um mesmo ambiente. É apresentada também uma série de experimentos simples de validação do sistema.

Palavras-chave: Robótica móvel. Robótica Analítica. Telerrobótica. Robô hexápode.

ABSTRACT

FORNECK, Leandro Vinicius Silva. LIE, Maiko Min Ian. SANTOS, Rafael Barcellos Lemos dos. Platform for analytical telerobotics development. 2016. 67 f. Trabalho de conclusão de curso (Graduação em Engenharia de Computação) – Departamento Acadêmico de Informática, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Mobile robotics has impressed with the sophistication of its results in many applications over the last decades, be them in industrial automation, military industry, space exploration, or entertainment. The gradative increase in the access to embedded development platforms, sensors, actuators, and other electronic devices, suggests a tendency to even more impressive results in the next years. However, there is still a great dependency on the trial-and-error approach in the development of mobile robots. This approach leads to less robust systems, which characterize only a proof-of-existence of the robot's functionality under certain circumstances. Analytical Robotics, that is, the application of the scientific method in the development of robotic systems, suggests methods for better understanding and modelling of robots and their interaction with their environment. Considering this, the present work describes the development of a mobile telerobotics platform, based on a hexapod robot, adequate for the development of Analytical Robotics. This is attained through two main functionalities: the quantitative description of the robot's state and the reproducibility of experiments with it. The former is implemented through the monitoring of adequate sensors (i.e. current consumed by the motors, distance in four directions, angular velocity, acceleration and vision), the latter through the remote and simultaneous operation of the robot, which allows the reproduction of experimental conditions through the operation of the same robot on the same environment. A series of simple validation experiments is also presented.

Keywords: Mobile robotics. Analytical robotics. Telerobotics. Hexapod robot.

LISTA DE FIGURAS

1	Paradigmas de controle de robôs.	17
2	Formas de locomoção terrestre mais comuns em robôs móveis.	18
3	<i>Strandbees</i> de Theo Jansen.	18
4	Sistema de telerrobótica.	21
5	Arquitetura do sistema.	24
6	Arquitetura do <i>middleware</i>	25
7	Formato de mensagens adotada no sistema.	26
8	Arquitetura do software embarcado.	28
9	Visualização 3D do modelo mecânico simulado.	30
10	Torque nas <i>junções 1 e 2</i> das <i>pernas 2 e 3</i>	31
11	Torque na <i>junção 0</i> de uma perna do robô.	31
12	Modelo 3D da estrutura física do robô — Perspectiva isométrica	32
13	Modelo 3D da estrutura física do robô — Vistas superior e inferior.	33
14	Estrutura do sistema integrado.	34
15	Descrição geral da operação do sistema.	35
16	Experimento de medição de distâncias. Vista superior.	39
17	Medições de distância.	39
18	Experimento de medição de aceleração vertical. Vista lateral.	40
19	Medições de aceleração.	41
20	Experimento de medição da orientação. Vista lateral.	42
21	Medição de correntes nos servomotores.	43
22	Experimento de acesso à visão — detecção de objeto.	45
23	RTT pela rede local e pela internet.	46
24	<i>Layout</i> do <i>Módulo de Locomoção</i> — <i>Junções (links)</i> para uma perna.	53
25	<i>Layout</i> do <i>Módulo de Locomoção</i> — Tronco.	54
26	<i>Layout</i> do <i>Módulo de Sensoriamento e Interfaceamento</i>	55
27	<i>Layout</i> do <i>Módulo de Comunicação e Visão</i>	56
28	Diagrama esquemático — Placa de sensoriamento.	57
29	<i>Layout</i> da placa de circuito impresso (face superior) — Placa de sensoriamento.	57
30	<i>Layout</i> da placa de circuito impresso (face inferior) — Placa de sensoriamento.	58
31	Placa de sensoriamento.	58

32	Diagrama esquemático – Placa de locomoção.	59
33	Diagrama esquemático – Placa de locomoção (continuação).	60
34	<i>Layout</i> da placa de circuito impresso (face superior) – Placa de locomoção. . .	61
35	<i>Layout</i> da placa de circuito impresso (face inferior) – Placa de locomoção. . .	62
36	Placa de locomoção.	63

LISTA DE TABELAS

1	Requisitos funcionais.	22
2	Requisitos não-funcionais.	23
3	Protocolo de comunicação — Mensagens de comando.	28
4	Protocolo de comunicação — Mensagens de resposta.	29
5	Requisitos funcionais, suas descrições e lista experimentos que os validam. . .	38
6	Inclinação real e inclinação estimada pelo giroscópio.	42

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVO GERAL	13
1.2	OBJETIVOS ESPECÍFICOS	13
1.3	JUSTIFICATIVA	14
1.4	TRABALHOS RELACIONADOS	14
1.5	ORGANIZAÇÃO DO TRABALHO	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	ROBÓTICA MÓVEL	16
2.2	ROBÓTICA ANALÍTICA	19
2.3	MIDDLEWARE PARA ROBÓTICA	19
2.4	TELERROBÓTICA	20
3	O SISTEMA DE TELERROBÓTICA	22
3.1	ESPECIFICAÇÃO DE REQUISITOS	22
3.2	ARQUITETURA	23
3.2.1	Módulo de Comunicação e Visão	24
3.2.2	Módulo de Sensoriamento e Interfaceamento	27
3.2.3	Módulo de Locomoção	29
3.2.4	Estrutura Física	32
3.3	O SISTEMA INTEGRADO	34
3.3.1	Comando de atuadores	36
3.3.2	Monitoramento de sensores	36
3.3.3	Acesso à visão	37
4	EXPERIMENTOS E DISCUSSÃO	38
4.1	EXPERIMENTO 1 – SENSORIAMENTO DE DISTÂNCIA	38
4.2	EXPERIMENTO 2 – SENSORIAMENTO DE ACELERAÇÃO	40
4.3	EXPERIMENTO 3 – SENSORIAMENTO DE ORIENTAÇÃO	41
4.4	EXPERIMENTO 4 – SENSORIAMENTO DE CORRENTE	42
4.5	EXPERIMENTO 5 – ACESSO À VISÃO	44
4.6	DISCUSSÃO GERAL	45

	12
5 CONCLUSÕES	47
A LAYOUT DA ESTRUTURA FÍSICA	53
B PLACAS - DIAGRAMA ESQUEMÁTICO, <i>LAYOUT</i> E FOTOS	57
C <i>SCRIPT</i> DE DETECÇÃO DE OBJETO	64
D DATASHEET	65

1 INTRODUÇÃO

Desde os seus primeiros resultados notáveis na década de 1950 (WALTER, 1950), a robótica móvel se tornou uma importante área de pesquisa, expandindo nichos que vão desde entretenimento (SONY, 1999) à exploração espacial (NASA, 2015). Entretanto, grande parte do projeto de robôs móveis na atualidade, seja na indústria ou na academia, ainda depende da abordagem de tentativa e erro (NEHMZOW, 2009). A necessidade de processos melhor fundamentados e mais precisos no projeto e construção de robôs móveis tem dado maior visibilidade a abordagens como a Robótica Analítica (NEHMZOW, 2009) onde a ênfase é na aplicação de métodos científicos de representação, análise e previsão — em detrimento de tentativa e erro.

Neste trabalho, é apresentado o desenvolvimento uma plataforma para telerrobótica móvel, com ênfase na descrição quantitativa do estado do robô, assim como na reprodutibilidade de experimentos com o mesmo, com o fim de oferecer uma ferramenta adequada ao desenvolvimento da Robótica Analítica. A descrição quantitativa se dá por meio do monitoramento de sensores, cujos dados podem ser acessados através de trocas de mensagens. A reprodutibilidade de experimentos, por sua vez, é facilitada através da teleoperação do robô, que sendo um só e mantido no mesmo ambiente, permite manter as condições experimentais relativamente constantes.

1.1 OBJETIVO GERAL

Desenvolvimento de uma plataforma para telerrobótica móvel baseada em robô hexápode, operável remotamente e com as seguintes modalidades de sensoriamento: corrente consumida por seus motores, distância em quatro direções, velocidade angular, aceleração e visão.

1.2 OBJETIVOS ESPECÍFICOS

- Projetar e construir a estrutura física do robô, que consiste no seu chassi e sua integração mecânica/eletrônica;
- Desenvolver o *middleware* que permite e abstrai o acesso distribuído aos sensores e atuadores do robô;
- Desenvolver os *device drivers* para interfaceamento com os sensores e atuadores;
- Validar e experimentar a plataforma desenvolvida.

1.3 JUSTIFICATIVA

A construção de robôs com base em avaliação meramente qualitativa e subsequente adaptação iterativa tendem a resultar em robôs pouco robustos, visto que constituem apenas uma prova de existência (NEHMZOW, 2009), ou seja, sabe-se que o robô se comporta de forma adequada no ambiente onde foi testado, mas não se sabe o suficiente para prever como se comportará em um ambiente diferente deste. Infelizmente, tal abordagem ainda é comumente utilizada. Para a construção de robôs mais robustos e previsíveis, é necessária a análise e descrição quantitativas do seu comportamento. Isso é possível por meio do monitoramento de sensores adequados.

A compreensão do comportamento de robôs móveis se dá por meio de experimentação. Desta forma, mostra-se desejável a possibilidade de reprodução de experimentos. Consideramos que a reprodutibilidade de experimentos é facilitada pela constância das condições experimentais, no caso, do robô e do ambiente. Apesar de não oferecer condições perfeitamente constantes, a teleoperação do robô oferece uma aproximação adequada, visto que permite que diferentes experimentos sejam feitos com o mesmo robô, no mesmo ambiente, mesmo que os operadores estejam experimentando a partir de locais distintos. Tais circunstâncias motivam o desenvolvimento de uma plataforma de telerrobótica para o controle e monitoração remotos de um robô móvel.

A aplicabilidade de um sistema de telerrobótica móvel vai além do âmbito da pesquisa científica, sendo útil também no ensino de robótica móvel, como mostra uma experiência recente na Universidade Técnica Tcheca (*Czech Technical University*), desenvolvida por Kulich et al. (2013). Consequentemente, tal sistema também encontra aplicabilidade em áreas contidas ou periféricas à robótica móvel, como sistemas embarcados, visão computacional e teoria de controle.

1.4 TRABALHOS RELACIONADOS

Atualmente existem diversos kits de robótica móvel disponíveis no mercado. No nicho de robôs hexápodes, existem alternativas com focos que vão desde o entretenimento de baixo custo (e.g. (ARCBOTICS, 2015)), a componentes de precisão com ênfase em pesquisa (e.g. (TROSSENROBOTICS, 2015)). Já entre os robôs com locomoção por rodas, as alternativas mais notáveis são o Khepera da K-team (K-TEAM, 2015) e o Roomba da iRobot (IROBOT, 2015). O Khepera teve origem em 1991, na Escola Politécnica Federal de Lausanne (*École Polytechnique Fédérale de Lausanne - EPFL*), na Suíça, onde já tinha ênfase em suporte para pesquisa (MONDADA;

FRANZI; GUIGNARD, 1999). O Roomba é um robô aspirador de pó, desenvolvido como um eletrodoméstico para auxílio a limpeza doméstica (IROBOT, 2015). Entretanto, devido à sua modularidade, relativo baixo custo, disponibilidade de sensores, entre outras características, o Roomba tem sido adaptado para uso em ensino e pesquisa envolvendo robótica móvel (MATARIC; KOENIG; FEIL-SEIFER, 2007; TRIBELHORN; DODDS, 2007).

Em relação a *middleware* para robótica, atualmente existe uma série de opções em uso, expandindo diversos escopos, alguns focando em aspectos de alto nível (e.g. Pyro), outros percorrendo desde os níveis de abstração mais baixos aos mais altos (e.g. MIRO, CLARAty) (ELKADY; SOBH, 2012). Apesar de não existir um *middleware* padrão bem estabelecido para robótica, o ROS (*Robot Operating System*) (QUIGLEY et al., 2009) tem emergido como uma das opções mais atraentes, mostrando-se a maior tentativa recente de se criar uma comunidade de robótica baseada em compartilhamento de código e desenvolvimento incremental (CRICK et al., 2011).

1.5 ORGANIZAÇÃO DO TRABALHO

No Capítulo 2 são apresentados o contexto histórico, questões notáveis, e paradigmas clássicos na robótica móvel. Também é apresentado um conceito simples, mas essencial à proposta deste trabalho — a Robótica Analítica. Por fim é apresentada a abordagem de operação de robôs que adotamos neste trabalho, a telerrobótica. No Capítulo 3 o sistema desenvolvido é apresentado. Primeiramente, descreve-se a arquitetura geral do sistema, em seguida cada um dos módulos que compõem a arquitetura são analisados internamente e explicados em mais detalhes. Por fim, apresenta-se o sistema resultante da integração destes módulos e as operações básicas para a sua operação. No Capítulo 4, é feita uma série de experimentos simples para validar as funções básicas do sistema. Também são feitos experimentos de aplicações simples envolvendo sensoriamento e atuação com o fim de apresentar uma prova de conceito da aplicabilidade do sistema. Concluindo, o Capítulo 5 apresenta um fechamento da discussão, mencionando aspectos do sistema passíveis de aprimoramentos futuros.

Os diagramas esquemáticos, *layouts* de placa de circuito impresso e *layouts* de cortes em acrílico utilizados no projeto são apresentados nos Apêndices A e B. O Apêndice C apresenta o *script* de detecção de objetos utilizado nos experimentos de validação e o Apêndice D apresenta um *datasheet* resumindo as características e especificações da plataforma desenvolvida.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ROBÓTICA MÓVEL

A robótica móvel tem se mostrado uma notável área de pesquisa pelo menos desde 1950, com os robôs autônomos analógicos de William Grey Walter, os *Machinae Speculatrixes* (WALTER, 1950). As questões tratadas na pesquisa com robôs móveis distinguem-se das tratadas na robótica de manipuladores, mais notavelmente, devido à sua ênfase nos aspectos de locomoção, navegação e — no caso de robôs móveis autônomos — aprendizado (NEHMZOW, 2006). Pesquisas em robótica móvel têm rendido resultados impressionantes nos últimos anos, em grande parte devido a sucessos nas indústrias militar (BOSTON DYNAMICS, 2015) e espacial (NASA, 2015). O potencial da pesquisa em robótica móvel se dá por conta dos inúmeros outros temas que envolve, como visão computacional, interpretação e integração de sensores, modelagem de ambientes reais, navegação, planejamento e execução de tarefas (IYENGAR; ELFES, 1991).

O desenvolvimento e a aplicação de robôs móveis geralmente se resume às questões de controle, navegação e locomoção:

Controle. Reduzida à sua ideia mais básica, o controle de um robô consiste na associação da percepção de sensores à ação de atuadores. Existem dois grandes paradigmas para tal na robótica: o funcional e o de subordinação (Figura 1) (NEHMZOW, 2009). No paradigma funcional, o robô processa continuamente um ciclo de percepção, decisão e ação — de forma inerentemente sequencial (NEHMZOW, 2009). No paradigma de subordinação, comportamentos complexos emergem de comportamentos simples, simultâneos e desacoplados entre si. Esta abordagem de composição apresenta robustez superior à abordagem funcional, tanto em termos de modularidade, conseqüente da minimização de interdependência entre os blocos comportamentais, quanto pela possibilidade de aprimoramento do comportamento através do simples acréscimo de novos blocos comportamentais (BROOKS, 1986). A desvantagem do paradigma de subordinação é a dificuldade de execução de planos, visto que a sua operação baseia-se na resposta direta à estímulos. Não é conhecida nenhuma forma simples de representação de planos neste paradigma (NEHMZOW, 2009).

Navegação. A navegação de robôs móveis pode ser descrita em termos de três competências: autolocalização — a capacidade do robô de se localizar em relação a uma

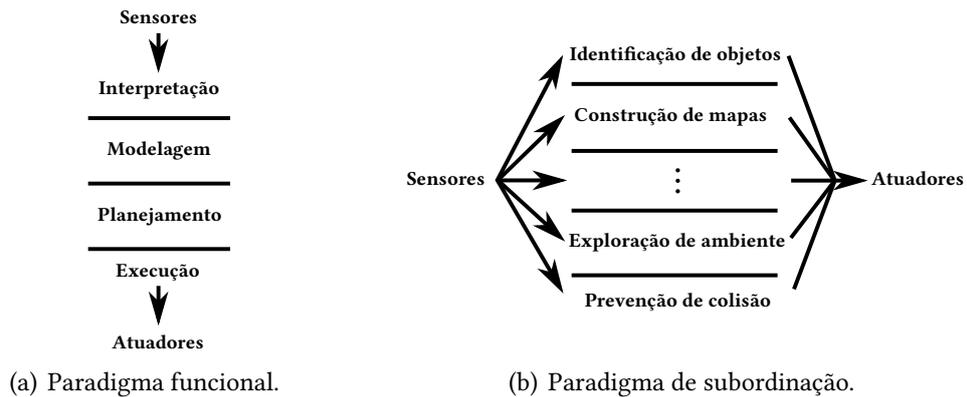


Figura 1: Paradigmas de controle de robôs.

Fonte: Adaptado de (BROOKS, 1986; JONES; FLYNN, 1993).

referência, planejamento de trajeto — a determinação de uma sequência de posições até um destino, e utilização de mapas — a representação e interpretação de locais em relação a uma referência (NEHMZOW, 2009).

Um sistema de navegação pode ser baseado em propriocepção, a utilização de medidas internas (e.g. integração da velocidade proveniente de um codificador (*encoder*) de eixo) para se localizar em relação a uma referência, ou em exterocepção, baseando-se em marcações no mundo real (e.g. fita magnética no ambiente). A primeira abordagem, diferentemente da segunda, está sujeita a erros cumulativos, provindos de deslocamentos que a percepção do robô é incapaz de contabilizar adequadamente (NEHMZOW, 2009).

Locomoção. Robôs móveis terrestres podem se locomover, geralmente, por meio de rodas, esteiras, ou pernas. A locomoção por rodas é a mais comum, devido à sua simplicidade e maior eficiência energética, mas mostra-se inadequada em terrenos irregulares — sendo sujeita a atolamento ao percorrer irregularidades com altura maior que o raio das suas rodas (Figura 2(a)) (JONES; FLYNN, 1993). A incorporação de esteiras mitiga este problema (Figura 2(b)), mas é energeticamente ineficiente devido à energia dissipada na fricção tanto das rodas com a esteira quanto das esteiras com o chão na derrapagem característica do seu movimento ao percorrer curvas (JONES; FLYNN, 1993). A locomoção por pernas apresenta duas grandes vantagens em relação às abordagens anteriores. A primeira é a sua mobilidade robusta, que permite locomoção por terrenos irregulares (Figura 2(c)). A segunda é a possibilidade de desacoplamento entre o trajeto do tronco e o das pernas — isto é, a carga pode se mover suavemente apesar da irregularidade do terreno (RAIBERT, 1986).

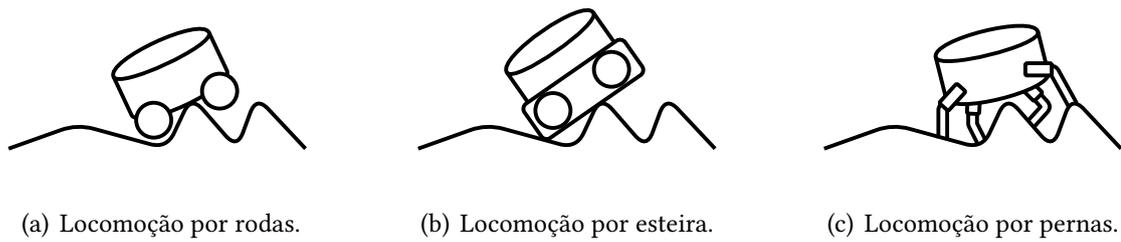


Figura 2: Formas de locomoção terrestre mais comuns em robôs móveis.

Na locomoção por pernas, a programabilidade das suas articulações é uma característica importante para garantir a robustez da sua marcha. Em um mecanismo baseado em articulações não programáveis, a marcha é fixa e definida pelo formato da estrutura mecânica. Um exemplo notável deste caso são os *Strandbeests* de Theo Jansen (JANSEN, 2016), que são estruturas móveis com locomoção por pernas onde as articulações são movimentadas pelas conexões que apresentam entre si, não sendo programáveis individualmente. Apenas uma destas articulações é programável, a que origina o movimento que é propagado para as outras articulações (Figura 3) – semelhante a um conjunto de engrenagens. Nesta abordagem a complexidade da marcha é delegada à estrutura mecânica (e.g. formato, proporções), sendo determinada na sua construção. Desta forma, o mecanismo é pouco robusto mas o seu controle é simples. Um mecanismo de locomoção com articulações independentemente programáveis apresenta robustez superior, tendo marcha adaptável por software, porém, acarreta custo de implementação e complexidade de controle significativamente maiores.

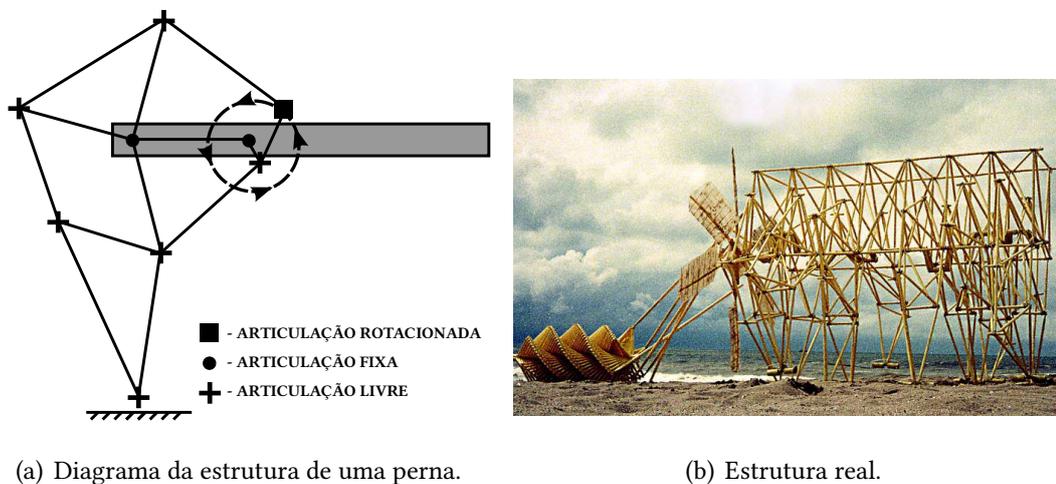


Figura 3: *Strandbeest* de Theo Jansen.
Fonte: Adaptados de (JANSEN, 2016).

2.2 ROBÓTICA ANALÍTICA

A importância da quantificação na modelagem, síntese e análise de fenômenos ou mecanismos é algo bem estabelecido há séculos na comunidade científica. A quantificação de dados relacionados a um problema permite a sua formalização na linguagem matemática, conveniente pela sua expressividade e imenso repertório de métodos para as mais diversas análises. A linguagem matemática permite não somente a expressão de leis da natureza, como também a descrição de mecanismos capazes de executar as mais complexas tarefas a partir do poder previsivo destas leis. As limitações e o potencial da expressividade possibilitada pela notação matemática foram concisamente descritas em 1960, por Eugene P. Wigner, em “*The Unreasonable Effectiveness of Mathematics in the Natural Sciences*” (WIGNER, 1960):

“(...) as leis da natureza podem ser utilizadas para prever eventos futuros somente sob circunstâncias excepcionais — quando todos os determinantes relevantes do presente estado do mundo são conhecidos. É de acordo com isso que a construção de máquinas, cujo funcionamento ele pode prever, constitui a mais espetacular conquista do físico. Nessas máquinas, o físico cria a situação na qual todas as coordenadas relevantes são conhecidas tal que o comportamento da máquina pode ser previsto.”

É em tal linha de pensamento que pesquisadores na área de robótica móvel, insatisfeitos com a abordagem de tentativa e erro da prática atual, têm promovido métodos melhor fundamentados para o projeto, implementação e análise de robôs móveis. Nehmzow (2006) chamou de “Robótica Analítica” a abordagem que enfatiza a aplicação de métodos científicos na representação, análise e previsão do comportamento de robôs móveis. É deste contexto que buscamos as características norteadoras do presente trabalho: descrição quantitativa e reprodutibilidade experimental na prática de pesquisa e desenvolvimento de robôs móveis.

2.3 MIDDLEWARE PARA ROBÓTICA

Um *middleware* para robótica consiste em uma camada de abstração cujo intuito é mediar o sistema operacional e as suas aplicações, oferecendo modularidade, abstração de *hardware* e portabilidade (ELKADY; SOBH, 2012). Existem diversas variedades de *middleware* para robótica, com focos que vão desde requisitos de tempo real (e.g. OroCoS) a planejamento simbólico de ações (e.g. SmartSoft) (ELKADY; SOBH, 2012). Apesar dos recentes avanços na robótica, não há um *middleware* padrão e bem estabelecido para esta, mesmo que, atualmente, seja possível

citar alguns mais proeminentes (e.g. ROS). Na realidade, é possível que nunca haja tal *middleware* comum para robótica, devido a uma série de motivos, mais notavelmente a acentuada heterogeneidade nesta área (SMART, 2007).

Observando as características de diversos sistemas de *middleware* para robótica, nota-se que — apesar das diferenças em termos de paradigmas, módulos, arquitetura, etc. — a contribuição destes consiste basicamente no desacoplamento entre tarefas de alto nível (e.g. navegação), módulos de baixo nível (e.g. controle de motores) e a comunicação entre estes. Alguns autores consideram que um *middleware* para robótica móvel deve incluir, além da infraestrutura básica de comunicação entre componentes, comportamentos básicos como desvio de obstáculos e até mesmo SLAM (*Simultaneous Localization and Mapping*) (SMART, 2007). Outros autores incentivam a abordagem oposta, enfatizando o desenvolvimento de algoritmos de robótica em bibliotecas separadas, para diminuir o acoplamento com o *middleware*, visto que isso dificulta a reutilização de código (QUIGLEY et al., 2009). Esta última abordagem tem encontrado recente sucesso no ROS (*Robot Operating System*) (QUIGLEY et al., 2009) que, apesar do nome, é um *middleware* e não um sistema operacional.

Uma abordagem emergente em *middleware* para robótica tem sido a implementação da infraestrutura de comunicação a partir de troca de mensagens no modelo *publish/subscribe*, no qual fornecedores publicam dados e consumidores assinam¹ tópicos de interesse, recebendo dados referentes aos tópicos somente e assim que forem publicados. Os publicadores e assinantes não sabem da existência uns dos outros — isto é, são desacoplados (MATTEUCCI, 2003). As características deste modelo encaixam-se bem no contexto da robótica, visto que neste é comum: existirem múltiplos produtores e consumidores de dados, a rede de produtores e consumidores ser dinâmica, e o comportamento dos componentes depender de eventos assíncronos (MATTEUCCI, 2003). A ampla implementação desta abordagem em *middleware* para robótica móvel (ELKADY; SOBH, 2012) sugere fortemente que esta é uma boa opção técnica.

2.4 TELERROBÓTICA

A telerrobótica tem origem no conceito de teleoperador, que é uma máquina que permite que um operador humano manipule e monitore objetos à distância (SHERIDAN, 1995). Um telerrobô é um caso específico de teleoperador, onde a máquina — neste caso o robô — executa tarefas utilizando os seus próprios mecanismos de inteligência e sensoriamento, mas baseando-se

¹Do verbo inglês *subscribe*. Não confundir com *sign*.

em informações sobre objetivos, restrições e planos, enviados por um supervisor humano, que por sua vez recebe informações sobre a atuação do robô (SHERIDAN, 1989). A Figura 4 mostra um sistema de telerrobótica, onde um operador humano acessa remotamente, a partir de um computador local, os sensores e atuadores disponíveis em um telerrobô. O robô possui o seu próprio computador, responsável por traduzir as instruções recebidas em operações concretas.

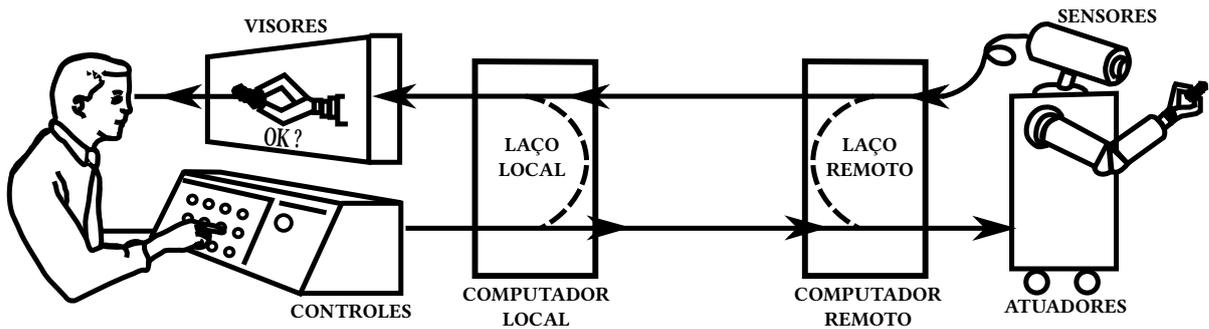


Figura 4: Sistema de telerrobótica.
Fonte: Adaptado de (SHERIDAN, 1989)

Consideramos que a telerrobótica é uma abordagem adequada para a prática da robótica analítica, por oferecer monitoração quantitativa do estado do robô, além de permitir o uso colaborativo do mesmo robô e ambiente, o que viabiliza a reprodução de experimentos. A aplicabilidade de uma plataforma de telerrobótica não se restringe apenas à pesquisa, mas também ao ensino, especialmente à distância, para o qual laboratórios remotos de robótica móvel já foram implantados e aplicados com considerável sucesso (KULICH et al., 2013). Desta forma, este trabalho foca na incorporação de recursos que permitam a utilização da plataforma de robótica móvel proposta na prática de telerrobótica e, conseqüentemente, de Robótica Analítica.

3 O SISTEMA DE TELERROBÓTICA

3.1 ESPECIFICAÇÃO DE REQUISITOS

Antes da descrição do sistema desenvolvido, são apresentados a seguir os requisitos adotados durante o seu projeto, bem como as suas justificativas. A Tabela 1 apresenta os requisitos funcionais:

Tabela 1: Requisitos funcionais.

Nome	Descrição
RF1	O robô deverá fazer sensoriamento de distância, aceleração e orientação.
RF2	O robô deverá fazer sensoriamento das correntes consumidas por cada servomotor.
RF3	Os sensores e atuadores do robô deverão ser acessíveis por meio de troca de mensagens com protocolo bem definido.
RF4	O robô deverá possuir visão por câmera.
RF5	O robô deverá se locomover por meio de pernas.
RF6	O robô deverá ser acessível por meio de conexão a rede Wi-Fi.

Os requisitos RF1 e RF2 são diretamente relacionados à capacidade de monitoramento do estado do robô, sendo necessários para atender à proposta de possibilitar a descrição quantitativa do comportamento do mesmo. RF3 é uma abordagem testada e amplamente adotada em sistemas de robótica (ELKADY; SOBH, 2012), sendo uma opção de considerável baixo risco. Devido a essa ampla adoção, tal abordagem também evita a alienação do sistema proposto em relação aos já existentes. RF4 foi incluída pelo fato da visão ser a forma de sensoriamento mais versátil na robótica (FU; GONZALEZ; LEE, 1988) e possível de ser incluída com custo relativamente baixo. RF5 deve-se à grande diversidade de movimentos possíveis na modalidade de locomoção por pernas, que oferece uma quantidade maior de parâmetros a serem explorados em relação às outras formas de locomoção terrestre. Por fim, RF6 atende ao requisito de operabilidade remota, uma das principais justificativas para a aplicabilidade do sistema no contexto proposto — telerrobótica para o emprego da Robótica Analítica.

Os requisitos não-funcionais adotados são apresentados na Tabela 2. RNF1, é possibilitado pelo uso de um roteador como principal eletrônica de controle, como será descrito mais à frente. O OpenWrt é uma distribuição do sistema operacional Linux para sistemas embarcados, amplamente adotada em roteadores *wireless*, possuindo uma comunidade com mais de 10 anos, suporte para mais de 1000 modelos de roteadores, repositórios com mais de 2000

pacotes de software oficiais — sem considerar os submetidos pelos usuários da sua comunidade (OPENWRT, 2016), sugerindo fortemente que é uma opção madura e confiável para a implementação do sistema proposto. RNF2 deve-se à popularidade e relativo baixo custo deste modelo de microcontrolador, o que implica maior chance de documentação e exemplos de aplicação. RNF3 permite a implementação de comunicação por troca de mensagens em alto nível, o que auxilia RF3 e é altamente desejável do ponto de vista de usabilidade, visto que um dos principais propósitos de uma plataforma é permitir que o usuário evite preocupação com aspectos de baixo nível do sistema.

Tabela 2: Requisitos não-funcionais.

Nome	Descrição
RNF1	O robô deverá operar sobre o sistema operacional OpenWrt.
RNF2	O interfaceamento com os sensores e atuadores, quando necessário, deverá ser feito por um microcontrolador Texas Instruments Tiva-C.
RNF3	A comunicação com os sensores e atuadores deverá ser feita por meio da biblioteca nanomsg.

No Capítulo 4 — Experimentos e Discussão, os requisitos funcionais serão retomados na medida em que os experimentos que os validam forem apresentados.

3.2 ARQUITETURA

O sistema desenvolvido é dividido em três módulos principais (Figura 5). O *Módulo de Comunicação e Visão* é responsável por trocar mensagens via Wi-Fi com os operadores do robô, além de transmitir as imagens que captura por meio de uma câmera. O *Módulo de Sensoriamento e Interfaceamento* concentra os sensores de propósito geral do robô (i.e. sonares, giroscópio e acelerômetro) e um microcontrolador, que media a interação entre os sensores e atuadores com o *Módulo de Comunicação e Visão*. O *Módulo de Locomoção* concentra os atuadores responsáveis pela locomoção do robô (i.e. servomotores) e a placa controladora dos mesmos. Neste módulo também estão incluídos os sensores de corrente para o monitoramento do estado dos servomotores.

A operação e monitoramento do sistema se dá por meio de um *middleware* próprio, descrito adiante, que abstrai estas ações como simples trocas de mensagens com os sensores e atuadores do robô. O *middleware* faz parte do *Módulo de Comunicação e Visão*, que propaga para os outros módulos as mensagens que recebe via Wi-Fi e vice-versa.

Os itens seguintes apresentam cada um dos módulos em mais detalhes.

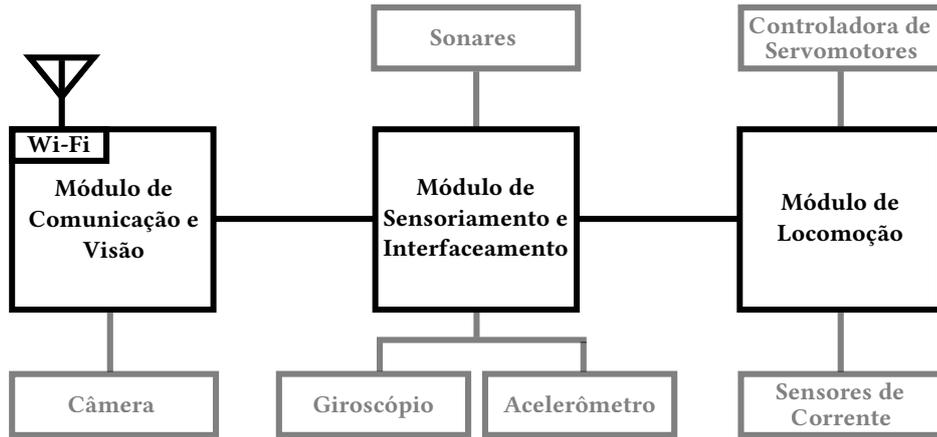


Figura 5: Arquitetura do sistema.

3.2.1 Módulo de Comunicação e Visão

Este é o principal módulo do sistema. Por meio deste, o robô é controlado e monitorado. Prezando por minimização do número de componentes, diminuição de tempo de desenvolvimento, abundância de soluções técnicas pré-existentes, assim como a disponibilidade de obtenção no varejo, optou-se pela implementação deste módulo em um roteador. O roteador oferece comunicação *wireless* (Wi-Fi), programabilidade (suporta sistema operacional Linux) e, dependendo do modelo, disponibilidade de interfaces de comunicação com os demais módulos via USB. A quantidade de entradas USB, entretanto, restringe-se comumente a um número pequeno (e.g. duas), sendo insuficiente para comunicar-se com o número de sensores e atuadores presentes no sistema, que chega a dezenas. Isso é mitigado por meio de um microcontrolador, que ocupa somente uma conexão USB com o roteador, enquanto pode possuir dezenas de terminais de entrada/saída — servindo basicamente como um multiplexador. O módulo também contém a câmera que realiza as funções de visão do sistema. Apesar de ser essencialmente um sensor, a câmera é mantida neste módulo por conveniência técnica, visto que o seu acesso é feito por meio de conexão USB com o roteador.

O roteador utilizado é o TP-LINK WDR3600 (TP-LINK, 2013), que possui um SOC (*system-on-chip*) AR9344 com *clock* de 560 MHz, 128 MB de memória RAM, 8 MB de memória Flash, e duas portas USB (OPENWRT, 2015b). O microcontrolador utilizado é o Texas Instruments Tiva-C, que possui uma CPU 32-bits ARM Cortex-M4F com *clock* de 80 MHz, 32 KB de memória SRAM, 256 KB de memória Flash, 2 KB de memória EEPROM (TEXAS INSTRUMENTS, 2015a). A câmera utilizada é a *webcam* Logitech C270, capaz de capturar vídeo com resolução de até 1280 x 720 pixels (LOGITECH, 2015).

Comunicação. Os operadores comunicam-se com o robô por meio de um *middleware* próprio em execução no roteador, sobre o sistema operacional OpenWrt – Barrier Breaker (OPENWRT, 2015a). Considerando que sistemas de *middleware* em aplicações críticas têm sido reescritos com o objetivo de minimizar tamanho e complexidade (DWORKAK; SLIWINSKI; SOBCZAK, 2011), consideramos desde o início uma abordagem minimalista na implementação do *middleware* do robô, incluindo o mínimo para possibilitar a troca de mensagens com sensores e atuadores. Supomos que o baixo consumo de memória e processamento, além da simplicidade de operações, compensam as desvantagens de não utilizarmos *middleware* mais populares na robótica (e.g. ROS). O minimalismo do *middleware* também é desejável por diminuir a influência que as suas abstrações implicam à arquitetura dos sistemas que o utilizam (NITTO; ROSENBLUM, 1999).

O *middleware* do robô é implementado utilizando a *nanomsg* (NANOMSG, 2015) – uma biblioteca para desenvolvimento de aplicações distribuídas por meio de *sockets* de alto nível. A biblioteca fornece uma série de padrões para troca de mensagens (e.g. *surveyor/respondent*, *publish/subscribe*, *request/reply*), bem como suporte para diferentes mecanismos de transporte (i.e. dentro de um processo – *in-process*, entre processos – IPC, e em rede – TCP). A estrutura do *middleware* é apresentada na Figura 6.

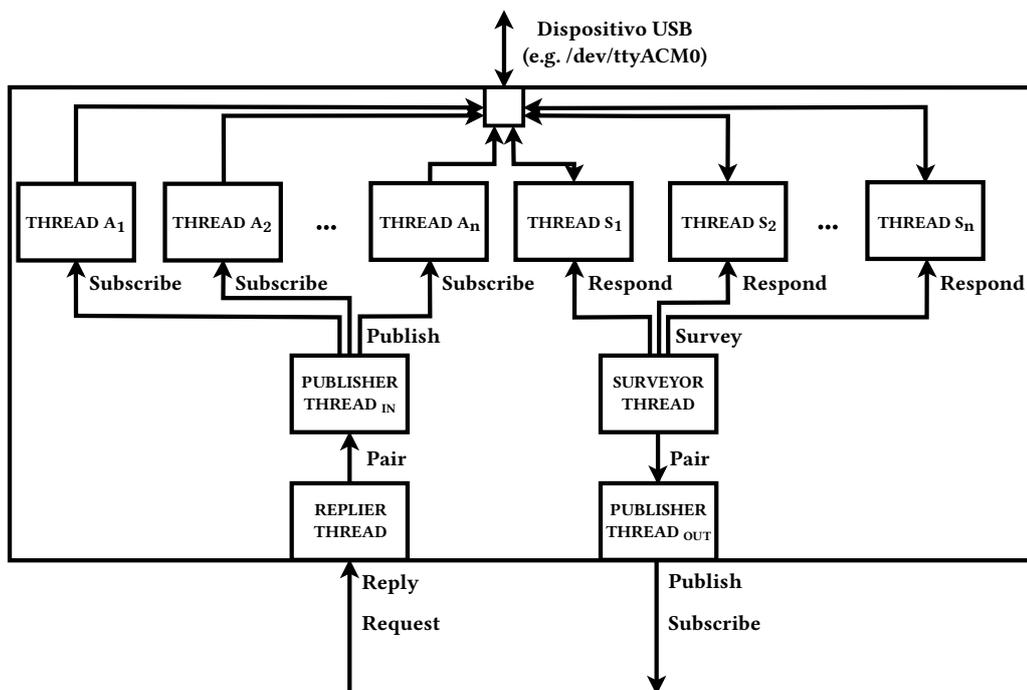


Figura 6: Arquitetura do *middleware*.

Existem duas formas do usuário se comunicar com o robô por meio do *middleware*: controle dos atuadores pelo envio de requisições (*request* do modelo *request/reply*) e monitoramento de sensores por meio de assinatura (*subscribe* do modelo *publish/subscribe*). Cada dispositivo, sensor ou atuador, possui um número identificador pelo qual este pode ser acessado. As mensagens de requisição possuem um formato específico, descrito na Figura 7.



Figura 7: Formato de mensagens adotada no sistema. A mensagem é codificada em caracteres ASCII.

A mensagem é delimitada pelos caracteres ASCII '#' (inicializador) e '!' (terminador). Entre os delimitadores, existem no mínimo dois campos de 3 caracteres cada: *ID*, *OPCODE*, e um de tamanho variável: *DATA*. O campo *ID* armazena o número (inteiro e positivo) de identificação do dispositivo. A semântica dos campos *OPCODE* e *DATA* dependem dos dispositivos que serão acessados, sendo sugerida a utilização do campo *OPCODE* para descrever a operação a ser efetuada no dispositivo e o campo *DATA* para descrever os parâmetros desta operação. O tamanho de *DATA* é definido como um múltiplo de 3 caracteres, por consistência e para facilitar a especificação da sua semântica (isso é feito na seção 3.2.2). O tamanho de cada campo foi convencionado como 3 caracteres pois a faixa que pode representar (i.e. 000-999) é adequada para os nossos propósitos. O formato de mensagem descrito também é utilizado pelo *middleware* na publicação de dados aos operadores.

Na sua organização interna, existe uma *thread* para cada atuador, que assina tópicos correspondentes ao seu próprio identificador, recebendo somente requisições de comando relevantes a si mesmo. Esses comandos são então direcionados para o *Módulo de Sensoriamento e Interfaceamento* via USB, onde o microcontrolador converte as mensagens dos comandos em ações concretas nos dispositivos (e.g. modulação de largura de pulso). Existe também uma *thread* para cada sensor, que é consultada (*survey* do modelo *surveyor/respondent*) a uma determinada frequência. Cada *thread* lê dados correspondentes ao seu respectivo sensor via USB, requisitando-as do *Módulo de Sensoriamento e Interfaceamento* e repassando para que sejam publicadas para os operadores que tenham assinado os tópicos referentes aos seus respectivos identificadores.

Visão. Como modalidade de sensoriamento mais versátil disponível, a visão fornece uma série de informações, como cor, textura, forma, tamanho e distância (FU; GONZALEZ; LEE, 1988; VIEIRA NETO, 2006). Por ser uma forma de percepção existente também em seres humanos, a visão também abre possibilidades de pesquisa como robôs para automatização de processos de inspeção visual (VIEIRA NETO, 2006) e validação de modelos computacionais de atenção visual (ITTI; KOCH; NIEBUR, 1998). A implementação da visão do robô se dá pela transmissão em *streaming* das imagens capturadas pela sua câmera, que é feito pelo software MJPG-streamer (MJPG-STREAMER, 2014).

3.2.2 Módulo de Sensoriamento e Interfaceamento

Este módulo concentra os sensores de propósito geral do sistema, isto é, os que não são atrelados a outros módulos seja por coerência lógica (e.g. sensores de corrente do *Módulo de Locomoção*) ou conveniência técnica (e.g. câmera no módulo de comunicação e controle). Além disso, este módulo contém o microcontrolador que media a comunicação dos sensores e atuadores com o *Módulo de Comunicação e Visão*.

Sensoriamento. Os sensores incorporados neste módulo são quatro de distância por ultrassom (um para cada direção — frente, trás, esquerda e direita), um giroscópio e um acelerômetro. Os sensores de distância são utilizados basicamente para a detecção de obstáculos, enquanto o giroscópio e acelerômetro são convenientes para determinar o estado espacial do robô (i.e. orientação) e quedas/vibrações. Estas variáveis são desejáveis pois, como o robô apresenta locomoção por pernas, espera-se que este seja utilizado para análise do seu comportamento em ambientes irregulares, onde esta forma de locomoção é mais adequada. Os dispositivos utilizados são o MPU-6050, *chip* que integra um giroscópio e um acelerômetro digital (INVENSENSE, 2015), sensor de distância por ultrassom HC-SR04 (ITEAD STUDIO, 2015). O diagrama da PCI do *Módulo de Sensoriamento*, seu diagrama esquemático e sua foto estão disponíveis no Apêndice B.

Interfaceamento. Devido ao número limitado de interfaces de comunicação que o roteador (*Módulo de Comunicação e Visão*) possui, é necessário expandir a capacidade de interfaceamento com sensores e atuadores. Isso é feito por meio de um microcontrolador com um sistema operacional de tempo real, o FreeRTOS (REAL TIME ENGINEERS, 2015). O sistema executa um conjunto de *threads* responsáveis por receber mensagens

via USB (no caso, requisições feitas pelo roteador), traduzi-las em comandos para os sensores e atuadores, retornando respostas no caso dos sensores. A organização do software embarcado que implementa essas *threads* é apresentada na Figura 8.

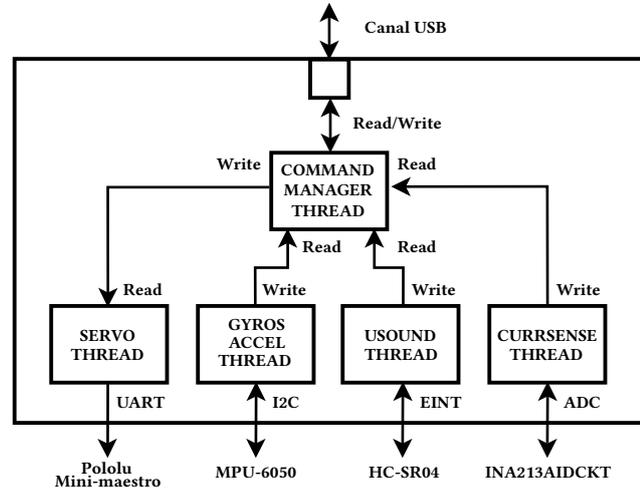


Figura 8: Arquitetura do software embarcado. Pololu Mini-maestro: placa de controle de servomotores, MPU-6050: *chip* integrado com acelerômetro e giroscópio, HC-SR04: sensor de distância por ultrassom, INA213AIDCKT: sensor de corrente.

As mensagens trocadas pelas *threads* no microcontrolador têm o mesmo formato das mensagens trocadas no *Módulo de Comunicação e Visão*, visto que são as mesmas mensagens — as dos atuadores foram recebidas pelo roteador e as dos sensores serão enviadas para o mesmo. O formato descrito anteriormente deixa a utilização e significado dos campos *OPCODE* e *DATA* conforme o dispositivo sendo acessado. O software embarcado interage diretamente com os sensores e atuadores, sendo adequada, neste momento, a descrição da semântica destes campos. As Tabelas 3 e 4 descrevem a semântica dos campos para, respectivamente, mensagens de comando aos dispositivos e as mensagens de resposta que o software embarcado retorna ao roteador.

Tabela 3: Protocolo de comunicação — Mensagens de comando.

Dispositivo	Mensagem		
	ID	OPCODE	DATA
Servomotores	0 - 23	0 - Altera ângulo	Ângulo
	0 - 23	1 - Altera velocidade	Velocidade
	0 - 23	2 - Altera aceleração	Aceleração
	0 - 23	3 - Lê corrente	—
Giroscópio	24	0 - Lê vel. angular	—
Acelerômetro	24	1 - Lê aceleração	—
Sonar	25 - 28	0 - Lê distância	—

Tabela 4: Protocolo de comunicação — Mensagens de resposta. Os subscritos *int* e *frac* indicam as partes inteira (à esquerda da vírgula) e fracionária (à direita da vírgula) dos dados. Para o giroscópio e o acelerômetro, o primeiro caractere da parte inteira indica o sinal.

Dispositivo	Mensagem							
	ID	OPCODE	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5
Servomotores	0 - 23	0 - Corrente	\dot{i}_{int}	\dot{i}_{frac}	—	—	—	—
Giroscópio	24	0 - Vel. angular	x_{int}	x_{frac}	y_{int}	y_{frac}	z_{int}	z_{frac}
Acelerômetro	24	1 - Aceleração	x_{int}	x_{frac}	y_{int}	y_{frac}	z_{int}	z_{frac}
Sonar	25 - 28	0 - Distância	d	—	—	—	—	—

3.2.3 Módulo de Locomoção

Este módulo concentra a maior parte da complexidade física do sistema devido à forma de locomoção adotada, a saber: locomoção por pernas. A locomoção é feita por seis pernas, cada uma com três graus de liberdade. Esta configuração foi adotada pois permite manter a condição de estabilidade estática mais facilmente do que com um número menor de pernas, podendo manter esta estabilidade desde que mantenha ao menos três pernas no chão (CLARK et al., 2001). Cada grau de liberdade das pernas é dado por um servomotor, resultando em um total de 18 servomotores no módulo. O acionamento destes motores é feito por meio de uma placa controladora de servomotores com 24 canais, a Pololu Mini-maestro (POLOLU, 2015), que é operada por comunicação serial UART pelo microcontrolador do *Módulo de Sensoriamento e Interfaceamento*. O monitoramento da corrente consumida por cada um dos servomotores é feito por meio de um conjunto de sensores Texas Instruments INA213AIDCKT (TEXAS INSTRUMENTS, 2015b), multiplexados devido à quantidade limitada de entradas do conversor analógico-digital disponíveis no microcontrolador do *Módulo de Sensoriamento e Interfaceamento*, o qual faz a leitura dos sensores. O diagrama da PCI do *Módulo de Locomoção*, bem como o seu diagrama esquemático e foto estão disponíveis no Apêndice B.

Para a determinação dos servomotores adequados, foi feita uma simulação da mecânica do robô no *Mathworks Simulink*, mais especificamente, no ambiente de simulação *SimMechanics* (MATHWORKS, 2015). O objetivo foi estimar a capacidade de torque necessária dos servomotores para que estes suportem a massa do robô. Foram feitas duas simplificações para facilitar a simulação e sua análise. A primeira foi a restrição da simulação à condição estática do robô, isto é, não foi simulado o seu movimento. A segunda foi considerar as pernas em posição de tripé, visto que esta corresponde à condição mínima para uma marcha estaticamente estável em um robô hexápode (PORTA; CELAYA, 1998).

Na Figura 9 é apresentado o modelo 3D gerado no *SimMechanics*, bem como a numeração que utilizamos para referência às pernas do robô. Cada perna do robô é modelada como três hastes (*links* na terminologia do ambiente de simulação) unidas por junções. Cada junção realiza uma articulação oferecendo um grau de liberdade adicional, onde os movimentos das junções 1 (entre os *links* 0 e 1) e 2 (entre os *links* 1 e 2) varrem o plano (x, z) no espaço enquanto a junção 0 (entre o tronco e o *link* 0) varre o plano (x, y) .

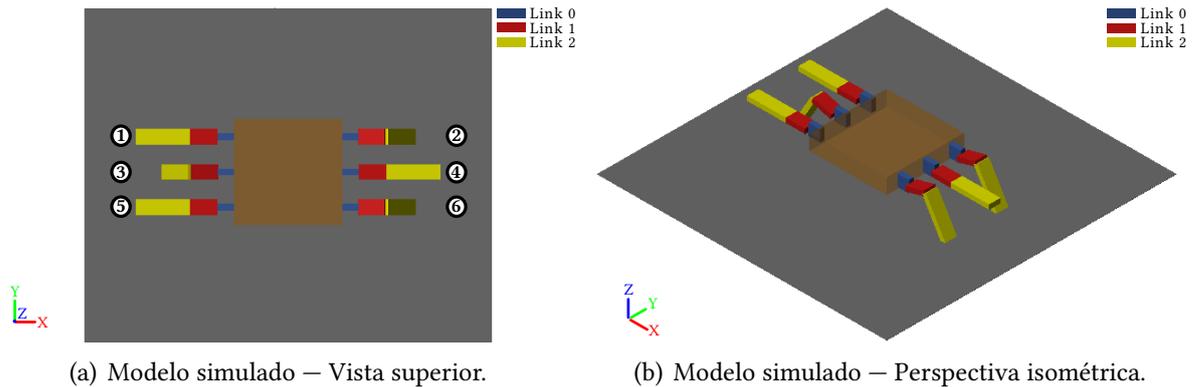
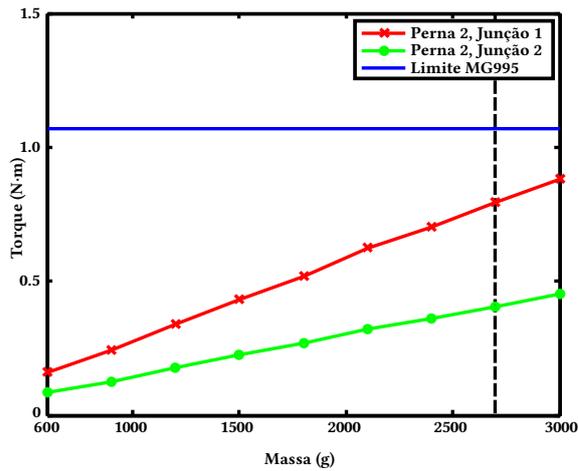


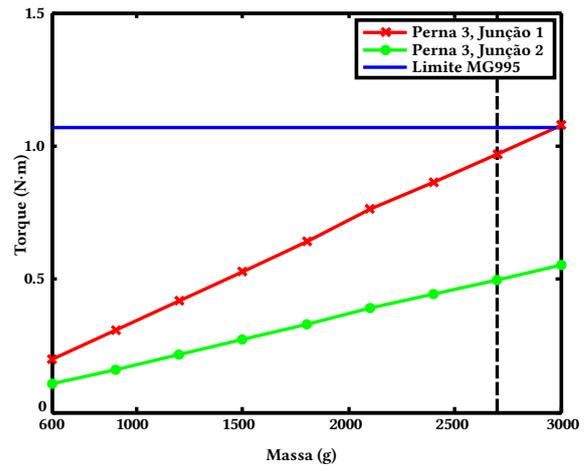
Figura 9: Visualização 3D do modelo mecânico simulado.

O ensaio da simulação se deu pela análise dos torques exercidos nas junções das pernas do robô de acordo com a variação de massa total do robô. Foram analisadas somente as pernas que estão em contato com o chão (i.e. 2, 3, e 6), pois são as que sustentam a carga da estrutura no posicionamento em tripé. Os valores de massa foram variados de 600 g a 3000 g, em intervalos de 300 g. A *junção 0* foi desconsiderada, visto que esta se articula no plano (x, y) , o qual não é influenciado pelo aumento de massa. Na Figura 10 são apresentados os resultados obtidos. Os resultados referentes à *perna 6* foram omitidos, visto que são idênticos aos da *perna 2*, devido à simetria da estrutura.

É possível verificar pelos gráficos que a *junção 1* na *perna 3* é que sofre mais esforço, sendo o fator limitante na escolha do modelo de servomotor adequado. Com base nestes dados, e considerando a estimativa da massa total do robô real (reta vertical tracejada na Figura 10), foram adotados para estas junções servomotores do modelo TowerPro MG995 (TOWERPRO, 2015). O torque máximo deste modelo é $1,08 \text{ N} \cdot \text{m}$ (operando a 6 V), indicado pela reta horizontal azul na Figura 10. Considerando que a massa total do robô real foi estimada como 2700 g e que o torque máximo deste servomotor é alcançado apenas para uma massa de 3000 g, pode-se dizer que existe uma margem de 10% (300 g) até a faixa de sobrecarga, sugerindo que a escolha do servomotor é adequada.



(a) Gráfico Massa-Torque – Perna 2.



(b) Gráfico Massa-Torque – Perna 3.

Figura 10: Torque nas *junções 1 e 2* das *pernas 2 e 3* para diferentes valores de massa. A linha azul indica o torque máximo ($1.08 \text{ N}\cdot\text{m}$) suportado pelo motor adotado nas juntas analisadas. A reta tracejada indica a estimativa da massa total do robô real (2700 g).

A *junção 0* não sofre esforço significativo no eixo (x, z) mas sofre no eixo (x, y), visto que ao executar uma marcha deve ser capaz de carregar a massa da sua respectiva perna. Como a simulação considera apenas a mecânica em condição estática, o ensaio da simulação se deu isolando-se uma perna e analisando o torque sobre a *junção 0* para diferentes valores de massa para a perna. O gráfico dos torques obtidos nesta simulação podem ser vistos na Figura 11. Como todas as pernas são idênticas e o torque na junção em questão é independente da sua posição, este gráfico representa todas as *junções 0* do robô.

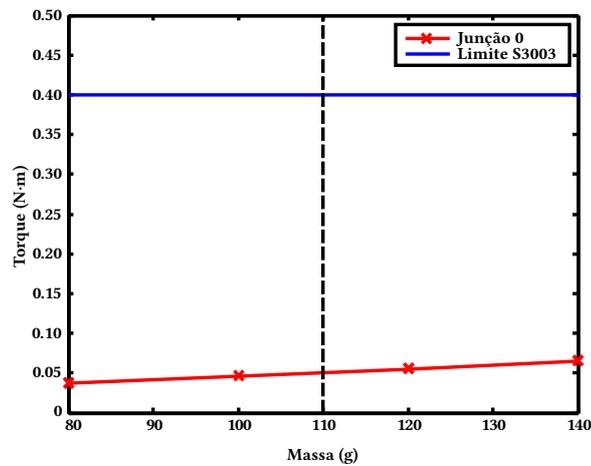


Figura 11: Torque na *junção 0* de uma perna do robô. A linha azul indica o torque máximo ($0.40 \text{ N}\cdot\text{m}$) suportado pelo motor adotado na *junta 0*. A reta tracejada indica a massa aproximada de uma perna (110 g).

Visto que o torque necessário na *junção 0* é consideravelmente inferior às das *junções 1* e *2*, adotou-se para esta um modelo de servomotor com especificações inferiores para diminuir o custo. O modelo adotado é o Futaba S3003 (FUTABA, 2015), cujo torque máximo suportado ($0.40 \text{ N} \cdot \text{m}$) é indicado pela reta horizontal azul no gráfico da Figura 11. Como pode ser visto, o torque necessário para suportar um valor de massa equivalente ao peso de uma perna (indicado pela reta tracejada) é consideravelmente inferior ao limite do servomotor adotado. O peso de uma perna foi estimado como 110 g, correspondente ao peso dos dois servomotores nas *junções 1* e *2* (um único servomotor TowerPro MG995 possui massa de 55 g (TOWERPRO, 2015)).

3.2.4 Estrutura Física

Foi elaborada uma estrutura própria para o robô, que foi construída em acrílico preto opaco. A abordagem foi separar a estrutura em termos dos módulos gerais da arquitetura do robô, a saber: *Módulo de Comunicação e Visão*, *Módulo de Sensoriamento e Interfaceamento* e *Módulo de Locomoção*. A Figura 12 apresenta o modelo 3D do robô em perspectiva isométrica, enquanto a Figura 13 apresenta as vistas superior e inferior. O *layout* da estrutura física está disponível no Apêndice A.

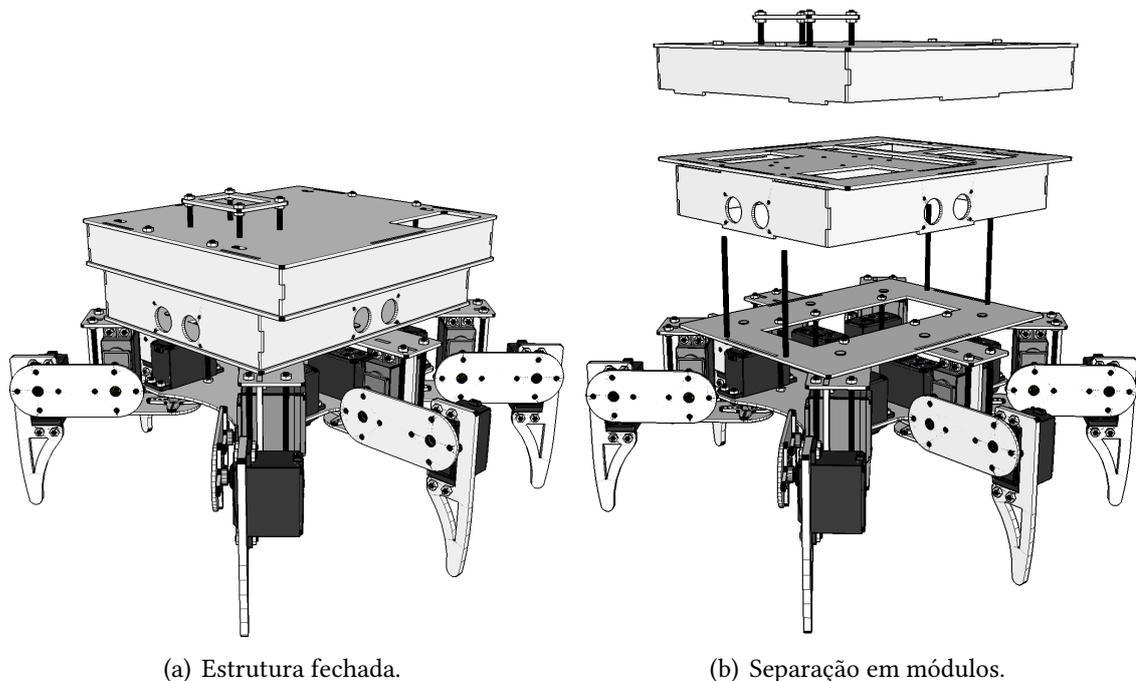


Figura 12: Modelo 3D da estrutura física do robô em perspectiva isométrica. A figura à direita apresenta a separação da estrutura em termos dos módulos, de cima para baixo: *Comunicação e Visão*, *Sensoriamento e Interfaceamento*, e *Locomoção*.

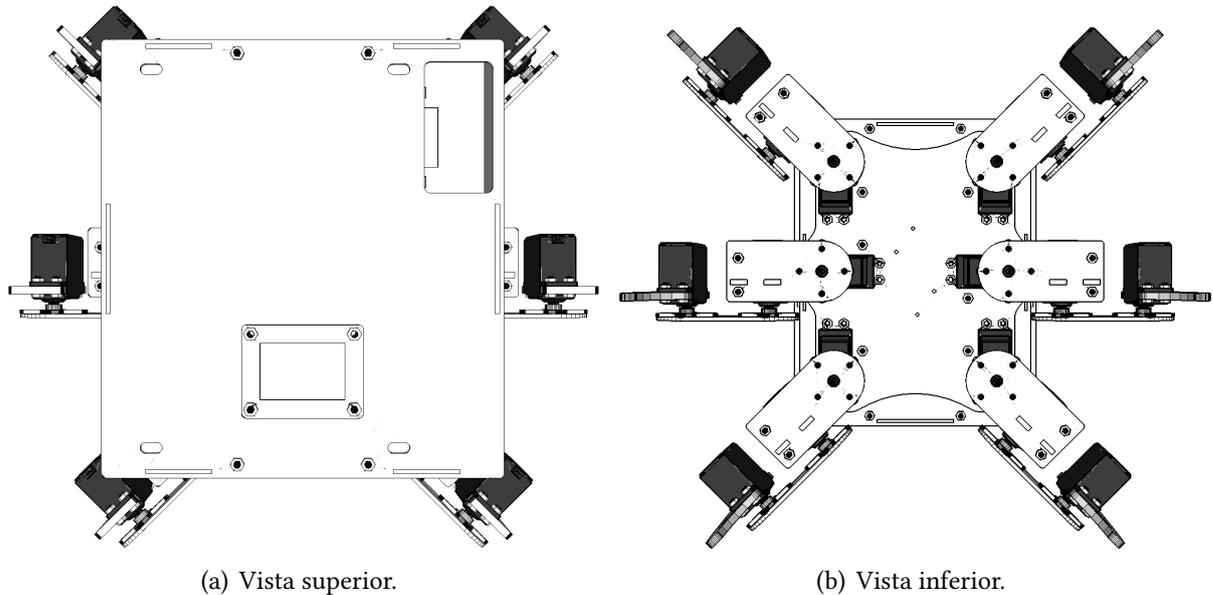


Figura 13: Modelo 3D da estrutura física do robô – Vistas superior e inferior.

A camada referente ao *Módulo de Comunicação e Visão* comporta a placa do roteador e permite a fixação da câmera na sua superfície superior. O mecanismo de fixação da câmera pode ser visto no topo do robô, nas Figuras 12(a) e 13(a), e consiste em uma peça de acrílico ajustável por parafusos. A camada referente ao *Módulo de Sensoriamento e Interfaceamento* permite a fixação do microcontrolador, bem como a fixação da placa que concentra os sensores. Um detalhe importante desta camada é que as suas laterais possuem aberturas para permitir a operação adequada dos sensores de distância por ultrassom.

A estrutura do *Módulo de Locomoção* é a mais elaborada, comportando a placa controladora de servomotores no seu centro e expandindo diversas outras peças fragmentadas entre as articulações das pernas. Diferente dos módulos anteriores, que tinham como requisito apenas a capacidade de conter e fixar componentes, este módulo possui peças cujos tamanhos interferem diretamente na distribuição de carga e, conseqüentemente, na capacidade dos motores sustentarem o robô. As peças em questão são os *links 0, 1 e 2* mencionados anteriormente na simulação, cujos comprimentos têm impacto direto na distribuição dos torques nas junções. Os seus comprimentos foram determinados experimentalmente considerando fatores como tamanho da placa controladora de servomotores, tamanho dos servomotores e formato dos cortes do tronco. A partir destas informações, o sistema foi simulado para uma série de comprimentos condizentes com os requisitos e adotou-se uma destas configurações de comprimento. A configuração adotada foi *link 0* – 3,4 cm; *link 1* – 5 cm; *link 2* – 10 cm.

3.3 O SISTEMA INTEGRADO

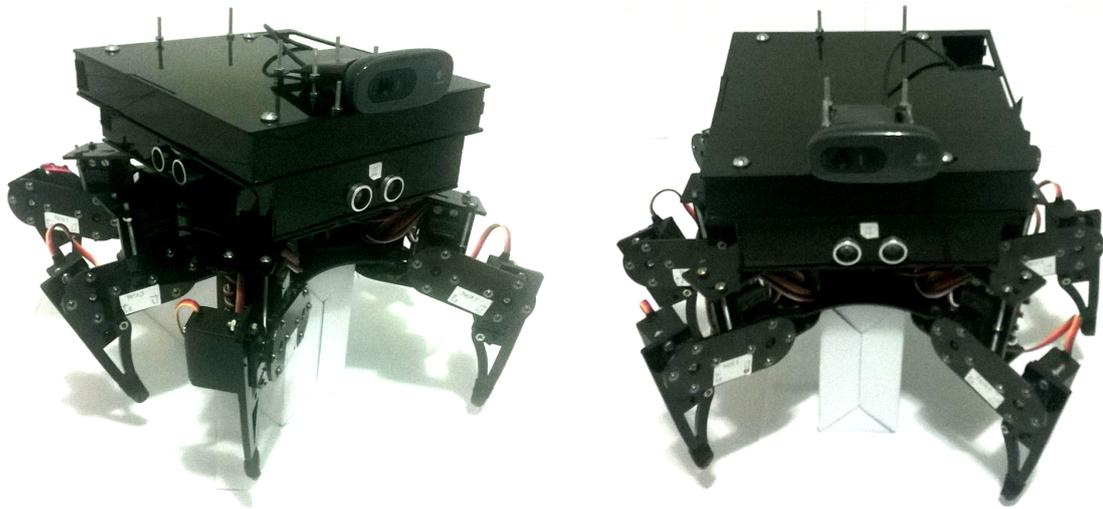


Figura 14: Estrutura do sistema integrado.

A estrutura consequente da integração dos módulos descritos na seção 3.2, o robô propriamente dito, é apresentada na Figura 14. O robô apresenta, na sua lateral, os conectores originais do roteador, os quais podem ser acessados para conexão *Ethernet*. Esta conexão é utilizada inicialmente para a configuração da sua conexão Wi-Fi ao *wireless access point* local. A partir deste ponto a conexão cabeada pode ser desfeita e o acesso ao robô pode ser feita de forma exclusivamente sem fio. Apesar de a comunicação ser sem fio, a alimentação do robô é cabeada e utiliza uma fonte DC de 360 W, 6 V (adaptada para 6 V) e 30 A.

A Figura 15 apresenta um exemplo ilustrativo da operação do sistema para as suas três principais funções: comando de atuadores, monitoramento de sensores e acesso à visão do robô. Estas funções podem ser iniciadas por um só operador ou diversos, em rede local ou remota, desde que sejam conhecidos o endereço IP do robô e as portas nas quais ele responde. A seguir são apresentados alguns aspectos técnicos da operação do sistema. As subseções seguintes explicam o exemplo descrito na Figura 15 em mais detalhes.

O acesso ao robô se dá por meio de três portas TCP, uma para envio de comando aos atuadores (9000), uma para assinatura de sensores (9001) e outra para acesso à visão do robô (8080). No caso do acesso aos sensores e atuadores, basta utilizar alguma linguagem de programação ou programa com suporte à biblioteca *nanomsg*, instanciar um *socket* do tipo adequado (*requester* para comandar atuadores, *subscriber* para monitorar sensores) e enviar ou receber as mensagens de acordo com o formato especificado nas Tabelas 3 e 4.

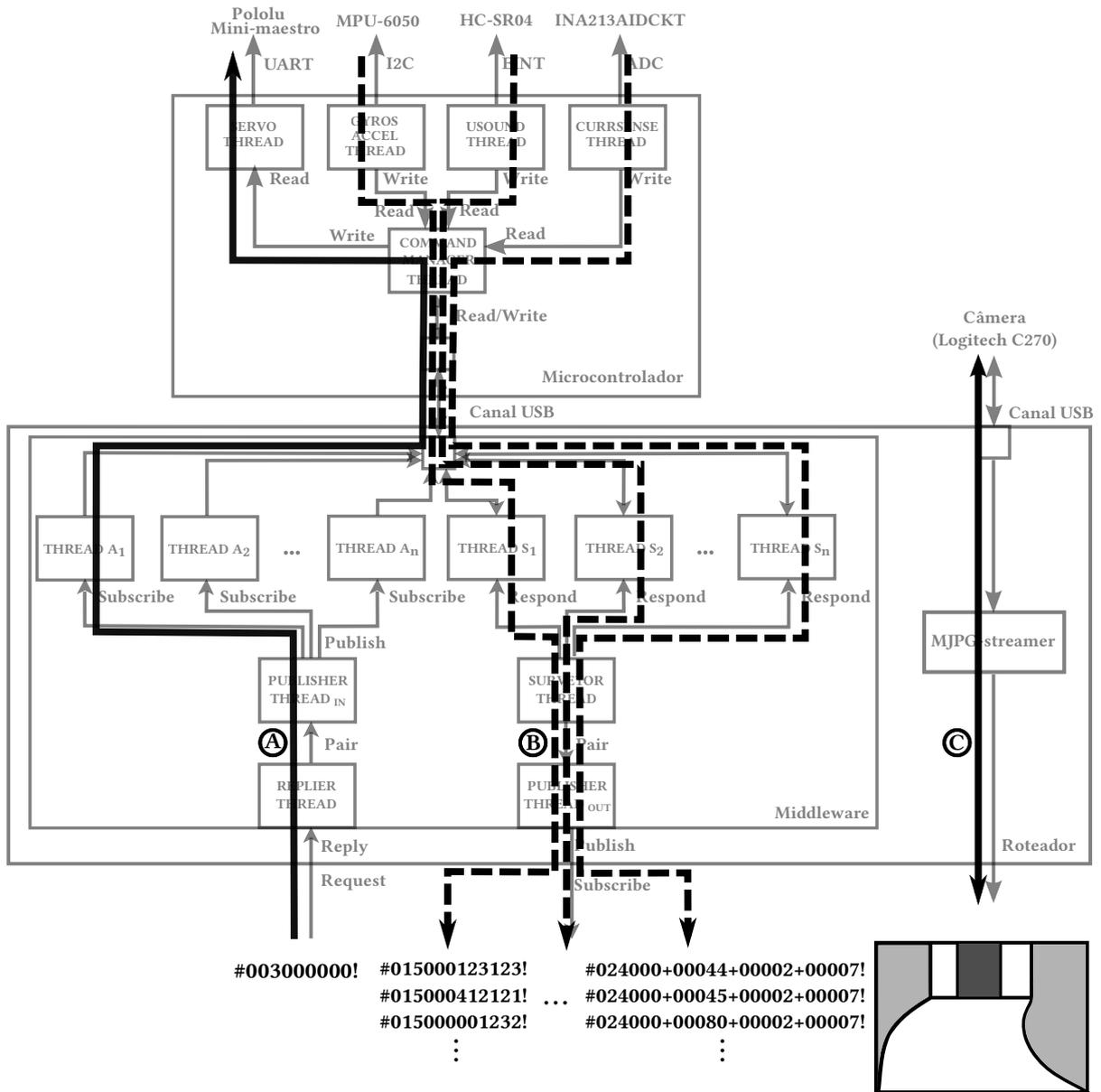


Figura 15: Descrição geral da operação do sistema. A – Envio de comando para atuadores. B – Monitoramento de sensores. C – Acesso à visão do robô. As cadeias de caracteres na parte inferior descrevem (da esquerda para a direita) um comando a atuador e leituras de sensores.

O nanocat é um aplicativo de linha de comando que acompanha a biblioteca nanomsg para fins de depuração, capaz de enviar e receber mensagens nos modelos que a biblioteca suporta (NANOMSG, 2015). Para manter a explicação breve, são apresentados exemplos de comando a atuador e monitoração de sensor utilizando o nanocat, em vez de implementação em alguma linguagem de programação. Se o custo computacional devido a chamadas de programas externos não for proibitivo, pode-se utilizar o nanocat como uma abordagem simples para comunicação com sistemas baseados na biblioteca nanomsg de forma praticamente independente de linguagem de programação, dispensando o uso de *wrappers* para cada linguagem.

3.3.1 Comando de atuadores

O fluxo de dados decorrente do envio de um comando a um atuador é indicado pela seta *A* na Figura 15. A chamada ao nanomsg neste caso seria a seguinte:

```
nanocat --req --data "#003000000!" --connect tcp://172.17.37.170:9000 --ascii
```

onde:

- req: define o socket como do tipo *requester*;
- data "#003000000!": define a mensagem de requisição de comando a ser enviada;
- connect tcp://172.17.37.170:9000: define o mecanismo de transporte e endereço;
- ascii: define que as mensagens de resposta devem ser impressas em codificação ASCII.

Primeiramente, uma mensagem de requisição é recebida por **REPLIER THREAD**, supondo que esta *thread* esteja aguardando conexões no endereço IP 172.17.37.170 na porta 9000, que a encaminha através de um *socket* de comunicação intraprocessos simples para **PUBLISHER THREAD**_{IN}. Cada atuador do sistema possui uma *thread* que a representa no *middleware*, assinando de **PUBLISHER THREAD**_{IN} o tópico referente ao seu próprio identificador — isto é, filtrando apenas mensagens direcionadas a si. Essas mensagens são então transmitidas via USB para **COMMAND MANAGER THREAD** no microcontrolador, que as processam, separando-as em seus respectivos campos e disparando as chamadas pertinentes aos atuadores adequados.

3.3.2 Monitoramento de sensores

O fluxo de dados consequente do monitoramento de sensores é representado pelo conjunto de setas *B* na Figura 15. Para monitorar um sensor, o operador deve assinar o tópico referente ao número identificador deste. A chamada ao nanomsg seria a seguinte:

```
nanocat --sub --subscribe "#015" --connect tcp://172.17.37.170:9001 --ascii
```

onde:

- sub: define o socket como do tipo *subscriber*;
- subscribe "#015": define o tópico a assinar;
- connect tcp://172.17.37.170:9001: define o mecanismo de transporte e endereço;
- ascii: define que as mensagens de resposta devem ser impressas em codificação ASCII.

A partir desta chamada, o operador assina de **PUBLISHER THREAD**_{OUT} o tópico referente ao número identificador 015, supondo que esta *thread* esteja aguardando conexões no endereço IP 172.17.37.170 na porta 9001. Deste momento em diante, **SURVEYOR THREAD**, que consulta os valores das *threads* referentes aos sensores a uma determinada taxa (por convenção, adotou-se um período de 100 ms), encaminha os valores recebidos para **PUBLISHER THREAD**_{OUT}, que publica as mensagens originadas de cada sensor aos assinantes adequados. As *threads* referentes aos sensores, ao serem consultadas por **SURVEYOR THREAD**, enviam uma requisição para o microcontrolador via USB. Esta requisição é tratada por **COMMAND MANAGER THREAD** que converte a requisição em um comando adequado de leitura ao sensor em requisitado e responde com o valor obtido.

3.3.3 Acesso à visão

A visão do robô é realizada por meio de imagens capturadas por uma *webcam*, que são transmitidas na forma de *streaming* no formato *Motion JPEG* através do software MJPG-streamer (MJPG-STREAMER, 2014). A transmissão pode ser acessada por meio de um navegador *web*, ou interface equivalente, no endereço IP do robô na porta 8080. O acesso à visão não tem relação alguma com o *middleware*, sendo tratada exclusivamente pelo MJPG-streamer. O fluxo de dados referentes ao acesso à visão do robô é representado pela seta C na Figura 15. Um exemplo de acesso à visão robô é apresentado no Experimento 5 do Capítulo 4, utilizando a linguagem de programação Python e a biblioteca de visão computacional OpenCV (OPENCV, 2016).

4 EXPERIMENTOS E DISCUSSÃO

Este capítulo apresenta uma série de experimentos com o fim de validar o sistema proposto em relação aos requisitos funcionais especificados no seu projeto. A ênfase é somente na validação dos requisitos, em detrimento da análise das características e comportamento dos dispositivos utilizados no sistema. A Tabela 5 apresenta cada requisito, suas descrições e os respectivos experimentos que os validam.

Tabela 5: Requisitos funcionais, suas descrições e lista experimentos que os validam.

Requisito	Descrição	Experimentos
RF1	O robô deverá fazer sensoriamento de distância, aceleração e orientação.	1, 2, 3
RF2	O robô deverá fazer sensoriamento das correntes consumidas por cada servomotor.	4
RF3	Os sensores e atuadores do robô deverão ser acessíveis por meio de troca de mensagens com protocolo bem definido.	1, 2, 3, 4
RF4	O robô deverá possuir visão por câmera.	5
RF5	O robô deverá se locomover por meio de pernas.	1
RF6	O robô deverá ser acessível por meio de conexão a rede Wi-Fi.	1, 2, 3, 4, 5

4.1 EXPERIMENTO 1 — SENSORIAMENTO DE DISTÂNCIA

Introdução. O sensoriamento de distâncias é comumente utilizado em robótica móvel para desvio de obstáculos e mapeamento de ambientes. O propósito do presente experimento foi validar a aplicação do sistema desenvolvido no que diz respeito ao monitoramento da distância do robô a obstáculos por meio de seus sonares.

Materiais e métodos. O experimento foi feito com o robô e uma parede circular concêntrica e ao redor deste. O sensoriamento de distância é feito por quatro sonares, acoplados cada um em uma direção do robô, que foi comandado a girar em torno de si mesmo. Durante este movimento (uma volta completa), as medidas dos sensores foram monitoradas para comparação com o valor esperado. Como o diâmetro D da parede circular é conhecido, espera-se que, em qualquer instante, a soma das medidas de cada par de sensores em direções opostas seja aproximadamente igual ao diâmetro D menos a distância S_i entre os sensores. A Figura 16 descreve as condições do experimento.

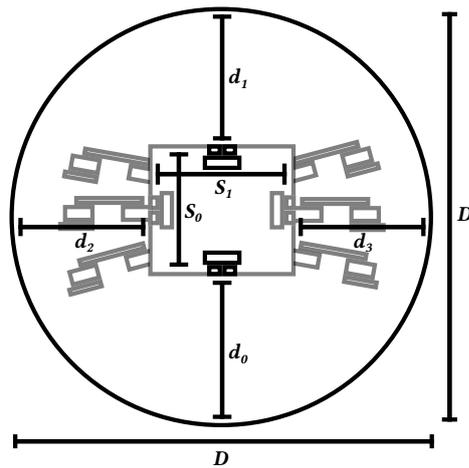


Figura 16: Experimento de medição de distâncias. Vista superior.

Resultados. As estimativas do diâmetro obtidas pela medição dos sensores são apresentadas na Figura 17. As distâncias entre pares de sensores opostos são, respectivamente, $s_0 = 20\text{ cm}$ e $s_1 = 16\text{ cm}$. Os valores obtidos mostram-se próximos do valor esperado ($D = 45\text{ cm}$), com médias e desvios padrão de, respectivamente: $44,43\text{ cm}$ e $0,51\text{ cm}$ para o par $d_0 - d_1$; $44,77\text{ cm}$ e $0,68\text{ cm}$ para o par $d_2 - d_3$. Consideramos que um desvio padrão de no máximo $0,68\text{ cm}$ é coerente e consideravelmente baixo para a grande maioria das aplicações do sistema, especialmente considerando a faixa de operação de $2\text{ cm} - 4\text{ m}$ do sensor, portanto validando a aplicação do sistema proposto em relação à monitoração de distância a obstáculos.

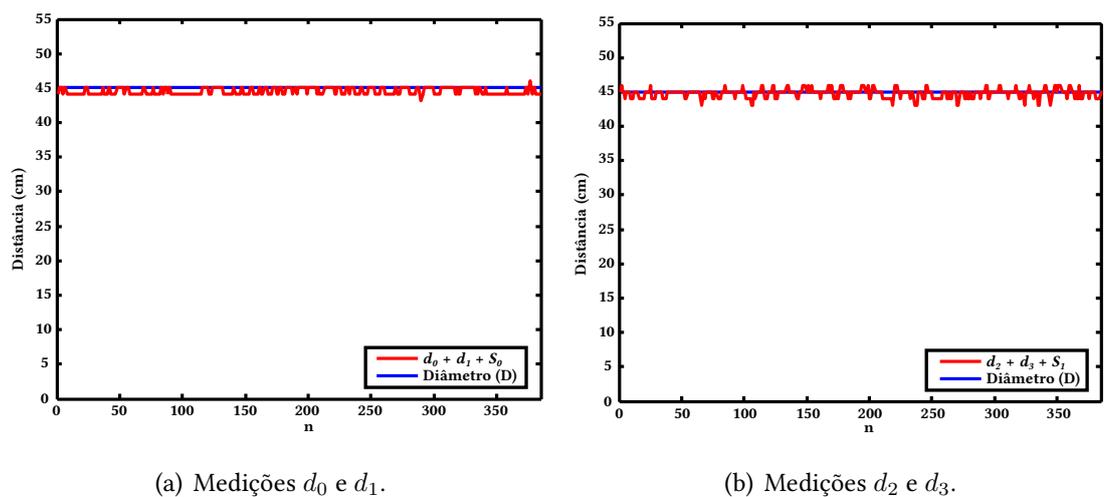


Figura 17: Medições de distância. A linha azul indica o valor esperado — o diâmetro real das paredes que cercam o robô. A linha vermelha indica as medições realizadas por meio dos sensores de distância.

4.2 EXPERIMENTO 2 – SENSORIAMENTO DE ACELERAÇÃO

Introdução. O sensoriamento de aceleração permite analisar a trajetória de um robô, bem como as características do seu movimento ao percorrê-la, visto que, por meio de integração no tempo, a aceleração fornece também a velocidade e o deslocamento. O propósito deste experimento foi validar a aplicação do sistema desenvolvido no que diz respeito ao monitoramento da aceleração do robô.

Materiais e métodos. O experimento foi feito com o robô, fio, polias e pesos. Para determinação das acelerações foi adotou-se o modelo da Máquina de Atwood, um experimento tradicional de mecânica clássica que permite determinar a aceleração de dois corpos a partir de suas massas e a aceleração da gravidade. O arranjo do experimento é descrito na Figura 18. A aceleração a é dada pela expressão (KLEPPNER; KOLENKOW, 1973):

$$a = \frac{(m_r - m_v) \cdot g}{m_r + m_v}$$

onde m_r é a massa do robô e o seu suporte, m_v é uma massa qualquer, g é a aceleração da gravidade (aproximada como $9,8 \text{ m/s}^2$). Espera-se que a leitura do sensor seja aproximadamente igual à aceleração obtida analiticamente pela expressão acima.

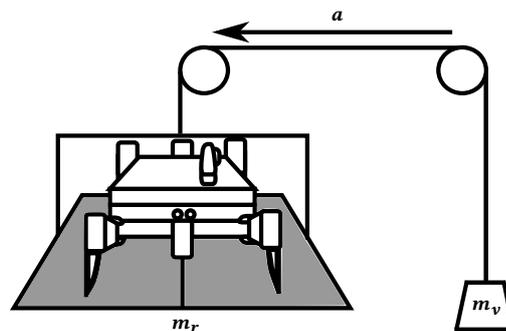


Figura 18: Experimento de medição de aceleração vertical. Vista lateral.

Resultados. A massa m_r é igual a 2880 g , enquanto m_v foi fixado em 1200 g para desequilibrar o sistema e causar uma aceleração conhecida, obtida analiticamente: $4,04 \text{ m/s}^2$. Foram feitos 10 ensaios do experimento, cujas medidas são apresentadas na Figura 19.

As medidas são razoavelmente próximas do valor esperado, com os ensaios apresentando uma medida média de $3,76 \text{ m/s}^2$. Entretanto, há uma dispersão considerável nos dados, que apresentam desvio padrão de $0,78 \text{ m/s}^2$, praticamente 20% do valor esperado. Já se esperava alguma diferença sistemática entre as medidas, devido a fatores como o atrito

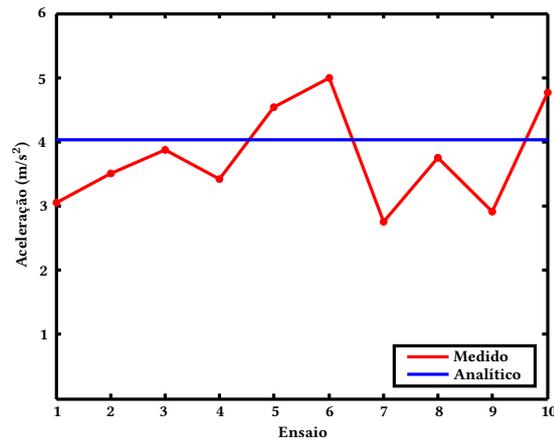


Figura 19: Medições de aceleração. A linha azul indica o valor esperado. A linha vermelha indica as medições realizadas por meio do acelerômetro.

do fio com as polias, que não é o caso para uma dispersão desta magnitude. Uma explicação plausível para isso são as condições experimentais, visto que no ambiente de testes os objetos tinham apenas uma altura de 80 cm para desenvolver o seu movimento, o que implica poucas medições durante a queda — tornando as medidas consideravelmente suscetíveis a ruído. Apesar disso, como o propósito do experimento não é a verificação da precisão do sensor, consideramos as medições como coerentes o suficiente para demonstrar a aplicabilidade do sistema no monitoramento de aceleração, cumprindo o propósito de validação deste aspecto do sistema.

4.3 EXPERIMENTO 3 — SENSORIAMENTO DE ORIENTAÇÃO

Introdução. O sensoriamento de orientação é útil na determinação da inclinação do plano do robô. Esta informação é aplicável, por exemplo, em situações onde deve-se adaptar o passo para que a carga do robô se movimente suavemente apesar de irregularidades no terreno. O propósito deste experimento foi validar a aplicação do sistema desenvolvido no que diz respeito ao monitoramento da orientação do robô. Apesar de giroscópios e acelerômetros serem comumente utilizados em par para estimar orientação, utilizamos somente o giroscópio para isolar cada experimento a um só tipo de sensor.

Materiais e métodos. O experimento foi feito com o robô, um transferidor e uma superfície plana. Fixou-se o robô sobre a superfície, que foi inclinada em uma série de ângulos medidos com o transferidor. Para cada ângulo, a orientação do robô foi estimada através da integração no tempo da velocidade angular medida pelo seu giroscópio

durante o movimento de 0° ao ângulo em questão. Para simplificar a análise, o experimento foi feito para somente um sentido. Espera-se que os ângulos estimados sejam aproximadamente iguais aos da inclinação real.

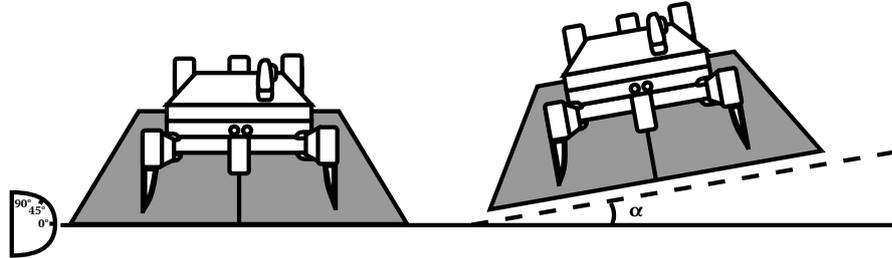


Figura 20: Experimento de medição da orientação. Vista lateral.

Resultados. Os ângulos adotados para o experimento foram 9° , 18° , 27° , 36° e 45° . Foram feitos 10 ensaios de medição para cada ângulo. As inclinações reais e suas respectivas medidas pelo sistema são apresentadas na Tabela 6.

Tabela 6: Inclinação real e inclinação estimada pelo giroscópio.

Real	Estimada	
	Média	Desvio padrão
9°	$9,39^\circ$	$0,49^\circ$
18°	$18,28^\circ$	$0,77^\circ$
27°	$26,63^\circ$	$0,33^\circ$
36°	$36,95^\circ$	$0,64^\circ$
45°	$44,20^\circ$	$0,93^\circ$

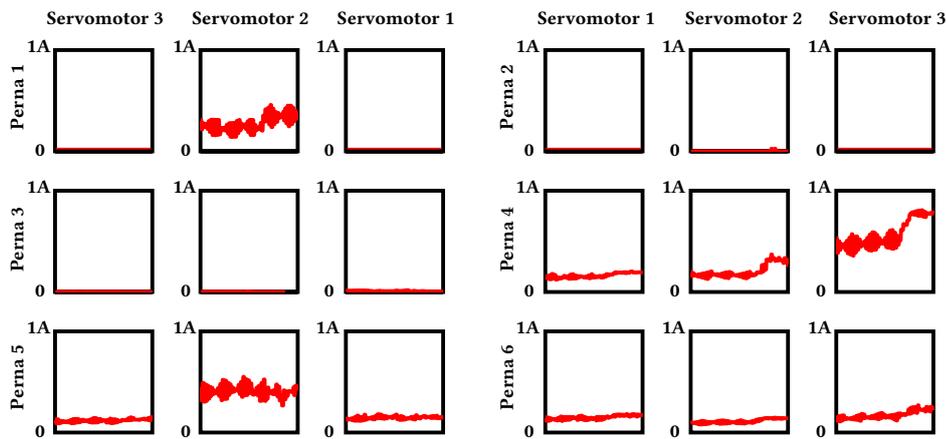
As medidas do sistema são satisfatoriamente próximas dos valores esperados, apresentando, em média, diferença menor que 1° em relação aos valores esperados e baixa dispersão — com desvio padrão também inferior a 1° . Os resultados são suficientes para validar o sistema em relação à sua aplicabilidade no monitoramento da sua orientação.

4.4 EXPERIMENTO 4 — SENSORIAMENTO DE CORRENTE

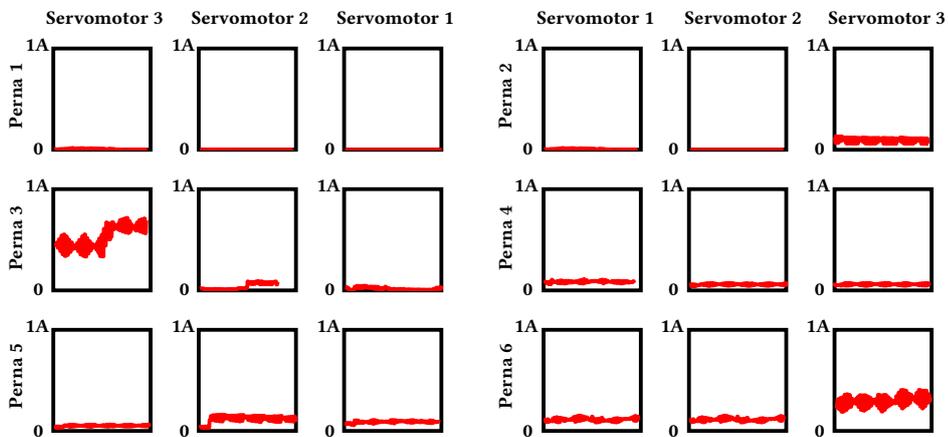
Introdução. O monitoramento da corrente através dos servomotores do robô é útil para estimar os torques gerados por estes. É possível a partir destas informações analisar o esforço local causado em cada articulação do robô, permitindo quantificar qualidade de padrões de marchas, efeitos de distribuições de carga, entre outros. O propósito deste experimento foi validar a aplicação do sistema desenvolvido no que diz respeito ao monitoramento das correntes nos servomotores do robô.

Materiais e métodos. O experimento foi feito com o robô e uma carga. As pernas do robô foram posicionadas em configuração de tripé, onde a estrutura é suportada por três das suas seis pernas: duas de um lado e uma de outro. A descrição deste posicionamento, bem como a numeração das pernas é a mesma utilizada na simulação do Capítulo 3 e descrita na Figura 9 (p. 30). As correntes consumidas por cada servomotor foram monitoradas em condição estática, com e sem a carga sobre a superfície superior do robô. Espera-se que as pernas de suporte apresentem maior consumo de corrente do que as suspensas, e que a adição de carga aumente o consumo destas correntes.

Resultados. As correntes consumidas pelos servomotores foram monitoradas por um intervalo de aproximadamente 20 segundos. A partir da metade deste intervalo, foi adicionada uma carga de 600 g na superfície superior do robô. As medições destas correntes foram feitas para as duas configurações de tripé possíveis: suporte nas pernas 1, 4, 5 e 2, 3, 6. Os resultados são apresentados na Figura 21.



(a) Pernas de suporte: 1, 4 e 5.



(b) Pernas de suporte: 2, 3 e 6.

Figura 21: Medição de correntes nos servomotores.

As correntes medidas são coerentes com o esperado, onde as pernas de suporte apresentam maior consumo de corrente em relação às pernas suspensas. O efeito da adição de carga também é bastante perceptível pelo crescimento de corrente, sendo mais acentuada no *servomotor 3* da perna que suporta um lado da estrutura sozinha: 4 na primeira configuração e 3 na segunda. Nota-se que os gráficos das duas configurações não são simétricos, muito provavelmente devido a diferenças entre as características elétricas e mecânicas dos servomotores e da própria estrutura. Os resultados validam a aplicabilidade do sistema no monitoramento das correntes consumidas pelos seus servomotores.

4.5 EXPERIMENTO 5 — ACESSO À VISÃO

Introdução. A capacidade de visão robótica viabiliza aplicações baseadas em inspeção, como detecção de falhas, monitoramento de ambientes, rastreamento de objetos, entre outros. No sistema proposto, o acesso remoto à visão permite processamento das imagens utilizando recursos de outros sistemas com maior poder computacional, viabilizando a aplicação de filtros e técnicas que estariam normalmente além das capacidades de hardware limitadas do robô. A visão do robô é isolada dos seus outros mecanismos, sendo implementado pelo software MJPG-streamer, que oferece acesso remoto às imagens capturadas pela câmera por HTTP. O propósito deste experimento foi validar a aplicação do sistema no que diz respeito à capacidade de acesso remoto à visão do robô. Diferentemente dos experimentos anteriores, este não apresenta uma análise quantitativa, caracterizando mais uma demonstração do que um experimento em si.

Materiais e métodos. Foi executado (em um computador na rede local) um *script* (disponível no Apêndice C) em Python/OpenCV para a captura das imagens da câmera do robô e detecção de um objeto nestas a partir da sua cor — azul. Cada quadro é convertido do espaço de cores RGB para o HSV, que é mais robusto a variações de luminosidade por possuir canais distintos para crominância e luminância. Em seguida é criada uma máscara com os pixels que possuem valor na faixa de matiz correspondente à cor azul. A máscara é submetida a uma abertura morfológica com elemento estruturante quadrado 5×5 para a remoção de ruído e uniformização dos objetos restantes. Dos pixels resultantes desta filtragem, seleciona-se o conjunto de pixels conectados entre si com a maior área, que é considerado como o objeto de interesse e tem o seu *bounding box* (retângulo mínimo que contém a sua área) sobreposto no quadro original.

Espera-se que seja possível acessar a visão do robô, processar as imagens providas desta e detectar um objeto com as características especificadas (i.e. azul), tudo isso a partir de outro computador da rede local.

Resultados. A Figura 22 apresenta o resultado da execução do *script* apresentado acima em um computador pessoal conectado à rede local. O *bounding box* do elemento detectado coincide com o redor do objeto de interesse, indicando que experimento de detecção foi bem-sucedido, validando o sistema proposto no que diz respeito à capacidade de visão por câmera.

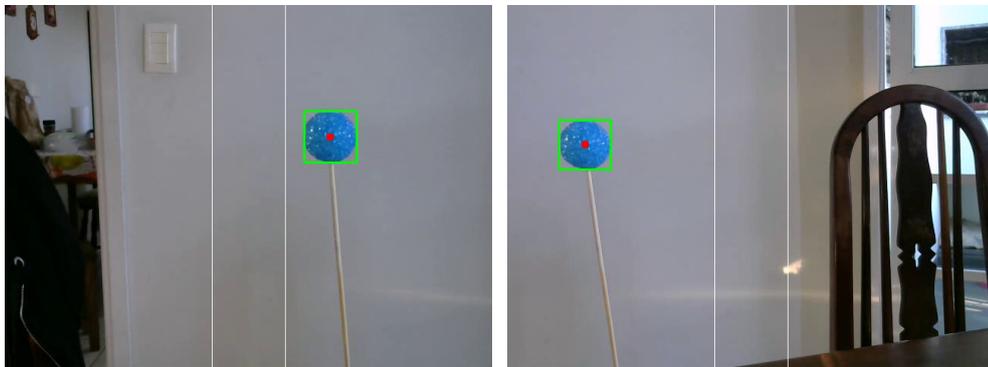


Figura 22: Experimento de acesso à visão — detecção de objeto.

4.6 DISCUSSÃO GERAL

Os experimentos descritos, apesar de simples, permitiram a validação dos requisitos propostos. Entretanto, foram notadas algumas características do sistema que anteriormente não eram explícitas. Primeiramente, a sincronização necessária entre os movimentos na marcha do robô torna inviável a sua implementação utilizando comandos individuais remotos aos atuadores. A latência da comunicação pela internet é intolerável, enquanto na rede local é relativamente tolerável, mas instável. Tanto em rede local quanto pela internet, o principal problema é a variabilidade entre o tempo de transmissão de cada mensagem (um atraso fixo não seria tão problemático), que adiciona atrasos não-determinísticos nos comandos enviados para os atuadores. Enquanto o tempo entre cada quadro de movimento da marcha do robô é 200 ms , a conexão, tanto pela rede local quanto pela internet, apresenta picos de RTT (*Round Trip Time*) com valor significativamente maior em relação ao tempo necessário para a marcha, atingindo valores próximos de 1000 ms . A Figura 23 apresenta o RTT para 30 requisições TCP na rede local e pela internet.

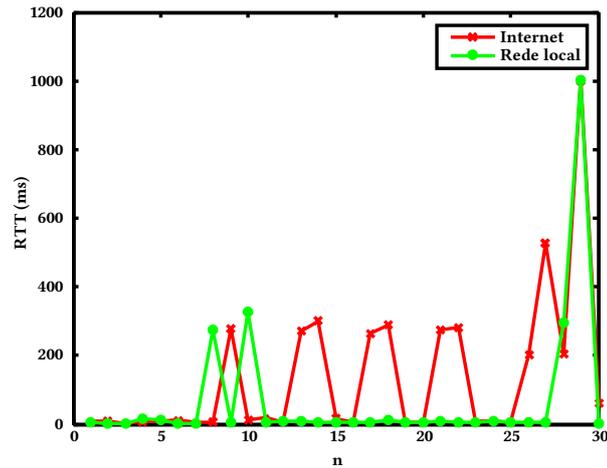


Figura 23: RTT pela rede local e pela internet.

A solução adotada para lidar com o atraso da rede foi mover os comandos que têm requisitos de tempo mais estritos para dentro do robô, isto é, em vez de diversas chamadas de comandos individuais são feitas chamadas individuais a sequências de comandos. Com isso não há atraso acumulado, somente o da chamada inicial. Isso diminui a aplicabilidade do sistema, visto que a granularidade com a qual se pode adaptar o comportamento do robô aumenta, entretanto, esta pode ser adaptada de acordo com a necessidade de cada caso. Uma solução alternativa possível, e menos intrusiva, seria o ajuste de qualidade de serviço na rede.

Outra característica observada durante os experimentos é que a distribuição de esforços causados pela carga do robô parecem não ser distribuídos somente entre os motores, mas também através da estrutura em acrílico. Apesar de parecer óbvio em retrospecto, não se esperava que o papel da estrutura fosse tão grande. A avaliação das correntes consumidas pelos motores demonstrou que a carga de 600 g utilizada no Experimento 4, significativo de acordo com as simulações apresentadas no Capítulo 3, causa uma variação muito pequena no consumo de corrente nos motores quando deslocada em diferentes posições do robô — diferentemente do esperado, que era um consumo de corrente proporcionalmente maior nos motores mais próximos da posição onde a carga foi adicionada. Isso sugere fortemente duas características (i) a estrutura em acrílico absorve parte considerável do esforço, o que apresenta a vantagem de o robô suportar cargas maiores do que as esperadas anteriormente, mas a desvantagem de que o esforço adicional é distribuído para um componente do sistema que não foi projetado para este propósito e sobre o qual tem-se pouco controle; (ii) o método de sensoriamento apresenta uma sensibilidade inadequada para capturar pequenas variações de esforço nos motores, o que limita a sua aplicabilidade, ainda mais considerando que as diferenças mecânicas e elétricas entre estes também interferem nas medições.

5 CONCLUSÕES

Com motivação nas questões levantadas pela Robótica Analítica, propusemos o desenvolvimento de um sistema de telerrobótica, com o fim de oferecer uma plataforma adequada para exploração do comportamento do robô com o seu ambiente por meio da avaliação quantitativa do seu estado. Tal motivação norteou a especificação dos requisitos do sistema, que foi abstraído em termos de três módulos: de locomoção e visão, de sensoriamento e interfaceamento, e de locomoção, cuja arquitetura e integração foram descritas. Em seguida foram apresentadas e exemplificadas as formas de acesso e controle a, respectivamente, os sensores e atuadores do robô. Por fim, foi descrita uma série de experimentos de validação, demonstrando que foram cumpridos os requisitos definidos durante o projeto do sistema.

O sistema desenvolvido cumpre com os objetivos especificados inicialmente, a saber: projetar e construir a estrutura física do robô (i.e. chassi, integração mecânica/eletrônica), desenvolver o *middleware* para acesso distribuído aos sensores e atuadores do robô, desenvolver os *device drivers* para interfaceamento com os sensores e atuadores, validar e experimentar com a plataforma desenvolvida. Entretanto, o sistema apresenta algumas limitações. Primeiramente, o atraso da sua operação por rede pode ser problemático dependendo da aplicação, como é o caso da marcha utilizada nos experimentos de validação, na qual os picos de atraso da rede são suficientes para dessincronizar os seus movimentos. Adotamos a solução de executar o código com requisitos estritos de tempo dentro do robô, que demonstrou ser o suficiente para os nossos propósitos. Outra limitação é a alimentação cabeada. Como a massa de uma bateria aumentaria a capacidade de torque necessária pelos motores e, conseqüentemente, o custo do sistema, optamos pela alimentação cabeada, que restringe consideravelmente a aplicação do sistema.

Considerando as limitações citadas, existe uma série de aprimoramentos possíveis para o sistema desenvolvido. O aprimoramento mais impactante seria a diminuição da massa do robô, que reduziria os requisitos dos servomotores, a área de acrílico e, conseqüentemente, o custo. Com a diminuição dos requisitos de torque do robô, o consumo de energia também diminuiria, viabilizando a utilização de uma bateria em vez da alimentação cabeada. É possível diminuir a área diminuindo primeiramente os circuitos utilizados. Isso é possível, por exemplo, utilizando um dispositivo lançado recentemente: o WrtNode (WRTNODE, 2016), que é uma pequena placa ($45\text{ mm} \times 50\text{ mm}$ vs. $180\text{ mm} \times 140\text{ mm}$ do roteador que adotamos) com comunicação Wi-Fi desenvolvida para utilização com o sistema operacional OpenWrt e com

ênfase em baixo consumo de energia.

Outro aprimoramento possível seria a análise e possível integração do sistema com outros *middleware* para robótica. Apesar de termos optado pela implementação de um *middleware* próprio, prezando pelo minimalismo, seria conveniente ao menos disponibilizar uma interface para comunicação com dispositivos executando o ROS, visto que este é o *middleware* para robótica com maiores chances de ser encontrado na prática.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARCBOTICS. *Hexy the Hexapod - Open Source Low Cost Fully Articulating DIY Robot Kit*. 2015. Acesso em 01/12/2015. Disponível em: <<http://arcbotics.com/products/hexy>>.
- BOSTON DYNAMICS. *BigDog - The Most Advanced Rough-Terrain Robot on Earth*. 2015. Acesso em 01/12/2015. Disponível em: <http://www.bostondynamics.com/robot_bigdog.html>.
- BROOKS, R. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, IEEE, v. 2, n. 1, p. 14–23, 1986.
- CLARK, J. E. et al. Biomimetic design and fabrication of a hexapedal running robot. In: *Proceedings of the IEEE International Conference on Robotics and Automation, 2001*. Seoul, Korea: IEEE, 2001. v. 4, p. 3643–3649.
- CRICK, C. et al. ROSbridge: ROS for non-ROS users. In: *Proceedings of the 15th International Symposium on Robotics Research*. Flagstaff, USA: Springer-Verlag Berlin Heidelberg, 2011. p. 1–12.
- DWORAK, A.; SLIWINSKI, W.; SOBCZAK, M. Middleware trends and market leaders 2011. In: *Proceedings of the 13th International Conference on Accelerator and Large Experimental Physics Control Systems*. Grenoble, France: JACoW, 2011.
- ELKADY, A.; SOBH, T. Robotics middleware: A comprehensive literature survey and attribute-based bibliography. *Journal of Robotics*, Hindawi Publishing Corporation, 2012.
- FU, K. S.; GONZALEZ, R.; LEE, C. *Robotics: Control, Sensing, Vision, and Intelligence*. New York, USA: Tata McGraw-Hill Education, 1988.
- FUTABA. *Futaba Standard Analog Servos*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.futabarc.com/servos/analog.html>>.
- INVENSENSE. *MPU6050 | InvenSense*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>>.
- IROBOT. *iRobot Roomba Vacuum Cleaning Robot*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.irobot.com/For-the-Home/Vacuum-Cleaning/Roomba.aspx>>.
- ITEAD STUDIO. *Ultrasonic Ranging Module HC-SR04*. 2015. Acesso em 01/12/2015. Disponível em: <http://wiki.iteadstudio.com/Ultrasonic_Ranging_Module_HC-SR04>.
- ITTI, L.; KOCH, C.; NIEBUR, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, IEEE, n. 11, p. 1254–1259, 1998.
- IYENGAR, S. S.; ELFES, A. *Autonomous Mobile Robots: Control, Planning, and Architecture*. Los Alamitos, USA: IEEE Computer Society Press, 1991.
- JANSEN, T. *STRANDBEEST*. 2016. Acesso em 20/01/2016. Disponível em: <<http://www.strandbeest.com>>.
- JONES, J. L.; FLYNN, A. M. *Mobile robots: inspiration to implementation*. Natick, USA: AK Peters, Ltd., 1993.

K-TEAM. *Khepera IV*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.k-team.com/mobile-robotics-products/khepera-iv>>.

KLEPPNER, D.; KOLENKOW, R. *An Introduction to Mechanics*. Cambridge, UK: McGraw-Hill Kogakusha, 1973.

KULICH, M. et al. SyRoTek - Distance teaching of mobile robotics. *IEEE Transactions on Education*, IEEE, v. 56, n. 1, p. 18–23, 2013.

LOGITECH. *HD Webcam C270 Simple 720p video calls*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.logitech.com/en-ca/product/hd-webcam-c270>>.

MATARIC, M. J.; KOENIG, N. P.; FEIL-SEIFER, D. Materials for Enabling Hands-On Robotics and STEM Education. In: *AAAI Spring Symposium: Semantic Scientific Knowledge Integration*. Palo Alto, USA: AAAI, 2007. p. 99–102.

MATHWORKS. *Multibody Simulation - SimMechanics - Simulink*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.mathworks.com/products/simmechanics/>>.

MATTEUCCI, M. Publish/subscribe middleware for robotics: requirements and state of the art. *Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, Tech. Rep. Technical Report*, 2003.

MJPEG-STREAMER. *MJPEG-streamer download - sourceForge.net*. 2014. Acesso em 01/12/2015. Disponível em: <<http://sourceforge.net/projects/mjpg-streamer/>>.

MONDADA, F.; FRANZI, E.; GUIGNARD, A. The Development of Khepera. In: *Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop*. [S.l.: s.n.], 1999. p. 7–14.

NANOMSG. 2015. Acesso em 01/12/2015. Disponível em: <<http://nanomsg.org>>.

NASA. *Rover - Mars Science Laboratory*. 2015. Acesso em 17/09/2015. Disponível em: <<http://mars.nasa.gov/msl/mission/rover/>>.

NEHMZOW, U. *Scientific Methods in Mobile Robotics: quantitative analysis of agent behaviour*. London, UK: Springer Science & Business Media, 2006.

NEHMZOW, U. *Robot behaviour*. London, UK: Springer, 2009.

NITTO, E. D.; ROSENBLUM, D. S. On the role of style in selecting middleware and underwear. *Proceedings of the ICSE '99 Workshop on Engineering Distributed Objects*, v. 99, p. 78–83, 1999.

OPENCV. 2016. Acesso em 08/03/2016. Disponível em: <<http://opencv.org/>>.

OPENWRT. *Barrier Braker [OpenWrt Wiki]*. 2015. Acesso em 01/12/2015. Disponível em: <<http://wiki.openwrt.org/doc/barrier.breaker>>.

OPENWRT. *TP-Link TL-WDR3600 [OpenWrt Wiki]*. 2015. Acesso em 01/12/2015. Disponível em: <<https://wiki.openwrt.org/toh/tp-link/tl-wdr3600>>.

OPENWRT. *About OpenWrt [OpenWrt Wiki]*. 2016. Acesso em 04/02/2016. Disponível em: <<https://wiki.openwrt.org/about/start>>.

- POLOLU. *Mini Maestro 24-Channel USB Servo Controller (Assembled)*. 2015. Acesso em 01/12/2015. Disponível em: <<https://www.pololu.com/product/1356>>.
- PORTA, J. M.; CELAYA, E. *Gait Analysis for Six-Legged Robots*. Barcelona, Spain, 1998.
- QUIGLEY, M. et al. ROS: an open-source Robot Operating System. In: *ICRA workshop on open source software*. Kobe, Japan: IEEE, 2009. v. 3, p. 5.
- RAIBERT, M. H. Legged robots. *Communications of the ACM*, v. 29, n. 6, p. 499–514, 1986.
- REAL TIME ENGINEERS. *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.freertos.org/>>.
- SHERIDAN, T. B. Telerobotics. *Automatica*, Elsevier, v. 25, n. 4, p. 487–507, 1989.
- SHERIDAN, T. B. Teleoperation, telerobotics and telepresence: A progress report. *Control Engineering Practice*, Elsevier, v. 3, n. 2, p. 205–214, 1995.
- SMART, W. D. Is a common middleware for robotics possible? In: *Proceedings of the IROS 2007 Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware*. San Diego, USA: IEEE, 2007.
- SONY. *Sony Launches Four-Legged Entertainment Robot "AIBO"*. 1999. Acesso em 17/09/2015. Disponível em: <<http://www.sony.net/SonyInfo/News/PressArchive/199905/99-046/index.html>>.
- TEXAS INSTRUMENTS. *ARM Cortex-M4F Based MCU TM4C123G LaunchPad Evaluation Kit - EK-TM4C123GXL*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.ti.com/tool/ek-tm4c123gxl>>.
- TEXAS INSTRUMENTS. *INA213 | Current Shunt Analog Output | Current Sense Amplifiers | Description & parametrics*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.ti.com/product/INA213>>.
- TOWERPRO. *MG995 Robot servo 180° Rotation | TowerPro*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.towerpro.com.tw/product/mg995-robot-servo-180-rotation-2/>>.
- TP-LINK. *N600 Wireless Dual Band Gigabit Router TL-WDR3600 - Welcome to TP-LINK*. 2013. Acesso em 01/12/2015. Disponível em: <http://www.tp-link.com/en/products/details/cat-9_TL-WDR3600.html>.
- TRIBELHORN, B.; DODDS, Z. Evaluating the roomba: A low-cost, ubiquitous platform for robotics research and education. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Roma, Italy: IEEE, 2007. p. 1393–1399.
- TROSSENROBOTICS. *PhantomX AX Metal Hexapod Mark III*. 2015. Acesso em 01/12/2015. Disponível em: <<http://www.trossenrobotics.com/phantomx-ax-hexapod.aspx>>.
- VIEIRA NETO, H. *Visual novelty detection for autonomous inspection robots*. Tese (Doutorado) – University of Essex, 2006.
- WALTER, W. G. An Imitation of Life. *Scientific American*, v. 182, p. 42–45, 1950.

WIGNER, E. P. The unreasonable effectiveness of mathematics in the natural sciences. *Communications on pure and applied mathematics*, Wiley Online Library, v. 13, n. 1, p. 1–14, 1960.

WRTNODE. 2016. Acesso em 09/03/2016. Disponível em: <<http://wrtnode.com/>>.

A LAYOUT DA ESTRUTURA FÍSICA

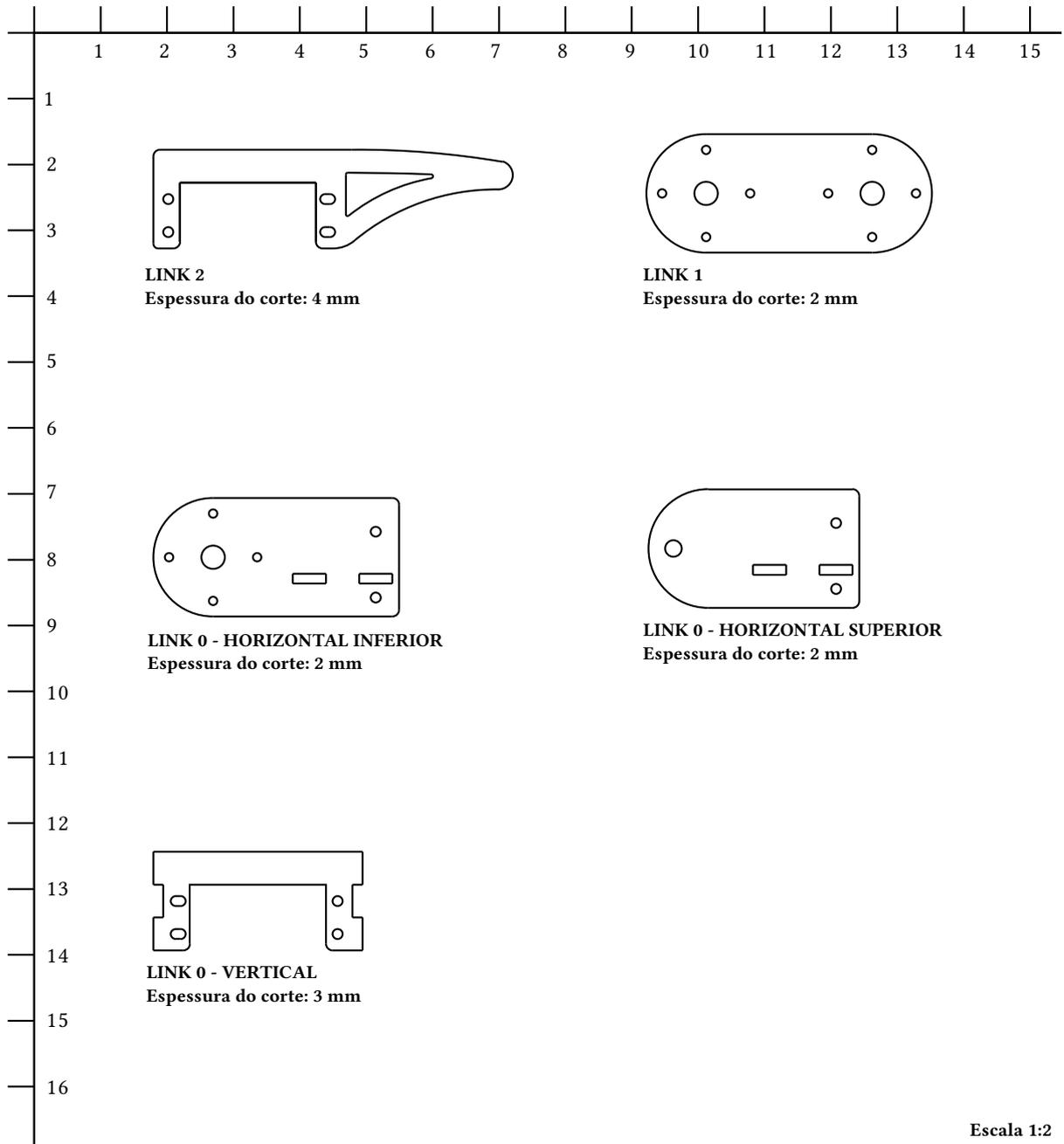


Figura 24: *Layout do Módulo de Locomoção – Junções (links) para uma perna.*

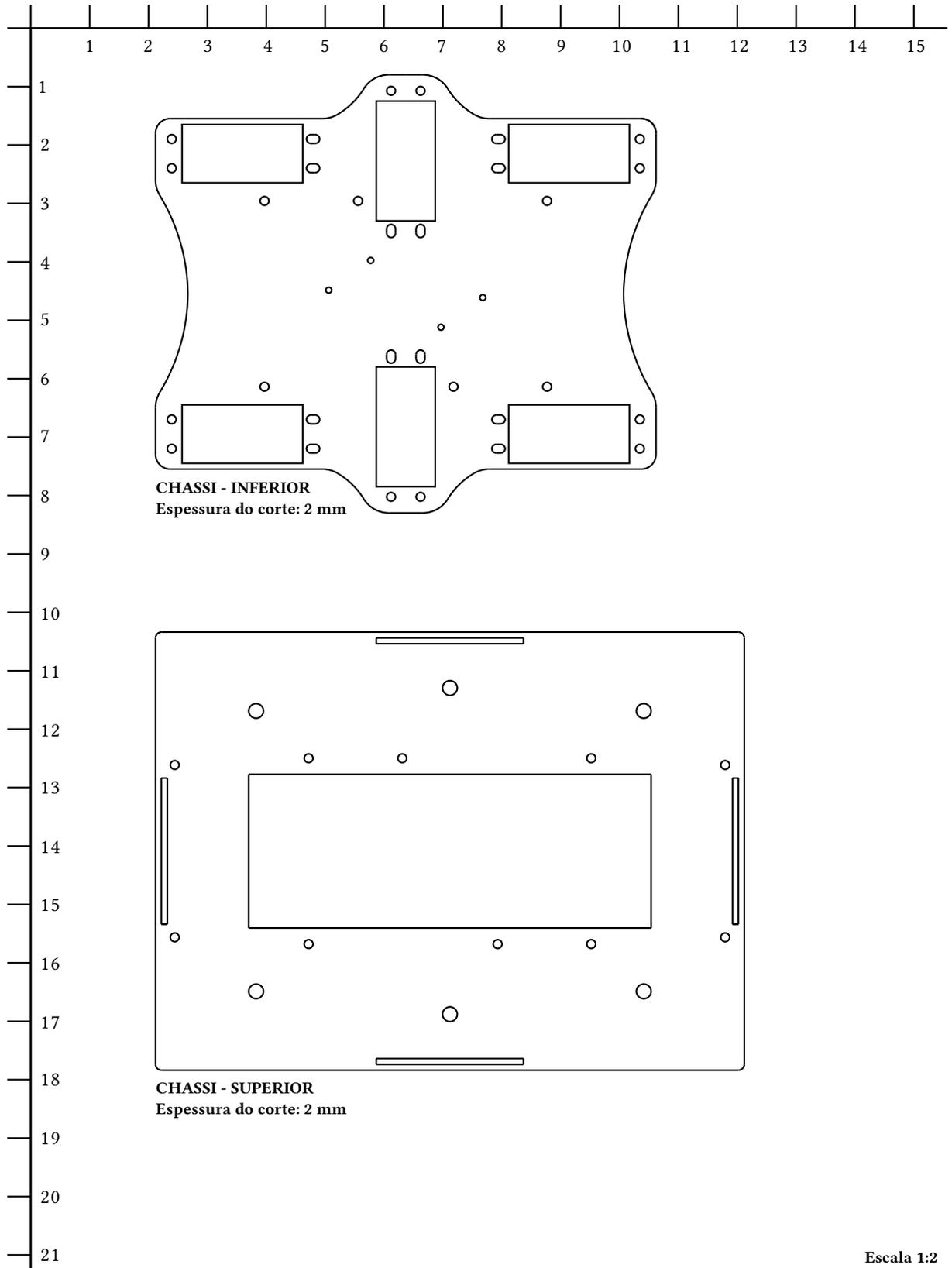
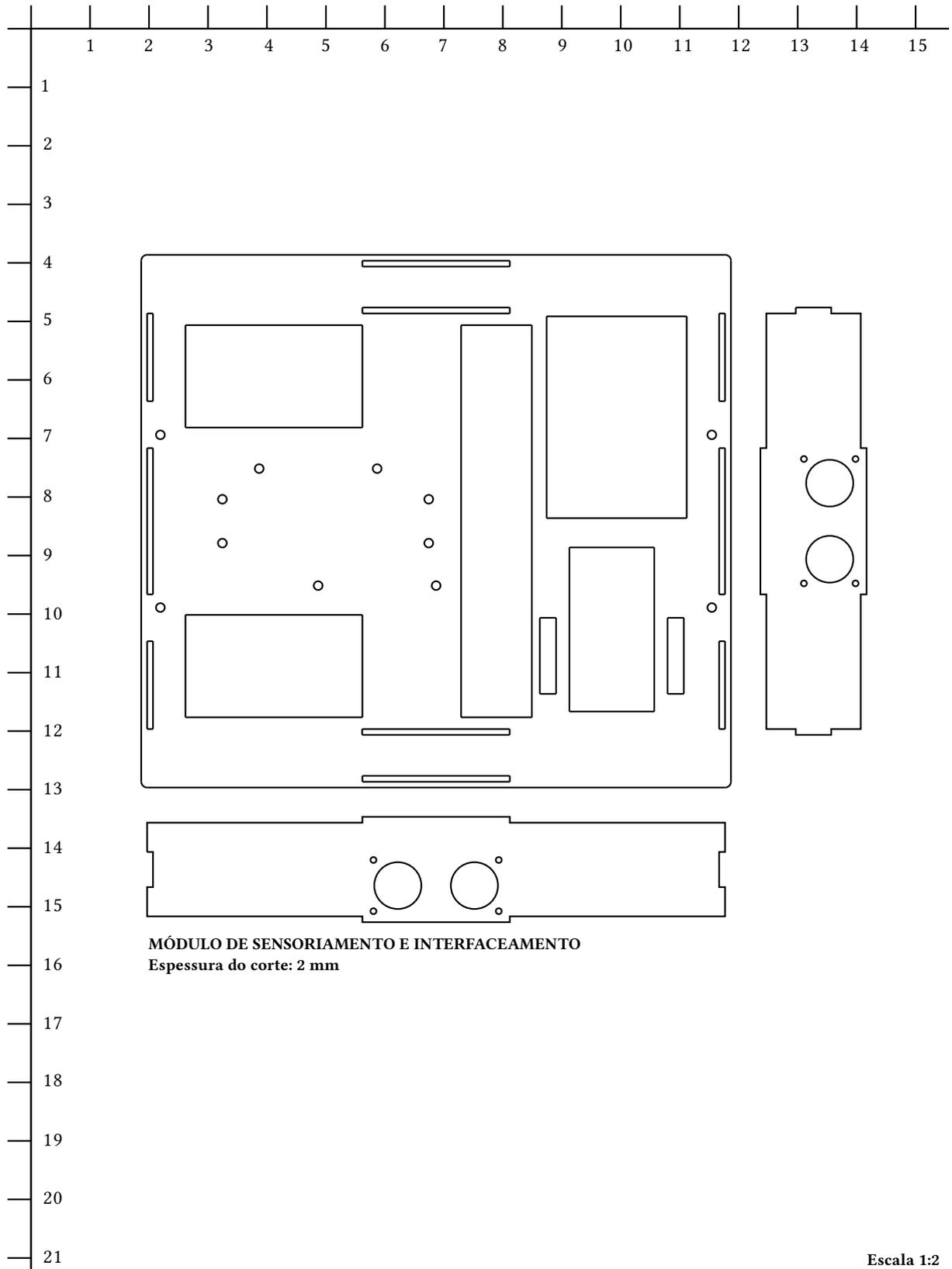


Figura 25: Layout do Módulo de Locomoção – Tronco.



Escala 1:2

Figura 26: *Layout do Módulo de Sensoriamento e Interfaceamento.*

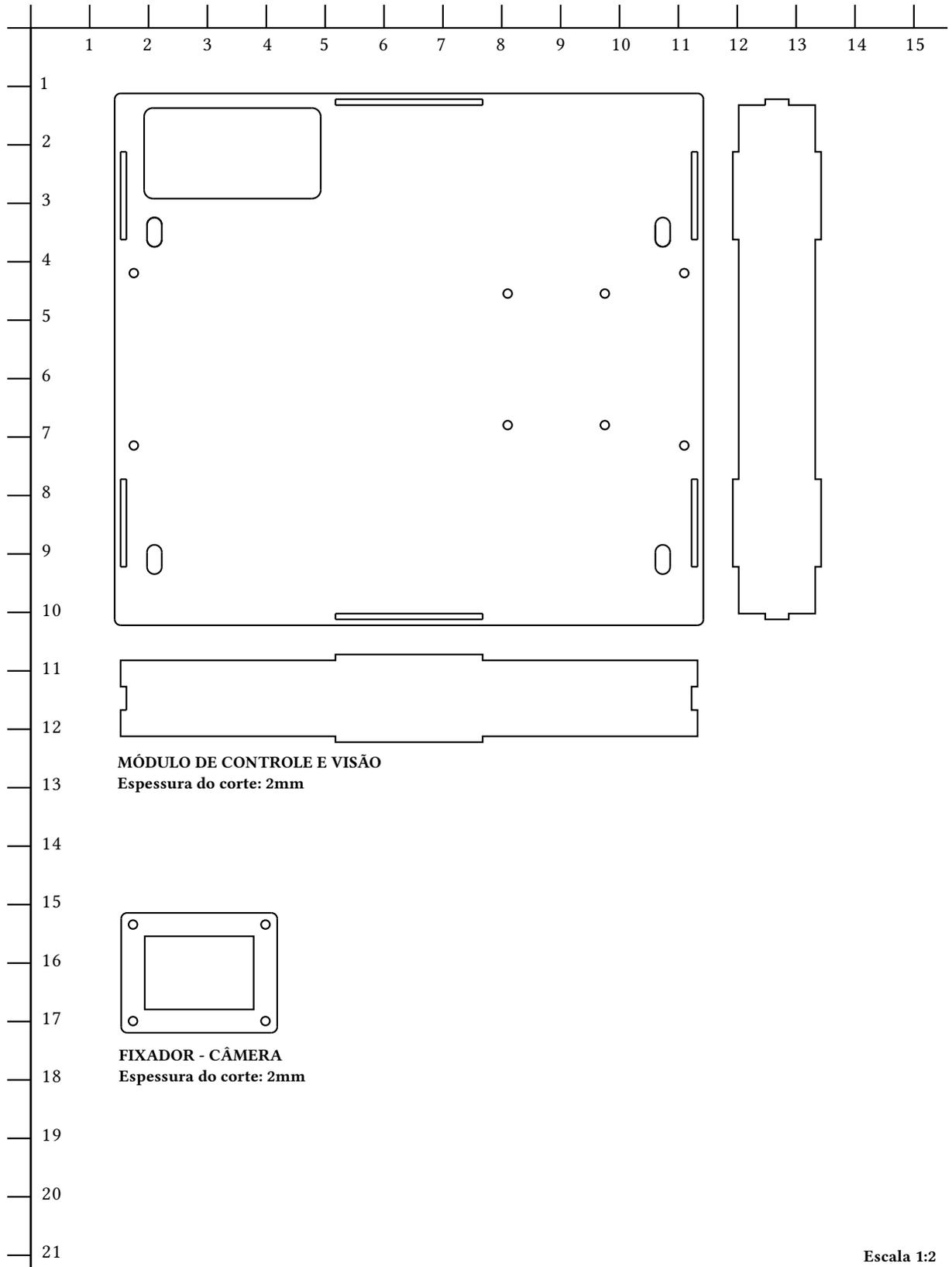


Figura 27: Layout do Módulo de Comunicação e Visão.

B PLACAS - DIAGRAMA ESQUEMÁTICO, LAYOUT E FOTOS

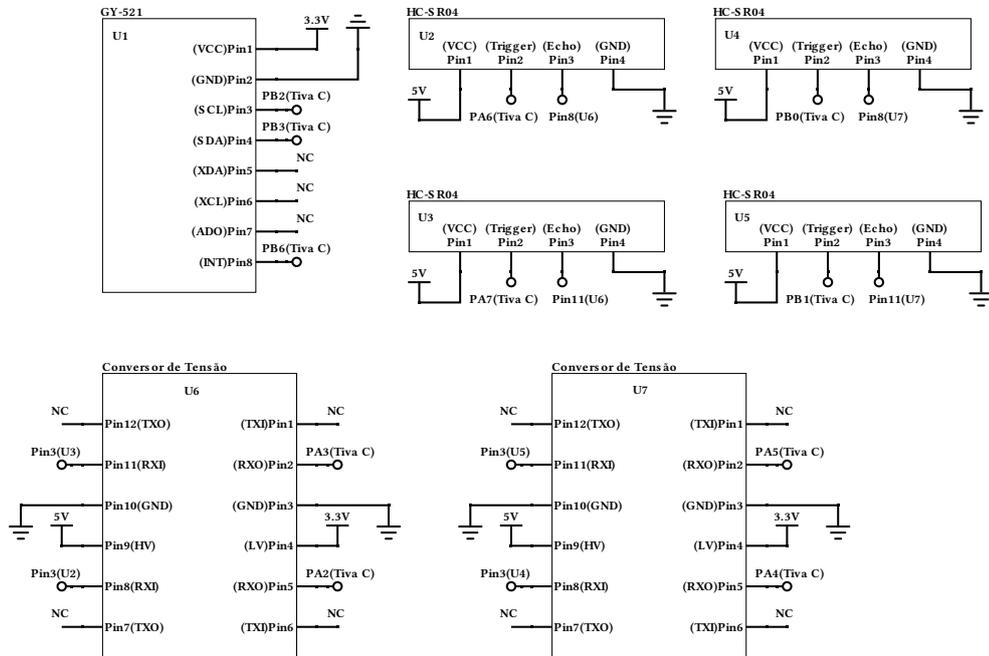


Figura 28: Diagrama esquemático – Placa de sensoriamento.

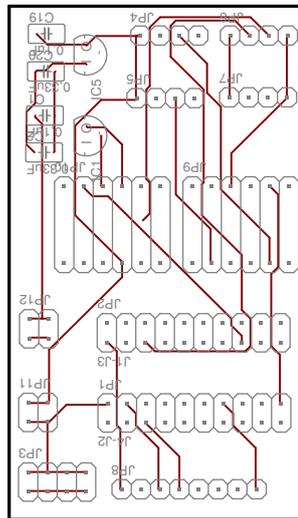


Figura 29: Layout da placa de circuito impresso (face superior) – Placa de sensoriamento.

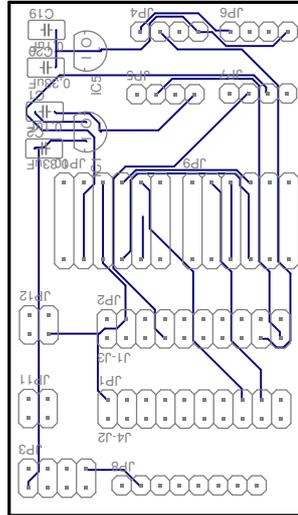


Figura 30: *Layout* da placa de circuito impresso (face inferior) – Placa de sensoriamento.

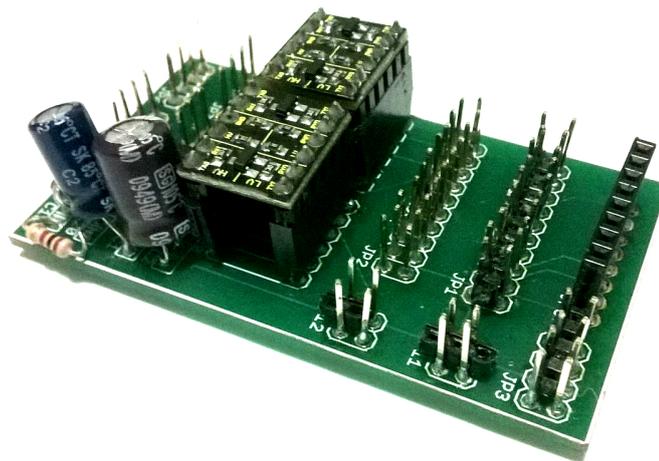


Figura 31: Placa de sensoriamento.

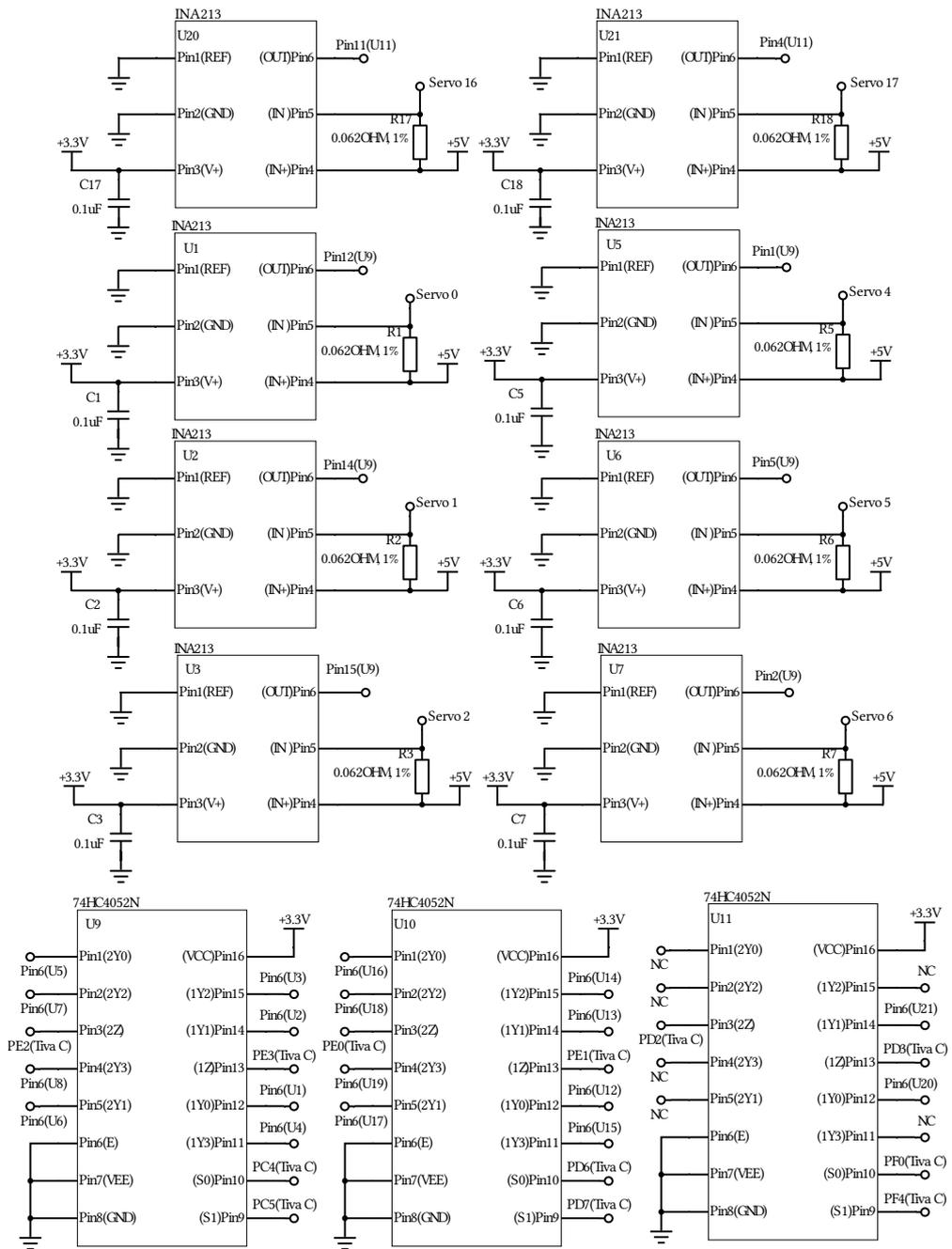


Figura 32: Diagrama esquemático – Placa de locomoção.

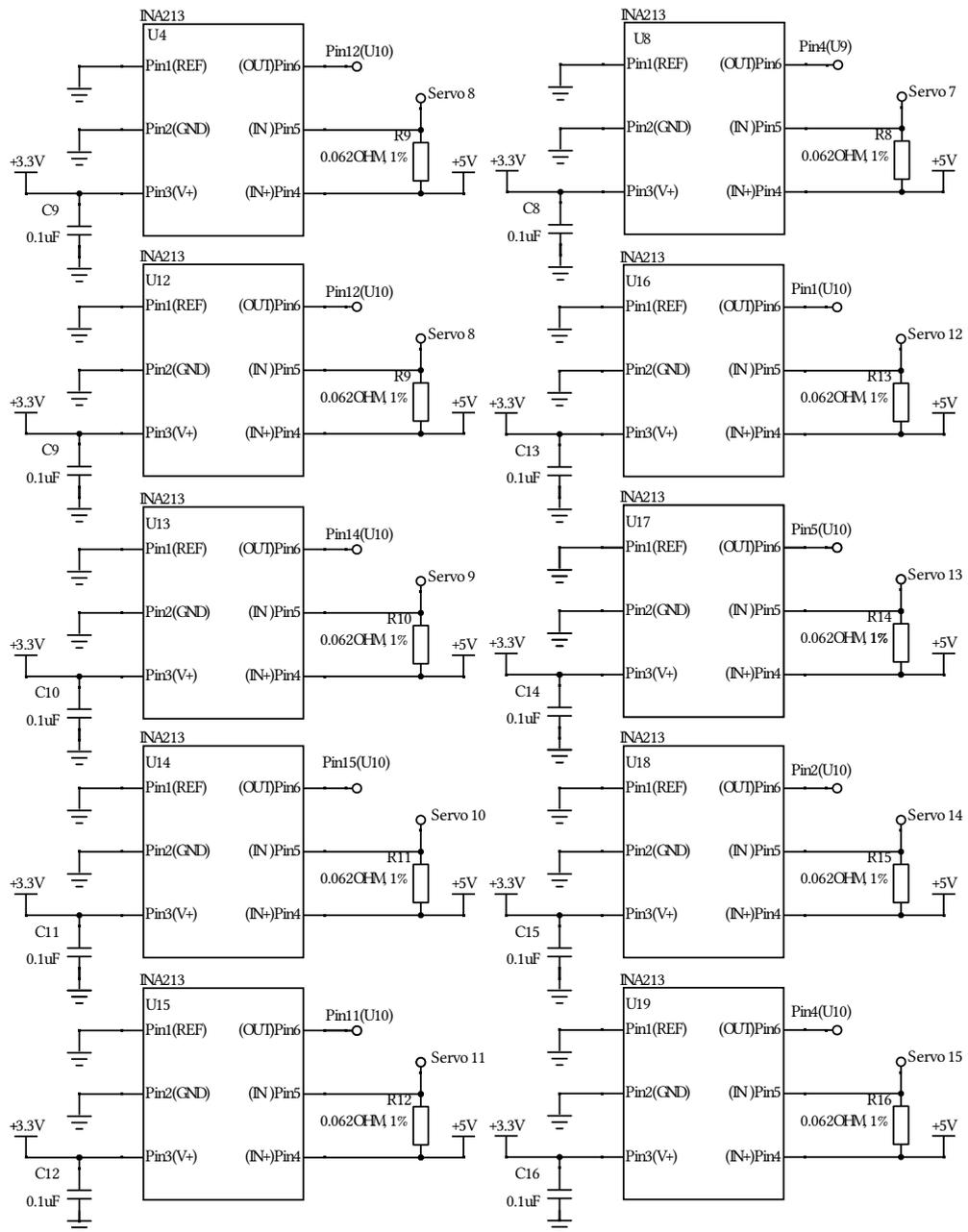


Figura 33: Diagrama esquemático – Placa de locomoção (continuação).

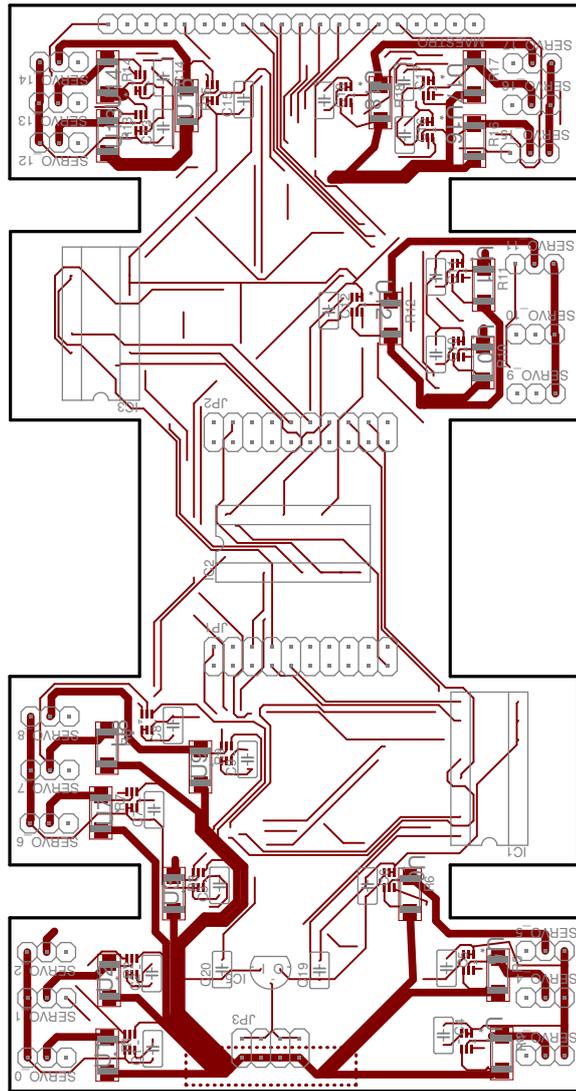


Figura 34: *Layout* da placa de circuito impresso (face superior) — Placa de locomoção.

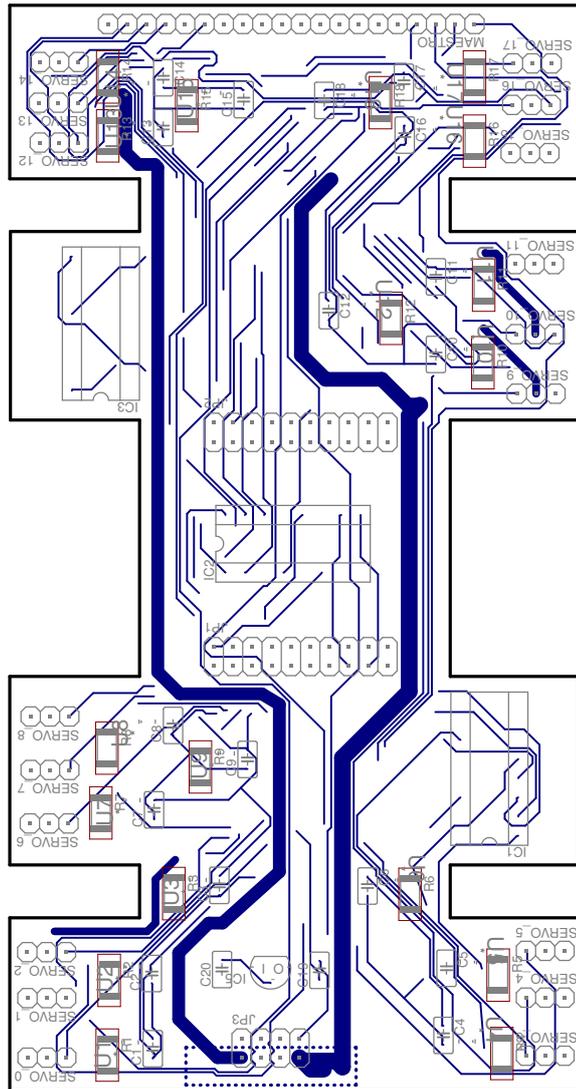


Figura 35: *Layout* da placa de circuito impresso (face inferior) – Placa de locomoção.

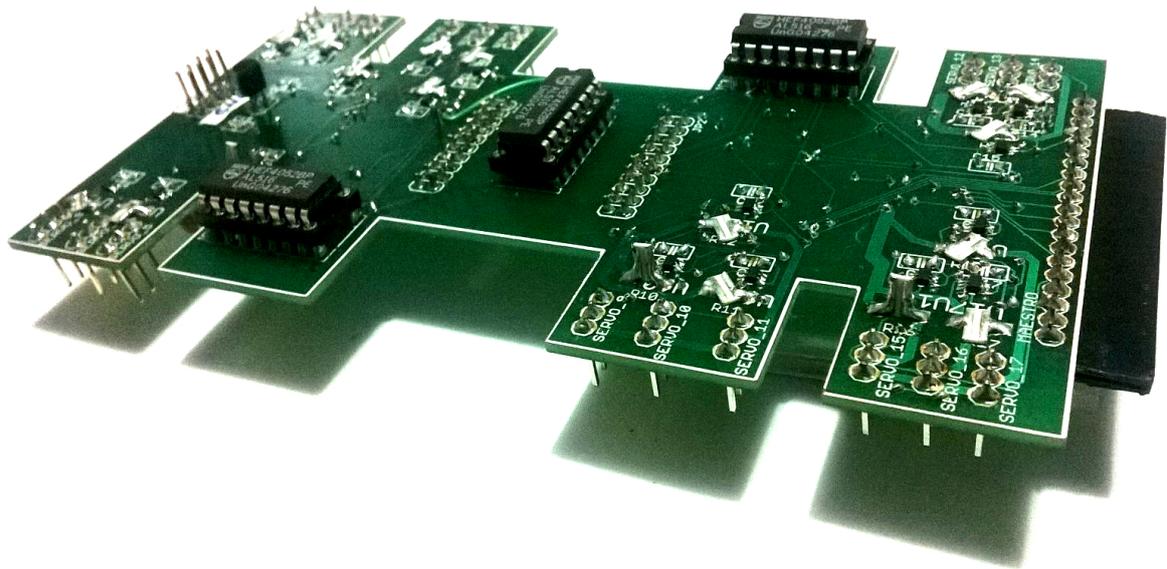


Figura 36: Placa de locomoção.

C SCRIPT DE DETECÇÃO DE OBJETO

```

1 import cv2
2 import numpy as np
3
4 STREAM_ADDR = 'http://192.168.0.16:8080/?action=stream?.mjpg'
5
6 # Instancia captura de video por streaming.
7 cap = cv2.VideoCapture(STREAM_ADDR)
8
9 while cap.isOpened():
10     # Captura quadro.
11     ret, frame_rgb = cap.read()
12
13     # Converte o quadro para o espaco de cores HSV (mais adequado que RGB)
14     # e cria mascara com objetos azuis.
15     frame_hsv = cv2.cvtColor(frame_rgb, cv2.COLOR_BGR2HSV)
16     lower = np.array([100,100,100])
17     upper = np.array([130,255,255])
18     mask = cv2.inRange(frame_hsv, lower, upper)
19
20     # Aplica abertura morfologica na mascara - remove pequenos objetos
21     # e uniformiza grandes objetos.
22     kernel = np.ones((5, 5), np.uint8)
23     thresh = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
24
25     # Seleciona o maior componente conectado e desenha o seu bounding box.
26     thresh, contours, hier = cv2.findContours(thresh, cv2.RETR_TREE,
27                                             cv2.CHAIN_APPROX_SIMPLE)
28
29     largest_area = 0
30     largest_index = 0;
31     area = 0
32     for i in range(0, len(contours)):
33         area = cv2.contourArea(contours[i])
34         if area > largest_area:
35             largest_area = area
36             largest_index = i
37     cnt = contours[largest_index]
38     x,y,w,h = cv2.boundingrect(cnt)
39     cv2.rectangle(frame_rgb, (x, y), (x + w, y + h), (0, 255, 0), 2)
40     cv2.circle(frame_rgb, (x+w/2, y+h/2), 5, (0, 0, 255), -1)
41
42     # Exibe o quadro atual.
43     cv2.imshow('Experimento 5', frame_rgb)
44     if cv2.waitKey(1) & 0xFF == ord('q'):
45         break
46
47 cap.release()
48 cv2.destroyAllWindows()

```


4. Especificações

Parâmetro	Mínimo	Típico	Máximo	Unidade
Sensibilidade do sonar ¹	2	—	400	cm
Sensibilidade do acelerômetro ²	-2 g	—	+2 g	m/s ²
Sensibilidade do giroscópio ²	-500	—	+500	°/s
Taxa máxima de transmissão de dados (Wi-Fi) ³	—	—	300	Mbps
Tensão de alimentação (DC)	—	6	—	V
Massa total	—	2700	—	g

¹ http://www.electfreaks.com/store/download/product/Sensor/HC-SR04/HC-SR04_Ultrasonic_Module_User_Guide.pdf

² <http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>

³ <http://www.tp-link.com/en/products/details/TL-WDR3600.html>

5. Operação

O acesso aos sensores e atuadores da plataforma é feito por trocas de mensagens utilizando a biblioteca nanomsg⁴. As mensagens são cadeias de caracteres ASCII, inicializados por '#' e terminados por '!', seguindo os formatos especificados nas tabelas a seguir.

- Comandos

Dispositivo	Mensagem		
	ID	OPCODE	DATA
Servomotores	0 - 23	0 - Altera ângulo	Ângulo
	0 - 23	1 - Altera velocidade	Velocidade
	0 - 23	2 - Altera aceleração	Aceleração
	0 - 23	3 - Lê corrente	—
Giroscópio	24	0 - Lê vel. angular	—
Acelerômetro	24	1 - Lê aceleração	—
Sonar	25 - 28	0 - Lê distância	—

- Monitoramento

Dispositivo	Mensagem							
	ID	OPCODE	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5
Servomotores	0 - 23	0 - Corrente	i_{int}	i_{frac}	—	—	—	—
Giroscópio	24	0 - Vel. angular	x_{int}	x_{frac}	y_{int}	y_{frac}	z_{int}	z_{frac}
Acelerômetro	24	1 - Aceleração	x_{int}	x_{frac}	y_{int}	y_{frac}	z_{int}	z_{frac}
Sonar	25 - 28	0 - Distância	d	—	—	—	—	—

⁴ <http://nanomsg.org/>

6. Exemplo de operação

Os exemplos seguintes utilizam o `nanocat`, um aplicativo que acompanha a biblioteca `nanomsg`, e consideram que o robô está na rede local com o endereço IP `192.168.1.1`. O envio de comando a atuadores é feito na porta `9000`, o monitoramento de sensores pela porta `9001`, e o acesso à visão pela porta `8080`.

- Comando `'000'` com argumento `'000'` ao atuador com ID `'003'`:

```
nanocat --req --data "#003000000!" --connect tcp://192.168.1.1:9000 --ascii
```

- Monitoração de um sensor com ID `'015'`:

```
nanocat --sub --subscribe "#015" --connect tcp://192.168.1.1:9001 --ascii
```

- Para acesso à visão do robô basta acessar o seu endereço na porta `8080` utilizando algum navegador web ou algum cliente de *streaming* com suporte para *stream* MJPG.