

INSTITUTO FEDERAL DO RIO GRANDE DO NORTE
CAMPUS NATAL - CENTRAL
DIRETORIA DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Título do trabalho

Jamillo G. da S. Santos

Natal-RN
Abril 2017

Jamillo G. da S. Santos

Título

Trabalho de conclusão de curso de graduação do curso de Tecnologia e Análise em Desenvolvimento de Sistemas da Diretoria de Gestão e Tecnologia de Informação do Instituto Federal do Rio Grande do Norte como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Linha de pesquisa:

Nome da linha de pesquisa

Orientador

Prof. Dr. Eduardo Braulio

TADS – CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DIATINF – DIRETORIA ACADÊMICA DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO
CNAT – CAMPUS NATAL - CENTRAL
IFRN – INSTITUTO FEDERAL DO RIO GRANDE DO NORTE

Natal-RN

Abril 2017

Trabalho de Conclusão de Curso de Graduação sob o título *Título* apresentada por Nome completo do autor e aceita pelo Diretoria de Gestão e Tecnologia da Informação do Instituto Federal do Rio Grande do Norte, sendo aprovada por todos os membros da banca examinadora abaixo especificada:

Nome completo do orientador e titulação

Presidente

DIATINF – Diretoria Acadêmica de Gestão e Tecnologia da
Informação

IFRN – Instituto Federal do Rio Grande do Norte

Nome completo do examinador e titulação

Examinador

Diretoria/Departamento

Instituto

Nome completo do examinador e titulação

Examinador

Diretoria/Departamento

Universidade

Natal-RN, data da defesa (dia, mês e ano).

Homenagem que o autor presta a uma ou mais pessoas.

Agradecimentos

Agradecimentos dirigidos àqueles que contribuíram de maneira relevante à elaboração do trabalho, sejam eles pessoas ou mesmo organizações.

Citação

Autor

Título do trabalho

Autor: Jamillo G. da S. Santos

Orientador(a): Prof. Dr. Eduardo Braulio

RESUMO

O resumo deve apresentar de forma concisa os pontos relevantes de um texto, fornecendo uma visão rápida e clara do conteúdo e das conclusões do trabalho. O texto, redigido na forma impessoal do verbo, é constituído de uma sequência de frases concisas e objetivas e não de uma simples enumeração de tópicos, não ultrapassando 500 palavras, seguido, logo abaixo, das palavras representativas do conteúdo do trabalho, isto é, palavras-chave e/ou descritores. Por fim, deve-se evitar, na redação do resumo, o uso de parágrafos (em geral resumos são escritos em parágrafo único), bem como de fórmulas, diagramas e símbolos, optando-se, quando necessário, pela transcrição na forma extensa, além de não incluir citações bibliográficas.

Palavras-chave: Palavra-chave 1, Palavra-chave 2, Palavra-chave 3.

Título do trabalho (em língua estrangeira)

Author: Jamillo G. da S. Santos

Supervisor: Prof. Doc. Eduardo Braulio

ABSTRACT

O resumo em língua estrangeira (em inglês *Abstract*, em espanhol *Resumen*, em francês *Résumé*) é uma versão do resumo escrito na língua vernícula para idioma de divulgação internacional. Ele deve apresentar as mesmas características do anterior (incluindo as mesmas palavras, isto é, seu conteúdo não deve diferir do resumo anterior), bem como ser seguido das palavras representativas do conteúdo do trabalho, isto é, palavras-chave e/ou descritores, na língua estrangeira. Embora a especificação abaixo considere o inglês como língua estrangeira (o mais comum), não fica impedido a adoção de outras linguas (a exemplo de espanhol ou francês) para redação do resumo em língua estrangeira.

Keywords: Keyword 1, Keyword 2, Keyword 3.

Lista de figuras

1	Teste de uma figura em formato .png	p. 18
2	Visão simplificada da arquitetura do sistema e seus fluxos.	p. 19
3	Diagrama de sequência do componente de visão.	p. 20
4	Diagrama de sequência do controlador de comportamento e suas dependências.	p. 21
5	Diagrama de sequência do <i>walking gait</i> e do leitor de orientação.	p. 22
6	Esquema de orientação dos atuadores de Arash	p. 23
7	Sentido do movimentos dos atuadores	p. 23

Lista de tabelas

1	Tabela sem sentido.	p. 26
---	-----------------------------	-------

Lista de abreviaturas e siglas

IFRN – Instituto Federal do Rio Grande do Norte

DIATINF – Diretoria Acadêmica de Gestão e Tecnologia da Informação

Sumário

1	Introdução	p. 13
1.1	Metodologia	p. 16
1.2	Organização do trabalho	p. 16
2	Capítulo 2	p. 17
2.1	Seção 1	p. 18
2.2	Seção 2	p. 18
2.3	Seção 3	p. 18
2.3.1	Subseção dentro da seção 3	p. 18
2.3.2	Subseção dentro da seção 3	p. 18
2.4	Seção 4	p. 18
3	Arquitetura	p. 19
3.1	Visão	p. 20
3.1.1	Graus de Liberdade e Atuadores	p. 22
3.2	Arquitetura de Software pré-solução	p. 24
3.3	Modificações na <i>OpenCM9.04</i>	p. 24
3.4	Seção 1	p. 26
3.5	Seção 2	p. 26
3.5.1	Subseção 2.1	p. 26
3.5.2	Subseção 2.2	p. 26
3.6	Seção 3	p. 27

3.7	Seção 4	p. 27
4	Capítulo 4	p. 28
4.1	Seção 1	p. 28
4.2	Seção 2	p. 28
5	Capítulo 5	p. 29
5.1	Seção 1	p. 29
5.2	Seção 2	p. 29
5.3	Seção 3	p. 29
6	Considerações finais	p. 30
6.1	Principais contribuições	p. 30
6.2	Limitações	p. 30
6.3	Trabalhos futuros	p. 30
	Apêndice A – Primeiro apêndice	p. 31
	Anexo A – Primeiro anexo	p. 32
	Todo list	p. 33

1 Introdução

GUIA:

- Contexto
 - Motivação
 - O problema
 - Soluções similares
 - Solução proposta
-

DRAFT: Nos dias atuais, a robótica está cada vez mais avançada e acessível. Seja em um pequeno robô aspirador de pó autônomo - que até mesmo retorna à estação de recarregamento quando sua bateria está fraca - ou nos sofisticados robôs que automatizam a fabricação de carros, aumentando a produção e diminuindo o custo. E eles vem em todos os tamanhos e formas.

Vivemos em um mundo feito e projetado para humanos. De escadas, portas, e até ferramentas como furadeiras, ou até o mouse são pensados para o uso diário de uma forma humanoide. Desta forma, nada mais obvio que projetar um robô humanoide que adapte-se de forma natural a este ambiente. Porém, a forma humanoide impõe diversos desafios de controle, basta uma breve busca no YouTube por "darpa challange fails".

GUIA:

Falar das dificuldades na robótica humanoide e fazer uma contextualização da caminhada citando alguns trabalhos recentes.

Normali-
to-
das
as
no-
men-
cla-
tu-
ras
do
wal-
king
gait,
as
ve-
zes
chama-
se
sis-
tema,

Em um esforço mútuo, as universidades *Amirkabir University of Technology* (AUT), do Irã, e a *University of Manitoba* (UofM), do Canadá, trabalham juntas para participar da Robocup - competição internacional de futebol de robôs cujo o objetivo é derrotar a seleção campeã mundial na copa de 2050. Nas edições 2015 e 2016, realizada em Hefei (China) e Leipzig (Alemanha), o sistema que controlou a caminhada de *Arash*; um robô humanóide de 100cm de altura rendeu o 3º lugar na modalidade de futebol em ambos os anos. Também, os 2º e 1º lugar no modalidade do desafio técnico em 2015 e 2016.

Desenvolvido utilizando placa microcontroladora *OpenCM 9.04*, compatível com Arduino, e batizado de *AUT-UofM-Walk-Engine*, o sistema de caminhada foi um dos grandes responsáveis pelo bom desempenho. Entretanto, o fato de rodar dentro da *OpenCM9.04* traz uma série de consequências não desejáveis ao palco.

Inicialmente, sendo um sistema complexo, a caminhada deve ser configurada para cada tipo de superfície. As configurações de uma superfície lisa e áspera, como concreto, podem ser bastante diferentes das configurações para caminhar num terreno como a grama artificial utilizada na Robocup. Essas mudanças nas configurações são determinantes para o bom equilíbrio do robô. Com mais de 100 parâmetros, esse processo de configuração requer muito tempo e paciência.

O primeiro problema com a implementação atual é que a configuração se dá através de alterações diretas dos valores dos parâmetros no código-fonte, em seguida compila-se, envia-se a nova versão à *OpenCM9*. Em seguida, inicializa-se o robô e verifica-se a caminhada. Além de massante, este esquema de atualização não favorece uma forma sadia de manter as alterações dos parâmetros organizada. Há uma frequente perda de quais valores de parâmetros são melhores para qual tipo de situação. Uma solução seria criar subversões do sistema para cada cenário, o que desfavorece a correção de *BUGs* e melhorias, já que cada modificação deve ser refletida em várias versões. Uma segunda possibilidade seria criar uma interface de comunicação entre um computador, que funcionaria como um gerenciador, e o *walking gait* via USB. Assim, o computador poderia guardar arquivos de configurações que seriam enviados à placa microcontroladora a medida que fossem necessários, o que seria uma saída simples e eficaz, mas não definitiva, tendo em vista outros problema citados abaixo.

O segundo problema com o *walking gait* atual é a dificuldade de implementação de simulações, testes e depuração. A única forma de obter visualização da saída gerada é através do *console serial* da IDE de programação, o que torna o processo de depuração um exercício de abstração com visualização de ângulos em 3 dimensões, ou arriscar-se

Introduz
a ex-
pres-
são
*wal-
king
gait*.

aplicando a saída direto aos motores, de alto custo. Em relação a simulação, ela seria até possível, porém seria necessária mais um dispositivo *OpenCM9.04* e algumas modificações no código original para que ele possa ser acoplado à algum framework de simulação de robôs.

O terceiro problema é a forma em que o paradigma estruturado foi utilizada no sistema. A complexidade do sistema implica para uma baixa curva de assimilação do funcionamento em um paradigma orientado a objetos ajudaria consideravelmente na compreensão dos mecanismos dentro do *walking gait*, quais os seus papéis e como eles se comunicam.

O quarto problema é a natureza multi-processada do sistema. "Ao mesmo tempo" em que ângulos devem ser enviados aos motores, a leitura dos sensores devem ser processadas e levadas em consideração para as próximas iterações. No *OpenCM0.04*, esse paralelismo é habilitado através da biblioteca *MapleFreeRTOS*. O fato de já haver muito processamento para rodar o sistema atual limita uma grupo de melhorias a serem desenvolvidas, o que poderia levar a próxima geração de robôs ainda mais estáveis.

Por fim, o quinto problema, que também limita o desenvolvimento da próxima geração, é o tamanho da memória ROM, que guarda o *walking gait*, da placa microcontroladora. Atualmente, o *firmware* da *OpenCM9.04* já foi modificado, via remoção de funções não utilizadas, para diminuir seu tamanho e assim liberar espaço para a versão do sistema.

Dadas, os problemas encontrados no funcionamento do sistema atual, e as considerações listadas até agora, este trabalho propõe mover a execução do componente *walking gait* do microcontrolador *OpenCM09.04* para o computador. Para o projeto da solução, foi lavada em consideração dois aspectos importantes: o desempenho e a interface de comunicação com os motores, decidindo-se assim pela adoção da linguagem *C++* versão 14. Um fator bastante importante foram os recursos da linguagem que vão prover uma arquitetura bem simples e eficiente.

Desta forma, este trabalho tem como objetivo a implementação de uma versão funcional do *walking gait*, melhorando o código atual - de forma a diminuir a curva de aprendizado, assim estimulando novas extensões e otimizações - com uma nova arquitetura orientada a objetos, ativando o sistema com suporte ao ROS, adicionando uma interface web responsiva de configuração.

Convers
com
o
ori-
en-
ta-
dor
so-
bre
a
lista

1.1 Metodologia

Na metodologia é descrito o método de investigação e pesquisa para o desenvolvimento e implementação do trabalho que está sendo proposto.

1.2 Organização do trabalho

Nesta seção deve ser apresentado como está organizado o trabalho, sendo descrito, portanto, do que trata cada capítulo.

Conversa
com
o
ori-
en-
ta-
dor
so-
bre
isto.

2 Capítulo 2

Definir
tí-
tulo

GUIA:

Falar sobre as tecnologias atuais de *walking gait*.

GUIA:

Falar sobre a abordagem tomada para este trabalho.

Este é o primeiro capítulo da parte central do trabalho, isto é, o desenvolvimento, a parte mais extensa de todo o trabalho. Geralmente o desenvolvimento é dividido em capítulos, cada um com subseções e subseções, cujo tamanho e número de divisões variam em função da natureza do conteúdo do trabalho.

Em geral, a parte de desenvolvimento é subdividida em quatro subpartes:

- *contextualização ou definição do problema* – consiste em descrever a situação ou o contexto geral referente ao assunto em questão, devem constar informações atualizadas visando a proporcionar maior consistência ao trabalho;
- *referencial ou embasamento teórico* – texto no qual se deve apresentar os aspectos teóricos, isto é, os conceitos utilizados e a definição dos mesmos; nesta parte faz-se a revisão de literatura sobre o assunto, resumindo-se os resultados de estudos feitos por outros autores, cujas obras citadas e consultadas devem constar nas referências;
- *metodologia do trabalho ou procedimentos metodológicos* – deve constar o instrumental, os métodos e as técnicas aplicados para a elaboração do trabalho;
- *resultados* – devem ser apresentados, de forma objetiva, precisa e clara, tanto os resultados positivos quanto os negativos que foram obtidos com o desenvolvimento

do trabalho, sendo feita uma discussão que consiste na avaliação circunstanciada, na qual se estabelecem relações, deduções e generalizações.

É recomendável que o número total de páginas referente à parte de desenvolvimento não ultrapasse 60 (sessenta) páginas.

2.1 Seção 1

Teste de figura:



Figura 1: Teste de uma figura em formato .png

2.2 Seção 2

Referenciamento da figura inserida na seção anterior: ??

2.3 Seção 3

Seção 3

2.3.1 Subseção dentro da seção 3

2.3.2 Subseção dentro da seção 3

2.4 Seção 4

Seção 4

3 Arquitetura

ARASH (Robô Antropomórfico Aumentado com Sentido de Ser Humano), é um robô humanoide de 7,5 Kg, 1 metro de altura. Com 20 graus de liberdade (ou DOF, de *degrees of freedom*), distribuídos em 6 DOF para cada perna, 3 em cada braço e 2 no pescoço.

Cada vez mais, a arquitetura distribuída é utilizada na robótica. A ideia é que cada componente funcione de forma independente para evitar uma eventual falha catastrófica onde todo o sistema venha a ser derrubado. Na figura 2 é possível observar os principais componentes, e o fluxo de dados entre eles, envolvidos no processo qualquer tarefa básica. Esses componentes podem ser executados em diferentes camadas do sistema. Na arquitetura, existem 3 camadas principais que podem ser descritas independentemente: A camada de software, a camada de software de baixo nível, e a camada de hardware.

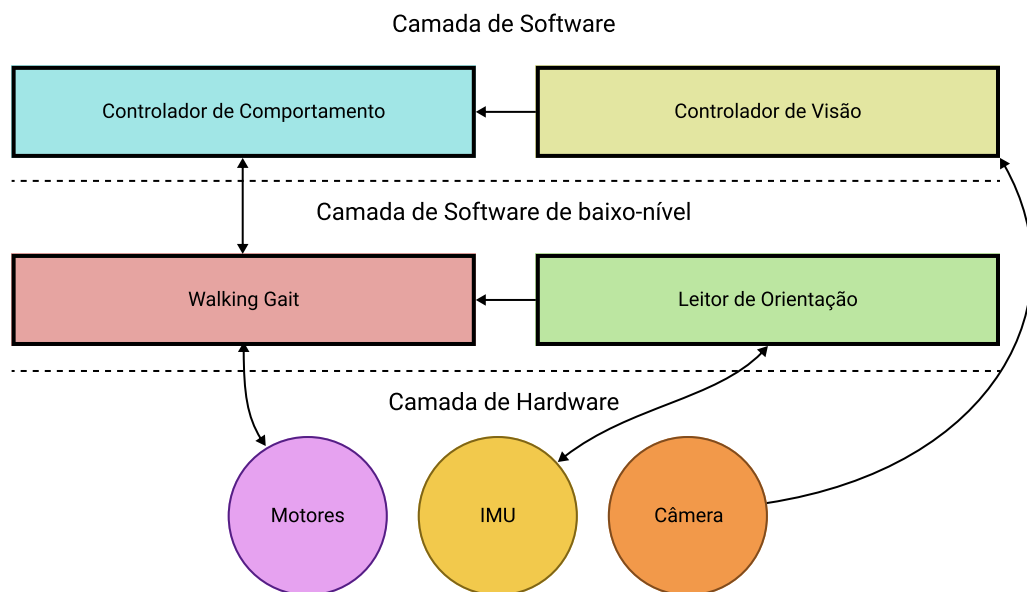


Figura 2: Visão simplificada da arquitetura do sistema e seus fluxos.

A camada de software é executada dentro do “controlador principal”, que é um computador *mini-box PC MAXData QutePC-3001* com um processador 64 bits de dois núcleos Intel® Celeron® 847E (1,1GHz e 2MB de *cache*), rodando o Ubuntu 14.04. Esta camada é responsável por rodar os componentes de software e enviar comandos de controle para

a camada inferior de baixo nível.

A camada de software de baixo nível é responsável pelo processamento de dados da IMU, juntamente com a execução do *walking gait*. Esta camada roda dentro da placa microcontroladora *Robotis.Co OpenCM9.04* (com um processador *32bit ARM Cortex-M3*, memória *flash* de 128Kb e 20Kb de *SRAM*), *arduino-like*.

3.1 Visão

Visão não é o foco deste trabalho. Entretanto, uma explanação geral é necessária afim de melhor entender o funcionamento geral do sistema.

O controle de visão é um software que conecta-se via USB com a câmera através. De forma geral, podemos abstrair seu funcionamento em duas *threads*.

A primeira é o *loop* principal - representada na figura 3 -, onde, sequencialmente, o controlador obtém a *imagem* da câmera. Em seguida, a extração de informações é realizada. Esse processo vai depender da tarefa que o robô deve realizar - detectar a bola, os adversários e as linhas do campo em caso de futebol ou detectar as linhas guia da pista de corrida, no caso da maratona são exemplos. Em seguida, a *thread* salva tudo o que foi detectado.

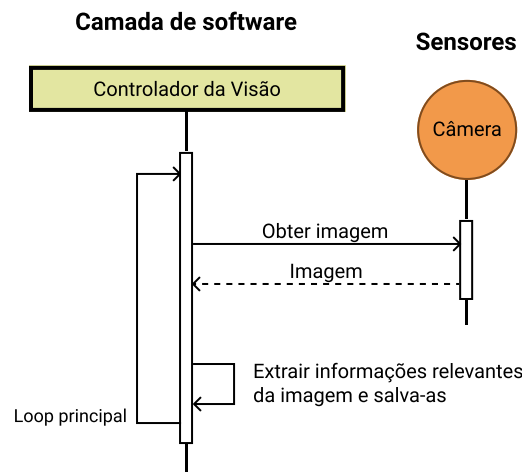


Figura 3: Diagrama de sequência do componente de visão.

A segunda *thread* - visualizada na figura 4 - do controle de visão executa o sistema de comunicação, que pode ser implementado de várias maneiras - desde *JSON over UDP* até um *message broker* com filas de consumo. Ao receber a mensagem de solicitação das informações, responde serializando os dados que ficaram salvos na primeira *thread*, fornecendo as informações com atraso mínimo.

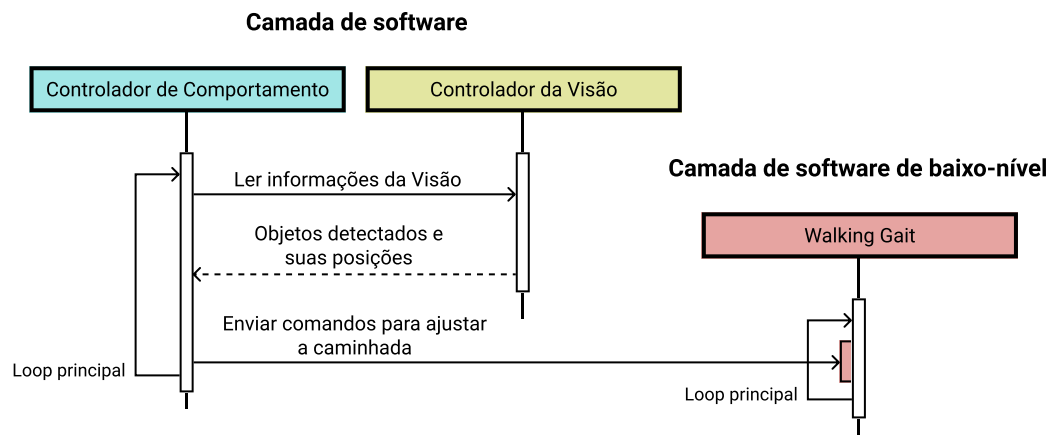


Figura 4: Diagrama de sequência do controlador de comportamento e suas dependências.

O controlador de comportamento implementa a tarefa que deve ser realizada pelo robô. Ele funciona como um gerente que coordena os outros componentes, recebendo informações e mandando comandos para ações específicas serem executadas. Na figura 4 é possível observar a sequência de passos realizados pelo controlador de comportamento. Por exemplo, durante uma partida de futebol, o controlador de comportamento “jogador” consulta o componente de visão, que responde a posição da bola - previamente obtida durante o processo da figura 3. Em seguida, baseado nessas informações o comportamento envia o comando ao *walking gait* para caminhar na direção correta.

DRAFT: Esta camada roda o *walking gait* e o *sistema de orientação*. Ela tem acesso direto a enviar e receber comandos dos motores e os sensores da *IMU*, conectando-se ao controlador principal via USB recebendo pacotes de controle e enviando comandos para os motores em uma conexão *TTL half-duplex* a 1 Mbps.

O *walking gait* monitora a USB aguardando os comandos de controle que contém as velocidades da caminhada, enviados a partir do controlador principal. Já o componente leitor de orientação desempenha um papel importante dentro do *walking gait*. Ele é o responsável pela verificação da orientação da rotação do torso, uma vez que a *IMU* está situada nas costas de Arash, em relação ao eixo da gravidade. Desta forma, o *walking gait* faz correções nas juntas para compensar um eventual desvio na trajetória. Mais detalhes sobre a implementação deste componente serão descritos nos próximos capítulos!!!

Na design original do time AUT-UofM, a *OpenCM9.04* é responsável por executar o componente de caminhada, coordenando os movimentos das juntas de acordo com parâmetros recebidos através da USB (enviados pelo controlador principal). Desta forma, os dados de leitura sobre a posição das juntas ou sobre a orientação, obtido pelos sensores

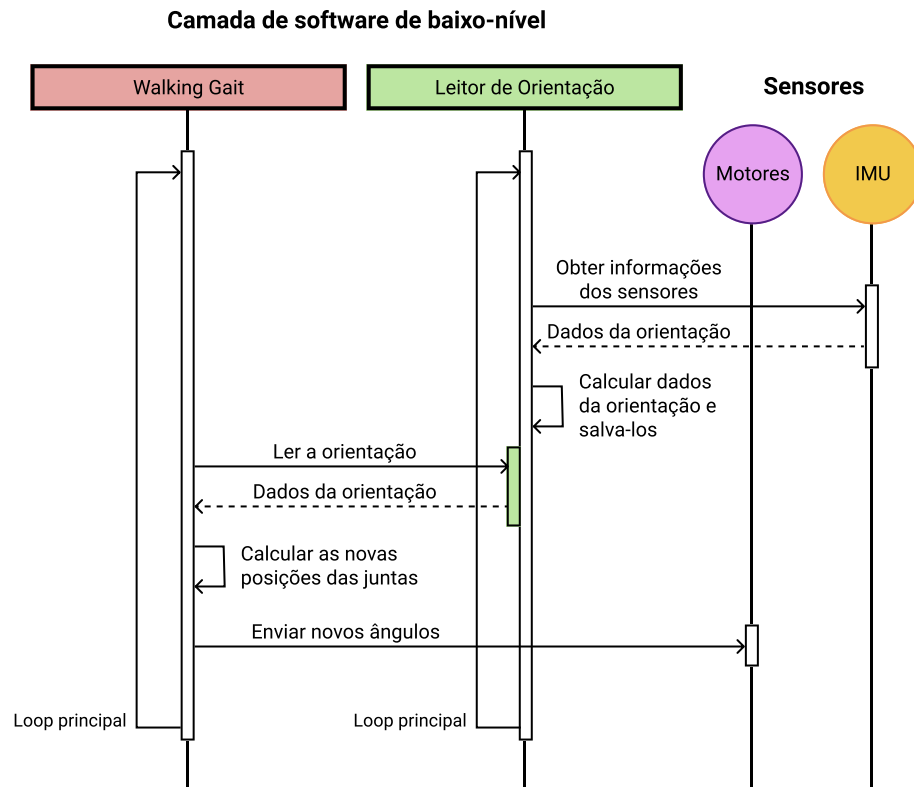


Figura 5: Diagrama de sequência do *walking gait* e do leitor de orientação.

da IMU, não são necessários fora da placa microcontroladora.

3.1.1 Graus de Liberdade e Atuadores

Em Arash, todos os atuadores são produzidos pela Robotics.Co. Porém, devido a variação de carga nas diversas juntas, foram utilizados modelos diferentes da mesma série MX, esta opção diminui o custo final do robô.

Tão importante quanto a configuração das juntas é a orientação dos atuadores para que os cálculos funcionem como o esperado.

Na figura acima, podemos observar o esquema de orientação dos atuadores em cada junta. No diagrama encontram-se 4 símbolos com significados diferentes:

O pequeno círculo preto preenchido, representa apenas o fim de cadeia. Assim, estes símbolos representam as mãos, que não possuem atuadores, e a ponta da cabeça, onde fica apenas a câmera. Adicionalmente, os "círculo vazado com outro círculo preto preenchido dentro" e os diamantes representam atuadores.

No caso do "círculo vazado com outro círculo preto preenchido dentro", o atuador será referenciado como "frontal" e propicia rotação orientada pelo eixo X (no plano YZ),

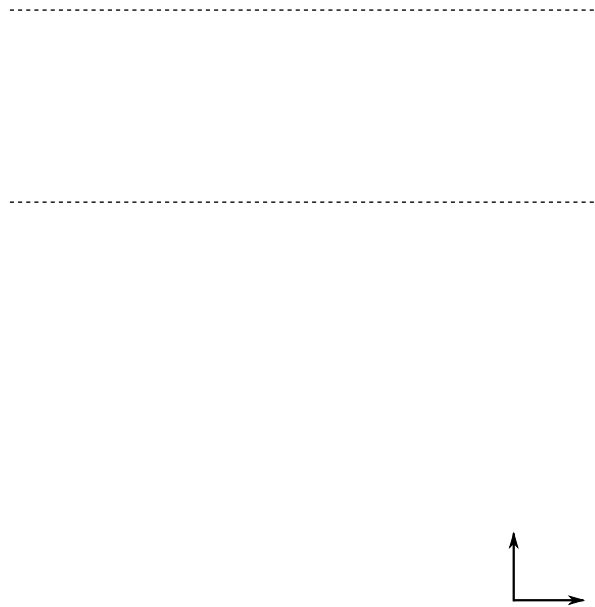


Figura 6: Esquema de orientação dos atuadores de Arash

que está aponta para o leitor (para fora do papel).

Nos diamantes, o atuador está apontando para a marcação preta na ponta do diamante. Atuadores que apontam para direita ou esquerda, são chamados de horizontais. Eles apresentam rotação com base no eixo Y. Já os que apontam para cima ou baixo são chamados de atuadores transversais que representam transformações na rotação no eixo Z.

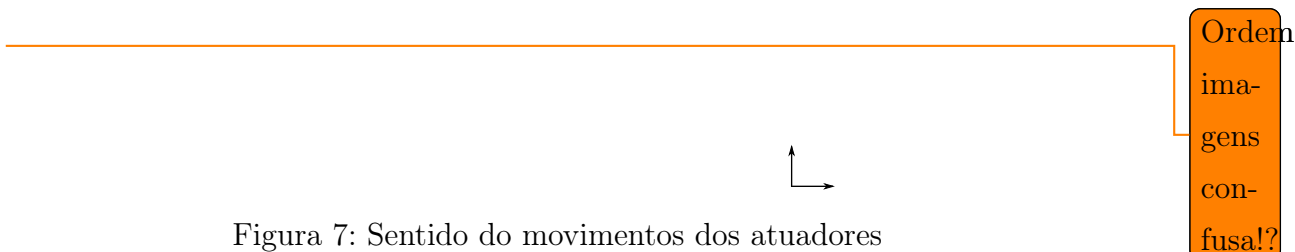


Figura 7: Sentido do movimentos dos atuadores

Estas orientações são importantes durante os cálculos dos ângulos enviados às juntas. Por exemplo, caso a orientação do cotovelo esteja trocada, no cálculo ou no robô real, qualquer ângulo enviado a ele resultará em uma rotação para trás, um movimento que o cotovelo humano não é capaz de realizar.

No pescoço, onde a carga é bem leve, 'MX-28' são suficientes. Dois atuadores, um na posição transversal (o atuador mais baixo) que fornece o movimento panorâmico a cabeça e outro na posição horizontal, fornecendo o movimento de inclinação vertical da cabeça.

Ordem
ima-
gens
con-
fusa!?

Referen-
ao
pa-
per
de
des-
cri-
ção
de
Arash

Nos braços, que podem sofrer uma carga maior, atuadores 'MX-64' são utilizados. Isso é importante já que existem modalidades de competições, como o levantamento de peso, que testam a capacidade do carregamento de cargas.

Para as pernas, foram utilizados atuadores 'MX-106' que são mais poderosos que os anteriores. Entretanto, na fase de projeto, simulações mostraram que durante o movimento de levantar-se do chão (em caso da recuperação de uma possível queda), o torque nas juntas do joelho era levado ao máximo suportado pelo 'MX-106', podendo assim levar este motor à falha. Desta forma, 2 atuadores sincronizados passaram a formar esta junta, afim de proporcionar um maior torque para os movimentos necessários. Esta junta dupla não oferece nenhum impacto na implementação, já que os motores da série 'MX-106' oferecem a capacidade de serem ligados e sincronizados via *hardware* - assim, a nível de software, controla-se apenas 1 único atuador. Desta forma, apesar das 20 DOF, Arash possui 22 motores.

3.2 Arquitetura de Software pré-solução

3.3 Modificações na *OpenCM9.04*

Na implementação da solução deste trabalho a geração das trajetórias do *walking gait* será transferida da *OpenCM9.04* para o controlador principal. Entretanto, a placa microcontroladora ainda terá um papel crucial dentro da nova implementação.

GUIA:

Detalhar arquitetura.

GUIA:

Detalhar componentes.

- ser claro, preciso, direto, objetivo e conciso, utilizando frases curtas e evitando ordens inversas desnecessárias;
- construir períodos com no máximo duas ou três linhas, bem como parágrafos com cinco linhas cheias, em média, e no máximo oito (ou seja, não construir parágrafos e períodos muito longos, pois isso cansa o(s) leitor(es) e pode fazer com que ele(s) percam a linha de raciocínio desenvolvida);

- a simplicidade deve ser condição essencial do texto; a simplicidade do texto não implica necessariamente repetição de formas e frases desgastadas, uso exagerado de voz passiva (como *será iniciado*, *será realizado*), pobreza vocabular etc. Com palavras conhecidas de todos, é possível escrever de maneira original e criativa e produzir frases elegantes, variadas, fluentes e bem alinhavadas;
- adotar como norma a ordem direta, por ser aquela que conduz mais facilmente o leitor à essência do texto, dispensando detalhes irrelevantes e indo diretamente ao que interessa, sem “rodeios” (verborragias);
- não começar períodos ou parágrafos seguidos com a mesma palavra, nem usar repetidamente a mesma estrutura de frase;
- desprezar as longas descrições e relatar o fato no menor número possível de palavras;
- recorrer aos termos técnicos somente quando absolutamente indispensáveis e nesse caso colocar o seu significado entre parênteses (ou seja, não se deve admitir que todos os que lerão o trabalho já dispõem de algum conhecimento desenvolvido no mesmo);
- dispensar palavras e formas empoladas ou rebuscadas, que tentem transmitir ao leitor mera ideia de erudição (até mesmo às vezes ilusória);
- não perder de vista o universo vocabular do leitor, adotando a seguinte regra prática: *nunca escrever o que não se diria*;
- termos coloquiais ou de gíria devem ser usados com extrema necessidade (ou mesmo nem serem utilizados) e apenas em casos muito especiais, para não darem ao leitor a ideia de vulgaridade e descaracterizar o trabalho;
- ser rigoroso na escolha das palavras do texto, desconfiando dos sinônimos perfeitos ou de termos que sirvam para todas as ocasiões; em geral, há uma palavra para definir uma situação;
- encadear o assunto de maneira suave e harmoniosa, evitando a criação de um texto onde os parágrafos se sucedem uns aos outros como compartimentos estanques, sem nenhuma fluência entre si;
- ter um extremo cuidado durante a redação do texto, principalmente com relação às regras gramaticais e ortográficas da língua; geralmente todo o texto é escrito

na forma impessoal do verbo, não se utilizando, portanto, de termos em primeira pessoa, seja do plural ou do singular.

Continuação do texto.

3.4 Seção 1

Teste de tabela.

Tabela 1: Tabela sem sentido.

Título Coluna 1	Título Coluna 2
Texto curto	Texto mais extenso, que requer mais de uma linha

3.5 Seção 2

Seção 2

3.5.1 Subseção 2.1

Referência à tabela definida no início: ??

3.5.2 Subseção 2.2

Texto a ser enumerado.

- 1. Item 1
- 2. Item 2, com nota explicativa¹
- 3. Item 3

¹Nota explicativa

3.6 Seção 3

Texto antes de equação.

$$x = y + z \tag{3.1}$$

Outra maneira de se usar equação.

$$\forall \pi : \pi \hookrightarrow \gamma$$

Texto depois de equação.

3.7 Seção 4

Exemplo de código

4 Capítulo 4

GUIA:

Detalhar matemática dos movimentos.

4.1 Seção 1

Teste para símbolo

λ

4.2 Seção 2

Teste para abreviatura

IFRN

DIATINF

5 Capítulo 5

GUIA:

Detalhar implementação, integração e simulação e testes.

5.1 Seção 1

GUIA:

Detalhar implementação.

5.2 Seção 2

GUIA:

Detalhar simulação.

5.3 Seção 3

GUIA:

Detalhar simulação.

Seção 3

6 Considerações finais

As considerações finais formam a parte final (fechamento) do texto, sendo dito de forma resumida (1) o que foi desenvolvido no presente trabalho e quais os resultados do mesmo, (2) o que se pôde concluir após o desenvolvimento bem como as principais contribuições do trabalho, e (3) perspectivas para o desenvolvimento de trabalhos futuros, como listado nos exemplos de seção abaixo. O texto referente às considerações finais do autor deve salientar a extensão e os resultados da contribuição do trabalho e os argumentos utilizados estar baseados em dados comprovados e fundamentados nos resultados e na discussão do texto, contendo deduções lógicas correspondentes aos objetivos do trabalho, propostos inicialmente.

6.1 Principais contribuições

Texto.

6.2 Limitações

Texto.

6.3 Trabalhos futuros

Texto.

APÊNDICE A – Primeiro apêndice

Os apêndices são textos ou documentos elaborados pelo autor, a fim de complementar sua argumentação, sem prejuízo da unidade nuclear do trabalho.

ANEXO A – Primeiro anexo

Os anexos são textos ou documentos não elaborado pelo autor, que servem de fundamentação, comprovação e ilustração.

Todo list

Normalizar todas as nomenclaturas do <i>walking gait</i> , as vezes chama-se sistema, as vezes chama-se componente.	p. 13
Introduzir a expressão <i>walking gait</i>	p. 14
Conversar com o orientador sobre a lista de objetivos.	p. 15
Melhorar a descrição dos objetivos.	p. 15
Introduzir o ROS	p. 15
Conversar com o orientador sobre isto.	p. 16
Definir título	p. 17
Referenciar seção específica	p. 21
Ordem imagens confusa!?	p. 23
Referenciar ao paper de descrição de Arash	p. 23
Conectar estes dois parágrafos	p. 23