



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO - DCA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

KASSIO JANIELSON DA SILVA EUGENIO

**IMPLEMENTAÇÃO DE ARQUITETURA DE HARDWARE E
SOFTWARE PARA A ÓRTESE ATIVA PARA OS MEMBROS
INFERIORES ORTHOLEG**

NATAL - RN, 2016



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO - DCA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

KASSIO JANIELSON DA SILVA EUGENIO

**IMPLEMENTAÇÃO DE ARQUITETURA DE HARDWARE E
SOFTWARE PARA A ÓRTESE ATIVA PARA OS MEMBROS
INFERIORES ORTHOLEG**

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte como parte dos requisitos para a obtenção do título de Engenheiro Mecatrônico, orientado pelo prof. Dr. Pablo Javier Alsina.

NATAL - RN, 2016



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO - DCA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

KASSIO JANIELSON DA SILVA EUGENIO

**IMPLEMENTAÇÃO DE ARQUITETURA DE HARDWARE E
SOFTWARE PARA A ÓRTESE ATIVA PARA OS MEMBROS
INFERIORES ORTHOLEG**

Banca Examinadora do Trabalho de Conclusão de Curso

Prof. Dr. Pablo Javier Alsina _____

Universidade Federal do Rio Grande do Norte . Orientador

Msc. Nicholas de Bastos Melo _____

Universidade Federal do Rio Grande do Norte - Avaliador Interno

Prof. Dr. Carlos Eduardo Trabuco Dórea _____

Universidade Federal do Rio Grande do Norte - Avaliador Interno

NATAL, Novembro de 2016

Aos meus pais e a minha noiva pelo incentivo e apoio ao longo deste período.

Aos amigos de laboratório pelo incentivo pelos apoios constantes.

A todos os professores do curso, q foram tão importantes na minha vida acadêmica.

A todos aqueles q de alguma forma estiveram estão próximos de mim.

AGRADECIMENTOS

Este trabalho não poderia ser concluído sem a ajuda de diversas pessoas as quais presto minha homenagem:

Agradeço primeiramente Deus, por ser essencial em minha vida, autor de meu destino, meu guia, socorro presente na hora da angústia, meu pai João Maria Eugenio, minha mãe Maria do Carmo da Silva Eugenio e meus irmãos.

A todos os companheiros de curso que vivenciaram comigo a árdua batalha de conseguir o título de engenheiro mecatrônico, pelas diversas brincadeiras e o bom humor, que traziam alegria nos momentos de estresse e cansaço.

Aos integrantes do laboratório de robótica do DCA, que me ajudaram muito durante todo o tempo que passei na iniciação científica, com dicas, artigos e livros, principalmente os integrantes do projeto Ortholeg.

A Vanessa Gabriela, minha amada noiva, que ao longo deste período soube me ajudar e apoiar.

A Válber Cesar Cavalcanti Roza, pela imensa ajuda nas dúvidas na programação e pelo incentivo e apoio.

A Nicholas de Bastos Melo por me auxiliar no desenvolvimento e implementação da arquitetura para a nova Ortholeg e na programação.

Tenho imensa gratidão ao meu professor orientador, Pablo Javier Alsina, por ter me concedido inicialmente a oportunidade de iniciação científica, que através de muitos desafios, dedicação e orientações, fez-me desenvolver esse trabalho de conclusão de curso, entre outros artigos científicos.

Por fim, agradeço ao CNPQ pelo suporte financeiro durante o projeto.

EUGENIO, K.J.S. Implementação de arquitetura de hardware e software para uma órtese ativa para os membros inferiores Ortholeg. 2016. 44p. Trabalho de Conclusão de Curso. (Graduação em Engenharia Mecatrônica) - Universidade Federal do Rio Grande do Norte, Natal, 2016.

RESUMO

Na garantia de aproveitar o melhor desempenho de um robô móvel, o qual apresenta diversos desafios, por exemplo, a locomoção, percepção, planejamento de trajetória e principalmente o processamento dos dados, se faz necessário utilizar uma arquitetura de *hardware* e *software* onde os componentes físicos (*hardware*) possam executar as tarefas ou comandos definidos na sua programação (*software*) fornecendo habilidades para resolver os problemas a que aquele robô está sendo submetido.

Sendo assim, uma arquitetura de hardware e software é importante na área de sistemas robóticos, possibilitando implementações de projetos de diversos níveis hierárquicos. A arquitetura também proporciona a interação entre o homem e a máquina, ou seja, o usuário de um sistema robótico tem a possibilidade de interagir em tempo real com o comportamento previamente planejado para o robô. As principais características de uma arquitetura são de assegurar o cumprimento das tarefas de maneira estável, robusto e possuir um bom desempenho em tempo real utilizando-se de seus recursos físicos e computacionais.

Este trabalho tem como objetivo desenvolver uma nova arquitetura de hardware e software para controle e acionamento de uma órtese ativa (Ortholeg 2.0). A arquitetura de *hardware* é composta por uma plataforma microcontroladora, um computador embarcado, uma placa de acionamento de motores CC dotados de sensor de posição angular, possibilitando que a órtese realize movimento antropomórfico. A arquitetura de *software* é organizada de modo que os componentes de *hardware* possam estar em constante comunicação e contém a estratégia para que as tarefas atribuídas à órtese sejam executadas. A arquitetura de *hardware* e *software* proposta deve ser aberta e modular, permitindo a inclusão de novos sensores e outros dispositivos externos.

Palavras-chave: Arquitetura de hardware e software. Órtese ativa. Computador embarcado. Microcontrolador. Placa de acionamento. Protocolo de comunicação I²C. Controle PID.

ABSTRACT

In order to guarantee the best performance of a mobile robot, where this robot presents several challenges, for example, locomotion, perception, path planning and especially data processing, it is necessary to use a hardware and software architecture where the components (Hardware) can perform the tasks or commands defined in their programming (software) providing skills to solve the problems that the robot is undergoing.

Thus, a hardware and software architecture is important in the area of robotic systems, enabling implementation of projects of different hierarchical levels. The architecture also provides the interaction between the man and the machine, that is, the user of a robotic system has the possibility of interacting in real time with the behavior previously planned for the robot. The main characteristics of an architecture are provided to ensure the fulfillment of the tasks in a stable, robust and have a good performance in real time using its physical and computational resources. This work aims to develop a new hardware and software architecture for the control and driving of a active orthosis (Ortholeg 2.0). The hardware architecture consists of a microcontroller platform, an embedded computer, a DC motor drive board and angular position sensor, allowing the orthosis to perform an anthropomorphic movement. The software architecture is organized in order to the hardware components can be in constant communication and contains the strategy for the tasks assigned to the orthosis to be executed. The proposed hardware and software architecture must be open and modular, allowing the inclusion of new sensors and other external devices.

Keywords: Hardware and software architecture. Active orthosis. Embedded computer. Microcontrollers. Drive plate. I²C communication protocol. PID control.

LISTA DE FIGURAS

Figura 1 - Protótipo órtese Ortholeg	5
Figura 2 - Órtese Ortholeg 2.0	5
Figura 3 - Arquitetura de <i>Software</i> presente na órtese Ortholeg	6
Figura 4 - Arquitetura da interface homem-máquina	7
Figura 5 - Arquitetura de <i>Hardware</i> presente na órtese Ortholeg.....	8
Figura 6 - Nova arquitetura de <i>software</i>	10
Figura 7 - Nova arquitetura de <i>hardware</i>	11
Figura 8 - Organização da arquitetura de <i>hardware</i> na órtese Ortholeg	12
Figura 9 - Modelos do microcomputador Raspberry Pi	13
Figura 10 - Tipos de plataforma Arduino	15
Figura 11 - Placa de acionamento TqRex <i>Robot Controler</i>	17
Figura 12 - Protocolo de comunicação I ² C	18
Figura 13 - Diagrama de comunicação via barramento I ² C	19
Figura 14 - Diagrama de blocos com controle PID.....	22
Figura 15 - Problema de leitura do sensor de Encoder na placa de acionamento TqRex	26
Figura 16 - Valores de pulsos conforme o aumento do sinal PWM.....	27
Figura 17 - Obtenção dos ganhos para K _p , K _i e K _d	28
Figura 18 - Ganhos obtidos para a estabilidade e controle dos atuadores.....	28
Figura 19 - Movimento de passo do joelho.....	29

LISTA DE TABELAS

Tabela 1 - Parâmetros do controlador PID.....	21
Tabela 2 - Regra de sintonia de Ziegler - Nichols baseada no τ_{11} e τ_{22}	24

LISTA DE ABREVIATURAS E SIGLAS

CC	Corrente Continua
DCA	Departamento de Automação e Computação
FTDI	<i>Future Technology Devices International</i>
I ² C	<i>Inter-Integrated Circuit</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IHM	Interface Homem Máquina
MISO	<i>Multiple Inputs and Single Output</i>
MOSI	<i>Multiple Output and Single Input</i>
OHMD	<i>Optical Head-Mounted Display</i>
PID	Proporcional . Integral - Derivativo
PTC	<i>Positive Temperature Coefficient</i>
PV	Variável de Processo
PWM	<i>Pulse Width Modulation</i>
RA	Robótica Assistiva
RX	Receptor
SAL	Sistema de Auxílio à Locomoção
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SP	<i>Set Point</i>
SPI	<i>Serial Peripheral Interface</i>
TTL	<i>Transistor-Transistor Logic</i>
TX	Transmissor

SUMÁRIO

1 INTRODUÇÃO	1
1.1 Robótica assistiva	2
1.2 Objetivos	2
1.3 Organização do texto	3
2 ÓRTESE ATIVA PARA MEMBROS INFERIORES	4
2.1 Órtese Ortholeg	4
2.2 Arquitetura embarcada	6
2.2.1 Arquitetura de Software existente	6
2.2.2 Arquitetura de Hardware existente	7
3 ARQUITETURA DE HARDWARE E SOFTWARE PROPOSTA	9
3.1 Arquitetura de Software proposta	9
3.2 Arquitetura de Hardware proposta	11
4 DISPOSITIVOS DE HARDWARE	13
4.1 Microcomputador Raspberry Pi	13
4.2 Plataforma eletrônica Arduino	14
4.3 Placa de acionamento <i>TqRex Controller Robot</i>	16
5 PROTOCOLO DE COMUNICAÇÃO I²C	18
6 CONTROLADOR	20
6.1 Controle PID	20
6.2 Parâmetros do controle PID	21
6.3 Método utilizado para obtenção dos parâmetros.	23
7 RESULTADOS EXPERIMENTAIS	26
8 CONCLUSÃO	29
9 REFERÊNCIAS BIBLIOGRÁFICAS	31

1 INTRODUÇÃO

A robótica tem auxiliado o homem em diversas tarefas, desde a exploração espacial até ao auxílio a pessoas com deficiências físicas. No caso de pessoas com algum tipo de deficiência física, de acordo com os dados estatísticos, apresenta um crescimento significativo de idosos em 19,3% da população total do Brasil (IBGE, 2010). O estudo mostra também que 1,3% da população têm algum tipo de deficiência física e quase a metade desse total (46,8%) tem grau intenso ou muito intenso de limitações. Com base nestes dados, é possível afirmar que haverá um aumento da demanda de atendimento específico para esse grupo de pessoas.

Nos últimos anos, houve um aumento de pesquisas e desenvolvimento de projetos para auxiliar na reabilitação de movimentos com auxílio de sistemas mecânicos e robóticos, desenvolvendo variações de robôs compostos por mecanismos ligados a circuitos elétricos, por exemplo, exoesqueletos, órteses e próteses ativas. Em conjunto a este desenvolvimento temos o crescimento de uma gama de sensores capazes de ler informações do ambiente, como por exemplo, sensores de detecção de obstáculos, de sistemas de visão, de posicionamento e outros.

Dentre os diversos sensores de posicionamento, destacam-se os sensores de posição angular ou, simplesmente, Encoders rotativos, os quais fornecem sinal de posição angular de um eixo ou objeto em rotação. Nas indústrias são utilizados para o controle e medição de velocidade dos motores e posição e sentido de giro das esteiras e na biomédica no controle dos atuadores elétricos localizados em juntas rotacionais, no caso das órteses e exoesqueletos.

Em exoesqueletos e órteses ativas, a evolução tecnológica destes sensores e sistemas eletrônicos organizados por uma arquitetura de hardware e software garante qualidade na execução de um determinado movimento pré-definido pela sua programação, com sistemas mais simples e compactos, não dificultando os movimentos dos pacientes e executando uma leitura biométrica mais efetiva dos movimentos mecânicos executados pelo usuário, além da maior confiabilidade e segurança do sistema como um todo.

Conforme a evolução tecnológica, o homem tem buscado inovar em diversas aplicações, tais como: na indústria - exercem funções repetitivas de produção direta ou de controle de produção de fábrica; espacial - na exploração do espaço e na

inspeção de estações espaciais; medicina - em cirurgia robótica. Além dessas inovações, outras áreas foram surgindo concomitantemente, como por exemplo, a domótica onde todos os equipamentos elétricos e eletrônicos em uma residência sejam possíveis de controlar e automatizar, através de um sistema integrado central e a robótica assistiva que proporciona meios que possam diminuir as dificuldades e ampliar habilidades funcionais de pessoas com deficiência física.

Dentro deste contexto, este trabalho apresenta o desenvolvimento de uma arquitetura de hardware e software para uma órtese ativa de membros inferiores, com a utilização de uma plataforma eletrônica microcontrolada, placa de acionamento de motores CC dotados de sensor Encoder.

1.1 Robótica assistiva

A robótica assistiva (RA) é uma das aplicações da robótica que visa a melhoria de vida e que tem como objetivo de proporcionar meios que possam diminuir os problemas e desenvolver habilidades funcionais de pessoas com deficiência física. As pesquisas em robótica assistiva têm contribuído em situações onde existe a necessidade de adaptações como, a ampliação de habilidades de comunicação; equipamentos de assistência na visão e audição; adaptações da postura e mobilidade (órteses e próteses).

As órteses disponíveis atualmente para a reabilitação de pessoas com deficiência nos membros inferiores são utilizadas para a correção e o auxílio para pacientes com perda parcial ou total dos movimentos, baseado nisto, diversas órteses ativas e exoesqueletos vêm sendo desenvolvidos [Mori et al. 2005].

A robótica assistiva é de grande importância para a sociedade. Um vasto e amplo campo pode ser explorado por tal tecnologia dando base para outras aplicações tanto na computação quanto na engenharia.

1.2 Objetivo

Este trabalho tem como objetivo apresentar a implementação de uma arquitetura de hardware e software para o controle e acionamento de uma órtese ativa de membros inferiores (Ortholeg).

Para tanto, busca-se:

- Desenvolver uma arquitetura aberta e modular, facilitando a inclusão de novos sensores;
- Desenvolver um algoritmo de controle para o funcionamento adequado e estável dos motores;
- Analisar os dados adquiridos nos testes para melhoramento da arquitetura proposta.

1.3 Organização do texto

A organização deste texto é dada da seguinte maneira:

- **Capítulo 2** - É apresentado a órtese Ortholeg e a sua arquitetura de hardware.
- **Capítulo 3** - Apresenta a nova arquitetura de hardware e software proposta.
- **Capítulo 4** - Mostra os principais dispositivos de hardware que são utilizados na nova arquitetura de *hardware*.
- **Capítulo 5** - Este capítulo apresenta tipo de protocolo de comunicação utilizado na conexão entre os dispositivos de *hardware*.
- **Capítulo 6** . Este capítulo aborda o tipo de controlador utilizado na programação para o acionamento desejado dos motores.
- **Capítulo 7** - Descrição dos resultados obtidos através dos testes feitos.
- **Capítulo 8** . Conclusão.

2 ÓRTESE ATIVA PARA MEMBROS INFERIORES

Órteses ativas são dispositivos robóticos vestíveis, com a principal função de corrigir ou melhorar a função das partes móveis do corpo. Estas órteses apresentam diversos tipos de atuadores para produzir o movimento nos membros de uma pessoa que é incapaz ou tem dificuldade de executá-los por si só. As órteses ativas para membros inferiores têm a função de permitir que pessoas sem movimentos ou com mobilidade reduzida nas pernas possam se locomover de maneira semelhante a um indivíduo saudável, executando movimentos antropomórficos.

2.1 Órtese Ortholeg

Na Universidade Federal do Rio Grande do Norte, foi desenvolvido, pelo grupo de Robótica e Sistemas Dedicados do Departamento de Computação e Automação (DCA/UFRN) um protótipo de órtese ativa de membros inferiores para ser utilizada por paraplégicos, a órtese Ortholeg (M. Araujo and P. Alsina, 2015).

A órtese Ortholeg é constituída por um conjunto de estruturas rígidas unidas por juntas rotacionais com um grau de liberdade cada uma e acionadas por atuadores elétricos fixadas nas articulações dos joelhos e do quadril. Esses quatro atuadores são motores elétricos CC e produzem os movimentos de extensão e flexão em ambas as juntas, possibilitando a realização dos movimentos de caminhada, subir e descer escadas, transpor pequenos obstáculos, sentar e levantar (Araújo, 2009). A Figura 1 mostra a imagem da órtese Ortholeg.

Figura 1 - Protótipo órtese Ortholeg.



Fonte: (Araújo, 2009).

Este protótipo foi projetado inicialmente para ser usada por paraplégicos com nível de lesão muscular T12, com peso entre 50 Kg e 60 Kg e altura entre 1,55 metros e 1,65 metros. A órtese pesa aproximadamente 20 kg incluindo o peso dos equipamentos eletrônicos e baterias. A órtese é fixada no usuário através de calhas plásticas que se ajustam ao corpo (M. Araujo and P. Alsina, 2015).

Um novo modelo da órtese Ortholeg, pensando cerca de 11 Kg sem os componentes eletrônicos, está sendo desenvolvido. A órtese Ortholeg 2.0 utiliza os mesmos motores CC para gerar os movimentos dos membros inferiores, esses movimentos são idênticos aos movimentos da versão antiga, porém a diferença é que existe a possibilidade de o usuário escolher se a junta do joelho pode ser passiva ou não. Essa possibilidade foi atribuída à órtese Ortholeg 2.0 com a intenção de um usuário com menor grau de lesão medular, que ainda detém os movimentos dos joelhos, não venha a necessitar acionar os motores (M.Gloger, 2015). O modelo da órtese Ortholeg 2.0 é mostrado na Figura 2.

Figura 2 - Órtese Ortholeg 2.0



Fonte: (M.Gloger, 2015)

A órtese Ortholeg 2.0, em comparação a versão antiga, suporta um usuário com peso de 70 Kg e altura acima de 1,65 m, este fato se dá pelo motivo da estrutura ser composta por tubos de alumínio mais resistentes e sua estrutura possibilita um ajuste de acordo com o corpo da pessoa.

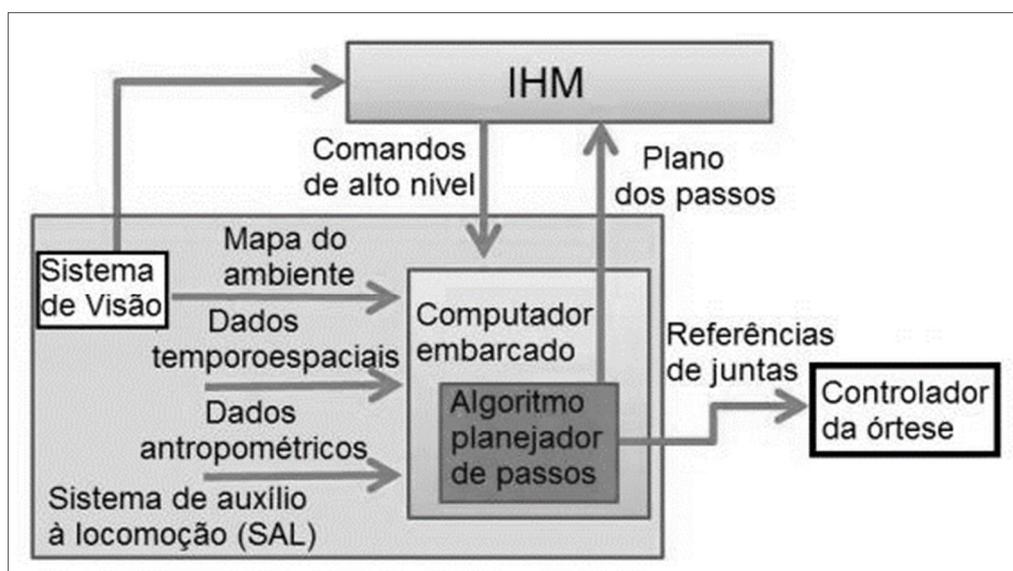
2.2 Arquitetura embarcada

A arquitetura presente na órtese Ortholeg é subdividida em: arquitetura de software, que tem a tarefa de fazer todo o planejamento de caminhada (trajeto ou percurso) e que possibilita também passar por cima ou desviar de obstáculos, e arquitetura de hardware, a qual utiliza um computador embarcado, uma placa de acionamento de motores CC e sensores.

2.2.1 Arquitetura de Software existente

A arquitetura de software presente na órtese Ortholeg é composta por um algoritmo de planejamento de passos que é executado dentro de um computador embarcado, com base nas informações que são enviadas pela interface homem-máquina através dos botões existentes na muleta, pelo sistema de visão ou pelo sistema de auxílio à locomoção (SAL). Como mostra a figura 3.

Figura 3 . Arquitetura de Software presente na órtese Ortholeg.



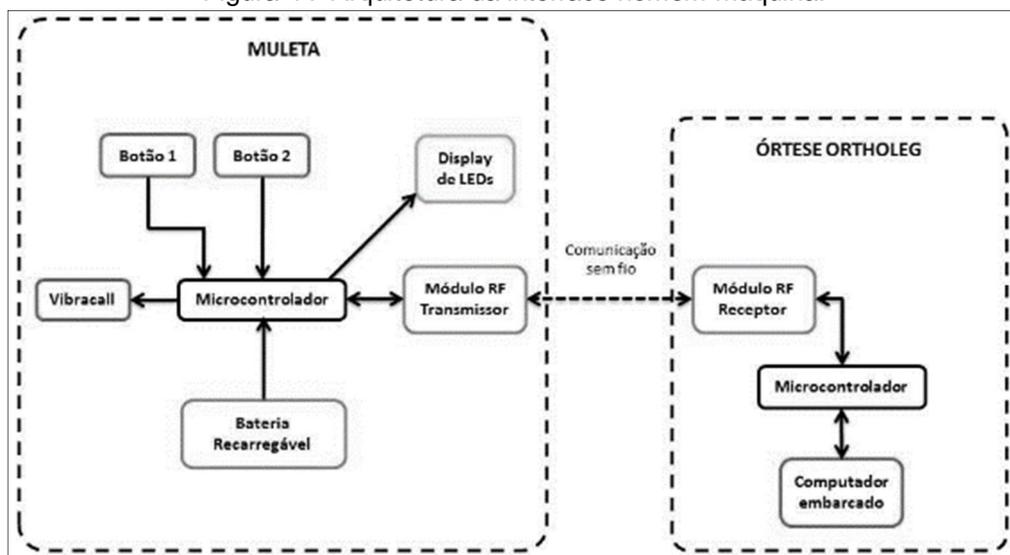
Fonte: (Araújo M.V. 2015).

A interface homem-máquina (IHM) envia um comando contendo o tipo de movimento, selecionado pelo usuário, para o algoritmo planejador de passos, que por sua vez executa as referências de juntas de acordo com o comando selecionado. O sistema de visão envia os dados referentes ao obstáculo e a distância em que se encontra a órtese do obstáculo. O SAL, com base nas características do usuário, como por exemplo, a altura e o tamanho do passo, planeja a trajetória e o envia para o algoritmo. (Araújo M.V. 2015).

2.2.2 Arquitetura de Hardware existente

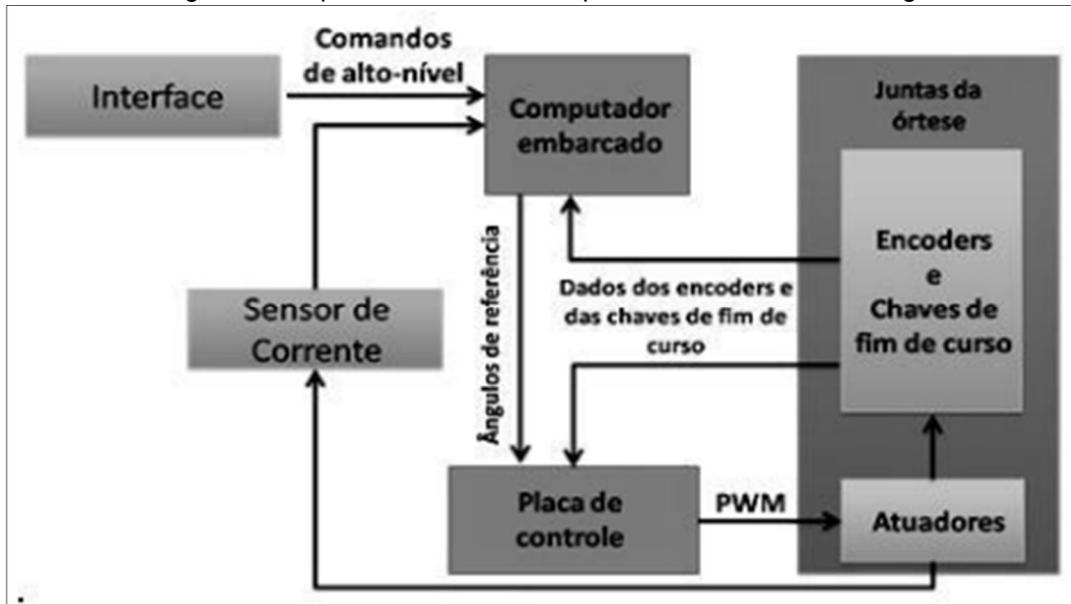
O sistema eletrônico que compõe esta arquitetura é organizado da seguinte maneira: um computador embarcado que tem a função de receber informações dos sensores instalados na órtese, receber os comandos de alto-nível, selecionado pelo usuário, através da interface homem-máquina (IHM) (Figura 4), e gerar dados de referência angulares de acordo com o movimento desejado, uma placa de controle dos motores, responsável por fazer o controle em malha fechada dos movimentos, tendo como a posição atual dos motores os valores de pulsos que os sensores de Encoder geram à medida que o motor gira. (M. Araujo and P. Alsina, 2015). A Figura 5 mostra a arquitetura de hardware presente na órtese.

Figura 4 . Arquitetura da interface homem-máquina.



Fonte: (Santos M.R. 2015).

Figura 5 - Arquitetura de Hardware presente na órtese Ortholeg.



Fonte: (Costa F. P. SBAI 2015)

A motivação para criar uma nova arquitetura de *hardware* e *software* se deu pelo fato de que a arquitetura existente não é aberta e modular, isto é, não possibilita a inclusão de novos sensores para avaliar o esforço físico do usuário e nem o desempenho da órtese Ortholeg e o seu código é fechado, o usuário não tem a liberdade de alterar o código para realizar ações diferentes.

3 ARQUITETURA DE *HARDWARE* E *SOFTWARE* PROPOSTA

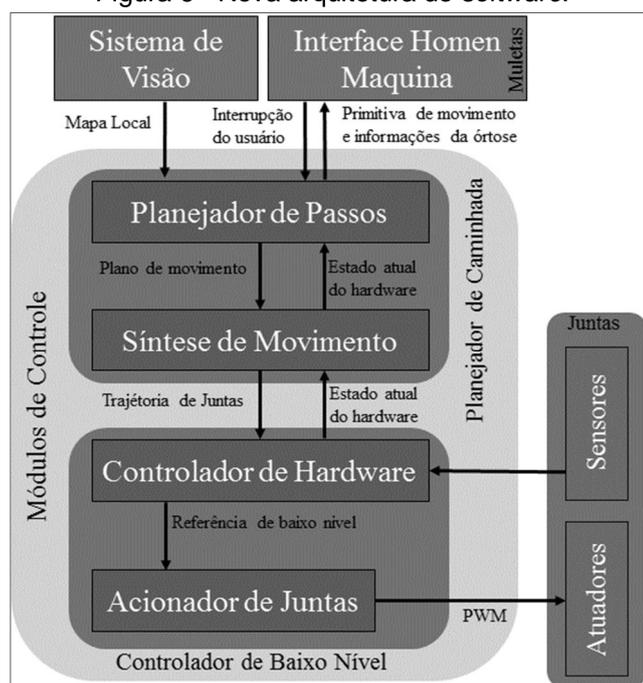
Para desenvolver a nova arquitetura, realizou-se uma organização das tarefas existentes da arquitetura anterior que possibilitou a inclusão de novas tarefas, foi implementado um algoritmo de controle PID para o controle dos atuadores e realizado uma conexão entre os dispositivos através de um protocolo de comunicação.

Portanto, na organização das tarefas, a nova arquitetura de *software* foi dividida em quatro módulos, cada um responsável por uma determinada ação de grande importância. A organização desses módulos se encontra da seguinte maneira: 1ª módulo: Sistema de visão; 2ª módulo: Planejador de movimento; 3ª módulo: Interface homem máquina (IHM); e o 4ª módulo: Controlador. A arquitetura de *hardware* ainda é constituída por um computador embarcado (microcomputador) responsável por processar dados de alto nível e realizar a comunicação com a IHM, e por uma plataforma eletrônica microcontrolada responsável por controlar as referências de juntas e as leituras dos sensores. Uma diferença entre a arquitetura de *hardware* antiga para a nova é que a placa de acionamento utilizada nessa nova arquitetura é a placa *TqRex Robot Controller*, no capítulo 4 apresentam-se as principais características dessa placa.

3.1 Arquitetura de Software

A arquitetura de *software*, conforme apresentado anteriormente, é dividida em: Sistema de visão, planejador de movimento ou caminhada, IHM e o controle. A figura 6 apresenta o diagrama da arquitetura proposta.

Figura 6 - Nova arquitetura de software.



Fonte: Ortholeg 2.0 . *Design of a Transparent Active Orthosis* (Artigo a ser submetido).

O sistema de visão incorpora uma câmera conectada à órtese para processar as imagens do ambiente. Esse sistema tem a principal função de identificar alterações no ambiente que estejam dentro da área de movimento e guardar as informações coletadas em um mapa local que é enviado para o planejador de passos. O sistema de visão não é tema deste trabalho e está sendo desenvolvido em outra pesquisa.

O planejador de movimento, ao receber o mapa local, interpreta as informações utilizando uma função característica deste módulo chamada de planejador de passos que tem a obrigação de determinar se o movimento pode ser realizado ou não. Esse movimento pode ser alterado devido a uma simples mudança na extensão máxima de uma junta, para o caso de transposição de um ou mais obstáculos, ou na mudança do ciclo de marcha, no caso de subir ou descer degraus. O usuário tem a opção de interromper o movimento planejado a partir da IHM.

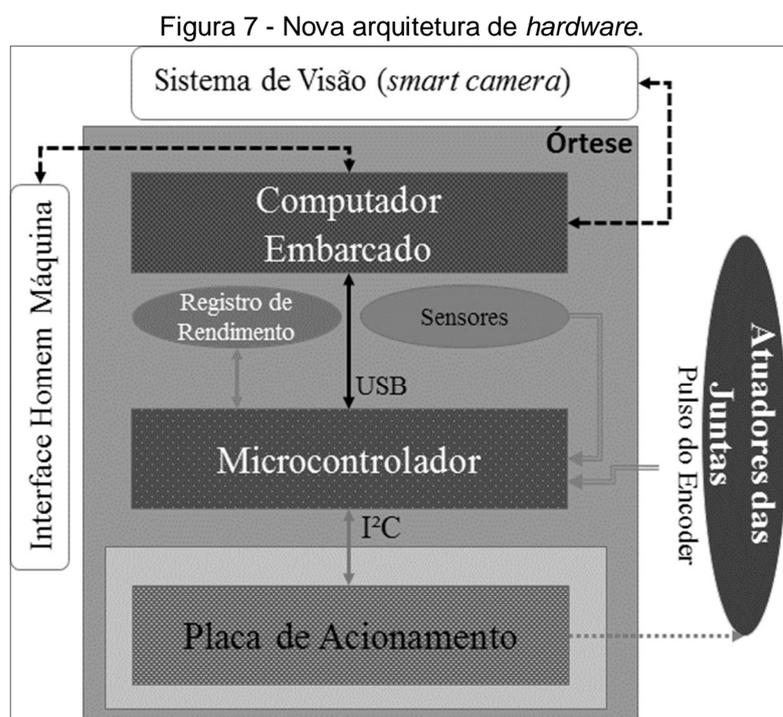
O objetivo principal da interface homem máquina é servir de conexão entre o usuário e o sistema autônomo geral da órtese. Neste módulo o usuário tem a possibilidade de visualizar e comandar os movimentos a serem realizados pela órtese. Se uma nova sequência de movimento for requisitado pelo usuário, essa sequência é realizada depois da verificação do planejador de passos.

Após a verificação do planejador de passos, um plano de movimento é enviado para a síntese de movimento a qual é responsável por criar uma trajetória de ângulos de juntas para a realização do movimento planejado.

3.2 Arquitetura de Hardware

A primeira versão da arquitetura de *hardware* da órtese Ortholeg sobrecarrega o computador central do dispositivo. Isso faz com que qualquer limitação de *hardware* comprometa o desempenho da órtese. Por exemplo, não existe processamento suficiente na órtese Ortholeg para executar funções de controle de baixo nível e planejamento de caminhada mais elaborada simultaneamente. Além disso, alguns dos dispositivos de *hardware* presentes na órtese não possuem código aberto, ou seja, o usuário fica impossibilitado de consultar ou modificar o algoritmo da forma que lhe convém.

Para se alcançar o perfeito funcionamento e adequar as funcionalidades de cada módulo da arquitetura de *software* apresentada, é importante criar uma arquitetura de *hardware* que possibilite a execução dos movimentos. Na Figura 7 é apresentado um diagrama geral da arquitetura de *hardware*.



Fonte: Ortholeg 2.0 . *Design of a Transparent Active Orthosis* (Artigo a ser submetido).

A IHM é representada por um par de botões nas muletas. Futuramente pode ser incluído, na IHM, *display* ou até mesmo dispositivo de realidade aumentada, como um OHMD (*optical head-mounted display*). A arquitetura de *hardware* permitirá também que o usuário possa usar um computador (*Desktop*) para analisar as condições atuais da órtese, por exemplo, a carga atual das baterias, o consumo energético ou até mesmo visualizar seu histórico de uso (distância percorrida).

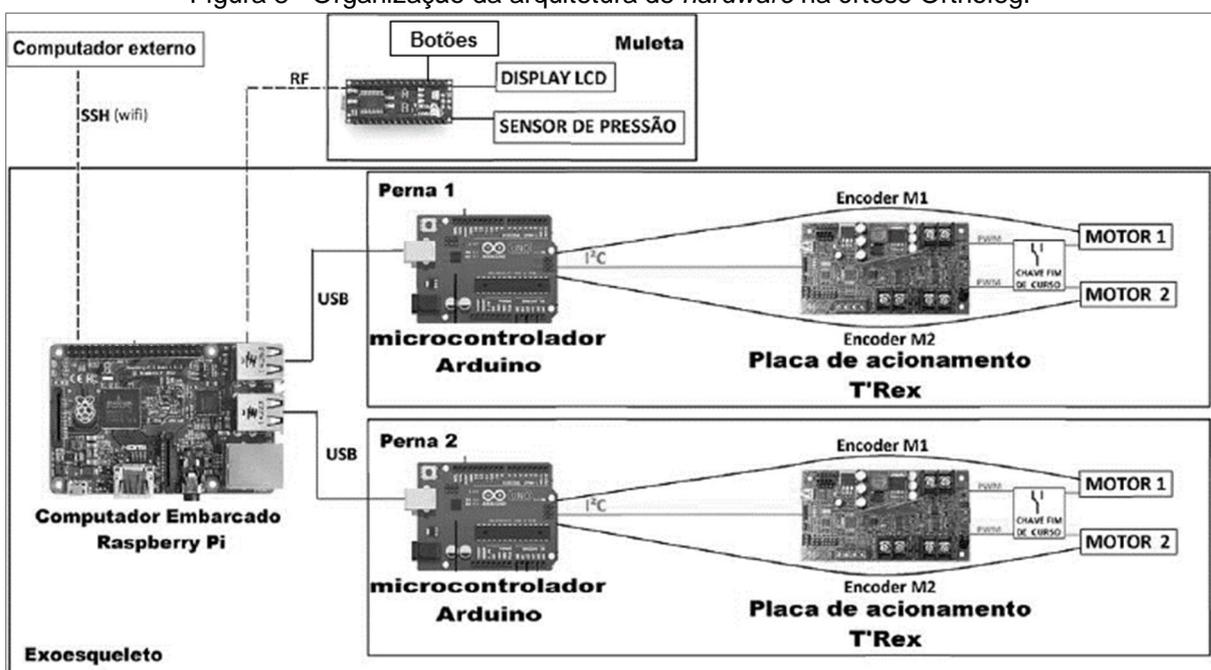
O computador embarcado (microcomputador) é encarregado de processar os dados de alto nível e fazer a comunicação com a IHM. O computador também executa as duas funções do planejador de caminhada: planejador de passos e síntese de caminhada.

A plataforma eletrônica microcontroladora controla as referências de juntas e faz a leitura de sensores ligados à órtese.

E por último, a placa de acionamento, a qual aciona as juntas do quadril e do joelho da órtese Ortholeg.

A Figura 8 ilustra a organização da arquitetura de *hardware* de maneira simples e clara para facilitar o entendimento do observador.

Figura 8 - Organização da arquitetura de *hardware* na órtese Ortholeg.



Fonte: Próprio autor.

4 DISPOSITIVOS DE HARDWARE

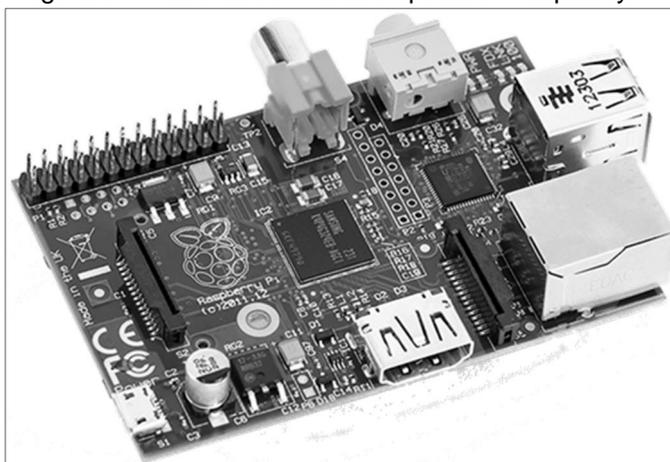
Neste capítulo serão apresentados os dispositivos eletrônicos, tais como uma plataforma eletrônica Arduino Mega, um computador embarcado Raspberry Pi e a placa de acionamento TdRex, que compõe a nova arquitetura de hardware proposta possibilitando o acionamento dos motores.

4.1 Microcomputador Raspberry Pi

Em uma arquitetura de *hardware* e *software* a utilização de um microcomputador (computador embarcado) é essencial no processamento dos dados, os quais realizam um conjunto de tarefas predefinidas, e pela sua segurança. O microcomputador Raspberry Pi tem a função de processar as informações que são recebidas, via ondas de rádio, pela muleta. Essas informações são as operações (andar, parar, levantar, sentar, etc.) escolhidas pelo usuário através dos botões. Depois de processar as informações desejadas, são enviados os dados acerca do movimento selecionado pelo indivíduo para a plataforma eletrônica Arduino.

O computador embarcado Raspberry Pi é um microcomputador que suporta diversas distribuições do sistema operacional Linux. De maneira oposta a um poderoso *desktop*, ele conta com um processador ARM que produz menos calor e consome menos energia. Este computador é usado em uma infinidade de projetos, como por exemplo, em robôs, em satélites de baixo custo, em automação residencial e entre outros, devido as suas especificações e o baixo custo.

Figura 9 . Modelos do microcomputador Raspberry Pi.



Fonte: www.raspberrypi.org (2016).

O modelo utilizado no projeto, foi o modelo Raspberry Pi B (Figura 9), o qual tem as seguintes especificações:

- Duas vias USB;
- Sistemas operacionais: Debian, Fedora, GNU/Linux e Raspbian;
- Armazenamento *onboard* . Cartão SD;
- Porta *Ethernet*;
- Saída de vídeo HDMI;
- Pinos de entrada e saída digital;
- Pinos de comunicação serial (TX e RX) e de comunicação via protocolo I²C;

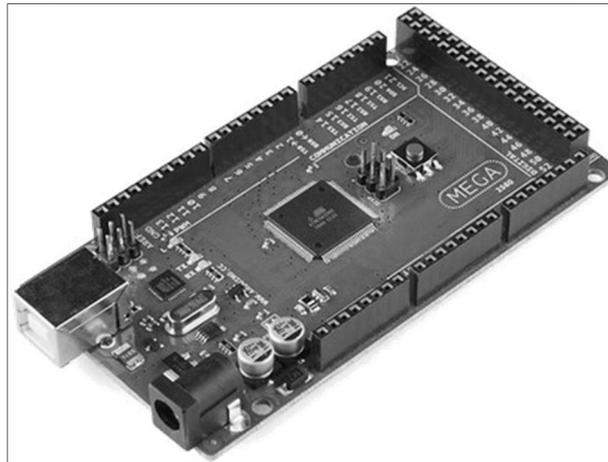
Além das especificações citadas acima, o microcomputador Raspberry Pi suporta as linguagens de programação C, C++ e Python os quais muitos programadores estão habituados.

4.2 Plataforma eletrônica Arduino

Na nova arquitetura de *hardware* é utilizada uma plataforma eletrônica Arduino que tem como principal objetivo enviar sinais de controle provenientes do controlador PID (controlador proporcional integral derivativo) para controlar os atuadores, com o propósito de se obter uma melhora significativa do sistema controlado, calculando um erro entre o valor medido na saída (pulsos) e o valor desejado no processo (dados recebidos pelo computador embarcado) fazendo com que o sinal de erro seja minimizado.

A plataforma eletrônica Arduino é uma placa composta por um microcontrolador Atmel e por um ambiente de desenvolvimento para escrever programas. A plataforma Arduino é usada para desenvolver diversos projetos que utilizam sensores, chaves, motores e outras saídas físicas. Em termos de software, a plataforma Arduino pode ter funcionalidades desenvolvidas utilizando as linguagens Assembly, C e C++ que podem ser executadas pelo dispositivo.

Figura 10 . Plataforma eletrônica Arduino Mega.



Fonte: www.arduino.cc (2016).

Para o desenvolvimento da arquitetura de hardware, foi utilizada uma plataforma Arduino Mega (Figura 10) que tem as configurações necessárias para o projeto, as quais são:

- EEPROM de 4 KB;
- Opera com uma fonte externa entre 7 a 12 volts;
- 54 pinos digitais de entrada ou saída, dos quais 15 fornece uma saída de sinal PWM;
- 16 pinos de entrada analógica que podem ser conectados sensores analógicos.

Além dessas configurações, existem alguns pinos que têm a função de comunicar com outros dispositivos de hardware e que usam algum tipo de protocolo:

- Comunicação Serial: Os pinos com as identificações TX e RX são utilizados para transmitir e receber dados, respectivamente.
- Comunicação SPI (*Serial Peripheral Interface*): é um protocolo de dados serial síncrono, onde um dispositivo é o mestre (*Master*) que envia os dados para os outros periféricos que são os escravos (*Slave*). Nesse protocolo são utilizados os pinos 50 (MISO . dados do *Slave* para o *Master*), pino 51 (MOSI . dados do *Master* para o *Slave*) e o pino 53 (SS . seleciona qual *Slave* receberá os dados). Para acessar este protocolo é necessário acessar a biblioteca de SPI.

- Comunicação I²C (*Inter-Integrated Circuit*): protocolo de comunicação que permite, através de barramento, que qualquer dispositivo *Master* troque informação com qualquer dispositivo *Slave*.

A plataforma Arduino também possui mais dois pinos especiais:

- Pinos de saída de sinal PWM: que controla o valor da alimentação entregue ao dispositivo conectado a este pino.
- Pinos de interrupções externo: é utilizado para interromper uma determinada ação do programa ou analisar algum dado de entrada.

O motivo de escolher a plataforma Arduino Mega foi pelo fato de existirem muitos pinos de saída e entrada (digital e analógica) e pela sua velocidade de comunicação entre os dispositivos externos.

4.3 Placa de acionamento T~~R~~ex Robot Controller

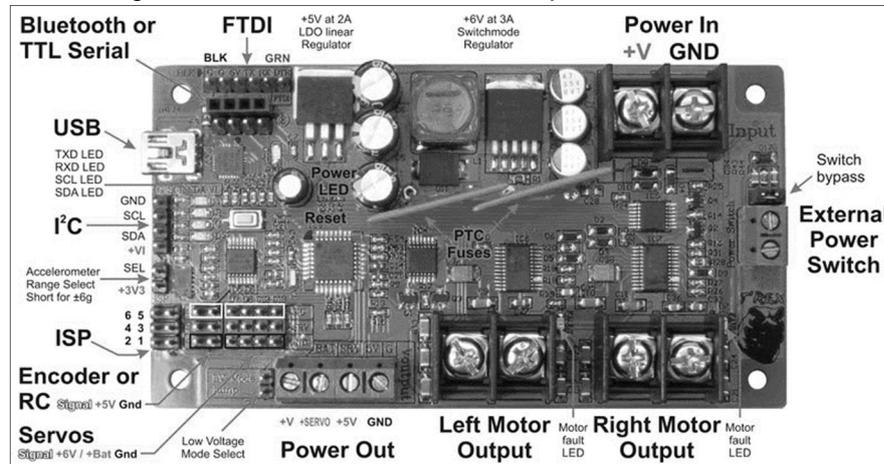
A placa de acionamento utilizada na arquitetura de *hardware* foi a T~~R~~ex Robot Controller, que é uma combinação de uma plataforma Arduino com circuito de acionamento. A placa T~~R~~ex tem o objetivo de acionar os atuadores das juntas do quadril e joelho fazendo com que a perna entre em movimento.

A preferência pela escolha dessa placa de acionamento foi pelo fato dela possuir código aberto, que pode ser facilmente acessado e alterado, pode trabalhar em uma ampla gama de tensões, de 6V a 30V e o baixo custo no mercado. Além dessas vantagens, a placa T~~R~~ex tem as seguintes características (Figura 11):

- *Bootloader* Arduino Nano w/ Atmega 328
- Corrente por motor de 9A . 40 A de pico;
- Tem sensores de fábrica para medição de consumo de corrente para cada motor;
- Acelerômetro MMA7361L com sensibilidade de $\pm 1.5g$ de três eixos que fornece dados de ângulos, aceleração e impacto;
- Cada motor tem frenagem eletrônica variável independente;
- Fusível PTC (Coeficiente de Temperatura Positivo) de 30V/9A para evitar danos causados por motores;

- Possui quatro entradas digitais para conexão de sensor Encoder.

Figura 11 . Placa de acionamento TqRex Robot Controller.



Fonte: (DAGU, 2016).

A placa TqRex nos fornece vários tipos de comunicação, tais como:

- Comunicação I²C;
- Comunicação TTL Serial (*Transistor-Transistor Logic*) . é um tipo de comunicação que recebe ou envia sinais com tensões entre 1.8 V e 5 V que pode ser conectada nas interfaces UART e USB utilizando um conversor TTL;
- Comunicação FTDI;
- Comunicação via Bluetooth . Onde os dados são enviados e recebidos através de um protocolo padronizado através de uma ligação sem fio de 2,4 GHz;
- Comunicação via rádio frequência.

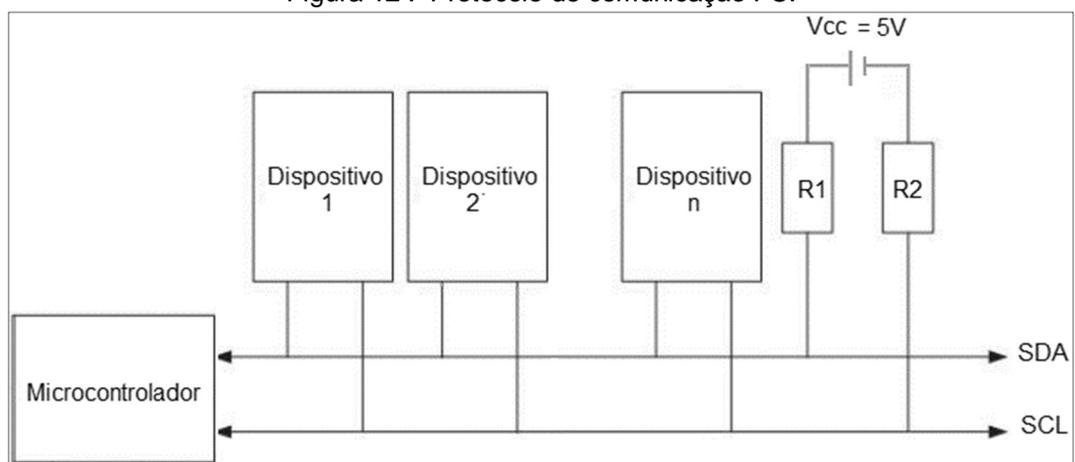
Em termos de linguagem de programação, a placa TqRex pode ter suas funcionalidades desenvolvidas utilizando as linguagens C e C++.

5 PROTOCOLO DE COMUNICAÇÃO I²C

O protocolo de comunicação I²C (*Inter Integrated Circuit*) é utilizado para a conexão de periféricos de baixa velocidade a uma placa mãe, a um sistema embarcado ou um telefone celular. Constitui uma interface simples e possibilita a troca eficaz de informações entre os dispositivos. A comunicação I²C foi criada na metade dos anos 90 pela empresa *Philips Semiconductor Division*, a atual NXP (NXP, 2014), com o propósito de reduzir a complexidade dos circuitos e assim melhorar sua produção. O envio de dados entre os dispositivos acontece através de dois fios: o SDA (*Serial Data*) linha bidirecional de dados e o SCL (*Serial Clock*) linha unidirecional de *clock*.

Duas características importantes deste protocolo de comunicação são: que não tem um valor fixo de frequência de transmissão de dados e tem a possibilidade de utilizar dispositivos distintos sem que haja incompatibilidade e nem interferência na comunicação. Cada dispositivo possui um endereço fixo e pode ser configurado para enviar ou receber dados pelo barramento (Figura 12), caracterizando os estados de mestre (*Master*) e de escravo (*Slave*), respectivamente. O protocolo I²C é também *multimaster*, isso significa que mais de um dispositivo pode ser configurado como *Master*.

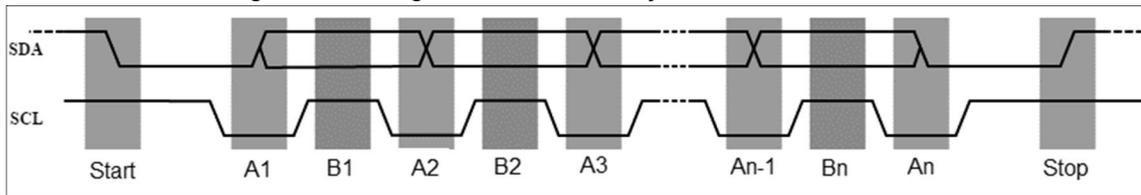
Figura 12 . Protocolo de comunicação I²C.



Fonte: Próprio autor.

A comunicação entre os dispositivos conectados ao barramento I²C é organizada da seguinte maneira (Figura 13):

Figura 13 . Diagrama de comunicação via barramento I²C.



Fonte: I2C-bus Specification and User Manual . NXP (2014).

- *Start* . condição de início quando o valor de bit do SDA vai para o nível baixo e o valor de bit do SCL continua em nível alto. Enquanto que a condição de *Stop* não ocorrer, o barramento é considerado ocupado;
- Quando o SCL está em nível baixo, o SDA está livre para enviar o primeiro dado (trechos A1, A2, A3, An-1 e An).
- A informação é recebida pelo dispositivo quando o SCL muda de estado de nível baixo para o nível alto (B1, B2, Bn);
- *Stop* . Quando o SDA muda de estado de nível baixo para o nível alto e o SCL permanece no nível alto;

Além dessa organização do envio de dados através do barramento, os dados podem ser organizados de outras maneiras, como por exemplo, em um pacote de oito bits (um byte), após o *start*, o primeiro bit é enviado pelo *master* contendo o endereço do *slave* que receberá os dados; depois que o *slave* é identificado, o mesmo enviará um bit de resposta indicando que está livre para receber os dados; após o *master* ter recebido o bit de resposta, este envia o dado através de seis bits e o último bit é o bit de *stop* indicando que a mensagem foi concluída.

Portanto, a escolha deste protocolo de comunicação na arquitetura de *hardware* se deu pelo fato de ser uma comunicação segura e fácil de implementar, além da possibilidade de conexão com vários dispositivos em um mesmo barramento, incluindo a opção de adicionar mais de um *master*.

6 CONTROLADOR

Com o contínuo avanço tecnológico, a automatização de diversos processos vem sendo base para o surgimento de novos controladores que podem ser classificados em: controladores convencionais ou clássicos e os controladores avançados (Unbenhauen, 1996). Os controladores do tipo PID (Proporcional . Integral . Derivativo) e em cascata são considerados convencionais, enquanto que os adaptativos, robustos, preditivos, não lineares e inteligentes são considerados avançados.

Neste capítulo será apresentado o controlador convencional do tipo PID utilizado na arquitetura de *software* e o método aplicado para a obtenção dos parâmetros do controlador.

6.1 Controle PID

O controlador implementado na arquitetura de *software* foi do tipo PID por ser utilizado com mais frequência nas indústrias e em sistemas de controle de processos, além do baixo custo, simples implementação e capaz de exercer um desempenho satisfatório, quando sintonizado adequadamente. No entanto, de acordo com os autores Astrom e Hanglung (ASTRON e HANGGLUND, 1995), apenas 20% das malhas de controle funcionam adequadamente com este tipo de controlador. Ainda assim, o controlador PID continua sendo o mais utilizado, esta afirmação pode ser confirmado por meio de algumas citações.

One of the most powerful but complex mode operations combines the proportional, integral, and derivative modes. This system can be used for virtually any process condition.
(JOHNSON, 1997, p. 424).

Despite the development of advanced process control techniques, proportional-integral (PI) control is still the most commonly used control technique in the process industries. The main reasons for this popularity are that PI controllers are often

effective and are easy to implement and maintain by plant personnel.+(CHEN e SEBORG, 2003, p. 27).

Com base nessas citações é possível afirmar que em sistemas de controle de processos, sabe-se que os esquemas básicos de controle PID provaram sua utilidade conferindo um controle satisfatório, embora em muitas situações eles podem não proporcionar um controle ótimo.

6.2 Parâmetros do controle PID

No controlador PID, existem três ações que, quando sintonizados corretamente, têm grande importância para alcançar a resposta ideal do sistema, garantindo um controle estável e preciso. Essas ações são: o proporcional (P), o integral (I) e o derivativo (D). A tabela 1 apresenta as características de cada ação (ARAÚJO 2007).

Tabela 1: Parâmetros do controlador PID.

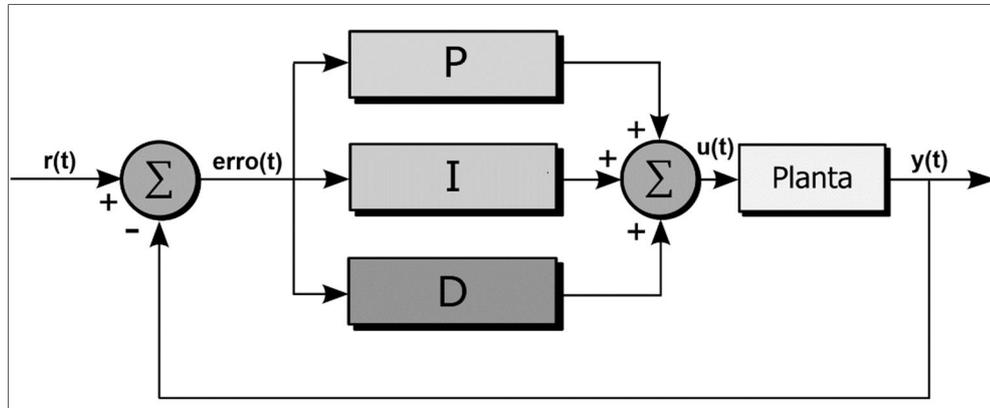
Parâmetro	Ação	Descrição
P	Produz um sinal de saída que é proporcional ao erro	Ir� fornecer uma resposta mais r�pida do sistema de acordo com a varia�o do sinal de entrada.
I	Produz um sinal de sa�da que � proporcional � magnitude e � dura�o do erro	Atua ao longo do tempo enquanto houver diferen�a entre o valor desejado e o valor controlado.
D	Produz um sinal de sa�da que � proporcional � taxa de varia�o do erro.	Antecipa a corre�o do valor de sa�da, reduzindo a varia�o da vari�vel controlada.

Fonte: www.novus.com.br (2016).

Para que o sistema exiba as propriedades desejadas,   necess rio ajustar os valores de cada a o atrav s das equa es matem ticas do PID. O valor calculado   transformado em um sinal adequado para o sistema utilizado. A Figura 14

apresenta o diagrama de blocos de malha fechada de um sistema com o controle PID.

Figura 14 . Diagrama de blocos com controle PID.



Fonte: www.csimn.com (2016).

Como pode ser visto na Figura 14, a equação mais comum do PID é organizada da seguinte forma:

$$u(t) = P + I + D \quad (1)$$

No qual, a ação proporcional (P) é definida por:

$$P = K_p \times e(t) \quad (2)$$

Onde

- K_p : Ganho proporcional;
- $e(t)$: Diferença entre o *Setpoint* (SP) e a variável de processo (PV). PV é a variável a ser controlada, no caso da órtese Ortholeg, a variável a ser controlada é o ângulo de junta. O SP é o valor que deseja alcançar pela PV. O SP e o PV são expressos nas mesmas unidades.

A resposta proporcional pode ser ajustada a partir da constante de ganho K_p , ou seja, aumentando o ganho proporcional irá aumentar a velocidade da resposta do sistema de controle. No entanto, se o ganho proporcional é muito grande, a variável de processo começará a oscilar. Porém se o valor do ganho K_p for muito pequeno, maior será o tempo de resposta.

A ação integral (I) é definida por:

$$I = K_i \times \int_0^t e(\tau) d\tau \quad (3)$$

Onde

- K_i : Ganho integral, onde, $K_i = \frac{1}{\tau_i}$;
- i : Constante de tempo integrativo.

De maneira oposta ao ganho proporcional, quando atribuído um valor muito baixo ao ganho K_i , o sistema pode levar muito tempo para atingir o valor de referência. No entanto, quando atribuído um valor alto, o sistema pode tornar-se instável e também pode provocar um *overshoot* ainda maior, ou seja, quando a PV ultrapassa o valor do SP.

A ação derivativa (D) é definida por:

$$D = K_d \times \frac{de(\tau)}{d\tau} \quad (4)$$

Onde

- K_d : Ganho derivativo, onde, $K_d = d$.
- d : Constante de tempo derivativo

Aumentando a constante de tempo derivativa fará com que o sistema de controle reaja mais fortemente a mudanças, aumentando a velocidade da resposta de controle do sistema. Na prática, a maioria dos sistemas de controle utilizam a constante de tempo derivativo (d) muito pequeno, pois a ação derivativa tem a desvantagem de amplificar os sinais de ruído, o que pode causar um efeito de saturação nos atuadores do sistema.

6.3 Métodos para obtenção dos parâmetros.

Existem diferentes métodos para determinar os valores dos parâmetros do controle PID, nesta seção serão apresentados o método de tentativa e erro (*guess and check*) e o método de Ziegler - Nichols.

No método de tentativa e erro, inicialmente é atribuído valor nulo aos ganhos K_i e K_d , já o valor da ação P é alterado conforme o aumento da constante de ganho K_p até que o sistema comece a oscilar. Com o aumento do ganho proporcional, o sistema torna-se mais rápido, porém não se deve deixar o sistema ficar instável.

Uma vez que o ganho K_p atingiu um valor o qual obteve uma resposta rápida desejada, o ganho K_i , por sua vez, é aumentado com o propósito de parar as oscilações. O aumento do ganho K_i faz com que o erro de estado estacionário seja reduzido, porém quanto maior o valor do ganho K_i , maior será a ação I consequentemente maior será o *overshoot*.

Para um sistema rápido, um valor de *overshoot* é necessário de modo que possa responder imediatamente às mudanças.

Após ter definido os valores dos ganhos K_p e K_i para que o sistema de controle se torne rápido, a constante de ganho K_d é aumentada até que o sistema torne ainda mais rápido. Com o aumento do ganho K_d o *overshoot* tende a diminuir, porém o sistema fica sensível ao ruído.

Os métodos de Ziegler-Nichols são cruciais para sistemas para os quais não se dispõe de modelos matemáticos. Um dos métodos consiste em determinar o valor do ganho proporcional através de um valor de ganho crítico (K_{cr}), em que o sistema em malha fechada entra em oscilação.

Para obter o valor da constante K_{cr} é necessário programar um controlador PID e então configura-lo para funcionar como um controlador P definindo as constantes de tempo $i = \infty$ e $d = 0$. O ganho K_{cr} é então aumentado até que a saída do sistema apresente oscilações, o período dessas oscilações é chamado de período crítico (P_{cr}).

Uma vez determinados os valores de K_{cr} e P_{cr} , eles são usados para determinar os parâmetros do controlador, de acordo com a tabela 2.

Tabela 2 . Regra de sintonia de Ziegler-Nichols baseada no K_{cr} e P_{cr} .

Tipo de controlador	K_p	i	d
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$\frac{0,5P_{cr}}{1,2}$	0

PID	0,6Kcr	0,5Pcr	0,125Pcr
------------	--------	--------	----------

Fonte: <http://www.univasf.edu.br/~eduard.montgomery/> (2014).

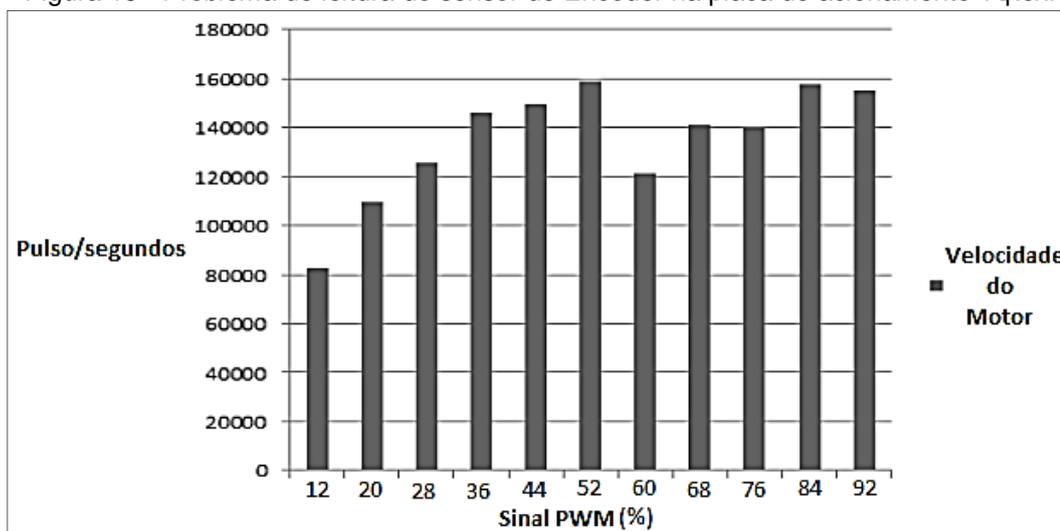
Os métodos de Zieger-Nichols vêm sendo muito utilizados para sintonizar os controladores PID em que a dinâmica da planta não é conhecida. Porém é importante destacar que: em sistemas que não estabilizam, independentemente do valor do ganho, não é possível aplicar esse método de Zieger-Nichols; e os métodos de Ziegler-Nichols fornecem aproximações nos resultados dos parâmetros, sendo necessário um ajuste posterior.

7 RESULTADOS EXPERIMENTAIS

Inicialmente foi planejado atribuir à placa de acionamento TdRex um algoritmo de controle PID, por isso foi conectado um sensor de Encoder em uma das entradas digitais da placa de acionamento para medir a velocidade de rotação e posição angular do motor através dos pulsos. Após a conexão do sensor de Encoder, foi realizada comunicação via protocolo I²C da plataforma eletrônica Arduino com a placa de acionamento TdRex, onde a plataforma Arduino enviava valores diferentes de sinais PWM para a TdRex, acionando um dos motores.

Foi observado que em altos valores de sinais PWM (*Pulse Width Modulation*), valores acima de 52% (Figura 15), os dados coletados do sensor não estavam coerentes com a velocidade de giro do motor, ou seja, em altas velocidades alguns pulsos eram perdidos na leitura do sensor durante os testes.

Figura 15 - Problema de leitura do sensor de Encoder na placa de acionamento TdRex.



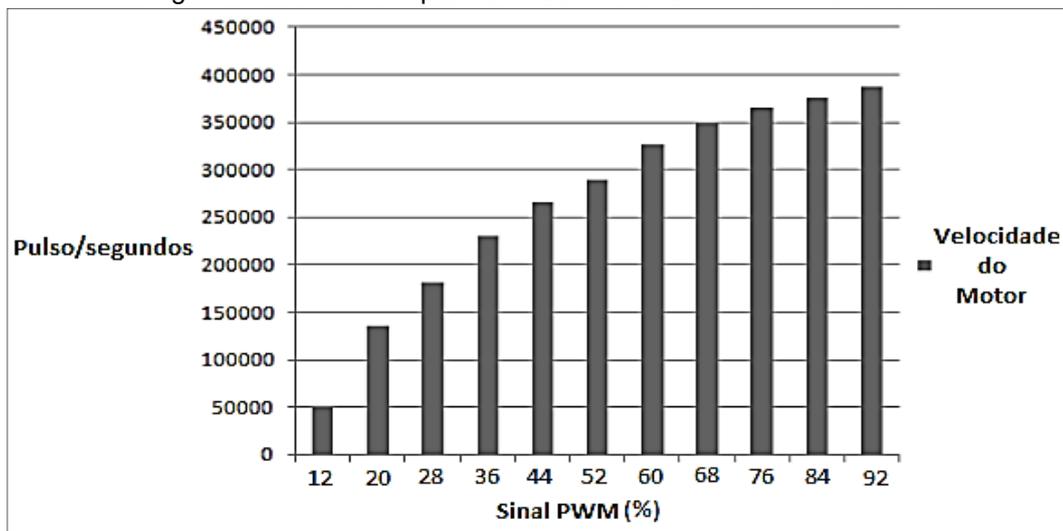
Fonte: Próprio autor.

Ao analisar mais a fundo para saber qual era o motivo do problema, foi notado que a frequência de leitura do processador da placa de acionamento é menor que a frequência em que os pulsos eram enviados pelo sensor, fazendo com que alguns desses pulsos fossem perdidos em altos valores de PWM.

Para resolver esse problema, o sensor de Encoder foi conectado em um dos pinos especiais da plataforma Arduino que é adequado para capturar os dados que são enviados pelo sensor, resolvendo o problema.

Portanto, os testes foram feitos da seguinte maneira: foi tirada a média de pulsos de dez ensaios com diferentes porcentagens de sinal PWM com duração de cinco segundos cada. Os resultados obtidos foram coerentes com a velocidade de giro do motor, quanto maior a porcentagem do sinal PWM maior é a velocidade e, conseqüentemente, maior a quantidade de pulsos lidos pela plataforma eletrônica com o mesmo intervalo de tempo. O resultado desses testes é mostrado na Figura 16.

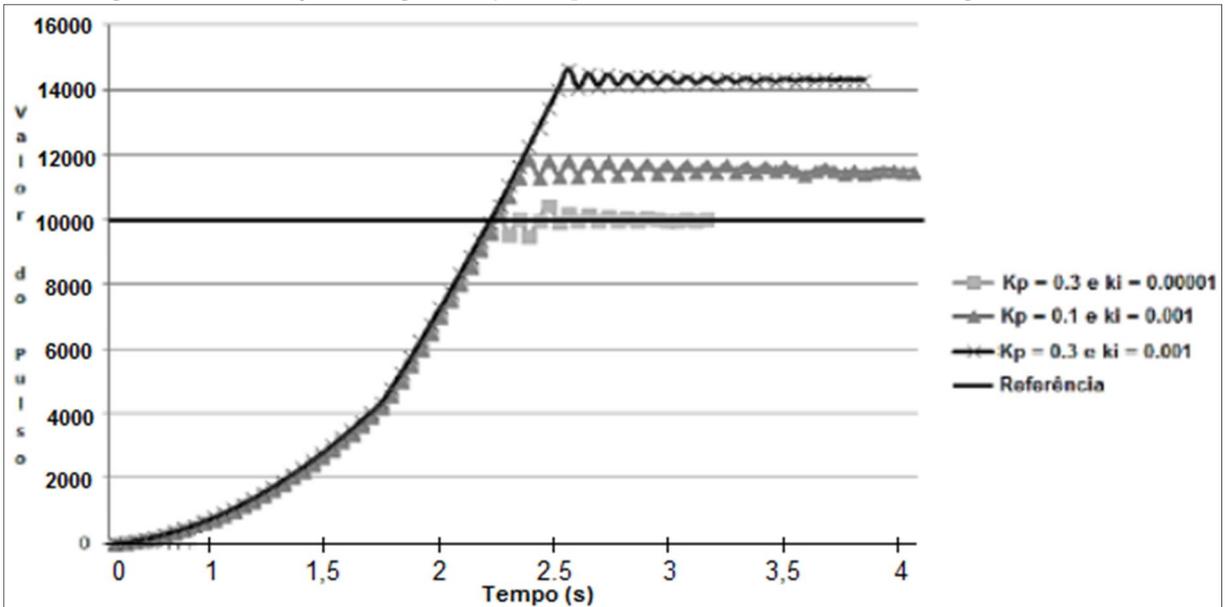
Figura 16 - Valores de pulsos conforme o aumento do sinal PWM.



Fonte: Próprio autor.

Como a plataforma Arduino irá receber o pacote de dados do microcomputador e os dados do sensor de Encoder, foi atribuído à plataforma um algoritmo de controle PID para atuar sobre o sistema e garantir um controle estável e preciso dos atuadores. Primeiramente foi aplicado o método de Ziegler-Nichols para obter os ganhos aproximados de todos os parâmetros do PID (K_p , K_i e K_d). A Figura 17 mostra o resultado obtido na utilização do método de Ziegler - Nichols.

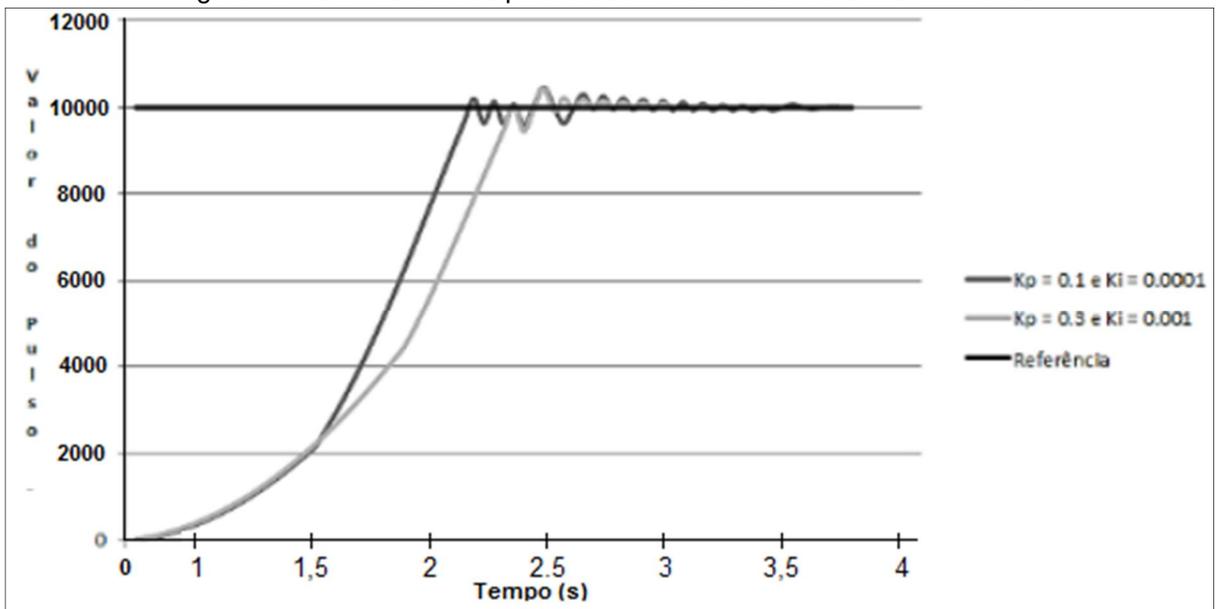
Figura 17 - Obtenção dos ganhos para K_p , K_i e K_d através do método Ziegler . Nichols.



Fonte: Próprio autor.

Em seguida foi utilizado o método de tentativa e erro, tomando como base os valores anteriores obtidos. Chegou-se a uma conclusão de que os valores desses dos ganhos utilizados no controle PID, que proporcionaram melhor estabilidade e controle dos atuadores, foram $K_p = 0.1$ e $K_i = 0.0001$ (Figura 18).

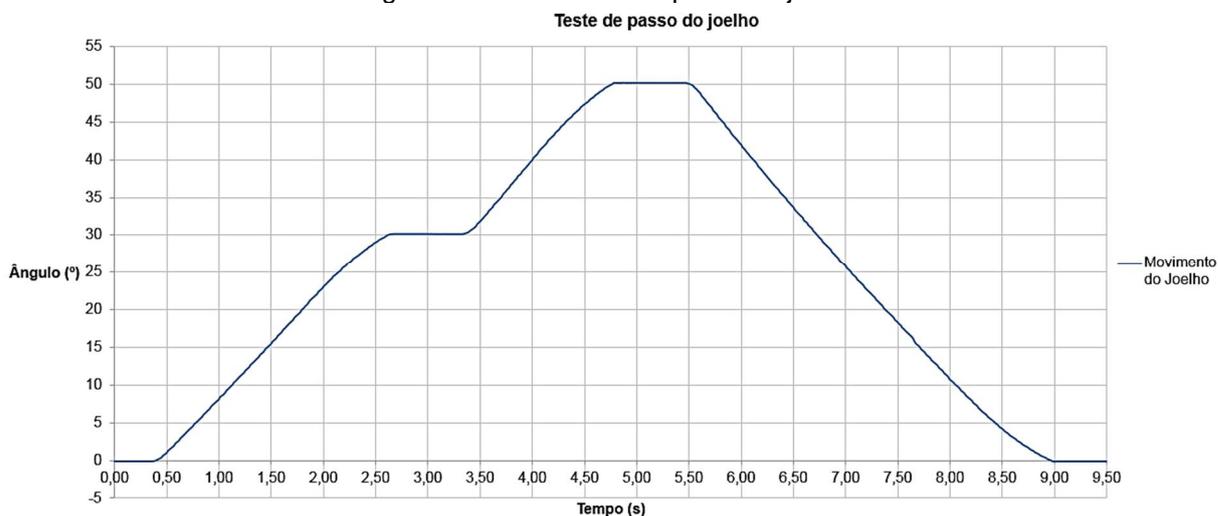
Figura 18 - Ganhos obtidos para a estabilidade e controle dos atuadores.



Fonte: Próprio autor.

A Figura 19 apresenta um simples teste feito de movimento de passo com o atuador do joelho.

Figura 19 - Movimento de passo do joelho.



8 CONCLUSÃO

Os testes feitos com a comunicação via protocolo I²C com a plataforma Arduino, atuando como mestre, e a placa de acionamento T_QRex, como escravo, foram o suficiente para coordenar a comunicação desses dois dispositivos. Não foi percebida nenhuma interferência na comunicação entre esses dois dispositivos. Porém o único e possível problema poderá ser o tempo de comunicação entre esses dois equipamentos, ao incluir mais funções e sensores na plataforma Arduino, podendo comprometer a execução do controle PID.

Os resultados obtidos na captura de valores de pulso usando o sensor de Encoder na plataforma foram bastante significativos, visto que os diferentes valores de sinais PWM que a plataforma Arduino enviava para a placa de acionamento faziam com que o motor fosse acionado aumentando a velocidade de giro dependendo do valor do sinal, depois de um tempo fixo, o movimento do motor é interrompido, e a plataforma, por sua vez, calculava a posição atual do motor através dos pulsos recebidos do sensor.

O controle PID foi executado da forma desejada, calculando a resposta de saída (sinal PWM) ajustando suas entradas através do cálculo proporcional e integral, a partir das informações referentes à trajetória de caminhada e da posição

atual do motor (valores de pulsos), garantindo o funcionamento adequado e estável dos motores.

Para trabalhos futuros, deve ser feita a análise do tempo de comunicação ao incorporar novas funções e sensores na plataforma Arduino, para não prejudicar o envio de pacotes de dados do computador Raspberry Pi na execução da função desejada pelo usuário e na geração dos sinais de PWM através do controle PID. A implementação de novos controladores para analisar o consumo e o desempenho da órtese. Também devem ser utilizados novos sensores para o devido funcionamento da nova órtese Ortholeg.

9 REFERÊNCIAS

- [1] ARAÚJO M. V., SILVA G. N., SOARES L. C., ALSINA P. J., **Ortholeg Project - Development of an Active Orthosis Prototype for Lower Limbs**. Proceedings of the 2012 Brazilian Robotics Symposium and Latin American Robotics Symposium . SBR LARS 2012, Fortaleza, Brasil, October, 2012, pp. 239-244.
- [2] ARAÚJO, F. M. U., **Sistemas de Controle**. Apostila. Natal/RN, fevereiro, 2007, p. 19 . 48.
- [3] ARAUJO, M. V, ALSINA, P. J, R, V. C., MELO, N. B. **Powered orthosis ortholeg: Design and development**. Latin America Transactions IEEE (Revista IFEE America Latina), vol. 13, no. 1, pp. 90-95, 2015.
- [4] ARAÚJO, M. V. ; ALSINA, P. J. ; MEDEIROS, A. A. D. ; PEREIRA, J. P. P. ; ARAUJO, F. M. U.; DOMINGOS, E. C. ; Silva, J. S. **Development of an Active Orthosis Prototype for Lower Limbs**. In: 20th International Congress of Mechanical Engineering - COBEM 2009, 2009, Gramado. Proceedings of the 20th International Congress of Mechanical Engineering - COBEM 2009, 2009. p. 1-6.
- [5] ASTROM, K. J. e HAGGLUND, **Revisiting the Ziegler-Nichols step response method for PID control**. Journal of process Control, 14, 635-650 (2004).
ASTROM, K. J. e HANGGLUND, T. **PID Controllers: Theory, Design, and Tuning**. 2ª ed. Research Triangle Park: Instrument Society of America (1995).
CHEN, D. e SEBORG, D. E. **PI/PID Controller Design Based on Direct Synthesis and Disturbance Rejection**. Ind. Eng. Cham. Res. 2002.
- [6] COSTA, F. P, ARAUJO, M.V, MORAES, V. G. S, ALSINA, P. J, **Medição e análise do consumo energético da órtese Ortholeg nos movimentos de levantar e sentar**. XII Simpósio Brasileiro de Automação Inteligente (SBAI), Brasil, outubro, 2015.

- [7] **DAGU HI-TECH ELECTRONIC ROBOTICS** (Zhongshan City Guangdong Province China) (Org.). TRex Controller Robot. 2016. Disponível em: <<http://www.dagurobot.com/index.php>>. Acesso em: 5 ago. 2016.
- [8] GLOGER M., ALSINA P. J., MELO N. B., **"Ortholeg 2.0 É A new Design for a Lower Limb Active Orthosis"**. 26th 2015 International Symposium on Micro-NanoMechatronics and Human Science, IEEE, Nagoya, Japan, (2015) pp.,1-6.
JOHNSON, C. D. **"Process Control Instrumentation Technology"**, 5ª ed. Upper Saddle River: Prentice-Hall (1997).
- [9] MELO N. B., ALSINA P. J., DÓREA C. E. T., ARAÚJO M. V., **"Gait Cycle Modeling for an Active Orthosis Using Principal Component Analysis"**, Proceedings of the 2013 IEEE Latin American Robotics Symposium . LARS 2013, Arequipa, Peru, 2013, p. 118-123.
- [10] MORI, Y., J. OKADA, K. TAKAYAMA (2005), **"Development of straight style transfer equipment for lower limbs disabled íableí"**, em **"Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics"** pp. 1176. 1181.
- [11] NICOLESCU, M.N.; MATARIC, M.J.; **"Linking perception and action in a control architecture for human-robot domains"**. Proceedings of the 36th Annual Hawaii International Conference on System Sciences. 2003.
- [12] SCHILDT A. N., SANCA A. S., GUIMARÃES J. P. F., Deus M. S., ALSINA P. J., **"Hardware and Telemetry Architectures for an Unmanned Aerial Vehicle of Quadrotor Type"**. Proceedings of the 5th Workshop in Applied Robotics and Automation . ROBOCONTROL 2012. Bauru, Brazil, junho, 2012.
- [13] UNBENHAUEN, L. (1996). **"Distributed Supervisory Industrial Control Systems and their Problems for Embedding Advanced Control Strategies"**, 7 o Congreso Latinoamericano de Control Automático e XV Simposio Nacional de Control Automático, vol. 1, Buenos Aires, Argentina, pp. I-VIII.