Federação das Indústrias do Estado da Bahia
CENTRO UNIVERSITÁRIO SENAI CIMATEC

Engenharia Mecânica

Trabalho de Conclusão do Curso

Desenvolvimento do robô de inspeção.

Apresentada por: Bruno Rodrigues
Bruno de Sousa
Frederico Garcia
Leandro S O Nozela
Victor V. Rezende

Orientador: Prof. Marco Reis, M.Eng.
Co-orientador: João Lucas da Hora

Dezembro de 2019

Bruno Rodrigues
Bruno de Sousa
Frederico Garcia
Leandro S O Nozela
Victor V. Rezende

Desenvolvimento do robô de inspeção.

Trabalho de Conclusão do Curso apresentada ao , Curso de Engenharia Mecânica do Centro Universitário SENAI CIMATEC, como requisito parcial para a obtenção do título de **Bacharel em Engenharia**.

Área de conhecimento: Interdisciplinar

Orientador: Prof. Marco Reis, M.Eng.

Salvador
Centro Universitário SENAI CIMATEC
2016

Agradecimentos

Resumo

Palavras-chave: Robô de Inspeção, Linhas de Transmissão, Navegação, Cinemática Inversa, Manipuladores

Abstract

Keywords: Inspection Robot,Transmission Lines,Navigation,Inverse Kinematics,Manipulators

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Siglas

ELIR	*Electrical Inspection Robot*
URDF	*Universal Robot Description Format*
ROS	*Robotic Operation System*
QFD	*Quality Function Deployment*
SOTA	*State of the Art*
USB	*Universal Serial Bus*

Lista de Simbolos

∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble
∂ .... Bla bla bla
$\prod$ .... ble ble ble

Capítulo Um

Introdução

"Faça ou não faça, tentativa não há."

(Mestre Yoda)

(**??**) O Brasil apresenta uma matriz energética diferente da do resto do mundo, onde as fontes renováveis representam uma grande parte da geração da energia. Segundo a (**??**), em 2016, a matriz energética mundial contava com somente 14,1% da matriz energética constituída por fontes renováveis, enquanto o brasil já apresentava 82% da sua matriz vinda de fontes renováveis, onde a geração hidrelétrica corresponde a 70% dessa geração.

A expectativa é de que a energia hidrelétrica continue sendo cada vez mais utilizada no país, devido ao crescimento previsto da demanda energética brasileira, onde segundo o (**??**) o consumo atual é de 405 TWh e a demanda esperada em 2030 é de 950 e 1.250 TWh/ano (**??**). Mesmo com a grande participação da geração hidrelétrica , somente 23% dos 260 GW totais de potencial hidrelétrico são aproveitados (**??**).

A concentração de demanda energética no Brasil está concentrada principalmente na região Sudeste devido a densidade populacional e elevada industrialização, isso faz com que dois terços do total da capacidade instalada estejam localizadas na Bacia do Rio Paraná que é a bacia mais próxima da região, enquanto as bacias com potencial menos aproveitado são as localizadas no norte e nordeste do país (**??**).

Com desenvolvimento do país é esperado um aumento na demanda de energia elétrica e consequentemente um aumento na geração de energia hidrelétrica, isso faz com que seja esperado um crescimento considerável na quantidade das linhas de transmissão, de acordo com (**??**), em setembro de 2018 o sistema elétrico brasileiro já atingiu 144.828 km de linhas de transmissão. Esse aumento na quantidade de linhas tende a ser amplificado pela tendência à exploração da geração de energia na região Norte, assim sendo necessário a construção de novas linhas para distribuir essa energia para as outras partes do País.

Quanto mais linhas de transmissão e maiores distâncias entre os centros geradores, maiores tendem a ser as perdas. Isso faz com que seja necessária um controle da qualidade dessa transmissão, o que se dá por meio de inspeções. A estrutura já existente apresenta precariedade em alguns aspectos, onde segundo (**??**) "no Brasil, há uma quantidade

considerável de linhas de transmissão que já ultrapassou os 40 anos de idade. Com o envelhecimento das linhas de transmissão, a manutenção preventiva é um fator de extrema relevância para garantir o perfeito funcionamento dos sistemas." A necessidade da constante manutenção e a alta periculosidade que os operadores são expostos faz com novas alternativas e tecnologias sejam aplicadas para a manutenção, o uso de Drones pilotados remotamente, com câmeras e sensores já é uma realidade em alguns países do mundo. O desenvolvimento de um robô próprio para inspeção de linha configura uma dessas novas alternativas, e possibilita uma expansão dos horizontes para as tecnologias aplicadas.

1.1 Objetivos

O objetivo do trabalho é implementar o sistema de movimentação do robô ELIR (*Electrical Line Inspection Robot*). Onde esse sistema é complementar aos outros existentes no robô, onde o conjunto dessas soluções busca fundamentar a implementação de uma Inspeção autônoma da linha.

1.1.1 Objetivos Específicos

Para o desenvolvimento do sistema é necessário realizar o estudo da movimentação, gestão de energia, controle e elaboração de trajetória para sistemas robóticos. A operação na linha faz com que seja necessário a gestão de energia do robô, assim como a integração com os outros subsistemas já desenvolvidos. De forma a garantir a operação na linha, os dispositivos e ferramentas utilizadas devem estar integradas no ROS (*Robot Operating System*), onde é necessário também a integração com outros pacotes já desenvolvidos para o Robô.

1.2 Justificativa

Tendo em vista a crescente demanda de energia elétrica do país bem como a previsão , a necessidade de um processo confiável de transmissão se torna amplamente necessário, afinal, diversas unidades consumidoras são alimentadas diariamente, além de instalações que exercem atividades críticas, como hospitais. As unidades geradoras de energia elétrica se encontram em regiões distantes de seus consumidores finais, portanto se faz necessário a utilização de linhas de transmissão de energia elétrica.

Uma linha de transmissão é uma linha composta por cabos condutores de energia

elétrica, utilizada para a transmissão de energia em alta tensão, saindo da origem geradora e indo até às cargas consumidoras.

A garantia da distribuição em condições favoráveis se dá pela confiabilidade das linhas de transmissão e os procedimentos de manutenção aplicados à elas, para isso, é realizada constantemente a rotina de inspeção nas linhas. A rotina de inspeção, se dá através da análise da integridade da estrutura das torres, a condição em que se encontram os isoladoras e as conexões das linhas de transmissão, uma vez que o tempo e a exposição a umidade e ao sol, além de diversos eventos climáticos, fazem com diversas falhas referentes ao desgaste do material venham a aparecer.

Estas análises têm como principal objetivo a detecção de eventuais pontos de ruptura. Outro meio para a localização dos eventuais pontos de ruptura se dá pelo uso de câmeras térmicas, onde existe o aumento da temperatura pontual devido à elevação na resistência elétrica.

Figura 1.1: Instalação típica de uma linha de transmissão

Fonte: (**??**)

Segundo (**??**), as rotinas de inspeção de linhas de transmissão se dão principalmente por dois métodos: inspeções por aeronaves e inspeções terrestres. A inspeção realizada por aeronaves, se dá tipicamente com o uso de helicópteros, que executam voos em baixa altitude, extremamente próximos das linhas de transmissão.

Em alguns casos as condições climáticas podem dificultar o procedimento de inspeção e controle da aeronave, além do risco inerente da atividade para os tripulantes, principalmente devido ao fato de que as aeronaves tipicamente operam na região de "homem-morto", uma zona de altura que representa perigo para os operadores a bordo das aeronaves em caso de uma queda.

A inspeção por vias terrestres possui uma grande dificuldade devido à dependência

Figura 1.2: Inspeção em linhas de transmissão por veículos aéreos tripulados.

Fonte: (**??**)

do terreno do local, o qual pode ser de difícil acesso devido às características geográficas.Diversos fatores tornam a inspeção de linhas de transmissão um procedimento não só perigoso, mas também altamente custoso.

Segundo (**??**) as principais desvantagens dos meios convencionais de inspeção de linhas de transmissão são os riscos de acidentes, devido a periculosidade do procedimento de inspeção; o alto custo, uma vez que é necessário a locação e deslocamento de equipamentos específicos para o transporte e inspeção das linhas de transmissão; a alta dependência das condições climáticas e geográficas, uma vez que se torna muito difícil realizar rotinas de inspeção em tempos chuvosos ou em locais de difícil acesso.

Outra grande desvantagem dos procedimentos de inspeção definida por (**??**) é justamente a necessidade de uma mão de obra qualificada para realização destes procedimentos. Se estes procedimentos de inspeção de linha de transmissão fossem realizados em linhas desenergizadas, o processo seria bem mais simples e rápido, porém existem diversos problemas atrelados ao fato de que existem inúmeras cargas consumidoras que necessitam da energia elétrica gerada.

1.3 Organização do Trabalho de Conclusão do Curso

O documento está organizado em cinco capítulos, seguindo a seguinte estrutura:

Capitulo 1 - Introdução: Faz a contextualização do âmbito no qual a pesquisa proposta está inserida. Apresenta, portanto, a problemática, objetivos e como este projeto Theoprax de conclusão de curso está estruturado

Capítulo 2 - Referencial Teórico: Apresenta a base teórica necessária para o desenvolvimento do projeto.

Capítulo 3 - Metodologia: Define o método adotado para o desenvolvimento do projeto, explicitando seu fluxo de atividades e premissas necessárias para aplicar a

metodologia.

Capítulo 4 - Desenvolvimento: Exibe os procedimentos realizados e resultados obtidos através de testes, unitários e integrados, durante o desenvolvimento do projeto.

Capítulo 5 - Conclusão: Apresenta as conclusões, contribuições e algumas sugestões de atividades de pesquisa a serem desenvolvidas futuramente.

Capítulo Dois

Fundamentação Teória

"Elementar , meu caro Watson."

(Scherlock Holmes)

O termo robô vem da palavra tcheca robota que tem como uma das possíveis traduções "trabalhador forçado" e ganhou o significado atual após o escritor tcheco Karel Capek (1809 - 1938), na sua obra de ficção científica "R.U.R. Rossumovi Univerzální Roboti", associar o termo às máquinas criadas pelo personagem principal para servi-lo. Mas a ideia de algo que desenvolva atividades de maneira autônoma é apresentada ao mundo muito tempo antes. (**??**) diz: "Se cada instrumento pudesse realizar sozinho a sua tarefa, obedecendo ou antecipando a nossa vontade, [...] os feitores não precisariam de servos, nem os senhores de escravos."

Diversas obras da ficção retratam diferentes tipos de robôs criados de forma a reproduzir comportamentos semelhantes aos de um ser humano. Com o passar do tempo, juntamente com o avanço tecnológico nas áreas da eletrônica, mecânica e informática, a construção dessas máquinas se tornou possível. A indústria observou nos robôs, o potencial para automatizar e otimizar as linhas de processo, onde atividades que pudessem demandar mais tempo se fossem executadas por seres humanos, seriam executadas de forma muito mais rápida e precisa com a utilização de máquinas programadas e autônomas, aumentando a produção.

A (**??**) define um robô como "mecanismo programável atuado em dois ou mais eixos com um grau de autonomia, movendo-se dentro do seu ambiente, para executar tarefas pretendidas". É resultado da integração de componentes como: Sensores; atuadores; unidade de controle; unidade de potência e manipulador mecânico. Sensores são os componentes que fornecem parâmetros sobre o ambiente em que o robô se encontra e sobre o comportamento do próprio sistema robótico. Já os atuadores são os dispositivos que movimentam as partes, quando convertem energia elétrica, hidráulica ou pneumática em mecânica. A energia necessária para o funcionamento dos atuadores é fornecida pela unidade de potência.

O gerenciamento dos parâmetros necessários para que o robô realize suas tarefas é de responsabilidade da unidade de controle. De onde também são emitidos os comandos para a movimentação. O manipulador mecânico é o conjunto de componentes estruturais

do robô, elos ou links, conectados entre si por articulações comumente denominadas de juntas. Graus de liberdade, segundo (**??**) "É o número mínimo de variáveis independentes de posição que precisam ser especificadas para se definir inequivocamente a localização de todas as partes de um mecanismo".

2.1 Cinemática

A cinemática é o ramo da física que descreve o movimento de um corpo, determinando características como posição, velocidade e aceleração. Na robótica, o estudo cinemático resulta em um conjunto de equações que caracterizam o movimento do robô, a complexidade da solução varia com a quantidade de graus de liberdade que esse robô tem. Em um manipulador mecânico composto por links que são conectados por juntas, cada conjunto link-junta caracteriza um grau de liberdade. Dessa maneira, um robô com n conjuntos link-junta tem n graus de liberdade, sendo o primeiro link a base de sustentação do robô no mundo e o último, onde está a seu end-effector.

2.1.1 Cinemática Direta

A cinemática direta é a solução para a movimentação de um robô com cálculo da posição e orientação do end-effector a partir de dadas posições das juntas. A notação de Denavit-Hartenberg é uma ferramenta utilizada para coordenar a descrição cinemática de sistemas mecânicos articulados com n graus de liberdade.

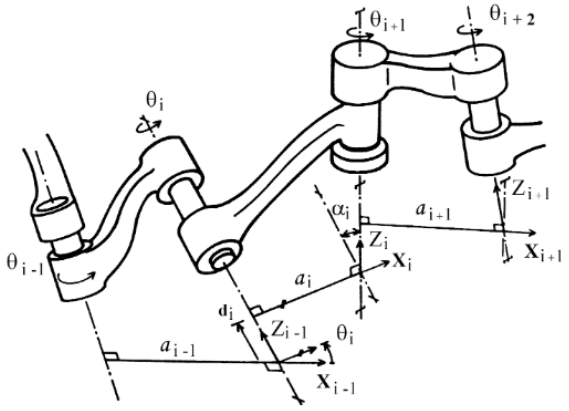


Figura 2.1: Parâmetros de Denavit-Hartenberg

Fonte: (**??**)

A figura mostra dois *links* ligados por uma junta de superfícies deslizantes uma sobre a outra. Um eixo de uma junta estabelece a conexão de dois *links*. Segundo (**??**), os eixos das juntas devem ter duas normais conectadas a eles, uma para cada um dos

links. Assim a posição relativa destes dois *links* conectados ($i-1$ e i) é dada por d_i, que é a distância medida ao longo do eixo da junta entre suas normais. O ângulo de junta θ_i entre as normais é medido em um plano normal ao eixo da junta. Dessa forma, d_i e θ_i são a distância e o ângulo entre os *links* adjacentes. Determinam a posição relativa de *links* vizinhos.

Um *link* pode apenas ser conectado a dois outros *links* ($i-1$ e $i+1$). Assim, dois eixos de juntas são estabelecidos em ambos terminais de conexão. Os *links* mantém uma configuração fixa entre as juntas e podem ser caracterizados pelos parâmetros a_i e α_i. O parâmetro a_i é a menor distância medida ao longo da normal comum entre os eixos da junta, chamado de comprimento de *twist*, já o α_i é o ângulo de *twist*. Esses quatro parâmetros determinam a estrutura do *link*, parâmetros da junta e a posição relativa aos *links* vizinhos.

A representação de Denavit-Hartenberg (**??**) tem como resultado uma matriz 4 x 4 representando cada sistema de coordenadas do *link* na junta em relação ao *link* anterior. Essa matriz é obtida através do produto das transformações: Translação de uma distância d_i ao longo do eixo Z_{i-1} para trazer os eixos X_{i-1} e X_i na coincidência; Rotação no eixo Z_{i-1} de um ângulo θ_i para alinhar os eixos X_{i-1} e X_i; Translação ao longo do eixo X_i de uma distância a_i para trazer as duas origens na coincidência; Rotação do eixo X_i um ângulo α_i para trazer os dois sistemas de coordenadas na coincidência. Isso resulta na matriz de transformação homogênea ${}^{i-1}A_i$.

$$ {}^{i-1}A_i = T_{z,d}T_{z,\theta}T_{x,a}T_{z,\alpha} \tag{2.1} $$

$$ {}^{i-1}A_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & 0 \\ sin\theta_i & cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha_i & -sin\alpha_i & 0 \\ 0 & sin\alpha_i & cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2} $$

$$ {}^{i-1}A_i = \begin{bmatrix} cos\theta_i & -cos\alpha_i sin\alpha_i & sen\alpha_i sin\theta_i & a_i cos\theta_i \\ sin\theta_i & cos\alpha_i cos\theta_i & -sin\alpha_i cos\theta_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3} $$

2.1.2 Cinemática Inversa

Segundo (**??**) os robôs estão em um espaço onde o objeto a ser manipulado tem sua posição expressa no sistema de coordenadas do ambiente. Com o objetivo de controlar a posição e orientação do *end-effector* do robô, a solução da cinemática inversa é mais

adequada. A cinemática inversa consiste em, partindo de uma posição e orientação desejada, calcula-se as posições das juntas para que o robô alcance esse objetivo, é o processo inverso da cinemática direta.

Há de se observar que a cinemática inversa pode ou não ter solução, caso a posição de interesse esteja fora do espaço de trabalho do robô, não há posições de juntas que execute a tarefa. Nos momentos em que a posição desejada pode ser alcançada, podem existir mais de uma solução. Um ponto importante na solução da cinemática inversa é, quando há mais de uma solução deve-se atentar para qual delas é a melhor opção, levando em consideração o ambiente em que o robô se encontra, principalmente os obstáculos à sua volta. A demanda energética para a execução dos possíveis movimentos e o esforço qual as juntas serão submetidas nesta ação, é crucial para o planejamento da movimentação do robô.

2.2 Modelagem Cinemática de um Braço Planar

O robô ELIR tem na sua estrutura, braços que se movimentam apenas em dois eixos, x e z, através da atuação de duas juntas, podendo assim ser modelado cinematicamente como um braço planar do tipo RR. A figura a seguir mostra um exemplo desse braço, RR por ter duas juntas rotativas, que se movimenta no plano $x - y$:

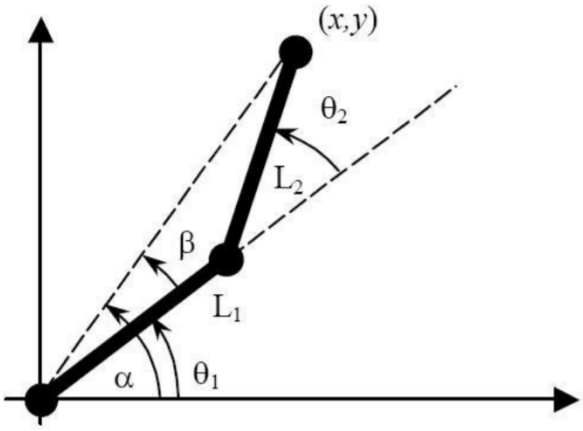


Figura 2.2: Braço planar do tipo RR

Fonte: (**??**)

Usando a análise da cinemática direta, consegue-se determinar a posição do *end-effector* com base nos ângulos θ_1 e θ_2 e nas dimensões L_1 e L_2. Logo tem-se que:

$$x = L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2) \tag{2.4}$$

$$y = L_1 sen(\theta_1) + L_2 sen(\theta_1 + \theta_2) \tag{2.5}$$

Aplicando a lei dos cossenos ao triângulo formado pelo braço e pela linha entre a origem do braço e o seu *end-effector* obtém-se:

$$\theta_2 = \pm arccos\frac{(x^2 + y^2 - (L_1)^2 - (L_2)^2)}{2L_1L_2} \tag{2.6}$$

Para determinar o θ_1 considera-se a relação trigonométrica:

$$tan(A - B) = \frac{tan(A) - tan(B)}{1 + tan(A)tan(B)} \tag{2.7}$$

e tomando:

$$tan(\beta) = \frac{L_2 sin\theta_2}{L_1 + L_2 cos\theta_2} \tag{2.8}$$

tem-se que:

$$\theta_1 = arctan[\frac{y(L_1 + L_2 cos\theta_2) - xL_2 sin\theta_2}{x(L_1 + L_2 cos\theta_2) - yL_2 sin\theta_2}] \tag{2.9}$$

Assim é possível fazer a solução da cinemática inversa para um braço planar RR.

2.3 Desenvolvimento de Robôs

Para o desenvolvimento de sistemas robóticos, é necessária a integração de vários dispositivos, assim sendo necessário utilizar ferramentas e tecnologias que poupem tempo no desenvolvimento, de forma a facilitar o processo de comunicação entre as diversas camadas de abstração. As camadas de abstração se referenciam ao alto e baixo nível da máquina, onde baixo nível é uma referência para aplicações mais simples, que estão mais próximas da linguagem da máquina, como por exemplo aplicação de comunicação somente via *bytes*. Um exemplo de um elemento que está numa camada de abstração de alto nível é uma Interface Homem-Máquina , onde o usuário consegue interagir com a máquina diretamente, sem ter que necessariamente entender o seu funcionamento interno.

2.3.1 Framework

Em ambientes computacionais, a utilização de ferramentas para realização de atividades e desenvolvimento de soluções é de extrema importância. Estas ferramentas podem ser softwares específicos para execução de uma determinada atividade ou *frameworks*. Segundo (**??**) "*Frameworks* são estruturas de classes que constituem implementações incompletas que, estendidas, permitem produzir diferentes artefatos de software". Os *frameworks* em geral permitem o desenvolvimento de soluções computacionais baseadas em determinadas funcionalidades, seguindo uma estrutura definida pelo *framework*. De acordo com (**??**) os *frameworks* definem uma arquitetura para um conjunto de subsiste-

mas, dando os construtores necessários para a sua criação.

A principal característica de um *framework* é a sua capacidade de reutilização, afinal a sua utilização permite que diversos conjuntos de produtos possam ser gerados partindo de uma única estrutura que possua os conceitos mais gerais. Segundo (**??**) *frameworks* podem ser classificados em dois tipos principais: *Frameworks* de Aplicações Orientado a Objetos e *Frameworks* de Componente. Os *frameworks* orientados a objetos geram famílias de aplicações orientadas a objetos e seus pontos de extensão são definidos como classes abstratas ou interfaces, onde se estendem por cada instância da família de aplicações. Para *frameworks* de componentes, o suporte é previsto para componentes que sigam um determinado modelo, possibilitando que as instâncias destes componentes sejam acopladas ao *framework*. Também são estabelecidas as condições necessárias para que um componente seja executado, regulando a sua interação entre as instâncias de outros componentes.

Os *frameworks* utilizados para robótica, são extremamente importantes, pois o uso de suas ferramentas possibilita o desenvolvimento e criação das soluções computacionais e códigos necessários para cada funcionalidade de um robô, de forma que o funcionamento delas em conjunto seja otimizado pela natureza do *framework* de realizar a compatibilização entre as estruturas.

2.3.2 Simulação

Em sistemas complexos, onde diversas variáveis definem o seu funcionamento, e por consequência as suas respostas a determinados estímulos, torna-se extremamente difícil e irresponsável executar a sua fabricação antes de realizar uma validação prévia de seu funcionamento.

Este tipo de procedimento de análise prévia do comportamento de um sistema é chamado de simulação. De acordo com (**??**) "Simulação refere-se a uma ampla coleção de métodos e aplicações para imitar o comportamento do sistema real, por meio de um computador com um *software* apropriado". Diversos tipos de sistemas se utilizam da ferramenta de simulação para validar previamente o funcionamento de projetos.

O processo de utilização de uma simulação consiste em basicamente recriar o sistema em questão em um ambiente computacional e então são fornecidas as entradas para o sistema, as rotinas de tratamento destas entradas e por fim as saídas da simulação.

Em sistemas robóticos, uma ferramenta extremamente útil e bastante utilizada, é a simulação. Segundo (**??**)

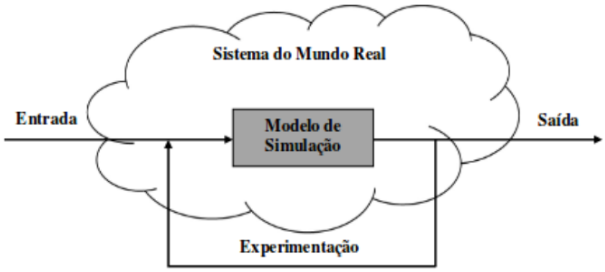


Figura 2.3: Diagrama de funcionamento de um processo de simulação

Fonte: (**??**)

> "Quando se trabalha com robótica, o uso de uma simulação é de importância significante. Por um lado, ela permite a validação de diferentes alternativas durante o design do sistema robótico, levando assim, a melhores decisões e preservação de custos. Por outro lado, auxilia o processo de desenvolvimento de *software*, disponibilizando uma reposição para robôs que não estejam em mãos".

Através do uso de *softwares* de simulação é possível criar uma representação computacional não só o modelo físico de um robô, mas também os parâmetros referentes à objetos do ambiente no qual o mesmo será posto em funcionamento. A avaliação prévia da execução das tarefas e do funcionamento do robô, permite a observação do comportamento do sistema em determinadas situações, facilitando assim, a tomada de decisões mais efetivas no processo de desenvolvimento do protótipo real.

2.3.3 Odometria

A odometria consiste no cálculo para estimar a mudança de posição do robô no tempo, onde isso pode se dar por meio de diversos dispositivos que possibilitem o cálculo de deslocamento. Onde segundo (**??**), "Odometria - a medição da distância - é um método fundamental usado por robôs para navegação". A medição de tempo é fácil utilizando o *clock* interno do computador embutido. Medir velocidade é mais difícil: em alguns robôs educacionais utilizam codificadores são usados para contar as rotações da rodas, enquanto em outros a velocidade é estimada das propriedades dos motores.

No caso da análise de deslocamento do robô na linha por meio de roldanas, o movimento é caracterizado como linear, já que o deslocamento ocorre em somente uma direção, analogamente a odometria utilizada é a linear, onde o deslocamento pode ser calculado simplesmente pela equação 2.10 onde s representa o espaço caminhando, v a velocidade e t o tempo.

$$s = v * t \tag{2.10}$$

Utilizando o medidor de tempo interno do computador embutido nos sistemas robóticos, pode se calcular a variação de espaço para um tempo muito pequeno, onde esses pequenos incrementos são somados ou subtraídos para encontrar o deslocamento do robô.

A velocidade de deslocamento das roldanas pode ser encontrada utilizando a equação 2.11 com as informações do raio da roldana r e a sua velocidade de giro w em radianos por segundo. A informação da velocidade de giro da roldana geralmente é extraída dos servomotores utilizados para tração.

$$v = 2\pi * r * w \tag{2.11}$$

O cálculo da odometria por meio da velocidade das rodas é denominado no âmbito da robótica como odometria de roda, *wheel odometry* em inglês, porém, outras técnicas são utilizadas, já que existem diversos tipos de deslocamento diferentes. Outro tipo aplicação muito encontrada é a odometria visual, que segundo (**??**) "Odometria Visual (OV) é o processo de estimação do deslocamento de um agente (ex: veículo, humano e robô) utilizando a entrada de uma ou múltiplas câmeras conectadas a ele". Os domínios da aplicação incluem robótica, realidade aumentada, automotiva e 'computadores vestíveis'.

2.3.4 Gestão de Energia

O conceito de gestão de energia se dá pela forma como a energia elétrica é utilizada em um sistema composto de diversos dispositivos elétricos e eletrônicos. Para sistemas robóticos, este conceito representa um fator importante para garantir uma operação autônoma de qualidade. Os robôs quando nesse tipo de operação, geralmente não dispõem de uma fonte de energia constante, e portanto, são geralmente alimentados por baterias e tendo interação por meio de conexões sem fio.

O uso de diversos dispositivos eletrônicos de baixo consumo energético, como sensores e interfaces microcontroladas, podem não se mostrar um problema para um curto período de operação, porém, para maximizar o tempo da atividade exercida pelo robô, é necessário encontrar uma forma eficiente de gerir a operação dos dispositivos conectados na rede de alimentação. Segundo (**??**)

"Gestão de energia é um conceito importante em redes de sensores, porque

> uma estrutura de energia cabeada geralmente não está disponível e um conceito óbvio é utilizar a energia disponível da bateria de forma eficiente".

Quanto mais atividades diferentes o robô desempenha maior será a demanda de energia entre os dispositivos interconectados, isso faz com que seja necessário que os desenvolvedores busquem uma forma de otimizar o custo de energia individual das atividades e do fluxo de operação como um todo. Os diversos dispositivos utilizados em sistemas robóticos fazem com que o mesmo se utilize de diferentes níveis de tensão e corrente, já que comumente, os dispositivos utilizados são comerciais, e devido às diferenças das suas características e parâmetros, definidos por empresas diferentes, responsáveis pela produção e fabricação das ferramentas, é necessário que a gestão de energia leve em consideração a compatibilidade entre diferentes dispositivos.

2.3.5 Conceito de segurança e Integridade

Em diversas áreas, é comum a verificação das condições antes da execução de atividades, a aviação é um grande exemplo de uso desse conceito. Neste seguimento, o *checklist* é utilizado toda vez antes de um avião decolar, assim é possível verificar se os sistemas vitais para o vôo estão em ordem. O principal objetivo dessa ação é identificar os riscos que existem para o cumprimento da atividade.

Para que um dispositivo robótico execute as tarefas para as quais ele foi desenvolvido, deve-se verificar se os seus sistemas, como um todo, e os componentes individualmente, estão em condições de funcionamento, garantindo assim a integridade do sistema como um todo. Essa análise deve ser feita levando em conta a importância de cada sistema e de cada componente desses sistemas, a fim de aumentar a capacidade de operação em condições adversas do robô. Podem existir sistemas que, mesmo quando não estão operando adequadamente, não comprometem a execução da missão do robô.

2.3.6 Comunicação em sistemas robóticos

Dispositivos eletrônicos são capazes de realizar transmissão de dados, afinal, a interconexão entre eles é de extrema importância em sistemas em que existam diversos dispositivos responsáveis por funções distintas. Dispositivos que se comunicam entre si, são capazes de criar uma rede em todo o sistema, permitindo um aumento na confiabilidade das funções do sistema, através da troca de informações de parâmetros que venham a ser importantes para o funcionamento do sistema como um todo.

Para que os dispositivos possam se comunicar entre si, os mesmos adotam o que se chama de protocolos de comunicação. Protocolos de comunicação são arquiteturas que estabelecem a troca de dados entre dispositivos eletrônicos. Os dispositivos comerciais possuem diferentes tipos de protocolos de comunicação e por isso, torna-se extremamente importante se atentar a qual protocolo utilizar durante a conceituação de um projeto que se tenha a necessidade da interconexão de dispositivos.

Uma das formas mais comuns de se realizar a transmissão de dados entre dispositivos embarcados é a comunicação serial. (**??**) define a comunicação serial como um envio de *bits* de forma serial, similar a uma fila. Possuindo dois canais principais: o canal TX para envio e o canal RX para recebimento. Dentro desse processo de comunicação alguns parâmetros devem ser levados em conta, como a taxa de transmissão de dados (*BaudRate*); bits de paridade, para assegurar que o número de bits no campo de dados é par ou ímpar; bits de parada para indicar o início ou fim de uma comunicação.

Outro meio de comunicação muito utilizado é o USB (*Universal Serial Bus*), criado com a intenção de tornar a comunicação serial mais simplificada e com uma taxa de transmissão muito mais elevada. Os cabos conectores USB possuem geralmente quatro fios condutores, sendo dois deles para alimentação e dois outros cabos de dados. Os cabos de dados são nomeados como D+ e D-, onde a comunicação entre os dispositivos se dá pela variação de tensão entre estes dois sinais. Dentro de ampla complexidade como um robô, onde diversos dispositivos necessitam estar trocando informações, a utilização de protocolos de comunicação serial se tornam extremamente importantes para a garantia da confiabilidade na execução de tarefas e operações.

Capítulo Três

Metodologia

"Tudo o que temos de decidir é o que fazer com o tempo que nos é dado."

(Gandalf)

De acordo com (**??**) metodologia é "o conjunto de métodos e técnicas aplicadas para um determinado fim. É o caminho percorrido, a maneira utilizada para atingir um objetivo". Por certo, descreve os métodos que padronizam uma produção, visando a chegada em um resultado. Em trabalhos acadêmicos a sua importância vai além de descrever o processo de confecção do projeto mas também permite que o mesmo possa ser replicado por outros pesquisadores.

A metodologia aplicada para para o projeto toma como base o desenvolvimento de sistemas robóticos, nesse caso sendo voltada para o desenvolvimento de um sistema de movimentação robótico, presente no robô *ELIR*. A divisão do projeto em fases maiores e menores facilita o fluxo para o desenvolvimento, assim sendo definidas quatro partes maiores, sendo elas: conceituação, *design*, desenvolvimento e operacionalização. O fluxo do projeto está explicitado no Figura 3.1 a seguir:

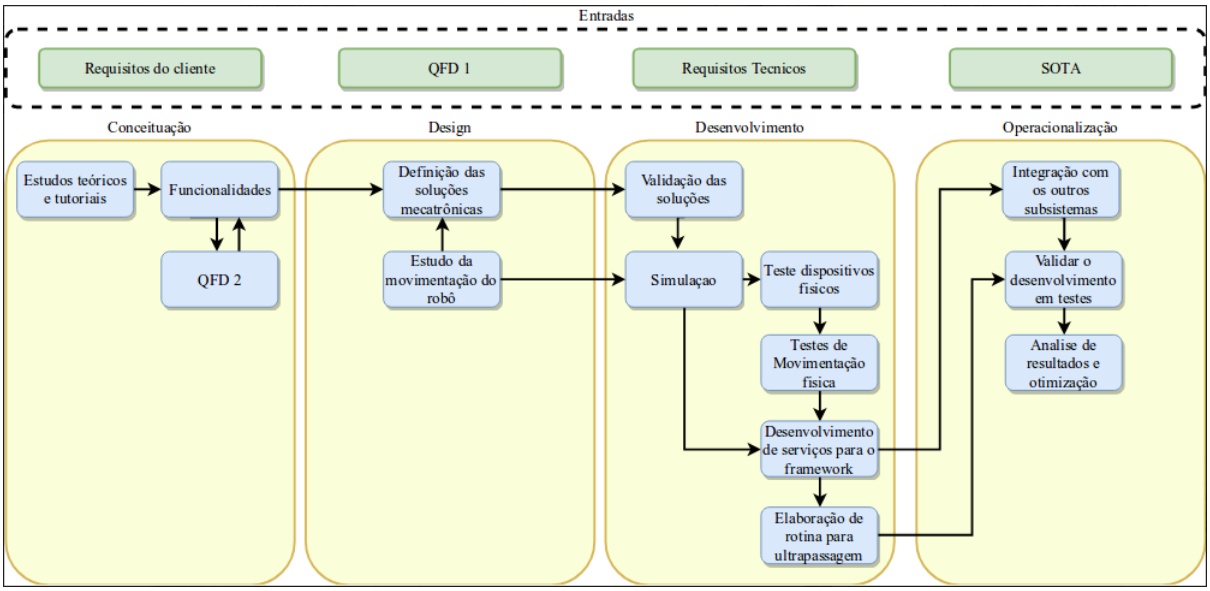


Figura 3.1: Fluxograma de desenvolvimento

Devido a complexidade envolvida nesse tipo de sistema, são utilizadas ferramentas específicas de desenvolvimento, assim como o cliente fornece certos parâmetros iniciais para impulsionar o projeto e guiar desenvolvimento para um resultado satisfatório. São

consideradas entradas para a metodologia os requisitos do cliente, requisitos técnicos, um *QFD* inicial, denominado *QFD* 1 e o Estudo do Estado da Arte (*SOTA*).

O requisito consiste na definição documentada de uma propriedade ou comportamento que um produto ou serviço particular deve atender. Existem os requisitos do cliente, no qual são as necessidades e as expectativas do cliente, e os requisitos técnicos possui uma visão técnica para atender as necessidades do cliente e os objetivos do projeto.

Durante a fase inicial do projeto, foi conversado com o cliente e deixado explícito seus requisitos para o *ELIR*, sendo eles:

- Realizar as funções de forma autônoma;
- Transpor obstáculos e cadeia de isoladores;
- Deslocar-se através do consumo de baterias;
- Deslocamento/movimento realizada por servomotores.

Foi determinado pelo cliente para que o projeto desempenhe corretamente os seguintes requisitos técnicos:

- Desempenho de deslocamento de 15km por dia
- Velocidade de deslocamento médio sem obstáculos de 0.5 m/s
- Ultrapassagem de obstáculos de volume máximo de 410x330x150mm
- Autonomia de potência de 2 horas
- Sistema operacional Linux,
- *Backend* em C++ e Python
- *Framework* ROS Kinetic Kame.

A ferramenta de Desdobramento da Função Qualidade (*QFD*) torna-se importante para guiar o projeto e a incorporar as reais necessidades do cliente. Por meio de um conjunto de matrizes parte-se dos requisitos expostos pelos clientes e realiza-se um processo de "desdobramento"transformando-os em especificações técnicas do produto. Esse desdobramento entre os requisitos do cliente, influencia no desenvolvimento, permitindo encontrar o que impacta mais no resultado final do projeto, assim fazendo com que a prioridade de certas atividades mude.

O estudo do estado da arte é o mapeamento que possibilitará o conhecimento e/ou reconhecimento de estudos que estão sendo, ou já foram realizados com temáticas, ou linhas de pesquisa , iguais ou parecidas a que está sendo estudando. No caso de projetos desenvolvidos em conjunto que incorporam diversas teses acadêmicas, esse estudo facilita a continuação do trabalho e dá uma base o projeto.

3.1 Conceituação

A fase da conceituação consiste na criação de um conceito para o sistema, sendo assim recolhido todo o embasamento teórico necessário para a confecção do projeto. Assim fazendo com que seja elaborada uma ideia para o sistema, o que é a base para todo o projeto, guiando as próximas fases.

As entradas do cliente são de suma importância para essa etapa, onde as mesmas são a base para a ideia do sistema. Com o estudo teórico do que será necessário e o uso das ferramentas como o *QFD*, é possível elaborar as funcionalidades que serão desempenhadas pelo sistema. A elaboração de um segundo *QFD* por parte da equipe acontece em paralelo com a elaboração das funcionalidades e se utiliza do *QFD* 1 junto com os requisitos técnicos e do cliente, buscando conceituar o sistema de forma concreta.

As funcionalidades recebem entradas e saídas, sendo assim, interligadas, esse tipo de metodologia se mostra muito eficiente pois consegue dividir o robô em subsistemas e funções a serem desempenhadas, podendo assim dar uma idéia de como será o seu funcionamento e troca de informações internas. Com a definição das funcionalidades do sistema, é possível partir para a forma da idéia, como ela será aplicada, o que acontece na etapa de *Design*;

3.2 Design

Com um conceito firme para o sistema, a etapa de *design* consiste na forma que a idéia irá ter, para que a mesma seja possível. Com as funcionalidades definidas, é possível decidir quais ferramentas devem ser utilizadas no projeto, de forma a garantir que as mesmas sejam executadas.

Por se tratar do desenvolvimento de um sistema de movimentação, é importante que seja realizado um estudo da forma como será necessário se realizar o movimento do robô, já que isso impacta profundamente na seleção das ferramentas, esse estudo consiste na busca do entendimento de como será sua aplicação real, e está relacionado também

com a simulação, que colabora para o sucesso dessa etapa. São tomados como critérios para a escolha da ferramentas: A quantidade de informação sobre cada ferramenta que há disponível, suporte da comunidade que a utiliza e os tutoriais que cada ferramenta possui. O estudo da movimentação é de grande valia no *design* das soluções mecatrônicas pois, conhecendo as maneiras quais o robô tem que se movimentar, definir os recursos necessários para esse objetivo se torna mais fácil.

3.3 Desenvolvimento

O desenvolvimento consiste na aplicação prática da idéia, sendo a parte que demanda mais tempo e a partir dela, já é possível ter a noção de como será o dispositivo físico final. Contém toda a produção de software, estruturas necessárias para o projeto, e também a construção do protótipo. Com a definição das ferramentas realizada na etapa de *design*, é possível começar a aplicação no sistema de interesse, validando o que foi decidido anteriormente.

3.3.1 Validação das ferramentas

A etapa de validação das ferramentas é onde se realiza os estudos e testes para compreender o funcionamento das mesmas e verificar se suas mecânicas e funcionalidades são adequadas para a solução e desenvolvimento do projeto. Uma vez que a ferramenta esteja escolhida, é realizada a análise do seu funcionamento, seus aspectos gerais, configurações, e como integrá-las ao desenvolvimento do projeto.

Após realizado o estudo da ferramenta, e de como suas mecânicas funcionam e podem auxiliar no desenvolvimento das funcionalidades do projeto, são realizados os primeiros usos das ferramentas, por meio de pequenos testes específicos, a fim de buscar o entendimento amplo de como se operacionalizar e implementar as soluções a partir das funcionalidades das ferramentas.

Uma vez que os testes se mostram efetivos e o conhecimento sobre as suas funcionalidades esteja adquirido, a ferramenta torna-se válida, e pode ser utilizada no desenvolvimento do projeto, podendo ser utilizada em todas as etapas onde se mostre necessária

3.3.2 Movimentação simulada/em simulação

Para a validação das ferramentas, o uso de simulações computacionais é fundamental, devido que a simulação consegue prever os comportamentos do protótipo antes do mesmo estar em operação e com uso das ferramentas disponíveis na robótica e assim validando a maioria das ferramentas e estratégias. A simulação torna-se presente em todos os testes, desde a validação de ferramentas definidas durante a fase de *Design* até do robô em operação, sendo uma poderosa ferramenta para validação dos dispositivos e a movimentação física.

3.3.3 Teste com dispositivos físicos e Movimentação física

Antes de realizar testes do robô se movimentando é necessário garantir que todos os dispositivos físicos estejam funcionando corretamente. Durante essa fase deverá ser realizado a montagem do robô e garantir que esteja conforme a simulação. Logo após, é de extrema importância realizar simulações, testes de esforço, velocidades e posição dos servomotores utilizados na estrutura para que não haja nenhum imprevisto durante os próximos testes de operação.

Com os testes de dispositivos físicos realizados unitariamente, é necessário integrá-los para realizar os testes de movimentação física. Esse teste é realizado tanto na simulação quanto em operação em linha, sendo importante verificar se o mesmo está andando corretamente, observando influência externa do vento.

3.3.4 Desenvolvimento de serviços para framework e rotina para ultrapassagem

A estrutura de serviços disponibilizada pelo *framework*, são subrotinas em código para desempenhar uma função específica, a comunicação é feita por um par de mensagens, uma de solicitação e outra de resposta, que quando necessário fazer uso do serviço, uma mensagem de solicitação é enviada, logo após o serviço executa o código contigo nele e retorna uma mensagem de resposta.

Para realizar o rotina de ultrapassagem como um todo é conveniente separar em serviços as partes dos movimento de ultrapassagem, facilitando a identificação de erros e inconsistências. Com os serviços feitos e testados é preciso unificá-los para realizar a ultrapassagem. É importante organizá-los de forma sequencial e analisar como cada serviço se comporta para garantir que foram bem codificados, e se necessário realizar

ajustes na programação.

3.4 Operacionalização

A operacionalização põe em funcionamento todo o conjunto dos dispositivos, sistemas e rotinas necessárias para o funcionamento do robô. A partir do momento em que há itens, rotinas e ferramentas que são parte de uma funcionalidade, desenvolvidas, passa a ser possível a junção destes para que essa funcionalidade tome forma. Com estas finalizadas, é realizada sua integração com outras funcionalidades, para que possa receber suas entradas e entregar suas saídas.

Deve-se também verificar o desempenho das funcionalidades, a validação do desenvolvimento é realizada com testes de operação dos sistemas. Nestes testes, são executados procedimentos que repliquem as situações de operação do robô. É importante também que as condições dessa operação, ambiente, carga e etc. sejam similares as quais o robô irá encontrar.

Após a realização de testes para a validação do desenvolvimento, é possível analisar os resultados dos mesmos. As informações levantadas nos testes mostram se as funcionalidades desenvolvidas realizam o que se espera e a sua eficiência. A partir daí, pontos do projeto a serem melhorados ficam evidentes, possibilitando assim a otimização de funções.

Capítulo Quatro

Desenvolvimento

"Insanidade é continuar fazendo sempre as mesmas coisas,
esperando resultados diferentes."

(Albert Einstein)

Para obter de forma efetiva o proposto na metodologia, foram elaborados dois conjuntos principais de entregáveis: os Tutoriais, e o Kit Físico. Após finalização do projeto, os Tutoriais se encontram em domínio virtual, na Wiki do repositório da Learnbotics no Github, e o Kit Físico foi fabricado e montado. A seguir encontra-se uma melhor explicação de como esses resultados foram obtidos.

4.1 Preparação do Robô

Para que tudo o que foi pensado pelo projeto pudesse ser desenvolvido e passado ao usuário, uma escolha cuidadosa do hardware foi efetuada. Posteriormente, uma configuração e testes deste hardware foram também realizados.

4.1.1 Hardware

4.1.1.1 Testes de Hardware

Como apresentado anteriormente, o kit físico irá conter os seguintes componentes:

Tabela 4.1: Componentes constituintes do kit fisíco.

Componentes	Quantidade
Raspberry Pi3B	1
Dynamixel MX-28	2
Câmera RGB	1
Rodas emborrachadas	2
Roda boba	1
Cabos e conexões	x
Bateria	1

Muito dos componentes acima são comumente utilizados, tendo uma alta qualidade atrelada, porém, somente quando atreladas à um uso comum. Como o projeto deste trabalho envolvia o uso de uma raspberry, tornou-se necessário o teste de integração entre os componentes e a raspberry.

4.1.1.2 Sistema operacional da Raspberry

Para a utilização dos ROS e OpenCV é preferível que estes estejam instalados em um SO (Sistema Operacional) com base em LINUX e com suporte ao ROS e OpenCV. Inicialmente foi testado o Ubuntu 16.04 server. Este SO é um derivativo do Ubuntu 16.04, a única diferença é que este não compõe a parte gráfica. Teoricamente, o Ubuntu server seria o Sistema operacional perfeito para a aplicação deste trabalho, por ser um sistema sólido, amplamente testado e que tem um dos melhores suportes às ferramentas utilizadas. Com tudo, há um pequeno problema na utilização dele, que não há suporte gráfico, ou seja, o aluno de cara teria um grande estranhamento de apenas utilizar o terminal para conseguir fazer as aplicações e desafios compostos no kit.

Com isso, foi preferível instalar o SO Raspbian, uma derivação do Debian. O Raspbian é um sistema operacional otimizado e próprio para a Raspberry, tendo suporte para ROS e OpenCV. Não é possível instalar o SO Ubuntu 16.04 com gráficos pois, a Raspberry não consegue comportar ele, já que ela conta com uma memória gráfica limitada.

O que é interessante neste SO que ele é disponibilizado pela própria Raspberry e mantido por ela. Com isso, este sistema vem com diversas aplicações educacionais, e possíveis projetos que se o aluno quiser explorar, poderá encontrar infinitas finalidades.

Para instalação do SO na Raspberry, foram utilizados os seguintes materiais, mostrados na figura 4.1:

1. Raspberry PI 3B+
2. Cartão SD 32 GB
3. Imagem do Sistema Operacional

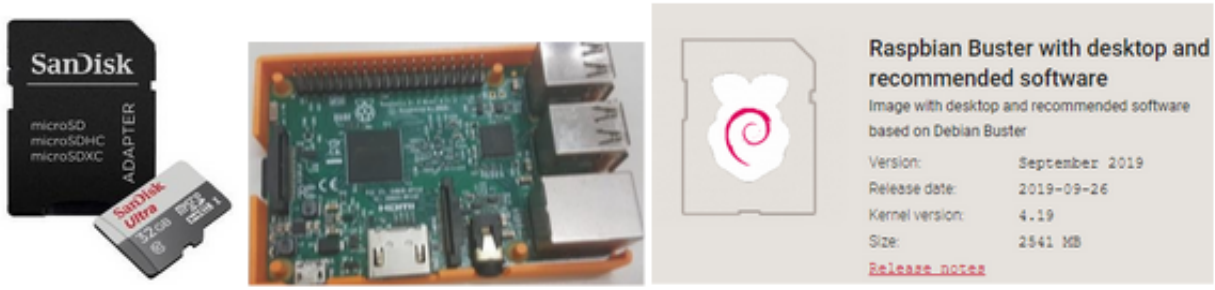


Figura 4.1: Componentes necessários para instalar o SO

A instalação do SO em si foi seguindo o tutorial disposto no próprio site da (RASBERRY, 2019), o raspberrypi.org. Lá contém tudo que é necessário para instalar, os passos a serem seguidos e como otimizar a Raspberry.

4.1.1.3 ROS na Raspberry

Para o comprimento dos requisitos funcionais do kit, faz-se necessário a instalação do framework de robótica ROS. Este passo normalmente é fácil e intuitivo, porém, quando se trata de uma ARM (Advanced RISC Machine) a instalação de frameworks como este tornam-se mais complexo. Esta dificuldade é consequência por dois fatores:

1. A arquitetura é mais simples se comparado com processadores utilizados em computadores pessoais;
2. O Sistema Operacional (SO) utilizado é o Raspbian, um SO baseado em Linux próprio para a Raspberry.

Por conta destes dois fatores, a instalação do framework não pode ser feita da mesma forma que é em um computador normal. Felizmente, há diversos tutoriais disponíveis na Internet para o auxílio desta tarefa, porém, isso não fez diminuir o nível esforço para o comprimento dela.

Tendo em vista esta complexidade, a equipe repensou como iria entregar o kit, mudando assim o requisito que o aluno deveria instalar o ROS na Raspberry. O que é interessante analisar é que o intuito deste trabalho é apresentar de forma fácil e prática o mundo da robótica aplicada para os alunos, então, para que não houvessem desistências prematuras do curso, foi preferido entregar o ROS já incluso.

4.1.1.4 Instalação do ROS na Raspberry

Inicialmente, foi utilizado o SO Ubuntu 16.04 server. Neste sistema a instalação do framework ROS foi simples, já que este sistema é amplamente utilizado pela comunidade, sendo assim, tem uma maior suporte.

Para ele, foi feito uma conexão via SSH, utilizando as entradas TX-RX da raspberry. Este tipo de conexão facilita a instalação, já que, pela raspberry somente havia o terminal, já que não havia a parte gráfica. A imagem 4.2 abaixo, mostra os componentes utilizados para a instalação.

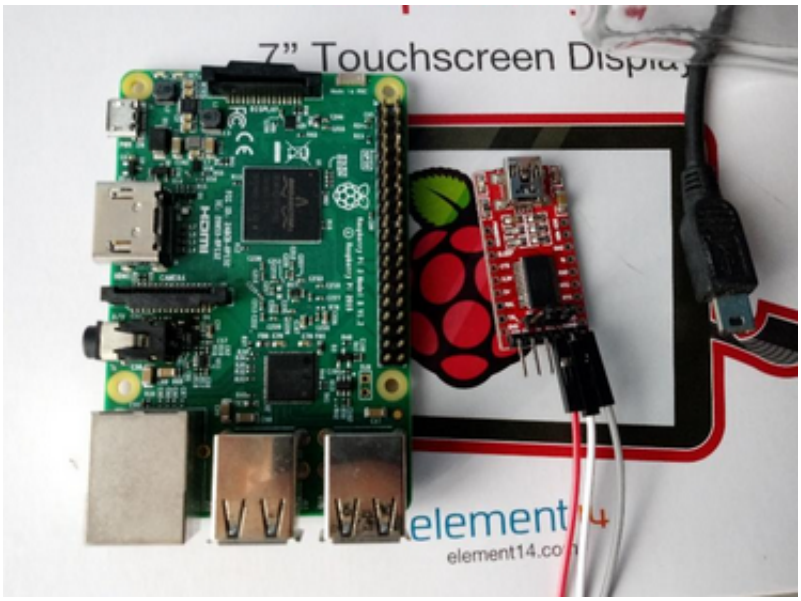


Figura 4.2: Componentes necessários para instalar o ROS

Como previamente comentado, foi feito a mudança do Ubuntu para o Raspbian. Com isso, não se fez necessário a conexão por via TX-RX, já que neste caso, havia o componente gráfico no SO.

Para a instalação do ROS no Rasbpian, foi seguido o tutorial disponível no próprio site do ROS, o ROS.org. Porém, houveram alguns problemas com a instalação do Framework neste SO, desde problemas com dependências, com o próprio ROS etc. Estes percalços ocasionaram em um atraso de alguns dias no projeto, já que até então, não eram conhecidos e mapeados.

4.1.1.5 Instalação do OPENCV na Raspberry

Para as aplicações de visão computacional, deve-se utilizar o OpenCV. Este contém inúmeras bibliotecas que viabilizam e possibilitam a identificação de cores, marcos fiduciais etc. Com isso, faz-se necessário a instalação desta ferramenta na Raspberry a fim de possibilitar ao aluno trabalhar com os princípios da visão computacional.

Há inúmeros tutoriais dispostos na internet para a instalação do OpenCV no sistema da Raspberry, especialmente com o Raspbian. Porém, a complexidade é tão alta quanto a instalação do ROS, por isso, a equipe concluiu que tanto o ROS quanto o OpenCV iriam ser entregues instalados na Raspberry.

Para testar se o OpenCV foi instalado corretamente, foi testado um algoritmo simples de identificação de Arucos, o resultado pode ser visto na figura 4.3 logo abaixo. Este mesmo algoritmo está disposto na WIKI no Github do (LEARNBOTICS, 2019f).

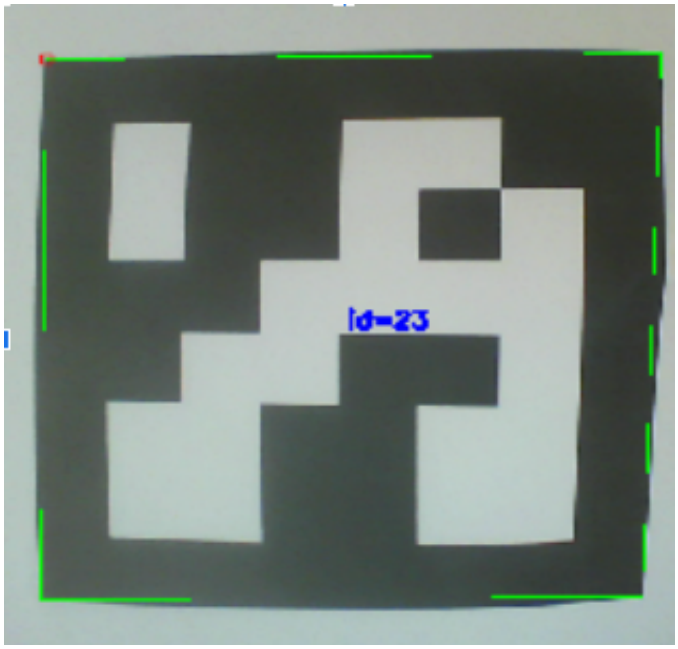


Figura 4.3: Aruco de teste

4.1.2 Software

4.1.2.1 Testes de Software

4.2 Tutoriais

De forma a melhor organizar a elaboração do conteúdo dos tutoriais, uma divisão dos mesmos foi feita de acordo com o assunto abordado, ocasionando assim uma menor abrangência de assuntos a serem pesquisados, de conteúdos a serem concentrados e de novas interpretações a serem elaboradas. As subseções a seguir tratarão de partes específicas dos tutoriais.

4.2.1 Um Breve Histórico da Robótica

siaufgasiogfçwaugfsçfçsuaigfawgfçksgfiauçgwfgsfsaf fsiuafawfioashf saif sfiaywo fasfhaufaw fsifshaf wif saif w

4.2.2 Introdução à Robótica atual e Alguns Conceitos Básicos

4.2.3 Introdução à Atuação

Para que o estudante possa compreender de uma forma mais aprofundada o que está fazendo, antes de começar a mexer com os Dynamixels, uma pequena introdução a atuação foi elaborada.

Essa Introdução trata de uma forma simplificada do conceito de atuadores, de tipos de atuadores, de conversão de energia, e apresenta exemplos cotidianos de atuadores explicando seu funcionamento e aprofundando um pouco mais os conceitos de conversão de energia. (LEARNBOTICS, 2019d)

4.2.4 Introdução à Visão Computacional

Para o ensino de visão computacional apresentado na wiki do Github do Learnbotics (LEARNBOTICS, 2019f), foi-se pensado na explanação dos conceitos de forma branda e intuitiva, já que, conceitos que se relacionam com visão computacional podem torna-se

um tanto rebuscados quando buscados na bibliografia. Os principais conceitos abordados foram:

- O uso da visão computacional;
- Características (*Features*)
- Cantos, arestas e linhas (*Corners,edges e lines*)
- OpenCV
- Segmentação e identificação de cores
- Marcos fiduciais
- Pose
- Identificação de arucos

Os conceitos apresentados são suficientes para o objetivo do trabalho. Após a construção destes textos na wiki (LEARNBOTICS, 2019e), a equipe disponibilizou para uma pequena amostragem de pessoas a fim de receber feedbacks. Dentro desta amostra, haviam pessoas que trabalhavam com o assunto, que conheciam o uso e que não havia conhecimento prévio de visão computacional. Os principais tópicos abordados nos feedbacks foram:

- Boa didática;
- Maior número de exemplos nos conteúdos abordados
- Conceitos apresentados de forma correta e correlacionando com exemplos do cotidiano
- Pequenos erros de digitação e ortografia

Os feedbacks, em resumo, foram positivos dos três grupos, tendo como consequência o auxílio a equipe a alcançar de forma satisfatória o intuito de explanar satisfatoriamente a área da visão computacional.

Como previamente abordado neste trabalho, um dos métodos de ensino proposto pela equipe foi a apresentação de desafios. Nos tutoriais foram apresentados os conceitos atrelados à desafios. Estes desafios tinham como intuito a validação dos temas abordados. Um exemplo de desafio proposto, foi o de segmentação de cores, no qual o usuário deve

segmentar a cor azul. Para que o aluno conseguisse ter êxito neste desafio, a equipe apresentou um algoritmo de segmentação da cor vermelha, previamente testado. O resultado pode ser observado na figura 4.4 abaixo:

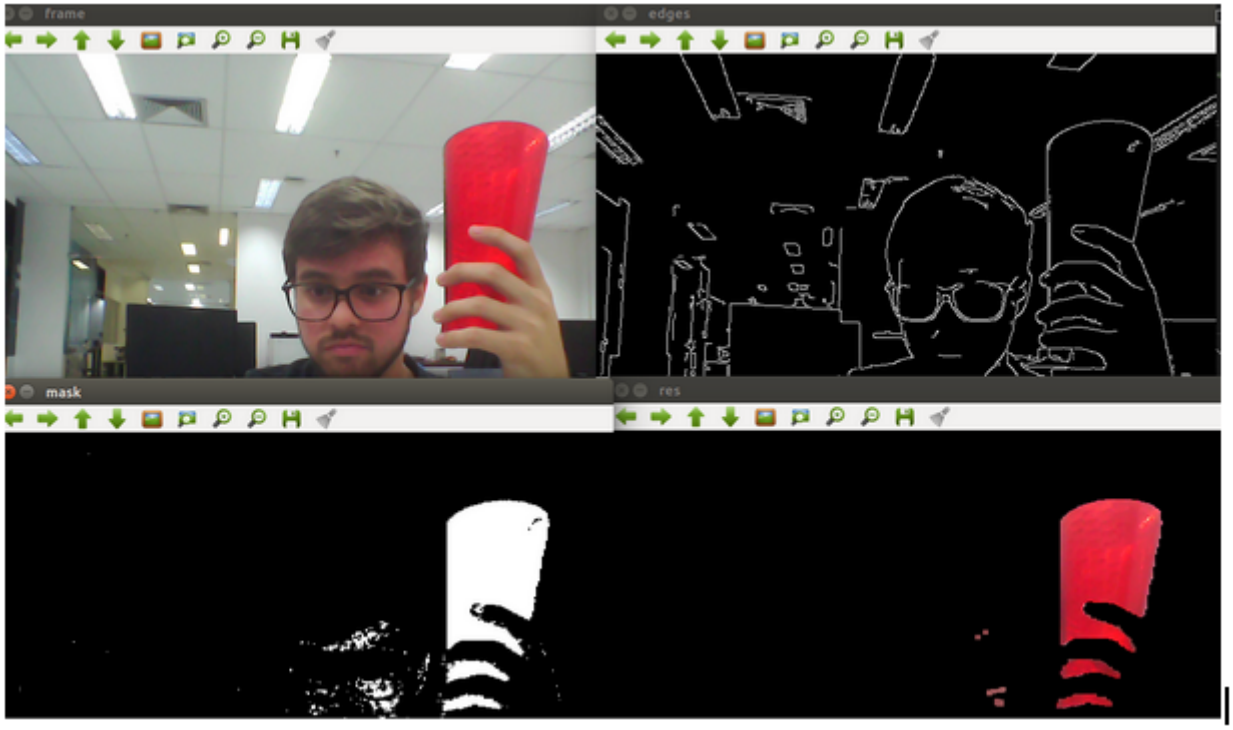


Figura 4.4: Exemplo de segmentação de cores

4.2.5 Tutoriais da Raspberry Pi

4.2.6 Tutoriais dos Dynamixels

Após ter tido contato com o conceito de atuadores, o estudante irá encontrar também um tutorial que faz uma introdução aos servo-motores inteligentes Dynamixel^{TM}.

Neste tutorial são apresentados os servo-motores inteligentes, suas diferenças para servo-motores comuns, suas vantagens sobre os comuns, qual o papel destes servo-motores no robô e no kit físico e mais precisamente porquê escolhemos utilizar os Dynamixels, e não servo-motores comuns. (LEARNBOTICS, 2019a)

4.2.7 Tutoriais do ROS

Tendo em vista a ídeia de apresentar ao estudante ferramentas que são de fato utilizadas por profissionais da área, buscamos realizar um material completo sobre as partes iniciais de utilização do framework ROS.

Devido ao nível de conteúdo que é abordado nos tutoriais nativos do ROS, foi feita uma análise de relevância dos conteúdos e uma reescrita completa do conteúdo abordado, trazendo novas abordagens para passar esse conhecimento para o estudante.

Este tutorial foi dividido em quatro partes, sendo elas, em ordem:

- Introdução: O que é o ROS e como funciona;
- Conceitos Básicos: Apresentação de terminologia e conceitos base utilizados pela comunidade do ROS.
- Entendendo como Funciona o ROS: Apresentação de conteúdo novo que foi elaborado com base em analogias para facilitar o entendimento do estudante sobre a ferramenta.
- Tutoriais do ROS: Os Tutoriais de fato, onde o aluno irá aprender a configurar e utilizar o ROS.

A parte quatro, ou parte dos tutoriais de fato, aborda todos os conceitos de nível iniciante apresentados nos tutoriais oficiais do ROS. Porém estes conceitos foram demonstrados de forma mais simplificada, com linguagem mais simples e de forma acompanhada passo a passo para uma melhor assimilação do estudante.

Todos os tutoriais foram traduzidos do inglês para o poruguês, visando assim facilitar o acesso à uma amostragem maior de estudantes. (LEARNBOTICS, 2019c)

4.2.8 Apresentação dos Scripts de Cinemática

Nesta parte dos tutoriais o estudante terá acesso ao programa que fará com que o seu robô ande. Além disso será ensinado também como o estudante deve proceder para que transforme seu código em um código executável e para rodá-lo.

De forma a estimular o estudante, uma análise mais minusciosa do código, com comentários parte a parte também foi feita. A partir da explicação do que os comandos do programa fazem o estudante terá condições de alterá-lo para concluir etapas e descobrir coisas por si só.

Para finalizar, o tutorial apresenta um desafio para que o estudante de fato assimile o que lhe está sendo proposto, alterando o código e vendo na prática o que isso ocasiona. (LEARNBOTICS, 2019b)

4.2.9 Integração dos assuntos anteriores

Como validação de todos os conceitos apresentados na wiki do github (**??**), a equipe pensou em um desafio que aliasse a visão computacional com os conceitos abordados de cinemática para um robô diferencial.

Inicialmente foi pensado um robô seguidor de cores, porém, este não iria abranger alguns conceitos importantes na robótica, como pose e marcos fiduciais. Tendo isto em vista, a equipe pensou em um desafio que aliasse arucos com a movimentação de um robô diferencial. Sendo assim, o desafio final foi um robô que conseguisse identificar um aruco e se movesse para determinada posição com as informações provenientes deste aruco.

Para que o aluno consiga ter êxito em completar este desafio, a equipe disponibilizou um algoritmo pré testado em que apresenta a comunicação e identificação de arucos, disponível na wiki do Learnbotics (**??**).

4.2.10 Teste do Desafio Final

O desafio final é o de maior complexidade apresentado pela equipe, não apenas pelos conceitos apresentados, mas também por todos os hardwares estarem em funcionamento ao mesmo tempo. Por isso, para o teste deste algoritmo, foi proposto a seguinte metodologia:

1. O primeiro teste foi feito apenas em nível de software, a fim de validar a lógica e encontrar possíveis erros;
2. A próxima etapa foi testar os dynamixels e a webcam utilizando um computador como “cérebro” .
3. A última etapa foi utilizar todos os componentes: Raspberry, dynamixels, webcam, bateria e conversores DCDC. A figura 4.5 apresenta como o teste foi feito.

Figura 4.5: Exemplo de segmentação de cores

4.2.11 Tutoriais de Montagem do Robô

4.3 Kit Físico

4.3.1 Design

4.3.2 Fabricação

4.3.3 Montagem

Capítulo Cinco

Conclusão

"Se vi mais longe foi por estar de pé sobre ombros de gigantes."

(Sir Isaac Newton)

Neste trabalho foi desenvolvido o sistema de movimentação para um protótipo de robô de inspeção de linha. O fluxo metodológico foi uma grande conquista desse projeto, onde o se buscou seguir as diretrizes de desenvolvimento de projetos robóticos de alto porte, utilizando ferramentas de referência para estabelecer um conceito concreto, e também produzindo diversos tipos de documentação que possibilitaram a produção de um conteúdo palpável em relação a esse trabalho.

O desenvolvimento de aplicações que foi realizado para o projeto, consiste em um conteúdo de suma importância para desenvolvimento de projetos similares, buscou-se utilizar as boas práticas de organização e implementar os padrões utilizados para outros projetos de desenvolvimento.

Inicialmente o projeto consistia em uma fuga da zona de conforto, apresentando diversos conteúdos novos e solicitando uma nova visão sobre o trabalho com engenharia, e com o seu caminhar, saiu da concepção da idéia e desafio do aprendizado para o desenvolvimento de um sistema robótico real.

O conceito gerado para o sistema, utiliza parâmetros de operação semelhantes à sistemas de alta complexidade, apresentando um fluxo de informações de alto nível, possibilitando uma noção sobre o funcionamento do sistema e integração com os outros subsistemas já desenvolvidos.

A organização do sistema, junto com suas ferramentas e conhecimentos produzidos, representa uma grande conquista do projeto. O conhecimento deixado relacionado ao *framework ROS*, as conclusões tiradas da ferramenta *MoveIt!*,assim como o comportamento observado pelos dispositivos físicos faz com que haja uma referência para projetos semelhantes, possibilitando a criação de conceitos cada vez mais consistentes e execução de idéias de forma efetiva.

A estrutura do protótipo montada, com suas ligações para comunicação e alimentação feitas devidamente, assim como esquemáticos desenvolvidos especialmente para o projeto,

possibilita que o mesmo seja tomado como base de estudo, ao se realizar o design e desenvolvimento de outros protótipos.

O uso da estrutura do *ROS* possibilita que seja feita a integração com outros sistemas sem muitos problemas, já que foram adotados os padrões do *framework*, assim possibilitando que o que foi desenvolvido para esse robô, seja utilizado em futuras aplicações, sejam elas tomando como base esse protótipo ou outros semelhantes.

A documentação técnica produzida seguiu de acordo com as orientações de projeto visando o reaproveitamento e continuação do trabalho, atendendo as demandas e solicitações para o desenvolvimento do projeto.

O projeto alcançou as expectativas, problemas inesperados encontrados durante o seu desenvolvimento foram contornados e os grandes desafios do projeto foram concluídos, onde a experiência como um todo foi muito enriquecedora e se deu de forma prazerosa, onde todos os participantes conseguiram aprimorar suas habilidades, crescer como pessoa e como profissional. Pelas palavras do sábio Platão: "A coisa mais indispensável a um homem é reconhecer o uso que deve fazer do seu próprio conhecimento."

Apêndice A

QFD

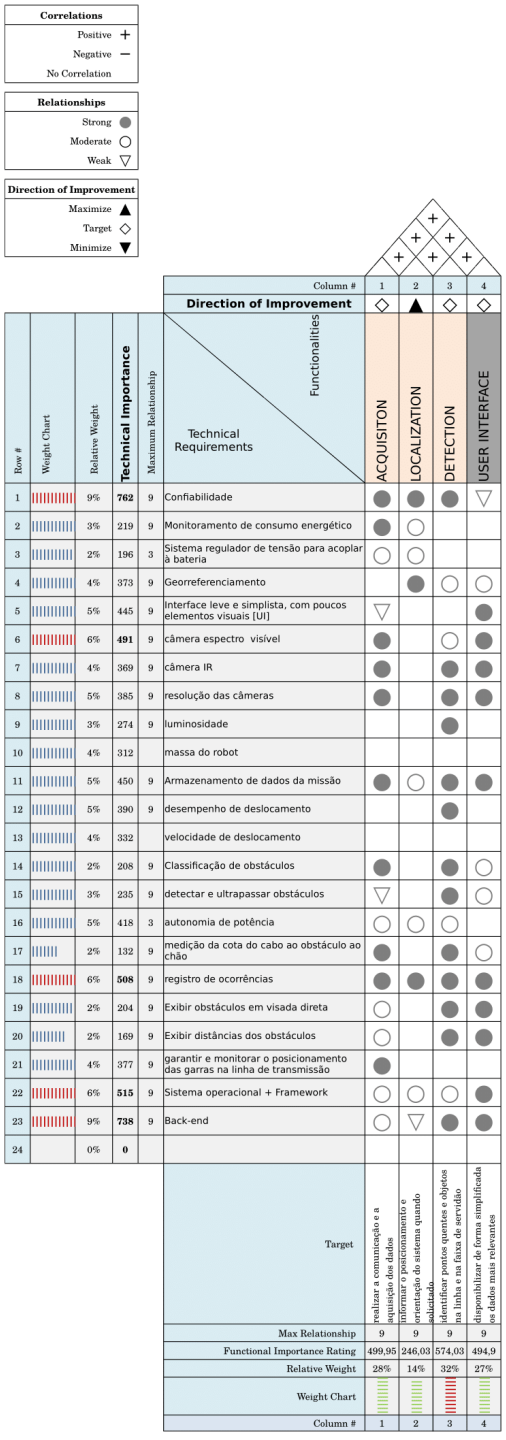


Figura A.1: *QFD* 2

Apêndice B

Arquitetura

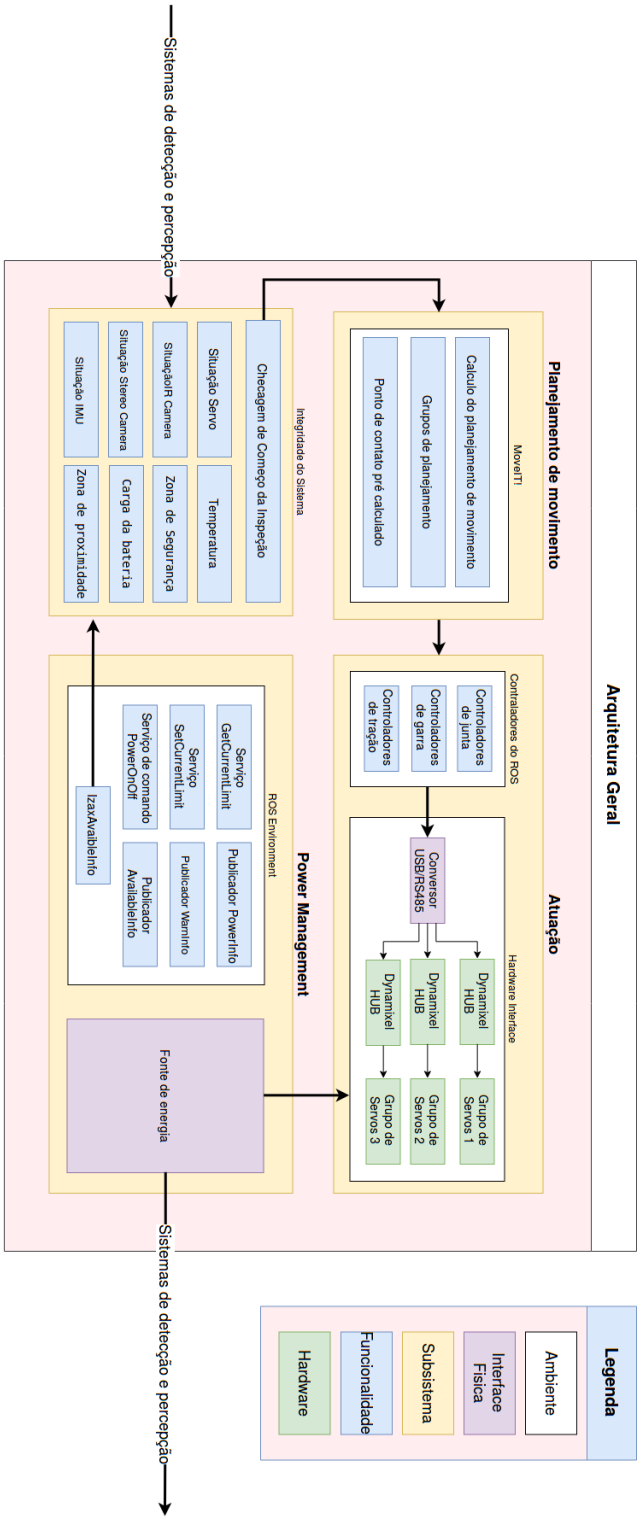


Figura B.1: Arquitetura geral do sistema de movimentação

Apêndice C

Logbook

ELIR

logbook

CONFIGURAÇÃO DOS LIMITES DE GIRO DOS MOTORES

Objetivos

O teste teve como objetivo estabelecer os limites de giro dos motores em seus controladores, com base nos limites físicos da estrutura do robô.

Descrição do teste

É criado um "Controller manager" que conecta os motores e publica em um tópico as informações destes. As juntas do robô são posicionadas manualmente em suas posições maximas e minimas, então o tópico "motor_states" é monitorado para verificar as posições dos motores.

DATA

8 AGOSTO 2018

LOCALIDADE

SENAI CIMATEC
SALVADOR - BAHIA

Mandruvah team

Cleber
Carlos
Ícaro
Davi

17:00

Foram coletados os limites de giro dos motores com id 11, 12, 13, 21, 22 e 23.

17:05

Ajustamos as posições iniciais dos controladores das juntas com base na posição *"home"* da simulação no *MoveIt!*. Nesse momento, verificamos que o valor que é publicado no controlador para mover a junta é a posição em radianos em relação à posição inicial que foi determinada no controlador.

ELIR

logbook

TESTE DE MOVIMENTAÇÃO DOS SERVO-MOTORES

Objetivos

Verificar o comportamento do robô executando alguns movimentos em um dos braços.

Descrição do teste

Os motores são alimentados e seus controladores executados. A partir daí, valores de posição são publicados e o comportamento do robô verificado.

DATA

10 AGOSTO 2018

LOCALIDADE

SENAI CIMATEC
SALVADOR - BAHIA

Mandruvah team

Cleber
Carlos
Ícaro
Davi

15:20

O braço do robo foi levantado até a posição *"home"* com as duas juntas sendo movimentadas ao mesmo tempo. Antes de atingir a posição determinada, o motor com id 21 apresentava erro de overload.

16:10

Quando o robô começa o movimento já proximo da posição final, *"home"*, o braço consegue alcançar o objetivo. Depois de cerca de 5 minutos nessa posição, um erro de overheat é apresentado.

ELIR

logbook

TESTE DE MOVIMENTAÇÃO DO BRAÇO

Objetivos

Verificar possíveis motivos para erro de "Overload" na junta 12-22 apresentado em teste anterior.

Descrição do teste

Os motores são alimentados e seus controladores executados. A partir daí, valores de posição são publicados e o comportamento do robô verificado.

DATA

13 AGOSTO 2018

LOCALIDADE

SENAI CIMATEC
SALVADOR - BAHIA

Mandruvah team

Cleber
Carlos
Ícaro
Davi

16:50

Foi verificado que os motores estão configurados para operar com 100% do torque, não sendo assim essa a causa do problema.

17:15

Percebemos também que o problema acontece com maior frequência quando as juntas do braço e da unidade de tração são acionadas ao mesmo tempo. Quando é acionada uma junta por vez, o "Overload" acontece menos vezes.

17:32

É levantada a suspeita de que a falta do cabo de sincronização nos motores da junta pode ser a causa da falha. Com o cabo conectado, o problema não aconteceu.

ELIR

logbook

TESTE DE CONVERSOR DA PLACA DE POWER MANAGEMENT

Objetivos

Verificar possíveis problemas do conversor da placa de power management e sua resposta de saída .

Descrição do teste

O conversor é retirado da placa e então testado com fonte de alimentação e sua saída observada com um multímetro. O conversor testado é do modelo UWE-12/10-Q12PB-C

DATA

25 SETEMBRO 2018

LOCALIDADE

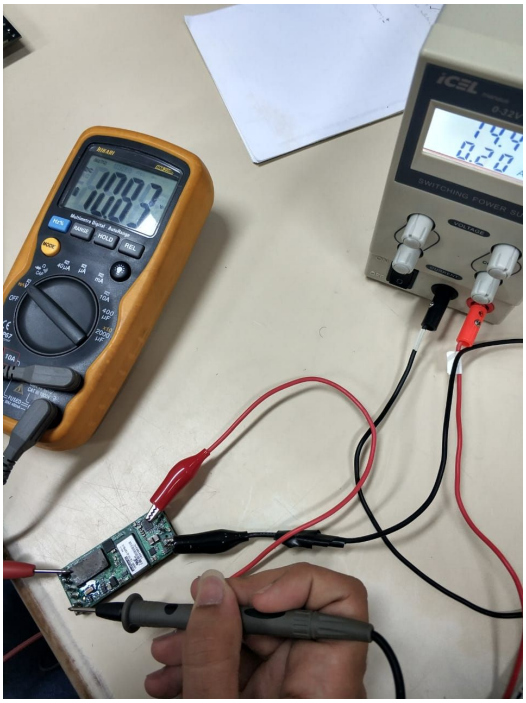
SENAI CIMATEC
SALVADOR - BAHIA

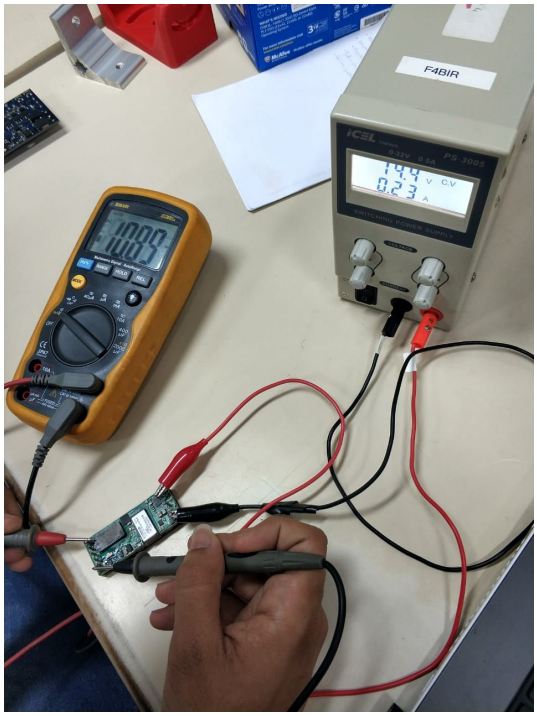
Mandruvah team

Cleber
Carlos
Ícaro
Davi

16:05

O conversor retirado da placa foi testado com alimentação de uma fonte de tensão com 14 Volts, e o mesmo consumiu um valor de cerca de 200 mA. Sua tensão de saída ficou em aproximadamente 10 Volts.



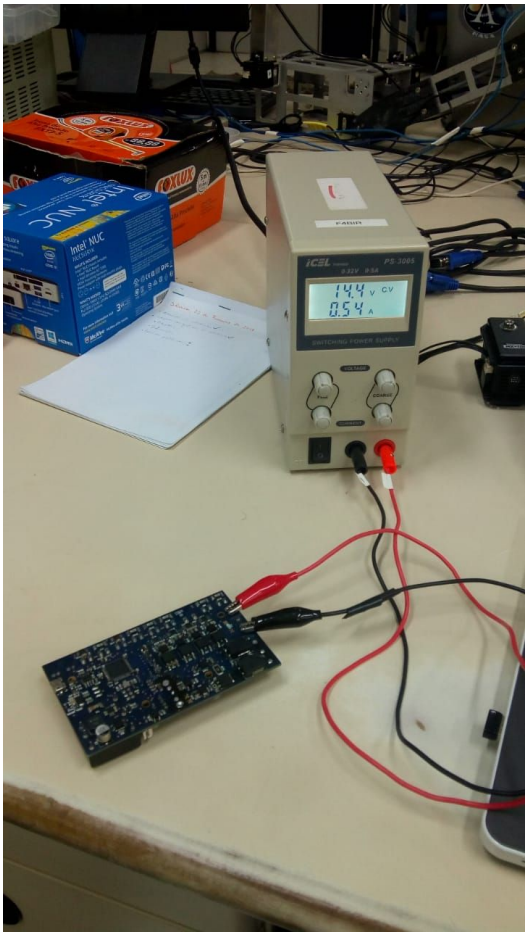


16:12

A corrente requisitada pelo conversor defeituoso oscila entre 200 mA e 250 mA. Sua temperatura, ao contrário do conversor que está funcionando corretamente, não aumenta e o conversor permanece frio.

16:28

O conversor que já está na Power Managemente permanece apresentando funcionamento correto. Sua temperatura aumenta quando permanece ligado.



ELIR

logbook

TESTE DOS MOTORES/CONTROLLER_MANAGER

Objetivos

Identificar se há algum motor defeituoso que pode estar "sujando" a comunicação dos motores.

Descrição do teste

Um motor é conectado e o arquivo controller_manager.launch é executado e verifica-se se o motor foi encontrado. Em seguida, são inseridos os demais motores, um a um, para que se perceba se a comunicação ainda acontece.

DATA

04 OUTUBRO 2018

LOCALIDADE

SENAI CIMATEC
SALVADOR - BAHIA

Mandruvah team

Cleber
Carlos
Ícaro
Davi

13:40

Todos os 18 motores estavam conectados, utilizando apenas os componentes (cabos e hub) da ROBOTIS. Quando o controller_manager.launch foi executado, nenhum motor foi encontrado.

13:42

Com apenas um motor conectado, o controller_manager o encontrou.

13:55

O teste prosseguiu até que, quando o motor de ID 14 foi conectado, o controller_manager não encontrou mais motores.

14:10

O motor de ID 14 foi removido e o teste continuou. O mesmo erro aconteceu quando o motor de ID 3 foi adicionado. Esse motor também foi retirado.

14:15

O teste seguiu até o último motor, nenhum motor aparentemente defeituoso foi encontrado. A comunicação funcionou bem com os 16 motores restantes.

14:45

Dois motores novos foram conectados. O controlador foi executado por volta de 20 vezes, em todos os testes todos os motores foram encontrados.

ELIR

logbook

TESTE DA PLACA DE POWER MANAGEMENT

Objetivos

Verificar possíveis problemas da montagem da placa de power management sem a presença de um dos conversores DC/DC

Descrição do teste

Os capacitores de acoplamento do regulador de tensão para o Atmega 32U4 são soldados e então a placa é alimentada com 14.4 Volts. A temperatura é monitorada com um multímetro com termopar, e os níveis de tensão com um multímetro comum.

DATA

05 OUTUBRO 2018

LOCALIDADE

SENAI CIMATEC
SALVADOR - BAHIA

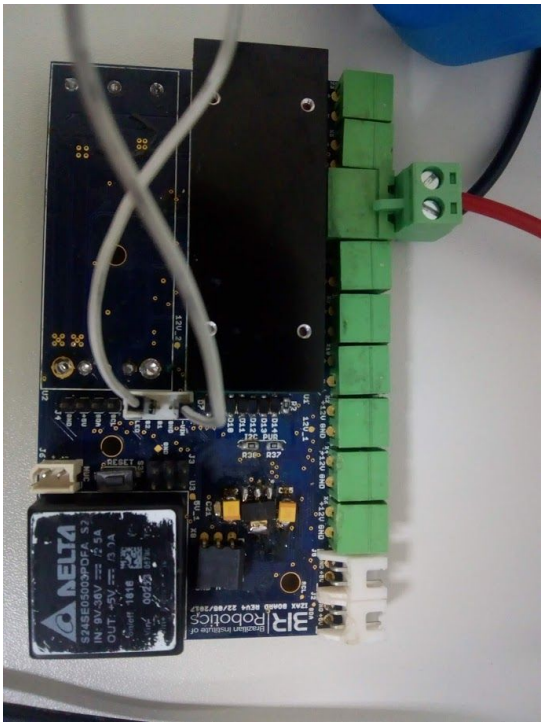
Mandruvah team

Cleber
Carlos
Ícaro
Davi

15:20
Capacitores de acoplamento foram soldados na placa, respeitando a polarização estabelecida no projeto de power management.
Capacitores de tântalo de 10uF de 16 Volts.

15:25
A placa foi alimentada com uma fonte de tensão à 14.4 Volts. Para que haja o funcionamento da placa é necessário realizar um curto entre os pinos -Vin e S2. Desta maneira inicia-se a operação da placa.


DELTA
S24SE05003PDFA S2
IN: 9V-36V
OUT: +5V
RESET
AIR
Robotics

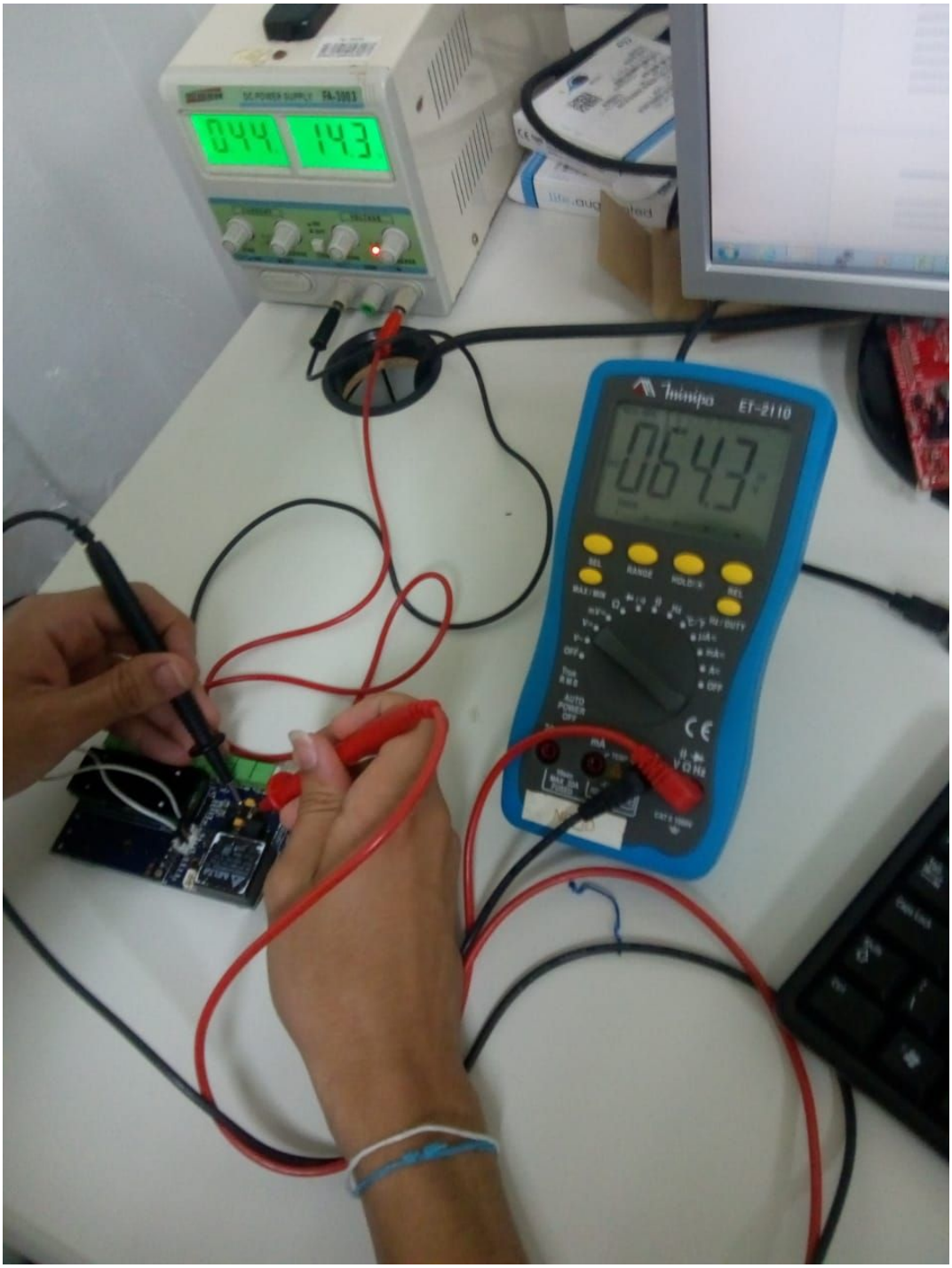

DELTA
S24SE05003PDFA S2
IN: 9V-36V
OUT: +5V
RESET
AIR
Robotics

15:27

A placa apresentou um aumento da temperatura do regulador de tensão 5 Volts, chegando a alcançar temperaturas em cerca de 110°C.

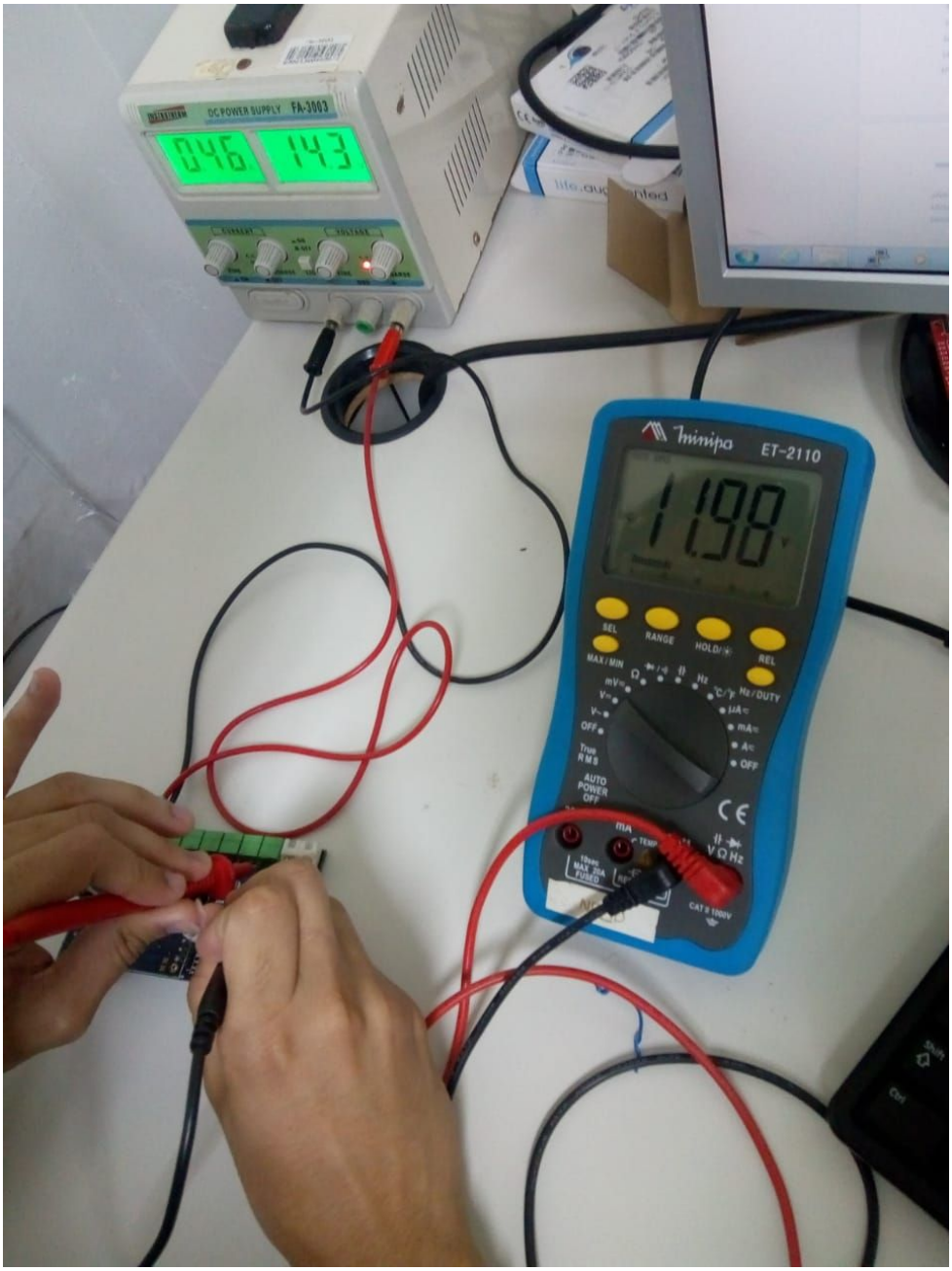
15:28

Os níveis de tensão gerados pelos reguladores também apresentaram níveis fora do esperado. O regulador de 5 volts apresentou tensões extremamente baixas, revelando um possível mal funcionamento, em torno de 70 milivolts



15:30

O conversor ainda presente na placa apresentou um funcionamento normal, gerando 12 Volts nas 3 portas de saída fixa, como esperado.



ELIR

logbook

TESTE COM O ROBÔ NA LINHA

Objetivos

Montar setup do teste, acessar a NUC remotamente e movimentar o robô na linha

Descrição do teste

O robô é posicionado na linha, é estabelecido um acesso remoto, são iniciados os controladores e movimenta-se o robô.

DATA

19 OUTUBRO 2018

LOCALIDADE

SENAI CIMATEC
SALVADOR - BAHIA

Mandruvah team

Cleber
Davi
Ícaro
Carlos

10:40

O robô foi levado até o estacionamento do SENAI CIMATEC onde há uma linha de transmissão de testes. O setup foi montado com uma bateria automotiva, um inversor de frequência 12 - 110V para conectar a fonte da NUC e os motores ligados diretamente à bateria.

11:05

Com o robô na linha e a NUC ligada e conectada a uma rede local exclusiva do teste, conectamos um notebook com o robô via SSH e assim tivemos acesso remoto ao terminal do robô.

11:10

Ao tentar carregar o *controller_manager.launch* nenhum motor foi encontrado. Conferimos então todas as conexões dos hubs de comunicação, nenhum mal contato foi encontrado. Percebemos que um dos terminais do conector do conversor USB-RS485 estava solto da placa.

13:20

Corrigimos o terminal solto do conversor. Novamente carregamos o *controller_manager.launch*, aconteceram erros de "*checksum"* e de *"wrong packet prefix"* e apenas 4 dos 18 motores foram encontrados.

13:40

Após os erros de comunicação, verificamos que quando corrigimos o conversor, os fios D+ e D- foram trocados acidentalmente.

13:45

Ainda depois de corrigir a conexão invertida, não conseguimos encontrar todos os motores, foi levantada a hipótese de que a bateria automotiva era o problema. Medimos a tensão e estava em 12,15V, decidimos voltar ao laboratório e testar com a fonte de bancada.

14:20

Repetimos o teste alimentando o robô com a fonte de bancada e o teste correu bem, nenhum dos erros anteriores aconteceu.

Apêndice D

Lista de componentes

ELIR project - BILL OF MATERIAL

$3.60

id	component	description	brand	part number	power/current	connection	unit cost [R$]	quantity	total cost [R$]
01	interface board		Phidgets	1019_1B	500mA (max)	USB	R$ 272.00	1	R$ 272.00
02	proximity sensor		ETT CO. Ltd	E18-D80NK npn	<25mA	Digital Output	R$ 29.00	5	R$ 145.00
03	temperature sensor		Texas Instruments	LM35	10mA	Analog Output	R$ 7.38	1	R$ 7.38
04	gps		Swift Navigation	Piksi 2.3.1	5V, 500mW	USB		1	R$ 0.00
05	imu		XSENS	Mti-1	44mW	USB		1	R$ 0.00
06	ultrassonic sensor		Maxbotix	EZ-1	5V, 2mA	Analog Output	R$ 107.82	1	R$ 107.82
07	current sensor		Phidgets	1122_0	5V, 10mA	Analog Output	R$ 106.02	3	R$ 318.06
08	lwir camera		FLIR	Lepton 1.0	140mW	I2C		1	R$ 0.00
09	bridge board I		STM	STM32F401 RE	160mA/0.64W	USB	R$ 49.79	1	R$ 49.79
10	bridge board II		STM	STM32L432	140mA/0.56W	USB	R$ 39.56	1	R$ 39.56
11	joint hub		Mandruvah	-	12V	RS485	R$ 25.20	3	R$ 75.60
12	servomotor I		Dynamixel	MX-28	1.4A/16.8W	RS485	R$ 900.00	5	R$ 4,500.00
13	servomotor II		Dynamixel	MX-106	5.2A/62.4W	RS485	R$ 1,800.00	13	R$ 23,400.00
14	adapter 485-usb							1	R$ 0.00
15	battery		Inspired Energy		89Wh/6,2Ah		R$ 879.98	2	R$ 1,759.97
16	power management board		-			USB	R$ 2,200.00	1	R$ 2,200.00
17	multiplex board		Inspired Energy	EB325A	15mA/0.36W	I2C	R$ 1,296.00	1	R$ 1,296.00
18	central processing		Intel	NUC515RYK			R$ 4,300.00	1	R$ 4,300.00
19	stereo camera		Stereolabs	ZED camera	380mA	USB	R$ 1,800.00	1	R$ 1,800.00
20	cabo usb								

Referências Bibliográficas

LEARNBOTICS. *Dynamixels*. 2019. Disponível em: ⟨https://github.com/leo5on/Learnbotics/wiki/Dynamixels-%232⟩. 4.2.6

LEARNBOTICS. *Mãos a Obra!* 2019. Disponível em: ⟨https://github.com/leo5on/Learnbotics/wiki/Dynamixels-%233⟩. 4.2.8

LEARNBOTICS. *Uma Conversa Sobre o ROS*. 2019. Disponível em: ⟨https://github.com/leo5on/Learnbotics/wiki/Uma-Conversa-Sobre-o-ROS⟩. 4.2.7

LEARNBOTICS. *Uma Rápida Conversa Sobre Atuadores*. 2019. Disponível em: ⟨https://github.com/leo5on/Learnbotics/wiki/Dynamixels-%231⟩. 4.2.3

LEARNBOTICS. *Uma rápida conversa sobre visão computacional*. 2019. Disponível em: ⟨https://github.com/leo5on/Learnbotics/wiki/Vis%C3%A3o-computacional-%231#uma-r%C3%A1pida-conversa-sobre-vis%C3%A3o-computacional⟩. 4.2.4

LEARNBOTICS. *Wiki do Github*. 2019. Disponível em: ⟨https://github.com/leo5on/Learnbotics/wiki⟩. 4.1.1.5, 4.2.4

RASBERRY. *Site RaspberryPi*. 2019. Disponível em: ⟨https://www.raspberrypi.org/⟩. 4.1.1.2