

JEFFERSON FERNANDES DE FREITAS

LUIZ GUILHERME SILVA SIMI

**DESENVOLVIMENTO DE UM ROBÔ AUTÔNOMO PARA
LIMPEZA DE UM TANQUE DE PROVAS COM OBSTÁCULOS**

Trabalho de Formatura apresentado à
Escola Politécnica da Universidade de
São Paulo.

Área de Concentração:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Eduardo Aoun Tannuri

São Paulo

2012

Simi, Luiz Guilherme Silva

Freitas, Jefferson Fernandes de

Desenvolvimento de um robô autônomo para limpeza de um tanque de provas com obstáculos / Luiz Guilherme Silva Simi e Jefferson Fernandes de Freitas. – São Paulo: Escola Politécnica da USP, 2012.

Orientador: Eduardo Aoun Tannuri

Trabalho de conclusão de curso – Escola Politécnica da USP / Mecatrônica, 2012

Agradecimentos

Agradecemos aos nossos pais pela ajuda na aquisição e transporte de alguns materiais empregados no trabalho e principalmente pelo grande incentivo.

Agradecemos ao nosso professor orientador, Eduardo Tannuri, pela liberdade que nos deu na concepção deste projeto abrangente, que foi grande desafio para nós. Pudemos por em prática vários conceitos de diversas áreas de mecânica, elétrica, eletrônica e hidráulica. Fomos bem orientados na metodologia do projeto de engenharia e sua documentação.

Agradecemos a FAPESP que, por intermédio do nosso orientador, financiou grande parte do projeto, de modo que não fôssemos restritos aos custos durante a concepção da solução.

Agradecemos aos técnicos que trabalham no laboratório de prototipagem da Poli, devido à ajuda na construção de peças com geometria complexa e nas dicas de manuseio das máquinas-ferramenta.

Agradecemos, enfim, a todos os que contribuíram direta ou indiretamente na realização deste trabalho.

Resumo

O trabalho tem como objetivo o projeto, fabricação e controle de um veículo que se movimenta de maneira autônoma no fundo do Tanque de Provas Numérico (TPN) da Escola Politécnica, pertencente à rede Galileu formada pela Petrobrás.

O robô deverá reconhecer e desviar das estruturas encontradas no fundo do tanque. Ele possuirá liberdade de translação nos três eixos e sua movimentação será feita por meio de jatos de água liberados por bombas de fluxo. O sensoramento será composto por dispositivos de fim de curso dispostos de maneira que seja possível diferenciar obstáculos de paredes.

O controle utilizará um dispositivo programável com taxa de aquisição adequada cujas saídas estarão conectadas a um circuito de potência responsável por ativar as bombas a cada momento. O sistema de controle também prevê um algoritmo que permita ao robô varrer o máximo possível de área do fundo.

Abstract

The objective of this project is the conception, manufacturing and posterior control of an autonomous vehicle that moves around the bottom of the water tank at the Tanque de Provas Numérico of Escola Politécnica, belonging to Galileu net founded by Petrobrás.

The robot must detect and dodge structures installed at the bottom of the tank. The robot will have freedom of movement on three axis and its translation will be done by water pumps water jet. Microswitches will be responsible for sensing and differing obstacles from walls.

Control system will utilize a programmable device with adequate acquisition rate in order to read switches signals. The robot will be provided with a power circuit responsible for pump activation. The algorithm embedded shall allow the vehicle to sweep the majority of the bottom surface of the water tank.

Sumário

Lista de Figuras	8
Lista de Tabelas.....	10
Lista de Gráficos	10
Lista de Símbolos.....	11
1. Introdução.....	12
<i>1.1 Objetivos.....</i>	<i>12</i>
<i>1.2 Justificativa</i>	<i>12</i>
2. Revisão Bibliográfica	13
3. Requisitos de Projeto.....	15
<i>3.1 Área varrida</i>	<i>15</i>
<i>3.2 Duração do ciclo</i>	<i>15</i>
<i>3.3 Velocidade.....</i>	<i>15</i>
<i>3.4 Coluna de água</i>	<i>15</i>
<i>3.5 Sensoriamento</i>	<i>16</i>
<i>3.6 Peso</i>	<i>17</i>
<i>3.7 Durabilidade</i>	<i>18</i>
<i>3.8 Requisitos qualitativos.....</i>	<i>18</i>
3.8.1 Facilidade de fabricação	18
3.8.2 Facilidade de manutenção.....	18
3.8.3 Segurança.....	18
4. Propostas iniciais	19
5. Projeto Básico	21
<i>5.1 Solução geral.....</i>	<i>21</i>
<i>5.2 Atendimento de requisitos</i>	<i>23</i>
5.2.1 Área varrida	23
5.2.2 Duração do ciclo	24
5.2.3 Velocidade	25
5.2.4 Coluna de água	25
5.2.5 Sensoriamento	25

5.2.6	Peso.....	27
5.2.7	Durabilidade	28
5.2.8	Facilidade de fabricação	29
5.2.9	Facilidade de manutenção.....	29
5.2.10	Segurança.....	30
6.	Projeto Executivo	31
6.1	<i>Projeto mecânico.....</i>	<i>31</i>
6.1.1	Movimentação	31
6.1.2	Bocal.....	35
6.1.3	Base dos sensores	39
6.1.4	Sistema de flutuação	41
6.1.5	Encapsulamento da eletrônica	42
6.2	<i>Projeto elétrico e eletrônico.....</i>	<i>43</i>
6.2.1	Contatos	43
6.2.2	Circuito de comando	45
6.2.3	Circuito de potência	49
6.3	<i>Programa embarcado.....</i>	<i>50</i>
6.3.1	Ciclo de limpeza	50
6.3.2	Lógica de rebatimento	51
6.3.3	Lógica de pulo	53
6.3.4	Watchdog de segurança	55
7.	Resultados	57
7.1	<i>Movimentação XY.....</i>	<i>57</i>
7.2	<i>Movimentação Z.....</i>	<i>57</i>
7.3	<i>Durabilidade</i>	<i>58</i>
7.4	<i>Requisitos qualitativos.....</i>	<i>59</i>
8.	Conclusões.....	60
9.	Referências bibliográficas.....	61
10.	Apêndices.....	62
10.1	<i>Modelo do fundo do tanque</i>	<i>62</i>
10.2	<i>Desenhos de fabricação do veículo</i>	<i>63</i>
10.3	<i>Desenhos de conjunto.....</i>	<i>74</i>
10.4	<i>Fotos.....</i>	<i>76</i>

<i>10.5 Algoritmo de otimização no Matlab</i>	<i>82</i>
<i>10.6 Codificação do programa embarcado no Arduino</i>	<i>86</i>

Lista de Figuras

Figura 2-1 Modelos de limpadores de piscina existentes no mercado	13
Figura 3-1 Fotos do TPN com detalhe para estruturas instaladas no fundo	16
Figura 3-2 Vista lateral do fundo do tanque com ênfase nos obstáculos	17
Figura 3-3 Vista superior do fundo do tanque com ênfase nos obstáculos	17
Figura 5-1 Modelo do veículo em vista isométrica	21
Figura 5-2 Vista lateral do modelo com detalhe interior	21
Figura 5-3 Foto do robô concluído	22
Figura 5-4 Esquema de movimentação	24
Figura 5-5 Modelo da base dos sensores na parede do veículo	27
Figura 5-6 Foto de parede com sensores e bomba montados	27
Figura 5-7 Modelo do sistema para redução do peso submerso	28
Figura 5-8 Foto do sistema de flutuação montado no topo do veículo	28
Figura 6-1 Esquema de experimento para levantar força das bombas	32
Figura 6-2 Vista lateral em corte do bocal dimensionado	38
Figura 6-3 Modelo do anel para fixação do bocal	38
Figura 6-4 Bocal pronto para montagem em bomba de porão	39
Figura 6-5 Base dos sensores inferiores	40
Figura 6-6 Base dos sensores superiores	40
Figura 6-7 Foto do tubo de encapsulamento do sistema eletrônico após construção	43
Figura 6-8 Esquema do circuito de comando (sinais de sensoriamento)	47
Figura 6-9 Circuito de comando montado	47
Figura 6-10 Esquema do circuito de comando (sinais de acionamento)	48

Figura 6-11 Projeto da trilha para o circuito de acionamento (à esquerda) e circuito concluído (à direita).....	49
Figura 6-12 Placa auxiliar com fusíveis de 5 A.....	50
Figura 6-13 Tomadas de decisão possíveis durante varredura	51
Figura 6-14 Ângulos de incidência e reflexão.....	52
Figura 6-15 Diagrama de atividades UML.....	53
Figura 6-16 Tomadas de decisão possíveis durante o pulo	54
Figura 6-17 Diagrama de atividades UML descrevendo o algoritmo	56
Figura 7-1 Fotos indicando sequência de salto.....	58
Figura 10-1 Vista lateral do fundo do tanque modelado	62
Figura 10-2 Base do veículo.....	63
Figura 10-3 Topo do veículo	64
Figura 10-4 Parede sem bomba	65
Figura 10-5 Parede com bomba.....	66
Figura 10-6 Base dos sensores superiores	67
Figura 10-7 Base dos sensores intermediários	68
Figura 10-8 Base dos sensores inferiores	69
Figura 10-9 Braçadeira tipo 1	70
Figura 10-10 Braçadeira tipo 2.....	71
Figura 10-11 Bocais	72
Figura 10-12 Braçadeira para fixação do bocal.....	73
Figura 10-13 Desenho de conjunto do veículo.....	74
Figura 10-14 Desenho de conjunto da parede do veículo	75
Figura 10-15 Parede do veículo com as três bases dos sensores.....	76
Figura 10-16 Parede do veículo com bases dos sensores e bomba com bocal.....	76
Figura 10-17 Parede concluída.....	77

Figura 10-18 Bombas montadas com braçadeiras e bocais	78
Figura 10-19 Fundo do veículo com rodízios, bombas e bocais montados.....	78
Figura 10-20 Chassis aberto	79
Figura 10-21 Eletrônica montada	79
Figura 10-22 Eletrônica vedada e acoplada ao veículo	80
Figura 10-23 Veículo montado sem vedação da eletrônica.....	80
Figura 10-24 Veículo concluído	81

Lista de Tabelas

Tabela 6-1 Características teóricas das bombas de porão adquiridas.....	32
Tabela 6-2 Resultados do experimento para levantar força das bombas.....	33
Tabela 6-3 Valores do coeficiente de perda de carga em bocais	36
Tabela 6-4 Massa dos elementos constitutivos do veículo.....	42
Tabela 6-5 Resultados do experimento com botões submersos não vedados	44
Tabela 6-6 Resultados do experimento com botões submersos vedados	44
Tabela 6-7 Transições de estado de acordo com sensores ativados	46

Lista de Gráficos

Gráfico 6-1 Curvas de carga máxima por tensão das bombas de 500GPH e 1000GPH	34
Gráfico 6-2 Curvas de carga por vazão a 13.5V das bombas de 500GPH e 1000GPH .	35
Gráfico 6-3 Força da bomba de 1000GPH em função das variáveis do problema	36
Gráfico 6-4 Resultado da otimização em curvas de nível com restrição.....	37

Lista de Símbolos

$A_{saída}$	Área de saída da bomba
A_1/A_2	Relação entre a área de entrada A_1 e a área de saída A_2 em bocais
E	Empuxo
F_{bomba}	Força exercida pela bomba de porão
H_{1000}	Carga manométrica da bomba de 1000GPH
H_{500}	Carga manométrica da bomba de 500GPH
ΔH	Perda de carga
K	Constante de perda de carga de bocais
L	Distância mínima perpendicular às áreas de saída e entrada do bocal
P	Peso de um corpo identificado por termo em subscrito
Q	Vazão volumétrica
V	Volume de corpo identificado por termo em subscrito
d_1	Diâmetro dos dutos usados na flutuação
d_2	Diâmetro do duto usado na eletrônica
g	Aceleração da gravidade
h	Altura de um cilindro
m	Massa do corpo indicado por termo subscrito
p	Porcentagem de ar no invólucro da eletrônica
v	Velocidade do fluido
ρ	Densidade do material indicado por termo subscrito
θ	Ângulo de abertura do bocal

1. Introdução

1.1 Objetivos

Projetar, fabricar e controlar um robô que se movimenta no fundo de um tanque de provas de forma autônoma. O sistema deve permitir o reconhecimento de obstáculos e garantir que o veículo desvie deles. O projeto também prevê a futura instalação de um sistema completamente operacional de limpeza que poderá ou não ser implementado até de acordo com o sucesso das etapas de fabricação e controle.

1.2 Justificativa

A motivação principal do trabalho é limpar o Tanque de Provas Numérico da USP. Atualmente a limpeza deste não é feita com adição de cloro, ao invés disso é utilizado um produto químico que aglutina as partículas sólidas em suspensão fazendo com que se depositem no fundo. Desse modo, um sistema que filtrasse a água do fundo do tanque seria eficiente na tarefa de eliminar tais resíduos, desde que fosse capaz de desviar de certos equipamentos instalados no local.

É interessante, também, para nós alunos da graduação da mecatrônica participar do projeto integrado de uma solução de engenharia que envolve áreas de projeto, mecanismos, fabricação, acionamento, sensoramento, programação e controle.

2. Revisão Bibliográfica

Antes de iniciar a análise do projeto que é tema desta dissertação, serão apresentados alguns modelos de limpadores de piscina encontrados no mercado, suas características e limitações com relação à implementação no TPN.

Uma das especificações do projeto é limpar o tanque usando o princípio de sucção e filtragem da água. No mercado existem diversos modelos de limpadores autônomos de piscina que também funcionam da mesma maneira.

Modelos como o Aquabot [1] e o Dolphin [2] mostrados na Figura 2-1, de origem norte-americana, podem varrer por completo o fundo da piscina e escová-la. Movimentam-se por meio de esteiras motorizadas cuja alimentação é externa. Eles são dotados de um sistema que suga a água por baixo e a libera por cima, que serve tanto para filtrar quanto para prender o veículo à superfície que está limpando. Assim, alguns modelos podem se movimentar pelas paredes da piscina até a linha de água. O modelo Verro da iRobot [3] possui, além dos demais, um sistema que libera cloro controladamente durante o ciclo de limpeza.

O interessante desses modelos é que eles possuem todos os sistemas embarcados. São autônomos, diferente de aspiradores de piscina que dependem da própria bomba de sucção da piscina para funcionar ou da presença de um operador.



Figura 2-1 Modelos de limpadores de piscina existentes no mercado

A principal limitação dos modelos existentes no mercado, no entanto, é que eles não são capazes de superar obstáculos de lateral não plana como é o caso da cremalheira circular instalada no fundo do TPN, esquematizada na seção 3.5 deste trabalho. Os

veículos que escalam paredes de piscinas, por exemplo, precisam de uma superfície contínua na qual possam se prender por meio de sucção. Por esse motivo se faz necessário o projeto de um robô que consiga lidar com obstáculos que são tão específicos a esse ambiente.

3. Requisitos de Projeto

Nesta seção serão discutidos os requisitos que o veículo deverá atender ao ser implementado no ambiente de trabalho. Os requisitos a seguir são de funcionalidade, manutenção e durabilidade, arbitrados a fim de viabilizar o emprego da solução e de não comprometer o funcionamento do TPN. Alguns aspectos não foram considerados relevantes, como estética e custo total do projeto. Consumo de energia elétrica também não foi limitado, pois o veículo possuirá fonte externa em vez de uma bateria embarcada.

3.1 Área varrida

Impõe-se que 80% da área do fundo do tanque deverá ser percorrida durante um ciclo de limpeza, no mínimo. Não existem áreas prioritárias, mas se faz essencial que o ciclo seja realizado dentro e fora da área delimitada pela cremalheira (discutida adiante na seção 3.5).

3.2 Duração do ciclo

A princípio, o tempo delegado ao robô para realizar a limpeza do ambiente de trabalho é de 3 horas. Entretanto, há uma grande tolerância quanto a esse período e o requisito pode ser alterado de acordo com o comportamento do veículo desde que não comprometa a alocação do tanque por projetos prioritários.

3.3 Velocidade

O robô deverá se locomover com velocidade média de 0.2m/s como mínimo necessário para percorrer o tanque no período de 3 horas estipulado para o ciclo de limpeza. Em caso do aumento da duração do ciclo especificada, a velocidade do veículo pode ser inferior ao especificado contanto que a área varrida seja de no mínimo 80%.

3.4 Coluna de água

Devido às condições padrões de operação do TPN, o veículo deverá ser capaz de realizar todas suas funcionalidades enquanto submerso, sujeito a até 4 metros de coluna de água.

3.5 Sensoriamento

O robô deverá detectar e desviar dos obstáculos presentes no fundo do tanque de provas. Deverá ser capaz também de distingui-los, uma vez que a tomada de decisão decorrente do encontro com cada conjunto de obstáculos é diferente.

A Figura 3-1 mostra o fundo do tanque onde podem ser notados os obstáculos instalados. Tratam-se de uma cremalheira circular com 4 m de raio e um aparato triangular apoiado sobre ela com liberdade de rotação em torno de um cilindro posicionado no centro do tanque. Há ainda pinos com 3 cm tanto de altura quanto diâmetro engastados arbitrariamente ao longo do fundo. Um modelo desses elementos pode ser observado em vista lateral na Figura 3-2 e vista lateral na Figura 3-3.

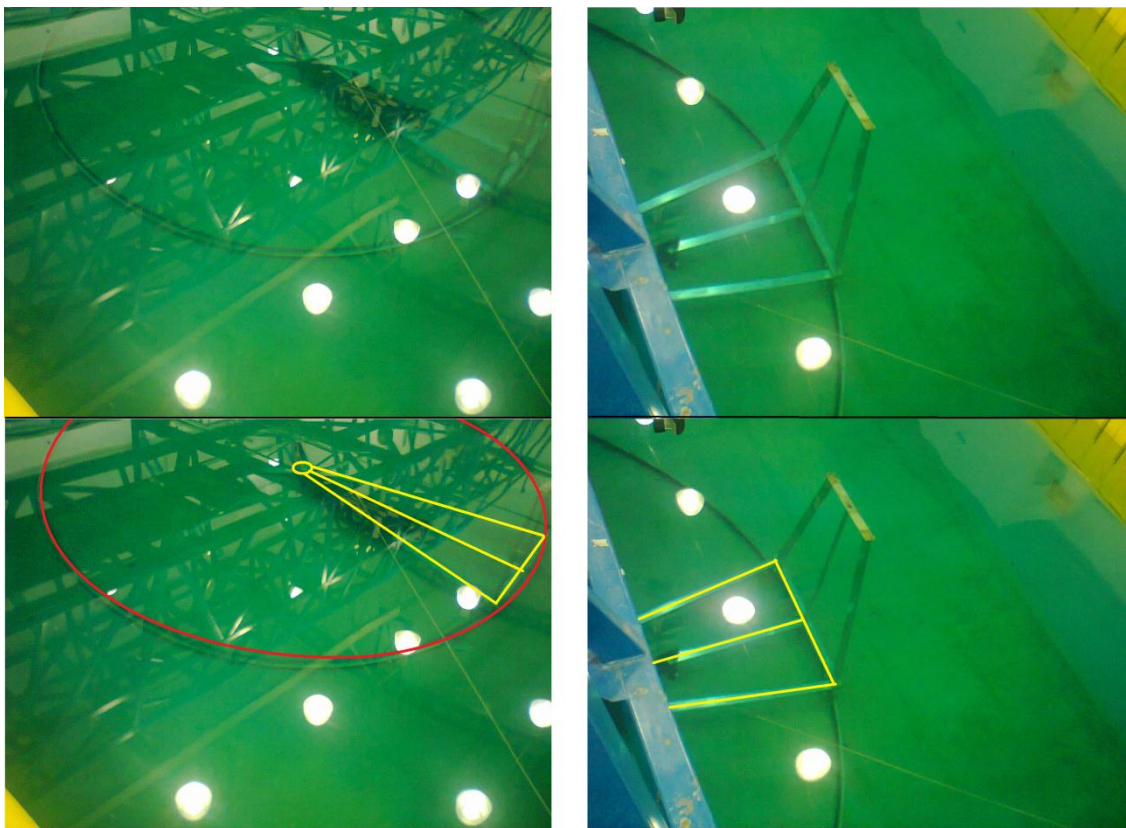


Figura 3-1 Fotos do TPN com detalhe para estruturas instaladas no fundo

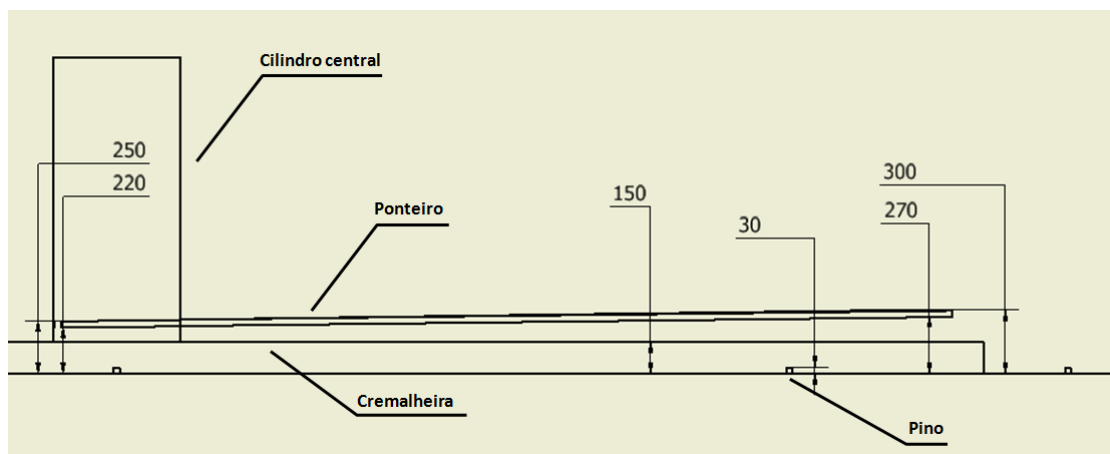


Figura 3-2 Vista lateral do fundo do tanque com ênfase nos obstáculos

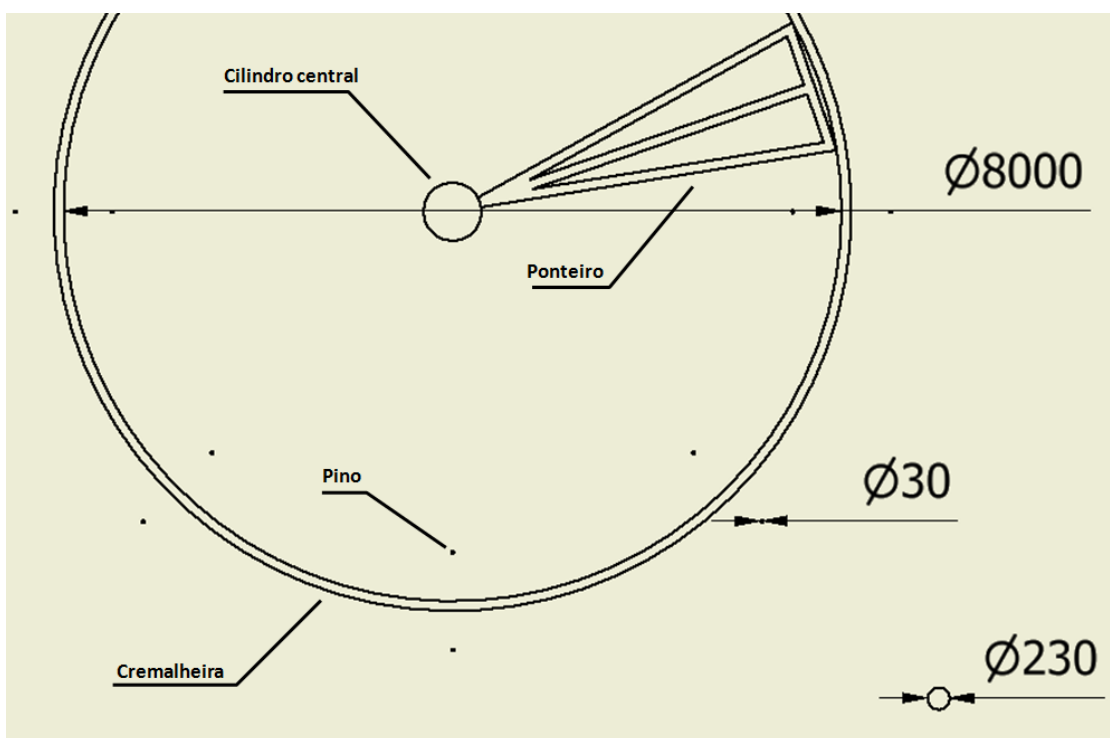


Figura 3-3 Vista superior do fundo do tanque com ênfase nos obstáculos

3.6 Peso

O peso submerso do robô deverá ser tal que ele mantenha contato com o fundo do tanque na maior parte do ciclo de limpeza exceto quando for ultrapassar um obstáculo, momento em que o peso deverá ser inferior ao esforço dos atuadores verticais.

3.7 Durabilidade

O robô deverá ter vida útil semelhante aos limpadores de piscina no mercado, cerca de dois anos, com carga de trabalho semanal de 6 horas.

3.8 Requisitos qualitativos

Há também requisitos qualitativos, mas ainda assim relevantes para as tomadas de decisão do projeto.

3.8.1 Facilidade de fabricação

Devido à disponibilidade de tempo e recursos, no desenvolvimento das peças que compõem o protótipo é preferível evitar o emprego de máquinas ferramentas programáveis, assim como reduzir a necessidade de máquinas que exijam elevado tempo de trabalho (torno, fresadora, etc.).

3.8.2 Facilidade de manutenção

O acesso, tanto às bombas e às peças dos mecanismos de fim de curso quanto aos elementos eletrônicos, deve ser facilitado, de modo a permitir a troca e devida manutenção dos mesmos na eventualidade de uma pane ou na simples necessidade de substituição por fim de vida útil.

3.8.3 Segurança

O veículo deve ser livre de elementos cortantes e as partes elétricas devem ser devidamente isoladas, de modo a não gerar correntes que sejam nocivas a um utilizador que esteja submerso ao lado, ou tocando, no robô.

4. Propostas iniciais

Foram levantadas diversas alternativas mecânicas, de acionamento, eletrônicas e sensoriais, a fim de atender os requisitos enumerados anteriormente. Nesse capítulo serão discutidas essas alternativas podendo-se assim justificar algumas escolhas iniciais do projeto.

Estrutura		Movimentação	Acionamento	Controle
Geometria	Material			
- Poligonal (4 faces)	- Alumínio	- Motores	- Driver	- Malha aberta
- Poligonal (n faces)	- Inox	- Bombas	baseado em L297/L298	- Malha fechada
- Circular	- Plástico		- Relês	

A respeito da estrutura do veículo, qualquer geometria cuja superfície lateral possa ser inteiramente rastreada é válida. A diferença está em como implementar o sensoriamento e como utilizar a aquisição de informações no acionamento do robô. A geometria também influencia nas características hidrodinâmicas do sistema. No princípio foram cogitadas geometrias com simetria radial quando vistas de cima, entre elas os formatos prismáticos com bases formadas por polígonos de diferente número de lados ou o formato cilíndrico. O prisma de base hexagonal foi escolhido devido ao fato do veículo possuir liberdade para se movimentar em seis sentidos no plano horizontal (tal característica de projeto é explicada na seção 5.2.1, que descreve como o protótipo realiza a varredura). Sendo hexagonal, independente de qual dentre os seis sentidos o carro se desloca haverá sempre uma parede dotada de sensores à dianteira mais capaz de identificar obstáculos do que uma aresta posicionada à frente durante o movimento.

Quanto ao material a ser empregado na estrutura do robô, a maior preocupação foi a vida útil enquanto submerso. Na aquisição de componentes para o veículo foi dada preferência aos compostos por alumínio, inox, plástico e latão.

A alternativa de movimentação via rodas motorizadas, empregada nos modelos de robôs limpadores de piscinas encontrados no mercado, foi descartada devido à decisão

de saltar a cremalheira encontrada no fundo do tanque. Para se movimentar horizontalmente sem atrito com o solo serão usados jatos d'água emitidos por bombas de porão instaladas nas laterais do veículo.

Depois de tomada a decisão de movimentação, o controle de posicionamento poderia ser feito em malha fechada, com o uso de acelerômetros para inferir a curva de deslocamento do veículo, ou malha aberta, em que os distúrbios não são rastreados pelo sistema de controle. Optou-se pelo uso de controle por malha aberta auxiliado por um algoritmo que não mapeia a área do fundo do tanque. Ao invés disso, o programa decide o sentido pelo qual o robô deve seguir após a detecção de cada obstáculo a fim de desviar do mesmo. Um controle menos robusto, mas de fácil implementação, cuja aleatoriedade inserida no movimento foi considerada não prejudicial quando levado em conta o período do ciclo de limpeza.

As alternativas de acionamento das bombas de porão consideradas foram o uso de relês, que apenas restringem ou permitem totalmente a passagem de corrente pela bomba, e o emprego de um driver baseado em CIs L297 e L298, capaz de variar analogicamente a corrente transferida para a movimentação. Ambas as aplicações são igualmente válidas no veículo devido ao não mapeamento do ambiente de trabalho, então se optou pelo uso de relês.

5. Projeto Básico

5.1 Solução geral

Após consideração de diversas alternativas, chegou-se à solução geral cujo modelo é apresentado abaixo em vistas isométrica na Figura 5-1 e vista lateral com chassis aberto na Figura 5-2. A Figura 5-3 exibe o protótipo em seu estágio final.

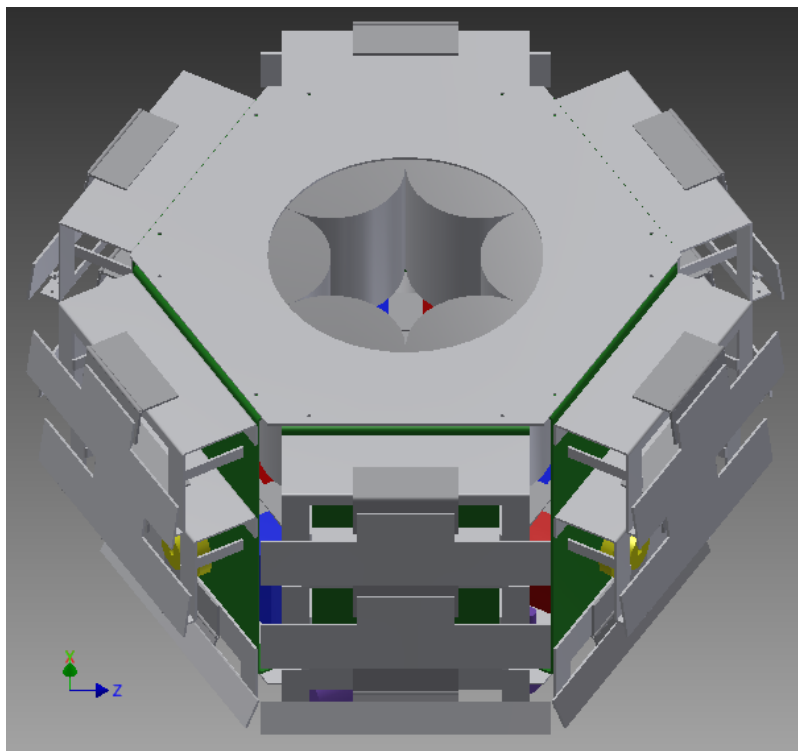


Figura 5-1 Modelo do veículo em vista isométrica

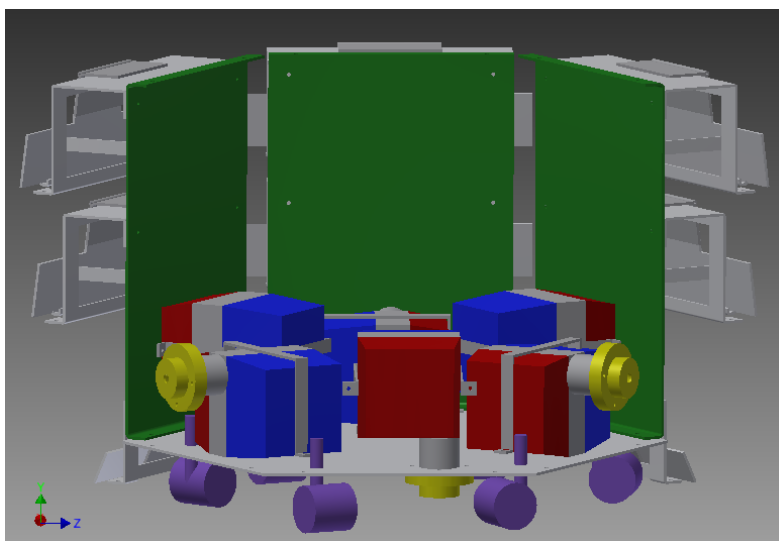


Figura 5-2 Vista lateral do modelo com detalhe interior

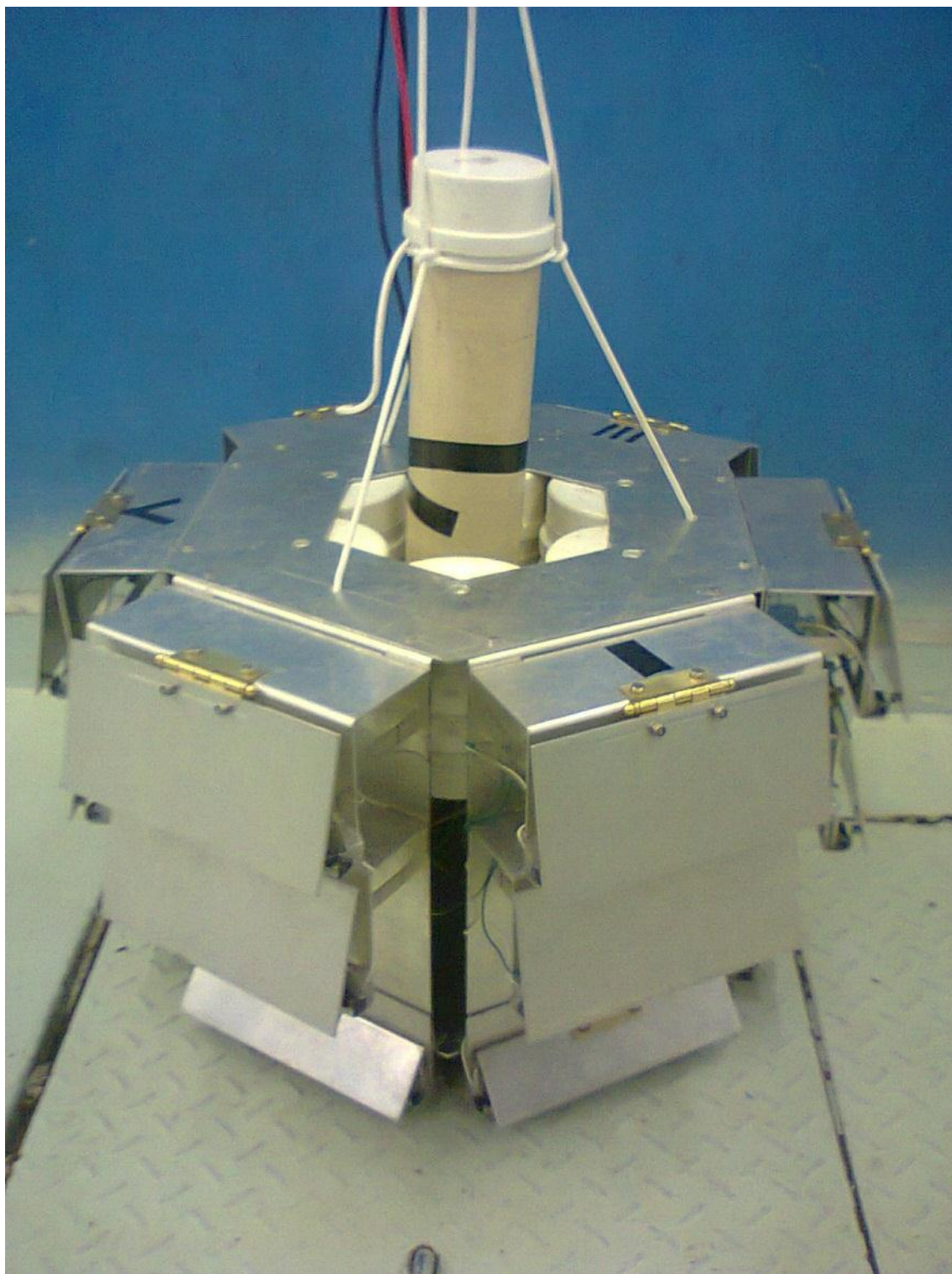


Figura 5-3 Foto do robô concluído

O veículo é composto por duas chapas hexagonais servindo como base e topo da estrutura. Possui seis paredes retangulares e cada parede é dotada de três conjuntos mecânicos possuindo botões de fim de curso com função de sensoriamento e distinção

entre obstáculos de altura baixa, média e alta. Todos os desenhos de fabricação se encontram na seção 10.2, e fotos gerais do projeto estão na seção 10.4.

A movimentação é feita com o uso de seis bombas de porão de 1000GPH [4] com bocais, três dispostas com a saída na horizontal, acopladas às paredes e outras três dispostas com a saída na vertical, acopladas à base do robô. A locomoção horizontal é facilitada pela inclusão de seis rodízios no inferior do veículo e a vertical é auxiliada por um sistema de flutuação composto por tubos de PVC vedados e contendo ar. A estrutura não é hermeticamente vedada, de modo que as bombas trabalham submersas, porém a eletrônica foi encapsulada por PVC.

O circuito que trata dos sinais enviados pelos botões é composto por resistências e um Arduino, que por sua vez aciona um conjunto de transistores e relês responsáveis por alimentar as bombas de acordo com a situação. A alimentação do veículo é realizada por cabos conectados a uma bateria externa.

5.2 Atendimento de requisitos

A seguir, serão justificadas brevemente quais decisões do projeto foram feitas de modo a atender os requisitos listados anteriormente.

5.2.1 Área varrida

Para varrer a área inferior do tanque, o veículo precisa ter liberdade de movimentação em dois eixos, definindo assim um plano de trabalho. Como os atuadores empregados são bombas hidráulicas não é possível deslocar o robô em dois sentidos opostos simplesmente revertendo a tensão de alimentação das bombas, pois a direção de entrada do fluido é perpendicular a de saída. Assim, para se deslocar ao longo do plano são usadas três bombas acopladas às paredes não adjacentes, com o duto de saída alinhado ao ponto médio da lateral. Acionando uma ou duas delas por vez é possível percorrer seis sentidos defasados de 60° cada.

Um esquema do sistema de movimentação é apresentado na Figura 5-4. Para avançar na direção das setas em azul o veículo, representado pelo hexágono como se visto por cima, precisa acionar as bombas em vermelho. Além disso, a Figura 5-2 indica o posicionamento dos rodízios que, embora não acionados, desempenham papel

importante na locomoção por possuírem um grau de liberdade azimutal podendo adaptar seu eixo de rotação de acordo com o deslocamento do veículo.

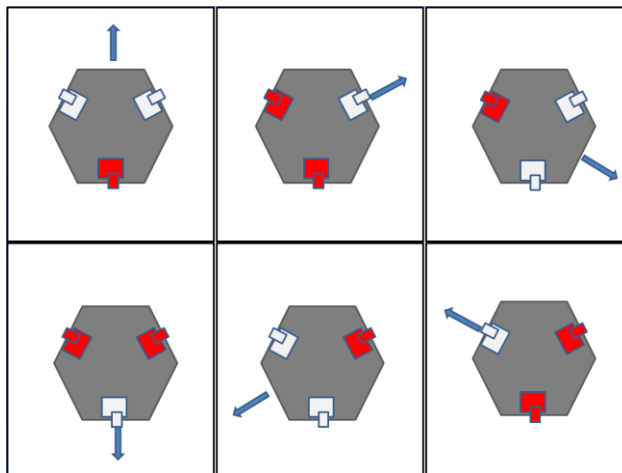


Figura 5-4 Esquema de movimentação

Dado então que o veículo consiga se locomover livremente no plano, ainda é preciso atender ao requisito de área mínima varrida. Essa tarefa está delegada ao algoritmo de tratamento dos sensores, embarcado no Arduino, que toma decisões visando impedir que o robô fique preso em algum local ou que ele entre em um laço repetitivo durante o trajeto.

5.2.2 Duração do ciclo

O cumprimento desse requisito é de responsabilidade total do programa implementado no sistema, que deve desligar o veículo após decorrido o tempo máximo delimitado previamente. É interessante notar que o algoritmo tenta reduzir ao mínimo a necessidade de se deslocar verticalmente, tarefa que exige cuidado e despende muito tempo. Para isso, o ciclo será dividido em três etapas: a primeira visará varrer a área entre as laterais do tanque e a cremalheira, a segunda se destinará ao salto que o robô dará para adentrar a circunferência e a terceira etapa cobrirá somente o círculo central. (As características do fundo do tanque estão apresentadas na seção 3.5)

Outra medida tomada para tentar garantir que maior parte do ciclo de trabalho seja destinada efetivamente à limpeza do tanque é a implementação de um temporizador que detecta se o veículo está preso ou andando em círculos e aciona brevemente o sistema de movimentação para se livrar e voltar à atividade anterior. O algoritmo está detalhado completamente na seção 6.3.

5.2.3 Velocidade

Para regular a velocidade do veículo através do acionamento por bombas de porão é possível tanto aumentar a alimentação delas ou reduzir suas áreas de saída. Serão apresentados no item 6.1.1 os ensaios realizados com as bombas de porão para determinar a resultante máxima exercida por elas e em seguida, no item 6.1.2, será apresentado o dimensionamento de um bocal cuja geometria e relação de redução de áreas aumenta consideravelmente essa resultante. Espera-se com essas medidas atender o requisito de velocidade mínima, embora não tenha sido realizada uma simulação dinâmica da estrutura final do veículo para estimar seu arrasto quando submerso nem o arrasto devido aos cabos de alimentação.

5.2.4 Coluna de água

O projeto do veículo leva em consideração a capacidade de cumprir suas funcionalidades enquanto submerso. A estrutura é composta predominantemente por alumínio, os elementos de fixação por aço inoxidável, os rodízios por plástico e o sistema de flutuação e o envoltório dos componentes eletrônicos por PVC de modo que nenhuma peça enferruje com o uso. Além disso, as bombas de porão trabalham normalmente quando submersas, de acordo com o fabricante.

Outro aspecto do projeto, devido à profundidade de trabalho, é o comprimento dos cabos de alimentação que ligam o veículo à fonte externa. Para impedir que esses cabos causem muito arrasto ou se enrolem durante a trajetória do carro, eles possuem flutuadores para mantê-los no nível da água. Além disso, são acompanhados de uma corda de nylon que permite ao usuário puxar o robô de volta para a superfície.

Um ponto crítico do trabalho em baixo da água é o uso de fins de curso como sensores principais. Os botões, quando submersos, permitem uma pequena passagem de corrente mesmo quando não acionados, o que pode ser preocupante quanto à segurança do equipamento. No item 6.2.1, serão apresentados resultados de ensaios com os botões submersos levando em conta a vedação aplicada nos contatos para minimizar efeitos nocivos.

5.2.5 Sensoriamento

Muitas formas de guiar o veículo através do ambiente de trabalho do tanque de provas podem ser desenvolvidas e colocadas em prática. Dentre as soluções estudadas

há opções robustas, que buscam minimizar o tempo de cumprimento da atividade enquanto garantem que toda a superfície será coberta [5]. Mas em tal caso, além de não se esperar a presença de obstáculos no ambiente, é utilizada a técnica de sensoramento por ultrassom, que foi considerada de demasiada complexidade para o dado projeto. Existem análises comparativas sobre algoritmos de varredura para veículos cujo objetivo é percorrer uma área sem direção previamente definida [6], mas vale mencionar, novamente, que o presente problema é muito particular e inclui tarefas não usuais, decidindo-se, portanto, pelo desenvolvimento de uma solução exclusiva que fosse simples e cumprisse as tarefas propostas.

A detecção de obstáculos será realizada por sensores de fim de curso. Com o contato, um sinal permitirá ao sistema de controle variar o movimento e desviar de objetos instalados no TPN assim como reverter o sentido do deslocamento após o encontro com as paredes. Além dos botões, mecanismos que serão melhor detalhados no item 6.1.3 garantirão a detecção dos obstáculos que se choquem com a maior parte da superfície lateral do veículo.

O sistema de sensores também prevê a diferenciação dos tipos de obstáculos, para isso cada parede é dotada de fins de curso afastados verticalmente. Os sensores inferiores só deverão detectar os pinos, os intermediários perceberão a cremalheira, enquanto os superiores detectarão as paredes e o ponteiro central. Para garantir que nunca haverá duas faixas de sensores ativos simultaneamente, o sistema de fixação dos botões possui um diferente espaçamento com relação à parede do veículo conforme mostrado na Figura 5-5 e Figura 5-6.

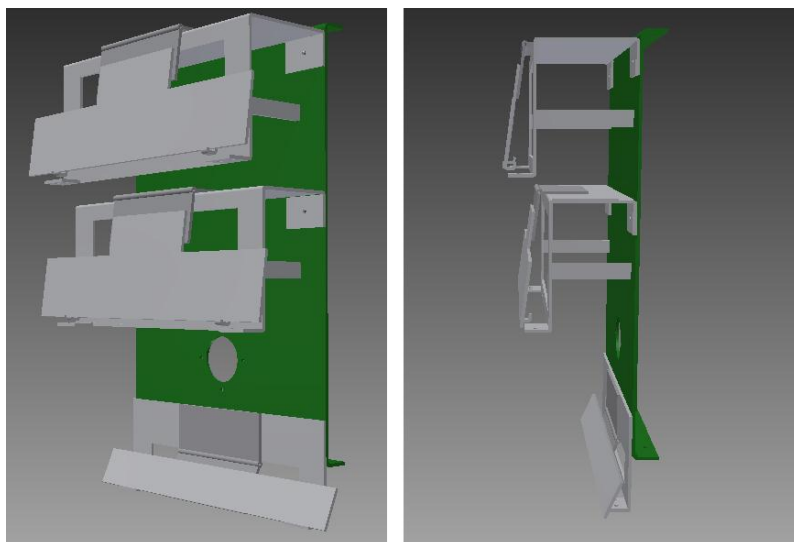


Figura 5-5 Modelo da base dos sensores na parede do veículo



Figura 5-6 Foto de parede com sensores e bomba montados

5.2.6 Peso

O robô é provido de tubos de PVC vedados e preenchidos com ar cujo objetivo é diminuir o peso submerso do robô. O volume de ar a ser embarcado foi calibrado de acordo com a massa total do veículo, a força resultante das três bombas responsáveis pelo salto e o empuxo a ser inserido no sistema. O objetivo é facilitar a tarefa de saltar obstáculos e melhor cumprir o requisito de peso submerso. Na Figura 5-7 é ilustrado esse sistema de dutos que está fixado por baixo da chapa de alumínio que constitui o topo do veículo. A Figura 5-8 mostra uma foto do sistema montado.

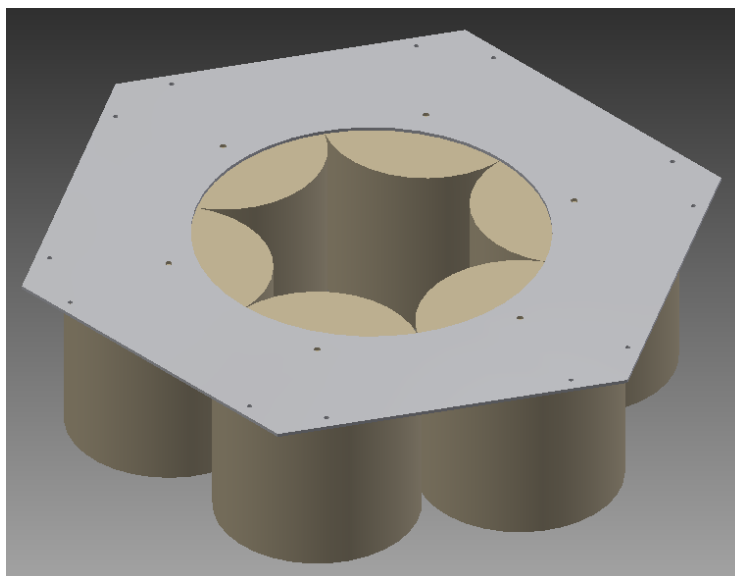


Figura 5-7 Modelo do sistema para redução do peso submerso



Figura 5-8 Foto do sistema de flutuação montado no topo do veículo

5.2.7 Durabilidade

A vida útil do veículo é limitada pela vida útil de seus componentes sendo preciso analisá-los separadamente:

Os materiais empregados na estrutura são, conforme discutido anteriormente, livres de oxidação quando em contato com a água, garantindo uma durabilidade teoricamente sem limites.

As bombas de porão de 1000GPH possuem estimativa de vida de 1500 horas de uso de acordo com o fabricante. Considerando que o ciclo de limpeza é de 3 horas e que

cada bomba permanece ligada por metade desse tempo, concluímos que elas não falharão com menos de 1000 usos, superando a expectativa de dois anos requerida.

Por último, a vida útil dos componentes eletrônicos é difícil de mensurar, porém cuidados serão tomados no sentido de impedir o aquecimento de transistores e a passagem de corrente além da máxima suportada. Também serão respeitados os limites de corrente por pino de entrada ou saída do microcontrolador usado para implementar a lógica de controle do veículo.

5.2.8 Facilidade de fabricação

A geometria das peças foi designada de modo a facilitar a fabricação limitando-se a realização de cortes, furos e dobras em 90°. Os procedimentos de maior dificuldade são cortes internos a chapas (necessários nos mecanismos de suporte dos sensores central e superior) e cones internos (caso dos bocais acoplados às bombas). Para os cortes foram necessários furos auxiliares nos vértices das geometrias desejadas e o uso de uma serra elétrica adequada, já para a realização dos cones foi preciso torner o interior de um cilindro maciço de material.

Outro ponto que facilita a fabricação é o fato da fixação das peças se basear em parafusos, que são fáceis de manusear, dispensando soldar ou rebitar componentes, que exigem aparato específico para montagem e impedem sua posterior desconstrução livre.

5.2.9 Facilidade de manutenção

Para facilitar a manutenção do veículo no caso de defeito dos botões, bombas ou componentes estruturais e eletrônicos, a montagem das peças é feita com parafusos, como dito no item acima. A possibilidade de desmontar qualquer componente permite sua substituição sem comprometer o módulo inteiro.

Outro aspecto relevante para manutenção é a facilidade de acesso às peças. Nesse sentido os botões são de mais fácil acesso, é possível retirá-los sem a retirada de nenhuma outra parte do veículo. Realizar substituições na eletrônica também é simples, bastando destampar o invólucro, puxar a placa com o circuito e desencaiar cabos ou componentes desejados. Para acessar as bombas, no entanto, é preciso retirar o topo do veículo, a braçadeira que prende a bomba no lugar e o acoplamento que une sua saída a uma das paredes ou base, dificultando o processo.

5.2.10 Segurança

A preocupação com segurança no projeto está focada principalmente no isolamento dos componentes eletrônicos. Já foi mencionado anteriormente o encapsulamento dos circuitos com PVC e o uso de vedantes nos contatos dos fins de curso. Além disso, fusíveis estão instalados para evitar problemas na fonte devido a picos de corrente ao ativar as bombas ou possíveis falhas que produzam curto-circuito.

6. Projeto Executivo

Neste capítulo será detalhado o processo de dimensionamento da solução concebida.

Após a compreensão do problema de engenharia a ser solucionado foram realizados ensaios, cálculos e esquemas apresentados ao longo dessa seção, que foram críticos para o projeto.

6.1 Projeto mecânico

6.1.1 Movimentação

Tomando como base limpadores de piscinas existentes no mercado, revisados na seção 2, é possível notar que o sistema de movimentação é geralmente constituído de rodas ou esteiras motorizadas. Alguns modelos se utilizam da força gerada por bombas hidráulicas para manterem-se em contato com o plano que desejam percorrer, mas não para se deslocarem paralelamente a tal plano. Porém, a necessidade de saltar um obstáculo vazado é um requisito que o veículo proposto por este projeto não compartilha com os robôs limpadores de piscina revisados.

Movimentar-se horizontalmente sem ter contato com o fundo do tanque é o principal motivo do uso de bombas de porão na locomoção. Além disso, elas são de pequeno porte e facilmente embarcáveis no sistema.

A fim de validar características fornecidas pelo fabricante das bombas, foram adquiridas duas de modelos diferentes. Foi possível calcular a força exercida por elas em operação nominal e em seguida foram realizados ensaios que permitiram extrapolar o comportamento das bombas com diferentes tensões de alimentação e com diferentes perdas de carga acopladas a suas saídas. Tais experimentos são relevantes quando se pretende explorar alternativas de aumentar a resultante exercida sobre o veículo durante a movimentação horizontal, para melhor atender ao requisito de velocidade mínima, e durante o salto, para aumentar a tolerância quanto ao peso submerso do robô.

Características teóricas das bombas adquiridas são apresentadas a seguir na Tabela 6-1. Corrente e diâmetro de saída foram retirados diretamente do site, enquanto que vazão nominal, área de saída e força exercida foram calculados.

Tabela 6-1 Características teóricas das bombas de porão adquiridas

Modelo GPH	Corrente com 13.6V (A)	Diâmetro de saída (polegadas)	Vazão nominal (m ³ /s)	Área de saída (m ²)	Força com 13.6V (N)
500	2.5	¾	5.26 * 10 ⁻⁴	2.9 * 10 ⁻⁴	0.970
1000	4.0	1-1/8	1.05 * 10 ⁻³	6.4 * 10 ⁻⁴	1.724

Sendo que a força exercida pela bomba é dada pela seguinte equação:

$$F_{bomba} = \frac{\rho_{\text{água}} * Q_{bomba}^2}{A_{saída}} \quad (6-1)$$

O experimento realizado para calcular na prática a força da bomba é ilustrado na Figura 6-1. À esquerda está um esquema do experimento realizado e à direita o diagrama de esforços que aproxima a situação real.

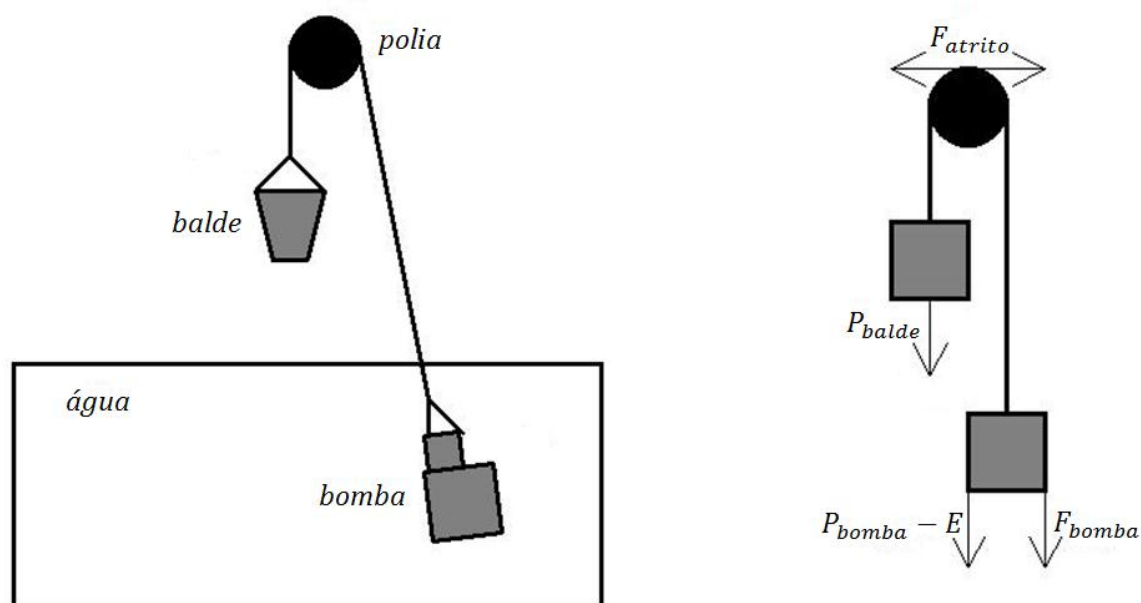


Figura 6-1 Esquema de experimento para levantar força das bombas

Tendo a bomba em mãos é possível obter sua massa e volume. Com isso, é possível calcular o peso e o empuxo da bomba cujas fórmulas são respectivamente $P_{bomba} = m_{bomba} * g$ e $E = \rho_{água} * V_{bomba} * g$, obtendo o peso submerso ($P_{bomba} - E$).

Para estimar o atrito inerente à montagem do experimento o primeiro cenário de teste foi realizado com a bomba submersa e desligada ($F_{bomba} = 0$). O balde foi preenchido com água até que estivesse na iminência do equilíbrio. Nessa situação a força de atrito é a diferença entre o peso do balde e o peso submerso da bomba.

Com o atrito é possível realizar um segundo cenário de testes, sendo feito o mesmo procedimento com a bomba ligada em sua tensão nominal de 13.5 V de modo que se possa obter a força exercida por ela.

É possível ainda realizar um último cenário em que a tensão de alimentação da bomba é tal que a corrente fornecida é inferior à máxima suportada podendo-se extrapolar a força exercida com diferentes tensões sobre a bomba.

Os resultados são apresentados a seguir na Tabela 6-2 tendo em vista que no cenário 1 o atrito aponta na direção do balde enquanto que nos cenários 2 e 3 aponta no sentido da bomba.

Tabela 6-2 Resultados do experimento para levantar força das bombas

	Cenário 1 – Desligada	Cenário 2 – Ligada em 13.5V	Cenário 3 – Ligada em 21.0V
Peso submerso da bomba	1.2 N	1.2 N	1.2 N
Peso do balde	0.7 N	2.7 N	4.3 N
Força de atrito	0.5 N	0.5 N	0.5 N
Força da bomba	0.0 N	1.0 N	2.6 N

É validado assim o fato de que a bomba exerce força de 1N na operação nominal, mas pode variar até 2.6N de acordo com a corrente fornecida.

Finalizado esse experimento, deu-se início o levantamento da curva de carga por vazão da bomba em sua operação nominal. A relação entre essas duas grandezas é

importante, nesse caso, para o posterior dimensionamento dos bocais que estarão acoplados às saídas de água e terão seu projeto detalhado na seção 6.1.2.

Para levantar a curva de carga por vazão volumétrica da bomba, bastaria obter em laboratório a carga máxima recalcada por ela (ponto em que a vazão é nula) e aproximar o formato da função por uma reta que passe por esse ponto e pelo de vazão nominal. Porém, a coluna de água que a bomba suporta com 13.5V é superior à altura do laboratório onde o ensaio foi realizado. Por esse motivo foi obtida primeiramente a curva de carga máxima por tensão, mostrada no Gráfico 6-1, através de regressão quadrática dos pontos encontrados no ensaio. Em seguida o valor assumido na tensão de alimentação foi interpolado e então a relação entre carga e vazão pôde ser aproximada conforme Gráfico 6-2.

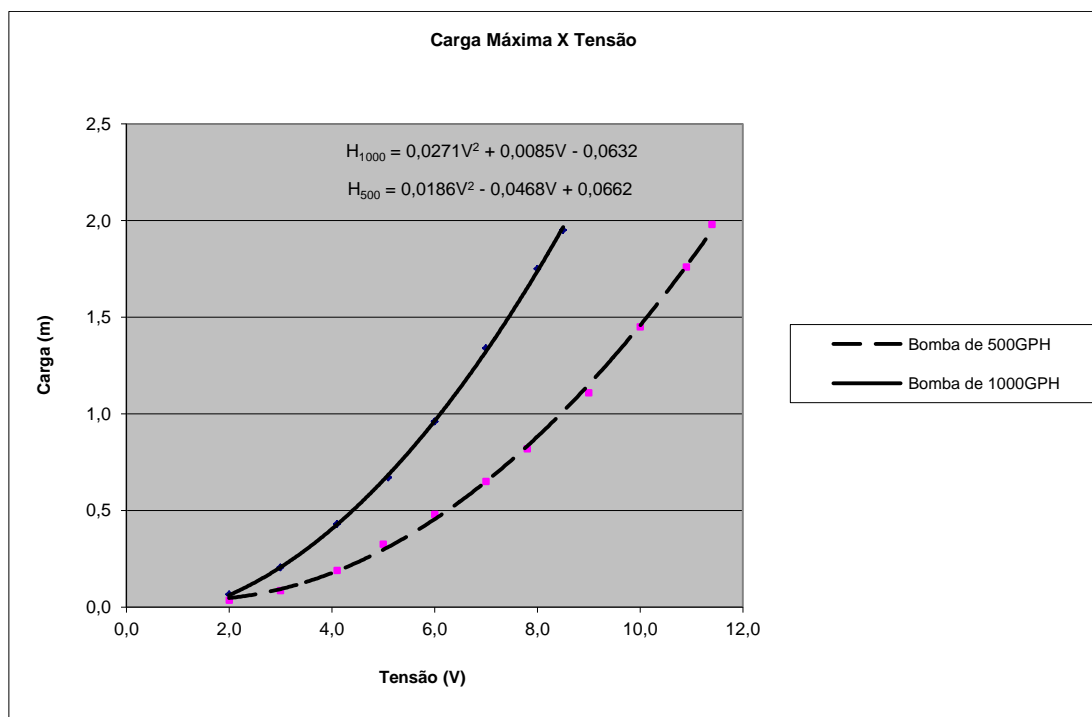


Gráfico 6-1 Curvas de carga máxima por tensão das bombas de 500GPH e 1000GPH

Pela linha de tendência do gráfico $H_{1000}(13.5) = 4.99 \text{ m}$ e $H_{500}(13.5) = 2.82 \text{ m}$.

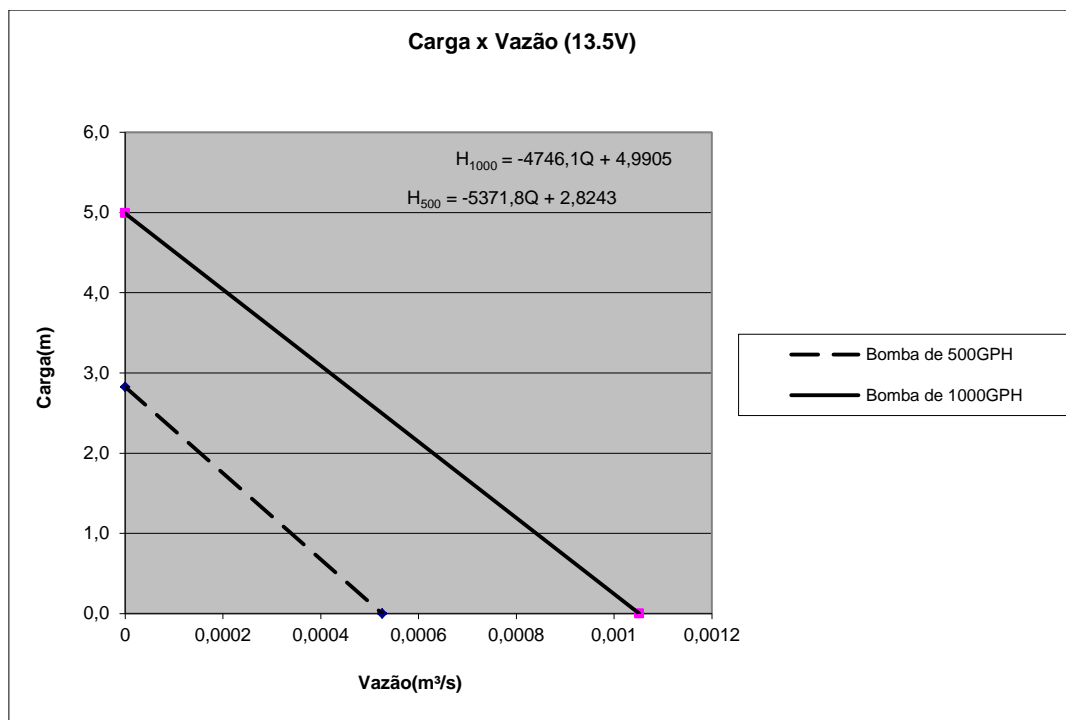


Gráfico 6-2 Curvas de carga por vazão a 13.5V das bombas de 500GPH e 1000GPH

A aproximação pela reta está sempre abaixo do valor real devido ao formato côncavo da curva de carga por vazão de uma bomba hidráulica [7]. Isso garante que os valores de vazão usados para dimensionamento dos bocais serão menores do que os reais e, por consequência, a força teórica exercida pelas bombas após o acoplamento dos redutores será menor que na prática, dando uma margem de segurança ao problema.

6.1.2 Bocal

Como discutido acima, a força exercida pela bomba varia proporcionalmente ao quadrado da vazão de água e ao inverso da área de saída da bomba. A inserção de um bocal no sistema representa simultaneamente uma diminuição da área e da vazão. Obter a melhor solução para a geometria do bocal de modo a maximizar a força é um problema de otimização cujas variáveis de projeto são as dimensões desse redutor.

A perda de carga decorrente da instalação de um bocal constritor é dada por:

$$\Delta H = (K_1 + K_2 + K_3) * \frac{\bar{v}^2}{2g} \quad (6-2)$$

Onde $\bar{v} = \text{MAX}(v_{\text{entrada}}, v_{\text{saída}})$.

Os coeficientes de perda de carga K_i referem-se a diferentes aspectos que causam perdas. K_1 depende do ângulo θ e da razão entre as áreas de saída e entrada [8] como

mostrado na Tabela 6-3. K_2 é um coeficiente devido à saída de canalização e vale sempre 1 independente de quão abrupta é a redução. K_3 deve-se à rugosidade relativa da parede interna do bocal.

Tabela 6-3 Valores do coeficiente de perda de carga em bocais

$K_{\text{contração}}$	θ							
	A_2 / A_1	10°	$15^\circ - 40^\circ$	$50^\circ - 60^\circ$	90°	120°	150°	180°
0,50	0,05	0,05	0,06	0,12	0,18	0,24	0,26	
0,25	0,05	0,04	0,07	0,17	0,27	0,35	0,41	
0,10	0,05	0,05	0,08	0,19	0,29	0,37	0,43	

Implementou-se em Matlab um algoritmo que calcula a força exercida pela bomba em um domínio com a relação A_1/A_2 entre 0.1 e 0.5 e θ entre 10° e 180° . Para isso é preciso, para cada ponto do domínio, encontrar a intersecção entre as curvas de carga do bocal e da bomba. A vazão volumétrica no ponto de encontro entre as curvas é usada na equação (6-1) para obter a força.

O algoritmo completo em linguagem .m está na seção 10.5. A malha de valores obtida é mostrada no Gráfico 6-3.

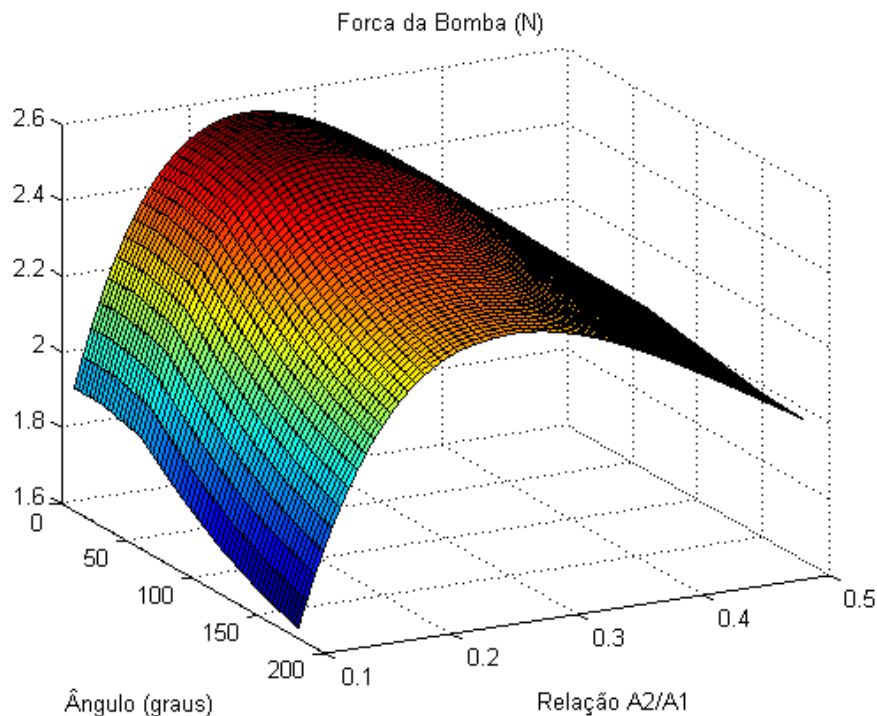


Gráfico 6-3 Força da bomba de 1000GPH em função das variáveis do problema

Há um problema adicional relacionado ao comprimento do bocal que se torna muito grande para valores pequenos de A_1/A_2 e θ o que atrapalha o posicionamento das

bombas no interior do veículo. Foi criada então uma restrição adicional de que o comprimento L fica limitado a 2 cm, sendo que L pode ser calculado utilizando a equação (6-3):

$$L = \frac{\sqrt{A_1} - \sqrt{A_2}}{\sqrt{\pi} * \tan(\theta/2)} \quad (6-3)$$

É apresentado no Gráfico 6-4 as curvas de nível da otimização com a reta de restrição. A região a direita da reta representa combinações de variáveis que satisfazem os requisitos. A partir do gráfico foi escolhido o ponto em que a relação entre as áreas de entrada e saída do bocal é 0.25 e o ângulo é 60°. Tal ponto é suficientemente próximo à restrição com geometria razoável e com força $F = 2.5N$, aproximadamente 2.5 vezes maior do que a exercida em operação nominal.

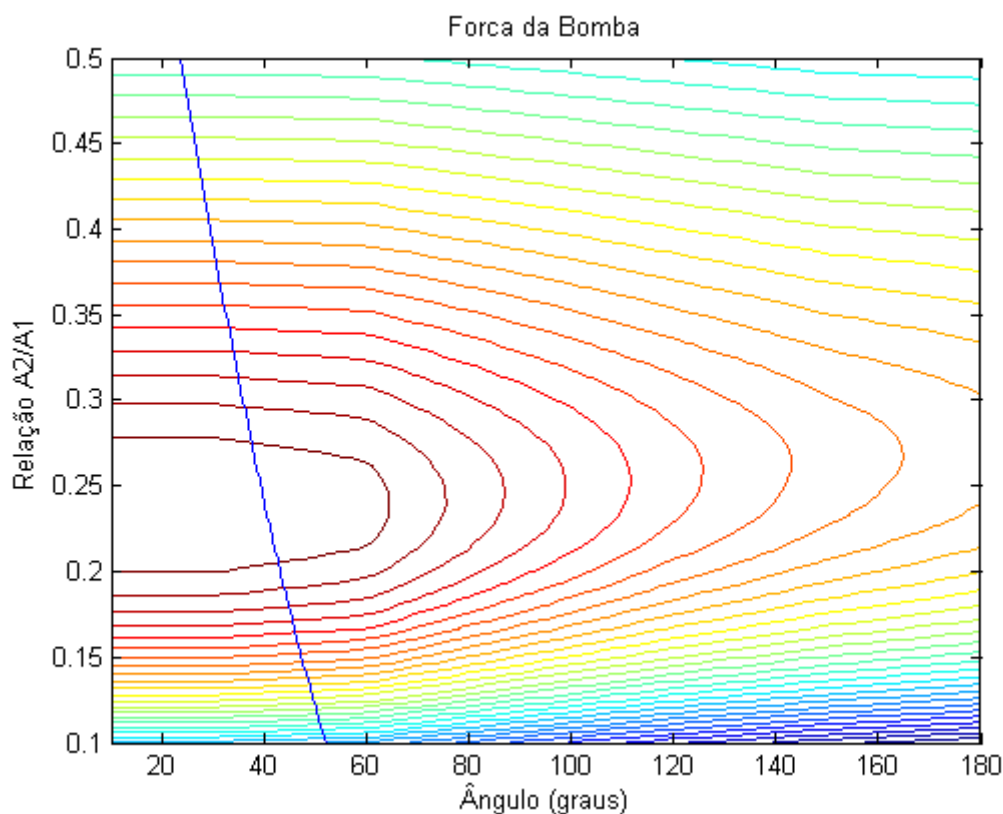


Gráfico 6-4 Resultado da otimização em curvas de nível com restrição

Finalmente, com as dimensões internas do bocal definidas, foi projetado o formato exterior para melhor acoplá-lo à bomba de porão. Há um rebaixo na seção de entrada que envolverá o cano de saída da bomba para garantir concentricidade e três furos próprios para parafusos M3 que são responsáveis por fixar o redutor à parede do veículo. O material escolhido para fabricação é PVC, como dito anteriormente no item

5.2.4. Na Figura 6-2 é mostrado um modelo visto por um corte lateral, seu desenho de fabricação encontra-se nos apêndices.

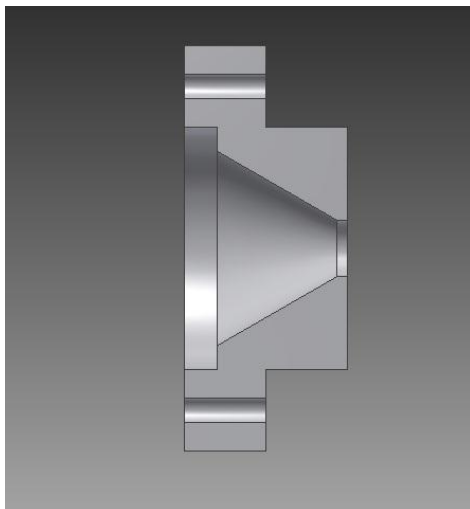


Figura 6-2 Vista lateral em corte do bocal dimensionado

Também foi projetada uma braçadeira de PVC com o intuito de impedir liberdade de movimento axial do conjunto bomba-bocal. Trata-se de um anel cujo raio interno envolve com ajuste forçado o duto de saída da bomba. Possui furos alinhados aos do bocal para inserção de parafusos e ainda um parafuso na transversal para reduzir qualquer folga. O modelo 3D dessa peça está mostrado na Figura 6-3. Uma foto do sistema pronto para montagem se encontra na Figura 6-4. Os desenhos de fabricação encontram-se nos apêndices.

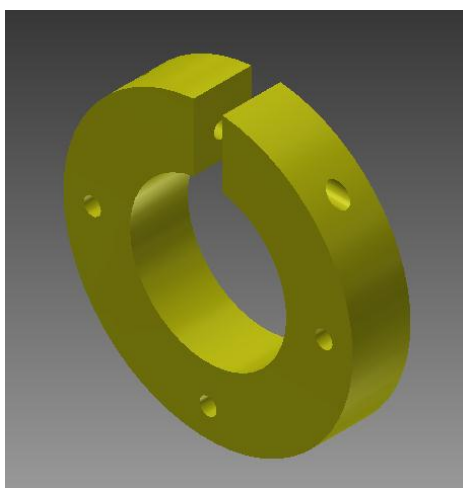


Figura 6-3 Modelo do anel para fixação do bocal



Figura 6-4 Bocal pronto para montagem em bomba de porão

6.1.3 Base dos sensores

O mecanismo para montagem dos fins de curso nas paredes foi concebido de modo a diferenciar os diversos tipos de obstáculos. Nesta seção será discutida brevemente a geometria desse conjunto de peças.

A base dos sensores inferiores é bastante simples. É composta por uma chapa dobrada em formato de L que fica com a seção vertical fixada com a parede enquanto que o fim de curso está preso na seção horizontal. Há também uma dobradiça com uma de suas abas parafusada ao veículo e a outra parafusada numa chapa com a mesma largura da parede, apoiada no botão. Um toque ao longo da superfície dessa chapa faz com que ela seja rotacionada e pressione o fim de curso.

Esse princípio de acionamento com dobradiça é utilizado no mecanismo de todos os níveis de sensoriamento e será referido de agora em diante como sistema básico.

Um modelo da base dos sensores inferiores pode ser visto na Figura 6-5.

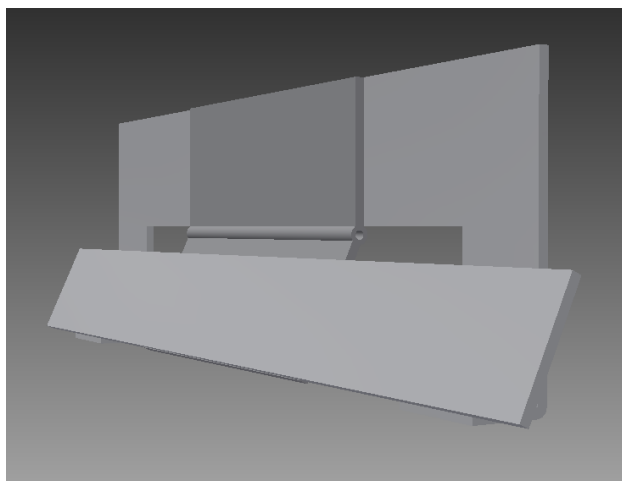


Figura 6-5 Base dos sensores inferiores

Por outro lado a base dos sensores intermediário e superior é de construção mais complicada porque não basta fixá-la diretamente à parede, é necessário impor uma distância entre ela e o fim de curso. Por isso, há uma dobra em horizontal antes do sistema básico descrito acima. O comprimento dessa dobra é tal que a cremalheira existente no fundo do tanque pressione o fim de curso intermediário antes que algum objeto pressione o inferior. Os sensores superiores são basicamente iguais, com exceção desse distanciamento horizontal, um pouco maior que o da estrutura logo abaixo.

Essas bases também são providas de dois retângulos dobrados para trás que permanecem apoiados ao veículo e impedem que a estrutura colapse durante o ciclo de limpeza com os impactos que sofrerá.

Vistas dos sensores superiores, análogos aos intermediários, são exibidas na Figura 6-6.

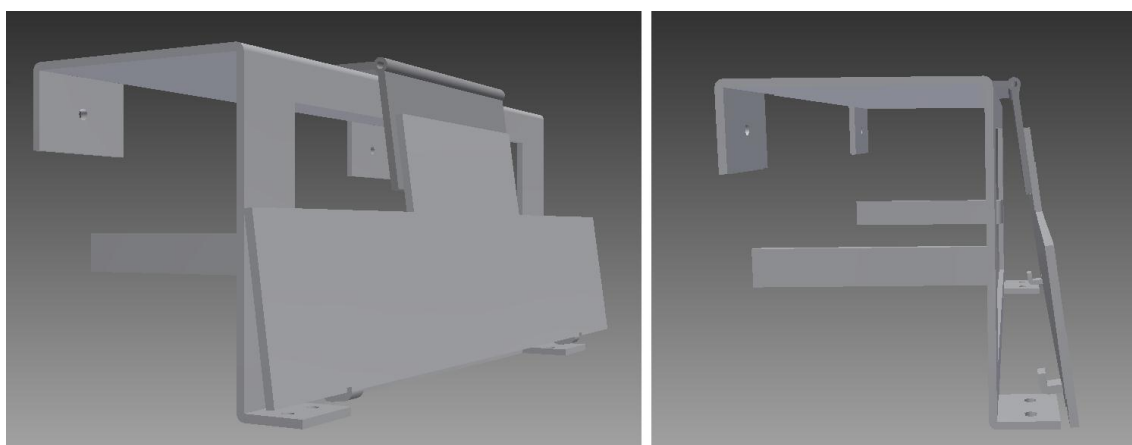


Figura 6-6 Base dos sensores superiores

6.1.4 Sistema de flutuação

Devido à geometria dos obstáculos encontrados no fundo do TPN, o veículo conta com um sistema de movimentação vertical, também constituído por bombas de porão extensivamente detalhadas nos itens anteriores do projeto mecânico. Para que o salto seja realizado com sucesso é preciso que a força resultante dos jatos de água voltados para baixo seja superior ao peso submerso do sistema.

Com o intuito de reduzir tal peso submerso, foi projetado um sistema flutuação provido de tubos de PVC preenchidos com ar. O empuxo gerado pelo volume de ar será tal que o veículo se movimente ao longo do fundo do tanque com o mínimo de atrito de rolamento nos rodízios e que o mínimo esforço seja necessário quando for desejável se deslocar para cima.

Será detalhado, então, o dimensionamento do volume de ar necessário para que, com as bombas verticais desligadas, o peso do veículo seja idêntico ao empuxo. Essa situação é crítica e não reflete exatamente o requisito almejado, porém é mais simples calibrar o peso submerso pela posterior inserção de massa no robô do que pela inserção de ar, que exigiria um redimensionamento dos canos empregados.

Dado que, na situação desejada:

$$\sum E = \sum P \quad (6-4)$$

$$P_{corpo} = m_{corpo} * g \quad (6-5)$$

$$E_{corpo} = V_{corpo} * \rho_{\text{água}} * g \quad (6-6)$$

$$E_{corpo} = m_{corpo} * \frac{\rho_{\text{água}}}{\rho_{corpo}} * g \quad (6-7)$$

Pode-se então reescrever a equação (6-4) isolando o empuxo do ar, que se deseja conhecer, e desconsiderando o peso referente à massa de ar. Além disso, substituindo as equações (6-5) e (6-6) na equação (6-4) obtém-se:

$$V_{ar} * \rho_{\text{água}} = \sum_{corpo} m_{corpo} * \left(1 - \frac{\rho_{\text{água}}}{\rho_{corpo}}\right) \quad (6-8)$$

O somatório considera todos os “corpos” relevantes na construção do veículo. Serão considerados o alumínio empregado na estrutura do robô, o latão usado nas dobradiças das bases dos sensores e as bombas de porão utilizadas.

Tabela 6-4 Massa dos elementos constitutivos do veículo

	Alumínio	Latão	Bombas
Massa total (g)	5720	745	2310
Densidade (g/cm ³)	2.7	8.5	1.4

Como $\rho_{\text{água}} = 1\text{g/cm}^3$ e com os dados da Tabela 6-4 conclui-se que são necessários 4920 cm³ de ar (4.92 litros) para equilibrar o peso e o empuxo do veículo, desconsiderando o efeito dos rodízios e das peças de PVC cuja contribuição na soma é duas ordens de grandeza menor que os itens mencionados acima.

O volume de ar dimensionado será inserido com a instalação de seis tubos de PVC com diâmetro interno $d_1 = 9.6\text{cm}$ destinados exclusivamente à flutuação. A altura deles será definida a seguir, a fim de obterem-se os 4.92 litros. Porém é preciso considerar o encapsulamento da eletrônica, que também consiste de um tubo com $d_2 = 7.1\text{cm}$ e altura $h_2 = 30\text{cm}$, que estará preenchido com uma porcentagem p de ar e $(1 - p)$ de componentes eletrônicos.

$$V_{\text{ar}} = 6 * \pi * \left(\frac{d_1^2}{4}\right) * h_1 + \pi * \left(\frac{d_2^2}{4}\right) * h_2 * p \quad (6-9)$$

Estimando $p = 0.3$, a altura dos dutos de flutuação será $h_1 = 10.5\text{cm}$.

O sistema de flutuação estará fixado com cola de silicone sob a chapa de alumínio que constitui o topo do veículo. Os centros dos tubos serão alinhados a furos existentes na chapa. A Figura 5-8, do capítulo do Projeto Básico, mostra os tubos montados no veículo.

6.1.5 Encapsulamento da eletrônica

Para garantir a vedação dos componentes eletrônicos empregados no controle do veículo, foi utilizado, conforme citado no item acima, um tubo de PVC. O sistema é lacrado e dispõe de orifícios para passagem dos cabos de alimentação e sensoramento. A vedação de tais cabos foi feita através de prensa-cabos, mecanismo comumente utilizado para isolar componentes eletrônicos de líquidos ou gases presentes no meio externo.

Tal sistema de encapsulamento deve, pela necessidade de eventuais manutenções, permitir acesso à placa eletrônica acomodada em seu interior. Então, uma das tampas do tubo é retirável. A solução com uma das tampas coladas e ajuste forçado da tampa móvel provou-se simples e eficaz na vedação dos componentes de seu interior. Uma aplicação de cola de silicone sobre os prensa-cabos foi necessária para que a água não penetrasse através de seus interstícios, uma vez que tal componente não é projetado para fixar grande quantidade de cabos passantes através de seu orifício, vide Figura 6-7. Fita veda rosca também é utilizada, apenas entre o tubo e a tampa fixada por ajuste forçado para garantir vedação, uma vez que não há cola entre tais elementos. O posicionamento do recipiente da eletrônica embarcada é feito junto ao cabo de suspensão do veículo como pode ser visualizado na foto do protótipo concluído (Figura 5-3).



Figura 6-7 Foto do tubo de encapsulamento do sistema eletrônico após construção

6.2 Projeto elétrico e eletrônico

6.2.1 Contatos

A solução de sensoriamento escolhida para o projeto foi a instalação de botões de fim de curso nas paredes do veículo para detectar o contato com obstáculos. No entanto seu comportamento quando submerso é preocupante devido à passagem de corrente pelo fluido e poderia causar choques durante o manuseio. Por este motivo, antes de sua implementação, foram realizados ensaios para determinar as características dos botões mergulhados em água.

Foram adquiridos dois botões, um naturalmente aberto e um naturalmente fechado, para realizar os ensaios. O circuito de provas foi composto basicamente de uma fonte de 12 V e um botão com um resistor de 1 K Ω ou 3 K Ω posto em série. Os resultados dos quatro cenários de teste obtidos são dados na Tabela 6-5.

Tabela 6-5 Resultados do experimento com botões submersos não vedados

Cenário	Tipo de botão	Resistor em série	Tensão no botão	Tensão no resistor	Corrente	Resistência do botão
1	NA	1 K Ω	11 V	1 V	1.00 mA	11 K Ω
2	NA	3 K Ω	10 V	2 V	0.67 mA	15 K Ω
3	NF	1 K Ω	10 V	2 V	2.00 mA	5 K Ω
4	NF	3 K Ω	8 V	4 V	1.33 mA	6 K Ω

Conclui-se que a resistência do botão quando submerso varia de acordo com o resistor que esteja em série, proporcionalmente ao módulo da resistência deste. Também varia de acordo com características específicas de cada botão. Outro aspecto relevante é que o botão naturalmente fechado irá transmitir corrente continuamente e os valores exibidos na tabela são os da situação em que o botão é pressionado, tal desperdício de potência justifica o uso apenas de botões naturalmente abertos no projeto final.

Durante a realização do teste notou-se o seguinte comportamento: um botão que esteve submerso continua dispensando corrente enquanto aberto por alguns instantes, mesmo fora da água. Essa dispersão é bastante amenizada após serem secadas as pontas metálicas localizadas na traseira dos contatos adquiridos. Com base nessa observação foi realizado um novo experimento no qual foi empregada cola quente envolvendo a conexão entre os fios e o botão. Os resultados são exibidos na Tabela 6-6. Desta vez consideram-se apenas os botões NA.

Tabela 6-6 Resultados do experimento com botões submersos vedados

Cenário	Tipo de botão	Resistor em série	Tensão no botão	Tensão no resistor	Corrente	Resistência do botão
1	NA	1 K Ω	12 V	0 V	0.0 mA	∞
2	NA	3 K Ω	12 V	0 V	0.0 mA	∞

Conclui-se que, com vedação apropriada, é possível eliminar a dissipação indesejada de corrente nos contatos submersos dado que a tensão não caiu sequer na ordem dos 0.01V. Tal resultado, no entanto, ainda é dependente do processo intrínseco de fabricação de cada botão. Quaisquer anomalias serão verificadas e discutidas na seção 7, que trata dos resultados dos testes com o veículo.

6.2.2 Circuito de comando

O circuito que controla o veículo, desde a aquisição dos sinais provindos dos botões até o acionamento das bombas de porão, pode ser dividido em duas partes: um circuito de comando e um circuito de potência. Nesta seção será detalhado o projeto do primeiro deles.

A princípio, há um circuito auxiliar que centraliza as entradas dos sensores das paredes e as transmite ao Arduino. Os botões e os respectivos resistores postos em série serão alimentados com 0 V e 5 V. Em seguida, há o circuito que envia os sinais de controle aos relês que alternarão a situação das bombas de porão.

Antes de prosseguir com a análise do circuito será descrita a lógica de agrupamento dos botões de forma que os sinais que regem o sensoramento de cada parede sejam reduzidos a dois ao invés de seis.

Há três conjuntos de sensores por parede, providos de dois contatos cada. O sinal de um deles é, portanto, redundante e pode ser conectado em paralelo a seu vizinho resultando em um sinal para o trecho inferior, um para o intermediário e um para o superior da lateral do veículo. Analisando esses três sinais logicamente, serão redundantes aqueles que resultem na mesma transição de estado partindo de uma mesma situação inicial. A Tabela 6-7 mostra os estados e transições de acordo com entradas dos sensores (desconsiderando transições ativadas devido a temporizadores e estados alcançados apenas por eventos temporizados).

Tabela 6-7 Transições de estado de acordo com sensores ativados

	Ciclo de limpeza		Varredura antes do pulo	
Entrada	Ação	Próx. estado	Ação	Próx. estado
Sinal do sensor inferior	Desviar	Ciclo de Limpeza	Desviar	Varredura antes do pulo
Sinal do sensor intermediário	Desviar	Ciclo de limpeza	Pular	Ciclo de limpeza
Sinal do sensor superior	Desviar	Ciclo de limpeza	Desviar	Varredura antes do pulo

Nota-se que as tomadas de decisão devidas à ativação dos sensores inferior e superior são idênticas. Os respectivos sinais são, portanto, considerados redundantes e ligados em paralelo.

O circuito de comando é ilustrado a seguir. Na Figura 6-8 o conector J7 liga o GND do Arduino aos doze conjuntos de sensores (dois por parede). Os sinais dos botões retornam à placa pelo conector J8 e são amostrados por seis pinos de entrada analógica e seis pinos de entrada digital. Não serão usadas todas as entradas digitais para amostragem, pois ainda são necessários alguns pinos reservados para saídas.

A Figura 6-9 mostra o circuito de comando montado com o Arduino. O flat cable indicado pelo número 1 recebe os sinais vindos dos sensores nas paredes enquanto o flat cable 2 transfere as saídas do Arduino para a placa com o circuito de acionamento. Os conectores de número 3 acoplam a placa auxiliar de resistências ao microcontrolador.

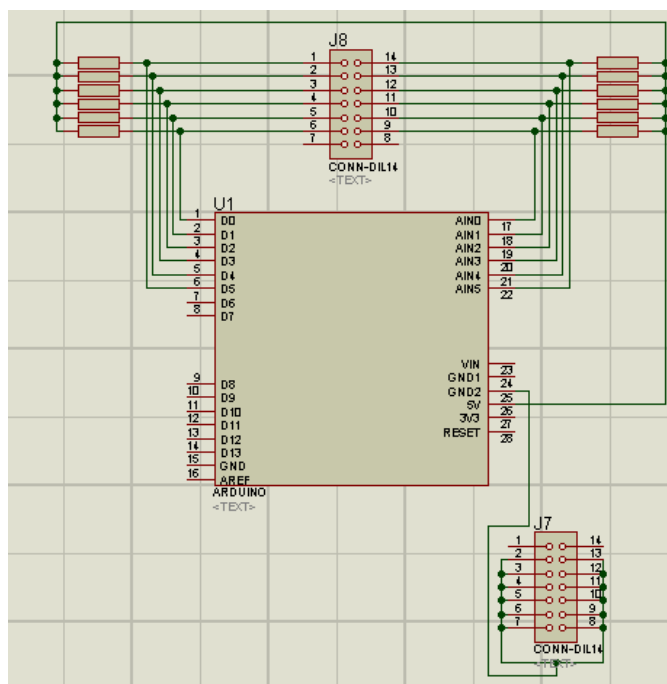


Figura 6-8 Esquema do circuito de comando (sinais de sensoriamento)

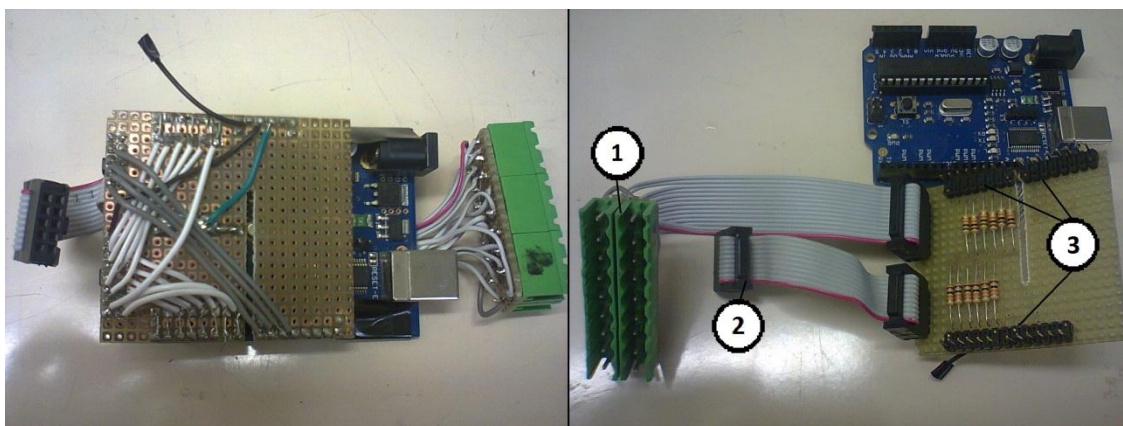


Figura 6-9 Circuito de comando montado

O circuito de acionamento, por sua vez, é composto pelos componentes esquematizados na Figura 6-10. O conector J9 liga as saídas do Arduino às bases de transistores que, quando saturados, acionam a bobina de relês que por sua vez permitem a passagem de corrente para as bombas que estarão ligadas nos conectores J1 à J6. Os três relês mostrados na parte inferior do esquema (RL23, RL24 e RL25) possuem acionamento independente, pois são responsáveis pelas bombas de movimentação horizontal. Ativando-as uma por vez ou em pares é possível se movimentar em seis sentidos distintos no plano, como já foi discutido anteriormente e ilustrado na Figura 5-4. Já os três relês na parte superior do esquema (RL20, RL21 e RL22) compartilham o mesmo sinal de acionamento, pois é interessante que as três bombas que realizam a movimentação vertical trabalhem em conjunto sempre.

Os diodos colocados em paralelo às bobinas dos relês servem para dissipar o fluxo de corrente gerado no indutor quando ocorre seu desligamento.

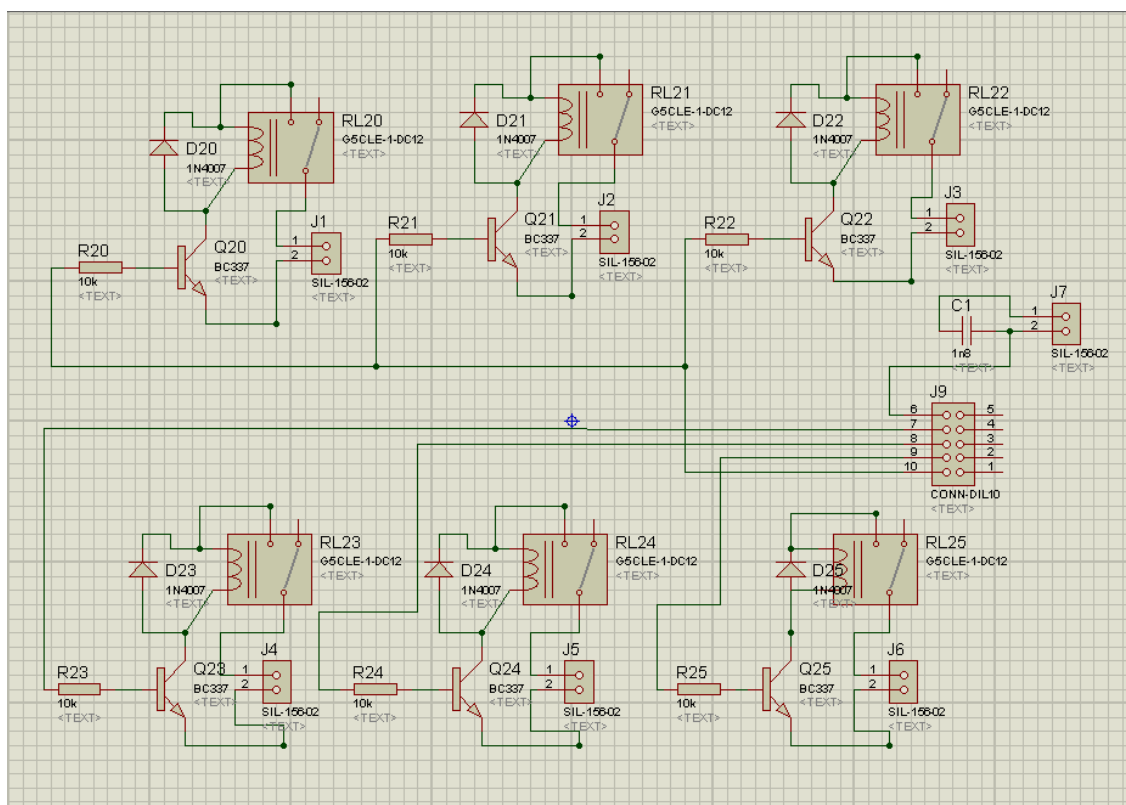


Figura 6-10 Esquema do circuito de comando (sinais de acionamento)

A Figura 6-11 mostra, à direita, o projeto de trilha concebida com auxílio de software para impressão em placa com apenas uma das faces acobreadas, já levando em consideração o tamanho real dos componentes. À esquerda é mostrado o circuito concluído. Os conectores indicados com o número 1 se referem às bombas verticais e os de número 2 se referem às horizontais. O conector 3 recebe os sinais de acionamento e o 4 serve para alimentar o Arduino.

Empregou-se um esforço extra em compactar a disposição dos componentes para que o invólucro em que se acomodará a eletrônica não represente um volume grande de ar que de alguma forma prejudique o dimensionamento do sistema de flutuação detalhado no item 6.1.4. Conclui-se que uma área de 7 cm por 5 cm é suficiente para acomodar a esse trecho do circuito.

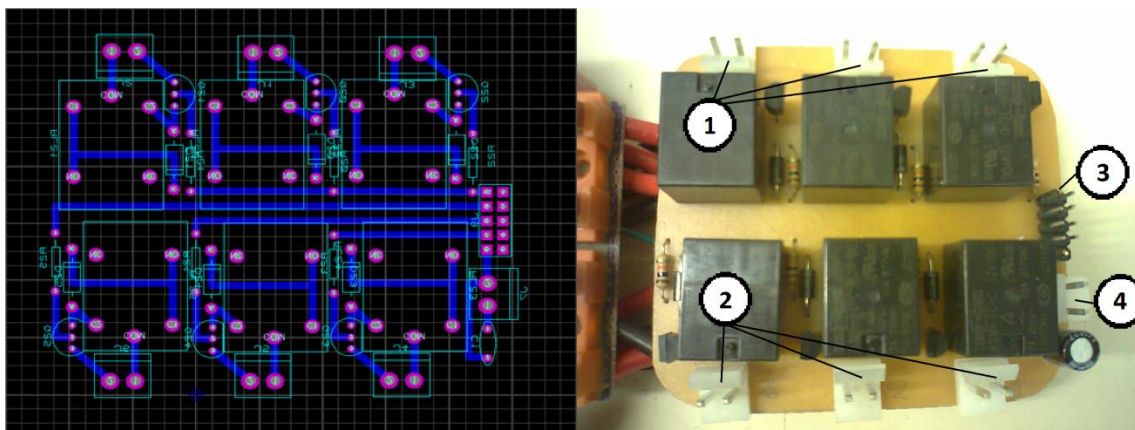


Figura 6-11 Projeto da trilha para o circuito de acionamento (à esquerda) e circuito concluído (à direita)

6.2.3 Circuito de potência

O circuito de potência do robô é o responsável pela alimentação das bombas. Ele é composto por uma fonte de 12 V com capacidade de fornecer até 15A, corrente solicitada para o acionamento do sistema de movimentação vertical no instante de pico.

Da fonte partem dois cabos providos de pequenas boias que o impedem de estar completamente submerso. Considera-se que, no caso crítico em que o veículo se encontra em um extremo do tanque e a fonte está no vértice oposto, o cabo percorra a diagonal da superfície da água e em seguida a distância até o fundo. O comprimento mínimo do cabo na situação crítica será de 23.8 m. Acrescenta-se a isso alguns metros por considerar a real distância entre a fonte e a borda do reservatório, embarrigamento do cabo e por mera segurança, de modo que o cabo de alimentação terá trinta metros.

Quando os cabos alcançam o invólucro da eletrônica são utilizados para alimentar o Arduino e em seguida passam por um conjunto de fusíveis de 5A antes de fornecer corrente para os relês. O objetivo é que não ocorra a passagem de sobrecorrente no circuito, que pode danificar a fonte e os cabos.

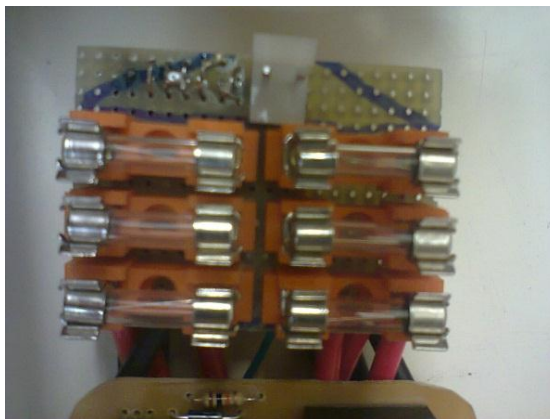


Figura 6-12 Placa auxiliar com fusíveis de 5 A

6.3 Programa embarcado

Com o intuito de controlar a trajetória do veículo conforme ocorrem interações com o ambiente, o sistema conta com um programa embarcado que decide qual rumo tomar após o contato com obstáculos. Esta seção descreve a lógica empregada no ciclo de limpeza e no sistema que alerta o controlador de eventuais irregularidades via *watchdog*.

6.3.1 Ciclo de limpeza

A princípio será definido o ciclo de limpeza de acordo com as tarefas a serem realizadas pelo veículo. É delegada ao usuário a responsabilidade de apenas fornecer energia ao sistema quando o robô estiver submerso, posicionado em um ponto arbitrário exterior à região circular delimitada no fundo do tanque. A partir do momento em que o microcontrolador é alimentado, o programa inicia o processo de controle. O veículo percorrerá a área externa ao círculo central pelo período de duas horas. Este período foi estabelecido com base no tempo total de limpeza de três horas definido no requisito 3.2 e a relação entre a área total do fundo do tanque e a área do círculo.

A solução básica de movimentação é feita seguindo o princípio de incidência e reflexão detalhado na seção 6.3.2. Para tratar da entrada do veículo na região circular delimitada no fundo do tanque, após o ciclo de limpeza externo o veículo deve se deslocar, seguindo ainda o mesmo princípio de movimentação, até o contato com a cremalheira. Esse contato é detectado devido à mudança de sinal nos sensores de nível intermediário de alguma parede.

A partir do contato, o microcontrolador envia um sinal que ativa as três bombas que movimentam o carro na vertical. O tempo que o sistema deve permanecer acionado depende da dinâmica do veículo enquanto submerso e pode ser calibrado com testes em campo. Após atingir altura adequada é preciso se locomover na horizontal no mesmo sentido em que ocorreu o contato. A locomoção dura o suficiente para que o robô adentre completamente a circunferência. Depois disso, será desativada a movimentação por um período suficiente para que o veículo toque o solo, período calibrável também em um ambiente controlado.

Dado o sucesso da manobra, o veículo passará o tempo restante do ciclo limpando o interior da circunferência utilizando o mesmo algoritmo de varredura discutido anteriormente. Após três horas de funcionamento, o robô cessará suas atividades.

6.3.2 Lógica de rebatimento

Durante a varredura o estado inicial do robô representa a movimentação horizontal em um sentido qualquer dada pelo acionamento de uma das bombas. Partindo desse estado é possível ocorrer o contato entre um obstáculo e uma das três paredes localizadas a dianteira. A Figura 6-13 ilustra esse fato.

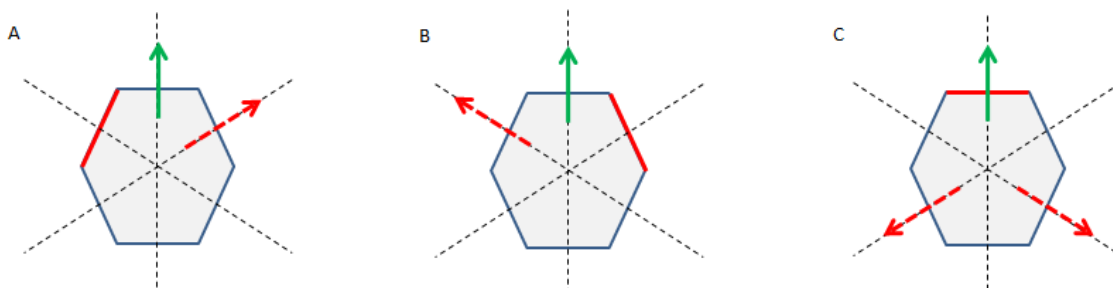


Figura 6-13 Tomadas de decisão possíveis durante varredura

Na configuração A, o veículo se move na direção da seta verde e entra em contato com um obstáculo um pouco à esquerda do movimento. No momento em que há a leitura do sinal de choque proveniente do sensor instalado nessa mesma parede, o algoritmo ativar as bombas que movem o carrinho no sentido indicado pela seta vermelha. Tal decisão ocorre independente da altura do obstáculo em questão. Tanto o ângulo de incidência quanto o ângulo de reflexão do robô serão idênticos na situação ideal, fazendo com que o comportamento do sistema se assemelhe, por exemplo, ao comportamento de um feixe de luz incidindo sobre uma superfície. A Figura 6-14 ilustra os ângulos mencionados, indicados na figura pelas áreas em azul.

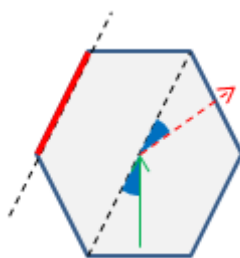


Figura 6-14 Ângulos de incidência e reflexão

Na configuração B a tomada de decisão é análoga, porém no sentido anti-horário. Já na configuração C, para que os ângulos de entrada e saída sejam semelhantes o veículo deveria retroceder, partindo no sentido oposto ao do movimento. Porém, em uma situação crítica, é possível que o veículo fique travado em uma única linha de atuação. Portanto o algoritmo escolherá entre um dos sentidos indicados em vermelho na Figura 6-13 e a ambiguidade poderá ser tratada aleatoriamente ou de acordo com o histórico de decisões tomadas pelo microcontrolador.

No caso de uma configuração de entrada diferente das citadas acima o algoritmo mantém o sentido atual de movimento, considerando que o veículo já está se afastando do obstáculo com o qual ocorreu o choque.

O fluxo de ações deste trecho do programa, responsável pela movimentação básica do veículo em seu ambiente e chamado de “lógica de rebatimento”, é descrito no diagrama de atividades UML ilustrado na Figura 6-15.

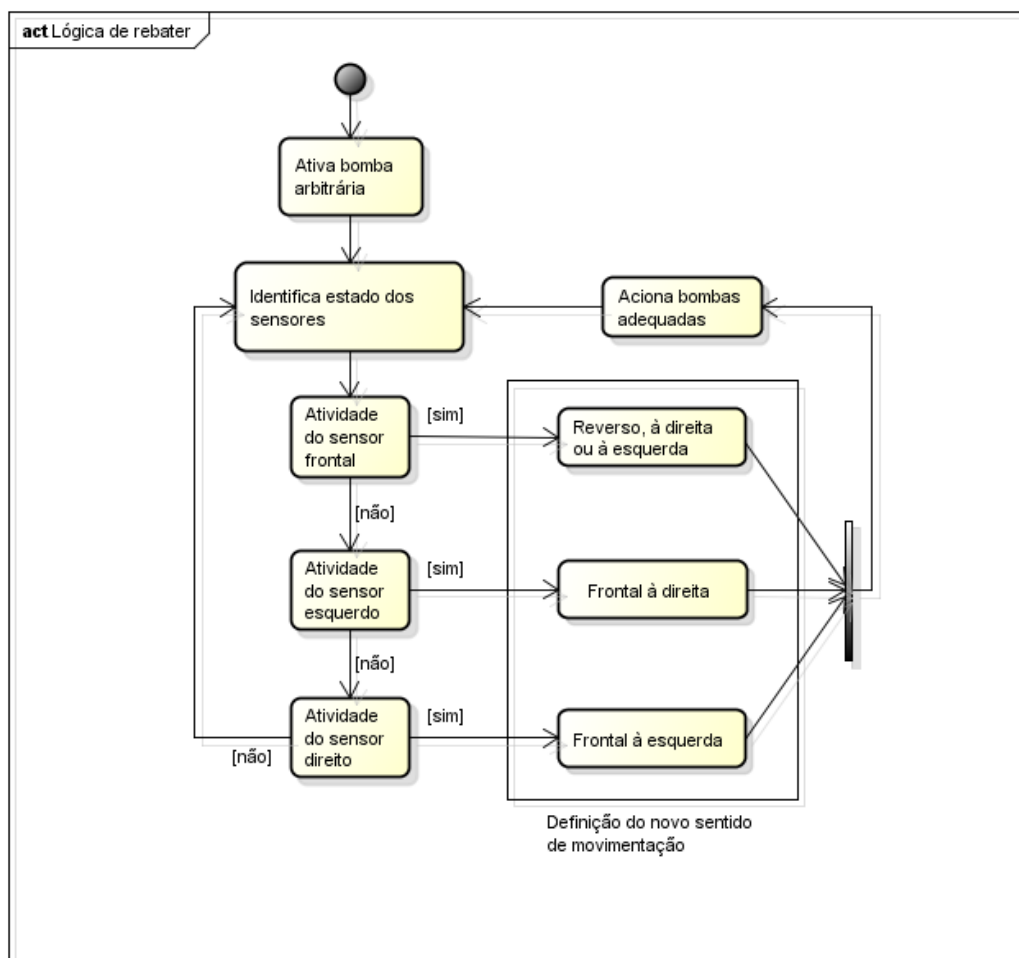


Figura 6-15 Diagrama de atividades UML

O algoritmo embarcado inclui uma interface que pode ser acessada pelo usuário ao se conectar o cabo USB embarcado no veículo a um computador e rodar o programa. É possível identificar o estado dos sensores e a sequência de decisões tomadas pelo sistema através das mensagens enviadas para a saída padrão do compilador.

6.3.3 Lógica de pulo

Depois de transcorrido o tempo do ciclo de limpeza referente à área externa a limitada pela cremalheira central do tanque, o veículo passa a buscar tal obstáculo para executar uma sequência de movimentos que constituem um “pulo” e adentrar a região interna. Nesse estado de busca pelo obstáculo, as atitudes de tomada de decisão do algoritmo são as mesmas das descritas na secção 6.3.2 Lógica de rebatimento, com diferenciação apenas para o sensor de pulo (também chamado “sensor intermediário”) frontal em relação à direção atual de movimento e os dois sensores de pulo adjacentes àquele.

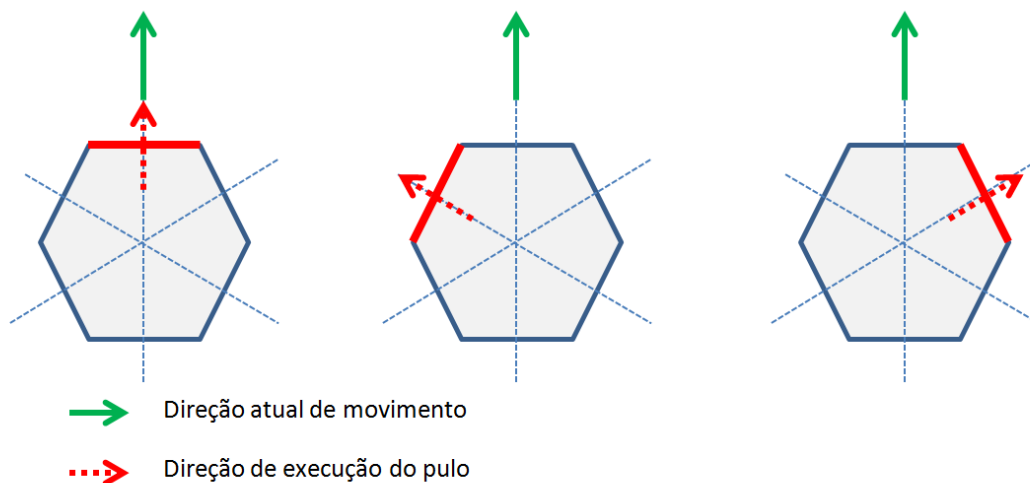


Figura 6-16 Tomadas de decisão possíveis durante o pulo

O veículo continua o procedimento de varredura até que encontre o obstáculo que irá ativar um dos sensores descritos acima e execute uma sequência de pulo na *mesma direção* do obstáculo encontrado. A sequência de movimentos do pulo é composta pelas seguintes etapas, cada uma com duração calibrada empiricamente:

Ré: O veículo impõe um pequeno espaço entre sua estrutura e o obstáculo para evitar qualquer tipo de travamento, para tanto, as bombas que implicam em movimentação no sentido oposto ao do pulo são ativadas por um breve período;

Ascensão: As bombas de movimento no eixo Z são ligadas para que o veículo deixe de tocar o chão. As bombas são então desligadas após o veículo atingir uma determinada altura com relação ao solo;

Avanço: O veículo se movimenta para frente através do acionamento das bombas responsáveis por impulsioná-lo na direção do obstáculo encontrado, passando então sobre a cremalheira;

Queda: Todas as bombas são desativadas e aguarda-se o contato do veículo com o solo para que retorne a uma situação de estabilidade;

Continuação: O veículo continua se movimentando pelo chão na mesma direção em que executou o pulo e volta a seguir a lógica de rebatimento.

6.3.4 *Watchdog* de segurança

Em funcionamento, o veículo é sujeito a situações não detectáveis pelo sistema de sensoriamento empregado. É possível que algum pino instalado no fundo do tanque enrosque na base de algum sensor, os rodízios utilizados na base do carro podem emperrar no canal de drenagem do TPN, o veículo pode saltar exatamente sobre o ponteiro localizado no centro do tanque e então, em uma dessas situações, o robô não se deslocará mesmo com o esforço realizado pelas bombas.

Com objetivo de se livrar de tais situações, o algoritmo é dotado de um trecho de código percorrido paralelamente ao ciclo de limpeza e que serve como sentinela. O princípio de funcionamento é inspirado no *watchdog* de sistemas embarcados, com o diferencial de não reiniciar efetivamente o programa dado seu acionamento.

O algoritmo temporizador é zerado a cada mudança de estado das entradas. Dessa forma, se o período entre duas mudanças de estado for superior a um tempo limite, o sistema presume primeiramente que houve um erro mecânico e que o veículo se encontra andando em círculos. A primeira atitude então é reverter o sentido de movimento. O temporizador é, então, zerado e a varredura continua normalmente.

Se o tempo limite for ultrapassado duas vezes consecutivas, o sistema presume que o veículo se encontra emperrado em algum obstáculo e realiza um salto exatamente como descrito na seção 6.3.3. Também é possível que um problema na alimentação das bombas impeça o movimento, falha que ocorre em malha aberta e o controlador, não tendo realimentação da situação dos acionadores, não detecta o fenômeno.

No caso de falha elétrica, o usuário pode ligar o veículo com o Arduino conectado a um PC via cabo USB e será enviada uma mensagem para a saída padrão do compilador identificando o sensor com erro. Já no caso de falha nas bombas, será preciso checar o estado dos fusíveis do circuito de potência.

A seguir, na Figura 6-17, é apresentado um diagrama UML descrevendo o algoritmo com varredura, salto e *watchdog* integrados.

Figura 6-17 Diagrama de atividades UML descrevendo o algoritmo

7. Resultados

Após a construção do veículo é necessário verificar que o mesmo atende aos requisitos estipulados no início do projeto a fim de se concluir se é possível ou não a implementação da solução concebida para o problema de engenharia. Nesta seção serão expostos os resultados da validação do protótipo.

7.1 Movimentação XY

O veículo se move conforme algoritmo descrito na seção 6.3 com exceção de dois fenômenos observados. Há propensão para rotação em torno do eixo vertical uma vez que esse grau de liberdade não é controlado. A rotação pode ocorrer por irregularidades no fluxo de saída das bombas, influência do cabo de alimentação e por diferenças na orientação dos rodízios quando ocorrem mudanças de sentido, pois o alinhamento dos mesmos não é instantâneo. No entanto, a movimentação do veículo não é mapeada durante a varredura mesmo quando o movimento é considerado ideal. Portanto, certa rotação em torno do eixo Z não afeta negativamente o funcionamento do protótipo.

Outro fenômeno que pode afetar a movimentação é o surgimento de ruído na leitura dos sensores, quando o microcontrolador detecta variação no nível de tensão de alguma entrada sem que haja o contato nos botões de fim de curso. A origem do ruído não foi completamente identificada, mas em testes realizados com o veículo fora da água, com e sem o acionamento das bombas, notou-se que o chaveamento dos relês pode influenciar a aquisição de dados.

Esse ruído, embora prejudicial, acontece com pouca frequência enquanto o veículo se encontra submerso e não se mostrou crítico nos testes. Eventuais problemas na movimentação XY podem ser desconsiderados caso os testes realizados no TPN indiquem que a área varrida é superior ao mínimo estabelecido no tempo delegado ao ciclo de limpeza.

7.2 Movimentação Z

Observou-se que a ultrapassagem de obstáculos por meio de um salto foi executada com sucesso, mas depende da calibração do peso submerso do veículo para que a resultante das bombas verticais seja suficiente para erguer o robô.

A calibração é realizada por meio de boias inseridas sob a estrutura da base dos sensores nas paredes. Embora esse procedimento seja dificultado pelo acumulo de bolhas de ar em diferentes partes da estrutura do veículo quando este é submerso, levando a uma calibração inicial errônea. Conforme o robô se movimenta e se choca com obstáculos, as bolhas se dispersam e o esforço requerido das bombas se torna alto demais e pode impossibilitar o pulo. Esse fato pode ser evitado se o usuário inclinar o robô após a imersão.

A sequência de imagens a seguir, indicadas como Figura 7-1, se trata de uma sequência de screenshots tirada de um dos vídeos gravados com o veículo submerso. O robô salta um obstáculo genérico e se mostra com hidrodinâmica bastante estável. Os três quadros superiores se referem ao movimento de ascensão do robô, em que ele se desloca apenas na vertical. Os quadros inferiores mostram o deslocamento horizontal até que o veículo tenha ultrapassado completamente o obstáculo. O desligamento de todas as bombas ocorre em seguida e o robô toca o fundo do tanque.

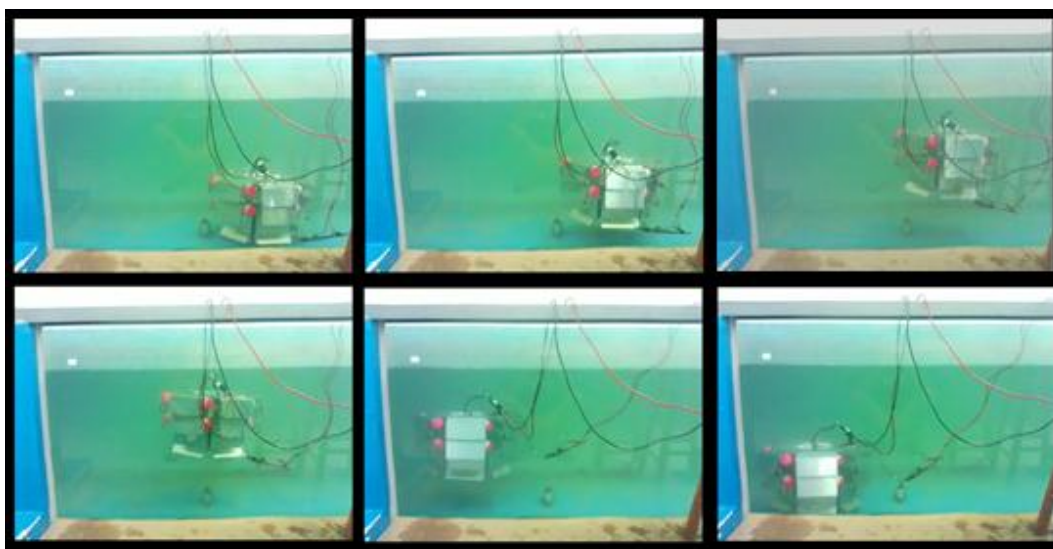


Figura 7-1 Fotos indicando sequência de salto

7.3 Durabilidade

O requisito de durabilidade é de difícil estimativa dado que diversos componentes estão em situação não usual de utilização, sendo necessário estimar o desgaste das bombas, dos contatos e eletrônica conforme o uso do veículo. É esperado que os dados dos fabricantes e os cuidados empregados na montagem do sistema sejam suficientes para garantir a vida útil dos componentes.

7.4 Requisitos qualitativos

O veículo é seguro, dado que a dissipação de corrente elétrica na água foi praticamente anulada com a vedação dos contatos utilizados no sensoramento. Além disso, o circuito de acionamento das bombas utilizadas na movimentação é provido de fusíveis que impedem uma possível sobrecorrente. E quanto à mecânica, o projeto não possui peças cortantes.

O projeto permite manutenção de praticamente todas as peças, que são desmontáveis, com exceção do sistema de flutuação. Problemas elétricos são identificáveis devido a um programa embarcado amigável, que indica o estado dos sensores, ou através da análise do circuito de fusíveis.

No entanto, há dificuldade de manutenção no sentido de complementos necessários para substituição de alguns componentes. Para trocar os fins de curso, por exemplo, é necessário dispor de cola quente para vedar os eletrodos na traseira dos botões e solda para os fios. Para acessar o programa embarcado é preciso instalar previamente os *drivers* do Arduino. Além disso, poucas peças são intercambiáveis, de modo que com exceção dos rodízios seria necessário fabricar peças novas no caso de falha catastrófica de paredes ou bases de sensores.

Outro quesito que entra na manutenção do veículo é a quantidade de parafusos e porcas utilizadas na montagem. Cada conjunto de paredes com três sensores utiliza 32 parafusos e porcas de tamanhos diversos. Dispende-se, então, um tempo razoável na substituição de qualquer peça.

8. Conclusões

Ao final do projeto, percebe-se que o problema de limpeza de um tanque de provas foi abordado de uma maneira não ortodoxa, de modo que, embora as etapas de um projeto de engenharia tenham sido seguidas (especificação de requisitos, projeto e desenvolvimento da solução, implementação da solução e realização de testes de verificação) o produto final é extremamente específico e se distancia dos modelos de limpadores de piscinas encontrados no mercado.

A metodologia empregada, descrita ao longo do texto, foi impulsionada pelos desafios encontrados durante o projeto e não considerados a princípio. Os ensaios realizados com as bombas e bocais, por exemplo, foram motivados pela identificação da pequena força exercida pelas bombas. Já os ensaios realizados com sensores de fim de curso se mostraram necessários após verificação da dissipação da corrente com os botões submersos. A busca pela solução desses e outros aspectos envolvidos no projeto (que abrange diferentes áreas da mecatrônica) possibilitou que ao final o protótipo tenha atendido a grande parte dos requisitos estipulados inicialmente.

Concluimos com base nos resultados obtidos que o veículo pode ser implementado no Tanque de Provas Numérico da USP mesmo possuindo movimentação errática. No entanto, medidas ainda podem ser tomadas a fim de avaliar o desempenho do protótipo, assim como tratar fenômenos nocivos ao deslocamento horizontal e vertical identificados nas seções 7.1 e 7.2.

Para dar sequência ao projeto é possível, por exemplo, monitorar seu comportamento durante um ciclo completo de limpeza no TPN, realizar simulações dinâmicas do veículo, realizar testes que evidenciem o comportamento estático do robô, melhorar o algoritmo de varredura e buscar uma solução para alimentação embarcada.

9. Referências bibliográficas

- [1] AQUABOT - Owner's manual. Disponível em:
<http://www.poolcenter.com/Aquabot_Manual.pdf>. Acesso em: 08 de Janeiro de 2013.
- [2] DOLPHIN - Robotic pool cleaners. Disponível em:
<<http://www.maytronics.com/robotic-pool-cleaners>>. Acesso em: 08 de Janeiro de 2013.
- [3] IROBOT Verro, pool cleaning robot. Disponível em:
<<http://www.irobot.com/en/us/robots/home/verro.aspx>>. Acesso em: 08 de Janeiro de 2013.
- [4] BILGE PUMPS. Disponível em:
<http://www.rulepumpsupply.com/Bilge_Pumps.html>. Acesso em: 08 de Janeiro de 2013.
- [5] M. SIMONCELLI, G. ZUNINO and H.I. CHRISTENSEN. Autonomous pool cleaning. CAS - Centre of Autonomous Systems, Royal Institute of Technology, Sweden, 1999.
- [6] R. BAYUK and S. RATERING. A comparison of robot navigation algorithms for an unknown goal. Computer Science Department, University of Wisconsin, Eau Claire, 2005. Disponível em:
<http://www.micsymposium.org/mics_2005/papers/paper108.pdf>. Acesso em: 08 de Janeiro de 2013.
- [7] R.F. MOREIRA e J.L. SOARES. Bombas. Departamento de Engenharia Química e Alimentos, UFSC - Universidade Federal de Santa Catarina, 2004. Disponível em:
<<http://www.enq.ufsc.br/disci/eqa5313/bombas.htm>>. Acesso em: 08 de Janeiro de 2013.
- [8] M. N. ESTANISLAUS REIS. Fenômenos de transporte. PUC - Pontifícia Universidade Católica, Minas Gerais, 2008. p.71-82.

10. Apêndices

10.1 Modelo do fundo do tanque

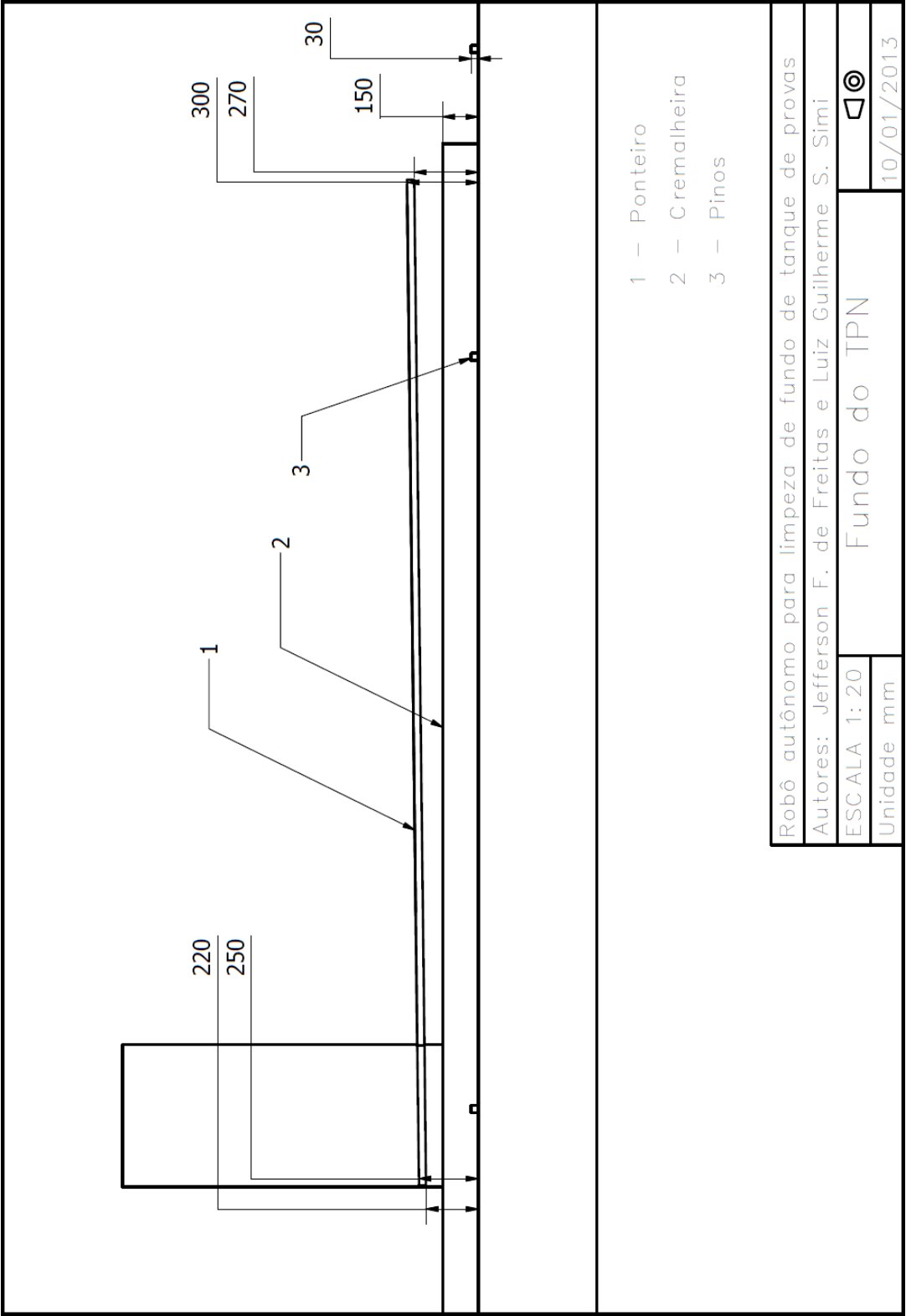


Figura 10-1 Vista lateral do fundo do tanque modelado

10.2 Desenhos de fabricação do veículo

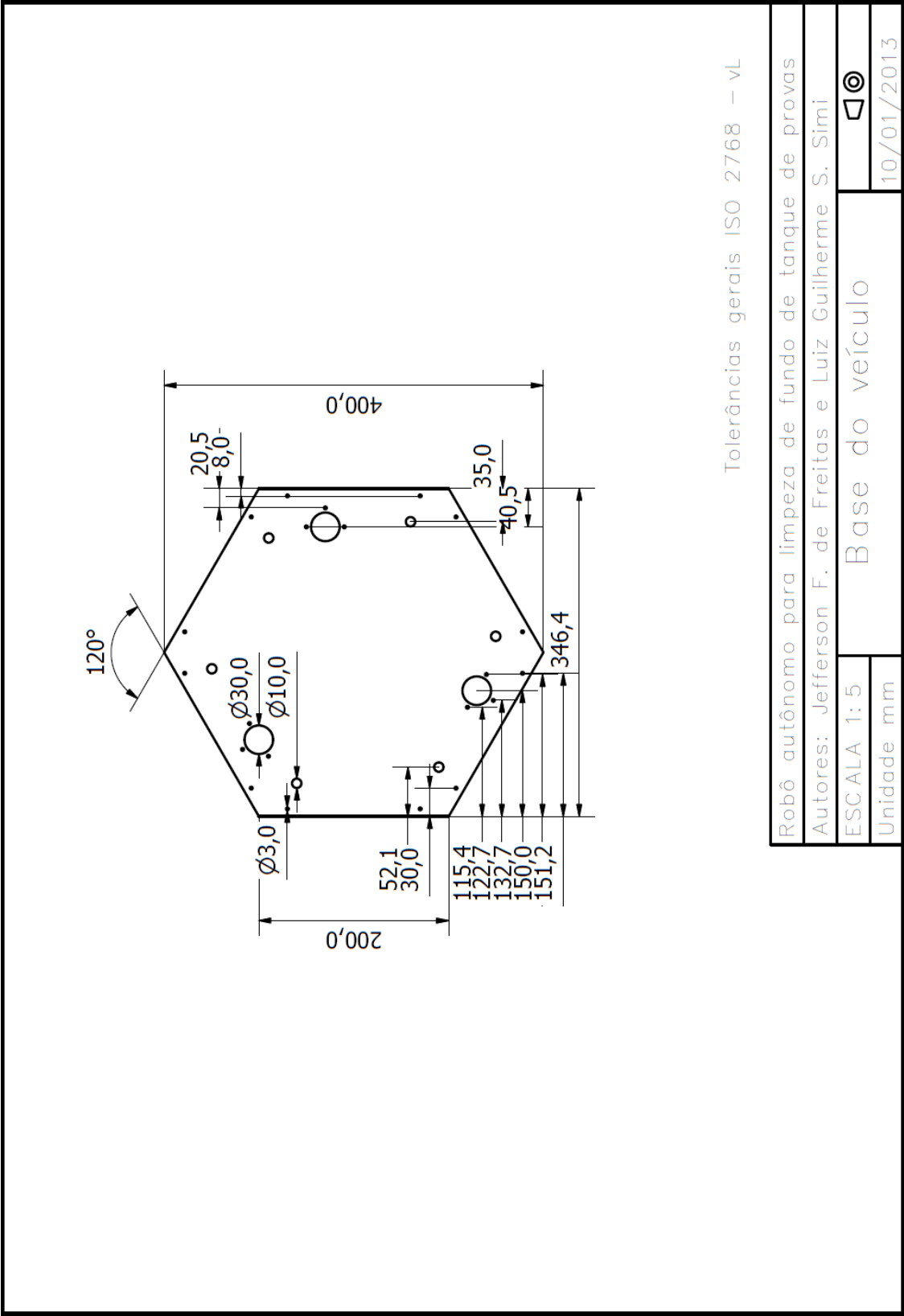


Figura 10-2 Base do veículo

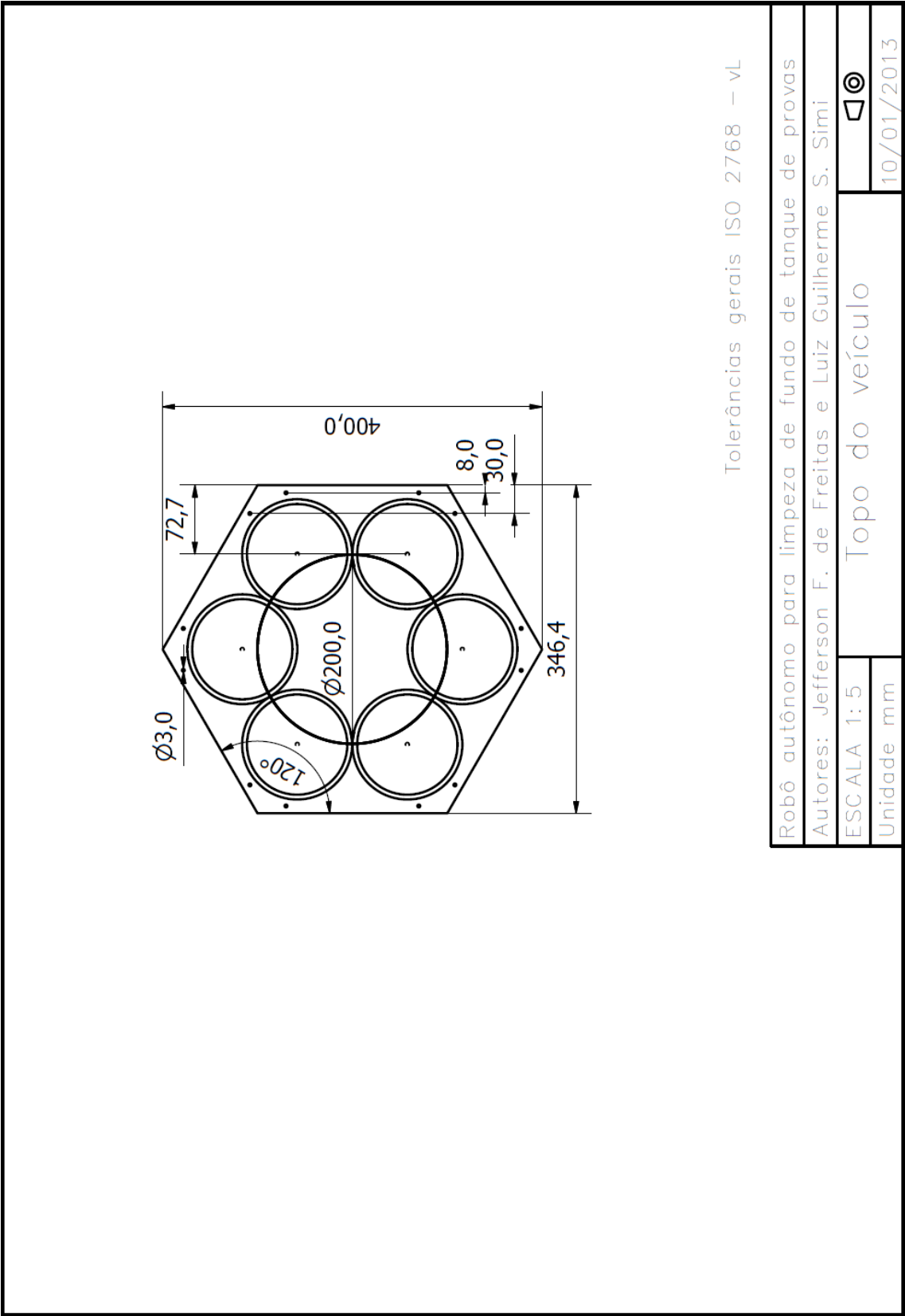


Figura 10-3 Topo do veículo

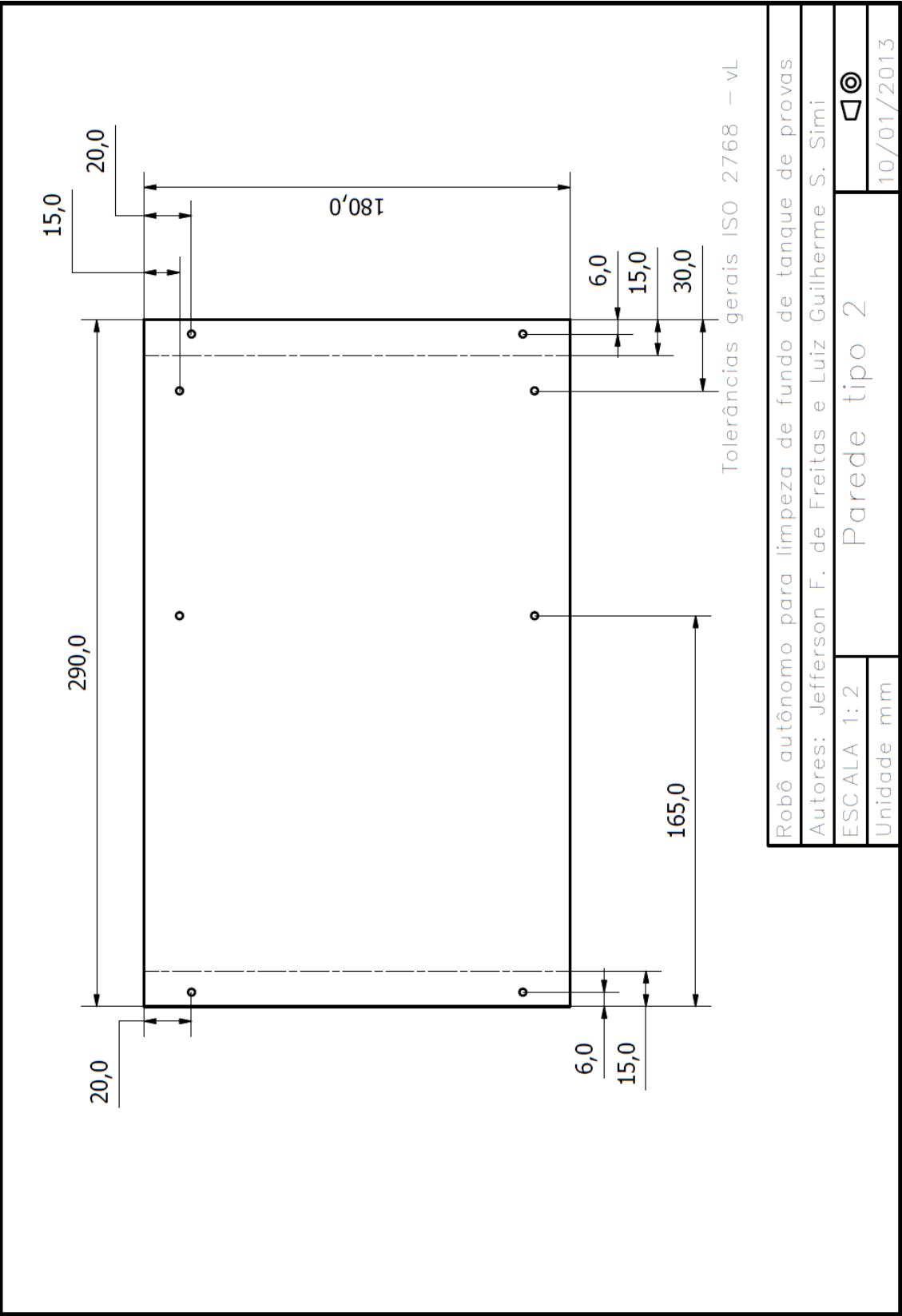


Figura 10-4 Parede sem bomba

a

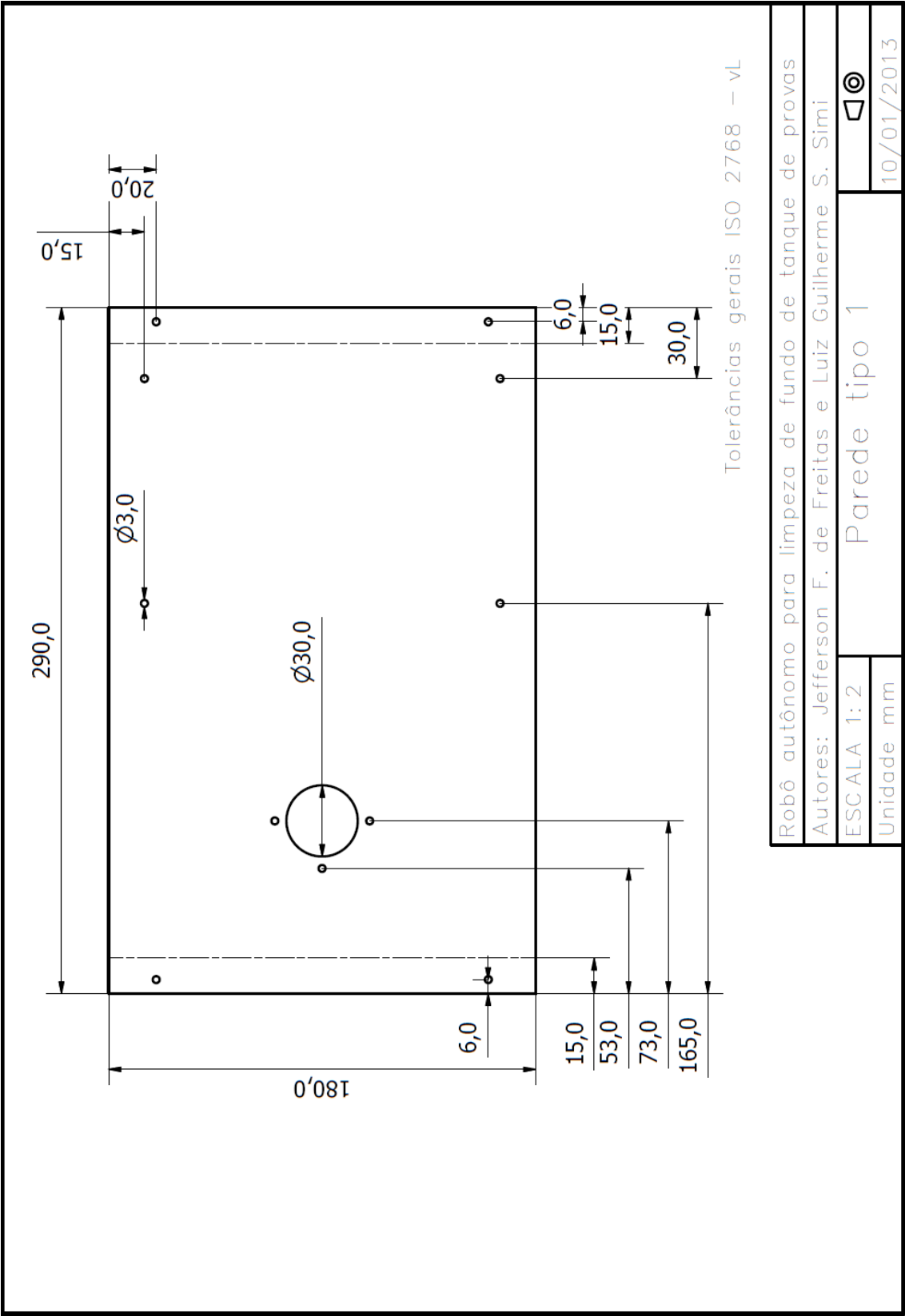


Figura 10-5 Parede com bomba

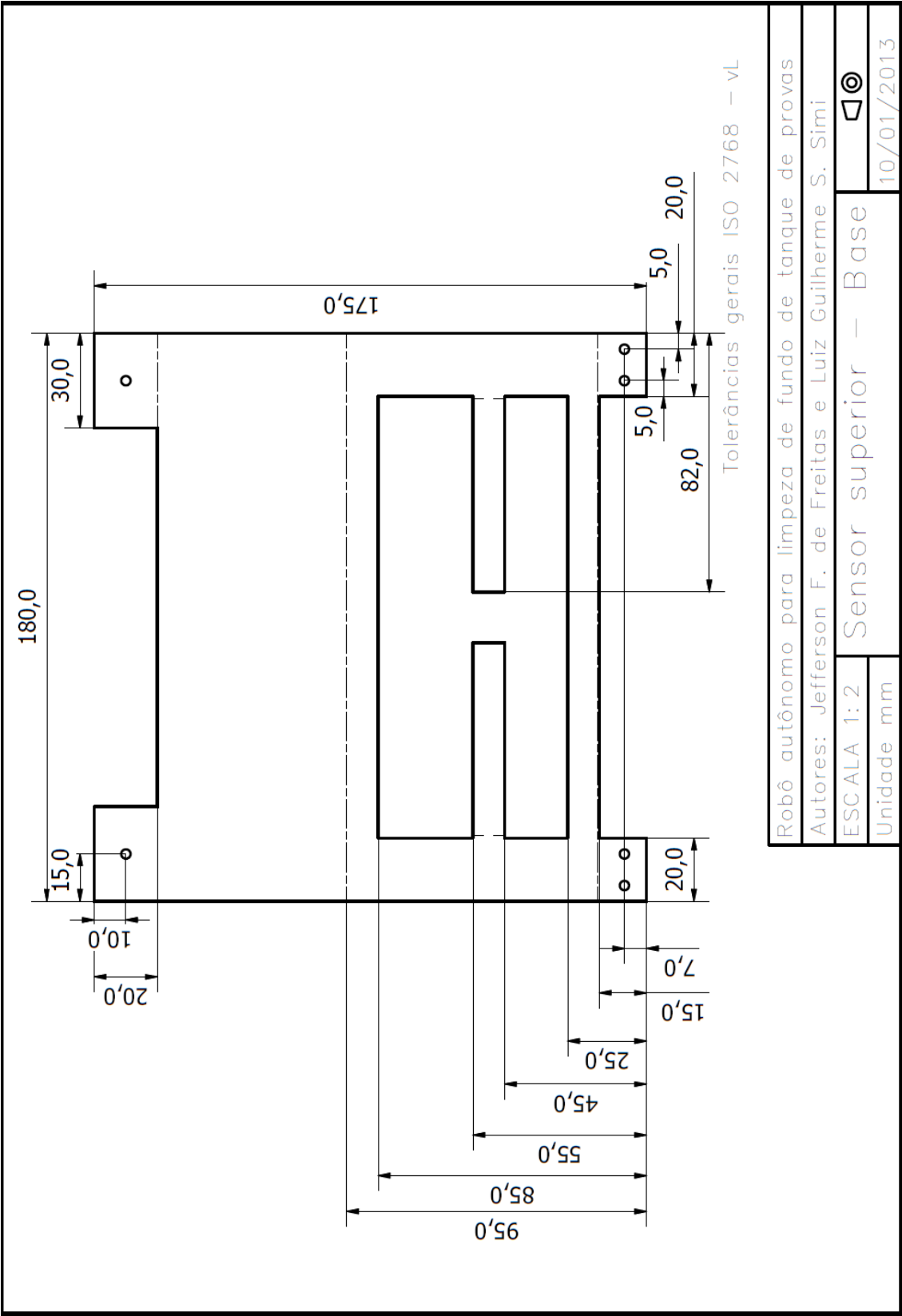
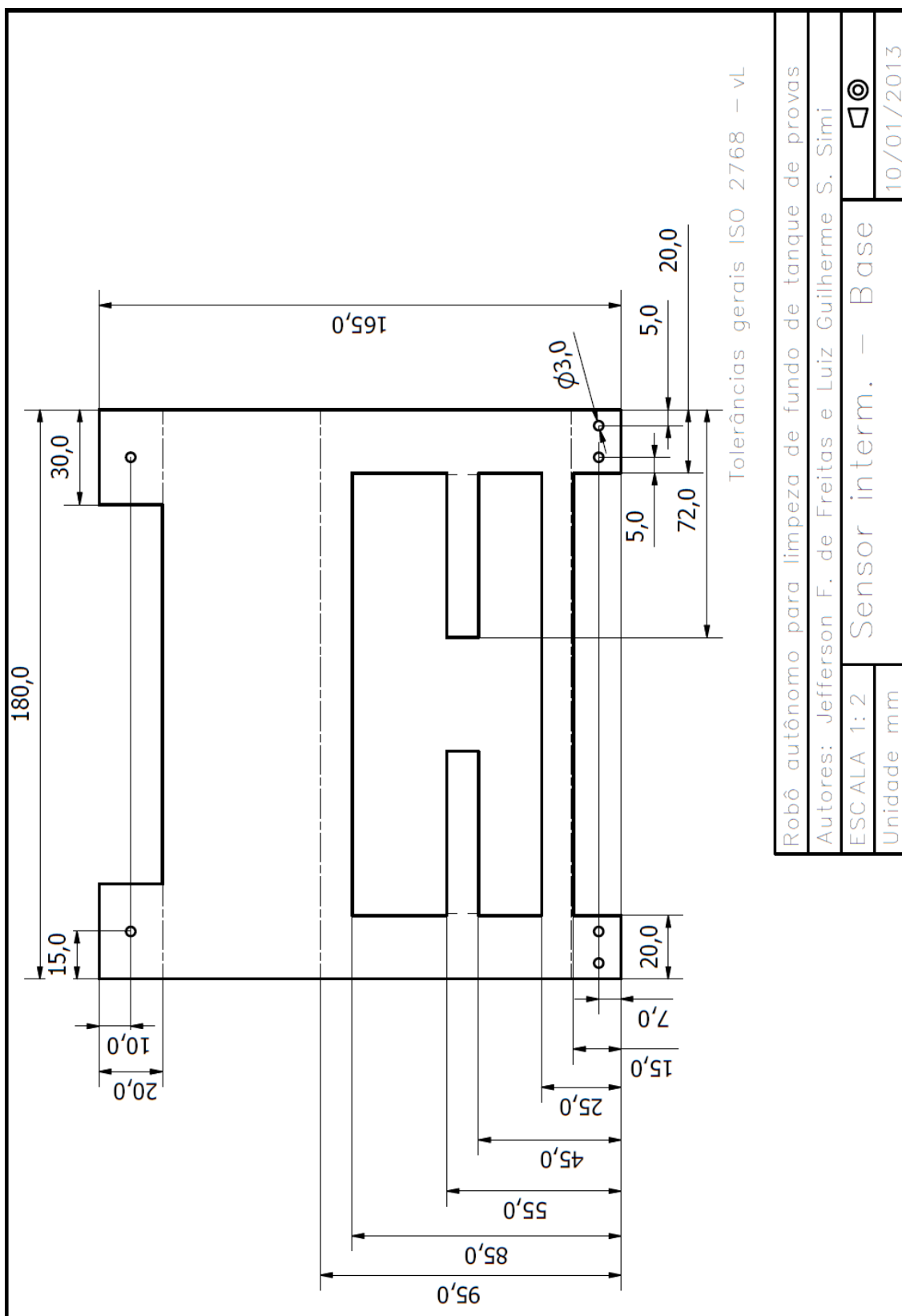


Figura 10-6 Base dos sensores superiores



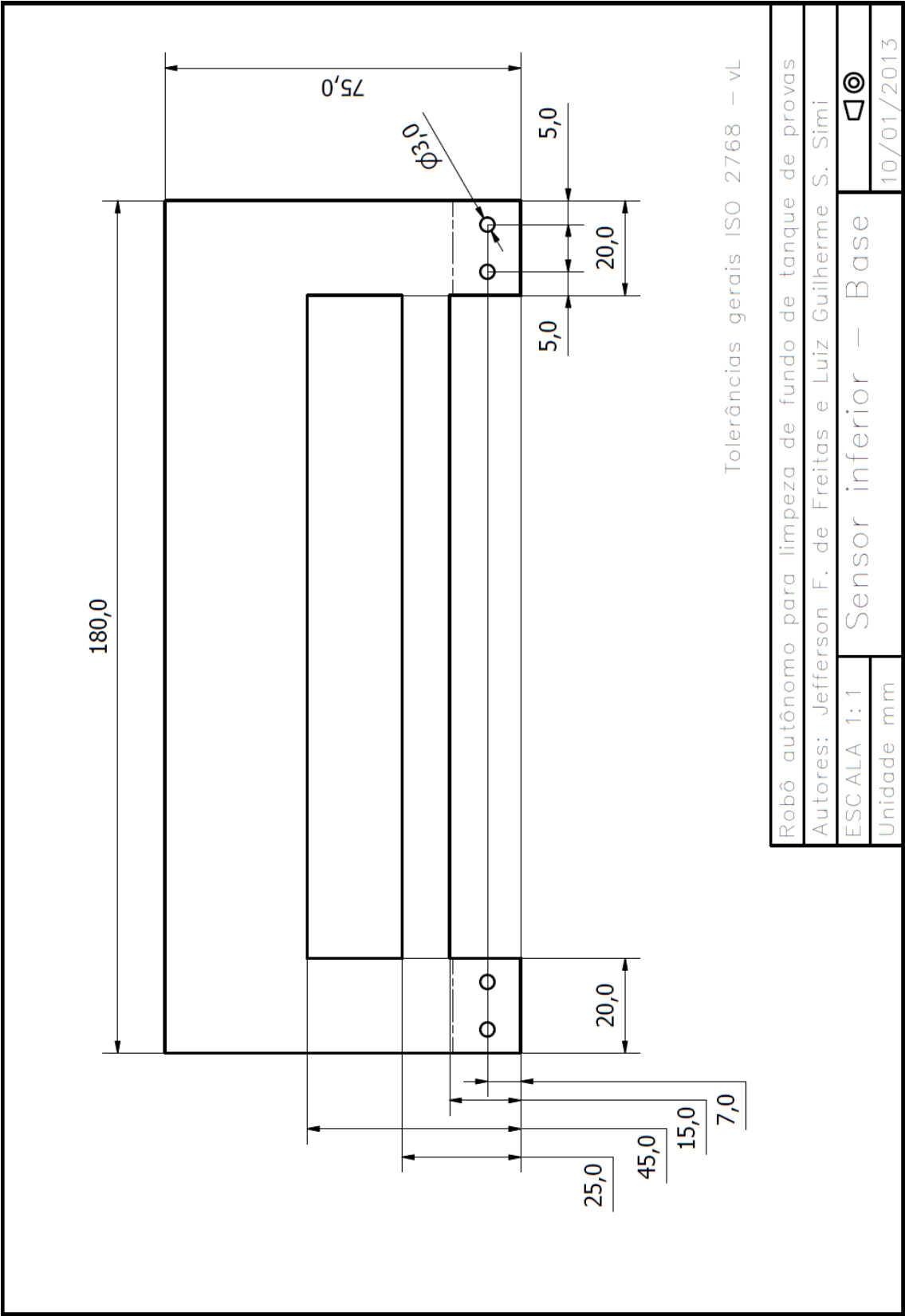


Figura 10-8 Base dos sensores inferiores

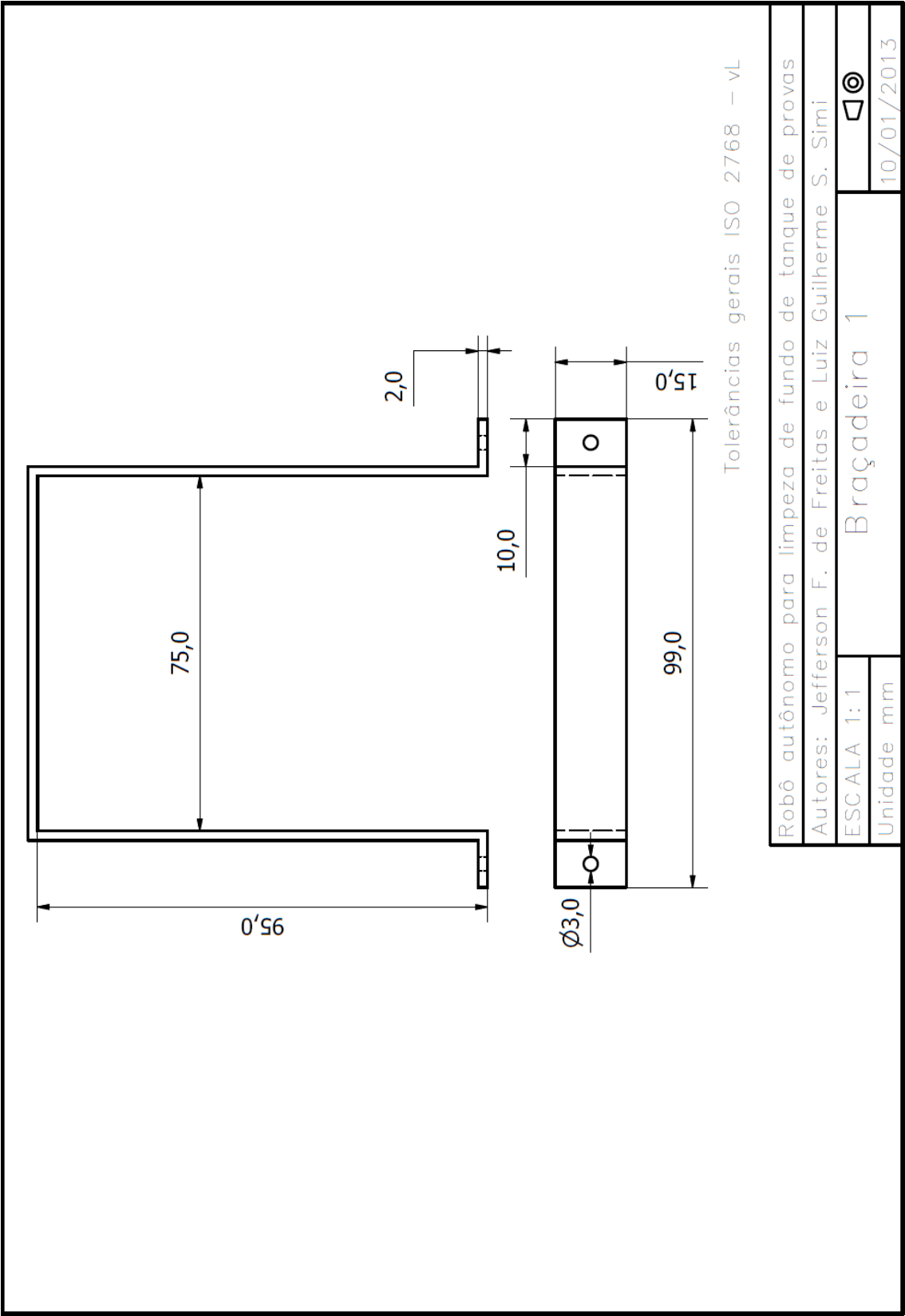


Figura 10-9 Braçadeira tipo 1

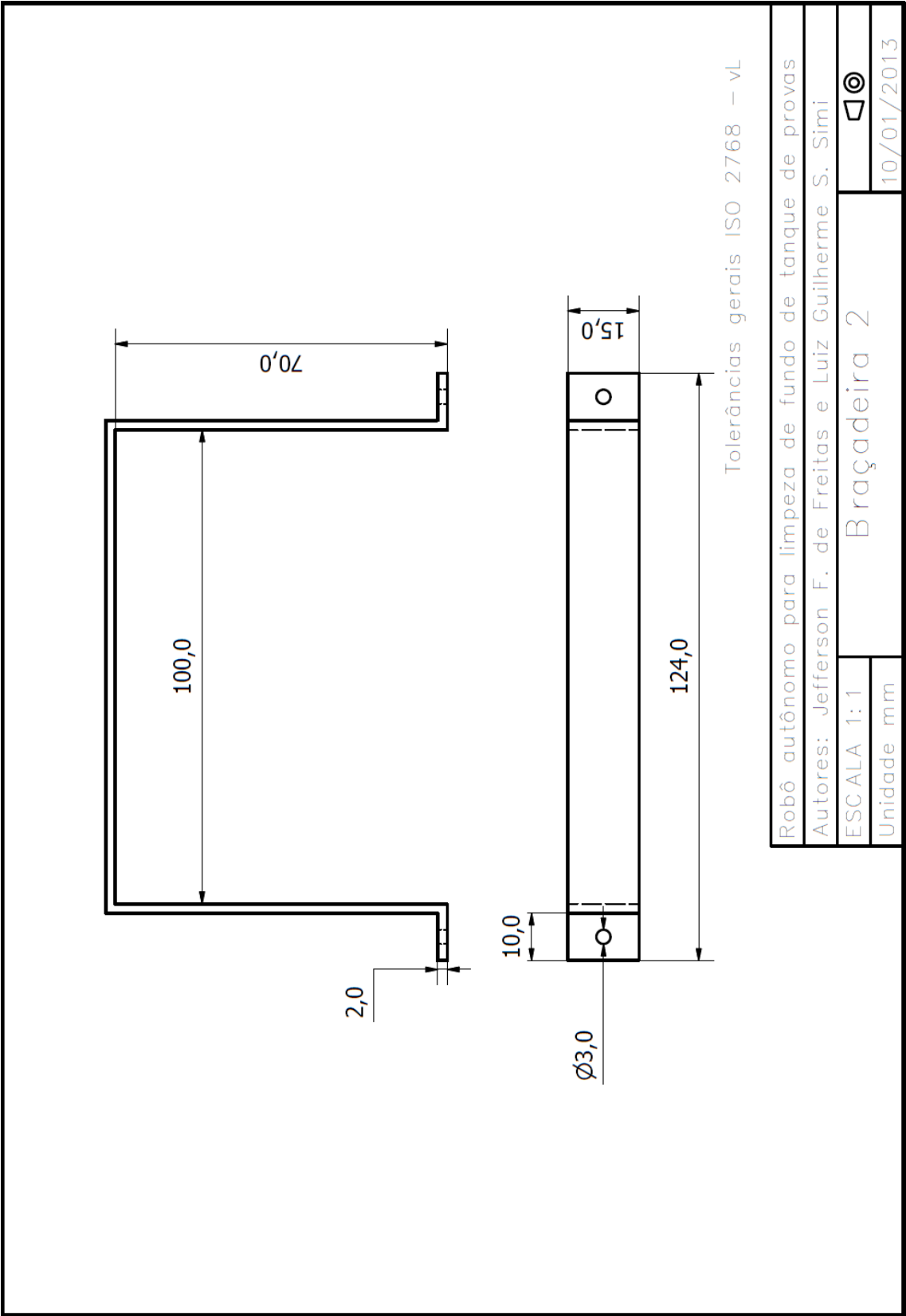


Figura 10-10 Braçadeira tipo 2

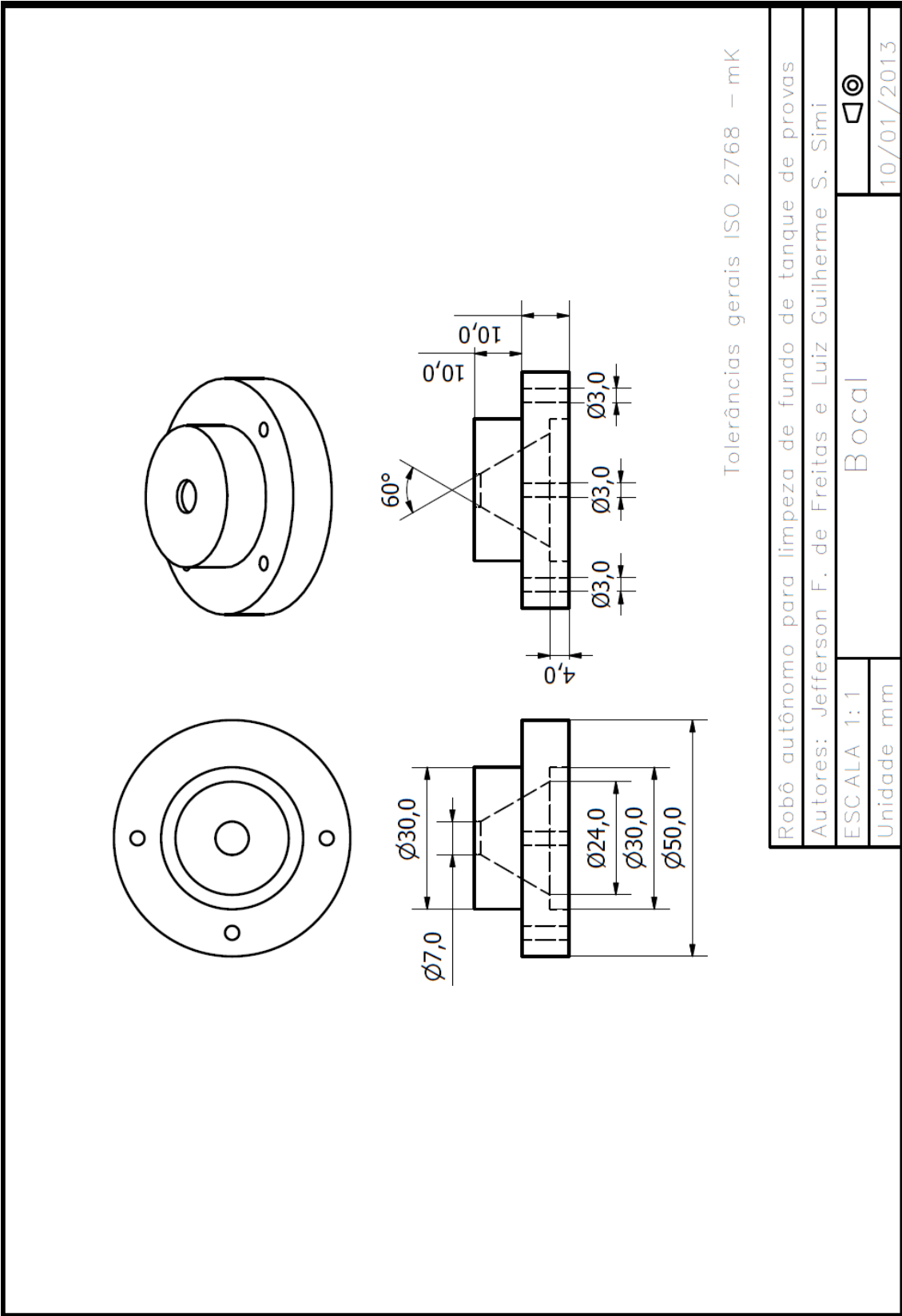


Figura 10-11 Bocais

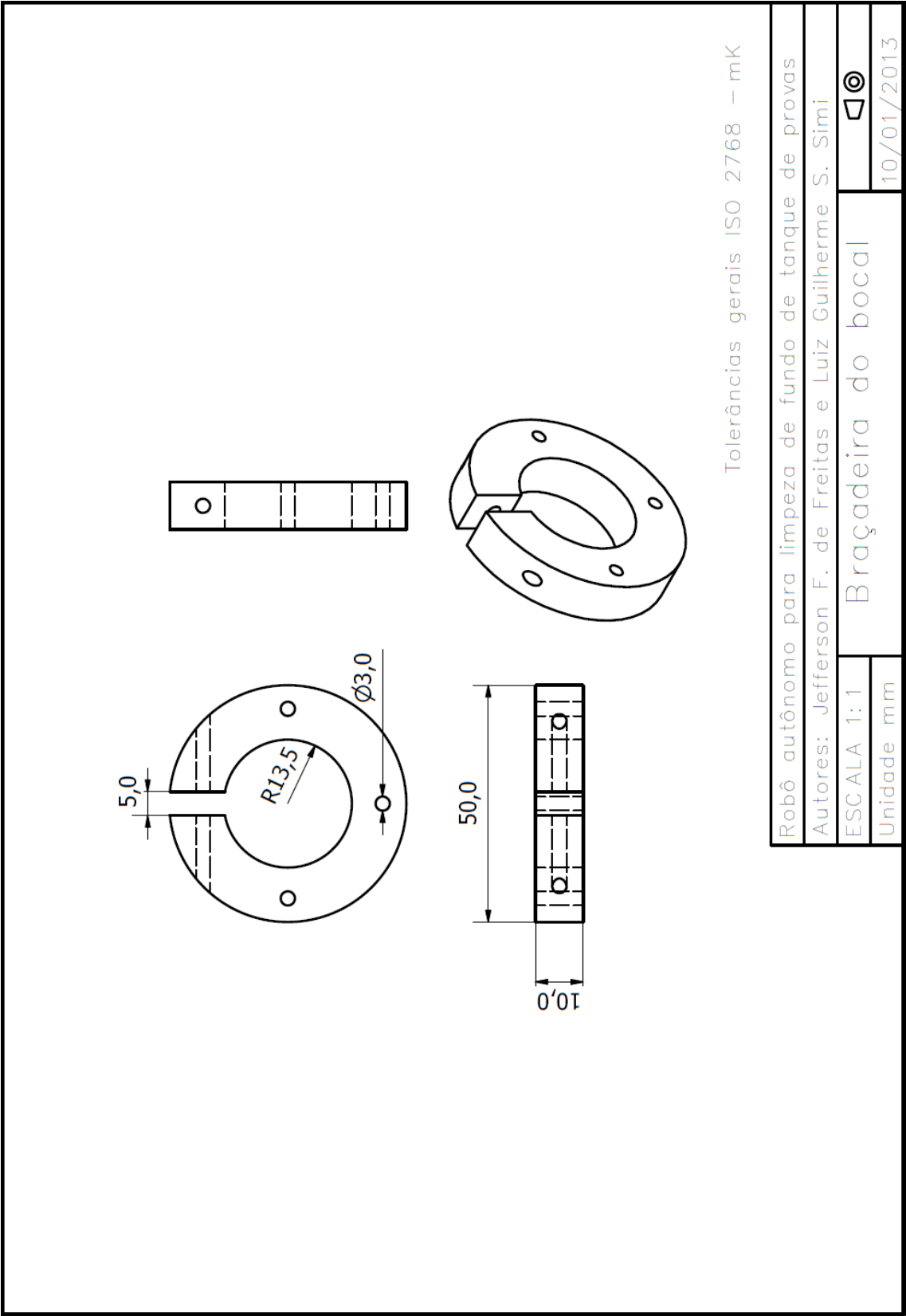


Figura 10-12 Braçadeira para fixação do bocal

10.3 Desenhos de conjunto

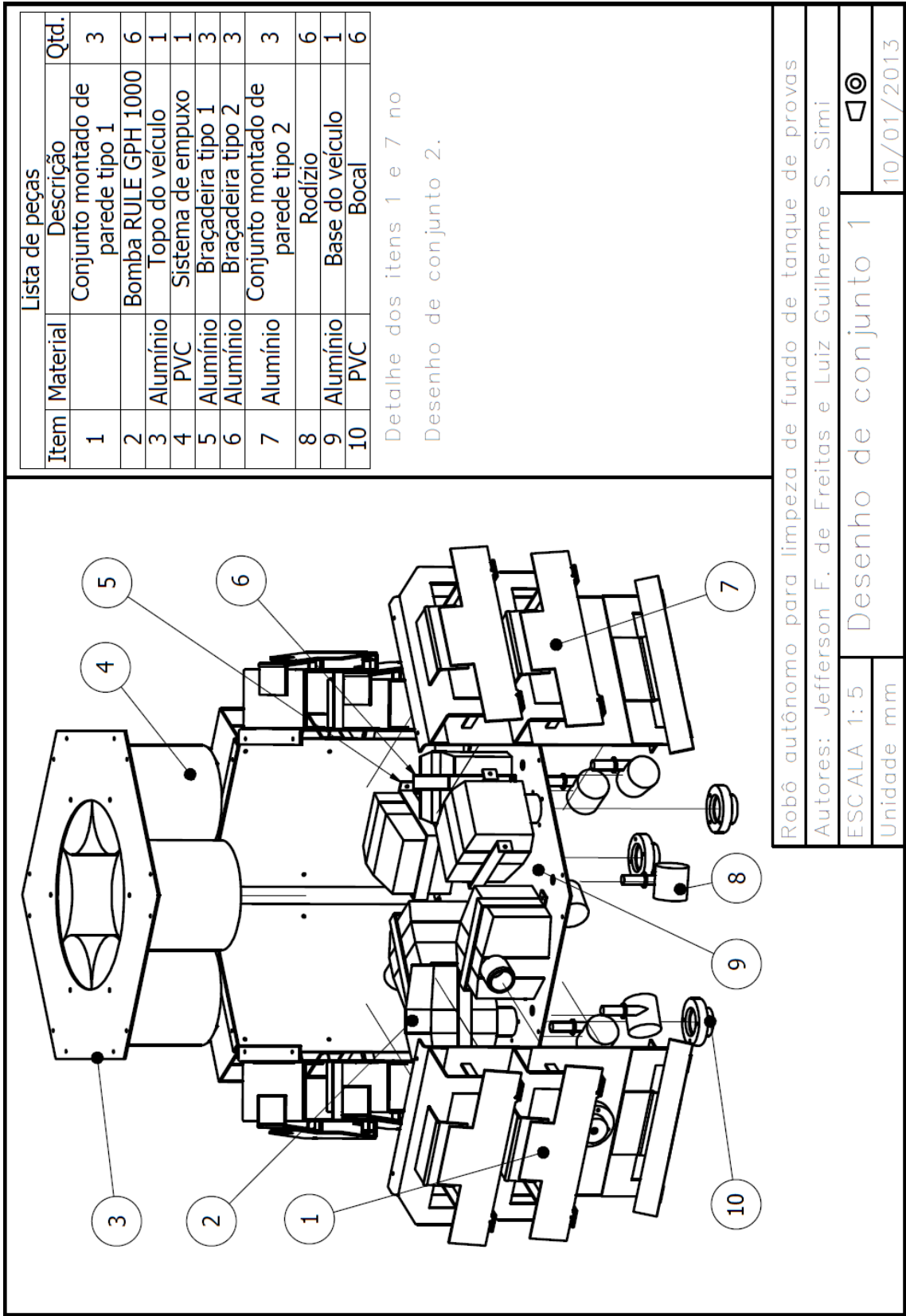


Figura 10-13 Desenho de conjunto do veículo

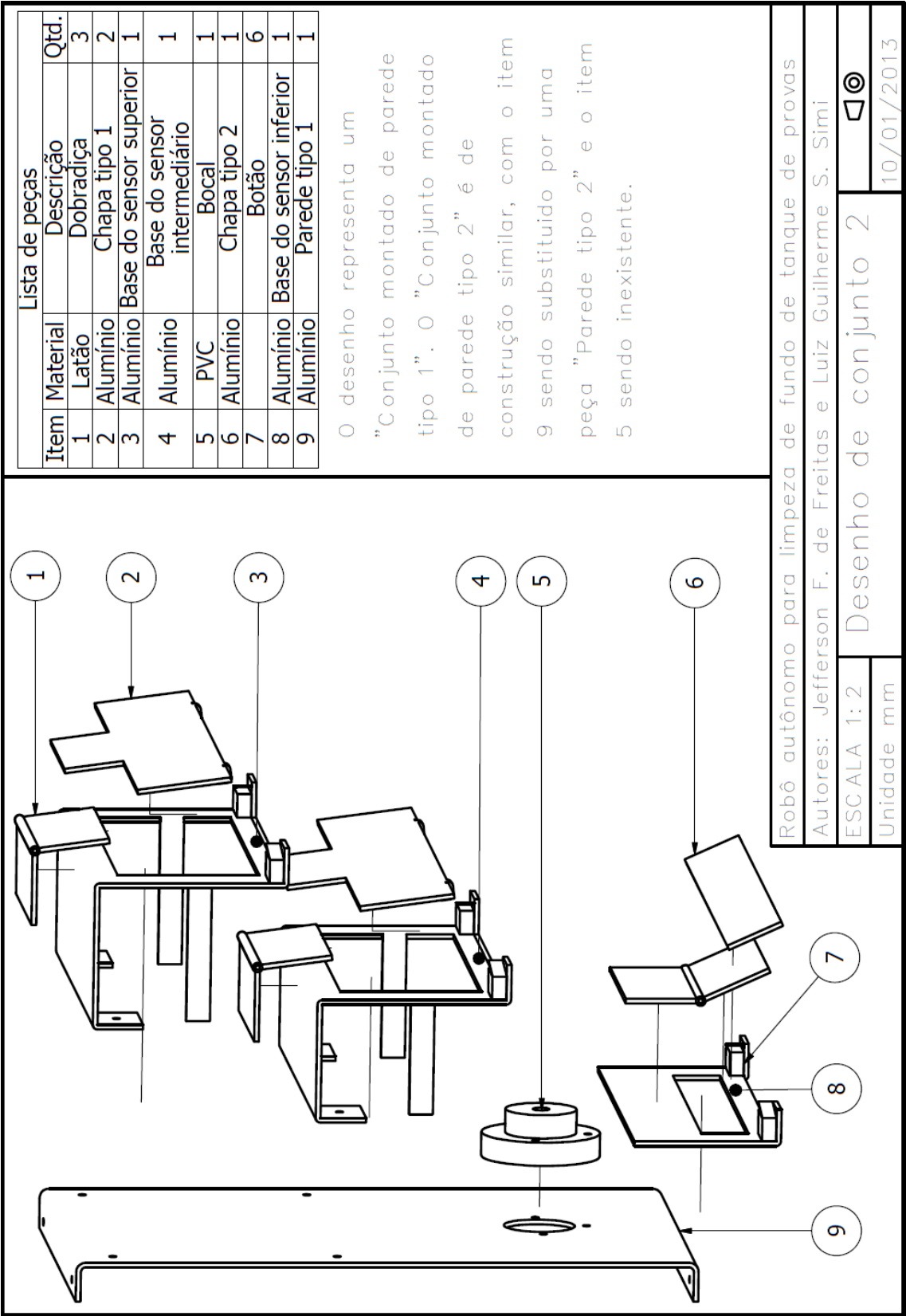


Figura 10-14 Desenho de conjunto da parede do veículo

10.4 Fotos

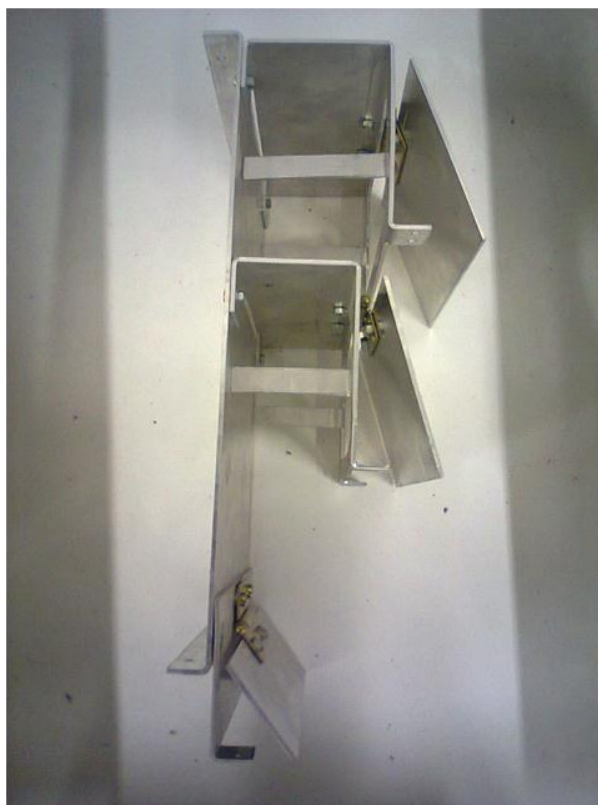


Figura 10-15 Parede do veículo com as três bases dos sensores



Figura 10-16 Parede do veículo com bases dos sensores e bomba com bocal

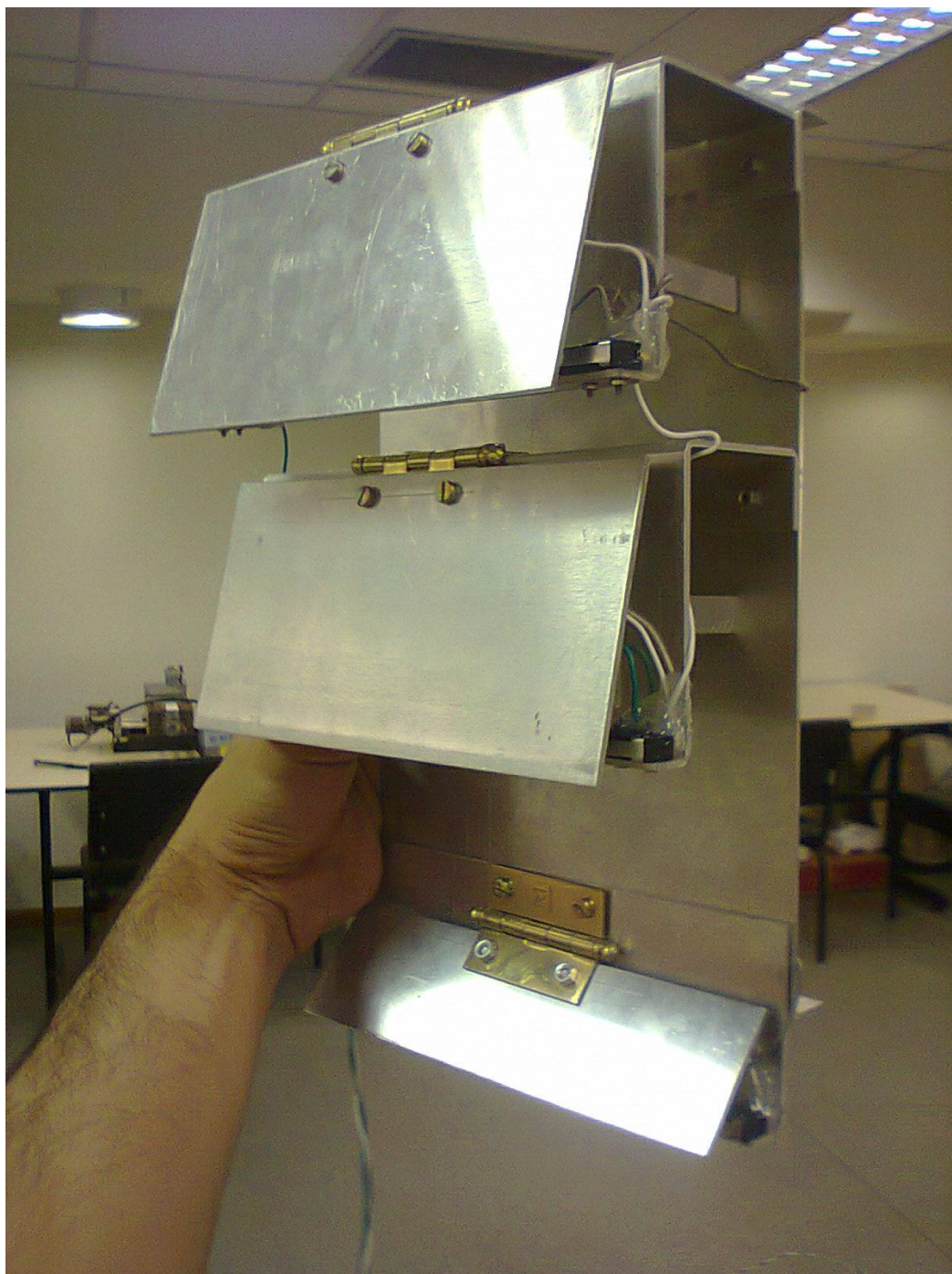


Figura 10-17 Parede concluída

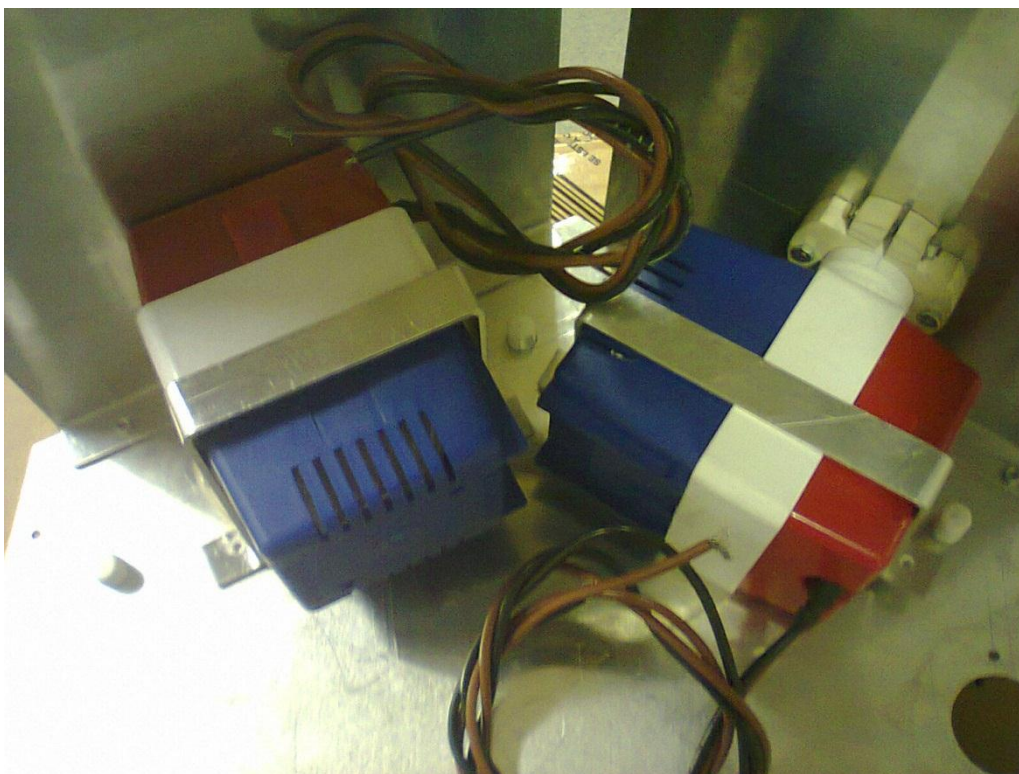


Figura 10-18 Bombas montadas com braçadeiras e bocais

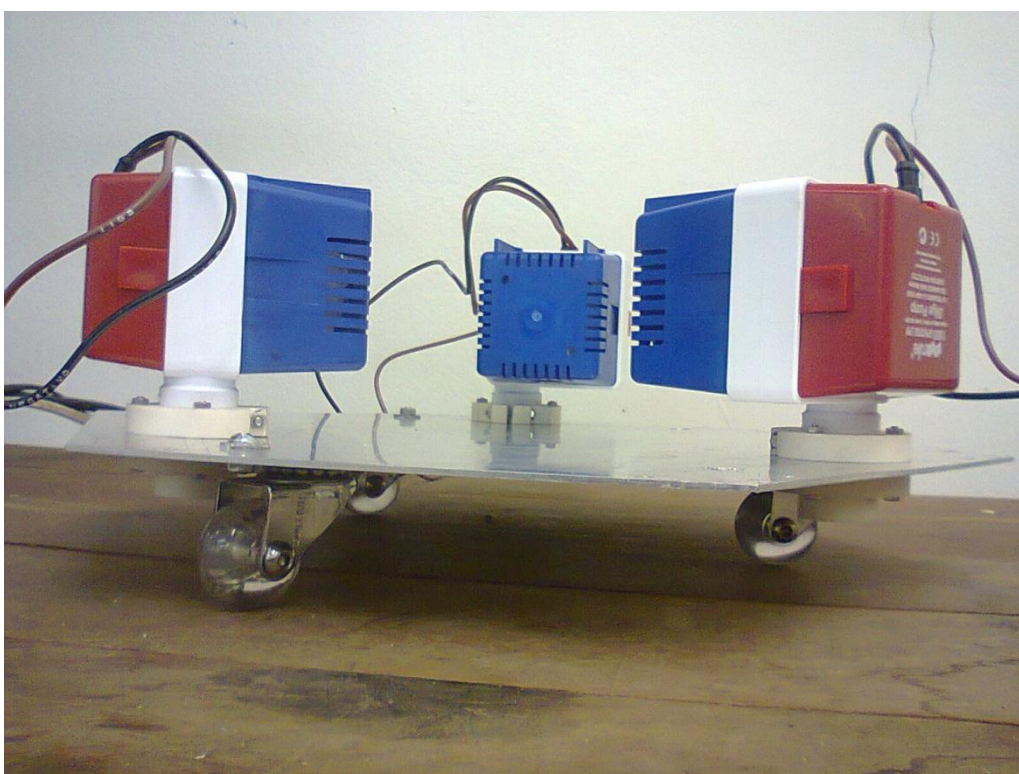


Figura 10-19 Fundo do veículo com rodízios, bombas e bocais montados

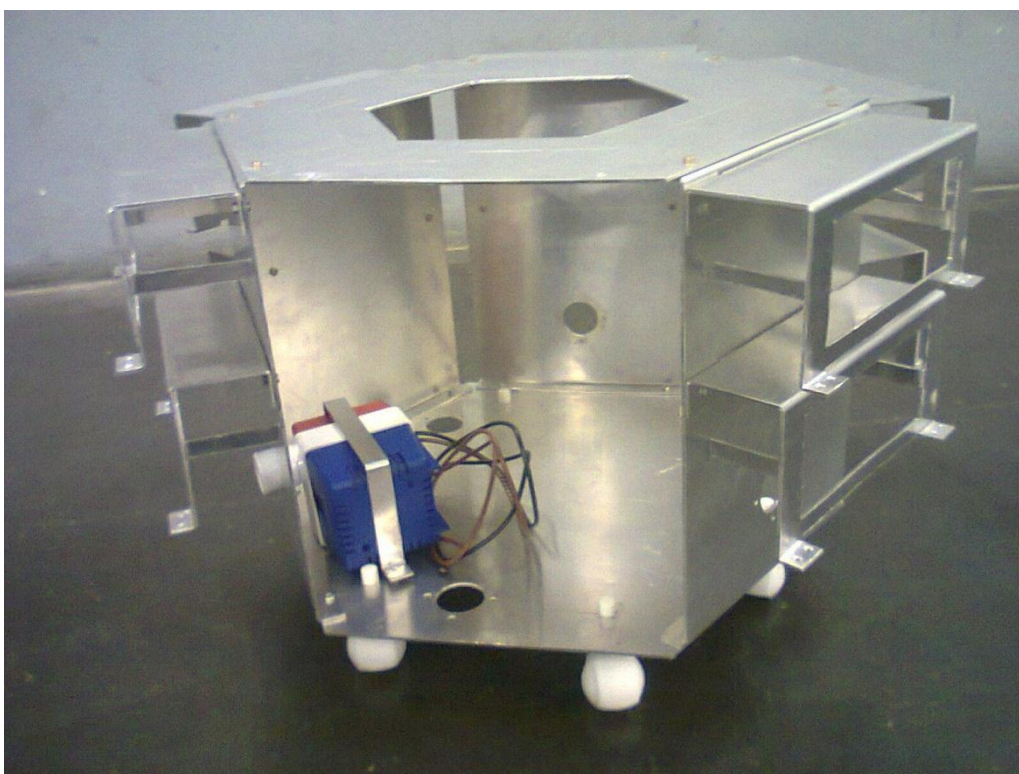


Figura 10-20 Chassis aberto

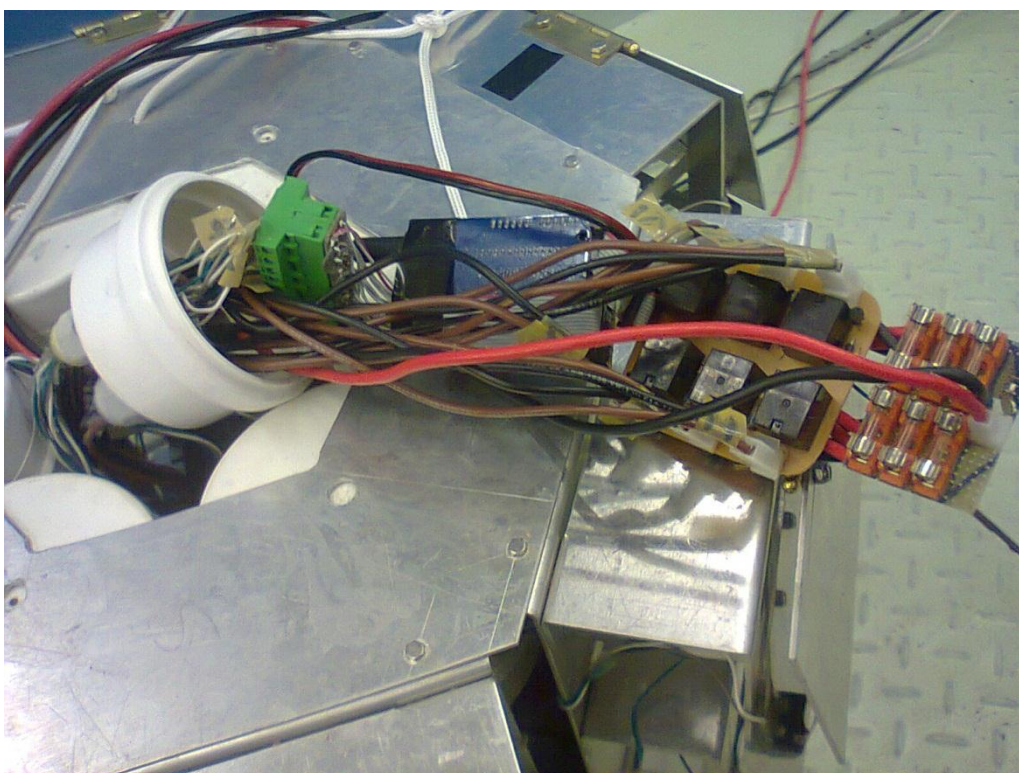


Figura 10-21 Eletrônica montada

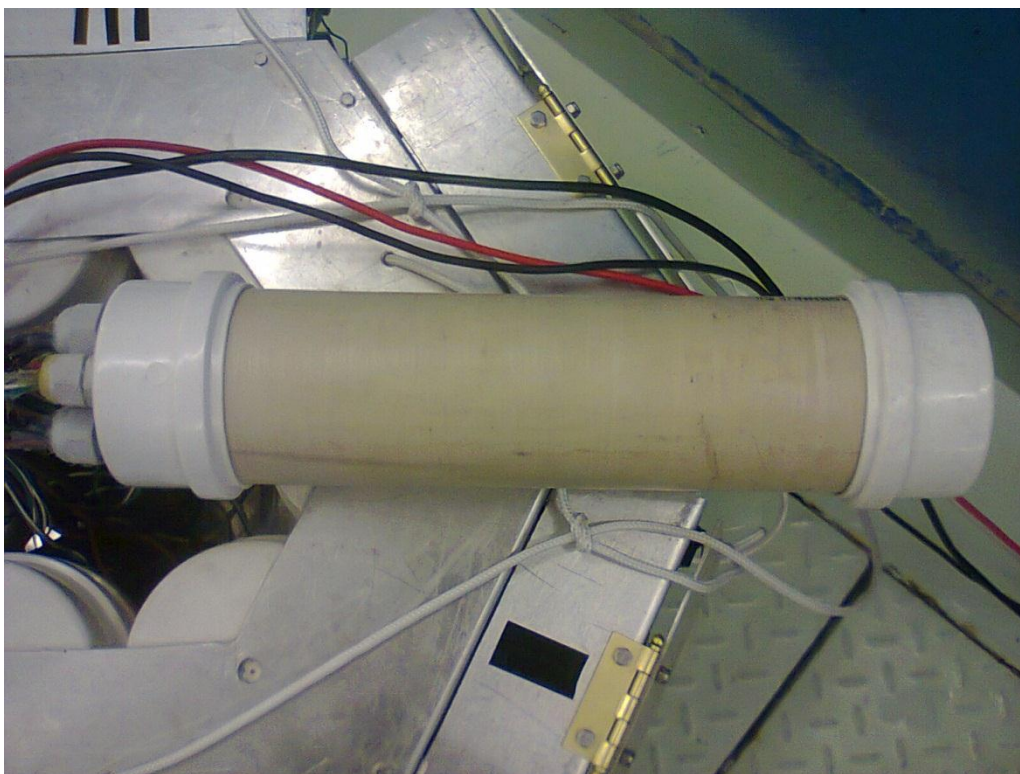


Figura 10-22 Eletrônica vedada e acoplada ao veículo



Figura 10-23 Veículo montado sem vedação da eletrônica

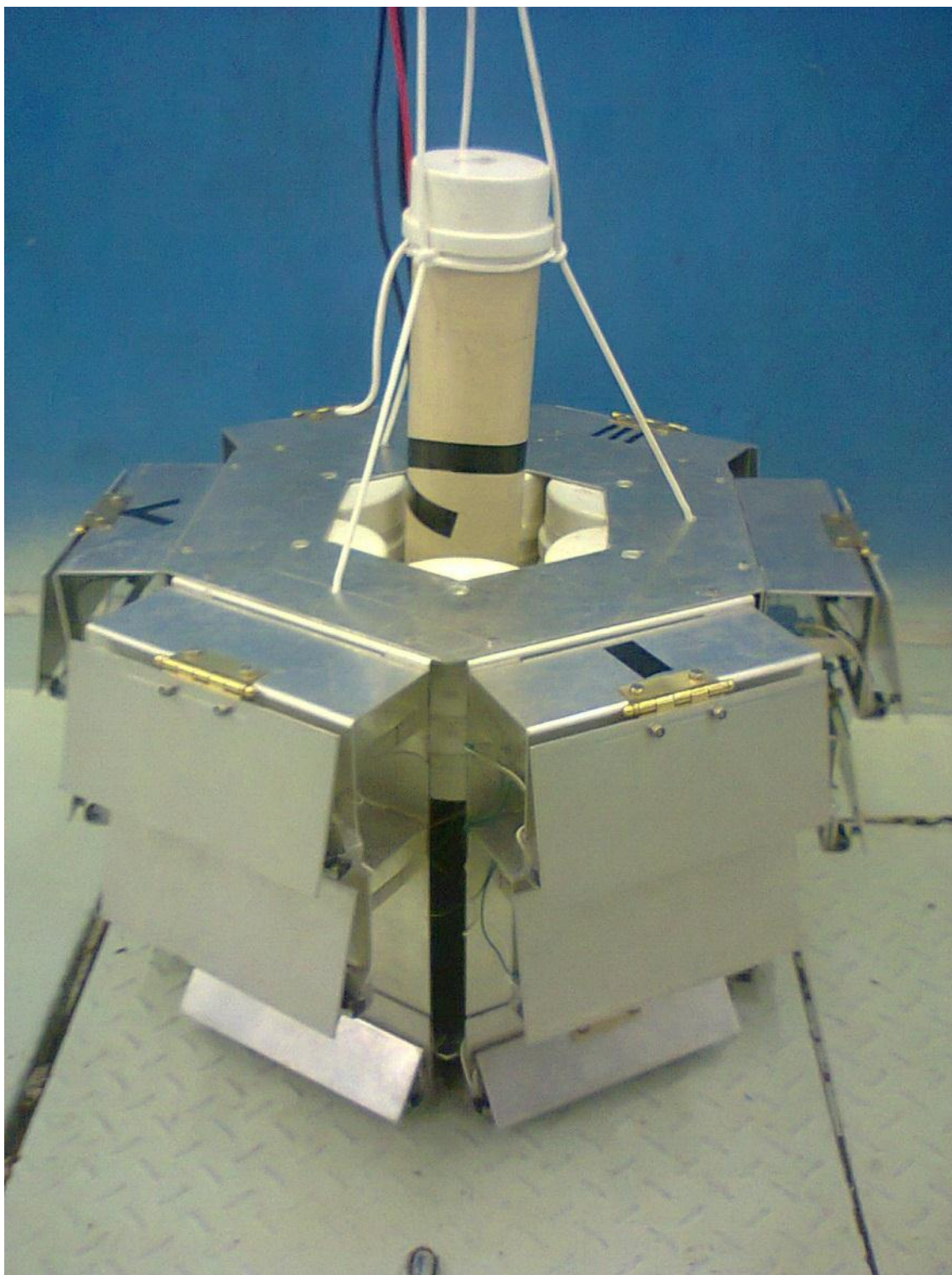


Figura 10-24 Veículo concluído

10.5 Algoritmo de otimização no Matlab

ForcaBomba.m

```
% Calculo da força da bomba em função das dimensões do redutor de
seção

A1 = 6.41*10^-4;
ro = 1000;
mi = 1.003*10^-3;
er = 0.01/0.03;
g = 9.8;
%Hmax = 5.1169;
Hmax = 3.9;
%Qmax = 1000 * 1.05*10^-6;
Qmax = 860 * 1.05*10^-6;

discr = 60; %numero de elementos da discretizacao para cada parametro

%área
A2ini = 0.1*A1;
A2fin = 0.5*A1;
dA2 = (A2fin-A2ini)/discr;

%ângulo
Tini = 10;
Tfin = 180;
dT = (Tfin-Tini)/discr;

A2 = A2ini:dA2:A2fin; %relacao de reducao
T = Tini:dT:Tfin; %angulo de reducao teta

lin = length(A2);
col = length(T);

F = zeros(lin,col);
maxk1 = zeros(lin,1);
maxk2 = zeros(lin,1);
maxk3 = zeros(lin,1);

for cont1=1:lin
    for cont2=1:col

        f = (-0.5/log10(er/3.7))^2; %regime turbulento rugoso

        %{
        %perda de carga distribuida regime turbulento liso
        Re = ro*Qmax/A2(cont1)*0.029*mi;
        f = 1; fant = 1; it=0;
        while ( it==0 || (it<1000 && abs(f-fant)/f > 0.001) ) %
enquanto erro > 0.1%
            fant = f;
            f = 1/( 2* log10(Re*sqrt(f)) - 0.8 )^2;
            it=it+1;
        end
        %}

        %coeficientes
```



```

k1 = coef(A2(cont1)/A1,T(cont2)); %perda de carga por redução
k2 = 1; %perda de carga por saída de canalização
k3 = f*0.02/sqrt(4*A2(cont1)/pi); %perda de carga distribuída

maxk1(cont1) = k1/(2*g)/A2(cont1)^2;
maxk2(cont1) = k2;
maxk3(cont1) = k3;

%Solução por Báskara
c1 = (k1+k2+k3)/(2*g)/A2(cont1)^2 ;%- k2/(2*g)/A1^2;
c2 = Hmax/Qmax;
c3 = - Hmax;

delta = c2^2 - 4*c1*c3;

if (delta<0)
    aviso = 'Erro: Delta Negativo'
end

Q = (-c2+sqrt(delta))/(2*c1);

F(cont1,cont2) = ro*Q^2/A2(cont1);

end

end

%plot

figure
contour(T,A2/A1,F,30); hold;
%restricao
L = 0.02; plot(360/pi*atan((sqrt(A1)-sqrt(A2))/sqrt(pi)/L),A2/A1);
hold;
title('Forca da Bomba');
xlabel('Ângulo (graus)');
ylabel('Relação A2/A1');

figure
surf(T,A2/A1,F);
title('Forca da Bomba (N)');
xlabel('Ângulo (graus)');
ylabel('Relação A2/A1');

figure
plot(A2/A1,F(:,round((60-Tini)/(Tfin-Tini)*discr)+1),'r')
title('Ângulo de redução 60°');
xlabel('Relação A2/A1');
ylabel('Força da Bomba (N)');

```

coef.m

```

function [ k ] = coef( rel, tet )

%Retorna o coeficiente de perda de carga para uma redução suave de secção
%transversal em função da relação entre as áreas e do ângulo de redução

```

```

%           ~30   60   90   120  150  180
kcontracao = [0.05 0.06 0.12 0.18 0.24 0.26;
              0.05 0.07 0.17 0.27 0.35 0.41;
              0.05 0.08 0.19 0.29 0.37 0.43];

lin=0;col=0;k=0;

if (tet>=10 && tet<30)
    k = 0.05;
elseif (tet>=30 && tet<60)
    col = 1;
elseif (tet>=60 && tet<90)
    col = 2;
elseif (tet>=90 && tet<120)
    col = 3;
elseif (tet>=120 && tet<150)
    col = 4;
elseif (tet>=150 && tet<=180)
    col = 5;
else aviso='ERRO em coef_col'
end

if (rel<=0.5 && rel>0.25)
    lin = 1;
elseif (rel<=0.25 && rel>=0.1)
    lin = 2;
else aviso='ERRO em coef_lin'
end

if (k~=0.05)

    dt1 = (tet-30*col)/30;
    dt2 = (30*(col+1)-tet)/30;

```

```
itp1 = kcontracao(lin,col)*dt2 + kcontracao(lin,col+1)*dt1;
itp2 = kcontracao(lin+1,col)*dt2 + kcontracao(lin+1,col+1)*dt1;

if (lin==1)
    dt1 = (0.5-rel)/0.25;
    dt2 = (rel-0.25)/0.25;
elseif (lin==2)
    dt1 = (0.25-rel)/0.15;
    dt2 = (rel-0.1)/0.15;
end

k = itp1*dt2 + itp2*dt1;

end

end
```

10.6 Codificação do programa embarcado no Arduino

// Escola Politécnica da USP - Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

// Robô autônomo para varredura em aplicação específica: Tanque de Provas Numérico

/******

Dezembro de 2012

Autores:

L. Simi

J. Freitas

*****/

/****** Defines *****/

// duration of jump stages in milliseconds

#define firstJumpTime 1000 // 1500

#define secondJumpTime 10000 // 10000

#define thirdJumpTime 12000 // 6000

#define fourthJumpTime 30000 // 15000

// duration of cycle stages in minutes

#define cycleTimeOne 1

#define cycleTimeTwo 1

// limit of time for watchdog alert in minutes

#define dogTimeLimit 1

#define numOfSensors 6 // of each type

#define moveOff 0

#define jumpOn 10

#define jumpOff 11

#define leftSide 0

#define rightSide 1

#define MOVE 1

#define JUMP 2

#define SEARCH 3

#define STOP 4

#define IDLE 0

#define OUTSIDE 1

#define SEARCHING 2

#define JUMPING 3

#define INSIDE 4

const int sensorMovePin[] = {
7, 6, 2, 3, 4, 5};

const int sensorJumpPin[] = {
14, 15, 16, 17, 18, 19};

const int actionPump2Pin = 10;

const int actionPump4Pin = 9;

const int actionPump6Pin = 11;

const int actionPumpJPin = 8;

/******

// Variables

```

int currentSensor = 0;

int sensorMoveState[numOfSensors];
int sensorJumpState[numOfSensors];

int vehicleState;
int thisDir;

int event;
unsigned long lastEvent;
unsigned long timeCounter;

boolean watchDogFlag;
unsigned long lastWatchDogTime;

// Setup
void setup(){

    // Input and Output pins
    for( currentSensor=0; currentSensor < numOfSensors; currentSensor++){
        pinMode(sensorMovePin[currentSensor],INPUT);
        pinMode(sensorJumpPin[currentSensor],INPUT);
    }
    pinMode(actionPump2Pin,OUTPUT);
    pinMode(actionPump4Pin,OUTPUT);
    pinMode(actionPump6Pin,OUTPUT);
    pinMode(actionPumpJPin,OUTPUT);

    // Serial for monitoring
    Serial.begin(19200);
    Serial.println("- Serial Begin -");
    Serial.println();

    // Deactivate all pumps on start
    setPumpsTo(moveOff);
    setPumpsTo(jumpOff);

    // Initial State and Direction
    vehicleState = OUTSIDE;
    Serial.println(" ----- OUTSIDE ----- ");
    thisDir = 1; setPumpsTo(thisDir);

    // Miscellaneous Variables
    event = 1;
    lastEvent = millis();
    timeCounter = 0;

    watchDogReset();
}

// Loop
void loop(){

    //executeJump(thisDir);
    //pumpTest();
    mainControlOfStates();
    //if( changeOfSensorState() )
    //printSensorState();

}

```



```

// MainControlOfStates
void mainControlOfStates(){

    if( watchDogAlert() ) watchDogAction();

    switch(vehicleState){

        case OUTSIDE:

            if( changeOfSensorState() ){
                thisDir = getMoveDirection(thisDir);
                setPumpsTo(thisDir);
                printSensorState();
            }

            if( clockWatch() == SEARCH ){
                Serial.println(" ----- SEARCHING ----- ");
                vehicleState = SEARCHING;
            }

            break;

        case SEARCHING:

            if( changeOfSensorState() == MOVE ){
                thisDir = getMoveDirection(thisDir);
                setPumpsTo(thisDir);
                printSensorState();
            }
            else if( getJumpDirection(thisDir) ){
                thisDir = getJumpDirection(thisDir);
                printSensorState();
                Serial.println(" ----- JUMPING ----- ");
                vehicleState = JUMPING;
            }

            break;

        case JUMPING:

            executeJump(thisDir);
            Serial.println(" ----- INSIDE ----- ");
            vehicleState = INSIDE;

            // start counting time inside
            timeCounter = millis();

            break;

        case INSIDE:

            if( changeOfSensorState() ){
                thisDir = getMoveDirection(thisDir);
                setPumpsTo(thisDir);
                printSensorState();
            }

            if( clockWatch() == STOP ){

```

```

    Serial.println(" ----- IDLE ----- ");
    vehicleState = SEARCH;
}

break;

default:
    setPumpsTo(moveOff);
    setPumpsTo(jumpOff);
}

}

// ChangeOfSensorState
int changeOfSensorState(){

    int stateChange = 0;

    for( currentSensor=0; currentSensor < numOfSensors; currentSensor++){

        if( sensorJumpState[currentSensor] == LOW && !digitalRead(sensorJumpPin[currentSensor]) ){
            sensorJumpState[currentSensor] = HIGH;
            stateChange = JUMP;
            watchDogReset();
        }
        else if( sensorJumpState[currentSensor] == HIGH && digitalRead(sensorJumpPin[currentSensor]) ){
            sensorJumpState[currentSensor] = LOW;
            watchDogReset();
        }

        if( sensorMoveState[currentSensor] == LOW && !digitalRead(sensorMovePin[currentSensor]) ){
            sensorMoveState[currentSensor] = HIGH;
            stateChange = MOVE;
            watchDogReset();
        }
        else if( sensorMoveState[currentSensor] == HIGH && digitalRead(sensorMovePin[currentSensor]) ){
            sensorMoveState[currentSensor] = LOW;
            watchDogReset();
        }

    }

    return stateChange;

}

// GetJumpDirection
int getJumpDirection(int currentDirection){

    // left jump sensor active
    if( sensorJumpState[hexShift(currentDirection,leftSide,1) - 1] ){
        return hexShift(currentDirection,leftSide,1);
    }
    // right jump sensor active
    else if( sensorJumpState[hexShift(currentDirection,rightSide,1) - 1] ){
        return hexShift(currentDirection,rightSide,1);
    }
    // front jump sensor active
    else if( sensorJumpState[currentDirection - 1] ){

```

```

    return currentDirection;
}
// any other jump sensor active
else
    return 0;
}

// GetMoveDirection
int getMoveDirection(int currentDirection){

    // any left sensor active
    if( sensorMoveState[hexShift(currentDirection,leftSide,1) - 1] ||
    sensorJumpState[hexShift(currentDirection,leftSide,1) - 1] ){
        return hexShift(currentDirection,rightSide,1);
    }
    // any right sensor active
    else if( sensorMoveState[hexShift(currentDirection,rightSide,1) - 1] ||
    sensorJumpState[hexShift(currentDirection,rightSide,1) - 1] ){
        return hexShift(currentDirection,leftSide,1);
    }
    // any front sensor active
    else if( sensorMoveState[currentDirection - 1] || sensorJumpState[currentDirection - 1] ){
        if( randomBool() )
            return hexShift(currentDirection,leftSide,2);
        else
            return hexShift(currentDirection,rightSide,2);
    }
    // any other sensor active
    else
        return currentDirection;
}

// ClockWatch
int clockWatch(){

    if( timeCounter!=0 && (millis()-timeCounter) > cicleTimeTwo*60000 )
        return STOP;
    else if( millis() > cicleTimeOne*60000 )
        return SEARCH;
    else return 0;

}

// WatchDogAlert
boolean watchDogAlert(){
    if( (millis()-lastWatchDogTime) > dogTimeLimit*60000 )
        return true;
    else return false;
}

// WatchDogAction
void watchDogAction(){
    if(watchDogFlag){

        // WatchDog activated two consecutive times, check sensors for direction and jump
        thisDir = getMoveDirection(thisDir);
        executeJump(thisDir);

        watchDogReset();
    }
}

```

```

else{

    // WatchDog activated just once, invert move direction
    thisDir = hexShift(thisDir,rightSide,3);
    setPumpsTo(thisDir);

    watchDogFlag = true;
    lastWatchDogTime = millis();
}
}

// WatchDogReset
void watchDogReset(){
    lastWatchDogTime = millis();
    watchDogFlag = false;
}

// SetPumpsTo
void setPumpsTo(int mode){
    switch (mode){
        case moveOff:
            digitalWrite(actionPump2Pin,LOW);
            digitalWrite(actionPump4Pin,LOW);
            digitalWrite(actionPump6Pin,LOW);
            break;
        case 1:
            digitalWrite(actionPump2Pin,LOW);
            digitalWrite(actionPump4Pin,HIGH);
            digitalWrite(actionPump6Pin,LOW);
            break;
        case 2:
            digitalWrite(actionPump2Pin,LOW);
            digitalWrite(actionPump4Pin,HIGH);
            digitalWrite(actionPump6Pin,HIGH);
            break;
        case 3:
            digitalWrite(actionPump2Pin,LOW);
            digitalWrite(actionPump4Pin,LOW);
            digitalWrite(actionPump6Pin,HIGH);
            break;
        case 4:
            digitalWrite(actionPump2Pin,HIGH);
            digitalWrite(actionPump4Pin,LOW);
            digitalWrite(actionPump6Pin,HIGH);
            break;
        case 5:
            digitalWrite(actionPump2Pin,HIGH);
            digitalWrite(actionPump4Pin,LOW);
            digitalWrite(actionPump6Pin,LOW);
            break;
        case 6:
            digitalWrite(actionPump2Pin,HIGH);
            digitalWrite(actionPump4Pin,HIGH);
            digitalWrite(actionPump6Pin,LOW);
            break;
        case jumpOn:
            digitalWrite(actionPumpJPin,HIGH);
            break;
        case jumpOff:
            digitalWrite(actionPumpJPin,LOW);

```

```

        break;
    }
}

// PumpTest
void pumpTest(){

    if( timerCount() > 3000 ){

        switch(event){
            case 1: digitalWrite(actionPumpJPin,LOW); digitalWrite(actionPump2Pin,HIGH); break;

            case 2: digitalWrite(actionPump2Pin,LOW); digitalWrite(actionPump4Pin,HIGH); break;

            case 3: digitalWrite(actionPump4Pin,LOW); digitalWrite(actionPump6Pin,HIGH); break;

            case 4: digitalWrite(actionPump6Pin,LOW); digitalWrite(actionPumpJPin,HIGH); break;
        }

        event++;
        if(event > 4)
            event = 1;

        timerReset();
    }
}

unsigned long timerCount(){
    return (millis()-lastEvent);
}

void timerReset(){
    lastEvent = millis();
}

// PrintSensorState
void printSensorState(){

    Serial.println("Sensors: 1 2 3 4 5 6");
    Serial.print("  Move:");
    for( currentSensor=0; currentSensor < numOfSensors; currentSensor++){
        Serial.print(" ");
        Serial.print(sensorMoveState[currentSensor]);
    } Serial.println();
    Serial.print("  Jump:");
    for( currentSensor=0; currentSensor < numOfSensors; currentSensor++){
        Serial.print(" ");
        Serial.print(sensorJumpState[currentSensor]);
    } Serial.println();
    Serial.print("Direction = ");
    Serial.println(thisDir);
    Serial.println();

}

// ExecuteJump
void executeJump(int currentDirection){

    // move backwards
    Serial.println(" - Back");
    setPumpsTo(hexShift(currentDirection,rightSide,3));
    delay(firstJumpTime);
}

```

```

// stop and start jump
Serial.println(" - Rise");
setPumpsTo(moveOff);
setPumpsTo(jumpOn);
delay(secondJumpTime);

// end jump and start moving
Serial.println(" - Move");
setPumpsTo(jumpOff);
setPumpsTo(currentDirection);
delay(thirdJumpTime);

// wait for contact with ground
Serial.println(" - Wait");
setPumpsTo(moveOff);
delay(fourthJumpTime);

// continue moving
Serial.println(" - GoOn");
Serial.println();
setPumpsTo(currentDirection);
}

// HexShift
int hexShift(int currentDirection, int side, int steps){
  if( side == leftSide )
    currentDirection += steps;
  else if( side == rightSide )
    currentDirection -= steps;
  else return currentDirection;

  if( currentDirection < 1 )

    return currentDirection + 6;
  else if( currentDirection > 6 )
    return currentDirection - 6;
  else return currentDirection;
}

// RandomBool
boolean randomBool(){
  if( random(1,1000) > 500 )
    return false;
  else return true;
}

/* END OF PROGRAM */

```