## linha horizontal



Pioneer Autônomo com Carmen-LCAD

Documentação

12/12/2019

**─**

Anderson Mozart, Gabriel Nunes, Genilson Cruz, Leandro de Lima, Marcos Piumbini, Thiago Cavalcante

Robótica Probabilística

Profª Ph.D Claudine Badue

UFES

2019

# Visão geral

O desafio principal do projeto é realizar a integração do Pioneer com o framework Carmen - LCAD e fazer o robô Pioneer trafegar autônomo indoor com o objetivo final de, se possível, trafegar até a cantina. Um dos desejos é verificar a possibilidade do robô trafegar em caminhos diferentes do que a IARA trafega, como ambientes indoor e caminhos como passarelas e calçadas. Para que a integração seja completa iremos simular o Pioneer, que é um robô diferencial, em um robô ackermann, ou seja, iremos colocar a função que se locomove feito um robô ackermann, com as rodas dianteiras que comandam a direção e as rodas traseiras que fornecem a tração.

# Sumário

[**Visão geral**](#_au51mny0sx6) **1**

[**Sumário**](#_1m5c2755xby2) **1**

[**Especificações do projeto**](#_k0eovateshfq) **2**

[**Etapas do projeto**](#_wepdv38s0s7q) **3**

[1 - Entendendo o Pioneer](#_sddvjienpzqk) 3

[2 - Entendendo a Sensorbox](#_ujvyeg82fr0j) 4

[3 - Integrando a Sensorbox ao Pioneer](#_rt3zecb3jesi) 4

[4 - Integrando o driver do Pioneer no Github do Carmen-LCAD [3]](#_49geo8j2x1s3) 5

[5 - Transformar o movimento do Pioneer em Ackermann](#_hcwhxcvyaqir) 9

[6 - Ajustar parâmetros do Pioneer no carmen-pioneer-sensorbox.ini](#_cbzngu7nat8p) 9

[7 - Controlar o Pioneer através do Joystick](#_qhfwh15pfz55) 10

[8 - Criar process (navigate, log, mapper) para o Pioneer com a Sensorbox](#_yb9577xye8lo) 11

[9 - Gravar um Log utilizando o Pioneer e a Sensorbox](#_w4562fu7ctqz) 11

[10 - Função de calibração da odometria sem GPS](#_4xgtvsmq4x4y) 12

[11 - Utilizar o log para criar um mapa](#_yod1iy3syfto) 12

[12 - Ajustar parâmetros de PID](#_riyh9g2j3wu9) 12

[13- Fazer o pioneer se localizar e se mover autônomo em um RDDF específico](#_vonll191svkh) 13

[**Discussões e Conclusões do projeto**](#_wwouvftpkr44) **14**

[Melhorias para serem realizadas no CARMEN-LCAD para facilitar integração](#_buwz1tcz7y35) 14

[Dificuldades que tivemos](#_p2nityf5kx5q) 14

[Aprendizado com o mundo real](#_o73cbp5hbz1) 14

# Especificações do projeto

* Fazer o Pioneer, integrado à Sensorbox, se mover através da função de Ackermann;
* Ajustar parâmetros de todas as medições do robô para o pioneer no carmen-pioneer-sensorbox.ini ;
* Controlar o Pioneer através do Joystick;
* Criar process (navigate, log, mapper) para o Pioneer com a Sensorbox;
* Gravar um Log utilizando o Pioneer e a Sensorbox;
* Criar o mapa usando as funções de SLAM;
* Ajustar parâmetros do PID diferencial para que o robô trafegue de forma suave sem sair do RDDF definido;
* Trafegar autônomo utilizando as funções de robótica probabilística implementadas no CARMEN-LCAD;
* Todos os códigos-fonte devem ficar disponibilizados no GitHub do CARMEN-LCAD [3].

# Etapas do projeto

## 1 - Entendendo o Pioneer

O Pioneer 3-DX é um pequeno robô de tração diferencial leve de duas rodas e dois motores, ideal para uso em laboratório ou em sala de aula [1]. O Pioneer P3-DX é uma base robótica com rodas projetada para navegação autônoma. É a plataforma de pesquisa móvel mais popular do mundo [2]. As especificações do Pioneer [2] são:

**CARACTERÍSTICAS**

Equipado com chassi robusto e plataforma de montagem com capacidade para 17 kg.

O robô possui três baterias de 7 amperes por hora, 12 volts de corrente contínua (VDC), seladas de chumbo/ácido (total de 252 watts-hora), acessíveis através de uma porta na parte traseira.

**ALTURA**

24 cm | 9.4 in

**COMPRIMENTO**

45 cm | 17.7 in

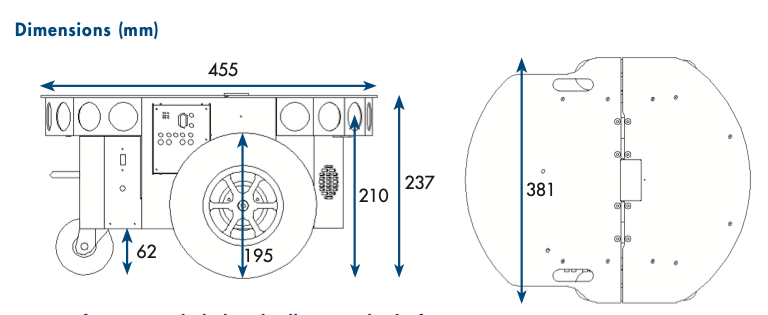
**LARGURA**

38 cm | 15 in

**PESO**

9 kg | 19.8 lb

**VELOCIDADE**

4.32 km/h | 2.7 mph

## 2 - Entendendo a Sensorbox

A Sensorbox é um conjunto de sensores em uma caixa de acrílico de aproximadamente 5 Kg que possui: uma unidade de LiDAR de 32 canais modelo Velodyne HDL-32E [6]; uma Câmera; uma IMU (Unidade de Medição Inercial); duas unidades de GPS Reach; um Roteador e um Raspberry Pi. Para se conectar na Sensorbox é utilizado um cabo RJ-45 e um cabo de alimentação.

As especificações dos IPs dos sensores da Sensorbox são:

Velodyne: 192.168.1.202 ; RaspberryPi: 192.168.1.15 ; Roteador Multilaser: 192.168.2.1 ; GPS Reach 1: 192.168.1.16:81 ; GPS Reach 2: 192.168.1.16:82.

## 3 - Integrando a Sensorbox ao Pioneer



A Sensorbox foi presa na parte superior do Pioneer através do uso de adesivos de velcro em uma posição específica. A posição definida foi escolhida de forma que o centro da Sensorbox ficasse o mais próximo possível do centro do eixo das rodas traseiras do Pioneer e o centro de gravidade do conjunto ficasse em uma região segura contra tombamento, entre a roda guia e o eixo das rodas traseiras.

O controle do Pioneer e a leitura dos dados da Sensorbox é feita por um notebook apoiado em cima da Pioneer e preso à Sensorbox com o uso de abraçadeiras de Nylon. A comunicação da Sensorbox com o notebook é feita através de um cabo Ethernet e a comunicação do notebook com o Pioneer é feita através de um cabo Serial RS232 para USB .

O Pioneer possui um conjunto de baterias internas que, além de alimentar o próprio robô, fornecem energia para o funcionamento da Sensorbox. Já o notebook é alimentado por bateria própria.

Para comunicar o computador com a Sensorbox é adicionada uma rede manual, renomeado para Sensorbox, e adicionado manualmente os seguintes IPs:

192.168.1.1 192.168.0.1 Ambos com netmask 24

## 4 - Integrando o driver do Pioneer no Github do Carmen-LCAD [3]

O framework Carmen utiliza IPC - Inter Process Communication para que os módulos se comuniquem entre si.

*O IPC fornece passagem flexível e eficiente de mensagens entre processos. Ele pode enviar e receber de forma transparente estruturas de dados complexas, incluindo listas e matrizes de tamanho variável, usando os paradigmas anônimos de "publicar / assinar" e "cliente / servidor". Uma grande variedade de idiomas e sistemas operacionais são suportados* [5].

Na versão do framework carmen 0.7.4 disponível em [4] há o driver para o Pioneer 3-DX através da comunicação da porta serial, dentro da pasta /src/base. Utilizamos esse driver e adaptamos para o framework Carmen-LCAD modificando e adicionando as seguintes linhas no arquivo **src/base/base\_main.c** para que o robô realize as funções de movimento e publique a odometria:

Adição das linhas com include do código na linha 31:

#include <carmen/base\_interface.h>

#include <carmen/base\_ackerman\_interface.h>

#include <carmen/robot\_ackerman\_interface.h>

Adição da variável global para transformá-lo em ackermann na linha 65:

static double distance\_between\_front\_and\_rear\_axles;

Para preencher essa variável, carregá-la do arquivo de parâmetros carmen-pioneer-sensorbox.ini, na linha 277 :

{"robot", "distance\_between\_front\_and\_rear\_axles", CARMEN\_PARAM\_DOUBLE,

&(distance\_between\_front\_and\_rear\_axles), 1, NULL}};

Adicionada as duas funções na linha 386, uma que irá se inscrever para receber o comando de movimento ackermann e outra para publicar a mensagem de odometria.

O handler utilizado tem o nome de base\_ackerman\_subscribe\_motion\_command\_handler, ele foi criado com base no método que já existia no código, com o nome velocity\_handler, mas foram feitas algumas modificações para funcionar com o framework Carmen-LCAD.

Foi criado o método publish\_odometry\_message no base\_main.c (necessário dar include no <carmen/robot\_ackerman\_interface.h>) e dar define nas mensagens tipo CARMEN\_ROBOT\_ACKERMAN\_VELOCITY\_NAME.

static void

base\_ackerman\_subscribe\_motion\_command\_handler(carmen\_base\_ackerman\_motion\_command\_message \*motion\_command\_message)

{

carmen\_ackerman\_motion\_command\_p vel;

vel = motion\_command\_message[0].motion\_command;

// double L = distance\_between\_front\_and\_rear\_axles; // Distancia entre eixos.

double delta\_theta;

int base\_err;

if(vel->v == 0 && vel->phi == 0)

{

if (moving)

{

do

{

base\_err = carmen\_base\_direct\_set\_deceleration(deceleration);

if (base\_err < 0)

initialize\_robot();

}

while (base\_err < 0);

moving = 0;

}

carmen\_warn("S");

}

else if (!moving)

{

moving = 1;

current\_acceleration = robot\_config.acceleration;

do

{

base\_err = carmen\_base\_direct\_set\_acceleration(current\_acceleration);

if (base\_err < 0)

initialize\_robot();

}

while (base\_err < 0);

carmen\_warn("V");

}

if (odometry\_inverted)

vel->v \*= -1;

vel->phi \*= -1; // trecho acrescentado por Gabriel Hendrix

if (!use\_hardware\_integrator)

{

vel->v /= relative\_wheelsize;

vel->phi /= relative\_wheelsize;

vel->phi /= relative\_wheelbase;

}

do

{

delta\_theta = (vel->v / distance\_between\_front\_and\_rear\_axles) \* tan(vel->phi);

base\_err = carmen\_base\_direct\_set\_velocity(vel->v, delta\_theta);

last\_motion\_command = carmen\_get\_time();

if (base\_err < 0)

initialize\_robot();

}

while (base\_err < 0);

}

void

publish\_odometry\_message(carmen\_base\_odometry\_message current\_odometry)

{

IPC\_RETURN\_TYPE err;

carmen\_robot\_ackerman\_velocity\_message odometry\_message;

odometry\_message.v = current\_odometry.tv;

//odometry\_message.phi = (odometry\_message.v/ distance\_between\_front\_and\_rear\_axles) \* tan(current\_odometry.rv);

if(odometry\_message.v == 0.0)

odometry\_message.phi = 0.0;

else

odometry\_message.phi = atan((distance\_between\_front\_and\_rear\_axles/odometry\_message.v) \* current\_odometry.rv);

odometry\_message.host = current\_odometry.host;

odometry\_message.timestamp = current\_odometry.timestamp;

err = IPC\_publishData(CARMEN\_ROBOT\_ACKERMAN\_VELOCITY\_NAME, &odometry\_message);

carmen\_test\_ipc\_exit(err, "Could not publish", CARMEN\_ROBOT\_ACKERMAN\_VELOCITY\_FMT);

}

Na pasta "base", o código que controla o pioneer é o base\_main.c, no método carmen\_base\_initialize\_ipc foi inserido o subscribe para receber a mensagem que o controle está publicando, foi utilizado o método carmen\_base\_ackerman\_subscribe\_motion\_command para isso (para utilizar esse método é necessário dar o include no carmen/base\_ackerman\_interface.h).

Foi adicionado na inicialização do IPC para definir a mensagem de velocidade linha 571:

err = IPC\_defineMsg(CARMEN\_ROBOT\_ACKERMAN\_VELOCITY\_NAME, IPC\_VARIABLE\_LENGTH,

CARMEN\_ROBOT\_ACKERMAN\_VELOCITY\_FMT);

carmen\_test\_ipc\_exit(err, "Could not define", CARMEN\_ROBOT\_ACKERMAN\_VELOCITY\_NAME);

E na linha 592 a inscrição para receber o comando de movimento:

carmen\_base\_ackerman\_subscribe\_motion\_command(NULL,

(carmen\_handler\_t) base\_ackerman\_subscribe\_motion\_command\_handler, CARMEN\_SUBSCRIBE\_LATEST);

Na função carmen\_base\_run, na linha 743, foi adicionado a publicação da odometria:

publish\_odometry\_message(odometry);

## 5 - Transformar o movimento do Pioneer em Ackermann

No base\_ackerman\_subscribe\_motion\_command\_handler foi inserido o cálculo para se obter a rotação de um veículo ackerman na linha 440, e então o resultado desse cálculo é enviado pelo método carmen\_base\_direct\_set\_velocity na linha 441 junto com a velocidade original.

delta\_theta = (vel->v / distance\_between\_front\_and\_rear\_axles) \* tan(vel->phi);

base\_err = carmen\_base\_direct\_set\_velocity(vel->v, delta\_theta);

## 6 - Ajustar parâmetros do Pioneer no carmen-pioneer-sensorbox.ini

Foi copiado o /src/carmen-ford-escape-sensorbox.ini para outro arquivo com o nome carmen-pioneer-sensorbox.ini. Nesse novo arquivo, foram inseridos os seguintes parâmetros:

base\_type

base\_dev

base\_model

base\_motion\_timeout

base\_use\_hardware\_integrator

base\_relative\_wheelsize

base\_relative\_wheelbase

robot\_odometry\_inverted

Foram modificados os seguintes parâmetros:

robot\_length

robot\_width

robot\_vertical\_displacement\_from\_center

robot\_wheel\_radius

robot\_acceleration

robot\_deceleration

robot\_distance\_between\_front\_and\_rear\_axles

Criada pasta pioneer\_p3dx em bin com o arquivo de colisão do robô. Trocado o parâmetro robot\_collision\_file e foi ligada a opção show\_collision\_range em carmen-ford-escape-sensorbox.ini, para representar o robô como um círculo no navigator\_gui.

Ao executar o process-volta\_no\_ct9\_log\_sensorbox.ini, o viewer 3d informava erro no arquivo de parâmetros carmen-pioneer-sensorbox.ini. Foram verificados e incluídas as seguintes seções no arquivo (retirado do arquivo carmen-ford-escape.ini):

-#--------- Bumblebee Basic 7 REALSENSE ------------

#--------- Stereo Velodyne 7 (Realsense)------------

#----------- Camera 7 - RealSense ------------

E foram alterados alguns parâmetros do Robot Ackermann para as medidas feitas para o pioneer. Os parâmetros alterados foram:

robot\_max\_velocity

robot\_distance\_between\_rear\_wheels

robot\_distance\_between\_rear\_car\_and\_rear\_wheels

robot\_distance\_between\_front\_car\_and\_front\_wheels

robot\_distance\_between\_rearview

robot\_turning\_radius

Também foram alterados parâmetros da sensor board:

sensor\_board\_1\_x

sensor\_board\_1\_y

sensor\_board\_1\_z

sensor\_board\_1\_yaw

sensor\_board\_1\_pitch

sensor\_board\_1\_roll

Segue Lista completa com as alterações de parâmetros:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SEÇÃO** | **PARÂMETRO** | **FORD ESCAPE** | **PIONEER** | **COMENTÁRIOS** |
| Localize Ackerman parameters | localize\_v\_uncertainty\_at\_zero\_v | N/A | 0.0 |  |
| Localize Ackerman parameters | localize\_remission\_variance\_multiplier | N/A | 2.0 |  |
| Map Server | map\_server\_initial\_waiting\_time | 3.0 | 0.0 |  |
| Map Server | map\_server\_map\_grid\_res | 0.2 | 0.05 |  |
| Map Server | map\_server\_map\_width | 210.0 | 15.0 |  |
| Map Server | map\_server\_map\_height | 210.0 | 15.0 |  |
| Map Server | map\_server\_time\_interval\_for\_map\_change | 0.3 | 0.1 |  |
| Mapper Parameters | mapper\_map\_grid\_res | 0.2 | 0.05 |  |
| Mapper Parameters | mapper\_map\_width | 210.0 | 15.0 |  |
| Mapper Parameters | mapper\_map\_height | 210.0 | 15.0 |  |
| Mapper Parameters | mapper\_mapping\_mode\_off\_velodyne\_range\_max | 70.0 | 10.0 |  |
| Mapper Parameters | mapper\_mapping\_mode\_on\_velodyne\_range\_max | 70.0 | 10.0 |  |
| Model Predictive Planner parameters | model\_predictive\_planner\_obstacles\_safe\_distance | 1.6 | 0.2 |  |
| Model Predictive Planner parameters | model\_predictive\_planner\_obstacles\_safe\_length\_distance | 5.63 | 0.47 |  |
| Model Predictive Planner parameters | model\_predictive\_planner\_max\_square\_distance\_to\_lane | 2.0 | 0.5 |  |
| Obstacle Avoider parameters | obstacle\_avoider\_obstacles\_safe\_distance | 1.0 | 0.1 |  |
| Behavior Selector parameters | behavior\_selector\_central\_lane\_obstacles\_safe\_distance | 1.6 | 0.6 |  |
| Behavior Selector parameters | behavior\_selector\_main\_central\_lane\_obstacles\_safe\_distance | 1.1 | 0.5 |  |
| Behavior Selector parameters | behavior\_selector\_lateral\_lane\_obstacles\_safe\_distance | 1.1 | 0.5 |  |
| Behavior Selector parameters | behavior\_selector\_lateral\_lane\_displacement | 2.2 | 1.0 |  |
| Behavior Selector parameters | behavior\_selector\_distance\_between\_waypoints | 13.0 | 1.5 |  |
| Behavior Selector parameters | behavior\_selector\_change\_goal\_distance | 13.0 | 1.5 |  |
| Robot parameters | base\_type | N/A | pioneer |  |
| Robot parameters | base\_dev | N/A | /dev/ttyUSB0 |  |
| Robot parameters | base\_model | N/A | p2d8+ |  |
| Robot parameters | base\_motion\_timeout | N/A | 1.0 |  |
| Robot parameters | base\_use\_hardware\_integrator | N/A | off |  |
| Robot parameters | base\_relative\_wheelsize | N/A | 1.0 |  |
| Robot parameters | base\_relative\_wheelbase | N/A | 1.0 |  |
| Robot parameters | robot\_length | 4.425 | 0.445 |  |
| Robot parameters | robot\_width | 1.806 | 0.393 |  |
| Robot parameters | robot\_vertical\_displacement\_from\_center | -1.375 | 0.0 |  |
| Robot parameters | robot\_wheel\_radius | 0.28 | 0.095 |  |
| Robot parameters | robot\_acceleration | 2.00 | 0.18 |  |
| Robot parameters | robot\_deceleration | 0.30 | 0.90 |  |
| Robot parameters | robot\_odometry\_inverted | N/A | on |  |
| Robot parameters | robot\_polygon\_file | ford\_escape / ford\_escape\_poly.txt | pioneer\_p3dx / pioneer\_p3dx\_poly.txt |  |
| Robot parameters | robot\_collision\_file | ford\_escape / ford\_escape\_col.txt | pioneer\_p3dx / pioneer\_p3dx\_col.txt |  |
| Robot Ackermann parameters | robot\_max\_steering\_angle | 0.5537 | 0.8 |  |
| Robot Ackermann parameters | robot\_max\_velocity | 8.33 | 0.4 | alterados parâmetros do Robot Ackermann para as medidas feitas para o pioneer |
| Robot Ackermann parameters | robot\_distance\_between\_front\_and\_rear\_axles | 2.625 | 0.4 | alterados parâmetros do Robot Ackermann para as medidas feitas para o pioneer |
| Robot Ackermann parameters | robot\_distance\_between\_rear\_wheels | 1.535 | 0.33 | alterados parâmetros do Robot Ackermann para as medidas feitas para o pioneer |
| Robot Ackermann parameters | robot\_distance\_between\_rear\_car\_and\_rear\_wheels | 0.96 | 0.19 | alterados parâmetros do Robot Ackermann para as medidas feitas para o pioneer |
| Robot Ackermann parameters | robot\_distance\_between\_front\_car\_and\_front\_wheels | 0.85 | 0.09 | alterados parâmetros do Robot Ackermann para as medidas feitas para o pioneer |
| Robot Ackermann parameters | robot\_distance\_between\_rearview | 2.065 | 0.4 | alterados parâmetros do Robot Ackermann para as medidas feitas para o pioneer |
| Robot Ackermann parameters | robot\_turning\_radius | 5.6 | 3.0 | alterados parâmetros do Robot Ackermann para as medidas feitas para o pioneer |
| Robot Ackermann parameters | robot\_maximum\_steering\_command\_rate | 0.335 | 1.0 |  |
| Robot Ackermann parameters | robot\_understeer\_coeficient | 0.0015 | 1.0 |  |
| Robot Ackermann parameters | robot\_maximum\_speed\_forward | 46.0 | 1.5 |  |
| Robot Ackermann parameters | robot\_maximum\_speed\_reverse | 20.0 | 1.0 |  |
| Robot Ackermann parameters | robot\_v\_multiplier | 1.015018 | 1.0 | Ajuste da odometria do carro |
| Robot Ackermann parameters | robot\_phi\_multiplier | 1.096677 | 1.0 | Ajuste da odometria do carro |
| Robot Ackermann parameters | robot\_PID\_steering\_kp | 689.4 | 1000.0 | Ajuste da odometria do carro |
| Robot Ackermann parameters | robot\_PID\_steering\_ki | 2008.7 | 2000.0 | Ajuste da odometria do carro |
| Robot Ackermann parameters | robot\_PID\_steering\_kd | 30.8 | 90.0 | Ajuste da odometria do carro |
| Grid Mapping Map parameters | grid\_mapping\_map\_grid\_res | 0.2 | 0.05 |  |
| Grid Mapping Map parameters | grid\_mapping\_map\_width | 210 | 15 |  |
| Grid Mapping Map parameters | grid\_mapping\_map\_height | 210 | 15 |  |
| Sensor Board 1 parameters | sensor\_board\_1\_x | 3.40 | 0.025 |  |
| Sensor Board 1 parameters | sensor\_board\_1\_z | 0.737 | 0.14 |  |
| Velodyne Variable Scan Message parameters |  |  |  | Criada seção no arquivo pioneer.ini mas não achei referência a alguns do parâmetros. Devemos documentar ou manter ? Eles foram usados apenas para teste ? |
| Velodyne parameters | velodyne\_z | 0.090805 | 0.35 |  |
| Velodyne parameters | velodyne\_pitch | -0.0227 | 0.0 |  |
| Velodyne parameters | velodyne\_yaw | -0.01 | 0.0 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_width | N/A | 640 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_height | N/A | 480 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_fx | N/A | 0.605578 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_fy | N/A | 0.807437 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_cu | N/A | 0.498175 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_cv | N/A | 0.512542 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_baseline | N/A | 0.05 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_pixel\_size | N/A | 0.00000375 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_is\_legacy | N/A | off |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_is\_rectified | N/A | on |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_model | N/A | xb3 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_fov | N/A | 0.6 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_tlight\_roi\_x | N/A | 0 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_tlight\_roi\_y | N/A | 0 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_tlight\_roi\_w | N/A | 640 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_tlight\_roi\_h | N/A | 480 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_tlight\_focal\_dist | N/A | 1500.0 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_tlight\_dist\_correction | N/A | 6.0 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_stereo\_stride\_x | N/A | 10.0 |  |
| Bumblebee Basic 7 REALSENSE | bumblebee\_basic7\_stereo\_stride\_y | N/A | 10.0 |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_num\_points\_cloud | N/A | 1 |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_flipped off | N/A | off |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_horizontal\_resolution | N/A | 640 |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_vertical\_resolution | N/A | 480 |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_range\_max | N/A | 2.0 |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_horizontal\_roi\_ini | N/A | 0 |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_horizontal\_roi\_end | N/A | 640 |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_vertical\_roi\_ini | N/A | 0 |  |
| Stereo Velodyne 7 (Realsense) | stereo\_velodyne7\_vertical\_roi\_end | N/A | 480 |  |
| Camera 7 – RealSense | camera7\_x | N/A | 3.0 |  |
| Camera 7 – RealSense | camera7\_y | N/A | 0.0 |  |
| Camera 7 – RealSense | camera7\_z | N/A | -0.9 |  |
| Camera 7 – RealSense | camera7\_roll | N/A | 0 |  |
| Camera 7 – RealSense | camera7\_pitch | N/A | 2.09 |  |
| Camera 7 – RealSense | camera7\_yaw | N/A | 3.14 |  |
| Viewer\_3D parameters | carmodel\_x | 1.2 | 0.09 | Parâmetros alterados para corrigir erro ao rodar módulo. |
| Viewer\_3D parameters | carmodel\_z | 0.28 | -0.28 | Parâmetros alterados para corrigir erro ao rodar módulo. |
| Viewer\_3D parameters | carmodel\_size\_x | 4.437 | 0.43 | Parâmetros alterados para corrigir erro ao rodar módulo. |
| Viewer\_3D parameters | carmodel\_size\_y | 1.806 | 0.38 | Parâmetros alterados para corrigir erro ao rodar módulo. |
| Viewer\_3D parameters | carmodel\_size\_z | 1.725 | 0.24 | Parâmetros alterados para corrigir erro ao rodar módulo. |
| Laser position and orientation in relation to car | xsens\_magnetic\_declination | N/A | -23.457849 |  |
| Logger parameters | logger\_imu\_pi | on | off |  |
| Logger parameters | logger\_xsens\_mtig | off | on |  |
| Logger parameters | logger\_ford\_escape\_status | on | off |  |

## 

## 7 - Controlar o Pioneer através do Joystick

Modificamos o código ~/carmen\_lcad/src/joystick/wingman\_control.c para que o método send\_base\_velocity\_command fosse capaz de enviar uma mensagem do tipo carmen\_base\_ackerman\_velocity\_message e realizar a publicação dessa mensagem. Para utilizar esse método, é necessário dar o include do carmen/obstacle\_avoider\_interface.h. Também foi modificado o método define\_messages para que seja definido as mensagens do tipo CARMEN\_BASE\_ACKERMAN\_MOTION\_COMMAND\_NAME. Também foram comentadas algumas seções do método main para que a opção "throttle\_mode" não fosse utilizada, já que não será necessária para a nossa aplicação e a ativação do modo acelerador estava causando bugs no funcionamento.

## 8 - Criar process (navigate, log, mapper) para o Pioneer com a Sensorbox

Foi criado o process-pioneer\_sensorbox\_playback\_viewer\_3D.ini e ao executá-lo, foi necessário incluir no carmen-pioneer-sensorbox.ini os seguintes parâmetros:

localize\_v\_uncertainty\_at\_zero\_v

localize\_remission\_variance\_multiplier

Foi criado o process-volta\_no\_ct9\_log\_sensorbox.ini para gravação do Log.

## 9 - Gravar um Log utilizando o Pioneer e a Sensorbox

Foi executado o process-volta\_no\_ct9\_log\_sensorbox.ini para realizar a gravação do log. Para a câmera ser gravada foi necessário alterar um parâmetro logger save on disk para “on” e ativar a câmera da sensorbox através do ssh.

Foi verificado que o log foi gravado com sucesso ao executar o process-pioneer\_sensorbox\_playback\_viewer\_3D.ini.

Para a criação do log, o controle do robô foi feito através de um operador humano usando um Joystick. O percurso inicial realizado foi do corredor do CT-9 próximo à porta do LCAD até o outro lado do corredor, conforme sequência de imagens abaixo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

## 10 - Função de calibração da odometria sem GPS

Foi criado um módulo [src](https://github.com/LCAD-UFES/carmen_lcad/tree/master/src)/odometry\_calibration\_nogps/ para realizar a calibração da odometria sem utilizar GPS.

Como calibrar a odometria:

1. Compile o módulo odometry\_calibration

2. Vá para o diretório bin: cd $CARMEN\_HOME/bin

3. Execute o programa:

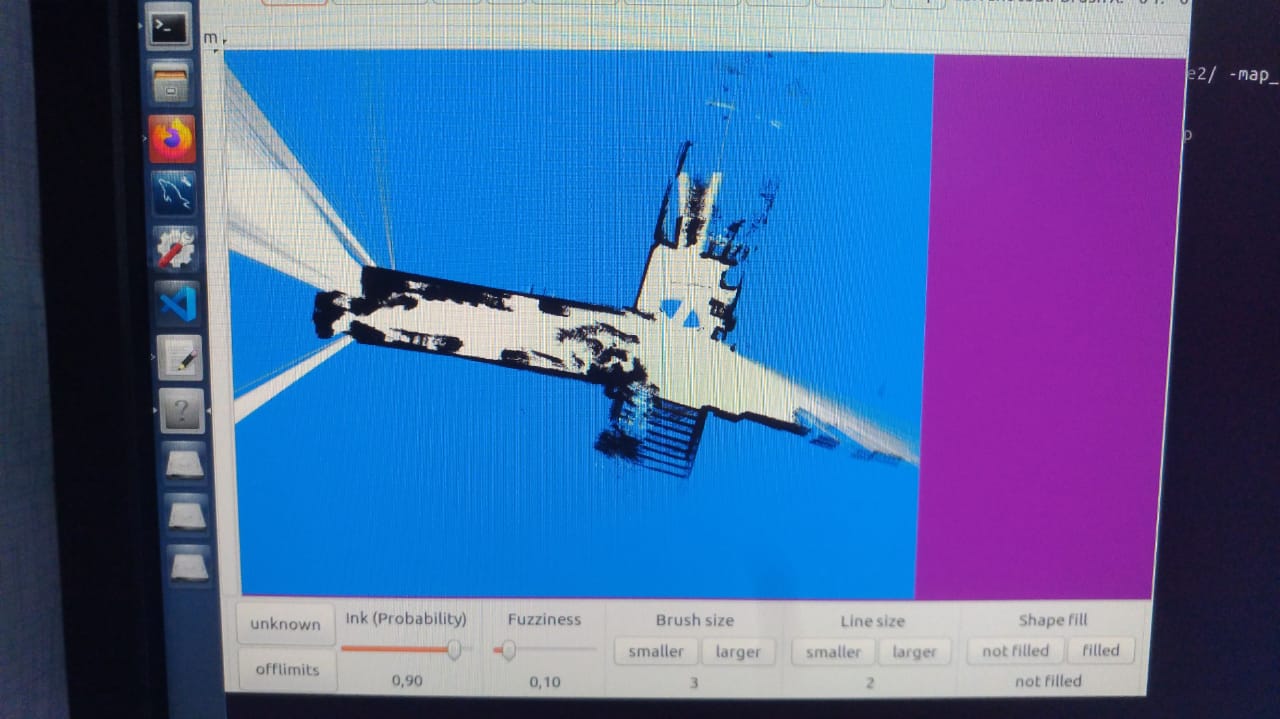
./calibrate\_bias\_from\_log --gps\_to\_use 2 --use\_non\_linear\_phi 0 -i 300 /dados/log\_volta-do-ct9.txt ../src/carmen-pioneer-sensorbox.ini caco.txt poses.txt poses\_opt.txt

4. Pode-se usar o poses\_opt.txt para fazer mapas sem graphslam! Basta usá-lo como se fosse a saída do graphslam para o log. Para isso, no process de fazer mapas, basta usar a linha do graphslam\_publish como abaixo:

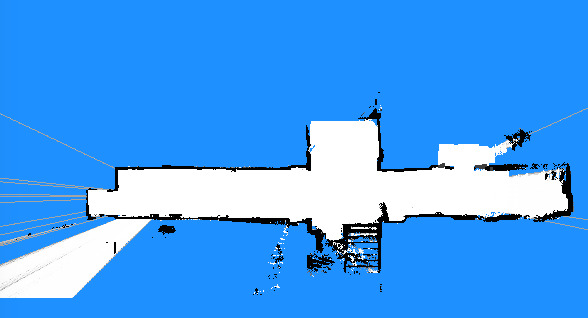
Publish\_poses graphslam 1 0 ./graphslam\_publish ../src/odometry\_calibration\_nogps/poses\_opt.txt

## 11 - Utilizar o log para criar um mapa

Utilizamos o process-pioneer\_sensorbox\_playback\_viewer\_3D\_map\_generation.ini para ajudar a criar o mapa. Depois fizemos a limpeza do mapa utilizando o map\_editor. As poses do robô foram utilizadas para fazer o RDDF.

Mapa antes da limpeza, com todos os obstáculos:

Mapa após a limpeza utilizando a ferramenta map\_editor:



## 12 - Ajustar parâmetros de PID

Ajustamos no carmen-pioneer-sensorbox.ini os parâmetros de PID para que os valores de integral e diferencial e correção ficassem os mais suaves possíveis para o Pioneer.

robot\_PID\_steering\_kp 2000.0

robot\_PID\_steering\_ki 100.0

robot\_PID\_steering\_kd 0.0

robot\_PID\_minimum\_delta\_velocity 0.2

robot\_PID\_velocity\_forward\_accelerating\_Kp 25.0 # 10.0

robot\_PID\_velocity\_forward\_accelerating\_Ki 6.0 # 3.0 # 0.35 #10.0

robot\_PID\_velocity\_forward\_accelerating\_Kd -0.2 # -0.025 #-0.5

robot\_PID\_velocity\_forward\_deccelerating\_Kp 30.0 # 20.0

robot\_PID\_velocity\_forward\_deccelerating\_Ki 2.0 # 9.0 # 9.0 # 0.2 #10.0

robot\_PID\_velocity\_forward\_deccelerating\_Kd -0.2 #-0.025 #-1.0

robot\_PID\_velocity\_backward\_accelerating\_Kp 10.0

robot\_PID\_velocity\_backward\_accelerating\_Ki 3.0 # 0.35 #10.0

robot\_PID\_velocity\_backward\_accelerating\_Kd 0.0 #-0.025 #-0.5

robot\_PID\_velocity\_backward\_deccelerating\_Kp 20.0

robot\_PID\_velocity\_backward\_deccelerating\_Ki 9.0 # 0.2 #10.0

robot\_PID\_velocity\_backward\_deccelerating\_Kd 0.0 #-0.025 #-1.0

robot\_PID\_velocity\_brake\_gap 17.0

## 13- Fazer o pioneer se localizar e se mover autônomo em um RDDF específico

Utilizamos o process-volta\_no\_ct9\_log\_sensorbox.ini para fazer o pioneer se locomover autônomo. Colocamos o pioneer no início do corredor, e com o RDDF definido ele trafegou sem colidir com nenhum obstáculo. Verificamos também que, quando há um obstáculo no caminho, ele para imediatamente conforme o esperado.

# Discussões e Conclusões do projeto

## Melhorias para serem realizadas no CARMEN-LCAD para facilitar integração

Verificamos que seria bom se, no arquivo de parâmetros, ao lado de cada parâmetro estivesse comentário sobre para o que o parâmetro é utilizado e qual medida reflete o valor desse parâmetro. Verificamos que há parâmetros duplicados, mas dado que são mais de 2 milhões de linhas de código é algo normal de ocorrer. Propomos uma interface para integração de um novo robô com o carmen que seja estilo um Wizard, em que o usuário insere as medidas do robô, as medidas da sensorbox em relação ao eixo traseiro do robô, todas as medidas necessárias, e com isso o arquivo carmen-robô-sensorbox.ini seria gerado automaticamente. Com isso a integração poderia ser feita de forma mais ágil em qualquer veículo, e também pode ter um apelo comercial maior.

## Dificuldades que tivemos

A maior dificuldade do trabalho foi sair do zero e entender o que deve ser integrado no código do pioneer. Depois que descobrimos como inserir a função de movimento com IPC para o movimento do robô, conseguimos caminhar na implementação para utilizar o robô com o joystick. Outra dificuldade que deu muito trabalho foi implementar a função de criar o mapa, visto que a função original da IARA utiliza o GPS para calibrar a odometria, e no nosso robô não há GPS disponível no ambiente indoor dado às restrições do sensor GPS.

## Aprendizado com o mundo real

Após superadas as dificuldades, verificamos que o mundo real é um ambiente em que se deve ajustar parâmetros e verificar o comportamento do robô a todo momento. Alteramos os parâmetros de PID de forma empírica e chegamos em uma configuração adequada para o Pioneer, fazendo-o trafegar autônomo dentro do corredor do CT. Acreditamos que com esse conhecimento podemos entender e integrar outros veículos com o Carmen-LCAD.

**Referências:**

1 - Pioneer 3-DX specifications. <https://www.generationrobots.com/media/Pioneer3DX-P3DX-RevA.pdf> . Acessado em 26/11/2019.

2 - Pioneer 3 - ROBOTS - YOUR GUIDE TO THE WORLD OF ROBOTICS. <https://robots.ieee.org/robots/pioneer/> . Acessado em 26/11/2019.

3 - GitHub - CARMEN-LCAD. <https://github.com/LCAD-UFES/carmen_lcad> Códigos-fonte disponíveis.

4 - CARMEN 0.7.4-beta <http://carmen.sourceforge.net/getting_carmen.html>

5 - Inter Process Communication (IPC). <http://www.cs.cmu.edu/~ipc/> . Acessado em 26/11/2019.

6 - HDL-32E Velodyne. <https://velodynelidar.com/hdl-32e.html> . Acessado em 27/11/2019.