

Navi++ 机器人配置手册

北京布科思科技有限公司

Beijing Boocax Technology Co.Ltd.

地址：北京市海淀区清河永泰园甲 1 号建金中心
606

网址：www.boocax.com

Version 2.0.0

Updated Date 2019.9.3

机器人参数配置

机器人配置文件： /home/boocax/data/config.dat

#为行注释符号

每项配置形式为 topic = content

topic 全部小写

该文档针对 navi++ 1.3.1 以上版本

详细配置内容

一、机器人本体设置：

1.机器人轮廓设置：

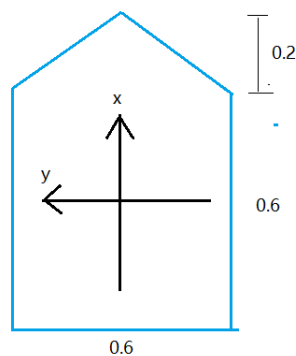
`robot_radius = 0.3`

`footprint = [[0.25,-0.25],[0.25,0.25],[-0.25,0.25],[-0.25,-0.25]]`

上面两个参数只需要设置其中一个，另外一个注释，分别表示圆形机器人的半径和非圆形机器人的轮廓。

设置机器人的轮廓时，以机器人运动中心为原点，朝前为 x 轴正方向，左为 y 轴正方向，各个顶点按照逆时针方向提供。如下图轮廓的机器人，footprint 配置为：

`footprint=[[0.5,0],[0.3,0.3],[-0.3,0.3],[-0.3,-0.3],[0.3,-0.3]]`



内部障碍物清理轮廓：

`pole_distance = 0.3`

`pole_footprint = [[0.5,0],[0.3,0.3],[-0.3,0.3],[-0.3,-0.3],[0.3,-0.3]]`

上面两个参数只需要设置其中一个，另外一个注释，分别表示圆形机器人的清理半径和非圆形机器人的清理轮廓。

2.机器人轴距设置：

`width_wheel = 0.3`

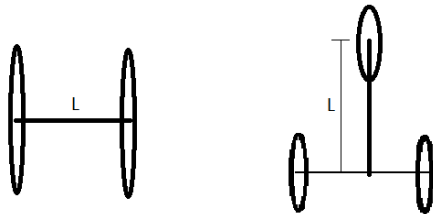
计算机器人的中心速度时要用到该值。

针对双轮差速机器人和 car-like 机器人取值有所不同。

双轮差速机器人表示两个轮子之间的距离。

Car-like 机器人表示主动轮和从动轮之间的距离。

图示如下，左边为双轮差速，右边为带方向盘的三轮车，由前轮驱动。



3. 机器人主动轮半径设置:

`radius_wheel = 0.3`

轴距和轮半径设置尽量要精确，影响机器人速度控制的精确性，进一步影响导航性能。这两个参数会通过底盘协议传输给下位机。

4. 机器人运动轮偏置设置:

`pre_robot_radius = 0.3`

机器人等似半径(以几何中心计算的外围轮廓半径)

`front_wheel = 0.122`

运动中心相对与几何中心的偏置

对于运动中心与几何中心不重合的模型的设置方法:以运动中心为中心点，设置激光、机器人轮廓等参数。在此基础上，设置等似半径和运动中心偏置距离

二、运动底盘设置

1. 底盘串口设置:

`base_baudrate = 9600`

`base_port = /dev/ttyS2`

底盘连接串口波特率和串口号，缺省接 COM3

2. 底盘通信协议

`base_link_format = 0`

0: 布科思可打印字符串协议

1: 康力优岚底盘协议

2: 瑞曼底盘协议

4: 布科思二进制底盘 1.0 协议

10: 凌科斯底盘协议

11: 布科思二进制底盘 2.0 协议

3. 电机减速比设置:

`reduction_gear_ratio = 1`

电机减速比将通过底盘协议传给下位机。

4. 真实里程误差

`odom_pos_var = 0.4` 位置误差百分比

`odom_theta_var = 0.6` 角度误差百分比

三、传感器设置

1. 传感器触发模式

`sensor_mode = 0`

0: 瞬间触发

1: 保持触发

开关量传感器例如碰撞防跌落等一旦触发, 在瞬间触发模式下当传感器状态恢复回非触发方式时则小车可以继续运动, 而在保持触发模式下, 只有断电重启或发送重置传感器状态后小车才可以运动。

2. 超声波设置

`have_sonar = 0`

已安装超声波数量, 缺省 0 个, 最多支持 6 个

`sonar_mode = 0`

超声波传感器类型, 0: 普通超声波 1: 防水超声波

`sonar0 = 315,0.3,0.2,2.5,15`

每个超声波的具体内容设置。

超声波按顺序名称分别为 sonar0, sonar1, …… sonar5

= 后面共 6 项内容, 用逗号隔开, 每项的含义依次为:

安装角度: 相对于机器人中心朝前方向的偏离角度, 逆时针顺序, 单位 度 (0-360 度)

安装距离: 超声波离机器人中心的距离, 单位 米

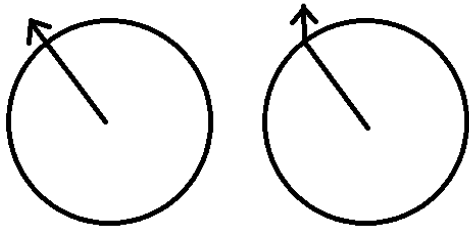
超声波读数最小距离: 单位(米)

超声波读数最大距离: 单位(米)

超声波探测视角: 单位 (度) 0-180 度。

超声波发射方向: 相对于机器人中心朝前方向的偏离角度。如果不设置该项缺省和安装角度相同。

安装位置角度和发射方向的区别, 下面的示意图中, 左图安装角度和发射方向都是 45 度, 右图安装角度 45 度, 发射方向是 0 度 (假设机器人方向朝上)



`sonar_fliter = 0`

是否开启超声波软件滤波

`sonar_zero_inver = 2.0`

如果开启软件滤波，那么数值 0 代表的探测最大值是多少

3. 红外传感器设置

`have_ir = 0`

已安装红外数量，缺省 0 个，最多支持 6 个

`ir0 = 315,0.3,0.2,2.5,15`

每个红外的具体内容设置与超声波相同。

4. 激光安装位置：

`laser_tf = 0, 0, 0, 0, 0, 0`

激光相对于机器人中心的坐标变换，前三位分别表示 x, y, z 轴的偏移量，单位米，后三位表示角度变换 roll, pitch, yaw，单位弧度。

注意：后三位 roll, pitch, yaw 要全部设置为 0，激光如果有旋转要在 laser.launch 中建立旋转坐标系

以激光倒装为例：

```
<node pkg="tf" type="static_transform_publisher" name="under_laser_transform" args="0 0 0 0 0 0 3.142 laser under_laser 100" />
```

并且此时将激光驱动发送出来的数据的话题名称要修改成除 /scan 外任何一个名字

将激光驱动发送出来的数据的 frame_id 也要改成除 /laser 外的任何一个名字

以下以西科激光为例：

```
<!-- Sick 雷达 -->
```

```
<node name="sick_tim551_2050001" pkg="sick_tim" type="sick_tim551_2050001" respawn="false" output="screen">
```

```
  <param name="min_ang" type="double" value="-2.35619449019" />
```

```
  <param name="max_ang" type="double" value="2.35619449019" />
```

```
  <param name="intensity" type="bool" value="false" />
```

```
  <param name="skip" type="int" value="0" />
```

```
  <param name="frame_id" type="str" value="under_laser" />
```

```
  <param name="time_offset" type="double" value="-0.001" />
```

```
  <param name="publish_datagram" type="bool" value="False" />
```

```
  <param name="subscribe_datagram" type="bool" value="false" />
```

```
  <param name="time_increment" type="double" value="0.000061722" />
```

```

    <param name="range_min" type="double" value="0.05" />
    <param name="range_max" type="double" value="10.0" />
    <param name="hostname" type="string" value="192.168.10.10" />
    <param name="port" type="string" value="2112" />
    <param name="timelimit" type="int" value="5" />
    <remap from="/scan" to="/under_scan" />
  </node>

```

然后配置激光融合节点的 launch 文件:

```

<launch>
  <node          name="multiple_laser_fusion_node"          pkg="multiple_laser_fusion"
type="multiple_laser_fusion" respawn="false" output="screen">
    <param name="frame_id" type="string" value="laser" />(输出数据的坐标系)
    <param name="input_topics" type="string" value="/under_scan" />
    <param name="output_topic" type="string" value="/scan" />
    <param name="angle_min" type="double" value="-2.35619449019" />
    <param name="angle_max" type="double" value="2.35619449019" />
    <param name="angle_increment" type="double" value="0.005236" />
    <param name="time_increment" type="double" value="0.000061722" />
    <param name="scan_time" type="double" value="0.05" />
    <param name="range_min" type="double" value="0.05" />
    <param name="range_max" type="double" value="10.0" />
  </node>
</launch>

```

5. 激光滤波设置

```
scan_filtered = 1
```

是否开启激光滤波，sick 等 270 度激光需要开启滤波
高级设置请参考附录 1

6. Uwb 串口设置

```
uwb_port = /dev/ttyS3
```

```
uwb_baudrate = 9600
```

uwb 缺省使用 COM4

7. Uwb 标签设置

```
uwb_tag =
```

每个 uwb 标签有一个 id 号，如果 id 号相同，会互相干扰。布科思早期生产的 uwb 标签 id 号都内置为 1，因此在一个空间有多台机器人同时运行时，需要设置该参数，自动调整 id。只有一台机器人时不需要设置。缺省为空，不设置。

设置内容格式： id 1 1 空格分开，后面两个 1 不能省略。

8. 里程计设置

```
have_odom = 1
```

0 表示不提供里程计，需要用户从其它程序输出里程计信息

- 1 表示提供虚拟里程计，用输出的速度当作实际速度
- 2 表示通过底盘提供真实里程计（例如通过实际的码盘信息获得）

`nav_odom = 0`

是否强制启动导航里程计【虚拟里程计】

9. 电池参数设置

`vol_max = 29.1`

`vol_min = 21.0`

电池最大电压和最小电压，主要用来计算电池剩余电量。如果底盘通过 obd 上报电池电量，该参数将不起作用。

10. 碰撞传感器设置

`have_collision = 1`

碰撞传感器最多支持 6 个

`collision0 = x, y, z, roll, pitch, yaw`

每个传感器的具体内容设置。

碰撞传感器按顺序名称分别为 collision0, collision1, …… collision5

= 后面共 6 项内容，用逗号隔开，每项的含义依次为：

以机器人底盘中心为原点，正前方为 x 轴，正左方为 y 轴，正上方为 z 轴。

xyz 表示传感器的中心在坐标系中的位置。单位：米

roll,pitch,yaw 表示传感器的姿态。单位：弧度

11. 防跌落传感器设置

`have_cliff = 4`

`cliff_mode = 0`

`cliff_distance = 11`

`cliff0 = x, y, z, roll, pitch, yaw`

防跌落传感器最多支持 4 个,cliff_mode 代表传感器型号

cliff_mode = 0: 跌落检测距离(cliff_distance)可以设置范围是 0-15cm

cliff_mode = 1: 跌落检测距离(cliff_distance)可以设置范围是 15-50cm

每个传感器的位姿内容设置与碰撞传感器相同。

12. 地磁传感器设置

`have_magnetic = 4`

`magnetic0 = x, y, z, roll, pitch, yaw`

地磁传感器最多支持 4 个

每个传感器的位姿内容设置与碰撞传感器相同。

13. 陀螺仪配置

`imu_flag = 0`

* 0 不使用 imu

* 1 布科思工控机外接 imu 需要一同设置 imu 端口 `imu_sensor_port = /dev/ttyUSB0`

* 2 米文动力 imu

* 3 内置 imu 机器 此模式只要保证将陀螺仪数据以 `sensor_msgs::Imu` 消息类型，topic 名

称为/imu/data 发布出来即可

* 4 由布科思底盘(BooBase)提供的 imu 数据

imu_pose = x, y, z, roll, pitch, yaw

陀螺仪的位姿内容设置与碰撞传感器相同。

14.AOA 转台配置

have_aoa = 1 是否存在 aoa 转台 0-无 1-有 转台的有无同时决定是否启用跟随功能。

aoa_port = /dev/ttyUSB0

controller_high = 0.7 遥控器与转台之间的高度(m)

aoa_pose = x, y, z, roll, pitch, yaw

转台的位姿内容设置与碰撞传感器相同。

15.视觉传感器配置

have_visual = 1 1 使用视觉传感器 0 不使用

16.串口屏配置

net_name = eth1 网卡名称

agv_screen_serial = /dev/ttyS1 串口屏连接的串口号配置

17.充电模式配置

charge_mode = 1 (0 红外对接 1 无线充电二维码对接充电 2 二维码对接的充电头充电)

dist_to_qr_code = 0.10 (对接完成时摄像头距离二维码的距离 只需要在无线充电时配置)

charge_camera_frame = usb_cam0 (二维码对接时相机的坐标系名称)

18.安全模式

safety_node = 1 (0 不启用安全节点 1 启用安全节点)

四、构图设置

在长走廊构图时回环特征比较少容易导致通道变歪，可以使用二维码辅助构图；

在/home/boocax/data/close_loop_data.json 中配置绑定的二维码关系即可，需要成对配置，配置好之后，成对的二维码在互相检测到后可以做回环并且每一个二维码可以在第二次再次看到自身的同时做自身回环；文件格式如下：

```
[
  {
    "begin_id": 1, (二维码的 ID)
    "end_id": 3,
    "x": -0.33, (begin_id 到 end_id 的坐标变换关系，每个二维码遵循左手坐标系)
    "y": 0,
    "roll": 0,
    "pitch": 0,
    "yaw": 3.142
  },
  {
```



```
"begin_id" : 7,  
"end_id" : 6,  
"x" : -0.225,  
"y" : 0,  
"roll" : 0,  
"pitch" : 0,  
"yaw" : 3.142  
}  
]
```

五、定位配置

1. 定位设置

use_odom = 1 是否使用在定位时使用 odom 数据, 1 使用, 0 不使用, 默认使用

odomTf = 1 设置定位时激光的位姿变换的位置, 1 在外部调用处, 0 在 slam 库

default_pos_start = 0.2, 0.3, 0 设置机器人启动的默认位置

2. 是否使用激光进行局部定位

use_lidar = 0 (1 使用 0 不使用 默认使用)

在设置为不适用激光局部定位时, 在使用激光全局匹配后, 不在使用激光进行定位, 全部使用里程计进行局部定位

3. 定位策略选择

- 固定周期(目前 1s)使用激光 SLAM 根据里程计当前定位和方差进行定位匹配, 在激光 SLAM 匹配率达到 60%以上时认为该位置可信, 并以此位置纠正里程计位置, 此方法适用于环境经常变化且变化大的场景; (**loc_strategy = 1**)
- 固定方差大小时使用激光 SLAM 根据里程计当前定位和协方差进行定位匹配, 在激光 SLAM 匹配率达到 60%以上时认为该位置可信, 并以此位置纠正里程计位置, 此方法适用于环境经常变化且变化大的场景; (**loc_strategy = 2**)
- 使用激光 SLAM 强制纠正定位, 每个周期都根据里程计的定位位置和方差进行激光 SLAM 匹配并且以此结果强制纠正里程计定位, 此方法适用于环境与原地图相似度较高并且周围环境几乎不变的场景; (**loc_strategy = 3**)
- 优先使用激光 SLAM 定位角度纠正里程计定位角度, 每个周期都根据里程计的定位位置和方差进行激光 SLAM 匹配有以下两种情况:
激光 SLAM 匹配率大于 40%, 则使用激光 SLAM 定位角度与里程计定位角度进行融合, 纠正里程计的角度误差;
激光 SLAM 匹配率大于 60%, 使用激光 SLAM 定位纠正里程计定位;
此方法适用于环境与原地图相似度较高并且周围环境几乎不变的场景(**loc_strategy = 4**)

4. 位置校验模式

verification_mode = 2 (1 RFID 到点打卡方式 2 二维码位置校验)

5. 特定位置激光定位验证

在 slam 构图之后, 在地图上选择特征明显且对激光 slam 匹配比较有利的区域, 将选定区域

以圆形进行描述，将圆心和半径写入相关配置文件中，在机器人一旦到达该区域附近，则在该区域进行局部定位，如果匹配结果好则用此结果和当前位置的相对误差来纠正当前定位。目的为了解决定位过程中里程计的长期累积误差。

区域形状：圆形

基础数据：圆心和半径(区域中心点坐标 x,y 和半径 r，每个数据以空格分开，每组数据以换行符分开)

数据类型:double

配置文件路径：data/verificate_area.dat

示例：1.2 3.5 4.0

6.二维码定位配置

机器人在移动过程中会产生位置丢失或偏差等问题，可以使用二维码进行辅助定位；在/home/boocax/data/ar_config.txt 中配置每个二维码在地图坐标系下的位置即可，配置好之后，当机器人行至二维码处时会对机器人位置进行纠偏改正，避免机器人位置丢失问题；文件格式如下：

```
{"ar_infos":[
{"name":3,"position":{"x":0.0579,"y":4.78086,"z":-0.0429,
"qx":0.7400,"qy":0.0007,"qz":0.01308,"qw":0.67247}},
{"name":13,"position":{"x":49.0577,"y":3.79627,"z":-0.0604,
"qx":0.71916,"qy":0.0066,"qz":-0.0035,"qw":0.69479}},
{"name":17,"position":{"x":1.3036,"y":3.5865,"z":-0.0494,
"qx":-0.0494,"qy":0.01998,"qz":0.00969,"qw":0.69779}}
]}
```

其中：name 代表二维码编号，x, y, z 代表二维码在地图坐标系下的位置，qx, qy, qz, qw 代表二维码在地图坐标系下的姿态。

六、应用设置

1.socket 服务器设置：

server_ip = 127.0.0.1

服务器 ip 地址，缺省服务器内置在机器人导航主板上。

server_port = 6789

缺省端口 6789

2.app 串口设置

app_use_serial = 0

应用是否通过串口来连接，0 不连接，1 连接。串口连接只传输一些简单的命令和状态，传感器数据和文件仍然通过 socket 传输。具体见 navi++应用层串口协议文档。

app_baudrate = 9600

app_port= /dev/ttyS0

应用缺省使用 COM1

3.缺省传感器数据发送到界面的频率

`ui_laser_timer = 2.0`

激光数据发送间隔，单位秒，如果为 0 表示不发送。因为激光数据比较多，缺省间隔设置为 2 秒。

`ui_uwb_timer = 0.2`

UWB 数据发送间隔，单位秒，如果为 0 表示不发送

`ui_sonar_timer = 0.2`

超声波数据发送间隔，单位秒，如果为 0 表示不发送

4.定制数据

`send_custom_to_app = 0`

用户自定义数据，从底盘串口传输的特定数据，通过该参数控制是否通过 socket 协议上报给 app

5.网卡配置(针对 AGV 串口屏)

`net_name = wlan0` 网卡名称

七、定位导航参数设置

1.导航命令级别设置

`task_level = 1`

1: 响应所有导航命令

0: 只响应巡线相关导航命令，不响应自主规划路径的导航命令，主要用在 agv 上，防止界面误触发任意位置导航。

2.初始定位等待 UWB 时间

`wait_uwb_time = 6.0`

在初始定位时，如果检测到 3 个 UWB，会立即利用 UWB 定位。如果检测到少于 3 个 UWB，将等待该参数指定的时间。在等待时间结束后，如果检测到 1 个或 2 个 UWB，将利用 UWB 辅助定位。如果没有 UWB，将直接用激光匹配定位。

3.导航地图参数

`map_publish_frequency = 0`

地图发布频率，调试用，不建议用户修改。

`local_window = 4`

如果启用自由跟随功能，`local_window = 10`。cpu 占用率有所提高，使用较高性能工控机。导航局部地图窗口大小，单位为米，不建议用户修改。

`map_update_frequency = 1`

导航地图的刷新频率，不建议用户修改。

4.全局路径搜索参数设置

Inflation = 0.5

导航路径距离障碍物的参考距离(非强制距离)

inflation_weight = 2.0

路径距离障碍物的衰减权重，不建议用户修改

search_goal_tolerance = 0.0

允许在目标点附近搜索可行替代点的容差半径，如果目标设在障碍物里，会在这个半径内搜索可到达的替代目标点。

search_path_tolerance = 0.0

允许在起始点附近搜索可行替代点的容差半径，不建议用户修改。

path_search_patience = 0.0

允许路径搜索的时间容差，如果大于 0 允许在时间容许的范围内再多次搜索，不建议用户修改。

feasibility_check_dist = 1.0

机器人避障预检查轨迹长度

5.局部导航参数设置

control_frequency = 10

局部导航控制频率

safe_cmd_dist = 0.15

手控安全模式下的安全距离，单位米

allow_cancel_patience = 60

在导航过程中遇阻或者其它异常情况，导致长时间不动，在超过该参数设置的时间后，将结束导航任务，并上报导航失败。单位为秒

report_cancel_patience = 30

在导航过程中遇阻或者其它异常情况，导致机器人不能运动，在超过该参数设置的时间后，将上报异常状态请求客户端介入，单位为秒

6.AGV 【V1】 参数设置

agv_check_yaw = 0

AGV 巡线功能起始状态下选择轨迹时是否检查角度

back_vel = 0.3

后退行走允许最大速度

待续

7.AGV 【V2】 参数设置

k_line1 = 1

k_line2 = 1

k_line3 = 1

k_line4 = 5

角度误差收敛速度

k_arc1 = 1

k_arc2 = 1

k_arc3 = 6

k_arc4 = 5

#脱轨检测

agv_max_ang_diff = 1

最大允许偏离角度

agv_max_dist_line = 0.5

最大允许偏离距离

allow_auto_rotate = 1

是否可以原地旋转

8.拖车模型参数配置【适用于 AGV_V1 模式下的拖车】

laser_static = 1

激光与车头是否相对静止

laser_base_param = -0.11,0

激光到转轴，转轴到车头中心的距离

body_to_laser_x = 0.805

激光到车体运动中心的距离，默认 0.805

trailer_limit_phi = 1.5707

角度约束，弧度制，默认 1.5707

link_body_node = 0.7

轴到机器人本体运动中心的距离，默认 0.7

body_footprint = [[0.95,0.3],[-0.15,0.3],[-0.15,-0.3],[0.95,-0.3]]

车本体轮廓

9. 自由导航优化局部路径修正策略

mod_local_path = 1

0 局部路径不拼接修正

1 局部路径拼接修正

机器人运动学参数适配

在设置完上面的参数后，我们还需要根据机器人自身的实际情况来设置运动参数，并保证执行机构以较好的精度执行下发的速度。以此来保证机器人可以获得良好的运动效果。配置文件为：/home/boocax/data/limit.dat，形式同上。

一、运动参数设置

1.线速度设置

max_vel_x = 0.5

X 方向最大线速度(m/s)

`min_vel_x = 0.05`

X 方向最小线速度(m/s)

`acc_lim_x = 0.5`

X 方向线加速度(m/s²)

`stop_acc_x = 0.5`

X 方向线刹车加速度(m/s²), 不设置则默认和 `acc_lim_x` 保持一致

`max_vel_y = 0`

Y 方向最大线速度(m/s)

`min_vel_y = 0`

Y 方向最小线速度(m/s)

`acc_lim_y = 0.0`

Y 方向线加速度(m/s²)

(TIPS:差动类, 类汽车类等机器人没有 Y 方向的速度)

2.角速度设置

`max_rot_vel = 0.5`

最大角速度(rad/s)

`min_rot_vel = 0.15`

最小角速度(rad/s)

`acc_lim_theta = 0.5`

角加速度(rad/s²)

`stop_acc_theta = 0.5`

刹车角加速度(m/s²), 不设置则默认和 `acc_lim_theta` 保持一致

3.二阶段限速设置

`limit_max_vel_x = 0.3`

X 方向限速速度(m/s)

`limit_max_vel_theta = 0.5`

角速度限速速度(m/s)

`limit_vel_dist = 0.5`

启动限速的障碍物距离限制

4.目的地参数设置

`xy_goal_tolerance = 0.1`

目的地位置容差(米)

`yaw_goal_tolerance = 0.05`

目的地角度容差(rad)

二、关于参数的适配

运动学参数可以通过电机参数, 主动轮半径, 底盘机构等数据计算得出, 但是理论的数

据总会有一些误差，下面简述测试方法。

1.X 方向线速度测试

标记测试路径的起点和终点。通过 ssh 登录到机器人，在命令行中输入下列命令，在机器人动作的同时开始计时，在终点停止计时，同时 ctrl + c 退出命令行。测量路径长度，算出线速度。

```
rostopic pub /cmd_vel geometry_msgs/Twist -r 10 '[vx,0,0]' '[0,0,0]'
```

按照上述方法，逐次增加 vx,测量对应的线速度误差，并得到最大线速度和最小线速度，实际速度和测量的速度误差越小越好，如果误差大于 5%，建议调整执行机构。

2.Y 方向线速度测试同上，命令如下

```
rostopic pub /cmd_vel geometry_msgs/Twist -r 10 '[0,,vy,0]' '[0,0,0]'
```

3.角速度测试

标记起始位置，在命令行中输入下列命令，在机器人动作的同时开始计时，N 周后停止计时，同时 ctrl + c 退出命令行。

```
rostopic pub /cmd_vel geometry_msgs/Twist -r 10 '[0,0,0]' '[0,0,vw]'
```

逐次增加 vw 值测量对应的角速度误差，并得到最大角速度和最小角速度。如果误差大于 5%，建议调整执行机构

4.加速度的测试

在静止状态，分别输入最大线速度和角速度，观察加速到最大速度所用时间，得到加速度值。

5.直线行走测试

给定 vx 一个值，输入 1 中的命令，观察机器人偏离直线程度。如果偏差较大建议调整执行机构。

给定 vy 一个值，输入 2 中的命令，观察机器人偏离直线程度。如果偏差较大建议调整执行机构。(如果没有 Y 方向速度，跳过)

6.转弯半径测试

分别给定 vx, vw 一个值，测量机器人的转弯半径。测量半径和理论半径不应误差过大。

```
rostopic pub /cmd_vel geometry_msgs/Twist -r 10 '[vx,0,0]' '[0,0,vw]'
```