

20191118

本周问题描述	issue号	是否解决	
自动运动模型标定:1、解决测试组反馈的自动运动模型标定出现的问题	3411	2019.11.23	
自动运动模型标定:2、与web组进行联调	3562	2019.11.23	
解决起步的过程，有一个后退再前进的动作。	2983	2019.11.19	
如果发送多个轨迹点，并且轨迹点的角度变化不大，生成贝塞尔轨迹会卡死。	3126		
YG2.0运动控制开发评估	3581	2019.11.18	Y
避障专项优化	3798		

1.bit msgs 支持多个软件包匹配的方法

```
1 cmake_minimum_required(VERSION 2.8.3)
2 project(bit_msgs)
3 set (bit_msgs_VERSION_MAJOR 2)
4 set (bit_msgs_VERSION_MINOR 0)
5
6 configure_file (
7     ${PROJECT_SOURCE_DIR}/include/bit_msgs/bit_msgsConfig.h.in
8     ${PROJECT_SOURCE_DIR}/include/bit_msgs/bit_msgsConfig.h
9 )
10
11 ## Find catkin macros and libraries
12 ## if COMPONENTS list like find_package(catkin REQUIRED COMPONENTS xyz)
```

编译的时候,bit_msgsConfig.h.in会生成一个bit_msgsConfig.h

bit_msgs/bit_msgsConfig.h.in:

```
1 #define BITO_MSGS_VERSION_MAJOR @bit_msgs_VERSION_MAJOR@
2 #define BITO_MSGS_VERSION_MINOR @bit_msgs_VERSION_MINOR@
```

其它包的程序中需要检查bit_msgs包中的版本,

```
1 #include "bit_msgs/bit_msgsConfig.h"
2 ...
3 #if BITO_MSGS_VERSION_MAJOR >= 2
4 ...
5 #else
6 ...
7 #endif
```

2.自动运动模型标定:与web组进行联调

发现运动规划组的tf树有问题, 在插件中加入tf发布的部分.

3.利用qt界面做一个搜索工具: 涉及语言: cpp, qt等, redmine统计本组的工作内容

4.如果发送多个轨迹点, 并且轨迹点的角度变化不大, 生成贝塞尔轨迹会卡死。

在NSA中加入一个判断, 对轨迹点输入进行判断.

5. 动态绕障

1) yugong修改:

(1) 远程登录小车

(2) 进入参数文件

```
1 roscd navigator_ros/json_data/  
2 vim obst_mode_interface.json  
3 按i进入编辑模式, 上下左右移动光标进行修改  
4 修改参数文件后, 输入  
5 :wq  
6 于是就可以保存退出编辑.
```

(3) 修改参数文件

navigator_ros/json_data/obst_mode_interface.json

```
1 {  
2  
3     "type" : "obst_mode",  
4     "mode":[  
5         {"footprints":[  
6             { "x" : 0.05,  
7               "y" : 0,  
8               "radius" : 0.8  
9             },  
10            { "x" : -0.05,  
11              "y" : 0,  
12              "radius" : 0.8  
13            }  
14          ]  
15        }, (注释:避障模式obst_mode 0)  
16        {"footprints":[  
17            { "x" : 0.05,  
18              "y" : 0,  
19              "radius" : 0.8  
20            },  
21            { "x" : -0.05,  
22              "y" : 0,
```

```

23         "radius" : 0.8
24     }
25 ]
26 }, (注释:避障模式obst_mode 1)
27 {"footprints":[
28     { "x" : 0.05,
29       "y" : 0,
30       "radius" : 0.8
31     },
32     { "x" : -0.05,
33       "y" : -0.0,
34       "radius" : 0.8
35     }
36 ]
37 }, (注释:避障模式obst_mode 2)
38 {"footprints":[
39     { "x" : 0.05,
40       "y" : 0,
41       "radius" : 0.8
42     },
43     { "x" : -0.05,
44       "y" : 0,
45       "radius" : 0.8
46     }
47 ]
48 } (注释:避障模式obst_mode 3)
49
50 ]
51
52 }
53

```

如上有4个用footprints组成的动态避障的模式.

每个footprint由两个圆组成, 圆的组成元素包括圆心在车体的坐标系的位置(x, y),以及半径大小radius.

单机重启navigator:

```
1 roslaunch navigator_ros pv_robot_navigator_ros.launch
```

这个launch文件在单机重启的时候,会自动启动.

2) 韩信修改:

参数配置文件：修改:~/hanxin_ws/src/hanxin/hanxin/param/dynamic_obst_conf.yaml

```

1 roscd hanxin/param/
2 vim dynamic_obst_conf.yaml
3 按i进入编辑模式, 上下左右移动光标进行修改
4 修改参数文件后, 输入
5 :wq
6 于是就可以保存退出编辑.

```

(1) 参数配置

配置tag避障属性, 这里的文档表示经过roadmap_node_id, 避障模式会切换为obst_mode.

```
1 tag_obst:
2   -
3     roadmap_node_id: 8 # 需调节避障距离对应的路网节点ID
4     obst_mode: 3 # 避障模式
5   -
6     roadmap_node_id: 9
7     obst_mode: 1
8   -
9     roadmap_node_id: 10
10    obst_mode: 2
11  -
12    roadmap_node_id: 11
13    obst_mode: 4
```

(2) 上述描述的详细解释

上述内容表示：小车在到达（经过）路网节点id=8时，避障模式会切换为模式3

在到达（经过）路网节点id=9时，避障模式会切换为模式1

在到达（经过）路网节点id=10时，避障模式会切换为模式2

在到达（经过）路网节点id=11时，避障模式会切换为模式4

1.初始化时，小车如果没在上述已配置的路网节点上，则小车会采用默认避障模式

2.小车在到达（经过）一个配置好的路网节点后，会一直采用对应的避障模式，直到小车到达（经过）其他配置好的节点

举例：采用上述配置时，小车在到达（经过）路网节点id=8后，会一直采用避障模式3，直到小车到达（经过）路网节点id=9或10或11，小车才会切换到避障模式1或2或4

(3)运行下面的launch文件

```
1 roslaunch hanxin node_dynamic_obst.launch
```

这个模式需要手动启动, 没有加入开机自启中.