

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**LẬP TRÌNH ROBOT VỚI ROS**  
**BÁO CÁO GIỮA KÌ XÂY DỰNG MÔ PHỎNG VÀ ĐIỀU KHIỂN**  
**ROBOT**

**Họ và tên:** Nguyễn Huy Thắng

**Lớp:** K67E-RE

**Mã lớp:** RBE3007-1

## I. Giới thiệu dự án

### 1. Mục tiêu dự án

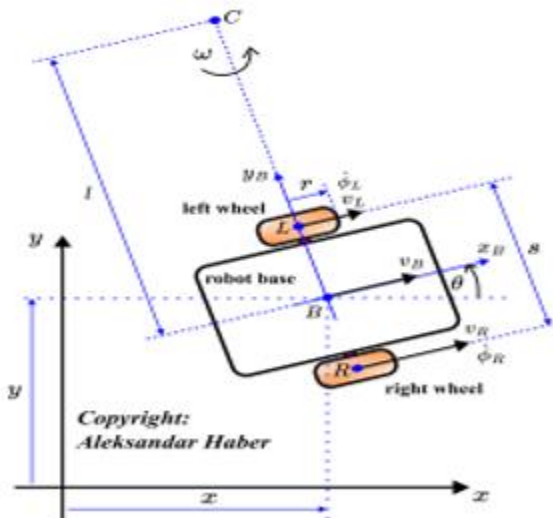
Dự án thiết kế robot hai bánh vi sai (Differential drive robot) với tay máy hai khớp quay (Rotation) sử dụng lidar, camera và imu để phát hiện vật thể, nhận diện vật thể và điều khiển vận tốc gia tốc của robot. Ngoài ra robot còn tích hợp tay máy hai khớp quay có thể di chuyển linh hoạt

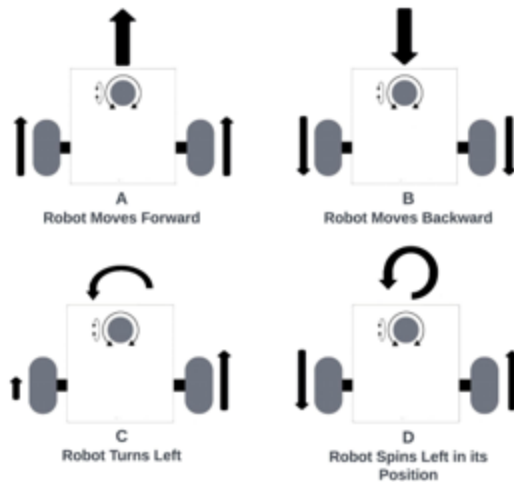
### 2. Yêu cầu đặt ra:

- Có mô hình vẽ 3D Solidworks
- Xuất được file URDF/Xacro và chạy tốt trong ROS
- Các cảm biến hoạt động tốt
- Mô phỏng điều khiển robot vi sai và tay máy di chuyển

### 3. Dạng robot, động học và kích thước dự đoán

- Dạng robot: Robot vi sai có hình dạng hình chữ nhật có hai bánh vi sai kết hợp một bánh đa hướng giúp xe di chuyển linh hoạt
- Động học





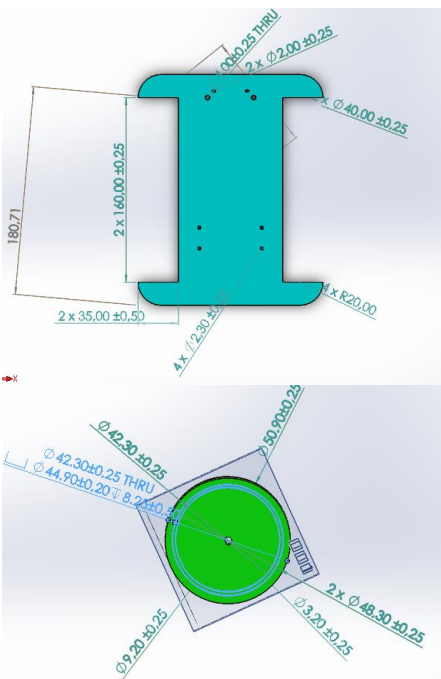
- Khi robot di chuyển cùng tốc độ ở hai bánh robot sẽ tiến hoặc lùi
- Khi vận tốc bánh phải lớn hơn vận tốc bánh trái robot sẽ rẽ trái và ngược lại
- Khi hai bánh di chuyển cùng vận tốc nhưng ngược chiều robot sẽ quay tròn tại một vị trí

#### c. Kích thước dự đoán

- Khung xe có kích thước 200x250 mm có thể chứa lidar, camera và imu.
- Bánh xe với kích thước dự đoán với bán kính 25-30mm.
- Tay máy sẽ được đặt ở tầng thứ 2 gồm: trụ tay máy liên kết khớp thứ nhất và liên kết khớp thứ hai với kích thước dự đoán khoảng 100mm

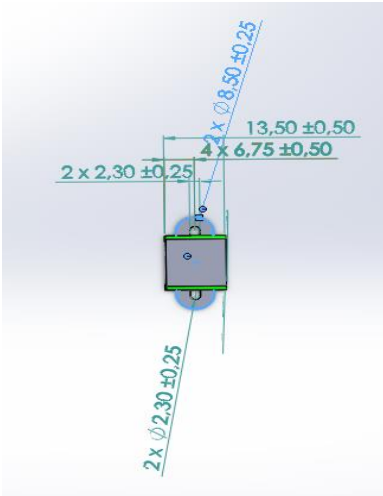
## II. Thiết kế Solidworks và đặt các trục tọa độ

### 1. Thiết kế các bộ phận trên Solidworks

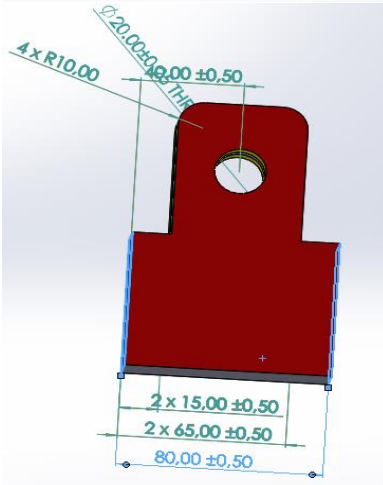


Thân xe

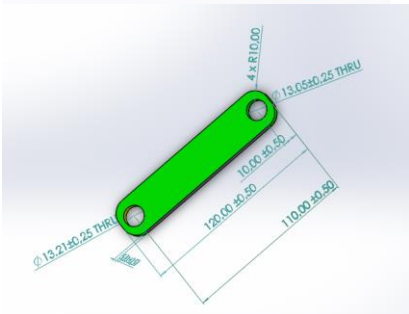
Bánh xe



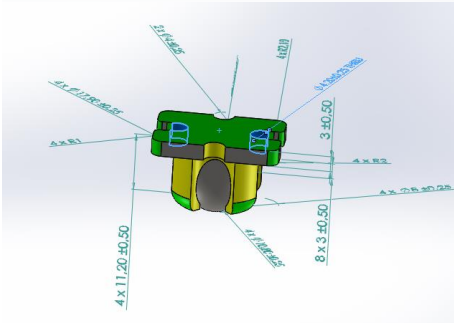
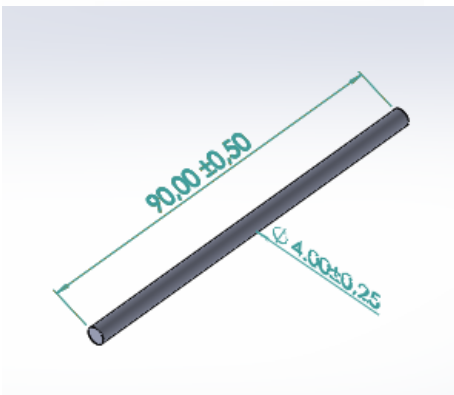
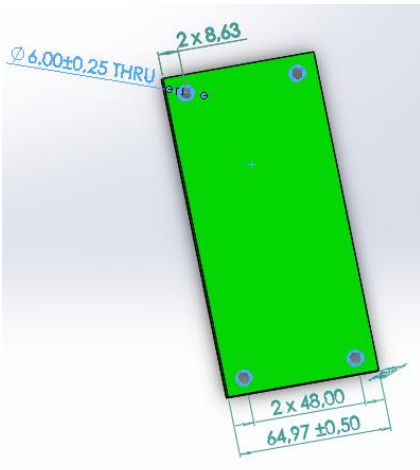
Trụ đỡ motor



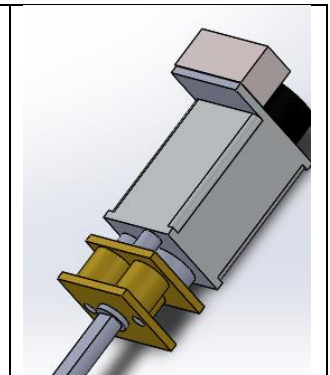
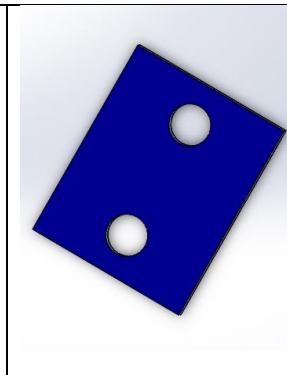
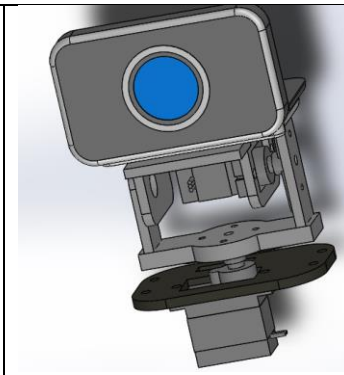
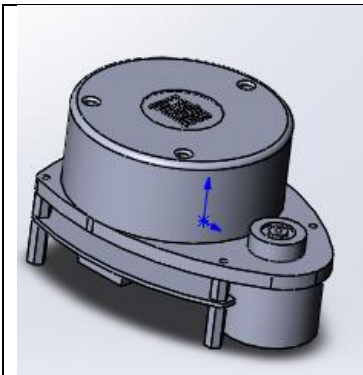
Trụ tay máy



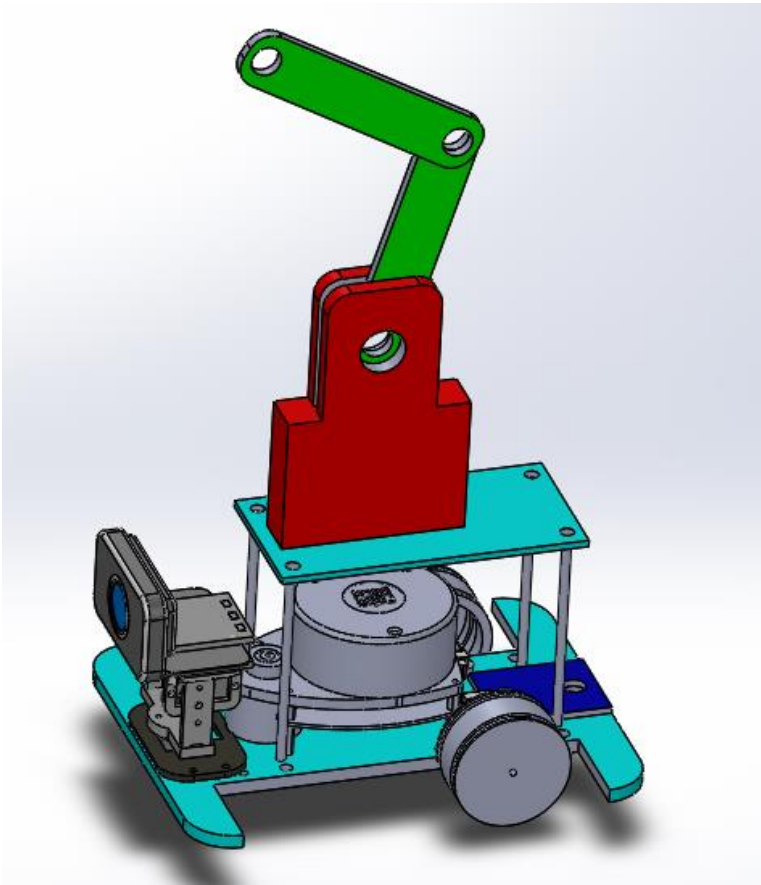
Link 1 và link 2



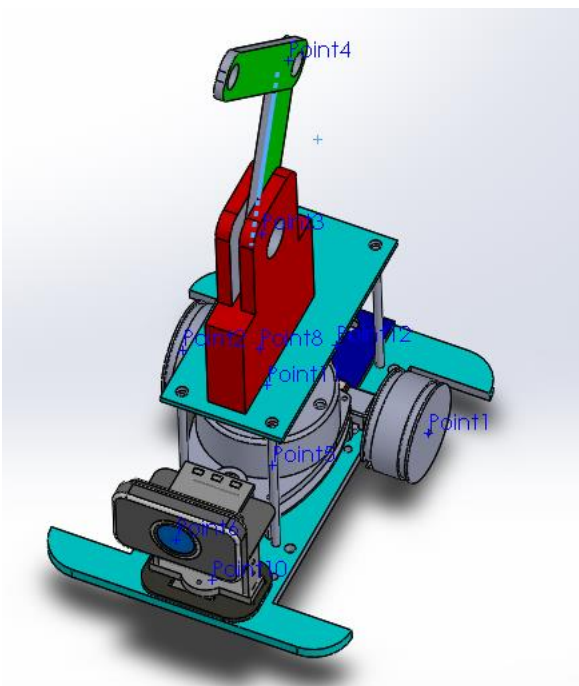
Ngoài ra còn các linh kiện điện tử sử dụng trên Grabcad như: lidar, camera và motor



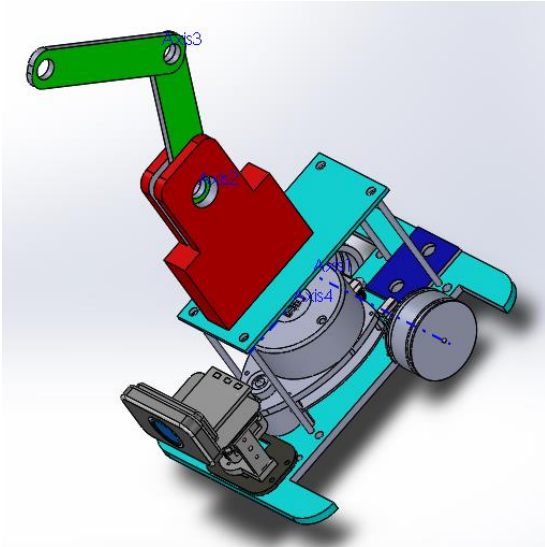
Tiến hành mate các chi tiết trên ta được robot như hình



## 2. Đặt các Point, Axis và Coordinate

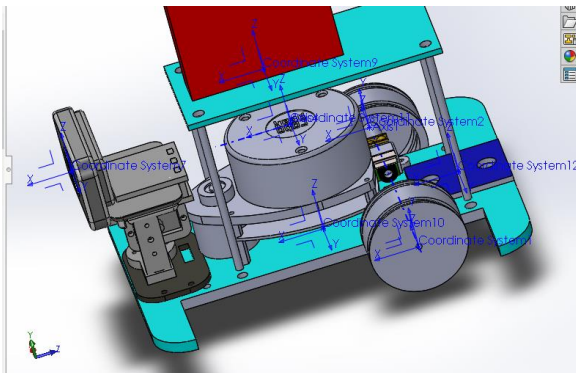


Ta xác định các điểm mà khớp quay quay: khớp tay máy, bánh xe,... ngoài ra còn các cảm biến ta sẽ sử dụng trong Gazebo và Rviz: lidar, camera, imu

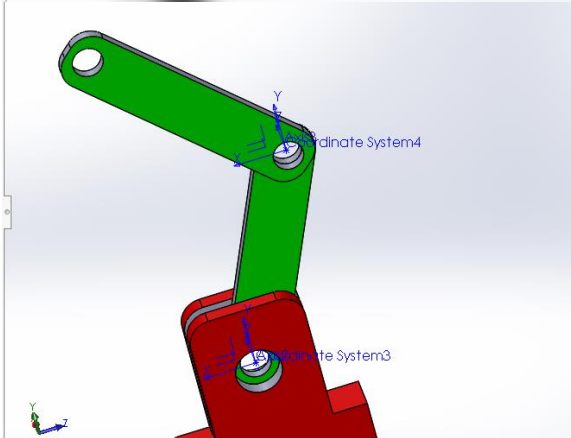


Xác định các trục quay quanh

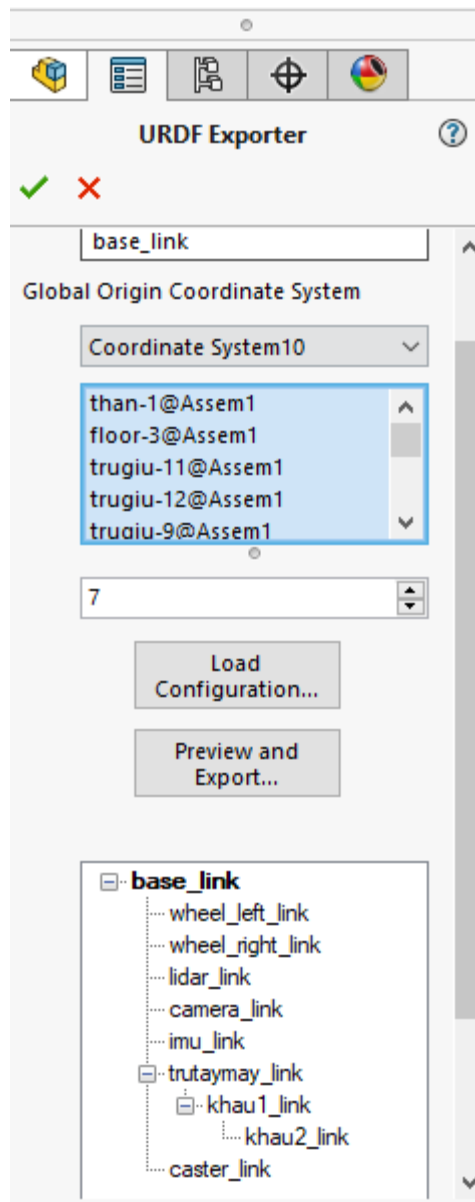
- Axis 1 cho hai bánh quay quanh
- Axis 2 cho khớp thứ nhất quay quanh
- Axis 3 cho khớp thứ hai quay quanh



Tiến hành đặt trục cho các point

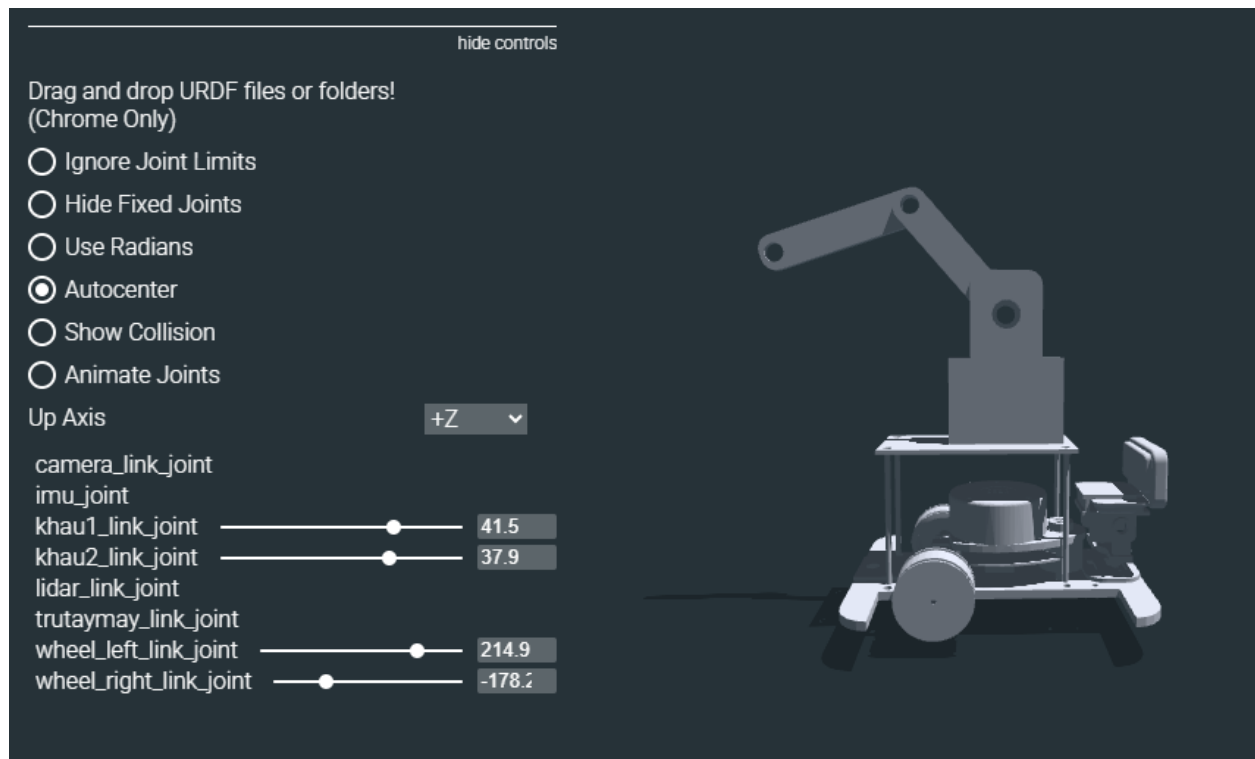


3. Export as URDF.



- Bước đầu ta cần chọn các phần sẽ cố định không cần thiết đặt là base\_link
- Bước thứ hai ta sẽ chọn các part nối với base link mà sẽ sử dụng để điều khiển trong Gazebo: hai bánh trái phải, lidar, camera, imu, trụ tay máy. Xác định các child của parent cụ thể là trụ tay máy sẽ gắn với khâu 1 và khâu 1 liên kết khâu 2.
- Bước thứ ba sẽ chọn các khâu fixed hay bánh xe (continuous) hoặc tay máy (revolute)





Sau đó xuất file URDF. Trước khi nhúng vào Gazebo có thể kiểm tra lại trên web URDF viewer example và test chuyển động tay máy và bánh xe nếu đặt trục tọa độ đúng

### III. Cấu trúc file URDF

#### 1. Giới thiệu về URDF

Các thư mục của một file URDF

<b>thangbnsh</b>	Initial commit for tn18 package	4f0251c · 2 hours ago	1 Commit
config	Initial commit for tn18 package	2 hours ago	
launch	Initial commit for tn18 package	2 hours ago	
meshes	Initial commit for tn18 package	2 hours ago	
rviz	Initial commit for tn18 package	2 hours ago	
scripts	Initial commit for tn18 package	2 hours ago	
urdf	Initial commit for tn18 package	2 hours ago	
CMakeLists.txt	Initial commit for tn18 package	2 hours ago	
export.log	Initial commit for tn18 package	2 hours ago	
package.xml	Initial commit for tn18 package	2 hours ago	

/config: chứa các file .yaml gồm cấu hình các link và điều khiển ROS Control

/launch: gồm các file .launch mở gazebo và rviz

/meshes: chứa các khâu, khớp, baselink và linh kiện của robot

/rviz: là file setup các topic trong rviz

/scripts: chứa các file .py chứa file control\_arm điều khiển tay máy và imu\_read đọc giá trị cảm biến imu

/urdf: chứa file .urdf bao gồm các link trong robot và plugins giúp điều khiển vi sai, tay máy và các cảm biến

/CMakeLists.txt: Cấu hình biên dịch cho chương trình

Package.xml: Khai báo các dependence của package

## 2. File .urdf

- Bao gồm các link code cấu trúc dạng .xml, một đoạn code về một link có các phần như sau:
  - <robot khai báo thể
  - <link name định nghĩa bộ phận
  - <link> các thuộc tính của link như hình dạng, độ rộng chiều dài, link tới .mesh, màu sắc, vật liệu.
  - <joint> định nghĩa các khớp nối gồm parent, child, kiểu khớp (quay, cố định,...), vị trí khớp, giới hạn lực effort, tốc độ quay,...
- Các phần plugins dùng để điều khiển robot vi sai, tay máy và các cảm biến

### a. Điều khiển vi sai

```
593 <gazebo>
594   <plugin name="differential_drive_controller" filename="libgazebo_ros_diff_drive.so">
595     <updateRate>100</updateRate>
596     <leftJoint>wheel_left_link_joint</leftJoint>
597     <rightJoint>wheel_right_link_joint</rightJoint>
598
599     <wheelSeparation>0.5380</wheelSeparation>
600     <wheelDiameter>0.2416</wheelDiameter>
601     <wheelAcceleration>0.5</wheelAcceleration>
602     <wheelTorque>20</wheelTorque>
603     <commandTopic>cmd_vel</commandTopic>
604     <odometryTopic>odom</odometryTopic>
605     <odometryFrame>odom</odometryFrame>
606     <robotBaseFrame>base_link</robotBaseFrame>
607     <odometrySource>1</odometrySource>
608     <publishWheelTF>true</publishWheelTF>
609     <publishOdom>true</publishOdom>
610     <publishWheelJointState>true</publishWheelJointState>
611
612     <legacyMode>false</legacyMode>
613   </plugin>
614 </gazebo>
```

Plugin nhận lệnh từ ROS topic /cmd\_vel (twist)

Plugin tính toán vận tốc từng bánh vào kinematic xe vì sai sau đó gửi momen xoắn tới các bánh qua joint effort

Plugin xuất odom lên /odom để dùng cho localization, có thể xuất TF robot.

## b. IMU

```
560 <gazebo reference="imu_link">
561   <sensor name="imu_sensor" type="imu">
562     <always_on>true</always_on>
563     <update_rate>100</update_rate>
564     <visualize>true</visualize>
565     <scale>0.1 0.1 0.1</scale>
566     <plugin filename="libgazebo_ros_imu_sensor.so" name="imu_plugin">
567       <topicName>/imu</topicName>
568       <bodyName>imu_link</bodyName>
569       <frameName>imu_link</frameName>
570       <gaussianNoise>0.01</gaussianNoise>
571       <xyzOffset>0 0 0</xyzOffset>
572       <rpyOffset>0 0 0</rpyOffset>
573       <initialOrientationAsReference>false</initialOrientationAsReference>
574     </plugin>
575     <pose>0 0 0 1.5708 0</pose>
576   </sensor>
577 </gazebo>
```

IMU thu thập dữ liệu gửi lên ROS sensor\_msgs/Imu

Cho phép các node khác có thể subscribe và topic này để sử dụng dữ liệu. Ứng dụng trong đo lường quỹ đạo di chuyển, giữ cân bằng và điều hướng odom

## c. Lidar

```
623 <gazebo reference="lidar_link">
624   <sensor type="ray" name="lidar_sensor">
625     <pose>0 0 0 0 0</pose>
626     <visualize>true</visualize>
627     <update_rate>30</update_rate>
628     <ray>
629       <scan>
630         <horizontal>
631           <samples>720</samples>
632           <resolution>1</resolution>
633           <min_angle>-3.1416</min_angle>
634           <max_angle>3.1416</max_angle>
635         </horizontal>
636       </scan>
637       <range>
638         <min>0.1</min>
639         <max>30.0</max>
640         <resolution>0.01</resolution>
641       </range>
642       <noise>
643         <type>gaussian</type>
644         <mean>0.0</mean>
645         <stddev>0.01</stddev>
646       </noise>
647     </ray>
648     <plugin name="gazebo_ros_laser" filename="libgazebo_ros_laser.so">
```

---

```

642         <stddev>0.01</stddev>
646     </noise>
647 </ray>
648 <plugin name="gazebo_ros_laser" filename="libgazebo_ros_laser.so">
649   <topicName>/scan</topicName>
650   <frameName>lidar_link</frameName>
651 </plugin>
652 </sensor>

```

Lidar phát xung laser theo hướng nhất định phát hiện vật cản sẽ phản lại.

Setup góc quay 360 độ từ (-3.14,3.14) giúp quét toàn bộ map.

Độ phân giải góc res=1 mỗi bước quét cách nhau một đơn vị góc

Khoảng cách quét từ 0.1 đến 30m

#### d. Camera

```

653 </gazebo>
654 <gazebo reference="camera_link">
655   <sensor type="camera" name="camera1">
656     <update_rate>30.0</update_rate>
657     <camera name="head">
658       <horizontal_fov>1.3962634</horizontal_fov>
659       <image>
660         <width>800</width>
661         <height>800</height>
662         <format>R8G8B8</format>
663       </image>
664       <clip>
665         <near>0.02</near>
666         <far>300</far>
667       </clip>
668       <noise>
669         <type>gaussian</type>
670         <mean>0.0</mean>
671         <stddev>0.007</stddev>
672       </noise>
673     </camera>
674     <plugin name="camera_controller" filename="libgazebo_ros_camera.so">
675       <alwaysOn>true</alwaysOn>
676       <updateRate>6.0</updateRate>
677       <cameraName>rrbot/camera1</cameraName>
678       <imageTopicName>image_raw</imageTopicName>
679       <cameraInfoTopicName>camera_info</cameraInfoTopicName>
680       <frameName>camera_link</frameName>
681       <hackBaseline>0.07</hackBaseline>
682       <distortionK1>0.0</distortionK1>
683       <distortionK2>0.0</distortionK2>
684       <distortionK3>0.0</distortionK3>
685       <distortionT1>0.0</distortionT1>
686       <distortionT2>0.0</distortionT2>
687     </plugin>
688   </sensor>
689 </gazebo>

```

Camera thu nhận ánh sáng từ môi trường sau đó tạo hình ảnh chuyển thành ảnh số, với mỗi pixel trong ảnh biểu diễn cường độ ánh sáng và màu sắc từ môi trường.

Plugin này cho phép camera hoạt động trong Gazebo và xuất dữ liệu ra ROS.

Camera chụp ảnh với tốc độ 30fps, ảnh được xuất lên topic ROS image\_raw. Mô phỏng dạng ống kính.

#### e. Điều khiển tay máy

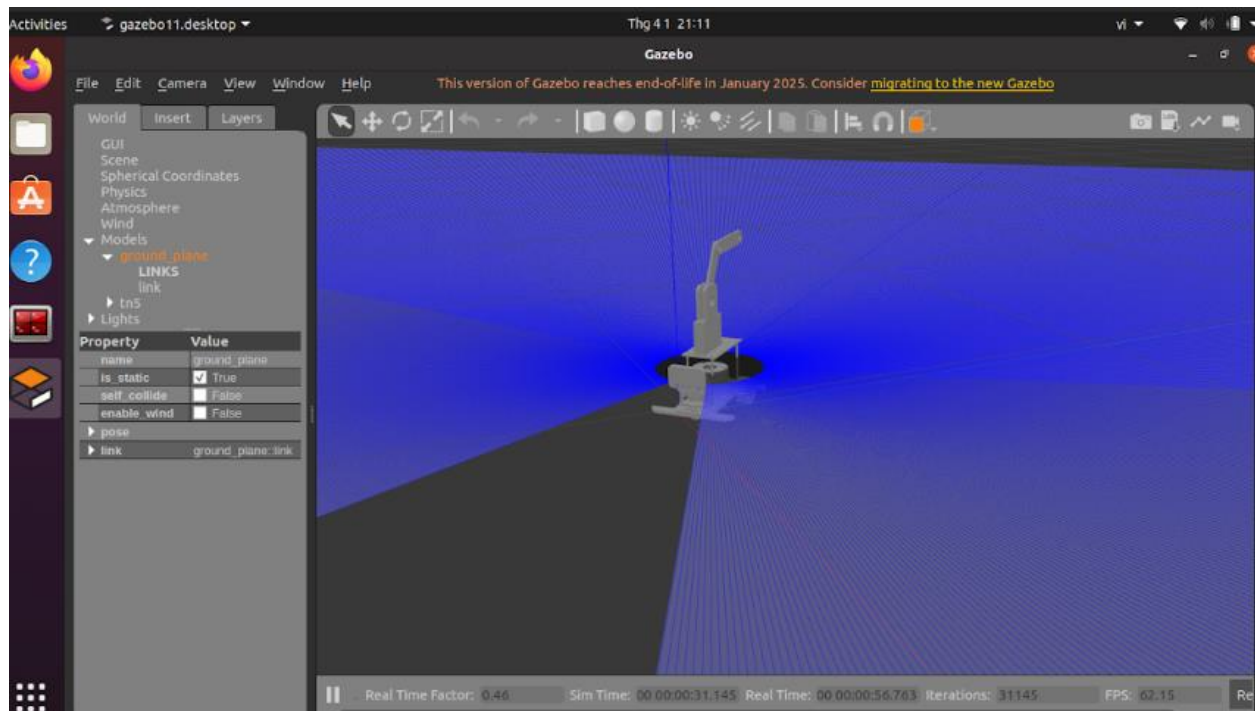
```
699 <transmission name="khau1_trans">
700   <type>transmission_interface/SimpleTransmission</type>
701   <joint name="khau1_link_joint">
702     <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
703   </joint>
704   <actuator name="khau1_actuator">
705     <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
706     <mechanicalReduction>1.0</mechanicalReduction>
707   </actuator>
708 </transmission>
709
710 <transmission name="khau2_trans">
711   <type>transmission_interface/SimpleTransmission</type>
712   <joint name="khau2_link_joint">
713     <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
714   </joint>
715   <actuator name="khau2_actuator">
716     <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
717     <mechanicalReduction>1.0</mechanicalReduction>
718   </actuator>
719 </transmission>
720
```

<transmission> truyền chuyển động cho hệ thống robot

Khai báo các khớp nối (joint). Bên trong <joint>, phần <hardwareInterface> xác định giao diện phần cứng mà khớp này sử dụng (EffortJointInterface).

Các bộ truyền động actuator đại diện cho các bộ điều khiển khớp EffortJointInterface truyền lực đến khớp.

Sau khi chạy file urdf trên ta được kết quả mô phỏng sau



### 3. Yaml

```

1 joint_state_controller:
2   type: joint_state_controller/JointStateController
3   publish_rate: 100
4
5 khau1_controller:
6   type: effort_controllers/JointPositionController
7   joint: khau1_link_joint
8   pid: {p: 5.0, i: 0.005, d: 0.05}
9   max_effort: 1.0
10  velocity_limit: 1
11
12 khau2_controller:
13   type: effort_controllers/JointPositionController
14   joint: khau2_link_joint
15   pid: {p: 5.0, i: 0.005, d: 0.05}
16   max_effort: 1.0
17   velocity_limit: 1
18

```

Sử dụng bộ điều khiển JointPositionController điều khiển vị trí khớp dựa trên PID.

Các thông số PID được điều chỉnh để tay máy chạy mượt mà.

Giới hạn effort và velo để tay máy quay không bị giật mạnh làm ảnh hưởng chuyển động.

### 4. Launch

File launch cho phép mở cùng lúc gazebo, rviz cùng lúc và link đến file configrviz.rviz đã có setup sẵn các topic cảm biến

---

```

1 <launch>
2 <!-- Load Gazebo -->
3 <include file="$(find gazebo_ros)/launch/empty_world.launch" />
4
5 <!-- Spawn Robot Model -->
6 <node name="spawn_model" pkg="gazebo_ros" type="spawn_model"
7   args="-file $(find tn13)/urdf/tn13.urdf -urdf -model tn13"
8   output="screen" />
9
10 <!-- Fake Joint Calibration -->
11 <node name="fake_joint_calibration" pkg="rostopic" type="rostopic"
12   args="pub /calibrated std_msgs/Bool true" />
13
14 <!-- Load robot URDF -->
15 <param name="robot_description" command="$(find xacro)/xacro $(find tn13)/urdf/tn13.urdf" />
16
17 <!-- Node xuất thông tin trạng thái khớp -->
18 <node name="joint_state_publisher_gui" pkg="joint_state_publisher_gui" type="joint_state_publisher_gui" />
19
20 <!-- Node xuất trạng thái robot -->
21 <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" />
22
23 <!-- RViz hiển thị robot và cảm biến -->
24 <node name="rviz" pkg="rviz" type="rviz" args="-d $(find tn13)/rviz/configrviz.rviz" output="screen" />
25 </launch>

```

---

## 5. Scripts

### a. Control\_arm

Khi chạy chương trình, nó sẽ đọc các phím nhấn (w, s, a, d, q) từ bàn phím để điều khiển các khớp (khau1\_link\_joint và khâu2\_link\_joint) của tay máy.

```

1  #!/usr/bin/env python3
2
3  import rospy
4  import sys, select, termios, tty
5  from std_msgs.msg import Float64
6
7  msg = ""
8  Điều khiển tay máy:
9
10     Nhấn 'w' để tăng khâu1_link_joint
11     Nhấn 's' để giảm khâu1_link_joint
12     Nhấn 'a' để tăng khâu2_link_joint
13     Nhấn 'd' để giảm khâu2_link_joint
14     Nhấn 'q' để thoát
15
16     """
17
18     # Mục điều chỉnh góc mỗi lần nhấn phím (radians)
19     step_size = 0.1
20
21     # Giá trị ban đầu của các joint
22     khâu1_angle = 0.0
23     khâu2_angle = 0.0
24
25     def get_key():
26         """Đọc phím từ terminal"""
27         tty.setraw(sys.stdin.fileno())
28         select.select([sys.stdin], [], [], 0)

```



Gửi giá trị góc (đơn vị radians) đến các topic ROS để điều chỉnh vị trí khớp.

Khi nhấn q, chương trình sẽ dừng lại.

```
25 def get_key():
26     """Đọc phím từ terminal"""
27     tty.setraw(sys.stdin.fileno())
28     select.select([sys.stdin], [], [], 0)
29     key = sys.stdin.read(1)
30     termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
31     return key
32
33 def teleop_control():
34     global khau1_angle, khau2_angle
35
36     rospy.init_node('arm_teleop')
37     khau1_pub = rospy.Publisher('/khau1_controller/command', Float64, queue_size=10)
38     khau2_pub = rospy.Publisher('/khau2_controller/command', Float64, queue_size=10)
39     rospy.sleep(1)
40
41     print(msg)
42
43     while not rospy.is_shutdown():
44         key = get_key()
45
46         if key == 'w':
47             khau1_angle += step_size
48         elif key == 's':
49             khau1_angle -= step_size
50         elif key == 'a':
51             khau2_angle += step_size
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

Khi ấn w/s/a/d để điều chỉnh góc, giá trị góc sẽ được gửi đến ROS qua publisher. Controller của ROS đọc dữ liệu từ topic và điều khiển động cơ.



## b. Imu read

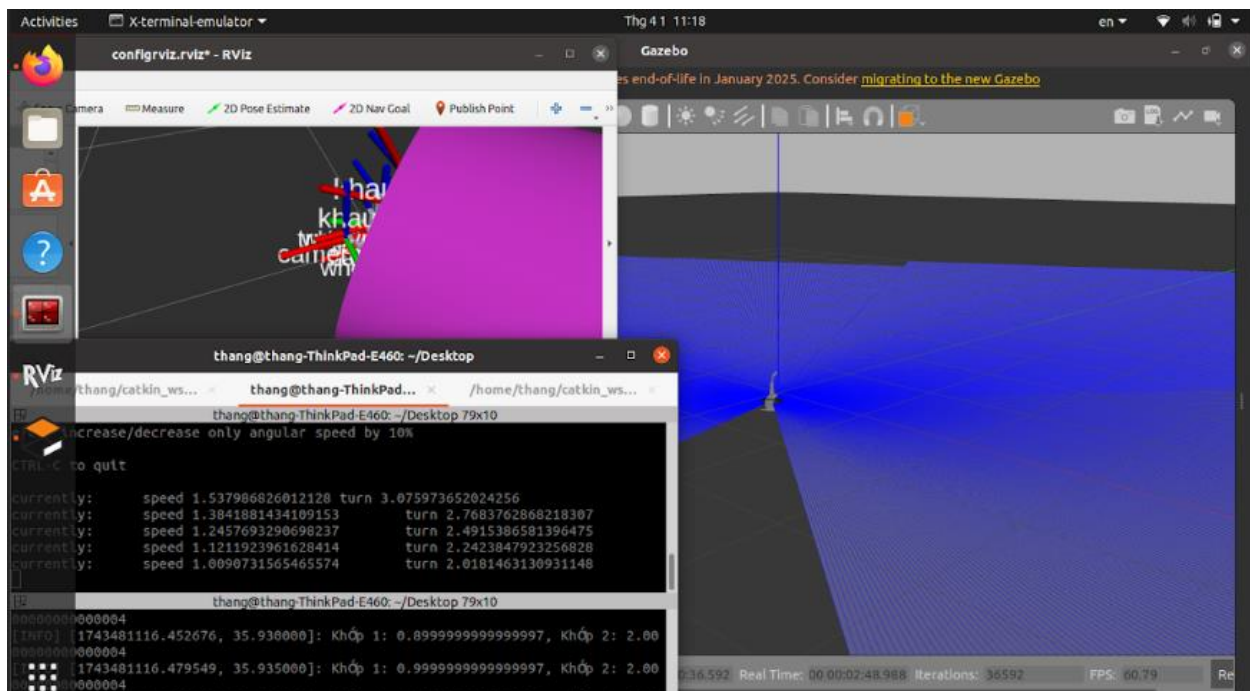
```
1  #!/usr/bin/env python3
2
3  import rospy
4  from sensor_msgs.msg import Imu
5  import time # Thêm thư viện để theo dõi thời gian
6
7  last_log_time = 0 # Lưu thời gian log lần trước
8  log_interval = 1.0 # Giảm log xuống 1 lần mỗi giây
9
10 def imu_callback(data):
11     global last_log_time
12     current_time = time.time()
13
14     # Chỉ log nếu đã qua ít nhất 'log_interval' giây
15     if current_time - last_log_time >= log_interval:
16         rospy.loginfo(f"Orientation: x={data.orientation.x}, y={data.orientation.y}, z={data.orientation.z}, w={data.orientation.w}")
17         rospy.loginfo(f"Angular Velocity: x={data.angular_velocity.x}, y={data.angular_velocity.y}, z={data.angular_velocity.z}")
18         rospy.loginfo(f"Linear Acceleration: x={data.linear_acceleration.x}, y={data.linear_acceleration.y}, z={data.linear_acceleration.z}")
19         rospy.loginfo("----")
20         last_log_time = current_time # Cập nhật thời gian log gần nhất
21
22 def imu_listener():
23     rospy.init_node('imu_listener', anonymous=True)
24     rospy.Subscriber("/imu/data", Imu, imu_callback)
25     rospy.spin()
26
27 if __name__ == '__main__':
28     imu_listener()
```

Đọc dữ liệu gia tốc và vận tốc từ imu sau đó có thể subscriber để lấy dữ liệu IMU.

## IV. Kết quả chạy gazebo, rviz

### 1. Điều khiển xe

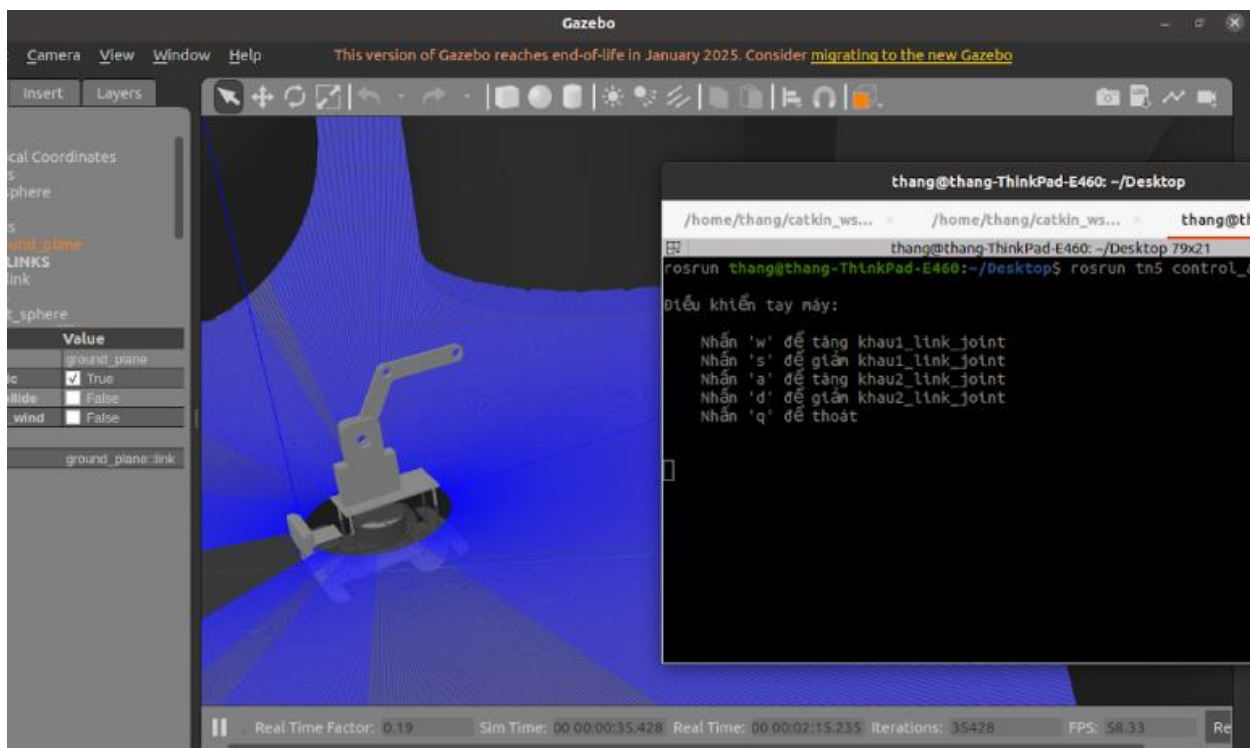
Điều khiển xe bằng teleop từ bàn phím. Hiện tại ta thấy 2 bánh đã xoay và dữ liệu đã được trả về



Bánh xe trên tf rviz đã quay

## 2. Điều khiển tay máy

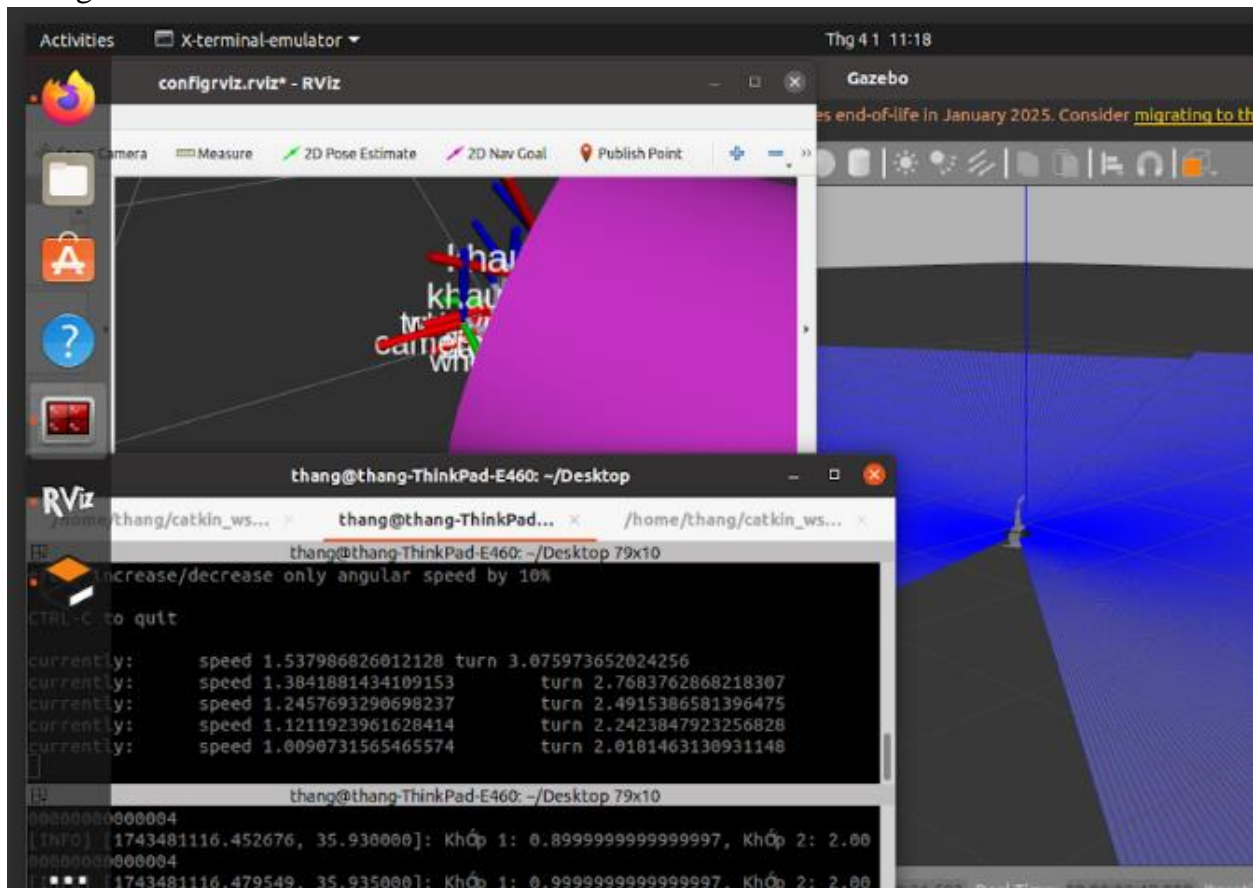
Tay máy điều khiển bằng w/s/a/d và check tf ta cũng thấy tay máy đã quay



Tay máy điều khiển bằng w/s/a/d và check tf ta cũng thấy tay máy đã quay

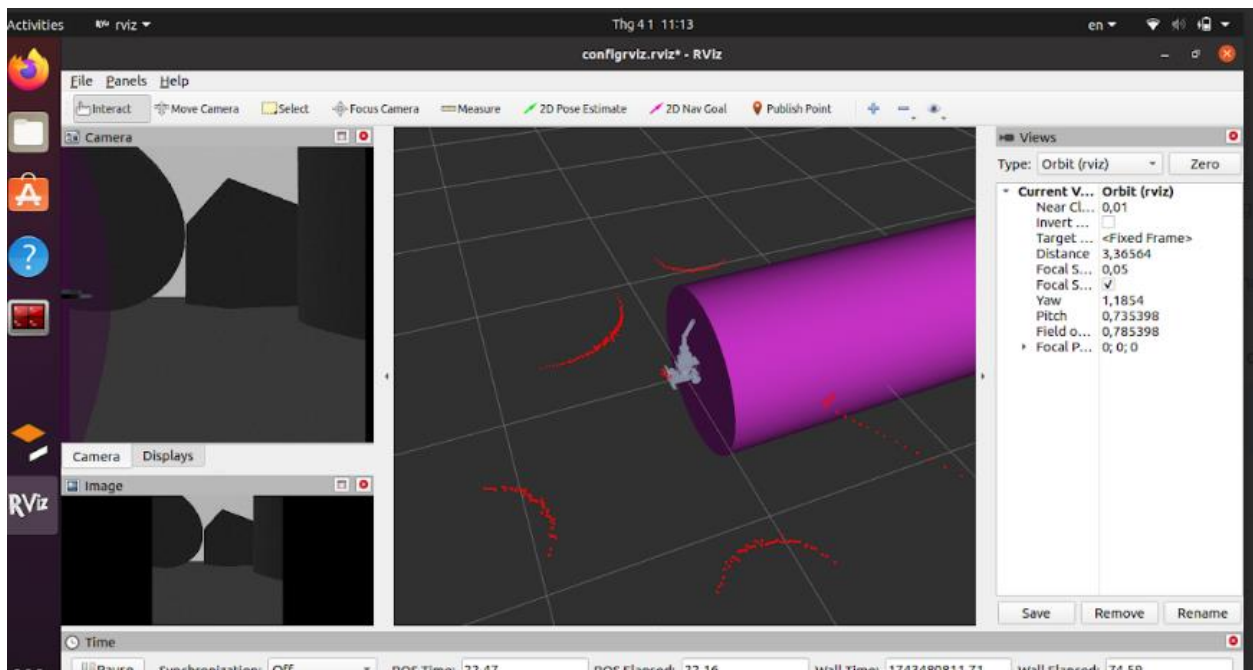
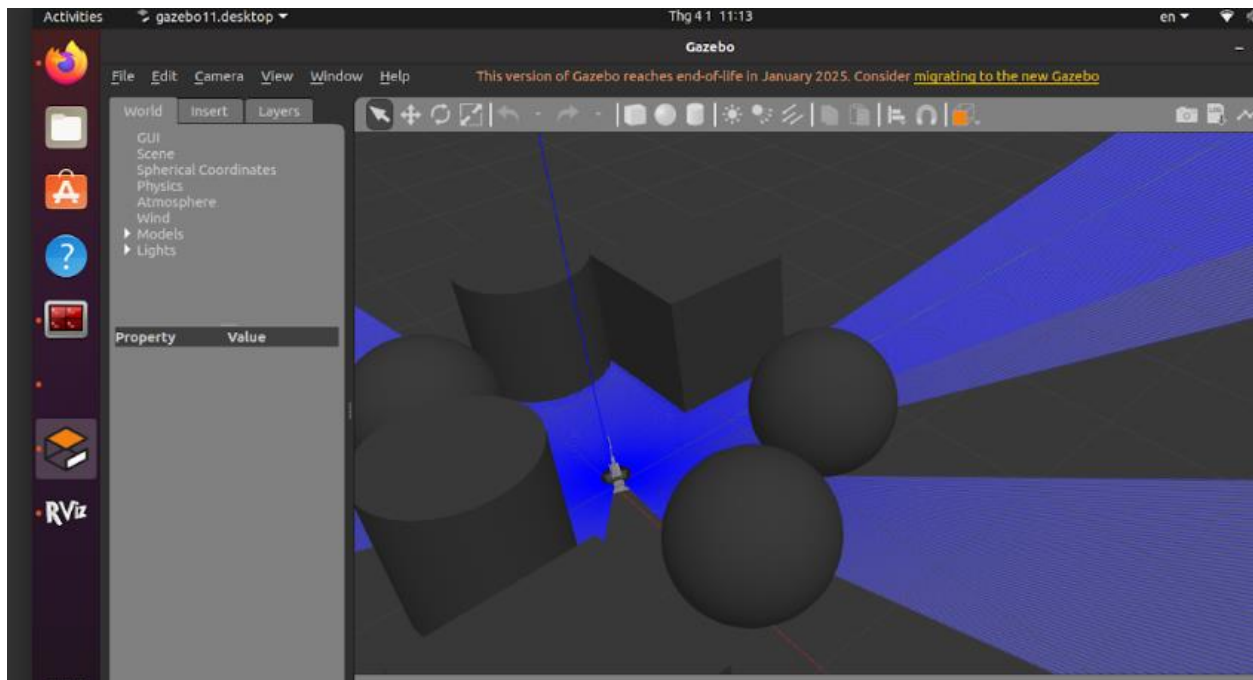
### 3. Imu

huong



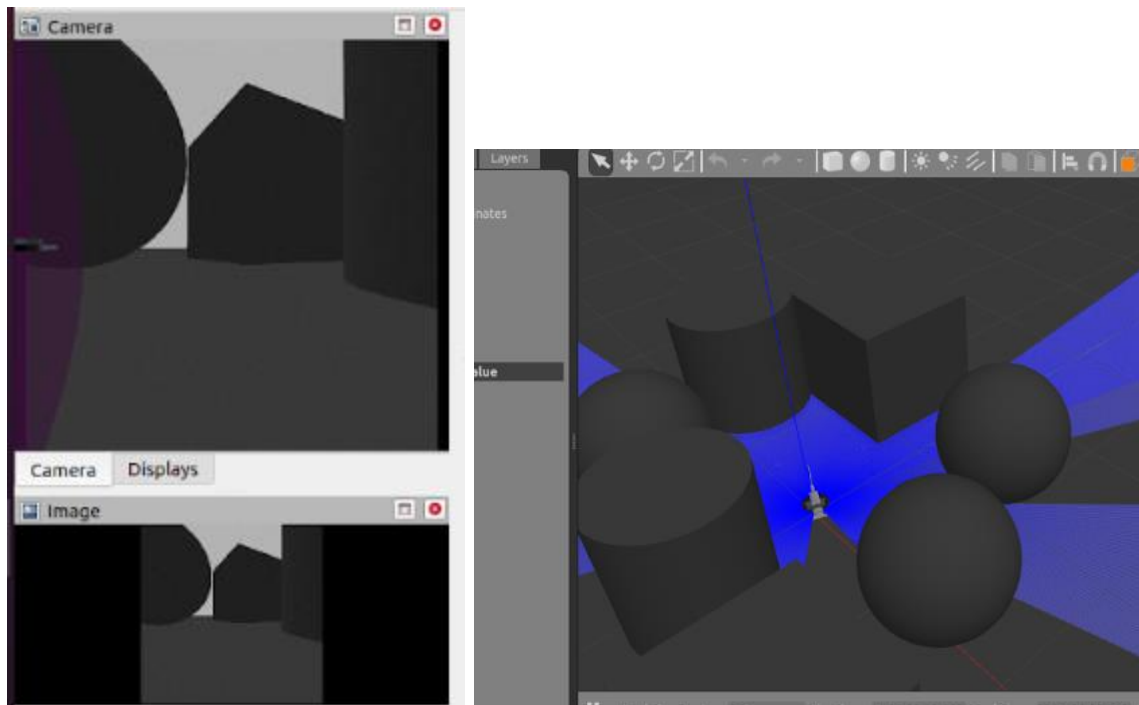
Khi robot chạy ta thấy mũi tên tím biểu thị cho topic imu đã hoạt động

### 4. Lidar



Lidar đọc các vật cản trên gazebo và hiển thị các viên chấm đỏ biểu thị có vật cản

## 5. Camera



Xe phát hiện các vật trước mặt và hiển thị lên topic camera