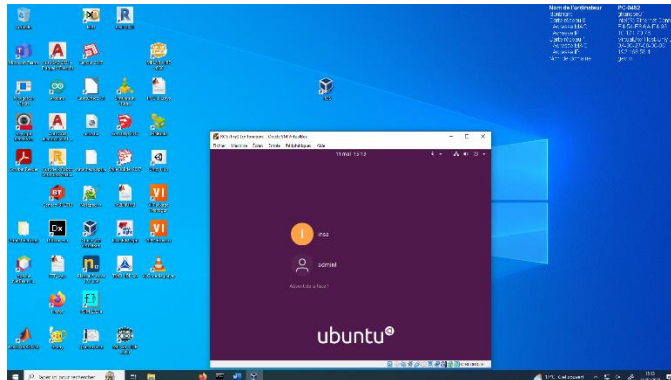


TUTO TEST

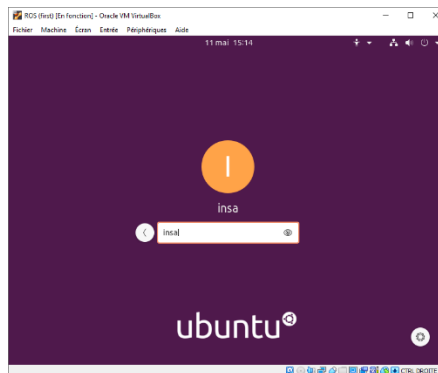
ATTENTION POUR TOUT LE TUTO SI MODIFICATION DU FICHIER « .BASHRC » OUVRIR UN NOUVEAU TERMINAL !!! (la modification ne se fait pas dans le terminal qui modifie le fichier)

QUESTION 1 : Lancer turlesim et contrôler sa tortue avec teleopkey et avec les cmd_vel donc avec le clavier (pour tous les sujets) :

- Etape 1 : ouvrir la machine virtuelle « ROS » :



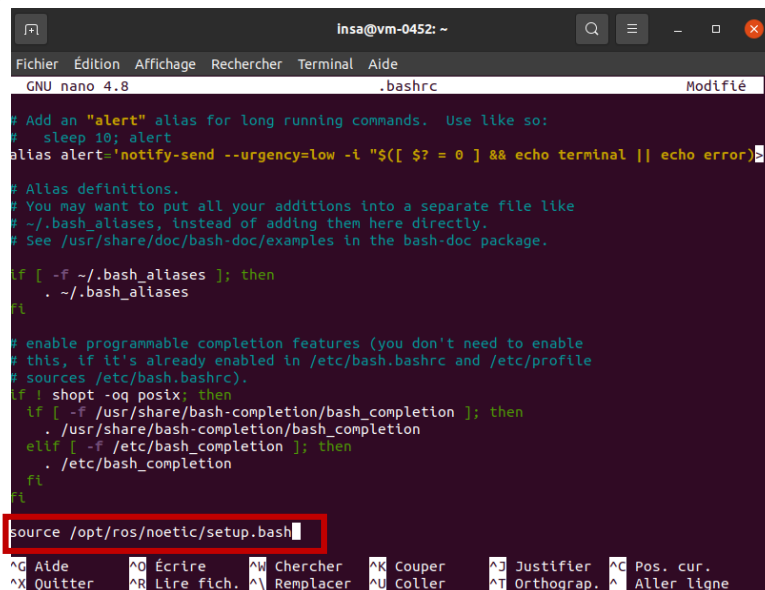
- Etape 2 : INSA avec login « insa » :



- Etape 3 : Ouvrir un Terminal avec « ctr + alt +T » puis modifier le fichier « .bashrc » avec la commande suivante :

```
insa@vm-0452:~$ sudo nano .bashrc  
[sudo] Mot de passe de insa :
```

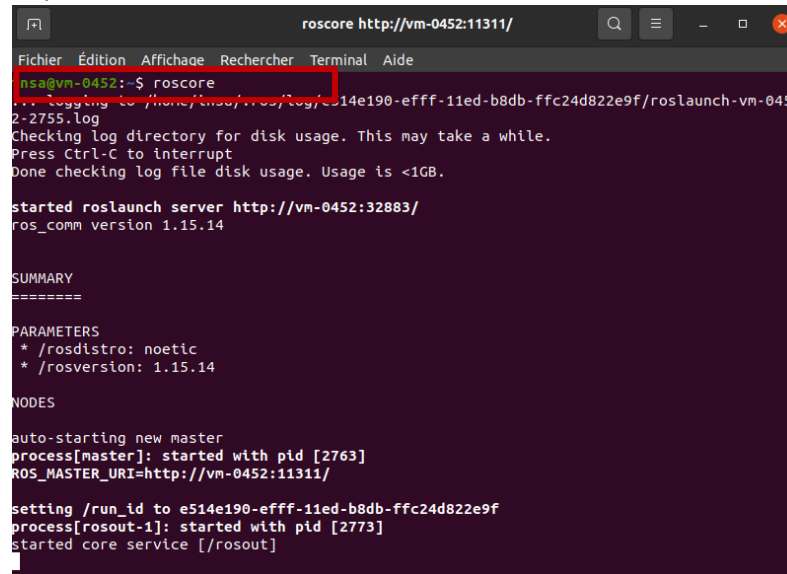
Rajouter la ligne suivante :



TUTO TEST

Enregistrer « ctr +x » puis « 0 » puis « entrée »

- Etape 4 : Sur un premier terminal [TERMINAL1], lancer la commande « roscore » :

A terminal window titled 'roscore http://vm-0452:11311/' with a menu bar (Fichier, Édition, Affichage, Rechercher, Terminal, Aide). The prompt is 'insa@vm-0452:~\$'. The command 'roscore' has been entered and executed. The output shows logging to a file, checking log directory, and starting the roslaunch server. It also displays summary, parameters (noetic, 1.15.14), and nodes (auto-starting new master, process[master] started with pid [2763], ROS_MASTER_URI=http://vm-0452:11311/, setting /run_id to e514e190-ffff-11ed-b8db-ffc24d822e9f, process[rosout-1] started with pid [2773], started core service [/rosout]).

```
roscore http://vm-0452:11311/
Fichier Édition Affichage Rechercher Terminal Aide
insa@vm-0452:~$ roscore
[1.14] Logging to /home/insa/.ros/log/e514e190-ffff-11ed-b8db-ffc24d822e9f/roslaunch-vm-0452-2755.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://vm-0452:32883/
ros_comm version 1.15.14

SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.15.14

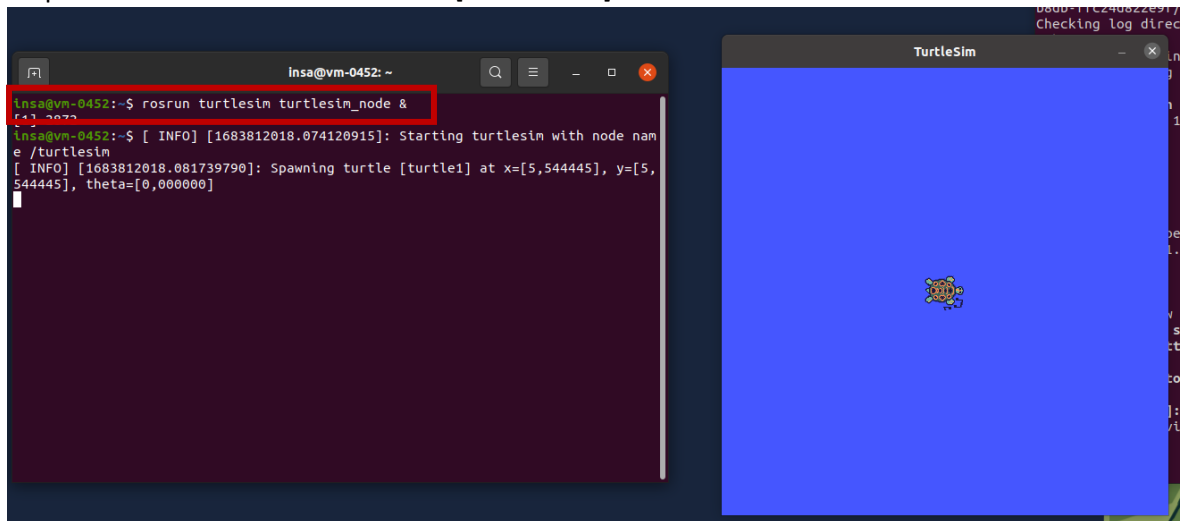
NODES

auto-starting new master
process[master]: started with pid [2763]
ROS_MASTER_URI=http://vm-0452:11311/

setting /run_id to e514e190-ffff-11ed-b8db-ffc24d822e9f
process[rosout-1]: started with pid [2773]
started core service [/rosout]
```

Attention le roscore devra toujours tourner dans un Terminal !

- Etape 5 : lancer un nouveau Terminal [TERMINAL2] et entrer la commande suivante :

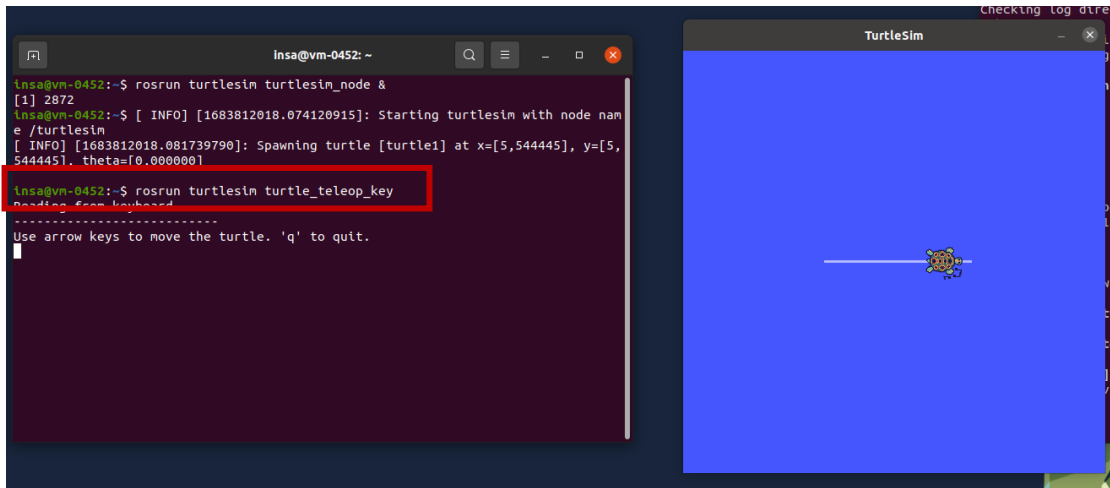
Two windows are shown side-by-side. The left window is a terminal titled 'insa@vm-0452: ~' with the prompt 'insa@vm-0452:~\$'. The command 'roslaunch turtlesim turtlesim_node &' has been entered and executed. The output shows starting turtlesim with node name /turtlesim and spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]. The right window is titled 'TurtleSim' and shows a blue square environment with a small yellow turtle icon in the center.

```
insa@vm-0452: ~
insa@vm-0452:~$ roslaunch turtlesim turtlesim_node &
[1.14] INFO [1683812018.074120915]: Starting turtlesim with node name /turtlesim
[1.14] INFO [1683812018.081739790]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
```

Une fenêtre « turtlesim » va alors s'ouvrir. Ensuite appuyer sur « entrée ».

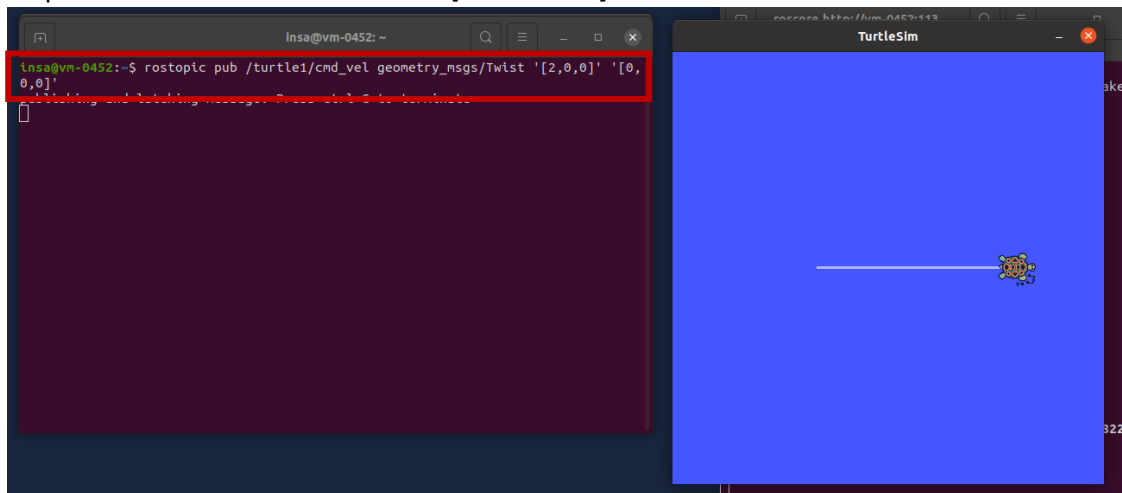
- Etape 6 : Lancer la commande suivante :

TUTO TEST



Faire bouger la tortue avec le clavier.

- Etape 7 : ouvrir un nouveau terminal [TERMINAL3] et mettre la commande suivante :



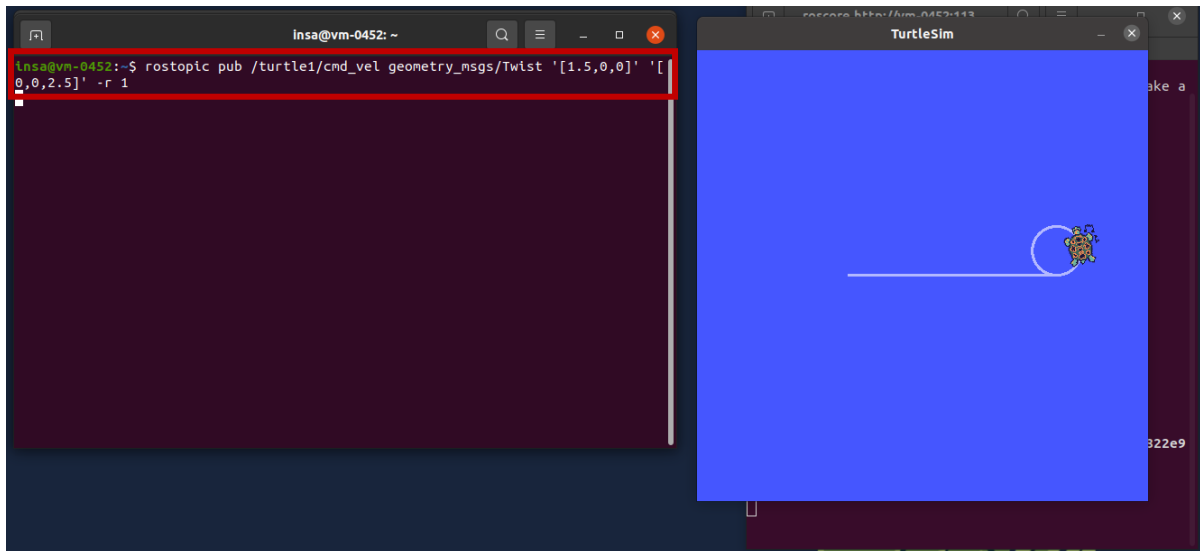
La tortue peut ainsi être commandée soit par « cmd_vel » ou « teleopkey ».

Mettre « ctrl+c » pour terminer la tâche ou mettre « & » à la fin de la commande.

QUESTION 2 : Faire tourner la tortue avec une vitesse de 1.5ms et un angle de 2.5 rad/s :

Rafraîchir le terminal avec la commande « clear » puis mettre la commande suivante :

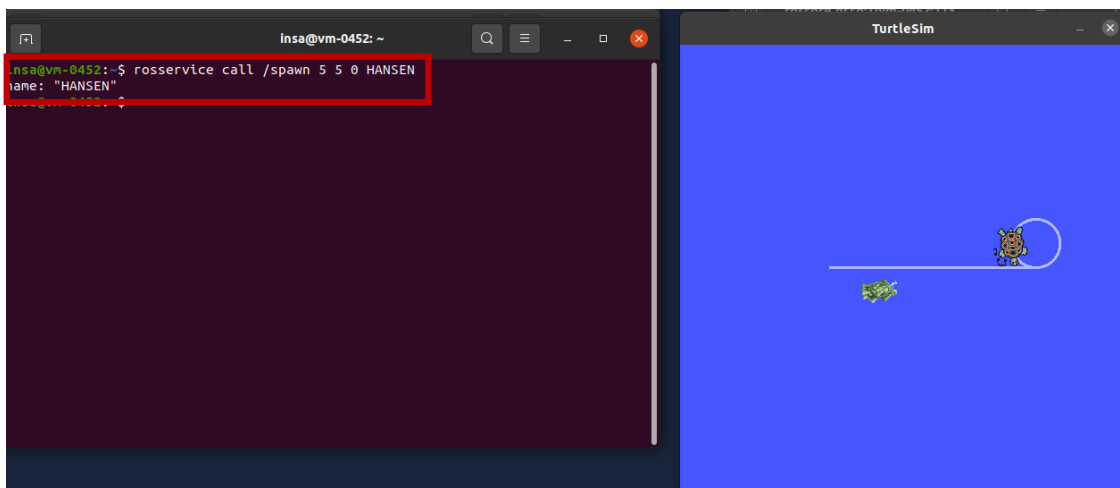
TUTO TEST



La tortue doit alors tourner avec une vitesse de 1.5ms et avec un angle de 2.5 rad/s tous les 1s.

QUESTION 3 : Ajouter une deuxième tortue avec son nom de famille :

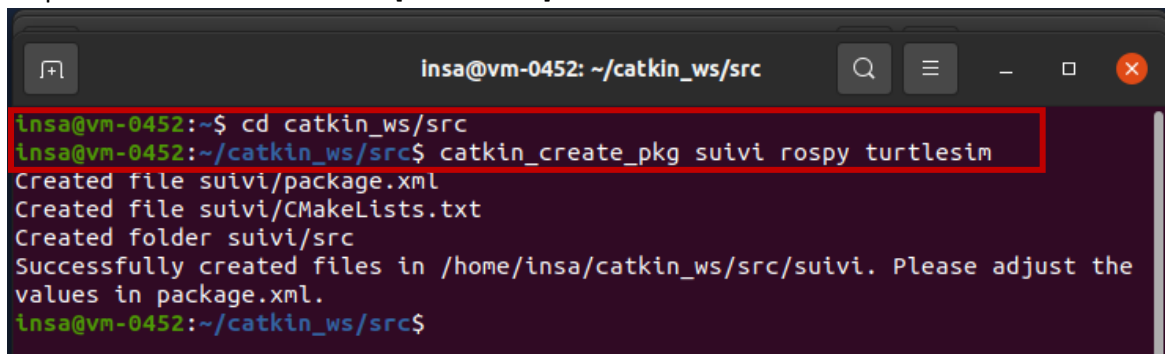
Ouvrir un nouveau Terminal [TERMINAL4] et lancer la commande suivante :



Une nouvelle tortue doit apparaître.

QUESTION 4 : SUJET CATKIN : La tortue 'HANSEN' doit suivre le mouvement de la tortue 'turtle1' :

- Etape 1 : rester dans le Terminal [TERMINAL4] et lancer les commandes :



- Etape 2 : Mettre les commandes suivantes et à la fin mettre le mot de passe 'insa' :

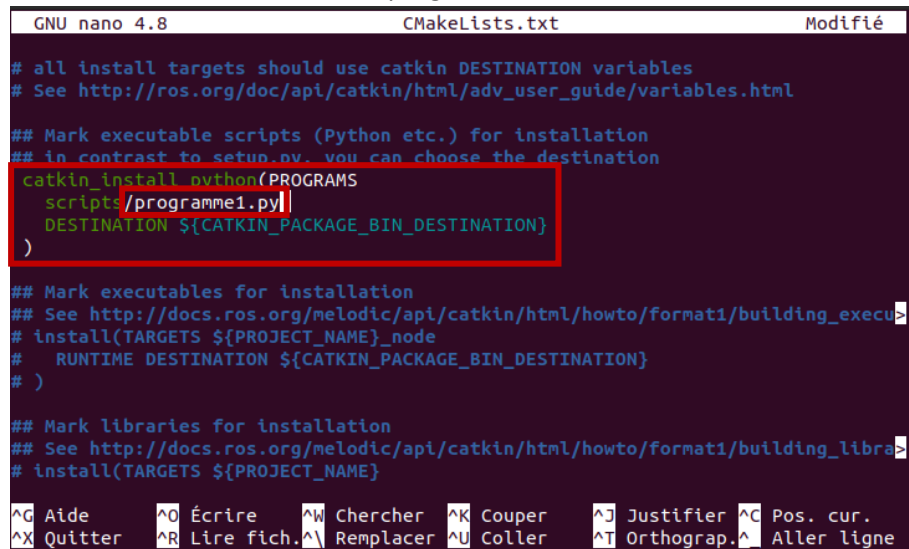
TUTO TEST

```
insa@vm-0452:~/catkin_ws/src$ cd suivi
insa@vm-0452:~/catkin_ws/src/suivi$ mkdir scripts
insa@vm-0452:~/catkin_ws/src/suivi$ cd scripts/
insa@vm-0452:~/catkin_ws/src/suivi/scripts$ sudo nano programme1.py
[sudo] Mot de passe de insa :
```

- Etape 3 : Mettre les commandes suivantes :

```
insa@vm-0452:~/catkin_ws/src/suivi/scripts$ cd ..
insa@vm-0452:~/catkin_ws/src/suivi$ sudo nano CMakeLists.txt
```

- Etape 4 : Modifier le fichier texte 'CMakeLists.txt' en décommentant les parties suivantes et en modifiant l'extension avec le nom du programme crée :



```
GNU nano 4.8 CMakeLists.txt Modifié

# all install targets should use catkin DESTINATION variables
# See http://ros.org/doc/api/catkin/html/adv_user_guide/variables.html

## Mark executable scripts (Python etc.) for installation
## in contrast to setup.py, you can choose the destination
catkin_install_python(PROGRAMS
  scripts/programme1.py
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
)

## Mark executables for installation
## See http://docs.ros.org/melodic/api/catkin/html/howto/format1/building_execu>
# install(TARGETS ${PROJECT_NAME}_node
#   RUNTIME DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
# )

## Mark libraries for installation
## See http://docs.ros.org/melodic/api/catkin/html/howto/format1/building_libra>
# install(TARGETS ${PROJECT_NAME}
```

Puis « ctrl+x » puis « 0 » puis « entrée »

- Etape 5 : Mettre les commandes suivantes et modifier le « programme1.py » :

```
insa@vm-0452:~/catkin_ws/src/suivi$ cd scripts
insa@vm-0452:~/catkin_ws/src/suivi/scripts$ sudo nano programme1.py
```

- Etape 6 : écrire le programme dans le fichier texte :

TUTO TEST

```
GNU nano 4.8                               programme1.py                          Modifié
#!/usr/bin/env python3
import rospy
from geometry_msgs.msg import Twist

twi = Twist()
def twi_callback(data):
    global twi
    twi = data

if __name__ == "__main__":
    rospy.init_node("Suivi_node")
    rospy.loginfo("Suivi node has been started.")

    sub = rospy.Subscriber("/turtle1/cmd_vel", Twist, callback = twi_callback)
    pub = rospy.Publisher("/HANSEN/cmd_vel", Twist)

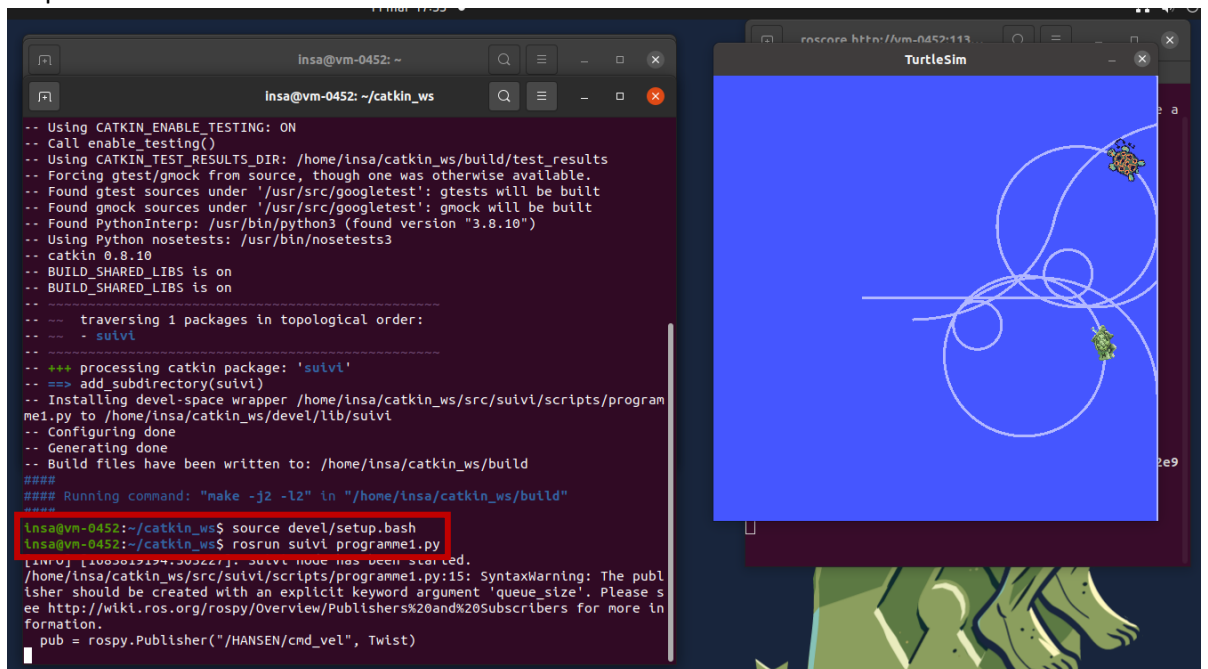
    rate = rospy.Rate(10)
    while not rospy.is_shutdown():
        msg = Twist()
        msg.linear.x = twi.linear.x
        msg.linear.y = twi.linear.y
        msg.linear.z = twi.linear.z
        msg.angular.x = twi.angular.x
        msg.angular.y = twi.angular.y
        msg.angular.z = twi.angular.z
        pub.publish(msg)
        rate.sleep()
```

Puis « ctrl+x » puis « 0 » puis « entrée »

- Etape 7 : Mettre les commandes suivantes :

```
insa@vm-0452:~/catkin_ws/src/suivi/scripts$ cd
insa@vm-0452:~$ cd catkin_ws/
insa@vm-0452:~/catkin_ws$ catkin_make
```

- Etape 8 : Mettre les commandes suivantes :



TUTO TEST

La tortue 'HANSEN' doit reproduire le mouvement de la tortue 'turtle1'.

QUESTION 4 : SUJET MATLAB : La tortue 'HANSEN' doit suivre le mouvement de la tortue 'turtle1' :

- Etape 1 : on relève l'adresse IP de la machine virtuelle avec la commande « ifconfig » :

```
insa@vm-0452: ~/catkin_ws
insa@vm-0452:~/catkin_ws$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.171.70.85 netmask 255.255.0.0 broadcast 10.171.255.255
    inet6 fe80::e4c:3008:dbf7:b72d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a4:07:ed txqueuelen 1000 (Ethernet)
    RX packets 215297 bytes 13705848 (13.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 563 bytes 58350 (58.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 35494 bytes 3617265 (3.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 35494 bytes 3617265 (3.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Etape 2 : écrire le script suivant dans MATLAB :

```
programme1.m  x +
1      rosinit('10.171.70.85');
2      |
```

- Etape 3 : vérifier que le node a bien été créé dans la machine virtuelle avec la commande suivante :

```
insa@vm-0452:~/catkin_ws$ rosnode list
/matlab_global_node_54169
/rosout
/rostopic_4443_1683819683277
/teleop_turtle
/turtlesim
insa@vm-0452:~/catkin_ws$
```

- Etape 4 : Ecrire le programme dans le script MATLAB :

TUTO TEST

```
programme1.m
1  rosshutdown
2  % ferme la connexion existante si il y en a une
3  rosinit('10.171.70.85')
4  % initialise la connexion
5
6  TwistSub = rossubscriber('/turtle1/cmd_vel','geometry_msgs/Twist')
7  %création d'un subscriber 'TwistSub' sur le Node 'turtle1' pour un topic de type
8  %'cmd_vel' sur le message 'geometry_msgs' avec des paramètres 'Twist'
9  cmPub = rospublisher('/HANSEN/cmd_vel','geometry_msgs/Twist')
10 %création d'un publisher 'cmPub' sur le Node 'HANSEN' pour un topic de type
11 %'cmd_vel' sur le message 'geometry_msgs' avec des paramètres 'Twist'
12 cmMsg = rosmessage(cmPub)
13 %création d'un message
14
15 while 1
16 %boucle infini
17
18 TwistMsg = receive(TwistSub)
19 %On définit un message 'TwistMsg' qui reçoit les paramètres du subscriber 'TwistSub'
20
21 cmMsg.Linear.X = TwistMsg.Linear.X
22 cmMsg.Linear.Y = TwistMsg.Linear.Y
23 cmMsg.Linear.Z = TwistMsg.Linear.Z
24 cmMsg.Angular.X = TwistMsg.Angular.X
25 cmMsg.Angular.Y = TwistMsg.Angular.Y
26 cmMsg.Angular.Z = TwistMsg.Angular.Z
27 %Mise à jour sur le publisher
28
29 send(cmPub,cmMsg)
30 %on envoie le message sur la machine virtuelle
31 pause(0.01)
32 %création d'une temporisation
33
34 end
```

- Etape 5 : Lancer le programme sous MATLAB.

La tortue 'HANSEN' doit reproduire le mouvement de la tortue 'turtle1'.

QUESTION 4 : SUJET SIMULINK : La tortue 'HANSEN' doit suivre le mouvement de la tortue 'turtle1' :

- Etape 1 : on relève l'adresse IP de la machine virtuelle avec la commande « ifconfig » :

```
lnsa@vm-0452: ~/catkin_ws
lnsa@vm-0452:~/catkin_ws$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.171.70.85 netmask 255.255.0.0 broadcast 10.171.255.255
    inet6 fe80::e4c:3008:dbf7:b72d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a4:07:ed txqueuelen 1000 (Ethernet)
    RX packets 215297 bytes 13705848 (13.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 563 bytes 58350 (58.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 35494 bytes 3617265 (3.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 35494 bytes 3617265 (3.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Etape 2 : écrire le script suivant dans MATLAB :

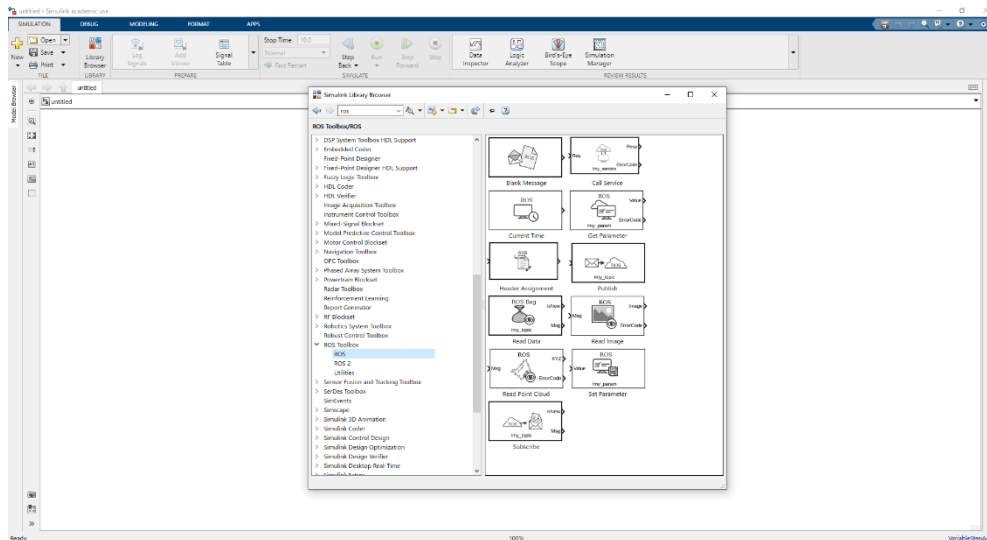
```
programme1.m
1  rosinit('10.171.70.85');
2  |
```


TUTO TEST

- Etape 3 : vérifier que le node a bien été créé dans la machine virtuelle avec la commande suivante :

```
lnsa@vm-0452:~/catkin_ws$ roscore list
/matlab_global_node_54169
/rosout
/rostopic_4443_1683819683277
/teleop_turtle
/turtlesim
lnsa@vm-0452:~/catkin_ws$
```

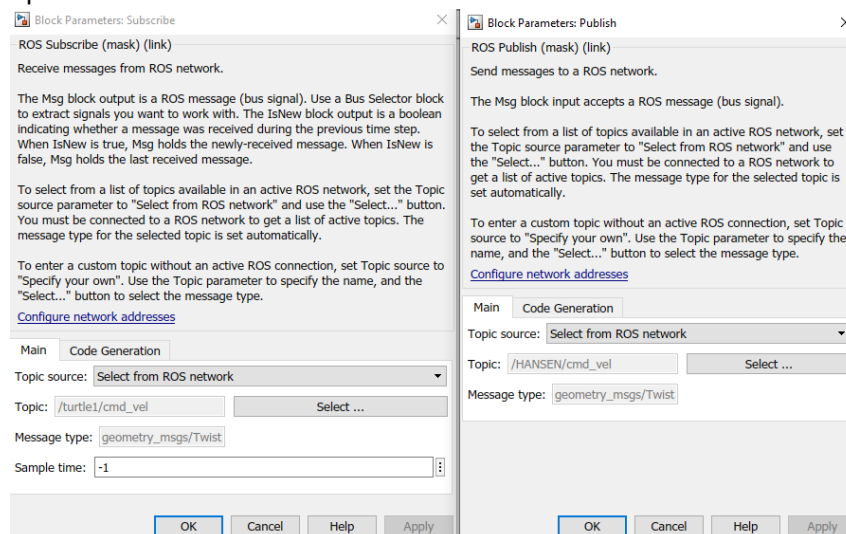
- Etape 4 : créer un fichier SIMULINK et ouvrir la « ROS toolbox » :



- Etape 5 : sélectionner les deux blocs « Subscribe » et « Publish » :

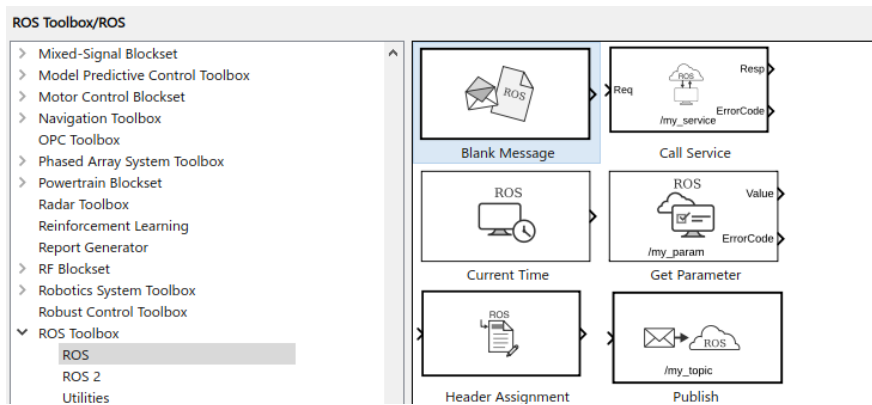


- Etape 6 : on paramètre ces deux blocs « Subscribe » et « Publish » :

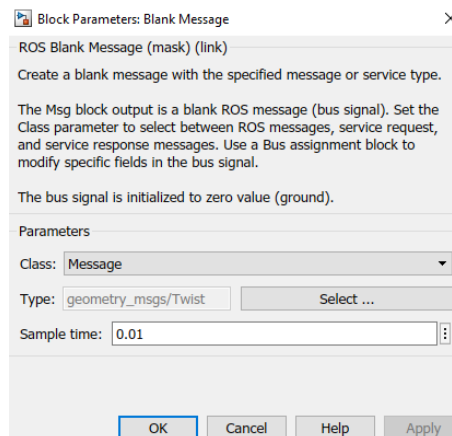


- Etape 7 : on ajoute un bloc « Blank Message » :

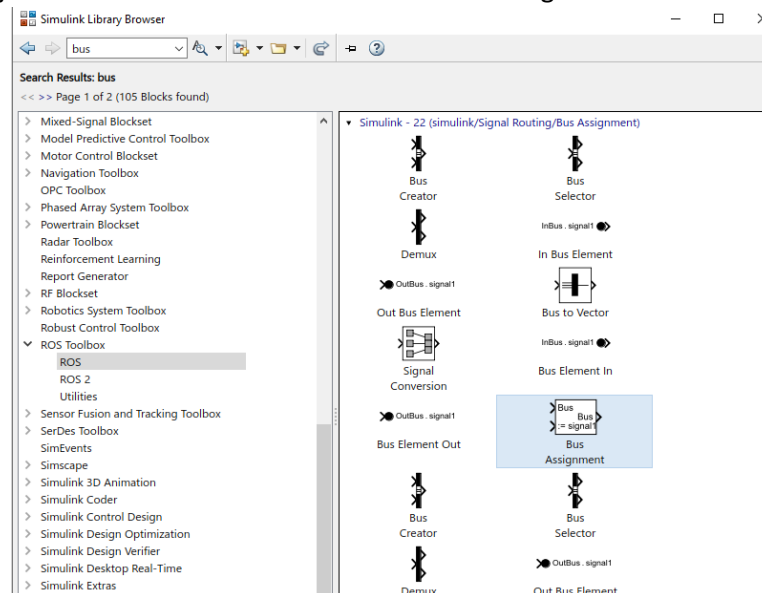
TUTO TEST



- Etape 8 : on paramètre ce bloc « Blank Message » :

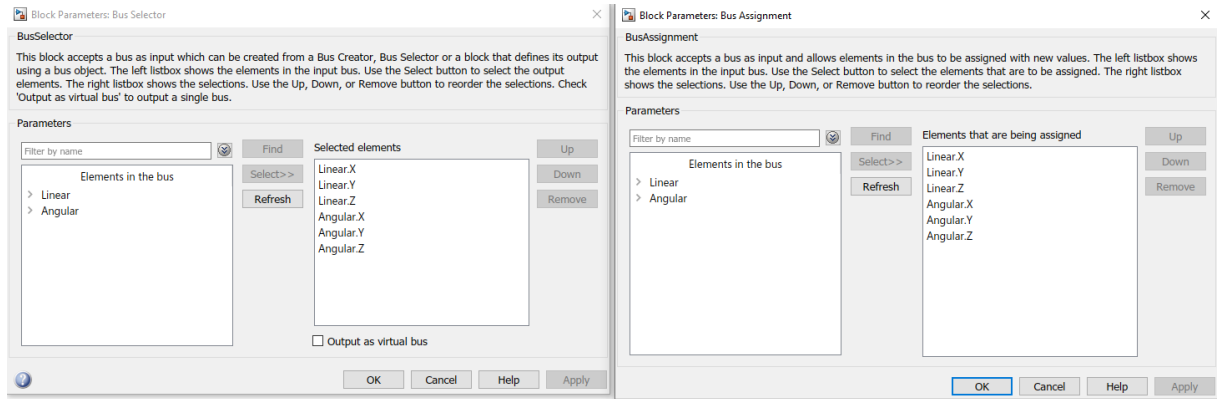


- Etape 9 : on ajoute deux blocs « Bus selector » et « Bus assignment » :

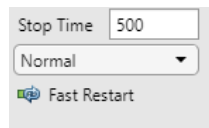


- Etape 10 : on paramètre ces deux blocs « Bus selector » et « Bus assignment » :

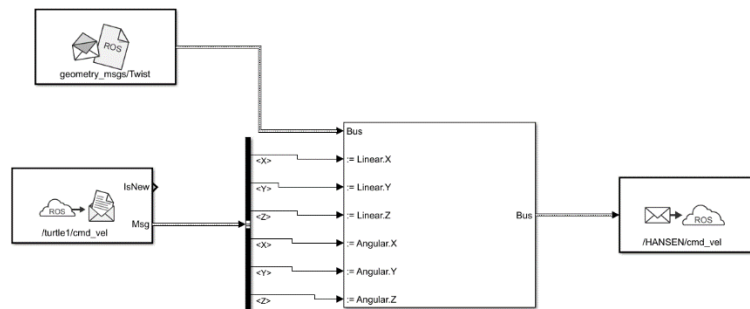
TUTO TEST



- Etape 11 : on change la durée de simulation :



- Etape 12 : on obtient le schéma suivant :



- Etape 13 : on lance le programme.

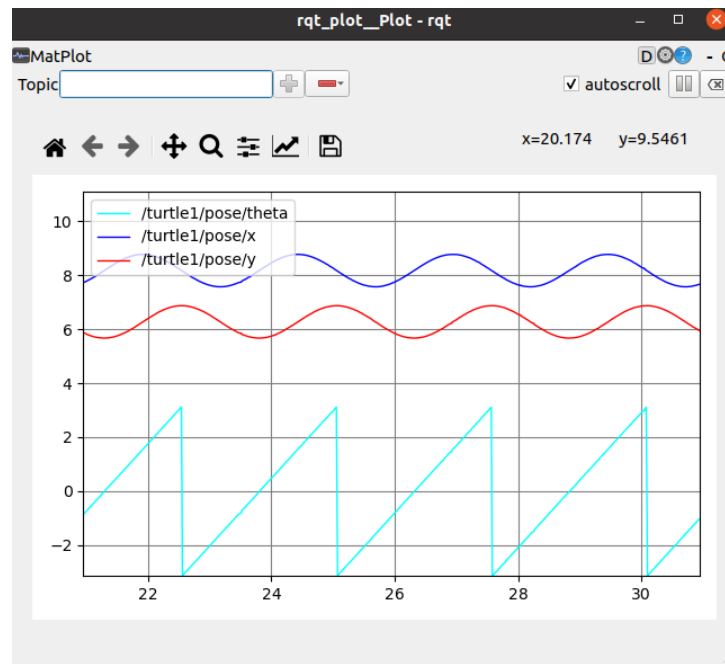
QUESTION 5 : Afficher les courbes de position pour les deux tortues via RQT :

- Etape 1 : ouvrir un nouveau terminal : « ctr + alt+ T »
- Etape 2 : Lancer la commande suivante :

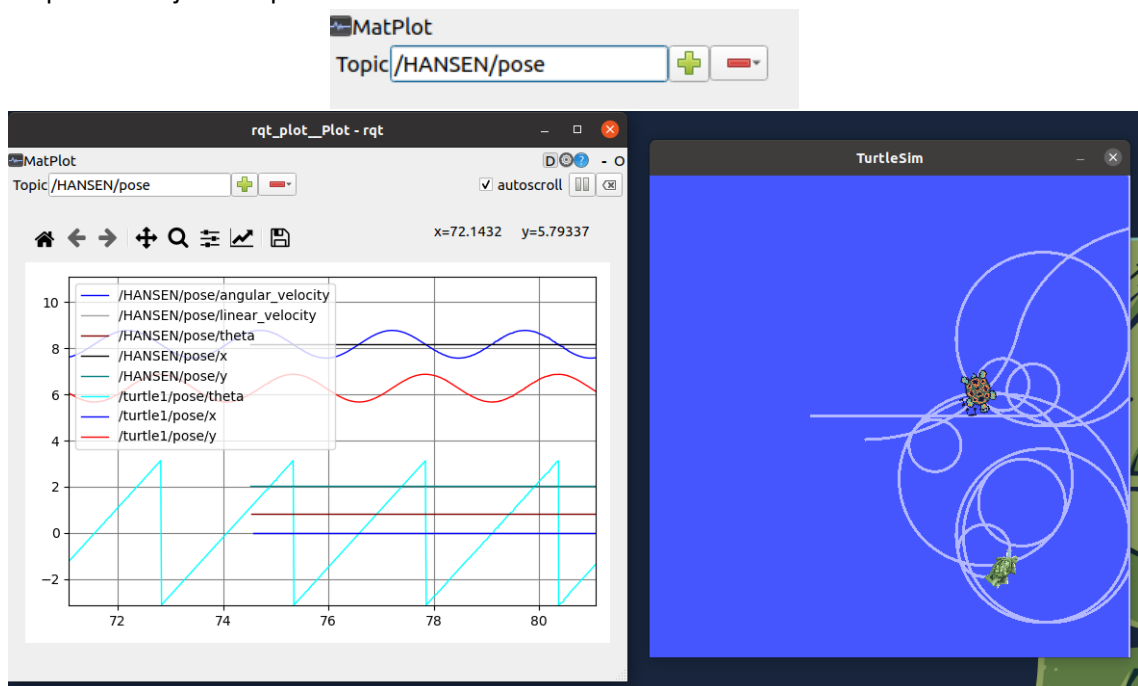
```
lnsa@vm-0452:~$ roslaunch rqt_plot rqt_plot
```

La fenêtre suivante s'ouvre alors pour la courbe de position de la tortue 'turtle1'

TUTO TEST



- Etape 4 : On ajoute la position de la tortue 'HANSEN' :



QUESTION BONUS : Faire bouger la tortue du PROF avec IP = « 10.171.70.80 » :

- Etape 1 : fermer le roscore ainsi que les autres Terminaux.
- Etape 2 : Ouvrir un Terminal avec « ctr + alt +T » puis modifier le fichier « .bashrc » avec la commande suivante :

```
ins@vm-0452:~$ sudo nano .bashrc
[sudo] Mot de passe de insa :
```

Rajouter les lignes suivantes en fin de fichier :

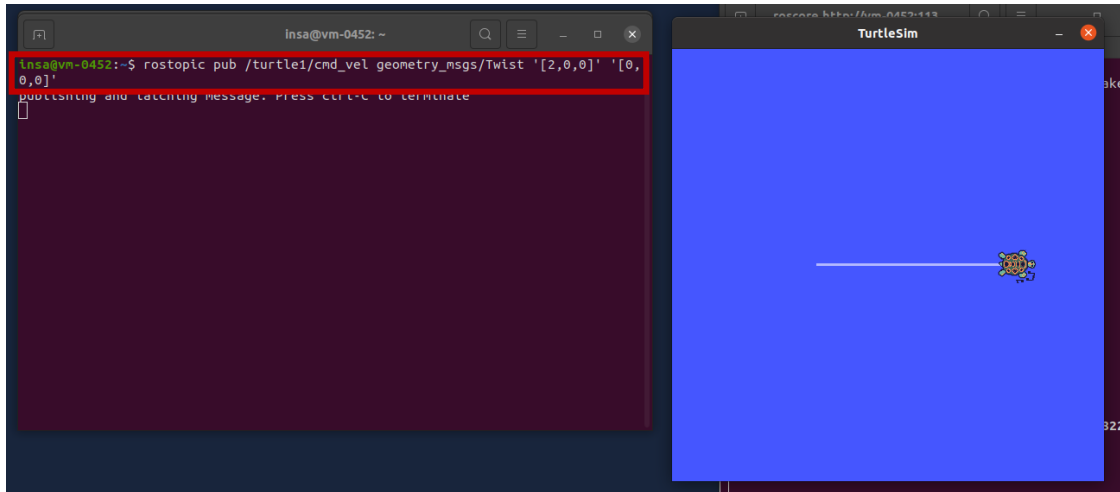
TUTO TEST

```
source /opt/ros/noetic/setup.bash

export ROS_MASTER_URI=http://10.171.70.80:11311
export HOSTNAME=:10.171.70.85
```

Enregistrer « ctr +x » puis « 0 » puis « entrée »

- Etape 3 : ouvrir un nouveau terminal et mettre la commande suivante :



La tortue du prof doit tourner.