Simulating a BlueBoat on an Ubuntu laptop using SITL and control it from ROS

Without BlueOS

Download **ardurover** SITL built for Linux x86_64:

https://firmware.ardupilot.org/Rover/stable-4.2.3/SITL_x86_64_linux_gnu/

or

https://firmware.ardupilot.org/Rover/stable-4.5.7/SITL_x86_64_linux_gnu/

In a terminal tab in the **Downloads** folder:

chmod +x ./ardurover

./ardurover --home 48.199,-3.015,122,0 --model rover-skid --speedup 1[1]

Then connect to it in TCP (e.g. 127.0.0.1, port 5760) with Mission Planner or QGroundControl and set ArduRover parameters:

- **BRD_SAFETY_DEFLT** to **0** (**BRD_SAFETYENABLE** in ArduRover < V4.4.0).
- **ARMING_CHECK** to **0**.
- **FRAME_CLASS** to **2** (means it is a boat).
- **SERVO1_FUNCTION** to **73** and **SERVO3_FUNCTION** to **74** since the BlueBoat is as described on https://ardupilot.org/rover/docs/rover-motor-and-servo-connections.html#skid-steering. Possibly swap **SERVO1_FUNCTION** and **SERVO3_FUNCTION**, toggle **SERVO1_REVERSED** and/or **SERVO3_REVERSED** if the simulated boat is turning/moving forward in the wrong direction compared to the real one.
- **PILOT_STEER_TYPE** to **3**? Other parameters?
- Maybe **SYSID_MYGCS** (might be removed in recent versions) or **MAV_GCS_SYSID** and **MAV_GCS_SYSID_HI** to **255**, as well as **SYSID_THISMAV** (might be removed in recent versions) or **MAV_SYSID** to **1**, with the corresponding mavros **apm2.launch** parameters **fcu_url:="tcp://127.0.0.1:5762/?ids=255,191" tgt_system:=1 tgt_component:=1**[2][3]. Change them accordingly if needed to match the parameters on the real boat (note that

[1] Note that there is also **motorboat-skid** instead of **rover-skid**, but its behavior appears to have issues when reaching waypoints in Guided mode…

[2] TCP 5762 is enabled by default by SITL as soon as something is connected to TCP 5760.

[3] Note that it would be rather **fcu_url:=" udp://127.0.0.1:14551@127.0.0.1:14550/?ids=255,191" tgt_system:=1 tgt_component:=1** for UDP if all is on the same computer, or **fcu_url:=" udp://:14550@192.168.2.2:14550/?ids=255,191" tgt_system:=1 tgt_component:=1** to communicate with a real BlueBoat, to be checked.

mavros appears to send by default to sysid 1, compid 1 as sysid 1, compid 191 (or 240 for ROS1)...).

In another tab for ROS Noetic:

sudo apt-get install ros-noetic-mavros ros-noetic-mavros-extras

sudo /opt/ros/noetic/lib/mavros/install_geographiclib_datasets.sh

roslaunch mavros apm2.launch fcu_url:="tcp://127.0.0.1:5762/?ids=255,191" tgt_system:=1 tgt_component:=1

Or for ROS 2 Jazzy:

sudo apt-get install ros-jazzy-mavros ros-jazzy-mavros-extras

sudo /opt/ros/jazzy/lib/mavros/install_geographiclib_datasets.sh

ros2 run mavros mavros_node --ros-args --param fcu_url:="tcp://127.0.0.1:5762" --param tgt_system:=1 --param tgt_component:=1 --param system_id:=255 --param component_id:=191

(note that for an unknown reason, **ros2 launch mavros apm.launch fcu_url:="tcp://127.0.0.1:5762/?ids=255,191" tgt_system:=1 tgt_component:=1** might ignore the **?ids=255,191** part...)

In another tab:

rqt

In **Plugins\Topics\Message Publisher**, add and then enable at 10 Hz:

/mavros/rc/override (with e.g. channel[0]=1400, channel[2]=1750)[4].

Ensure the simulated boat is armed[5] and in **Manual** mode e.g. using Mission Planner or QGroundControl buttons[6] and ensure their joystick is disabled (they may conflict since Mission Planner and QGroundControl might have the same sysid as mavros).

---

[4] Or equivalently from the terminal with ROS2:
ros2 topic pub -r 1 /mavros/rc/override mavros_msgs/msg/OverrideRCIn "{channels: [1400,0,1750,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}"

[5] Forcing to arm with ROS2:
ros2 service call /mavros/cmd/command mavros_msgs/srv/CommandLong "{command: 400, confirmation: 0, param1: 1.0, param2: 21196.0, param3: 0.0, param4: 0.0, param5: 0.0, param6: 0.0, param7: 0.0}"

[6] Or to set **Manual** mode from the terminal with ROS2:
ros2 service call /mavros/set_mode mavros_msgs/srv/SetMode "{custom_mode: 'MANUAL'}"

Alternatively, **/mavros/setpoint_velocity/cmd_vel**[7] can be used instead of **/mavros/rc/override** if the boat is in **Guided** mode[8].

The simulated boat should move in Mission Planner or QGroundControl view, this can be also checked by showing the values of e.g. **/mavros/global_position/compass_hdg** and **/mavros/global_position/global** in rqt **Plugins\Topics\Topic Monitor**.

Then you can use your own ROS nodes to send/receive those ROS messages in place of rqt, those nodes should be compatible with both the real BlueBoat and the simulated one **(to be checked with a real BlueBoat)**.

With BlueOS

To be closer to the real BlueBoat, try https://github.com/bluerobotics/BlueOS#running-via-docker-docker-run[9] and see how to configure the BlueOS docker and SITL to connect together and get mavros to connect to the MAVLink endpoints created in BlueOS (instead of mavros connecting directly to SITL):

```
sudo cp -f /etc/resolv.conf /etc/resolv.conf.host
mkdir -p /tmp/workspace/var/logs/blueos
mkdir -p /tmp/workspace/.config
mkdir -p /tmp/workspace/tmp/wpa_playground
mkdir -p /tmp/workspace/etc/blueos
mkdir -p /tmp/workspace/usr/blueos/{bin,extensions,userdata}
docker run --privileged --network=host --pid=host --name=blueos-core \
 --mount type=bind,source=/dev/,target=/dev/,readonly=false \
 --mount type=bind,source=/sys/,target=/sys/,readonly=false \
 --mount
type=bind,source=/var/run/wpa_supplicant,target=/var/run/wpa_supplicant,readonly=fals
e \
 --mount
type=bind,source=/tmp/workspace/tmp/wpa_playground,target=/tmp/wpa_playground,re
adonly=false \
 --mount
type=bind,source=/var/run/docker.sock,target=/var/run/docker.sock,readonly=false \
```

---

[7] From the terminal with ROS2:
ros2 topic pub -r 1 /mavros/setpoint_velocity/cmd_vel geometry_msgs/msg/TwistStamped "twist: {linear: {x: 0.5, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.5}}"

[8] Wait a little for the AHRS to be fully initialized before changing mode, otherwise Guided might be refused.

[9] You might need to remove the comments and any extra space or character after the ****, run before **sudo cp -f /etc/resolv.conf /etc/resolv.conf.host**, install an SSH server and ensure password authentication is enabled on it, change SSH_USER and SSH_PASSWORD to the ones of the computer, change **bluerobotics/blueos-core:master** to e.g. **bluerobotics/blueos-core:1.4.2** to use a specific version of BlueOS, recreate the temporary folders after each reboot, reboot if you get error messages saying that some UDP or TCP ports are already in use...

```
 --mount
type=bind,source=/tmp/workspace/var/logs/blueos,target=/var/logs/blueos,readonly=fals
e \
 --mount type=bind,source=/run/udev,target=/run/udev,readonly=true \
 --mount type=bind,source=$HOME/.ssh,target=/home/pi/.ssh,readonly=false \
 --mount
type=bind,source=/tmp/workspace/etc/blueos,target=/etc/blueos,readonly=false \
 --mount type=bind,source=/etc/machine-id,target=/etc/machine-id,readonly=true \
 --mount type=bind,source=/etc/dhcpcd.conf,target=/etc/dhcpcd.conf,readonly=false \
 --mount
type=bind,source=/tmp/workspace/usr/blueos/userdata,target=/usr/blueos/userdata,rea
donly=false \
 --mount
type=bind,source=/tmp/workspace/usr/blueos/extensions,target=/usr/blueos/extensions,
readonly=false \
 --mount
type=bind,source=/tmp/workspace/usr/blueos/bin,target=/usr/blueos/bin,readonly=false
\
 --mount
type=bind,source=/etc/resolv.conf.host,target=/etc/resolv.conf.host,readonly=true \
 --mount type=bind,source=/var/run/dbus,target=/var/run/dbus,readonly=false \
 --mount type=bind,source=/tmp/workspace/.config,target=/root/.config,readonly=false \
 --mount type=bind,source=/run/log/journal,target=/run/log/journal,readonly=true \
 --mount type=bind,source=/var/log/journal,target=/var/log/journal,readonly=true \
 -e BLUEOS_DISABLE_MEMORY_LIMIT=true \
 -e BLUEOS_DISABLE_STARTUP_UPDATE=true \
 -e SSH_USER=pi \
 -e SSH_PASSWORD=raspberry \
 bluerobotics/blueos-core:1.4.2
```

Then, open Firefox with address 127.0.0.1:80 to get BlueOS webpage, skip the wizard, enable **Pirate Mode**, go to **Autopilot Firmware**, click the **CHANGE BOARD** button, select SITL and click **SET** button, click **UPLOAD FIRMWARE FILE** button, browse to the **ardurover** executable, click **INSTALL FIRMWARE** button and then **RESTART AUTOPILOT** button. Then: **Vehicle Setup > CONFIGURE > PARAMETERS > RESET ALL PARAMETERS > REBOOT AUTOPILOT**, and go to **ArduPilot Parameters** and load the parameters proposed in the beginning of this document. Then using the **File Browser**, change **sitl_frame** from **vectored** to **rover-skid** in **configs/ardupilot-manager/settings.json** and add a line with **ROVER_SKID = "rover-skid"** after **ROVER = "rover"**[10] in **system_root/home/pi/services/ardupilot_manager/typedefs.py** and run **docker restart blueos-core** to restart the container.

Finally, Mission Planner, QGroundControl and/or mavros should be able to connect to TCP 5777 or UDP 14550, see previous part[11]...

---

[10] If it does not exist yet.

[11] In the previous part, TCP 5762 was enabled by default by SITL as soon as something was connected to TCP 5760. In BlueOS, TCP 5777 seems to appear by default instead of 5760 or 5762. Check the enabled MAVLink Endpoints in BlueOS and create new ones if needed to separate streams per program, and check the different SYSID parameters in the different programs...

In case BlueOS cannot connect any more to the SITL, try sudo **rm -f /tmp/workspace/.config/ardupilot-manager/firmware/eeprom.bin** to delete ArduRover parameters, or kill and delete all docker images, folder, files involved in the commands ran, i.e.

docker kill blueos-core
docker image rm -f $(docker images -q)
docker image prune -f
docker rm $(docker ps -qa --no-trunc --filter "status=exited")
sudo rm -rf /etc/resolv.conf.host
sudo rm -rf /tmp/workspace/var/logs/blueos
sudo rm -rf /tmp/workspace/.config
sudo rm -rf /tmp/workspace/tmp/wpa_playground
sudo rm -rf /tmp/workspace/etc/blueos
sudo rm -rf /tmp/workspace/usr/blueos/{bin,extensions,userdata}

References

https://firmware.ardupilot.org/Rover

http://wiki.ros.org/mavros

https://github.com/bluerobotics/Blueos-Parameter-Repository

https://ardupilot.org/rover/docs/parameters.html

https://mavlink.io/en/guide/routing.html

https://ardupilot.org/dev/docs/using-sitl-for-ardupilot-testing.html

https://www.ensta-bretagne.fr/lebars/tutorials/TD_robots_sensors_actuators.pdf

https://bluerobotics.com/learn/blueboat-software-setup/

https://discuss.bluerobotics.com/t/blueos-ros2-extension-v0-0-2-is-here/19324/

https://github.com/aminabyaneh/blueboat_rl_mavros