Deceptively Fast and Highly Stable Robot Dog

Deceptively fast and highly stable? Yes you heard right, today I am writing this guide for a robotic dog named A1 by a Chinese manufacturer Unitree Robotics.

A1 is a medium sized, fast and stable robot available in the market. The robot weighs 12KG and can have a payload of 5KG. It can run upto a velocity of 11.66 kph. A1 has an excelent athletic performance and is highly stable. Its motors are quite powerful and comes with different modes of control through the API provided by Unitree Robotics.

1 Unitree A1

The robot ships in a suitcase. The accessories included are ofcourse the robot itself, battery, adapter, controller and emergency stop. A1 configuration can be ordered either with Rasberry pi or with Nvidia boards. Here a video can be seen for unboxing of the A1 and here basic remote control usage can be seen. One thing which one need to make sure is that the robot is alligned on the ground with its legged folded like a real dog lie down(Can be seen in the above unboxing video) to power it on. If not the case the robot will not startup and stand. Here I will not go into more details of how to use it with the remote control but be focused on how to setup the programming environment and using ROS drivers.

Unitree Robotics have provided an SDK for their legged robots usage. This SDK at the backend is dependent upon LCM a lighweight message passing tool and for communication purposes they are using UDP for their so called realtime operation and communcation.

The first reccomended thing to run and test is the unitree_legged_sdk. A few examples are provided on how to use the SDK. So lets start first by building the driver.

1.1 Network Connection

Before the anything is installed or build, its good to connect to the robot via a static connection, to use the driver and SDK out of the box without any errors.

The robot is on `192.168.123.xxx` so you can either create a static connection driectly or may be with the commands below.

Check your network port first, network details can be seen by:

`$ ifconfig`

Let's say your network port is `enx000ec6612921`, so to create a static connection:

```
sudo ifconfig lo multicast
sudo route add -net 224.0.0.0 netmask 240.0.0.0 dev lo
sudo ifconfig enx000ec6612921 down
sudo ifconfig enx000ec6612921 up 192.168.123.162 netmask
    255.255.255.0
```

1.2 Installation

The driver dependencies are:

1. LCM(V1.4.0 or higher)

   LCM can be downloaded from here and can be installed by the following commands:

```
cd lcm-x.x.x
mkdir build
cd build
cmake ..
make
sudo make install
```

2. Boost(V1.5.4 or higher)

```
sudo apt install libboost-dev
```

3. Cmake(V2.8.3 or higher)

   `$ sudo apt install cmake`

Now lets build the unitree_legged_sdk by following commands:

```
git clone https://github.com/unitreerobotics/
    unitree_legged_sdk.git
mkdir build
cd build
cmake ../
make
```

After a successfull build now you have all the examples in `build`, try running some example just not forget the memeory locking and use `sudo` with every command you execute e.g.

```
sudo ./example_position
sudo ./example_walking
```

A1 is have two control modes

1. High level: Mainly for walking/running
2. Low level: For position/velocity/torque control

Beware: The robot will fall when in staring up the low level control.

1.3 ROS Driver

Unitree has provided a ROS driver here. This driver has also example uses of their high level and low level control modes. Lets start first by building this driver.

Dependencies:

```
sudo apt-get install ros-$ROS_DISTRO-controller-interface ros-
    $ROS_DISTRO-gazebo-ros-control ros-$ROS_DISTRO-joint-state-
    controller ros-$ROS_DISTRO-effort-controllers ros-
    $ROS_DISTRO-joint-trajectory-controller

mkdir -p ~/catkin_ws/src
cd ~/catkin_ws
git clone https://github.com/unitreerobotics/unitree_ros.git
    src/unitree_ros
```

You need to add the following lines to the file `unitree_gazebo/worlds/stairs.world`

```
<include>
  <uri>model:///home/unitree/catkin_ws/src/unitree_ros/
     unitree_gazebo/worlds/building_editor_models/stairs</
     uri>
</include>
```

Some environment variables settings required by the driver:

```
source /opt/ros/melodic/setup.bash
source /usr/share/gazebo-8/setup.sh # For ROS melodic Gazebo
    9 works(I have tested)
export ROS_PACKAGE_PATH=~/catkin_ws:${ROS_PACKAGE_PATH}
export GAZEBO_PLUGIN_PATH=~/catkin_ws/devel/lib:${
   GAZEBO_PLUGIN_PATH}
export LD_LIBRARY_PATH=~/catkin_ws/devel/lib:${
   LD_LIBRARY_PATH}
export UNITREE_LEGGED_SDK_PATH=~/unitree_legged_sdk
export ALIENGO_SDK_PATH=~/aliengo_sdk
#amd64, arm32, arm64
export UNITREE_PLATFORM="amd64"
```

Now just to build the driver run:

```
catkin build
source ~/catkin_ws/devel/setup.bash
```

Now you can run the examples both low and high level:

```
sudo su
source ~/catkin_ws/devel/setup.bash
roslaunch unitree_legged_real real.launch rname:=a1
   ctrl_level:=highlevel firmwork:=3_2
# In another terminal
rosrun unitree_legged_real walk_lcm
```

This above driver is just a wrapper around the examples provided in the `unitree_legged_sdk` with no specific ROS interfaces. Also the Gazebo simulation provided don't have walking robot.

In an effort at Quadruped Robotics, a ROS driver is being developed which can be found here. This driver has standard ROS communication interfaces and is currently being actively developed. Both high level and low level working modes are provided. In high level mode the robot can be teleoperated with standard ROS teleop tools. Same as in low level control robot can be controlled with rostopics and rosservices at the moment. An example controller for low level position control is provided here as well. A working simulation for the robot in Webots simulator is provided here with two demos to use.

To build the driver download the source code from here and copy/paste it inside 'catkin_ws/src' and run the following commands:

```
source /opt/ros/$ROS_DISTRO/setup.bash
cd ~/catkin_ws && catkin build
source devel/setup.bash
sudo apt update -qq
rosdep update
rosdep install --from-paths src --ignore-src -y
catkin build
source devel/setup.bash
```

You can use the driver in both modes for example:

```
sudo su
source ~/catkin_ws/devel/setup.bash
roslaunch mbs_unitree_ros high_level_mode.launch
```

Note: In case the links provided for the ROS driver does not work, please contact tahir.mehmood@mybotshop.de to get your copy.

1.4 References

1. Unitree Robotics, A1 More Dextrety More Possibilities.
   https://www.unitree.com/products/a1

2. Unitree Robotics, High performance robot manufacturer.
   https://github.com/unitreerobotics

3. Quadruped Robotics, Your future research partner.
   https://Quadruped.de/