# ОПРЕДЕЛЕНИЕ НАПРАВЛЕНИЯ ОСЕЙ

Когда все шарниры находятся в нулевом положении, система координат выглядит так, как показано на рисунке 1. Красная ось обозначает ось X, зелёная — ось Y, а синяя — ось Z. Изображение взято из [официальной документации Unitree](https://support.unitree.com/home/en/H1_developer)**.**

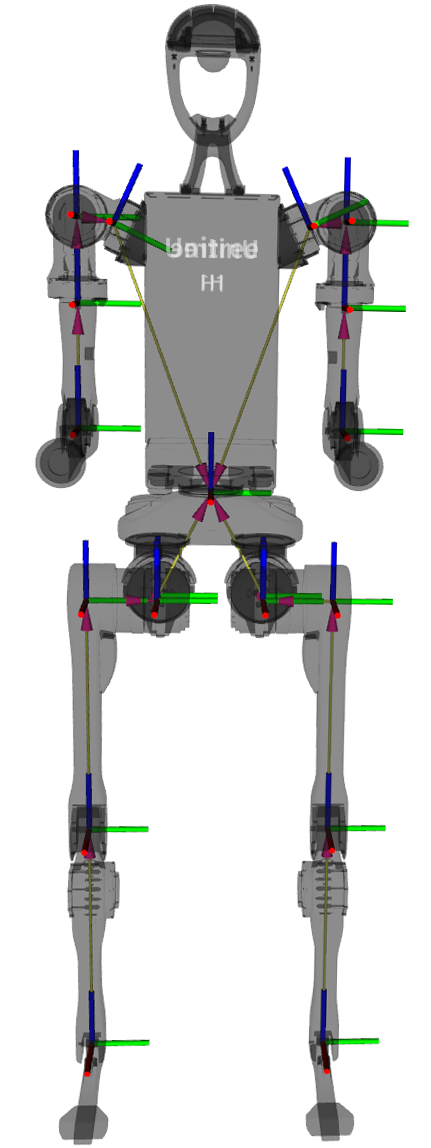


Рисунок 1 - *Оси координат, когда шарниры находятся в нулевом положении*

## Опционально:

Вы также можете посмотреть эту визуализацию вживую и поэкспериментировать с моделью робота в Rviz. Для этого необходимо:

1. **Установите Rviz** (обычно входит в состав ROS2).
2. **Установите и соберите репозиторий** [unitree\_h1\_visualization\_ws](https://github.com/cyberbanana777/unitree_h1_visualization_ws) — следуя приложенной инструкции.
3. **Запустите систему.**

Запустите launch-файл для визуализации модели робота и графического интерфейса для управления положением суставов:

ros2 launch completed\_scripts\_visualization show.launch.py mode:=without\_hands launch\_rviz:=True robot:=empty launch\_control\_by\_gui:=True

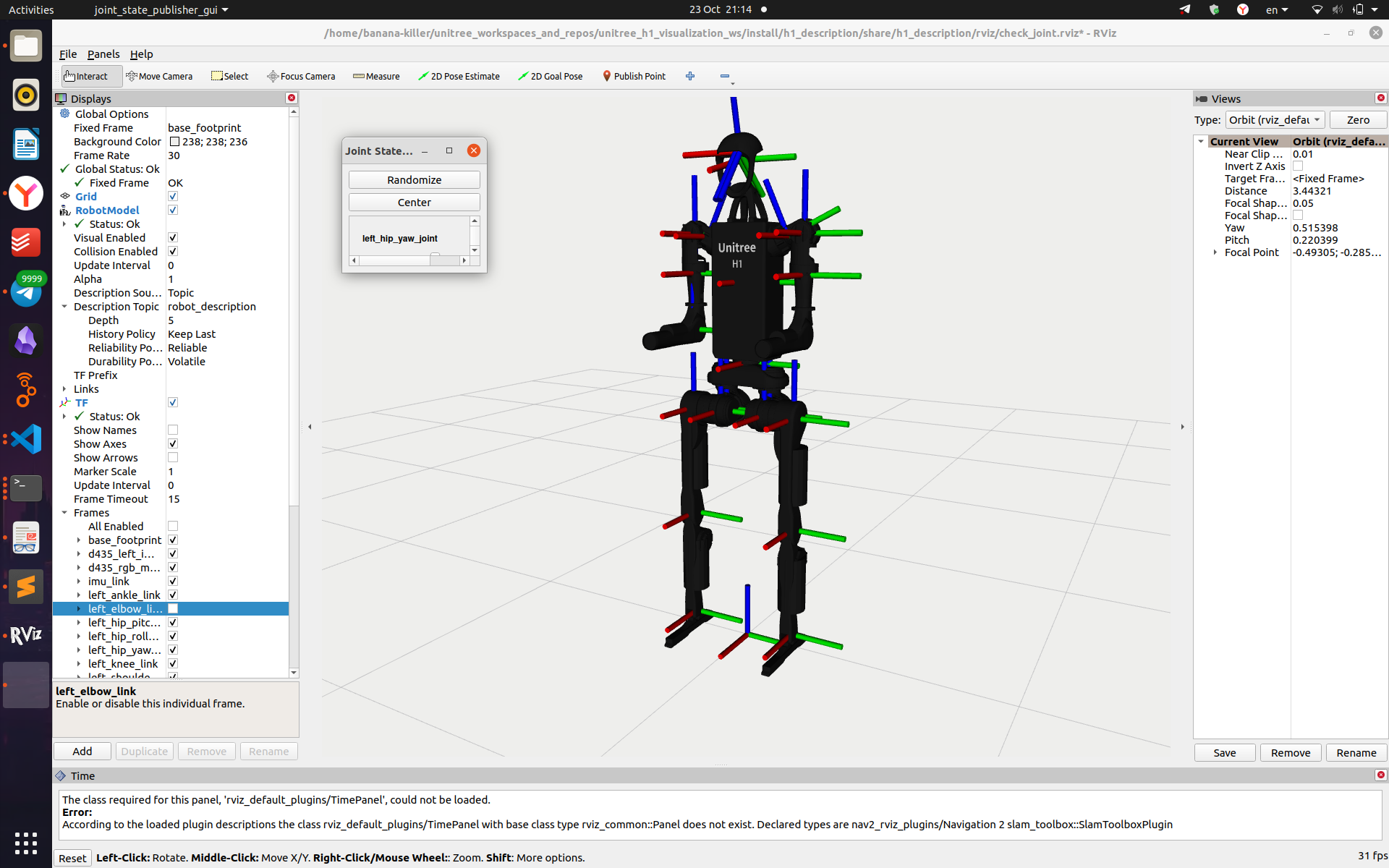


Рисунок 2 - Робот в Rviz. Красная ось обозначает ось X, зелёная — ось Y, а синяя — ось Z.

В блоке TF:

* Пункт *Show Axes* — позволяет включить или отключить отображение всех осей.
* В разделе *Frames* — можно включить или отключить отображение определённых осей.

Растяните окно ***Joint State Publisher***, что бы увидеть все ползунки. Передвигая полузнки, управляйте углами в суставах робота.

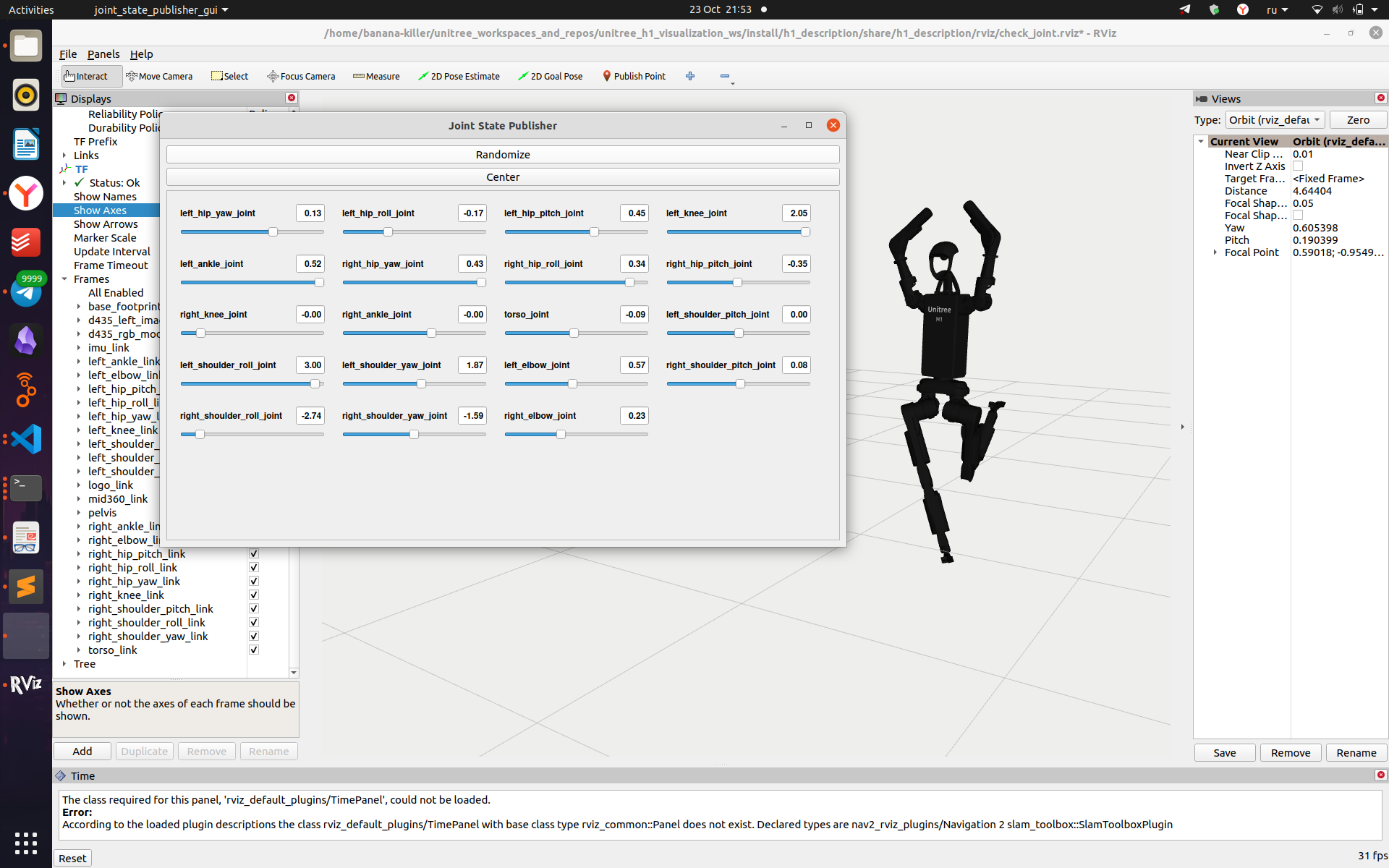


Рисунок 3 - Управление углами в суставах робота в Rviz с помощью gui интерфейса.

### Дополнительная возможность: управление реальным роботом с помощью графического интерфейса

Для управления роботом с помощью графического интерфейса ознакомьтесь с методическими указаниями [7\_Как\_отправить\_управляющий\_сигнал\_на\_робота](https://github.com/cyberbanana777/unitree_h1_docs/blob/main/instructions_and_manuals/7_Как_отправить_управляющий_сигнал_на_робота.md). Установите и соберите репозиторий [unitree\_h1\_control\_ws](https://github.com/cyberbanana777/unitree_h1_control_ws) — следуя приложенной инструкции, если ещё этого не сделали.

Затем в отдельном терминале запустите ноду для отправки управляющих сигналов, чтобы задавать движение роботу:

ros2 run low\_level\_control low\_level\_control\_without\_hands\_node  
# (по умолчанию target\_topic\_param = "arm\_sdk")

Запустите графический интерфейс ***gui\_control\_node***:

ros2 run gui\_control gui\_control\_node

Эта нода публикует в топик ***/position\_to\_unitree***. Подробнее смотрите здесь [gui\_control\_node](https://github.com/cyberbanana777/unitree_h1_control_ws?tab=readme-ov-file#gui_control).

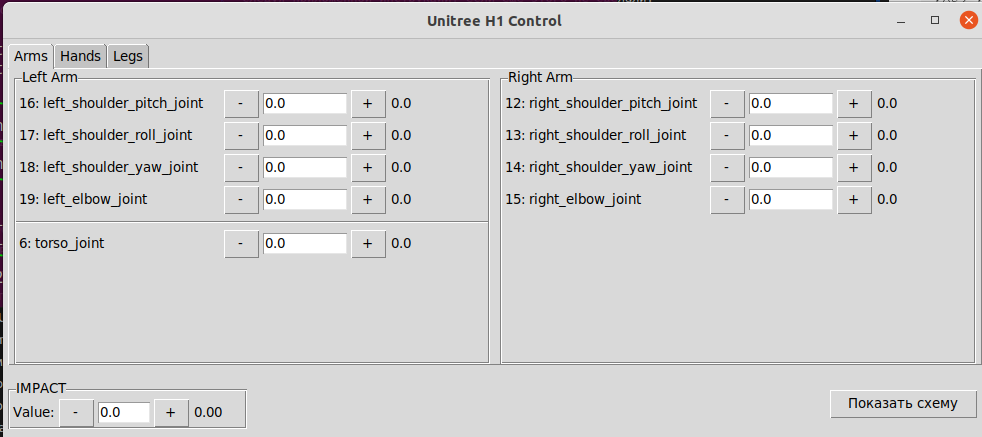


Рисунок 4 - Графический интерфейс для управления роботом.

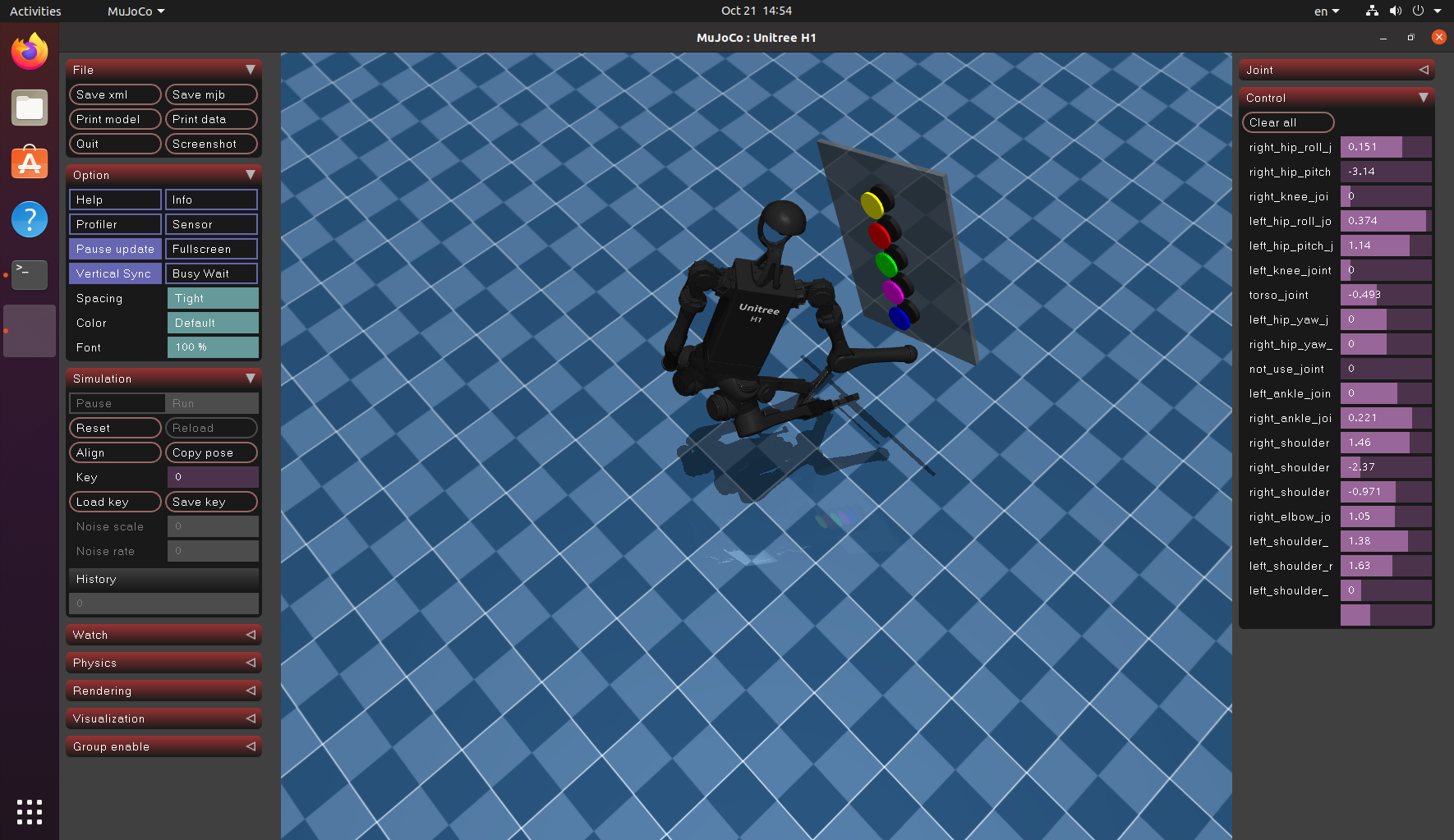
Можно вводить координаты в поле с клавиатуры или изменять их кнопками «+» и «–».

## Определение направления вращения для каждого звена в зависимости от поданного на мотор значения (угла)

Если выполнить инструкцию из опционального раздела предыдущего пункта, вы сможете управлять роботом через графический интерфейс в Rviz и наблюдать, в какую сторону будет вращаться каждое звено.

В общем случае можно реализовать то же самое в *MuJoCo*. Для этого:

1. **Установите MuJoCo** по инструкции [5\_Как\_установить\_Mujoco\_вместе\_с\_моделями\_Unitree](https://github.com/cyberbanana777/unitree_h1_docs/blob/main/instructions_and_manuals/5_Как_установить_Mujoco_вместе_с_моделями_Unitree.md).
2. **Запустите симуляцию** — в правой части окна визуализатора вы найдёте раскрывающееся поле ***Control***.
3. **Экспериментируйте** — через этот интерфейс можно независимо управлять каждым мотором и визуально определять положительное/отрицательное направление вращения.



*Рисунок* 5 - Управление моторами в MuJoCo.