

Physics engines and instability problem

Running Gazebo with one of the 4 supported physics engines

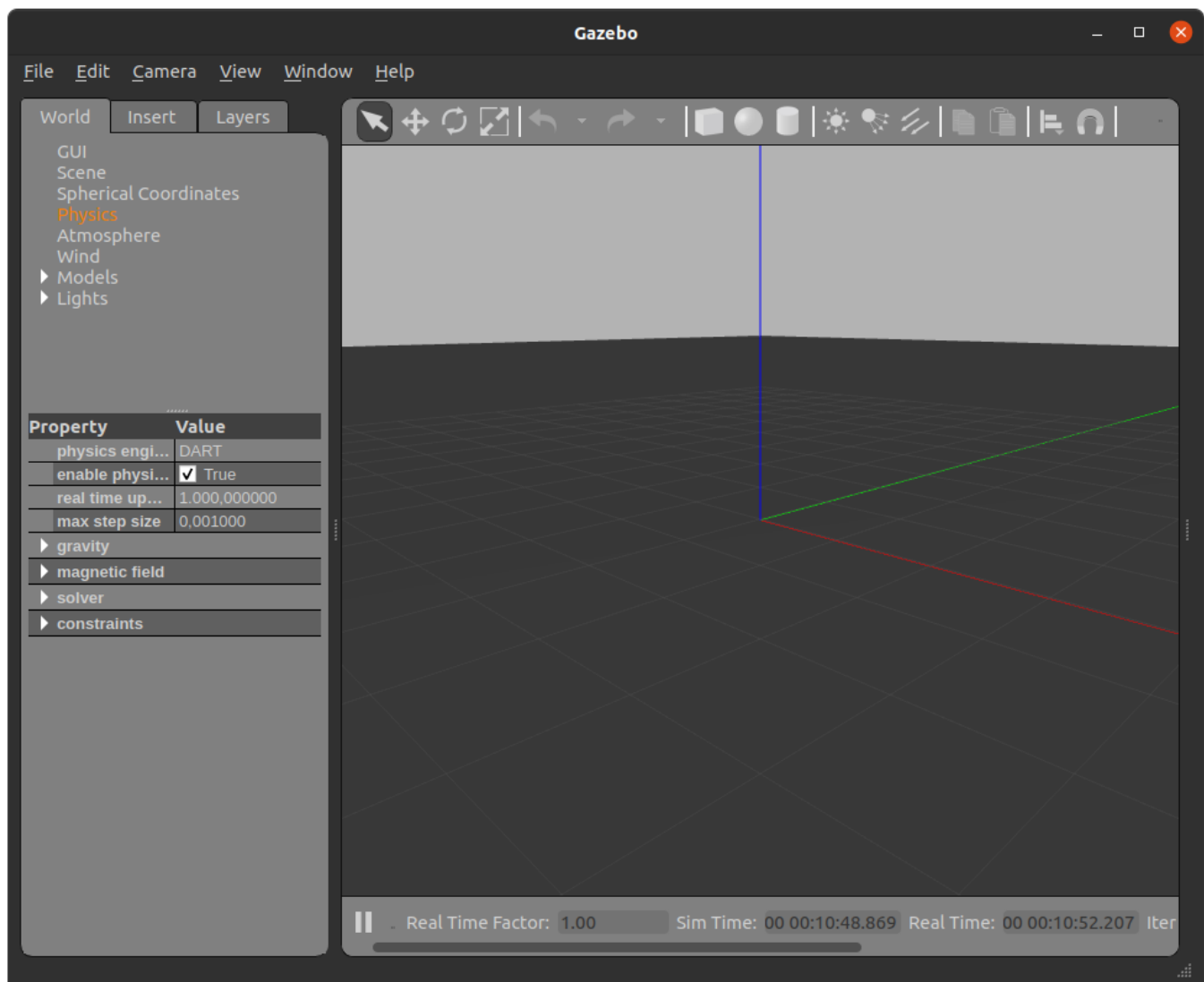
Gazebo [supports](#) 4 different physics engines bullet, DART, Simbody and the default ODE.

One can specify a physics engine as:

```
$roslaunch gazebo_ros gazebo -e [ --physics ] arg
```

where arg ode, bullet, dart or simbody. The gazebo then launches using a default "empty" world where no robot models exist but there is a ground plane and a light source (sun).

The selected engine name appears in the physics tab at the left part of the gazebo gui.



Problem

If any engine other than ode is selected the following error appears in the terminal.

```
[ERROR] [1663868150.470065076, 0.013000000]: ROS get_physics_properties
service
call does not yet support physics engine [engine name].
```

```

vasilisp@yoga: ~
roscore http://yoga:11311/

vasilisp@yoga:~$ roslaunch gazebo_ros gazebo -e dart
[ INFO] [1663868097.863143311]: Finished loading Gazebo ROS API Plugin.
[ INFO] [1663868097.863950318]: waitForService: Service [/gazebo/set_physics_properties] has not been advertised, waiting...
[ INFO] [1663868098.191739494]: waitForService: Service [/gazebo/set_physics_properties] is now available.
Warning [BodyNode.cpp:469] [BodyNode] A negative or zero mass [0] is set to BodyNode [link], which can cause invalid physical behavior or segfault. Consider setting positive value instead.
[ERROR] [1663868098.201251778, 0.003000000]: ROS get_physics_properties service call does not yet support physics engine [dart].
[ INFO] [1663868098.205314170, 0.007000000]: Physics dynamic reconfigure ready.
vasilisp@yoga:~$ Warning [World.cpp:353] [World::removeSkeleton] Skeleton [ground_plane] is not in the world.

vasilisp@yoga:~$ roslaunch gazebo_ros gazebo -e bullet
[ INFO] [1663868116.198847446]: Finished loading Gazebo ROS API Plugin.
[ INFO] [1663868116.199858649]: waitForService: Service [/gazebo/set_physics_properties] has not been advertised, waiting...
[ INFO] [1663868116.527441332]: waitForService: Service [/gazebo/set_physics_properties] is now available.
[ERROR] [1663868116.539557719, 0.003000000]: ROS get_physics_properties service call does not yet support physics engine [bullet].
[ INFO] [1663868116.543257821, 0.006000000]: Physics dynamic reconfigure ready.
vasilisp@yoga:~$ roslaunch gazebo_ros gazebo -e simbody
[ INFO] [1663868150.133000248]: Finished loading Gazebo ROS API Plugin.
[ INFO] [1663868150.133858937]: waitForService: Service [/gazebo/set_physics_properties] has not been advertised, waiting...
[ INFO] [1663868150.461747813, 0.006000000]: waitForService: Service [/gazebo/set_physics_properties] is now available.
[ERROR] [1663868150.470065076, 0.013000000]: ROS get_physics_properties service call does not yet support physics engine [simbody].
[ INFO] [1663868150.473967444, 0.017000000]: Physics dynamic reconfigure ready.
vasilisp@yoga:~$

```

In the [latest report](#), that i found, of this issue is stated that the DART engine works as expected despite of this error, `get_physics_properties` is a function that is not yet implemented for other engines than ode.

In addition to this error DART physics engine throws 2 warnings. The first is referring to a link in the world even though no links exist in the empty.world loaded by default. The second may appears after the gazebo gui is closed where `ground_plane` no longer exist, it is safe to ignore it.

Going back to ODE

Due to the problem above i decided to go back to ODE and try to stabilize the platform. I also noticed that sometimes the platform acts abnormally in DART.

In an effort to isolate the issue and make it easier for myself to fix the issue i simplified the robot's xacro file so that no libraries are loaded, only links and joints leaving me with `simple_robot.xacro`.

How to spawn the xacro in gazebo using any engine

To spawn the model in gazebo, run the ros master node

```
$roslaunch roscore
```

then in another terminal launch gazebo

```
$roslaunch gazebo_ros gazebo
```

Gazebo can also run with `-e` tag as before.

Then in a third terminal the xacro file has to be converted to urdf.

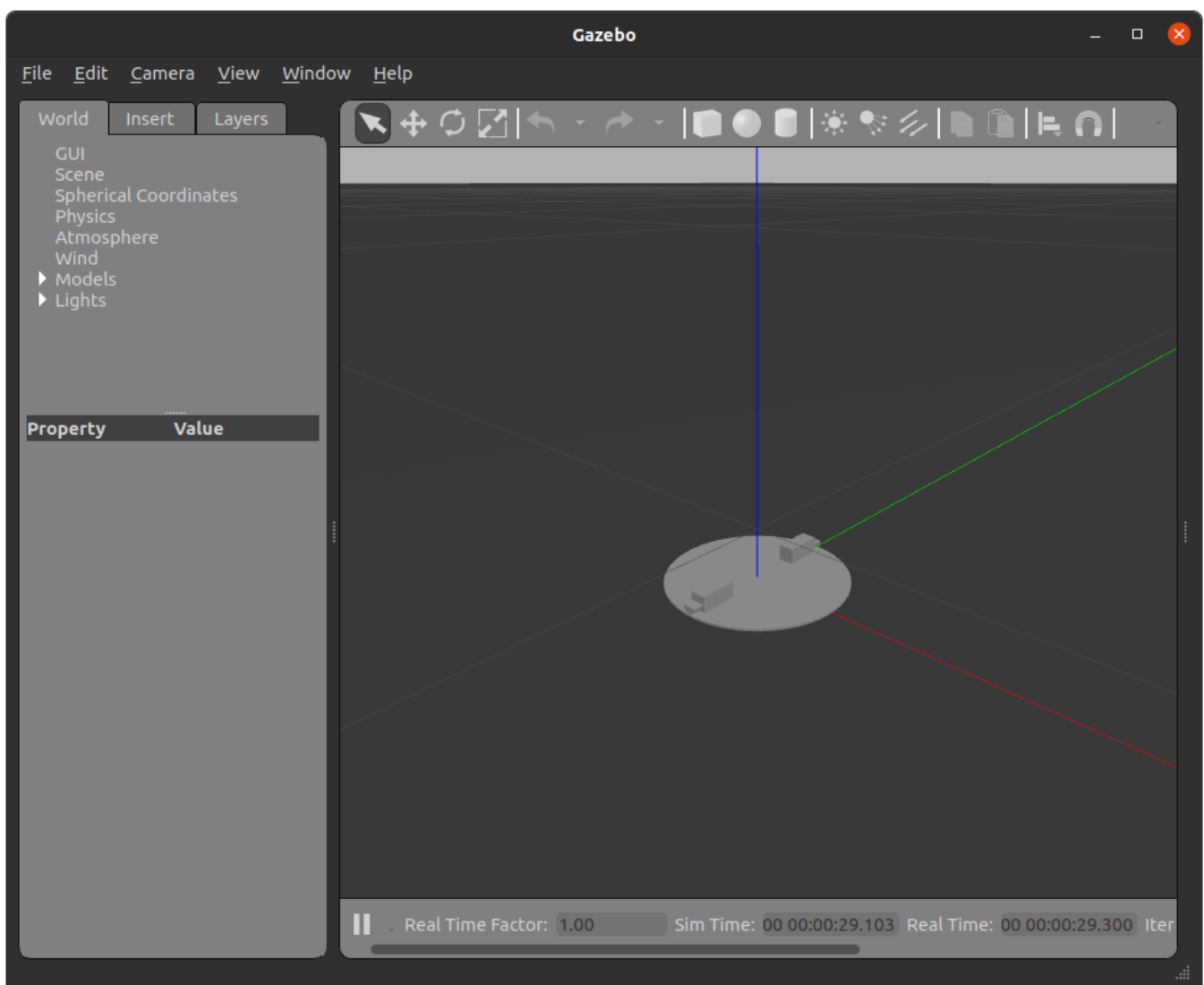
```
$roslaunch xacro xacro -o model.urdf simple_robot.urdf
```

where `model.urdf` is the name of the produced urdf.

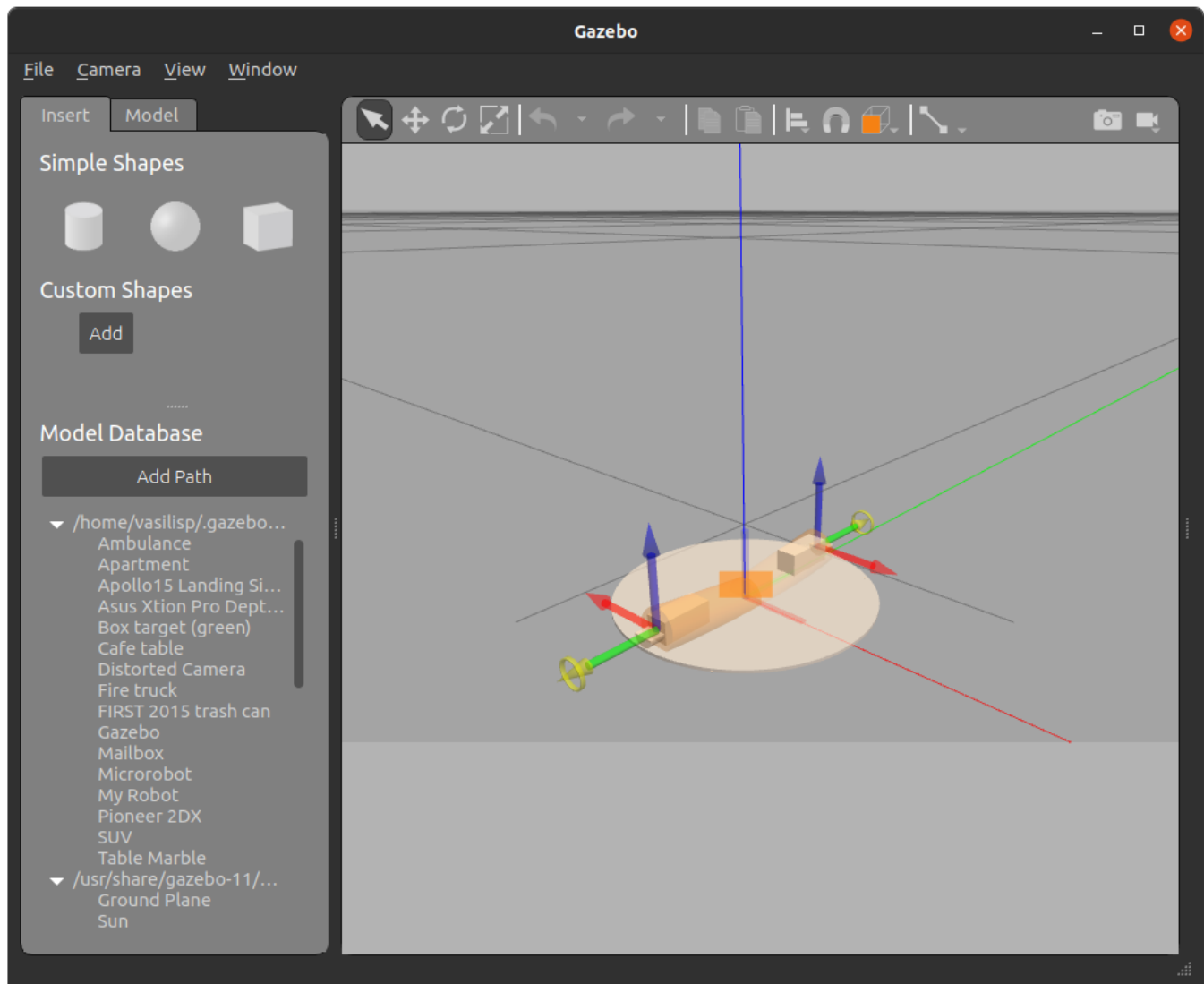
Finally to spawn the urdf use:

```
$roslaunch gazebo_ros spawn_model -file model.urdf -urdf -model microbot
```

you should see the robot in the gazebo gui.



The robot can be edited by selecting right clicking it and selecting "edit model".



Further simplification, save the model for faster trial and error

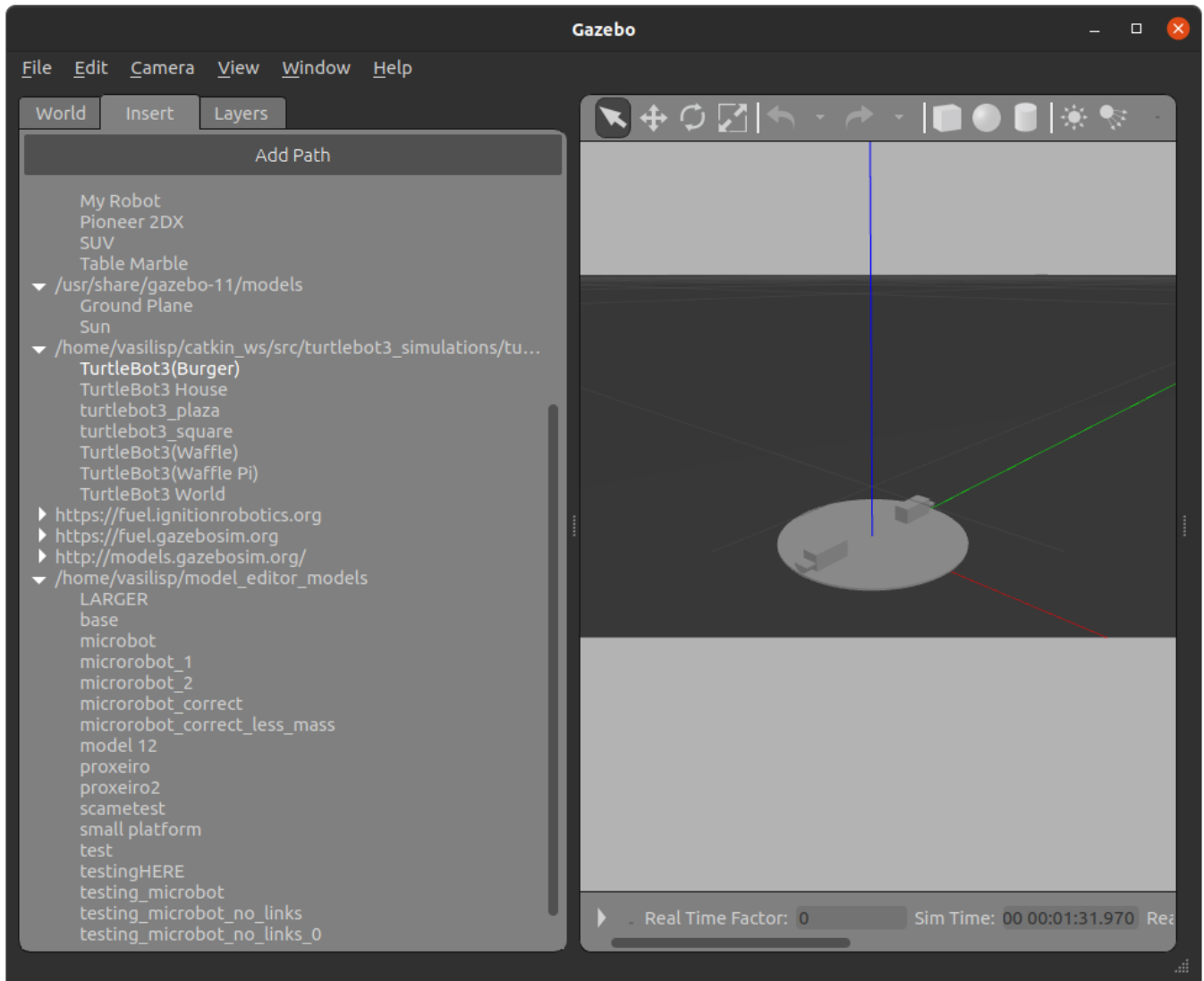
Going into the model editor right click the robot and select link inspector.



Both links (collision tab) and joints have a lot of physical variables, so i simplified the robot even further by removing the eccenters and links, leaving only the platform of the robot (Selected them in the model editor and pressed the delete key). I exited the model editor saving the model.

The instability of the platform was the same as before the simplification of the model confirming that problem lies in the physics parameters.

As i had the platform saved, i could edit the robot further using the model editor, i could delete the model and drag and drop it in the gazebo world from the insert tab (very bottom of the list "testing_microbot").



After a lot of trial and error, i noticed that the physics parameters that affected the instability were K_p , K_d , and Min depth. A very brief explanation for each one of those variables is given [here](#).

I managed to stabilize the back support completely with the following values.

- $k_p = 10^{16}$
- $k_d = 25 * 10^8$
- $\text{min_depth} = 1 * 10^{-6}$

Link Inspector

Name: base_link

Link

Visual

Collision

Restitution coefficient

0,00000000

Bounce threshold

1,00000000

Soft cfm

0,00000000

Soft erp

0,20000000

Kp

100000000000000000,000

N/m

Kd

2500000000,00000000

N/m

Max vel

0,01000000

m/s

Min depth

0,00000100

m

Collide without contact

☐ True

☒ False

+ Bounding Box

+ Bounding Sphere

+ Bounding Cylinder

VIS

COL

Reset

Cancel

OK