Atlas Balancing and Positioning in drcsim3.1
Wyatt Newman Nov 17, 2013

There are numerous problems with simulation under drcsim3.1, which makes even stabilizing Atlas standing difficult. These notes are a current update (and likely of temporary value, as drcsim evolves).

Changes to world model:
The task-7 world model is suitable for balancing Atlas. It can be launched with:
VRC_CHEATS_ENABLED=1 roslaunch drcsim_gazebo drc_practice_task_7.launch

However, the task-6 model has a diamond-plate surface that seems to present chattering problems in drcsim3.1. The diamond plate surface is replaced with a flat surface with the following edits to the world file:

```
+    <include>
+      <uri>model://ground_plane</uri>
+    </include>
-    <!-- <include> -->
-    <!--   <uri>model://pavement_plane</uri> -->
-    <!-- </include> -->
-    <!-- <include> -->
-    <!--   <uri>model://starting_pen</uri> -->
-    <!--   <pose>-12.31 -15 0 0 0 -1.5707</pose> -->
-    <!-- </include> -->
```

The world file for drc practice task 6 has been edited in this fashion and installed in the directory: hku_worlds/worlds/task_6.world.

Launch files have been modified to use this world model and installed in:
hku_worlds/launch/task_6.launch
hku_worlds/launch/drc_practice6_hku.launch

To start up drcsim using this modified world model, run:
VRC_CHEATS_ENABLED=1 roslaunch hku_worlds task_6.launch

Modifications to gains:
The gazebo/drcsim/ODE simulator interprets the gain parameter “kp_velocity” oddly. It indirectly affects simulation damping properties, and setting good values for these helps to suppress some of the simulator's vibrations. *However*, it seems that using values other than 0 for the ankles results in chatter between the robot and the ground surface. Suggested gains are:

```
  drcsim_3pt1_kvp_tork_ = {
        50.0, 50.0, 50.0, 50.0,
        50.0, 50.0, 50.0, 50.0, 0.0, 0.0,
        50.0, 50.0, 50.0, 50.0, 0.0, 0.0,
        50.0, 50.0, 50.0, 50.0, 50.0, 50.0,
        50.0, 50.0, 50.0, 50.0, 50.0, 50.0};
```

Additionally, a set of control gains that stabilizes Atlas while standing includes:

```
drcsim_3pt1_kp_tork_ = {
      3600, 6000.0, 6000.0, 60.0,
      2100.0, 3500.0, 5000.0, 4200.0, 13400.0, 1700.0,
      2100.0, 3500.0, 5000.0, 4200.0, 13400.0, 1700.0,
```

3000.0, 2500.0, 1600.0, 1600.0, 1600.0, 860,
3000.0, 2500.0, 1600.0, 1600.0, 1600.0, 860 };
for kp_position, and:
drcsim_3pt1_kd_tork_ = {
18.0, 3.0, 3.0, 3.0,
4.0, 34.0, 25.0, 42.0, 30.0, 6.0,
4.0, 34.0, 25.0, 42.0, 30.0, 6.0,
5.0, 49, 25, 25, 3.3, 1.7,
5.0, 49, 25, 25, 3.3, 1.7 };
for kd_position.

The above gains are installed as the default in the package:
lowLevelJointServoV3

However, these gains are not optimized. It is expected that there are better values that could be used.

Starting User Control:
As noted above, start Atlas with the modified task-6 world with:
VRC_CHEATS_ENABLED=1 roslaunch hku_worlds task_6.launch

After launching, wait for Atlas to descend and go into dynamic stand mode.

Start up the gravity calculator:
rosrun mass_calcs stand_torque_calc

Start up the low-level joint controller:
rosrun lowLevelJointServoV3 lowLevelJointServoV3

Choose the option for drcsim and the option for initializing in stand pose. After a few seconds of warm-up, Atlas will switch over to user control of all joints and balance under user control.

Repositioning Atlas:
Repositioning Atlas without him falling over can be tedious.

For task 7, repositioning is relatively simple. After Atlas has lowered and gone into dynamic-stand mode, you can “click” on his image, which will bring up a white, wireframe bounding box. Select the right/left/up/down icon from the menu bar above the graphical display. Three colored axes will appear. However, it is best not to use these for translating Atlas, since the x-axis (red axis) is not parallel to the ground. Instead, “hover” over Atlas with the mouse until the mouse pointer turns to a hand icon (but not coincident with one of the colored axes). You can then click-drag Atlas horizontally along the floor, and he will keep his balance. In this manner, you can position him in front of the valve wall within a viable reach of a valve. For such translations, Atlas can be repositioned equally well in start-up (dynamic-stand) mode, or under control of the version-3 low-level controller.

For task 6, to reposition Atlas in front of the table with the drill requires both rotation and translation. Rotation is the harder task, and it is helpful to do this first. Also, Atlas seems to be more tolerant of incremental, imperfect rotations using the version-3 low-level controller than in start-up, dynamic-stand control mode.

To rotate Atlas, first “select” him by clicking on the model. When successful with this selection, he will be enclosed in a white, wire-frame box. It may be difficult to select him instead of the wall, but this can be made easier by rotating to a top-down view, then selecting Atlas.

To rotate Atlas, select the “rotate” icon from Gazebo's menu bar above the graphical display. Red, green and blue “orbit” rings will appear to enable commanding rotations about x, y and z axes, respectively. Unfortunately, Atlas' z reference axis is not normal to the floor, so performing a z-rotation causes misalignment between his foot and the floor, causing him to fall down. However, this can be accomplished in increments.

With the orbit circles visible, resume Gazebo simulation (click the “pause” button to toggle back to “run”). Hover over the blue ring (rotation about z), and its color will intensify. You can then move the cursor (click-drag) by a small amount. If the resulting disturbance is relatively small, Atlas will restabilize. You can then command another rotation increment. Continue in this fashion until you have Atlas reoriented. If he falls during this process, you can use the “reset model poses” command under the “edit” drop-down menu and try again. (You may need to reset a few times to get him to restabilize at his starting position).

Once Atlas is reoriented, you can translate him to the table. This can be done by clicking the right/left/up/down icon in the menu bar above the graphical display. Red, green and blue axes will appear. If you hover over one of these, it will intensify its color, and you can then click-drag Atlas along the selected axis. The y-axis (green arrow) is nearly parallel to the ground, so large motions can be commanded with this axis without inducing falls. The x-axis (red arrow), however, is not quite parallel to the ground, so motions in this direction should be performed incrementally, allowing Atlas to restabilize. Alternatively, if you “grab” the wireframe box (not selecting any colored arrow), you can translate Atlas arbitrarily without inducing falls.

After repositioning Atlas, in order to focus on manipulation, it can be useful to pin the pelvis (or torso), which can be done with the command:
rostopic pub --once /atlas/mode std_msgs/String '{data: "pinned_with_gravity"}'

If Atlas falls, choosing “reset model poses” from the edit menu should let you start over with repositioning him. You may need to invoke “reset model poses” multiple times to get him balanced again. This works under v3 low-level control, but not under “dynamic stand” default control.

Acknowledgements:
My thanks to our collaborator from TORK, Masaki Murooka, for his advice on editing the task-6 world model, setting the kvp gains, and suggesting a set of kp and kd gains to stabilize Atlas!