CS26410 Report 2
Simulation vs Reality

Chris Savill

chs17@aber.ac.uk

March 10, 2013

Contents

1 Difference Between Simulation and Real Robots

The simulation of a robot in a virtual environment and the reality of a real robot in a real world environment can yield very different results. The main cause for this is that the real world is a very complex environment with many uncertainties. Not all of these uncertainties can be dealt with or even simulated properly.

In the case of the a Pioneer robot being simulated in a simple 2D world in Stage, the default *simple.world* file provides an extremely simple environment with just a 'T' shaped corridor with an odd extra cell. Also not only is the world simulated simple, but the robot simulated will work almost perfectly so the PGAIN values chosen will work well. In a sense the simulation is of an ideal scenario. This is not te case however in the real world.

The real world in which the real Pioneer robot will be tested has a floor which has areas which are not level and may cause the robot to turn slightly. Also there will be many different kinds of surfaces that the sonar sensors may or may not scan/interpret properly. Another key thing to remember is that the robot may not be in ideal shape, the robot may not be able to turn as accurately as it does in the simulations (highly likely) and the robot may not move as far or maybe too far compared to the simulations (also highly likely). This means that when it comes to moving the moving the robot from simulation testing to real world testing the PGAIN values may need adjusting to get more accurate results in the real world.

2 Possible Causes of Differences

There are many possible causes for differences in the results between simulated tests and real world tests. The causes can exist in both the robot and the environment it is tested in. Firstly the main thing to take into account is that the simulated world environment may not be and probably is not an accurate representation of the real world environment. To get a proper perspective of the differences in the results you must first make sure that you have the testing environments as similar as possible.

2.1 Possible Robot Related Causes

In terms of the robot in real world there are many ways in which the robot may not portray the same simulated version of itself and produce different results.

2.1.1 Wheels and Movement

Firstly the wheels may not be at even pressures causing the robot to always turn slightly due to uneven pressures between the sides. Also in terms of power, different results may be yielded in how long a robot can last on battery due to different tyre pressures. Relating to the wheels there is something that may yield different moving results. The surface that the wheel travels on as well as the surface of the wheel may give or not give enough traction to allow the robot to actually move. In the case where an algorithm to calculate

distance moved just uses time and the speed the robot should of been travelling at, the algorithm relies on the fact that the robot actually moves at the speed stated within the time so if the robot in slipping on ice for example then robot may not actually move at all but the robot thinks it has.

The robot may also count the number of revolutions a wheel goes round and the instrument that measures that may not provide accurate or reliable results thus resulting in different results. The wheel motors may give unequal power to both wheels as well producing the robot to turn when it is not meant to. As you can see there are many things just related to a robot's wheels (if it has any) that can cause unexpected results that differ from those on a simulation.

2.1.2 Sensors

Depending on what kinds of sensors your robot uses (if any) there are different ways in which results can be distorted. When using sonar sensors there is a problem called **reflection** where a sonar ping may hit a curved surface and ping off it in a different direction and not hitting the same sonar it was emitted from. This causes the sonar to think that there is no obstacle in front of it for the maximum distance of that sensor. Also there is another problem known as **interference** where a different sonar ping hits a sonar after it has just pinged thus making the sonar think that there is an obstacle in front of it. Maybe there is actually a obstacle there but it may be at a different distance, and as the sonar received a ping back already, the other ping (its actual ping) isn't received or processed thus giving the sonar a false reading. This is quite likely to happen when there are multiple sonar sensors on different robots in an area. Interference through reflection is also possible where one robot pings a curved surface and the angle causes the ping to bounce to another robot's sonar facing that direction. Also your own robots sonar sensors could be tricked by interference through reflection.

Another thing to take into account is that on a 2D simulation the sonar sensors do not take into account the height of an obstacle or the height at which the so sonar sensor sit. This can cause quite a lot of problem and different results as on a real robot, the sonar sensors map be sitting at a height at which they cannot detect an obstacle that is too low this leading the robot to crash into the obstacle that could not be detected. With the Pioneer robots, the sonar sensors are not able to detect most chair legs on swivel wheeled office chairs but can detect the stem of the chairs, also if a person put their foot in front of a sonar sensor, the foot probably won't get detected but the person's leg probably would.

2.2 Possible Real World Causes

The real world is full of uncertainties and the only way to ensure that that the robot in question is actually being tested properly and fully is to make sure that the simulated world is as complex and as uncertain as the real world is.

2.2.1 Landscape

A simulated 2D world would not perceive heights of objects or inclines that a robot may encounter. This can be a problem when you have an algorithm that tells a robot to move a set distance but hits an incline or is already on one. Your robot may have moved the set distance but not fully in the direction expected as the robot may of travelled uphill or downhill rather than along a flat surface. Also the terrain/surface the robot may be travelling along could cause the robot's wheels to slip thus either causing a turn or the robot thinking it has a moved a set distance when the reality is that it didn't. A simple simulated world would not consider terrain surfaces and wheel material and thus the traction between the two, but the real world would.

2.2.2 Weather Conditions

Another thing to take into consideration is how the robot would perform under certain weather conditions. In a simple simulated environment there might not be any form of *weather* for the robot to deal with. Different weather conditions could effect certain forms of sensor by distorting the readings. Also the robot may not travel well in certain weather conditions, or the internals of the robot may start to malfunction. Take for example a wheeled robot in a snow storm, the internals may start to freeze up in the cold causing all kinds of errors/problems to occur as well as the whole robot seizing to function.

2.3 Summary

In summary, there are many reasons/causes for the various kinds of differences between simulation testing and real world testing. Simulation testing is really good for simple tests. More realistic simulation tests can be programmed to help develop robots better for real world scenarios but this is a extremely complex task.