uSimMarine Final Project Report
Bad Compass Simulation

November 2019

Logan Zhang, youcheng@mit.edu

Visiting Student, MIT

---

1 Motivation
2 Overview
3 Configuration for Bad Compass Simulation
4 Bad Compass Simulation
5 How Bad Compass Simulation Works?

---

1 Motivation

When developing a naive DP system for our autonomous vehicle Heron, we discovered that the heading which is provided by the front seat device isn't accurate all the time. Though the MOOS application we developed for keeping vehicle's heading and position works in simulation, but it doesn't work at all during the on-water testing. Therefore, this gave us an idea of adding a new feature in uSimMarine, which can simulate the situation when the compass isn't giving the correct heading. This new feature enables user and developer to take a first glimpse at how their application will react under this situation before the on-water testing, allowing them to make some adjustments in advance. We'll called this new feature "Bad compass simulation".

2 Overview

After we added a new feature in uSimMarine, there are total 3 modes in uSimMarine so far. They are Dual mode, Drift mode and Heading error mode.

- Dual mode :

  When Dual mode is on, two vehicles appear on pMarineViewer. One is ideal vehicle, and the other one is ground truth vehicle. The ground truth vehicle will be affect by drift or heading error while the ideal one won't. Thus, the path of both vehicle will be will be different if drift mode or heading error mode is on.

- Drift mode :

  In Drift mode, vehicle will be drifting base on the artificial wind gust we provide. We can create wind gust by posting external force vectors to the MOOS variable DRIFT_VECTOR_ADD, which is read by the uSimMarine application to alter the prevailing external force vector, which by default has a magnitude of zero. Drift mode will be on once MOOS variable DRIFT_VECTOR_ADD is received by uSimMarine.

- Heading error mode :

  This is the one of the new feature we added in uSimMarine. In this mode, user and developer can set a heading error, this error will be added to vehicle's current heading, simulating the situation when the heading provided from the compass is never correct and always has a constant error.

Combine three modes mentioned above, we'll have 8 different kinds of simulation.

Case\Mode	Dual	Drift	Heading error
Case 1	off	off	off
Case 2	off	off	on
Case 3	off	on	off
Case 4	off	on	on
Case 5	on	off	off
Case 6	on	off	on
Case 7	on	on	off
Case 8	on	on	on

This table shows 8 different kinds of simulation in the new version of uSimMarine. User and developer can decide which case fits their requirements the most, and use it to help them develop their application.

Bad compass simulation can be achieved by turning Dual mode and Heading error mode on, which is case 6 in the table above. We'll describe more details of this two mode in the next two section.

More details about this simulation will be described later.

3 Configuration for Bad Compass Simulation

For Bad Compass simulation, we have to turn on both Dual mode and Heading error mode. We'll describe how to configure Dual mode and Heading error mode in this section.

1. Dual mode :

Dual Mode can be activated by setting dual_state parameter to `true` in uSimMarine configuration block.

Listing 3.1 - Example configuration of the uSimMarine application.

```
1  ============================================================
2  uSimMarine Example MOOS Configuration
3  ============================================================
4
5  ProcessConfig = uSimMarine
6  {
7    AppTick   = 4
8    CommsTick = 4
9
10   dual_state    = false // or {true}          <=== set dual_state to true
11  }
```

2. Heading error mode :

Heading error mode will be activated once MOOS variable DRIFT_VECTOR_ADD is received by uSimMarine.The uSimMarine application registers MOOS variable HEADING_ERROR_ADD, providing user and developer a way to simulate the situation when the front seat device isn't giving the correct heading

and always have a constant error during the entire mission. MOOS variable HEADING_ERROR_ADD will be added to the current heading, set by either the initial configuration or the last received mail concerning the error of the heading. Details are described as below:

(a) Configure .moos file, and add the line below to uSimMarine configuration block :

Listing 3.2 - Example configuration of the uSimMarine application.

```
 1  ============================================================
 2  uSimMarine Example MOOS Configuration
 3  ============================================================
 4
 5  ProcessConfig = uSimMarine
 6  {
 7    AppTick   = 4
 8    CommsTick = 4
 9
10    dual_state    = true
11    heading_error = 10            <=== set heading error
12   }
```

This will launch the vehicle with the 10 degrees error(clockwise), and the vehicle will always has a 10 degrees error heading(compare to ground truth heading, where vehicle is exactly pointing to) until the end of the mission.

(b) We can also dynamically change the heading of the vehicle by posting MOOS variable HEADING_ERROR_ADD :

```
event = var=HEADING_ERROR_ADD, val= 45, time=20
event = var=HEADING_ERROR_ADD, val=-45, time=30
```

The script above poke by uTimerScript will give the vehicle a 45 degrees of error clockwise after the mission is launch after 20 seconds, and 10 seconds later, the second message will give the vehicle a 45 degrees of error counter-clockwise, which means after 30 seconds, the net heading error will be zero.

4 Bad Compass Simulation

To explain Bad Compass Simulation more clearly, I'll take s3_charlie mission as an example. The charlie mission may be launched from the command line in the following manner:

```
$ cd moos-ivp/missions/s3_charlie/
$ ./launch.sh --warp=10
```

After Charlie mission is launched, selecting HEADING_ERROR=45 from the Action pull-down menu. You'll see vehicle "charlie_GT" suddenly get a 45 degrees heading error clockwise. Next, press DEPLOY button in the lower right corner, and see what happen.

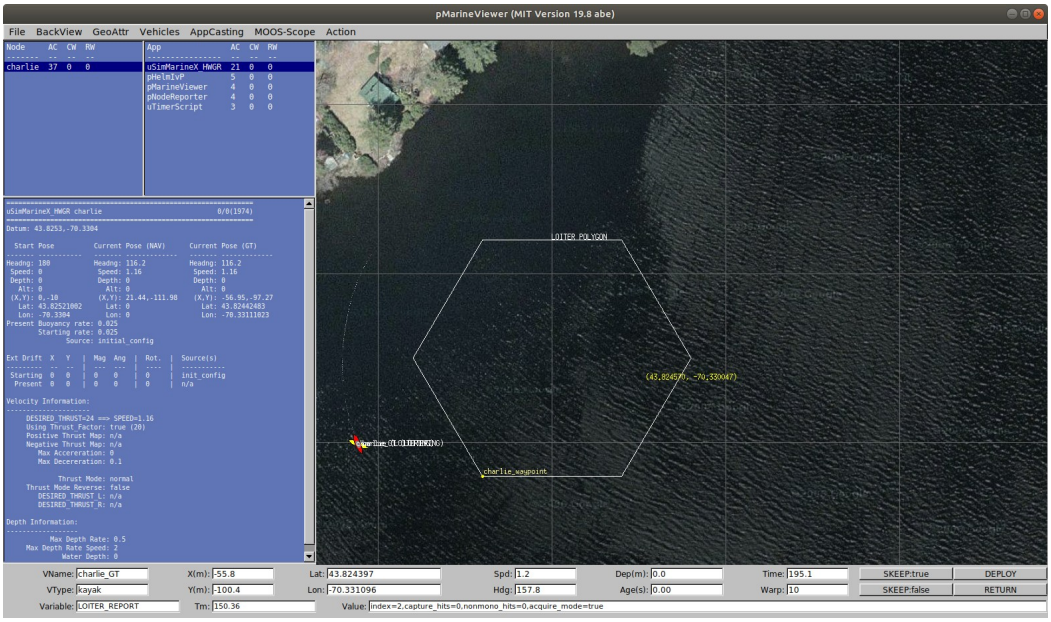

Figure 3.1: **The Charlie Mission:** The vehicle, "charlie_GT" is never point to the direction where pHelmIvp tells it to go because of the heading error.

After vehicle is deployed, “charlie” vehicle(yellow one) is always pointing to the next charlie waypoint but the ground truth vehicle “charlie_GT”(red one) is never pointing to the same direction and always have a 45 degrees difference. This simulate the time when you doesn’t calibrate your compass well or you have a bad compass causing your heading will always have a 45 degrees error clockwise. The yellow vehicle represents the direction where pHelmIvp tells the vehicle to go, and the red vehicle represents the direction where the vehicle is actually going. Therefore, the ground truth vehicle “charlie_GT” will never reach the waypoint. It can only get close to waypoint but never reach it.

This Bad Compass Simulation we added in Charlie mission can help user and developer understand how their application will react and adjust when the heading provided from the front seat device isn’t correct.

5 How Bad Compass Simulation Works?

To get a better understanding of how Bad Compass Simulation works, we can start from understanding how MOOS variables are passed and calculated between every MOOS application in this simulation. Take a look at the flow chart below.

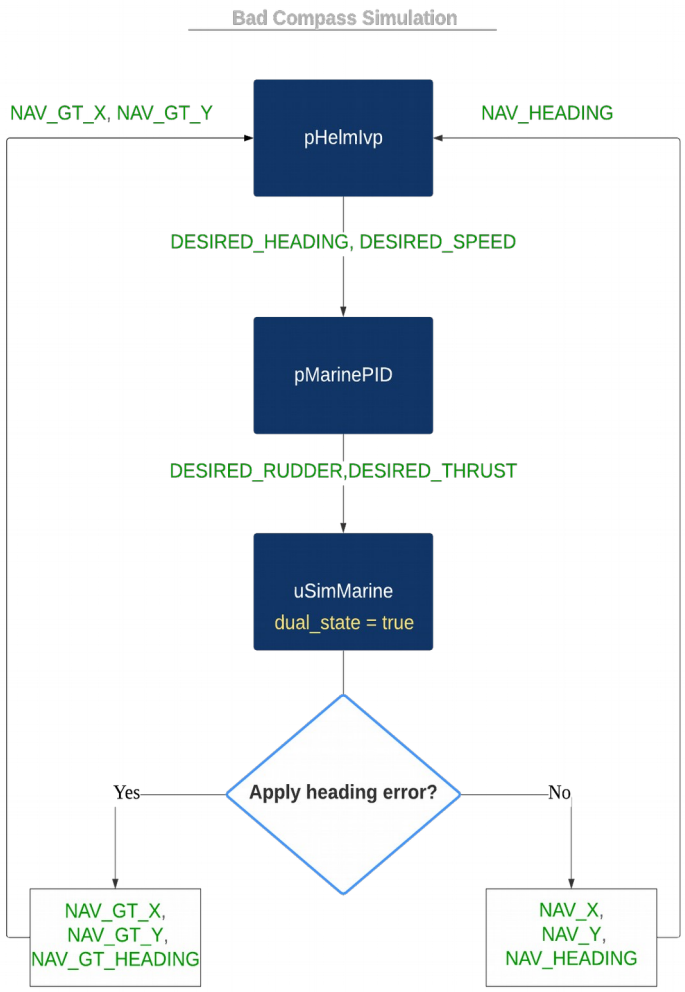

Figure 3.2: **The Charlie Mission Flow chart of Bad Compass Simulation:** In Bad Compass simulation, pHelmIvp make decisions base on ground truth X and Y, but wrong heading.

As you can see, when we set dual_state parameter to `true`, there will be two branches in uSimMarine, one is ground truth result which take external force and heading error into calculation, the other one is ideal result which won't be affected by both external force and heading error. In this simulation, pHelmIvp will get all the information it needs from ground truth result(NAV_GT_X and NAV_GT_Y) except for heading. Only heading will be provided from the ideal branch(NAV_HEADING). This means pHelmIvp is making decision base on the correct position(NAV_GT_X and NAV_GT_Y) but wrong heading(NAV_HEADING), causing the result that the vehicle will never reach the destination, and that's what we see in the Bad Compass Simulation.