PoseKeeping Final Project Report
pPoseKeeping Version 1

November 2019

Logan Zhang, youcheng@mit.edu

Visiting Student, MIT

1 Motivation

For multi-vehicle mission at the pavilion, we have to deploy every single vehicle with remote controller before switching to autonomous mode, and normally we'll have one person to take care of one vehicle at a time. Therefore, if we have a huge mission which includes six vehicles, then we'll need at least six people to help with this on-water testing. However, we can't always find that many people to help with the experiment. Thus, it will be great if we have an application which will take care all of this, including taking the vehicle to the destination we want and hold it's position and heading before we start the mission. This would be a great capability to have for multi-vehicle testing.

2 Overview

The goal of this project is to create a "naive" DP system for this particular boat(Heron) and pair of thrusters. We called this application pPoseKeeping. pPoseKeeping allows vehicle,

(1) More-or-less pointing in a chosen direction, e.g., away from the dock, or within a given range of heading values.

(2) More-or-less in one spot, within a certain radius, tolerance.

(3) Holding the above with only minor adjustments, e.g., not invoking full turns or loops to get back in position.

(4) Speaking directly to the Heron front seat in terms of Thrust Left/Right by the IM200 interface, making this application works completely without pHelmIvp.

(5) Can freely switch between pHelmIvp and pPoseKeeping.

With pPoseKeeping application, we can deploy N herons out to station-keeping points just off the dock, putting them into a pose-keeping mode (Helm not active), deploying the mission/help which would halt the pose-keeping mode.

3 pPoseKeeping Version1

The pPoseKeeping application is organized by hierarchical mode declarations into three modes: Forward mode, Backward mode and Keeping Heading mode.

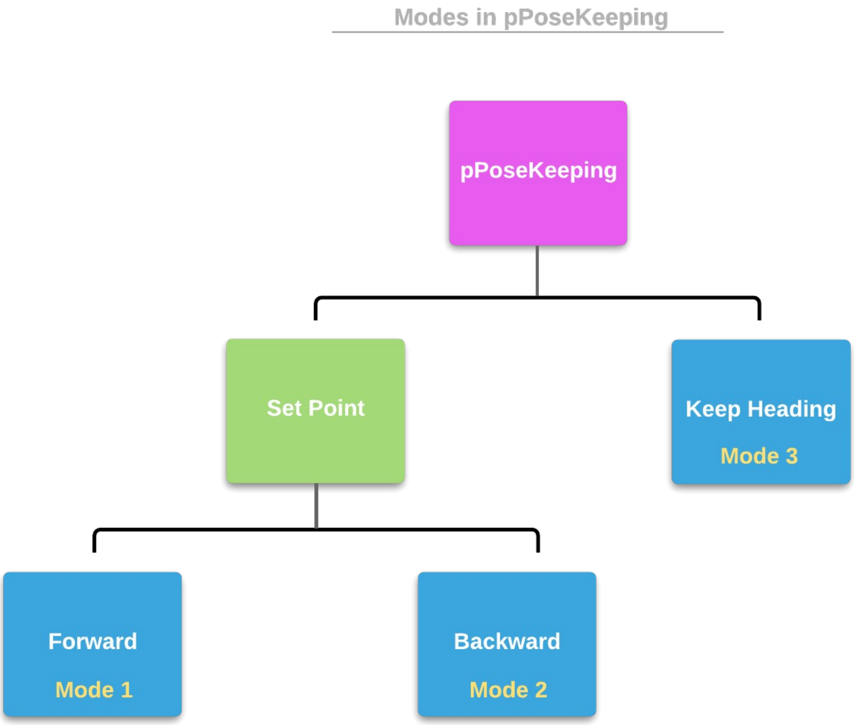


Figure 3.1: **pPoseKeeping Mode Declarations:** When pPoseKeeping application is activated, the vehicle will be in one of the three modes , Forward mode, Backward mode or Keep Heading mode.

When pPoseKeeping application is activated, the vehicle will be in one of the three modes indicated by the leaf nodes of the tree. Details of each mode are described as below.

1. Set Point Mode :

In Set Point Mode, vehicle will be heading to the pose-keeping region that we set in pPoseKeeping configuration block, and under Set Point Mode, there are Forward Mode and Backward Mode.

(a) Forward Mode :

Vehicle will be in Forward Mode when it is far away from our pose-keeping region. This mode is used to take our vehicle to the destination we set.

(b) Backward Mode :

Backward Mode is only used when the vehicle is really close to the pose-keeping region and the region is right behind the vehicle. When the vehicle drifts out of the pose keeping region and the region is behind it, instead of making a full turn back to the region, it will be more efficient if we just use backward thrust to get back in the region.

2. Keep Heading mode :

Once the vehicle is in the pose-keeping region, it will switch to Keep Heading mode. In this mode, vehicle will keep the heading we set in pPoseKeeping config block as much as possible using PID control.

4 Configuration Parameters for pPoseKeeping

The following parameters are defined for pPoseKeeping. A more detailed description is provided in other parts of this section. Parameters having default values are indicated so in parentheses below.

Listing 4.1 - Configuration Parameters for pPoseKeeping.

position:	Decide the center of the pose-keeping region.(50,-50)
heading:	Decide the desired heading in keep heading mode.(180)
kp:	PID parameters proportional gain. The default is 1.5.
ki:	PID parameters integral gain. The default is 0.
kd:	PID parameters derivative gain. The default is 0.
tolerance_radius:	Decide the range of pose-keeping region.(5)

An Example MOOS Configuration Block

An example MOOS configuration block is provided in Listing 2.2 below. This can also be obtained from a terminal window with:

```
$ pPoseKeeping --example or -e
```

Listing 4.2 - Example configuration of the pPoseKeeping application.

```
===============================================================
pPoseKeeping Example MOOS Configuration
===============================================================

ProcessConfig = pPoseKeeping
{
  AppTick   = 4
  CommsTick = 4

  position       = 0
  heading        = 0
  Kp             = 0
  Kd             = 0
  Ki             = 0
}
```

5 Publications and Subscriptions for pPoseKeeping

The interface for pPoseKeeping, in terms of publications and subscriptions, is described below. This same information may also be obtained from the terminal with:

```
  $ pPoseKeeping --interface or -i
```

5.1 Variables Published by pPoseKeeping

The pPoseKeeping application will publish for the following MOOS variables:

- DESIRED_THRUST_L: Left thrust output, [-100,100].

- DESIRED_THRUST_R: Right thrust output, [-100,100].

5.2 Variables Subscribed for by pPoseKeeping

The pPoseKeeping application will subscribe for the following MOOS variables:

- NAV_HEADING_CPNVG: Compass heading. Direction of travel in degrees clockwise from true north, [0,359].
- NAV_HEADING: GPS heading. Direction of travel in degrees clockwise from true north, [0,359].

- NAV_X: Vehicle’s current position X, [-inf/+inf].

- NAV_Y: Vehicle’s current position Y, [-inf/+inf].

- THRUST_MODE_DIFFERENTIAL: pPoseKeeping on/off switch, either `true` or `false`.

6 An Example Mission for pPoseKeeping

The pPoseKeeping mission may be launched from the command line in the following manner:

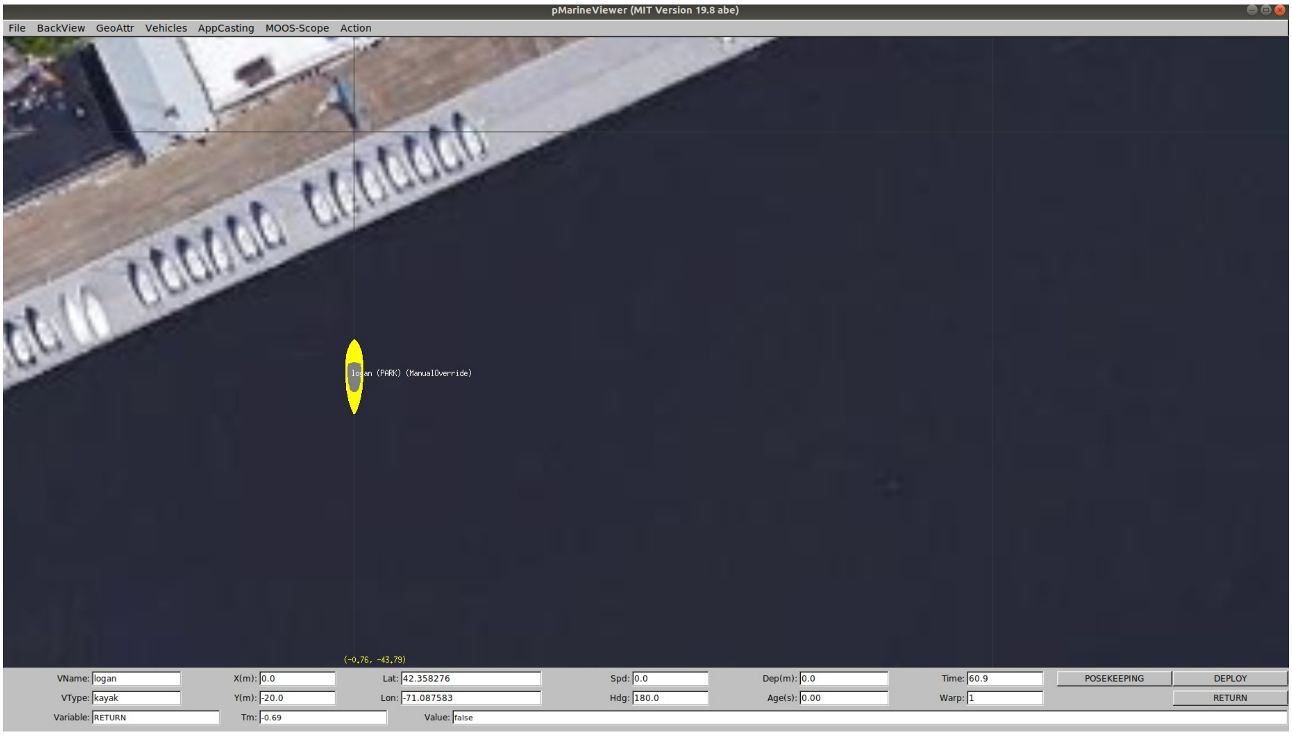
```
  $ cd moos-ivp-logan/missions/PoseKeeping/simulation_mission
  $ ./launch.sh 10
```

This should bring up a pMarineViewer window like that shown in Figure 6.1, with a single vehicle, *logan*, initially in the PARK mode.

(1) After hitting the POSEKEEPING button in the lower right corner, the vehicle enters the "Forward" mode and begins to proceed to the pose-keeping region which is shown by the red polygon.(Figure 6.2) The green arrow represent compass heading(NAV_HEADING_CPNVG). pPoseKeeping application only used compass heading to decide output thrust. GPS heading(NAV_HEADING) is never used in pPoseKeeping application. More details about heading will be described in the next section.

(2) A soon as logan reach the destination, its will switch to "Keep Heading" mode, and the pose-keeping region will turn blue.(Figure 6.3) Logan vehicle will keep trying to hold its heading, 180 degrees clock-wise to the true north by default until it is out of the pose-keeping region.

(3) You can keeping switching between “PoseKeeping” and “Helm” by pressing POSEKEEPING button, DEPLOY button, and RETURN button randomly.

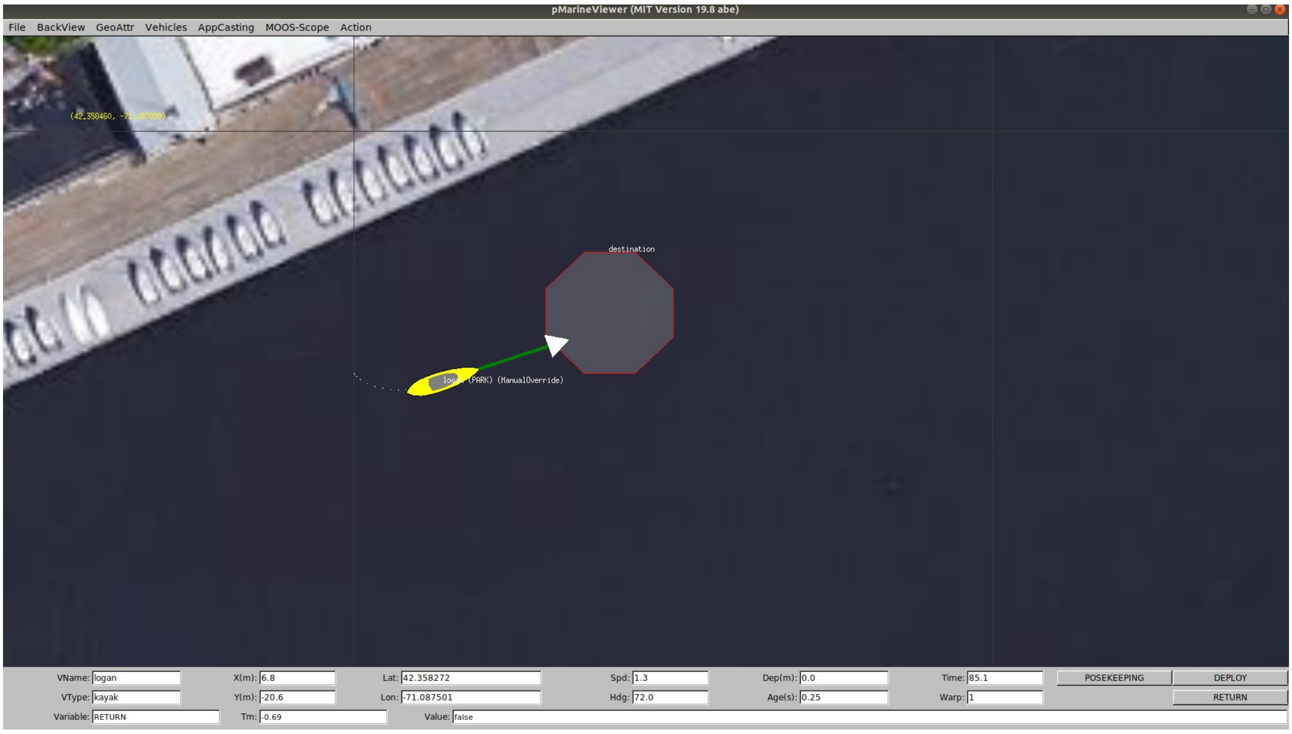


VName: logan X(m): 0.0 Lat: 42.358276 Spd: 0.0 Dep(m): 0.0 Time: 60.9

VType: kayak Y(m): -20.0 Lon: -71.087583 Hdg: 180.0 Age(s): 0.00 Warp: 1

Variable: RETURN Tm: -0.69 Value: false

Figure 6.1: **The PoseKeeping Mission(1):** After launching the mission, a single vehicle logan will show up and it's initially in the PARK mode.

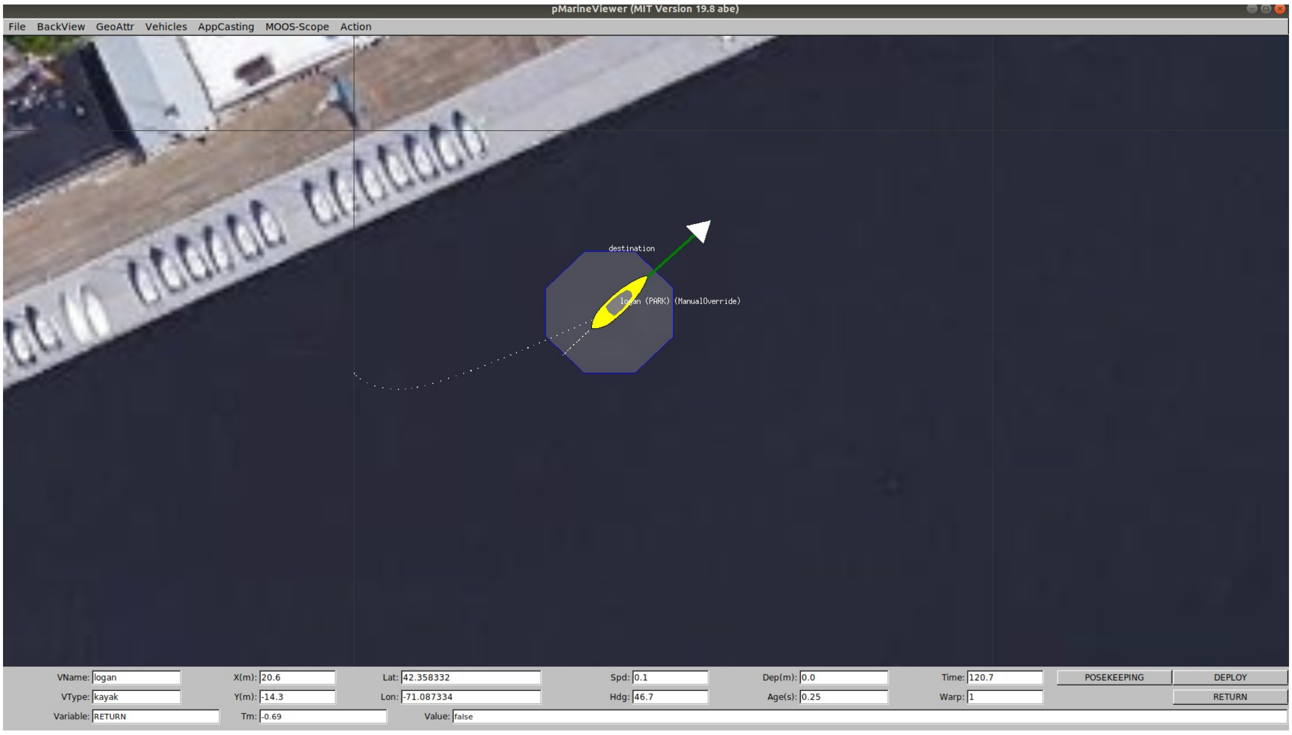


VName: logan X(m): 6.8 Lat: 42.358272 Spd: 1.3 Dep(m): 0.0 Time: 85.1

VType: kayak Y(m): -20.6 Lon: -71.087501 Hdg: 72.0 Age(s): 0.25 Warp: 1

Variable: RETURN Tm: -0.69 Value: false

Figure 6.2: **The PoseKeeping Mission(2):** The vehicle, logan is under "Forward" mode and it is heading to the destination point. The pose-keeping region will be red when under "Set Point" mode, which includes "Forward" mode and "Backward" mode.



VName: logan X(m): 20.6 Lat: 42.358332 Spd: 0.1 Dep(m): 0.0 Time: 120.7
VType: kayak Y(m): -14.3 Lon: -71.087334 Hdg: 46.7 Age(s): 0.25 Warp: 1
Variable: RETURN Tm: -0.69 Value: false

Figure 6.3: **The PoseKeeping Mission(3):** The vehicle, "logan" will be in “Keep Heading” mode once it reach its destination. In this mode, logan will turn its heading to the heading we set in the pPoseKeeping configuration block. Before it drift out of the pose-keeping region, it will hold its heading as much as possible.

7 Adjustments after on-water Testing

After the on-water testing, we made some adjustments in pPoseKeeping, these adjustments are made only for Heron.

(1) Subscribe compass heading(NAV_HEADING_CPNVG) instead of GPS heading(NAV_HEADING):

> In iM200, it notify GPS heading as NAV_HEADING, compass heading as NAV_HEADING_CPNVG. Therefore , in pHelmIvp, we’re using GPS heading all the time. However, there are two main reasons why we can’t use GPS heading(NAV_HEADING) in pPoseKeeping. First, GPS heading is incredibly inaccurate under low speed, and in pPoseKeeping, vehicle we remain low speed most of the time. Second, the heading will be completely wrong under “Backward” mode, since GPS heading is calculating heading by vehicle’s displacement. Hence, these two major drawbacks of GPS heading makes it not suitable for pPoseKeeping.
>
> To get compass heading from front seat, remember to restart ROS service everytime when you activate Heron.

```
   sudo service kingfisher-core restart
```

(2) Some values of output thrust won’t be accept by Heron:

> Under direct-thrust mode, if we publish thrust range from -33 to +17, heron's thruster won't move at all. Instead, the lights on heron will flash quickly. We consider it as a sign of warning, so we change the output thrust to zero if the it is in the range of -33 to +17.

(3) Forward thrust is much stronger than backward thrust:

In KeepHeading mode, the method we use to turn vehicle's heading without moving its position is simply by giving left and right thruster the same value but one positive and one negative. For example, "DESIRED_THRUST_L= 10, DESIRED_THRUST_R = -10". This works perfectly in simulation. But when we test it on river, vehicle can't hold its position since forward thrust(10) is much stronger than backward thrust(-10) even they have the same value. We tried several values to find the balance between forward and backward thrust, and we found that the ratio of forward thrust and backward thrust is approximately 1 to 3.

(4) Compass heading isn’t stable all the time:

Though compass heading is more accurate compare to GPS heading in this case, but sometimes it will still have a suddenly huge change in heading. To solve this, we add an filter in pPoseKeeping which will remove the heading if the difference between current heading and previous heading is larger than 120 degrees.

8 Future Work

The future work will be:

(1) Finding a better ratio between forward and backward thrust. Under Keep Heading Mode, if the force of forward and backward thrust are not even, this will cause the vehicle to drift. Therefore, the vehicle won’t be able to hold its heading and remain its position at the same time. The ratio we’re using is constant now. Maybe setting up a thrust map for forward and backward thrust is a good way to solve this problem.

(2) Figure out the reason why some of the output thrust won’t be accepted by Heron, since this will affect the output thrust we can use for PID control.