

Lab 3 - Introduction to the Helm

Accelerated Marine Vehicle Autonomy,
Sensing, and Communications

May, 2017
國立台灣大學
National Taiwan University - Taiwan

Michael Benjamin, mikerb@mit.edu
Henrik Schmidt, henrik@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

1	Overview and Objectives	3
2	Exercises	4
2.1	Exercise 1 - Add a New Behavior to the Alpha Mission	4
2.2	Exercise 2 - Add Command and Control Switches to pMarineViewer	6
2.3	Exercise 3 - Use uTimerScript to Control Mission State	7
2.4	Exercise 4 - Integrate the iSay App to Behavior Events	8
2.5	Exercise 5 - Station Keeping at Waypoints	9
2.6	Exercise 6 - Station Keeping at Waypoints, for Fixed Durations	10
3	Where to Find the Solutions	11

1 Overview and Objectives

This lab will introduce the IvP Helm to new users. It assumes nothing regarding prior experience with the Helm, other than the material included in the accompanying lecture "Introduction to the IvP Helm". The goals of this lab are to (a) understand the role of mission files (.moos) and behavior files (.bhv) in constructing new missions, (b) get comfortable with different ways to generate mission changing events, with endflags, pMarineViewer and the uTimerScript, (c) explore the iSay MOOS App, (d) explore the StationKeeping behavior.

- Mission and Behavior Files
- Adding new behaviors to a mission
- Behavior endflags
- Waypoint behavior wptflags
- StationKeep behavior
- The iSay MOOS App
- The uTimerScript MOOS App

Preliminaries

This lab assumes you have a working MOOS-IvP tree checked out and built on your computer. To verify this make sure that the following executables are built and findable in your shell path:

```
$ which MOOSDB
/Users/you/moos-ivp/bin/MOOSDB
$ which pHelmIvP
/Users/you/moos-ivp/bin/pHelmIvP
```

If unsuccessful with the above, return to the steps in the first lab and proceed until you are able to run the s1_alpha mission.

http://oceanai.mit.edu/ivpman/labs/machine_setup

IMPORTANT TIP:

All exercises begin by copying an existing mission folder (containing an alpha.moos and alpha.bhv file. We create the new folder and new mission.

Create your own missions folder, missions_lab, and put all your work in this folder.

A new copy of a folder can be created with:

```
$ cp -rp moos-ivp/ivp/missions/s1_alpha missions_lab/s2_alpha
```

`./launch.sh 25` ->可以調整wrap，數字越大跑越快，但太快會脫離軌跡

2 Exercises

2.1 Exercise 1 - Add a New Behavior to the Alpha Mission

In this exercise the goal is make a version of the Alpha mission that uses another waypoint behavior to perform a second survey before returning. It should look something like:

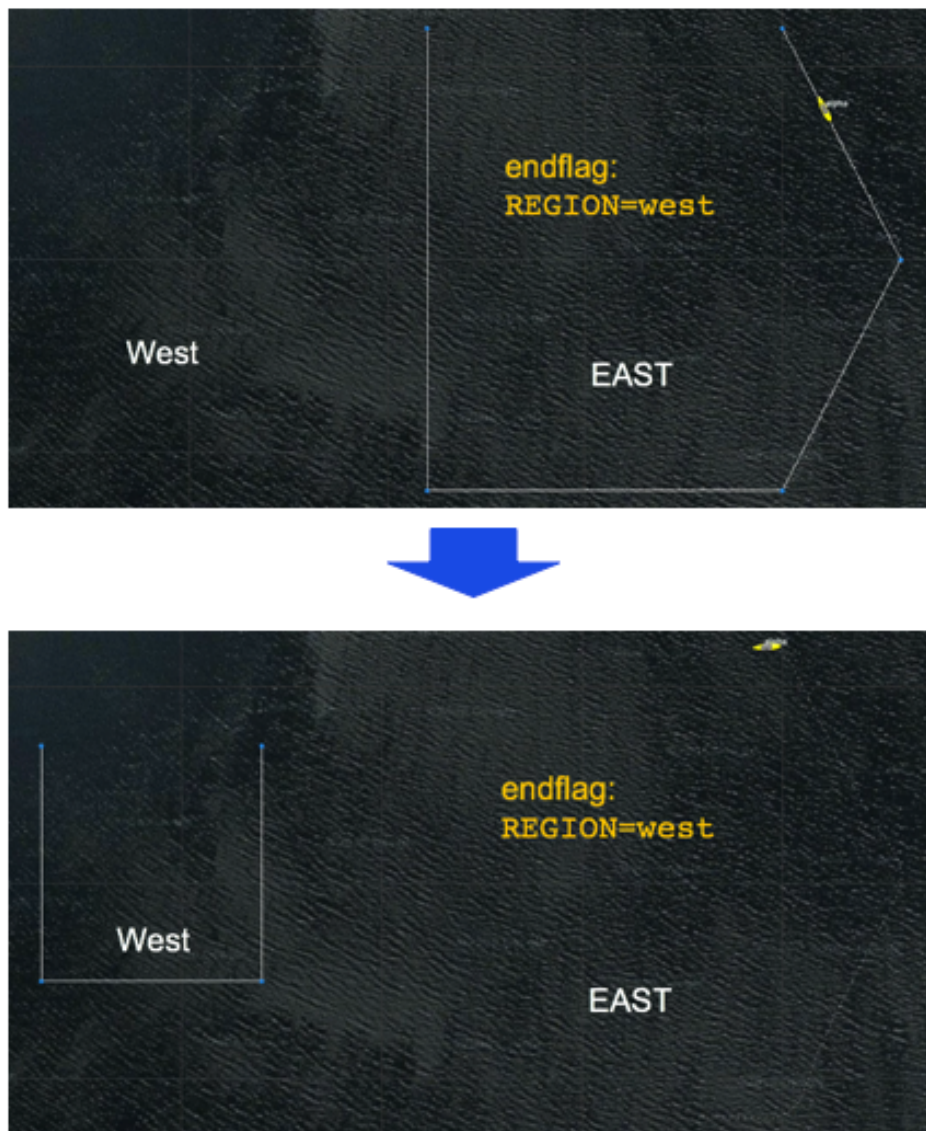


Figure 1: **s8_alpha**: Using the Loiter Behavior

You will need to do the following steps:

- Copy the `s1_alpha` mission, naming it `s2_alpha`
- Add second Waypoint survey behavior

- Make the end flags for the first behavior, trigger the second survey behavior.
- Make the end flags for the second behavior trigger the return behavior

For the second survey behavior, give it the name "west_wpt_survey". And give it the survey points below. Give a run condition requiring the survey region mode (given by the REGION MOOS variable) is "west". Finally give it an end flag that triggers the return home behavior. The relevant parameters are:

```
//-----
Behavior = BHV_Waypoint
{
    . . .
    name = west_wpt_survey
    condition = REGION = west
    endflag   = RETURN = true
    points = -35,-65 : -35,-125 : 20,-125 : 20, -65
    . . .
}
```

在.bhv裡面

For the first survey behavior, give it the name "east_wpt_survey". Give a run condition requiring the survey region is "east". Finally give it an end flag that triggers the second survey behavior. The relevant parameters are:

```
//-----
Behavior = BHV_Waypoint
{
    . . .
    name = east_wpt_survey
    condition = REGION = east
    endflag   = REGION = west
    points = -35,-65 : -35,-125 : 20,-125 : 20, -65
    . . .
}
```

Lastly, you need to initialize the survey region to be east initially. At the top of the behavior file:

```
//-----      FILE: alpha.bhv      -----
initialize    DEPLOY = false
initialize    RETURN = false
initialize    REGION = east
```

2.2 Exercise 2 - Add Command and Control Switches to pMarineViewer

In this exercise the goal is make a version of the Alpha mission that requires the user to command the vehicle between the east and west regions. Otherwise the vehicle will remain surveying in its current region indefinitely.

You will need to do the following steps:

- Copy the s2_alpha mission, naming it s3_alpha
- Reconfigure each survey behavior to repeat forever
- Configure pMarineviewer to have two new buttons for commanding the vehicle between the east and west regions

To change the survey behaviors to repeat forever, change the repeat paramter to be "forever"

```
//-----  
Behavior = BHV_Waypoint  
{  
    . . .  
    repeat = forever  
    . . .  
}  
ktm
```

The pMarineViewer app can be configured to add additional buttons besides the DEPLOY and RETURN buttons in the Alpha mission. The documentation for pMarineViewer can be found here:
<http://oceanai.mit.edu/ivpman/apps/pMarineViewer>

The pMarineViewer buttons are added as below.

```
//-----  
ProcessConfig = pMarineViewer  
{  
    . . .  
    button_three = WEST # REGION=west  
    button_four  = EAST # REGION=east  
    . . .  
}
```

2.3 Exercise 3 - Use uTimerScript to Control Mission State

In this exercise the goal is to further modify the previous Alpha mission where the vehicle changes between east and west based on the output of a script, running in the **uTimerScript** MOOS Application.

You will need to do the following steps:

- Copy the `s3_alpha` mission, naming it `s4_alpha`
- Add **uTimerScript** to the list of apps launched in the mission, inside the Antler configuration block of `alpha.moos`.
- Configure **uTimerScript** with a simple script to switch between east and west mission every 90 seconds.

The **uTimerScript** documentation can be found here:

<http://oceanai.mit.edu/ivpman/apps/uTimerScript>

The timer script for this exercise is below: You can copy it into your mission file, but try to understand the configuration parameters by reading the documentation.

```
//-----  
// uTimerScript config block  
ProcessConfig = uTimerScript  
{  
  AppTick      = 4  
  CommsTick    = 4  
  
  condition    = DEPLOY = true  
  event        = var=REGION, val="west", time=90  
  event        = var=REGION, val="east", time=180  
  reset_max    = nolimit  
  reset_time   = all-posted  
}
```

直接在moos檔裡面改wrap值也是可在最上面

開啟儀表板裡面 timer的時間 0.9s 1.8s的時候

2.4 Exercise 4 - Integrate the iSay App to Behavior Events

In this exercise the goal is to further modify the previous Alpha mission to generate audio events when the vehicle completes surveying in either region. Audio events will be generated with the **iSay** MOOS App.

You will need to do the following steps:

讓他說話～

- Copy the s2_alpha mission, naming it s5_alpha
- Add **iSay** to the list of apps launched in the mission, inside the Antler configuration block of alpha.moos.
- Configure **iSay** with its default configuration block and add it to alpha.moos.
- Download to audio files shipbell.wav, and buzzer.wav from the MIT server.
- Configure the east survey mission to sound the shipbell sound upon completion.
- Configure the west survey mission to sound the buzzer sound upon completion.

The **iSay** documentation can be found here:

<http://oceanai.mit.edu/ivpman/apps/iSay>

say hello就會講話耶～有夠潮或是用espeak

You will need to add an iSay event as an endflag for each behavior. For example:

```
//-----  
Behavior = BHV_Waypoint  
{  
  name      = east_waypt_survey  
  pwt       = 100  
  condition = RETURN = false  
  condition = DEPLOY = true  
  condition = REGION = east  
  endflag   = REGION = west  
  endflag   = SAY_MOOS = file=shipbell.wav  
  
  . . .  
}
```

And:

```
//-----  
Behavior = BHV_Waypoint  
{  
  name      = west_waypt_survey  
  pwt       = 100  
  condition = RETURN = false  
  condition = DEPLOY = true  
  condition = REGION = east  
  endflag   = REGION = west  
  endflag   = SAY_MOOS = file=buzzer.wav  
  
  . . .  
}
```


2.5 Exercise 5 - Station Keeping at Waypoints

In this exercise the goal is to modify the original Alpha mission to station-keep each time it arrives at one of its waypoints. The user then re-clicks the DEPLOY button in `pMarineViewer` to send the vehicle in motion again.

You will need to do the following steps:

- Copy the `s1_alpha` mission, naming it `s6_alpha`
- Add a `StationKeep` behavior to the behavior file.
- Add a `wptflag` configuration to the Waypoint behavior, such that it *turns off* the Waypoint behavior, and turns *on* the StationKeeping behavior.
- Modify the `pMarineViewer` DEPLOY button such that it also turns off StationKeeping when it is clicked.

2.6 Exercise 6 - Station Keeping at Waypoints, for Fixed Durations

In this exercise the goal is to modify the previous Alpha mission, which will station-keep each time it arrives at one of its waypoints, to automatically resume its deployment after a short period of time. No user involvement required.

You will need to do the following steps:

- Copy the `s6_alpha` mission, naming it `sy_alpha`
- Modify the `StationKeep` behavior to have a duration.
- Add an endflag to the `StationKeep` behavior such that it shuts itself off after the duration expires.
- Declare the `StationKeep` behavior to be perpetual - so that it can be used again after it completes.

3 Where to Find the Solutions

All solutions can be downloaded from the server:

Exercise 1:

- http://oceanai.mit.edu/ntu/s2_alpha/alpha.moos
- http://oceanai.mit.edu/ntu/s2_alpha/alpha.bhv

Exercise 2:

- http://oceanai.mit.edu/ntu/s3_alpha/alpha.moos
- http://oceanai.mit.edu/ntu/s3_alpha/alpha.bhv

Exercise 3:

- http://oceanai.mit.edu/ntu/s4_alpha/alpha.moos
- http://oceanai.mit.edu/ntu/s4_alpha/alpha.bhv

Exercise 4:

- http://oceanai.mit.edu/ntu/s5_alpha/alpha.moos
- http://oceanai.mit.edu/ntu/s5_alpha/alpha.bhv

Exercise 5:

- http://oceanai.mit.edu/ntu/s6_alpha/alpha.moos
- http://oceanai.mit.edu/ntu/s6_alpha/alpha.bhv

Exercise 6:

- http://oceanai.mit.edu/ntu/s7_alpha/alpha.moos
- http://oceanai.mit.edu/ntu/s7_alpha/alpha.bhv