uFldPathCheck: Monitoring Vehicle Path Properties

June 2018

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

1 Overview

The `uFldPathCheck` application is a tool for summarizing a few properties in a field of vehicles remotely deployed. The primary focus is on summarizing the speed and distance travelled for each vehicle. Rather than relying on the vehicles themselves to calculate and report this information, the `uFldPathCheck` tool determines this information independently based on node reports from the vehicles. The assumption is that `uFldPathCheck` is running on a topside or shoreside computer, and receiving information about deployed vehicles primarily through node reports. The typical layout is shown in Figure 1

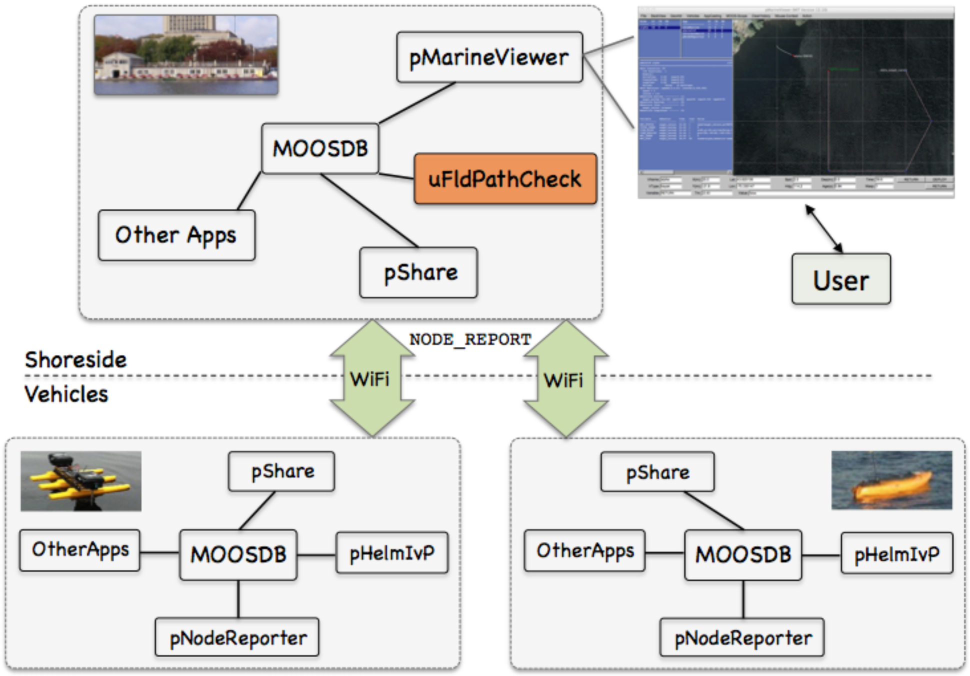

Figure 1: **Typical uFldPathCheck Topology:** A shoreside or topside community is receiving information from several deployed vehicles, in the form of node reports. The node reports contain time-stamped updated vehicle positions, from which the speed and distance measurements are derived and posted to the shoreside MOOSDB.

In short, `uFldPathCheck` subscribes for incoming node reports for any number of vehicles, and keeps a running history of positions for each vehicle. It uses this recent-history to post (a) the current speed noted per vehicle and (b) the distance travelled per vehicle. These two reports are posted in the `UPC_SPEED_REPORT` and `UPC_ODOMETRY_REPORT` variables to the MOOSDB. The odemetry report includes a total distance travelled, and a "trip-ometer" distance travelled since the last trip reset. The trip-ometer may be reset for a vehicle *alpha* when mail is received of the form `UPC_TRIP_RESET=alpha`. Examples of the posted output form are given below in Section 3.1.

2 Overview of the uFldPathCheck Interface and Configuration Options

The `uFldPathCheck` application may be configured with a configuration block within a `.moos` file. Its interface is defined by its publications and subscriptions for MOOS variables consumed and generated by other MOOS applications. An overview of the set of configuration options and interface is provided in this section.

2.1 Configuration Parameters of uFldPathCheck

The following parameters are defined for `uFldPathCheck`.

Listing 2.1: Configuration parameters for `uFldPathCheck`.

history: Length of queue used for calculating present speed. The default is 10.

3 Publications and Subscriptions for uFldPathCheck

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

```
$ uFldPathCheck --interface or -i
```

3.1 Variables Published by uFldPathCheck

The primary output of uFldPathCheck to the MOOSDB are the speed and odometry reports.

- APPCAST: Contains an appcast report identical to the terminal output. Appcasts are posted only after an appcast request is received from an appcast viewing utility.
- UPC_ODOMETRY_REPORT: The current odometry information for a given vehicle.
- UPC_SPEED_REPORT: The current speed for a given vehicle.

An example of the odemetry report:

```
UPC_ODOMETRY_REPORT  = "vname=alpha,total_dist=4205.4,trip_dist=1105.2"
```

An example of the odemetry report:

```
UPC_SPEED_REPORT  = "vname=alpha,avg_speed=2.21"
```

3.2 Variables Subscribed for by uFldPathCheck

The uFldPathCheck application subscribes to the following MOOS variables:

- APPCAST_REQ: A request to generate and post a new apppcast report, with reporting criteria, and expiration.
- NODE_REPORT: A node report for a given vehicle from pNodeReporter.
- NODE_REPORT_LOCAL: A node report for a given vehicle from pNodeReporter.
- UPC_TRIP_RESET: The name of a vehicle to have its trip odemeter reset.

3.3 An Example MOOS Configuration Block

An example MOOS configuration block can be obtained by entering the following from the command-line:

```
$ uFldPathCheck --example or -e
```

Listing 3.2: Example configuration for uFldPathCheck.

```
===============================================================
uFldPathCheck Example MOOS Configuration
```

```
=============================================================

ProcessConfig = uFldPathCheck
{
  AppTick   = 4
  CommsTick = 4

  history   = 10  (Default)
}
```

4 Usage Scenarios the uFldPathCheck Utility

The motivation for this tool is to have a module running on the shoreside capable of being used in a competition scenario. Especially in a simulated competition, there may be a need to limit the upper speed of a participating vehicle to ensure a level playing field between competitors. Since the vehicle simulators may be running on separate machines with particpants merely reporting node reports, there is no way to directly control the upper speed of the vehicles. By *monitoring* the upper speed, this leaves open the option of either (a) disqualifying a vehicle caught moving too fast, or (b) imposing a penalty on a speeding vehicle. The penalty chosen is outside the scope of this module but may include reducing the number of points awarded for certain accomlishments, adding more noise to simulated sensors, and so on. Since this usage scenario implies a "non-compliant" vehicle, we don't want to base the monitored speed on a value reported by the vehicle. Instead we calculate the speed based on successive time-stamped node reports with positions.

Likewise, the odometry information calculated and posted is also intended for use in a competition context. Conceivably, part of a competition may require a vehicle to periodically "re-fuel" after travelling a certain distance. By having the odemetry information calculated independently on the shoreside, constraints on total distance, or fuel calculations may be generated to impose on vehicle competitions. The `uFldPathCheck` application accepts the `UPC_ODOMETRY_RESET=VNAME` posting to reset the trip-ometer for a given vehicle, presumably after a re-fueling event is noted. The odometry information may also be used directly in competitions where minimizing the total path length is the primary objective.