

The uField Toolbox Overview

June 2018

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

1	Overview	1
2	Module Overview	1
3	The uField Toolbox Design	2
4	Motivations for the uField Toolbox	3

1 Overview

The uField Toolbox contains a number of tools for supporting multi-vehicle missions where each vehicle is connected to a shoreside community. This includes both simulation and real field experiments. It also contains a number of simulated sensors that run on off-board the vehicle on the shoreside.

2 Module Overview

- **pHostInfo**: Automatically detect the vehicle's host information including the IP addresses, port being used by the MOOSDB, the port being used by local **pShare** for UDP listening, and the community name for the local MOOSDB. Post these to facilitate automatic inter-vehicle communications in especially in multi-vehicle scenarios where the local IP address changes with DHCP.
- **uFldNodeBroker**: Typically run on a vehicle or simulated vehicle in a multi-vehicle context. Used for making a connection to a shoreside community by sending local information about the vehicle such as the IP address, community name, and port number being used by **pShare** for incoming UDP messages. Presumably the shoreside community uses this to know where to send outgoing UDP messages to the vehicle.
- **uFldShoreBroker**: Typically run in a shoreside community. Takes reports from remote vehicles describing how they may be reached. Posts registration requests to shoreside **pShare** to bridge user-provided list of variables out to vehicles. Upon learning of vehicle JAKE will create bridges **FOO_ALL** and **FOO_JAKE** to JAKE, for all such user-configured variables.
- **uFldNodeComms**: A shoreside tool for managing communications between vehicles. It has knowledge of all vehicle positions based on incoming node reports. Communications may be limited based on vehicle range, frequency of messages, or size of message. Messages may also be blocked based on a team affiliation.
- **uFldMessageHandler**: A tool for handling incoming messages from other nodes. The message is a string that contains the source and destination of the message as well as the MOOS variable

and value. This app simply posts to the local MOOSDB the variable-value pair contents of the message.

- **uFldScope**: Typically run in a shoreside community. Takes information from user-configured set of incoming reports and parses out key information into a concise table format. Reports may be any report in the form of comma-separated parameter-value pairs.
- **uFldPathCheck**: Typically run in a shoreside community. Takes node reports from remote vehicles and calculates the current vehicle speed and total distance traveled and posts them in two concise reports. Odometry tallies may be re-set to zero by other apps.
- **uFldHazardSensor**: Typically run in a shoreside community. Configured with a set objects with a given x,y location and classification (hazard or benign). The sensor simulator receives a series of requests from a remote vehicle. When sensor determines that an object is within the sensor field of a requesting vehicle, it may or may not return a sensor detection report for the object, and perhaps also a proper classification. The odds of receiving a detection and proper classification depend on the sensor configuration and the user's preference for P_D/P_FA on the prevailing ROC curve.
- **uFldHazardMetric**: An application for grading incoming hazard reports, presumably generated by users of the **uFldHazardSensor** after exploring a simulated hazard field.
- **uFldHazardMgr**: A straw-man MOOS app for managing hazard sensor information and generation of a hazard report over the course of an autonomous search mission.
- **uFldBeaconRangeSensor**: Typically run in a shoreside community. Configured with one or more beacons with known beacon locations. Takes range requests from a remote vehicle and returns a range report indicating that vehicle's range to nearby beacons. Range requests may or may not be answered depending on range to beacon. Reports may have noise added and may or may not include beacon ID.
- **uFldContactRangeSensor**: Typically run in a shoreside community. Takes reports from remote vehicles, notes their position. Takes a range request from a remote vehicle and returns a range report indicating that vehicle's range to nearby vehicles. Range requests may or may not be answered dependent on inter-vehicle range. Reports may also have noise added to their range values.

3 The uField Toolbox Design

The uField Toolbox is a set of tools to facilitate the deployment of and simulation of multiple fielded marine vehicles. It pre-supposes the arrangement depicted in Figure 1 below. A number of vehicles are deployed and are connected to a single shoreside command and control computer. Each vehicle, as well as the shoreside computer, contain a dedicated MOOS community. A community is comprised of a MOOSDB process and a number of connected MOOS applications.

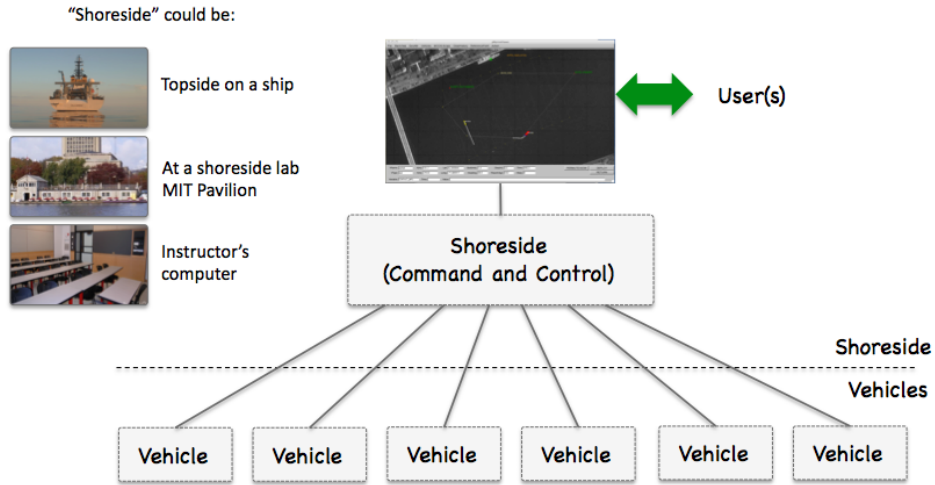


Figure 1: **Shoreside to Multi-Vehicle Topology:** A number of vehicles are deployed with each vehicle maintaining some level of connectivity to a shoreside command and control computer. Each node (vehicles and the shoreside) are comprised of a dedicated MOOS community. Modes and limits of communication may vary.

The uField Toolbox is designed to support both laboratory simulations as well as fielded experiments with multiple marine vehicles.

The uField Toolbox in Laboratory Simulations

In laboratory or classroom simulations, the nodes in Figure 1 may be either (a) all running on a single user's machine during initial development and testing, or (b) each running on a separate laptop with the shoreside community rendered on a central laboratory screen.

The uField Toolbox in Fielded Experimentation

During field experiments, the vehicle nodes in Figure 1 are actual fielded vehicles. The shoreside node is a machine on the shore with connectivity to each of the vehicles. For example, at the MIT Sailing Pavilion lab, the vehicles are either kayaks, kingfishers or a man-portable UUV. Communications is typically achieved with WiFi, and occasionally over acoustic link, or cellular network link.

4 Motivations for the uField Toolbox

The uField Toolbox is meant to facilitate and simulate. It facilitates aspects of configuration for inter-vehicle communications with [pShare](#) by letting nodes auto-configure with one another. It facilitates the monitoring of properties of several fielded or simulated vehicles by collecting and displaying user configurable information. It simulates properties inter-vehicle communication such as range dependency and bandwidth limitations. It simulates inter-vehicle contact detection as would otherwise occur through AIS or vehicle mounted sensors. And it simulates a range-only sensor to fixed beacon locations and moving contacts.

Facilitation of Inter-Vehicle Connections

When vehicles are on the Internet, or common local subnet, their inter-vehicle communications are handled by `pShare`. This is the case during multi-vehicle simulations on a single machine, multi-vehicle simulations between a room full of laptops, and field exercises at a facility like the MIT Sailing Pavilion when all vehicles are connected by a local WiFi router. It is even the case when vehicles are connected using the cellular telephone network. When or if a vehicle is connected via an acoustic modem, or via a satellite link, `pShare` is replaced with a dedicated device interface.

A hurdle to using `pShare` has been the configuration of `pShare` on each machine to properly point the IP address and community name of the vehicles to which it wishes to communicate. If the IP address and vehicle/community names of all machines never change, this may be tractable but cumbersome. When the IP addresses are not known at run time, the configuration problem is often a deal-breaker. In conjunction with the development of the `uField` Toolbox, a key modification to `pShare` was made to allow it to accept incoming mail specifying new bridge configurations. The `uFldNodeBroker` app may be run on each vehicle to communicate in a generic manner to the shoreside MOOS community running `uFldShoreBroker`. These two apps communicate to share IP and host information and request modifications to their own local `pShare` instances, automatically configuring them for all further communications. All that is needed on the vehicles is a list of possible shoreside IP addresses to try for initial connections. Collectively these tools greatly simplify the `pShare` comms configurations in both simulation and fielded exercises.

Facilitation of Multi-Vehicle Status Fields

Deploying multiple vehicles makes it increasingly important to have access to key information across all vehicles in a concise, user-configurable format. For example, it may be good to know which vehicles have had start-up problems, which vehicles are running low on power, or the total trip distance or maximum noted speed for all vehicles. The `uFldScope` tool is built to address this. It exploits a convention that many MOOS status messages are in the format of comma-separated parameter=value pairs. The `uFldScope` tool may be configured to register for any such status message, provided a field to scope, and provided a field indicating the vehicle name. It produces a series of terminal-output reports in a simple tabular format with a row for each vehicle and column for each scoped piece of information. The tool may be configured to produce reports on multiple different sets of scoped data, allowing the user to toggle through the reports.

Simulating Inter-Vehicle Contact Detection

Detecting the position of another vehicle may be necessary (a) for performing collision avoidance, (b) for coordination with a collaborating vehicle, or (c) for strategizing against a competitor vehicle. Knowing another vehicle's position may come from either communication of vehicle position through a general system such as AIS, through direct vehicle to vehicle communication, or through on-board sensors uses to detect and track another vehicle.

The goal of the `uFldNodeComms` utility is simulate the above scenarios. It accepts node reports from *all* fielded or simulated vehicles, and passes on this information to other vehicles. The user may choose a configuration where every vehicle knows the position of every other vehicle all the time. Or a configuration may be chosen where vehicle report sharing is based on the vehicle range.

The user may also choose a configuration where vehicle group or team membership determines, in part, the sharing of node reports.

Simulating Range-Only Sensors

A set of common sensing problems for marine vehicles include (a) determining the location of a fixed object given only a sequence of range-only measurements, (b) determining the location of one's own vehicle given range-only measurements to an object with a known location, and (c) determining the location and trajectory of a moving contact given a sequence of range-only measurements to the contact.

The uField Toolbox includes two tools for simulating these range-only sensors, the `uFldBeaconRangeSensor` and `uFldContactRangeSensor` applications. Each tool is situated on the shoreside computer and awaits the arrival of a "range request" from a deployed or simulated vehicle. The shoreside computer is also receiving node reports from all vehicles and is thus able to factor in relative vehicle position in determining whether or not to reply. Range replies contain the information otherwise perhaps derived from a range-only sensor residing on the vehicle.

Simulating Inter-Vehicle Message Transmission

Message transmission between vehicles is never a given. There is almost always some degree of range dependency, and some degree of uncertainty at any range as to whether the message will get through. The message length itself may also be limited depending on how it is sent. Underwater messaging is particularly subject to all three considerations.

The `uFldNodeComms` utility is designed to simulate message passing between vehicles. This utility resides on a shoreside computer and operates by requiring inter-vehicle message passing to be routed through this utility back out to the receiving vehicles. Since `uFldNodeComms` also is receiving node reports from all fielded vehicles, it may apply the relative position of vehicles in its determination of whether or not to send the message. It may also arbitrarily restrict the size of the message being sent.