iModemSim Operation Handbook

iModemSim was designed and built over a short period of time in March, 2008 to allow the simulated testing of MLBL algorithms that required a MicroModem compatible system to ensure that the delays would not cause algorithm failure.

Overall design

The main modem simulator module is loosely coupled to a MOOS interface, allowing simple repurposing of the module to other simulator systems as needed.

The only inputs needed for the modem simulator core are an IP port, a static or dynamic location (and updates), and a serial port to connect to, to which it would look like a MicroModem.

The design of the communication backend (between multiple iModemSim processes) allows all of the processes running on a local network subnet to talk with each other (UDP broadcasts) as well as with multiple processes on a single machne (internal TCP server), with no configuration needed beyond the port number and network broadcast address.

Usage manual

Command line

iModemSim is a standard MOOS application, it can be started in the same ways as most other MOOS applications:

- `iModemSim your_moos_file_here.moos`

If you want to run iModemSim and not use a physical serial port, you can use the companion program `serial_loopback` to do so (run as root):

- `serial_loopback /dev/ttyLOOPA1 /dev/ttyLOOPA2`

This will create the `/dev/ttyLOOPA1` and `/dev/ttyLOOPA2` as UNIX pseudo-terminal devices (which have an identical interface as a real serial port) and establish a virtual loopback connection between the two. iModemSim would be configured to use one of the "serial" ports, and iMicroModem (or similar) would be configured to use the other.

MOOS mission configuration block

There are a number of configuration variables, many of which need to be set, and they are described below.

- `AppTick`, `CommsTick` - The standard MOOS main loop variables, the defaults of 4 and 4 work reasonably well, although setting it to at least the value (in Hz) of position updates in the system is a good idea

- `Port` - The serial port to connect the simulated modem to (can be a virtual serial port, see above)
- `Speed` - The speed to configure the serial port to (modem is generally at 19200)
- `DefConf` - Default configurations for the modem; the format is `VAR,<int>`. There can be multiple lines of this type. Some useful ones:
  - `SRC` - Modem source ID (i.e `DefConf = SRC,1`)
  - `TOA` - Modem will show time-of-arrival information (`$CATOA`), set to 1 to enable, 0 to disable
  - `SNV` - Modem will only transmit on (simulated) sync pulse, used in conjunction with `TOA` to allow testing of one-way-range algorithms.
- `IPPort` - The port number that iModemSims will use to communicate amongst themselves.
- `BroadcastAddr` - The broadcast address that iModemSim will transmit packets to; what this should be set to depends on the `Bcast` address of the interface you want to transmit the packets on. For example, if all of your computers are on a network `192.168.0.0-254`, then you would set this to `192.168.0.255`. If you want to send the broadcast packets to the interface with the default route (internet), you can set it to `255.255.255.255`; note that these packets will not leave your local subnet.
- `InputLocType` - How iModemSim knows where the modem "is"; this can be set to one of a number of settings, with correspondingly dependant configuration values. If it's set to:
  - `global` - Globally referenced input location that is dynamic, i.e. input variables are doubles of latitude, longitude. In this mode, these variables also need to be set:
    - `InputLocLat` - The name of the variable that contains the latitude (for instance, `GPS_LATITUDE`).
    - `InputLocLon` - The name of the variable that contains the longitude (for instance, `GPS_LONGITUDE`).
    - `InputDepth` - The name of the variable that contains the depth (for instance, `NAV_DEPTH`). Note that all three of these variables need to be updated at least once after iModemSim starts.
  - `constant_global` - Globally referenced input location that is static, i.e. a modem on a simulated dock. In this mode, these variables also need to be set:
    - `ConstantPosLat` - A double that contains the constant latitude assigned to this modem (for instance, `42.3584`).

    - `ConstantPosLon` - A double that contains the constant longitude assigned to this modem (for instance, `-71.08745`).
    - `ConstantDepth` - A double that contains the constant depth assigned to this modem (for instance, `0.0`).
  - `local` - Locally referenced input location that is dynamic, i.e. a modem running with iKayakSim (for example). In this mode, these variables also need to be set:
    - `InputLocX` - The name of the variable that contains the local X position (for instance, `NAV_X`).
    - `InputLocY` - The name of the variable that contains the local Y position (for instance, `NAV_Y`).
    - `InputDepth` - The name of the variable that contains the local depth (for instance, `NAV_DEPTH`). Note that all of these variables need to be set in the database at least once after iModemSim is started.
  - `constant_local` - Locally referenced input location that is static, i.e. a modem at the LatOrigin/LonOrigin. In this mode, these variables also need to be set:
    - `ConstantPosX` - A double that contains the constant X location assigned to the modem.
    - `ConstantPosY` - A double that contains the constant Y location assigned to the modem.
    - `ConstantDepth` - A double that contains the constant depth assigned to the modem.

Note that in both local modes, the Origin (`LatOrigin` and `LongOrigin`) need to be set in the top level of the mission file.