

t. schneider tes@mit.edu 12.08.08
ocean engineering graduate student - mit / whoi joint program
massachusetts institute of technology (mit)
laboratory for autonomous marine sensing systems (lamss)

documentation for uTPB

this process is a program running under the MOOS robotic control software system developed by paul newman: <http://www.robots.ox.ac.uk/~Epnewman/TheMOOS/index.html>

1 problem

often it is beneficial for post mission analysis or new software development to "playback" a log file of a previous mission into a new MOOSDB. this allows the user to interact with the data that was present during the initial mission as if all those processes were running now and experiencing the same inputs. MOOS comes with a similar utility uPlayBack, but uTPB (**u**tility: **T**erminal **P**lay**B**ack) does not require an X-windows environment or the graphical package ftk. it also handles large files with more ease.

2 solution

uTPB:

- allows playback of an ".alog" MOOS log file (asynchronous log, generated by pLogger or zLoggerNG)
- allows playback at rates higher than realtime and seeking to specified locations within the file.

3 usage

3.1 compilation

uTPB uses the ncurses (<http://www.gnu.org/software/ncurses/>) terminal "gui" package in addition to the libraries included in MOOS and moos-ivp-local. all modern linux distributions should include an installer package for ncurses (libncurses-dev in debian/ubuntu).

3.2 example moos file

uTPB does not require a MOOS file to run properly if you are using the MOOS defaults (`ServerHost = localhost` and `ServerPort = 9000`). The usual `AppTick` and `CommTick` parameters are set by uTPB and thus any specified in the .moos file will not be used. To use a different server/port, use a barebones file like this:

```
ServerHost = 18.195.129.21
ServerPort = 9100
```

3.3 running uTPB

you will need an .alog file to read in. pLogger¹ generates such files. another tool, zLoggerNG generates gzip compressed log files (.zlog). to generate an appropriate .alog file from a .zlog, use the the zlogtool with the -t filestart hook:

```
zlogtool -t filestart somelog.zlog > somelog.alog
```

now that you have an alog file either from pLogger or from zlogtool (let's say somelog.alog) you will need to launch uTPB by the following command:

```
uTPB somelog.alog
```

if you want to specify a moos file (for something other than the defaults: ServerHost = localhost and ServerPort = 9000), use

```
uTPB somelog.alog someconfig.moos
```

or

```
uTPB someconfig.moos somelog.alog
```

uTPB will open the alog file and connect to the server. if all is successful you will see a screen like this in your terminal emulator:

```
file: somelog.alog, lines: 1360463
status: stopped
position in file: 00:00:00 | 0.00%
file length: 04:15:58
playback rate: 1
```

```
play | pause | stop | rate | seek
```

use '<' '>' keys (i.e. ',' and '.') to navigate options. type enter to select an option. 'q' quits.

use the less than/greater than keys ('<','>') keys to navigate the options and hit the enter key to select an option. the options do the following

- play: begins or resumes playback of the file into the MOOSDB

¹see p. newman's website for documentation:<http://www.robots.ox.ac.uk/~7Epnewman/TheMOOS/index.html>

- pause: pauses playback at the current point in the MOOS file
- stop: stops playback and resets the file to the beginning (think like pressing stop on a CD player)
- rate: change the playback rate. enter a number between 1 and 20.
- seek: jump to a time location in the file. enter the time in HH:MM:SS format. (e.g. if i wanted to go 11 min, 23 secs into the file, i would enter '0' '0' '1' '1' '2' '3' 'enter'. the screen will give you feedback.

to quit, type 'q'.

4 details

- the entire .alog is read into memory. hopefully you have enough ram. otherwise consider shrinking your file with grep. if you wanted just NAV_X and NAV_Y from "logfile.alog" you could do this²:

```
head -n 5 logfile.alog > shortfile.alog;
egrep " NAV_[X|Y]" logfile.alog >> shortfile.alog;
```

then, run uTPB on shortfile.alog. in the future, something like this may be automatically available in uTPB.

- it is possible, depending on the amount of data in the .alog and the speed of your computer, MOOS will get behind in processing mail if you run uTPB at high rates. reduce the rate, or reduce your file by using the above tip (or buy a faster computer, but that's a bad solution).
- the alog file format is missing data, namely a key as to whether a MOOS variable is (double) or (string). for example, (double)4.52356 is written to the MOOS file the same way as (string)"4.52356", namely simply just 4.52356. uTPB deals with this issue by calling all strings with just characters '0-9', or a leading '-' or a single '.' doubles and everything else strings. this is a problem if the original MOOSDB contained the string "3.214", for example.
- variables are published with the appropriate original MOOS process, rather than uTPB. the community is not preserved since it is not logged.

²the head command grabs the necessary five lines of header in the alog file