

Tutorial for LTLMoP Toolbox – DRAFT v2

This is a tutorial for the LTLMoP (Linear Temporal Logic for Mission Planning) toolbox. If you have any concerns or suggestions, please contact us via email (cpf37@cornell.edu or gj56@cornell.edu).

Installation:

Currently, LTLMoP is supported on Linux (tested with Ubuntu), OS X, and Windows1 (tested with XP).

Below is a general installation guide for all system. A guide for Linux can be found in the ltlmop package

There are, unfortunately, a large number of dependencies that must be installed before you can run LTLMoP. In order of necessity, they are:

- An SVN client
Most versions of Linux come with an SVN client installed. For Windows, we recommend TortoiseSVN (<http://tortoisesvn.net/downloads>).
- A Java Runtime Environment (JRE)
This may already be installed on your computer; otherwise, see <http://www.java.com/en/download/manual.jsp>
- Python 2.x (<http://www.python.org/>)
Most versions of Linux come with Python. For Windows, an installer is available: <http://www.python.org/download/releases/2.6.5/>.
- wxPython (<http://wxpython.org/>)
A cross-platform GUI library.
- NumPy (<http://numpy.scipy.org/>) and SciPy (<http://www.scipy.org/>)
Python libraries that provide matrix operations and optimization routines.
- Player/Stage (<http://playerstage.sourceforge.net/>)
For simulation and connection to Player-enabled robots. Be sure that the Python Player bindings are also built and installed (if compiling from source, this will be visible in the output of the configure script; if installing from a package manager, it's probably already enabled). We recommend that you use 2.0.x versions.
- Polygon (<http://polygon.origo.ethz.ch/download>)

For decomposing concave polygon to convex polygons. We recommend that you use 2.0.3 version

However, be forewarned that Player/Stage may be difficult to get working with Windows. (This is not important, of course, if you have no interest in interfacing with Player/Stage).

Also, optionally:

- Dotty (<http://www.graphviz.org/>)

Used to visualize automata.

- wxGlade (<http://wxglade.sourceforge.net/>)

For developing the Specification Editor GUI

- Java JDK

For recompiling the automaton synthesis code, if you wish to alter it

Download the latest version of LTLMoP from SVN (<http://code.google.com/p/ltlmop/source/checkout>).

On Linux/OS X: From a terminal, run `svn co http://ltlmop.googlecode.com/svn/trunk/ ltlmop-svn`

On Windows: Create a folder somewhere, right-click on it and do an SVN checkout of <http://ltlmop.googlecode.com/svn/trunk/>

For answers to commonly encountered problems during installation, please see the Troubleshooting section on the Google Code wiki: <http://code.google.com/p/ltlmop/wiki/TroubleshootingAndFAQ>.

Using LTLMoP:

Create a new project:

Run SpecEditor (specEditor.py)

Give your project a name by saving it (File >> Save)

Import a robot by File >> Import >> Robot Description file. The program will show all sensors and actions available for the robot. You can uncheck the sensors or actions that are not used for current project (to make the automaton smaller). You can define a new proposition by pressing “New” button below “Custom Proposition”.

Next we’ll need to define our workspace that the robot will move around in. Open up RegionEditor by pressing the “Edit Regions” button near the region proposition list or import a pre-existing region file by File >> Import >> Region file. SpecEditor will update its region proposition list to reflect this new workspace. Make sure the region file is saved in the same folder with the project. In addition, one region named “boundary” is required in order to run the ltlmop. Draw a region that covers every other region in region editor and rename it to be “boundary”. The region will become hollow indicating it is a boundary.

Write the specifications. You can enter a region name by just typing it, selecting it from a map (Hit “Select from Map” button and select the region) or by selecting it from the region name list. You can also enter a sensor, action or custom proposition name by double click it in the list.

Save the project after finishing writing the specifications.

Compile the specification to an automaton by Run >> Compile. If the log at the bottom part shows “Automation Successfully Synthesized”, your specifications are ready for experiment. Otherwise, please check your specifications.

Configure new experiment:

After the specifications are successfully compiled, you can configure setting for experiment by Run >> Configure Simulation.

Click “Add” button to add a new experiment. You can change the experiment name by editing “Experiment Name”.

Check initial sensor, action and custom proposition in the list below. Make sure the settings are consistent with your specifications.

Choose simulation environment and robot.

Click “calibrate” to run the calibration interface (note that you will need to have defined calibration points (two points should be enough) for your map in RegionEditor) and follow the instructions to determine the coordinate transformation between your map of the region and the regions in your lab environment.

Click “OK”, if you are done with experiment configuration setting or you can add more configurations. Be sure to set the current experiment you are going to work on by selecting the experiment name from the list before you click “OK”.

Run the experiment:

Before you run the experiment, please save the project file by File >> Save.

Then you can run the experiment/simulation by Run >> Simulate

Thank you for choosing LTLMoP

We are looking forward to hearing feedback from you.