

Published October 26, 2015 by sauravagarwal

FAQ: Belief Space Planning With OMPL

This page contains answers to some common questions about the belief space planning library based on OMPL that I have developed in my research group based on my own work and that of Dr. Ali-akbar Agha-mohammadi. You can find the library on [Github](#).

Who uses this library?

Please visit our research group [EDPLab](#) for more information about our research, you can find links to videos and papers that use this library. This library has also been used by students at MIT for running FIRM on a quadrotor, you can find their code [here](#).

Can you initialize the graph/roadmap with a stored one or save the one generated at runtime?

We have the choice to do both, i.e., we can load a previously computed roadmap or make the planner generate a new one. When the planner starts up, it looks for a FIRMRoadMap.xml file in the top directory. We use this file to load a pre-computed roadmap. If this file is found, the planner will load it and use it (so you get to re-use an old map). After completing planning, the FIRM planner will update the file if it already exists in case new FIRM nodes/edges are added to the graph. If there is no such file, a new roadmap will be generated and saved to the same file name (FIRMRoadMap.xml). You can then move this xml file to the SavedRoadmaps folder for later use.

What are these c11, c12 values in the FIRMRoadmap.xml file?

C11 etc are the covariance matrix values for the FIRM nodes. When the state space is x,y,theta (2d). The covariance matrix is a 3×3 matrix and these are its entries.

RECENT POSTS

January 2021

July 2018

December 2017

August 2017

July 2017

June 2017

November 2016

October 2016

July 2016

May 2016

February 2016

January 2016

November 2015

October 2015

September 2015

How do you handle collision checking?

We use the FCL collision checker library wrapper provided by OMPL to handle the collision checking and cad objects (mesh). Please check with the FCL / OMPL developers for your questions about how these work.

What are the landmarks and how can I provide my own map of world features?

First of all, by map here we mean the map of the features / landmarks in the world. There is a set of setup files in the “SetupFiles” folder which contain information about the landmarks, the robot’s motion and observations model etc. We can choose which setup file to use by setting the correct path in the main.cpp file. You can create your own setup by creating a new setup xml file and adding the marker locations, robot motion noise, observation model noise etc. accordingly.

How are the robot and environment mesh loaded?

Please have a look at the file FIRM2DSetup.h, in the “loadParameters” function to see how we load in the robot mesh.

Can I load obstacles while the planner is running, i.e., dynamic obstacles?

Currently, it is only possible to load a mesh in the beginning in the beginning of the simulation. You cannot insert meshes while the planner/simulation is running (dynamic obstacles). This would be a great feature to have but I haven’t implemented it yet :(.

Is it possible to use a Quadrotor or any other robot model?

Yes yes yes! The planner is independent of what robot you have. We define a motion model and observation model class. You can define a new motion model and observation model for your robot (could be a quadrotor). Just like we have [UnicycleMotionModel.cpp](#) you can create your own ABCMotionModel derived from MotionModelMethod (cpp and header file). Please look at our code which is commented for your convenience. For example, since a quadrotor pose has 6 dimensions (x,y,z,roll,pitch,yaw), you would need to create a new belief space (SE3BeliefSpace). Good news is that some students at MIT have already implemented this stuff in their own fork of our library. You can see their page [here](#). Please fork our library if you would like to add more robot models, we would love to add this stuff back to our repository for everybody’s benefit.

August 2015

April 2015

March 2015

February 2015

October 2014

September 2014

June 2014

CATEGORIES

Drones

Entrepreneurship

Food

General

Life Hacks

Research

robotics

Software

Thoughts

Tutorial

[Previous Post](#)

[How To: Tips and Tricks in Latex](#)

[Next Post](#)

[How To: Setup Ubuntu on MacBook Pro 2015](#)



[sauravagarwal](#)

 [RESEARCH](#) [SOFTWARE](#)

 [BELIEF SPACE](#) [C++](#) [FIRM](#) [M3P](#) [MOTION PLANNING](#) [NBM3P](#) [OMPL](#) [SOFTWARE](#)

2 Comments



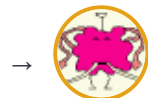
Ali said:

Hello Saurav,

Does Dynamic Feedback Linearization (DFL) control used in these codes? I couldn't find where it is implemented.

Thanks for your help

March 18, 2016 | [Reply](#)



sauravagarwal said:

Hi Ali,

DFL is not implemented in this code currently. If you're interested in implementing DFL FIRM, let us know we would be happy to take your contribution.

Regards
Saurav

RECENT COMMENTS

sauravagarwal on [Why I work on Forklift Safety](#)

Bill Ryan on [Why I work on Forklift Safety](#)

Seon on [A practical introduction to pose-graph SLAM with ROS](#)

Bolaji on [Startups: A great early-stage startup CEO needs to be a killer COO](#)

Abdulwaheed on [The Best Way to Convert Latex Math Equations into Word Equations](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Enter Your Comment...

Your Name*

Your Email*

Your URL (optional)

Post Comment

