4/26/17

current issue:

1. Add the condition about the obj improvement to determine the trust region modification to see if any better result can happen. Current version does not work well.

The resutls computed from cvx and from analytical computation are different. One possible reason is that in the objective function I removed a term that involves the determinant of covariance matrix.

1. A big bug: the sensor modeling method may not work when the sensing angle is greater than or equal to 180 (explain this in the paper to say we can always decompose into a disjoint set of convex shapes)
2. debug the whole program, think about using motion primitive for multiple initial guesses (e.g. moving towards MAP)

9/5/17

Nothing new since last note. Things to do:

1. the obj quadratization seems incorrect. May need to quadratize wrt z and u, not x and P? Re: after thinking about it, I think it’s fine to be wrt x and P.
2. changed SQP according to Abbeel’s work. The problem now is that numerical Hessian is very very slow to compute. Fix this! May try using the diagonal approximate hessian method that Abbeel used. (Abbeel’s work only uses 1st order approx! Now I have changed to also use the 1st order.)
3. Trust region now only affects z(1:3). I feel that trust region should be applied to all states and inputs. Now the problem is that the actual merit function improvement is not large enough. Try is putting trust region for all states and inputs help.(add trust region to all z and u)
4. SCP always gives wrong result. Debug!
   1. Debug using KF. Check the gradient to understand if the infeasibility issue. Add feasibility recovery function.
   2. Debug using PF. The result is quite wrong. Debug code to see why it’s different from the KF code. Then understand why KF planner makes the robot move past the target
   3. try the method from Num Opt book, which seems to use Lagrangian.
5. add code for FOV shape computation when obstacles exist. Use prob(occlusion) as gamma
6. add collision avoidance constraint
7. consider stochastic component (probability of occlusion, collision avoidance)