

Objective

Autonomously navigate the Jackal between any two points A and B in a mapped environment with some level of dynamism (e.g. Tech 1st floor). ROS will be used as the framework, and the majority of the required code will be written in C++. Tie in concepts from concurrent computer vision and sensor network courses wherever appropriate, as well as past course work (ME495, EECS469).

Process, Milestones, Timeline

- | | |
|--|---------|
| 1. Pework: gather required info, determine acceptable environment (e.g. Tech 1st floor), get or make map of environment, assess feasibility with Jarvis / prune project scope, etc | 1 week |
| 2. Communication between Jackal and laptop over NU wireless (for development, e-stop / controls override, video feed, debugging, etc), control via wireless commands (from laptop) | 1 week |
| 3. Velodyne/Xtion/Kinect communicating with Jackal's on-board computer (and transmitting data to laptop, or at least a subset of it, if bandwidth allows) | 1 week |
| 4. Implement localization / Bayesian filter (NavStack or fix my defunct particle filter?) | 2 weeks |
| 5. Implement local and global planners (NavStack or my crappy version of A*?) | 2 weeks |
| 6. Create an algorithm to differentiate between static and dynamic obstacles, and potentially between human / non-human. | 3 weeks |
| 7. Project demonstration, documentation, wrap-up | 1 week |

Stretch Goals

- full SLAM implementation instead of providing environment map a priori (just use NavStack)
- tie in machine learning / AI (no specific ideas on what exactly to tie in, could be something to address a deficiency discovered during the project)
- identify the weakest component of the project and implement my own improved version, compare performance between the two
- expand operating range from initial environment

Questions / Unknowns

- How powerful is the Jackal computer? Will it meet my computational needs?
- LIDAR produces a ton of data - can we transport the data via wireless or are we going to have to do all LIDAR/depth camera processing related stuff on-board? It would be nice to be able to visualize this...
- Velodyne unnecessary? Use Xtion/Kinect instead? If using Xtion/Kinect, I'd like to have a way to turn and look in different directions.
- How much work will it be just to get the Jackal moving? Does it have its own control software preloaded?
- Would this project be better suited for the YouBot?