Gemini 101-I

Ajinkya Jain

With help from Max and Yuchen, Thanks guys!

Outline: Part -1

- System Set-up
- Manipulation
- Perception using AR-Tags
- Navigation
- Coding Style guide
- Demos

How to setup the system?

- Core code repos: https://github.com/StanleyInnovation/vector_v1
  - Branch: Gemini-7dof
  - Parameters: (~/vector_ws/vector_v1/vector_common/vector_config/vector_config.sh)
    - VECTOR_HAS_TWO_KINOVA_ARMS=true (Gemini/Poli ⇒ True/False)
    - VECTOR_HAS_TWO_ROBOTIQ_GRIPPERS=true (Gemini/Poli ⇒ True/False)
    - VECTOR_HAS_KINOVA_7DOF_ARM=true (7dof/6dof⇒ True/False)

- Hlpr- Repos:
  - Documentation: https://github.com/HLP-R/hlpr_documentation/wiki
  - Branch: Gemini- 7dof

How to start the simulation?

- Run each in a new terminal:
  - Launch Gazebo (Equivalent to powering on robot):

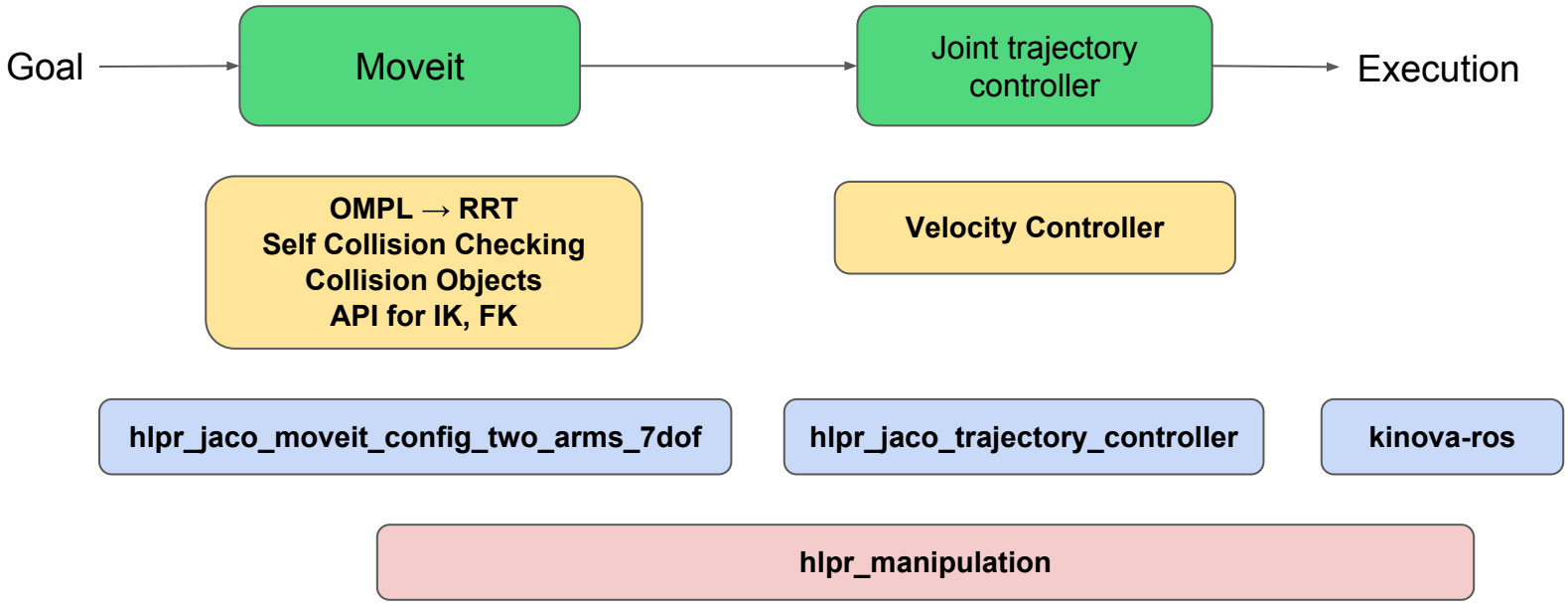
    ```
    roslaunch hlpr_gazebo vector.launch
    ```

  - Launch Moveit: `roslaunch hlpr_jaco_moveit_config_two_arms_7dof hlpr_simple_moveit.launch`
  - Launch rviz: `rosrun rviz rviz`

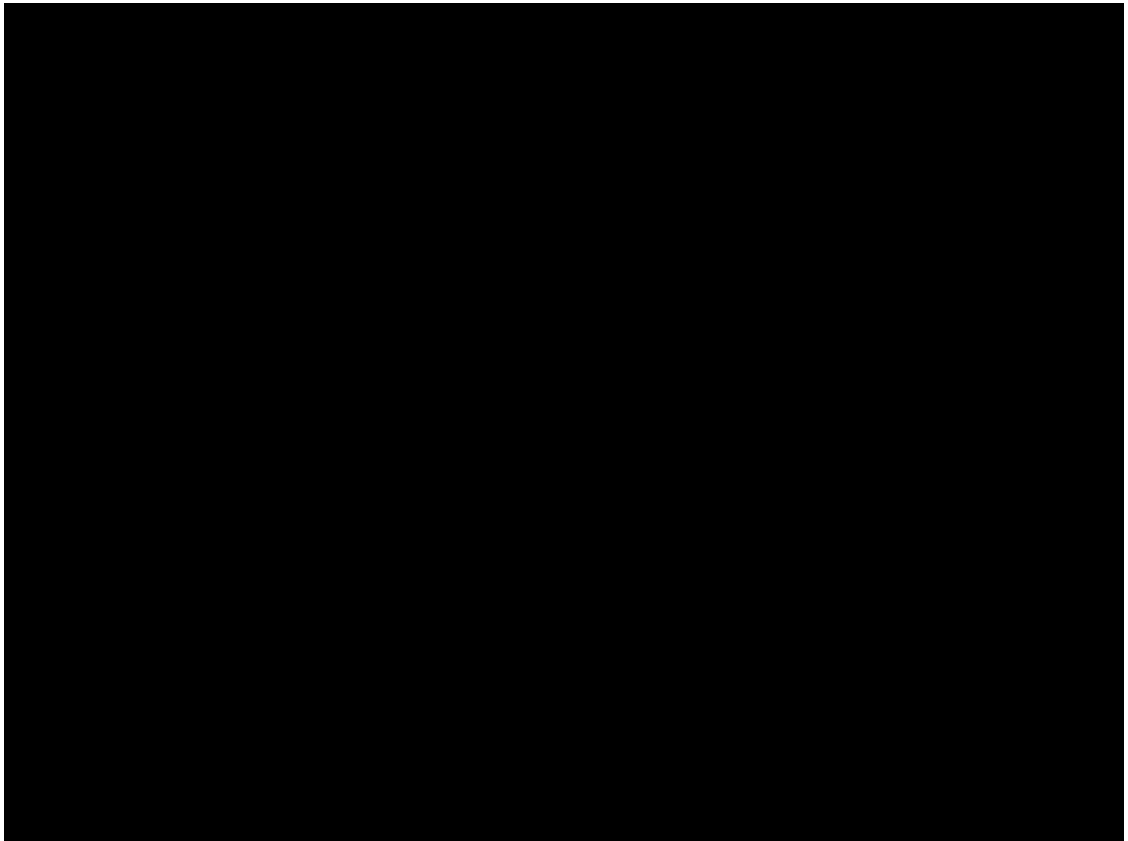
How to start the robot?

- Hardware Power switch:
  - Normal Mode → Press and release
  - Bootloader Mode (To update system firmware) → Press, hold for 3-4 seconds, release
- Software Power switch: vstop, vstart
  - When to use these?
    - When you recompile vector_ws on the robot
    - Drivers aren’t responding
- Debugging:
  - vchk

Manipulation Pipeline



Arm Teleop: RViz → Motion Planning Panel



Manipulation: Development

- Package: hlpr_manipulation (https://github.com/HLP-R/hlpr_manipulation)
  - Branch: gemini-7dof
- Hlpr classes:
  - Arm Control: hlpr_manipulation_utils/arm_moveit2.py
  - Gripper: hlpr_manipulation_utils/manipulator.py
- Gripper:
  - Teleop via commandline: `rosrun gemini_experimental_utils gripper_utils right/left`
- Reference: Demo 1

How to control: Linear Actuator

- Publish to topic:
  - Linear Actuator
    - Topic: /vector/linear_actuator_cmd
    - Data type: vector_msgs/LinearActuatorCmd
    - Range: [0., 1.] ⇒ Height
    - Command:

```
rostopic pub /vector/linear_actuatormd vector_msgs/LinearActuatorCmd
"header:
  seq: 0
  stamp:
     secs: 0
     nsecs: 0
  frame_id: ''
desired_position_m: 0.5
fdfwd_vel_mps: 0.0"
```

How to control: Pan

- Publish to topic
  - Pan
    - Topic: /pan_controller/command
    - Data type: std_msgs/Float64
    - Range: [-1.0, 1.0] ⇒ [0 deg, +90 deg] along z (base frame)
    - Command:

```
rostopic pub /pan_controller/command std_msgs/Float64 "data: -0.75"
```

How to control: Tilt

- Publish to topic
  - Tilt
    - Topic: /tilt_controller/command
    - Data type: std_msgs/Float64
    - Range: [-1.0, 1.0] ⇒ ~[-60 deg, +60 deg] along y (base frame)
    - Command:

```
rostopic pub /tilt_controller/command std_msgs/Float64 "data: 0.5"
```

Perception: AR tags

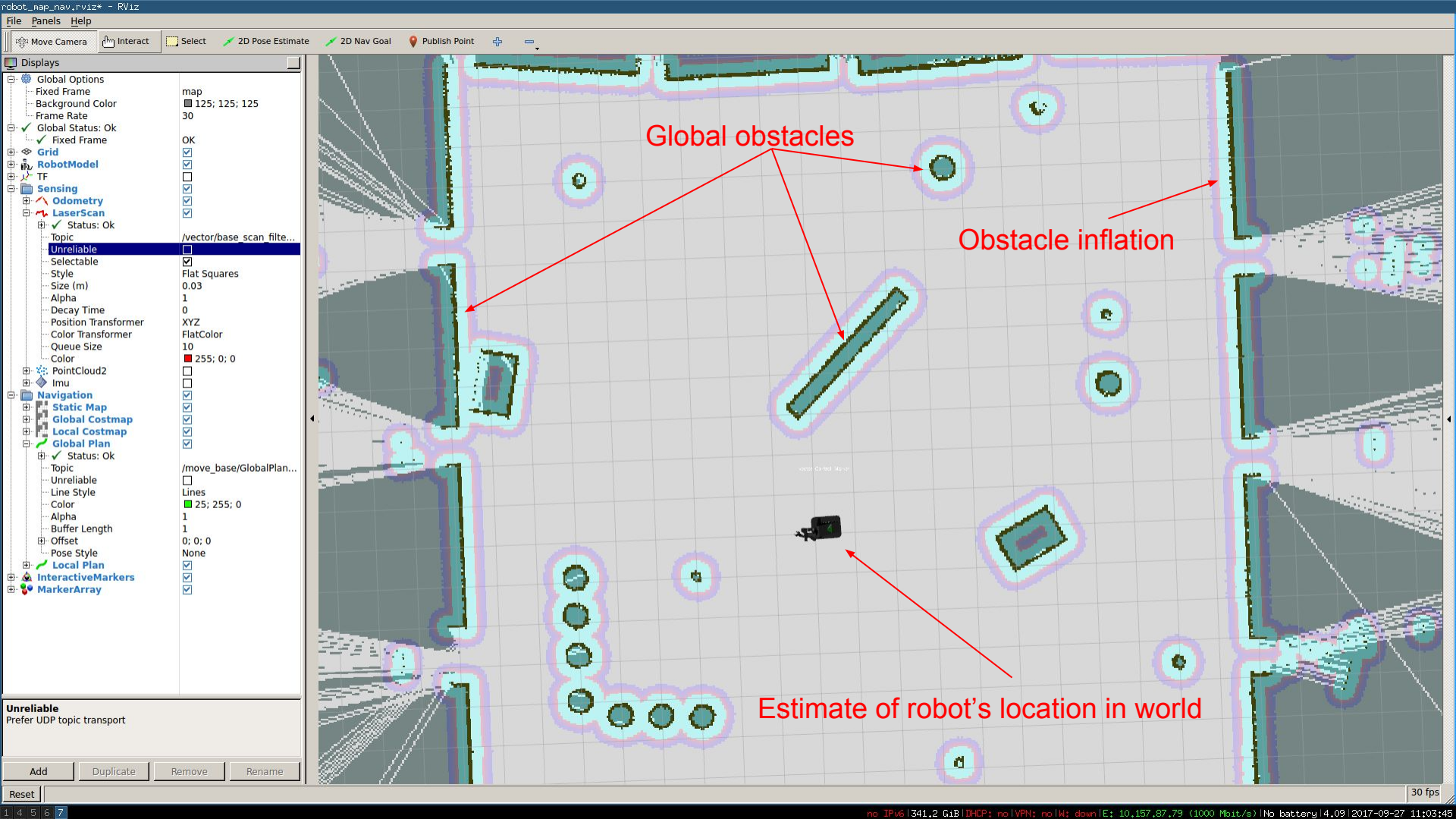
- Ar-tag tracking:
  - Package: http://wiki.ros.org/ar_track_alvar
  - Installation: vector2
  - Setup:
    - Individual Markers
    - Multi-tag Bundles → Generate Bundle file
  - Launch:

```
roslaunch ar_track_alvar <launch_file>
```

Vector Navigation Overview

(courtesy: Max)

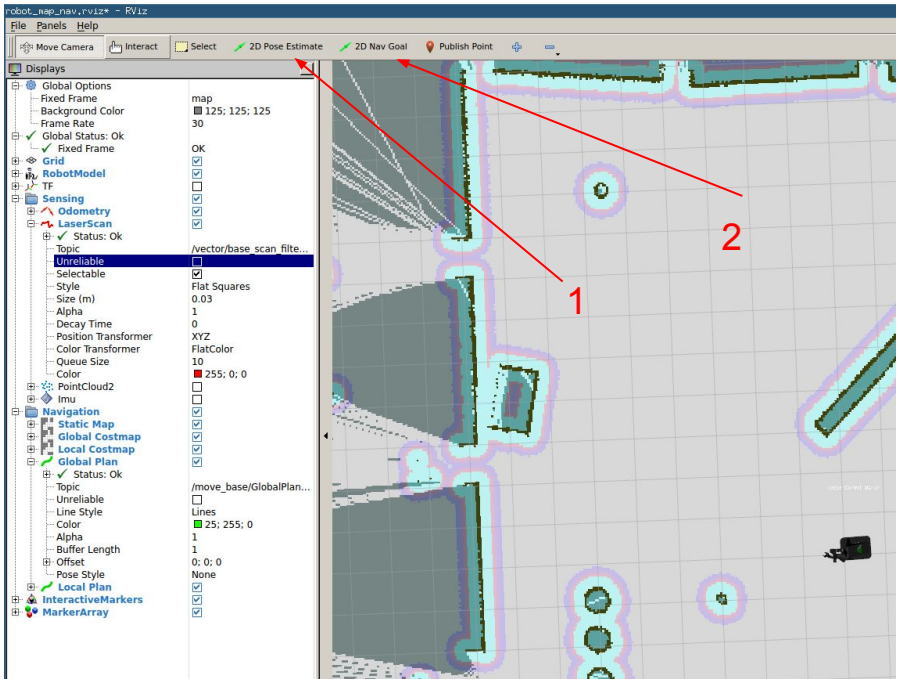
- Uses sensors and odometry feedback to localize to environment
- Environment is defined by a 2-D map
- Goal coordinates are accepted in this global map frame
- Obstacles are tracked in 2-D occupancy grids
  - Global obstacles - affect initial navigation plan generated; also used for localization
  - Local obstacles - cause the navigation plan to be updated to avoid collision; obscure localization features
  - All obstacles inflated with cost values
- Is not informed of 3D space outside of base perimeter
  - Unaware of arm position
  - Unaware of obstacles below and above laser plane
  - Unaware of cliffs (i.e. stairs)


robot_map_nav.rviz* - RViz
File
Panels
Help
Move Camera
Interact
Select
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
map
Background Color
125; 125; 125
Frame Rate
30
Global Status: Ok
Fixed Frame
OK
Grid
RobotModel
TF
Sensing
Odometry
LaserScan
Status: Ok
Topic
/vector/base_scan_filte...
Unreliable
Selectable
Style
Flat Squares
Size (m)
0.03
Alpha
1
Decay Time
0
Position Transformer
XYZ
Color Transformer
FlatColor
Queue Size
10
Color
255; 0; 0
PointCloud2
Imu
Navigation
Static Map
Global Costmap
Local Costmap
Global Plan
Status: Ok
Topic
/move_base/GlobalPlan...
Unreliable
Line Style
Lines
Color
25; 255; 0
Alpha
1
Buffer Length
1
Offset
0; 0; 0
Pose Style
None
Local Plan
InteractiveMarkers
MarkerArray
Unreliable
Prefer UDP topic transport
Add
Duplicate
Remove
Rename
Reset
30 fps
Global obstacles
Obstacle inflation
Estimate of robot's location in world

Vector Navigation Areas

- Creating a map
  - Not covered here; check the Vector / Hlpr docs
  - Do ***NOT*** use Gmapping
- Navigation through RViz
  - Set initial pose
  - Set goal!

Vector GUI Navigation

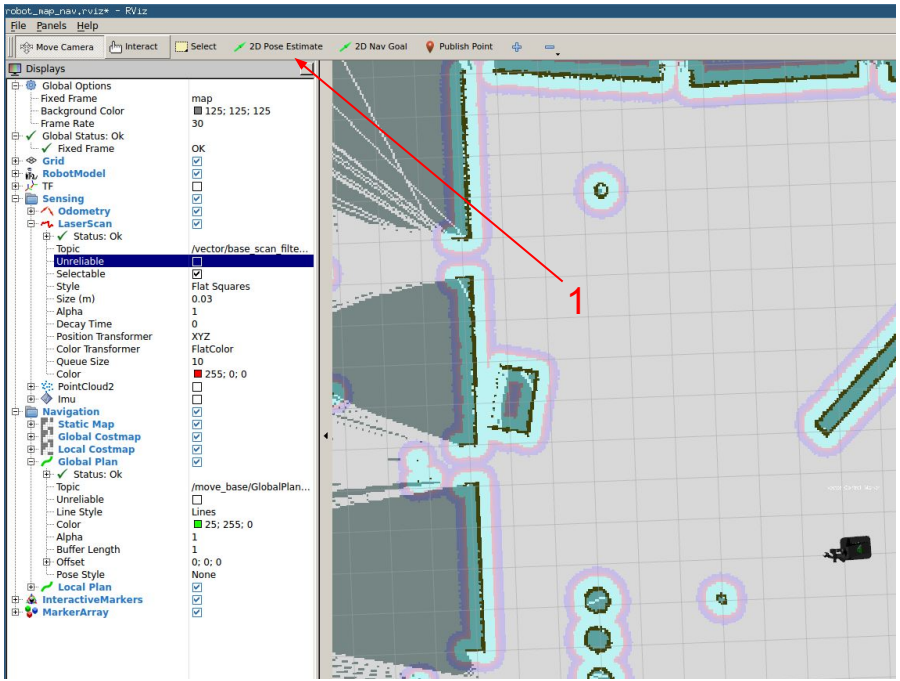


1. Set initial pose estimate
   a. Necessary for localization algorithm (AMCL)
   b. Try to set it accurately so the map aligns with the robot's laser and location in room
   c. You can always give another estimate- AMCL will always use the latest one
2. Set Navigation goal
   a. Click and drag to set location and orientation
   b. Goal is immediately sent to movebase
      i. Plan is generated
      ii. Robot carries out plan

Vector Navigation Areas

- Navigation programmatically
  - Set initial pose
  - Wait for move_base action server
  - Set goal message parameters
  - Send goal to action client

Vector Programmatic Navigation



1. Set initial pose estimate
2. Set Navigation goal
    a. C++
    b. Python (needs slight modification)
    c. When setting goal message fields, make sure the frame_id is correct
        i. Generally frame_id = “map”
        ii. Depends on specific usage

Running navigation on Vector

- Simulation
  - Start gazebo simulator
    - roslaunch vector_gazebo vector.launch
  - Launch navigation stack
    - roslaunch vector_navigation_apps 2d_map_nav.launch sim:=true
- Robot
  - Launch navigation stack
    - roslaunch vector_navigation_apps 2d_map_nav.launch
- Visualize robot
  - roslaunch vector_viz view_robot.launch function:=map_nav

Final Note on Navigation

- Much work has gone into configuring navigation for Poli
  - All of which is on the Poli_config branch of vector_v1
  - Changes include
    - Local planner change
    - Navigation smoother
    - Navigation less likely to run into objects
  - Use Poli's navigation configuration as reference if you run into weird behavior
- If you
  - run into general problems
    - Ask Max!
  - need to generate a map or help using that map for navigation
    - Ask Max!
  - want to venture further and use multi-level navigation and use the elevator
    - Ask Max!

How to conduct experiments?

- Test it in simulation
- Create a new worskpace on the robot. Install your packages in there. Do catkin make. Source your workspace.
- Vector_ws → Only for the basic system functionalities
- Code Style Guidelines: https://github.com/HLP-R/hlpr_documentation/wiki/Coding-Standards

Thanks!