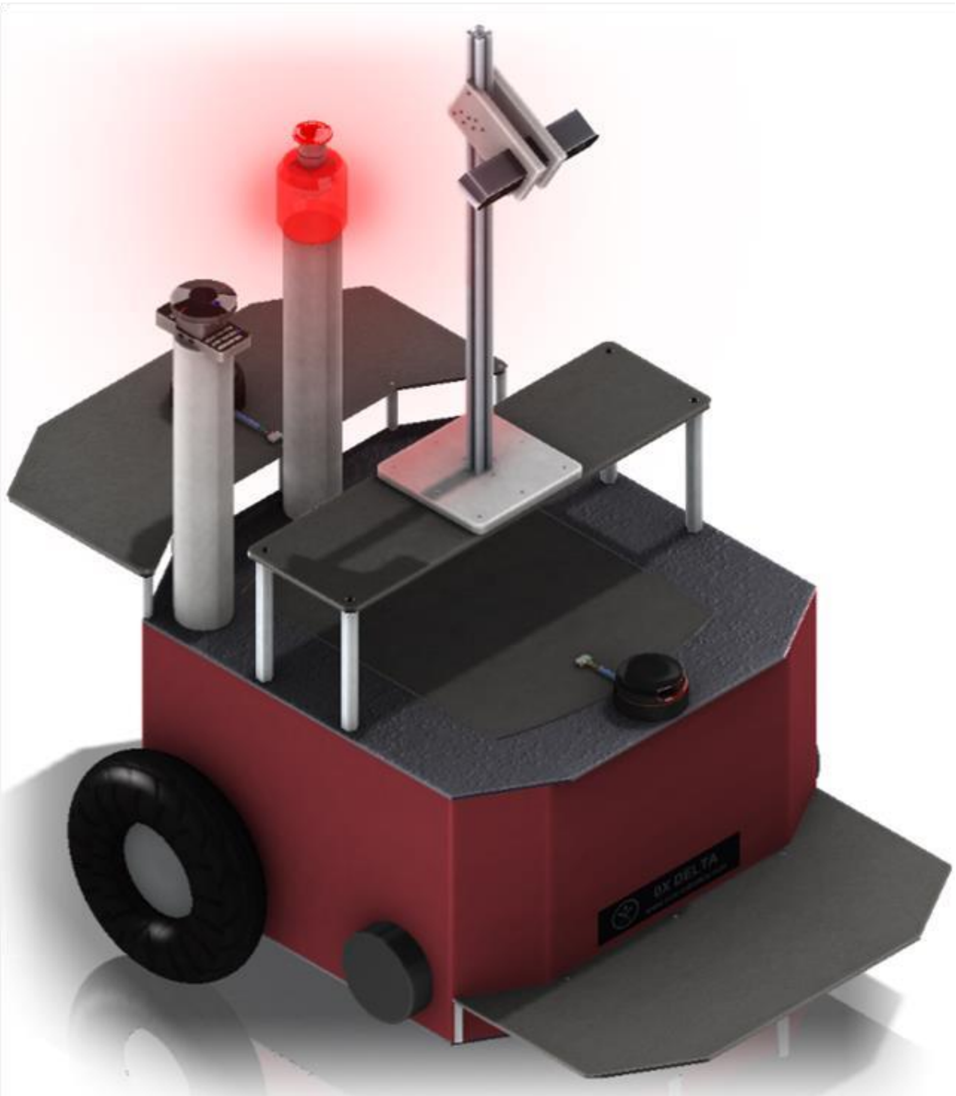
Autonomous Navigation



Mangal Kothari

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Kanpur – 208016

mangal@iitk.ac.in

9460255282

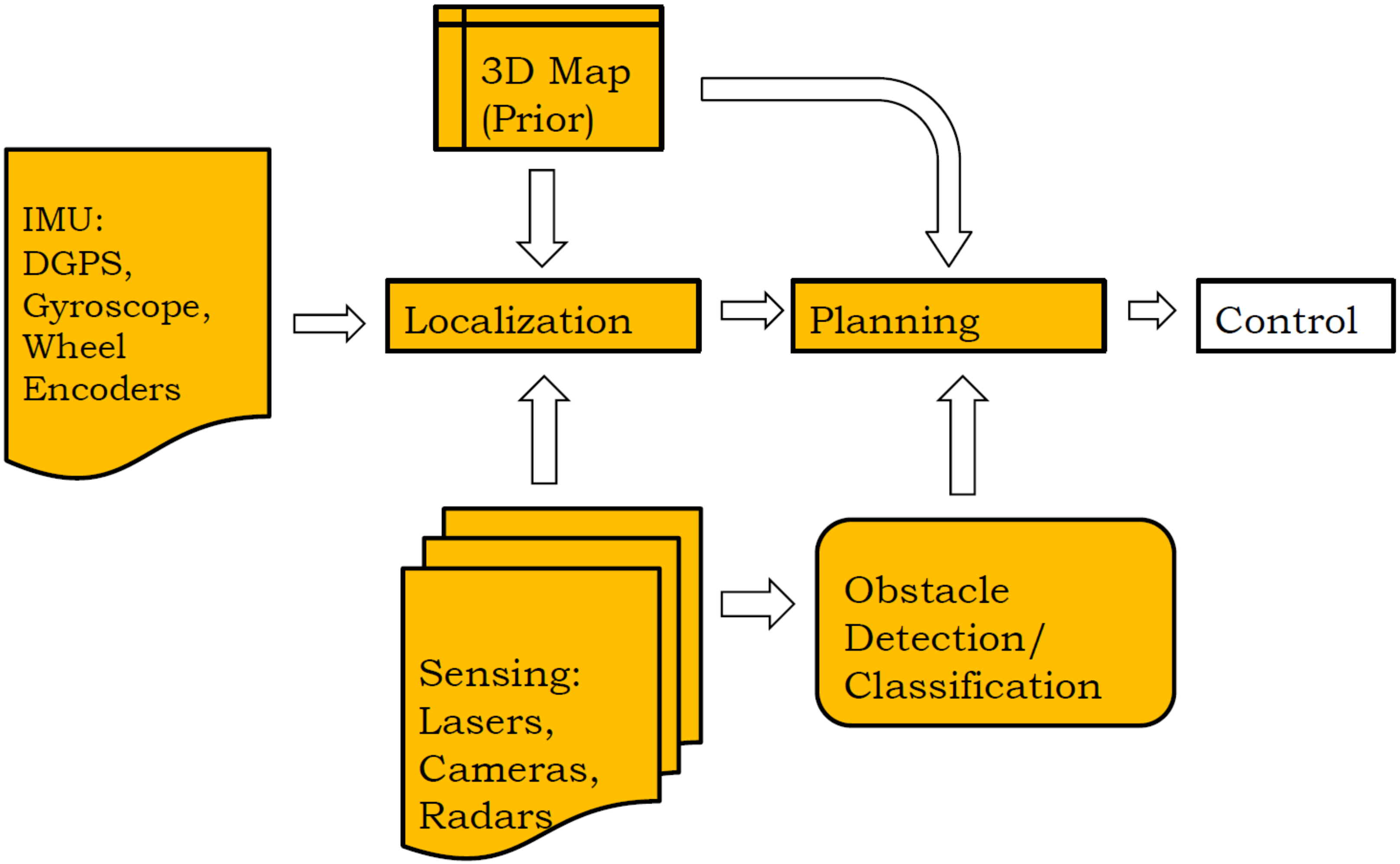
Class Timing: M-12:00-13:15

T-09:00-10:15

TA: Mr. Aalap A Saha

Autonomous Navigation

Autonomous Navigation System



Course Content

I I T K A N P U R

- Introduction: practical examples and challenges – IGVC, SAVe, Mehar Baba competition
- ROS and state estimation (Bayesian filter-Kalman Filter, Extended Kalman Filter, Unscented Kalman Filter), Nonparametric filter (particle filter), Localization, SLAM, Cooperative localization
- Path planning algorithms: Deterministic and probabilistic algorithms, Task allocation algorithms
- Vision and communication systems
- Topics can be added and removed based on **feedback!!!**

Reference

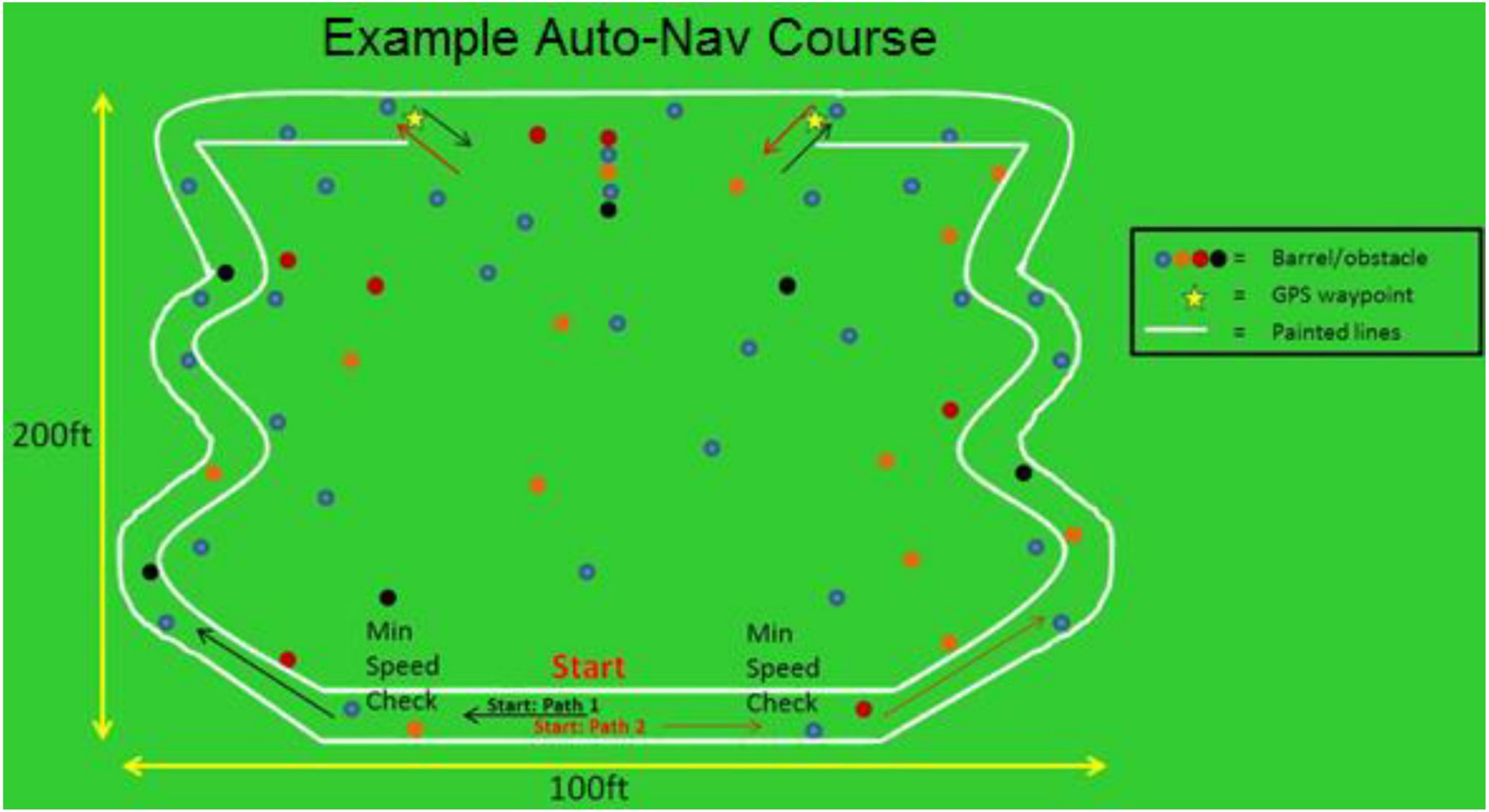
I I T K A N P U R

- Probabilistic Robotics. Sebastian Thrun, Wolfram Burgard and Dieter Fox. MIT press, 2005.
- Principles of Robot Motion: Theory, Algorithms and Implementations, Howie Choset *et al.*. MIT Press, 2005.
- State Estimation for Robotics: Timothy D. Barfoot. Cambridge University Press, 2017.
- A Gentle Introduction to ROS: Jason M. O’Kane. 2013.

Evaluation

- **Assignment – 40%**
- **Project (maximum 3 students) – 40%**
- **Midterm exam – 10%**
- **Quizzes (after midsem) – 10%**
- **Plagiarism – de-register/failed**

Problem Statement



Specifications

- Length: Min 3 feet, Max 7 feet
- Width: Min 2 feet, Max 4 feet
- Height: Max 6 feet
- Propulsion: Battery powered
- Average speed: 1 mph
- Minimum speed: 1 mph for the first 44 feet
- Maximum speed: 5 mph
- Mechanical E stop: Hardware base
- Wireless stop: Effective with in 100 feet
- Safety light: Must be on when vehicle is on
- Payload: 20 pounds, 18”x8”x8”

Qualification

- Mechanical stop and E stop evaluation
- Lane following (with U turn)
- Obstacle avoidance
- Waypoint following

Intelligent Ground Vehicle Competition (IGVC)

I I T K A N P U R

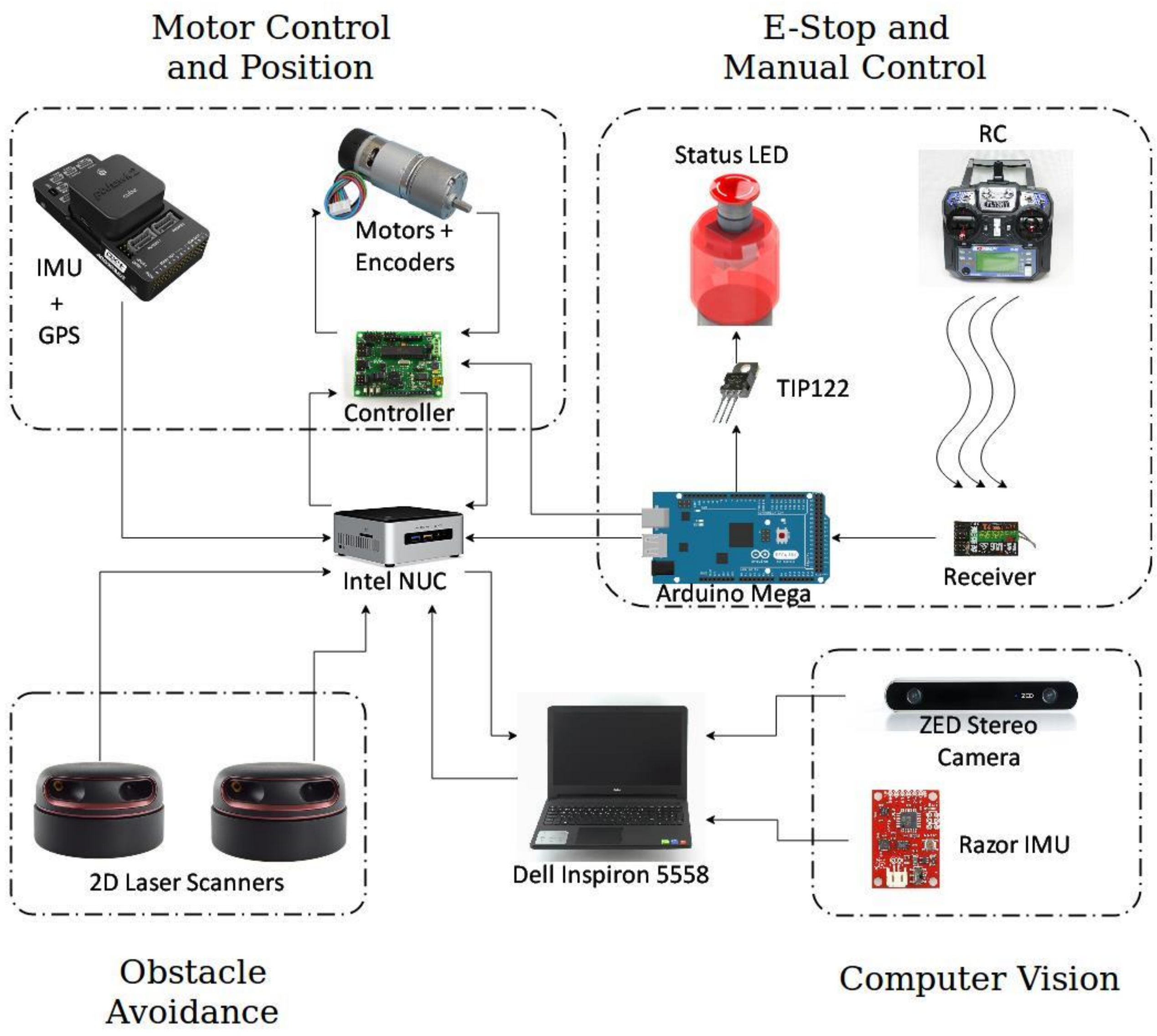
IGVC 2018 GPS Waypoints

North	42.6791159989	-83.1949250546
Midpoint	42.6789603912	-83.1951132036
South	42.6788151958	-83.1949093082
Practice1	42.6783260449	-83.1946867275
Practice2	42.6781974127	-83.1949338822
Qualification2	42.6778987274	-83.1954820799

Vehicle Model

- **Compact design**
- **Switchable vehicle design**
- **Spring based suspension system**
- **Height and angle adjustable camera mount**

System Architecture



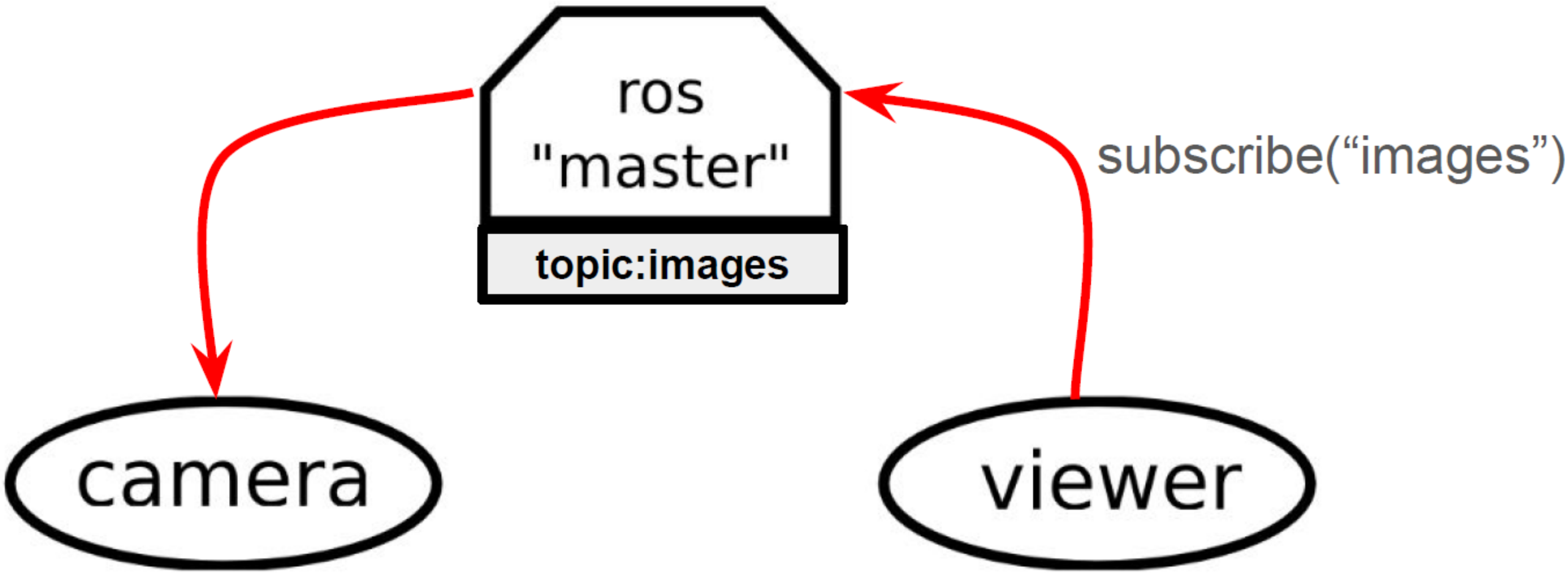
Robot Operating Systems

- **A meta operating system for robot**
- **A collection of packaging, software building tools**
- **An architecture for distributed interprocess/ inter-machine communication and configuration**
- **A language-independent architecture (C++, python, lisp, java, and more)**

ROS Communication Layer: ROS Core

- **ROS master**
  - **Centralized communication server based on XML and RPC**
  - **Registers and looks up names for ROS graph resources**
- **Nodes**
  - **Distributed process over the network (executable runs a separate thread)**
  - **Serve as source and sink for data**
- **Topics**
  - **Asynchronous many-to-many communication**
  - **Publish and subscribe structure**

Asynchronous Distributed Communication



ROS Master

Manage communication among nodes

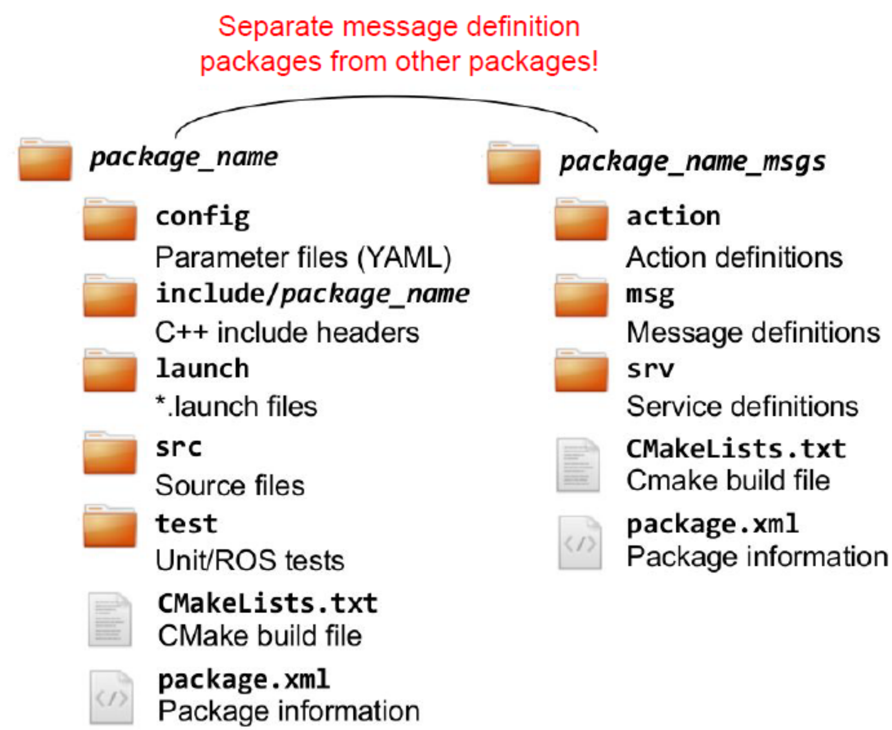
Every node register when at start up with the master

$ roscore

ROS Package

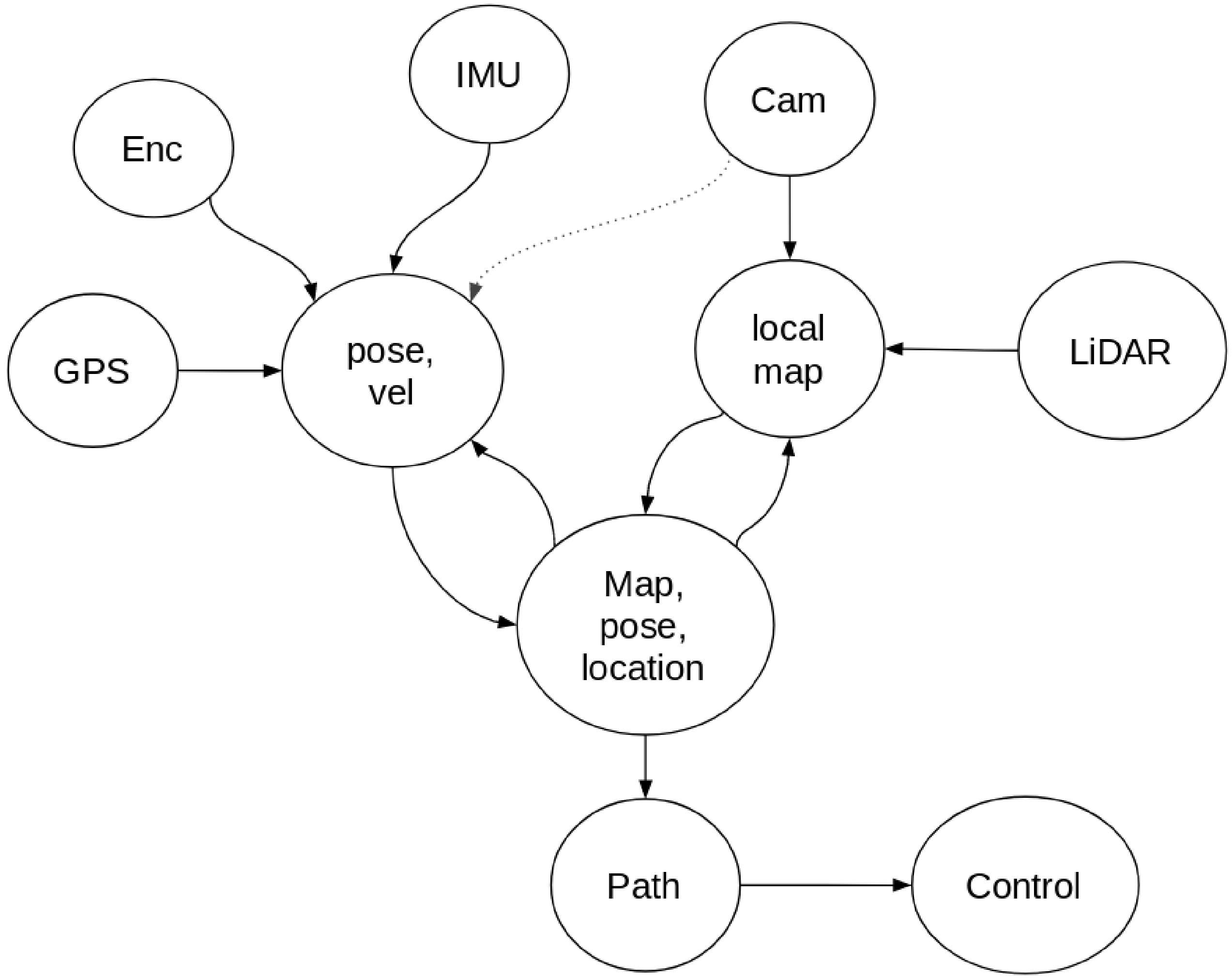
ROS Packages

- ROS software is organized into packages, which can contain source code, launch files,configuration files, message definitions, data, and documentation
- A package that builds up on/requires other packages (e.g. message definitions), declares these as dependencies

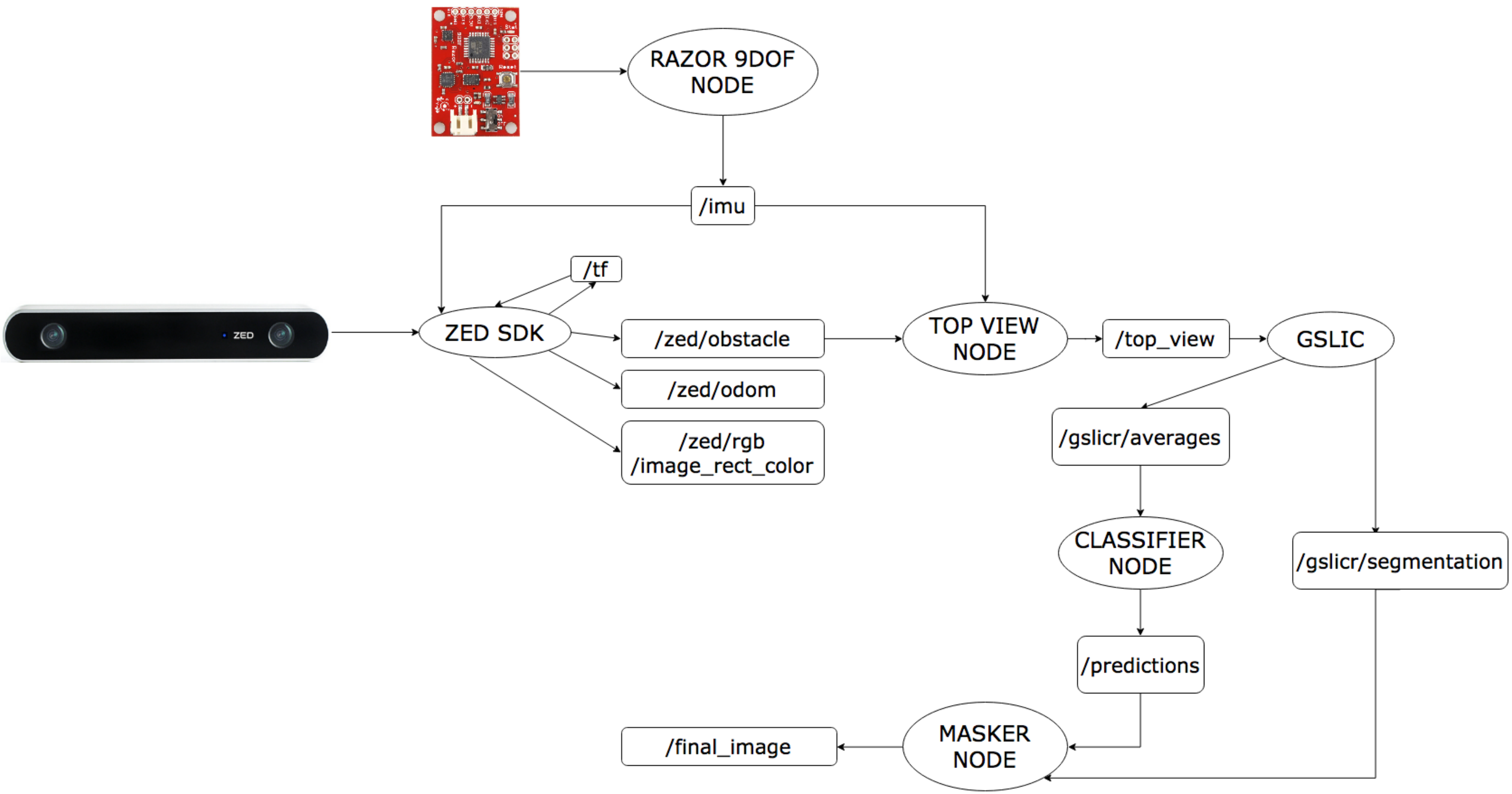


Software Architecture

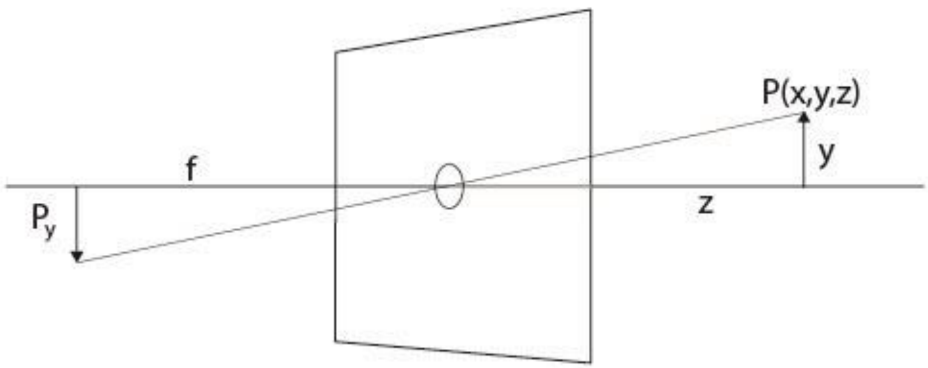
I I T K A N P U R



Lane Detection: Computer Vision



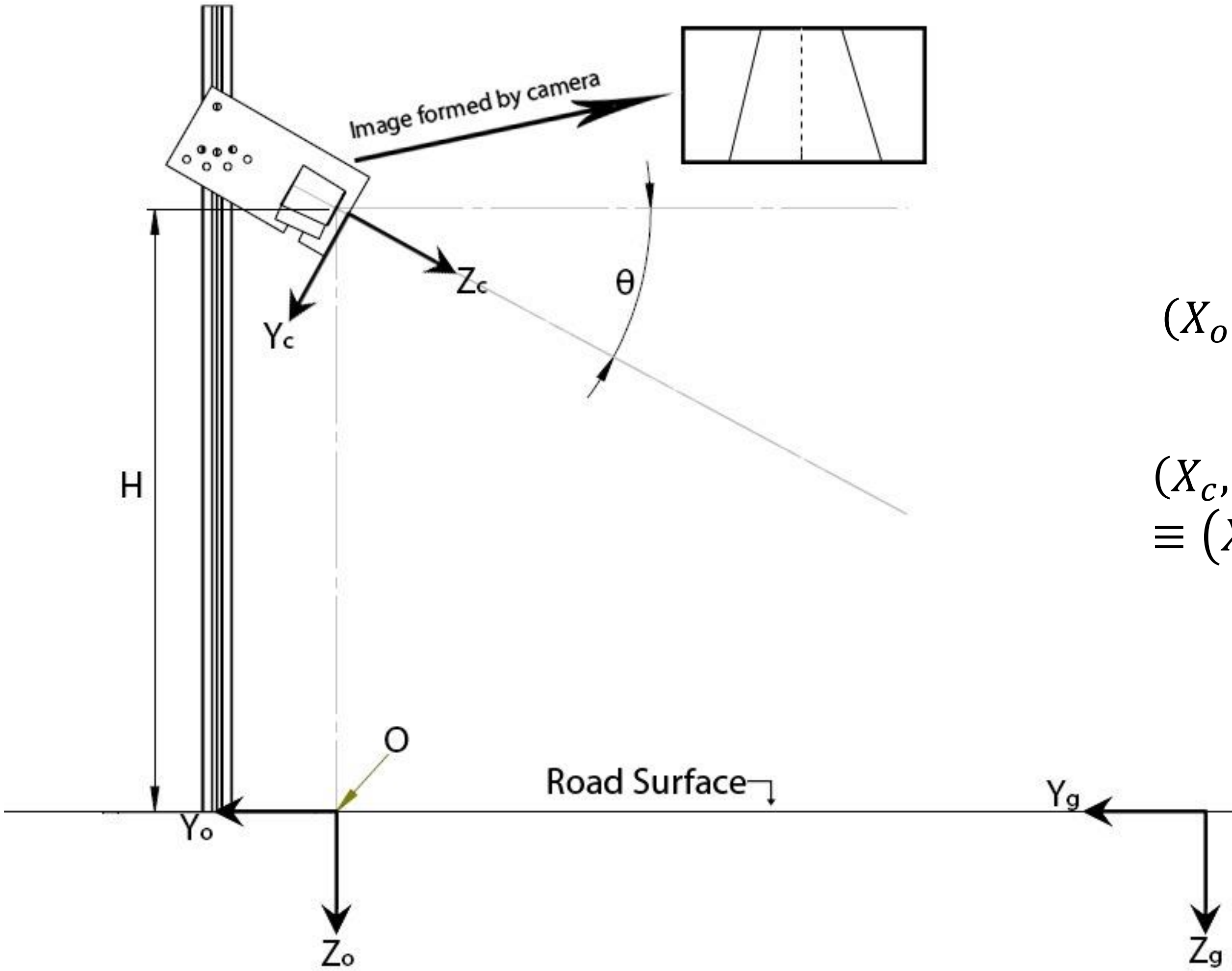
Pinhole Camera Model



$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x = \frac{x'}{w}, y = \frac{y'}{w}$$

Inverse Perspective Transformation



Lane following transformation

$$(X_f, Y_f, Z_f) \equiv \left(\frac{x_f}{s}, \frac{y_f}{s}, 0\right)$$

$$(X_o, Y_o, Z_o) \equiv \left(X_f - O_X, Y_f - O_Y, Z_f\right)$$

$$(X_c, Y_c, Z_c) \equiv \left(X_o, (H + Z_o)\cos(\theta) + Y_o \sin(\theta), (H\right.$$

$$(x_c, y_c) \equiv \left(\frac{X_c}{Z_c} f_x + c_x, \frac{Y_c}{Z_c} f_y + c_y\right)$$

Top View Transformation

$$(x_c, y_c) \equiv \left(\frac{\left(\frac{x_f}{s} - O_X\right)}{H\sin(\theta) - \left(\frac{y_f}{s} - O_Y\right)\cos(\theta)} f_x + c_x, \frac{H\cos(\theta) + \left(\frac{y_f}{s} - O_Y\right)\sin(\theta)}{H\sin(\theta) - \left(\frac{y_f}{s} - O_Y\right)\cos(\theta)} f_y + c_y \right)$$

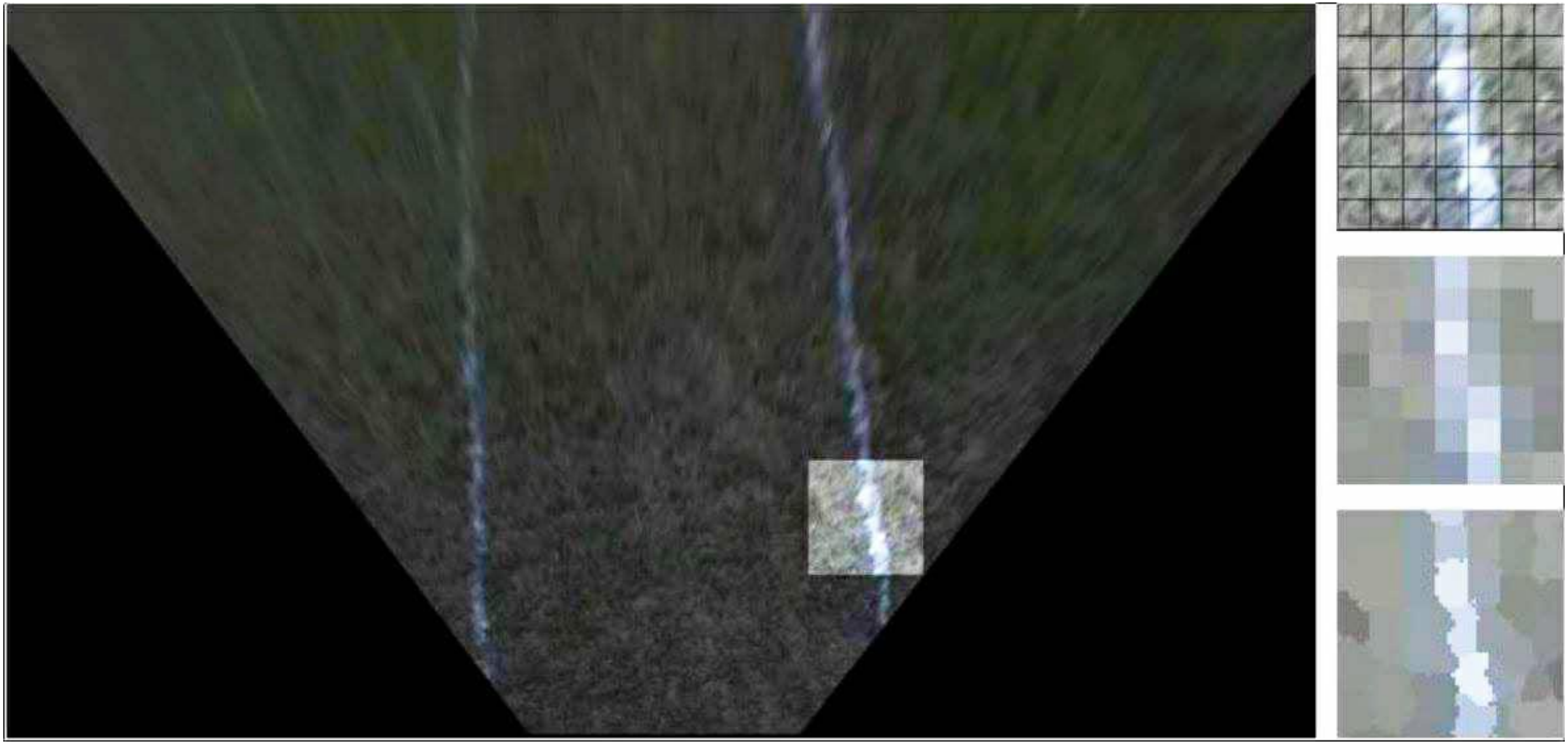
OpenCV implementation

$$\texttt{dst}(x, y) = \texttt{src}\left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\right)$$

Ref: https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html#warpperspective

Super-Pixel Segmentation

I I T K A N P U R

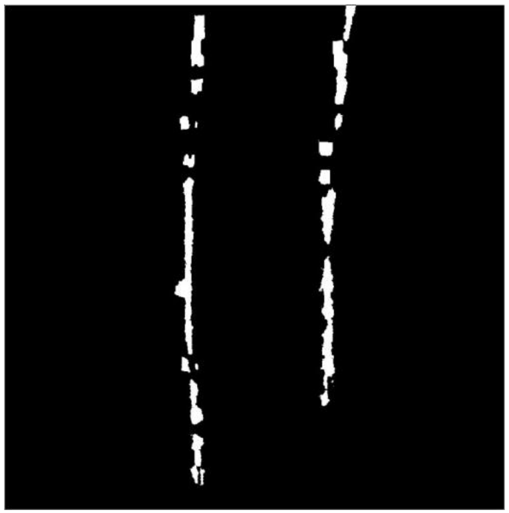


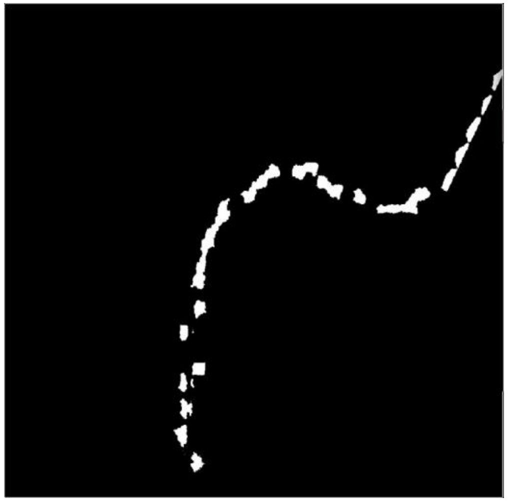
Reducing the dimensionality of data without loss of important information

Super-Pixel Segmentation

- Performing an unsupervied algorithm for image clustering
- Using an open Source and GPU acclerated implementation of SLIC (Simple Linear Iterative Clustering)
- SLIC divides the image into segments based on 5 dimensional distance
- Three dimensions are for RGB colors and 2 dimensions are for XY coordinates

Lane Detection

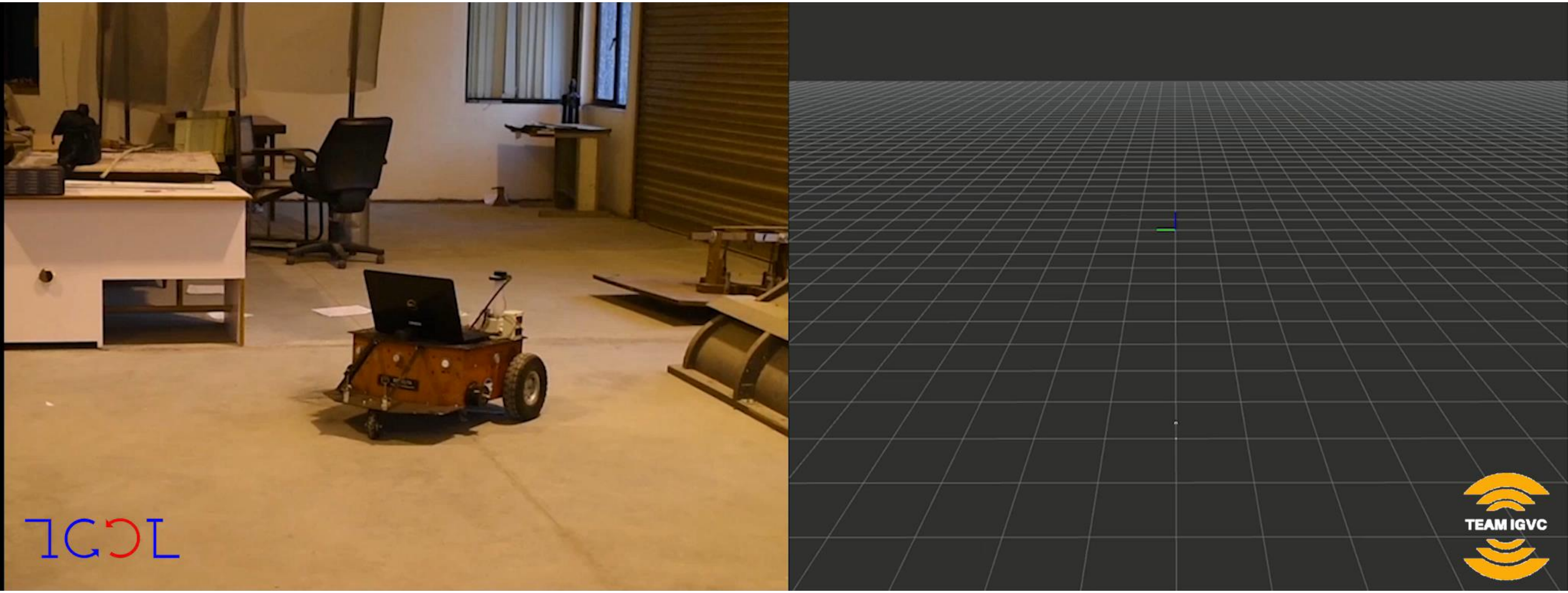




Obstacle Detection

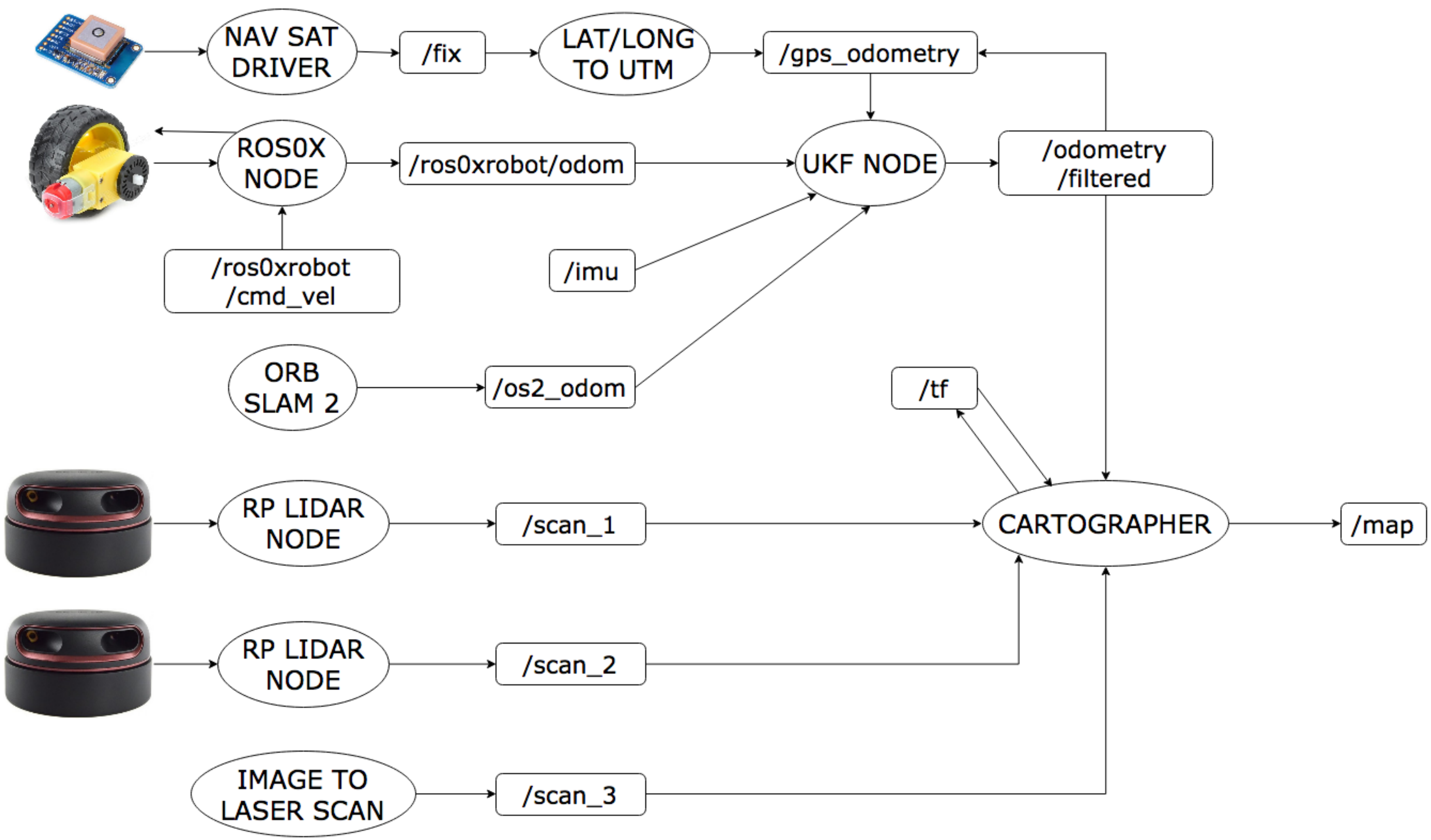
- Obstacle detection is done using the depth from stereo camera
- Alternatively, Lidar is used for avoidance

Indoor Navigation



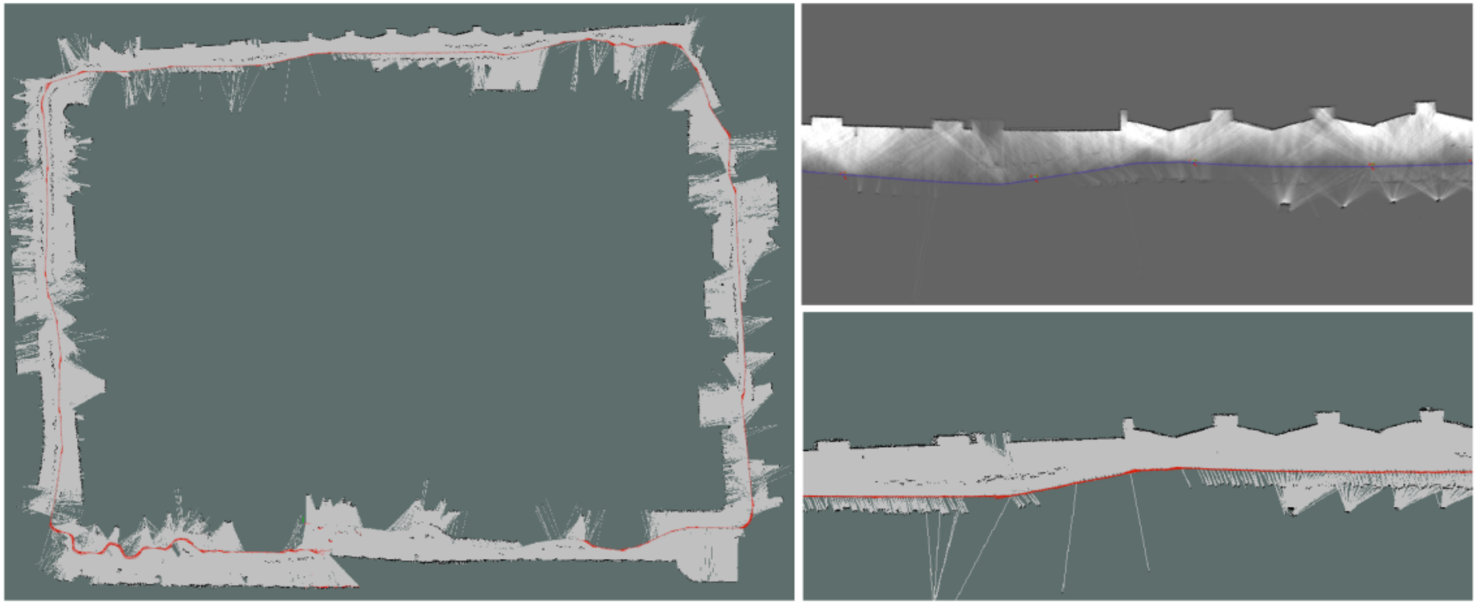
Simultaneous Localization and Mapping

I I T K A N P U R

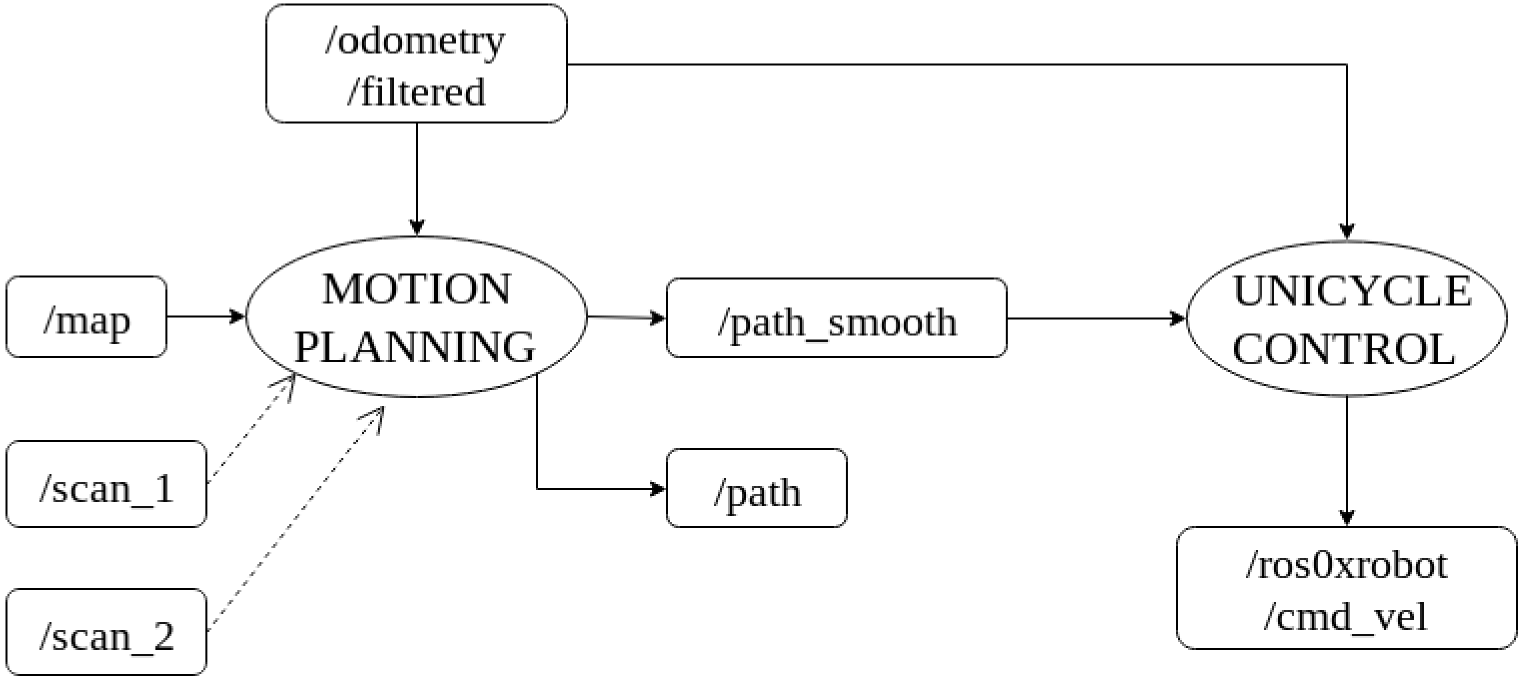


Simultaneous Localization and Mapping

- UKF
- Cartographer
- Odometry

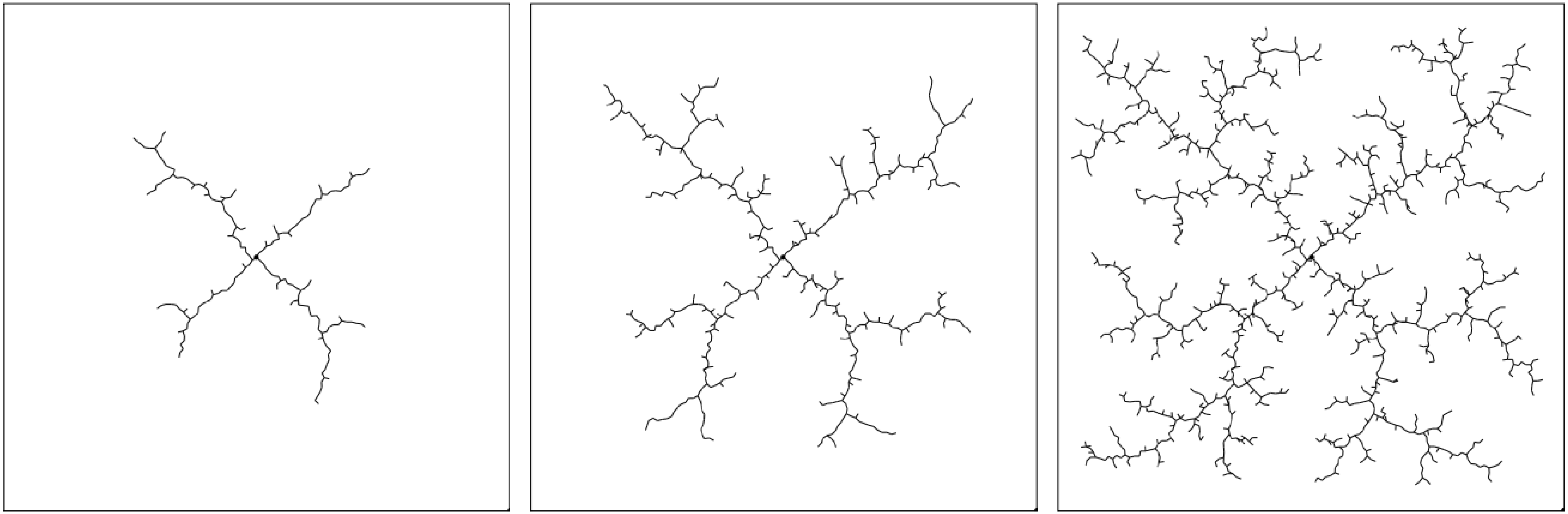


Motion Planning and Control



Sampling-based Algorithm

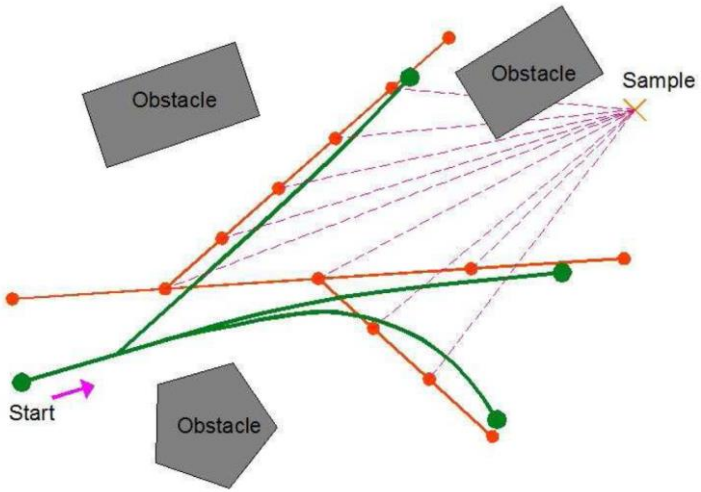
Path planning using modified RRT

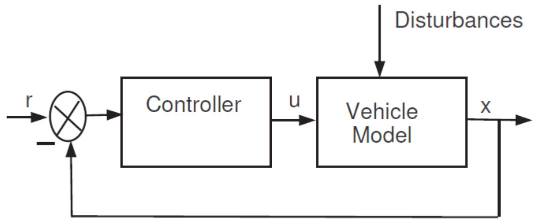


Chance constrained RRT (CC-RRT) Algorithm

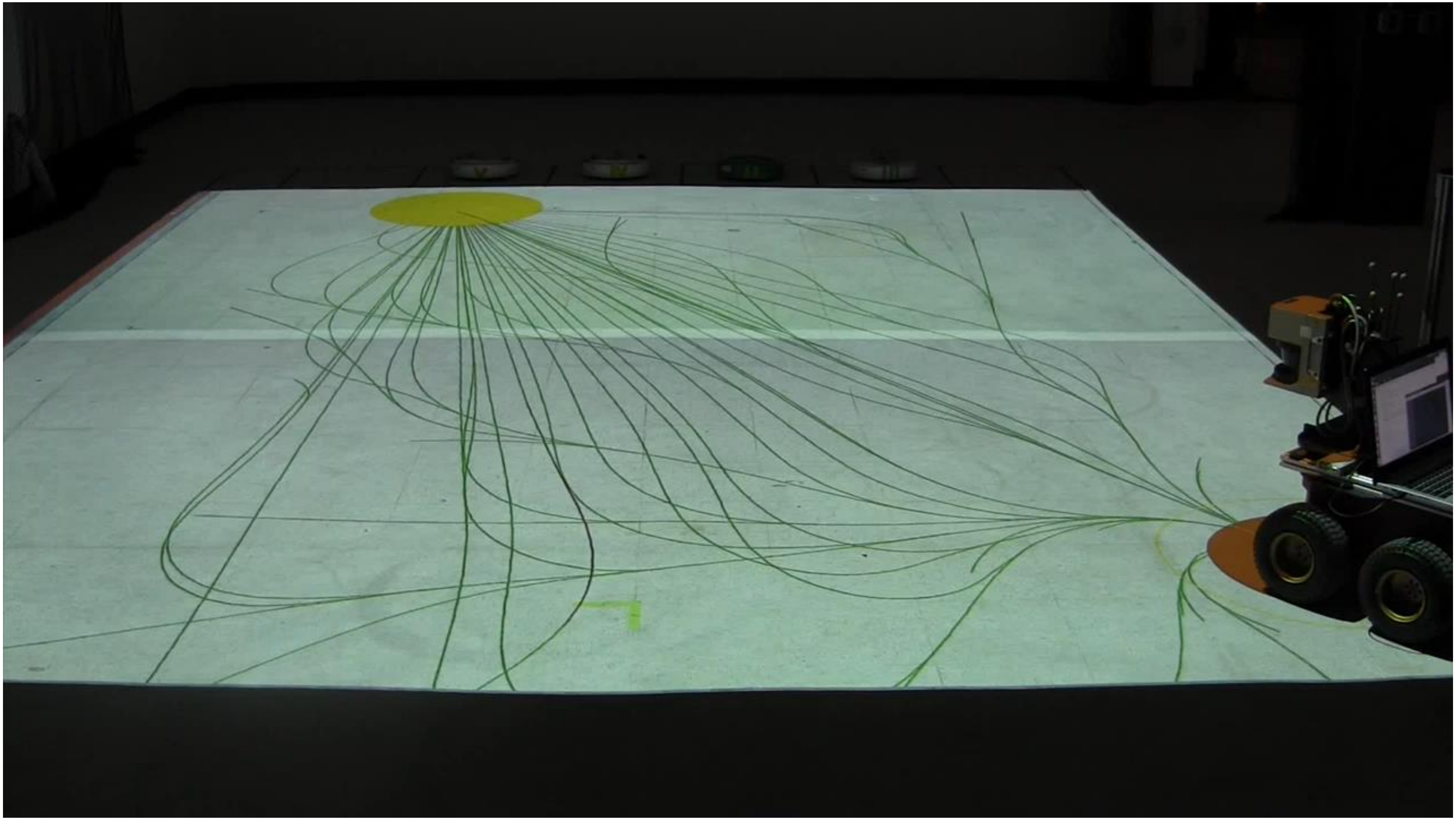
I I T K A N P U R

- Grow a tree of state distributions for a given time
  - sample reference path (similar to waypoint selection)
  - generate trajectory for the sampled path (use a control/guidance law to generate trajectory)
  - evaluate the feasibility of the generated trajectory (using chance constraint)
  - include the path in the existing tree if it is feasible





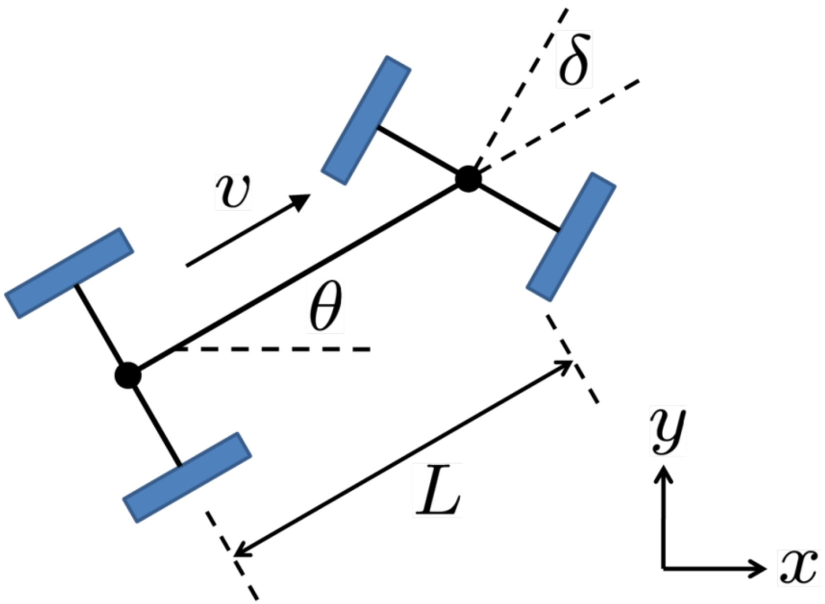
Robust:ACL@MIT



Kinematic Model: Ackerman Steering

$$\dot{x} = v\cos\theta,$$
$$\dot{y} = v\sin\theta,$$
$$\dot{\theta} = \frac{v}{L}\tan\delta.$$

$$\dot{\delta} = \frac{1}{T_\delta}(\delta_R - \delta),$$
$$\dot{v} = a,$$
$$\dot{a} = \frac{1}{T_a}(a_R - a),$$

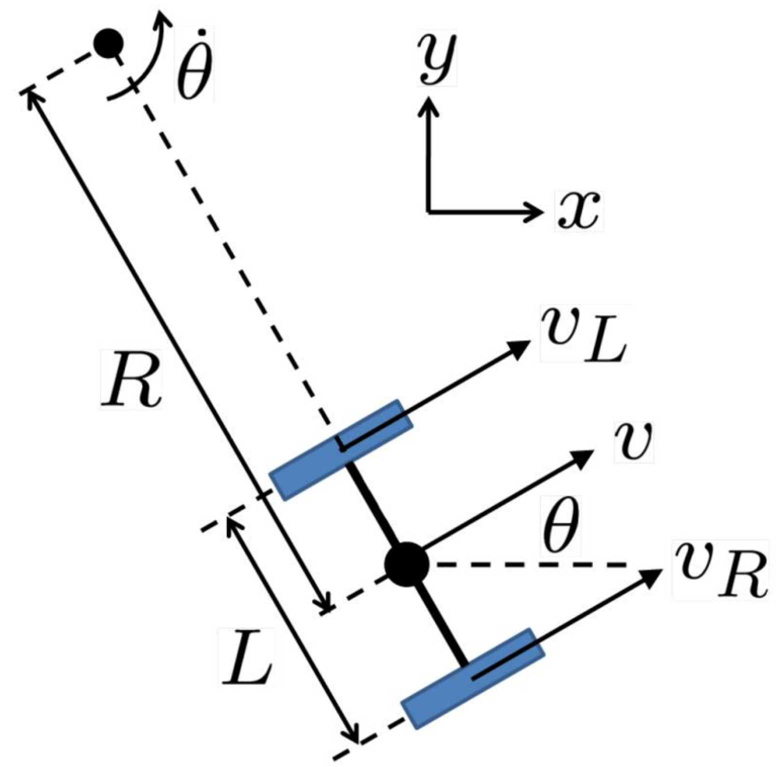


Explicit steering vehicle model

Skid Steering Model

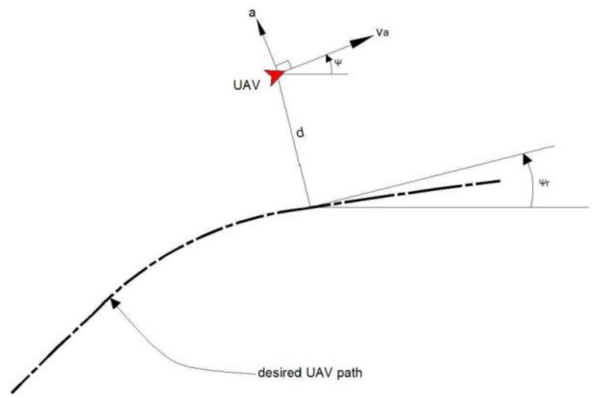
$$\begin{aligned} \dot{x} &= v\cos\theta, \\ \dot{y} &= v\sin\theta, \\ \dot{\theta} &= \frac{\Delta v}{L}. \end{aligned}$$

$$\begin{aligned} \dot{x} &= \frac{1}{2}v_R\cos\theta + \frac{1}{2}v_L\cos\theta, \\ \dot{y} &= \frac{1}{2}v_R\sin\theta + \frac{1}{2}v_L\sin\theta, \\ \dot{\theta} &= \frac{1}{L}v_R - \frac{1}{L}v_L. \end{aligned}$$



Path Following Problem

I I T K A N P U R



- Calculate lateral acceleration

$$\text{(i)} \lim_{t \to \infty} d = 0,$$
$$\text{(ii)} \lim_{t \to \infty} \tilde{\psi} = 0,$$
$$\text{(iii)} |u| \leq u_{max}.$$

Mathematical Formulation

I I T K A N P U R

kinematic model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v_a \cos\psi \\ v_a \sin\psi \\ \frac{g}{v_a}\tan\phi \\ -k(\phi - \phi^d) \end{bmatrix}$$

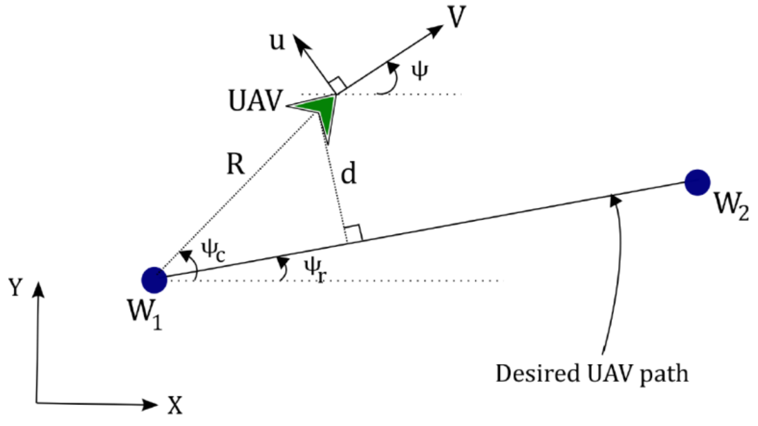
Path following law: pursuit and LOS components

$$\phi^d = \tan^{-1}\left(\frac{k_1(\psi_r - \psi) + k_2 d}{g}\right)$$

Error dynamics

$$\begin{aligned} \dot{d} &= v_a \sin(\psi_r - \psi) \\ \dot{\tilde{\psi}} &= -\frac{k_1}{v_a}(\psi_r - \psi) - \frac{k_2}{v_a}\, d \end{aligned}$$

Straight Line following



$$d = R\sin(\psi_c - \psi_r)$$
$$\tilde{\psi} = \psi - \psi_r$$

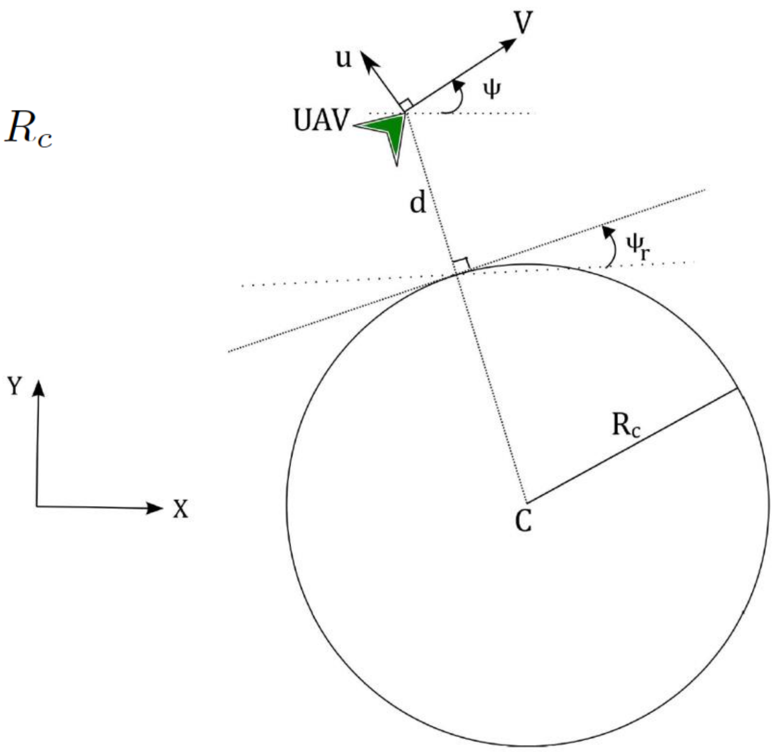
Circle following

I I T K A N P U R

$$d = \sqrt{(x-x_c)^2+(y-y_c)^2} - R_c$$

$$\tilde{\psi} = \psi - \psi_r$$

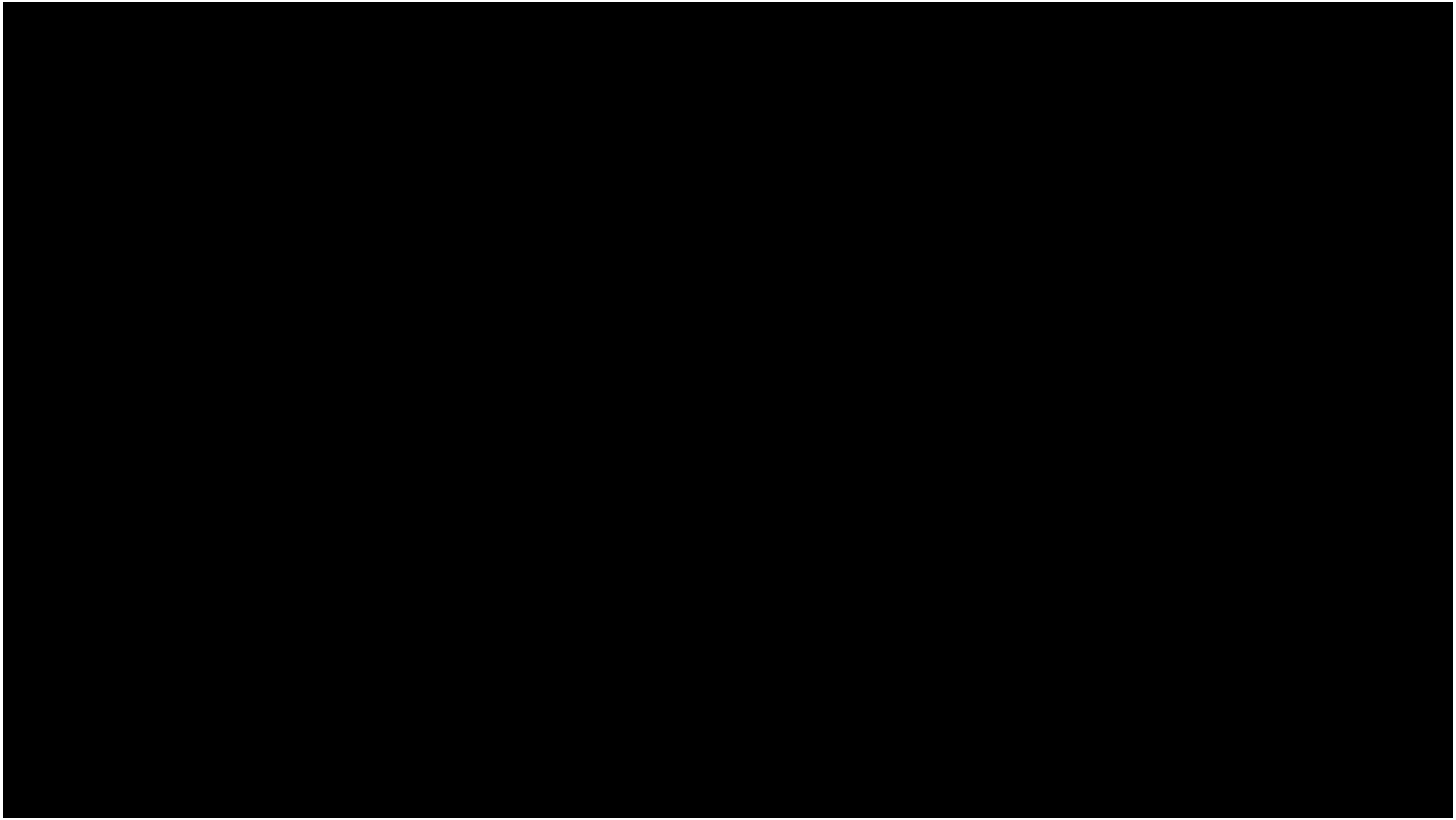
$$\psi_r = \tan^{-1}\frac{y_c-y}{x_c-x} \pm \frac{\pi}{2}$$



SLAM and Motion Planning

I I T K A N P U R

Vision based Autonomous Tracking and Landing



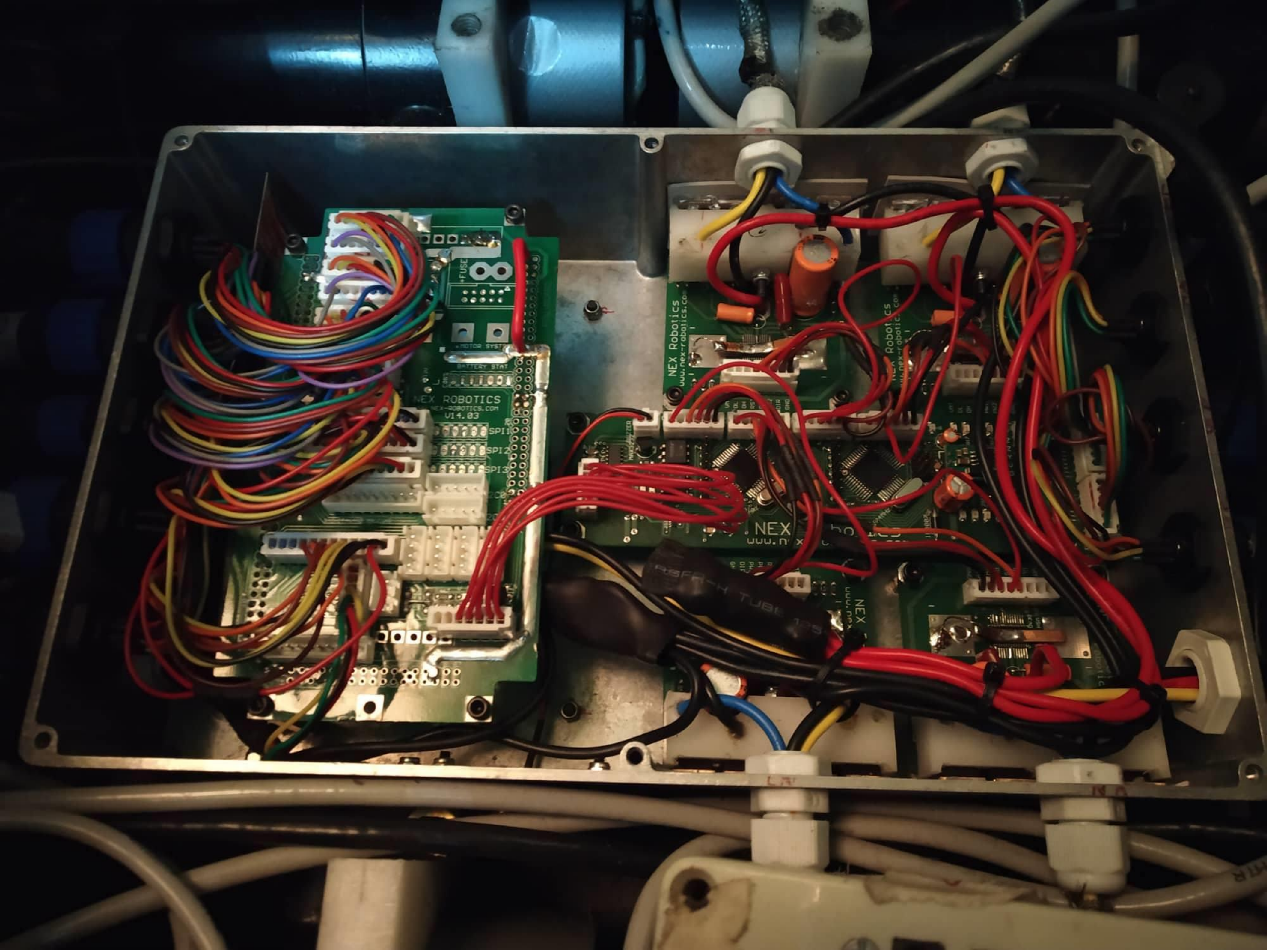
Acknowledgement

I I T K A N P U R

- **IGVC Team**
- **Harsh Sinha, Shubh Gupta, Swati Gupta, Deepak Gangwar**
- **Aalap Saha, Hemanth Bollamreddi, Abhishek Yadav, Vaibhav Agarwal, Vardhan Gupta**

Thanks


pixhawk
2
Aalap Shah
Harsh Sinha
Abhishek Yadav
Vardhan Gupta
Fifth Place is awarded to
IIT Kanpur
Daksh
Design Competition
26TH INTELLIGENT GROUND VEHICLE COMPETITION
ROCHESTER, MICHIGAN
JUNE 1-4, 2018
AUVSI
MOBIS
robonation
NDIA
Continental
NORTHROP GRUMMAN
molex
veoneer
OPENJAUS
ROUSH
FCA
ROBOTIC RESEARCH
MAGNA
MAGNA ELECTRONICS
GENERAL DYNAMICS
Land Systems
MathWorks
TARDEC


NEX ROBOTICS
NEX-ROBOTICS.COM
V14.03
SPI1
SPI2
SPI3
BATTERY STAT
NEX Robotics
www.nex-robotics.com