

Lecture 2

- Lecture 2: <https://www.youtube.com/watch?v=k4FQ-dZ0Lp8>
- Written by Milad Abdi with some AI assistance.

Automatic Emergency Braking (AEB) on the road

- **Relative Speed:**
 - Need the relative speed of our car and the one in front.
 - A sudden change in the lead vehicle's speed requires harder braking.
- **Effectiveness:**
 - Very effective and mandated as standard in many vehicles.
 - Time-to-collision-based braking will be implemented.
- **AEB in Today's Vehicles:**
 1. Detect objects.
 2. Find range, heading, pose, and velocity.
 3. Determine/isolate the critical object (may not be the closest).
 4. Check the trajectory for imminent collision.
- **Different information about detected objects:**
 - **Range:** Distance to the object.
 - **Heading:** Object orientation.
 - **Pose:** Object's position and orientation in space (includes range and heading).
 - **Velocity:** Speed and direction of movement.
- **Binary Classifier Approach:**
 - **False Positives:** Acceptable but may annoy users, leading them to disable the system. Could also cause crashes if the car behind reacts unexpectedly.
 - **False Negatives:** Extremely dangerous; people are killed. This is the most critical metric, and insurance heavily considers this rate.
 - **Goal:** Minimize false positives while avoiding false negatives entirely.

Range sensors for autonomous vehicles

- **Overview:**
 - Multiple sensors are needed for AVs to achieve **360° situational awareness**.
- **Sensor roles:**
 - **Long-range radar:** 200m+ lookahead for vehicle trajectory planning.
 - **LIDAR:** Captures distance (and can infer relative motion over time).
 - **Cameras:** Track objects for better context.

- **Ultrasound:** Blind-spot warnings and parking assistance.

RGB-D Camera

- **Demo Hardware:** Intel Realsense D435i
- **Working Principle:** Active infrared stereo
- **Advantages:** 3D point cloud (great for visual SLAM), works in any environment, mature SDK, calibrated IMU.
- **Disadvantages:** Limited range, can be noisy, affected by lighting conditions, requires building Linux kernel.

Stereo Camera

- **Demo Hardware:** ZED Camera
- **Working Principle:** Stereo/multiview geometry
- **Advantages:** Range, outdoor performance, scalability, calibrated IMU.
- **Disadvantages:** Low-texture surfaces reduce accuracy, baseline width determines range, high processing requirements.

Monocular Depth

- **Example Hardware:** Logitech Webcam & Monodepth Net
- **Working Principle:** Learns depth from stereo camera data (unsupervised) to map monocular input to depth.
- **Advantages:** No special hardware, failure modes orthogonal to other options.
- **Disadvantages:** Relatively poor accuracy (good only for near vs. far distinction).

Long-Range Radar

- **Example Hardware:** Continental (etc.)
- **Working Principle:** Emit RF energy, measure echo; Doppler shift can provide velocity.
- **Advantages:** Cheap, long range, orthogonal failure modes.
- **Disadvantages:** Poor spatial resolution and field of view; false positives on overhead signs.
- **Note:** Most AVs rely on radar.

Ultrasonic Proximity Sensor

- **Example Hardware:** VEX IQ Ultrasonic Distance Sensor
- **Working Principle:** Measures time-of-flight of high-frequency sound waves.
- **Advantages:** Accuracy, low cost, compact size.
- **Disadvantages:** Short range (~5 m), limited resolution, mostly for parking/blind-spot applications.

Planar 2D LIDAR

- **Demo Hardware:** Hokuyo 30LX
- **Working Principle:** Time-of-flight using laser.
- **Advantages:** Relatively low cost, simple data structure, high update rate, low processing requirements.
- **Disadvantages:** Primarily works in flat environments, harder to detect objects.

3D LIDAR

- **Demo Hardware:** Ouster OS-2
- **Working Principle:** Time-of-flight laser (different wavelengths depending on product).
- **Advantages:** Full 3D information, can get image-like information.
- **Disadvantages:** Costly, mechanical reliability issues, heavy point-cloud processing.

Solid-State LIDAR

- **Demo Hardware:** Velodyne Velarray
- **Working Principle:** Time-of-flight, steerable laser with solid-state components.
- **Advantages:** Compact, no moving parts, good range.
- **Disadvantages:** Limited field of view, availability, technical feasibility issues.

Working with laser scan data

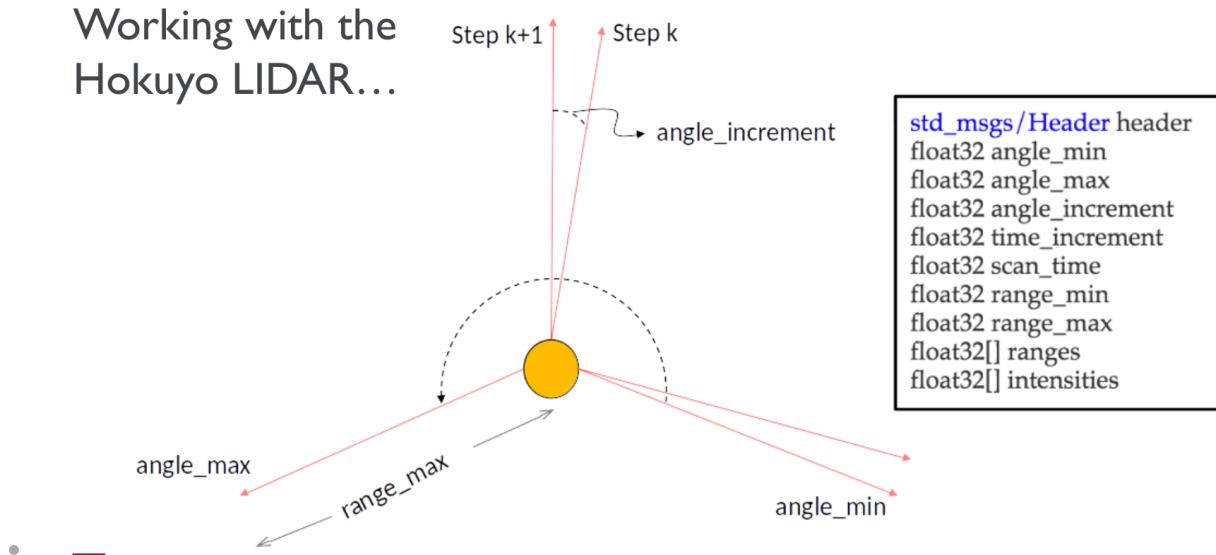
- **LIDAR Placement:**
 - Mounted on top of the car to avoid obstructions.
 - Effective range for this setup: ~10 meters.
- **Basic Principle:**
 - LIDAR shoots a laser beam at an object and measures the **time for the beam to reflect back**.
 - **Distance to obstacle:** Distance = $\frac{(\text{speed of light})(\text{time traveled})}{2}$.
 - Maximum range is limited by the laser energy.
- **Signal-to-Noise Considerations:**
 - Energy of reflected ray depends on:
 - Laser divergence
 - Bouncing/reflections
 - Humidity
 - Target reflectivity
 - Detector sensitivity
 - For indoor environments, LIDAR is generally very reliable.

- **Scan Parameters:**
 - Scan rate: 40 Hz (1 laser every 25 ms).
 - Each scan produces **1080 rays**, separated by **0.25°**.
 - Angular resolution: covers **270° horizontal field of view**, 40 times per second.

- **Visual:**

- Single scan from a planar laser range-finder:

Working with the Hokuyo LIDAR...



```
std_msgs/Header header
float32 angle_min
float32 angle_max
float32 angle_increment
float32 time_increment
float32 scan_time
float32 range_min
float32 range_max
float32[] ranges
float32[] intensities
```

- **Header header:** Timestamp is the acquisition time of the first ray in the scan.
- **float32 angle_min:** Start angle of the scan (rad).
- **float32 angle_max:** End angle of the scan (rad).
- **float32 angle_increment:** Angular distance between measurements (rad).
- **float32 time_increment:** Time between individual measurements (s).
- **float32 scan_time:** Time between scans (s).
- **float32 range_min:** Minimum valid range (m).
- **float32 range_max:** Maximum valid range (m).
- **float32[] ranges:** Distance measurements (m); discard values <<< range_min or >>> range_max.
- **float32[] intensities:** Intensity measurements (device-specific units).

- **Key Point:**

- The most important field is `ranges` : it contains distances to obstacles.
- Array `A[1080]` : `A[i]` = distance measurement of the **i-th step**.
- Measurements beyond min/max range (e.g., `inf`) should be discarded. `inf` simply indicates “beyond range.”

Measuring safety

- Safety should be considered **on a spectrum**, not as a binary safe/unsafe decision

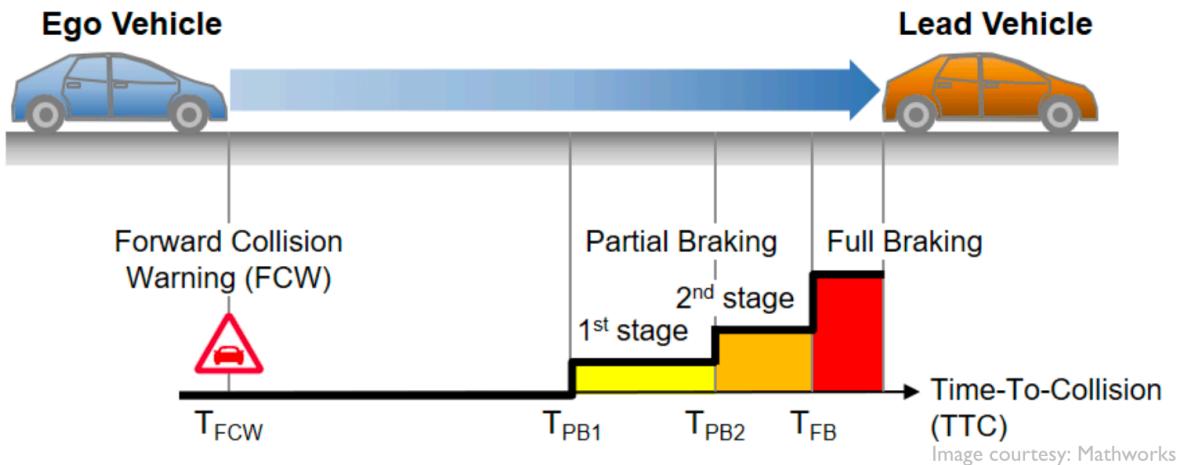
Euclidean Distance

- **Pros:**
 - Easy to compute.
 - Sensors can measure it accurately with well-understood uncertainty.
- **Cons:**
 - Classifies safe situations as “near unsafe.”
 - Can cause unnecessary braking (e.g., already stopped at a T-junction, tailgater behind).

Therefore, we need to consider both **relative distance** and the **rate of change of relative distance**.

Time-to-Collision (TTC)

- **Time-to-Collision (TTC):** TTC is the time it would take for the ego vehicle and an object to collide, assuming both maintain their current heading and velocity.
- **Visualization:**



- **Definition:** $TTC_i(t) = \frac{r_i(t)}{[-\dot{r}_i(t)]_+}$, where $[x]_+ = \max(x, 0)$.
- **Range:** $r_i(t)$ is the distance between the vehicle and object i .
- **Range-rate:** $\dot{r}_i(t)$ is the time derivative of the range (range-rate).
- **Property:** $TTC_i(t) \rightarrow \infty$ when $\dot{r}_i(t) > 0$ (object moving away).
- **Range-rate computation:** Obtained by projecting the relative velocity onto the vector between the vehicle and the object's pose.
- **Static obstacle case:** $\dot{r}_i = v_x \cos(\theta_i)$, where v_x is forward speed in the vehicle's reference frame and θ_i is the laser beam angle.

Planar LIDAR Application

- Using a ROS interface, we can control the field of view and capture distance estimates from a planar LIDAR.

- **Planar LIDAR application:** Compute TTC_i for each laser beam.
- **Braking threshold:** Define an acceptable braking threshold T .
- **Decision rule:** Apply brakes if $\text{TTC}_i \leq T$.

Safety Lab Overview

- Implement a Time-to-Collision (TTC) based safety node in the simulator.
- Tune the implementation to reduce false positives and avoid false negatives.
- Use keyboard control to intentionally crash the car into a wall; AEB should engage.
- Demonstrate a crash-free lap with AEB activations.