Augmenting Physical Safety by Modifying Vehicular Networking Communication Methods

Eric Newberry, Hsun-Wei Cho, Richard Higgins
University of Michigan

π March 14, 2019 π

Key Components/Challenges

1. Understanding physical safety from real network traces
    a. Incident and situation analysis
2. Augmenting physical safety
    a. How can we measure this?
    b. What choices could improve it?
3. Modifying vehicular networking
    a. Current Basic Safety Messages (BSMs) communicate every 100ms
    b. BSMs include basic information about the vehicle
    c. Certain situations may benefit from faster or different inter-vehicle communication
4. Combining simulation with reinforcement learning
    a. Allows experimentation with a multi-vehicle interactive environment
    b. Understanding how differently constrained network situations can impact vehicle safety

Related Work: Physical Safety

1. How do you evaluate physical safety (instead of a derived measure)?
    a. A paper [1] by Intel/Mobieye formulates a mathematical model for driving behavior and safety
    b. Derives the model from 5 basic driving principles
    c. Longitudinal and lateral safe distances
    d. Longitudinal and lateral evasive manoeuvres
    e. Also considers multiple lanes, winding roads, priorities of routes
    f. Each car is an RL agents. These mathematical results are used by agents to make a driving decision.
    g. Mathematically, each agent tries to maximize a *Q function*, which measures the physical safety
    h. The Q function is a summation of instantaneous rewards, which may depend on:
        i. Relative position/velocities/acceleration to other cars
        ii. The difference of the current speed and the desired speed
        iii. Route selection, level of comfort …

[1] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2017. On a Formal Model of Safe and Scalable Self-driving Cars. arXiv:1708.06374. Retrieved from https://arxiv.org/abs/1708.06374

Related Work: DSRC, BSMs, and CAMs

- Connected vehicles will communicate with each other and infrastructure
- Dedicated Short-Range Communications (DSRC)
  - WiFi-like (802.11p) protocol
  - Theoretical maximum range of 1km
  - Entirely ad-hoc
- Basic Safety Messages (BSMs)
  - American protocol for communicating (standardized by SAE [1])
  - Includes info such as position, heading, speed, acceleration, vehicle size, path
  - Broadcasted on a static interval (100ms) -- is this frequency enough or unnecessarily high?
- Cooperative Awareness Messages (CAMs)
  - Like BSMs, but European (standardized by ETSI [2])
  - Dynamic interval (between 100ms and 1s) -- much more complex than BSMs...

[1] SAE International, *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, March 2016.

[2] European Telecommunications Standards Institute, *Intelligent Transportation Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, January 2019.

Related Work: CAMs

- Two alternate conditions to broadcast a CAM
- Time since default interval and congestion-set minimum both exceeded:
  - Default interval defaults to 1s, but can be adjusted based upon “environmental conditions” [1]
  - Default interval must still be in range [100ms, 1s]
- Change in vehicle status:
  - When heading changes by >4°
  - When distance traveled since last CAM >4m
  - When speed since last CAM changes by >0.5m/s
  - But only if time since last CAM tx >= congestion-set minimum
    - This value must be in range [100ms, default interval]

[1] European Telecommunications Standards Institute, *Intelligent Transportation Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, January 2019.

Step 1: Exploratory Data Analysis

1. Safety Pilot Dataset
    a. Consists of traces from vehicles recorded driving around Ann Arbor
    b. Includes 100s of Gb of data, around a Tb
    c. Includes vehicles and their location, headings, speed, communications

Attribute
Attribute Label: MsgCount (column F)
Attribute Definition: This field contains a message ID that gets incremented by one with each BSM.
Attribute Domain Values: Integer

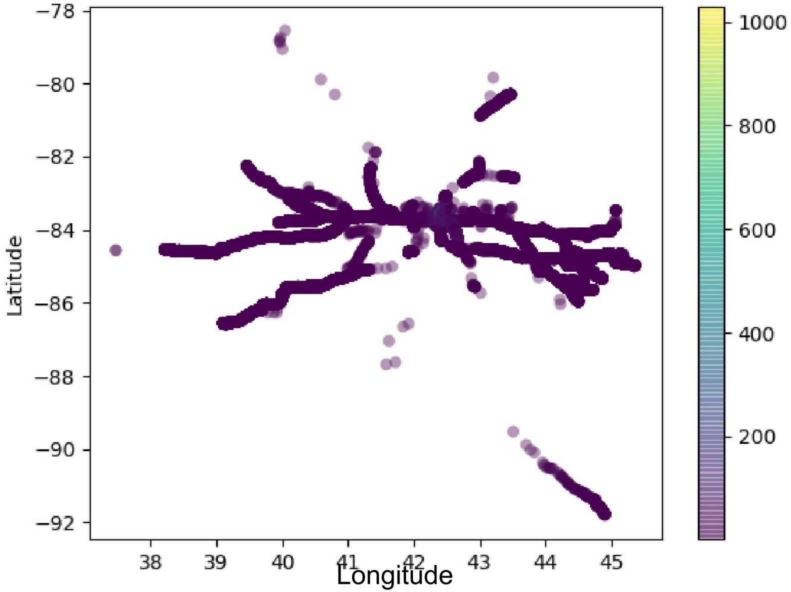
Attribute
Attribute Label: DSecond (column G)
Attribute Definition: This field contains the time in deciseconds since ignition started.
Attribute Domain Values: Integer

Attribute
Attribute Label: Latitude (column H)
Attribute Definition: This field contains the current latitude, in degrees, of the vehicle.
Attribute Domain Values: Float

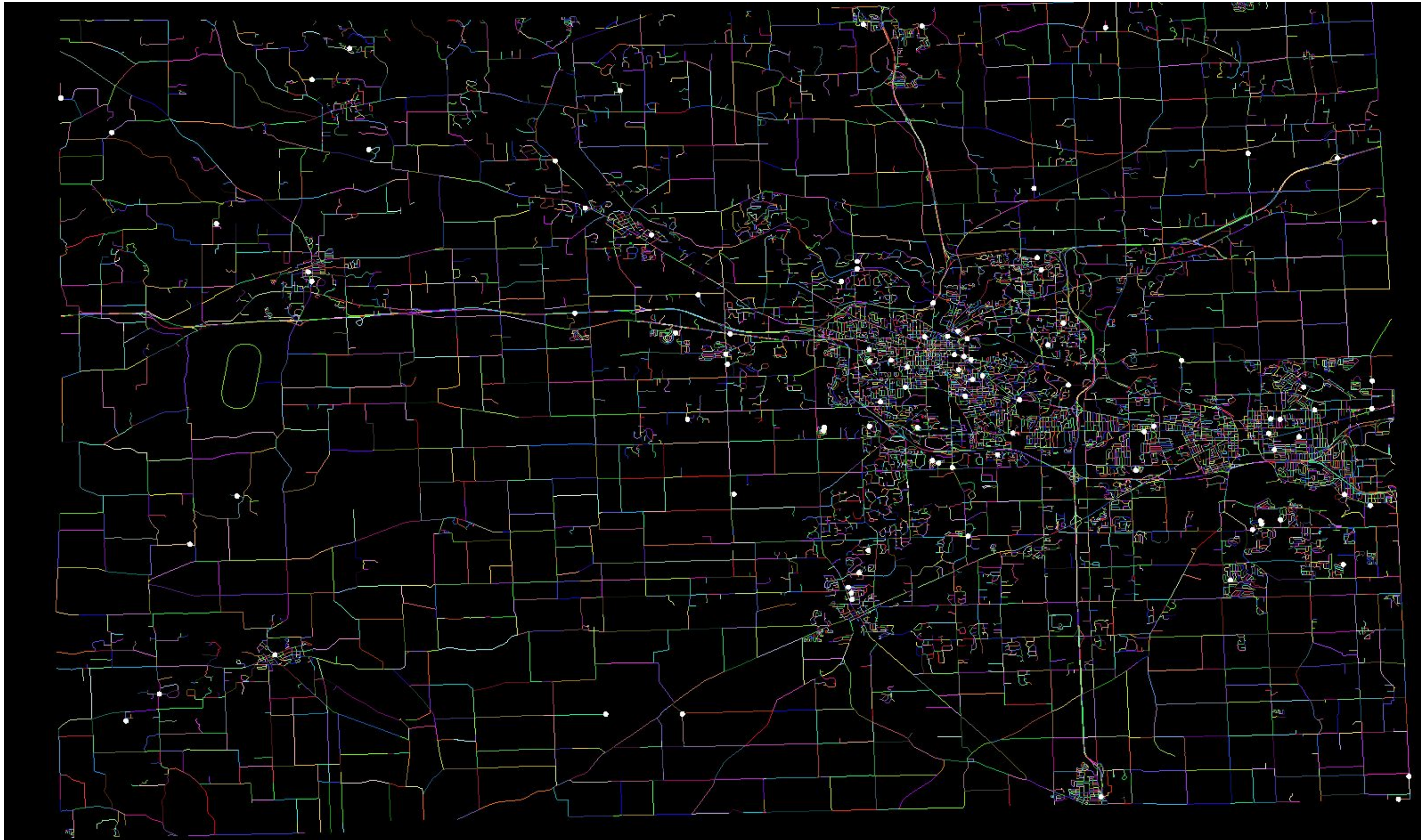
Attribute
Attribute Label: Longitude (column I)
Attribute Definition: This field contains the current longitude, in degrees, of the vehicle.
Attribute Domain Values: Float

Attribute
Attribute Label: Elevation (column J)
Attribute Definition: This field contains the current elevation, in meters, of vehicle according to GPS.
Attribute Domain Values: Float

Attribute
Attribute Label: Speed (column K)
Attribute Definition: This field contains the vehicle speed in m/sec.
Attribute Domain Values: Real



Results: Exploratory Data Analysis



Step 2: Vehicular Networking Simulation

1. Each RL agent drives a car on a real street map
2. Each agent makes accelerate/decelerate decisions to maximize the Q function
3. V2V Communication: Each agent may not have all states of other cars, or states at the finest time granularity
4. Different communication pattern affects the quality of the decisions made
5. Evaluate the overall safety of the system

Will handle details like transmission range of BSMs

Step 2: Vehicular Networking Simulation

- Selecting a Q function:
- The paper derives the safety distances each car should maintain:

  Q = The time percentage where safety distances are violated

- Alternatively, we can use relative position/velocities/acceleration information:

  For example, the kinetic energy in a collision $\propto$ (relative velocity)^2

  Therefore, we can set Q = $\displaystyle\sum_{\text{cars expected to collide in near future}} (\text{relative velocity})^2$

Step 2: Vehicular Networking Simulation

- The simulation program uses the real map of Ann Arbor as the input
- Parse the input shape file/SVG into a *roads* structure
- Analyze the endpoints of roads to find intersections
- Each car is driven along the piecewise linear segments of a road. When reaching the end:
    - Dead-end: Turn around
    - Only one connecting road: Drive to this road
    - Multiple connecting road: Stop at the intersection and inform the agent
- The simulator is written is Python and renders the map and cars by OpenGL
- For the scope of this project, we assume each road is single-lane and bidirectional

Current: Vehicular Networking Simulation

1. Currently our state contains:
   a. Available directions to turn
   b. Location
   c. Position/velocity/acceleration
2. For evaluation we also measure the minimum distance to another vehicle
3. We need to next include the network information

Step 3: Next Experiments

1. Varying Sensor Fidelity
    a. Sensor Rich
    b. Sensor Poor
    c. Autonomous
2. Varying Communication Frequency
    a. 10ms
    b. 100ms,
    c. 500ms,
    d. 1000ms
    e. 2000ms
    f. 5000ms