Experience-Based Iterative Path Optimization for Autonomous Tramming using Inertial Sensors

Jeremy Roy and Joshua Marshall

Motivation

- Autonomous tramming research for **mobile robots** has focused on reducing path following errors
- Little research has questioned the **optimality of the original path**, with notable exceptions [1,2]
- **Environmental factors** may cause the terrain to change over time, e.g. potholes, wet soil, ice

Problem Setup

- A **dynamic vehicle** is instructed to traverse a terrain a large number of times given a prior path
- The **terrain is susceptible to changes**
- The vehicle is equipped with an **inertial measurement unit (IMU)** that can measure the vehicle's angular velocities and linear accelerations
- Using the IMU, the vehicle can **assign a cost to each position** it has traversed
- On each subsequent traversal, the vehicle can **alter the path** in an attempt to find a path with a **lower overall cost**
- The path resulting from these alterations must conform to the **vehicle's turning constraints**

Assumptions

- The desirability of the terrain (cost) is determined by the **magnitude of the vehicle's vibrations**
- The vehicle is constrained by **radius of curvature** (ROC) only
- The vehicle travels at a **constant speed**
- The vehicle is capable of **perfect localization**
- **Timing** considerations are **ignored**

Selected References

[1] M Mazuran, C Sprunk, W Burgard, and G D Tipaldi. LexTOR: Lexicographic teach optimize and repeat based on user preferences. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 2780–2786, May 2015.

[2] F Tsang, R A Macdonald, and S L Smith. Learning Motion Planning Policies in Uncertain Environments through Repeated Task Executions. In 2019 IEEE International Conference on Robotics and Automation (ICRA), pages 8–14, May 2019.

Mathematical Formulation

Definitions and Notation:

Terrain	$T \subset \mathbb{R}^2$
Objective function over terrain	$f: T \to \mathbb{R}$
Path	$P = [p_0\ p_1\ \dots\ p_m]^{\mathrm{T}}$, $p_i = (x_i, y_i) \in T$ $\lvert\Delta x\rvert, \lvert\Delta y\rvert < \epsilon$ for some $\epsilon \in \mathbb{R}$
Prior (original) path	P_0
Lines normal to prior path	$N = [n_0\ n_1\ \dots\ n_m]^{\mathrm{T}}$ $n_i \perp P_0$ at p_i
Objective function along normal	$f_i(p_i) = f(p_i)$ $(p_i) \in n_i$
Specified min ROC	$R_{\min}$
Actual min ROC of path	$r_{\min}$

Problem Formulation:

$$\begin{aligned} \text{minimize} \quad & f_i(x, y) \quad i \in \{0, 1, \dots, m\} \\ \text{s.t.} \quad & r_{\min} \geq R_{\min} \end{aligned}$$

Method

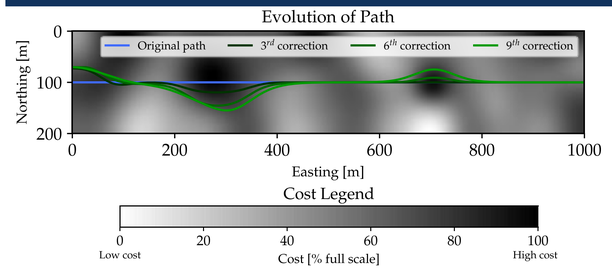
Steps:

1. Drive once along P_0, assign cost to each path point p_i
2. Offset path to one side along path normals n_i
3. Drive along new path, assign cost to each path point p_i
4. Calculate finite differences along n_i
5. Adjust path by μ (learning rate) in direction of gradient
6. Smooth path using a weak gaussian filter $h(i)$ until $R_{\min}$ is met
7. Repeat 3 – 7

Algorithm:

for k in iterations **do**
  for p_i in P_k **do**
    d_i = direction of gradient at p_i along n_i
    $P = P + \mu \cdot d_i \cdot \hat{n}_i$
  while $r_{\min} < R_{\min}$ **do**
    $P = P * h(i)$

Simulation



Description

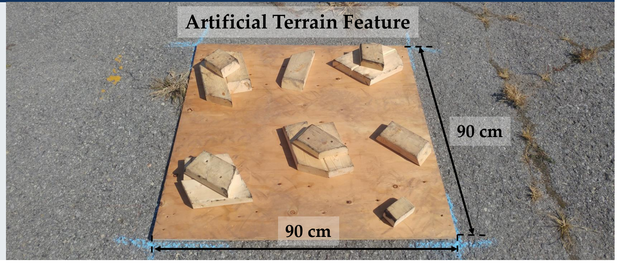
- **New feature** introduced after 5th driving iteration

Observations

- Algorithm can find a **path around** new high-cost **environmental features**
- Path's overall **cost generally decreases** as vehicle gains experience

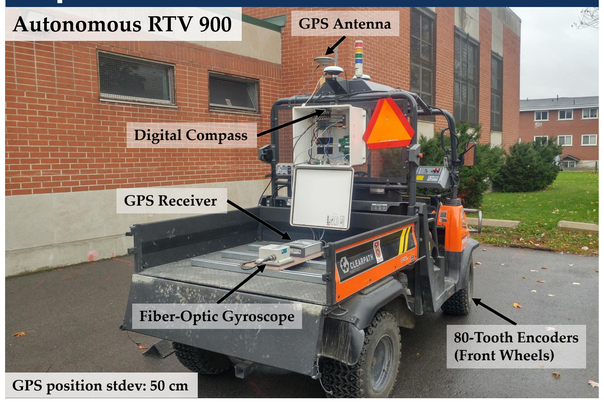
Description of Experiment

1. Place an **artificial terrain feature** on a paved surface
2. Record a path while driving over this feature
3. Drive the vehicle to the start of the path
4. Instruct the vehicle to drive along the path
5. Process the IMU data
6. Compute a "better" path
7. Repeat steps 3-6 until the path converges

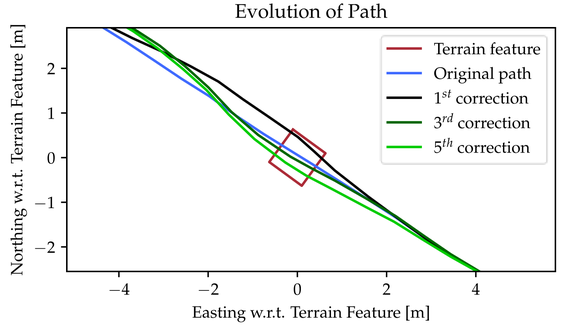


For more information: jeremy.roy@queensu.ca; joshua.marshall@queensu.ca; https://offroad.engineering.queensu.ca

Experimental Platform



Preliminary Field Results



Analysis

- The preliminary results show that the vehicle **doesn't learn to circumnavigate** the terrain feature
- Likely causes are:
  - **Poor localization**: standard deviation of position estimate is half the width of the terrain feature
  - **Over-simplified optimization method**: gradient descent

Envisioned Improvements

- Improve localization solution: use **Real-Time Kinematic (RTK)** differential GNSS for position stdev of 2 cm
- Explore **surrogate optimization** methods