Autonomous Navigation of UGV Based on LiDAR and AHRS

John Liu

Advisor:

Professor Ying-Jeng Wu

Measurement Laboratory, National Yunlin University of Science & Technology

March 12, 2014

1 Path Planning and Control Rules

Path Planning

Here I'm going to put a picture!

Flow Chart of Navigation Algorithm

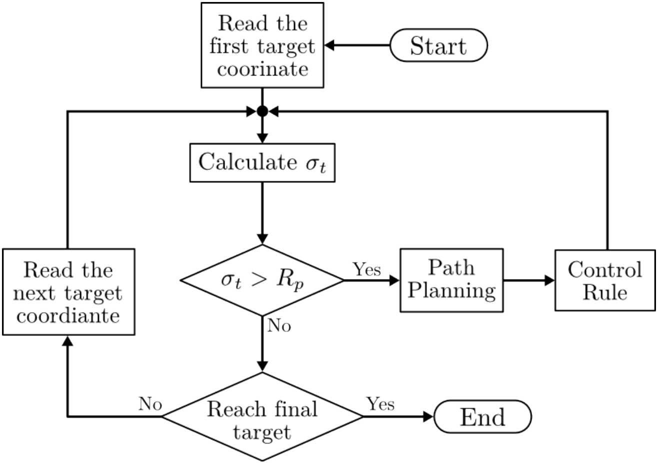
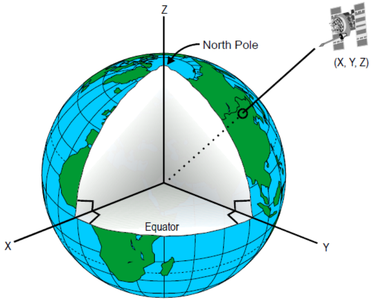


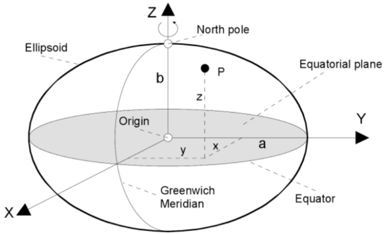
Table of Contents

Geographic Coordinate System

Geographic coordinate system is a reference system used to describe a position on the earth. There are two kinds of such system: *ECI* and *ECEF*.



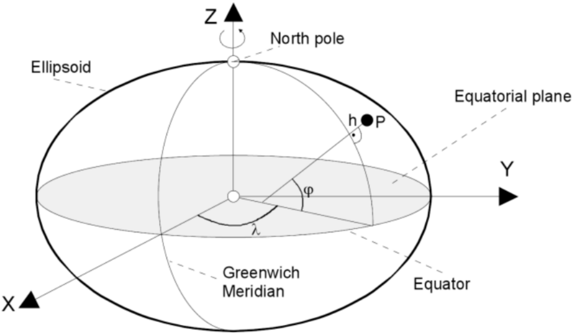
ECI



ECEF

ECEF Ellipsoidal Coordinates

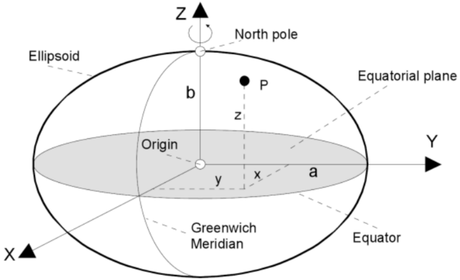
Ellipsoidal coordinates is the most common coordinate system in describing a position on earth, which defined by *Latitude* ϕ, *Longitude* λ and *Altitude* h.



Datum

Different definition of ellipsoid will also change its coordinate system, and the ellipsoid used to define the earth is called a *datum*.

- NAD27
- NAD83
- *WGS84*
- ...



WGS84 Datum

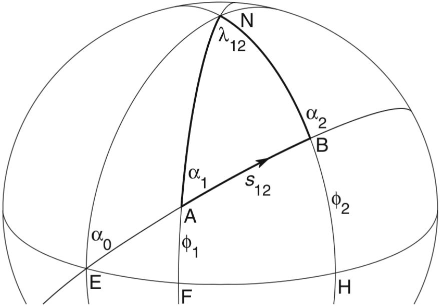
WGS84 (World Geodetic System 1984) is the standard ellipsoidal coordinate system used by MTi-G position sensor and most of the GPS.

a	$6378137m$
b	$6356752.3142m$
f	$= (a-b)/a = 1/298.257223563$

Geodesic

The shortest path between two points on the earth, customarily treated as an ellipsoid of revolution, is called a *geodesic*. Two geodesic problems are usually considered:

1. Direct
2. Inverse

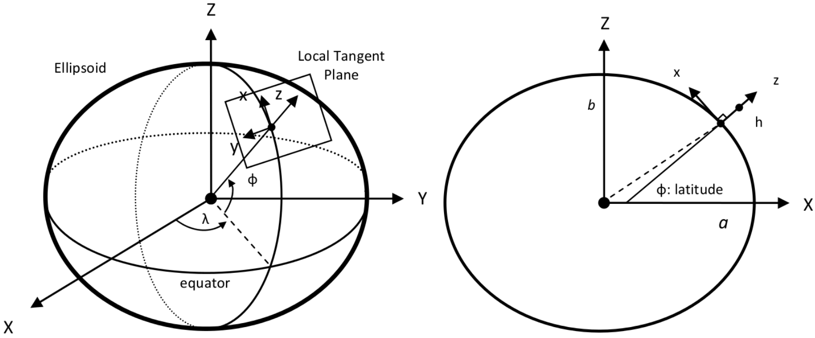


Inverse Problem

The relative distance and direction between two location is required for autonomous navigation, therefore inverse problem in geodesic is considered.

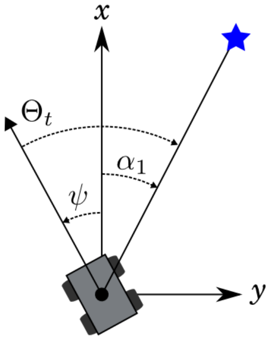
Local Tangent Plane

Local tangent plane is the reference system used by AHRS.



Target Angle

By the definition of azimuth α_1 and yaw ψ, the target angle Θ_t relative to robot could be determined:

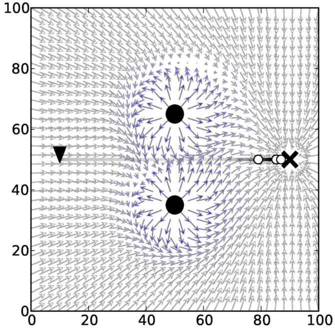


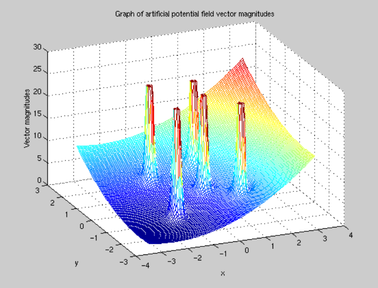
$$\Theta_t = -\alpha_1 - \psi$$

Table of Contents

Artificial Potential Field

Artificial Potential Field creates a virtual force field which attracts the robot toward the target, and retracts it away from the obstacle.



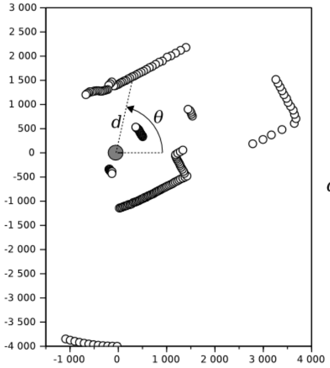


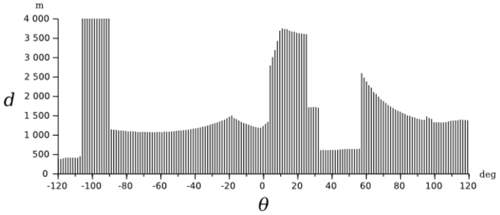
Artificial Potential Field

- Advantages:
  - Global path planning
  - Efficient Calculation
  - Easily adapt to the data acquired by LiDAR
- Disadvantages:
  - Ignore the kinematic and dynamic constraints
  - Ignore robot's geometry

Vector Field Histogram (VFH)

VFH generates a polar histogram of the environment around the robot, identifies wide-enough spaces and calculates corresponding steering direction.





Vector Field Histogram (VFH)

A cost function G is then applied to every candidate directions, and the direction which generates the smallest value is then selected:

$$G = u_1 \cdot \alpha + u_2 \cdot \beta + u_3 \cdot \gamma$$

where

$\alpha =$ difference between target and candidate direction

$\beta =$ difference between current direction and candidate direction

$\gamma =$ difference between the previously selected direction and candidate di

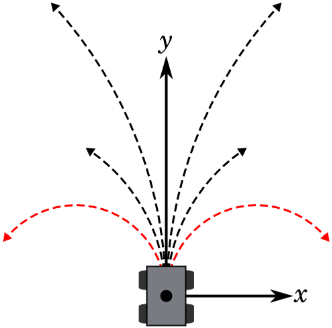
u_1, u_2 and u_3 are weighting constants

Vector Field Histogram (VFH)

- Advantages:
  - Easily adapt to the data acquired by LiDAR
  - Efficient Calculation
  - Adjustable characteristic
- Disadvantages:
  - Ignore the kinematic and dynamic constraints
  - Ignore robot's geometry
  - **Direction depends on free-spaces**

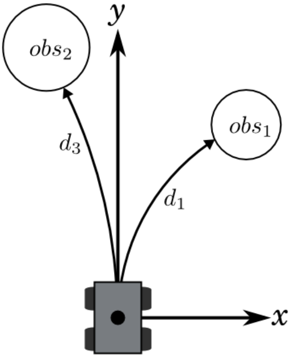
Curvature Velocity Method (CVM)

CVM takes robot's kinematic constraints into account, assumes it only travels along circular trajectories with curvature $c = \omega/\nu$, which uses *velocity space* for path planning.



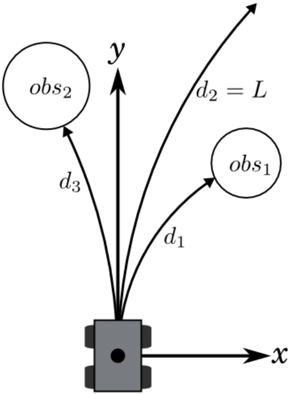
Curvature Velocity Method (CVM)

In order to transform obstacles into velocity space, a distance function $D(c, obs)$ calculates the distance d before collides with obstacle obs, along the curvature c.



Curvature Velocity Method (CVM)

Since the sensor has detection distance, CVM sets a limit L according to its limitation. Therefore, the distance function D becomes $D_{limit} = min(L, D(c, OBS))$.



Curvature Velocity Method (CVM)

The final decision of new ω and ν is made by an object function, which resembles the cost function of previous method:

$$f(\omega,\nu) = u_1 \cdot speed(\nu) + u_2 \cdot dist(\omega,\nu) + u_3 \cdot head(\omega)$$

where

$$\begin{aligned} speed(\nu) &= \nu/\nu_{max} \\ dist(\omega,\nu) &= D_{limit}(\frac{\omega}{v}, OBS)/L \\ head(\omega) &= 1 - |\theta_{target} - \omega \cdot T_c|/\pi \end{aligned}$$

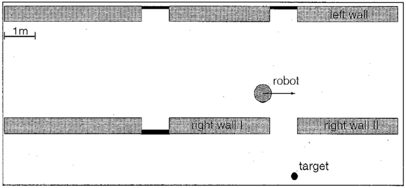
The velocities which generate the largest value will be chosen!

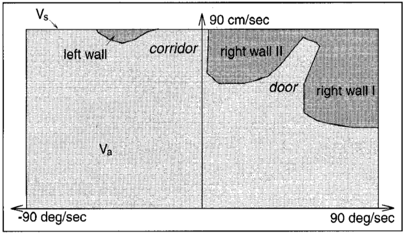
Curvature Velocity Method (CVM)

- Advantages:
  - Kinematic and dynamic constraints
  - Robot's geometry constraint
  - Adjustable characteristic
- Disadvantages:
  - Simplified circular obstacle
  - **Velocity sensors are required**

Dynamic Window Approach (DW)

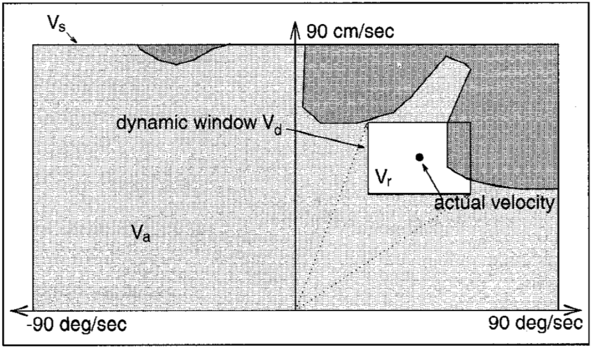
DW also assumes the robot only travelled in circular path with rotational velocity ω and translational velocity ν. The sensed environment is then transformed into **velocity space**.





Dynamic Window Approach (DW)

In velocity space, a *dynamic window* is constructed according to its dynamic constraints and current velocities. Again, an object function is used to choose the optimized velocities.



Dynamic Window Approach (DW)

- Advantages:
  - Kinematic and dynamic constraints
  - Robot's geometry constraint
  - Adjustable characteristic
- Disadvantages:
  - Complexity
  - **Velocity sensors are required**

Table of Contents

Vector Field Histogram Plus (VFH^+) - Introduction

VFH^+ algorithm is an enhanced version of original VFH which offers several improvements:

1. Kinematic constraints
2. Robot's geometry constraints
3. **Direction no longer depends on spaces**

VFH$^+$ - Four-Stage Process

The VFH$^+$ employs a four-stage data reduction process in order to compute the new direction of motion:

1. Primary Polar Histogram
2. Binary Polar Histogram
3. Masked Polar Histogram
4. Selection of Steering Direcion

VFH$^+$ - with Laser Range Finder

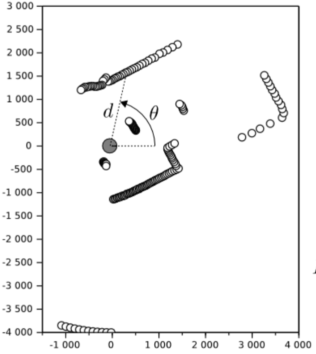
However, some modification is required in order to implement VFH$^+$ with laser range finder, therefore the process become:

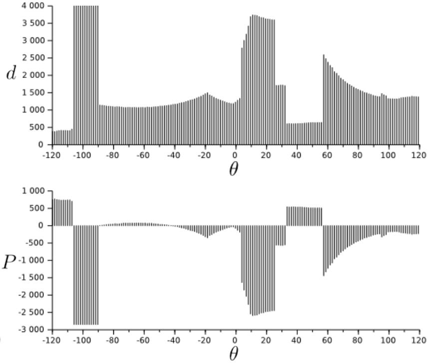
1. Primary Polar Histogram
2. Identifying Free Spaces
3. Blocked Directions
4. Selection of Steering Direction

1: Primary Polar Histogram

A polar histogram P_i of corresponding measured distance and angle d_i can be generated with following formula:

$$P_i = a + b \cdot d_i$$

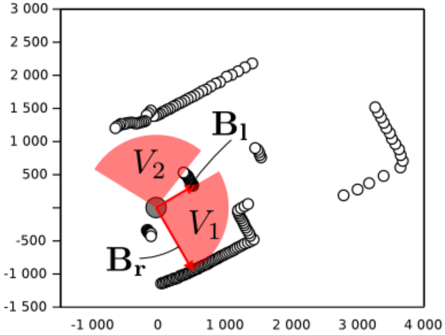




2: Identifying Free Spaces - Boundary Vector

Both VFH and VFH$^+$ try to identify free spaces V - spaces capable for the robot to pass through, by different method. Each free space V_j is defined by two boundary vectors $(\mathbf{B_l}, \mathbf{B_r})_j$:

$$\mathbf{B_l} = \begin{bmatrix} \theta_l & d_l \end{bmatrix}$$
$$\mathbf{B_r} = \begin{bmatrix} \theta_r & d_r \end{bmatrix}$$



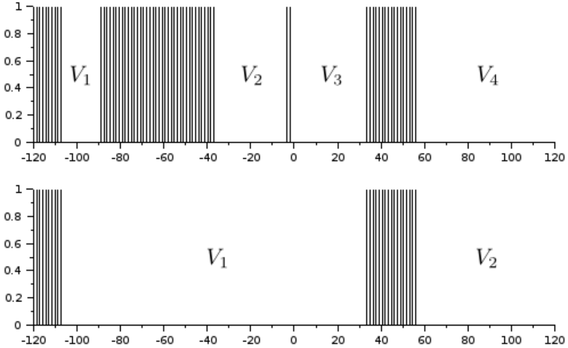
2: Identifying Free Spaces - Hysteresis Filter

VFH$^+$ uses two thresholds τ_{max} and τ_{min} instead of single threshold τ in VFH to generate a *Binary Histogram* H_i, identifying all the free spaces.

$$H_i = \begin{cases} 1 & \text{if } P_i \geq \tau_{max} \\ 0 & \text{if } P_i \leq \tau_{min} \\ H_{i-1} & \text{otherwise} \end{cases}$$

2: Identifying Free Spaces - Hysteresis Filter

By hysteresis filter, VFH$^+$ has reduced the number of free spaces, which overcomes the frequent oscillations of VFH in narrow indoor environment.



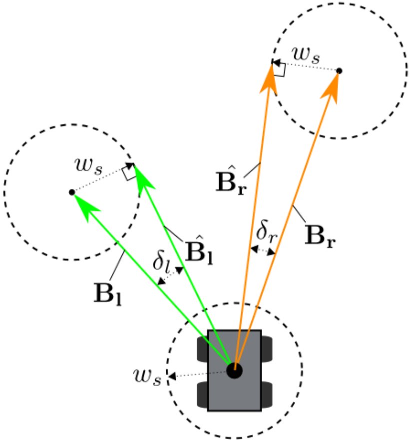
2: Free Spaces - Robot's Geometry

With geometry constraints, free spaces with shrinked boundaries $\hat{V}_j = (\hat{\mathbf{B}}_\mathbf{l}, \hat{\mathbf{B}}_\mathbf{r})_j$ of each V_j is calculated:

$$\hat{\mathbf{B}}_\mathbf{l} = \begin{bmatrix} \theta_l - \delta_l & d_l \cos \delta_l \end{bmatrix}$$
$$\hat{\mathbf{B}}_\mathbf{r} = \begin{bmatrix} \theta_r + \delta_r & d_r \cos \delta_r \end{bmatrix}$$

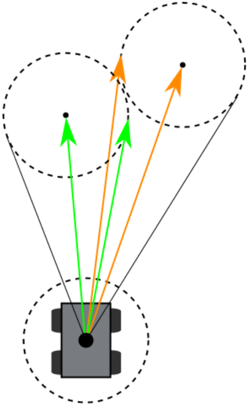
where

$$\delta_l = \arcsin(\frac{w_s}{d_l})$$
$$\delta_r = \arcsin(\frac{w_s}{d_r})$$



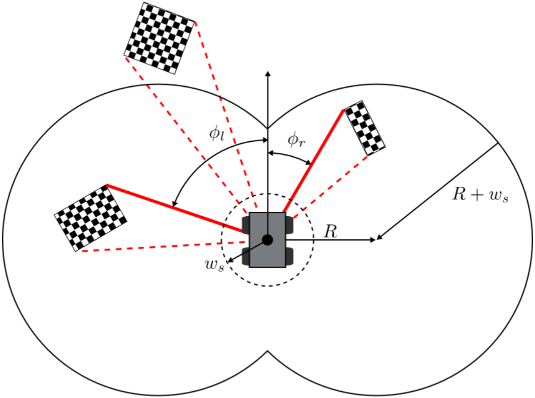
2: Free Spaces - Boundary Overlapped

The V_j' with overlapped boundaries where $\theta_r + \delta_r > \theta_l - \delta_l$ are abandoned, since they are considered too narrow to pass through.



3: Blocked Directions

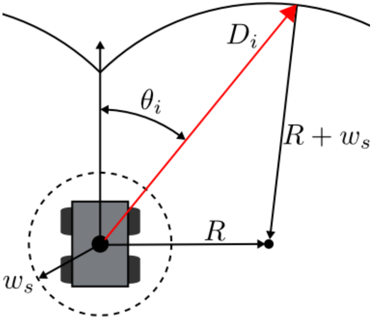
VFH$^+$ takes the minimum radius of rotation of robot into account, determines the limitation of steering angles ϕ_r and ϕ_l.



3. Blocked Directions - Detection Histogram

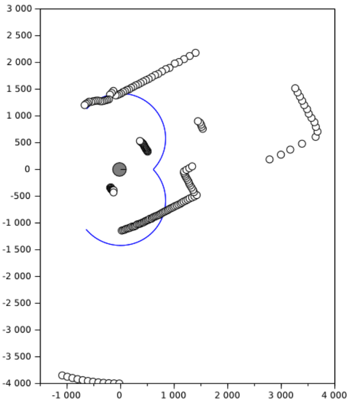
In order to calculate ϕ_r and ϕ_l, the detection histogram D_i is generated first:

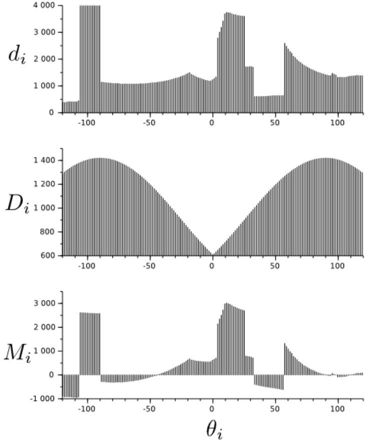
$$D_i = |R \sin \theta_i| + \sqrt{R^2 \sin^2 \theta_i + w_s^2 + 2Rw_s}$$



3. Blocked Directions - Masked Histogram

The masked histogram $M_i = d_i - D_i$ shows whethter the steering angle is blocked by obstacles.





3. Blocked Directions - Determine ϕ_r and ϕ_l

ϕ_r and ϕ_l can be efficiently found by following method:

1) Initially set $\phi_r = -\pi$ and $\phi_l = \pi$

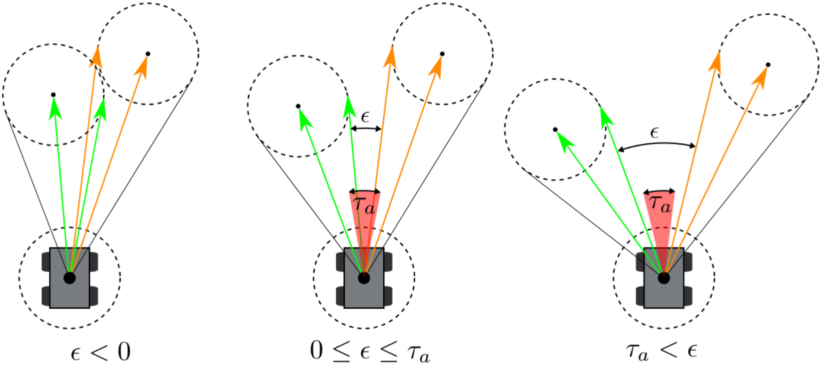
2) For every $M_i < 0$:

  a) If $\theta_i < 0$ and $\theta_i > \phi_r$, set ϕ_r to θ_i

  a) If $\theta_i > 0$ and $\theta_i < \phi_l$, set ϕ_l to θ_i

4. Selection of Steering Direction - Width of Free Spaces

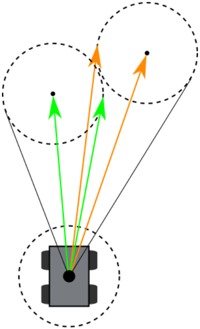
According to the width of each free space $\hat{V}_j$, single or multiple candidate directions c could be found. The width of a free space is determined by its spanning angle $\epsilon = \theta_l - \theta_r$ and a threshold τ_a, which has 3 kinds of different situation:



4. Selection of Steering Direction - Candidate Directions

$\epsilon < 0$ represents free space with overlapped boundary, which is abandoned.

- No candidate direction!

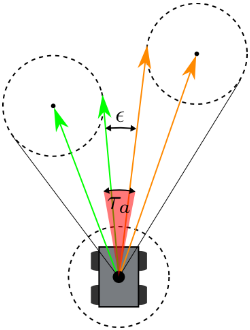


$\epsilon < 0$

4. Selection of Steering Direction - Candidate Directions

For a free space with $0 \leq \epsilon \leq \tau_a$, the centered direction is the only candidate direction.

- $c_n = \frac{\theta_l + \theta_r}{2}$

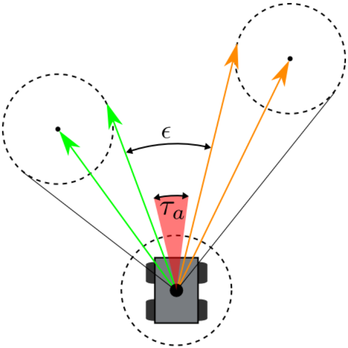


$0 \leq \epsilon \leq \tau_a$

4. Selection of Steering Direction - Candidate Directions

For a free space with $0 < \tau_a < \epsilon$, there are 2 or 3 candidate directions.

- $c_r = \theta_r$
- $c_l = \theta_l$
- If $\theta_l < \Theta_t < \theta_r$, $c_t = \Theta_t$



$\tau_a < \epsilon$

4. Selection of Steering Direction - Cost Function

Like VFH, VFH$^+$ also uses a cost function to select the preferred direction c_t:

$$G(c) = \mu_1 \cdot (|c - \Theta_t|) + \mu_2 \cdot |c| + \mu_3 \cdot (|c - c_{t-1}|)$$

and

$$c_t = min\,\{G(c)\}$$

where

$$\begin{aligned} \Theta_t &= \text{Target direction} \\ c &= \text{Candidate directions} \\ c_{t-1} &= \text{Previously selected direction} \end{aligned}$$

No Candidate Directions!

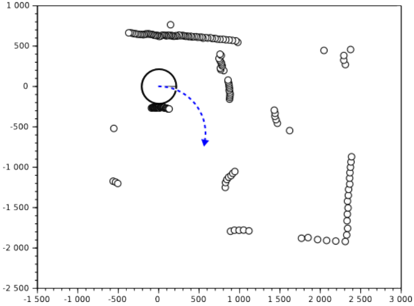


Table of Contents

Obstacle Density Function

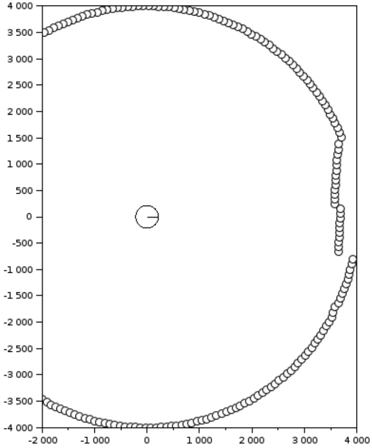
VFH$^+$ uses *Obstacle density function* D to calculate the speed of robot in the environment:

$$D(d_i) = 1 - \frac{1}{N}\sum_{i=1}^{N}\frac{d_i}{d_{max}}$$

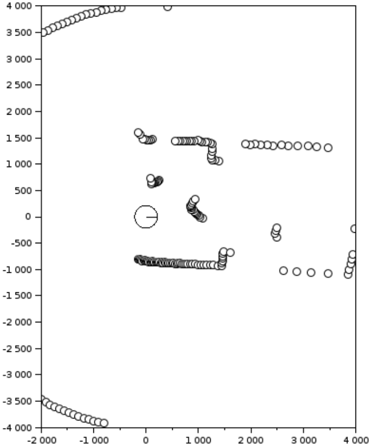
The value of D lies between 0 and 1. Therefore, defined a maximum speed v_{max} and minimum speed v_{min}, the speed of robot in the environment v could be determined:

$$v = v_{min} + (1 - D(d_i)) \cdot (v_{max} - v_{min})$$

Obstacle Density Function



$D = 0.01$

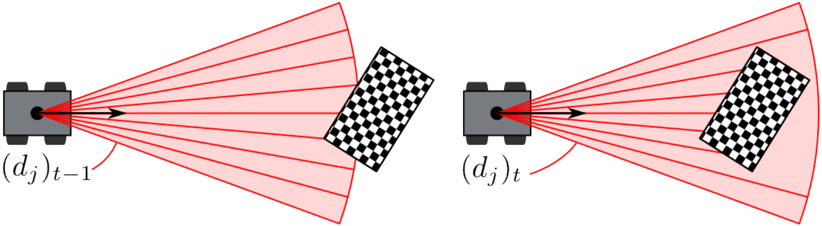


$D = 0.49$

Obstacle Approaching Rate

Speed controlled by D considered only current environment, which is insufficient for high speed robot. Therefore, *Obstacle approaching rate* δ is introduced:

$$\delta = -\frac{1}{N}\sum_j \frac{(d_j)_t - (d_j)_{t-1}}{T}$$



Obstacle Approaching Rate

For high speed robot, the ability of deceleration while approaching an obstacle with high speed is critical. Therefore only *approaching* rate of change is considered:

$$\delta_a = -\frac{1}{N}\sum_j \frac{P((d_j)_t - (d_j)_{t-1})}{T}$$

where

$$P(x) = \begin{cases} x & \text{if } x < 0 \\ 0 & \text{otherwise} \end{cases}$$

Obstacle Approaching Rate

In order to combine obstacle approaching rate with obstacle density, it has to be normalize to the range between 0 and 1 with maximum speed v_{max}:

$$\delta_n = -\frac{1}{N \cdot v_{max}} \sum_j \frac{P((d_j)_t - (d_j)_{t-1})}{T}$$

And the speed v becomes:

$$v = v_{min} + (1 - \frac{D(d_i) + \delta_n}{2}) \cdot (v_{max} - v_{min})$$

Conclusion

Compare to original VFH method, VFH^{+} eventually overcomes some defects:

- Overcome the primary problem of VFH where steering angle is determined by spaces
- Create smooth trajectory by hysteresis threshold
- Take robot's geometry and kinematic constraints into account

However, it still suffers from some problems:

- The geometry of the robot is assumed to be circular
- Leads the robot into dead end which can be avoided