VFH^{+} : Vector Field Histogram Plus

Enhanced Obstacle Avoidance Algorithm

John Liu

Advisor:

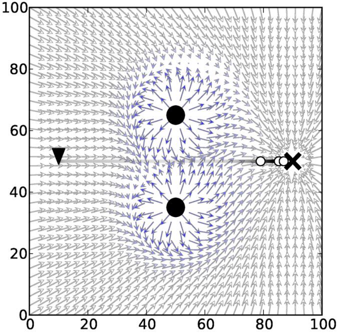
Professor Ying-Jeng Wu

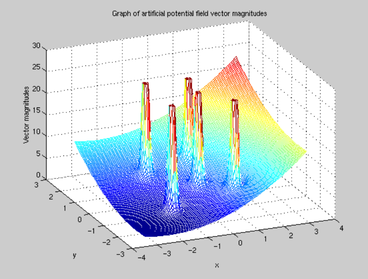
Measurement Laboratory, National Yunlin University of Science & Technology

March 12, 2014

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.



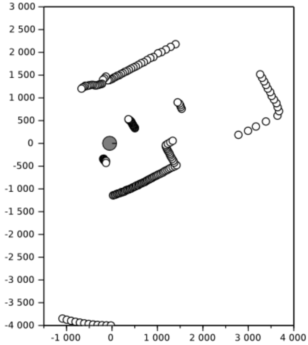


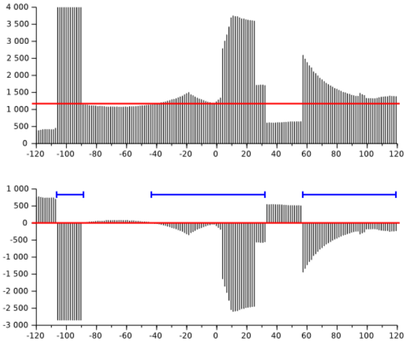
Artificial Potential Field

- 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

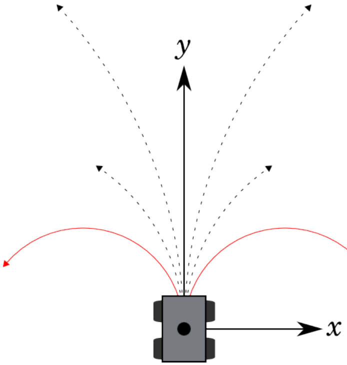
u_1, u_2 and u_3 are weighting constants

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)

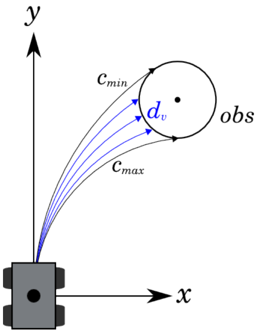
CVM takes robot's kinematic constraints into account, assumes it only travels along circular trajectories with curvature $c = \omega/\nu$, where ω is the rotational velocity and ν is the translational velocity with limits.



Curvature Velocity Method (CVM)

The travelled distance d_v of each obstacle can be calculated, and selected by a cost function. For real-time implementation, There are some simplifications:

- Obstacles are circular shaped
- d_v between each c_{min} and c_{max} of an obstacle is divided onto few intervals with constant value



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

$$\begin{aligned} speed(\nu) &= \nu / \nu_{max} \\ dist(\omega, \nu) &= d_v / d_{max} \\ head(\omega) &= 1 - |\theta_{target} - \omega \cdot T_c| / \pi \end{aligned}$$

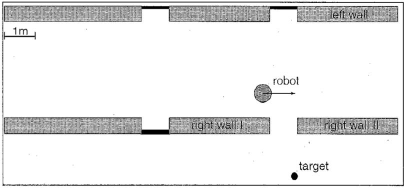
The velocities which generate the largest value will be chosen!

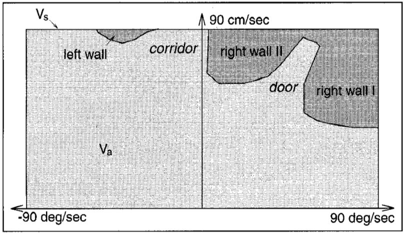
Curvature Velocity Method (CVM)

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

Dynamic Window Approach (DW)

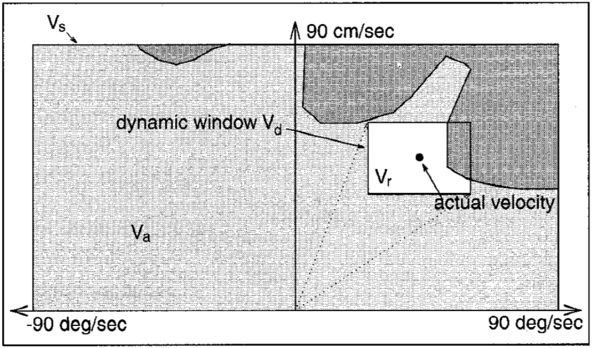
DW also assumes the robot only travelled in circular path with rotational velocity ω and translational velocity ν. The sensed environment is then transformed into **velocity space**.





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**

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

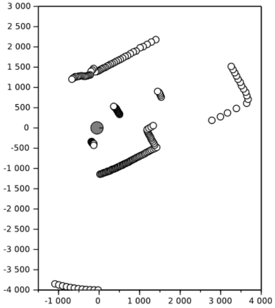
However, some modification is required in order to implement VFH$^+$ with laser range finder, therefore the process become:

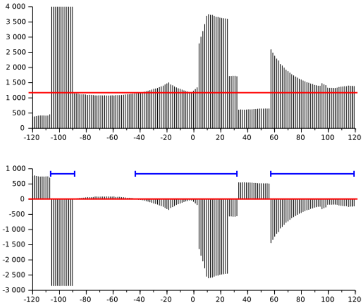
1. Primary Polar Histogram
2. Free Spaces
3. Blocked Directions
4. Selection of Steering Direction

1: Primary Polar Histogram

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

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

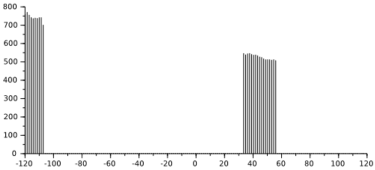


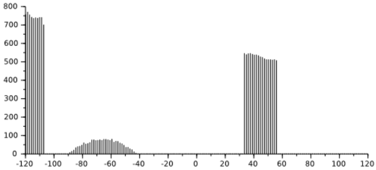


2: Free Spaces - Hysteresis Filter

VFH$^+$ uses two thresholds τ_{max} and τ_{min} instead of single threshold τ in VFH to overcome the oscillation motion in environments with several narrow opennings:

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



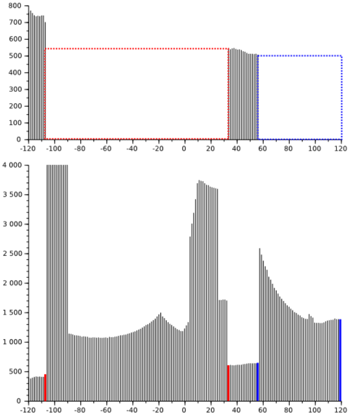


2: Free Spaces - Boundary

Each valley of filtered polar histogram P_i' indicates safe directions for the robot, namely V_j.

The boundaries of each V_j are defined by two angle:

$$V_j = (\theta_j^r, \theta_j^l)$$



2: Free Spaces - Robot's Geometry

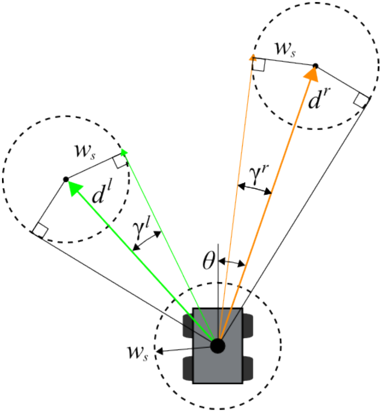
With geometry constraints, new boundaries $V_j' = (\theta_j^{r\prime}, \theta_j^{l\prime})$ of each V_j are calculated:

$$\begin{aligned}\theta_j^{r\prime} &= \theta_j^r + \gamma^r \\ &= \theta_j^r + \arcsin(w_s/d^r)\end{aligned}$$

and

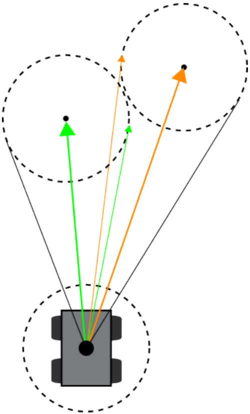
$$\begin{aligned}\theta_j^{l\prime} &= \theta_j^l - \gamma^l \\ &= \theta_j^l - \arcsin(w_s/d^l)\end{aligned}$$

where d^r and d^l are corresponding measured distances.



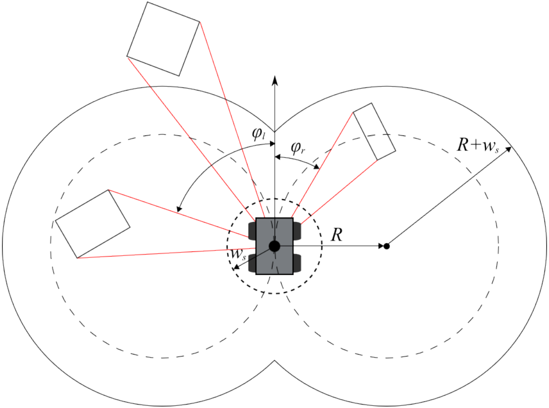
2: Free Spaces - Overlapped

The V_j' with overlapped boundaries where $\theta_j^{r\prime} \geq \theta_j^{l\prime}$ are abandoned, since they are considered too narrow to pass through.



3: Blocked Directions

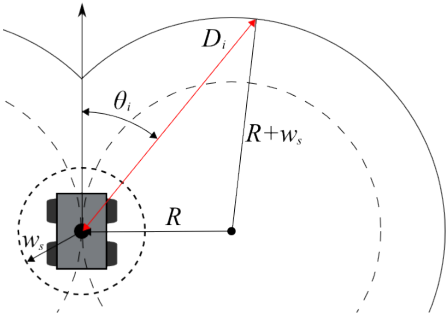
VFH$^+$ also assumes circular trajectories of robot's motion. However, this assumpation only used to determine the limitation of steering angles φ_r and φ_l:



3. Blocked Directions - Detection Histogram

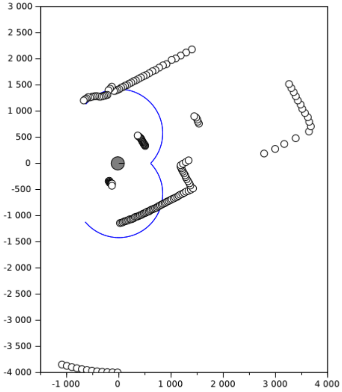
In order to calculate φ_r and φ_l, the detection histogram D_i is generated first:

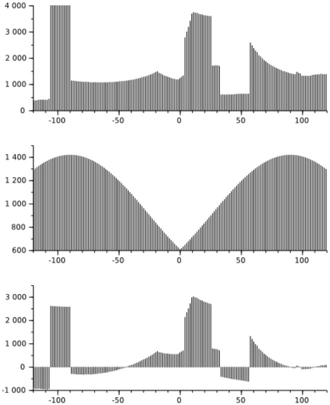
$$D_i = |R \sin \theta_i| + \sqrt{R^2 \sin^2 \theta_i + w_s^2 + 2Rw_s}$$



3. Blocked Directions - Masked Histogram

The masked histogram $M_i = d_i - D_i$, which 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 $\varphi_r = -\pi$ and $\varphi_l = \pi$

2) For every $M_i < 0$:

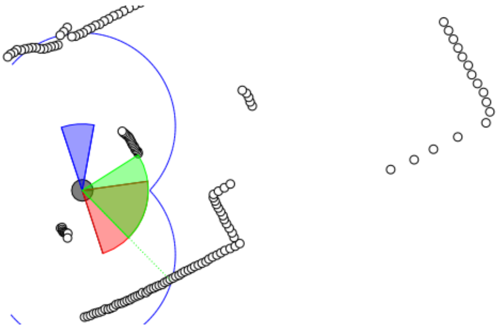
   a) If $\theta_i < 0$ and $\theta_i > \varphi_r$, set φ_r to θ_i

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

4. Selection of Steering Direction - Candidate Direcions

A set of candidate directions c_n are selected from θ_i which satisfied:

$$\theta_i \in (\varphi_r, \varphi_l) \cap (V_1' \cup V_2' \cup \cdots \cup V_j')$$



4. Selection of Steering Direction - Cost Function

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

$$G(c_n) = u_1 \cdot (|c_n - \varphi_t|) + u_2 \cdot (c_n) + u_3 \cdot (|c_n - c_{t-1}|)$$

and

$$c_t = min\left\{G(c_n)\right\}$$

where

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

Algorithm of Speed

Speed of the robot is controled by the density function $D(d_i)$ of surrounding objects with certain threshold τ_{obj}:

$$D(d_i) = \frac{1}{N}\sum_{i=1}^{N} H_i$$

where

$$H_i = \begin{cases} 0 & \text{if } d_i \geq \tau_{obj} \\ 1 & \text{if } d_i < \tau_{obj} \end{cases}$$

With maximum speed ν_{max} and minimum speed ν_{min}:

$$\nu = (\nu_{max} - \nu_{min}) \cdot (1 - D(d_i)) + \nu_{min}$$

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