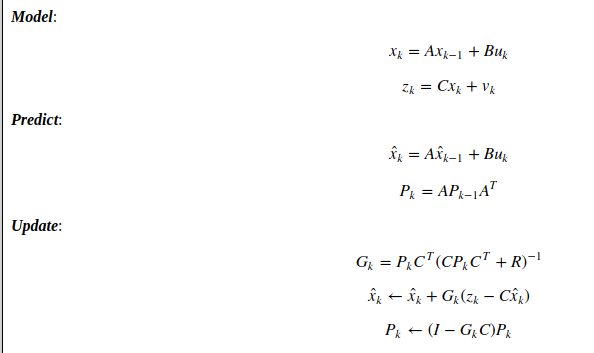
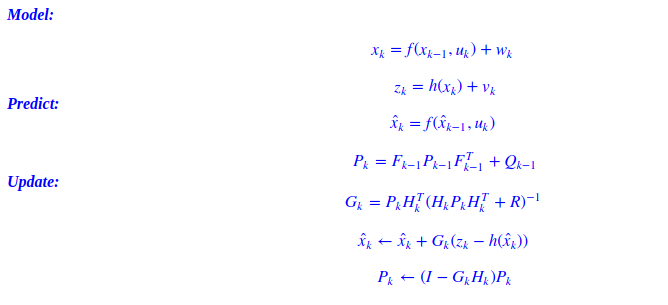
# Filter

Kalman Filter



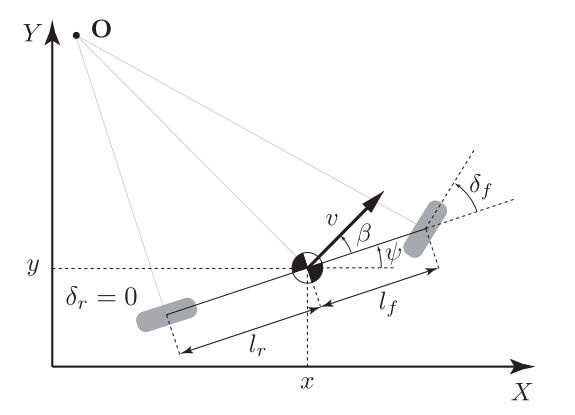


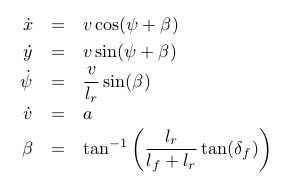
# Udacity Self Driving Car

Term2

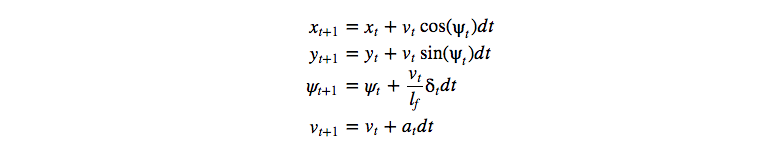
### 1. CarND-MPC-Project-P5

Ref:





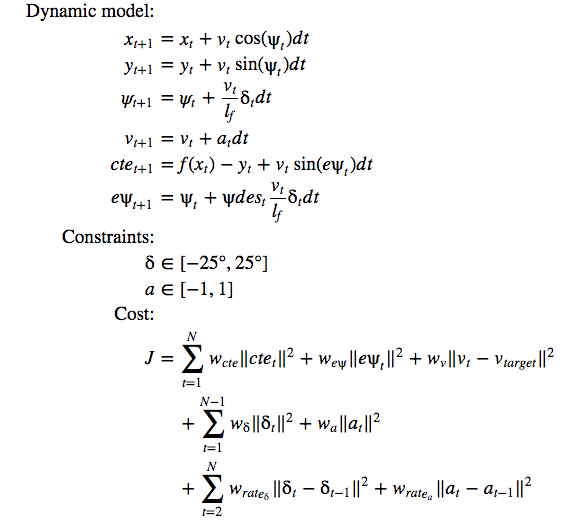
以上是论文里面给出来的Kinematic Bicycle Model，下图是Udacity里面推出来的位置迭代迭代，我们可以出来，关于角度ψ的迭代公式，角度的变化与lf成反比，而按照上面的模型来判断，应该是与lr成反比。这是一个比较大的疑问？？？？



<https://medium.com/@cacheop/implementing-a-model-predictive-control-for-a-self-driving-car-7ee6212a04a8>

<https://medium.com/@hsinchengchao/%E8%87%AA%E9%A7%95%E8%BB%8A%E4%B9%8B%E8%B7%AF7-mpc-control-4552b10aa877>

[https://medium.com/@jonathan\_ hui/lane-keeping-in-autonomous-driving-with-model-predictive-control-50f06e989bc9](https://medium.com/@jonathan_hui/lane-keeping-in-autonomous-driving-with-model-predictive-control-50f06e989bc9)



### 2. Ackermann Model

Term3

### CarND-Path-Planning-Project-P1

#### Frenet坐标系

关于waffle的使用调查

### CarND-Capstone

#### 3.1 Ref

<https://github.com/kung-fu-panda-automotive/carla-driver>

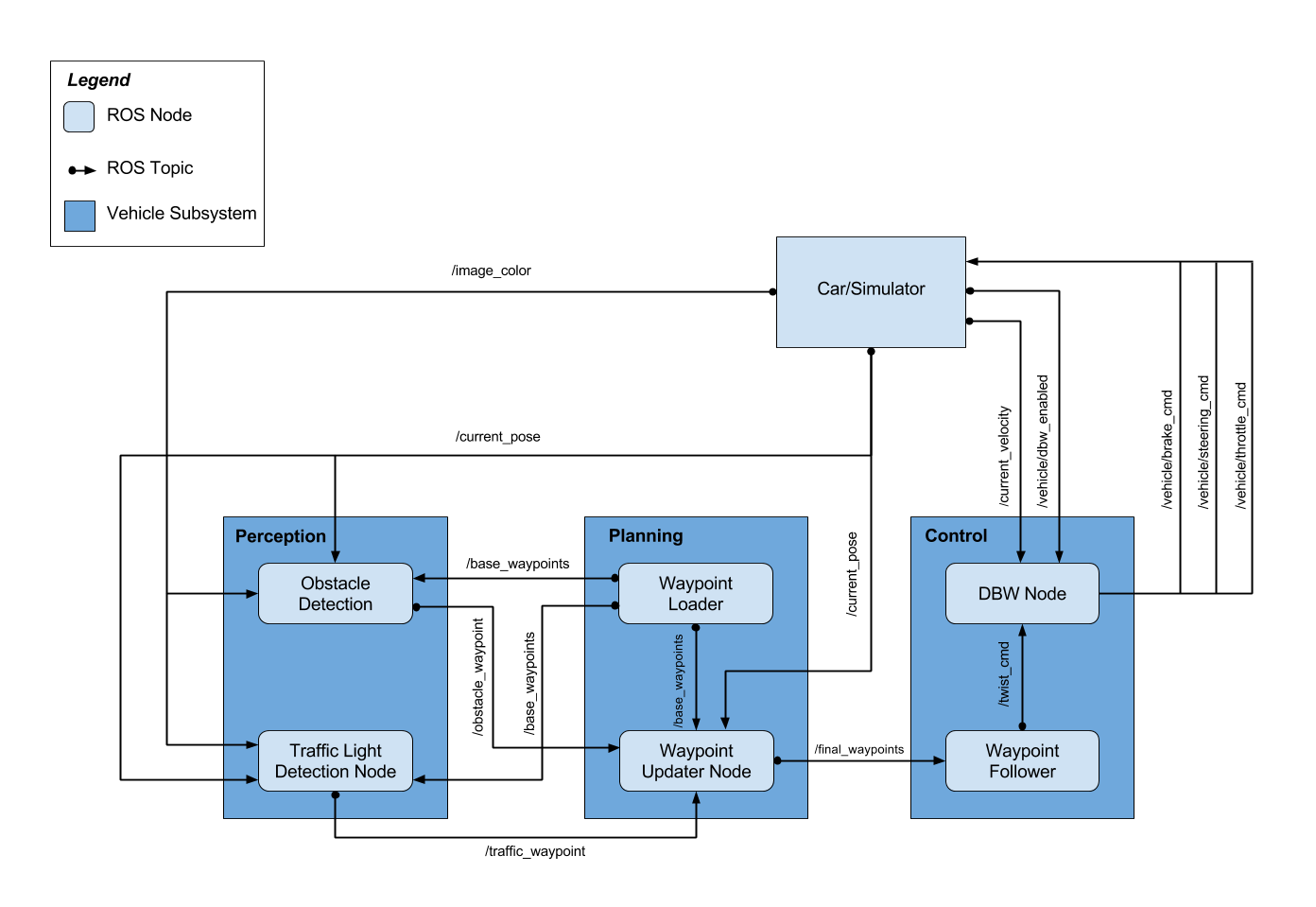
<https://github.com/SiliconCar/CarND-Capstone>

#### 3.2 项目模块分析

项目主要分成两个部分，

1. 直接用rosbag来模拟仿真数据
2. 直接在Carla上面运行

Rosgraph



##### 3.2.1 Perception

**Traffic Light Detection Node**

主要是通过车载摄像头进行识别，然后将识别到的信息发送个/traffic\_waypoint用来决定机器人是不是需要停车。模块供能分为两个阶段， traffic light detection (localization) and (light) color classification

# Autoware

规划算法 Lavalle

路径规划大神 Koening

## 1. Vector map

？？什么是vector map and 怎么生成vector map

## 2. PathSmoothing Thrun Conjugate gradient

Dp\_planner Conjugate gradient路径平滑的方式很简单，

PathSmooth方法就是Sebastian将的方法，复习一下原理，blog

/home/wangmin/Autoware/ros/src/computing/planning/common/lib/openplanner/op\_planner/src/LocalPlannerH.cpp的Line 732

## 3. OpenPlanner

采用的方法是

Open Source Integrated Planner for Autonomous Navigation in Highly Dynamic Environments.pdf

#### 3.1 Introduction

**全局路径规划现状**

传统的方法有A星，D星，Voronoi图等方法，他们都是基于costmap格式的，但是对于上千里的路径规划，如果用他们效率就实在是根本上。

目前在无人车领域用的全局路径规划的方法是Vector Map格式， RNDF

**行为状态机**

进行状态任务切换。

**局部路径规划**

1. Potential Field
2. 位置预测方法【19】
3. 动态窗口法
4. 考虑人类因素的动作规划
5. 根据参考路线，计算平行路线的方式，有就是本文用的方法

**常用的开源环境**

1. Open Motion Planning Library
2. ROS navigation
3. Open-rdc

Openplanner的主要优势是

主要针对的就是无人驾驶，采用vector map格式，能够使用交通灯信息，交通标识，交叉路口，停车线，这样在路径规划的时候，就没有没么多计算的复杂度。在停车以及非道路的情况，还是使用了RRT星和Hybrid A星

Ref：https://www.youtube.com/watch?v=JeEk\_CWcRFI

OpenPlanner可以用作Non-holonomic平台

OpenPlanner可以添加行为状态机，添加的方式非常简单

总之，OpenPlanner更加的适用于自动机器人因为他能够遵守交通规则，而且在全局路径规划的时候，他只需要vector map和全局目标点，在局部路径规划的时候，他只需要当前位置以及检测到障碍物

OpenPlanner主要包含三个部分全局路径规划、局部路径规划和行为状态机

全局路径规划：

输入为vector map，开始点和结束点，通过动态规划，寻找最短或者代价最低的路径。

行为状态机：

通过当前交通状况以及交通规则来进行状态切换

局部路径规划：

输入为全局参考路径 以及当前位置，输出为几条候选路径以及当前选择的代价最低的路径。

#### 3.2 全局规划

4.1.1 vector map

Vector map主要用来区分GIS地图，包含了一些在全局路径规划过程中需要的信息，如果用多项式表达道路中心的话，那么计算的成本将会非常的高。

4. 2.1 全局路径规划

主要是根据vector-map然后进行类似Dijastra路径搜索

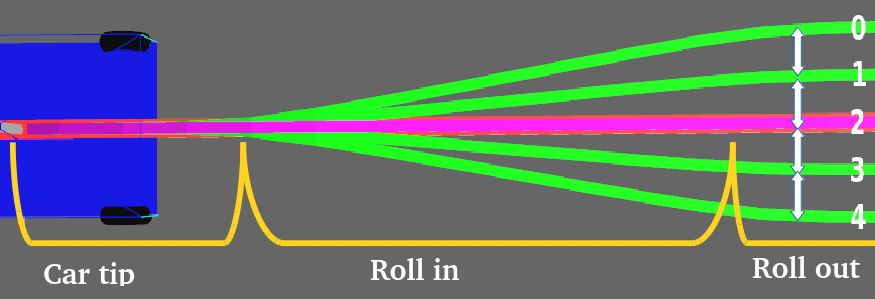
#### 3.3 局部规划

局部规划主要是生成直接可以跟踪的路径

##### 3.3.1 Roll-out生成

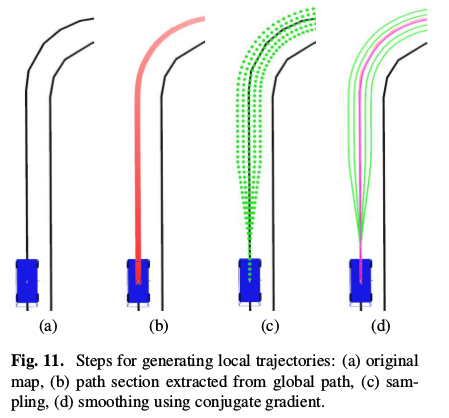
Roll-out最少的生成频率为10hz，输入为当前的位置，规划的距离，roll-outs路径的个数n和下一段参考的全局路径，输出为n条平滑的局部路径，从机器人中心一直延伸到最大规划距离

Roll-outs路径可以可以分为三个部分：car-tip margin（决定了路径在切换的时候的平滑度）， roll-in margin (从car-tip终端到平行起始点，这段距离也汽车的速度成正比), roll-out margin这段就是中roll-in终点到最大平滑的距离，每条线距离全局参考路线一定距离（roll -outs density来表示这个距离）。



Roll-outs局部路径规划也包含3步：

1. 根据车辆当前的位置以及最大规划距离，获取全局的参考路径。
2. 根据全局路径生成与之垂直的路线点。生成的路径从car tip开始，首先的平行距离为0，然后开始进入roll-In段，然后通过index增量慢慢的增长到roll-outs density。
3. 通过conjugate gradient方法对路径进行平滑，减少了在采样生成过程中，路径的不连续性。提高了曲率，同时平滑了旋转路径。



生成的路径通过piece wise interpolation（分段差值）的方式来改变density，很多的参数差值方式对噪声很敏感并且会将噪声传到后面的结果，因此这里通过piece wise interpolation和conjugate gradient结合的方式来生成平滑的路径。

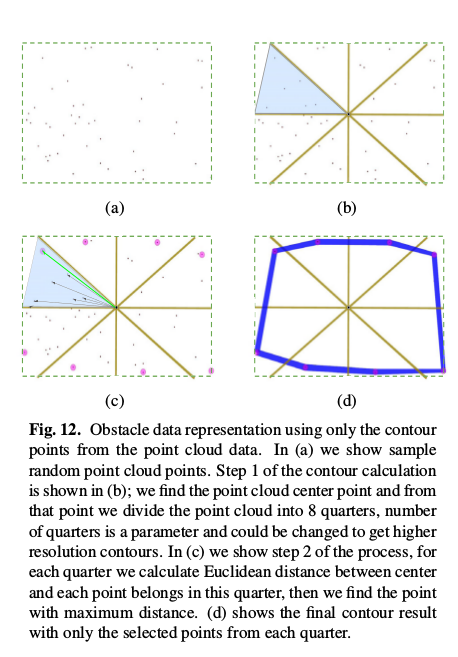
##### 3.3.2 cost calculation

代价计算主要是用来选择roll-outs生成的最优轨迹，算法的输入是roll-outs路径以及检测到的障碍。输出是选择的最优轨迹。代价主要由3个部分代价相加得到：priority cost（偏向选择中间路径）， collision cost，transition cost。

Autoware中的障碍物有Bouding Box以及points Cluster两种，Bounding Box的准确度高，但是进度低；而Points Cluster的进度高，但是性能差，因此在这里采用的是contour points的方法来提取每一Points Cluster的轮廓，每个Cluster最多用16个点表示。

Contour Points方法的步骤如下：

1. 把xy空间分为n sector（n是一个调节的参数）
2. 找到每一个点与中心点的距离和角度并
3. 得到距离中心点最远的点作为这个sector的表示点。



5.2.1 center cost

Center cost用来约束汽车行驶在中间路线中，用他们离中间路线的距离来表示

5.2.2 transition cost

约束汽车在不同的路径中进行切换，通过计算当前路径与不同路径之间的归一化垂直距离作为代价

5.2.3 Collision cost

计算分为2个阶段：

1. 测试rolls-in与障碍物的距离，计算每一条路径到障碍物轮廓点的距离；测试距离的方法计算的方法很简单，计算waypoints到每条障碍物轮廓边的距离，判断这个距离与车宽度/2加上障碍物测量误差
2. 测试rolls-out这一段与障碍物的距离。由于经过roll-in阶段，所有的路径都与中间路径平行，因此在rolls-in之后的路线，不需要直接测试他们的距离。因此我们只需要计算障碍物轮廓到中心路径的距离，然后加上每个路线到中心路线的距离就可以了

#### 3.4 基于状态机的行为生成

1. 从一个状态切换到另外一个状态基本上是通过不同的触发条件得到的，这样在一些临界值的时候，就会出现在一个状态和另外一个状态中来回切换。为此，引入定时函数或者计数器，能够解决这个问题，而且在交通灯检测这些问题中，还可以通过定时来判断红灯有没有变绿，这种方式比直接的检测更加的靠谱。

## 4. Way\_planner

视频教程

https://www.youtube.com/watch?v=FKM8v79X3\_s&t=41s

Lattice\_velocity\_planner

1. V\_scan 从哪里过来，可以通过find\_in\_project 来查找

VScan 是从节点VitualScanImage中的points2vscan过来

Lattice\_trajectory\_gen

1. computeWaypointGoal怎么三个点的计算曲率,也就是Lattice\_Trajectory\_gen的核心是什么，具体可以参考PythonRobotics

## 5. Points2image

## 6. Obstacle\_Avoid

根据当前点，之前的obstacle\_index, 计算跟踪的路径，如果用到的AStart用作避障碍，也就是临时修改障碍区域的跟踪路径 。

## 7. Velocity\_set

主要根据当前的位置，在路径上最近点，然后检测周围的obstacle， crosswalk，判断是否需要停车还是减速，然后根据停车和减速点，修改路径中的跟踪速度。

## 8. Pure\_pursuit

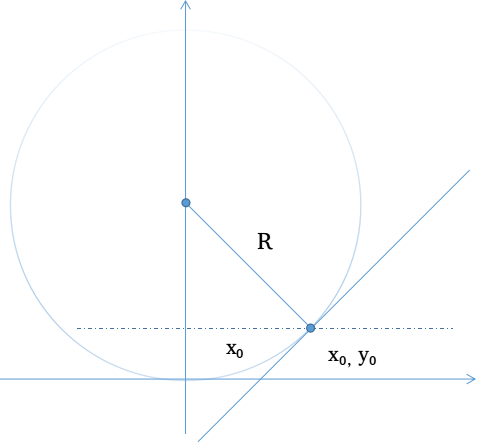
需要搞清楚的几点：

1. 怎么通过两点计算曲率

两点之间的曲率计算，当前点方向相切，并且与目标方向的相切的点，

首先需要将目标点转换到当前点的坐标系下，然后曲率是画圆弧线。

曲率计算的过程，首先将目标点转换到当前汽车的局部坐标系中，然后计算公式



*R2 = X02 + (R - Y0)2*

1. 怎么通过曲率计算出ackeman car模型的转向角度。

具体可以参考Catvehicle中的视频

https://www.youtube.com/watch?v=i6uBwudwA5o&list=PLcXdwRQiNMIoAOcJZvq5\_c2xsoegDxu4N

## 9. Lane Planner软件包

包含节点

lane\_navi

lane\_rule

根据vmap发布对应的traffic点。

lane\_select

lane\_stop

没有什么特别的

**Lane\_select**

**需要搞清楚什么是Hermite Curve**

**概要**

1. 接收多条路线
2. 从当前位置计算所有路线的最近邻居
3. 设置距离当前正在运行的路线最近的车道
4. 检测到当前路线的左右路径
5. 由当前路线的最近邻居持有的车道改变标志被保持为路线的车道改变标志
6. 最近与右转或左转的标志旁找到点，产生的路径，目标点是该计划的点和变道的车道埃尔米特插值，？？？？，融合的路径和车道变更计划路线后的目标点将路线定义为换道路线
7. 当未更换车道时，分别发布当前路线，最近邻近点和车道更换标志
8. 更换车道时，分别公布车道变换路线，最近邻点和车道变更标志

**注意**

In order to change the lane, it is necessary to read multiple route files of ver3 format. (See waypoint\_maker package)

**Subscribed Topics**

traffic\_waypoints\_array (waypoint\_follower/LaneArray)

current\_pose (geometry\_msgs/PoseStamped)

current\_velocity (geometry\_msgs/TwistStamped)

state (std\_msgs/String)

config/lane\_select (runtime\_manager/ConfigLaneSelect)

Published Topics

base\_waypoints (waypoint\_follower/lane)

closest\_waypoint (std\_msgs/Int32)

change\_flag (std\_msgs/Int32)

lane\_select\_marker (visualization\_msgs/MarkerArray) : for debug

Parameter from Runtime Manager

Distance threshold to neighbor lanes  
lanes Represents the threshold when detecting a valid lane around the current route. Lanes lying far from this threshold are not recognized as lanes.

Lane Change Interval After Lane Merge

Represents the value of how many meters after running a lane change it will be possible to change the lane again.

Lane Change Target Ratio

The value used when defining the target point on the lane to be changed lane by the distance proportional to the speed (m / s). The starting point of the target point search is the nearest point of the point having the lane change flag of right turn or left turn at the point on the lane where the lane is scheduled to change.

Lane Change Target Minimum

This represents the minimum distance to the target point on the lane to be changed lane. The starting point of the target point search is the nearest point of the point having the lane change flag of right turn or left turn at the point on the lane where the lane is scheduled to change.

Vector Length of Hermite Curve

Represents the magnitude of vector when complementing with Hermitian curve.

参考资料

<https://github.com/CPFL/Autoware/blob/master/ros/src/computing/planning/mission/packages/lane_planner/README.md>

**waypoint\_maker**

**Overview**

This package has following nodes.

waypoint\_clicker

waypoint\_saver

waypoint\_marker\_publisher

waypoint\_loader

waypoint\_velocity\_visualizer

3-formats of waypoints.csv handled by waypoint\_maker

ver1： consist of x, y, z, velocity（no velocity in the first line）

ex)

3699.6206,-99426.6719,85.8506   
3700.6453,-99426.6562,85.8224,3.1646   
3701.7373,-99426.6250,85.8017,4.4036   
3702.7729,-99426.6094,85.7969,4.7972   
3703.9048,-99426.6094,85.7766,4.7954   
3704.9192,-99426.5938,85.7504,4.5168   
3705.9497,-99426.6094,85.7181,3.6313   
3706.9897,-99426.5859,85.6877,4.0757   
3708.0266,-99426.4453,85.6608,4.9097   
...

ver2： consist of x, y, z, yaw, velocity（no velocity in the first line）

ex)

3804.5657,-99443.0156,85.6206,3.1251   
3803.5195,-99443.0078,85.6004,3.1258,4.8800   
3802.3425,-99442.9766,85.5950,3.1279,7.2200   
3801.2092,-99442.9844,85.5920,3.1293,8.8600   
3800.1633,-99442.9688,85.5619,3.1308,10.6000   
3798.9702,-99442.9609,85.5814,3.1326,10.5200   
3796.5706,-99442.9375,85.6056,3.1359,10.2200   
3795.3232,-99442.9453,85.6082,3.1357,11.0900   
3794.0771,-99442.9375,85.6148,3.1367,11.2300   
...

ver3： category names are on the first line

ex） consist of x,y,z,yaw,velocity,change\_flag

x,y,z,yaw,velocity,change\_flag   
3742.216,-99411.311,85.728,3.141593,0,0   
3741.725,-99411.311,85.728,3.141593,10,0   
3740.725,-99411.311,85.733,3.141593,10,0   
3739.725,-99411.311,85.723,3.141593,10,0   
3738.725,-99411.311,85.719,3.141593,10,0   
3737.725,-99411.311,85.695,3.141593,10,0   
3736.725,-99411.311,85.667,3.141593,10,0   
3735.725,-99411.311,85.654,3.141593,10,0   
...

**Nodes**

**waypoint\_loader**

**Overview**

Convert waypoints.csv to ROS message type.

Correspond to the above 3 types of csv.

Adjust waypoints offline (resample and replan velocity)

Save waypoints.csv as ver3 format.

How to use

How to start

At Computing->waypoint\_loader:

Check app->disable\_decision\_maker. If you want to use decision\_maker, switch to false. Otherwise switch to true.

Check waypoint\_loader and start.

Idea of velocity replanning

The velocity plan is based on the following idea.

On a straight line, it accelerates to the maximum speed.

At one curve:

Finish decelerating before entering the curve. If the curve is sharper, the more deceleration is required.

Maintain a constant speed in the curve.

Start to accelerate after getting out of the curve.

Detail of app tab

On multi\_lane, please select multiple input files. If you want lane change with lane\_select, prepare ver3 type.

Check replanning\_mode if you want to replan velocity.

On replanning mode:

Check resample\_mode if you want to resample waypoints. On resample mode, please set resample\_interval.

Velocity replanning parameter

Vmax is max velocity.

Rth is radius threshold for extracting curve in waypoints. Increasing this, you can extract curves more sensitively.

Rmin and Vmin are pairs of values used for velocity replanning. Designed velocity plan that minimizes velocity in the assumed sharpest curve. In the i-th curve, the minimum radius ri and the velocity vi are expressed by the following expressions. vi = Vmax - (Vmax - Vmin)/(Rth - Rmin) \* (Rth - ri)

Accel limit is acceleration value for limitting velocity.

Decel limit is deceleration value for limitting velocity.

Velocity Offset is offset amount preceding the velocity plan.

End Point Offset is the number of 0 velocity points at the end of waypoints.

Subscribed Topics

/config/waypoint\_loader (autoware\_msgs/ConfigWaypointLoadre)

/config/waypoint\_loader\_output (std\_msgs/Bool)

Published Topics

/lane\_waypoints\_array (autoware\_msgs/LaneArray)

Parameters

~disable\_decision\_maker

waypoint\_saver

Overview

When activated, subscribe /current\_pose, /current\_velocity(option) and save waypoint in the file at specified intervals.

change\_flag is basically stored as 0 (straight ahead), so if you want to change the lane, edit by yourself. (1 turn right, 2 turn left)

This node corresponds to preservation of ver3 format.

How to use

On app:

Ref on the Save File and specify the save file name.

Check Save/current\_velocity if you want to save velocity. In otherwise, saved as 0 velocity.

Using Interval, it is set how many meters to store waypoint.

**Subscribed Topics**

/current\_pose (geometry\_msgs/PoseStamped) : default

/current\_velocity (geometry\_msgs/TwistStamped) : default

**Published Topics**

nothing

**Parameters**

~save\_filename

~interval

~velocity\_topic

~pose\_topic

~save\_velocity

# PythonRobotics

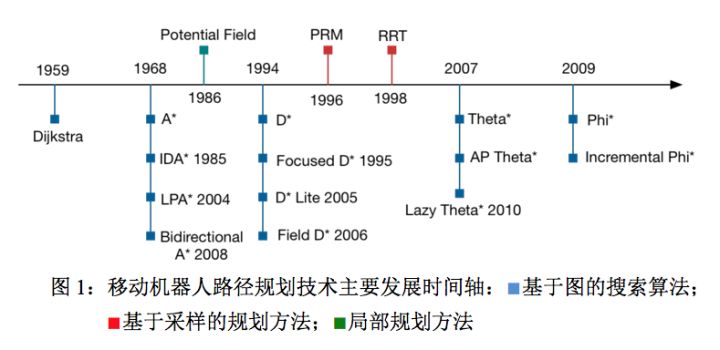
### PathPlanning

#### RRT

A\*

D\*

# 路径规划



## 完备算法：

优点：算法是完备的

缺点：计算量大，特别是在大尺度以及高维度的情况下

深度优先寻路算法

广度优先寻路算法

Dijkstra(迪杰斯特拉)算法

A\*（主要是在Dijkstra上面添加了启发函数）

D\*

Theta\* 增加了line of sight

## 基于采样的算法

主要应用于高维空间，但是在Narrow space的情况下容易死掉。

代表算法：

PRM

RRT

Dubins Path

## 计算几何

## Configuration Space

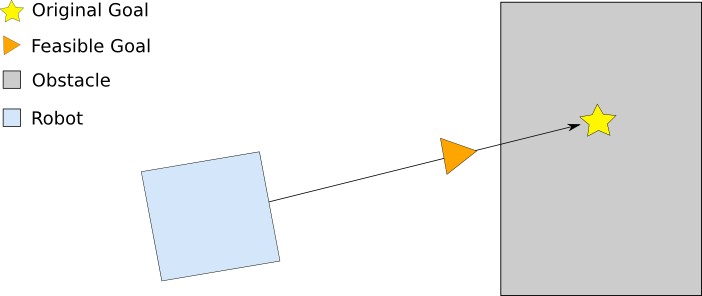
## CostMap的原理以及使用

## ROS Global Planner

### Carrot\_Planner

<http://wiki.ros.org/carrot_planner?distro=melodic>，具体的使用方法

这个规划算法的Idea很简单，规划的路径只有两个点，当前点和目标点，当目标点的位置在障碍物里面的时候，沿着向起始点方向寻找可行的目标点。



# 机器人超车过程

## 行为决策

## 行为规划

# C++

## 封装

## 继承

## 多态

## 模板编程

提高程序员生产率的一个非常有效的方法就是“代码复用”，而面向对象很重要的一个贡献就是通过内部紧耦合和外部松耦合将“思想”转化成一个一个容易复用的“概念”。但是面向对象提供的工具箱里面所包含的继承，组合与多态并不能完全满足实际编程中对于代码复用的全部要求，于是模板就应运而生了。

# 算法和数据结构

## 程序的事件复杂度

Master method主定理

T(n) = T(n-1) + T(n/2) + 1

## 动态规划

状态转移

重叠子结构

## 贪心算法

没有修正功能，因此他得到最优解必须是全局最优就是局部最优。

## 图

### 图的最短路径：Dijkstra、Bellman-Ford、SPFA、Floyd、A\*算法

## 最小生成树

无向图23题

## 二分图

P12 Q4

## 拓扑排序

## 全排列算法

## Top k minimum