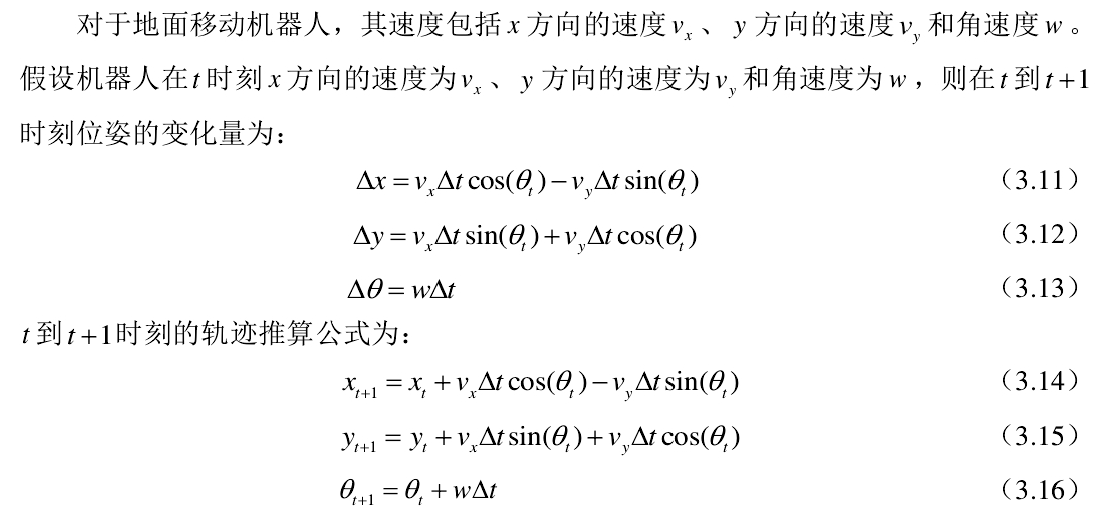
DWA算法的基本思想是，在机器人的控制空间(线速度、角速度指令)中离散地采样，对于每个样本指令，在机器人当前状态和地图信息的基础上，推演未来很短的一段时间内的运动轨迹。 根据发生碰撞的可能性、目标点的接近程度、全局轨迹的跟踪近似度、速度限制等多方面的指标评价这些轨迹并打分，选取得分最高的轨迹，将其对应的指令下发给底盘，控制机器人运动。

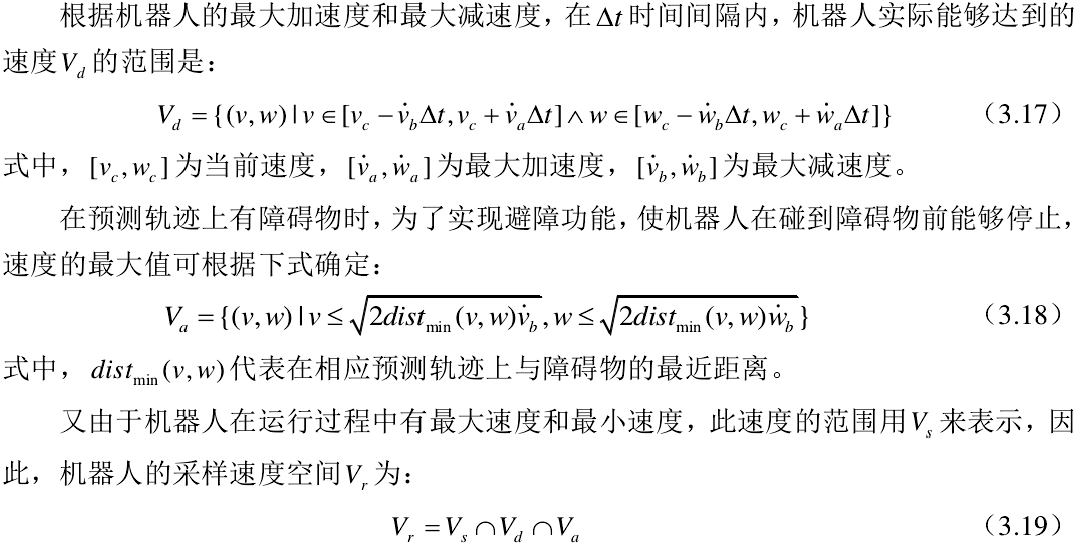
DWA算法的大致步骤如下：

1. 运动模型搭建

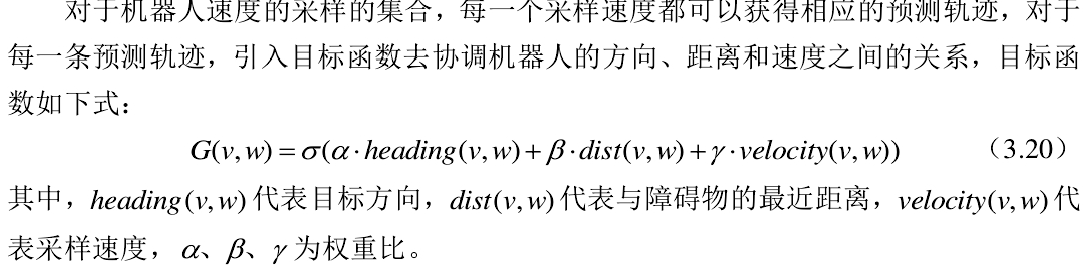


1. 速度采样

采样之前需要对机器人实际能达到的速度范围进行限制，然后在范围内进行采样操作。

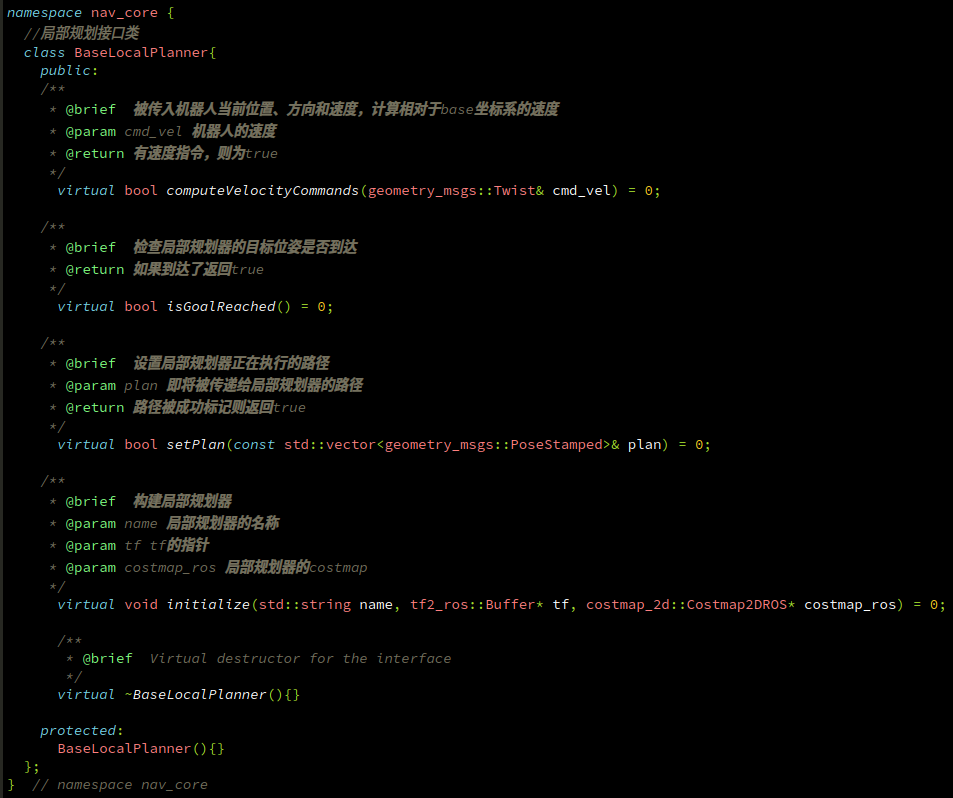


1. 评价函数



**1. 接口——BaseLocalPlanner**

BaseLobalPlanner是move\_base获取局部规划能力的接口， 它是定义在包nav\_core的头文件[base\_lobal\_planner.h](https://github.com/ros-planning/navigation/blob/melodic-devel/nav_core/include/nav_core/base_lobal_planner.h)中的一个虚类。



**2. 封装——DWAPlannerROS**

DWAPlannerROS是接口BaseLobalPlanner的一种实现，是对局部规划器DWAPlanner的封装。下面是DWAPlannerROS的类定义片段，可以看到它继承自BaseLobalPlanner，并实现了上述的四个接口函数。

void DWAPlannerROS::initialize():初始化函数

bool DWAPlannerROS::computeVelocityCommands()：输出局部规划器计算的机器人控制指令

bool DWAPlannerROS::dwaComputeVelocityCommands()：DWAPlannerROS通过成员函数dwaComputeVelocityCommands完成局部的路径规划。

**3. 实现——DWAPlanner**

DWAPlanner并不具体实现局部路径规划算法。它使用包[base\_local\_planner](http://wiki.ros.org/base_local_planner)中的SimpleTrajectoryGenerator来生成预测轨迹， 并为预测轨迹定义了一系列的评价标准。实际的规划则是由SimpleScoredSamplingPlanner进行的。

下图是它所用的各个评价对象，它们都是在base\_local\_planner中实现的，具有相同的基类TrajectoryCostFunction。

