
1. Setup: GetPathService() in get_path_service.cpp

1.1 One instance of the CarVehicle class is created

1.1.a Load all motion primitives generated by matlab

1.1.b Load and create additional information (.adat file +

All motion primitives are stored in the map modelMotionPrimitivesLT_ (VehicleModel class) using the key (startingOrientationID, startingSteeringID).

NOTE: The key should be unique.

.adat Additional Motion Primitive Data

Generation: File generated by the function generatePrimitiveAdditionalData (called in line 223 of CarModel.cpp the file is not already existing).

Each line (corresponding to a motion primitive) has the following format:

startingOrientationID startingSteeringID distanceCovered [{cell_x cell_y} all the swept cells of the motion primitive] CELL_DELIMITER CELL_DELIMITER [{cell_x cell_y} all the occupied cells in the end point of the motion primitive]\n

Usage: see lines 225-306 CarModel.cpp to fill such fields in the lookup table (for each motion primitive)

1.2 GetPathService subscription: this service communicate the start, the goal and the map [input], instantiate a PathPlanner which attempts to solve the planning and returns.

.hst heuristic lookup table

Generation: File generated by function setEntryInHeuristicTable (called in line 492 of VehicleModel.cpp)

each line has two value:

a key calculated in the function HeuristicTableKey (443 VehicleModel.cpp) that consider a start pose and a end pose.

a value calculated in the function extractHTDataFromNode (line 50 in AStarPathPalnner.cpp) and is the "cost to ???".

extractHTDataFromNode is called from the AStarPathPlanner :: solve()

2) Mission : getPathCB(orunav_msgs) in get_path_service.cpp

2.1 read The robotTarget form orunav_msgs

2.2 creates and queues a mission (line 118-123)

2.3 line 129 Pathfinder -> solve launch the path planner

2.3 vector<Configuration*> Pathfinder :: solve(PathFinder.cpp line 78)

create world

create inicial node // perchè più configurazioni, più missioni??

startNode = new PathNode

2.3.1 setPathPlanner .. AStarPathPlanner ..
AState

insertNewNodeForExpansion(startNode) -- > openList

2.3.2 planner → solve (line 163)

2.3.2 vector<Node*> AStarPathPlanner :: solve() (AStarPathPlanner line 100)

2.3.2a line 144 candidate = extractNode():

Extract the best Node candidate from the expansionQueue and
insert it into the closedList. Returns 0 in case no nodes are left

2.3.2b line 166 extractHTDataFromNode(candidate);

Extract the data from the Node to save in the heuristic
table of the vehicle model:

calculate the cost from this configuration to itself + cost
from each parent to the candidate.

if candidate == goal break

else

2.3.2c line 194 children = candidate → generateChildren(PathNode101)
Generate the children reachable from the current PathNode

2.3.2e line 199 For each child:

check if we already have a PathNode with the same content:

1) new → save it into the uniqueList

2) old is better → delete child

3) new is better → substitute (non ben capito la
sostituzione.. puntatori?)

2.3.2c generateChildren(PathNode 101)

there are 4 different methods

- NAIVE NAIVE Base method,
- EP Expansion Pruning
- FSG Fast Successor Generation
- EPFSG Expansion Pruning +Fast Successor Generation

line 116 newConfigurations = generateNewConfigurations(carConfiguration
24)

Generate new Configuration starting from the current one
considering the primitives
29 getApplicablePrimitives(getOrientationID(), getSteeringID)
(vehicleModel 407)

line 124 add the position of the other vehicles as obstacles
(noi consideriamo pathPlanning per uno solo e poi lasciamo
gestire al coordinatore ?)

line 130 create the new nodes
checkCollisionFree
146 newChild

Node Structure

```
/** Actual cost to reach the Node from starting point (G) */  
double G_  
/** Heuristic estimated cost to reach the goal from the Node (H) */  
double H_  
/** Total estimated cost from start to goal through Node (F: F = G + H)  
double F_;
```

summary

1. Setup: GetPathService()

- instance of the CarVehicle class is created
- load all motion primitives + additional data

2 Mission : getPathCB(orunav_msgs)

- add mission and solve it
 - PathFinder: set PathPlanner and initial Node
 - AStarPathPlanner: solve Path planner + HeuristicTable ?
 - extract candidate
 - expand Nodes
 - newConfig considering Primitives
 - selectBest Parent Parent for config
 - if Candidate == goal_node
 - return path to Candidate

