README:

% The V-REP file rrl\_ideal.ttt must be opened and the simulation started before running the Matlab code.

Project\_2\_Astar.m % Main Script

Sets up start and goal positions (StartNode, GoalNode), resolution of movement (res), and if you want to plot the final node costs as colors (cm); saves inputs and path as “InOut.mat”, saves plot of expanded nodes and path as “Astar\_[res]\_[cm].jpg”.

[Path] = Astar\_Search (StartNode, GoalNode, res, cm) % Main Function

Expands nodes from start node to goal node according to A\* algorithm, backtracks path from goal node to start node

res = resd2res(resd) % Sub-function of Astar\_Search

Determines the correct radius (res), given in meters, to check for ‘duplicate’ nodes given the increments of expansion (resd) expressed in degrees. The value of res is given as the minimum distance between endpoints of an expanision as determined from the function xy.

[x,y] = xy (td) % Sub-function of resd2res

Determines the path (x,y), expressed in meters, taken by the differential drive robot for a given change in angle (td), expressed in degrees.

[] = InitDisplay(StartNode, GoalNode, res) % Sub-function of Astar\_Search

Sets up display with start and goal as well as obstacles (determined by EvalCrash)

[Que] = Move (dx, dy, CurrentNode, CurrentNodeInfo, Que, res) % Sub-function of Astar\_Search

Action of moving in direction specified by (dx,dy) by setting the new nodes position as the parent (“current”) nodes position +(dx,dy), updates the record of expanded nodes, the Que, and the display

[Status] = getStatus (Node, res) % Sub-Function of Move

Takes the position of the new node and returns “Status” of the Node as: Status = [Visited Crash Goal]. Has this node position already been visited? (Visited : y=[node number of the old node in this position]/n=0); is this node in an obstacle? (Crash: y=1/n=0); and is this the goal node? (Goal: y=1/n=0).

[Crash] = EvalCrash (Node, res) % Sub-Function of getStatus & InitDisplay

Takes the position of the new node and evaluates is the corners of a Node’s cell (based on res) are in an obstacle (Crash : y=1/n=0)

[Crash] = EvalCrash\_2 (x,y) % Sub-Function of EvalCrash

Takes the position coordinates (x,y) and evaluates if that point is in an obstacle (Crash : y=1/n=0)

[NewNodeInfo] = getNewNodeInfo (NewNode, CurrentNodeInfo, dx, dy) % Sub-Function of Move

Takes the new nodes position, the parent nodes info (CurrentNodeInfo) and the movement taken to come from the parent node to the current node (dx,dy) and generates the new nodes info NewNodeInfo = [NewNodeNo, ParentNodeNo, Cost, Steps, Cost2Come], where the new node number is 1 greater than the previous highest node number, the parent node number is exactly as it sounds, the Steps is the number of moves, “Cost2Come” is the total distance traveled from the start node (different from steps due to diagonal moves), and Cost is the “Cost2Come” + the remaining Euclidian distance to the goal node.

[Que] = UpdateQueAndDisplay (NewNode, NewNodeInfo, Que, Status, res) % Sub-Function of Move

Takes the new nodes position and “info” as well as “Status” and updates the record of expanded nodes, the Que, and the display. If the new node’s position has not been visited previously and is not in an obstacle it is added to the record, the Que, and displayed. If the new node’s position has been visited previously, but this new node has a lower cost than the old one, then the new node’s information (including the parent node number) replaces the old node’s info (except for the old nodes number) in the record of expanded nodes.

[Que] = deletechildnodes(nn,Que) % Sub-function of UpdateQueAndDisplay

Finds and removes all the children, grand-children, and great-grand-children of a node (given a node number “nn”) from the record of nodes (NodeSet) and the Que. Usually the nodes being replaced do not even have grand-children or great-grand-children, so I think it is safe to stop there.

[] = AddNode (NewNode,NewNodeInfo) % Sub-function of Astar\_Search

Updates record of expanded nodes … only actually called once in Astar\_Search (within UpdateQueAndDisplay I just write out the lines of code contained within AddNode since it is only two lines anyway.)