# An Accelerated Path Planning Approach


Esraa Morsi Ahmed
*Computer and Systems Engineering Department*
*University of Ainshames*
*cairo,Egypt*
*12476@eng.asu.edu.eg*

Hossam E. Abd El Munim
*Computer and Systems Engineering Department*
*University of Ainshames*
*cairo,Egypt*
*Hossameldin.hassan@eng.asu.edu.eg*

Hassan shehta bedour
*Computer and Systems Engineering Department*
*University of Ainshames*
*cairo,Egypt*
*hassan_bedour@eng.asu.edu.eg*



***Abstract* - Path planning is critical in robotics as well as autonomous driving applications. This research provides a modified path planning algorithm by enhancing the performance of the probabilistic roadmap (PRM) approach. The proposed technique is based on dividing the domain of motion and solves the relative path in each division. Our results on synthetic maps show a dramatic reduction on processing time compared with the conventional algorithm.**



***Keywords - Path Planning, Absolute Motion, Relative Motion, PRM (Probabilistic roadmap approach), Processing Time (PT), Path Length (PL), k.absolute (number of nodes in Absolute mode), k.relative ((number of nodes in Relative mode).***


## I .INTRODUTION

Practically, in the world we live everything is moving. There is no absolute origin nor absolute coordinates. All natural positioning systems are based on relative positions and corrections of these relative positions .First, it is necessary to explain absolute motion and relative motion, an absolute motion always moves to a taught position without being influenced by the distance from the present position. Since a relative motion is a motion move by a showed current position of the result of executing the previous motion command, the previous motion command influences the motion. Second, the need to discuss the difference between path finding, path planning and path tracking, path finding is the affirmation of each possible course; therefore, path finding depends on factors including how to get from source to goal, how to get around blocks in the route, how to find the briefest possible way and how to find the way rapidly [1]. Path planning is the art of picking which course to take and relies upon the cost, path tracking and path finding techniques worried about how to choose a speed and managing settings at every moment of time for the robot to take after a particular way, shown at fig.1. Fig.1 consists of three pictures, first is path planning; red sign refers to start state point, blue sign refers to a goal state point, and black blocks refer to obstacles. Second is path tracking; shown a vehicle, red sign refers to start state point, blue sign refers to goal state point, black curve refers to the referenced path of a vehicle, and the dotted line refers to vehicle trajectory. Third is path finding, ; shown obstacles in black blocks, red point refers to start state point, blue point refers to goal state point.Third, problems. The first problem is that powerful planners are slow.Second, the fast planner is less accurate, and more nodes need to be generated. Third, processing time in absolute mode is very huge, trying to get better results in relative motion mode, enhancing the performance of the PRM approach and apply relative positioning to get optimal results. PRM has two parameters processing time and path length, processing time is the time of adding nodes all around domain plus time of robot movement from start state point to goal state point. Path Length is a distance from start state point to goal state point, trying to solve performance problem by dividing domain to more than one domain with overlapping edges. Target here is reducing processing time from absolute path planning mode to get an optimal processing time at relative path planning mode. The importance of this problem is to get goal state at an optimal time, no wasted time by adding nodes all around the whole domain.The challenge here is reducing processing time. This research is useful in many applications like robot navigation, automation, the driverless car, robotic surgery and digital character animation and also in many fields like an army, agriculture, etc…. At 1997 Lydia E.Kavraki and Jean-Claude Latombe published research titled "Probabilistic Roadmaps for Robot Path Planning''[12], but they didn't reduce errors and processing time and used more than 2000 nodes. These nodes made processing time huge and map they work on it isn't complicate map, its sample. Also as Mika Rantanen [5], M.Kazemi and M.Mehrandezh [8] and Frank Lingelbach [11]. In this paper we introduce PRM approach, explaining of approach, PRM approach at absolute path planning mode, PRM algorithm at relative path planning mode, and the difference between absolute mode and relative mode, with different variables, as shown in in Experiments Results and validation. The remainder of this paper is organized as follows: Sect (II) provides methods of absolute and relative (PRM), and implementation of relative and absolute (PRM) algorithm. Experimental Results are presented and discussed thoroughly in Sect (III).Conclusion are given in Sect (IV).

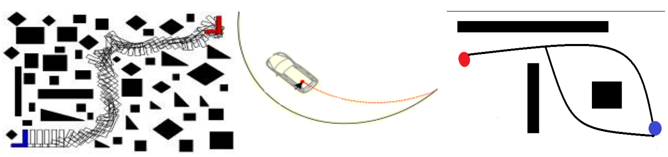

Fig.1 : (a) Path Planning , (b) path tracking , (c) path Finding

## II. METHODS

PRM approach is unequivocal geometry based planners illogical in high dimensional spaces. Exact solutions with complex geometries are provably exponential. Sampling-based planners can regularly make plans in high-dimensional spaces proficiently. We seek to clarify distinction between probabilistic complete planner approaches, complete planner approach and resolution complete planner approach.First, probabilistic complete planner approach, when a solution exists, planner will lastly find it, using random sampling [13]. Second, complete planner approach replay to a path planning query correctly in fixed time. Third, resolution complete planner approach is same as above, buts build on a deterministic sampling [6][7][8][9].

### *A. updated PRM planner algorithm at absolute mode*

PRM approach at absolute mode consists of two steps, knowing the domain and finding a trajectory.
Given as inputs, start state point (q_int), goal state point (q_goal), k.absolute (k.absol) is number of nodes at absolute mode i.e (K=100,200,300,400,500…etc), and map.When number of nodes increase processing time also increase.Given as outputs, processing time is the time of nodes distribution plus the time of calculating the distance between the start state point and goal state point by (seconds), and path length is the distance from start state point to goal state point by (meters).
The first step is knowing the domain; first, drawing a vertex (L).Second, implies collision-free path between robot configurations as decreases sharpness in new turns, increase smoothness of curves and check path correction. The second step is finding a path; first, calculate edges (q,q') using euclidean distance relation. Second, uses coarse sampling of the absolute nodes (k.absolute) and fine sampling of the edges (q,q').Third, getting processing time and path length, as shown at method.1.

### *B. PRM planner algorithm at relative mode*

PRM planner approach at relative mode consist of two operations, the first operation calculating processing time and path length from start state absolute point to goal state absolute point at map1 with overlapping regions (m1).The second operation calculating processing time and path length from start state relative mapped point to goal state relative mapped point at map2 with overlapping regions (m2). At the first operation, given as inputs, start state absolute point (q_intabsol),goalstate absolute point (q_goalabsol), (k.relative) is a number of relative nodes i.e. (K=45,60,75,90,105…etc), when a number of nodes increase processing time increase too, and map1 with overlapping regions (m1). Given as outputs, the processing time of map1(m1) is the time of nodes distribution plus the time of calculating distance between the start state absolute point and goal state absolute point by (seconds). And path length of map1 (m1) is the distance from start state absolute point to goal state point by (meters).

The first step of first operation, knowing map (m1); first, drawing a vertex (L) .Second, implies collision-free path between robot configurations as decreases sharpness in new turns, increase smoothness of curves and check path correction. The second step of first operation is finding a path; first, calculate edges (q,q') using euclidean distance relation. Second, uses coarse sampling of the nodes (k. relative) and fine sampling of the edges (q, q').On another hand the second operation is repeating these steps on relative mapped start state and goal state points using map (2) with overlapping regions (m2). And adding results of first operation to results of second operation to get total path length and total optimal processing time as shown at method 2

**Method.1: PRM planner algorithm at absolute mode**

**Inputs:** q_init: start point, q_goal: goal point, k: number of nodes, R: empty graph.
**Outputs**: processing time [PT], path length [PL], and Drawed path.

1: [L] ←vertices.
2: [q'] ← neighbor node.
3: Δ ←local planner.
4: [q, q'] ←edge.
5:q←next step.
6: R = φ.
7: [q] is randomly chosen *(q ≥ 1)*.
8: If [q] → φ then added to R (collision detection needed).
9: else none.
10: end if.
11: Repeats until [L] vertices chosen.
12: For each [q] select [k] closest neighbours.
13: Δ connects [q] to neighbor [q'].
14: If connect successful then add edge [q, q'].
15: else none.
16: end if.
17: Given [q_init] and [q_goal], connect each to the roadmap.
18: Find[k]nearest neighbors of [q_init] and [q_goal] in
19: Calculate historic cost, heuristic cost, and total cost.
20: Get PT and PL.

**Method.2: PRM planner algorithm at relative mode.**

**Inputs:** q_intabsol : start absolute point , q_goalabol: goal absolute point ,m1:as graph ,q_intrel:start relative point q_goalrel: goal relative point,k.relative:number of nodes in relative mode ,m2:as graph .
**Outputs:** total processing time [T.pt] and total path length [T.pl].

1: [L] ←vertices.
2: [q'] ← neighbor node.
3: Δ ←local planner.
4: [q, q'] ←edge.
5: [q] ←next step.
6: m1 = φ.
7: [q] is randomly chosen *(q ≥ 1)*.
8: If q → φ then added to m1 (collision detection needed).
9: else none.
10: End if.
11: Repeats until [L] vertices chosen.
12: For each [q] select k.relative closest neighbours.
13: Δ connects [q] to neighbor [q'].
19: Repeat above all previous 5 steps of method 1 on map1
20: Get PT and PL of m1.
21: Press any key to continue.
41: Repeat above all previous 20 steps of method 1 on map2
42: Press any key to continue.
43: Adding PL, PT of m1 andm2 and get T.PL and T.PT.

*C. Implementation of PRM path planning at absolute mode*

This implementation expects that the source and objective of the robot are unequivocally provided. This approach has two stages: an offline roadmap building stage and an online planning stage [2][3][14]. The point of the offline roadmap building stage is to haphazardly draw a little graph over the workspace. All vertices (L) and edges (q,q') of the graph should be collision-free. The PRM selects various number of irregular points as states in the workspace as the vertices so as to qualify to be a vertex, a haphazardly chose state point must not be inside some snag [5]. May there be k (number of nodes) which is an approach parameter. Higher are the quantity of k, better would be the outcomes with a loss of processing time. The approach at that point endeavors to interface all pairs of haphazardly chosen vertices. In the event that any two vertices can be associated by a straight line, the straight line is added as an edge as shown in figure 2.

*D. Implementation of PRM path planning at relative mode*

Dividing the domain into more than one domain (four maps); each map (500*500) pixels with overlapping regions to get relative movement target and reduce processing time [10][11]; as shown in Figure (3,4).

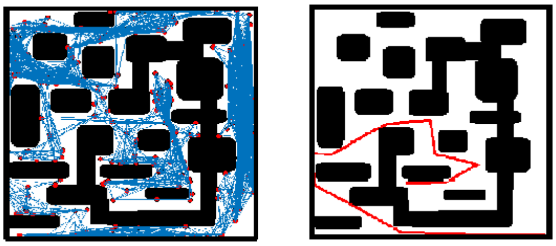

Fig. 2: PRM approach (a) roadmap (b) Path computed

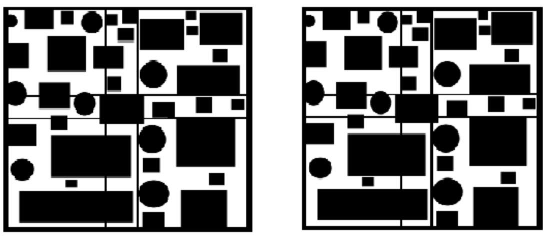

Fig.3: (a) Map (1000*1000) Pixels, (b) division domain {(1000*1000) pixels} to {(500*500) pixels} four domains with overlapping edges.

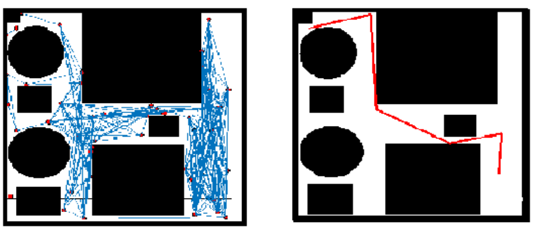

Fig.4: (a) Roadmap for map1 (m1) (500*500) pixels, (b) Path computed.

These overlapping regions have a lot of intermediate points between all four maps.These named intermediate point regions. At first map1 (m1) , we choose any start state point and goal state point, this goal state point is laying on intermediate points region also called intermediate goal point. Then getting relative start and goal state points from the second map2 (m2) as shown in figure (3, 4) and tables (1, 2, 3). These tables (1,2,3) show absolute points of (PRM) path planning at map1 (m1) and relative mapped points of (PRM) Path planning at map2 (m2) at different situations.First situation is (K)s {number of nodes} are different variables and all other parameters are constant. The second situation is input points are variable and all other parameters are constant. The third situation is output points are variable and all other parameters are constant.Parameters are input, output, k (number of nodes).

Table.1: Mapping points from absolute positioning to relative positioning at first situation ((K) s {number of nodes} are difference variables and all other paramerters are constant).

| Absolute Points (y x) | | | | Relative points(y x) | | | |
|---|---|---|---|---|---|---|---|
| S.A | E.A | S.A | E.A | S.A | E.A | S.R | E.R |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |

Table.2: Mapping points from absolute positioning to relative positioning at second situation (input points are variable and all other paramerters are constant).

| Absolute Points (y x) | | | | Relative points(y x) | | | |
|---|---|---|---|---|---|---|---|
| S.A | E.A | S.A | E.A | ST.A | E.A | S.R | E.R |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |
| 63 81 | 554 524 | 554 524 | 960 957 | 63 81 | 554 524 | 54 24 | 460 457 |
| 160 76 | 554 524 | 554 524 | 960 957 | 160 76 | 554 524 | 54 24 | 460 457 |
| 187 138 | 554 524 | 554 524 | 960 957 | 187 138 | 554 524 | 54 24 | 460 457 |
| 277 124 | 554 524 | 554 524 | 960 957 | 277 124 | 554 524 | 54 24 | 460 457 |
| 301 25 | 554 524 | 554 524 | 960 957 | 301 25 | 554 524 | 54 24 | 460 457 |
| 345 85 | 554 524 | 554 524 | 960 957 | 345 85 | 554 524 | 54 24 | 460 457 |
| 448 141 | 554 524 | 554 524 | 960 957 | 448 141 | 554 524 | 54 24 | 460 457 |

Table.3: Mapping points from absolute positioning to relative positioning at third situation (output points are variable and all other paramerters are constant).

| Absolute Points (y x) | | | | Relative points(y x) | | | |
|---|---|---|---|---|---|---|---|
| S.A | E.A | S.A | E.A | S.A | E.A | S.R | E.R |
| 24 22 | 554 524 | 554 524 | 960 957 | 24 22 | 554 524 | 54 24 | 460 457 |
| 24 22 | 554 524 | 554 524 | 9381000 | 24 22 | 554 524 | 54 24 | 438 500 |
| 24 22 | 554 524 | 554 524 | 904 675 | 24 22 | 554 524 | 54 24 | 404 175 |
| 24 22 | 554 524 | 554 524 | 818 533 | 24 22 | 554 524 | 54 24 | 318 33 |
| 24 22 | 554 524 | 554 524 | 916 687 | 24 22 | 554 524 | 54 24 | 416 187 |
| 24 22 | 554 524 | 554 524 | 745 969 | 24 22 | 554 524 | 54 24 | 245 469 |
| 24 22 | 554 524 | 554 524 | 544 987 | 24 22 | 554 524 | 54 24 | 44 487 |

## III. EXPERIMENTS RESUITS AND VALIDATION

*A. first situation (K)s {number of nodes} are different variables and all other parameters are constant. As shown in Table.5*

Relativesection:-absolute start points (st.absol) are constant as shown in column one. Absolute end points (end.absol) are constant as shown in column two. Relative start points (st.rel) are constant as shown in column three. Relative end points (end.rel) are constant as shown in column four. Relative (K)s (number of nodes) (k.rel) are variable as shown in column five. Total relative path length (t.pl.rel) as shown in column six. Total relative processing times (t.pt.rel) as shown in column seven.
Absolute section:- start points (t.st.absol) are constant as shown in column eight. Absolute end points are constant as shown in column nine.Absolute (K)s (number of nodes) (K.absol) are variables as shown in column ten. Absolute path lengths (pl.absol) as shown in column eleven. Absolute processing times (pt.absol) as shown in column twleve. It is clear that processing time decrease at relative mode and increases at an absolute mode. increasing of relative (k)s reflected on processing time in relative mode. There is a symmetry between the two curves in the form of curves, what happens in the absolute processing time increasing curve is reflected on the relative processing time decreasing curve as shown in figure 5. Path length in relative mode increase in nine states and decrease in one state. Choosing always to take the decision to decrease processing time;otherwise not important if path length is longer or not as shown in figure 6.
Calculating relative (K)s according to new maps(1,2), adding relative nodes (K.rel) according to this relation
N = k/A ; N=density of nodes , A= pixels, k=number of nodes=150,200,250,300,350,400,450,500,550,600.

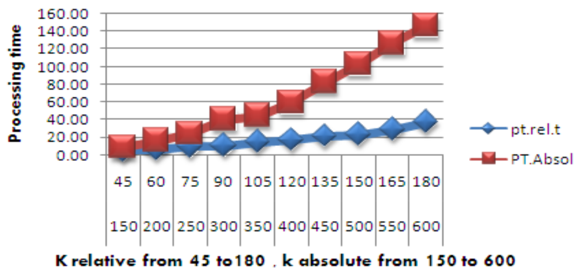


Fig.5: Relationship between processing time (vertical line from 0 to 160), consists of two curves {relative processing time (blue line) and absolute processing time (red line)}, and (k)s (number of nodes in absolute mode and relative mode)

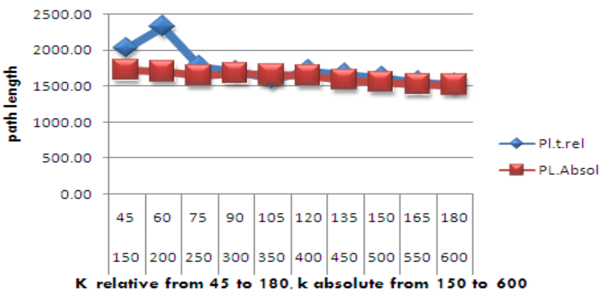


Fig.6: Relationship between the path length {relative path length (blue line) and absolute path length (red line)}, and (k)s (number of nodes in absolute state and relative state).

*B. second situation (input points are variable and all other parameters are constant), as shown in Table.7.*

Relative section:- absolute start points (st.absol)are variable as shown in column one. Absolute end points (end.absol) are constant as shown in column two.
relative start points (st.rel) are constant as shown in column three. relative end points(end.rel) are constant as shown in column four. Relative (K)s (number of nodes) (k.rel) are constant as shown in column five. total relative path length (t.pl.rel) as shown in column six. total relative processing times (t.pt.rel)as shown in column seven.
Absolute section:- absolute start points(t.st.absol) are variable as shown in column eight. Absolute end points(t.end.absol) are constant as shown in column nine. Absolute (K)s (number of nodes) (K.absol) are constant as shown in column ten. Absolute path lengths (pl.absol) as shown in column eleven. Absolute processing times (pt.absol) as shown in column twleve. It is clear, that different inputs make processing time increasing or decreasing depends on the distance between the start point and goal point at relative mode and absolute mode. Processing time decreases at relative mode and increases at absolute mode. There is symmetry between the two curves in the form of curves, what happens in absolute processing time increasing curve is reflected on the relative processing time decreasing curve as shown in figure 7. Path length in relative state increases in six states and decrease in two states, choosing always to take the decision to decrease processing time otherwise; not important if path length is longer or not, as shown in figure 8.Drawing dependent variable (path length and processing time) versus independent variable (minimum path length). Minimum path length as shown at table.4.

Table.4: Minimum path lengths in second situation (input points are variables and all other paramerters are constant).

| Start point | | End point | | results |
|---|---|---|---|---|
| X1 | Y1 | X2 | Y2 | = |
| 22 | 24 | 957 | 960 | 1322.00 |
| 81 | 63 | 957 | 960 | 1253.79 |
| 76 | 160 | 957 | 960 | 1190.03 |
| 138 | 187 | 957 | 960 | 1126.18 |
| 124 | 277 | 957 | 960 | 1077.21 |
| 25 | 301 | 957 | 960 | 1141.45 |
| 85 | 345 | 957 | 960 | 1067.06 |
| 141 | 448 | 957 | 960 | 963.33 |

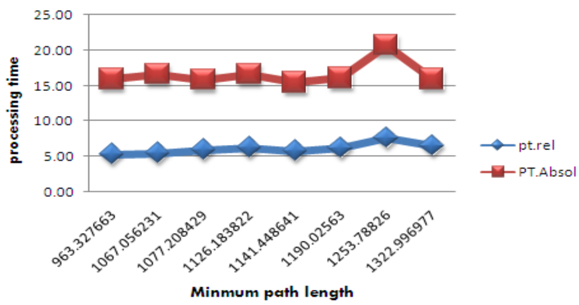


Fig.7: Relationship between processing time {absolute processing time (red line) and relative processing time (blue line)} and minimum path length

Table .5: first situation; (K) s {number of nodes} are variables and all other paramerters are constant

| | | | Relative Section | | | | | | Absolute Section | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| st.absol | end.absol | st.rel | end.rel | K.rel | t.pl.rel (m) | t.pt.rel(s) | t.st.absol | t.end.absol | K absol | pl.absol (m) | Pt.absol (s) |
| 24 22 | 554 524 | 54 24 | 460 457 | 45 | 2028.05 | 4.48 | 24 22 | 960 957 | 150 | 1717.92 | 8.73 |
| 24 22 | 554 524 | 54 24 | 460 457 | 60 | 2326.40 | 6.56 | 24 22 | 960 957 | 200 | 1697.53 | 15.90 |
| 24 22 | 554 524 | 54 24 | 460 457 | 75 | 1776.73 | 9.57 | 24 22 | 960 957 | 250 | 1642.81 | 23.92 |
| 24 22 | 554 524 | 54 24 | 460 457 | 90 | 1708.17 | 10.35 | 24 22 | 960 957 | 300 | 1672.95 | 40.24 |
| 24 22 | 554 524 | 54 24 | 460 457 | 105 | 1601.27 | 14.79 | 24 22 | 960 957 | 350 | 1653.38 | 44.27 |
| 24 22 | 554 524 | 54 24 | 460 457 | 120 | 1717.63 | 16.92 | 24 22 | 960 957 | 400 | 1641.59 | 58.86 |
| 24 22 | 554 524 | 54 24 | 460 457 | 135 | 1658.00 | 21.42 | 24 22 | 960 957 | 450 | 1585.97 | 82.27 |
| 24 22 | 554 524 | 54 24 | 460 457 | 150 | 1620.85 | 23.40 | 24 22 | 960 957 | 500 | 1555.69 | 102.58 |
| 24 22 | 554 524 | 54 24 | 460 457 | 165 | 1558.60 | 28.70 | 24 22 | 960 957 | 550 | 1519.87 | 125.99 |
| 24 22 | 554 524 | 54 24 | 460 457 | 180 | 1526.44 | 37.00 | 24 22 | 960 957 | 600 | 1512.24 | 146.34 |

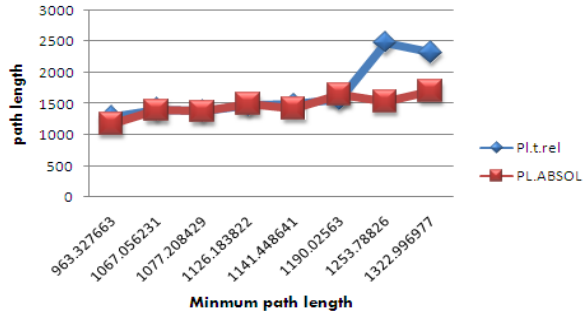


Fig.8: Relationship between path length {relative (blue line) and absolute (red line)} and minimum path length.

*C. Third situation is output points are variable and all other parameters are constant.,as shown in Table.8 .*

Relative section:-absolute start points (st.absol) are variable as shown in column one.Absolute endpoints (end.absol) are constant as shown in column two. Relative start points (st.rel) are variable as shown in column three. relative endpoints (end.rel) are variable as shown in column four. Relative (K)s (number of nodes)(k.rel) are constant as shown at column five. total relative path length (t.pl.rel) as shown at column six. total relative processing times (t.pt.rel) as shown at column seven. Absolute section:- absolute start points (t.st.absol) are constant as shown in column eight. Absolute end points(t.end.absol) are variable as shown in column nine .Absolute {(K)s (number of nodes )} (K.absol) are constant as shown in column ten. Absolute path length (pl.absol) as shown in column eleven.Absolute processing time (pt.absol) as shown in column twleve. It is clear, that different outputs make processing time increase or decrease depends on the distance between the goal point and start point at relative mode and absolute mode. There is a symmetry between the two curves in the form of curves, what happens in absolute processing time increasing curve is reflected on the relative processing time decreasing curve shown in figure 9.

Path length in relative state increases in seven states and does not decrease in any state, choosing always to take the decision to decrease processing time otherwise; not important if path length is longer or not,as shown in figure 10. Drawing dependent variable (path length and processing time) versus independent variable (minimum path length) as shown at table 6.

Table.6: Minimum path lengths in third situation (output points are variables and all other paramerters are constant).

| Start point | | End point | | Results |
|---|---|---|---|---|
| X1 | Y1 | X2 | Y2 | = |
| 22 | 24 | 957 | 960 | 1323.00 |
| 22 | 24 | 1000 | 938 | 1338.61 |
| 22 | 24 | 675 | 904 | 1095.81 |
| 22 | 24 | 533 | 818 | 944.22 |
| 22 | 24 | 687 | 916 | 1112.60 |
| 22 | 24 | 969 | 745 | 1190.23 |
| 22 | 24 | 987 | 544 | 1096.19 |

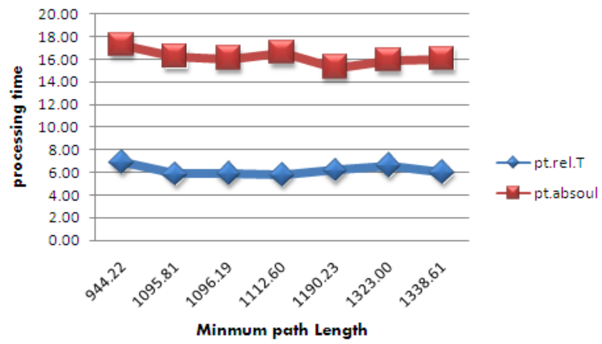


Fig .9: Relationship between processing time {absolute processing time (redline) and relative processing time (blue line)} and minimum path length.

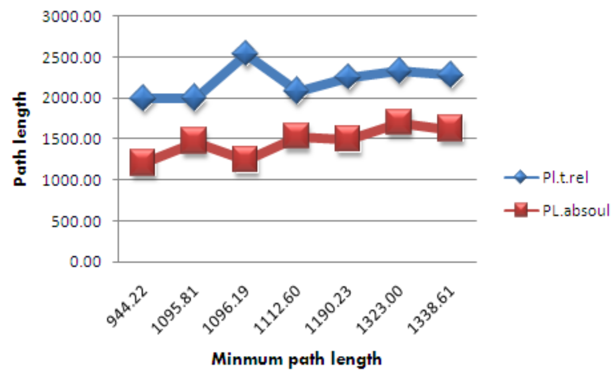


Fig.10: Relationship between path length {relative (blue line) and absolute (red line)} and minimum path length.

Table.7: second situation (input points are variable and all other paramerters are constant).

| Relative section | | | | | | | Absolute Section | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **st.absol** | **end.absol** | **st.rel** | **end.rel** | **K.rel** | **t.pl.rel (m)** | **t.pt.rel (S)** | **t.st.absol** | **t.end.absol** | **K absol** | **pl.absol (m)** | **Pt.absol (s)** |
| 24 22 | 554 524 | 54 24 | 460 457 | 60 | 2326.40 | 6.56 | 24 22 | 960 957 | 200 | 1697.53 | 15.90 |
| 24 22 | 554 524 | 54 24 | 438 500 | 60 | 2271.78 | 5.94 | 24 22 | 938 1000 | 200 | 1617.90 | 16.02 |
| 24 22 | 554 524 | 54 24 | 404 175 | 60 | 1994.97 | 5.81 | 24 22 | 904 675 | 200 | 1464.30 | 16.21 |
| 24 22 | 554 524 | 54 24 | 318 33 | 60 | 1987.58 | 6.86 | 24 22 | 818 533 | 200 | 1199.49 | 17.25 |
| 24 22 | 554 524 | 54 24 | 416 187 | 60 | 2077.76 | 5.73 | 24 22 | 916 687 | 200 | 1525.64 | 16.57 |
| 24 22 | 554 524 | 54 24 | 245 469 | 60 | 2252.40 | 6.15 | 24 22 | 745 969 | 200 | 1488.56 | 15.26 |
| 24 22 | 554 524 | 54 24 | 44 487 | 60 | 2536.62 | 5.85 | 24 22 | 544 987 | 200 | 1235.52 | 16.05 |

Table.8: third situation (output points are variable and all other paramerters are constant).

| Relative section | | | | | | | Absolute Section | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **st.absol** | **end.absol** | **st.rel** | **end.rel** | **K.rel** | **t.pl.rel (m)** | **t.pt.rel (S)** | **t.st.absol** | **t.end.absol** | **K absol** | **pl.absol (m)** | **Pt.absol (s)** |
| 24 22 | 554 524 | 54 24 | 460 457 | 60 | 2,323.00 | 6.56 | 24 22 | 960 957 | 200 | 1,700.00 | 15.90 |
| 63 81 | 554 524 | 54 24 | 460 457 | 60 | 2,483.00 | 7.64 | 63 81 | 960 957 | 200 | 1,530.00 | 20.70 |
| 160 76 | 554 524 | 54 24 | 460 457 | 60 | 1,566.00 | 6.20 | 160 76 | 960 957 | 200 | 1,640.00 | 16.00 |
| 187 138 | 554 524 | 54 24 | 460 457 | 60 | 1,470.00 | 6.29 | 187 138 | 960 957 | 200 | 1,490.00 | 16.60 |
| 277 124 | 554 524 | 54 24 | 460 457 | 60 | 1,380.00 | 5.94 | 277 124 | 960 957 | 200 | 1,370.00 | 15.80 |
| 301 25 | 554 524 | 54 24 | 460 457 | 60 | 1,484.00 | 5.85 | 301 25 | 960 957 | 200 | 1,410.00 | 15.40 |
| 345 85 | 554 524 | 54 24 | 460 457 | 60 | 1,403.00 | 5.50 | 345 85 | 960 957 | 200 | 1,390.00 | 16.60 |
| 448 141 | 554 524 | 54 24 | 460 457 | 60 | 1,284.00 | 5.32 | 448 141 | 960 957 | 200 | 1,170.00 | 15.90 |

## IV. CONCLUSION

In this paper, we have demonstrated the absolute and relative PRM algorithms for path planning in dynamic environments. Our modifications enhanced the processing time especially in the corners and u-turns to have smooth Paths. The proposed algorithms avoid collision by redistribution of nodes to include the source and target positions. The path is checked priori to assure avoiding any obstacle. Our experimental results are demonstrated for different maps and showed the efficiency and effectiveness of the proposed approaches.

## REFERENCES


[1] Steven M. LaValle, "planning algorithms", first Edition, published by "Cambridge University Press", first published (2006).

[2] Dennis Nieuwenhuisen and Mark H. Overmars, "Useful Cycles in Probabilistic Roadmap Graphs", published by "institute of information and computing sciences, Utrecht University technical", first published (December 2004).

[3] Thierry Siméon, Jean-Paul Laumond, Juan Cortés, and Anis Sahbani, "Manipulation Planning with Probabilistic Roadmaps", published by "The International Journal of Robotics Research ", first published (August 2004).

[4] Roland Geraerts, "Sampling and node adding in probabilistic roadmap planners", published by "Utrecht University Institute of Information and Computing Sciences", first published (December 2005).

[5] Mika Rantanen, "Improving Probabilistic Roadmap Methods for Fast Motion Planning", published by "Tampere University Press and the author", first published (2014).

[6] Roland Geraerts and Mark H Overmars, "A Comparative Study of Probabilistic Roadmap Planners", published by "Institute of Information and Computing Sciences Utrecht University," first published (2004).

[7] M. Kazemi and M. Mehrandezh, "Robotic Navigation Using Harmonic Function-based Probabilistic Roadmap", published "by IEEE International Conference on Robotics 8 Automation New Orleans", first published (April 2004).

[8] Khaled Belghith, Froduald Kabanza, Leo Hartman and Roger Nkambou, "Anytime Dynamic Path-Planning with Flexible Probabilistic Roadmaps", published by "Government of Canada", first published (2006).

[9] Khaled Belghith, Froduald Kabanza, Leo Hartman and Roger Nkambou, "Anytime Dynamic Path-Planning with Flexible Probabilistic Roadmaps", published by "Government of Canada", first published (2006).

[10] Patrycja E. Missiuro and Nicholas Roy, "Adapting Probabilistic Roadmaps to Handle Uncertain Maps", published by "MIT Computer Science and Artificial Intelligence", (June 2006).

[11] Frank Lingelbach, "Path Planning using Probabilistic Cell Decomposition", first published (2005).

[12] Lydia E.Kavraki and J-c Latombe,"Probabilistic Roadmaps for Robot Path Planning", published by "John Wiley & Sons", first published (1997).

[13] Stiller, "A Probabilistic Roadmap Planner with Sampling on the Medial Axis of the Free Space", Published by "Department of Computer Science, Texas A&M University", First published (November 23, 1998).

[14] T.Simeon, J-P. Lamound and C. Nissoux, "Visibility-based PRM for motion planning Advanced Robotics", published By "VSP and Robotics Society of Japan", first published (April 2000).

[15] P. Svestka and M.H. Overmars, This is the fifth chapter of the book "Robot Motion Planning and Control"[Jean-Paul Laumond (Editor)] published by "Springer", first published (1998).