RRTs Review and Statistical Analysis

Ahmad Abbadi and Radomil Matousek

***Abstract*—** Path planning is one of the important issues in robotics area. There are many ideas to deal with this issue one of them is RRT (Rapidly Exploring Random Tree). This method is not optimal but it reduces the time needed for obtaining solutions. This algorithm is based on portability, the result of this algorithm is a tortuous path which has a lot of useless points. In this paper we introduce some variants of RRTs and a method for reduce a degree of tortuous, making the path shorter and omitting useless points. Also because of RRT's randomizes we make some statistical test on many variations of RRT, to make decisions about the best variations.

***Keywords*—** RRT, Rapidly Exploring Random Tree, Path Planning, Path optimizing, RRT Statistic**.**

I. Introduction

Path planning is an important issue in robotic movements, it focuses on finding the way between two points in given space. Many methods was introduced like Potential Field, Neural Network, and heuristic algorithm like A* algorithm and evolutionary algorithms like GA (Genetic algorithm) and Ant Colony system algorithm (ACS) [14]. In this paper we will talk about RRT (Rapidly Exploring Random Tree) which is a successful method applied in this area, it depends on randomized approaches on its work, there are many variations of this method [1], and we talk briefly about them in chapter 2. In this paper we test many variations of RRT using 4 different types of space all of them have 2D dimensions. We developed also an algorithm to shorten the path between the X_{init} point and the X_{target} point. We also made some statistical analysis on test results to ensure and support our contentment about them.

II. RRT (Rapidly Exploring Random Tree)

RRT was introduced as planning algorithm for rapidly exploring spaces in two or more dimensions [7], with either holonomic or nonholonomic movements, by considering kinematics and movement constraints or not. Using this algorithm we can model the movement of a car, a robot or any other moving machines. RRT is not optimal way because it depends on probability of choosing path points.

Manuscript received August 3, 2011: Revised version received December 27, 2011. This work was supported by the research projects of MSM 0021630529, GACR No.: 102/09/1680 " Control Algorithm Design Using Evolutionary Approach", and IGA FSI-S-11-31 "Application of AI".

All authors are from Brno University of Technology, Faculty of Mechanical Engineering, Dept. of Applied Computer Science, Technicka 2896/2, 61600 Brno, Czech Republic, corresponding authors have e-mails: abbadi83@gmail.com, matousek@fme.vutbr.cz.

The tree starts from X_{init} point and tries to reach X_{target} point (Fig. 1.). It selects a point X_{rnd} of the space randomly, then it chooses the nearest point of the tree X_{near} to X_{rnd}, after that it tries to extend a branch from X_{near} to the X_{rnd} by length of ε to get X_{new}, if there are no obstacles in the way to X_{new} we considers it as new point of the tree. We repeat those steps many times until X_{new} is the same as X_{target} or very near to it.

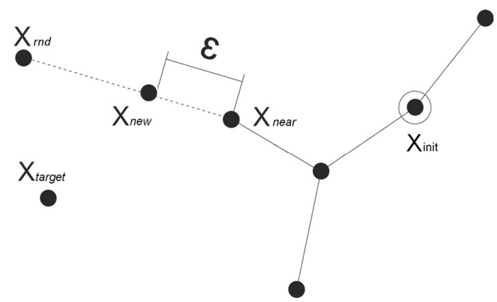

Fig. 1 The principle of RRT expanding by ε toward X_{rnd}.

There have been introduced and developed many variants of this algorithm to make it much faster and suitable for many types of environments. In [1] are listed some RRT variations with more detail in explanations. One of RRT variations depending on bias to X_{target} by choosing X_{target} point instead of X_{rnd} and try to extend to it (see Fig. 2.), we choose X_{target} in p probability and other points in 1-p probability, usually p is small like 5% to avoid the local minimums [4].

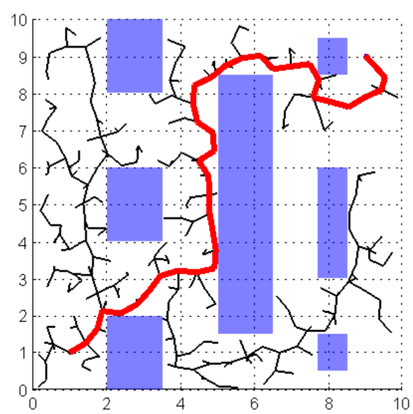

Fig. 2 Bias tree to the goal point.

The second variation depends on extending the tree to X_{rnd} directly if it is possible and this variation is called Connect [5]. Another variation is Vlrrt (Variable Length RRT) which means we store the length ε on every point of the tree and use this value when we want to make a new branch from this point.

The value of ε changes from point to point depending on the obstacle; if the extension fails because of obstacles then we decrease ε value, else we increase it and store the new value with X_{new} information form next extensions [2]. The other variation is similar to the pervious one it's called Dvlrrt (Directional Vlrrt) which means the increase or decrease of ε value depends on the direction of the obstacles not only the existence of them beside the tree points [2].

Another approach developed RRT is based on the number of trees. Bidirectional trees try to grow two trees, one from X_{init} and the other from X_{target}. (See Fig. 3.) The two trees try to connect to each other; we may grow the trees with bias to other trees branches in p probability or without bias [4].

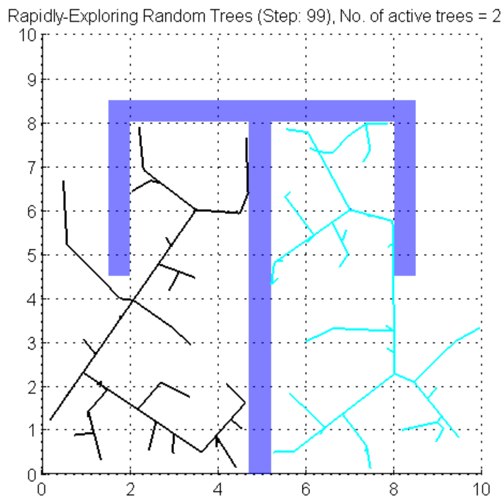

Fig. 3 Connect method with 2 trees.

Many local trees is like pervious variation of RRT, but instead of using two trees from the start and goal we use many trees two from the start and goal points and other trees grows from random point in the space [6].

RRT-Blossom is a variation of RRT which suppose that the X_{new} must be far from already existing points of the tree, so the new branch is forced to grow to unexplored space and the advantage is decrease in number of useless branches [3].

III. Shortening the Path

After we apply RRT we get tree path which is tortuous path and has a lot of points, we introduce an algorithm to make the path shorter in length and as straightforward as possible by trying to omit useless points from it (see Fig. 4.).

Initially End_{pnt} is last point in the path –could be X_{target}-. And $Start_{pnt}$ is first point in the path –could be X_{init} -.

The algorithm (Fig. 5) tries to connect the End_{pnt} with $Start_{pnt}$ in the path. If there are collisions with obstacles, we put $Start_{pnt} = Start_{pnt} + 1$ (next point in the path), then we try again to connect it with End_{pnt}, when the connection is successful we delete the points between End_{pnt} and $Start_{pnt}$ from the original path, then we put $End_{pnt} = End_{pnt}$ -1 and repeat above steps, until $End_{pnt} = 2$.

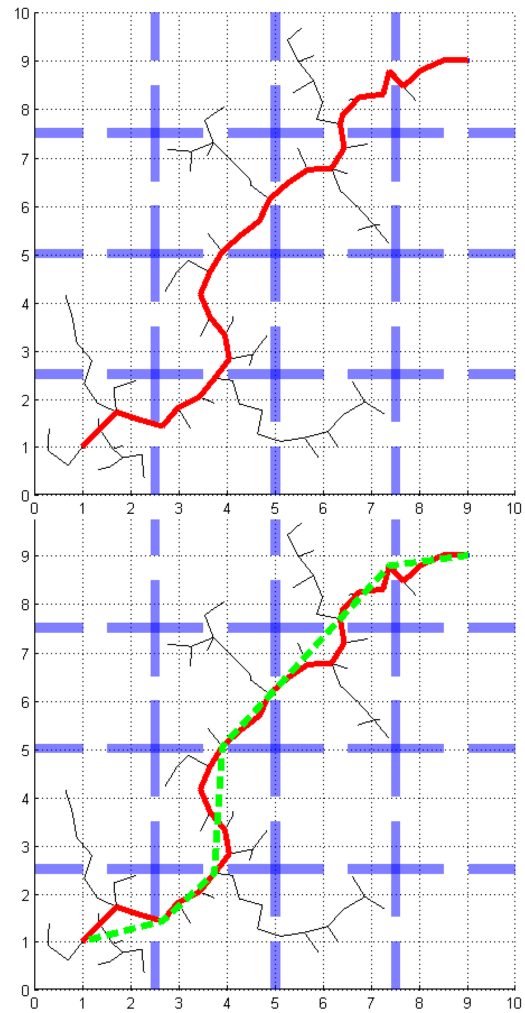

Fig. 4 (above): original RRT Path (38 point, Len =18.13), (bottom): shortened path (6 point, Len= 14.2).

```
Endpnt =index of last point in the path;
Startpnt =1;
while (Endpnt  ~= 2 )
        pnt2 = path( Endpnt );
        for ( Startpnt =1 ; Startpnt < Endpnt -1 ; Startpnt ++ )
                pnt1 = path( Startpnt );
                if ~collisionCheck(p1,p2)
                        path = path(1 to Startpnt )
                                + path( Endpnt to the end);
                        Endpnt = index of previous point to Endpnt;
                        break;
                end
        end
end
```

Fig. 5 Short path algorithm.

After generating *path* by RRT function *collisionCheck()* is used to check if there are collisions between obstacles and the line from point *pnt1* to point *pnt2*.

Because this algorithm is based on the original path - generated by RRT - it is not optimal. The result is a path with fewer points. Fig. 4 shows the original path generated by the RRT (a) and Shortened path (b) generated by this algorithm.

IV. Testing results

We made tests for 13 RRT variations on 4 spaces Fig. 6. The first space is low density of obstacles (a). The second is T-trap obstacle (b); in (c) high density of obstacle and the last obstacles are doors (d).

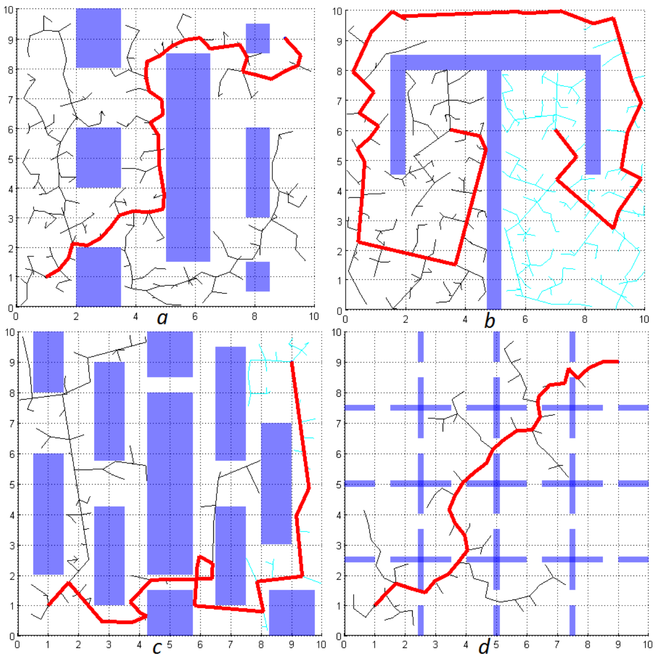

Fig. 6 Testing space for RRT variation.
a: low density space, b: trip space, c: high density, d : doors obstacles.

The test is applied on every space separately; we test 13 variations of RRT 100 times for each. We suppose the fail occurs when RRT variation tried to extend a branch 2000 times without reaching the goal. The extension length $\varepsilon = 0.5$, the limit of each space is [10, 10]. We have used PC equipped with 2.5 GHz Core2Duo CPU, 2 GB RAM.

The implementation of RRT variation is in Matlab and the statistical result is in Minitab.

The tests are based on time, success rate of reaching the goal and path length. We focus on path length result because time testing depends on the power of hardware and algorithms implementation, the best implementation, the best result. Time tests could be useful for low power hardware.

A. Time ration of RRT variations

The tests results show that the best variation in Low obstacles space is Vlrrt(2) the mean time of reaching the goal is 0.0467 second and the median is 0.0418, the second best variation is Dvlrrt(2) with 0.0484 second in mean and 0.0407 in median. Table 1 shows the result for all testing results in low obstacle space and Fig. 7 show boxplots.

Table 1: Tests results of the low density of obstacle.

Variation	Mean	StDev	Variance	Median	Success
Bias	0.1035	0.0484	0.0023	0.0890	100
Blossom	0.3552	0.2584	0.0668	0.2714	94
Blossom(2)	0.0615	0.0255	0.0007	0.0564	100
Con	0.3434	0.2546	0.0648	0.2526	93
Con(2)	0.0578	0.0198	0.0004	0.0559	100
ConExt	0.0617	0.0202	0.0004	0.0585	100
Dvlrrt	0.0893	0.0493	0.0024	0.0734	100
Dvlrrt(2)	**0.0484**	0.0259	0.0007	**0.0407**	100
Ext	0.2806	0.1991	0.0396	0.2380	95
Ext(2)	0.0516	0.0249	0.0006	0.0444	100
ExtCon	0.0637	0.0234	0.0006	0.0621	100
Vlrrt	0.0840	0.0436	0.0019	0.0698	100
Vlrrt(2)	***0.0467**	***0.01754**	***0.0003**	***0.0418**	100

** is the best result*

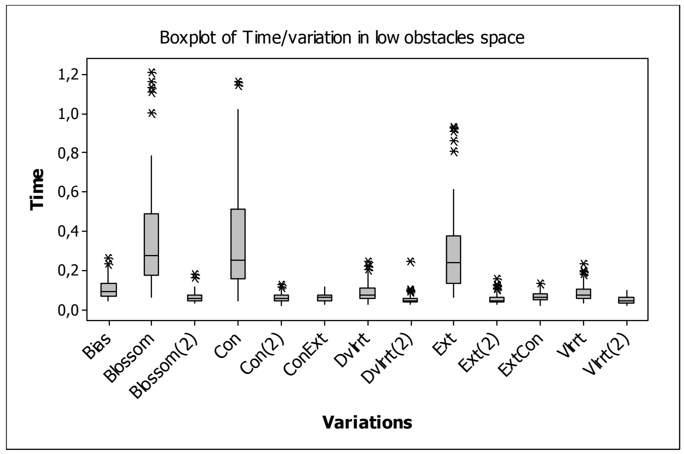

Fig. 7 Tests results of in the low density obstacles.

In the T obstacle the best result is Vlrrt but with one fail of reaching the goal. Mean is 0.3740 and the median is 0.3713. The second best result is Vlrrt(2) with mean 0.3984 and median 0.3849 without a fail. We make some statistical test on the best results to ensure that if we use Vlrrt(2) – the second best variation without a fail - it would give the same result in confidence level 95%, Fig. 8. Show testing hypothesis.

```
Two-sample T for Vlrrt vs Vlrrt(2)

              N    Mean    StDev   SE Mean
Vlrrt        99  0.3740   0.0984    0.0099
Vlrrt(2)    100   0.398    0.122     0.012

Difference = mu (Vlrrt) - mu (Vlrrt(2))
Estimate for difference:  -0.0244
95% CI for difference:  (-0.0554; 0.0067)
T-Test of difference = 0
(vs not =): T-Value = -1.55
P-Value = 0.123   DF = 189
```

Fig. 8 T-test for Hypothesis "Vlrrt and Vlrrt(2) not equal" in T.

We can infer from the P-Value which is >5% that there is no sufficient difference between the two variations, and we can use Vlrrt(2) without fail in result instead of Vlrrt , by keeping the same result in confidence level 95%. Fig. 9 shows boxplot for the results.

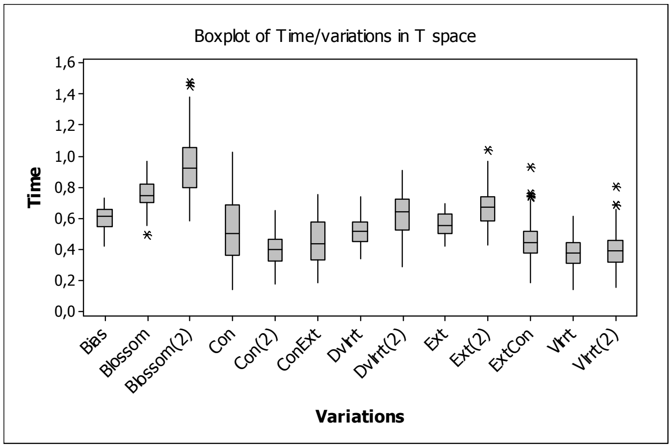

Fig. 9 Tests results in T obstacle.

In high obstacle the Con(2) is the best variation the mean is 0.1871 and the median is 0.1844. Fig. 11 shows boxplot for results. By statistical tests we get *P-Value* >5% in T-test, so there is no sufficient difference in using Con(2) the best, or Vlrrt(2) the 3rd best (Fig. 10) in confidence level 95%.

```
             N    Mean   StDev  SE Mean
Con(2)     100  0.1871  0.0712   0.0071
Vlrrt(2)   100  0.2072  0.0837   0.0084

Difference = mu (Con(2)) - mu (Vlrrt(2))
Estimate for difference:  -0.0201
95% CI for difference:  (-0.0418; 0.0015)
T-Test of difference = 0
(vs not =): T-Value = -1.83
P-Value = 0.068  DF = 193
```

Fig. 10 T-test for Hypothesis "Con(2) and Vlrrt(2) not equal" in high.

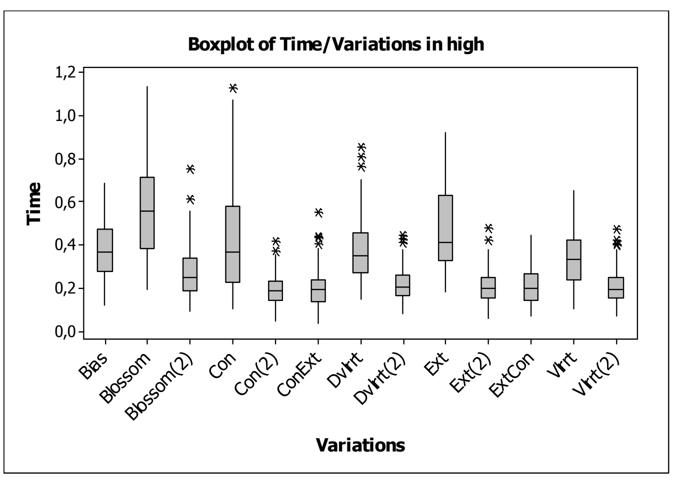

Fig. 11 Tests results in high density obstacles.

For the last space "Doors obstacle", the best variations is Dvlrrt(2) with mean 0.2961 and median 0.2623. Fig. 13 shows the boxplot for all variations. We make a T-test for the hypothesis which assume that there is difference between the best variation Dvlrrt(2) and the 2nd best variation Vlrrt(2) Fig.12.

```
             N   Mean  StDev  SE Mean
Dvlrrt(2)  100  0.296  0.148    0.015
Vlrrt(2)   100  0.317  0.198    0.020

Difference = mu (Dvlrrt(2)) - mu (Vlrrt(2))
Estimate for difference:  -0.0213
95% CI for difference:  (-0.0702; 0.0275)
T-Test of difference = 0
(vs not =): T-Value = -0.86
P-Value = 0.390  DF = 183
```

Fig. 12 T-test for Hypothesis "Dvlrrt(2) and Vlrrt(2) not equal" in doors obstacle

Based on this test we reject the hypothesis, because we get *P-Value*> 0.05 which mean there is no sufficient difference between the two best variations.

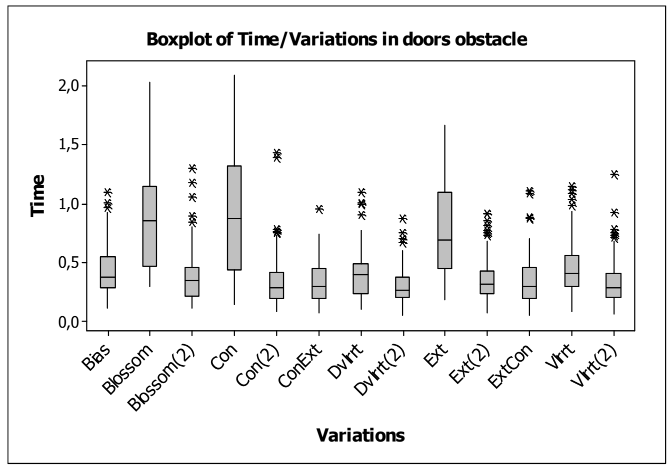

Fig. 13 Tests results in the doors obstacles.

Form the last result and statistical analysis we can use Vlrrt(2) in all spaces without sufficient difference between it and the best variations in each space in confidence level 95%.

Some of variations didn't reach the goal in our limitations so we consider this a fail. From the tests we can get idea about the fail in variations. Next chapter discuss this situation.

B. Successful of Tests

The tests show some variations fail to reach the goal in the obstacles spaces. Table 2 shows success rate of reaching the goal in 100 iterations of each variation in the four spaces. In each iteration we try to grow the trees 2000 times and we consider a fail when the tree didn't reach the goal within this limit. Table 2 shows the successful result of these tests.

Depending on the result we can see that the fail accrued just in one directional algorithm, the bidirectional algorithms didn't have any fail. (Fig. 14, Fig. 15, Fig. 16, Fig. 17) shows the unsuccessful in obstacle spaces.

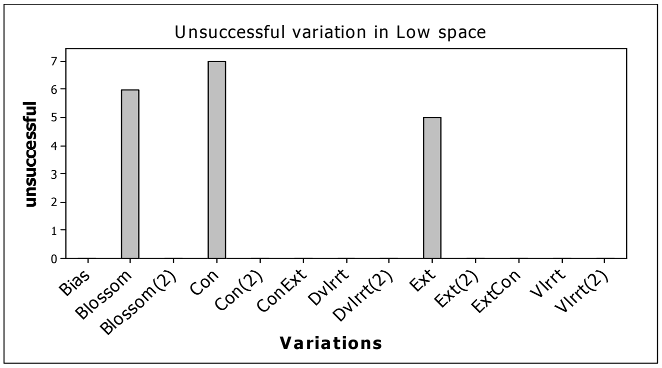

Fig. 14 Unsuccessful results in the low density obstacles.

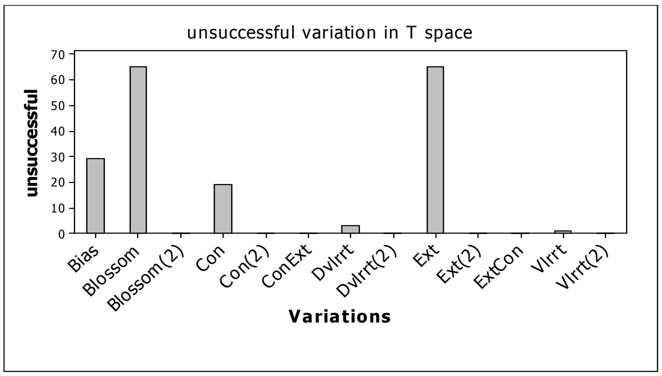

Fig. 15 Unsuccessful results in the T obstacle.

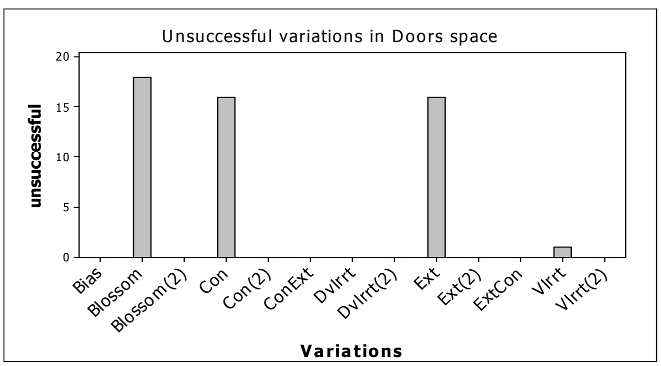

Fig. 16 Unsuccessful results in the doors obstacles.

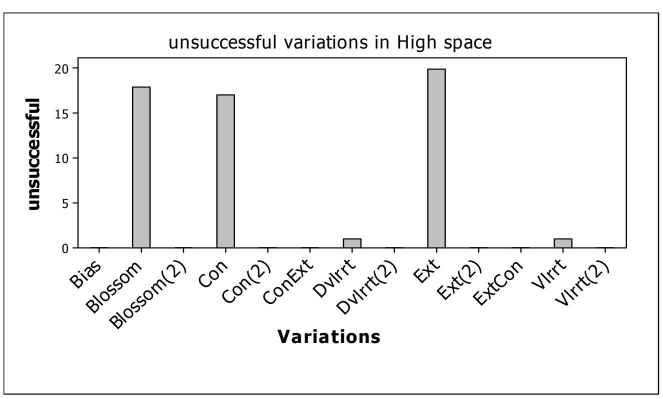

Fig. 17 Unsuccessful results in the high density obstacles.

Table 2 Successful results of the variations in obstacles spaces.

	Low	**T**	**High**	**Doors**
Bias	100	71	100	100
Blossom	94	35	82	82
Blossom(2)	100	100	100	100
Con	93	81	83	84
Con(2)	100	100	100	100
ConExt	100	100	100	100
Dvlrrt	100	97	99	100
Dvlrrt(2)	100	100	100	100
Ext	95	35	80	84
Ext(2)	100	100	100	100
ExtCon	100	100	100	100
Vlrrt	100	99	99	99
Vlrrt(2)	100	100	100	100

From the last two chapters we infer that the bidirectional variations can find a solution in every tested environments and execution time is much faster than the unidirectional variations.

C. Path Tests

As we have mentioned before the time ration results was variable because of algorithms' implementations. So we test the variations for the path length.

The tests results are shown in Table 3, Table 4, Table 5 and Table 6 for low, T, High and Doors obstacles respectively. Also the boxplot for RRT variations on every space is presented for every case.

Table 3: Path length results on the low density of obstacle.

	Path Median	Path Min	S-path Median	S-Path Min	Rate %
Bias	***14.336**	12.770	11.856	11.478	17.30
Blossom	15.373	13.316	11.817	**11.471**	23.13
Blossom(2)	14.644	12.947	11.787	11.490	19.51
Con	17.359	13.366	14.085	11.553	18.86
Con(2)	18.880	12.210	14.195	11.532	24.81
ConExt	16.534	12.796	12.010	11.556	27.36
Dvlrrt	14.540	12.444	11.834	11.504	18.61
Dvlrrt(2)	14.773	12.808	11.862	11.499	19.70
Ext	15.189	13.235	11.831	11.473	22.11
Ext(2)	14.604	13.155	***11.810**	11.496	19.13
ExtCon	16.929	***12.062**	12.058	11.530	***28.77**
Vlrrt	14.846	12.565	11.946	11.565	19.53
Vlrrt(2)	**14.545**	12.629	11.846	11.476	18.56

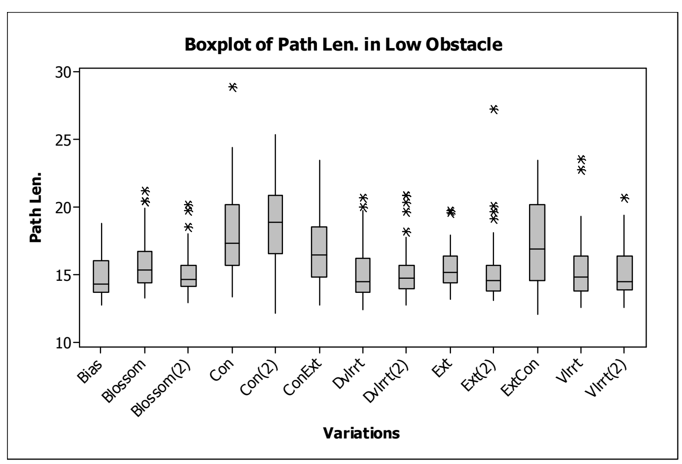

Fig. 18 Path length Boxplot on the low density of obstacle.

In Table 3 we see the best results are for Bias variation. This variation has unidirectional tree. So we try to statically test for hypnosis which assumes that there is no sufficient deference between the Bias variation and the best bidirectional variation, in order to get a variation which we can use and benefit from its advantages (speed, completeness).

```
Two-sample T for bias-PLen vs Blossom2-PLen
           N   Mean  StDev  SE Mean
b-PLen    100  14.85   1.51     0.15
Bl2-PLen  100  15.03   1.37     0.14

Difference = mu (b-PLen) - mu (Bl-PLen)
Estimate for difference:  -0.176
95% CI for difference:  (-0.579; 0.226)
T-Test of difference = 0 (vs not =):
T-Value = -0.86  P-Value = 0.388  DF = 196
```

Fig 19.T test for testing hypothesis length of path for “blossom(2) and Bias variations are not equal”.

For the statistical test P-value = 0.388> 5%, so we reject the hypothesis (blossom(2) and Bias not equal) and we infer that we can use blossom(2) variation instead of bias variation in low obstacle space. Also for the short path the best variation is Ext(2).

For the T obstacle the best result is blossom, we try to find another bidirectional variation with similar result in confidence level 95%. Table 4 show the testing result. And Fig. 21 shows the boxplot for these results.

Table 4: Path length on the T obstacle.

	Path Median	Path Min	S-Path Median	S-Path Min	Rate %
Bias	30.124	25.814	***24.089**	22.309	20.03
Blossom	***29.976**	26.635	24.488	22.191	18.31
Blossom(2)	**32.166**	27.269	**24.498**	22.643	***23.84**
Con	33.658	26.281	25.934	22.277	22.95
Con(2)	33.618	26.742	25.831	22.462	23.16
ConExt	33.170	27.120	25.427	21.740	23.34
Dvlrrt	31.148	25.324	25.104	22.522	19.40
Dvlrrt(2)	33.177	26.489	25.831	22.367	22.14
Ext	30.385	26.432	23.908	22.491	21.32
Ext(2)	32.562	26.460	25.091	22.127	22.94
ExtCon	32.815	25.764	25.378	22.050	22.66
Vlrrt	32.041	***25.318**	25.906	22.534	19.15
Vlrrt(2)	33.436	28.006	26.165	23.233	21.75

We make a T-test for hypothesis the best variation “Blossom” and the bidirectional variation Blossom(2) are not equal. Fig. 20 shows the result for this test.

```
Two-sample T for Bl-PLen vs Bl2-PLen
            N   Mean  StDev  SE Mean
Bl-PLen    35  30.30   2.14     0.36
Bl2-PLen  100  32.21   2.53     0.25

Difference = mu (Bl-PLen) - mu (Bl2-PLen)
Estimate for difference:  -1.917
95% CI for difference:  (-2.796; -1.038)
T-Test of difference = 0 (vs not =):
T-Value = -4.35  P-Value = 0.000  DF = 69
```

Fig 20.T test for testing hypothesis length of path for “blossom(2) and blossom variations are not equal” in T space

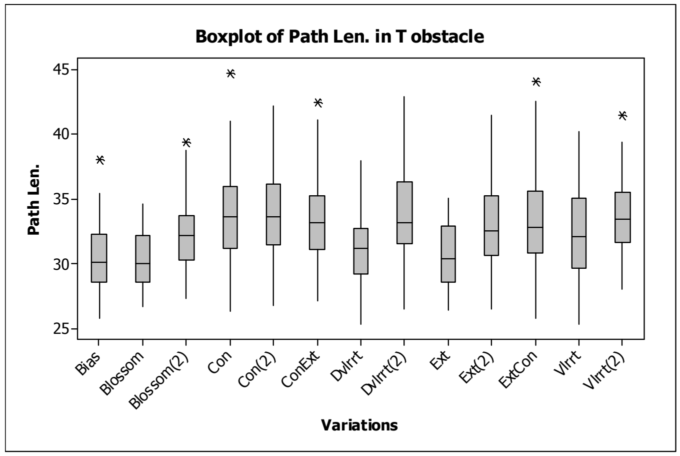

Fig. 21. Path length Boxplot on the T obstacle.

For the T-test result in Fig. 20 we infer that the hypothesis is accepted and there is a difference in the results between the two variations. But depending on the short path and the testing between the best variation Bias and the best bidirectional variation Blossom(2) we see that there is no sufficient difference between the two variations Fig. 22.

```
Two-sample T for Bias-SP vs Bl-SP
          N   Mean  StDev  SE Mean
Bias-SP   71  24.43   1.45     0.17
Bl-SP    100  24.76   1.30     0.13
Difference = mu (Bias-SP) - mu (Bl-SP)
Estimate for difference:  -0.328
95% CI for difference:  (-0.753; 0.098)
T-Test of difference = 0 (vs not =):
T-Value = -1.52  P-Value = 0.130  DF = 140
```

Fig 22.T test for hypothesis that short path in blossom(2) and bias variations are not equal in T space

In the high obstacle space we test the variations and get the result in Table 5. The boxplot figure is shown in Fig. 24.

Table 5: Path length of the high density of obstacle.

	Path Median	Path Min	S-Path Median	S-Path Min	Rate %
Bias	17.911	***14.233**	15.045	13.269	16.00
Blossom	17.766	15.349	***14.70**	13.353	***17.26**
Blossom(2)	**19.879**	14.977	16.716	13.321	15.91
Con	19.004	15.260	16.697	13.252	12.14
Con(2)	21.144	15.253	17.541	13.498	17.04
ConExt	21.363	15.079	18.164	13.191	14.97
Dvlrrt	18.263	14.415	**16.185**	13.236	11.38
Dvlrrt(2)	20.350	15.087	17.376	13.479	14.61
Ext	17.752	15.542	14.702	13.286	17.18
Ext(2)	19.947	15.328	17.277	13.222	13.39
ExtCon	20.535	14.613	17.109	13.393	16.68
Vlrrt	***17.528**	14.846	14.919	13.347	14.88
Vlrrt(2)	20.022	14.730	16.832	13.412	15.93

We make some analytical tests in order to find bidirectional variation we can use instead of unidirectional variation, because as we listed before, the bidirectional is much faster than unidirectional and it have found solution in all tested environments. Fig. 23 shows T-test for hypothesis “blossom(2) and Vlrrt variations are not equal in high space”, we infer from this test that the hypothesis is accepted and we can’t use blossom(2) instead of Vlrrt.

```
Two-sample T for Bl-PLen vs Vl-Plen
          N   Mean  StDev  SE Mean
Bl2-PLen  100  19.60   2.95     0.29
Vl-Plen   99  18.41   2.69     0.27

Difference = mu (Bl-PLen) - mu (Vl-Plen)
Estimate for difference:  1.197
95% CI for difference:  (0.408; 1.985)
T-Test of difference = 0 (vs not =):
T-Value = 2.99  P-Value = 0.003  DF = 195
```

Fig 23. T test for testing hypothesis length of path for “blossom(2) and Vlrrt variations are not equal” in high space.

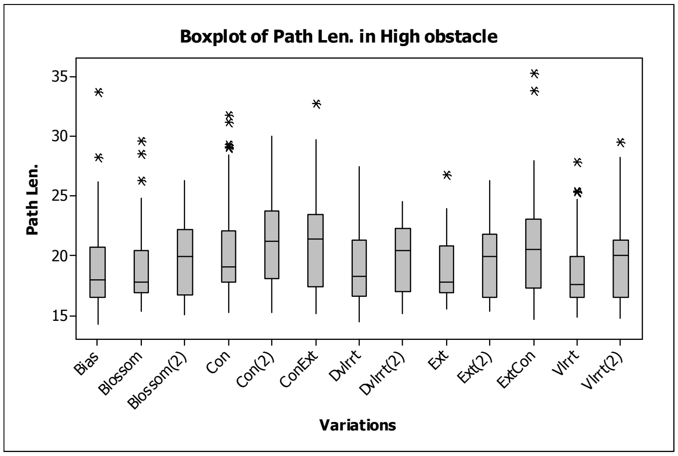

Fig. 24. Path length Boxplot on the high density of obstacle.

From the Fig. 23 the statistical analysis for the hypothesis blossom(2) and Vlrrt are not equal is accepted and that means we have to develop RRT variation in this space or find another strategy to improve the results.

In door obstacle the tests result are in Table 6. And Fig. 26 shows the boxplot chart for this result. Also in this space we test the hypothesis “there is a difference between Vlrrt the best variation in path length and the Blossom(2) the best bidirectional variation”, Fig. 25 shows the testing result.

Table 6: Path length on doors obstacles.

	Path Median	Path Min	S-Path Median	S-Path Min	Rate %
Bias	17.089	14.145	14.285	11.826	16.41
Blossom	17.389	13.787	***13.960**	11.771	***19.72**
Blossom(2)	**17.256**	14.153	**14.127**	11.740	18.13
Con	18.072	15.407	14.883	11.869	17.65
Con(2)	17.810	13.757	14.792	11.797	16.95
ConExt	17.422	13.574	14.749	11.930	15.34
Dvlrrt	16.934	13.604	14.250	11.850	15.85
Dvlrrt(2)	17.532	14.104	14.134	11.787	19.38
Ext	17.263	14.872	14.157	11.802	17.99
Ext(2)	17.413	13.685	14.304	11.764	17.85
ExtCon	17.583	***12.278**	14.524	11.799	17.40
Vlrrt	***16.878**	13.957	14.411	11.870	14.62
Vlrrt(2)	17.577	13.046	14.165	11.835	19.41

Form the T-test P-Value is 0.751>0.05 so we reject this hypothesis, and conclude that there is no sufficient difference between the two variations in confidence level 95%. Fig. 27 shows summary information about this statistical test.

```
Two-sample T for Bl-PLen vs Vl-PLen
          N   Mean  StDev  SE Mean
Bl2-PLen  100  17.30   1.57     0.16
Vl-PLen   99  17.23   1.85     0.19

Difference = mu (Bl-PLen) - mu (Vl-PLen)
Estimate for difference:  0.077
95% CI for difference:  (-0.403; 0.557)
T-Test of difference = 0 (vs not =):
T-Value = 0.32  P-Value = 0.751  DF = 191
```

Fig 25. T test for testing hypothesis length of path for “blossom(2) and Vlrrt variations are not equal” in doors space.

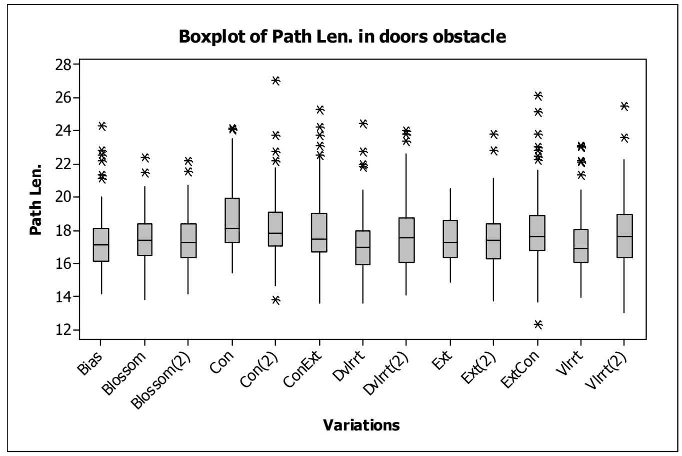

Fig. 26. Path length Boxplot on the doors obstacle.

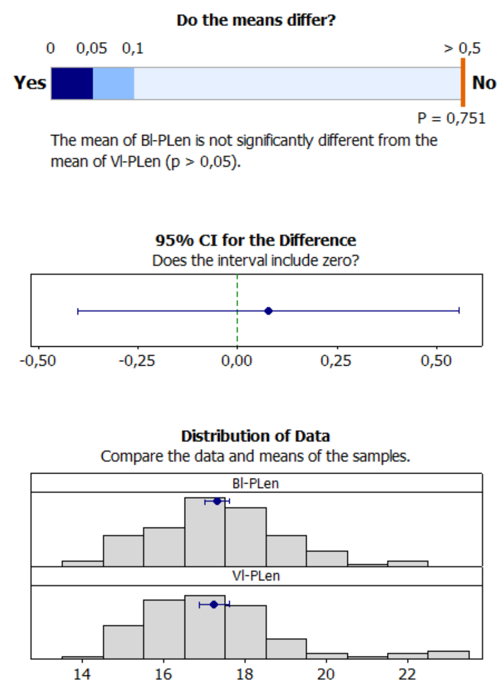

Fig. 27.T-test summary for testing hypothesis blossom(2) and Vlrrt variations are not equal in the T space

We see that when we care about the path length, the single tree variations is better than the double trees variations and that because the way of extending the tree depend on grow a branch from the nearest node in the tree and that makes the path shorter than the path in two tree which is two paths connecting to each other. But the successful rate of unidirectional variations is low.

Fig. 28 shows the short path in bidirectional variation. The thick line is the path generated from connecting the two trees. And the dashed line is the shortened path.

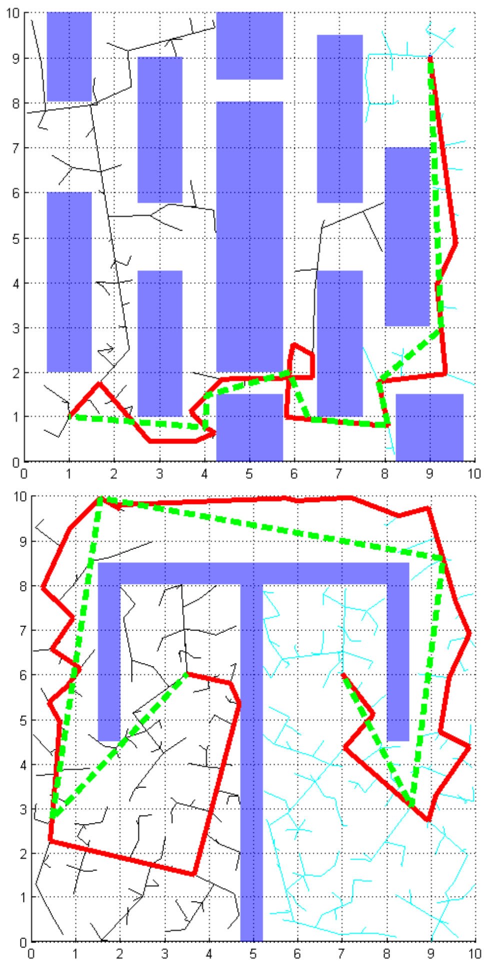

Fig.28 Two RRT variation with Path(thick) and short Path(dashed).

From the testing result the advantage of making the path shorter in length is varying from 13% – 28% in rate, depending on the testing environment, obstacles and variation. And we can use this result to use bidirectional variations when we care about the path length.

V. Conclusion

In this paper we introduce an algorithm to shorten RRT path in length, number of points and degree of tortuous. Also we test many types of RRT variations in 4 different type of obstacle and make some statistical analysis on the result to ensure and support our decision about using of one variation.

We conclude that if we care about the time of execution, in low density obstacles the best result is Vlrrt(2) with 100% successful in the result. For T obstacle the best result is with Vlrrt but with one fail in the result. So we choose to use the Vlrrt(2) based in the statistical result which show that there is no sufficient difference in the two variations with confidence level 95%. For high density of obstacle the best variation is Con(2) but based on the statistical result we see that we can use Vlrrt(2) and Ext(2) with no sufficient.

The last space is doors obstacle and the results give the best variation Dvlrrt(2) and we can use Vlrrt(2) based on statistical tests. After all, we conclude that we can use Vlrrt(2) variation in all spaces, and it will give good result as the best with confidence level 95%.

But if we care about the length of path, in doors and low spaces we can find alternative bidirectional variation but in high and T spaces we can't. For that we should find another method or strategy to get solution. One of these strategies is to develop the optimization of path and making it shorter and smoother.

References

[1] A. Abbadi, R. Matousek, P. Minar, P. Soustek. "RRTs Review and Options", computational Engineering in Systems Applications, Volume II, 2011, IAASAT Press, pp. 194-199, Iasi, Romania. ISBN: 978-1-61804-014-5, ISSN: 2223-9812.

[2] Filipe Militao. Karl Naden. Bernardo Toninho. Improving RRT with Context Sensitivity. April 13. 2010.

[3] M. Kalisiak. M. Panne. RRT-blossom: RRT with a local flood-fill behavior. In International Conference on Robotics & Automation. Orlando. Florida. 2006

[4] S. M. LaValle. J. J. Kuffner. "Rapidly-exploring random trees: Progress and prospects". Workshop on Algorithmic Foundations of Robotics. 2000.

[5] Kuffner J.J.. LaValle S.M.. "RRT-connect: An efficient approach to single-query path planning". IEEE Int. Conf. on Robotics and Automation. pp. 995- 1001. 2000.

[6] M. Strandberg. "Augmenting RRT-planners with local trees." in IEEE Int. Conf. on Robotics & Automation. 2004. pp. 3258–3262.

[7] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. In Technical Report No. 98-11. October 1998.

[8] D.M. Himmelblau . "Process Analysis by Statistical Methods". John Wiley & Sons Inc (June 1970). ISBN 10 . 047139985X

[9] J. Bruce . M. Veloso. Real-Time Randomized Path Planning for Robot Navigation . Intelligent Robots and Systems. 2002

[10] P. Cheng. S. M. LaValle. Reducing metric sensitivity in randomized trajectory design. In IEEE/RSJ International Conference on Intelligent Robots and Systems. 2001.

[11] P. Popela. J. Dupacova. "Melt Control: Charge Optimization via Stochastic Programming". In W. Zieamba. S. W.Wallace (eds.): Applications of Stochastic Programming. Chapter 15. pp. 277–289. ISBN 0-89871-555-5. 2005. SIAM

[12] R. Grepl. J. Vejlupek. V. Lambersky. M. Jasansky. F. Vadlejch. P. Coupek. P. "Development of 4WS/4WD Experimental Vehicle: platform for research and education in mechatronics". IEEE International Conference on Mechatronics (ICM 2011-13-15). 2011

[13] Oplatkova. Z.. Senkerik. R.. Belaskova. S.. Zelinka. I.: Synthesis of Control Rule for Synthesized Chaotic System by means of Evolutionary Techniques. BUT. 16th International Conference on Soft computing. Brno. 2010. 91-98. ISBN 978-80-214-4120-0.

[14] Buniyamin N., Sariff N., Wan Ngah W.A.J., Mohamad Z. "Robot global path planning overview and a variation of ant colony system algorithm", International journal of mathematics and computers in simulation, Issue 1, Volume 5, 2011, ISSN: 1998-0159.

MSc. Radomil Matousek, Ph.D. He is from Czech Republic; he graduated from Brno University of Technology in the Czech Republic. Master of Science in Applied Informatics. Doctorate of the Cybernetics at the Faculty of Electrical Engineering in 2004. Researcher and Ph.D. supervisor in Artificial Intelligence and Control Design area. Head of Dept. of Applied Informatics.

Eng. Ahmad Abbadi. He is from Syria, he graduated from Aleppo University in Syria in 2006 as Electronic Engineer (computer specialist), started PhD program at the Brno University of Technology in the Czech Republic in 2010.