International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

Enhanced Informed Probabilistic Road Map Algorithm with Parameter Optimization

Salma Rahmani[1*], M Aria Rajasa[2*]

[1] Departemen Teknik Elektro, Universitas Komputer Indonesia, Indonesia

[2] Departemen Teknik Elektro, Universitas Komputer Indonesia, Indonesia

Email: *muhammad.aria@email.unikom.ac.id

Abstract. This paper presents the design and performance testing of the Enhanced Informed Probabilistic Roadmap (EI-PRM) algorithm in path planning in various environments, such as simple environments, dense environments and narrow paths. This research evaluates the effectiveness of the algorithm in terms of solution cost and computation time by testing different parameter configurations, including number of sample points (n_{sample}) and cost scaling factor ($\propto \hat{C}_{best}$). The results show that the EI-PRM algorithm can adjust the sampling strategy based on the available information, resulting in an optimal solution with high efficiency. During the test, in a simple environment with the parameter value of nsample between 200 and 400 and $\propto \hat{C}_{best}$ parameter value between 1.2 and 1.4, the best solution cost is 344.93 and the computation time is 1.9 seconds. However, in a denser environment, the optimal solution cost reaches 141,586 with a computation time of 1.16 seconds, a parameter value of nsample 200, and a parameter value of $\propto \hat{C}_{best}$ 1.5. Furthermore, the algorithm shows good performance on narrow paths with an optimal solution cost of about 293.39 and the best computation time of 0.38 seconds at a parameter value of n_{sample} 400 $\propto \hat{C}_{best}$ parameter 1.3. This research focuses on the importance of parameter optimization and efficient sampling strategies to improve path quality and speed up computation time. In general, the results indicate that the EI-PRM algorithm is effective for path planning under various environmental conditions. The process of the EI-PRM algorithm consists of several steps. First, sample points are created at random. In the second step, the computer will link the example locations to produce a roadmap. In the last step, the shortest path inside an ellipsoid-bounded search area will be determined. The size of the ellipsoid will increase gradually until the best path solution is found. This research is expected to contribute significantly to the development of path planning algorithms that are more efficient, faster and capable of producing high-quality paths in complex environments. This research has the potential to improve applications in transportation and logistics that require optimal path planning in order to reduce operational costs and improve safety.

Keywords: Path planning, Probabilistic roadmap, Tunning parameter, Time computation, Cost.

1. Introduction

A subfield of robotics research, path planning algorithms are widely used in vehicle and robot programming applications [1]. The goal of the path planning algorithms is to find a path that allows the robot to move from it's home position to it's destination while avoiding obstacles [2, 3]. Path planning algorithms are also widely used in a variety of applications, including automation [4], robot navigation [5], autonomous vehicles [6], digital character encoding [7],robotic surgery [8], and even in chemistry, such as protein folding processes [9]. The planning time and path optimization cost are two metrics used to assess the performance of various path planning algorithms, which vary in terms of performance [10], [11]. As a result, it is anticipated that these algorithms would generate ideal pathways quickly [12]. When enough time or iterations provide an optimal solution, a path planning method is considered to be asymptotically optimal [13], [14]. Shortest distance, comfort, least amount of risk, and energy efficiency are a few examples of optimality criteria [15]. Random sampling, graphics, and sensor-based path planning are often employed techniques [16].

Many scholars have suggested algorithms with asymptotic optimality after studying path planning algorithms in great detail. All algorithms, how ever, operate differently. Proposed by Karaman and Frazzoli, Rapidly-exploring Random Tree Star (RRT*) is well-liked approach because it can yield asymptotically optimal solutions [17]. Nevertheless, RRT* has a drawback in that its computation time to reach an optimal solution still needs improvement. One factor that slows down RRT*'s computation is its requirement to sample the entire search space [18], [19], [20]. Researchers have also explored various methods, including hybridization, to enhance performance and achieve better solutions [21-24].

Gammell et al. [25] introduced the Informed RRT* algorithm, which utilizes sampling by restricting the configuration space based on information from the currently discovered path. According to Wang's research [19], Informed RRT* outperforms the RRT* algorithm in terms of reaching more optimal solutions. Pakaya and Pohan's Informed Probabilistic Road Map (Informed PRM) technique is another that has asymptotic optimality [26]. This Informed PRM method is a hybridization of the PRM and Informed RRT* algorithms. The study shown that Informed PRM can generate almost ideal routes in a range of testing conditions. Furthermore, this approach performs better in terms of calculation time and generated path quality than both RRT* and Informed RRT*. Nevertheless, one limitation of Informed PRM is that, depending on the situation, it may not always outperform Informed RRT*. Additionally, only the utilization of ellipsoid space between the initial and goal designs was suggested by this study [27].

Pohan [3] has again proposed the hybridization of path planning algorithms by merging the BFS method and path smoothing. The investigation proved that his approach can create pathways that are higher quality than those produced by the RRT* algorithm. In contrast to the RRT* approach, this hybridization has a computational disadvantage. Path quality is significantly increased when BFS and path smoothing are combined, although computation time is still an issue, particularly in complicated circumstances.

A hybrid sampling strategy was developed by Fauzi et al. [28] to expedite the pathfinding process of the RRT algorithm. Experiments showed that this method resulted in faster computation times, especially when sampling with 90% target biasing, 5% border, and 5% random. This method also proved to be more effective in complex environments, such as cluttered and narrow spaces. However, the study has some drawbacks. One of them is the reliance on the percentage combination of the sampling methods, which can influence the results. If the combination is not properly adjusted, the outcome may not be optimal.

International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

Additionally, boundary sampling failed in several trials, indicating that sampling too close to obstacle surfaces can hinder the search process.

Sopa et al. [29] designed an integrated sampling method by combining goal biasing, Gaussian, and boundary in the RRT* algorithm. Testing showed that this integrated method produced shorter paths and faster computation times compared to using the Gaussian and boundary methods separately. However, the study lacked exploration of sampling method optimization for real-world scenarios and broader applications of this integrated method.

Malik and Pohan. [23] proposed a research on path planning algorithms that combine RRT with PSO. This study show that the proposed method outperforms RRT* and Informed RRT* in terms of path quality, computation time, and number of iterations required. However, there are certain drawbacks to the study, such as the narrow spectrum of disorders it examined, which may affect how generally relevant the results are. The complexity and processing time of this approach under more complex or dynamic conditions necessitate more testing to guarantee its implementation in real-world applications.

The Enhanced Informed Probabilistic Roadmap (EI-PRM) algorithm is being developed and tested as a more reliable and efficient method of path planning. By improving the present Enhanced PRM technique, this work aims to solve some of the drawbacks found in previous research, such as long computation times, poor solution quality, and inefficient sampling throughout the search area under a variety of environmental situations. It is expected that this research will produce better solutions compared to the “RRT” and “Informed RRT” algorithms, as well as hybrid approaches such as “Informed PRM”. These algorithms produce near-optimal paths, but they have limitations under some conditions. Testing and evaluation of EI-PRM will be done by simulation using LabVIEW programming language to compare its performance with RRT* and Informed RRT* algorithms. Additionally, its adequacy will be tried with different benchmark settings, such as a confined way, a congested region, and a square space. As a result, it is anticipated that this inquire about will offer assistance create way arranging calculations that are quicker, more productive, and competent of making superior courses in troublesome circumstances.

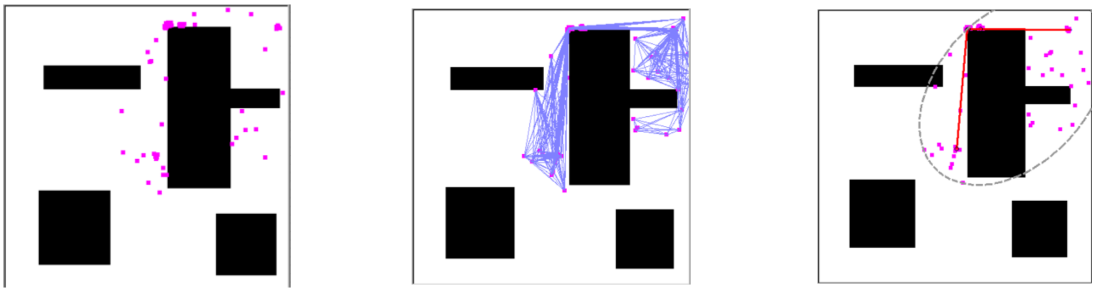
2. Method

The goal of this project is to build an Informed PRM algorithm that, given the positions and lengths of the start and destination points, finds the optimal path by progressively shrinking the ellipsoid region after gradually expanding it to find the path’s starting point. This proposed approach is called the Enhanced Informed PRM (EI-PRM) algorithm. This path planning algorithm was tested with two parameters: sample points (n_{sample}) and cost scalling factor ($\propto \hat{C}_{best}$). The optimal parameter value tunning process is done by grid search method. An illustration of the pathfinding procedure informed by the EI-PRM algorithm (see Figure 1).

International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech



(a) The result of sample point generation

(b) Roadmap creation

(c) The shortest path result

Figure 1. Illustration of the pathfinding procedure informed by the EI-PRM algorithm

The illustration shown depicts the path planning process with the EI-PRM algorithm through three main steps. In the first step, sample points (denoted by red) are randomly generated in a search space containing various obstacles (denoted by black areas). This sampling aims to explore the search space and distribute points that are later used in roadmap generation. This process helps the algorithm gradually recognize areas that can be traversed by the robot.

The second step is road map generation. Once the sample points are established, the algorithm connects the points using blue lines, which depict potential paths between two traversable points. This network of paths forms a road map, which is used to find a route from the starting point to the destination. The lines formed indicate connections between the sample points that are not obstructed by obstacles in the search space. The more connections formed, the more optimized the search space exploration process.

In the final stage, the algorithm determines the shortest path that can be taken. When the iteration starts, the search area is bounded by an ellipsoid. If in the first iteration no solution or path connecting the starting point and the destination has been found, then the size of the ellipsoid will be gradually enlarged until a solution is found. After the solution path is found, in the next iteration the ellipsoid size will be reduced if a shorter path is found. This process continues to repeat until it reaches the maximum iteration limit set. The performance comparison of the EI-PRM algorithm with the RRT* and Informed RRT* algorithms will be discussed in Chapter III.

The complete proposed algorithm (see Figure 2). Line 1 initializes the best value for path cost using the Euclidean distance from the starting position (q_{init}) to the destination position (q_{goal}). This value is multiplied by a constant factor ($\propto \hat{C}_{best}$) to get an initial estimate of the best path cost. This initial estimate will be updated in subsequent iterations. In line 2, the algorithm will loop until a stopping condition is met. This stopping condition is usually determined based on the maximum number of iterations or when the optimal path solution has been found. Lines 3 and 4 initialize the set of nodes (V) and edges (E) connecting the nodes in the graph. V is the set of valid sample nodes (free from obstacles), while E is the set of possible paths between two nodes. Line 5 initializes the set of path solutions (X_{sol}) as an empty set that will later be filled with the shortest path solutions. In lines 6 to 10, random sampling from the search space is performed. The algorithm keeps adding random points (X_{rand}) taken at random until it reaches the desired number of samples (n_{sample}). For each random point, the algorithm checks if the point is collision-free. If it is collision-free, the

International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

point is added to the set V. In lines 12 to 19, the algorithm builds a graph from the collected points. For each point q in the set V, the algorithm selects neighbors (N_q) based on the closest distance. If there is an obstacle-free path between two points (q and q'), then the algorithm adds the path to the set E as a valid edge. On lines 20 to 21, the algorithm forms a graph T consisting of nodes (V) and edges (E). The algorithm then uses Dijkstra's algorithm to find the shortest path from the starting point (q_{init}) to the destination point (q_{goal}). The result is stored in the solution set X_{sol}. Once the path is found, the algorithm compares the cost of the new path with the previous solution. On line 22, the best cost value ($\hat{C}_{best}$) is updated by selecting the path that has the minimum cost from the set of solution paths (X_{sol}). If the stopping condition has not been met, the algorithm will iterate again to find a better solution path, by updating the $\hat{C}_{best}$value to minimize the search space.

```
Algorithm 1. X_sol = (n, map)
1.  Ĉ_best ← euclidean distance (q_in, q_goal) ×∝ Ĉ_best
2.  while termination condition not met do
3.      V←∅
4.      E←∅
5.      X_sol←∅
6.      while |V| < n_sample do
7.          repeat
8.              X_rand ← Sample(X_start, X_goal, Ĉ_best)
9.          until q is collision-free
10.         V ← V ∪ {q}
11.     end while
12.     for all q ∈ V do
13.         N_q ← the neighbors of q chosen from V according to do dist
14.         for all q′ ∈ N_q do
15.             if (q, q′) is collision-free then
16.                 E ← E ∪ {(q, q′)}
17.             end if
18.         end for
19.     end for
20.     T = (V, E)
21.     X_sol ← shortest path(q_init, q_goal, T) using Djikstra Algorithm
22.     Ĉ_best ← min (X_soln ∈ X_soln){Cost(X_soln)}
23. End while
```

Figure 2. EI-PRM algorithm

The sampling process will be carried out using algorithm 2 (see Figure 3). The pseudocode of the EI-PRM sample algorithm mimics the Informed PRM algorithm proposed by H. O. Pakaya and M. A. R. Pohan [26]. This algorithm aims to generate random samples that will be used in the path planning process, taking into account the information from the best path that has been found before. It modifies the sampling strategy dynamically based on the search conditions. The process is divided into two main conditions: if the value of C_{max} (the maximum distance between the starting point and the destination) is finite, or otherwise. If $C_{max} < \infty$ (line 1) the algorithm starts by checking if the maximum distance (C_{max}) is finite. If yes, this means the best path has been found and random samples will be generated in the space limited by C_{max}. This approach aims to optimize the sampling process and speed up the search for a better solution. This usually happens when a path solution has been found in the previous iteration, and the algorithm wants to find a more optimal path around the best path. If the number of samples (|V|) collected is still less than half of the desired total number of samples (n/2), the algorithm will perform sampling within more focused constraints (lines 2-3). At this stage, the shortest distance (C_{min}) between the start and goal points is calculated using the Euclidean norm (straight line distance between two points). Then, the center of the best path (X_{centre}) is calculated as the midpoint between the start point (X_{start}) and the goal point (X_{goal}) (lines 3-4). The algorithm converts the local coordinate system to the world coordinate system using a rotation matrix. This is necessary so that the resulting sample points are in the correct reference frame in the search space (line 5). The algorithm calculates the parameters of the ellipsoid that will be used to bound the sampling space. r1 is half of the C_{max} value, which determines the principal radius of the ellipsoid along the principal axis. For the other axes, the algorithm calculates ri using a formula involving the difference between C_{max} and C_{min}. A diagonal matrix L is then created to represent the size of the ellipsoid along each axis (lines 6-8). At this stage, the algorithm generates a random sample point inside the unit ball (SampleUnitBall), then converts this point into a point inside the ellipsoid by applying the previously calculated rotation and scale. The result is a sample point (X_{rand}) that is confined in the ellipsoid space around the best path (lines 9-10).

If the Number of Samples is Sufficient (Lines 11-13). If the number of samples that have been generated is more than half of the desired number (n/2), the algorithm changes the sampling strategy. In this case, the samples will be taken around the previously found best path, with the aim of improving the quality of the found path. The algorithm samples around the best path (X_{sol}) using a certain radius (d). This process allows the algorithm to focus on areas that are already known to be near-optimal, thus accelerating convergence towards the optimal solution (line 12).

If $C_{max} = \infty$ (Lines 14-16). If the value of C_{max} is infinite, it means that the path solution has not been found or there is no limit that can be used to narrow the search space. In this case, the algorithm again uses the conventional random sampling strategy. The algorithm generates random samples evenly across the search space. This approach is taken when the information regarding the best path is still very limited or no solution has been found yet (line 15). After determining the appropriate sampling method, the algorithm returns random sample points (X_{rand}) that will be used to expand the graph in the next iteration.

Algorithm 2. $X_{rand} \leftarrow sample(X_{start}, X_{goal}, \hat{C}_{best})$

1. **if** $C_{max} < \infty$ **then**
2.   **if** $|V| < n/2$ **then**
3. $C_{min} \leftarrow \|X_{goal} - X_{start}\|_2$
4. $X_{centre} \leftarrow (X_{goal} + X_{start})/2$
5. $C \leftarrow \mathrm{RotationToWorldFrame}(X_{start}, X_{goal})$
6. $r_1 \leftarrow C_{max}/2$
7. $\{r_i\}_{i=2,\dots,n} \leftarrow \left(\sqrt{C^2{}_{max} - C^2{}_{min}}\right)/2$
8. $L \leftarrow diag\{r_1, r_2, \dots, r_n\}$
9. $X_{ball} \leftarrow \mathrm{SampleUnitBall}$
10. $X_{rand} \leftarrow (CLX_{ball} + X_{centre}) \cap X$
11.   **else**
12. $X_{rand} \leftarrow \mathrm{SamplingNearBestPath}(X_{sol}, d)$
13.   **end if**
14. **else**
15. $X_{rand} \leftarrow \mathrm{RandomSampling}(map)$
16. **end if**
17. **return** X_{rand}

Figure 3. EI-PRM algorithm sample pseudocode

This study also evaluates and compares the performance of the EI-PRM algorithm based on sample points (n_{sample}) and cost scaling factor ($\propto \hat{C}_{best}$) in various path planning scenarios with different complexity levels, including environmental conditions such as clutter, narrow, and square field. This study involved six test environments that were repeated 10 times. The EI-PRM method was created using the LabVIEW programming language and the path planning library [30]. The tuning findings from earlier experiments or recommendations from the literature served as the foundation for the algorithm's setup and settings. This approach is expected to yield a reliable and accurate path planning performance evaluation using the EI-PRM algorithm.

3. Results and Discussion

In research conducted by Malik and Pohan [23], discussed the combination of path planning algorithms with the RRT algorithm and the PSO algorithm using several test environments such as clutter, multiple narrow, square field and tough passage environments. The study also used an iteration value of 5000, but this algorithm still has shortcomings in computation time compared to the RRT * algorithm. Meanwhile, research conducted by Rahajoeningroem and Gunastuti [31], related to the performance study of path planning algorithms based on the informed rrt * algorithm. The study compared several variants of the informed RRT* algorithm using several test environments with different complexities. Such as a maze test environment, an environment with 50 obstacles, an environment with 100 obstacles, and an environment with 200 obstacles.

The output performance investigated in this study is the quality of the final solution, based on sample points (n_{sample}) and cost scaling factor ($\propto \hat{C}_{best}$). The tests, which include square field, clutter, and narrow test scenarios are based on LabVIEW software simulations. The exam consists of 10 trials, with an iteration value of 3000 for each environtment. This research also compares the EI-PRM algorithm, RRT* algorithm and Informed RRT* algorithm. This performance study can also help provide a better understanding of the selection of better path planning algorithms for more specific path planning applications.

3.1 The Performance of the EI-PRM algorithm in a Environtment 02 – Simple Obstacle

Table 1. show the best solution cost results in environment 02 - simple obstacle. While Table 2. shows the best computation time results in the 02-simple obstacle environment. The best solution cost value obtained by the EI-PRM algorithm is 344.93, at a value of $\propto \hat{C}_{best}$ 1.2 to 1.4 with an n_{sample} value of 200 to 400. While the best computation time obtained by the EI-PRM algorithm is 1.9 seconds.

International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

Table 1. The best solution cost results of the EI-PRM algorithm in test environment 02 - simple obstacle

n_{sample}	Solution Cost $\propto \hat{C}_{best}$									
	1.1	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
100	344.96	344.97	344.932	344.93	344.932	344.935	344.935	344.933	344.95	344.94
200	344.931	344.93	344.93	344.931	344.93	344.93	344.932	344.932	344.935	344.931
300	344.93	344.931	344.93	344.93	344.932	344.932	344.93	344.93	344.93	344.93
400	344.931	344.931	344.932	344.93	344.932	344.931	344.931	344.931	344.93	344.932
500	344.931	344.932	344.931	344.931	344.931	344.93	344.93	344.93	344.931	344.93

Table 2. The best computation time results of EI-PRM algorithm in environment 02-simple obstacle environment

n_{sample}	Time $\propto \hat{C}_{best}$									
	1.1	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
100	6.5	4.3	3.1	3.5	3	4.2	5.7	5.2	4.9	3.8
200	5.8	2.6	3.9	3.2	2.7	3.7	4.1	3.4	5.6	2.1
300	7.6	5.4	4.4	4.2	3.5	2.8	4.4	4.2	3.8	4.7
400	3.8	3.3	2.2	2.9	2.4	2.1	3.1	3.8	4.5	4.3
500	7.8	2.9	2	2.6	1.9	3.7	2.8	2.5	5.3	2.9

The path with the best solution cost value generated by the EI-PRM algorithm after 10 trials with the nsample value parameter from 100 to 500, and the value parameter $\propto \hat{C}_{best}$ from 1.1 to 2.0 (see Figure 4).

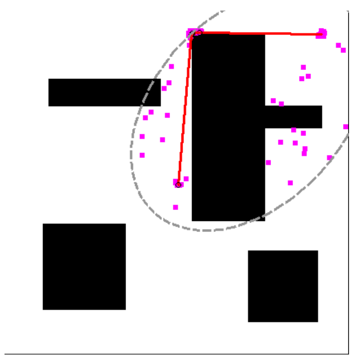


Figure 4. The path with the lowest solution cost generated by the EI-PRM algorithm in the Environtment 02-Simple Obstacle

3.2 The Performance of the EI-PRM algorithm in a Environtment 03 – Clutter 1

Table 3. show the best solution cost results in environment 03 – clutter 1. While Table 4. shows the best computation time results in the environment 03-clutter 1. Overall, the best

Solution Cost often appears at $\propto \hat{C}_{best}$ = 1.2, 1.6, and 2.0 with a minimum value of 141,580, especially at n_{sample} 100 to 400. The best time occurs at n_{sample} 200, 300, and 500, with α = 1.5, where the computation time is very low, which is below 1.5 seconds. The best combination of Solution Cost and Time was found at n_{sample} 200 with $\propto \hat{C}_{best}$= 1.5, where Solution Cost 141.586 and Time: 1.16 seconds.

Table 3. The best solution cost results of the EI-PRM algorithm in test environment 03 – clutter 1

n_{sample}	Solution Cost $\propto \hat{C}_{best}$ 1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
100	141.586	141.58	141.586	141.586	141.586	141.586	141.586	141.586	141.586	141.58
200	141.586	141.586	141.586	141.587	141.586	141.58	141.588	141.586	141.586	141.586
300	141.588	141.586	141.586	141.58	141.586	141.588	141.586	141.586	141.586	141.58
400	141.58	141.586	141.585	141.586	141.587	141.586	141.586	141.585	141.587	141.58
500	141.586	141.586	141.587	141.585	141.586	141.586	141.586	141.586	141.586	141.586

Table 4. The best computation time results of EI-PRM algorithm in environment 03 – clutter 1

n_{sample}	Time $\propto \hat{C}_{best}$ 1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
100	2.1	3.41	5.9	3.2	1.8	2.31	2.4	3.55	4.7	4.1
200	8.7	2.7	2	1.81	1.16	2.64	2.3	3.25	4.75	4
300	6.9	2.65	2.17	2.35	1.37	1.79	2.1	3.1	4.4	3.75
400	2.4	2.11	1.8	2	1.49	2.43	1.36	1.67	1.99	1.83
500	3.4	2.9	3	1.74	1.5	1.86	2.32	3.43	4.54	3.98

The path with the best solution cost value generated by the EI-PRM algorithm after 10 trials with the n_{sample} value parameter from 100 to 500, and the $\propto \hat{C}_{best}$ value parameter from 1.1 to 2.0 (see Figure 5).

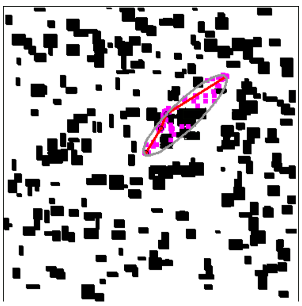


Figure 5. The path with the lowest solution cost generated by the EI-PRM algorithm in the Environtment 03 - Clutter 1

International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

3.3 The Performance of the EI-PRM algorithm in a Environtment 04 – Clutter 2

Table 5. show the best solution cost results in environment 04 – clutter 2. While Table 6. shows the best computation time results in the environment 04 - clutter 2. The best Solution Cost was found at n_{sample} 200 and 400, with = 1.5, which resulted in a minimum value of 355,055. The best computation time occurs in n_{sample} 500 with $\propto \hat{C}_{best}$= 1.5, where the computation time is only 2.07 seconds. In addition, n_{sample} 400 also has a good time (4.41 seconds at $\propto \hat{C}_{best}$= 1.5). The best combination of Solution Cost and Time is found in n_{sample} 400 with $\propto \hat{C}_{best}$= 1.5, with a Solution Cost value of 355.055 and Time of 2.07 seconds.

Table 5. The best solution cost results of the EI-PRM algorithm in test environment 04 – clutter 2

Solution Cost										
n_{sample}	$\propto \hat{C}_{best}$									
	1.1	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
100	355.058	355.058	355.058	355.058	355.058	355.058	355.058	355.065	355.061	355.062
200	355.056	355.058	355.056	355.058	355.055	355.057	355.057	355.06	355.06	355.056
300	355.056	355.056	355.056	355.058	355.056	355.057	355.056	355.058	355.056	355.056
400	355.065	355.057	355.068	355.056	355.055	355.055	355.065	355.056	355.065	355.057
500	355.056	355.056	355.056	355.057	355.056	355.056	355.056	355.056	355.057	355.056

Table 6. The best computation time results of EI-PRM algorithm in environment 04 – clutter 2

Time										
n_{sample}	$\propto \hat{C}_{best}$									
	1.1	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
100	9.84	9.07	8.3	7.39	6.49	7.09	7.7	8.95	10.2	9.57
200	8.2	8.29	8.38	6.71	5.05	6.44	7.84	6.69	5.55	6.12
300	7.78	6.69	5.6	5.04	4.49	5.05	5.62	5.32	5.02	5.17
400	5.5	5.67	5.84	5.12	4.41	5.31	6.22	5.51	4.81	5.16
500	4.85	5.04	5.24	3.65	2.07	2.9	3.73	3.67	3.61	3.64

The path with the best solution cost value generated by the EI-PRM algorithm after 10 trials with the n_{sample} value parameter from 100 to 500, and the $\propto \hat{C}_{best}$ value parameter from 1.1 to 2.0 (see Figure 6).

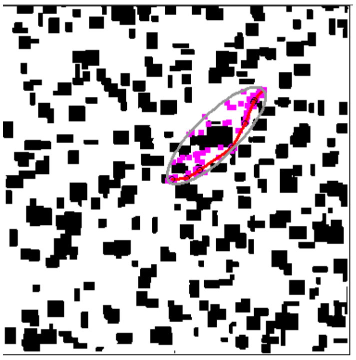


Figure 6. The path with the lowest solution cost generated by the EI-PRM algorithm in the Environtment 04 - Clutter 2

3.4 The Performance of the EI-PRM algorithm in a BIT* Clutter Map

Table 7. show the best solution cost results in BIT* clutter map. While Table 8. shows the best computation time results in the BIT* clutter map. The best Solution Cost is found at n_{sample} 300 with $\propto \hat{C}_{best}$= 1.1, where the Solution Cost value is 432.12 (minimum value). The best computation time occurs at n_{sample} 500 with $\propto \hat{C}_{best}$= 1.5, where the computation time is only 2.02 seconds. In addition, n_{sample} 300 and 400 also have very fast times around 2.14 - 2.29 seconds (at $\propto \hat{C}_{best}$= 1.5). The best combination of Solution Cost and Time is at n_{sample} 300 with $\propto \hat{C}_{best}$= 1.1, where Solution Cost reaches a minimum value of 432.12 even though the computation time is not the fastest (5.33 seconds).

Table 7. The best solution cost results of the EI-PRM algorithm in test BIT* clutter map

Solution Cost										
n_{sample}	$\propto \hat{C}_{best}$									
	1.1	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
100	441.77	467.35	492.94	433.84	433.75	441.45	449.14	445.85	442.56	442.2
200	441.28	442.75	488.36	460.52	432.69	437.62	442.56	442.56	441.77	442.16
300	432.12	441.77	452.35	441.95	441.56	449.14	442.56	441.77	442.65	442.21
400	442.56	442.56	441.75	442.77	433.73	441.77	441.77	441.77	441.77	442.56
500	441.77	441.77	442.56	441.56	433.75	442.56	442.56	442.16	441.77	442.96

International Journal of Research and Applied Technology
5(1)(2025) 65-81
Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

Table 8. The best computation time results of EI-PRM algorithm in BIT* clutter map

n_{sample}	Time $\propto \hat{C}_{best}$									
	1.1	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
100	5.58	5.88	6.18	6.55	6.93	7.64	8.36	6.85	5.35	6.1
200	8.3	6.95	5.61	4.16	2.71	6	9.29	6.74	4.2	5.47
300	8.78	7.05	5.33	3.81	2.29	4.26	6.23	5.03	3.84	4.43
400	6.76	4.73	2.68	2.41	2.14	3.37	4.6	4.58	4.56	4.57
500	5.59	3.84	2.1	2.06	2.02	3.23	4.45	5.16	5.88	5.52

The path with the best solution cost value generated by the EI-PRM algorithm after 10 trials with the n_{sample} value parameter from 100 to 500, and the $\propto \hat{C}_{best}$ value parameter from 1.1 to 2.0 (see Figure 7).

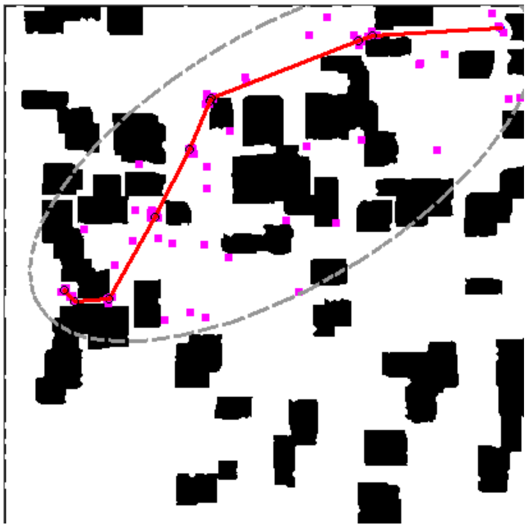


Figure 7. The path with the lowest solution cost generated by the EI-PRM algorithm in the BIT* clutter map

3.5 The Performance of the EI-PRM algorithm in a Multiple Narrow Environtment

Table 9. show the best solution cost results in multiple narrow environtment. While Table 10. shows the best computation time results in the multiple narrow environtment. The best Solution Cost is found in several combinations, namely n_{sample} 200, 300, 400, and 500 with $\propto \hat{C}_{best}$= 1.1, 1.3, 1.7, where the Solution Cost value is 427.16. The best computation time occurs at n_{sample} 500 with $\propto \hat{C}_{best}$= 1.3, where the computation time is only 2.04 seconds. In general, α = 1.3 produces the fastest time on almost all n_{sample}. The best combination of Solution Cost and Time is at n_{sample}300 with $\propto \hat{C}_{best}$= 1.3, where the Solution Cost reaches a minimum value of 427.16 and the computation time is a very fast 2.4 seconds.

International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

Table 9. The best solution cost results of the EI-PRM algorithm in test multiple narrow environtment

n_{sample}	Solution Cost $\propto \hat{C}_{best}$ 1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
100	427.52	427.43	427.34	427.52	427.52	427.52	427.52	427.4	427.29	427.52
200	427.16	427.29	427.16	427.16	427.34	427.34	427.34	427.29	427.34	427.52
300	427.16	427.53	427.16	427.16	427.52	427.52	427.16	427.29	427.34	427.29
400	428.29	427.16	427.16	427.34	427.29	427.29	427.52	427.52	427.16	427.16
500	427.16	427.16	427.23	427.29	427.16	427.16	427.52	427.16	427.29	427.34

Table 10. The best computation time results of EI-PRM algorithm in multiple narrow environtment

n_{sample}	Time $\propto \hat{C}_{best}$ 1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
100	3.86	3.25	2.64	5.24	7.82	6.7	5.57	5.41	5.25	5.46
200	8.27	5.39	2.52	3.9	5.27	5.24	5.21	5.15	5.1	5.12
300	7.82	5.11	2.4	3.7	5.01	5.02	5.04	4.84	4.65	4.74
400	6.23	4.19	2.16	3.39	4.63	4.76	4.89	4.08	3.27	3.67
500	4.77	3.4	2.04	2.42	2.8	3.24	3.68	3.31	2.94	3.12

The path with the best solution cost value generated by the EI-PRM algorithm after 10 trials with the n_{sample} value parameter from 100 to 500, and the $\propto \hat{C}_{best}$ value parameter from 1.1 to 2.0 (see Figure 8).

International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

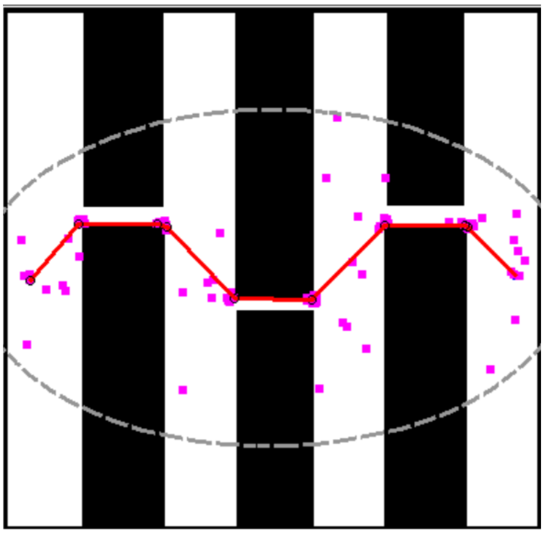


Figure 8. The path with the lowest solution cost generated by the EI-PRM algorithm in the multiple narrow environtment

3.6 The Performance of the EI-PRM algorithm in a Narrow Passage

Table 11. show the best solution cost results in narrow passage. While Table 12. shows the best computation time results in the narrow passage. In general, at all n_{sample} sizes (100, 200, 300, 400, 500), the best values for solution cost are found to be around 293.39, especially in most configurations.

At n_{sample} 100, the lowest solution cost result is 293.12 in configuration 1.4, which is one of the lowest values in the whole table. While in n_{sample} 300, the lowest result was also found to be 293.12 in 1.8. Overall, the most optimal solution cost is seen in nsample 300 and 400, with small and stable variations. Overall, the best (fastest) computation time was obtained in n_{sample} 400 with configuration 1.3 which gave a time of 0.38 seconds. This very low time value indicates good efficiency in this sample. To get the best solution cost and time results, it is recommended to use n_{sample} 400, because it produces the fastest time (0.38 seconds) with a very stable solution cost.

Table 11. The best solution cost results of the EI-PRM algorithm in test narrow passage

n_{sample}	Solution Cost									
	$\propto \widehat{C}_{best}$									
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
100	293.39	294.09	294.09	293.12	294.09	294.09	293.39	293.45	294.59	294.09
200	293.39	293.34	293.39	293.45	295.03	293.39	293.43	293.45	293.63	293.58
300	293.39	293.39	293.39	293.39	293.39	293.39	293.39	293.12	293.62	293.39
400	293.45	293.39	293.39	293.12	293.39	293.29	293.29	293.45	293.39	293.39
500	293.39	293.39	293.34	293.39	293.39	293.39	293.45	293.39	293.45	293.16

International Journal of Research and Applied Technology

5(1)(2025) 65-81

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

INJURATECH

Table 12. The best computation time results of EI-PRM algorithm in narrow passage

n_{sample}	Time									
	$\propto \hat{C}_{best}$									
	1.1	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
100	1.72	1.26	0.8	1.02	1.25	1.58	1.92	5.81	9.7	7.75
200	7.3	7.91	0.61	0.81	1.01	1.31	1.62	5.53	9.44	7.48
300	1.45	1.67	1.89	1.3	0.72	1.03	1.35	5.12	8.89	7
400	5.44	2.91	0.38	1.1	1.84	2.86	1.02	4.54	8.07	6.3
500	3.55	2.65	1.75	106	0.38	0.57	0.76	3.89	7.03	5.46

The path with the best solution cost value generated by the EI-PRM algorithm after 10 trials with the n_{sample} value parameter from 100 to 500, and the $\propto \hat{C}_{best}$ value parameter from 1.1 to 2.0 (see Figure 9).

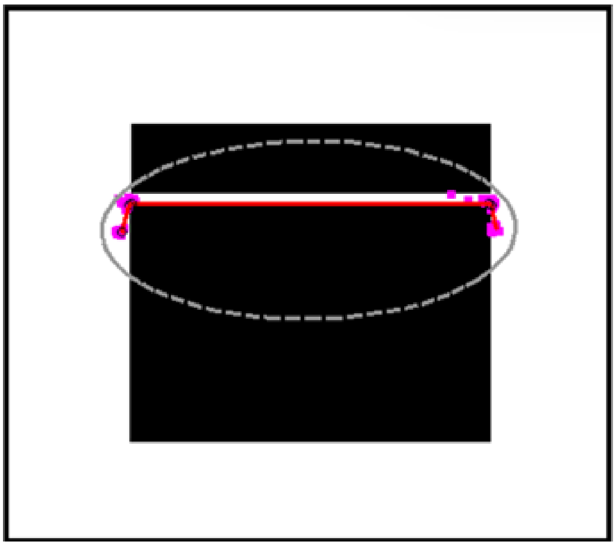


Figure 9. The path with the lowest solution cost generated by the EI-PRM algorithm in the narrow passage

Path planning algorithm using this variant of the PRM algorithm has undergone many developments. In this EI-PRM algorithm research there are several interesting areas to be explored further. It is also possible to develop this algorithm using the integration between the EI-PRM algorithm and the techniques underlying deep learning [32]. Integrating deep learning improves the algorithm's ability to predict and overcome obstacles in unpredictable environments. This approach can help in reducing computation time and improving the quality of the generated paths. This EI-PRM algorithm can also be developed by conducting performance studies of other PRM algorithm-based path planning algorithms, such as in the research conducted by Rahajoeningroem and Gunastuti [31] who compared the performance of Informed RRT*-based path planning algorithms to enable the selection of more appropriate algorithms in more specific path planning applications.

4. Conclusion

This work investigates, designs and tests the path planning performance of the EI-PRM algorithm across a range of scenarios, including congested areas and constrained spaces. EI-

PRM exhibits a good ability to strike a balance between solution cost and calculation time. Based on the available optimal path data, the algorithm can modify the sample preparation method. Six different scenarios are used to evaluate the performance of the algorithm, and the results show that the cost scaling factor ($\propto \hat{C}_{best}$) and the number of sample points ($\boldsymbol{n_{sample}}$) have a significant impact on the results. For example, the bias of the optimal solution is 344.93 in the initial condition, and the computation takes 1.9 days with optimal parameters 1.2-1.4 and 20. With an optimal parameter of 1.3 and a sample size of 400, the approach shows effectiveness under challenging conditions. A good computation time is 0.38 seconds. This research also highlights how important it is to optimize the parameters by using grid search techniques to maximize the algorithm's capacity to determine the ideal spinning path.

References

[1] Elbanhawi, M., & Simic, M. (2014). Sampling-based robot motion planning: A review. *IEEE Access*, *2*, 56-77.

[2] Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., & Teller, S. (2011, May). Anytime motion planning using the RRT. In *2011 IEEE international conference on robotics and automation* (pp. 1478-1483). IEEE.

[3] Aria, M. (2020). Algoritma Perencanaan Jalur Kendaraan Otonom berbasis Hibridisasi Algoritma BFS dan Path Smoothing. *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, *8*(1), 13-22.

[4] Friedrich, C., Csiszar, A., Lechler, A., & Verl, A. (2017). Efficient task and path planning for maintenance automation using a robot system. *IEEE Transactions on Automation Science and Engineering*, *15*(3), 1205-1215.

[5] Tsai, C. C., Huang, H. C., & Chan, C. K. (2011). Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. *IEEE Transactions on Industrial Electronics*, *58*(10), 4813-4821.

[6] Rasekhipour, Y., Khajepour, A., Chen, S. K., & Litkouhi, B. (2016). A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *18*(5), 1255-1267.

[7] Qiu, Z., & Liu, C. (2012, July). The motion planning in the automatic generation of mobile phone 3D animation. In *Proceedings of the 10th World Congress on Intelligent Control and Automation* (pp. 725-731). IEEE.

[8] Sudhakara, P., Ganapathy, V., & Sundaran, K. (2018, March). Trajectory Planning Using Enhanced Probabilistic Roadmaps For Pliable Needle Robotic Surgery. In *2018 International Conference on Recent Trends in Electrical, Control and Communication (RTECC)* (pp. 61-64). IEEE.

[9] Ekenna, C., Thomas, S., & Amato, N. M. (2016). Adaptive local learning in sampling based motion planning for protein folding. *BMC systems biology*, *10*, 165-179.

[10] Yang, Y., Pan, J., & Wan, W. (2019). Survey of optimal motion planning. *IET Cyber-systems and Robotics*, *1*(1), 13-19.

[11] Mashayekhi, R., Idris, M. Y. I., Anisi, M. H., & Ahmedy, I. (2020). Hybrid RRT: A semi-dual-tree RRT-based motion planner. *IEEE Access*, *8*, 18658-18668.

[12] Dijkstra, E. W. (2022). A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy* (pp. 287-290).

[13] Marble, J. D., & Bekris, K. E. (2013). Asymptotically near-optimal planning with probabilistic roadmap spanners. *IEEE Transactions on Robotics*, *29*(2), 432-444.

[14] Guo, Y., Liu, X., Liu, X., Yang, Y., & Zhang, W. (2022). FC-RRT*: An improved path planning algorithm for UAV in 3D complex environment. *ISPRS International Journal of Geo-Information*, *11*(2), 112.
[15] Noreen, I., Khan, A., & Habib, Z. (2016). Optimal path planning using RRT* based approaches: a survey and future directions. *International Journal of Advanced Computer Science and Applications*, *7*(11).
[16] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, *4*(2), 100-107.
[17] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, *30*(7), 846-894.
[18] Qureshi, A. H., Mumtaz, S., Iqbal, K. F., Ayaz, Y., Muhammad, M. S., Hasan, O., ... & Ra, M. (2014, March). Triangular geometry based optimal motion planning using RRT*-motion planner. In *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)* (pp. 380-385). IEEE.
[19] Wang, W., Gao, H., Yi, Q., Zheng, K., & Gu, T. (2020, June). An improved rrt* path planning algorithm for service robot. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (Vol. 1, pp. 1824-1828). IEEE.
[20] Yuncheng, L., & Jie, S. (2017, April). A revised Gaussian distribution sampling scheme based on RRT* algorithms in robot motion planning. In *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)* (pp. 22-26). IEEE.
[21] Pohan, M. A. R., Trilaksono, B. R., Santosa, S. P., & Rohman, A. S. (2024). Path planning using combined informed rapidly-exploring random tree star and particle swarm optimization algorithms. *IEEE Access*.
[22] Pohan, M. A. R., Trilaksono, B. R., Santosa, S. P., & Rohman, A. S. (2021). Path planning algorithm using the hybridization of the rapidly-exploring random tree and ant Colony systems. *IEEE Access*, *9*, 153599-153615.
[23] Malik, A. D. A. M., & Pohan, M. A. R. (2022). The development of a path planning algorithm combining the rapidly-exploring random tree algorithm and the particle swarm optimization algorithm. *J. Eng. Sci. Technol.*, *17*(6), 3742-3754.
[24] Aria, M. (2020). Path planning algorithm using informed rapidly exploring random tree*-connect with local search. *Journal of Engineering Science and Technology Special Issue on INCITEST*, 50-57.
[25] Gammell, J. D., Srinivasa, S. S., & Barfoot, T. D. (2014, September). Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2997-3004). IEEE.
[26] Pakaya, H. O., & Pohan, M. A. R. (2021). Metode Improved Gaussian Sampling pada Algoritma Rapidly Exploring Random Tree. *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, *9*(2), 106-118.
[27] Novosad, M., Penicka, R., & Vonasek, V. (2023). CTopPRM: Clustering topological PRM for planning multiple distinct paths in 3D environments. *IEEE Robotics and Automation Letters*, *8*(11), 7336-7343.
[28] Fauzi, M., & Pohan, M. A. R. (2021). Informed-RRT* using hybrid sampling to finding fast final path solution. *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, *9*(2), 94-105.

[29] Sopa, A., & Hartono, R. (2021). Algoritma Rapidly Exploring Random Tree Star Dengan Integrasi Metode Sampling Goal Biassing, Gaussian, Dan Boundary. *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, *9*(2), 129-138.
[30] Pohan, M. A. (2022). LabVIEW Libraries untuk Algoritma Perencanaan Jalur Robotik. *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, *10*(1), 47-62.
[31] Rahajoeningroem, T., & Gunastuti, D. A. (2024). Studi Performansi Algoritma Perencanaan Jalur Berbasis Informed Rapidly-exploring Random Tree Star. *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, *12*(1), 50-67.
[32] Ramba, L. S., & Aria, M. (2020). Design of a voice controlled home automation system using deep learning convolutional neural network (DL-CNN). *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali Dan Elektronika Terapan*, *8*(1), 57-73.