

# Enhancing Path Quality of Real-Time Path Planning Algorithms for Mobile Robots: A Sequential Linear Paths Approach

**RAOUF FAREH<sup>1</sup>, MOHAMMED BAZIYAD<sup>2</sup>, TAMER RABIE<sup>3</sup>, (Senior Member, IEEE), AND MAAMAR BETTAYEB<sup>1,4</sup>, (Senior Member, IEEE)**

<sup>1</sup>Electrical Engineering Department, University of Sharjah, Sharjah 27272, United Arab Emirates

<sup>2</sup>Research Institute of Sciences & Engineering (RISE), University of Sharjah, Sharjah 27272, United Arab Emirates

<sup>3</sup>Computer Engineering Department, University of Sharjah, Sharjah 27272, United Arab Emirates

<sup>4</sup>Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Raouf Fareh (rfareh@sharjah.ac.ae)

**ABSTRACT** The traditional trade-off between execution speed and path quality has forced real-time robotic path planning algorithms to sacrifice path quality in order to execute in real-time. Producing a path planning algorithm that targets enhancing both, the path quality and swiftness is a challenging problem. However, this article proposes a novel path planning strategy that aims to break this traditional trade-off, by targeting both, increasing the swiftness, and enhancing the path quality represented by the path length and smoothness. The proposed strategy is based on the observation that most path planning algorithms waste the processing efforts in less critical areas of the map. Therefore, the proposed path planning strategy tends to focus on critical areas such as the areas around obstacles and areas around the goal point, and exhausts the processing power on these critical areas. This is done by neglecting all static obstacles that do not lie between the robot and the destination. For obstacles that intersect with the linear line from the robot to the destination, a basis traditional path planning algorithm such as A\*, D\* or the Probabilistic RoadMap (PRM) technique is only implemented around the obstacles in order to find a feasible path around each selected obstacle. This procedure would minimize the computational efforts compared to applying the basis algorithm on the whole map. Finally, the path quality is enhanced by finding any linear shortcuts between any two points in the path and fix these shortcuts as the final path from the starting point to the goal point. The proposed path planning strategy was tested on a P3-DX Pioneer mobile robot using a kinematic controller. The experimental results have proven that the path planning strategy was able to show a superior advantage over other path planning techniques in both aspects, computational time (reached up to 97.05% improvement) and path quality (reached up to 16.21% improvement for path length and 98.50% for smoothness).

**INDEX TERMS** Path planning, A\*, D\*, PRM, mobile robot, depth sensor, kinematic control, real-time, tracking control, Lyapunov, stability.

## I. INTRODUCTION

Research in mobile robots and its requirements is witnessing growing serious interest among researchers and scientists in the last few years [1]–[5]. Robotics experts are in a frequent research mode to design and produce robots with a high level of automation and flexibility. The ongoing improvements in the capabilities of mobile robots will positively impact many vivid fields such as industry, economics, health-care, and

lifestyle [6]–[10]. Due to the increased flexibility that can be achieved by modern mobile robots, robots are involved in critical applications in the health-care field, from diagnostic tests to rehabilitation to sophisticated surgeries. These operations require a top level of precision and accuracy, where a minor error can lead to an enormous trouble [8].

In addition to the accuracy attribute, another attribute arises its criticality when dealing with mobile robots, which is swiftness. To build an effective autonomous system, the controlling system must be capable to deal with the real-time nature of the applications of autonomous systems. Some examples

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng H. Zhu.

of real-time decisions include avoiding moving obstacles, finding a safe and optimal path to the desired destination, object detection and recognition. For a mobile robot environment, real-time controlling algorithms suffer from the massive computational complexity required by several critical applications. Therefore, a challenging trade-off problem between accuracy and swiftness arises since enhancing the accuracy increases the complexity of the system which affects negatively on the time required to achieve real-time tasks.

A critical real-time objective for an autonomous robotic system is to search for a collision-free path from its current location to the desired destination. To compute a feasible path, an autonomous robotic system must frequently sense the environment in order to detect obstacles. The swiftness of the path planning problem becomes more critical when operating in an environment with fast-moving obstacles. However, the optimal shortest path is usually obtained after several iterations where in each iteration, the quality of the path is enhanced either by shortening the path, or by smoothening the tracking. Therefore, considering improving both objectives, the swiftness and quality, is a challenging problem in the robotic path planning field. The seriousness of the swiftness-accuracy trade-off problem manifests clearly when operating with smart surface-to-air missiles. A surface-to-air missile is designed to be launched from the ground to destroy an aircraft or guided missiles such as a ballistic missile. It is extremely important to locate the missile, launch an anti-ballistic missile, generate the free-obstacle path and destroy the ballistic missile, before it reaches the target. On the other hand, the criticality of the path quality attribute manifests clearly from the surface-to-air missiles example, since the path planning algorithm must generate a precise path, so that the anti-missile does not miss the ballistic missile. Thus, having an acceptable level of path quality is always required to avoid obstacles, and generate a feasible path to the desired destination.

The path planning and motion control problem of an autonomous robotic system have been the focus of many researchers in the last few years [11]–[16]. With the enormous evolution of computation power of modern processors, computer vision techniques have become a major contributor in robotic applications [17], [18]. In [17], vision algorithms play an important role in detecting obstacles in the robot's environment. Then, a non-holonomic mobile robot tracks the path generated with the help of the vision system. Moreover, in [18] an autonomous system that calibrates a vision system and a robot is presented. The proposed system consists of multiple calibrated cameras to localize obstacles or objects of interest with reference to the robot's initial position.

The objective of a path planning technique is to calculate an obstacle-free path to the destination. The sampling-based schemes and the grid-based schemes are examples of classical path planning schemes. The sampling-based schemes operate by scattering a number of points randomly in the free space. After that, the scattered points are connected forming a web-like set of obstacle-free paths to the destination. Finally,

the final path is the shortest path among all feasible paths from the starting point to the goal point. The Probabilistic RoadMap algorithm (PRM) is an example of a sampling-based path planning scheme [19]. In a PRM planner, there are two critical parameters that directly affect the performance and form as tuning parameters between quality and swiftness, the number of samples parameter ( $N$ ), and the distance parameter ( $D$ ). The parameter  $N$  represents the number of scattered points in the free space. A PRM planner connects all points separated by a maximum distance of the parameter  $D$  in the map.

The quality-swiftness trade-off manifests clearly with the  $N$  and  $D$  parameters. PRM is provably probabilistically complete, meaning that as the number of sampled points ( $N$ ) approaches infinity, the probability to find a path approaches zero. However, a very high value of  $N$  will increase the complexity and thus, will increase the computational time required to find the shortest path. On the other hand, electing smaller  $N$  and  $D$  values will reduce the computational time, but the PRM planner might fail to calculate a feasible path, even if it exists. A new sampling-based path planning scheme is presented in [20]. The proposed path planning scheme is designed to find the “easiest” path in a mountainous rough area. The path planning scheme calculates a cost function and implements it on the map. The proposed scheme operates by scattering random points in areas that represent valleys and plain areas and assigns low cost values to the randomly scattered points. The Semi-Lazy Probabilistic RoadMap (SLPRM) is another sampling-based path planning scheme proposed in [21]. The SLPRM path planning scheme is a new variation of sampling-based schemes designed for motion planning of robotic manipulators. The SLPRM scheme combines some features of the lazy-PRM method and the basic Probabilistic RoadMap (PRM) method. The goal is to reduce the number of collision checks performed during planning. Therefore, the SLPRM scheme is able to achieve improvements in terms of the computational time and show some advantages in the reduced computational time compared to the classical PRM-based schemes or the lazy-PRM techniques.

On the other hand, the Grid-Based Schemes [22] place a grid on the map where each grid line intersection is represented by a grid point. The robot can freely move between adjacent grid points if there is no obstacle between the two grid points. Thus, compared to the  $N$  and  $D$  tuning PRM parameters, there is a critical parameter with grid-based schemes which is the grid resolution. The resolution parameter refers to the density of the grid which is represented by the number of grid points used to establish the grid. The criticality of selecting the resolution parameter reveals when dealing with real-time applications. Increasing the resolution of the grid will minimize the probability of not finding a feasible path if it exists. However, since the number of grid points will increase, the complexity will naturally increase which will affect the computational time negatively. On the other hand, creating a low-resolution grid can reduce the quality

of the generated path in terms of shortness or smoothness. In the worst case, the planner might even fail to find a feasible path even though an obstacle-free path exists from the starting point to the destination.

The A\* path planning technique is an example of a grid-based scheme [23], [24]. The methodology in finding the shortest path is to assign a cost value to each grid point which reflects the distance from each grid point to the destination. The final path obtained by the A\* planner is the path that minimizes the cost function which is represented in many A\* implementations by the distance to the destination. Although the A\* algorithm is widely known as a high-performance path planning technique, it suffers from the high complexity level which can be described as  $O(b)^d$ .

The D\* search algorithm is another grid-based path planning scheme [25]. A D\* planner begins searching for the obstacle-free path inversely from the destination to the starting point. Similar to A\*, D\* calculates a cost function and overlays it on the grid. The planner traces points with minimal cost values and when the next point is the starting point, the process ends, and the path is generated by tracking the pointer back to the destination.

The reduced path planning strategy was proposed recently in [2]. The main focus of the reduced path planning strategy is to reduce the computational time of sampling-based and grid-based techniques. The proposed reduced path planning strategy is based on the observation that sampling-based and grid-based techniques waste the processing efforts in less critical areas. For example, sampling-based schemes scatter the sample points randomly everywhere in the map even within areas that are far away from obstacles and from the destination. Therefore, the reduced strategy was proposed to optimize the operation of sampling-based and grid-based techniques by simply forcing the planner to only operate in areas around the obstacle. This is done by first establishing a straight line between the starting point and the destination. Then, the planner detects the obstacles that intersect with the straight line. After that, the basis algorithm such as PRM, A\*, or D\* is only executed in areas around each obstacle. Finally, all paths are connected to form the final obstacle-free path to the destination. Based on the experimental result provided in the paper, the reduced strategy had outperformed other schemes in terms of the execution speed. However, this improvement was at the expense of the path quality, since the reduced path planning strategy had produced relatively longer paths in some scenarios compared with PRM, D\*, and A\*.

Another state-of-the-art scheme was proposed in [5]. The proposed technique was based on the artificial potential field concept adopted in recent path planning schemes. The idea was to derive an objective function that imitates the effects of repulsive and attractive forces where obstacles generate repulsive forces while the goal point will be the source of the repulsive forces. The final path is obtained by minimizing the objective function using heuristic optimization techniques such as the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and the Artificial Bee Colony (ABC).

To overcome the swiftness-quality problem, this article introduces the new path planning strategy the “Sequential Linear Path (SLP)”. The proposed SLP strategy builds on the same main concept of the reduced scheme proposed in [2] which is the aim of reducing the search area to be within significant areas such as the areas around obstacles and areas surrounding the goal point. After computing the reduced path, as in [2], the planner enhances the path by finding possible sequential linear shortcuts in the path.

The proposed SLP path planning technique consists of five main steps. First, a linear line is designed from the starting point to the goal point. In the second step, all the obstacles that intersect with the linear path are detected through a specific planner. Next, the basis algorithm is applied on each obstacle in order to obtain a feasible path around the obstacle. After that, the planner connects all the paths together to establish an obstacle-free path from the starting point to the target. Finally, the path is enhanced by finding any possible linear shortcuts in the path.

This article proposes a new path planning strategy that aims to break the barrier between quality and swiftness. The strategy can be applied using any path planning algorithm as a basis algorithm, and can show improved results in terms of the computational time required and the path quality as well. Therefore, the proposed system is tested on well-known, very popular path planning techniques such as A\*, D\*, and PRM. To test the proposed path planning technique, computer vision algorithms are used to gather information about the surrounding environment and to create a map of the environment. Then, an obstacle-free path is calculated using the proposed SLP path planning strategy. Finally, to test the path quality in terms of smoothness, a kinematic controller is developed to ensure a good tracking of the desired path. Experimental comparative results prove the efficiency and the effectiveness of the proposed SLP strategy.

The contributions of the paper can be summarized as follows:

- A path planning system that aims to deal efficiently with the swiftness-quality trade-off. That is executing a short obstacle-free path to the destination in a real-time manner.
- A low-cost obstacle tracking mechanism.

The rest of this article is organized as follows. In Section II, a general description of the robotic system is illustrated. Section III briefly reviews the computer vision and image processing algorithms used to obtain the map. The proposed SLP path strategy is described in detail in Section IV. Section V models the robot dynamically and kinematically and presents the control design law. Section VI presents the comparative results. Finally, Section VII provides the conclusion.

## II. SYSTEM DESCRIPTION

The mobile robot used in the system is a wheeled mobile robot with two independent primary wheels fixed in the same axis in front, and a small caster wheel placed at the back of the

robot. A RGBD camera is fixed on the ceiling with a height of  $T$ . Figure 1 illustrates a general description of the robotic system.

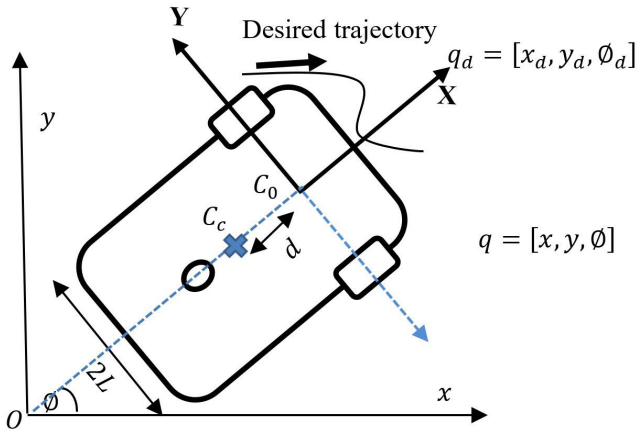


FIGURE 1. General description of the robotic system.

The desired position and orientation poses calculated by the proposed SLP path planning are represented by  $q_d = [x_d, y_d, \phi_d]^T$ . The coordinate  $q = [x, y, \phi]^T$  is the actual position and orientation measured by the robot's sensors.

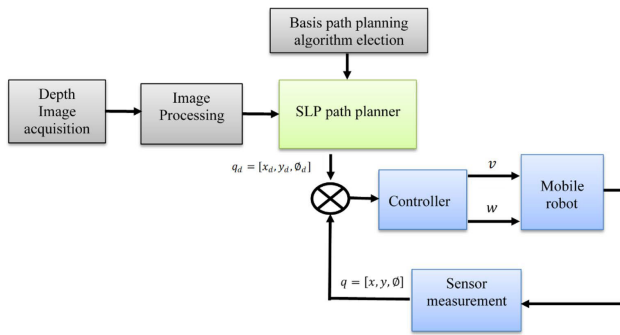


FIGURE 2. Block diagram of the control system.

In order to test the proposed path planning algorithm, a control system is implemented to track the path generated by the proposed path planning technique. Figure 2 illustrates a general block diagram of the control system. The figure illustrates the three main phases of the system, namely, the computer vision part, the proposed SLP path planning planner, and the kinematic controller. Computer vision algorithms are used to gather information about the surrounding environment such as the locations of obstacles. This information is essential to build the map that the proposed path planning technique will operate on. Once the obstacle-free path is generated from the starting point to the goal point, a kinematic controller assists the robot to track the desired trajectory, by generating the instantaneous linear and rotational velocities ( $v$  and  $\omega$ ) needed to follow the path. This is done based on the current measurements of the instantaneous pose of the robot with respect to the origin frame.

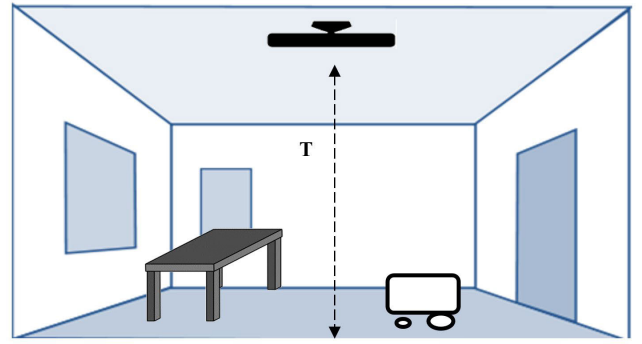


FIGURE 3. Obstacle detection method.

### III. VISION AND IMAGE PROCESSING ALGORITHMS

To build the map that represents the environment surrounding the mobile robot, vision and image processing algorithms are utilized to detect and find the coordinates of obstacles. The technique used to detect obstacles is illustrated in Figure 3. A depth-RGB sensor is fixed on the ceiling to detect and track obstacles. A depth sensor is a device that obtains an image where its pixel's values represent the corresponding real distance from the sensor. Since the sensor is mounted on the ceiling, and the height of the ceiling ( $T$ ) is known, then any pixel with a value less than  $T$ , can be considered as an obstacle.

Path planning techniques need a map that emphasizes the locations of obstacles. One of the most common formats of a map is the binary image format where black areas represent obstacles, and free-space areas are represented by a white area. To create such maps, the raw depth image must run first into a pre-processing phase. The goal of the pre-processing phase is to convert the raw depth image into a binary image emphasizing the locations of obstacles. The pre-processing phase is composed of three operations, namely the noise filtering operation, the binary conversion operation, and the obstacle padding operation.

#### A. NOISE FILTERING

It is known that the depth sensor is not an ideal sensor as it might fail to calculate the distance for some pixels. The reason is that modern depth sensors utilize the Time-of-Flight (ToF) technology. A modern depth camera such as Kinect has an infrared light emitter and an infrared light sensor. First, it emits a beam of an infrared light, and waits for reflections. Once the infrared light sensor receives a reflected infrared beam at a certain location, it marks the time needed for the reflection. The distance now can be easily calculated by simple kinematic equations since the speed of the infrared lights is known [26].

Sometimes, due to some refraction effects, the infrared light sensor does not receive any reflected infrared light at a certain location. Thus, the depth sensor fails to calculate the distance at that specific location (pixel). As a result, and to mark this failure, the depth sensor marks the corresponding pixel with a value of zero to indicate the failure.



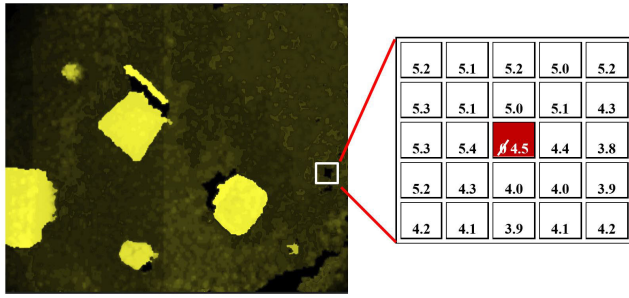


FIGURE 4. Noise elimination process.

These zero-valued pixels can be thought of a noise that must be removed before further processing. In this work, to eliminate this noise, a zero-valued pixel is replaced by the average value of the surrounding neighbor pixels. Figure 4 presents an example of the noise elimination process. In this example, the zero-valued pixel was replaced by the average value of 4.5.

### B. BINARY CONVERSION

Next, after the noise has been removed, the binary conversion process begins. The conversion is done through a simple thresholding process. The value of each pixel is checked. If the value is less than a certain threshold, which is  $T$  (the height of the depth sensor), then the pixel's value is converted to zero (black). Otherwise, the pixel is turned to 255 (white), and considered to be in an obstacle-free area.

$$\text{map}(x, y) = \begin{cases} 0, & \text{if } \text{depth\_image}(x, y) < T \\ 255, & \text{otherwise} \end{cases} \quad (1)$$

where  $T$  is the threshold value; *depth\_image* is the raw depth matrix after noise filtering; *map* is the new binary image.

### C. OBSTACLE PADDING

After creating the map, a safety margin is added around each obstacle. The purpose of this operation is that path planning algorithms treat robots as a point in the map. Thus, it might consider a path which is adjacent to an obstacle. In reality, this may cause some parts of the robot to collide into the obstacle. The size of the safety margin should not be less than half of the robot's size.

## IV. THE PROPOSED SLP PATH PLANNING ALGORITHM

Path planning techniques suffer from a classical trade-off between computational execution time and the path quality. Usually, targeting the path quality in a path planning algorithm, effects the execution time of the algorithm adversely. On the other hand, fast and simple path planning techniques do not guarantee having a high-quality path in terms of length and smoothness. Thus, path planning techniques published in the literature, focus on improving only one aspect, either reducing the computational time such as the real-time path planning algorithms, or enhancing the path quality such as the shortest-path path planning algorithms.

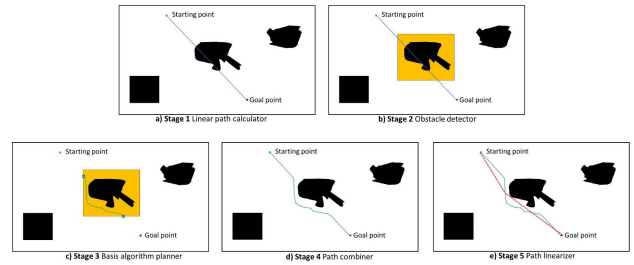


FIGURE 5. The different stages of the SLP strategy.

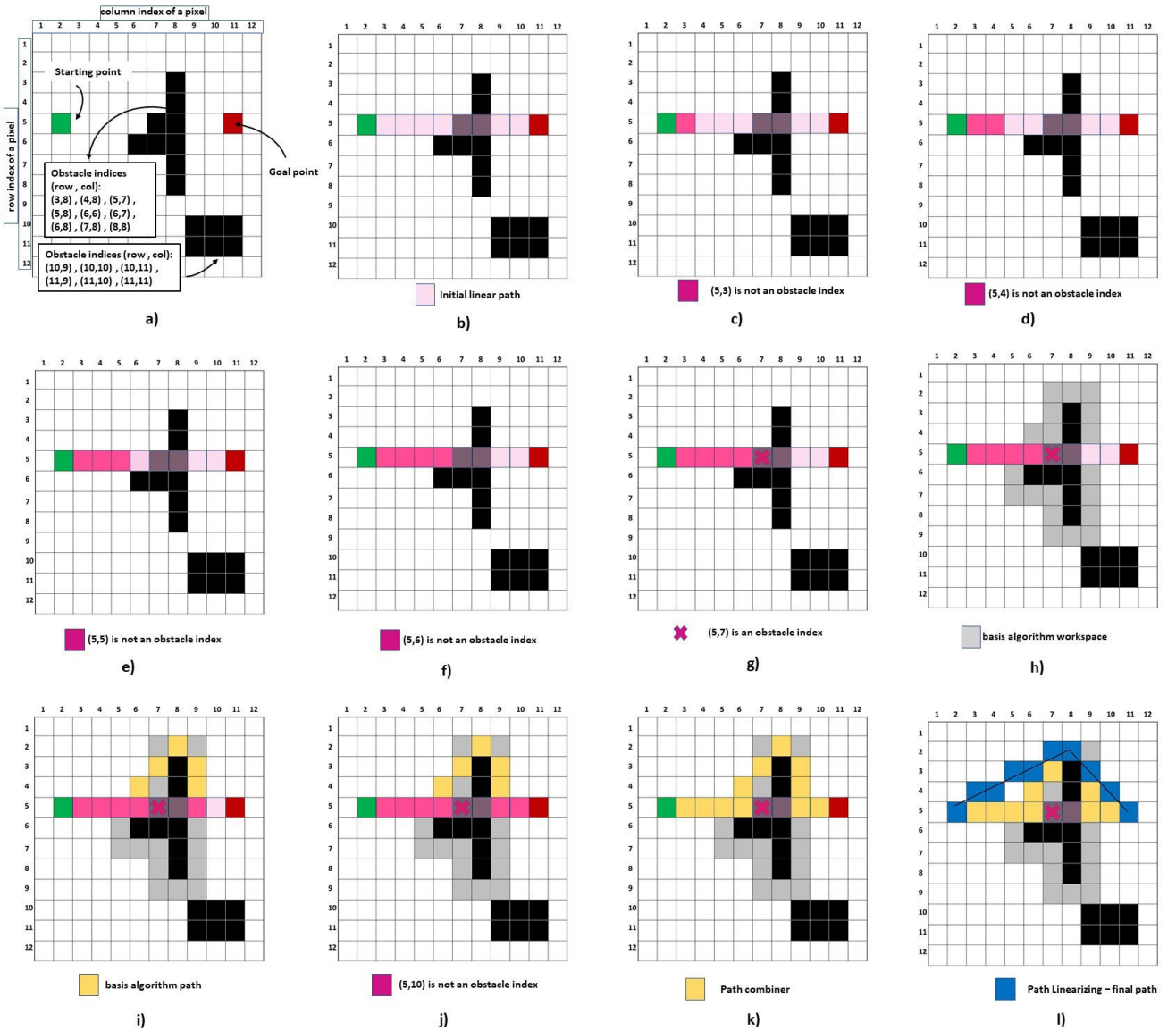
The proposed path planning strategy aims to break the trade-off barrier between the computational execution time and the path quality. The fundamental concept that allowed the proposed path planning strategy is the fact that “the shortest distance between two points is the straight line”. The strategy tries to find multiple linear paths from the starting point to the goal point, and combines these linear paths to find a feasible path from the starting point to the goal point.

The proposed path planning strategy needs a “basis path planning algorithm” to build on in order to find the final path. Thus, before starting with the proposed SLP technique, a well-known algorithm such as A\*, D\*, PRM is elected to be the basis algorithm to be used in the proposed SLP path planning strategy. Therefore, the proposed SLP technique have different versions based on the basis path planning algorithm used. For instance, SLP-A\*, SLP-D\*, and SLP-PRM are the path planning algorithms produced when using A\*, D\*, and PRM as the basis algorithms, respectively.

The proposed “SLP-x” uses the “x” path planning algorithm only around obstacles that intersect the linear path from the starting point to the goal point. This minimized working space area is the key feature that has allowed the proposed SLP path planning algorithm to show advanced results over the standard path planning algorithms (such as A\*, D\*, PRM) in terms of the computational time. On the other hand, the final path produced by the SLP algorithm is a combination of linear paths. These linear paths are linear shortcuts of the path produced by the basis algorithm around each obstacle. As known, the shortest path between any two points is the straight line. Thus, this linearizing process is the main reason for enhancing the path quality as well.

The proposed SLP-x path planning algorithm starts by computing a linear path from the starting point to the goal point. Then, the proposed algorithm finds obstacles intersecting with the linear path. Next, the basis algorithm is used to find an obstacle-free path around each obstacle intersecting with the linear path. Then, all paths, linear and non-linear paths are combined together forming an obstacle-free path from the starting point to the goal point. Finally, the path is enhanced by linearizing the path as much as possible, forming the final enhanced multiple linear paths from the starting point to the goal point.

Therefore, the proposed SLP path planning algorithm can be composed of five main stages. Figure 5 illustrates the five stages clearly. The first stage is the linear path calculator



**FIGURE 6.** Detailed illustration of the SLP path planning strategy.

which calculates a linear line from the starting point to the goal point (Figure 5-a). The second stage is the obstacle detector which finds all obstacles intersecting with the linear path (Figure 5-b). The third stage is the basis algorithm planner, where an obstacle-free path is generated around each obstacle using a selected basis path planning algorithm (Figure 5-c). The fourth stage is the path combiner, where all paths are combined to form a feasible path from the starting point to the goal point (Figure 5-d). Finally, the linearizer stage where the path is enhanced by finding any possible shortcut segments in the path (Figure 5-e).

The following sub-sections provide detailed discussions on the different parts of the proposed strategy. Section IV-A discusses the linear path calculator. The obstacle detector is presented in Section IV-B. Section IV-C describes the basis algorithm planner. Section IV-D demonstrates the

path combiner. Finally, Section IV-E illustrates, in a detailed discussion, the path linearizing process. Figure 6 shows a detailed step-by-step implementation of the proposed SLP path planning technique. Figure 6 shows a map of  $12 \times 12$  pixel. This small size is chosen just for clarification and illustration purposes. Figure 6-a shows the index of the starting point (5, 2), the index of the goal point (5, 11), and the indices of all obstacles. The next sub-sections will use this figure in their discussions.

#### A. LINEAR PATH CALCULATOR

The main concept of the proposed SLP technique is to make the path from the starting point to the goal point as linear as possible. Thus, it starts by establishing a linear line from the starting point to the goal point using the following equation:

$$y = m(x - x_1) + y_1 \quad (2)$$

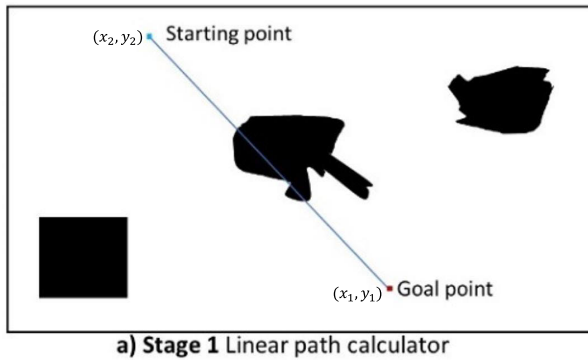


FIGURE 7. Linear path calculator.

where:  $x_1, y_1$  are the  $(x, y)$  coordinates of the starting point or the goal point.  $x$  is a vector that has the  $x$ -coordinate of every point in the line.  $y$  is a vector that has the  $y$ -coordinate of every point in the line.  $m$  is the slope of the linear line given as follows:

$$m = (y_2 - y_1) / (x_2 - x_1) \quad (3)$$

where the two points  $(x_1, y_1)$  and  $(x_2, y_2)$  are any two points in the line. The only constrain is that  $y_1$  must be smaller than  $y_2$ . Up to this point, the only known points are the starting point and the goal point. Thus, the election of the two points  $(x_1, y_1)$  and  $(x_2, y_2)$  will be based on the coordinate value of both points. If the  $y$ -coordinate of the goal point is less than the  $y$ -coordinate of the starting point, then  $(x_1, y_1)$  is the goal point, and  $(x_2, y_2)$  is the starting point. Otherwise, if the  $y$ -coordinate of the starting point is less than the  $y$ -coordinate of the goal point, then  $(x_1, y_1)$  is the starting point and  $(x_2, y_2)$  is the goal point. For example, in Figure 7, the  $y$ -coordinate of the goal point is less than the  $y$ -coordinate of the starting point. Thus,  $(x_1, y_1)$  is the goal point and  $(x_2, y_2)$  is the starting point. Figure 7 shows the output of the linear path calculator stage. If there are no obstacles intersecting with the linear path, then this linear path is the final path produced by the proposed SLP technique.

In Figure 6-b, the starting point and the goal point have the same  $y$ -coordinate value. Hence, the slope  $m$  will equal to zero, which means that the initial line will be a horizontal line that crosses the two points.

## B. OBSTACLE DETECTOR

The obstacle detector is now used to find obstacles that intersect with the initial linear path calculated from the previous stage. This is simply done by inspecting the indices of every obstacle point and every point in the linear path. The obstacle detector starts checking each point in the linear path, starting from the starting point towards the goal point and checks whether the point index matches any obstacle index.

The operation of the obstacle detector is illustrated in Figures (6-[c-g]). The process starts with the linear path point located after the starting point which is the point (5, 3) in Figure 6-c. The point (5, 3) does not match any obstacle index listed in Figure 6-a. Thus, this point (5, 3) is

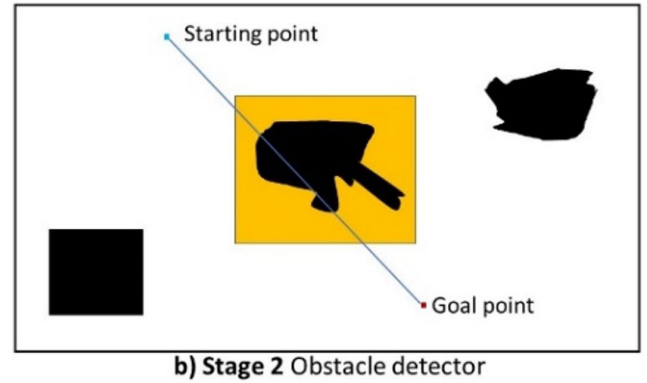


FIGURE 8. Obstacle detector.

accepted, and added to the final path points list. The obstacle detector moves to the next linear path point toward the goal point (5, 4), which is also not an obstacle index. The index (5, 4) is also accepted to be in the final path points list (Figure 6-d). The same process is performed in Figures 6-e, and 6-f.

In Figure 6-g, the linear path checking point (5, 7) is founded to match an obstacle index. Once the detector detects that the linear line intersects with an obstacle, it creates a small margin around this obstacle. The margin with the obstacle creates a mini-map which is sent to the basis path planner. The objective of the basis planner is to find a path around this obstacle. Thus, the first point of the linear path located within the margin (index 5, 6) is now the starting point for the basis function. On the other hand, the last point of the linear path located within the margin (index 5, 10) is the goal for the basis function. The obstacle detector does this operation for each obstacle intersecting with the linear path.

The width of the margin is a performance tuning parameter that is related to the swiftness-quality trade-off. Increasing the width of the margin increases the complexity of the algorithm, which results in increasing the computational time. The reason is that by increasing the width of the margin, the number of Samples  $N$  (for PRM), or the resolution parameter (for A\* and D\*) will naturally increase, which means more points to be processed. As a result, the technique will be affected adversely in terms of the computational time. However, the increase in the margin width increases the possibility to have a better path quality in terms of length and smoothness.

Figure 8 shows the output of the obstacle detector operation. The yellow box area is sent to the basis algorithm to find a feasible path around this obstacle.

## C. BASIS ALGORITHM PLANNER

The basis planner can be any path planning technique published in the literature. The basis planner receives a mini-map where it has an obstacle with a free-space around it (the margin added in the previous stage), along with the starting point and the goal point determined from the previous stage.

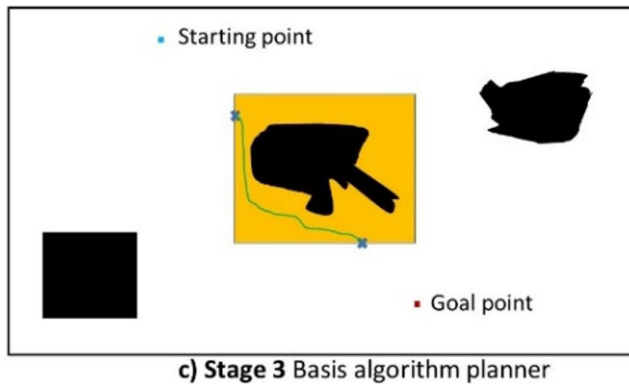


FIGURE 9. Basis algorithm planner.

The objective of the basis planner is to find an obstacle-free path from the starting point to the goal point.

In the basis planner, there are another trade-off tuning parameters. Depending on the basis algorithm used, these parameters can vary. For grid-based algorithms (such as A\* and D\*), the grid size is a critical parameter that can affect the overall performance of the proposed SLP technique. On the other hand, sampling-based algorithms (such as PRM) have the “number of samples N” parameter. As explained in the previous stage, minimizing the values of these parameters can help reducing the computational time, but that may affect the quality of the generated path. In the proposed SLP technique, these values will be relatively small since the working area is a mini-map which is much smaller area then the whole map. However, these small values suit the working space, which means that there is no significant adverse effect on the path quality.

Figure 6-i shows the generated path from the basis planner stage. To be noted that the path shown in Figure 6-i is just for illustration, and not the actual path to be generated by the basis algorithm, as the path will vary based on the basis algorithm used. Figure 9 shows the output of the basis algorithm planning operation.

#### D. PATH COMBINER

The path combiner stage starts once all of the “basis paths” are generated around each obstacle. In this stage, the different non-linear segments of the path are combined with the initial linear segments, which produces the final obstacle-free path from the starting point to the goal point. Figure 6-k shows the path combiner process, while Figure 10 shows the output of the path combiner operation.

#### E. PATH LINEARIZER

The path linearizer is responsible of finding linear shortcuts in the path. This is a critical process in the proposed SLP strategy which aims to improve the quality of the path in terms of length and smoothness. The result of this process is a path composed of multiple sequential linear paths from the starting point to the goal point. The process of the path linearizer is illustrated clearly in Figure 11. Figure 11-a shows

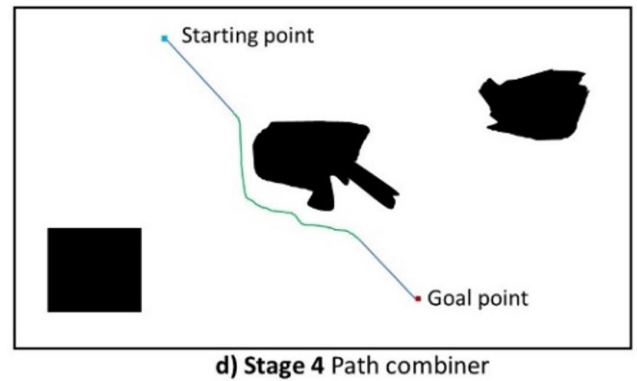


FIGURE 10. Path combiner.

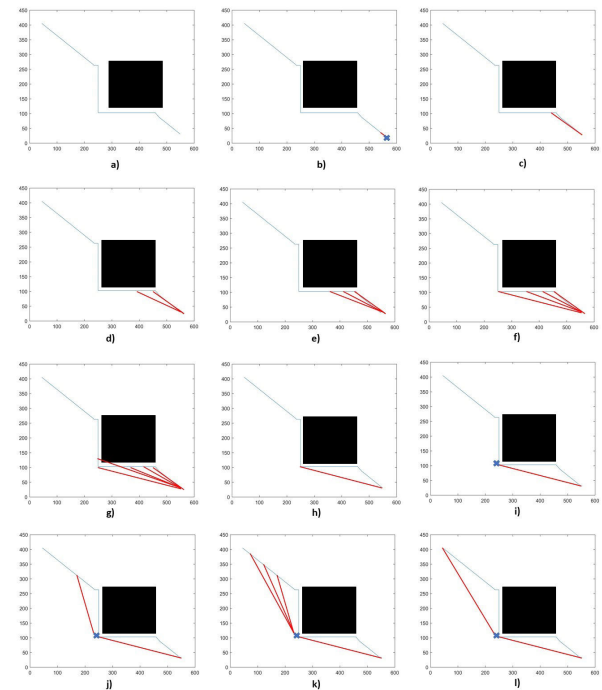


FIGURE 11. Detailed illustration of the linearization process.

the SLP path before linearization. The linearization process starts adversely from the goal point, and tries to figure out possible linear shortcuts towards the starting point (Figure 11-b). The linearizer works by establishing straight lines using Equation 2 between the goal point and points from the path. If the line does not intersect with an obstacle, then this line is a valid shortcut. At the end, the linearizer selects the straight line with the longest length, since it is the best shortcut possible.

To make the idea clearer, Let us assume that the path has n points, the point n will be the starting point for the linearizer, and the point n-1 is the goal point for the linearizer. The linearizer establishes a straight line between the two points using Equation 2. If the line does not intersect with an obstacle, then this line is a valid shortcut (Figure 11-b). Next, the goal point (of the linearizer) moves to the next point, and checks if the linear shortcut is valid (Figures 11-c,11-d,11-e,11-f). Once



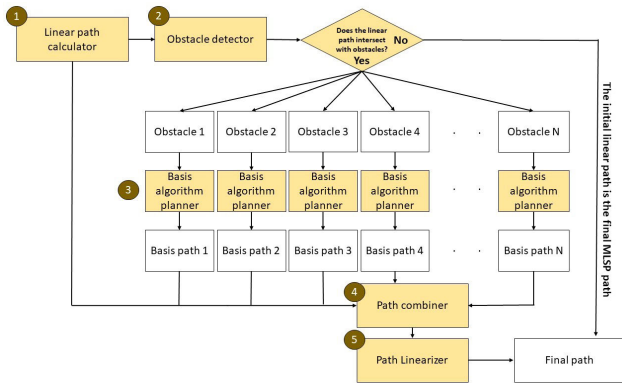


FIGURE 12. The general diagram of the SLP strategy.

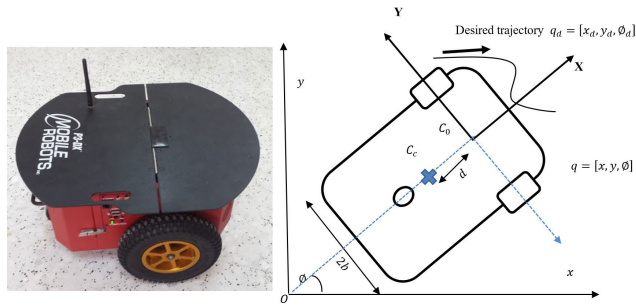


FIGURE 13. Mobile robot with two actuated wheels.

the straight line intersects with an obstacle as in Figure 11-g, the linearizer selects the previous line (in Figure 11-f) to be part of the final SLP path (Figure 11-h). After that, the goal point of the previous step becomes the starting point of the next step (Figure 11-i), and the same checking process is repeated (Figures 11-j, 11-k). If the goal point of the linearizer reaches the end of the path, the latest straight line is added to be part of the final SLP path, and the linearization process ends (Figure 11-l).

Figure 12 shows the five different stages as a diagram. The diagram shows an important feature supported by the strategy which is parallelization. It is clear from Figure 12, that Stage 3 can be done in parallel. Once the obstacle detector detects the obstacles that are intersecting with the linear path, multiple basis algorithm planners can operate in parallel to calculate an obstacle-free path around each obstacle.

## V. DYNAMIC AND CONTROL DESIGN

We consider the mobile robot with two actuated wheels, which is shown in Figure 13 [27].

With regard to the mobile robot shown in Figure 13, the width of the mobile robot is  $2b$ , the radius of the wheel is  $r$ ,  $O_{xy}$  is the reference coordinate system and  $C_{0XY}$  is the coordinate system fixed to the mobile robot. The origin of the coordinate system  $P_{0XY}$  is  $C_0$ . The center of mass of the mobile robot is  $C_c$ , which is on the X-axis.  $d$  represents the distance from  $P_0$  to  $C_c$ . The general dynamic equation of the mobile robot with two actuated wheels is described as

follows [28]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)\tau - A^T(q)\lambda \quad (4)$$

where  $\tau = [\tau_r, \tau_l]^T \in R^r$  is the input torque vector.  $\tau_r$  and  $\tau_l$  are the right and left torque respectively.  $M(q) \in R^{nn}$  is a symmetric and positive-definite inertia matrix,  $C(q, \dot{q}) \in R^{n \times n}$  is the centripetal and Coriolis vector,  $G(q) \in R^n$  is the gravitational vector,  $B(q) \in R^{nr}$  is the input transformation matrix, and  $A(q) \in R^{mn}$  is the matrix associated with the constraints.  $\lambda \in R^m$  is the vector of constraint forces. The configuration of the mobile robot can be represented by five generalized coordinates:  $q = [x, y, \phi, \theta_r, \theta_l]^T$ ; where  $x$  and  $y$  are the coordinates of  $P_0$ ,  $\phi$  is the direction, and  $\theta_r, \theta_l$  are the angles of the right and left driving wheels. The wheels are assumed to roll and do not slip. Then, there exist three constraints; the velocity of  $P_0$  must be in the direction of the axis of symmetry and the wheels must not slip.

$$\dot{y} \cos \phi - \dot{x} \sin \phi = 0 \quad (5)$$

$$\dot{x} \cos \phi - \dot{y} \sin \phi + b\dot{\phi} = r\dot{\theta}_r \quad (6)$$

$$\dot{x} \cos \phi - \dot{y} \sin \phi - b\dot{\phi} = r\dot{\theta}_l \quad (7)$$

The kinematic constraints can be denoted as:

$$A(q)\dot{q} = 0 \quad (8)$$

where

$$A(q) = \begin{bmatrix} \sin \phi & -\cos \phi & 0 & 0 & 0 \\ \cos \phi & \sin \phi & b & -r & 0 \\ \cos \phi & \sin \phi & -b & 0 & -r \end{bmatrix}.$$

when selecting a full rank matrix  $S_1(q)$  to be a basis of null space  $A(q)$ , the constraint equation becomes:

$$A(q)S_1(q) = 0 \quad (9)$$

where

$$S_1 = \begin{bmatrix} \frac{r}{2} \cos \phi & \frac{r}{2} \cos \phi \\ \frac{r}{2} \sin \phi & \frac{r}{2} \sin \phi \\ \frac{2b}{1} & -\frac{r}{2b} \\ 0 & 1 \end{bmatrix}.$$

Therefore, we have:

$$\dot{q} = S_1(q) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (10)$$

where  $v_1$  and  $v_2$  represent the angular velocities of the right and left wheels. If we consider  $v; \omega$  as the linear and angular velocities of the mobile robot, the relation between  $v; \omega$  and  $v_1; v_2$  can be explained as:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (11)$$

The ordinary form of a mobile robot with two actuated wheels can be obtained by substituting 11 in 10 as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ \sin \phi & -\cos \phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (12)$$

Note that the objective is to track a reference trajectory given from the proposed SLP path planner. The velocity control based on kinematic model is designed to develop the desired forward and rotational velocities.  $x_r$ ,  $y_r$  and  $\phi_r$  denote the reference trajectory. The desired velocity is denoted by  $v_r$  and  $\omega_r$ . Using the Kanayama transformation [28], the tracking errors are obtained as follows:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \phi_r - \phi \end{bmatrix} \quad (13)$$

$e_1$ ,  $e_2$  and  $e_3$  describe the difference of position and direction of the reference robot from the real robot. The related error dynamics are expressed as follows:

$$\dot{e}_1 = \omega e_2 - v + v_r \cos \phi \quad (14)$$

$$\dot{e}_2 = -\omega + v_r \sin \phi \quad (15)$$

$$\dot{e}_3 = \omega_r - \omega \quad (16)$$

**Proposition:** The input  $v$  and  $\omega$  which make  $e_1$ ,  $e_2$  and  $e_3$  converge to zero are given by the following expression.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 \sin e_3 \end{bmatrix} \quad (17)$$

where  $k_1 > 0$ ;  $k_2 > 0$  and  $k_3 > 0$  are the controller gains [28]. To prove this proposition, let us define the following positive Lyapunov function:

$$V = \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2 + \frac{1 - \cos e_3}{k_2} \quad (18)$$

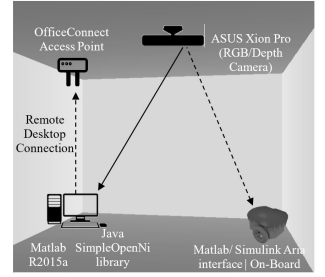
The time derivative of  $V$  is given as follows:

$$\dot{V} = e_1 \dot{e}_1 + e_2 \dot{e}_2 + \dot{e}_3 \frac{\sin e_3}{k_2} \quad (19)$$

Using Equations 14, 15, 16 and 17, we obtain:

$$\dot{V} = -k_1 e_1^2 - k_3 \frac{\sin^2 e_3}{k_2} \leq 0 \quad (20)$$

The time derivative of  $V(t)$  is negative because  $k_1$  and  $k_3$  are positive gain. By Equation 20,  $V$  is monotonously decreasing. Therefore,  $e_1 \in L_\infty$ ,  $e_2 \in L_\infty$  and  $e_3 \in L_\infty$ , and  $e_1 \in L_2$ ;  $e_3 \in L_2$ . According to Equations 14, 15, and 16, we know that  $\dot{e}_1$ ,  $\dot{e}_2$  and  $\dot{e}_3$  are bounded and therefore  $\dot{e}_1$ ,  $\dot{e}_2$ , and  $\dot{e}_3$  are uniformly continuous. Then, using Barbalat's theorem [27], we can conclude that  $e_1 \rightarrow 0$ ,  $e_2 \rightarrow 0$ , and  $e_3 \rightarrow 0$  as  $t \rightarrow \infty$  and the error dynamics are asymptotically stable.



**FIGURE 14.** (Left figure): The Pioneer P3-DX robot. (Right figure): The experimental setup.

**TABLE 1.** Experiment parameters.

Parameters	Values
Image Processing:	
$H$ [Threshold] ( $m$ )	3.8
Robot:	
$m_c$ ( $kg$ )	7.4
$m_\omega$ ( $kg$ )	0.8
$L$ ( $m$ )	0.19
$r$ ( $m$ )	0.0975
$d$ ( $m$ )	0.1
$I_c$ ( $kgm^2$ )	0.020
$I_m$ ( $kgm^2$ )	0.00475



**FIGURE 15.** RGB image of the lab.

## VI. EXPERIMENTAL RESULTS

The experiments were performed using an Asus Xion pro depth camera which is mounted on the ceiling. The robotic system used in these experimental results is Pioneer P3-DX shown in Figure 13. The experimental setup is also shown in Figure 14.

The parameters used in the following experiments are shown in Table 1. The robot parameters are specific to the robot used in the experiment (The Pioneer P3-DX). These parameters describes the P3-DX robot such as the weight, inertia and dimensions. The Image Processing

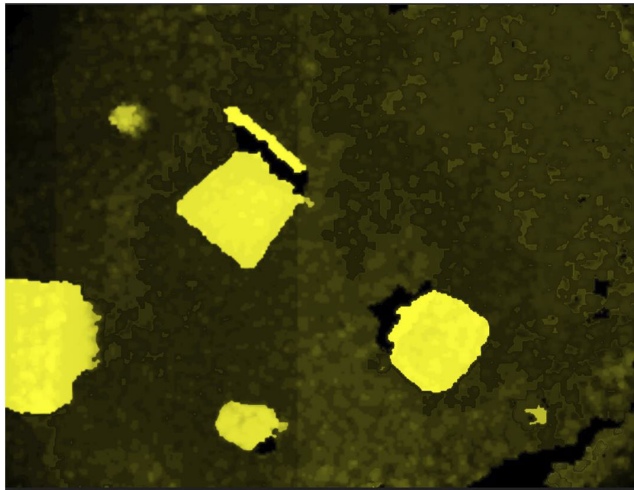


FIGURE 16. Raw depth image of the lab.

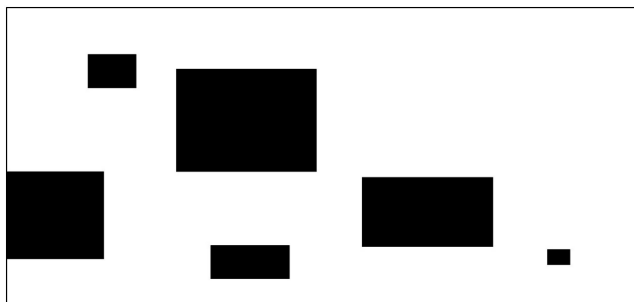


FIGURE 17. Raw depth image converted to binary map.

parameter shown in Table 1 (Threshold  $H$ ) is the height of the lab in meters, so these are not tunable parameters.

Figures 15 and 16 show the lab where the experiment was done. Figure 15 shows the RGB image of the lab, while Figure 16 shows the depth image of the lab.

It is clear from Figure 16, that obstacles are emphasized with a brighter color. Thus, as in Figure 17, the image pre-processing phase was able to successfully convert the raw depth image into a binary image, where black pixels are obstacles, and white pixels represent free space. This binary image will be the map that next experiments will be tested on. Figure 18 shows the different scenarios to be tested. In Scenario A, the initial linear line does not intersect with any obstacle - even though some obstacles are available in the map - (Figure 18-a). One obstacle intersects with the linear line in Scenario B (Figure 18-b). Figure 18-c shows the third scenario C where two obstacles intersect with the linear line. Finally, Figure 18-d shows Scenario D where the linear line intersects with four obstacles.

Figure 19 shows the implementation of the SLP strategy with a comparison with its corresponding basis scheme on the fourth scenario (Scenario D), since this scenario has the maximum number of obstacles intersecting with the initial linear path. The red dashed line in the SLP Figures (b, d and f) is the final SLP path produced by the path linearizer. It is

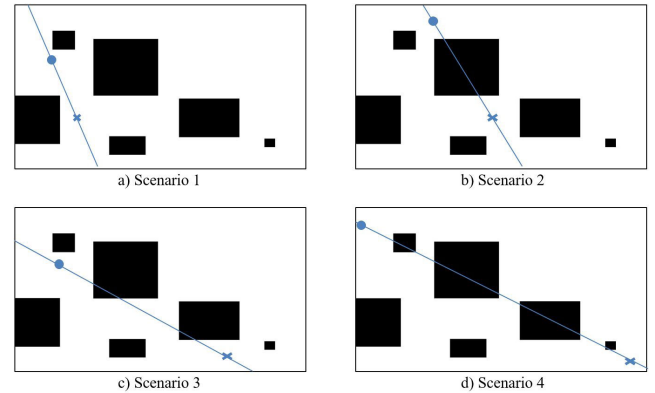


FIGURE 18. Different scenarios based on the number of obstacles intersecting with the initial linear path.

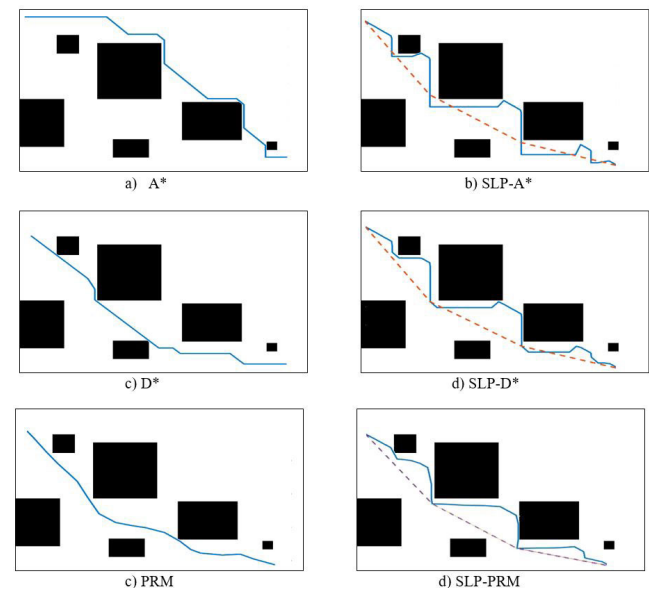


FIGURE 19. SLP path versus the basis path. The red dashed line in the SLP figures (b, d and f) is the final SLP path produced by the path linearizer.

clear from the figures, that SLP paths appear much smoother than the corresponding basis path.

#### A. COMPUTATION TIME COMPARISON

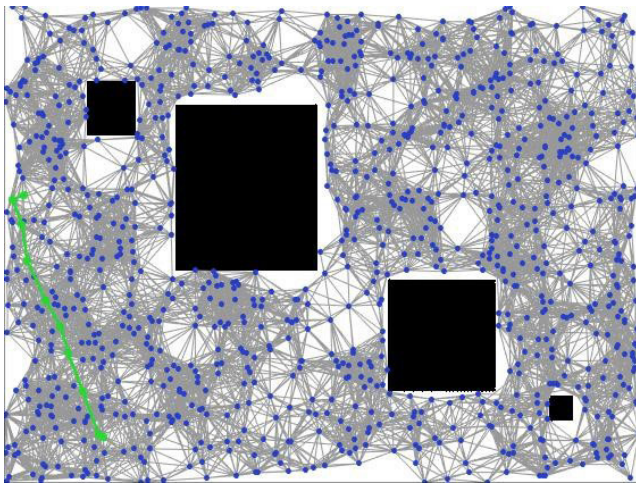
This section provides a computational time comparative results discussion. The proposed scheme is a real-time path planning scheme, and thus, it should show improved results in terms of computation time. Table 2 shows comparative results between the proposed SLP scheme and its corresponding basis scheme.

Table 2 manifests clearly the advantage of the proposed SLP path planning strategy over the corresponding basis path planning algorithm. To be noted here, that the computer used in the experiments has a limited processing power with an intel i5 processor with 4 GB RAM. Therefore, the computational time listed in Table 2 is expected to be less with a higher processing power. However, even though, the proposed SLP scheme was able to execute and obtain a path in a real-time manner, all scenarios and different SLP versions were



**TABLE 2. Computational time comparison.**

Scenario A (No obstacles intersecting with the linear line) Figure 18-a						
Algorithm	A*	SLP-A*	D*	SLP-D*	PRM	SLP-PRM
Average execution time (sec)	0.395	0.093	1.122	0.093	3.150	0.093
Improvement percentage	76.46%		91.71%		97.05%	
Scenario B (1 obstacle intersecting with the linear line) Figure 18-b						
Algorithm	A*	SLP-A*	D*	SLP-D*	PRM	SLP-PRM
Average execution time (sec)	0.494	0.191	1.156	0.376	3.496	0.538
Improvement percentage	61.34%		67.47%		84.61%	
Scenario C (2 obstacles intersecting with the linear line) Figure 18-c						
Algorithm	A*	SLP-A*	D*	SLP-D*	PRM	SLP-PRM
Average execution time (sec)	0.577	0.371	1.449	0.641	3.321	0.770
Improvement percentage	35.70%		55.76%		76.81%	
Scenario D (4 obstacles intersecting with the linear line) Figure 18-d						
Algorithm	A*	SLP-A*	D*	SLP-D*	PRM	SLP-PRM
Average execution time (sec)	0.872	0.457	1.263	0.5439	3.484	0.979
Improvement percentage	47.59%		56.94%		71.90%	

**FIGURE 20. Using PRM in Scenario 1.**

able to calculate the free-obstacle path in less than a second. A noticeable result, SLP strategy was able to reduce the computational to over than 50% for most cases. Moreover, for Scenario A, the computational time has reduced from 3.150 seconds for PRM to 0.093 seconds for SLP-PRM with an improvement percentage of 97.05%. It is noticeable that Scenario A (the initial linear path does not intersect with an obstacle) has achieved the best improvement percentage, especially for SLP-PRM over PRM. The reason can be known easily from Figure 20. Figure 20 shows how it is inefficient to use a classical path planning method (such as PRM) with such scenario. PRM distributes a huge number of points (the N parameter) and connect each two points having a distance less than (the D parameter). This operation usually increases the computational time to find a feasible path from the starting point to the goal point. However, in a scenario such as the one showed in Figure 20, it does not make sense to distribute points and connect lines, as all what is required is a straight line from the starting point to the goal point.

Table 3 shows another comparison with the state-of-the-art path planning scheme [5] described earlier in Section I. Based on Table 3, the proposed scheme was able to outperform the technique proposed in [5]. The variation that used the PSO technique has achieved the fastest execution speed. This is due that the PSO optimization technique converges faster

**TABLE 3. Comparing the three optimization techniques GA, PSO and ABC for path planning.**

No. of obstacles	Technique	Execution time (sec)
1	Baziyad et. al.-GA [5]	2.2
	Baziyad et. al.-PSO [5]	0.26
	Baziyad et. al.-ABC [5]	1.4
	Proposed SLP-A*	0.191
	Proposed SLP-D*	0.376
	Proposed SLP-PRM	0.538
2	Baziyad et. al.-GA [5]	2.4
	Baziyad et. al.-PSO [5]	0.4
	Baziyad et. al.-ABC [5]	1.3
	Proposed SLP-A*	0.371
	Proposed SLP-D*	0.641
	Proposed SLP-PRM	0.770
4	Baziyad et. al.-GA [5]	0.457
	Baziyad et. al.-PSO [5]	0.5
	Baziyad et. al.-ABC [5]	1.3
	Proposed SLP-A*	0.457
	Proposed SLP-D*	0.5439
	Proposed SLP-PRM	0.9790

**TABLE 4. Path length comparison.**

Scenario A (No obstacles intersecting with the linear line) Figure 18-a						
Algorithm	A*	SLP-A*	D*	SLP-D*	PRM	SLP-PRM
Path length (m)	1.65	1.50	1.52	1.50	1.049	1.036
Improvement percentage	9.09%		1.32%		1.24%	
Scenario B (1 obstacle intersecting with the linear line) Figure 18-b						
Algorithm	A*	SLP-A*	D*	SLP-D*	PRM	SLP-PRM
Path length (m)	2.31	2.31	2.20	2.30	1.801	1.509
Improvement percentage	0.00%		-4.55%		16.21%	
Scenario C (2 obstacles intersecting with the linear line) Figure 18-c						
Algorithm	A*	SLP-A*	D*	SLP-D*	PRM	SLP-PRM
Path length (m)	3.17	2.82	2.68	2.84	3.850	4.507
Improvement percentage	11.04%		-5.97%		-17.06%	
Scenario D (4 obstacle intersecting with the linear line) Figure 18-d						
Algorithm	A*	SLP-A*	D*	SLP-D*	PRM	SLP-PRM
Path length (m)	4.94	4.52	4.45	4.54	4.10	4.52
Improvement percentage	8.50%		-2.02%		-10.24%	

than other heuristic optimization techniques. However, even though, the proposed SLP-A\* scheme was able to execute faster than the PSO variation of the proposed technique in [5].

## B. PATH LENGTH COMPARISON

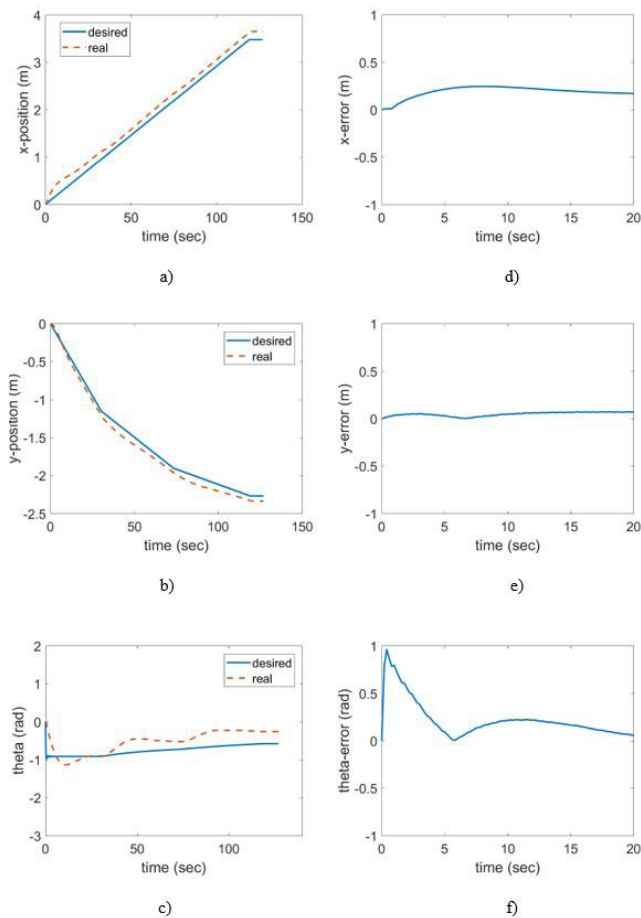
The quality of a path can be studied in its length and smoothness. The proposed SLP strategy aims to introduce a real-time path planning technique by reducing the computational time needed to find a feasible path. Thus, this should show an adverse effect on the path quality represented by the path length and smoothness. Table 3 shows a detailed comparison between the proposed SLP strategy and the corresponding basis algorithm in terms of path length.

Table 3 shows an outstanding performance. The traditional trade-off between computational time and path quality did not adversely affect significantly. The proposed SLP strategy has actually, in many cases, improved the path quality by reducing the path length. For example, in scenario C, the proposed SLP was able to reduce the path length from 3.17 meters by



**TABLE 5.** Mean Square Error comparison (MSE).

Scenario A (No obstacles intersecting with the linear line) Figure 18-a									
	A*	SLP-A*	IP	D*	SLP-D*	IP	PRM	SLP-PRM	IP
MSE X	0.0075	0.0231	-67.53%	0.0593	0.0260	56.16%	0.0624	0.0272	56.41%
MSE Y	0.1719	0.0190	88.95%	0.1476	0.0147	90.04%	0.1372	0.0130	90.52%
MSE Theta	0.4257	0.0686	83.89%	0.2865	0.1002	65.03%	0.1768	0.1243	29.69%
Scenario B (1 obstacle intersecting with the linear line) Figure 18-b									
	A*	SLP-A*	IP	D*	SLP-D*	IP	PRM	SLP-PRM	IP
MSE X	0.0337	0.0056	83.38%	0.0265	0.0049	81.51%	0.0905	0.0119	86.85%
MSE Y	0.0957	0.0836	12.64%	0.0958	0.1017	-6.16%	0.2628	0.0500	80.97%
MSE Theta	3.4846	0.4982	85.70%	2.5509	0.5858	77.04%	0.1620	0.4612	-64.87%
Scenario C (2 obstacles intersecting with the linear line) Figure 18-c									
	A*	SLP-A*	IP	D*	SLP-D*	IP	PRM	SLP-PRM	IP
MSE X	0.0460	0.0296	35.65%	0.0903	0.0276	69.44%	0.5788	0.0263	95.46%
MSE Y	0.0353	0.0028	92.07%	0.0982	0.0040	95.93%	0.3118	0.0057	98.17%
MSE Theta	0.0385	0.0393	-2.04%	0.4202	0.0546	87.01%	0.0584	0.0817	-28.15%
Scenario D (4 obstacles intersecting with the linear line) Figure 18-d									
	A*	SLP-A*	IP	D*	SLP-D*	IP	PRM	SLP-PRM	IP
MSE X	0.0579	0.0236	59.24%	0.0864	0.0226	73.84%	0.7997	0.0238	97.02%
MSE Y	0.1019	0.0064	93.72%	0.1359	0.0072	94.70%	0.4198	0.0063	98.50%
MSE Theta	0.0663	0.0869	-23.71%	0.5979	0.0952	84.08%	0.0491	0.0846	-41.96%

**FIGURE 21.** Kinematic control of the platform subsystem: (a) Tracking trajectory of x-position, (b) Tracking trajectory of y-position (c) Tracking trajectory of  $\phi$ -direction, (d) Tracking error of x-position, (e) Tracking error of y-position (f) Tracking error of  $\phi$ -direction.

A\* to 2.82 meters by SLP-A\* with a percentage improvement of 11.04%. Scenario A has showed an improvement for all scenarios and different SLP versions. That is since the shortest distance between any two points is simply the straight line.

**TABLE 6.** Summaries improvement percentages.

Scenario	Method	Improvement percentage			
		In time execution	In path length	In smoothness	Average
Scenario A	A*	76.46%	9.09%	35.10%	40.22%
	SLP-A*				
	D*	91.71%	1.32%	70.41%	54.48%
	SLP-D*				
	PRM				
Scenario B	SLP-PRM	97.05%	1.24%	58.87%	52.39%
	A*	61.34%	0.00%	60.57%	40.64%
	SLP-A*				
	D*	67.47%	-4.55%	50.80%	37.91%
	SLP-D*				
Scenario C	PRM				
	SLP-PRM	84.61%	16.21%	5.63%	31.73%
	A*	35.70%	11.04%	41.88%	29.54%
	SLP-A*				
	D*	55.76%	-5.97%	84.13%	44.64%
Scenario D	SLP-D*				
	PRM				
	SLP-PRM	76.81%	-17.06%	51.24%	37.00%
	A*	47.59%	8.50%	40.63%	32.24%
	SLP-A*				
Scenario E	D*	56.94%	-2.02%	84.21%	46.38%
	SLP-D*				
	PRM				
	SLP-PRM	71.90%	-10.24%	41.07%	34.24%
	SLP-PRM				

### C. PATH SMOOTHNESS COMPARISON (TRACKABILITY)

The last evaluation metric in the experimental section is the path smoothness, or the trackability degree. For the following experiments, the control law designed in Section 6 is used to help the robot tracking the path. Scenario D has been selected in the following tracking experiments. Table 4 shows the Mean Square Error (MSE) value between the desired path calculated from the path planning technique and the actual tracked path.

The tracking experimental results for SLP-A\* obtained are shown in Figure 21.

Table 6 summarizes the performance of the proposed SLP technique in a table. To be noted that the smoothness improvement percentage shown in Table 6 is the average MSE value of the X,Y and Theta MSE values shown in Table 5. The table clearly shows that the SLP technique was always able to improve the technique with a very high improvement percentage. Even though that for some scenarios, there were some adverse effect on the path length which reached in the worst

case to  $-17.06\%$ , the time and smoothness improvement percentages compensate this negative effect. In that worst path length case, the overall improvement percentage average reached up to  $37.00\%$ . Moreover, the overall improvement percentage did not achieve lower than  $29.54\%$  improvement percentage, which indicates an outstanding performance of the proposed SLP technique.

## VII. CONCLUSION

This article has presented the new path planning technique “Sequential Linear Path (SLP)”. The proposed path planning technique aimed to overcome the traditional trade-off between swiftness and path quality. The main idea is to find sequential linear paths from the starting point to the goal point. The objective of the proposed system is to improve and enhance the operation of any existing path planning technique. Therefore, the proposed system was tested on well-known, very popular path planning techniques such as  $A^*$ ,  $D^*$ , and PRM. The execution time required to find a feasible path to the destination is minimized by operating the basis path planning algorithm only within areas around each obstacle that intersects with the straight line to the goal point. This minimized working space is the main reason for obtaining reduced computational time results. Finally, the path linearizer improves the quality of the path by finding linear shortcuts in the path. Experimental results have shown an outstanding performance of the proposed SLP path planning system. The technique was able to reach an overall improvement percentage of  $54.48\%$  for some cases.

Future enhancements on the proposed scheme are planned to be in the direction of supporting dynamic environments. The proposed technique can be modified to support path findings in dynamic environments. Moreover, the proposed path planning system will be adapted to operate on 3D maps for Unmanned Aerial Vehicle (UAV) applications.

## REFERENCES

- [1] N. Gageik, P. Benz, and S. Montenegro, “Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors,” *IEEE Access*, vol. 3, pp. 599–609, 2015.
- [2] R. Fareh, M. Baziyad, M. H. Rahman, T. Rabie, and M. Bettayeb, “Investigating reduced path planning strategy for differential wheeled mobile robot,” *Robotica*, vol. 38, no. 2, pp. 235–255, Feb. 2020.
- [3] E. I. Al Khatib, M. A. K. Jaradat, and M. F. Abdel-Hafez, “Low-cost reduced navigation system for mobile robot in indoor/outdoor environments,” *IEEE Access*, vol. 8, pp. 25014–25026, 2020.
- [4] M. Wehner, R. L. Truby, D. J. Fitzgerald, B. Mosadegh, G. M. Whitesides, J. A. Lewis, and R. J. Wood, “An integrated design and fabrication strategy for entirely soft, autonomous robots,” *Nature*, vol. 536, no. 7617, pp. 451–455, Aug. 2016.
- [5] M. Baziyad, A. B. Nassif, T. Rabie, and R. Fareh, “Comparative study on the performance of heuristic optimization techniques in robotic path planning,” in *Proc. 3rd Int. Conf. Adv. Artif. Intell.*, Oct. 2019, pp. 157–161.
- [6] A. Shukla and H. Karki, “Application of robotics in onshore oil and gas industry—A review part I,” *Robot. Auto. Syst.*, vol. 75, pp. 490–507, Jan. 2016.
- [7] M. Sitti, “Miniature soft robots—road to the clinic,” *Nature Rev. Mater.*, vol. 3, no. 6, p. 74, 2018.
- [8] R. H. Taylor, A. Menciassi, G. Fichtinger, P. Fiorini, and P. Dario, “Medical robotics and computer-integrated surgery,” in *Springer Handbook of Robotics*. London, U.K.: Springer, 2016, pp. 1657–1684.
- [9] K. Sharma, M. Mishra, and S. Bhardwaj, “Social impact of artificial intelligence: Robotics,” *Res. About Res.*, vol. 1, p. 19, 2016.
- [10] D. Brougham and J. Haar, “Smart technology, artificial intelligence, robotics, and algorithms (STARA): Employees’ perceptions of our future workplace,” *J. Manage. Org.*, vol. 24, no. 2, pp. 239–257, Mar. 2018.
- [11] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, “Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 952–964, Feb. 2017.
- [12] A. Pandey and D. R. Parhi, “Optimum path planning of mobile robot in unknown static and dynamic environments using fuzzy-wind driven optimization algorithm,” *Defence Technol.*, vol. 13, no. 1, pp. 47–58, Feb. 2017.
- [13] B. K. Patle, D. Parhi, A. Jagadeesh, and O. Sahu, “Real time navigation approach for mobile robot,” *J. Comput.*, vol. 12, no. 2, pp. 135–142, 2017.
- [14] S. Jeong, O. Simeone, and J. Kang, “Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.
- [15] M. Garber, “A grid-based path planning approach for a two vehicle team with localization constraints,” Ph.D. dissertation, Texas A&M Univ., College Station, TX, USA, 2017.
- [16] Y. Singh, S. Sharma, R. Sutton, and D. Hatton, “Path planning of an autonomous surface vehicle based on artificial potential fields in a real time marine environment,” Univ. Plymouth, Plymouth, U.K., Tech. Rep. 8717, 2017.
- [17] R. Fareh, T. Rabie, S. Grami, and M. Baziyad, “A vision-based kinematic tracking control system using enhanced-PRM for differential wheeled mobile robot,” *Int. J. Robot. Autom.*, vol. 34, no. 6, pp. 206–221, 2019.
- [18] X. C. Chen, Z. Q. Cao, Y. Q. Yang, and C. Zhou, “Vision-based fuzzy controller for quadrotor tracking a ground target,” *Adv. Sci. Technol.*, vol. 101, pp. 121–126, 2017.
- [19] D.-Q. Zhu and M.-Z. Yan, “Survey on technology of mobile robot path planning,” *Control Decis.*, vol. 25, no. 7, pp. 961–967, 2010.
- [20] L. Jailliet, J. Cortés, and T. Siméon, “Sampling-based path planning on configuration-space costmaps,” *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 635–646, Aug. 2010.
- [21] H. Akbaripour and E. Masehian, “Semi-lazy probabilistic roadmap: A parameter-tuned, resilient and robust path planning method for manipulator robots,” *Int. J. Adv. Manuf. Technol.*, vol. 89, nos. 5–8, pp. 1401–1430, Mar. 2017.
- [22] Z. Li, C. Yang, C.-Y. Su, J. Deng, and W. Zhang, “Vision-based model predictive control for steering of a nonholonomic mobile robot,” *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 553–564, Mar. 2016.
- [23] J. Leitner, S. Harding, M. Frank, A. Förster, and J. Schmidhuber, “An integrated, modular framework for computer vision and cognitive robotics research (icVision),” in *Biologically Inspired Cognitive Architectures*. Berlin, Germany: Springer, 2013, pp. 205–210.
- [24] H. Choset, “Coverage for robotics—a survey of recent results,” *Ann. Math. Artif. Intell.*, vol. 31, nos. 1–4, pp. 113–126, 2001.
- [25] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, “Path planning with modified a star algorithm for a mobile robot,” *Procedia Eng.*, vol. 96, pp. 59–69, 2014.
- [26] H. Sarbolandi, D. Lefloch, and A. Kolb, “Kinect range sensing: Structured-light versus Time-of-Flight Kinect,” *Comput. Vis. Image Understand.*, vol. 139, pp. 1–20, Oct. 2015.
- [27] J.-J. E. Slotine and W. L. Ping, *Applied Nonlinear Control*, vol. 199, no. 1. Englewood Cliffs, NJ, USA: Prentice-Hall, 1991.
- [28] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1990, pp. 384–389.



**RAOUF FAREH** received the master’s and Ph.D. degrees in electrical engineering from the École de Technologie Supérieure, University of Quebec, Canada, in 2008 and 2013, respectively. In 2013, he joined Ottawa University as a Postdoctoral Researcher. In 2014, he joined the University of Sharjah, where he is currently teaching robotics and control systems. His research interests include robotics, nonlinear control, and mathematical modeling.



vision integration, nonlinear robotic control, robot tracking/path planning, simulating human emotion for robotics, advanced software architectures for decision making, and stereo and color analysis for dynamic obstacle avoidance and navigation.

**MOHAMMED BAZIYAD** received the bachelor's degree in network engineering from the Canadian University of Dubai, Dubai, United Arab Emirates, in 2015. He is currently a Research Assistant with the Autonomous Robotics & Active Vision (ARAV) Research Group, University of Sharjah. He is also involved with many research projects in different areas mostly related to robotics. His research interests include information security, steganography, active computer



ing, computer vision, image steganography, and intelligent robotic systems.

**TAMER RABIE** (Senior Member, IEEE) received the M.Sc. degree from the Department of Electrical & Computer Engineering, University of Calgary, in 1993, and the Ph.D. degree from the Department of Electrical & Computer Engineering, University of Toronto, in January 1999. He is currently an Associate Professor of computer engineering with the Electrical & Computer Engineering Department, University of Sharjah. His current research interests include digital image process-



imation, signal and image processing, process control, networked control systems, fractional dynamics and control, nonlinear estimation and filtering, and renewable energies.

**MAAMAR BETTAYEB** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1976, 1978, and 1981, respectively. In 2000, he joined the University of Sharjah. He is also a member of the Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdulaziz University, Jeddah, Saudi Arabia. His current research interests include  $H_\infty$  optimal control, rational approx-

...