Jurnal Mantik, 7 (1) (2023), ISSN 2685-4236 (Online)

Published by: Institute of Computer Science (IOCS)

Jurnal Mantik

Journal homepage: www.iocscience.org/ejournal/index.php/mantik

Real world design and implementation of pathfinding sewer inspection robot using a-star algorithm

Atyanta Nika Rumaksari[1], Adri Gabriel Sooai[2], Gloria Song Abimanyu[3], Gunawan Dewantoro[4], Hartanto Kusuma Wardana[5], Budihardja Murtianta[6], Lukas Bambang Setyawan[7]

[1,3,4,5,6,7] Faculty of Electrical Engineering, Universitas Kristen Satya Wacana, Indonesia

[2]Computer Science Department, Universitas Katolik Widya Mandira, Indonesia

ARTICLE INFO

Article history:

Received Apr 17, 2023
Revised Apr 24, 2023
Accepted Apr 28, 2023

Keywords:

A Star Algorithm
Application of Graph Theory
Autonomous Navigation
Pathfinding Algorithm
Sewer Inspection Robot

ABSTRACT

This paper presents the design and implementation of a sewer inspection robot that utilizes the A-Star algorithm for pathfinding. The robot is intended to provide a more efficient solution for culvert workers in inspecting sewer pipes, particularly in hard-to-reach areas. The A-Star algorithm was chosen due to its ease of implementation and low computational resource requirements, making it suitable for real-time applications. The robot was designed with a modular approach, allowing for flexibility in adapting to different pipe sizes and configurations. It is equipped with various sensors and cameras, allowing for accurate inspection of pipe conditions and identification of potential issues. The A-Star algorithm was used to plan the robot's path through the sewer pipes, minimizing the time required for inspection and reducing the risk of damage to the pipes. The results of the implementation showed that the sewer inspection robot using the A-Star algorithm was able to efficiently navigate through the sewer pipes, reducing the time required for inspection and minimizing the need for manual labor. In order to check the performance, we performed experiments on six test models through simulation. On average, the proposed algorithm showed remarkable results, where all models can generate path planning to find the target from the start position. We obtained an average time completion from Models 1 to 6 with a maximum travel distance of 30 meters of 12.96, 4.47, 18.59, 20.71, 24.93, and 19.34 seconds.

Corresponding Author:

Atyanta Nika Rumaksari,
Electrical Engineering Faculty,
Universitas Kristen Satya Wacana,
Address Diponegoro 52-60, Salatiga, Central Java, 50711, Indonesia.
Email: atyanta.rumaksari@uksw.edu

1. INTRODUCTION

It is undeniable that densely populated cities would affect the development of supporting facilities. It should be compensated by the provision of adequate public infrastructure, one of which is the construction of culverts. It can be built underground, under bridges, or even under train tracks, which is often called drainage. Currently, the culverts that are being built have various shapes and sizes, long and narrow paths and are mostly underground. This can be a problem for workers to check regularly, considering the risk that workers must inspect culverts themselves (Alyasin et al., 2019; Leng & Chanson, 2020; Montes et al., 2019; Ross et al., 2021; Sharif & Gormley, 2021). For this reason, a sewer robot is proposed to monitor culverts and work under water with a variety of sensors mounted on it, such as optical, acoustic, inertial, thermal, and pressure sensors (Aitken et al., 2021). Therefore, this robot is expected to be used to directly assist the human task or role in terms of monitoring culverts while minimizing the risk of work accidents (Yeo et al., 2021).

However, most research has been carried out in the application of machine vision technology with robot control technology (Ni et al., 2021). They offer swift and effective services in numerous domains that hold growing significance in the contemporary manufacturing and inspection sectors, benefiting individuals. For sewer robots, it is possible to enter the sewers and find their way controlled by the operator. However, if the robot is trapped, it will be in trouble because the operator does not know the route of the culvert, and the robot also does not record the traces that have been passed. As a result, the robot can be lost in the middle of the culvert that has many paths (Ciszewski et al., 2018). Therefore, a pathfinding method is needed for the robot so that it can find its own way out after finishing the task or reaching a dead end. One of the pathfinding methods that can be used on this robot is the A* algorithm (read as "A STAR").

This algorithm was first described in 1968 by Peter Hart, Nils Nilsson, and Bertram Raphael. In their writing, this algorithm is simply called algorithm A. Then, with heuristic optimization, it is called A star (A*) (Edelkamp, Stefan;Schrödl, 2012; Hart et al., 1968). The A* algorithm is one of the most optimal and complete route search algorithms with complex data input compared with other well-known algorithms such as Greedy and Dijkstra (Wayahdi et al., 2021). What is meant by optimal is that the resulting route is robust and has low-cost source-destination paths (Hu et al., 1993).

The heuristic search algorithm has superior performance to blind search (Ismail & Agwu, 2019). When complex data are implemented, the blind search requires a longer access time and more memory than the heuristic algorithm. Therefore, it can be said that the A* is the best search algorithm for finding the shortest path with the simplest calculation on the path from the initial node to the final node. In line with the previous opinion, the A* is known as one of the most frequently used algorithms for pathfinding and graph traversal, which can be presented as nodes (Russell, Stuart; Norvig, 2021).

In the mathematical process, A* employs a pair list, namely, the OPEN and CLOSED lists, to store the required nodes that have been previously generated to calculate the raw heuristic value. In other words, OPEN contains nodes that still have a chance to be selected as the best node, while CLOSED is a list to store the nodes that have been generated and selected as the best node (the chances of being selected are closed). However, in the practical solution, there are numerous algorithms that have more efficient results in finding the shortest path than A*(Kim & Kim, 2023). It can be easily outperformed by an algorithm that pre-processes the graph. Moreover, the difficulty of requiring completion costs makes A* unsuitable for many shortest path applications. However, in reality, it is ideal for applications with complete source information, such as GIS (Geographic Information System) (Zeng & Church, 2009).

The most common applications of A* include, among others, unmanned aerial vehicles (UAVs), such as making the direction path of radar in 3D aerial space environments (Alhadi et al., 2021; J. Li et al., 2023; Z. Zhang et al., 2022), path planning for guided vehicles in choosing parking spaces (Sedighi et al., 2019), and automatic routing networks for petrochemical inspection robots (Lai et al., 2021).

Improvement of A* is needed since the traditional method has the disadvantage of taking too long to reach the desired target (Cai et al., 2019). The improvement in A* is achieved by using an extra scope of inspections aimed at reducing redundant paths to achieve a highly efficient path cost (X. Li et al., 2020). On the processing side, the genetic algorithm can be used to accelerate the convergence speed during the search (Y. Li et al., 2020). Robot agility can be increased by removing the sharp turn (Gunawan et al., 2019), giving consideration to obstacle avoidance (Shang et al., 2020), filtering the nodes in the close list by means of $P(x,y)$and $W(x,y)$ and removing the redundant nodes to enhance regular paths (Tang et al., 2021).

However, environmental conditions can be a huge problem because the path's dynamic position and noise can make the robot ineffectively process the path (Y. Zhang et al., 2019). Therefore, the organization of path planning can be achieved by organizing other information, such as berthing paths and collision avoidance planning (C. Liu et al., 2019). This arrangement can be seen in automated guided vehicles (AGVs) for logistics and manufacturing, where improving work efficiency is a very important factor and has become a major focus for industry (Zheng et al., 2019). Numerous advanced algorithms have been developed for the field of SLAM research, all of which are based on the A* algorithm as their fundamental building block (Jiang et al., 2023). However, the complexities are becoming a concern in the fields of computation and implementation. It is undeniable that the improvement leads to greater complexity in computational phase.

As a lecture for elementary-level students, this algorithm should be presented and student should be required to understand the concept, especially in structured data architecture, or on the foundations of artificial intelligence subjects. We tailored this algorithm to make the student understand, initiating short and clear command parts through the sensor-scanning process. Then, we proceed with the calculation of each search block and the costs it incurs to finally obtain the most efficient path. When applied to a mapping robot, students can place it well because it does not require complex understanding.

Culvert inspection is a job that requires three important parts, namely, the process of mapping the transportation route, the inspection process, and finding the shortest path from the beginning to the destination or vice versa [28]. Based on these research objectives, the A* algorithm is very suitable because it meets the two criteria for the localization process. In this research, we emphasize not only the effectiveness of the problem-solving of the implementation of mapping and the path finding algorithm but also the simplicity of algorithm design.

2. RESEARCH METHOD

Analytical methods and design of the robot's structure, such as hardware configuration, preparation of path testing, designing algorithm and software implementation.

2.1. Hardware Design

In this research, we implemented the A* algorithm into a robot that is designed to work in culverts. The robot has an undercarriage drive. The robot is also equipped with a GPS (global positioning system) sensor for locating the coordinates, a digital compass for direction determination, a camera for machine vision, and ultrasonic-based distance sensors. Because in this study we focus on the functionality of the tool, the robot is deliberately placed in ditches that have dry soil.

The hardware design uses ultrasonic sensors as the detector of obstacles that exist around the robot and a camera sensor as a visual obstacle detector that can be directly seen by the operator. Since it is general positioning as described by (Darmawan et al., 2023), the robot employs GPS and an electronic compass to sense the current position relative to the movement and purpose. The main controller functions to control the moving robot and record positions that have been acquired by the robot sensors. The robot is powered by a 7AH 12 VDC battery, which can last 4-5 hours. We design a torque controller without using a special controller such as Linear Quadratic Gaussian (LOG)

(MOUNIS et al., 2022) because we retain the design simplicity. Fig. 1 shows the design of the prototype, and Fig. 2 shows its block system design.

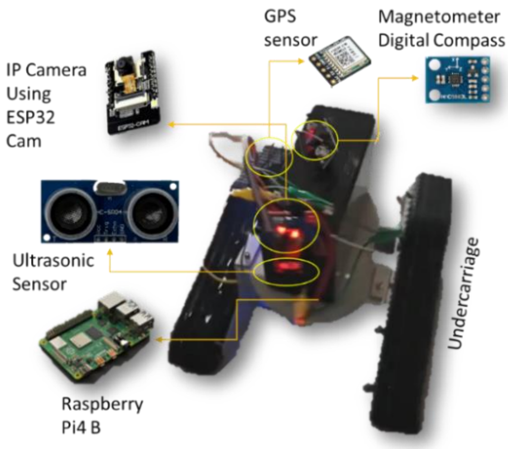

Figure. 1. The prototype of the Sewer Robot Design, which has an IP Camera ESP32, an ultrasonic sensor, a GPS sensor, a magnetometer sensor, and a Raspberry Pi 4B for its controller. The undercarriage enables the robot to move off-road.

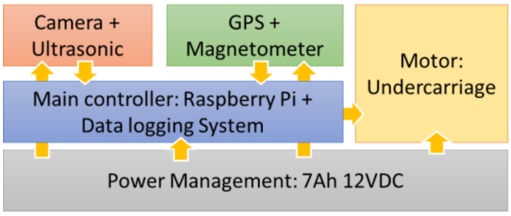

Figure. 2. Block System Design of the Sewer Robot Prototype. The voltage regulator will step-down the voltage into 5 VDC and 3.3 VDC for camera, ultrasonic, GPS and magnetometer sensors. The motor driver uses both 5 VDC and 12 VDC sources

The robot is designed using several processes. Initially, we manually entered the target's location as a reference. The robot will actively search for the target while placing marks according to the map provided. In practice, as seen visually in the software, green marks imply that the robot is led to pass a predetermined path. We put this location in an open list format. Meanwhile, along the paths that have been traveled, the successor is determined. After the process is done, we put it into the Close list (colored in red). The start-stop paths are colored purple. Finally, the finding path is used for navigating the robot from finish to start.

2.2. Optimizing Software Testing through Path Preparation

We manually map the culverts at the author's workplace, namely, the main culverts at Satya Wacana Christian University (SWCU) in Indonesia, as shown in Fig. 3. Then, a special route is chosen to test the effectiveness of the algorithm. The culverts at SWCU have a slightly steep and muddy path. When it rains, the water can fill the culverts but does not have a strong current flow. As shown in Fig. 4, the culvert has a width of 1 meter and a height of 75 cm. The path can be shown as a bold red line. Based on this map, we prepare several scenarios to test the effectiveness of the algorithm.

Figure 3. A combination of the actual map (right) with manual sewer-route inspection (left) in UKSW. Using this data, we then create a test simulation.

a. Sewer entrance/exit

b. Water tunnel/sewer from outside

c. Condition of Water tunnel/sewer from inside.

Figure 4. The conditions of the SWCU sewer before testing since the robot was not covered with waterproof

2.3. Algorithm Design

The A* algorithm consists of two conditions: OPEN-CLOSE list and a n-step heuristic function $h(n)$ based on the equation $f(n) = g(n) + h(n)$. Function $g(n)$ computes the cost between the start and destination through a particular sequence of paths. Each possible adjacent node of the current position is evaluated by f(n). Since a sewer has the possibility of dynamic paths due to blockage, heuristic information can be utilized to guide the robot to find the path. Consequently, to optimize the algorithm, in this research, we assume that the dynamic arrangement must not exist within the OPEN-CLOSE list.

We assume that the path of our condition is a group of n-block arrangements. Each block occupies (x, y) coordinates, with the robot being our point of reference. Therefore, we can say that the robot is needed to search and move from the starting to the destination point. While the robot is running, it will calculate the cost of the evaluation function $f(x, y)$. The minimum function $\min(f(x, y))$ will be stored in the buffer as a path finding reference. In other words, the robot can only get the local information around (one-block size). The collection of this information is done one-by-one until the target is found. Based on this consideration, we assume that the target must exist somewhere in the robot's path.

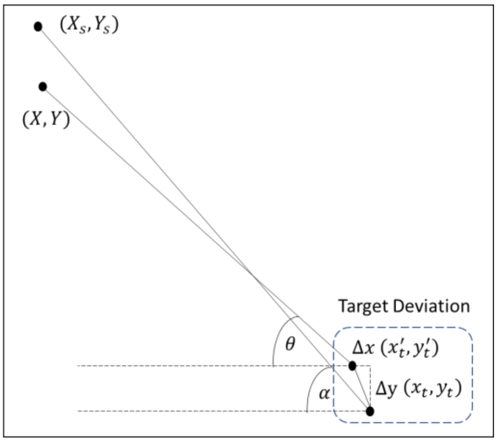

Figure 5. Target deviation arrangement of Δx and Δy of the starting point versus the target point.

To minimize the computational load, we must limit the furthest traveling distance to 30 meters. As stated in (X. Liu & Gong, 2011), the greater the distance, the higher the uncertainty in the heuristic information (Rumaksari, 2019). To constrain the robot, we incorporate the target point's deviations into the A* heuristic function. Fig. 5 illustrates the computation of the deviation area.

$$x' = x_t + \Delta\mathrm{x},$$
$$y' = y_t + \Delta\mathrm{y} \tag{1}$$

Where (x_t, y_t) is the actual target's location, (x_t', y_t') is the detected target's location and $(\Delta x, \Delta y)$ is the error caused by sensor readings and compensation of robot's movement. We can approximate the errors as in Equation 2. k is an error constant, and L is the distance between the starting point and target point. In our research, we calculate L as the Euclidean distance between start (x, y) and target (x_t, y_t) point. Function $rand(-1,1)$ is random between -1 to 1.

$$\Delta x = k\frac{1}{L}rand(-1,1),$$
$$\Delta y = k\frac{1}{L}rand(-1,1) \tag{2}$$

2.4. Software Implementation

Basically, the graph search algorithm for path planning and searching method employs the breadth-first search algorithm, as shown in Fig. 6. This means that we conduct the node-visiting process as a tree transversal. Therefore, the next node can be visited after we have visited all current nodes of a branch. Consequently, it may sound inefficient because it conducts a blind search. However, in this research, we maintain a feasible path by scaling and transforming a real location map into grid. The processes are (1) scaling the original area into a reference area that can be traveled by the robot, (2) dividing the reference area into robot movement boxes, (3) manually marking which reference boxes are obstacles and sewer wall structures (the black color in Fig. 8 shows the manual marks).

In this research, we created 50×50 blocks (Error! Reference source not found.). Each block has a size of approximately 50 centimetres. The tolerance for movement in the box is a radius of 30 centimetres, which is the size of the robot area. The movement analysis from position $P_{start}(x_1, y_1)$ to $P_{target}(x_t, y_t)$ is shown in Error! Reference source not found Error! Visually, the movement starts from the centre position of the box, namely, position (x_1, y_1). The heuristic distance is the shortest distance between the start position and the target. Along its movement to get to the target position, the robot must calculate seven cost values, they are $C2, C3, C4, C5, C6, C7$, and $C8$. Where $C_i = d(P(x_i, y_i), P(x_t, y_t))$. In this study, d is the Euclidean distance $d(a, b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$.

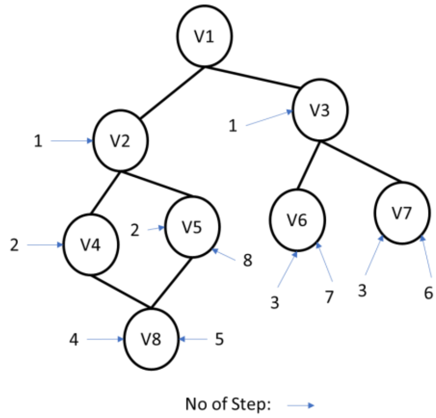

Figure 6. Step-by-step Breadth-First Search Algorithm. The numbers on the left and right of the nodes are the number of step-by-step process algorithms.

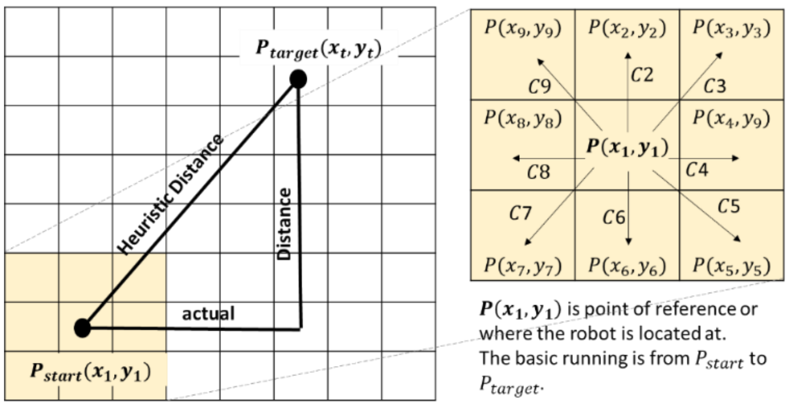

Figure 7. . Grids and movement analysis (left) is a general map. (Right) is the movement pattern and cost calculation of the robot.

Since the movement is modeled on the basis of an $n \times n$ grids, we know that there is compensation for motion which becomes an inevitable error where the accuracy of the robot is defined as the distance between the position calculated by the software and the position of the robot in real calculations. However, in designing the robot, we do not focus on including this case in our algorithm design, because in practice our robot is good enough to work according to applicable standards.

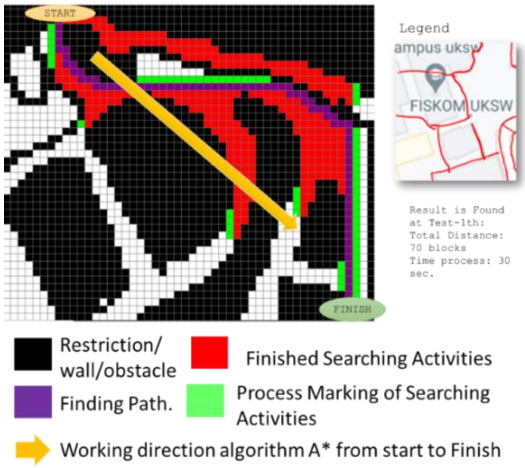

Figure 8. Display of output A* algorithm program after running

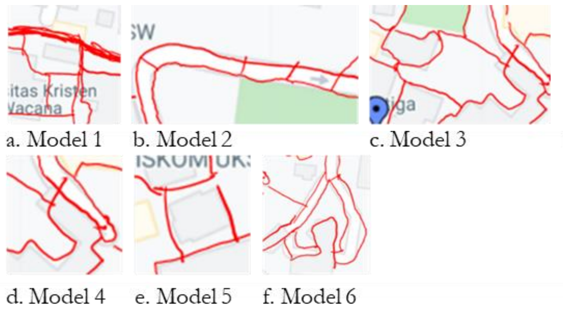

Figure 9. Six scenarios of tests. Models 1 and 2 have the same sharp turn, and Models 3, 4, 5, and 6 have hidden paths.

In this research, we use *Python Pygame* to design the visualization of the A* Algorithm. As shown in Fig. 8, there is a configuration of the A* Algorithm implementation. We divide the box into four colors, namely, green for the search plan box performed by the robot, red for the area that has been visited by the robot (the cost function value has been obtained), black for obstacles, and purple is the found path.

After all parameters and variables are prepared, we program the algorithm by assigning the target node as *NodeGoal* and the start node as *NodeStart*. We maintain two lists: OPEN and CLOSE. The OPEN list consists of every node that has been visited, but not expanded (meaning that successors have not yet been explored). This is the list of pending tasks. The CLOSE list consists of nodes that have been visited and expanded (successors have been explored already and included in the open list, if this was the case). The pseudocode of the A* can be seen in Algorithm 1. Our tests are set by six scenarios with different curves, angles and positions, as shown in Fig. 9. This test is implemented based on the work schedule of the plumber, since the robot is designed to help the plumber monitor the sewer conditions.

Algorithm 1. Pseudocode Algorithm

```
Put NodeStart in the OPEN list with f(NodeStart) = h(NodeStart) as initialization
while the OPEN list is not empty:
{ Take from OPEN list a NodeCurrent with the lowest value:
            f(NodeCurrent) = g(NodeCurrent)
                  +h(NodeCurrent)
if NodeCurrent is NodeGoal, then break.
Generate NodeSuccessor that code after NodeCurrent.
for each NodeSuccessor of NodeCurrent:
{ Set SuccessorCurrentCost =
        g(NodeCurrent) + w(NodeCurrent, NodeSuccessor)
IfNodeSuccessor is the OPEN list: or in CLOSE list:
{if g(NodeSuccessor) ≤ SuccessorCurrentCost then GO TO line 23}
else if NodeSuccessor is in the CLOSE list:
Move NodeSuccessor from the CLOSE list to OPEN list. }
else: { Add NodeSuccessor to the OPEN list. ; Set h(NodeSuccesor) to be HD(NodeGoal). }
Set g(NodeSuccessor) = NodeSuccessorCurrentCost.
Set the parent of NodeSuccessor to NodeCurrent.
}
Add NodeCurrent to the CLOSE list.
}
if(NodeCurrent ! = NodeGoal) exit
with error (the OPEN list is empty).
```

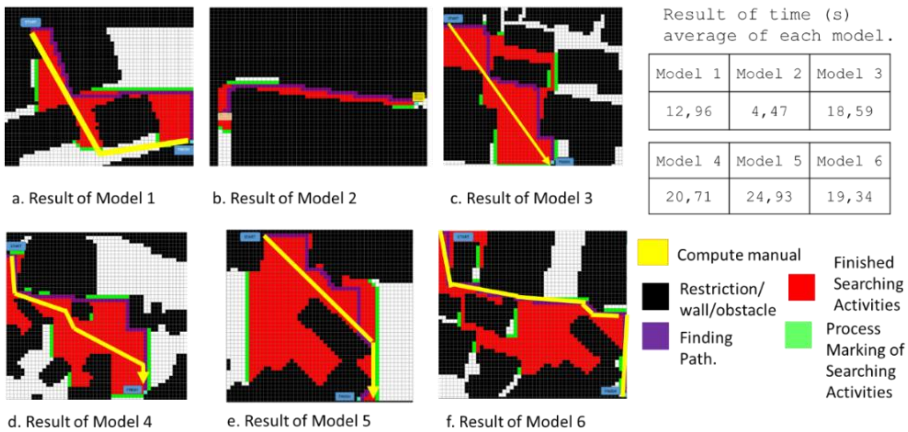

Figure 10. Each model is computed fairly under the same conditions. Model 5 is the longest time completion because of widest searching area (red dot is larger than others).

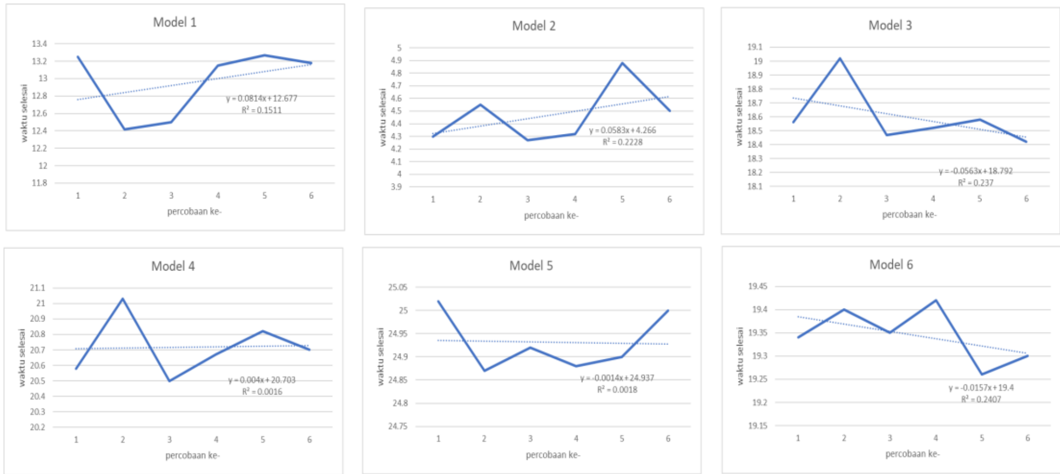

Figure 11. Average time of experiment of each model

3. RESULTS AND DISCUSSIONS

In this section, we focus on showing the two major parts of the results of the analysis and considerations during the research in order from the successful development of mechanic in robot that can pass through obstacles and the volume of sewers to the success of the algorithm in solving the path finding problem. Algorithm testing is very important because the current mapping of culverts has been done manually. The purpose of this research is to make it easier for operators to carry out their duties in inspection activities. When carrying out its activities the robot experiences direction problems, through this path finding algorithm, it can show the direction back and the location where the robot's last position was.

In designing the robot, we don't pay attention to aerodynamics because we focus on the successful use of the A* algorithm in relation to making paths for the robot to work. As a result, we designed the robot to be semi-automatic, while still retaining the power cable directly, which is useful besides being always-on-power and the cable is used to mark the position of the robot's movement when it is in the sewer line. However, we considered the total size of robot’s body as small as possible which should not exceed the volume size of the culvert.

Figure 1 shows the mechanical specifications of the robot. By being given a 7 Ampere Hour battery power, the robot can safely last for 8 hours continuously, namely during the work period of the operator employee. Equipped with a camera robot can visually show the conditions in the field. So that in practice if there is a path adjustment due to external factors and the failure of the algorithm in solving the problem, the operator can take over control of the robot's movement. Apart from that, in carrying out the experiment, we were assisted by operators who understand the position and direction of the culvert path pattern (Fig. 12). In Fig.12 it is shown that before starting the test, we carry out the process of calibrating the magnetometer and digital encoder sensors to ensure that we calculate the steps of the robot's movement. In this design, we found problems in reading the change orientation and the position of the robot, so that in every experiment process, we have to calibrate all sensors first. This is because we do not save the motion reference and the position of the robot based on the location where the robot is used. Therefore, we hope that further research can solve this problem.

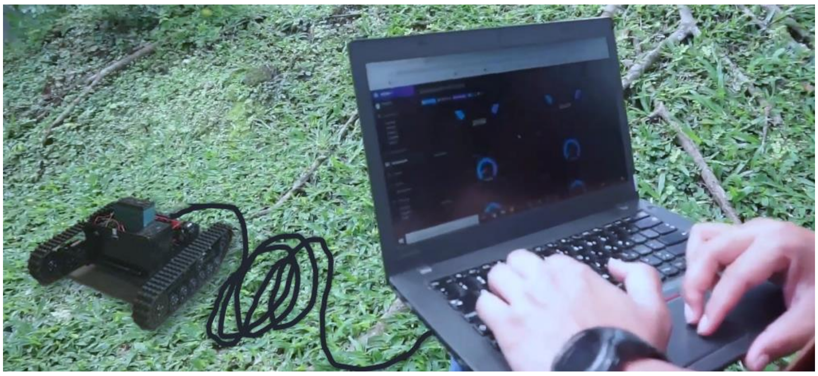
Figure 12. The operator is controlling the robot.

In the experiments, we observe every simulated robot in each testing scenario. Fig. 1 shows the result of the scenario. From the chart, we can see in each scenario that the robot has outstanding performance, with a 100% success rate to find the target, while the completion time varies with maximum 0.35 standard deviation. The average completion times (in seconds) for model scenarios 1 to 6 are 12.96, 4.47, 18.59, 20.71, 24.93, and 19.34, respectively.

Model 1 and Model 2 have a similar concept of sharp turnover of almost 90^0. However, in scenario Model 1, the robot needs to turn sharp left two and a third blocks, while in Model 2, the robot only needs to turn right directly after turning sharply left two and a third blocks. The other difference that leads to a higher computational load in Model 1 is having three path options before arriving at the target; therefore, the algorithm needs to mark the search activities more than in Model 2. From the experiment, we obtained one unnecessary process before it returned to the correct path. We also observe that the resulting path is not the most efficient one (shown in yellow) because the algorithm has a tendency to search the right-upper hand first, since this area is obstacle-free. Fig. 10 and 11 show the comparison between each model.

We prove that the A* algorithm will prioritize a direction where the target exists. Not only Models 1 and 2, but also Model 3 will show the same result, that is it goes left straight before moving diagonally down to where the target is located. A* does not allow it to move diagonally freely because the searching activities do not have preloaded memory and it has one-to-one movement based on a single cost function without considering an obstacle (blind search). The experiment shows that Model 3 has the widest result discrepancy between manual computation and A*.

Models 4 and 5 have the same concept as Model 3, which has a higher degree of discrepancy between manual computation and A*'s result. The best performance is obtained from Model 6 and Model 2, which have the same result in both manual and A*. Unlike Model 2, in which the obstacle forms the path, Model 6 has many compartments, but the target position is exclusive inside the obstacle with unwinding paths. Therefore, the result will show the best finding.

4. CONCLUSION

In this paper, we propose the fundamental A* Algorithm to give an example of real-world problem A* Algorithm applications. With this simplified algorithm, we can teach beginners the basic structure of how the algorithm addresses path finding problems. In addition, to check the performance, we performed experiments on six test models through simulation. On average, the proposed algorithm showed remarkable results, where all models can generate path planning to find the target from the start position. We obtained an average time completion from Models 1 to 6 of 12.96, 4.47, 18.59, 20.71, 24.93, and 19.34 seconds. In conclusion, our implementations already succeed in achieving the goal of applying the A* algorithm to a homemade sewer robot to help the plumber inspect the sewer's condition. Industrial sewer robots are very expensive for SWCU; therefore, we made them ourselves.

However, one of the limitations of this research is the design of the robot's structure. Since it works in the sewer, it should be completed with supporting waterproofing and mud cutting technology. Therefore, in future research, we will make more use of CNC technology to design agile, waterproof and strong robots to make robots that are durable and can be applied to muddy culverts.

ACKNOWLEDGEMENTS

We wish to acknowledge the generous support of the Management Campus Division (BMK). I would like to express my appreciation to Kedai Reka Matching Fund DIKTI 2022 for funding the project of an AI-Mining System for Data Acquisition. Many thanks to Dr. Hanna Arini Parhusip, Dr. Suryasatria Trihandaru, Prof. Yohanes Sardjono, Gede Sutresna Wijaya M.Sc., Ir. Isman Mulyadi Triatmoko, Hindriyanto Dwi Purnomo, Ph.D, Putri Hergianasari, S.IP, M.IP., Daniel Eliazar Latumaerissa M.SC, Mr. Adrianus Herry Hariadi, and Mr. Petrus Priyo Santosa.

REFERENCES

Aitken, J. M., Evans, M. H., Worley, R., Edwards, S., Zhang, R., Dodd, T., Mihaylova, L., & Anderson, S. R. (2021). Simultaneous Localization and Mapping for Inspection Robots in Water and Sewer Pipe Networks: A Review. In *IEEE Access* (Vol. 9, pp. 140173–140198). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2021.3115981

Alhadi, S., Rianmora, S., & Phlernjai, M. (2021). Conceptual Design and Analysis of Small Power Station for Supporting Unmanned Aerial Vehicle (UAV) Deployment. *Engineering Journal, 25*(8), 51–71. https://doi.org/10.4186/ej.2021.25.8.51

Alyasin, A., Abbas, E. I., & Hasan, S. D. (2019). An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method. *4th Scientific International Conference Najaf, SICN 2019*, 11–14. https://doi.org/10.1109/SICN47020.2019.9019345

Cai, Y., Xi, Q., Xing, X., Gui, H., & Liu, Q. (2019). Path planning for UAV tracking target based on improved A-star algorithm. *1st International Conference on Industrial Artificial Intelligence, IAI 2019*, 1–6. https://doi.org/10.1109/ICIAI.2019.8850744

Ciszewski, M., Buratowski, T., & Giergiel, M. (2018). Modeling, Simulation and Control of a Pipe Inspection Mobile Robot with an Active Adaptation System. *IFAC-PapersOnLine*, *51*(22), 132–137. https://doi.org/10.1016/j.ifacol.2018.11.530

Darmawan, A., Ratnadewi, R., & Prijono, A. (2023). Designing a wireless robot plotter as a supporting tool for understanding logical thinking. *Jurnal Mantik*, *7*(1), 155–167. https://doi.org/10.35335/MANTIK.V7I1.3640

Edelkamp, Stefan;Schrödl, S. (2012). *Heuristic Search: Theory and Applications* (D. Roumeliotis, Rachel; Bevans (Ed.)). Morgan Kaufmann : Elsevier Inc. https://doi.org/10.1016/C2009-0-16511-X

Gunawan, S. A., Pratama, G. N. P., Cahyadi, A. I., Winduratna, B., Yuwono, Y. C. H., & Wahyunggoro, O. (2019). Smoothed a-star algorithm for nonholonomic mobile robot path planning. *2019 International Conference on Information and Communications Technology, ICOIACT 2019*, 654–658. https://doi.org/10.1109/ICOIACT46704.2019.8938467

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In *IEEE Transactions on Systems Science and Cybernetics* (Vol. 4, Issue 2, pp. 100–107). https://doi.org/10.1109/TSSC.1968.300136

Hu, T. C., Kahng, A. B., & Robins, G. (1993). Optimal Robust Path Planning in General Environments. *IEEE Transactions on Robotics and Automation*, *9*(6), 775–784. https://doi.org/10.1109/70.265921

Ismail, I. M., & Agwu, N. N. (2019, February 4). Influence of heuristic functions on real-time heuristic search methods. *14th International Conference on Electronics Computer and Computation, ICECCO 2018*. https://doi.org/10.1109/ICECCO.2018.8634782

Jiang, C., Meng, Q., Xu, B., Gao, W., Wu, P., Guan, L., Li, Z., Zheng, S., Wang, J., Rizos, C., Ding, W., & El-Mowafy, A. (2023). Simultaneous Localization and Mapping (SLAM) for Autonomous Driving: Concept and Analysis. *Remote Sensing 2023, Vol. 15, Page 1156*, *15*(4), 1156. https://doi.org/10.3390/RS15041156

Kim, C., & Kim, C.-J. (2023). Development of Autonomous Driving and Motion Control System for a Patient Transfer Robot. *Actuators*, *12*(3), 106. https://doi.org/10.3390/ACT12030106

Lai, X., Li, J. H., & Chambers, J. (2021). Enhanced Center Constraint Weighted A* Algorithm for Path Planning of Petrochemical Inspection Robot. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *102*(4), 1–15. https://doi.org/10.1007/s10846-021-01437-8

Leng, X., & Chanson, H. (2020). Hybrid modelling of low velocity zones in box culverts to assist fish passage: Why simple is better! In *River Research and Applications* (Vol. 36, Issue 9, pp. 1765–1777). John Wiley & Sons, Ltd. https://doi.org/10.1002/rra.3710

Li, J., Zhang, W., Hu, Y., Fu, S., Liao, C., & Yu, W. (2023). RJA-Star Algorithm for UAV Path Planning Based on Improved R5DOS Model. *Applied Sciences (Switzerland)*, *13*(2). https://doi.org/10.3390/app13021105

Li, X., Hu, X., Wang, Z., & Du, Z. (2020). Path planning based on combination of improved A-STAR Algorithm and DWA algorithm. *Proceedings - 2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture, AIAM 2020*, 99–103. https://doi.org/10.1109/AIAM50918.2020.00025

Li, Y., Dong, D., & Guo, X. (2020). *Mobile robot path planning based on improved genetic algorithm with A-star heuristic method*. 1306–1311. https://doi.org/10.1109/ITAIC49862.2020.9338968

Liu, C., Mao, Q., Chu, X., & Xie, S. (2019). An Improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning. *Applied Sciences (Switzerland)*, *9*(6), 1057. https://doi.org/10.3390/app9061057

Liu, X., & Gong, D. (2011). A comparative study of A-star algorithms for search and rescue in perfect maze. *2011 International Conference on Electric Information and Control Engineering, ICEICE 2011 - Proceedings*, 24–27. https://doi.org/10.1109/ICEICE.2011.5777723

Montes, C., Kapelan, Z., & Saldarriaga, J. (2019). Impact of self-cleansing criteria choice on the optimal design of sewer networks in South America. *Water (Switzerland)*, *11*(6), 1148. https://doi.org/10.3390/w11061148

MOUNIS, S. Y. A., AZLAN, N. Z., & ZAKIA, H. (2022). Design and Control of a Robotic Device for Upper Limb Rehabilitation Therapy. *Journal of Engineering Science and Technology*, *17*(2), 1306–1327.

Ni, Y., Yang, X., Cheng, Y., Dong, Z., & Tang, M. (2021). Machine Vision based Grabbing Objects with Manipulator System Design. *Engineering Journal*, *25*(2), 327–334. https://doi.org/10.4186/ej.2021.25.2.327

Ross, R., Stumpf, A., Barnett, D., & Hall, R. (2021). Condition assessment for concrete sewer pipes using displacement probes: A robotic design case study. *Robotics*, *10*(2), 64. https://doi.org/10.3390/robotics10020064

Rumaksari, A. N. (2019). Vehicle Classification using IPCP and EsKNN Algorithm for Surveillance Camera. *Techné : Jurnal Ilmiah Elektroteknika*, *18*(01), 15–30. https://doi.org/10.31358/techne.v18i01.181

Russell, Stuart; Norvig, P. (2021). *Artificial Intelligence: A Modern Approach, 4th US ed.* Pearson. http://aima.cs.berkeley.edu/

Sedighi, S., Nguyen, D. Van, & Kuhnert, K. D. (2019). Guided Hybrid A-star Path Planning Algorithm for Valet Parking Applications. *2019 5th International Conference on Control, Automation and Robotics, ICCAR 2019*, 570–575. https://doi.org/10.1109/ICCAR.2019.8813752

Shang, E., Dai, B., Nie, Y., Zhu, Q., Xiao, L., & Zhao, D. (2020). A Guide-line and Key-point based A-star Path Planning Algorithm for Autonomous Land Vehicles. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*, 1–7. https://doi.org/10.1109/ITSC45102.2020.9294336

Sharif, K., & Gormley, M. (2021). Integrating the design of tall building, wastewater drainage systems into the public sewer network: A review of the current state of the art. In *Water (Switzerland)* (Vol. 13, Issue 22, p. 3242). Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/w13223242

Tang, G., Tang, C., Claramunt, C., Hu, X., & Zhou, P. (2021). Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment. *IEEE Access*, *9*,

59196–59210. https://doi.org/10.1109/ACCESS.2021.3070054
Wayahdi, M. R., Ginting, S. H. N., & Syahputra, D. (2021). Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path. *International Journal of Advances in Data and Information Systems*, *2*(1), 45–52. https://doi.org/10.25008/ijadis.v2i1.1206
Yeo, M. S. K., Ng, A. Q. B., Ng, T. Y. J., Mudiyanselage, S., Samarakoon, B. P., Muthugala, M. A. V. J., Mohan, R. E., & Ng, D. T. (2021). Robot-Inclusive Guidelines for Drain Inspection. *2021 8th International Conference on Information Technology, Computer and Electrical Engineering, ICITACEE 2021*, 7–12. https://doi.org/10.1109/ICITACEE53184.2021.9617537
Zeng, W., & Church, R. L. (2009). Finding shortest paths on real road networks: The case for A. *International Journal of Geographical Information Science*, *23*(4), 531–543. https://doi.org/10.1080/13658810801949850
Zhang, Y., Li, L. L., Lin, H. C., Ma, Z., & Zhao, J. (2019). Development of path planning approach using improved a-star algorithm in AGV system. *Journal of Internet Technology*, *20*(3), 915–924. https://doi.org/10.3966/160792642019052003023
Zhang, Z., Wu, J., Dai, J., & He, C. (2022). Optimal path planning with modified A-Star algorithm for stealth unmanned aerial vehicles in 3D network radar environment. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *236*(1), 72–81. https://doi.org/10.1177/09544100211007381
Zheng, T., Xu, Y., & Zheng, D. (2019). AGV Path Planning based on Improved A-star Algorithm. *Proceedings of 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2019*, 1534–1538. https://doi.org/10.1109/IMCEC46724.2019.8983841