Adv Syst Sci Appl 2022; **04**:193–201
Published online at https://ijassa.ipu.ru.

Extending Obstacle Map of Autonomous Vehicles Based on Network Model of Local Positioning

Alexander Abdulov*

V. A. Trapeznikov Institute of Control Sciences, Moscow, Russia

Abstract: At present, an unmanned autonomous vehicle (AV) to provide accurate navigation during motion depends on GPS. The search and study for the alternative methods of AV localization is the demand to implement smart city concepts because in real-world conditions, the GPS signal may either be absent, or its accuracy may be insufficient to trajectory following and perform maneuvers. For collision avoidance to raise AV safety, the network model of local positioning is proposed. Control systems based on obtained local maps partially solve the problem of a safety motion. The presented simulation results confirm the effectiveness of the described approach to improving traffic safety in uncontrolled dynamic environments.

Keywords: autonomous vehicle, local positioning, collision avoidance, simulation modeling

1. INTRODUCTION

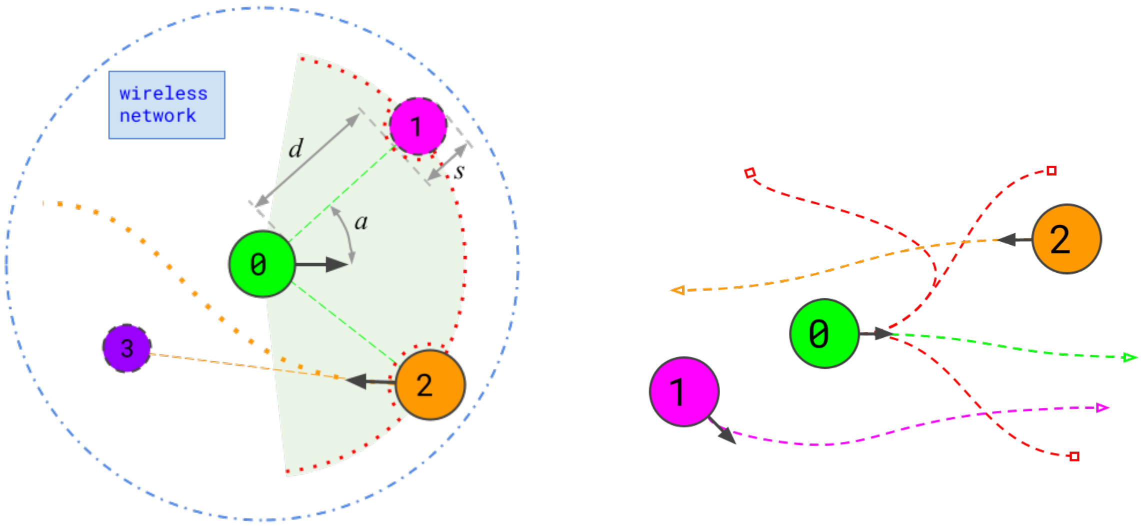
An unmanned autonomous vehicle (AV) for safety motion applies optimal control rules enabling one to avoid collisions. Due to that, AVs form models of observed obstacles, accounting for dynamic characteristics of moving objects (including other vehicles). The accuracy of determining obstacle model parameters is directly concerned with the probability of avoiding a collision [1]. One should point out that in real-time, it is not always possible to estimate the position and dynamics of other road traffic participants accurately directly using only onboard computational tools. As a rule, all road traffic participants are interested in minimizing the collision number, so it is reasonable for them to exchange helpful information with each other under the availability of communication channels. Then the communication can become a key factor in searching for safe motion paths.

Let us consider a case of motion in a "crowd" with a possibility of AVs communicating with each other by the wireless network. The AVs are equipped with rangefinders enabling them to evaluate distances to obstacles and wireless communication modules to receive and transmit messages. Suppose the radius of action of the wireless communication module is comparable with the AV dimension and its speed characteristics to receive helpful information from other participants from the network in advance.

2. PROBLEM STATEMENT

All AVs have the possibility to build their local map of observed objects, transmit it to other network participants, and receive their maps of obstacles. For the two-dimensional case, a local map of obstacles (see Fig. 2.1a) which is represented by the data type $MapMsg$: $< ID, G = \{g_i\}, U = (u_j) >$, where $g_i = (d, a, s)$ – sets of relative positions of directly

*Corresponding author: aabdulov@asmon.ru



(a) local map of obstacles (b) principle of a safe path choosing

Fig. 2.1. Simple model of the mobile autonomous vehicle.

observed objects (distant at the distance d and deviated on the angle a from observer heading), and s – approximate dimensions of the obstacle (since they are unknown, they should be estimated). Note that it is quite difficult to focus on the dimensions estimated by rangefinders in practice. Therefore, this information is rather needed for the subsequent evasion from unidentified objects, which in theory appear to be stationary objects with some size.

In turn, U is the state vector of the AV (dimension, planned path in local coordinate system, and other useful information). It is advisable to use wireless network IP addresses or MAC addresses of network devices as unique identifiers ID. All examples are used simple integer identifiers.

There are N observable objects for each AV in the general case, and M received messages from the network. So, a problem arises to determine message sender location at the local map of obstacles.

Choosing safety path control (see Fig. 2.1b) can be implemented by modifications of DWA [2], TEB [3], or MPC [4].

3. ALGORITHM DESCRIPTION

Partially this uncertainty can be decreased by solving for each AV the local positioning problem, which is to determine a relative location and parameters of motion of different objects. A feature of the problem is the absence of common orient beacons for all road traffic participants. Groups of AVs with communication channels can apply network models of local positioning based on known approaches: triangulation, trilateration, and navigation numeration [5].

The proposed method is applicable, taking into account the following statement: the distance between the participants of the movement observing each other is the same. The complete cycle of the algorithm for obtaining an extended local obstacle map is shown in Fig. 3.2. Let's consider it in more detail.

3.1. Rangefinder data capture

First of all, it is necessary to capture readings from the rangefinders installed on board, these can be circular or multi-beam lidars, depth or stereo cameras, etc.

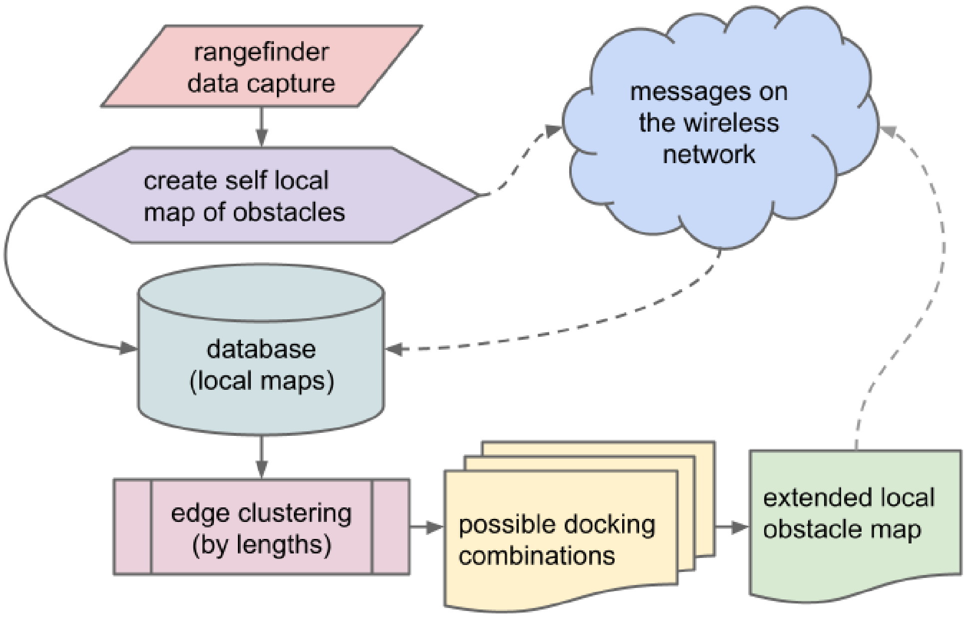


Fig. 3.2. Algorithm cycle to obtaining an extended local obstacle map.

3.2. Create self local map of obstacles

Then create self local obstacle map G_{self}. Here, rangefinder readings are used as input, but they should be prepared. It is required to isolate individual objects (obstacles) based on raw data g_i: determine the distance d_i, the angleai of deviation from its course on it and additionally evaluate its dimensions s_i. This is a very important stage, especially in practice when the sensor readings are noisy. The prepared map G_{self} should be packed into a message $msg_{self} \in MapMsg$ (supplemented with information U_{self} about its dimensions and the estimated trajectory of movement relative to the current position), and sent to the network so that other participants can also use it.

3.3. Database of local maps

Self message $msg_{self} =< self, G_{self}, U_{self} >$ and others $\{msg_{id}\}$ received from the network are stored on board in a special database for further operational use. Outdated records from the database can be automatically deleted. In this case, the component G of messages in the database will be represented in the form of a directed tree $G(V, E)$.

So each local obstacle map consists of a root node $id \in V$ and its associated terminal vertices $\{v = (id, a_i)|d(v) = 1\} \subset V$ as "star" topology. All nodes of this graph have the attribute s (object dimensions), and edges $e = (id, (id, a)) \in E$ with attribute d (distance of the obstacle). Examples of such "stars" are shown in Fig. 3.3.

3.4. Edge clustering by its lengths

From all relevant messages in the database edges $e_i \in E(G_{self} \cup G_{id})$ are extracted to perform clustering based on their lengths d with an unknown number of clusters. Methods like DBSCAN [6] are suitable for this task.

Note: When clustering edges, one can take into account not only their lengths, but also the dimensions of the nodes connected by them. Through this clustering, many sets $C_k \subset E$ of similar edges are generated. If any two edges $e_i, e_j \in C_k$ are from the same cluster, then the corresponding nodes (AV) are potential candidates for docking into a combination. In Fig. 3.3 edges from the same cluster have the same color.

3.5. *Many possible docking combinations*

For each cluster with multiple nodes (i.e. $|C_k| > 1$), its docking edges $e_i \in C_k$ can be represented by nodes of a connected graph $G^k = G(C_k, C_k \times C_k)$ of possible permutations within a given cluster. Joints within such a graph of permutations are mutually exclusive. Let $G^k!!$ denote their number.

Note: For a fully connected graph (upper bound), the number of permutations of the choice of all possible docking edges will be equal to the double factorial (3.1) of the number of nodes. If the number of nodes n is even, then take the nearest less odd: $n = n - 1$.

$$n!! = \frac{n!}{2^{\frac{n-1}{2}} \cdot (\frac{n-1}{2})!}. \tag{3.1}$$

If $n = 9$ then $n!! = 945$. Therefore, even with a small number of candidates, the total number of permutations may turn out to be quite large. The total number of permutations of all combinations is (3.2).

$$\prod_{|C_k|>1} G^k!! \tag{3.2}$$

Taking into account the specifics of the problem, the number of permutations can be significantly reduced to improve performance, for calculations on board. First, take

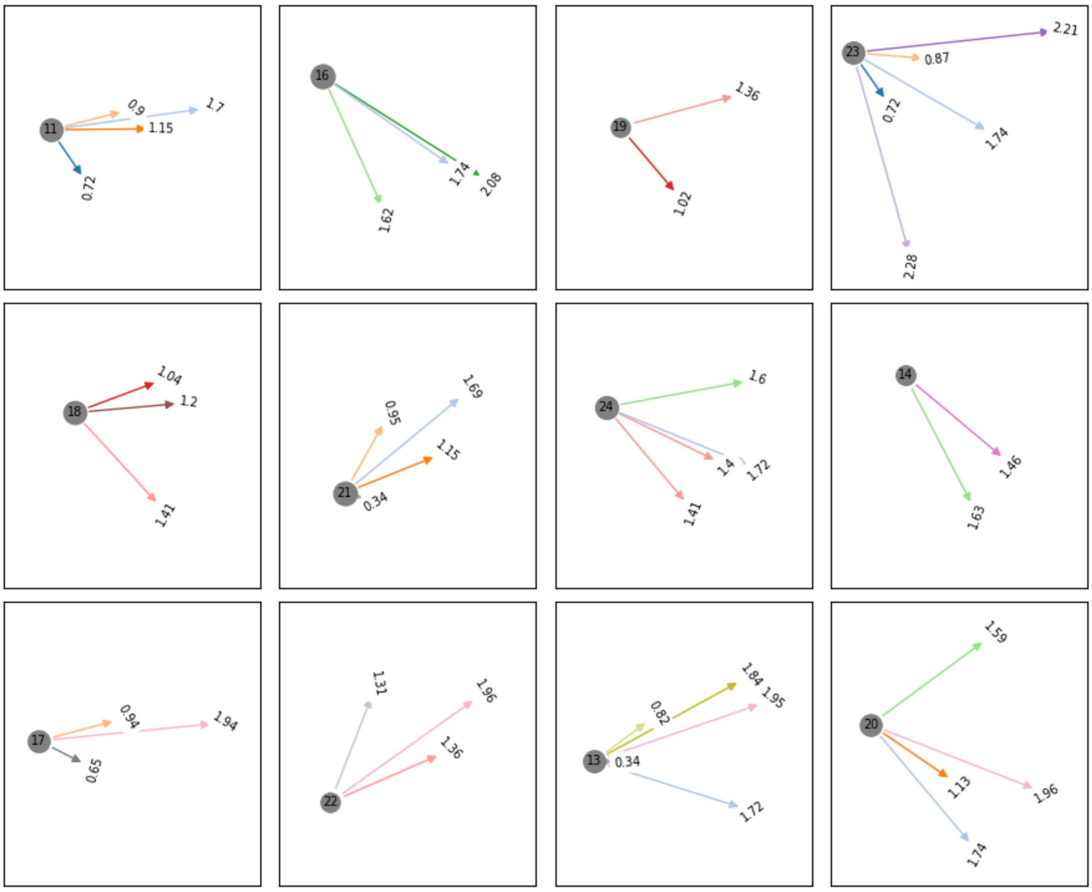


Fig. 3.3. Examples of docking items.

the possible docking with self AV edges obtained directly. Further, may be use not all combinations available on the network, and append the list of possible connections only twice ($K_1 = 2$).

1. Docking observed by self AV, because for successful docking the AVs must be in FoV (field of view) of each other.
2. Control docks for the appended list in the previous iteration.

In the process of forming possible combinations, invalid ones should be eliminated immediately, for example, based on limited FoV. The proposed docking may add to the participants' FoV some objects that they did not initially observe – these should be excluded. There are also cases when, in false positives, the identified node goes beyond the radius of the wireless network signal – it shouldn't be there. Another way to reduce the number of permutations is the probabilistic approach. The more samples in the cluster (more candidates), the more likely it is to misplace the nodes. Given the mutually exclusiveness of the docking, only a few of the most probable combinations can be selected. For simulation experiments, the value of the corresponding parameter $K_2 = 3$ was chosen (the algorithm still reliable).

3.6. *Optimal extending of the local obstacle map*

Of the many admissible combinations of docking, the most suitable one will be selected in accordance with the given criterion of optimality, which will be further used for navigation. As a result, its own extended local obstacle map $\hat{G}_{self} = (G_i)$ is a certain combination (ordered set) of docking permutations connecting all currently relevant "stars" into a single map. As an optimality criterion when choosing the most suitable combination of joints used:

$$\frac{|\{v \in V(G_{self})|d(v) = 1\}|}{|E(G_{self} \cup \{G_{id}\})|} \to \min \tag{3.3}$$

Here, the denominator is the total number of joining edges, and the numerator is the number of remaining terminal vertices. Also have ideas for using weighting factors to take into account the importance of joining certain nodes when calculating the combination scores.

<u>Note</u>: The resulting extended local obstacle map can also be sent to the network as the initial one, especially if there is high confidence in its reliability. An example of the result is shown in Fig. 3.4. The negative identifiers mean that this node is a common obstacle for several observers, but has never been identified on the network. The presence of such common obstacles, one might say, increases the reliability of the docking. The optimality criterion (3.3) does not guarantee perfect results and "false positives" are quite acceptable.

4. SIMULATION MODELING

Experiments have been implemented on the specially developed program simulator to evaluate the efficiency of the communication factor and collisions minimization. The AV model with a given frequency of the main cycle builds its local map, sends data to the network, receives messages from other ones, analyzes the situation, and selects maximally safe control approaching the destination based on the extended obstacle map.

Simulation scene is the corridor 10 m (see Fig. 4.5). The intensity of the AV model generator is 0.5 - 2.5 (units per second). There are 2 generators on the scene. AV models move in counterflows: one group (from generator 1, cyan) travels towards generator 2 (magenta) and vice versa. For each intensity, the experiment lasted 10 min. simulation time, statistics were collected every dt = 0.1 s. During this time, two generators are released on the stage somewhere from 600 (0.5 Hz) to 3000 (2.5 Hz) AV models. There are no side walls in this experiment to eliminate the congestion effect.

The parameters of the AV models at:

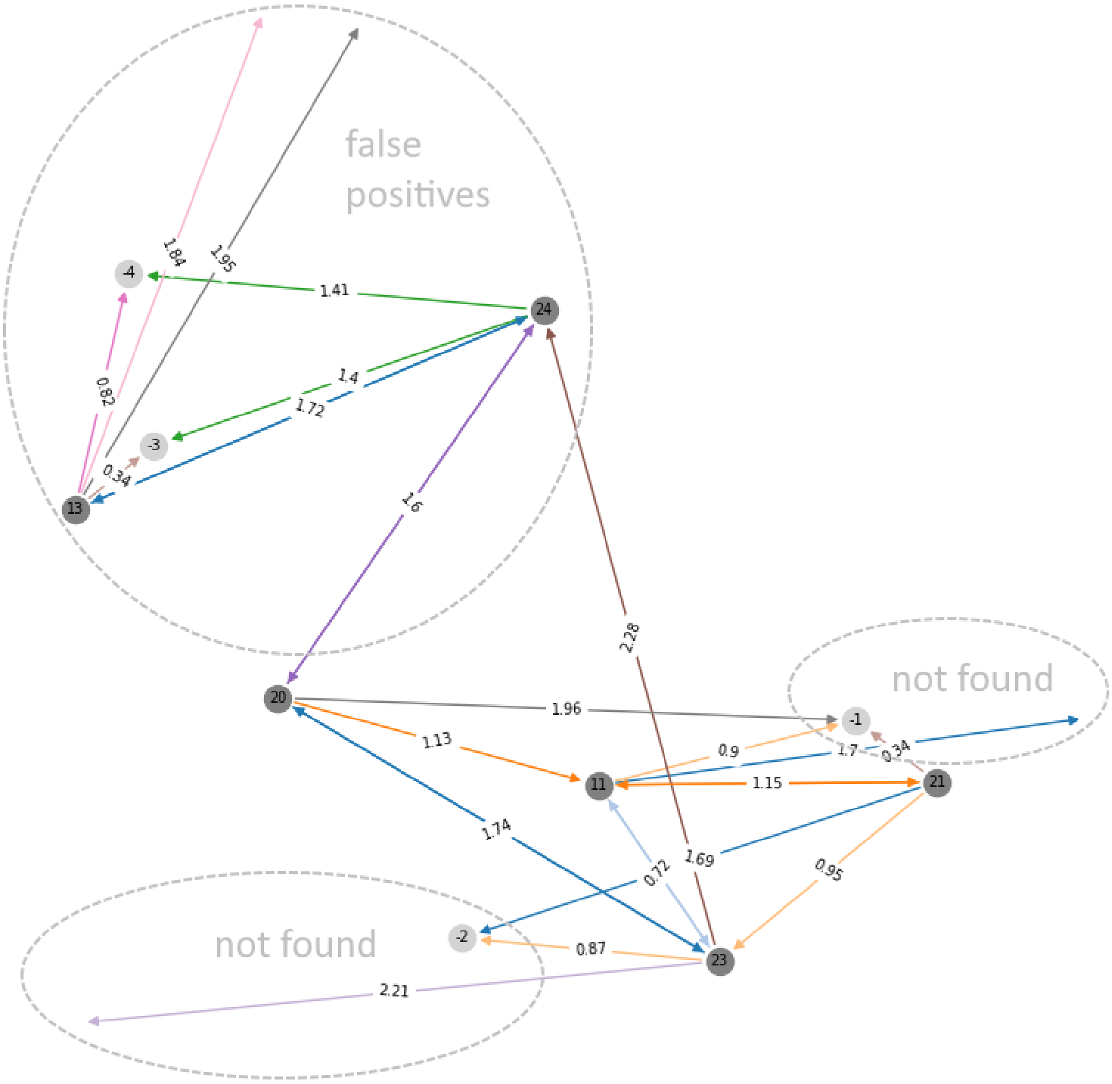


Fig. 3.4. Example of an extended obstacle map.

- linear velocities: 1.5 ± 0.5 m/s;
- speed of rotations: 180 ± 60 deg/s;
- dimensions (radii): 0.15 ± 0.05 m;
- wireless network signal range (radii): 2.5 ± 0.5 m;
- rangefinder maximum distance: 1.5 ± 0.5 m;
- field of view (sector): 180 ± 60 deg.

Series of experiments:

- T1 - everything is known about those who fall into direct observation (uses known dimensions and planned trajectories);
- T2 - direct observation (information from the network is not available, all observed obstacles are considered as static);
- T3 - the proposed approach based on an extended obstacle map (uses information from the network for those network participants whose position has been established, including false positives, all other nodes are considered as static obstacles).

5. EXPERIMENTAL RESULTS.

During the experiments, the following average indicators were evaluated:

- Count of $Hits$ (collisions).
- $Delay$ on the way.

$$Delay = \frac{ExitTime - StartTime}{MinTime} - 1$$

If it is possible to reach the goal in the minimum possible time $MinTime$, then the $Delay$ is 0, and if 1, then the $Delay$ relative to the minimum possible time is 100%.

The simulation results (see Fig. 5.6) have confirmed the efficiency of the proposed approach: the developed model enables one to considerably restrict the number of collisions under motion over non-regulated counter flows. Such a model of local positioning enables one to increase the awareness of the road traffic participants without a common regulation center. This approach will work both in networks with centralized common access points and in distributed self-organizing mesh networks [7].

In addition, a number of subtasks appear to reduce the number of permutations and develop a more reliable criterion of optimality to reduce the number of false positives, so that the method continues to work not only in conditions of artificial noisy readings of rangefinders, but also in reality, where there are a lot of other inaccuracies, for example, in the the experiments did not take into account the delays in data transmission over the network. In addition, in connection with the combinatorial problem, the question remains about the minimum sufficient number of successfully identified participants to ensure safe movement in clusters. At high intensities, even with $K_1 = 2$ and $K_2 = 3$, lengthy calculations had to be performed at the stage of enumerating combinations.

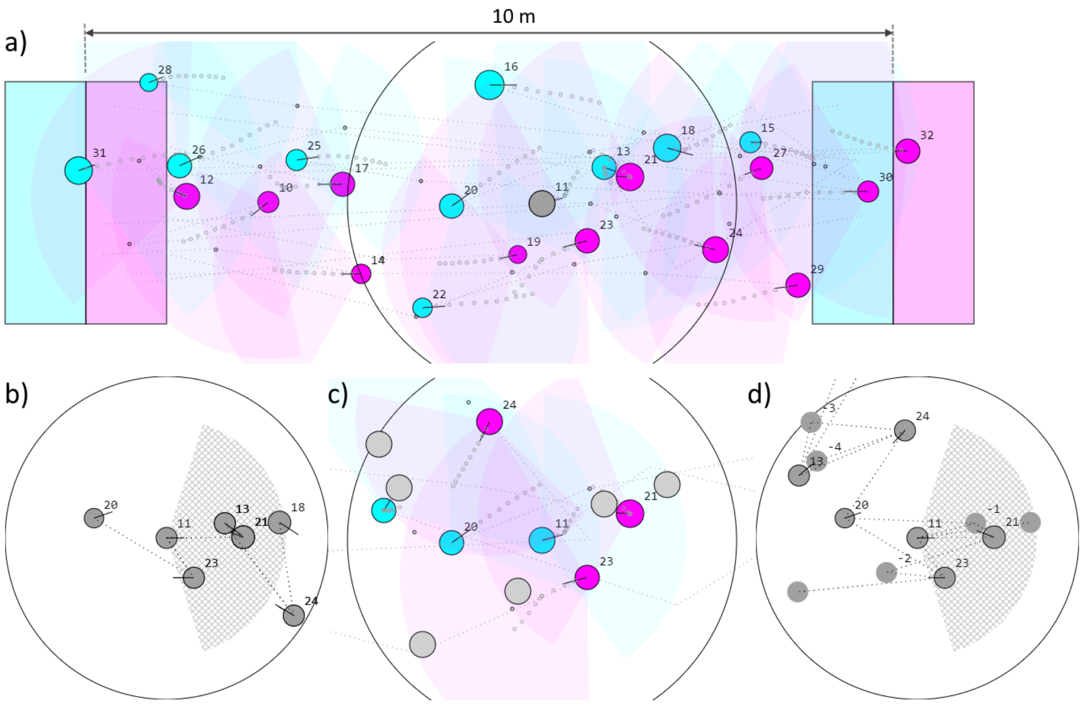


Fig. 4.5. Scene example of the developed simulator: (a) move area; (b) ideal extended local map; (c) proposed method in scene; (d) extended map by the proposed method with normed angles.

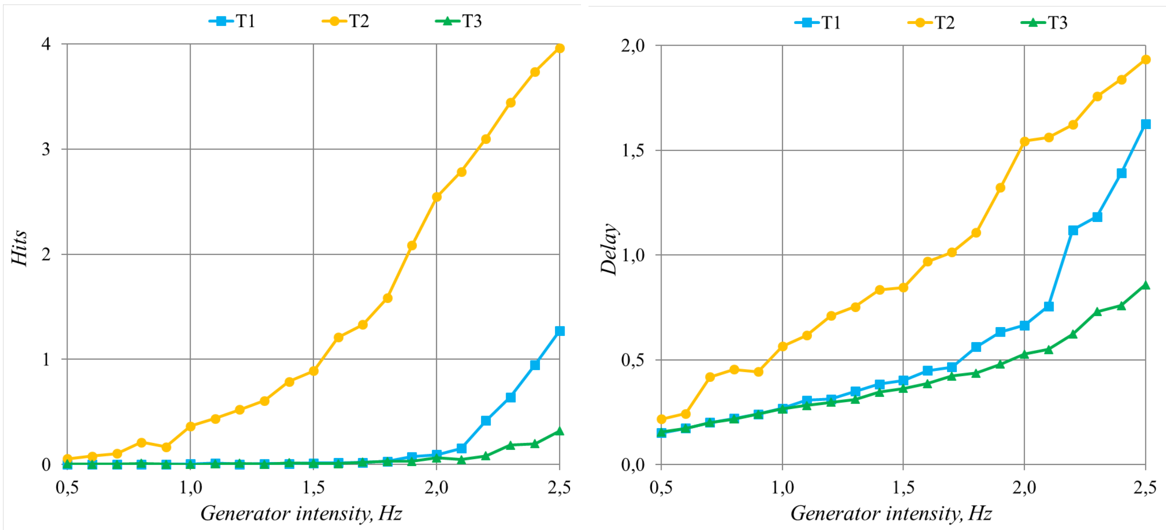


Fig. 5.6. Experimental results.

6. CONCLUSION

The smart city strategically considers transport development. The transport structure becomes intelligent. The tendency to apply dynamic and multi-modal information is appreciable in control of the city logistics. Big data are sampled from AVs sensors, safety cameras, RFID (Radio Frequency IDentification) labels, etc. The AVs take a particular place in the transport infrastructure, which requires the development of technologies: artificial intelligence, telecommunication systems, cyber-physical interfaces, information- and cybersecurity [1, 5, 8]. The "smart city" approach, oriented to safety, is to react not only to existing but also to occurring vulnerabilities, account dangers arising under developing new technologies. Du to that, Safety Management Systems [9–12] based on determining risks of different kinds (cybersecurity, efficiency, and reliability of software and hardware tools, etc.), will enable one to reveal vulnerabilities, dangerous factors, and to develop corresponding compensation measures, what, in turn, will increase the smart city CII safety.

Applying the network models of local positioning as additional method localization to organize the AVs traffic in the smart city will enable one to increase motion safety even in the case of the absence of GPS. In the case of successfully extending the local map of obstacles by comparing relative mutual locations, i.e., if for the AV will be correspondence between network addresses of message senders with their location on the map has been established, then it can able to use data received from the network. The simulation results show that the proposed approach allows to reduce the number of collisions when driving in counterflow traffic on the basis of the considered algorithm for extending the local obstacle map.

ACKNOWLEDGEMENTS

Section 3 of the reported study was funded by RFBR, project number 19-29-06044.

REFERENCES

1. Abdulov, A. & Abramenkov, A. (2018) Collision avoidance by communication for autonomous mobile robots in crowd. In *2018 Eleventh International*

Conference "Management of large-scale system development" (MLSD), 1–4, https://dx.doi.org/10.1109/MLSD.2018.8551804.

2. Fox, D., Burgard, W., & Thrun, S. (1997) The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, **4(1)**, 23–33, https://dx.doi.org/10.1109/100.580977.
3. Rösmann, C., Hoffmann, F., & Bertram, T. (2017) Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, **88**, 142–153, https://dx.doi.org/https://doi.org/10.1016/j.robot.2016.11.007.
4. Li, J., Ran, M., Wang, H., & Xie, L. (2019) Mpc-based unified trajectory planning and tracking control approach for automated guided vehicles. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, 374–380, https://dx.doi.org/10.1109/ICCA.2019.8899955.
5. Abdulov, A., Abramenkov, A., & Shevlyakov, A. (2018) Visual odometry approaches to autonomous navigation for multicopter model in virtual indoor environment. *Advances in Systems Science and Applications*, **18(3)**, 17–28, https://dx.doi.org/10.25728/assa.2018.18.3.583.
6. Khan, K., Rehman, S. U., Aziz, K., Fong, S., & Sarasvady, S. (2014) Dbscan: Past, present and future. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*, 232–238, https://dx.doi.org/10.1109/ICADIWT.2014.6814687.
7. Guss, S. (2016) Private wireless mesh networks. *Mathematical Structures and Modeling*, **4**, 102–115.
8. Promyslov, V., Sakrutina, E., & Meshcheryakov, R. (2019) Coherence criterion for security architecture of digital control system. In *2019 International Russian Automation Conference (RusAutoCon)*, 1–5, https://dx.doi.org/10.1109/RUSAUTOCON.2019.8867615.
9. Li, C.-Y., Wang, J.-H., Zhi, Y.-R., Wang, Z.-R., & Gong, J.-H. (2018) Simulation of the chlorination process safety management system based on system dynamics approach. *Procedia Engineering*, **211**, 332–342, 2017 8th International Conference on Fire Science and Fire Protection Engineering (ICFSFPE 2017), https://dx.doi.org/10.1016/j.proeng.2017.12.020.
10. Kalashnikov, A. & Sakrutina, E. (2019) Safety management system and significant plants of critical information infrastructure. *IFAC-PapersOnLine*, **52**, 1391–1396, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019, https://dx.doi.org/10.1016/j.ifacol.2019.11.393.
11. Sakrutina, E. & Kalashnikov, A. (2020) Towards the issue of the cybersecurity analysis of a significant object of critical information infrastructure. In *2020 13th International Conference "Management of large-scale system development" (MLSD)*, 1–5, https://dx.doi.org/10.1109/MLSD49919.2020.9247693.
12. Jharko, E., Abdulova, E., & Iskhakov, A. (2021) Unmanned vehicles: Safety management systems and safety functions. *Communications in Computer and Information Science*, **1396**, 112–121, https://dx.doi.org/10.1007/978-981-16-1483-5.

