

<http://doi.org/10.35784/iapgos.1512>

## SOLVING THE FAILING TRACK MARKER PROBLEM IN AUTOMATED GUIDED VEHICLE SYSTEMS – A CASE STUDY

Tomasz Lewowski<sup>1,2</sup>

<sup>1</sup>Wrocław University of Science and Technology, Faculty of Computer Science and Management, Department of Applied Informatics, Wrocław, Poland,

<sup>2</sup>Octant sp. z o.o., [www.octant.pl](http://www.octant.pl)

**Abstract.** *This paper is a case study of the development of a localization and positioning subsystem of an Automated Guided Vehicle-based transportation system. The described system uses primarily RFID markers for localization. In some deployments, those markers occasionally fail, mostly due to being crushed by cargo platforms operated by a human or due to internal defects. Those failures are not common enough to warrant switching from marker-based localization to a more sophisticated technique, but they require additional effort from maintenance staff. In this case study, we present our solution to this problem – a self-tuning algorithm that is able to detect marker failures and, in most cases, keep the system operational. The paper briefly discusses business circumstances under which such a solution is reasonable and then describes in detail the entire technical process, including data acquisition, verification, algorithm development and finally, the result of deploying the system in production.*

**Keywords:** industrial control, unmanned vehicles, fault tolerance, maintenance

### ROZWIĄZYWANIE PROBLEMU USZKODZEŃ MARKERÓW TRASY W SYSTEMIE OPARTYM O WÓZKI SAMOJEZDNE – STUDIUM PRZYPADKU

**Streszczenie.** *Ten artykuł opisuje studium przypadku rozwoju podsystemu lokalizacji i pozycjonowania w systemie opartym na wózkach samojezdnych. Opisany system używa markerów RFID w celu lokalizacji wózków. Markery te w niektórych wdrożeniach okazjonalnie ulegają uszkodzeniom – najczęściej mechanicznym, ze względu zgniecenia powstałe w wyniku przejechania przez platformę z ładunkiem kierowaną przez człowieka lub też wewnętrzne defekty. Uszkodzenia te występują na tyle rzadko, że nie uzasadniają zmiany sposobu lokalizacji na bardziej zaawansowany, jednakże wymagają dodatkowego wysiłku od kadry zajmującej się utrzymaniem ruchu. W tym studium przypadku opisane zostało rozwiązanie przyjęte w firmie Octant – samostrojący się algorytm wykrywający uszkodzenia markerów, w przypadku typowych uszkodzeń umożliwiającą kontynuację pracy systemu. Publikacja ogólnie opisuje sytuację biznesową w której zastosowanie takiego rozwiązania jest racjonalne, a następnie opisuje szczegóły techniczne podsystemów odpowiedzialnych za ruch i pozycjonowanie pojazdu – zarówno fizycznych, jak i w zakresie oprogramowania – oraz uzasadnienia dla podjętych decyzji technicznych.*

**Słowa kluczowe:** sterowanie przemysłowe, pojazdy samojezdne, odporność na uszkodzenia, utrzymanie ruchu

### Introduction

Octant sp. z o.o. is a company dedicated to industrial automation. Its flagship offering is an Automated Guided Vehicle (AGV for short) transportation system which is flexible and adjustable to the many custom workflows encountered inside factories. One of the supported versions uses passive RFID markers as the primary location indicator. Unfortunately, to provide sufficient precision, the markers have to be glued to the floor. As a result, they are subject to multiple causes of failures – usually mechanical. e.g. breaking the marker by running over it. While these failures are not frequent, they may sometimes happen up to several times per month. Since maintenance of the system is not performed on a 24/7 basis, the clients rightfully expect that failures of non-critical pieces of infrastructure, such as intermediate markers, will not cause the system to become inoperable, even if they may degrade system performance.

In this case study, we describe in detail the implemented subsystem responsible for AGV movement and positioning, describe and briefly analyse from a business standpoint several possible solutions to the problem of failing markers, show the details of the technical processes we undertook, explain them in detail and report the results.

The rest of the paper is structured as follows: section 1 describes the localization and positioning subsystems of other AGV systems available on the market and briefly mentions the architecture of our system. In section 2, we provide details on the movement and positioning subsystem in our system. In section 3, we discuss the causes of the problem, what solutions are feasible and what the costs and benefits are of each of them. Then, in section 4, we provide the details of our implementation and employed processes. We continue the discussion on the chosen solution in section 5, describing possible future system evolution scenarios, as well as accepted limitations. The paper concludes in section 6 with advice for future implementers of similar systems.

### 1. AGV systems

Automated Guided Vehicle (AGV) systems have been implemented in industry since at least since 1953 [1]. They have caught traction in recent years due to the sudden increase in available computing power mixed with increasing staff costs, more powerful algorithms, and development tools maturing to the point where they are ready for public usage. In this section, we will discuss localization and positioning in some of the AGV solutions present in the literature and in industry.

#### 1.1. Other solutions

According to [1], the first ever AGV was developed in 1953 by Barrett Electronics of Northbrook. Since then, AGVs have come a long way, including several navigation mechanisms – optic [6] and magnetic [10] line following, vision [7], laser [11] and others [14, 15]. Each of those solutions has been deployed in various circumstances and has a unique set of strengths and weaknesses, which means that each of them has one or more applications for which it is best fitted, and ones for which it may not be a good choice.

For example, tape- or wire-based navigation mechanisms cannot be applied in dynamic environments, where routes often change and attaching them to the floor would be impractical. On the other hand, they are some of the simpler mechanisms, requiring well-known hardware and not much algorithmic knowledge or processing power.

On the other side of the spectrum are AGVs using composite techniques, such as joint odometry and LiDAR [13] or location 3D mapping (SLAM, [9]). They are much more versatile, but require more sophisticated hardware, more processing power and much more complex control algorithms. As a result, one needs to find a balance between implementation simplicity, solution capabilities, maintenance effort and the need to retrofit the factory layout to support AGV-based transportation.

It is reasonable to assume that one should start with the simplest solution that is sufficient to solve the problem at hand. This is why we chose to implement magnetic tape-following as the first positioning technique in our AGV system. While there are multiple newer techniques (e.g. SLAM [9]), at the same time, we also cannot underestimate the additional effort coming from the need to manage 3D imaging (regardless of whether the data is delivered by a laser or by cameras) and managing collisions in environments that require a fairly stable route.

A system similar to ours was presented in [11] – it also used magnetic tape following as the main technique of guiding the vehicle and RFID tags for deciding the driving mode. The main difference is that our system uses data matrices and laser for precise positioning, while the system presented in [11] uses specific landmarks.

### 1.2. Our architecture

Our system consists of four types of active nodes: central supervision server, vehicles, stations and chargers. The first is a software-only node, deployed on a standard rack server. Technically, it can be deployed in an external data centre, but we recommend on-premise installation for both security and stability reasons. The other three types of nodes are direct hardware controllers, with mostly operational logic, i.e. the charger node knows how to perform the “start charging” operation, but does not know when to do it – all of the orchestration and task coordination is performed by the supervision server. A high-level view of the system architecture is shown in Figure 1.

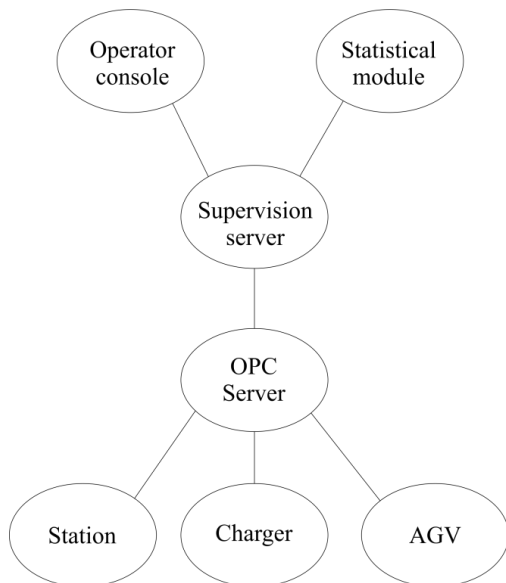


Fig. 1. Overview of system architecture

While having a single supervision server introduces a single point of failure, deployments of this system were not big enough to warrant the additional effort that we would need to put into guaranteeing processing continuity in case of major system outages (minor failures on hardware side are handled separately and will not be discussed in this paper). The hardware controllers are implemented in the ST language on Mitsubishi PLC devices. Those PLCs communicate with an intermediate server using a proprietary protocol, and the intermediate server communicates with our supervision server using the OPC UA protocol [8]. Of course, the whole system also contains other components – such as those dedicated to alerting, monitoring, dashboards and emergencies – but they are out-of-scope for this paper. The presented components are relevant here as they are either directly involved in positioning and track control (such as the supervision server or AGV), contain data used for control (statistical module),

or indirectly affect the routing by denoting special positions in the track (chargers and stations). In the rest of the paper, we will focus mostly on the central supervision server and the vehicle nodes.

## 2. Movement and positioning subsystem

The movement and positioning subsystem integrates data from four separate hardware mechanisms – line following, marker-based location, data matrix-based positioning and laser-based precise positioning. Their physical implementation spans over the station, charger and AGV components from Figure 1. These four subsystems and the core reasons for their implementation will now be discussed, and later on in the study, we will focus on the former two, since those are the ones that cause the most trouble. We believe that sharing our solutions will be beneficial for the research community as well as for practitioners. A summary of the four subsystems is presented in Table 1 and a diagram of their physical localizations on an AGV and station is presented in Figure 2.

Table 1. Summary of implemented movement and positioning subsystems

Subsystem	Function	Used hardware
Line following	Guiding vehicle along a track	Magnetic tape on the floor, reader on vehicle
Marker-based location	Marking important spots on the track – forks, checkpoints, action points	RFID markers glued to the floor, RFID reader on vehicle
Data matrix-based positioning	Reducing speed before action points	Data matrices glued to the floor, data matrix reader on vehicle
Laser-based precise positioning	Precise positioning between platform and vehicle	Cone-shaped cut in platform, laser rangefinder on vehicle

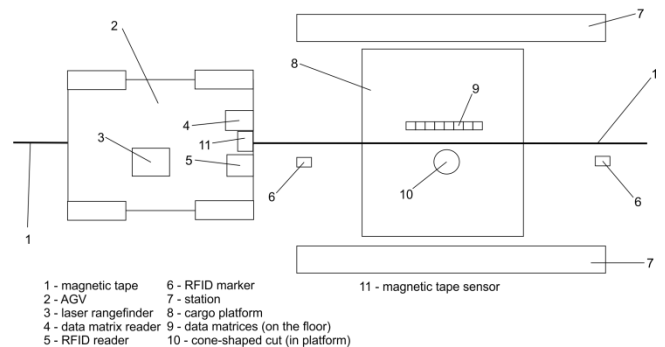


Fig. 2. Layout of components for movement and positioning systems

### 2.1. Line following

Line following is the primary movement pattern used in this system. A track made out of magnetic tape (numbered 1 in Figure 2) is glued to the floor. Each vehicle has a sensor (numbered 11) that lets it position itself on the tape and follow it, both when driving forward and backward. A vehicle cannot move to the side without rotating, but it can rotate in place – a feature that can be used to turn around or to perform a short search if the track is lost. Track segments are mostly unidirectional, though this is not an inherent restriction but rather an optimization choice – fewer bidirectional segments mean less need to manage potential collisions, therefore using unidirectional segments is likely to increase system throughput. A segment can fork into a maximum of two segments (“left” and “right”) – when a vehicle encounters a fork, it has to choose which one to follow. This choice is configured by the supervision server from Figure 1 before reaching the fork. Two segments can also merge in one point, in which case, the vehicle simply follows along. We do our best to make the graph (created from forks, merges and unidirectional

segments) strongly connected, but in some factory layouts, this is simply not possible. In such cases, we need to treat some segments as bidirectional and sometimes, in especially space-constrained factories, even provide a turnaround point for the vehicle. Segments are delimited by special RFID markers located next to the magnetic tape.

## 2.2. Marker-based location

RFID markers (numbered 6 in Figure 2) are used to provide the approximate vehicle location to the supervision system, so that future orders (such as taking the nearest turn or loading cargo on the segment) can be sent to the vehicle before the action needs to be undertaken. Sending the data upfront is important because factories can generate a substantial amount of radio noise and the connection between the vehicle and the supervising server may be broken. Additionally, there is always a delay between when vehicle reports its new status and when the supervising server is able to act on it (and vice versa), mostly introduced by the OPC server from Figure 1. These two factors considered together lead us make the design choice that, at any point of time, the vehicle should “know” what to do on its current segment and the following two. Such a buffer was experimentally proven to be sufficient in our setting. Sending future actions to the vehicle is possible because the route that the vehicle shall take is chosen when a new task is assigned to the vehicle, and not on the fly (however, if a chosen path becomes blocked – for example with cargo – the supervising server is able to recalculate the route). Markers are placed in important route locations – such as before and after a fork, and before and after a station or charger – and every few meters along the way. The RFID reader (numbered 5 in Figure 2) is slightly shifted relative to the middle of the vehicle. Thanks to this design choice, we can assign two RFID markers, one on each side of the track, to each physical location on a bidirectional segment and, on the system supervisor layer, treat each segment as unidirectional. This abstraction is not full – we still need to know which markers cannot be simultaneously reached by two different vehicles – but it is very helpful for route planning. For example, we can treat the whole track as a strongly connected graph where the segments are physically represented by the tape and the nodes are physically represented by the RFID markers. This allows us to calculate routes using well-known graph traversal algorithms [5]. Markers are also used to provide all sorts of on-segment navigation hints to the vehicle, such as the maximum speed allowed on the segment, an action that should be taken on the segment or an order to stop after encountering the next marker. Marker-based positioning is precise enough to locate places to stop or to turn around, but unfortunately had proven to not be precise enough for other operations – most notably positioning for charging and cargo acquisition. For those actions, we implemented an additional mechanism, based on data matrices.

## 2.3. Data matrix-based positioning

A data matrix is a two-dimensional binary image that is able to encode some information (the exact amount depends on the size of the data matrix). A single data matrix holds roughly the same amount of information as a single RFID marker, but multiple data matrices can be placed one next to another without interfering with each other. We use this tool to provide a method of reducing the vehicle speed – data matrices glued to the ground (numbered 9 in Figure 2) encode consecutive numbers in ascending order, while the supervision server sends the number encoded on the target data matrix. Since the vehicle internally knows how many matrices are left, it can reduce the speed and stop gracefully, without harsh breaks. This gracefulness decreases the maintenance costs of the vehicle and makes it consume less energy, which is very important in busy factories, where charging breaks cannot be long due to the amount of cargo that requires transporting.

For a long period of time, we thought that this technique should be sufficient to provide precise positioning. However, it turned out that it was not. The cargo acquisition mechanism consists of two pins mounted on the vehicle that are extended to fit slots present on cargo transportation platforms. While in this setup with data matrix-based positioning, the cargo was successfully acquired each time, the pins quite often did not fit well and they sometimes broke. This problem was increasing the maintenance costs of the system substantially, since every time a pin broke, the vehicle was rendered inoperable until the pin was replaced. Therefore, we decided to implement one final subsystem, dedicated to precise positioning – a laser-based one.

## 2.4. Laser-based precise positioning

Since we could not accept unplanned outages, we needed to solve the problem of breaking pins. After some research, it turned out that the problem was primarily caused by misalignment of the pin with its slot – the misalignment that was fairly hard to detect, as at least one of the pins ultimately made it into the slot each time (with high friction). However, the mechanism was designed so that the load would be carried by two pins – therefore while empty platforms could be carried after misalignment without much of a problem, loaded ones could not and caused the pins to break prematurely.

Unfortunately, it turned out that we needed to introduce another method of positioning – this time a very precise one. The chosen design included a laser rangefinder on the vehicle and a small cone-shaped cut in the cargo platform (numbered 3 and 10 in Figure 2). Once the laser hits the furthest point in the cone, the vehicle stops and begins the cargo acquisition sequence. This simple system worked surprisingly well, and was the final addition to the movement and positioning subsystem of Octant AGV. Interestingly, since this is a low-level system responsible only for operational behavior, if one does not explicitly require insight into the parameters (and we imagine that many users will not be interested in this insight anyway), it does not even need to be exposed to the supervision system, much less to the user. As such, this subsystem is implemented only in the AGV component from Figure 1 and is not propagated to the supervision system. Detailed data for diagnostics can be read from the OPC server.

## 3. The problem and the possible solutions

While single cargo transports were working well and factory workers were able to use the transportation system for several days in a row, every few days, the system had downtime. This downtime was related to the fact that an RFID marker had failed. These marker failures were at first related to internal marker failures, but after switching vendors and marker type, the failures did not stop, even though they became rarer. After an investigation, we tracked the root cause. It was predominantly a hardware issue – it was either broken due to physical impact – e.g. a manually-driven cart drove through a marker – or displaced (the glue used was not able to hold it in the correct position). Due to the failure mechanism, the most commonly failing markers were the ones placed in the open space, not the ones placed near critical track points (stations, chargers etc.).

From a business perspective, there were several ways to address such a problem:

1. Treat replacing broken markers as part of standard on-site maintenance and do not modify the system;
2. Switch to a different localization mechanism:
  - a. Slightly different – for example, use data matrices instead of RFID markers;
  - b. Substantially different - for example, use SLAM for localization;
3. Implement a mechanism to allow the vehicles to continue their task if the failed marker was not critical on the track.

In this section, we provide a detailed analysis of the costs and benefits of each of those solutions together with recommendations for when each of the presented choices may be optimal. A short overview is presented in Table 2.

Table 2. Overview of possible solutions for failing markers problem

Solution	When to use
Treat replacing broken markers as part of standard on-site maintenance	<ul style="list-style-type: none"> <li>System receives only minimal maintenance, non-critical defects are no longer fixed</li> <li>There are no engineers able to implement changes</li> <li>Income from this type of system is not a substantial part of the company business</li> </ul>
Switch to a slightly different localization mechanism	<ul style="list-style-type: none"> <li>System is already deployed, but still under active maintenance</li> </ul>
Switch to a substantially different localization mechanism	<ul style="list-style-type: none"> <li>System is still under active development and substantial changes in its design are acceptable</li> <li>Hardware can still be adjusted</li> <li>Pricing is not yet fixed</li> </ul>
Improve system resilience	<ul style="list-style-type: none"> <li>System is already deployed</li> <li>There are engineers able to implement changes</li> <li>System will be offered to future customers</li> </ul>

### 3.1. Treat replacing broken markers as part of standard on-site maintenance

This choice, while terrifying for an engineer, is surprisingly often reasonable from a business perspective. If a defect in the system does not cause additional harm and the work done by the system either can wait or there are sufficient failover procedures, the defect may not be worth fixing. This is most common in systems where the system original creators and maintainers are no longer supporting it or the support is very expensive.

The main benefit of this choice is its simplicity – it does not require any additional action except those which have already been taken (e.g., providing failover procedures). The cost – for the client – includes increased maintenance, and an increased need for support and disturbances in the production flow. If the expected frequency of failures is known, these costs can be reasonably estimated. However, we act not as a client, but as a manufacturer. In this context, the most important costs are those related to the company’s image. Obviously, if a freshly deployed system requires constant maintenance, this means that it has defects. Even if leaving these defects in the system would be the financially optimal solution in the short term, its effect on the company’s image would be devastating. As a young company, we could not afford dissatisfied clients, so we needed to provide a solution.

### 3.2. Switch to a slightly different localization mechanism

Since the localization RFID markers were found to not be sufficiently resilient (even after switching to a different RFID vendor), and we had a different localization mechanism already in place (data matrices) an obvious decision would be to drop the dependency on RFID markers altogether and simply introduce a distinction between data matrices made for speed decreases and data matrices serving as location markers. However, the characteristics of those two types of readers (RFID reader and data matrix reader) are slightly different and they cannot be used as a drop-in replacement for each other.

This solution might have required some changes in vehicle physical equipment – for example using a faster or more precise camera – but definitely would require significant changes in the track – namely removing all RFID markers and replacing them with data matrices. However, it was not clear whether implementing such a substitution would actually help, or would the data matrices break as well – considering also that the system works in industrial settings, there can be a substantial amount of dirt, which may affect ability of the camera to read the markers. While this is not a problem in the areas where data matrices are

used (such as stations and chargers), it might be on the open floor. The data matrices would also be glued to the floor the same way RFID markers are, which means that they could also be accidentally removed the same way that the RFID markers were, for example by cleaning machines.

### 3.3. Switch to a substantially different localization mechanism

Some of our potential clients complained about the need for a line and the concept of line following – while it was technically working, we were not able to convince them that it can withstand the tough conditions inside a factory. Therefore, another idea was born – we could entirely replace line following with a system such as SLAM [9]. This would definitely solve any problem we had with the markers (effectively rendering them useless), while also greatly improving our offering by removing the need to modify the factory space for the purposes of our system.

However, the costs of such solutions would be too high to implement just to address a defect in the earlier design – the additional scanners alone would consume any profit our company made on the deal, not to mention the additional software development costs (a substantial portion of the system supervisor and all of the code from the AGV component from Figure 1 would have to be rewritten) and probably also significant hardware changes because the amount of raw data generated by the additional devices would require it to be, at least partially, processed on the vehicle.

While the costs of this solution made it infeasible for us to implement it for this particular client, it is possible that a SLAM-based solution will be one of our future offerings.

### 3.4. Improve system tolerance for marker failures

Finally, we could leave all of the hardware and overall architecture and adjust only the parts of the system responsible for handling failure cases, such as reaching a wrong marker or losing the track. Since this would be a software-only change, it could be deployed with relative ease and without introducing an additional maintenance break – only a software update would be needed, and those took no more than several minutes. Having no downtime and no changes in the physical part of the system are valuable benefits, but there are costs as well. Most importantly, this would only help assuming we could estimate the expected marker location with good precision – to be precise, if a failed marker is used to mark the location of an important action point, such as a station or charger, we need to know that we lost it before the action needs to be started. There is also one more caveat – if failing markers do not cause visible loss of operability, such a failure might go unnoticed. One unnoticed failure would be fine, but if many markers failed, the system would start to behave in an unpredictable way, probably in situations where it would be most unexpected. Choosing this a strategy would mean a serious investment in system monitoring and alerting, not only during development, but also permanently in maintenance – alerts are no good if nobody sees them. This means that improving system tolerance would not only require that the system work when a marker has failed, but also that the system can recognize a failed marker and report this failure to the maintenance staff and, further, that the maintenance staff know how to address the problem.

### 3.5. Why we chose improved failure tolerance

After many discussions, we settled on improving system tolerance for marker failures. This decision was based on our business situation – we were the manufacturer, so we could not ignore those failures, and the system was freshly deployed, so at that stage we could not afford to introduce significant changes in hardware. We could either substitute RFID markers with data

matrix markers or improve system resiliency. Since marker substitution at the time seemed like the choice bearing the greater risk – high maintenance impact and hardware changes without really any guarantee that it would improve the situation – we decided to improve system tolerance for marker failures. This decision was also predicated on the assumption that such a feature would be an enhancement to any future deployments we may contract and would be a valuable addition to our current know-how, especially in the area of system monitoring and alerting.

## 4. Implementation details

After deciding to improve system tolerance for marker failures, we needed to choose how to implement this feature. The symptoms of marker failures were fairly consistent, and involved markers that could not be read – never incorrect reads. Due to this, it was obvious from the beginning that for each point in which our vehicle finds itself, we needed to be able to determine which marker was the last one read if the system behaves correctly. To determine this, the obvious solution is to use the distance driven – since the distance between the markers is constant as long as the track itself is constant, we should be able to measure the distance between the markers and estimate whether the vehicle has driven far enough yet.

### 4.1. Measuring distance driven by a vehicle

The vehicles are equipped with two electrical engines – one on the left and one on the right. On both of these, we can control the speed and measure it using embedded encoders. This speed should be easily convertible into covered distance, assuming a known wheel size (which we of course know). However, there are several cases in which this conversion is not so obvious. Some of those cases include wheels slipping, which indicates various problems – for example an invalid hardware configuration in the case of harsh start under load or insufficient floor cleaning in case of oil puddles. Others are expected – since we measure the distance on the left and right engines separately, we effectively get the distance traveled by the left wheel of the vehicle and the distance traveled by the right wheel, as presented in Figure 3.

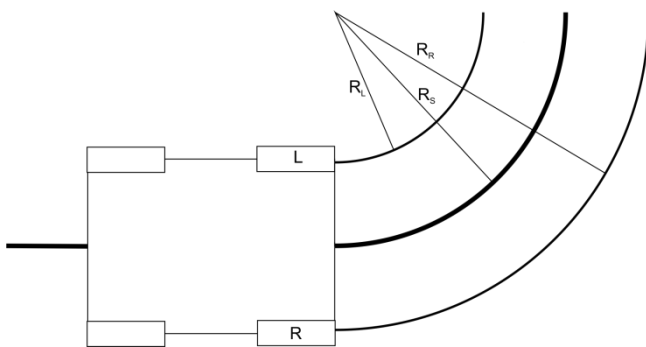


Fig. 3. Distances measured by the AGV at a perfect turn

When driving forward these distances both equal, but on forks and turns they are not, and it is even possible that one of the wheels will drive a negative distance (for example during in-place turnaround, as presented in Figure 4). Calculating the covered distance from this data seems simple – the equations for the position of a non-slipping differential drive vehicle are well-known and can be found in multiple publications, for example [2]. However, in practice, this problem turned out to be challenging – in many cases, the wheels were slipping, the track was glued in unexpected patterns (for example to avoid some obstacles) and the PID regulators did occasionally behave in an unexpected way. While there is body of research that would allow us to deal with those issues – for example a model of slipping was shown in [11]

– including all of the relevant phenomena would substantially complicate the implementation.

Not only that – in fact, calculating the distance would not make our system any more accurate, as we are really interested in any distance measure – it does not really matter how exactly the measurement is performed, as long as it measures the right thing (which we know it does, as we explicitly measure distance) and is consistently done the same way the entire time.

Therefore, we decided to follow a different path – use as little data processing as possible and, wherever possible – work on data gathered directly from the sensors and stored in statistical module shown in Figure 1.

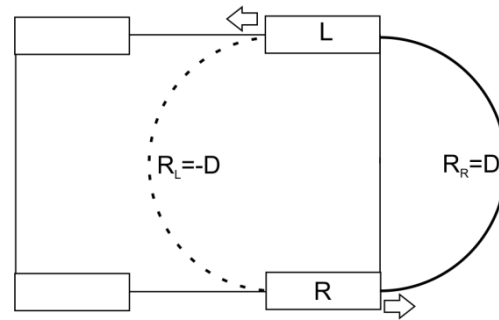


Fig. 4. Distances measured during a perfect turnaround

### 4.2. Estimating our distance expectations

We started the implementation by gathering actual distance data from the running system. We took measurements for two days in a controlled environment, with maintenance staff on-site, as well as for the next four weeks when the system was running without direct supervision. This allowed us to gather over 300 samples for each segment distance. Those measurements were meant to be used to enrich the track graph with the distances between the markers. Finally, we needed to decide what values for those distances would be acceptable.

After the initial measurements we had a good understanding of the measured distance distribution – this was clearly a normal distribution. For all marker pairs that we measured, we ran a Kolmogorov-Smirnov test and received a p-value  $> 0.999$ .

The variances differed between segments – both in absolute values and in the segment distance percentages. This was not a problem, since we could generate proper distributions for each segment on system start up without incurring too high a load on the system.

Still, there were a few other decisions that needed to be made: How many false positives could we accept? How big a deviation in measurements is acceptable, and when does it become an issue?

### 4.3. Tuning system parameters

These decisions resemble ones that were the subject of quality control research done by Deming [4] and Taguchi [3]. For example, the acceptable deviation can be considered to be the signal-to-noise ratio in Taguchi's research. While we are aware of the applicability of those techniques, at the time, we believed that they were slightly too formalized for our needs. Nevertheless, the route we took was heavily inspired by them.

In our case, a false positive is a situation when a marker exists on the track, but the system decides it is not there. The only situation where this happens is when measured distance  $d > d_{max}$ , so we are only interested in one side of the distance distribution tail. From a business perspective, on-site maintenance was performed once per week anyway, so the specialized staff could deal with the false positives then, together with the real failures. However, false positives should not happen any more

often. During regular system operations, we measured that an AGV was reaching markers roughly once every minute. This means that our false positive rate could be greater than  $\frac{1}{7 \cdot 24 \cdot 60} \approx 10^{-4}$  for a single vehicle. The deployment had two, so the maximum rate was  $5 \cdot 10^{-5}$ . On the other hand, we did not want to increase the distance too much, so that we would not encounter the next marker before the end of the estimated distance (this constraint was dictated by another part of system architecture, not discussed in detail in this paper). These hard constraints are schematically shown in Figure 5 –  $d_{max}$ , the maximum distance that a vehicle can drive between T1 and T2 must be bigger than  $d_{min}$  (the “real” distance between T1 and T2) – otherwise virtually every run would be a false positive – and less than  $d_{err}$  (the distance between T1 and T3) – otherwise the vehicle would reach T3 before driving the maximum distance, and that is not very useful (except for cases when two consecutive markers fail at the same time, but those cases are very rare and we can safely ignore them in the current discussion).

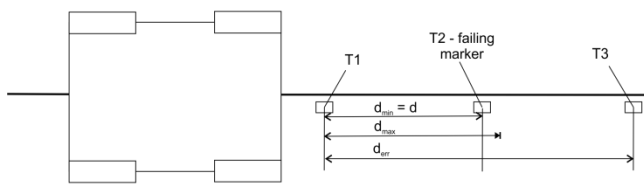


Fig. 5. Acceptable maximum distance range

Since we confirmed that our distance variable has a normal distribution and we are interested in one-sided probability of a sample being outside our window, we have reduced our issue to a standard statistical problem with well-known solutions, namely finding value for cumulative distribution function of normal distribution such that  $CDF(\mu + A\sigma) \geq p$ , where  $\mu$  is the average for the distribution,  $\sigma$  is its standard deviation,  $p$  is 0.99995 and  $A$  is the variable for which we are solving the inequality. Cumulative distribution function for normal distribution cannot be expressed in terms of elementary functions, but nevertheless it can be either looked up in pre-calculated tables or calculated with arbitrary precision by mathematical software. The smallest  $A$  for which our hard constraints were fulfilled was 3.89.

We decided to add a little buffer and in the implementation use  $A = 4$ , which means that measurements up to four standard deviations above the average were considered valid results.

In the end, we were expecting roughly 1 false positive every 10 days on a two-vehicle setup and 1 false positive every 21 days on a one-vehicle setup, which was acceptable. The markers failed more often than that and the maintenance staff needed to replace them, so checking up on a single false positive was no longer a significant problem.

To address the variance in the measurements, we analysed the data acquired when the system was running under supervision. The measurements were very stable and even on the shortest tracks, the maximum standard deviation did not exceed 1% of the total length, and generally were equal to at most a few centimeters, often below 2 cm. With such measurements, we were confident that, as long as the system continues to operate correctly, the distance at which a marker failure is detected will be smaller than the distance to the next marker or to the decision point. We could be sure of that, because for no two consecutive segments  $S1, S2$ , the following inequality holds:  $S1 + S2 < 1.04 \cdot S1$ .

Since 1% of the total distance as acceptable for us as a threshold value, and the existing design did not cross this threshold, we simply decided that a standard deviation should not

exceed 1% of the total length of the segment. If it exceeds this, the system shall report a warning that a given segment length is unreliable and the maintenance staff should take a closer look at it. Increased variance is usually result of ad-hoc track changes (e.g. gluing the marker back in a slightly different location). The length distribution is calculated on start up from data that was obtained during earlier transports. Such an approach could cause trouble should we not take special precautions. Those precautions include providing the maintenance staff with a possibility to manually discard measurements that are no longer valid (e.g., due to track layout changes) and automatic discarding of suspicious measurements. “Suspicious measurements” include outliers and records where an unexpected situation – such as a track loss or emergency break – occurred.

#### 4.4. Reporting the failures

After the system recognizes that a marker is missing, it needs to do two things: first, it needs to report this failure to the operator or maintenance staff, and second, it needs to somehow adjust the current task assignments and expected track, so that the system remains operational. The first action is simple – the system can show it on the UI, send an email or perform any other notification action – and we will not discuss it any further, since any further action is up to the maintenance staff. The second one, however, is much more complex and needs to be analysed in several contexts, since it may cause various effects – this case is shown in Figures 6 and 7.

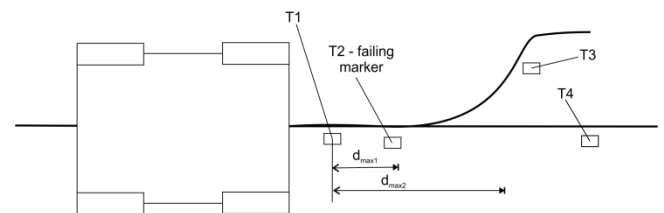


Fig. 6. Failure of a marker that denotes an action

If a marker is lost when driving forward without any intention of turning or performing an action, the only thing that needs to be done is to readjust the next marker and next action - nothing would happen anyway, so stopping the system here would make no sense.

If failing marker denotes a turn, like in Figure 6, the behavior depends on the value of  $d_{max}$ , which can be anywhere between  $d_{max1}$  and  $d_{max2}$ . If it is close to  $d_{max1}$ , the turn will always be taken correctly – in which case no further adjustments need to be taken. If it is close to  $d_{max2}$ , the behavior is effectively undefined (to be precise, it depends on the initial conditions – previous turns taken), so the vehicle might take the wrong one. In such case, the supervision system may also need to readjust the current track or even tasks of the vehicle – for instance, it may request the vehicle to turn around at the nearest possible point and start the task from the beginning, or assign another one, depending on the details of the scheduler configuration.

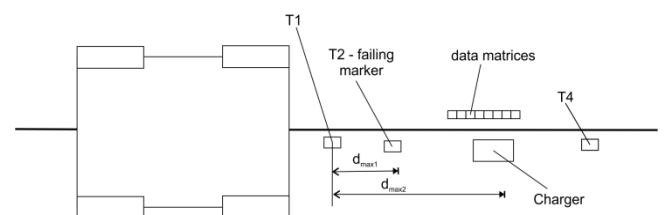


Fig. 7. Failure on marker denoting action point

If failing marker denotes an important action point, like in Figure 7, again everything depends on the accuracy of distance estimation – if the measurement distribution is accurate ( $d_{max} \approx d_{max1}$ ) and it is already known that the marker is missed before the data matrices start, the system can behave like when a marker is lost on the way forward and simply continue its work. On the other hand, if the distribution is flat, it may be impossible to position correctly on this action point. This case may require immediate maintenance, if the vehicle drives into the station with too high speed. Luckily, action points are the most often used points in the whole track, so there is usually a sufficient number of measurements to provide an accurate estimation. They are also shielded against human error by the fact of being located inside a station, and the platform only fits into the station in one position, so it is difficult to accidentally break it.

Two markers lost in a row usually mean that there is some external failure, for example the RFID scanner, so immediate system maintenance is recommended, although not always strictly needed.

## 5. Limitations and future work

Our solution, while working well enough for us and our customers, has some imperfections and constraints. First, it only works if the distribution of measurements is Gaussian – this means that negligence in either vehicle engine configuration (by our engineers) or in floor maintenance (by our client's cleaning staff) may cause this method to occasionally fail and return false positives. We did confirm that distributions were indeed Gaussian in the deployment where we needed it by acquiring measurements during regular system operations over several weeks and verifying the results with the Kolmogorov-Smirnov test. However, such an analysis needs to be applied to every deployment, which complicates maintenance, especially track modification. After several months, we found that some distances do not adhere to the Gaussian distribution. Luckily, the markers on the affected segments were protected and the distribution was a result of other phenomenon (mostly wheels slipping during initial acceleration), hence no software updates were needed.

One important limitation of this solution is that, while we are able to easily reassign tasks that were not started, it is not that easy if the cargo has already been acquired and the vehicle is driving with a platform. A platform is a fairly big physical object and there are parts of the track which can only be navigated platformless – for example, an empty vehicle can drive through a station (even if there is a platform on said station), but a vehicle carrying a platform cannot do that. Due to this limitation, if a new marker is lost on a route taken with cargo, the vehicle may drive into an area that cannot be left with cargo. In such cases, manual action is required – however, this is a simple action of releasing the cargo and moving it to the closest station, so it does not require attention from specialized maintenance staff.

In the implemented version of this technique, we recalculate the measurement distributions only during system start up. While this works well enough in our scenarios, it may occasionally cause major changes in distances – for example after an ad-hoc repairs, when a marker is attached in a slightly different location. It also requires a short maintenance window, just to adjust the distances. This does not seem necessary, and it is likely that the system could implement a more flexible approach, especially since track changes are by no means rare.

Track reconfiguration – such as adding new forks and segments – require some effort to be incorporated into the system – if a marker is moved, earlier readings have to be marked as “no longer valid”, and at least a few measurements should be taken in a supervised scenario. However, if a new segment is added without moving any existing markers, the system will gradually adapt to the expected length of this segment – in the first run, it

will be expected that the marker is in the right place, but later – after some measurements of the segment are taken – the system will automatically start to monitor the distance traveled and will detect any failure. Adding new segments and markers requires changes in configuration files, and while this could be automated so that during a test run, the vehicle would perform such a reconfiguration itself, such a feature is not yet needed.

In hindsight, it might have been possible to avoid using RFID markers, if we had understood early enough that we would need to support data matrix detection. In such a case we could probably just replace the markers with data matrices and avoid installing one more piece of equipment on the vehicle. On the other hand, the characteristics of those two types of readers are vastly different, and it might not be possible to read a data matrix when driving at the desired speed, so more research will be needed before making such decisions.

Regardless, many enterprises seem to prefer solutions which do not require line following. It is conceivable that one of the future generations of Octant AGVs will not be line-following vehicles, but instead will use some other means of orienting in space, such as SLAM. However, we still believe that this case study may be useful to future implementers as an example of iterative system development.

## 6. Conclusions

In this paper, we presented a detailed case study of a problem with failing location markers that we encountered at one of the deployments of our AGV system. We presented details of the location and positioning subsystems, made of four hardware components – line following, marker-based location, data matrix-based positioning and laser-based precise positioning. Then we discussed in more detail the problem of failing markers and four possible solutions – treating failures as part of regular maintenance, changing the location system to data matrices alone, changing the location system to SLAM and implementing extra features for improving system resilience for marker failures. Next, we explained that in our case, we chose to improve system resilience to provide this feature to future users while not spending an immense effort on implementing something that should be just a defect correction.

In the case of our system, the most beneficial choice was to improve system resilience. To do that, we gathered data from a long period, analysed it to understand the underlying distributions, then made a simple statistical model based on the gathered data and applied it to the vehicle in operation. We also made sure that all special cases – namely action locations and forks – are taken care of, and any possible error that might occur is correctly handled.

When addressing the challenge presented in this paper, we analysed several options at each implementation step (whether the problem should be addressed at all, how much effort to spend on it, what the expected effects are). We learned that while we, as a manufacturer, care about the system being in a good overall shape, the clients (and, more importantly, factory staff) care most about the system being operational, regardless of whether internally it works correctly or not. In hindsight, this seems obvious, but at the time, we perceived this as a substantial threat – failures might go unnoticed in a system that stays operational, and when the system breaks down, the resulting pressure makes it a hard time to perform a proper diagnosis, especially when there are multiple failures affecting each other. This is why system monitoring is critical to providing an acceptable level of service.

With the effort put into monitoring, we could react to failures before the client realized that there is one, even without having permanent, on-site, 24/7 support. Software systems used today allow for an immense amount of monitoring and we believe that using these and implementing proper system supervision and support is the core of maintaining system health.

We hope that this paper will be useful to future implementers of similar systems and will help them avoid some of the pitfalls we encountered along the way.

## Acknowledgement

This research is partially financed by European Union project RPDS.01.02.02-02-0027/17-00, "Opracowanie innowacyjnego systemu transportowego z zastosowaniem pojazdów autonomicznych" under the Lower Silesian Regional Operational Programme 2014–2020.

## References

- [1] Bandyopadhyay S.: Intelligent Vehicles and Materials Transportation in the Manufacturing Sector: Emerging Research and Opportunities. IGI Global 2017.
- [2] Campion G., Bastin G., D'Andréa-Novel B.: Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. Proceedings IEEE International Conference on Robotics and Automation (IEEE) 1993, [http://doi.org/10.1109/ROBOT.1993.292023].
- [3] Clausing D.: Taguchi methods to improve the development process. IEEE International Conference on Communications – Spanning the Universe 2, 1988 826–832, [http://doi.org/10.1109/ICC.1988.13674].
- [4] Deming W. E.: Sample Design in Business Research. Wiley-Interscience, 1990.
- [5] Dreyfus S.: An Appraisal of Some Shortest-Path Algorithms. Operations Research 17(3), 1969, [http://doi.org/10.1287/opre.17.3.395].
- [6] Ismail A. H., Ramli H. R., Ahmad M. H., Marhaban M. H.: Vision-based system for line following mobile robot. IEEE Symposium on Industrial Electronics & Applications 2009, 642–645, [http://doi.org/10.1109/ISIEA.2009.5356366].
- [7] Lee J-W., Choi S-U., Lee C-H., Lee Y-J., Lee K-S: A study for AGV steering control and identification using vision system. IEEE International Symposium on Industrial Electronics Proceedings 3, 2001, 1575–1578 (Cat. No. 01TH8570), [http://doi.org/10.1109/ISIE.2001.931941].
- [8] Leitner S. H., Mahnke W.: OPC UA—service-oriented architecture for industrial applications. ABB Corporate Research Center 48, 2006, 61–66.
- [9] Li L., Schultze L.: Comparison and Evaluation of SLAM Algorithms for AGV Navigation. F.-J. Villmer, E. Padoano (Eds.): Department of Production Engineering and Management. Production Engineering and Management. Lemgo 2018.
- [10] Pang Y., De La Cruz A. L., Lodewijks G.: Bipolar magnetic positioning system for automated guided vehicles. IEEE Intelligent Vehicles Symposium 2008, 883–888, [http://doi.org/10.1109/IVS.2008.4621228].
- [11] Park J., Kim J. Y., Kim B., Kim S.: Global Map Generation using LiDAR and Stereo Camera for Initial Positioning of Mobile Robot. International Conference on Information and Communication Technology Robotics (ICT-ROBOT), 2018, 1–4, [http://doi.org/10.1109/ICT-ROBOT.2018.8549897].
- [12] Park J., Lee J., Park Y., Kim S. W.: AGV parking system based on tracking landmark. 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology 2009, 340–343 [http://doi.org/10.1109/ECTICON.2009.5137022].
- [13] Quan S., Chen J.: AGV Localization Based on Odometry and LiDAR. 2nd World Conference on Mechanical Engineering and Intelligent Manufacturing 2019, 483–486, [http://doi.org/10.1109/WCMEIM48965.2019.00102].
- [14] Ray K. A., Gupta M., Behera L., Jamshidi M.: Sonar based Autonomous Automatic Guided Vehicle (AGV) navigation. IEEE International Conference on Systems and System Engineering 2008, 1–6, [http://doi.org/10.1109/SYSOSE.2008.4724179].
- [15] Wiklund U., Andersson U., Hyypä K.: AGV navigation by angle measurements. Automated guided vehicle systems: Proceedings of the 6th International Conference 1988, 199–212.
- [16] Wang D., Low C. B.: Modeling Skidding and Slipping in Wheeled Mobile Robots: Control Design Perspective. IEEE/RSJ International Conference on Intelligent Robots and Systems 2006, 1867–1872, [http://doi.org/10.1109/IROS.2006.282309].

### M.Sc. Tomasz Lewowski

e-mail: tomasz.lewowski@pwr.edu.pl

Tomasz Lewowski is currently a Ph.D. student in the Department of Applied Informatics of Wrocław University of Science and Technology. His primary research interests include software engineering and software quality. He is also heavily involved in the software industry, designing and implementing systems in a variety of areas, including transportation, telematics and banking.



<http://orcid.org/0000-0003-4897-1263>

otrzymano/received: 2.03.2020

przyjęto do druku/accepted: 15.09.2020