Approaches to fast sequential inventory and path following in RFID-enriched environments

Arne Bosien* and Volker Turau

Institute of Telematics
Hamburg University of Technology,
Hamburg, Germany
Email: arne.bosien@tuhh.de
Email: turau@tuhh.de
*Corresponding author

Franco Zambonelli

Dipartimento di Scienze e Metodi dell'Ingegneria,
Universitá di Modena e Reggio Emilia,
Via Amendola 2, Pad. Morselli,
42122 Reggio Emilia, Italy
Email: franco.zambonelli@unimore.it

Abstract: Although RFID is mostly used for ticketing, e-passports and supply chain management applications, the technology is also suitable for transponder based navigation systems. There it can be applied to leave a virtual trace on writeable RFID tags. These applications use RFID in a different way, because the tags are stationary and related to each other, since each tag of the path points to the following one. This offers the possibility to develop more efficient algorithms for their detection. In this paper, we suggest to store the identifying masks of subsequent tag IDs on the tags. The approach can easily be used with query-tree anti-collision methods. We examine the requirements for such an approach and describe a concrete algorithm, which is simulated and compared with different anti-collision algorithms.

Keywords: RFID; inventory; anti-collision; pheromones; ants; path following; navigation; algorithms; query-tree; ISO-15693; mask storing; virtual trace.

Biographical notes: Arne Bosien is a PhD student of Computer Science at the Hamburg University of Technology. He received his master in 2007 and started his research on RFID-based navigation systems at the Institute of Telematics. In 2010 he received a scholarship to spend a research period at the University of Modena and Reggio Emilia.

Volker Turau is a Professor of Distributed Systems at the Hamburg University of Technology in Germany. Since 2008 he is the Head of the Institute of Telematics. He received a PhD in 1984 from the University in Mainz and held post-doctoral positions at the Universities of Manchester and Karlsruhe. After that he was a Visiting Scientist at the University of Berkeley in California and Hewlett-Packard Labs in Palo Alto. His general research interest falls within

the scope of distributed systems with a focus on wireless sensor networks, energy aware computing and fault tolerance. He has published five books and more than 50 scientific papers in international journals and conferences. He regularly serves as a member of programme committees of international conferences.

Franco Zambonelli is a Professor of Computer Science at the University of Modena and Reggio Emilia. His research interests include distributed and pervasive computing, complex and adaptive systems and autonomic computing. He received his PhD in Computer Science from the University of Bologna in 1997. He is senior member of IEEE and ACM.

1 Introduction

RFID tags are a powerful tool for pervasive computing scenarios (Roussos and Kostakos, 2009). RFID-enriched environments can be used for a variety of purposes, such as localisation tools, for realising the Internet of Things (Fleisch and Mattern, 2005) vision, as well as for helping navigating in a physical environment. Different scenarios can be imagined, where RFID-based navigation can be applied. This can be the case for Automated Guided Vehicles (AGVs) in industrial environments, better orientation for blind people or simply to help people to move in the environment (Raoui et al., 2009).

The basic idea is to leave a virtual path [similar as ants do with pheromones (Bonabeau et al., 1999)] by writing relevant information on the RFID tags. Afterwards the paths can be retrieved using this information (Mamei and Zambonelli, 2005). However, there are subtle problems involved in correctly reading the tags and enabling an effective following of the path. While common RFID applications use a stationary reader and mobile tags, RFID navigation uses this the other way around. The main challenge is the movement of the reader.

Because the reader moves continuously, the communication process is more difficult. Especially if a smooth movement without interruptions is to be achieved, all interactions with the tags are time critical. The tags have to be identified, their positions have to be estimated, and the stored content has been read and processed to determine the correct path direction. In order to offer a high navigation speed, a fast detection (inventory) of relevant tags is crucial. This requires very efficient algorithms.

In this paper, we explore this issue and propose new algorithms. These are evaluated through a sequence of simulations. We show that it is possible to identify a suitable algorithmic approach to make the inventory and path following problems effective. Although this paper is written with navigation scenarios in mind (e.g. for AGVs), we are well aware that the problem is a general one. Thus, the study of algorithms for moving RFID readers is important, since they can be employed in a variety of situations where RFID tags are related to each other.

The remainder of this paper is organised as follows: In the next section, we describe scenarios for the application of RFID-based navigation as well as motivations for new anti-collision algorithms. Section 3 gives an introduction into the main principles of path following and inventory. Section 4 presents the approach for a new inventory that is based on the storage of masks. The following section discusses algorithms that are used for the inventory process. Furthermore, the mask storing algorithm is explained in detail. Simulation results of these algorithms are shown and discussed in Section 6. Section 7 reports on related work, and the paper closes with the conclusion in Section 8.

2   Scenarios and motivations

2.1   Scenarios of RFID-enriched environments

RFID is an example for a technique to realise automatic identification system. Passive (i.e. batteryless) tags can be mounted on entities to allow an easy identification by information systems. If the tags are stationary, they can be used as landmarks for the identification of a fixed position. Furthermore, many tags allow the limited storage of arbitrary data, opening new usage scenarios. Due to the frequent use of RFID, cheap tags are currently available, which allows their massive deployment in applications (Bohn and Mattern, 2004).

Nowadays, RFID-based navigation is used in different ways. The Container Terminal Altenwerder (CTA) in Hamburg (Metro AG, 2006) and a similar project in Rotterdam (Texas Instruments, 2000) use a grid of RFID tags to enable the exact positioning of AGVs. Via the tag ID and a map with corresponding geographic positions, the position of a vehicle can be detected. Several thousand tags are embedded into the pavement to allow this usage.

The University of Hanover (Niemann et al., 2006) has researched a system where the embedded tags store their geographic position. That way it is no longer necessary to maintain a global map with all tag positions. Furthermore, the tags can contain the position of the next tag the vehicle is supposed to move to.

In Laveno Mombello in Italy a system has been tested to give blind people orientation in an unknown area (Peck, 2008). For this purpose, an RFID reader has been integrated into a white stick and connected to a PDA. Headphones allow the communication with the person. As in the previous example, RFID tags are embedded in the environment and denote positions via their ID. The PDA uses a map to inform the person about its current position and to give other relevant information, for example whether the person is standing near a traffic light where it is required to wait for green light.

Pheromone-based systems use a different approach. They follow the example of the behaviour of ants which use a scent to mark the path from a source of food to the ant hill. Thus, other ants are also able to trace this path. This is similar to the concept of continuous track guidance of robots in industrial environments. RFID tags can be used to emulate this principle.

Mamei and Zambonelli (2007) suggested a pheromone-based approach for relocating lost objects. A person is carrying an RFID reader that detects tags, which can be distinguished as object tags (mounted on objects) or as location tags (mounted at fixed positions). When a person moves with an object, the ID of this object is written into the location tags, enabling the discovery of this object at a later point in time.

This approach can also be used for AGVs (Bosien et al., 2008). During a teach phase a vehicle is writing markers onto tags in the vicinity. Later, during the follow phase, other AGVs can detect the marked tags and thereby discover the path.

2.2   Motivations

Path following has to be built on a reliable and fast inventory process. The application has to detect the available tags, read out their contents, process and evaluate the provided data and compute the desired direction. The performance of the path following algorithm is strongly connected to the underlying inventory. It is obvious that the speed of the path following vehicle is tied to the inventory speed (Bosien and Turau, 2009).

To enable the communication with a specific tag, the RFID reader has to know the ID of the tag (Finkenzeller, 2003). The process of gaining all IDs of tags in range is the so-called inventory process. The details of this process depend on the used RFID technology and the implementation, but the principles are equal. First of all, the reader sends a command causing all tags in range to respond: if more than one tag responds, a collision does occur. In the next steps, the reader tries to resolve these collisions by refining its commands to cause fewer tags to respond until finally just one tag responds so that its ID can be detected.

The requirements for an inventory to enable path following differ from those for a traditional inventory. In our application the tags remain stationary while the reader moves. The inventory is no longer a static process rather it has to be regarded as a continuous process. This leads to new challenges the algorithms have to face. For example it is likely that tags will reappear in two sequential inventory processes when the reader has moved only slightly. In this case, the newly appeared tags are of greater interest for the navigation algorithm than the old ones. Furthermore, the path following algorithm is not interested in all detectable tags. Important are those tags which contain path-related data.

In RFID-based navigation approaches tags remain stationary and have a fixed set of neighbours. This situation offers new possibilities commonly used inventory algorithms are not able to take advantage of. Once the path is settled, the order of appearing tags during the path following is fixed. This condition can be used to speed up the inventory process.

3 Basic principles of path following and inventory

This section gives an overview of the principles of RFID inventory and path following. Since the path following relies on an efficient inventory, both tasks are strongly connected.

3.1 Path following

The main task of the path following is to determine the relative position of the reader to the path and to calculate the direction in which the path continues. For this purpose it has to use the information that has been left on the tags during the teaching phase. This information can consist of a sequence number (to enable self-crossing paths and a correct recovery of the path), a path identifier (to allow different paths at the same time) and the current driving direction of the vehicle (to ease the task of finding the next tag). The teaching of the path can be accomplished by a human-steered vehicle that is driving the same route; another vehicle should follow later automatically. Figure 1 shows an example for the applied approach of Bosien et al. (2008).

Only if the path following works in a sufficient way is the inventory algorithm able to detect relevant tags. Otherwise the vehicle may be offside the path in an area where no relevant tags are available (see Figure 2). Since path following is also a complex task, we do not deal with this problem in this paper. By this means it is avoided that the result of the performance of the inventory is influenced by the quality of the path following algorithm. We create suitable conditions for efficient navigation algorithms. The navigation system also has to face problems related to the physical issues (e.g. the accuracy of the AGV driving system).

Figure 1   Basic example of an environment for path following, L/M/R indicate the positions of the tags, also sequence numbers are shown (see online version for colours)
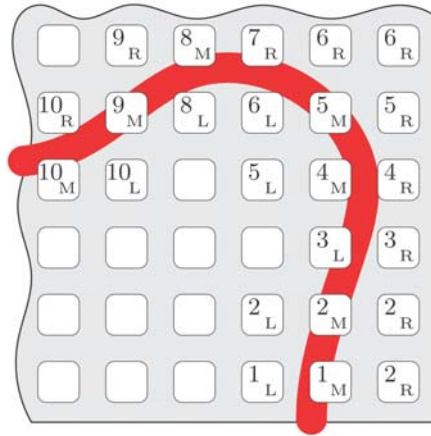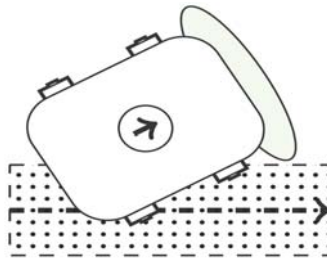

Figure 2   Vehicle besides the path and heading in the wrong direction

3.2   Inventory and identifying mask

ISO-15693 is one of the standards specifying the communication protocol and physical characteristics for RFID readers and tags. It works in the high frequency domain of 13.56 MHz and offers a maximum read distance of more than 1 metre. In general, it is well suited for the applications under consideration in this work (Baum et al., 2007).

This paper uses ISO-15693 as reference for all implementation specific details. Nevertheless, the discussed problems are general ones and the approach is suited for all query-tree anti-collision algorithms. The overall idea is adoptable to other technologies.

An advantage of ISO-15693 is that the complete anti-collision process is controlled by the reader via query-tree algorithms. This is a stateless approach posing low demands on the tags and allowing the development of new algorithms by just changing one component of the RFID system while the remaining parts of the system remain untouched.

The query-tree anti-collision process tries to detect a mask that only matches with one tag ID. A mask (or prefix) is a part of the tag ID that distinguishes the tag from other tags handled by the algorithm. Given this mask, it is possible to recover the whole tag ID. Those masks which uniquely identify a tag within a certain detection area are called the *identifying masks* of the tag.

Consider an example where the IDs of two tags are 1110 and 1100. A tag will respond to an inventory request if the least significant bits of its ID match the transmitted mask. While the mask 0 matches both tags[1] (11x0), the mask 10 identifies the first tag (11<u>10</u>) and the mask 00 identifies the second tag (11<u>00</u>).

The complete tag ID is globally unique, whereas the identifying mask only uniquely identifies a tag within a given area. If the number of tags or IDs in range changes, the identifying masks will also change. In general, the larger the area wherein the mask must be unique (which means a higher number of involved tags), the longer the mask will become.

Figure 3 shows another example of four tag IDs (0111, 1011, 1001 and 1000) and their respective identifying nodes[2]. The masks 111, 011, 01 and 0 identify the given tags in the case of these four tag IDs.

Figure 3   Unbalanced tree of four tag IDs, the identifying nodes are highlighted (see online version for colours)
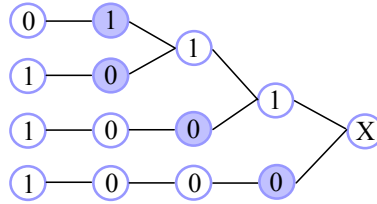
The detection of identifying masks and the complete tag ID is time critical since the reader is constantly moving and the amount of time in which tags are available for detection is limited. Obviously, the more tags that require detection, the longer the anti-collision algorithm will take. Some existing algorithms perform the mask detection by tree-walking using depth-first or breadth-first search, a rather time consuming process.
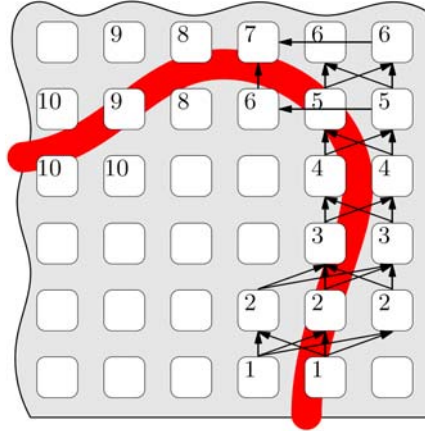
4   Approach

The anti-collision algorithm spends most of the time searching for identifying masks. Clearly, the inventory can be accelerated if the identifying masks are already known. In this case it would not be necessary to use the inventory in its traditional style and it would be possible to probe directly for the given masks. Therefore, we propose to store the identifying masks of the tags that come next on the path into the tags. Moreover, this gives the possibility to distinguish between relevant and irrelevant tags and the inventory is no longer in need of detecting all tags.

Figure 4 illustrates this approach in respect to Bosien et al. (2008). The path is divided into ten steps. Each tag belonging to a certain step stores the identifying masks that are necessary to detect the tags of the next step. For example all tags of step 2 point to all tags of step 3, this means that all tags of step 2 store the two masks to identify the tags of step 3.

This approach is feasible if the route and the IDs of the next tags are already known. In this paper, we do not deal with the problem how such a path can be set-up. In general, this is possible if the path is written starting from the end heading towards the beginning (like ants do) during the teaching phase. Then all future tags will be known. It is also imaginable that an AGV uses two readers during the teaching phase, one head-mounted and one on the rear side. The front reader is used to calculate the identifying masks which are written by the rear reader.

Figure 4   Approach of mask storing applied to an RFID path (see online version for colours)

The approach is based on the results of Bosien et al. (2008). If another navigation technique is used, a different handling of the masks can be considered as possible, but the main idea of storing identifying masks should remain valid.

4.1   Determination of the identifying mask

To identify a tag by an identifying mask, the area in which the identifying mask is valid has to be chosen. This area depends on the range of the reader. Furthermore, attention has to be paid to the question of when a tag can be expected to become detectable.

- *Optimal case*: In this case (see Figure 5) the tag is detected immediately when it enters the interrogation area of the reader. This means that the mask only has to be unique in the current field of the reader.

- *Average case*: It is more probable to assume that a tag is detected at any time while it is in range of the reader (Figure 6). In this case the mask has to be unique within the whole area that the reader could cover while the tag is in range.

- *Worst case*: It also is imaginable that the tag is not detected while it is in range of the reader. In this case the used mask will lead to the detection of other tags or none. To prevent this, the mask would have to be unique within the whole area. Since this is not reasonable, this case must not occur, hence, the speed of the reader has to be reduced.

Figure 5   Detection of a tag when entering the interrogation area. The reader is moving from left to right, the detected tag is highlighted, the circle represents the range of the reader (see online version for colours)
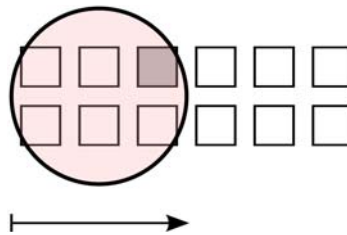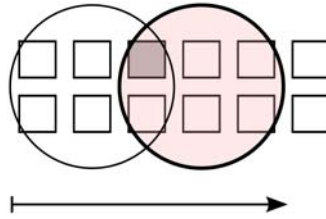
Figure 6   Detection of tag at any time while it is in the interrogation path. The two circles represent the possible extreme interrogation areas (see online version for colours)

4.2   Mask length

If the identifying mask is very short (e.g. 1 bit or even empty), the probability to detect a non-intended tag is very high when changes are applied to the tag field or the reader will detect more tags than initially thought. Hence, a minimal mask length must be adhered to. The length of the mask depends on the number of tags in range.

4.3   Further possibilities

In former approaches a path identifier has been used to ascertain if a detected tag belongs to the desired path or not (Mamei and Zambonelli, 2005; Bosien et al., 2008). If each tag may only be used to store one data set of a single path, it is feasible to omit this path identifier, since our approach delivers the sought-after tags from the beginning. But it must be noted that this procedure lacks robustness. For example if the two tags of step 3 (Figure 4) fail, it is no longer possible to recover the remaining part of the path. Therefore, this possibility can only be used when a high redundancy of tags is given and the detection of the next step can be guaranteed.

     The usage of the next masks provides a natural order. Therefore, it may not be necessary to use sequence numbers to detect which are the tags for the next step. On the other hand, this can lead to complications if tags are not used exclusively for one data set and if self-crossing of one path should be allowed.

4.4   Drawbacks

Besides the detection of the tag IDs the reading of the contents of the tag also consumes time. In general, it holds that more data require more time. Since data on ISO-15693 tags are organised block-wise, this may not apply, but to store and read many or long parts of tag IDs, which requires the usage of more than one block, will reduce the overall speed. Moreover, it should be kept in mind that the memory of tags is limited and other data also have to be stored.

5   Algorithms

In the next part the following algorithms for the inventory process are presented:

- a recursive anti-collision algorithm, as given in the ISO-15693 specification
- an improved recursive algorithm that uses a start query
- the proposed mask storing algorithm

5.1   *Different anti-collision algorithms*

This section deals with two existing anti-collision algorithms that are later used in the evaluation of our approach. The non-optimised anti-collision algorithm (noAC, Algorithm 1) has been chosen for simulation. This is the simplest algorithm possible. It is described in the ISO-15693 standard as a basic algorithm (ISO/IEC 2000). It uses recursion to implement a simple depth-first search. Although this standard is a little dated, it is still used in many cases and the principles of the algorithm are still valid.

The algorithm starts with an empty mask. With high probability the response to a request without a mask will have collisions. In this case the mask is extended by 0 and 1 and the recursion starts. If no collisions have occurred, then at most one tag will have responded. Thus, either a tag is identified or no tag is in the range of the reader.

Algorithm 1: noAC

Data: mask
response = sendRequest(mask)
if *response has collisions* **then**
    noAC(0 + mask)
    noAC(1 + mask)
else
    if *reponse contains ID* **then** return ID
    end
end

Bosien and Turau (2009) analysed different algorithms regarding their suitability for path following. The Q2 algorithm uses a start query of length two and other optimisations and showed the best performance in most cases. Therefore, we also use this algorithm to evaluate our mask storing algorithms.

The Q2 algorithm is a modification of the noAC and can be split into two parts. It uses a start query (Algorithm 2) which contains all possible masks of length two (00, 01, 10 and 11). Furthermore, it uses the two optimisations *jump to collision position* and *two at once* for the real anti-collision (Algorithm 3). Details can be found in the Bosien and Turau's (2009) work.

Algorithm 2: Q2

---

startMasks = {00, 01, 10, 11}
for all the *mask $\in$ startMasks* **do**
    enhAC(mask)
end

If the response just contains one collision, it means that two tags have responded and the corresponding IDs have a Hamming distance of 1. In this case we can directly identify two tags without an additional request. If more collisions have occurred, it will be examined at which bit positions of these have occurred and the next mask will be expanded until this position.

Algorithm 3: enhAC

Data: mask
response = sendRequest(mask)
if *response has 1 collision* **then**
    ID1 = replace collision with '0'
    ID2 = replace collision with '1'
    return ID1 and ID2
else if *response has collisions* **then**
    mask = expanded mask until 1st collision
    enhAC (0 + mask)
    enhAC (1 + mask)
else
    if *response contains ID* **then**
      return ID
    end
end

5.2 Mask storing anti-collision algorithm

5.2.1 Preliminary thoughts

If a mask storing anti-collision algorithm has no information about the set-up of the tags either in the field or the actual speed of the reader, it has to deal with the problem of deciding what to do in case no new tags are detected. The detection of no tags may have two reasons:

Reader moved insufficient distance (due to low speed): This yields that the identifying mask is not yet valid in this area. This may lead to:

- no responding tags, because the right tag is not yet in range

- collisions, but this can only happen if the area wherein the identifying mask is valid is too small and old tags reappear in the current anti-collision run.

Reader moved too far (due to high speed): This yields that the identifying mask is no longer valid in this area, this may lead to:

- no responding tags, because all tags involved in the inventory process have changed and the original intended tag has already left the range of the reader

- collisions, because now other tags also match the same mask (if the area wherein the identifying mask is valid is too small) or complete other tags are involved in the inventory process if the original tag has already left the range of the reader.

In the first case, when the reader movement has been insufficient, the algorithm should probe the same mask again in the next loop execution. If the reader has moved too far and in addition to that, there are no responding tags, the algorithm would have to restart in order to do a complete inventory. If collisions occur, the algorithm could also resolve these collision in a standard anti-collision procedure (e.g. with noAC or Q2).

5.2.2 *Mask storing anti-collision algorithm*

The anti-collision algorithm presented in the following considers the second case, since we are more interested in a good behaviour for higher speeds of the reader. If the first case occurs, the algorithm will also work but with a lower performance.

The mask storing algorithm (MS, Algorithm 4) works as follows. If the algorithm starts or the previous run has not delivered (`isEmpty()`) any identifying mask (`identMasks`), `startMasks` is set as an empty mask. Otherwise – and this is the desired case – `startMasks` is set as the found identifying masks. For each `startMask` a common anti-collision algorithm is used to find the whole tag ID. In case the `startMask` is a valid identifying mask, the algorithm should immediately detect the tag ID. If this does not happen due to collisions (e.g. in case the `startMasks` field is set as an empty mask), the collisions will be solved in the standard way.

Algorithm 4: MS

---

if *not isEmpty(identMasks)* **then**
    startMasks = identMasks
else
    startMasks = {}
end
for all the *startMask* ∈ *startMasks* do
    foundTags += anticollision (startMask)
end
 clear (identMasks)
forall the *foundTag* ∈ *foundTags* **do**
if *isOld(foundTag)* **then**
        continue
end
readD at a (foundTag)
if *belongsToPath(foundTag)* **then**
        nextMasks = extractMasksFromData()
        if *isNewData(nextMasks)* **then**
           identMasks += nextMasks
   **end**
end
return identMasks
end

All found tags are collected in the `foundTags` field. After this step `identMasks` will be cleared, since it is necessary to store the new identifying masks.

Next, each found tag is handled separately. First, it will be checked if the tag has already been used before in a previous loop (`isOld()`). In this case, the algorithm proceeds to the next tag, since we are only interested in new data. Furthermore, even if we were interested in the data of these tags, it would not be required to read the tag's content again, since this has been done in a previous run and it would be possible to reuse the (hopefully) cached information.

After reading the contents of the tag, it is decided whether the tag belongs to the desired path (`belongsToPath()`). If this is the case, the stored masks are read (`extractMasksFromData()`).

Finally, it is checked whether the found identifying masks do not point to tags that have already been found in the current run (`isNewData()`). This case can occur if the reader detected tags of two different steps (see Figure 7). Then the stored masks of the earlier step will point to tags that have already been detected. This filtering is also of importance if the loop starts without identifying masks.

Figure 7   If the reader detects masks of previous steps (step 4 or 5), filtering will be required (see online version for colours)
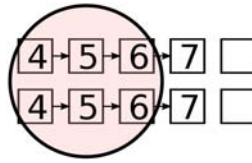
The employed standard anti-collision algorithm was changed in a way so it will not try to find all tags in range and will just return the first encountered tag. This improves the start behaviour, because the initial detection and reading of all tags in range consumes much time. After the first tag is detected, the algorithm benefits from its own strength to find following tags quickly.

5.2.3 *Constraints, implementation*

The algorithm does not use a path identifier to distinguish between near or crossing paths. For the simulation we will always use one path per time. The algorithm could easily be enhanced to use such identifiers. Also, sequence numbers are not used to determine an ordering over the tags.

The used memory has been limited to one block of 32 bits. In this block we are able to store up to seven entries with a maximum mask length of 7 bits.

6   Simulation

The used software simulates the ISO-15693 protocol regarding timings and packet format. It does not apply a channel model; therefore, all data are exchanged without loss within the specified range of the reader. If the tag is not within the interrogation area, no communication is possible. The position of the moving reader is rectified after each communication step.

For the simulation we chose a symmetric tag field (as shown in Figure 4) and set up a straight path. The reader is always placed in the centre of the path and is moved alongside.

To allow the tracking of a path, the navigation algorithm depends on an efficient inventory. On the other hand, if the navigation does not work properly, the path following algorithm will also influences the performance of the inventory (according to

Section 3.1). By choosing a straight path (which surely is not a real world scenario), we prevent the result of the simulation from being influenced by the quality of a path following algorithm, since no navigation is required.

Although the field is symmetric, the exact positions of the tags are varied within their given area. Hence, big leaps in the number of involved tags are avoided when the range of the reader is altered only a little.

The simulation uses realistic values regarding the number of tags in range. While some papers evaluate the behaviour of their algorithms against several thousand tags, here the tag number results from tag distance and reader range.

6.1 Simulation objectives

One goal of the simulation is to compare our approach with existing anti-collision algorithms. We want to know how the algorithm behaves regarding different speeds. The simulation tries to prove that our algorithm is superior to existing algorithms if only a subset of tags requires detection.

Furthermore, we want to examine the influences of varying the reader's range and the tag density. Both parameters affect the number of tags within the interrogation area.

In Section 6.6, it is of interest how the mask storing algorithm reacts to changes in the tags field. These changes are caused by removing tags (e.g. in case of failure).

6.2 Simulation of different AGV speeds

As described in Section 5.1, we chose noAC and Q2 to compare our mask storing approach by means of simulation. noAC is chosen as reference to show the impact of the mask storing algorithm compared to Q2, which stands as a representative of an already improved algorithm for RFID navigation.

Furthermore, three slightly different mask storing algorithms have been simulated. MS is the algorithm that uses the noAC algorithm to solve collisions. In case no masks have been detected previously, it will search for the first available tag as described in Section 5.2.2. MS-Q2t is a variant that uses Q2 for collision handling. MS-Q2f does the same, but in contrast to MS-Q2t, it tries to find all tags in range and not just the first one.

It must be noted that the simulation in this paper puts higher demands on the algorithms than the simulations of Bosien and Turau (2009). In our case a tag is considered detected if its ID has been discovered and, furthermore, its content could be read. This additional reading requires accessorily time and becomes critical if the tag is detected relatively late before it leaves the interrogation area again. Beyond that, the available time for the anti-collision part diminishes.

The simulation was run with a fixed setting: The path consisted of 60 tags, had a length of 2 m, and the width as well as the diameter of the reader were set to 0.30 m.

Figure 8 shows the results of the speed simulation. In a first run it was required to detect all tags in range. In the second run the range of the reader was expanded to 0.24 m, which meant that about 100 tags were in range, but the same 60 tags should be detected. By the expansion the algorithms now had to deal with tags that were not the target of the detection and disturbed the anti-collision process (Figure 9).

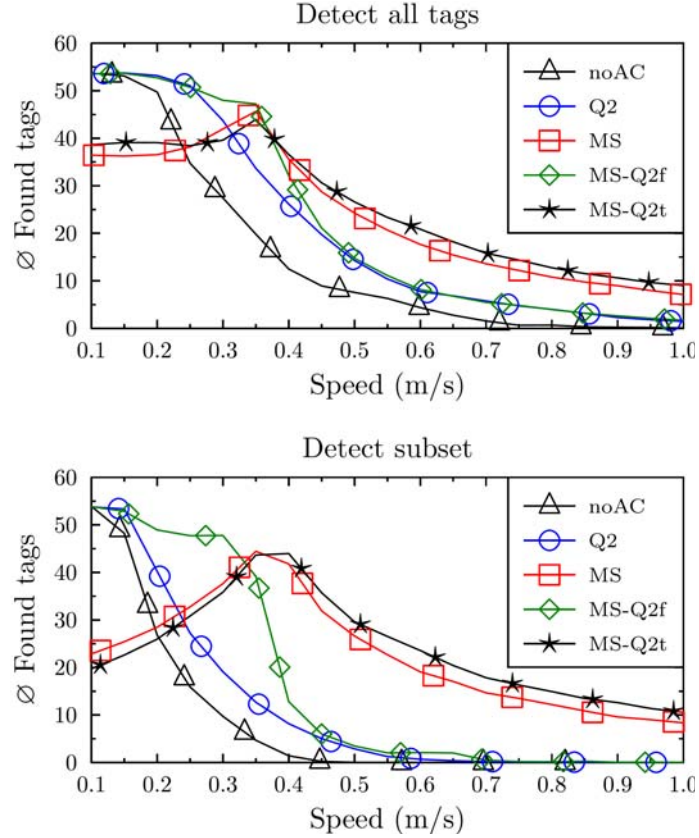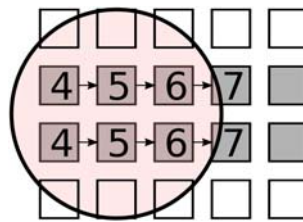**Figure 8** Detection of tags along a path (see online version for colours)

Figure 9 The reader covers more tags than necessary for the task of path following. The additional tags disturb the performance of the anti-collision algorithms that do a complete inventory (noAC, Q2). MS does not need to detect these tags (see online version for colours)

The first remarkable observation is that the MS algorithm has a global maximum at an optimal speed. This has a priori been expected. At this speed, the case in which the identifying mask 'is not valid' (see Section 5.2.1) does not occur. As expected, Q2 performs better than noAC, but for higher speeds, MS is superior.

The comparison of the different mask storing algorithms shows that the Q2f variant has a better performance for low speeds and a worse performance for higher speeds. MS-Q2t behaves similar to MS but has slight advantages.

If the mask storing algorithm were allowed to use data on the distance between tags and from odometry, it would be possible to improve the performance of Q2t for low speeds by switching the collision handling to Q2f.

It can be observed that the mask storing algorithms MS and MS-Q2t do not suffer much from the modification of the reader range for higher speeds, while noAC and Q2 show worse performances. Again, MS-Q2f has better results for low speeds but behaves much worse for higher speeds.

For MS, the usage of a bigger range results in longer identifying masks that have to be stored.

6.3   *Influence of the range of the RFID reader*

When examining the range, there are at least two important parameters that can be varied: the range of the reader itself and the width of the path.

In the previous section, it was shown that the performance for MS does not drop much if the range of the reader is increased and the path width remains untouched. Figure 10 allows a closer analysis of this behaviour. The setting is the same as in the previous simulation runs. The range is increased from 0.15 m to 0.4 m.

**Figure 10** Speed versus range (see online version for colours)
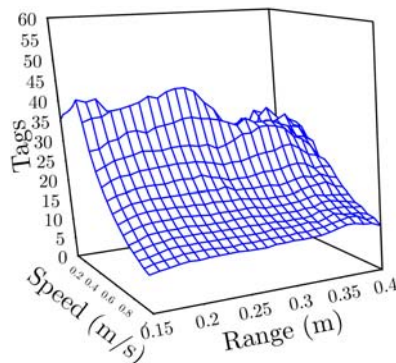
It can be observed that the performance decreases with a higher range. This is caused by the higher number of tags that have to be dealt with when the path is lost and a complete inventory has to be executed. A higher range also means that each tag remains in the interrogation area for a longer period. Therefore, the detection rate decreases only slowly.

Next, we analyse how the number of detected tags changes if we increase the range of the reader and in the same manner the path width. Figure 11 shows the result. For a higher range, more tags are detected at low and medium speeds. For higher speeds, the number of detected tags aligns. The widening of the path also increases the tag memory usage.

Figure 11 Speed versus range/path width (see online version for colours)
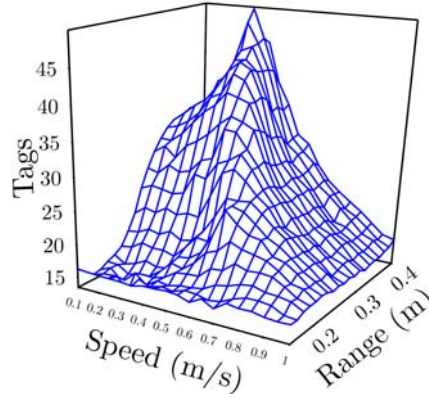
6.4 Scalability of the algorithm

The next simulation (see Figure 12) used a setting where range of reader, tag distance and path length have been scaled synchronously. This means that a constant number of tags are potentially in range.

**Figure 12** Speed versus range/density (see online version for colours)
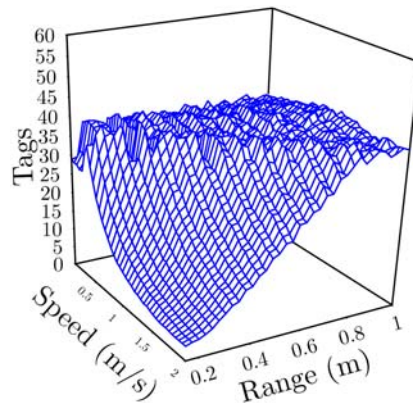
The figure shows that with scaling the reader range and the tag distance, the speed where most tags become detected also increases. But it should also be noticed that the number of tags per length unit decreases since the found tags are detected on the scaled course.

6.5 Interaction of speed, range and density

Increasing the range of the reader and increasing the tag density are similar operations but have different consequences. Both of them lead to more tags within the interrogation area. Increasing the range also results in a longer time in which the reader is able to detect the tags. In contrast, increasing the range puts higher demands on the anti-collision algorithms since more tags have to be detected within the same time-frame.

In Figure 13 the effects of changing the range and the density are compared directly. The density is expressed as the distance between the tags running from 1.0 m to 0.1 m. Also, the range is selected from 0.1 m to 1.0 m. For each distance/range pair the percentage of tags for speeds between 0.1 m/s and 7.0 m/s has been determined and averaged. This value is plotted on the *x*-axis of the first image. The second figure is taken from the same simulation, but here the *x*-axis shows the absolute number of found tags per metre averaged over the speed.

Figure 13 Range versus density (see online version for colours)
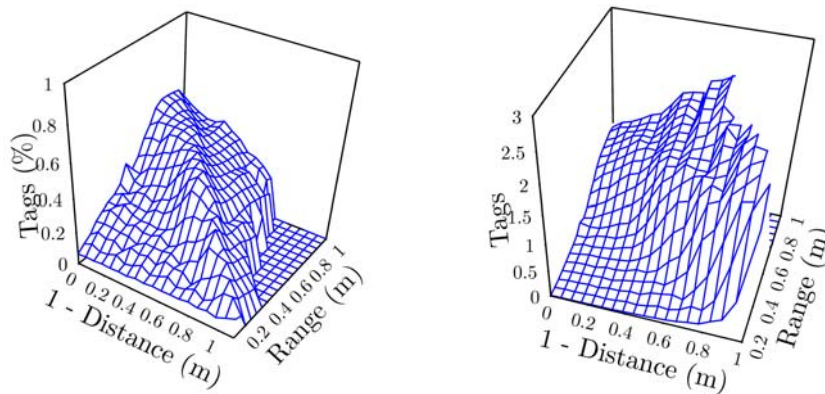

For long ranges and high density, many tags are in range of the reader. In these cases no evaluation is possible since the reserved tag memory of our simulation has exceeded. Therefore, the *x*-value is set to zero.

It can be observed that the highest percentage can be achieved for a low tag density and a high range. But it must be kept in mind that a low tag density also means that fewer tags per distance are available, even if the detection rate is higher. This can be observed from the second figure. For high density and high range, most tags can be detected, although the detection rate is low.

6.6   Robustness

Next, we checked how MS behaves under failures. For this purpose we randomly removed 5%, 10%, 20% and 30% of all tags. The remaining conditions were the same as described in Section 6.2. Removing tags is the hardest challenge that MS has to face because this could lead to gaps within the path. In contrast to other algorithms, MS uses the information of prior tags, which is also the strength of this algorithm.

Figure 14 shows the result of this simulation. The number of found tags decreases since fewer tags are available. For higher speeds, no substantial difference can be observed. Nevertheless, removing many tags results in gaps within the path. Therefore, for low speeds the maximum number of found tags decreases stronger than the number of absent tags would suggest.

Figure 14 Failure of tags (see online version for colours)
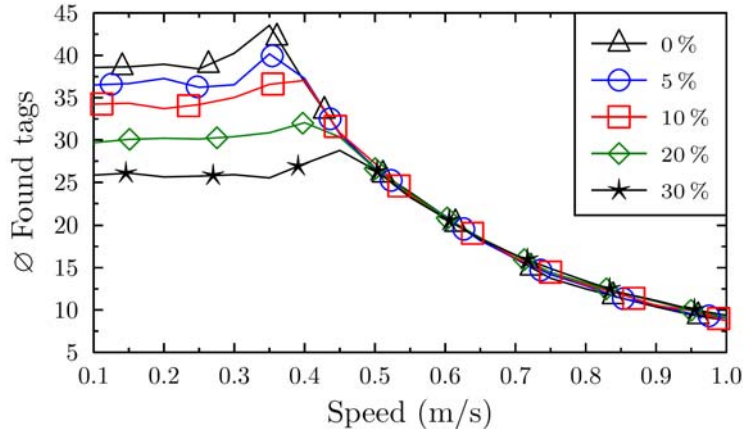
6.7 Discussion of the results

Concerning the goal of detecting as many tags as possible for high speeds, two possibilities have to be distinguished. On the one hand, it is possible to try to detect a high absolute number of tags; on the other hand it is possible to optimise the conditions to reach a high percentage of detectable tags and hence a high effectiveness of tag usage. What is required depends on the application.

The usage of stored masks for an anti-collision algorithm has two main advantages. First of all the inventory speeds up because it is no longer necessary to perform a time consuming search through the whole ID tree. Second, fewer tags have to become detected because the stored masks distinguish directly at the start between wanted and unwanted tags. This can be best observed in the speed simulation. It shows that the developed algorithm has advantages in contrast to other known anti-collision algorithms. These advantages increase if we change the requirement to that effect that a subset of tags must be detected.

Increasing the path width (and adapting the range of the reader) significantly leads to more tags for lower speed but drops after the optimal speed is reached.

Therefore, this should not be applied if a more constant behaviour for different speeds is required.

By scaling the dimensions the result is also scaled. When range, density and path length are increased, the number of detected tags grows linearly. This is a very promising result because it allows the transformation of a smaller model into a real application.

For a given range, the best performance regarding the detection rate is achieved if the tag distance is set to a similar value. If the effectiveness of tag usage is not the prime objective, most tags can be detected for a high density and a high reader range. But this also leads to high memory usage in case the mask storing approach is applied.

The conclusion can be drawn that the range and the density only have a limited influence on the detection rate and that only the combination of both parameters allows a notable influence on this. To allow higher moving speeds, it is necessary to scale all dimensions.

The test for robustness delivers the interesting observation that the algorithm is not susceptible to failures for higher speeds. On the other hand, it shows that the upper limit of detectable tags is reached and that the number of tags in range does not have an influence on the result any more.

For most simulations the mask storing approach has been used. But it should also be possible to adapt the simulation results for the examination of the influence of range and density on other anti-collision algorithms.

7   Related work

In common applications, e.g. the shopping cart, an RFID reader is required to detect simultaneously all tags in range. However, the next inventory deals with other tags. This is the scenario most anti-collision algorithms are developed for. Consequently, also RFID-based navigation systems use this standard behaviour to detect tags, even though these algorithms are not well suited for this kind of application. In a first step all tags in range are detected, the contents of each tag are read and in the end it is decided for each tag whether it belongs to the path (Bosien et al., 2008).

There are several approaches to speed up the inventory. Although they are developed with another scenario in mind, they provide useful hints for further acceleration of the inventory. Most approaches try to reduce the steps that it takes the anti-collision algorithm to find an identifying mask. While this is not necessary for MS when the next masks are available, it is of interest when a complete inventory has to be executed. Furthermore, with the Q2 algorithm we have already made use of an accelerated algorithm.

The following general approaches are notable.

- *Response analysis*: Kim et al. (2008) and Liu et al. (2005) apply a more sophisticated analysis on the response to each request. They examine under which conditions it is possible to identify tags even when collisions have occurred.

- *Tag distribution*: If it is possible to make assumptions about the distribution of the tags IDs, Koh et al. (2008) show a feasible approach that is applied to detect tags IDs which form a balanced tree. The distribution of the IDs is also interesting for MS since this has a direct influence on the length of the identifying masks and, therefore, on the required tag memory.

- *Adaptation of query*: Hsu et al. (2008) guessed the number of involved tags in order to adapt an optimal query for fast detection. The procedure is similar to the Q2 algorithm. But in contrast, the length of the mask extension is adapted to the number of remaining tags during the ongoing inventory. This makes sense if many tags are supposed to be detected.

- *Repeated inventories*: The reuse of found tags IDs can be used efficiently if only few tags change between two inventory runs (Myung and Lee, 2005). In this case it is possible to check directly if the previously found tags reappeared. This approach is related to ours, since it also uses known masks to shorten the anti-collision process.

8 Conclusion

In this paper, we presented a new algorithm that implements the idea of mask storing, which is intended for the use in path following applications and provides its foundation. The algorithm has been simulated and compared to different anti-collision algorithms. By this means we examined under which conditions we are able to accomplish our goal of detecting as many tags as possible for high speeds.

It has been proven that the developed algorithm has advantages in contrast to other known anti-collision algorithms and is suitable for application where RFID tags are stationary and related to each other, as this applies to RFID-based navigation systems. Although we used the ISO-15693 standard as a basis for our considerations and simulations, the approach is also applicable on other systems.

Acknowledgements

References

Baum, M., Niemann, B. and Overmeyer, L. (2007) 'Passive 13.56 MHz RFID transponders for vehicle navigation and lane guidance', *Proceedings of the 1st International EURASIP Workshop on RFID Technology*, Wien, Österreich, pp.83–86.

Bohn, J. and Mattern, F. (2004) 'Super-distributed RFID tag infrastructures', *Proceeding of the 2nd European Symposium on Ambient Intelligence (EUSAI 2004)*, Springer-Verlag, Eindhoven, The Netherlands, pp.1–12.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999) *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Inc., New York, NY, USA.

Bosien, A. and Turau, V. (2009) 'RFID for mobile applications', *Proceedings of the Workshop on Self-Organising Wireless Sensor and Communication Networks*, pp.27–33.

Bosien, A., Venzke, M. and Turau, V. (2008) 'A rewritable RFID environment for AGV navigation', *Proceeding of the 5th International Workshop on Intelligent Transportation (WIT'08)*, Hamburg, Germany.

Finkenzeller, K. (2003) *RFID Handbook*, 2nd ed., John Wiley & Sons, Ltd., New York.

Fleisch, E. and Mattern, F. (Eds) (2005) *Das Internet der Dinge*, Springer, Berlin, Heidelberg.

Hsu, C-H., Yu, C-H., Huang, Y-P. and Ha, K-J. (2008) 'An enhanced query tree (EQT) protocol for memoryless tag anti-collision in RFID systems', *Second International Conference on Future Generation Communication and Networking*, Vol. 1, 13–15 December, pp.427–432.

ISO/IEC (2000) *ISO/IEC FCD 15693-3, Identification cards – Contactless integrated circuit(s) cards – Vicinity cards Part 3: Anti-collision and transmission protocol*.

Kim, S., Kim, Y., Lee, S. and Ahn, K. (2008) 'An improved anti collision algorithm using parity bit in RFID system', *Seventh IEEE International Symposium on Network Computing and Applications*, 10–12 July 2008, IEEE, Cambridge, MA, pp.224–227.

Koh, H., Yun, S. and Kim, H. (2008) 'Sidewalk: a RFID tag anti-collision algorithm exploiting sequential arrangements of tags', *Proceedings of IEEE International Conference on Communication (ICC)*, pp.2597–2601.

Liu, L., Xie, Z., Xi, J. and Lai, S. (2005) 'An improved anti-collision algorithm in RFID system', *Second International Conference on Mobile Technology, Applications and Systems*, 15–17 November, IEEE, Guangzhou, p.5

Mamei, M. and Zambonelli, F. (2005) 'Physical deployment of digital pheromones through RFID technology', *Proceedings of Autonomous Agents and Multiagent Systems (AAMAS)*, ACM Press, New York, pp.1353–1354.

Mamei, M. and Zambonelli, F. (2007) 'Pervasive pheromone-based interaction with RFID tags', *ACM Transactions on Autonomous and Adaptive Systems*, Vol. 2, No. 2, p.4.

Metro, AG (2006) 'Der Dynamik des globalen Handels begegnen', *RFID-Newsletter 01/2006*, pp.8–9.

Myung, J. and Lee, W. (2005) 'An adaptive memoryless tag anti-collision protocol for RFID networks', *24th IEEE Annual IEEE Conference on Computer Communications (INFOCOM-2005)*, Poster Session, Miami, Florida.

Niemann, B., Baum, M., Overmeyer, L. and Fricke, D-H. (2006) 'Aufbau von fahrerlosen Transportsystemen (FTS) durch eine dezentrale Datenstruktur', *Logistics Journal*, Vol. 2006, pp.1–13.

Peck, M.E. (2008) 'RFID tags guide the blind', *IEEE Spectrum.* Available online at: http://spectrum.ieee.org/biomedical/devices/rfid-tags-guide-the-blind

Raoui, Y., Goller, M., Devy, M., Kerscher, T., Zollner, J., Dillmann, R. and Coustou, A. (2009) 'RFID-based topological and metrical self-localization in a structured environment', *International Conference on Advanced Robotics (ICAR 2009)*, 22–26 June 2009, pp.1–6.

Roussos, G. and Kostakos, V. (2009) 'RFID in pervasive computing: state-of-the-art and outlook', *Pervasive and Mobile Computing*, Vol. 5, No. 1, pp.110–131.

Texas Instruments (2000) 'RFID tracks containers throughout Europe', *RFID News*, p.10.