Multipacket Reception of Passive UHF RFID Tags: A Communication Theoretic Approach

Nielsen, Karsten Fyhn; Melchior Jacobsen, Rasmus; Popovski, Petar; Scaglione, Anna ; Larsen, Torben

Published in:
I E E E Transactions on Signal Processing

DOI (link to publication from Publisher):
10.1109/TSP.2011.2159499

Publication date:
2011

Document Version
Accepted author manuscript, peer reviewed version

Multipacket reception of passive UHF RFID tags: a communication theoretic approach

Karsten Fyhn, *Member, IEEE,* Rasmus M. Jacobsen, *Member, IEEE,* Petar Popovski, *Senior Member, IEEE,* Anna Scaglione, *Fellow, IEEE,* and Torben Larsen, *Senior Member, IEEE*

Abstract

This work develops a communication theoretic model for the design and analysis at the physical layer of a reader receiver structure for passive UHF RFID. The objective is attaining multi-packet reception capabilities which in turn help the fast resolution of multiple tags through a more rapid and power efficient arbitration of the tags collisions. In particular, we derive a parametric continuous time model for the subspace of a tag signal at the noisy receiver/reader, which in addition to being affected by fading and receiver delay, exhibits wide variations in the symbol frequency and transmission delay, due to imperfections in the RFID hardware design. Our main contribution is in showing that channel fading, the difference in delay and the tags frequency dispersion can be transformed from foes to friends by exploiting them in a multipacket receiver. In fact, signals colliding from different tags are more easily separable by estimating the sensor specific variation in frequency and delay and using these estimates in a multiuser receiver. In our study, we specifically consider a successive interference cancellation algorithm followed by a maximum likelihood sequence decoder, that iteratively reconstructs one signal contribution at a time and then removes it from the received signal.

Numerical simulations show that the estimates and proposed algorithm are effective in recovering collisions. The proposed algorithm is then incorporated into a numerical simulation of the Q–protocol for UHF RFID tags and is shown to be effective in providing fast and power efficient arbitration.

Index Terms

The authors e–mails are: {kfn,raller,petarp,tl}@es.aau.dk, and ascaglione@ucdavis.edu. The first two authors are students at the Aalborg University elite–programme, http://eliteeducation.aau.dk.

RFID, physical collision recovery, reader receivers, multiuser decoding

I. Introduction

In the current standard for UHF Radio Frequency IDentification (RFID), the protocol imposes a simple tag–to–reader communication to allow for a simple tag structure [1]. Collisions occur at the reader when multiple tags simultaneously reply to a query sent from a reader. To combat this, a range of anti–collision, or arbitration protocols have been designed to ensure that eventually, during the arbitration, all tags are queried individually. Since tags need to be cheaply produced, they modulate the backscattered signal with large variations, explicitly allowed in the EPCglobal UHF Class–1 Generation–2 (EPC Gen2) standard. The two parameters that vary across tags are the link frequency and the time of reply, see Fig. 1. The most significant difference is in the tag reply symbol frequency, for which the tolerance limits, as defined in the EPC Gen2 standard [1], can vary up to $\pm 22\%$ per message from the *nominal* link frequency, denoted as the backscatter link frequency (BLF). The arrival times of tag responses also vary in a range as large as 24μs, for some BLFs, which corresponds to the duration of several encoded symbols; this discrepancy is not caused by propagation effects, but rather by differences in the response of the tag, which is also assumed to be variable to minimize the synchronization demands for the tag.

Currently, readers are assumed to be equipped with a coherent receiver structure to *cope with the side-effects* of these variations [2]. In this work we propose, instead, to leverage on them, considering the attenuation, frequency and delay differences as the enabling features that allow to resolve multiple tag signals received concurrently. To do so, we introduce a communication theoretic framework, in which such variations across tags are accurately captured in the receiver observation model. We argue that the production tolerances can be seen as a *cheap form of CDMA* encoding, that facilitates the separation of the signal contributions.

There is a vast literature on wireless multipacket reception (see e.g. [3], [4], [5], [6]) which documents its potential benefits on wireless networks stable throughput, but the typical assumption is that there are multiple active communication transmitters. In the specific context of passive transmitters most of the papers have investigated the problem of tag population estimation, see e.g. [7], [8], [9], [10]; some of these works also suggest to exploit the difference in the RFID parameters to detect the number of tags in a collision. However, these papers do not provide algorithms to decode multiple tag replies.

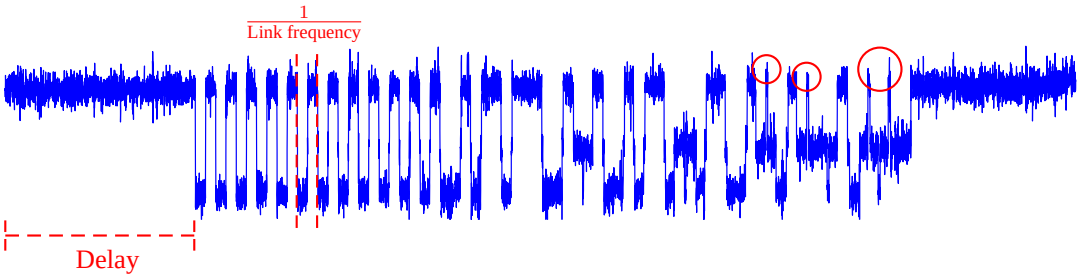


Fig. 1: Signal level of a collided tag signal, with two participating tags, as measured by a reader. The nominal link frequency (BLF) is 44.44kHz, which is the reason for the small delay difference. The tags are synchronized to begin with, but differ later in the communication, as shown by the red circles.

Multi-tag signal decoding is investigated specifically in [11] and [12]. In [11] the authors show how to decode up to four LF tags using joint detection on the I/Q components of the signal. However, they assume that a centralized, reader–controlled link frequency exists and, in light of the discussion above, it is clear that this is not a valid assumption for UHF tags. In [12] the authors propose a method to decode up to two UHF tags replies, using zero forcing and successive interference cancellation. Neither work uses a maximum likelihood sequence detector (MLSD) to improve the decoding. Other authors have suggested using multiple antennas for the separation of multiple tags [13], [14], [15]. Our model is based on a single antenna system, but can yield improved performance if generalized to a multi-antenna receiver setting.

To the best of our knowledge, our work is the first to propose a method for resolving at the physical layer more than two concurrent replies from UHF RFID tags by exploiting the intrinsic diversification of the tag parameters.

Compared to the prior art, we adopt the classic communication theoretic methodology of defining the communication signal space, deriving first the detailed continuous time parametric model of the received signal from the digital noisy link between tag and reader, and then elaborating our technique on the proposed model. The mathematical representation of the UHF RFID (see Section II) is novel, and provides the basis for the frequency and delay estimation (derived in Section III), for MLSD and subsequent interference cancellation technique, (shown in Section IV). To verify the usefulness of the receiver design we include the multipacket reception capability in the so called Q–protocol from the EPC Gen2 standard in Section V, which is used for arbitration of tags.

The Q–protocol is a slotted ALOHA–based protocol, meaning that a reader splits up the time domain into slots, in which the tags are then asked to respond. The tags choose one of the possible slots at

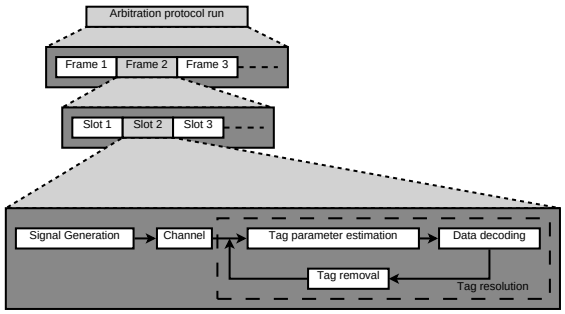


Fig. 2: An overview of our implementation and where it is applied in the Q–protocol.

random, without using any carrier sensing, and transmits there. A slot is therefore classified into one of three categories: Single if only one tag responds, Collision if more than one tag responds or Idle if no tags respond. In the present implementation of the Q–protocol only if a slot is Single can the contained tag be resolved.

A visualization of our implementation and its place in the Q–protocol is shown in Fig. 2.

We show in Section VI that by using our work to resolve some of the Collision slots, we can attain a significant gain compared to using the Q–protocol without multi-packet reception. Note that the ideas presented here are applicable beyond UHF RFID, to a wide range of scenarios with cheap, passive, clock-less tags and sensors.

II. Signal and Channel Model

This section describes the mathematical framework we have derived for representing tag signals and the channel model we employ to simulate their transmission over the air. We begin with the derivation of basis functions and the signal space representation of UHF RFID tag to reader communication, which is based on either FM0 or Miller encoding [1]. As tag to reader communication is based on backscattering [16] of a carrier wave, the tag transmission signal should be seen as a *control signal*, specifying whether a tag backscatters the carrier wave or not.

An example of the control signal for the short preamble in FM0 encoding is shown in Fig. 3. This example is used in the remainder of this section to explain the signal encoding. We first define the basis functions and signal waveforms used to generate each individual symbol. Then we describe the state machine that generates sequences of FM0 and Miller symbols.

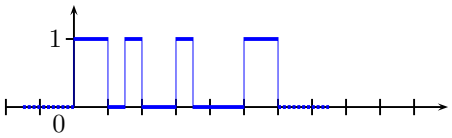


Fig. 3: FM0 preamble control signal with $TRext = 0$ (short preamble). The bit sequence is $\{1, 0, 1, 0, v, 1\}$, where the v is a symbol breaking the encoding (more on this later). The ticks on the x-axis denote a symbol duration.

A. Basis Functions for FM0 and Miller Encoding

Let $\mathcal{M}_p = \{m_0, m_1, \ldots, m_{N_\mathcal{M}-1}\}, m_n \in \{0, 1\}$ be the data message backscattered by tag p after a reader query, not including pre– and postamble. This message is the reply message in a slot during arbitration with the *Q*–protocol. It contains a 16–bit random number (RN16), and so $N_\mathcal{M} = 16$. To transmit this message, a tag first encodes it using FM0 or Miller codes, and then the signal is backscattered to the reader, through the channel. The FM0 and Miller basis functions are not rigorously defined in [1], but the signal waveforms for the respective encoding schemes are specified.

Let $\phi_k^T(t)$, $k = 0, 1$ be basis functions having *support duration* MT. That is $\phi_k^T(t) = 0$ for $t < 0$ and $t > MT$, where M is a symbol period multiplier. For FM0, $M = 1$, and the basis functions are:

$$\phi_0^{FM0,T}(t) = \frac{1}{\sqrt{T}} \left\{ \text{rect}\left(\frac{t - \frac{T}{4}}{\frac{T}{2}} \right) - \text{rect}\left(\frac{t - \frac{3T}{4}}{\frac{T}{2}} \right) \right\} \qquad \phi_1^{FM0,T}(t) = \frac{1}{\sqrt{T}} \text{rect}\left(\frac{t - \frac{T}{2}}{T} \right), \quad (1)$$

where the uses of $\text{rect}(\cdot)$ are scaled so the bases have unit energy[1]. The tag signal is generated here with ideal on–off keying. Noise and the hardware limitations do not allow pulse shaping with an instantaneous transition, however the difference is considered negligible, as in e. g. [11], [17]. For Miller, $M = 2, 4, 8$,

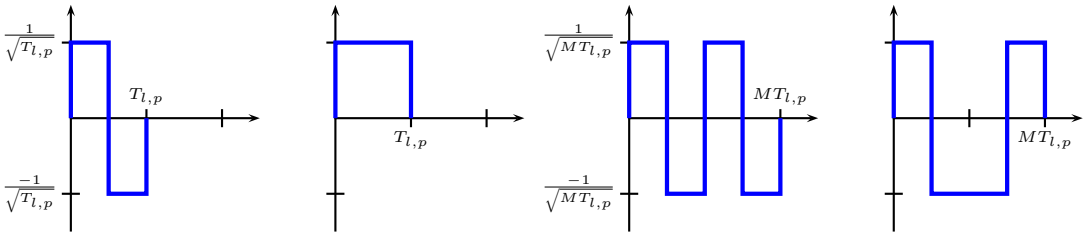
[1]Notice that the basis functions do *not* have zero mean, i.e. they correlate highly with the readers carrier wave echo. However, because the tag signal has overall zero mean, as is shown later, this is not a problem.

corresponding to the number of subcarrier cycles in the basis function:

$$\phi_0^{Miller,M,T}(t) = \frac{1}{\sqrt{MT}} \sum_{j=0}^{M-1} \left[\text{rect}\left(\frac{t - \left(j + \frac{1}{4}\right) T}{\frac{T}{2}} \right) - \text{rect}\left(\frac{t - \left(j + \frac{3}{4}\right) T}{\frac{T}{2}} \right) \right],$$

$$\phi_1^{Miller,M,T}(t) = \frac{1}{\sqrt{MT}} \left\{ \sum_{j=0}^{\frac{M}{2}-1} \left[\text{rect}\left(\frac{t - \left(j + \frac{1}{4}\right) T}{\frac{T}{2}} \right) - \text{rect}\left(\frac{t - \left(j + \frac{3}{4}\right) T}{\frac{T}{2}} \right) \right] \right.$$
$$\left. - \sum_{j=\frac{M}{2}}^{M-1} \left[\text{rect}\left(\frac{t - \left(j + \frac{1}{4}\right) T}{\frac{T}{2}} \right) - \text{rect}\left(\frac{t - \left(j + \frac{3}{4}\right) T}{\frac{T}{2}} \right) \right] \right\}. \quad (2)$$

The basis functions are depicted in Fig. 4, when evaluated for the *tag dependent subcarrier period* defined from the tag dependent link frequency as $T_{l,p} = \frac{1}{f_{l,p}}$, where $f_{l,p}$ is the link frequency for tag p where the allowed frequency variation is included.



(a) $\phi_0^{FM0,T_{l,p}}(t)$. (b) $\phi_1^{FM0,T_{l,p}}(t)$. (c) $\phi_0^{Miller,M=2,T_{l,p}}(t)$. (d) $\phi_1^{Miller,M=2,T_{l,p}}(t)$.

Fig. 4: Basis functions for FM0 ($M = 1$) and Miller with $M = 2$.

B. Possible Signal Waveforms using Basis Functions

RFID tags modulate the received waveform by alternating the control signal between two states: 0 (OFF) and 1 (ON). In the OFF state the tag absorbs the power it receives and in the ON state it reflects it back. The control signal is generated in two steps:

1) Signal waveforms are derived from the encoding dependent basis functions in Eqns. 1 and 2 with signal levels $\pm\frac{1}{2}$.
2) A constant offset of $\frac{1}{2}$ is added to the encoded message to create the control signal.

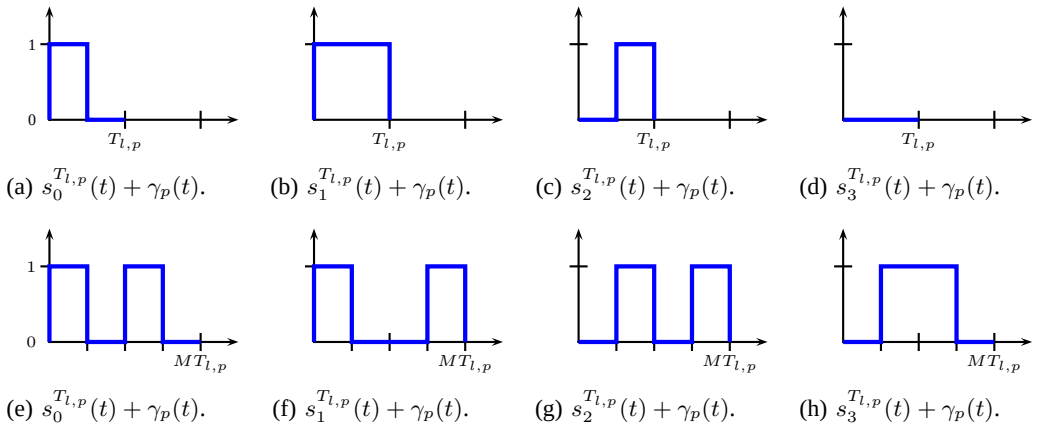
In the first step, a set of signal waveforms are found as a linear combination of the basis functions. This is accomplished using the following signal space representation:

$$\mathbf{V}_{\mathcal{E}} = \frac{\sqrt{MT_{l,p}}}{2}\mathbf{V}, \qquad \mathbf{V} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \tag{3}$$

This matrix is used in later sections as a dictionary of possible signals. Using this signal space representation, the possible signal waveforms are generated as:

$$s_i^{T_{l,p}}(t) = \sum_{j=0}^{1} v_{j,i}\phi_j^{T_{l,p}}(t), \tag{4}$$

where $v_{j,i}$ picks out an element from $\mathbf{V}_{\mathcal{E}}$ in Eqn. (3), $\phi_j^{T_{l,p}}(t)$, $j = 0, 1$ are the encoding dependent basis functions and where the signal level is $-\frac{1}{2}$ when the tag absorbs and $\frac{1}{2}$ when it reflects. Adding the signal level offset $\frac{1}{2}$ to these signal waveforms in their support duration gives the control waveforms illustrated in Fig. 5.



(a) $s_0^{T_{l,p}}(t) + \gamma_p(t)$. (b) $s_1^{T_{l,p}}(t) + \gamma_p(t)$. (c) $s_2^{T_{l,p}}(t) + \gamma_p(t)$. (d) $s_3^{T_{l,p}}(t) + \gamma_p(t)$.

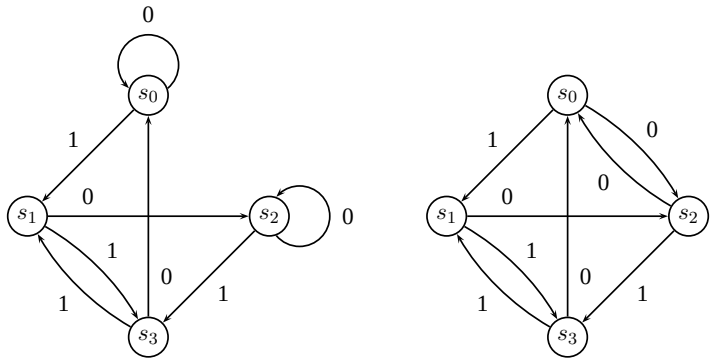
(e) $s_0^{T_{l,p}}(t) + \gamma_p(t)$. (f) $s_1^{T_{l,p}}(t) + \gamma_p(t)$. (g) $s_2^{T_{l,p}}(t) + \gamma_p(t)$. (h) $s_3^{T_{l,p}}(t) + \gamma_p(t)$.

Fig. 5: Control waveforms for FM0 (top) and Miller with $M = 2$ (bottom) where $s_0(t)$ and $s_2(t)$ encodes symbol–0 and $s_1(t)$ and $s_3(t)$ symbol–1. $\gamma_p(t) = \frac{1}{2}\,\mathrm{rect}\left(\frac{t-\frac{MT_{l,p}}{2}}{MT_{l,p}}\right)$ is the offset added in the support duration.

The control waveforms now allow us to generate single symbols. The following describes how symbol sequences are generated using the memory in the encoding schemes. We exploit this memory later in the decoding, which significantly improves the decoding.

C. Generating the Control Signal using the Inherent Encoding Memory

An important property for FM0 and Miller encoding is the inherent memory of the data encoding, i.e. the signal waveforms used for encoding of the symbol m_n depends on the previously sent symbol m_{n-1}. Let the signal waveforms $s_k^{T_{l,p}}(t)$ correspond to *state* s_k, then the state machine for FM0 and Miller is in Fig. 6, from which we obtain the symbol–dependent transition matrices $\mathbf{H}_{m_n}$, $m_n = \{0, 1\}$:



(a) FM0. (b) Miller.

Fig. 6: State diagrams for FM0 and Miller encoding. A 0 and 1 indicates the symbol sent for the transition to take place, and s_k indicates the state representing the signal waveform $s_k^{T_{l,p}}(t)$ used to encode a symbol.

$$\mathbf{H}_{m_n=0}^{FM0} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{H}_{m_n=1}^{FM0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\mathbf{H}_{m_n=0}^{Miller} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{H}_{m_n=1}^{Miller} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

where the (k, k')th entry equal to 1 indicates a valid transition from state $s_{k'-1}$ to state s_{k-1}. Also, let $\mathbf{S}_{\mathcal{M}_p}$ be a $4 \times N_{\mathcal{M}}$ state select matrix generated using $\mathbf{H}_0$, $\mathbf{H}_1$ and the message $\mathcal{M}_p$. Each column vector $\mathbf{s}_{\mathcal{M}_p,n}$ in $\mathbf{S}_{\mathcal{M}_p}$ is one of the coordinate column vectors $\mathbf{e}_k$, where $k = 1, 2, 3, 4$ denotes the state s_0, s_1,

s_2, or s_3, respectively, used to encode the nth symbol in $\mathcal{M}_p$:

$$\mathbf{S}_{\mathcal{M}_p} = \begin{bmatrix} \mathbf{s}_{\mathcal{M}_p,0} & \mathbf{s}_{\mathcal{M}_p,1} & \cdots & \mathbf{s}_{\mathcal{M}_p,N_{\mathcal{M}}-1} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{m_0}\mathbf{s}_{\text{init}} & \mathbf{H}_{m_1}\mathbf{s}_0 & \cdots & \mathbf{H}_{m_{N_{\mathcal{M}}-1}}\mathbf{s}_{N_{\mathcal{M}}-2} \end{bmatrix}, \quad (5)$$

where $\mathbf{s}_{\text{init}}$ denotes the state prior to the first symbol in $\mathcal{M}_p$. This state follows from the last symbol in the preamble.

An example is the state select matrix used to generate the signal in Fig. 3:

$$\mathbf{S}_{\mathcal{M}_p} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

From EPC Gen2 it is known that the state for the last transmitted symbol in the preamble is s_1 for FM0 and s_3 for Miller, and the respective initialization vectors are:

$$\mathbf{s}_{\text{init}}^{FM0} = \mathbf{e}_2, \qquad \text{and} \qquad \mathbf{s}_{\text{init}}^{Miller} = \mathbf{e}_4. \quad (6)$$

The control signal waveform describing the message part for tag p directly follows as:

$$\begin{aligned} c_{\mathcal{M}_p}(t) &= \sum_{n=0}^{N_{\mathcal{M}}-1} \sum_{k=0}^{3} \mathbf{e}_{k+1}^{\intercal} \mathbf{S}_{\mathcal{M}_p} \mathbf{e}_{n+1} s_k^{T_{l,p}}(t - nMT_{l,p}) + \gamma_p(t), \\ &= \sum_{n=0}^{N_{\mathcal{M}}-1} \sum_{k=0}^{1} \mathbf{e}_{k+1}^{\intercal} \mathbf{S}_{\mathcal{M}_p} \mathbf{e}_{n+1} \mathbf{V}_{\mathcal{E}} \phi_k^{T_{l,p}}(t - nMT_{l,p}) + \gamma_p(t), \end{aligned} \quad (7)$$

where $D_{\mathcal{M}_p} = T_{l,p} M N_{\mathcal{M}}$ is the duration of the data message, and $\gamma_p(t) = \frac{1}{2} \operatorname{rect}\left(\frac{t - \frac{D_{\mathcal{M}_p}}{2}}{D_{\mathcal{M}_p}} \right)$ adds the offset ensuring that the control signal has signal levels 0 or 1.

Pre– and postamble control signals are added to the message control signal, where the preamble depends on whether FM0 or Miller is used for encoding. Let $c_{pr,p}(t)$ be the preamble control signal generated with tag link frequency $f_{l,p}$, and let $D_{pr,p}$ be the support duration of the preamble. Also, let $c_{po,p}(t)$ be the postamble control signal, with support duration equivalent to the duration of one symbol. The complete transmitted signal from tag p is:

$$c_p(t) = c_{pr,p}(t) + c_{\mathcal{M},p}(t - D_{pr,p}) + c_{po,p}\left(t - D_{pr,p} - D_{\mathcal{M}_p}\right).$$

In the following sections we use an alternative representation of the above equation, where the message and the postamble symbol are included in a combined structure. It is known from EPC Gen2 that the postamble symbol is symbol–1 which corresponds to state s_1 or s_3 depending on the state of the last encoded symbol in $\mathcal{M}_p$. Let the state select matrix $\mathbf{S}_p$ be $4 \times (N_{\mathcal{M}} + 1)$ where the last entry is for the postamble symbol:

$$\mathbf{S}_p = \begin{bmatrix} \mathbf{S}_{\mathcal{M}_p} & \mathbf{H}_1 \mathbf{s}_{N_{\mathcal{M}}-1} \end{bmatrix}$$

which follows from Eqn. (5). We then have the following definition of a tag signal, rewritten as an extended version of Eqn. (7):

$$c_p(t) = c_{pr,p}(t) + \sum_{n=0}^{N_{\mathcal{M}}} \sum_{k=0}^{3} \mathbf{e}_{k+1}^{\intercal} \mathbf{S}_p \mathbf{e}_{n+1} s_k^{T_{l,p}}(t - D_{pr,p} - nMT_{l,p}) + \gamma_p(t), \tag{8}$$

where the support duration of $\gamma_p(t)$ is increased to include the postamble symbol $\gamma_p(t) = \frac{1}{2} \operatorname{rect}\left(\frac{t - D_{pr,p} - \frac{T_{l,p} M (N_{\mathcal{M}}+1)}{2}}{T_{l,p} M (N_{\mathcal{M}}+1)} \right)$.

D. Channel Model and Received Signal

Let $y_p(t)$ be the signal corresponding to a single tag reply where the effect of the channel between tag p and the reader is captured. We assume a linear time–invariant (LTI) channel, i.e. the tags do not move during the communication and the channel is assumed static during one reading of the tag population. As RFID is a narrowband communication system, this also means the entire channel is more likely to be coherent. Therefore, assuming a LTI channel with flat fading during the short period of communication:

$$y_p(t) = H_{RTR,p} \; A \; T_b \; c_p(t), \tag{9}$$

where $c_p(t)$ is the on–off key modulated square wave control signal for tag p from Eqn. (8), T_b is the fraction of the power, which the tag is able to backscatter [18], and A is the amplitude of the transmitted carrier wave from the reader. $H_{RTR,p}$ is the complex channel coefficient $H_{RTR,p} = H_{RT,p}^2 = H_{TR,p}^2$, which models that the channel coefficient between the reader and tag p, $H_{RT,p}$, is the same as the channel between tag p and the reader, $H_{TR,p}$, due to reciprocity. $H_{RTR,p}$ captures fading, antenna gains, and path–loss. Then the received signal at the reader is:

$$z'(t) = \sum_{p=0}^{P-1} y_p(t - \tau_p) + L + O(t), \tag{10}$$

where P is the number of tags that participate in the response, τ_p is the unknown random delay for tag p, L is the leakage from the reader's transmit antenna and the scatteres of the unmodulated carrier wave, and $O(t)$ is additive white Gaussian noise (AWGN) added at the reader. More specifically, the antenna leakage can be decomposed as $L = H_{RR}A + L_{ant}$, where L_{ant} is the antenna leakage, and H_{RR} is the complex channel coefficient for the reader–to–reader channel. We model $H_{RR,p}$ with Rayleigh fading because the line of sight component is captured by the direct antenna leakage L_{ant}.

Note that $y_p(t)$ in Eqn. (9) has infinite bandwidth because of the on–off keyed control signal $c_p(t)$. However all other components in $y_p(t)$ are low–pass. To model the effect of the receive filter at the reader, we introduce an ideal low–pass filter at the reader side, defined as $h_l(t) = 2W\,\mathrm{sinc}(2Wt)$, where W is the positive bandwidth of the low–pass equivalent signal. Additionally, all tag replies are amplitude modulated, thus only the envelope of the received signal contributes to the information in $z'(t)$. This envelope detection is done in some RFID reader systems, but rarely [18], as more advanced readers are able to extract the complex envelope of the signal, preserving the linearity of the overall link. In our case, we use the envelope detection because it is simple, but future work should investigate other, more advanced, receiver structures. The received low–pass envelope on the reader is:

$$\begin{aligned} z(t) &= \left| \int_{-\infty}^{\infty} z'(\tau) h_l(t-\tau) d\tau \right| \\ &= \left| \int_{-\infty}^{\infty} \left[\sum_{p=0}^{P-1} H_{RTR,p} A T_b c_p(\tau - \tau_p) + H_{RR}A + L_{ant} + O(\tau) \right] h_l(t-\tau) d\tau \right| \end{aligned} \tag{11}$$

III. Parameter Set Estimation

After having developed a detailed model for a tag signal and of its output through the communication channel, the next step is to derive the structure for the signal parameters estimation. Specifically, the estimation module described in this section estimates the link frequency, $T_{l,p}$ and delay, τ_p, of the strongest tag in the received signal, as defined in Eqn. (11).

The information in a tag reply is encoded using the tag dependent control signal $c_p(t)$, which is true for all tags in the reply. For estimating the two parameters, link frequency and delay, it is important to have a–priori known information about the structure of the tag replies in $z(t)$. The estimation procedure is designed to exploit the fact that in a reply *all* tags, independent of the link frequency and delay chosen by tag p, *use the same structure in the preamble control signal* $c_{pr,p}(t)$ to control the absorb and reflect

state during backscatter of the preamble in a reply. The structure in the preamble is the key used in the estimation framework introduced, where a *mother function*, $\psi(t)$, representing the preamble structure is designed. Then, a number of derivatives of this mother function are created, denoted as *daughter functions*, $\psi_{a,b}(t)$, which are *scaled* (a) and *translated* (b) versions of the mother function. This framework is similar to that used in wavelet signal processing, see e.g. [19]. However, a wavelet transform is an orthogonal transform and is used to decompose a signal, where our framework is non-orthogonal and is used only as a correlation framework to identify which signals are most likely present in the received signal.

A daughter function is defined as:

$$\psi_{a,b}(t) = \psi\left(\frac{t-b}{a}\right) \tag{12}$$

Each daughter function is *correlated* with the received signal $z(t)$ and the largest magnitude is used to estimate the frequency and delay of the strongest tag in the incoming signal. This approach is motivated by the fact that the mean of the received preamble signal may be approximated by:

$$E[z(t)] = \alpha\psi\left(\frac{t-b}{a}\right) + \beta, \qquad b < t \leq b + D_{pr}, \tag{13}$$

where α is an estimate of the signal level, β is the offset added to remove the zero mean property of the mother function (more about this later), and D_{pr} is the duration of the preamble. The expectancy operation averages across the white noise.

The correlation framework is defined using the received signal and a daughter function:

$$T(a,b) = \langle z(t), \psi_{a,b}(t)\rangle = \int_{-\infty}^{\infty} z(t)\psi_{a,b}(t)dt, \tag{14}$$

Calculation of $T(a,b)$ for a range of a and b results in a three–dimensional representation, where a measure of the correlation of the received signal $z(t)$ with various daughter functions are given. Similarly, if the scalogram $E(a,b) = T^2(a,b)$ is considered, then:

$$(a_p, b_p) = \arg \max_{a\in\mathcal{A},\ b\in\mathcal{B}} E(a,b) \tag{15}$$

is the pair telling that it is very likely that tag p, with link frequency $f_{l,p} = \frac{1}{a_p}$, delayed b_p is present in $z(t)$. We use scalogram since if the channel has incurred a phase shift, the received signal has a large negative peak in the correlation representation. The search ranges for a and b, $\mathcal{A}$ and $\mathcal{B}$, respectively, are

defined in the standard and depend on the settings of the reader.

In an implementation, the cardinality of $\mathcal{A}$ and $\mathcal{B}$ must of course be finite, meaning that we must generate the scalogram according to some grid. But the tags choose the two parameters, link frequency and delay, from a continuous distribution. This therefore results in scenarios where two tags are different in one or both parameters, but because of the chosen grid size, they appear identical in the scalogram. This fact, and the fact that neither tag signals nor daughter functions are orthogonal to each other, means that the more tags participating in a response, the harder it is to decode several of them.

The objective is then to explicitly evaluate Eqn. (15). The mother function is designed to capture the preamble structure in Section III-A, and the scaling and translation of the mother function leads to the definition of the daughter function in Section III-B.

A. Mother Function

Let $\psi(t)$ be the real valued mother function that satisfies the following two requirements:

- $\psi(t)$ must have finite energy, i.e. $\int_{-\infty}^{\infty} \psi^2(t)dt < \infty$. This ensures that the correlation is bounded in time.
- $\psi(t)$ must have no zero–frequency (DC) component in its support duration, i.e. it must be zero mean. Thereby the function is able to differentiate between signals based on their structure rather than their signal level.

Furthermore, the mother function is designed such that each bit in a symbol has duration $\frac{1}{2}$ seconds ensuring that the link frequency of the mother function is 1Hz in the support duration, and so each symbol in the preamble has unit energy. Thanks to this normalization the link frequency of a daughter function (which is designed in the following section) becomes $1/a$ when evaluated with the scaling parameter a. The preamble structure consists of linear combinations of the basis functions derived in Section II, and the signal waveforms with unit energy are:

$$\begin{aligned} s_0^{T=1}(t) &= \phi_0^{T=1}(t), & s_1^{T=1}(t) &= \phi_1^{T=1}(t), \\ s_2^{T=1}(t) &= -\phi_0^{T=1}(t), & s_3^{T=1}(t) &= -\phi_1^{T=1}(t). \end{aligned} \tag{16}$$

The mother function depends on whether FM0 or Miller is used. Additionally, in the query sent by the reader, the parameter $TRext$ specifies, which of two different preambles to use for a given encoding

when a tag replies. Let N_{pr} be the number of symbols in a preamble, and a $4 \times N_{pr}$ state select matrix $\mathbf{S}_{pr}$ is generated in the same way as in Section II from the preamble structures in EPC Gen2 as:

$$\mathbf{S}_{pr}^{FM0,TRext=0} = \begin{bmatrix} \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_4 & \mathbf{e}_1 & \mathbf{e}_4 & \mathbf{e}_2 \end{bmatrix}$$
$$\mathbf{S}_{pr}^{FM0,TRext=1} = \begin{bmatrix} \underbrace{\mathbf{e}_1 \quad \mathbf{e}_1 \quad \cdots \quad \mathbf{e}_1}_{12} & \mathbf{S}_{pr}^{FM0,TRext=0} \end{bmatrix}$$

$$\mathbf{S}_{pr}^{Miller,TRext=0} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_1 & \mathbf{e}_1 & \mathbf{e}_1 & \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_4 & \mathbf{e}_2 & \mathbf{e}_4 \end{bmatrix}$$
$$\mathbf{S}_{pr}^{Miller,TRext=1} = \begin{bmatrix} \underbrace{\mathbf{e}_1 \quad \mathbf{e}_1 \quad \cdots \quad \mathbf{e}_1}_{16} & \mathbf{S}_{pr}^{Miller,TRext=0} \end{bmatrix},$$

where $\mathbf{e}_k$ indicates the state s_{k-1}, or equivalently that signal waveform $s_{k-1}(t)$ in Eqn. (16) is used to generate the mother function. With this in mind, let the square–wave modulated mother function be:

$$\psi(t) = \sum_{n=0}^{N_{pr}-1} \sum_{k=0}^{3} \mathbf{e}_{k+1}^{\intercal} \mathbf{S}_{pr} \mathbf{e}_{n+1} s_k^{T=1}(t - nM) = \sum_{n=0}^{N_{pr}-1} \sum_{k=0}^{1} \mathbf{e}_{k+1}^{\intercal} \mathbf{V} \mathbf{S}_{pr} \mathbf{e}_{n+1} \phi_k^{T=1}(t - nM), \qquad (17)$$

where $\mathbf{V}$ is the signal constellation matrix from Eqn. (3). Notice that the preamble structure has zero mean and that the inherent memory structure of both FM0 and Miller encoding is violated in the preambles. This ensures that it is not possible for the designed function to correlate as strongly to the data as to the preamble.

A final consideration on the design is that $z(t)$ contains the low-pass filtered signal. However, since each daughter function is recalculated for each value of a and b, to contain the computational complexity we consider the daughter functions square-wave modulated.

B. Daughter Function

A daughter function of the mother function is defined as:

$$\psi_{a,b}(t) = w(a)\psi\left(\frac{t-b}{a}\right), \qquad (18)$$

where the scaling and translation parameters are used in the evaluation of the mother function, and where $w(a)$ is a weight, which ensures that all configurations of a daughter function are equally weighted and not biased by the parameter set (a, b) when matched onto the input signal in the correlation $\langle z(t), \psi_{a,b}(t) \rangle$. The amplitude level for a tag when a bit is backscattered is the same during communication no matter what link frequency is used. Thus the energy in a tag preamble oscillating with a fast frequency is less than the energy in a tag preamble oscillating with a slow frequency. The match constraint to determine $w(a)$ therefore takes into account $z(t)$.

Lemma 1: (Proved in the Appendix) The weight ensuring that daughter functions are correctly scaled for all values of a and b in a reply containing only one tag reply is:

$$w(a) = \frac{1}{a}. \tag{19}$$

IV. Data Decoding

The estimates of the link frequency $\frac{1}{a}$ and the delay offset b obtained using the framework presented above makes it possible to decode one and possibly several tag replies in a received signal, even when they are dispersed in time and frequency. This corresponds to the tag resolution part shown in Fig. 2, which is the subject treated in this section.

One shortcoming of using UHF RFID as a use case for general multiple sensor decoding is that EPC Gen2 does not currently support multiple Ack commands after a collided reply, which clearly affects the possible gain in data decoding. This is not included in the current standard, as it has no use for it and its introduction would require careful system design to avoid deadlocks, etc. Our method would work even with the current standard, but would then only be able to provide optimized decoding of a single tag, by estimating its parameters and then decoding its message. When acknowledging that tag, the other participating tags would return to their arbitrate state and be unresolvable in that slot. In the description of the data decoding algorithm that follows, we assume that multiple Ack commands are implemented in the standard, to show the true potential of our method.

To decode multiple tag replies an iterative, greedy algorithm, Successive Interference Cancellation (SIC), is used, as opposed to the more complex joint detection. Before explaining the SIC algorithm further, the next section derives the optimum decoder structure for a single tag, using the above described estimators.

A. Optimized Single Tag Decoding

The optimal algorithm for a detector, where the memory in a sequence satisfies the Markov property as the nth symbol in a sequence *only* depends on the $(n-1)$th decoded symbol, is the Viterbi algorithm [20]. In addition to the memory structure in the encoding scheme, there are two additional a–priori known structures in EPC Gen2 to improve the decoding:

1) The last symbol in the preamble and the signal waveform used to create it is a–priori known and are previously found in Eqn. (6).
2) The postamble symbol after the data part is a–priori known to be symbol–1.

Recall from Section II-C that the structure containing *both* the memory, the data message $\mathcal{M}$ and the postamble symbol is the 4–by–$(N_{\mathcal{M}}+1)$ state select matrix $\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_{N_{\mathcal{M}}+1} \end{bmatrix}$, where each $\mathbf{s}_i$ is a coordinate vector $\mathbf{e}_k$ and $k = 1, 2, 3, 4$ indicates which of the respective signal waveforms $s_0(t)$, $s_1(t)$, $s_2(t)$, and $s_3(t)$ is used to encode a symbol in the tag reply. The objective is therefore reformulated to estimate the state select matrix $\mathbf{S}$ as it contains the memory structure, encoded message, and postamble.

Let the signal processed in the ith iteration of the SIC algorithm be $r_i(t)$, with $r_0(t) = z(t)$, and assume that it only contains one tag reply with parameters a and b. The expected value of $r_i(t)$ is then similar to the control signal $c_p(t)$ in Eqn. (8):

$$E[r_i(t)] = \alpha_i \Big(\psi^1_{a_i,b_i}(t) + \sum_{n=0}^{N_{\mathcal{M}}} \sum_{k=0}^{3} \mathbf{e}_{k+1}^{\intercal} \mathbf{S} \mathbf{e}_{n+1} s_k^{a_i} \left(t - b_i - D_{pr,i} - n a_i M\right) + \gamma_i(t) \Big) + \beta, \qquad (20)$$

where α_i is the signal level at the reader side. Notice here that the daughter function has a superscript 1 attached. This is because this daughter function must be scaled to have *unit energy per symbol*, rather than to have correct scaling for all values of a and b in a reply containing only one tag signal. This results in changing the weight in Eqn. (19) from $\frac{1}{a}$ to $\frac{1}{\sqrt{aM}}$. This change in scaling allows for the correct scaling afterwards to the signal level, α_i. $\gamma_i(t)$ is the offset introduced to ensure signal levels that correspond to the way a tag backscatters its reply (recall Eqn. (7)), β is an estimate of the reader leak and $D_{pr,i} = a_i M N_{pr}$ is the estimated duration of the preamble. The signal waveforms:

$$s_0^{a_i}(t) = \phi_0^{a_i}(t) \qquad s_1^{a_i}(t) = \phi_1^{a_i}(t) \qquad s_2^{a_i}(t) = -\phi_0^{a_i}(t) \qquad s_3^{a_i}(t) = -\phi_1^{a_i}(t), \qquad (21)$$

follow from the symbol basis functions in Eqn. (1) and Eqn. (2) and can be represented in terms of the signal space representation matrix $\mathbf{V}$ as:

$$s_k^{a_i}(t) = \sum_{j=0}^{1} \mathbf{e}_{j+1}^{\mathsf{T}} \mathbf{V} \mathbf{e}_{k+1} \phi_j^{a_i}(t). \tag{22}$$

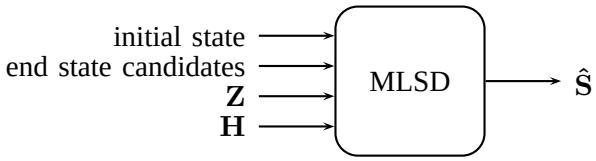


Fig. 7: Black box illustration of the MLSD. A correlation matrix $\mathbf{Z}$ and a transition matrix $\mathbf{H}$ are used with the information of the initial state and possible end states to estimate the most likely transmitted state select matrix $\hat{\mathbf{S}}$.

Let the outcome of the MLSD be an estimate of $\mathbf{S}$, denoted $\hat{\mathbf{S}}$, and consider the Viterbi algorithm as the black box for MLSD in Fig. 7; for optimal decoding it uses the three data structures:

- Initial state vector $\mathbf{s}_{\text{init}}$ — The initial state seeding the decoding which follows from the state corresponding to the last symbol in the preamble. For FM0, this state is s_1 and for Miller it is s_3, and the vector is defined in Eqn. (6).
- A $4 \times (N_{\mathcal{M}} + 1)$ matrix $\mathbf{Z}$ — Cost matrix not considering memory, where each element represents how well each of the four signal waveforms $\tilde{s}_k(t)$, $k = 0, 1, 2, 3$, match a singled out symbol part in the residual, where a large element value indicates a good match. Values are found for the $N_{\mathcal{M}} = 16$ data symbols and the postamble symbol.
- A 4×4 matrix $\mathbf{H}$ — Memory structure representing the allowed paths between states observed for two adjacent encoded symbols.

The memory structure matrix $\mathbf{H}$ is given by the valid state transitions. In the generation of the control signal at the tag in Section II-C, the two matrices $\mathbf{H}_{m_n=0}$ and $\mathbf{H}_{m_n=1}$ are derived describing the transition to make, conditioned on the symbol m_n to send. $\mathbf{H}$ follows as the version where the transmitted symbol is unknown:

$$\mathbf{H}^{FM0} = \mathbf{H}_0^{FM0} + \mathbf{H}_1^{FM0} \qquad \mathbf{H}^{Miller} = \mathbf{H}_0^{Miller} + \mathbf{H}_1^{Miller}$$

and $\mathbf{H}_{k,k'} = 1$ indicates that it is possible to go from state $s_{k'-1}$ to state s_{k-1}.

The two matrices $\mathbf{Z}$ and $\mathbf{H}$ can be represented as a trellis, where each node (except the node in the preamble) represent the entries of $\mathbf{Z}$ and the possible transitions are given by the entries in $\mathbf{H}$. The reason why there are only two possible states for the first symbol is that the previous state is known a–priori. The same applies for the postamble as it is known to be a symbol–1.

It is useful to remark that, if the channel for the tag to be decoded in the ith iteration incurs a complete phase shift on the backscattered reply in the residual, this means that the state of the last symbol in the preamble is s_3 instead of s_1 for FM0, and s_1 instead of s_3 for Miller. The phase shift can be detected in several ways; for example if $\alpha_i < 0$ or $T_i(a_i, b_i) < 0$ a phase shift is introduced. In the sequel this effect is neglected to simplify the description of the decoding, however, it is important for an implementation to detect the phase shift and flip the initializing state in the decoder.

B. Successive Interference Cancellation for Data Decoding

The SIC algorithm has originally been used for multiuser detection in CDMA systems [21], [22], [23] and is a general, iterative framework for multiuser detection, where a user is singled out and removed in each iteration, treating the remaining users as interference in the signal. The algorithm is often coupled with e.g. least-squares or matching pursuit [23], which is used to find the strongest signal component in each iteration. The ith iteration of the algorithm is defined as follows:

$$r_{i+1}(t) = r_i(t) - q(t),$$

where $r_i(t)$ is the current residual signal, $q(t)$ is an estimate of the strongest signal component in $r_i(t)$ and $r_{i+1}(t)$ is the resulting residual, used in the next iteration.

The estimate of the ith tag signal, which shall be subtracted from the residual, follow from Eqn. (25). To model the transmitted message signal $\hat{q}_i(t)$, we use the decoded message $\hat{\mathbf{S}}$:

$$\hat{q}_i(t) = \alpha_i \Big(\psi^1_{a_i,b_i}(t) + \sum_{n=0}^{N_{\mathcal{M}}} \sum_{k=0}^{3} \mathbf{e}^{\intercal}_{k+1} \hat{\mathbf{S}} \mathbf{e}_{n+1} \cdot s_k^{a_i} \left(t - b_i - D_{pr,i} - n a_i M\right) + \gamma_i(t) \Big), \tag{23}$$

where the ith modulation depth estimate α_i follow from Lemma 2 in the Appendices and $\gamma_i(t) = w^1(a_i) \operatorname{rect}\left(\frac{t - b_i - \frac{D_{pr,i} + a_i(N_{\mathcal{M}}+1)}{2}}{D_{pr,i} + a_i(N_{\mathcal{M}}+1)} \right)$. The weight, $w^1(a_i)$, is to ensure unit energy per symbol of the signal before scaling with α_i.

If the signals being estimated and removed are orthogonal, the SIC algorithm would correspond simply

to projecting a signal down into each component subspace for decoding. This is however not the case in this scenario, where the signal are non-orthogonal and the SIC must therefore instead estimate and remove them one at a time. This means that when the amplitude coefficient α_i is estimated, it may be influenced by other tag signals in the residual. This degrades the performance of the decoding, but is an inherent property of tag signals, differentiated by parameters chosen from a continuous distribution.

The algorithm is summarized as follows: Let $\mathcal{H} = \{v_0, v_1, \ldots\}, v_i = (a_i, b_i)$ be the history of estimates, assume $z(t)$ contains only one tag reply, set $r_0(t) = z(t)$ and let $i = 0$, then:

1) Find the ith scalogram, as in Eqn. (14), and determine whether a phase shift has occurred.
2) Find estimates for the strongest contribution in the newly found scalogram:

$$v_i = (a_i, b_i) = \underset{\substack{a \in \mathcal{A}, \\ b \in \mathcal{B}}}{\arg\max}\, E_i(a, b)$$

3) Decode a message from $r_i(t)$ with location parameters (a_i, b_i) using the Viterbi algorithm described in the previous section.
4) Generate an estimate of the complete tag contribution $\hat{q}_i(t)$ from Eqn. (23).
5) Subtract the estimate from the residual $r_{i+1}(t) = r_i(t) - \hat{q}_i(t)$.
6) Let $\mathcal{H} = \{\mathcal{H}, v_i\}$ if $\mathcal{H} \cap v_i = \{\}$.
7) Based on a termination criteria, decide whether to break or increment i and re–iterate. This termination criteria may be e.g. the saturation of the residual $r_{i+1}(t)$, a fixed number of iterations based on the probability of decoding any remaining tags, see Fig. 8 or perhaps a threshold decision on the scalogram in the next iteration. It is tempting to use a simple digital termination technique, where the reader attempts to decode and request acknowledgement from tags, until it receives no more acknowledgements, However, this is inefficient because sending an acknowledgement request is very expensive, as is shown in Table I. Because the termination criteria has a large impact on the resolution efficiency, it should be analyzed thoroughly to find the best solution, which is outside the scope of this paper. Therefore, in the numerical simulations we assume that the reader knows when a signal is either idle or contains no more tag signals. This is done for both the classical reader and the multipacket reception reader.

It is interesting to make a short analysis of the complexity of this algorithm. The novelty in this work is in the creation of the scalogram, which may be described as follows: Let the signal under analysis

Reader		
Name	**Contents other than payload**	**No. of bits in payload**
Query	delimiter, data-0, RTcal, TRcal	22
QueryRep	delimiter, data-0, RTcal	5
Ack	delimiter, data-0, RTcal	18
Tag		
Name	**Contents other than payload**	**No. of bits in payload**
RN16	preamble, postamble	16
PC/XPC + EPC + CRC	preamble, postamble	128

TABLE I: Transmitted reader and tag commands [1].

consist of N samples and let the cardinality of the ranges $\mathcal{A}$ and $\mathcal{B}$ be K, then the computation of the scalogram consists of K^2N multiplications and $K^2(N-1)$ additions. These computations may be done using the FFT, which would greatly increase the efficiency. This must be analyzed further, but as this paper focuses on the information theoretic possibilities of the idea, rather than their implementation, this is not treated further here.

V. Implementing Multiple Tag Decoding in the Q–Protocol

With the algorithm concluding the previous section, it is possible to decode a single slot shown in Fig. 2. This section describes how this is extended to become part of an entire arbitration protocol run, using the Q–protocol of EPC Gen2. It is useful to evaluate the effect multiple tag decoding has on the Q-protocol, if acknowledging multiple tags is allowed. To enable this, the Q-protocol is implemented in MATLAB and Monte Carlo simulations are run, to determine how much *time* it takes to resolve an entire tag population and how many *transmissions* it takes. The results can be used to evaluate the following (1) Is multiple tag decoding in the Q–protocol more time efficient than single tag decoding? (2) Is multiple tag decoding more energy efficient, with respect to transmissions count from the reader?

In the numerical simulations the transmissions by both reader and tag are counted as listed in Table I. Based on the duration of each of these commands and the three timeouts T_1, T_2 and T_3 from the standard, the duration of the inventorying can be calculated.

Design Assumptions for Q–Protocol Implementation

Prior to an experiment, we assume a Select command has been issued and received correctly by all tags. This defines the scope of the experiment to inventorying of tags alone. Additionally, all tags have

their inventoried flag for the selected session set to the same value, either A or B. This means that out of N tags, N tags participate in the inventorying.

For simplicity the command QueryAdjust is not used *during* a round to change the value of Q. Instead, a round is always completed with a chosen Q after which a new Query command is issued with a new value of Q. A QueryAdjust can increment or decrement Q during an inventory round, and when to use it during arbitration must be analyzed thoroughly first, to understand its effect on time and energy usage. We have therefore not optimized the Q–protocol for multiple tag decoding and we expect that a higher gain is achievable if this is done.

It is assumed that the reader can determine perfectly whether a slot is idle or not and whether a slot contains any remaining tag signals. As mentioned in the description of the SIC algorithm in the previous section, this is of course not possible, but we do this to focus our results on tag decoding, not on detecting whether there are tags to be decoded. Our results therefore provide an upper bound on the performance with respect to this parameter. In a future implementation, this detection could instead be done based on the saturation of the residual. As the variance of the noise can be estimated before tag to reader communication, it can be decided whether one or more tags are present, if enough samples cross a detection threshold based on the variance during tag to reader communication. This threshold may also be used for detecting when the residual in the SIC algorithm contains no more tags. In the case of UHF RFID tags an Ack transmission from a reader is quite expensive and should be avoided if possible. Otherwise, the termination criteria of the algorithm could be based on a *digital decision*, where the SIC algorithm terminates if no sensor replies the Ack.

Because each iteration of the SIC algorithm is dependent on the previous iterations and the accuracy of the estimates of a, b and the signal level, the estimation of the signal level is assumed perfect in this implementation, to focus exclusively on the impact of multiple tag decoding based on estimation of link frequency, a, and delay, b. Also, the reader is assumed to be unable of detecting a collision, it will always attempt to decode the strongest tag. If a tag is correctly singled out, the Ack and EPC are correctly transferred and decoded. This, to allow for simplicity and because if a tag RN16 has already been successfully singled out, the probability of error in receiving the Ack and transmitting the EPC is smaller. Additionally, we assume that the forward link (reader-to-tag) is error free, to be able to focus fully on multiple tag decoding.

When a single tag is resolved, an extra frame is conducted, to ensure that no weaker tags are unresolved. For fair comparison, this is done for both the original and new reader. A change in the UHF RFID standard for the tags is assumed, namely that when a tag receives an Ack with a *wrong* RN16, it does not transition to state arbitrate, but remains in state Reply. Only when a Query or QueryRep command is received does the tag transition to state Arbitrate. This allows for multiple tag acknowledging by sending an Ack and receiving and decoding the EPC of the resolved tag in each iteration of the SIC, rather than only being able to send one Ack per slot.

VI. Results

To show the benefit of having multiple tag decoding in a reader, we have performed two simulation tests. Both tests are performed as Monte Carlo simulations to verify that the result converges to some value. This of course does not prove convergence, but shows statistical evidence indicating convergence. The first test illustrates the probability of decoding a given number of tags. The second test is a comparison of the duration of an inventory round using the Q–protocol when using a reader with and without multiple tag decoding. The tests have been made for a scenario where the tags choose their link frequency according to a Gaussian distribution with mean equal to a nominal BLF of $50\mathrm{kHz}$ and a variance such that 99.73% (3σ using the empirical rule) of the generated link frequencies and delays fall within the EPC Gen2 requirements. Also, a long preamble ($TRext = 1$), $FM0$ encoding is used, the distance between reader and tags is set to 1 meter and the low–pass filter employed has a bandwidth of $1.5\mathrm{MHz}$. The noise power at the reader antenna is set to $-50\mathrm{dBm}$.

In the first test a number of experiments and runs are performed. A *run* is defined as the generation of a received signal, as in Eqn. (11), and the following decoding of that signal. After decoding, the estimated message and the actual encoded message is compared, and if the decoding was incorrect, the run is marked as *erroneous*. An *experiment* is a series of runs. For the first test, 100 experiments, each containing 100 runs, has been performed. The results are shown in Fig. 8, where the gain for four different tag cardinalities, $P = \{2, 3, 4, 5\}$ is illustrated. Note that our proposed method resolves multiple tags at the physical layer, where these tags transmit in a single slot. With a directed antennae, it may be possible to spatially isolate 5 or less tags with high probability, however this cannot be assumed with an omnidirectional antennae. Since the correlation framework is based on non-orthogonal waveforms, at $P > 5$ the probability of decoding $P > 5$ tags in a single slot is small. On the other hand, on top of

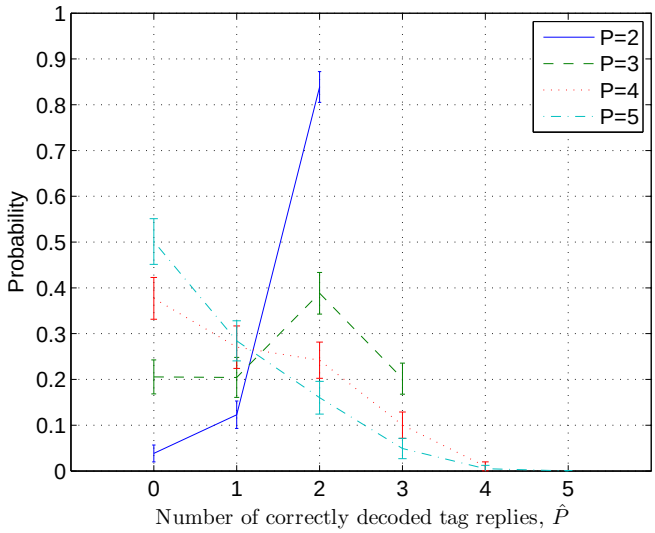


Fig. 8: Numerically simulated collision resolution success probability for multiple tag decoding. The distance is 1 meter and $P = \{2, 3, 4, 5\}$ tag replies are collided. The vertical lines represent the standard deviation at a given parameter point.

this physical layer mechanism, there is a MAC-layer mechanism, notably the Q-protocol, that randomly divides the total set of tags into smaller subsets in order to enable the collision resolution process. Hence, if the number of tags in a given collision is larger than e. g. 5, then the proposed physical layer mechanism will likely not resolve any tags and the slot needs to be considered as a collision. Therefore, for a larger tag set, the Q-protocol operates in conjunction with the proposed multi-tag detection. It can be understood that the proposed mechanism only takes effect when the collision slot contains a small number of tag signals. The result of one experiment is used to calculate the percentage of runs ending in a given number of decoded tags. The results from all the experiments are then used to calculate the standard deviation of this statistic. The figure shows that it is possible to decode multiple tags, even when there are up to five tags present in the collision. Even though the total five tags are rarely decoded, the results show that in 50% of the cases some of the tags are decoded, which is a gain compared to presently used methods. On the figure, it might look confusing that for $P = 3, 4$, the probability of resolving only one of the P tags is higher than for $P = 2$. However, it should be noted that this is because when only two tags collide, the probability of resolving both tags in one slot is very high.

In the other test, we implement the Q–protocol with multiple tag decoding and compare it to a normal reader, which only decodes one tag per slot. The initial value of Q is set to 4 and 1000 runs are conducted.

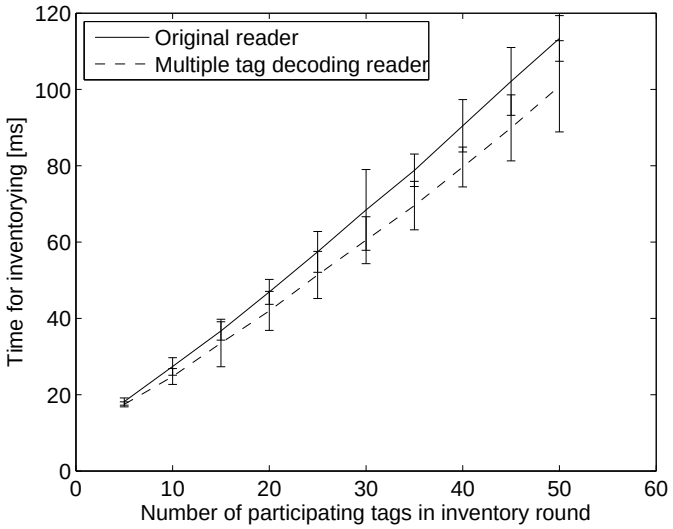


Fig. 9: The duration of the inventorying in the numerical simulation of multiple tag decoding in the Q–protocol. The vertical lines is the standard deviation at a given parameter point.

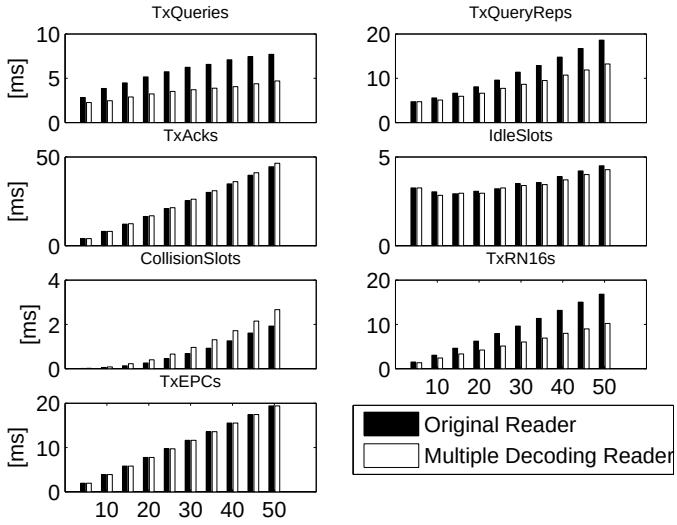


Fig. 10: The distribution of the transmitted commands during inventorying in the numerical simulation of multiple tag decoding in the Q–protocol.

In each run, a randomly generated tag set is resolved using the Q protocol. The time it takes to resolve the tag set is found, by counting the transmitted commands, as specified in the previous section. The result is averaged over the 1000 runs and plotted in Fig. 9. The distribution of the commands is further elaborated on in Fig. 10. As can be seen from Fig. 9, multiple tag decoding decreases the duration of

the inventorying, especially for a large number of tags. From Fig. 10, it is clear that with multiple tag decoding, fewer Queries and QueryReps are sent. The RN16 count must be further explained. If more than one tag transmit their RN16 at the same time, there is a collision, which is counted as one RN16, as the duration is independent of the number of participating tags in the collision. The number of RN16s sent out has also decreased dramatically, as several tags can be decoded in one slot. The number of collisions has increased when using multiple tag decoding, but by a very small amount when compared to the savings. The number of idle slots and acknowledgements are roughly the same for both single and multiple tag decoding. The number of EPC commands is exactly identical, as is expected for full resolution. The reason why the difference is not larger in Fig. 9 is that the number of the most expensive transmissions, the acknowledgement, is unchanged. If the protocol is changed to allow for acknowledging multiple tags with a single composite Ack, the performance will greatly improve. Overall, the results show that multiple tag decoding does provide savings in time and energy and this gain increases approximately linearly, meaning that for tag populations in the hundreds and thousands, this would provide a significant increase in time and power efficiency.

VII. Conclusion

The concepts presented in this paper show that tag variability can be transformed from foe to friend, by using such differences to decode multiple colliding UHF RFID tag replies. We first presented a detailed mathematical model of the tag signals using standard signal representation techniques, which, to the knowledge of the authors, has not been presented in this level of detail before for both FM0 and Miller encoding. This model may be of use in future work in this area or for other problems related to RFID at the physical layer. It may also allow for easier abstraction of these areas, by capitalizing on the more general representation presented here.

Using our own model and by utilizing the knowledge of the tag signal, we then show that it is possible to distinguish and decode individual tag signals in numerical simulations by using an estimation framework that estimates the delay and frequency of the individual tag signals, to enable better decoding. For the decoding, we use first the iterative SIC algorithm to estimate individual tag signals, which are then removed from the residual, enabling further decoding of weaker tag signals. Using these estimates and the Viterbi algorithm, we are able to decode the individual tag replies.

The final tests where multiple tag decoding is incorporated into the Q–protocol shows a potential for time and transmission savings, in terms of fewer transmitted commands from the tags.

APPENDIX A
PROOF OF WEIGHT ENSURING CORRECT SCALING OF DAUGHTER FUNCTIONS

Lemma 1: The weight ensuring that daughter functions are correctly scaled for all values of a and b in a reply containing only one tag reply is:

$$w(a) = \frac{1}{a}.$$

Proof: The following simplifications are made for the derivation:

- Only one tag is assumed to be present in the reply $z(t)$ with link frequency $\frac{1}{a}$, and the duration of the encoded preamble in the reply is D_{pr}, that is, the tag preamble contributes to $z(t)$ for $b < t < b+D_{pr}$.
- $w(a)$ is determined for the pair a, b that leads to a maximum or minimum in Eqn. (14), i. e. only the case where the duty cycle duration a for the encoded tag in $z(t)$ and the duration of the mother function MN_{pr} satisfy $D_{pr} \equiv aMN$, where M is the number of subcarrier cycles per symbol and N_{pr} is the number of symbols in the preamble.

The property to be satisfied is that a daughter function, when correlated with $z(t)$ should satisfy a parameter independent correlation level:

$$\langle E[z_1(t)], \psi_{a_1,b_1}(t)\rangle = \langle E[z_2(t)], \psi_{a_2,b_2}(t)\rangle \tag{24}$$

should be satisfied, where the tag reply in $z_j(t)$ is encoded with *different* link frequency $\frac{1}{a_j}$ and delay b_j but with the *same channel*, thus the signal levels in $z_1(t)$ and $z_2(t)$ are equal. In the interval $b < t < b+D_{pr}$ the received signal $z(t)$ has the property that its expected value can be written in terms of a weighted mother function with the same configuration as the control signal used to model $z(t)$:

$$E[z(t)] = \alpha\psi\left(\frac{t-b}{a}\right) + \beta, \qquad b < t < b + D_{pr}, \tag{25}$$

where α controls the signal level and β the DC component in $z(t)$. Rewriting, letting $t' = \frac{t-b}{a}$:

$$E[z(at'+b)] = \alpha\psi(t') + \beta, \qquad 0 < t' < MN_{pr}. \tag{26}$$

As the signal levels in $z(t)$ clearly does not depend on the encoding parameters a and b, then the correlation of Eqn. (26) with the mother function is therefore *constant for all encoded tag replies with different a and b.* That is:

$$\langle E[z_1(a_1 t + b_1)], \psi(t) \rangle = \langle E[z_2(a_2 t + b_2)], \psi(t) \rangle \tag{27}$$

is a property that is always satisfied when $z_j(\cdot)$ is encoded with parameters a_j and b_j, and thus is the property requested in Eqn. (24). As the daughter function is a scaled and translated version of the mother function, combine Eqn. (24) and Eqn. (27):

$$\int_{-\infty}^{\infty} E[z(at+b)]\psi(t)dt = \int_{-\infty}^{\infty} E[z(t)]w(a)\psi\left(\frac{t-b}{a}\right)dt = w(a)a\int_{-\infty}^{\infty} E[z(at'+b)]\psi(t')dt'$$

where $t' = \frac{t-b}{a}$ and $dt' = \frac{dt}{a}$, and the weight is $w(a) = \frac{1}{a}$ which completes the proof. ■

APPENDIX B
PROOF OF OPTIMAL ESTIMATOR OF α

Lemma 2: Aim for the lowest contribution for the tag with parameter configuration (a_i, b_i) in the scalogram evaluated in the next iteration $i+1$, then the optimal estimator for α_i is:

$$\alpha_i = \frac{\sqrt{a_i} T_i(a_i, b_i)}{\sqrt{M} N_{pr}}, \tag{28}$$

where $T_i(a_i, b_i)$ is the value from Eqn. (14) corresponding to the estimated a_i and b_i, M is the number of subcarrier cycles per symbol in the encoding scheme and N_{pr} is the number of symbols in the preamble.

Proof: The problem to be optimized is:

$$\alpha_i = \arg\min_{\alpha \in \mathbb{R}} E_{i+1}(a_i, b_i),$$

i. e. find the α_i which minimizes the contribution in iteration $(i+1)$ of the tag found in iteration i. The

scalogram as a function of the ith residual and the contribution to be removed is:

$$E_{i+1}(a_i,b_i) = \left(\int_{-\infty}^{\infty} E[r_{i+1}(t)]\psi_{a_i,b_i}(t)dt\right)^2 = \left(\int_{-\infty}^{\infty} \left[E[r_i(t)] - \hat{q}_i(t)\right]\psi_{a_i,b_i}(t)dt\right)^2$$
$$\stackrel{i}{=} \left(\int_{-\infty}^{\infty} \left[E[r_i(t)] - \alpha_i(\psi^1_{a_i,b_i}(t) + \gamma_i(t))\right]\psi_{a_i,b_i}(t)dt\right)^2$$
$$= \left(T_i(a_i,b_i) - \alpha_i\int_{-\infty}^{\infty}\psi^1_{a_i,b_i}(t)\psi_{a_i,b_i}(t)dt - \alpha_i\int_{-\infty}^{\infty}\gamma_i(t)\psi_{a_i,b_i}(t)dt\right)^2 \stackrel{ii}{=} \left(T_i(a_i,b_i) - \frac{\alpha_i\sqrt{M}N_{pr}}{\sqrt{a_i}}\right)^2$$
$$= \alpha_i^2\frac{MN_{pr}^2}{a_i} - \alpha_i\frac{2\sqrt{M}N_{pr}T_i(a_i,b_i)}{\sqrt{a_i}} + E_i(a_i,b_i), \tag{29}$$

where $i)$ follows as the support duration of $\psi_{a_i,b_i}(t)$ ensures that $\hat{q}_i(t)$ is not evaluated in the part where the extra half symbol is added to $\hat{q}_i(t)$, and where $ii)$ follows firstly because the daughter function is zero mean in the interval of the support duration of $\gamma_i(t)$ and secondly by evaluating the daughter function and the daughter function with unit energy per symbol in terms of the mother function:

$$\int_{-\infty}^{\infty}\psi^1_{a_i,b_i}(t)\psi_{a_i,b_i}(t)dt = w^1(a)w(a)\int_{-\infty}^{\infty}\psi^2\left(\frac{t-b_i}{a_i}\right)dt = \frac{1}{\sqrt{Ma_i}}\int_{-\infty}^{\infty}\psi^2(t')dt' = \frac{\sqrt{M}N_{pr}}{\sqrt{a_i}}.$$

Eqn. (29) is a quadratic function where the quadric coefficient is positive, the function is convex, and thus its minimum is where the derivative is zero:

$$\frac{dE_{i+1}(a_i,b_i)}{d\alpha_i} = \alpha_i\frac{2MN_{pr}^2}{a_i} - \frac{2\sqrt{M}N_{pr}T_i(a_i,b_i)}{\sqrt{a_i}} = 0.$$

Solving for α_i completes the proof. ■

REFERENCES

[1] *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz*, 1st ed., EPCglobal Inc., October 2008.

[2] C. Angerer and M. Rupp, "Advanced Synchronisation and Decoding in RFID Reader Receivers," in *IEEE Radio and Wireless Symposium*, San Diego, USA, 2009.

[3] L. Tong, Q. Zhao, and G. Mergen, "Multipacket Reception in Random Access Wireless Networks: From Signal Processing to Optimal Medium Access Control," *Communications Magazine, IEEE*, vol. 39, no. 11, pp. 108–112, November 2001.

[4] M. Tsatsanis, R. Zhang, and S. Banerjee, "Network-assisted Diversity for Random Access Wireless Networks," *Signal Processing, IEEE Transactions on*, vol. 48, no. 3, pp. 702–711, March 2000.

[5] R. Lin and A. Petropulu, "A New Wireless Network Medium Access Protocol Based on Cooperation," *IEEE Transactions on Signal Processing*, vol. 53, no. 12, pp. 4675–4684, December 2005.

[6] M. Tsatsanis, R. Zhang, and S. Banerjee, "Network-assisted Diversity for Random Access Wireless Networks," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 702–711, March 2000.

[7] B. Knerr, M. Holzer, C. Angerer, and M. Rupp, "Slot-wise Maximum Likelihood Estimation of the Tag Population Size in FSA Protocols," *IEEE Transactions on Communications*, vol. 58, pp. 578–585, February 2010.

[8] R. S. Khasgiwale, R. U. Adyanthaya, and D. W. Engels, "Extracting Information from Tag Collisions," in *IEEE international conference on RFID*, Orlando, USA, 2009.

[9] I. Cidon and M. Sidi, "Conflict Multiplicity Estimation and Batch Resolution Algorithms," *IEEE Transactions on Information Theory*, vol. 34, no. 1, pp. 101–110, 1988.

[10] P. Popovski, F. H. P. Fitzek, and R. Prasad, "A Class of Algorithms for Collision Resolution with Multiplicity Estimation," *Algorithmica*, vol. 49, no. 4, pp. 286–317, 2007.

[11] D. Shen, G. Woo, D. P. Reed, A. B. Lippman, and J. Wang, "Separation of Multiple Passive RFID Signals Using Software Defined Radio," in *IEEE international conference on RFID*, Orlando, USA, 2009.

[12] C. Angerer, R. Langwieser, and M. Rupp, "RFID Reader Receivers for Physical Layer Collision Recovery," *IEEE Transactions on Communications*, vol. 58, no. 12, pp. 3526–3537, December 2010.

[13] J. Yu, K. H. Liu, X. Huang, and G. Yan, "An Anti-collision Algorithm Based on Smart Antenna in RFID System," in *International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, Nanjing, China, April 2008.

[14] J. Lee, T. Kwon, Y. Choi, S. K. Das, and K.-a. Kim, "Analysis of RFID Anti-collision Algorithms using Smart Antennas," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, 2004.

[15] A. Mindikoglu and A. J. van der Veen, "Separation of Overlapping RFID Signals by Antenna Arrays," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, NV, USA, April 2008.

[16] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification.* John Wiley & Sons, West Sussex, England, 2003.

[17] C. Angerer, G. Maier, M. V. B. Delgado, M. Rupp, and J. V. Alonso, "Single Antenna Physical Layer Collision Recovery Receivers for RFID Readers," in *Proceedings of the International Conference on Industrial Electronics*, Vina del Mar, Chile, 2010.

[18] D. M. Dobkin, *The RF in RFID: Passive UHF RFID in Practice.* Newton, MA, USA: Newnes, 2007.

[19] P. S. Addison, *The Illustrated Wavelet Transform Handbook.* London, UK: IOP Publishing Ltd., 2002.

[20] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York, NY, USA: McGraw–Hill, 2008.

[21] P. Patel and J. Holtzman, "Analysis of a simple successive interference cancellation scheme in a ds/cdma system," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 5, pp. 796 –807, June 1994.

[22] R. Iltis and S. Kim, "Geometric derivation of expectation-maximization and generalized successive interference cancellation algorithms with applications to cdma channel estimation," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1367 – 1377, May 2003.

[23] S. Kim and R. Iltis, "A matching-pursuit/gsic-based algorithm for ds-cdma sparse-channel estimation," *IEEE Signal Processing Letters*, vol. 11, no. 1, pp. 12 – 15, January 2004.

Karsten Fyhn obtained both his B. Sc. in electrical engineering and M.Sc. in wireless communication from Aalborg University, Denmark in 2008 and 2010, respectively. The master degree was obtained at the ELITE-programme at Aalborg University, a special research programme for highly skilled and motivated students. He is currently a Ph.D. student at Aalborg University, working on compressive sensing in wireless communication. His interests include signal processing, multi-packet reception, power-efficient wireless communication systems and wireless communication receiver structures.

Rasmus Melchior Jacobsen is with the radio group at Kamstrup, Denmark, designing wireless mesh, multi-hop automatic metering infrastructure for low-power heat/water/gas meters. He obtained his M.Sc. at the wireless communications elite-branch at Aalborg University in 2010, where he also obtained his B.Sc. in communication systems in 2008. His interests include low-power routing protocols, network coding, models for slow topology dynamics, and receiver structures.

Petar Popovski received the Dipl.-Ing. in electrical engineering and Magister Ing. in communication engineering from Sts. Cyril and Methodius University, Skopje, Macedonia, in 1997 and 2000, respectively and Ph. D. from Aalborg University, Denmark, in 2004. He was Assistant Professor at Aalborg University from 2004 to 2009. From 2008 to 2009 he held part-time position as a wireless architect at Oticon A/S. Since 2009 he is an Associate Professor at Aalborg University. He has more than 100 publications in journals, conference proceedings and books and has more than 25 patents and patent applications. He has received the Young Elite Researcher award from the Danish Free Research Council. He has received several best paper awards from IEEE. Dr. Popovski serves on the editorial board of several journals, including IEEE Transactions on Wireless Communications and IEEE Communications Letters. His research interests are in the broad area of wireless communication, information theory and protocol design.

Anna Scaglione received the Laurea (M.Sc. degree) in 1995 and the Ph.D. degree in 1999 from the University of Rome, ”La Sapienza.” She is currently Professor in Electrical and Computer Engineering at University of California at Davis, where she joined in 2008. She was previously at Cornell University, Ithaca, NY, from 2001 where she became Associate Professor in 2006; prior to joining Cornell she was Assistant Professor in the year 2000-2001, at the University of New Mexico. She served in Editorial Board of the IEEE Transactions on Wireless Communications from 2002 to 2005, and of IEEE Transactions on Signal Processing from 2008-2011, where she was Area Editor in 2010-2011. She has also served in several IEEE conferences and special issues. She has been in the Signal Processing for Communication Committee from 2004 to 2009. She was general chair of the workshop SPAWC 2005 and keynote speaker in SPAWC 2008. Dr. Scaglione is the first author of the paper that received the 2000 IEEE Signal Processing Transactions Best Paper Award; she has also received the NSF Career Award in 2002 and she is co-recipient of the Ellersick Best Paper Award (MILCOM 2005). Her expertise is in the broad area of signal processing for communication systems and networks. Her current research focuses on RF communications, cooperative wireless networks and sensors’ systems for monitoring and control applications and energy management, and signal processing for network science.

Torben Larsen received the M.Sc. and Dr.Techn. degrees from Aalborg University, Denmark, in 1988 and 1998, respectively. Since 2001, he has been a Full Professor at Aalborg University. He has industrial experience with Bosch Telecom and Siemens Mobile Phones. He was member in 2005-2010 and vice-chairman in 2009-2010 of the Danish Research Council for Technology and Production Sciences. He has authored or co-authored more than 100 peer-reviewed journal and conference papers. He received the Spar Nord research prize in 1999 for excellent research. He has successfully supervised more than 15 PhD students. His research interests include scientific computing, compressive sensing, GPU computing, RF system modeling and simulation, wireless communication, and transceiver design.