On Number of Tags Estimation in RFID Systems

Der-Jiunn Deng, *Member, IEEE*, Chun-Cheng Lin, Tzu-Hsun Huang, and Hsu-Chun Yen

***Abstract*—According to the latest version of the radio-frequency identification (RFID) standard, EPCglobal UHF Gen2, dynamic framed slotted ALOHA has been accepted and employed as the *de facto* collision resolution algorithm to share the channel usage when multiple tags respond to the reader's signal command simultaneously. However, the frame size in read cycle is usually far from optimal when the reader does not know the number of unidentified tags in its interrogation zone, and this will lead to several performance issues and technical limitations such as power consumption, longer reading time, and degraded system performance. In this paper, we show that the number of unidentified tags can be expressed as a function of the collision rate received by the reader when the reader collects tags' information, and then we use an extended Kalman filter-based tag number estimation algorithm to estimate the number of unidentified tags in an RFID system on run-time measurements. Simulation results show that our scheme provides much better accuracy than existing well-known approaches even in a fast-changing environment.**

***Index Terms*—Dynamic framed slotted ALOHA (DFSA), EPCglobal UHF Gen2, extended Kalman filter (EKF), radio-frequency identification (RFID), tag identification.**

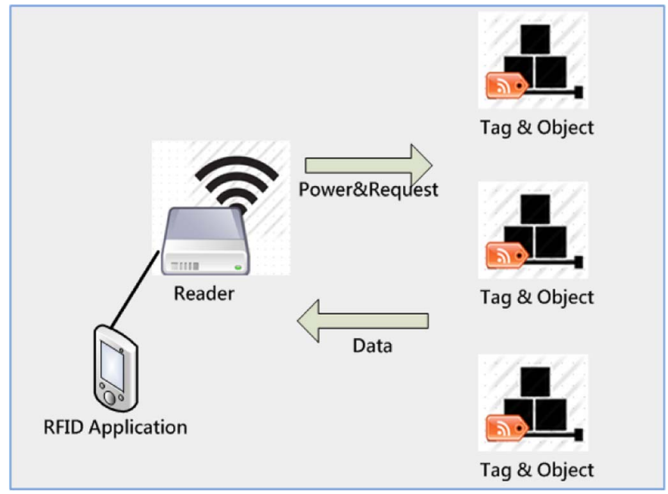

Fig. 1. Illustration of the RFID system.

I. Introduction

RADIO-FREQUENCY IDENTIFICATION (RFID) is one of the wireless telecommunication techniques for recognizing objects [1]–[3]. Nowadays, RFID systems have been broadly applied in a variety of areas, including banknote security, electronic toll collection system, contactless payment system, animal and goods tracking, and so on [4], [5].

In this paper, we consider the RFID system framework illustrated in Fig. 1, which consists of one reader and a number of tags. When tags need to be recognized, the reader sends a request to ask for data from the tags, and each tag will transmit its data to the reader. However, sometimes, collisions occur when tags transmit their data to the reader. That is, when there is more than one tag conducting uplink transmission in the same slot, collisions occur and this will lead to failed transmission.

From the literature, a variety of anticollision algorithms have been proposed. They are generally classified into tree-based algorithms [6] and ALOHA-based algorithms [7]. Tree-based algorithms repeat the procedure of clustering the collided tags to avoid collision. On the other hand, the ALOHA-based algorithm randomly selects a time slot for data transmission to avoid collisions between tags. One of the most popular anticollision algorithms for RFID systems is the dynamic framed slotted ALOHA (DFSA) algorithm [8], which groups multiple time slots into a frame, and usually, the frame length is decided according to the current number of unidentified tags. In particular, according to previous works, the performance of the DFSA algorithm is optimal when the frame size is equal to the number of unidentified tags inside the interrogation zone. However, based on our research results [9], when the frame size is equal to the number of tags, collision still occurs frequently, and this severely affects the system performance because it causes power consumption and longer tag reading time. Hence, in [9], we develop an analytical model to study the system throughput of DFSA-based RFID systems, and then we use this model to search for an optimal frame size that maximizes the system throughput based on the current number of unidentified tags.

Although now we know that we should tune the frame size based on the current number of unidentified tags [8], [9], however, the information on the precise number of unidentified tags is difficult to get. In general, if the frame length is smaller than the optimal one, collisions occur frequently; contrarily (i.e., the frame length is larger than the optimal one), idle time slots produced and system performance degraded. Therefore, correct estimation of the current number of unidentified tags is the key factor to successfully get an appropriate frame length for high system performance. In the past, there were several works along this topic, including the lower bound (LB) [10],

Manuscript received December 15, 2014; revised June 17, 2015; accepted July 30, 2015. This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 102-2221-E018-012-MY3 and Grant MOST 104-2221-E-009-134-MY2. *(Corresponding author: Hsu-Chun Yen.)*

D.-J. Deng is with the Department of Computer Science and Information Engineering, National Changhua University of Education, Changhua 500, Taiwan (e-mail: djdeng@cc.ncue.edu.tw).

C.-C. Lin is with the Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: cclin321@nctu.edu.tw).

T.-H. Huang and H.-C. Yen are with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: yen@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/JSYST.2015.2466594

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

maximum throughput (MT) [11], RFID explicit tag estimation scheme (RETES) [12], Chebyshev-based method [13], tag estimation method (TEM) [11], enhanced TEM (ETEM) [14], and accurate TEM (ATEM) [15]. The simulation results of these works showed satisfactory performance, but the accuracy of the tag number estimation still leaves room for improvement.

In this paper, a precise on-time tag number estimation method for RFID systems is proposed so that the accurate estimated number of unidentified tags can be used to tune the frame length in the DFSA algorithm for improving the performance of the system. The proposed tag number estimation method is based on the extended Kalman filter (EKF), which has been shown to have a high estimation performance compared with minimum mean-squared-error method [13], [16], [17]. The observed channel statuses of idle slot rate, success slot rate, and collision slot rate is considered in the proposed scheme to establish a more fairly accurate tag estimation model. Finally, the performance of our proposed method is verified via a detailed simulation.

The remainder of this paper is organized as follows. Section II surveys the previous works on tag number estimation algorithms, respectively, and introduces the Kalman filter. Section III describes the proposed algorithm. Section IV shows the experimental results for our proposed algorithm and compares them with some existing well-known algorithms. The conclusion and future work are presented in Section V.

II. Preliminaries

Some existing well-known popular TEMs are surveyed in this section, and then we introduce the Kalman filter. Note that the simulation results in Section IV include the comparisons between the proposed method and these methods.

A. Tag Number Estimation Methods

1) LB: In a *collision slot*, no less than two tags transmit simultaneously; in a *success slot*, only one tag transmits; in an *idle slot*, no tag transmits. The LB method [10] is proposed to estimate the number of tags in the last read cycle. Let S and C be the numbers of success and collision slots in the last read cycle, respectively. The estimated tag number $n_{\rm est}$ can be formulated as

$$n_{\rm est} = S + 2C.$$

However, the multiple two for C in this equation may not be correct because there could be more than two tags transmitted in the collision slot; hence, $n_{\rm est}$ might not be a precise estimation.

2) MT: The MT method [8] is based on the hypothesis that the frame length is equivalent to the number of unidentified tags since this is a direct way to optimize the system throughput. The MT method also supposes that the number of unidentified tags n_u could be infinite. Let $P_{\rm coll}$ be the probability that a slot is a collision slot and $P_{\rm succ}$ be the probability that a slot is a success slot. Then, the estimated collision rate $C_{\rm rate}$ is expressed as follows:

$$C_{\rm rate} = R_{\rm coll}/(1 - P_{\rm succ}).$$

Since it supposes that $n_u \to \infty$, the rate $C_{\rm rate}$ can be calculated as follows (the details are referred from [11]):

$$C_{\rm rate} = \lim_{n_u \to \infty} \frac{P_{\rm coll}}{1 - P_{\rm succ}} \cong 0.418.$$

The average number of the tags involved in a collision slot $C_{\rm tag}$ is then easily computed as follows:

$$C_{\rm tag} = 1/C_{\rm rate} \cong 2.3.922.$$

That is, the MT method estimates the number of unidentified tags $n_{\rm est}$ as follows:

$$n_{\rm est} = S + 2.3922C.$$

However, the supposed conditions in this method are too strict that some deviations would be generated if the real situation differs much from the strict conditions.

3) RETES: Different from the LB and MT methods, the RETES [12] additionally considers the number of idle slots denoted by I. The estimated tag number $n_{\rm est}$ is then formulated as follows:

$$n_{\rm est} = S + w_1 C + w_2 I$$

where w_1 and w_2 are the scaling factors for C and I, respectively. According to the study in [12], the best accuracy for $n_{\rm est}$ can be obtained with $2.0 \le w_1 \le 2.2$ and $0.1 \le w_2 \le 0.2$.

4) Chebyshev-Based Method: Chebyshev's inequality indicates that, in any probability distribution, almost all random variables are close to the expected value. Hence, the Chebyshev-based method [13] models the tag number $n_{\rm est}$ as follows:

$$n_{\rm est} = \arg\min_n \left\| \begin{pmatrix} E(I) \\ E(S) \\ E(C) \end{pmatrix} - \begin{pmatrix} I \\ S \\ C \end{pmatrix} \right\|$$

where n is the number of tags; I, S, and C represent the numbers of idle slots, success slots, and collision slots, respectively; and their expected values are $E(I)$, $E(S)$, and $E(C)$, respectively, which can be computed as follows:

$$E(I) = N\left(1 - \frac{1}{N}\right)^n$$

$$E(I) = n\left(1 - \frac{1}{N}\right)^{n-1}$$

$$E(C) = 1 - E(I) - E(S)$$

where N is the frame length.

5) TEM: Let N and n denote the frame length and the number of tags, respectively. The probability of collision happening in a slot $P_{\rm coll}$ is then formulated as follows:

$$\begin{aligned} P_{\rm coll} &= 1 - P_{\rm idle} - P_{\rm succ} \\ &= 1 - \left(1 - \frac{1}{N}\right)^n - \frac{n}{N}\left(1 - \frac{1}{N}\right)^{n-1} \\ &= 1 - \left(1 - \frac{1}{N}\right)^n \left(1 + \frac{n}{N-1}\right) \end{aligned}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

where P_{idle} is the probability that no tags transmit in a slot, and P_{succ} is the probability that one tag successfully transmits in a slot. Note that P_{coll} can be also computed by using the number of collision slots C as follows:

$$P_{\mathrm{coll}} = \frac{C}{N}.$$

Finally, the estimated tag number n in the TEM [11] can be obtained by solving the following equation:

$$\frac{C}{N} - \left(1 - \left(1 - \frac{1}{N}\right)^n \left(1 + \frac{n}{N-1}\right)\right) = 0.$$

6) ETEM: The TEM aforementioned only utilizes the probability of the collision slot when estimating tag number but does not leverage the information on idle and success slot probability. Hence, the ETEM [14] fully utilizes the aforementioned information to increase accuracy. Similar to the MT method aforementioned, the ETEM also needs to compute the average number of the tags included in a collision slot C_{tag}, but it does not suppose any condition on the tag number and the frame length, as shown as follows:

$$\begin{aligned} C_{\mathrm{tag}} &= \frac{n - P_{\mathrm{succ}}N}{P_{\mathrm{coll}}N} \\ &= \frac{n - n(1-1/N)^{n-1}}{N - N(1-1/N)^n - n(1-1/N)^{n-1}}. \end{aligned} \tag{1}$$

The number of tags n can be obtained as follows:

$$n = S + C_{\mathrm{tag}}C. \tag{2}$$

The ETEM then estimates the final results by solving the following equation from substituting (1) into (2):

$$C\frac{n - n(1-1/N)^{n-1}}{N - N(1-1/N)^n - n(1-1/N)^{n-1}} + S - n = 0.$$

7) ATEM: The ATEM [15] first denotes β as the ratio of the frame length N to the tag number n and γ as the average tag number of the collision slots, as shown as follows:

$$\beta = \frac{N}{n} \text{ and } \gamma = \frac{n-S}{C}. \tag{3}$$

In addition, as N is large enough, we have

$$(1-1/N)^n \approx (e^{-1})^{\frac{n}{N}} = e^{-\frac{n}{N}}.$$

Hence, the expected values of S and C can be employed to derive the γ value as follows:

$$\begin{aligned} \gamma &= \frac{n - E(S)}{E(C)} = \frac{n - n(1-1/N)^{n-1}}{N - N(1-1/N)^n - n(1-1/N)^{n-1}} \\ &\approx \frac{n\left(1 - e^{-\frac{n}{N}}\right)}{N\left(1 - \left(1 + \frac{n}{N}\right)e^{-\frac{n}{N}}\right)} = \frac{1 - e^{\frac{1}{\beta}}}{\beta\left(1 - \left(1 + \frac{1}{\beta}\right)e^{-\frac{1}{\beta}}\right)}. \end{aligned} \tag{4}$$

After the real S and C values are obtained by real observation, n can be computed by (3) and (4). However, it is hard to directly obtain the solution; thus, Eom and Lee utilized an iterative algorithm to approximate the solution of the equation system to estimate the number of tags [15].

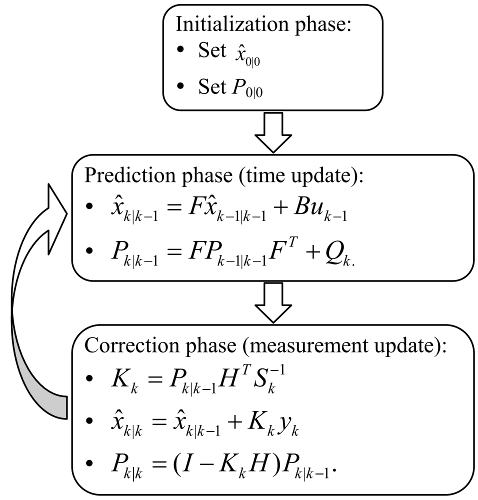

Fig. 2. Flowchart of the Kalman filter.

B. Kalman Filter

Based on linear algebra and hidden Markov model [18], the Kalman filter [16], [17] recursively traces the state of a dynamic system, which is used in many fields, e.g., radar systems, missile tracking, pattern recognition, and tracking of various dynamic systems. One of the most crucial advantages is that, if the assumptions are respected, the estimation results of using Kalman filter have the minimum mean squared error and very high precision. This section introduces the basic Kalman filter and EKF, respectively.

1) Basic Kalman Filter: The Kalman filter requires the process model and the observation model. To cope with a discrete-time controlled process system, the process model can be described by a linear stochastic difference equation, which describes how the system state x_k at time k is evolved from the system state x_{k-1} at time $k-1$, which can be written as

$$x_k = Fx_{k-1} + Bu_k + w_k$$

where F is the state transition model, u_k is the control vector, B is the control-input model, and w_k is the process noise with the normal distribution $N(0, Q_k)$.

On the other hand, the observation model describes the correlation between the observation z_k and the system state x_k at time k, which can be expressed as

$$z_k = Hx_k + v_k$$

where H is the observation model, and v_k is the observation noise with the normal distribution $N(0, R_k)$. Assume that the initial system state x_0, the process noises, and observation noises $\{w_0, w_1, \ldots, w_k, v_1, \ldots, v_k\}$ at every time step are mutually independent.

Then, we explain how Kalman filter uses the process model and observation model for system state estimation. As shown in Fig. 2, let $\hat{x}_{k|j}$ denote the estimated system state at time k with the information at time j and $P_{k|j}$ denote the error covariance matrix of the estimated state at time k with the information at time j, representing the uncertainty of estimated results. At the initialization stage, $\hat{x}_{0|0}$ and $P_{0|0}$ are initialized. Then, each iteration of the main loop includes the prediction stage and the correlation stage. At the prediction stage, the process model and the information at time $k-1$ are used to predict $\hat{x}_{k|k}$ and $P_{k|k}$ at time k, as shown as follows:

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + Bu_{k-1}$$
$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q_k.$$

Then, at the correction phase, the observation is used to update the results of the prediction phase. First, the measurement residual produced by observation z_k at time k is calculated as follows:

$$y_k = z_k - H\hat{x}_{k|k-1} \tag{5}$$

and the covariance matrix S_k for the measurement residual is calculated as follows:

$$S_k = HP_{k|k-1}H^T + R_k. \tag{6}$$

Then, the optimal Kalman gain K_k is calculated as follows:

$$K_k = P_{k|k-1}H^T S_k^{-1}. \tag{7}$$

Finally, the above Kalman gain is used to update the estimated state $\hat{x}_{k|k}$ and the error covariance $P_{k|k}$ for the system state at time k, as shown as follows:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k$$
$$P_{k|k} = (I - K_k H)P_{k|k-1}.$$

2) EKF: The basic Kalman filter previously described can only be applied to linear systems. However, real-world cases are often nonlinear systems, i.e., at least one of the process model and the observation model is nonlinear. To cope with nonlinear systems, the EKF is adopted. Considering that the process model and the observation model are differentiable functions but not necessarily linear functions, then we have the process model and the observation model as follows:

$$\hat{x}_k = f(\hat{x}_{k-1}, u_k) + w_k$$
$$z_k = h(\hat{x}_k) + v_k$$

where f is the process model and h is the observation model. Note that f and h cannot be used directly on covariance matrix yet. In order to solve this problem, they should be linearized by using the current estimated state to calculate the Jacobian matrices, as shown as follows:

$$F_k = \left.\frac{\partial f(x,u)}{\partial x}\right|_{x=\hat{x}_{k-1|k-1};u=u_k}$$
$$H_k = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}_{k-1|k-1}}$$

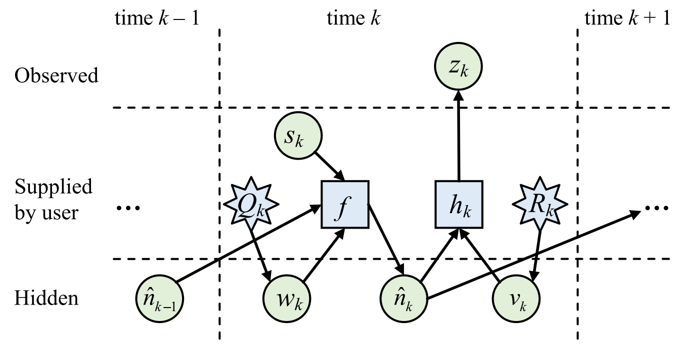

Fig. 3. Hidden Markov model of tag number estimation in RFID systems.

where a time subscript k is considered to make the process model be consistent with the observation model at each time step. The remaining part of the EKF is identical to the basic Kalman filter described previously.

III. Proposed Scheme

The Kalman filter has been shown to minimize the estimation results with a minimum mean squared error; thus, our idea is to apply the EKF to achieve more accurate tag estimation, called the EKF-based tag estimation (EKFBTE) method. To use the EKFBTE to solve this problem, a hidden Markov model is used to describe our RFID system, as illustrated in Fig. 3, in which some variable notations are the same as those in the previous section. In the remainder of this section, we describe in detail how the proposed EKFBTE method uses the EKF for tag number estimation.

A. Process Model and Observation Model

To use the EKF to estimate the number of tags for the RFID system, we have to design how to first update the rules for the system state in the process model and then to identify the correlation between the system state and observation in the observation model.

Consider estimating the number of tags at each read cycle, which is called a *round* in the remainder of this paper. That is, at the end of each read cycle, our objective is to estimate the current number of unidentified tags, meaning that the system state x_k at round k is the number of unidentified tags n_k. Hence, both the vectors of our system state and the observation are of only one element, and a single scalar is used to replace the vector representation in the EKF. As all the matrices and vectors are scalars, the operation rules in the EKF become less restrictive and simpler.

In fact, the number of unidentified tags n_k at round k is the difference of the number of unidentified tags n_{k-1} at round $k-1$ from the number of success slots s_{k-1} at round $k-1$. Hence, s_{k-1} can be viewed as the control vector u_k in the EKF, and the process model is shown as follows:

$$n_k = f(n_{k-1}, s_{k-1}) + w_k = n_{k-1} - s_{k-1} + w_k$$

where w_k is a random variable and also the process noise for EKF, which is the bias possibly produced by the process model,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

and it follows the normal distribution $N(0, Q_k)$. In general, during the process of tag identification, there is no new tag suddenly entering the system nor any original tag leaving the system; hence, in our estimation process, Q_k is set to zero.

Then, the occurrence ratio of idle slots at round k is used in the corresponding round as observation z_k, calculated as follows:

$$z_k = \frac{i_k}{N_k}$$

where i_k is the number of idle slots at round k, and N_k is the frame length of round k. On the other hand, z_k can be obtained from the following equation:

$$z_k = h_k(n_k) + v_k = \left(1 - \frac{1}{N_k}\right)^{n_k} + v_k$$

where

$$h_k(n_k) = \left(1 - \frac{1}{N_k}\right)^{n_k} \tag{8}$$

and v_k is the deviation produced by observation, which follows the normal distribution $(0, Var(v_k))$ where $Var(v_k)$ is the variance of v_k. Note that subscript k is added to function h because the observation model at each round varies from the different dynamic frame lengths N_k.

To calculate the value of $Var(v_k)$ in the observation noise v_k, we first observe the probability distribution of z_k, which is approximate to a binomial distribution

$$P\left(z_k = \frac{i_k}{N_k}\right) = \binom{N_k}{i} p^i (1-p)^{N_k - i}$$

where $i_k \in \{0, 1, \ldots, N_k\}$, and p is the occurrence probability of idle slots at round k theoretically. Hence, the mean and variance of z_k (following a binomial distribution) are shown as follows:

$$E(z_k) = p = \left(1 - \frac{1}{N_k}\right)^{n_k} \tag{9}$$

$$Var(z_k) = \frac{p(1-p)}{N_k}. \tag{10}$$

From (9) and (10), we obtain

$$v_k = z_k - h_k(n_k) = z_k - \left(1 - \frac{1}{N_k}\right)^{n_k} = z_k - p.$$

Hence

$$p = h_k(n_k).$$

From the above equation and (10), we have

$$Var(v_k) = \frac{p(1-p)}{N_k} = \frac{h(n_k)\left(1 - h(n_k)\right)}{N_k}. \tag{11}$$

B. Applying EKF

The process model and the observation model have been established in the previous section. Then, in this section, we start to apply the EKF to tag number estimation. At the prediction phase of the EKF, the system state $\hat{n}_k$ and the error covariance matrix of the system state $P_{k|k-1}$ are predicted as follows:

$$\hat{n}_{k|k-1} = \hat{n}_{k-1|k-1} - s_{k-1} \tag{12}$$

$$P_{k|k-1} = P_{k-1|k-1} + Q_k \tag{13}$$

where Q_k is set to be 0 as previously described.

Then, at the correction phase, the measurement residual y_k at round k is calculated as follows:

$$\begin{aligned} y_k &= z_k - h_k(\hat{n}_{k|k-1}) = z_k - \left(1 - \frac{1}{N_k}\right)^{\hat{n}_{k|k-1}} \\ &= z_k - \left(1 - \frac{1}{N_k}\right)^{\hat{n}_{k-1|k-1} - s_{k-1}}. \end{aligned}$$

Since H_k is a scalar, the covariance matrix of measurement residual S_k is calculated by substituting (13) into (6):

$$S_k = (P_{k-1|k-1} + Q_k)H_k^2 + R_k \tag{14}$$

where H_k is calculated from (8) and (12), as shown as follows:

$$\begin{aligned} H_k &= \left.\frac{\partial h_k(n)}{\partial n}\right|_{n=\hat{n}_{k|k-1}} \\ &= \left(1 - \frac{1}{N_k}\right)^{\hat{n}_{k-1|k-1} - s_{k-1}} \ln\left(1 - \frac{1}{N_k}\right). \end{aligned}$$

Then, the optimal Kalman gain K_k at round k is calculated by (13) and (14)

$$\begin{aligned} K_k &= P_{k|k-1} H_k^T S_k^{-1} \\ &= \frac{(P_{k-1|k-1} + Q_k)H_k}{(P_{k-1|k-1} + Q_k)H_k^2 + R_k} \end{aligned}$$

where R_k is the variance of v_k as calculated in (11), and the actual value n_k is replaced by the predictive value $n_{k|k-1}$, as calculated as follows:

$$R_k = \frac{h_k(\hat{n}_{k|k-1})\left(1 - h_k(\hat{n}_{k|k-1})\right)}{N_k}.$$

Finally, the measurement residual and the Kalman gain are used to calculate the estimated value $n_{k|k}$ for the unidentified tag number at round k and the error variance $P_{k|k}$, as shown as follows:

$$\hat{n}_{k|k} = \hat{n}_{k|k-1} + K_k y_k$$

$$P_{k|k} = (1 - K_k H_k)(P_{k-1|k-1} + Q_k).$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

C. Using Full Observations

In the previous section, we only use the idle slot rate observation. In the real case, however, in addition to the idle slots, the success slots' rate and collision slots' rate also have a direct function mapping relationship with the system state. Theoretically, if we can fully leverage these full observations, the estimation results would be more accurate than those using only the observations of idle slots' rate. Therefore, in this section, we try to adopt the aforementioned full observations. Note that only the different parts from the previous section are explained.

The full observation z_k at round k is rewritten as follows:

$$z_k = \begin{bmatrix} z_{k1} \\ z_{k2} \\ z_{k3} \end{bmatrix} = \begin{bmatrix} \frac{i_k}{N_k} \\ \frac{s_k}{N_k} \\ \frac{c_k}{N_k} \end{bmatrix}$$

where i_k, s_k, and c_k are the numbers of idle slots, success slots, and collision slots at round k, respectively.

Similar to the previous section, under the assumption that all the probability distributions of z_{k1}, z_{k2}, and z_{k3} are binomial distributions, their means and variances can be calculated as follows:

$$E(z_{k1}) = p_{\text{idle}_k} = \left(1 - \frac{1}{N_k}\right)^{n_k}$$

$$E(z_{k2}) = p_{\text{succ}_k} = \frac{n}{N}\left(1 - \frac{1}{N_k}\right)^{n_k - 1}$$

$$E(z_{k3}) = p_{\text{coll}_k} = 1 - p_{\text{idle}} - p_{\text{succ}}$$

$$Var(v_{k1}) = Var(z_{k1}) = \frac{p_{\text{idle}}(1 - p_{\text{idle}})}{N_k}$$

$$Var(v_{k2}) = Var(z_{k2}) = \frac{p_{\text{succ}}(1 - p_{\text{succ}})}{N_k}$$

$$Var(v_{k3}) = Var(z_{k3}) = \frac{p_{\text{coll}}(1 - p_{\text{coll}})}{N_k}.$$

The observation model using full observations is rewritten as follows:

$$z_k = h_k(n_k) + v_k = \begin{bmatrix} p_{\text{idle}_k} \\ p_{\text{succ}_k} \\ p_{\text{coll}_k} \end{bmatrix} + v_k$$

where $v_k = [v_{k1}, v_{k2}, v_{k3}]^T$.

The steps of the prediction phase are the same as those in the previous section. As for the correction phase, the formulas of the measurement residual y_k and the covariance matrix of measurement residual S_k are the same as (5) and (6), respectively, except for the Jacobian matrix H_k and the covariance matrix R_k of v_k. First, the Jacobian matrix H_k is rewritten as follows:

$$H_k = \left.\frac{\partial h_k(n)}{\partial n}\right|_{n=\hat{n}_{k|k-1}} = \left.\begin{bmatrix} \frac{\partial p_{\text{idle}_k}}{\partial n} \\ \frac{\partial p_{\text{succ}_k}}{\partial n} \\ \frac{\partial p_{\text{coll}_k}}{\partial n} \end{bmatrix}\right|_{n=\hat{n}_{k|k-1}}.$$

Then, before computing R_k, we first calculate $\text{cov}(v_{k1}, v_{k2})$. Since

$$\text{cov}(v_{k1}, v_{k2}) = \frac{Var(v_{k1} + v_{k2}) - Var(v_{k1}) - Var(v_{k2})}{2}$$

and $v_{k1} + v_{k2} = (z_{k1} - p_{\text{idle}_k}) + (z_{k2} - p_{\text{succ}_k}) = -(z_{k3} - p_{\text{coll}_k}) = -v_{k3}$, thus

$$\text{cov}(v_{k1}, v_{k2}) = \frac{Var(v_{k3}) - Var(v_{k1}) - Var(v_{k2})}{2}.$$

Similarly, $\text{cov}(v_{k2}, v_{k3})$ and $\text{cov}(v_{k1}, v_{k3})$ can be obtained as follows:

$$\text{cov}(v_{k2}, v_{k3}) = \frac{Var(v_{k1}) - Var(v_{k2}) - Var(v_{k3})}{2}$$

$$\text{cov}(v_{k1}, v_{k3}) = \frac{Var(v_{k2}) - Var(v_{k1}) - Var(v_{k3})}{2}.$$

With these three equations, R_k can be determined as follows:

$$R_k = \begin{bmatrix} Var(v_{k1}) & \text{cov}(v_{k1}, v_{k2}) & \text{cov}(v_{k1}, v_{k3}) \\ \text{cov}(v_{k1}, v_{k2}) & Var(v_{k2}) & \text{cov}(v_{k2}, v_{k3}) \\ \text{cov}(v_{k1}, v_{k3}) & \text{cov}(v_{k2}, v_{k3}) & Var(v_{k3}) \end{bmatrix}.$$

Then, the formula of calculating the optimal Kalman gain K_k at round k is the same as (7). However, in our application of tag number estimation, the determinant for the covariance matrix of measurement residual S_k is very close to zero, as close to a singular matrix, which causes a serious error on computing or judging whether the inverse matrix of S_k exists. Hence, let us recall the derivation of the optimal Kalman gain K_k. Equation (7) can be rewritten as follows:

$$K_k S_k = P_{k|k-1} H_k^T$$

where K_k is a 3-D row vector, S_k is a 3 × 3 matrix, $P_{k|k-1}$ is a scalar, and H_k is a 3-D column vector. Hence, the transpose on the two sides of the above equation can be viewed as the following linear system:

$$Ax = b$$

where $A = S_k^T$, $x = K_k^T$, and $b = P_{k|k-1} H_k$.

In the case where the determinant of the A matrix is very close to zero, the inverse matrix of A is not calculated in practical, but the Moore–Penrose pseudoinverse A^+ of A is adopted instead, where a tolerance is set to remove those values that are too close to zero (i.e., using the $pinv()$ function in MATLAB, there will be a default tolerance for this action), in order to reduce the errors. By doing so, x can be solved as follows:

$$x = A^+ b.$$

Hence, $K_k = x^T$ is the required optimal Kalman gain. Then, the part of updating $P_{k|k}$ and $n_{k|k}$ is the same as the previous section.

TABLE I
DEFAULT ATTRIBUTE VALUES USED IN THE SIMULATION

Parameters	Value	Meaning
N_0	150	Initial frame size
$P_{0\|0}$	10000	Initial error variance
$\hat{n}_{0\|0}$	150	Initial number of tags
Q_k	0	Covariance matrix of process noise

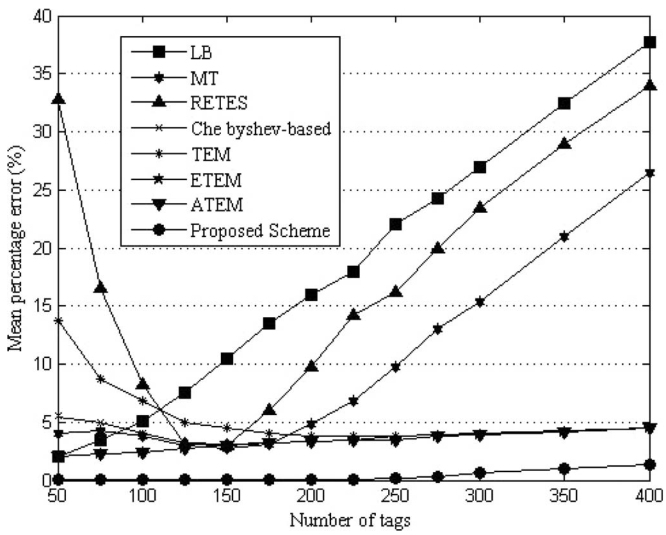

Fig. 4. Mean percentage error versus number of tags (fixed frame size).

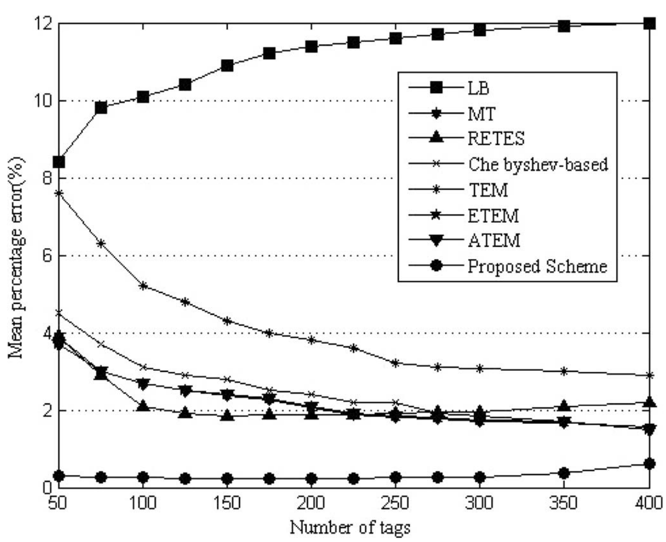

Fig. 5. Mean percentage error versus number of tags (dynamic frame size).

IV. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed scheme. The default parameters used in the simulation are listed in Table I. We used an event-driven custom simulation program, which is written by the C++ programming language, and MATLAB to verify the performance of the proposed scheme.

The accuracy of the proposed extend Kalman filter-based estimation algorithm is demonstrated in Figs. 4 and 5. As shown in Fig. 4, when the fixed frame size is considered, the accuracy of the LB and MT methods is quite acceptable when the number of tags is below 100, but their performance gets worse in dense environments. That is, the higher the number of tags, the lower the accuracy. In general, the Chebyshev-based method has a better performance because of the occurrence ratio of idle slots, success slots, and collision slots stably corresponding to the expectations. However, among all the existing well-known estimation models, our scheme obtains much higher accuracy in both high- and low-density environments.

When the dynamic frame size is considered, the LB method has the worst performance. This is because the number of tags is always underestimated, and hence, its frame size is usually too small. However, as shown in Fig. 5, the accuracy of the MT method has greatly improved when a dynamic frame size is considered because it will constantly update its frame size to reach the assumption of $N = n$. The RETES method has the similar results as well. Finally, as we expected, our scheme provides much better accuracy than all the existing well-known estimation models no matter in sparse or dense environments, and please note that its performance is stable.

V. CONCLUSION

The DFSA algorithm is the *de facto* anticollision algorithm for RFID systems, but it requires a good tag number estimation method in order to achieve high efficiency. The EKF has been shown to theoretically obtain the estimated results with the minimum squared error for nonlinear systems. In this paper, we have shown that, even if the actual situation of the tag number estimation for RFID systems is not fully consistent with the assumptions of the EKF, our method is still undoubtedly able to achieve a quite remarkable system performance. Our EKF-based tag number estimation method considers the relationship among idle slots, success slots, collision slots, and the number of tags as the observation model to accurately estimate the number of tags on run-time measurements. We also find that using a full observation model indeed achieves higher accuracy than using a single observation model. Finally, the experimental results show that the proposed scheme has the highest precision, as compared with various previously proposed methods, and has stable performance in diverse environments.

REFERENCES

[1] R. Want, “An introduction to RFID technology,” *IEEE Pervasive Comput.*, vol. 5, no. 1, pp. 25–33, Jan.–Mar. 2006.
[2] K. Domdouzis, B. Kumar, and C. Anumba, “Radio frequency identification (RFID) applications: A brief introduction,” *Adv. Eng. Informat.*, vol. 21, no. 4, pp. 350–355, Oct. 2007.
[3] H. Liu *et al.*, “Grouping-proofs based authentication protocol for distributed RFID systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no.7, pp. 1321–1330, Jul. 2013
[4] Y. Zhang *et al.*, “Home M2M networks: Architectures, standards, and QoS improvement,” *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 44–52, Apr. 2011.
[5] Y. Zhang *et al.*, “Cognitive machine-to-machine communications: Visions and potentials for the smart grid,” *IEEE Netw. Mag.*, vol. 26, no. 3, pp. 6–13, May/Jun. 2012
[6] K. Ali, H. Hassanein, and A.-E. M. Taha, “RFID anti-collision protocol for dense passive tag environments,” in *Proc. IEEE Conf. Local Comput. Netw.*, 2007, pp. 819–824.
[7] Z. W. Yuan and L. D. Duan, “Research on the tag estimation and frame length of the ALOHA algorithm,” in *Proc. Int. Conf. Biomed. Eng. Informat.*, 2012, pp. 1511–1514.
[8] F. C. Schoute, “Dynamic frame length ALOHA,” *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 565–568, Apr. 1983.

[9] D. J. Deng and H. W. Tsao, "Optimal dynamic framed slotted ALOHA based anti-collision algorithm for RFID systems," *Wireless Pers. Commun.*, vol. 59, no. 1, pp. 109–122, Jul. 2011.

[10] H. Vogt, "Multiple object identification with passive RFID tags," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2002, vol. 3, pp. 6–13.

[11] J. R. Cha and J. H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in *Proc. Int. Conf. Parallel Distrib. Syst.*, 2005, vol. 2, pp. 63–67.

[12] P. Pupunwiwat and B. Stantic, "A RFID explicit tag estimation scheme for dynamic framed-slot ALOHA anti-collision," in *Proc. Int. Conf. Wireless Commun. Netw. Mobile Comput.*, 2010, pp. 1–4.

[13] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Int. Conf. Pervasive Comput.*, 2002, pp. 98–113.

[14] Z. Li, C. He, and H. Z. Tan, "An enhanced tag estimation method applied to tag anti-collision algorithm in RFID systems," in *Proc. Int. Conf. Inf. Sci. Technol.*, 2011, pp. 703–708.

[15] J. B. Eom and T. J. Lee, "Accurate tag estimation for dynamic framed-slotted ALOHA in RFID systems," *IEEE Commun. Lett.*, vol. 14, no. 1, pp. 60–62, Jan. 2010.

[16] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, vol. 82, pp. 35–45, 1960.

[17] G. Welch and G. Bishop, "An introduction to the Kalman filter," Univ. North Carolina at Chapel Hill, Chapel Hill, NC, USA, Notes ACM SIGGRAPH Tutorial Kalman Filter, 2001, pp. 19–34.

[18] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Comput.*, vol. 11, no. 2, pp. 305–345, 1999.

Der-Jiunn Deng (M'10) received the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2005.

He joined National Changhua University of Education, Changhua, Taiwan, as an Assistant Professor with the Department of Computer Science and Information Engineering in August 2005 and then became a Full Professor in February 2012. His research interests include multimedia communication, quality of service, and wireless networks.

In 2010, 2011, and 2012, he received the Research Excellency Award of National Changhua University of Education. In 2012, he also received the Outstanding Faculty Research Award of National Changhua University of Education. Dr. Deng served or is serving as an editor and guest editor for several technical journals. He also served or is serving on several symposium chairs and technical program committees for IEEE and other international conferences.

Chun-Cheng Lin received the B.S. degree in mathematics, the M.B.A. degree in business administration, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2000, 2007, and 2009, respectively.

He was an Assistant Professor with the University of Taipei, Taipei (2010–2011), and National Kaohsiung University of Applied Sciences, Kaohsiung (2009–2010). Since August 2013, he has been an Associate Professor with National Chiao Tung University, Hsinchu, Taiwan, which he joined as an Assistant Professor in February 2011. His main research interests include graph drawing and information visualization, wireless networks, design and analysis of algorithms, and computational management science.

Tzu-Hsun Huang received the B.S. degree in computer science and information engineering from National Taipei University, Taipei, Taiwan, in 2011 and the M.S. degree in electrical engineering from National Taiwan University, Taiwan, in 2013. His research interests include the theory of computation, image recognition, and optimization.

Hsu-Chun Yen received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1980; the M.S. degree in computer engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1982; and the Ph.D. degree in computer science from The University of Texas at Austin, Austin, TX, USA, in 1986.

From August 1986 to July 1990, he was an Assistant Professor of computer science with Iowa State University, Ames, IA, USA. In August 1990, he joined National Taiwan University, where he is currently a Professor of electrical engineering. His current research interests include automata theory and formal languages, Petri net theory, graph drawing, design and analysis of algorithms, and formal methods.