Dynamic Channel Calibration on a Crystal-Free Mote-on-a-Chip

Ioana Suciu, Filip Maksimovic, Brad Wheeler, David Burnett, Osama Khan, Thomas Watteyne, Xavier Vilajosana, Kristofer Pister

Dynamic Channel Calibration on a Crystal-Free Mote-on-a-Chip

IOANA SUCIU[1], FILIP MAKSIMOVIC[2], BRAD WHEELER[2], DAVID BURNETT[2], OSAMA KHAN[2], THOMAS WATTEYNE[3], XAVIER VILAJOSANA[1,4], KRISTOFER S.J. PISTER[2]

[1]Worldsensing S.L., Barcelona, Spain. (isuciu@worldsensing.com)
[2]University of California Berkeley, Berkeley, CA, USA (fil,brad.wheeler,db,oukhan,ksjp@berkeley.edu)
[3]Inria, Paris, France (thomas.watteyne@inria.fr)
[4]Universitat Oberta de Catalunya, Barcelona, Spain (xvilajosana@uoc.edu)

Corresponding author: Ioana Suciu (isuciu@worldsensing.com).

ABSTRACT The single chip integration of a wireless sensor node would allow for cheap, low-power, dust-size devices. The key to realizing this vision is to eliminate bulky off-chip frequency references such as crystal oscillators or resonators, and their associated power-hungry circuitry. The immediate challenge of removing off-chip references is that there is no accurate on-chip frequency references, which makes it hard to tune the radio to the right frequency, and to keep an accurate sense of time. This article offers a full solution for crystal-free devices, which includes (1) initiating communication in an IEEE802.15.4 network, (2) synthesizing the 16 communication channels at startup temperature, and (3) continuously applying corrections to the inaccurate timing source to allow keeping frequency synchronization on all communication channels over a 5–55°C temperature range. The proposed methods are accompanied by simulations and an experimental validation on the first fully-functional Single Chip Micro Mote hardware implementation. Our simulations and experimental results validate that the proposed approach achieves radio clock synchronization accuracy close to the 40 ppm limit imposed by the IEEE802.15.4 standard.

INDEX TERMS Clock calibration, crystal-free radio, IEEE802.15.4, low-power wireless mesh networking, reference frequency stability, short-range wireless, standard-compliance.

I. INTRODUCTION

THE explosive growth of IoT is pushing the market (consumer, industrial, military) towards cheap, low-power devices with a strong focus on miniaturization. Application domains include activity trackers, wearables, smart materials, factory automation, in-body and personal health monitoring, and microrobots. To be able to scale to trillions of deployed wireless devices [1], they need to be small and cheap. This is the primary motivation for designing radios that do not require any external components [2].

Crystal oscillators are used in most electronic circuits as the primary reference for time and frequency. They offer high frequency accuracy and stability over time and temperature [3]. Crystal oscillators have undoubtedly become cheaper and smaller and offered better performance [4]. Their main drawback remains that they are inherently external (off-chip) components: using them requires special fabrication and packaging steps, and process integration beyond CMOS [5].

There is important trend towards adopting MEMS oscillators as an alternative to crystals [6]. MEMS oscillators are complex, silicon-based structures, consisting of a resonator, a fractional-N phase-locked-loop (PLL) with a temperature sensor and compensation circuitry [6]. MEMS oscillators allow for a more customizable package size, a much higher accuracy than equivalent crystals over high temperature variation, and increased resistance to shock and vibration [7], [8]. While crystals have to be cut into a fixed size, MEMS oscillators can have a $60 - 80\%$ smaller footprint, placed on top of the ASIC [9]. They typically feature higher phase noise and energy consumption than equivalent crystals [7]. Texas Instruments [10], a global semiconductor design and manufacturing company, has very recently released the industry's first crystal-free microcontroller unit, the SimpleLink
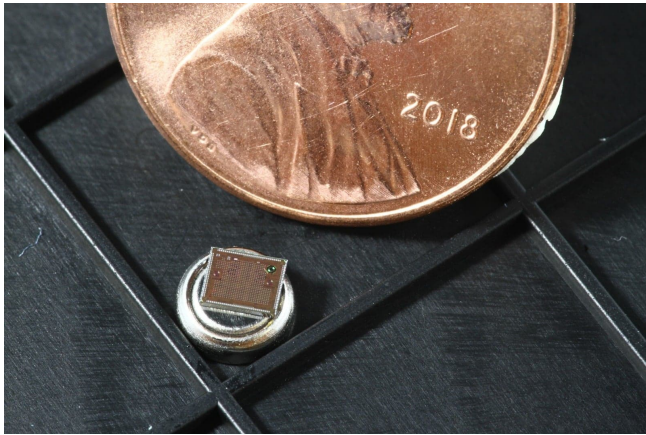
FIGURE 1: The Single-Chip Mote next to a 1 U.S. cent coin.

CC2652 [11], that uses in its structure a MEMS-based acoustic resonator [12]. MEMS oscillators are, exactly as crystal oscillators, off-chip components. And while the CC2652 is a good example of a MEMS+CMOS integration in a single package, it is still a packaging exercise, rather than a full single-chip solution.

Our goal is to go one step further on true single-chip radio design, which requires neither a crystal nor a MEMS oscillator. We also want to remove the need for power-hungry circuitry such as a Phase-Locked Loop (PLL) [13]. The objective is to achieve good system performance with this cheap, dust-size, ultra low power, single-chip integration of a wireless sensor node. The space savings of removing off-chip elements is significant. Wireless communication systems ("radio chips") have been demonstrated that are as small as $0.46\ mm^2$ [14], $2.025\ mm^2$ [15], $\approx 4.4\ mm^2$ [16], $\approx 9\ mm^2$ [17]. As a point of comparison, the area occupied by a small crystal oscillator is $\approx 2mm^2$ [1], [4].

We validate our approach experimentally on the SCuM "Single-Chip μMote", a prototype implementation of the "Single-Chip Mote" [18], [19]. SCuM has a die area of $2.5mm \times 3mm$ and features a 2.4 GHz transceiver, a Cortex-M0 microprocessor, 128 kB of RAM memory, and all the support hardware. Fig. 1 is a picture of SCuM. The radio offers a 10 m communication range, and draws 670 uA in reception, 1 mA when transmitting at -10 dBm. As a point of comparison, the crystal-based industry leading low-power technology (2.4 GHz, IEEE802.15.4 TSCH) is the 10 mm × 10 mm LTC5800-IPM [20], consuming 4.5 mA when receiving and 5.4 mA when transmitting at 0 dBm.

Without a high accuracy time reference, it is hard to maintain compatibility with standards. The goal of our work is to be compliant to the IEEE802.15.4 [21] standard, which mandates that the radio frequency drift stays below 40 ppm. This translates into two challenges.

The first challenge is initiating communication, i.e. being able to receive a first IEEE802.15.4-compliant frame using SCuM. Unlike crystal-based radios, at startup, a crystal-free radio such a SCuM can have its on-chip oscillators 10,000+ ppm off their nominal value [22]. The tuning code that outputs a channel's center frequency is initially unknown to the device and has to be determined in order to be able to initiate communication. After finding the tuning code value X that tunes the radio on the communication channel Y, the crystal-free radio exhibits frequency errors limited by the phase noise of the local oscillator and by the quantization errors caused by the integer nature of the tuning codes. The accumulated effects of the phase noise and quantization errors can be limited in the design phase of the respective oscillators to be within the ± 40 ppm, at constant temperature [23].

The second challenge is maintaining communication when the temperature changes. On-chip oscillators beat at a different frequency when the temperature changes. This can cause a loss of channel calibration (the chip is communicating at the wrong frequency) and of time synchronization (the chip is communicating at the wrong time). Figure 2 shows the behaviour of an on-chip oscillator that is calibrated for communication on channel Y: at constant temperature, the frequency errors measured on this oscillator are limited from the design phase to be within 40 ppm; as the temperature increases, this oscillator starts to accumulate errors until it oversteps the 40 ppm accuracy and eventually exceeds the channel limits. There are a number of related works which look at the temperature stability of on-chip frequency references [24]–[28]. Yet, to stay within the 40 ppm drift limit and maintain compliance with IEEE802.15.4, calibration algorithms need to run continuously to detect temperature changes and correct its effect on frequency drift [23].

This article introduces a mechanism to bootstrap and calibrate the on-chip Radio Frequency (RF) oscillator of a crystal-free radio to enable the normal operation of an IEEE802.15.4 TSCH network. Specifically, we tackle the problem of synthesizing the IEEE802.15.4 communication channels with low frequency errors at both constant and variable temperature. This work is, to best of our knowledge, the first to experimentally validate this approach on the SCuM platform, a prototype implementation of Single Chip Mote [18], [19].

The remainder of the article is organized as follows. Section II summarizes the related work on frequency correction techniques. Section III describes the network formation particularities of IEEE802.15.4 standard and its consequences on crystal-free designs. Section IV summarizes the process of initial RF clock calibration of a crystal-free device and the corrections to be applied to the remaining on-chip oscillators. Sections V, VI and VII present the algorithms and simulations for synthesizing the IEEE802.15.4 communication channels at constant and variable temperature. Section VIII focuses on the experimental work and discusses the results of an extensive experimentation campaign. Section IX discusses the challenges of transmitting data with crystal free radios and from the best of our knowledge, this is the first work to address these challenges. Section X concludes this article.
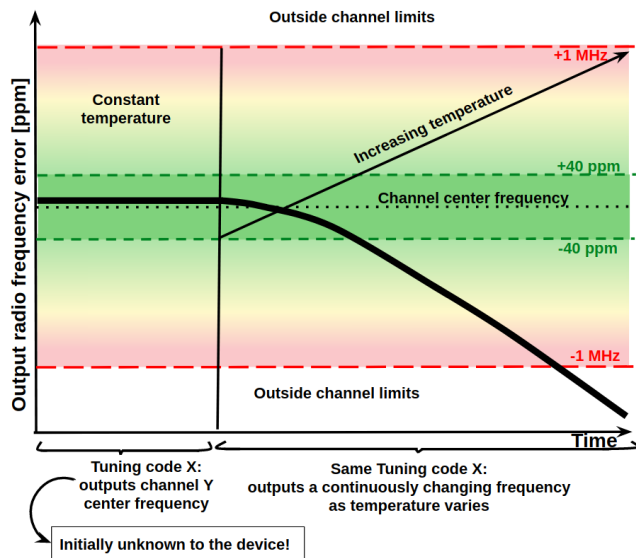
FIGURE 2: The temperature impact on the output frequency of a crystal-free radio calibrated for communication on channel Y. While at constant temperature, the frequency errors are limited, a continuous change in temperature causes the radio to drift outside the ± 40 ppm accuracy limits and eventually, outside of the communication channel.

II. RELATED WORK

Applying compensation techniques to correct the deviation from the expected value of the radio frequency of a wireless device is a topic that has been addressed in different contexts in the literature.

Staszewski *et al.* [29] propose a novel method to perform frequency synchronization of a local oscillator in a mobile phone to a reference sent by the base-station (BS). The usual approach is for the mobile device to determine the deviation of its reference frequency by using the accurate carrier frequency received from a base-station, and use that to directly adjust the frequency of an (expensive) digitally controlled crystal oscillator (DCXO). In this way, the known frequency command word (FCW) applied to the frequency synthesizer to synthesize the communication channel frequency is always the same. In what Staszewski *et al.*proposes, the local oscillator (LO) has a free-running crystal oscillator as a reference (XO) with no option for applying frequency corrections. Their method proposes compensating the deviations by continuously modifying the frequency command word (FCW) of the frequency synthesizer while leaving the XO frequency unchanged. In this way, the FCW of a communication channel is no longer fixed, which allows an XO to be used, which is cheaper than a DCXO.

Bae *et al.* [30] focus on a crystal-free wireless body area network (WBAN) sensor node. To avoid using a bulky and expensive crystal, they replace it with an injection-locking digitally controlled oscillator (IL-DCO), while not altering the remaining transceiver blocks. By applying a calibration method, they managed to keep the 80 MHz DCO within

100 kHz accuracy over a 100°C temperature variation. Every 100 ms, the base-station sends an accurate 80 MHz tone, during which the sensor nodes calibrate their DCOs. This calibration technique is based on a specific property of the IL oscillators: they are able to measure the phase difference between the signal received from the BS and the oscillation frequency of the IL-DCO. By sweeping the N bit tuning code of the DCO, Bae *et al.*are able to identify the range of codes within which the DCO operates in the injection-locked phase, and tune the DCO to the center of this range (assuming that the DCO output frequency is linear with the input tuning code). The output frequency of the DCO is as close as possible to that of the reference signal. The calibrated DCO is further used as a reference in an UWB-FM transmitter block that contains a power-hungry phase-locked loop frequency synthesizer, in the same way a crystal oscillator would have been used. This calibration method is similar to the one used by Staszewski *et al.* [29].

Griffith *et al.* [31] use a BAW-based 2.52 GHz oscillator. This oscillator has a stability of ± 10 ppm over a 165°C temperature range. The authors further use of a PLL and a temperature sensor. In order to compensate for the strong negative temperature coefficient of the silicon resonator in the BAW structure, every time the radio is needed, the temperature value is read, a prediction of the frequency offset is made, and the FCW of the PLL is changed to compensate for this offset. The prediction is made based on a second-order polynomial that describes the BAW frequency offset over temperature. The parameters of this polynomial are determined by measurements made at three temperatures during production testing, and are stored in the device flash. Using this calibrated off-chip reference, the on-chip oscillators (for example 32 kHz) are periodically corrected.

Khan *et al.* [22] show that an RC oscillator could be disciplined with network feedback to provide a time reference with sufficient accuracy. In order to demonstrate the feasibility of their proposal, the authors use the off-the-shelf OpenMote-CC2538 [32]. One OpenMote-CC2538 is programmed to use its crystal and transmit. The other two OpenMote-CC2538 boards simulate a crystal-free device: RX and RC connected to an FPGA that calibrates the RC oscillator. Khan *et al.*obtain an accuracy of 70 ppm for a 1 MHz RC oscillator. In [2], the authors extend this work by testing their approach on a FPGA implementation of the digital system of a crystal-free mote. Using a wired setting, the accuracy obtained using network calibration is 47 ppm for a 25 MHz oscillator.

While traditional radios require a XTAL and a PLL-based synthesizer for selecting the communication channel of interest, Wang *et al.* [33] propose a PLL and XTAL-free radio architecture to be used in a 2.4 GHz IEEE802.15.4 network. They replace the crystal oscillator with a 0.1 mm² off-chip temperature-compensated thin-film bulk-acoustic-wave resonator (FBAR) that has a ± 50 ppm stability over a 100 °C temperature range. By using a fixed 2.4 GHz LO generated by the FBAR and a second configurable LO, they are able

to directly select any of the 16 IEEE802.15.4 channels with good accuracy at any temperature within the operational range.

Chuo *et al.* [34] consider the communication of a $3 \times 3 \times 3$ mm^3 XTAL and PLL-free sensor node with a gateway, at 915 MHz. The system architecture is designed so as to resonate with the attached magnetic antenna at 915 MHz, in both RX and TX modes. Because there is no PLL in TX mode, the authors use a more complex gateway that enables a wider frequency search, so no calibration is needed on the sensor side. In RX mode, the high quality factor of the resonator tank at 915 MHz is enough not to need an accurate off-chip reference. This approach is, however, only valid for single-channel communication.

Tu *et al.* [35] propose a frequency compensation technique to tackle the temperature drifts experienced by a DCO. At startup, the DCO frequency is defined as a target frequency at the startup room temperature, and its value is digitized and stored in a register. While the DCO frequency variation has a positive temperature coefficient, the frequency compensation algorithm provides negative temperature coefficients to compensate for the drift and bring the DCO back to the target value. The authors make use of dedicated circuits that enable them to determine and adjust the value of the DCO's temperature coefficient.

Wheeler *et al.* [5] are the first to demonstrate the crystal-free demodulator feedback that we use in this article. They use a free-running on-chip LC tank as the local oscillator of an IEEE802.15.4 transceiver that is characterized by a temperature drift of 95 ppm/°C. To deal with temperature variations, the authors use demodulator-based feedback to allow the receiver to track the drift of the transmitter when it is placed in a temperature chamber and subjected to a temperature variation of $2^\circ\ C/min$. The receiver is able to track the transmitter's signal by adjusting the LO at each received frame to keep the value of the intermediate frequency (IF) constant. The devices used in the experiments are composed of 65 nm CMOS RF chips with the digital part running on FPGA boards.

Our previous work [23] analyzes the stability over time and temperature of two fundamental on-chip oscillators of the prototype implementation of the Single-Chip Mote. These oscillators drive the wireless communication capabilities of a radio in typical IEEE802.15.4 networks. We propose a generic search method by which any crystal-free platform with an uncalibrated on-chip oscillator tunes its radio to receive IEEE802.15.4-compliant frames. The algorithm is based on demodulator feedback and tracks received IEEE802.15.4 frames. It maintains calibration of all on-chip oscillators so they stay within the IEEE802.15.4 oscillator specifications, even when subject to temperature variations. This work is, however, limited to finding and maintaining communication on a single channel, and hence does not address multi-channel communication.

In this article, we present a set of algorithms for clock calibration on a crystal-free IEEE802.15.4 compliant device.

Clock calibration is necessary to tune the radio's frequency to any of the 16 IEEE802.15.4 channels. The radio we use only features low-cost on-chip oscillators, and no high quality resonators, PLLs or frequency synthesizers. We show through simulations and experimentation that, using our algorithms, the radio maintains communication even when undergoing temperature changes, with an accuracy close to the ± 40 ppm limit imposed by the IEEE802.15.4 standard.

III. NETWORK FORMATION

We focus on the IEEE802.15.4 Time Synchronized Channel Hopping (TSCH) mode of the IEEE802.15.4 standard. TSCH is a technique which has been designed to provide ultra high reliability and ultra low power operation, and is at the heart of standards such as WirelessHART [36] and 6TiSCH [37], [38].

When joining a TSCH network, a node first listens for a beacon frame. Beacon frames are regularly sent by nodes already part of the network. Once the joining node has received a beacon, it is synchronized to the network, and initiates a secure join handshake. This handshake ensures mutual authentication between the joining node and the network [39].
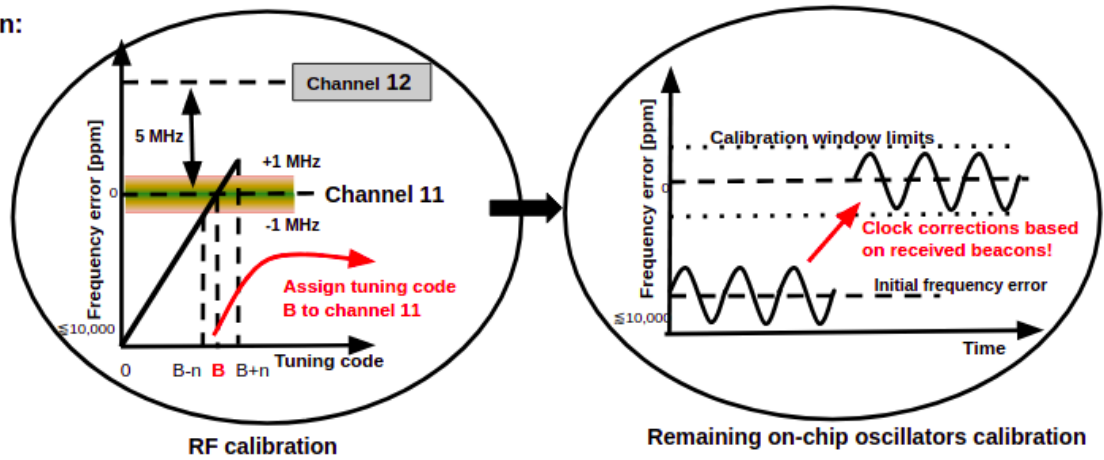
TSCH devices are typically clocked by a low-power 32 kHz crystal with a drift of 10-30 ppm. Because of this drift, a node needs to regularly re-synchronize to its time source neighbor to remain synchronized to the network. Assuming a maximum de-synchronization error of 1 ms and a 30 ppm drift, this re-synchronization needs to happen at least every 30 s.

The drift of the internal oscillators of a crystal-free device can be thousands of times higher than that of crystals. This impacts not only the rate at which nodes de-synchronize, but also their capability to tune the communication frequency and communication data rate. This article details a series of algorithms to compensate for these drifts. In an initial calibration phase, a node measures the ratio between its radio frequency and its clock frequency based on the periodic IEEE802.15.4 beacons sent by a crystal-based node. In a startup phase, it discovers the clock settings for each communication channel, at startup temperature. During normal operation, it continuously adjusts these clock settings to compensate for clock drifts caused by temperature variations. Fig. 3 summarizes these phases. In future networks composed only of crystal-free devices, a crystal-based sensor node that is always powered could act similar to a base station, sending periodic beacons on all channels so as to ensure that all the sensor nodes are able to maintain their frequency accuracy.
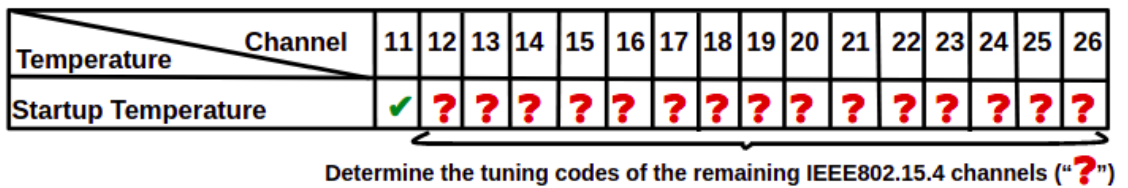
IV. INITIAL CALIBRATION

The crystal-free radio uses Digitally-Controlled Oscillators (DCOs) as time references. When a crystal-free radio joins a network, since its DCO can 10,000 ppm off its target frequency [22], it doesn't know which DCO setting corresponds to which frequency.

Initial calibration:

RF calibration

Remaining on-chip oscillators calibration

Startup:

	Channel	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Temperature																	
Startup Temperature		✓	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Determine the tuning codes of the remaining IEEE802.15.4 channels ("?")

Normal operation:

	Channel	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Temperature																	
Startup Temperature		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
New Temperature		✓		?					?				?				

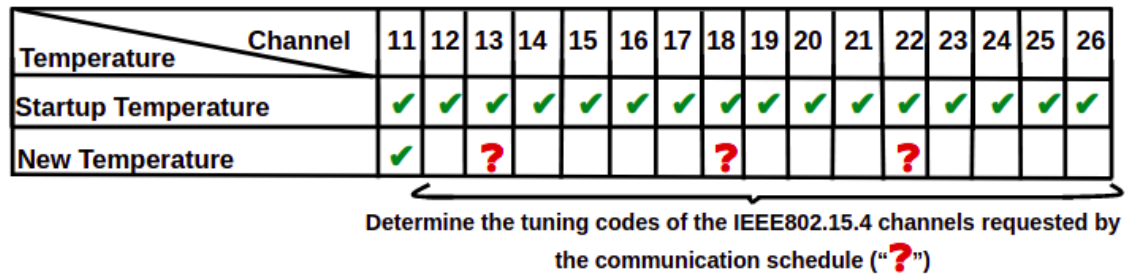Determine the tuning codes of the IEEE802.15.4 channels requested by the communication schedule ("?")

FIGURE 3: Crystal-free radio. Steps for establishing communication on all channels, at any environmental temperature.

The goal of the initial calibration is to find that relationship. It only happens once. During the initial calibration, a node turns on its radio in reception mode and sweeps the tuning code of its DCO across all possible values. It does so until it successfully receives a frame.

$$duration_{RF\_cal} \approx (B + n) \times t_L$$
$$t_L \approx 10 \times T_B \qquad (1)$$
$$T_B = 160 \text{ms}$$

Eq. (1) expresses the duration of the initial calibration phase. $(B + n)$ is the number of tuning code adjustments until reaching the upper limit of channel 11 (see Fig. 3). t_L is the amount of time the radio listens for each tuning code setting. t_L duration ($t_L \approx 10 \times T_B$) is large enough for the radio to be able to receive multiple beacons, which is needed in case the wireless link from the crystal-based node sending the beacons is lossy. $T_B = 160$ ms is the time it takes for the

node sending the beacons to hop across all 16 IEEE802.15.4 frequencies (in TSCH, a 10 ms timeslot is typical).

The IEEE802.15.4 standard uses 16 frequencies – numbered 11-26 – in the 2.4 GHz band. The beacons are sent on all 16 frequencies using a known channel hopping sequence that lasts for T_B. This means that the joining node, when listening on a single frequency, receives at most one beacon every T_B.

The search process starts on the minimum supported code (tuning code 0). The device listens for t_L seconds on the synthesized frequency (F_{min}), then increments the tuning code by 1 (which changes the radio frequency to $F_{min} + \Delta F$, where ΔF is the RF oscillator tuning resolution). The radio does not receive any beacon until its tuning code reaches $B - n$ (as shown in Fig. 3), which corresponds to a synthesized frequency at the inferior limit of IEEE802.15.4 channel 11. The number of beacons received with a correct Cyclic Redundancy Check (CRC_OK) increases as the tuning code

reaches value B, corresponding to a synthesized frequency close to the center frequency of channel 11. The number of received beacons decreases as the synthesized frequency approaches the upper limit of the channel, as the error between the center frequency of the channel and the synthesized frequency increases. The search process stops after the synthesized radio frequency is outside the communication channel (tuning code value $B + n$), at which point no more beacons are received. The crystal-free device tunes its radio using tuning code B (corresponding to the maximum reception performance) and is ready for receiving the periodic beacons sent on channel 11 with an accuracy within ± 40 ppm.

The time interval between successive beacons received on channel 11 is used to calibrate the remaining on-chip oscillators, as shown in Fig. 3 as the step to follow after RF calibration. These oscillators generate other standard-specified rates, such as the communication bit rate and the real-time clock. The correction to be applied to an on-chip oscillator is given by (2), where $\Delta F_{oscillator[Hz]}$ is the tuning resolution of that oscillator.

$$correction = -\frac{ticks_{oscillator} - ticks_{idealF[Hz]}}{\Delta F_{oscillator[Hz]}} \quad (2)$$

To correct the frequency error, and assuming that the oscillator should have a nominal value of F Hz, we compare the number of ticks this oscillator counts with the value an ideal F Hz clock would count between two successive beacons (T_B).

The duration of the complete initial calibration phase depends on $duration_{RF\_cal}$ and on how restrictive the set calibration window for other on-chip oscillators is. More discussions about the search and correction algorithms and the experimental results are included in our previous work [23].

The initial calibration phase is valid for any crystal-free device starting its operation at any temperature. At the end of this phase, each device has all of its on-chip oscillators calibrated and has the knowledge of one communication channel, which is primarily channel 11 in IEEE802.15.4 networks. Tuning code value B, corresponding to channel 11, has a different value for each device and for each startup temperature. Each device stores B as the tuning code needed for synthesizing channel 11 at startup temperature, which can be measured by an on-chip sensor. The following sections discuss the challenges of synthesizing the remaining 15 IEEE802.15.4 channels, first at the startup temperature, then dynamically as temperature changes.

V. FREQUENCY SYNTHESIS DISCUSSION
The IEEE802.15.4 standard at 2.4 GHz mandates the use of at most 16 channels (numbered 11-26), spaced at 5 MHz, occupying the 2405-2480 MHz band [40]. ΔF^{approx} [Hz] is the frequency resolution of a tuning code, a value chosen in the design phase of the chip. This value may slightly vary from chip to chip and can be influenced by the environmental temperature and operational frequency.

As shown in Fig. 3, the startup phase is dedicated to determining the tuning codes of the remaining IEEE802.15.4 channels at startup temperature, once the tuning code B corresponding to channel 11 is found and all the remaining on-chip oscillators are calibrated. As the next IEEE802.15.4 channel, channel 12, has a center frequency spaced 5 MHz above channel 11, the first guess for synthesizing channel 12 would be to use a tuning code C, which can be inferred using (3).

$$C = B + offset$$
$$offset = \lceil \frac{5MHz}{\Delta F^{approx}\ Hz} \rceil \quad (3)$$

The oscillation frequency of an LC oscillator is given by (4) [41]:

$$f = \frac{1}{2\pi\sqrt{L \times C_e}} \quad (4)$$

L is the inductance ([H]); C_e is the capacitance ([F]). A linear change in C_e (determined by the change in tuning code value) around a quiescent point has second order influence on frequency. By analyzing the frequency response to the linear change in tuning codes of the RF oscillator through measurements, we observed that a second degree function approximates it better than a linear function and that this is valid for all the tested temperatures ($5 - 55°$ C). As a consequence, using (3) to synthesize the center frequency of channel 12 will introduce three sources of error:

- ΔF^{approx} is an approximation of the particular ΔF of the chip (error ϵ_Δ)
- $offset$ is an integer approximation of a real value (error $\epsilon_{integer}$)
- C is a linear approximation of a second degree function (error ϵ_{linear})

The total frequency error (ϵ^1) with respect to center frequency of channel 12 obtained when using tuning code C is expressed by (5):

$$\epsilon^1 = \epsilon_\Delta + \epsilon_{integer} + \epsilon_{linear} \quad (5)$$

where ϵ_Δ represents the tuning error introduced by using an approximation of ΔF, $\epsilon_{integer}$ is the error introduced by having to approximate the float $offset$ to an integer and ϵ_{linear} is the error added by using a linear approximation for the second degree function that describes the frequency response of the oscillator with the linear change in tuning codes. Extrapolating from tuning code B used for channel 11, to synthesizing directly channel $11 + i$ ($i \leqslant 15$) using a linear approximation code $Q = B + i \times offset$, would mean synthesizing a frequency that has an error E with respect to the center frequency of the channel $11 + i$: $E \approx i \times \epsilon^1$. The value of E could become so significant that the synthesized frequency is more than 1 channel apart from the desired communication channel.

In order to determine the tuning codes needed for synthesizing the remaining 15 IEEE802.15.4 channels, and not to

have to deal with cumulative errors, we propose to synthesize channels in consecutive order and refine the channel's tuning code at each step by using intermediate frequency (IF) based calibration, as expressed by (6).

$$
\begin{aligned}
&B \text{ (channel 11)} \\
C = &B + offset_i \text{ (channel 12)} \\
D = &C_{IF} + offset_j \text{ (channel 13)} \\
E = &D_{IF} + offset_k \text{ (channel 14)} \\
&\qquad\qquad\qquad \cdots
\end{aligned} \tag{6}
$$

This approach synthesizes each channel frequency with an initial error $\epsilon \leqslant \epsilon^1 \ll 1$ MHz (channel edge), that is reduced to $\epsilon_{integer}$ after IF-based calibration. As an example, when synthesizing a channel's center frequency (F_c) using the tuning function of the radio oscillator f with the tuning codes C and C_{IF} (after IF calibration), the returned errors are ϵ and $\epsilon_{integer}$ respectively, expressed by (7).

$$
\begin{aligned}
f(C) = F, \quad f(C_{IF}) &= F_{IF} \\
|F - F_c| = \epsilon, \quad |F_{IF} - F_c| &= \epsilon_{integer} \\
\epsilon_{integer} &\ll \epsilon
\end{aligned} \tag{7}
$$

The approaches for adjusting the value of $offset$ to $offset_{i,j,k..}$ in order to determine all the communication channels are the subject of Section VI. In the remainder of this section, we detail IF-based calibration.

As shown in Fig. 4, when the crystal-free device uses an initial tuning code (e.g. C) for receiving packets on channel Y, once a packet is received on that channel, the device can count the intermediate frequency (IF) value [19]. The IF is created by mixing the incoming signal on channel Y with the local oscillator signal, resulting in a signal at a difference frequency. If the IF value is within 40 ppm of the expected value (2.5 MHz, a design parameter of the chip hardware), the LO is tuned on channel Y. If there is an offset from the expected value, the LO initial tuning code needs to be adjusted to compensate for that offset, until it becomes null (or a value within ±40 ppm accuracy). The adjusted tuning code is C_{IF}. This means that, for determining with a good accuracy all 16 IEEE802.15.4 channels in the startup phase, at least one beacon has to be received on each channel, which ensures that the radio frequency accuracy meets the ±40 ppm accuracy requested by the standard.

This IF-based tuning of a crystal-free radio is one of the key techniques which allow low-accuracy low-frequency clocks to be used to track, and indeed generate, standards-compliant RF oscillations with an accuracy of 40 ppm. The nonlinear nature of an RF mixer means that a frequency difference Δf of 40 ppm at 2.4 GHz turns into the same Δf at 2.5 MHz, where it is now nearly 40,000 ppm of error relative to a 2.5 MHz clock. So the fractional frequency error in the 2.5 MHz measurement is divided by nearly a factor of one thousand when it shows up at 2.4 GHz.
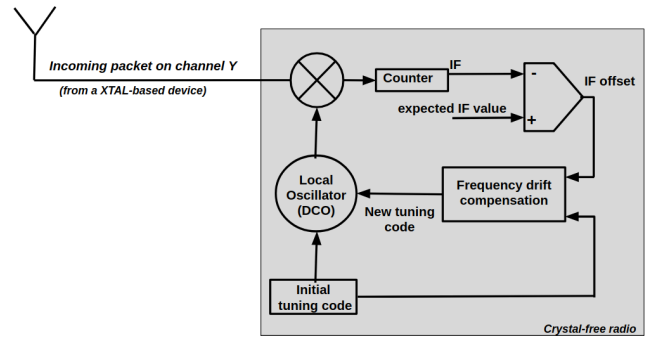

FIGURE 4: IF-based calibration scheme. The tuning code of the LO is continuously adjusted until the IF value reaches the expected value. This results in a LO frequency as close as possible to the center frequency of channel Y.

VI. STARTUP PHASE

In this section, we discuss two solutions for determining the tuning codes of the remaining IEEE802.15.4 channels at startup temperature, assuming that channel 11 is synthesized using tuning code B. For this purpose, we discuss two algorithms: one based on Recursive Least Squares (RLS), another based on Moving Average (MA). We analyze the initial frequency errors of synthesizing new channels (ϵ, before applying IF corrections) of the two approaches through simulations and experimental validation.

A. RLS CHANNEL TUNING

We propose the use of a Recursive Least Squares-based algorithm for determining the tuning codes of each IEEE802.15.4 channel at startup temperature. This algorithm outputs parameters a of a fit function y_{fit} that approximates f, the oscillator's tuning function. RLS is known to perform well when input data is provided sequentially [42], improving the fitting with every data point received. The RLS procedure is illustrated in Fig. 5.

The RLS-based algorithm requires as inputs the tuning code B for synthesizing channel 11, the approximate frequency resolution of a tuning code ΔF^{approx} and the forgetting factor λ. The algorithm is already set to fitting the measured frequency data on a second degree function:

$$
y_{fit}(x) = a_3 \times x^2 + a_2 \times x + a_1 \tag{8}
$$

where $y_{fit}(x)$ represents the synthesized frequency when using tuning code value x. The value of the forgetting factor λ defines the system memory and has an impact to the ability to track the changes in the input sequence and to the stability of the coefficients [43]:

$$
\lambda = \frac{N}{N+1} \tag{9}
$$

where N is the number of iterations before the effect of a disturbance has decayed to e^{-1} of its initial value [43].

The algorithm will predict the tuning code to be used for synthesizing each IEEE802.15.4 channel, from channel 12

RLS-based startup channel discovery(B,ΔF_approx,λ):

```
p = [0 1 2];
a = [0 0 0];
a = improve_offset(a, B, CH_freq);
Pm = eye(3);

for j = 1:16
    index = j;
    y_target = CH_freq(index);
    if y_target ∈ {channel_11, channel_12 or channel_13}
        x_predicted = linear_approximation(B, ΔF_approx);
    else
        x_predicted = roots([a(3) a(2 ) a(1)-y_target]);
        Process(x_predicted);
    end
    tuned_codes(index) = x_predicted;
    for i =1: 3
        x_m (:, i) = x_predicted.^p(i);
    end
    Km = Pm*x_m'*(λ+x_m*Pm*x_m')^(-1);
    Pm = (Pm - Km*xm*Pm)./λ;
    y_fit(index) = x_m*a;

    Wait_for_packet_reception();

    y_m = LO(x_predicted);
    ε(index)= y_m-y_fit(index);
    a = a + Km*ε(index);
    IF_correct_tuning_codes(tuned_code(index), ε);
end
```

Proposed tuning code for the target channel

Operations possible only after packet reception on the target IEEE802.15.4 channel

Update fitting parameters

FIGURE 5: RLS-based approach for consecutively determining the tuning code of each IEEE802.15.4 channel at startup temperature. At each step, a prediction of the tuning code to be used is made, *tuned_codes*. After a packet reception, the fitting function coefficients *a* are improved and IF correction is applied to the tuning code prediction.

to channel 26. The initial values of the coefficients *a* are null, but the a_1 term (the offset) guess can be improved (*improve_offset*), to speed up the fitting process:

$$a_1 \approx 2405 - B \times \Delta F^{approx} \text{ [MHz]} \qquad (10)$$

The tuning codes for the channels (11-known) 12 and 13 will be determined using an approximation function (*linear_approximation*), as expressed in equation (6), because the algorithm needs at least 3 (*m*) data points before fitting to a second (*m*−1) degree function. After this, the tuning code of target channel *j* is the solution ($x\_predicted = x$) of the fitting provided up to that point (j):

$$y\_target(j) = a_3(j) \times x^2 + a_2(j) \times x + a_1(j) \qquad (11)$$

where $y\_target(j)$ is the frequency of the target channel and j is the algorithm iteration ($j \in 1..16$, one iteration per channel). A problem with this approach is that the solution of a second degree polynomial is not unique. We tackled this issue in $Process()$ function, by providing positive (integer) solutions that fit with respect to the tuning codes of already discovered channels, provided the approximate frequency resolution of a code, ΔF^{approx}.

After a beacon is received on the target channel, the synthesized frequency can be measured,

$$y_m = LO(x\_predicted) = y\_target + IF\_offset \qquad (12)$$

as well as the error with respect to the target frequency, $\epsilon = IF\_offset$. This error will be further used to adjust the predicted tuning code (IF calibration) and to improve the fitting coefficients, *a*. At this point the algorithm continues

---

MA-based startup channel discovery(B,ΔF_approx,W):

```
CH_spacing(1) = round(5*1e6/ΔF_approx);
tuned_codes(1) = B;

for index = 2:16
    y_target = CH_freq(index);
    window = max(index-W,1):max(index-1,1);
    average_CH_spacing = round(mean(CH_spacing(window)));
    x_predicted = tuned_codes(index-1) + average_CH_spacing;
    tuned_codes(index) = x_predicted;

    Wait_for_packet_reception();

    y_m = LO(x_predicted);
    ε(index)= y_m-y_target;
    IF_correct_tuning_codes(tuned_code(index), ε);
    CH_spacing(index-1) = tuned_code(index) - tuned_code(index-1);
end
```

Proposed tuning code for the target channel

Operations possible only after packet reception on the target IEEE802.15.4 channel

FIGURE 6: MA-based approach for consecutively determining the tuning code of each IEEE802.15.4 channel at startup temperature. At each step, a tuning code to be used is proposed based on previous channel spacing, *tuned_codes*. After packet reception, IF correction is applied to the current tuning code.

by synthesizing the next target channel ($j = j + 1$). The algorithm ends when the tuning code needed for synthesizing all 16 channels are determined ($j = 16$, corresponding to channel 26).

B. MA CHANNEL TUNING

The Moving Average based algorithm (Figure 6) returns a value for the tuning code to be used for each IEEE802.15.4 channel, consecutively, having as inputs the tuning code value B (channel 11), the approximate frequency resolution of a tuning code ΔF^{approx} and the window size W. For determining the tuning code of a new channel, the algorithm will use the closest integer value to the average spacing between the channels inside the window W. After tuning on a channel using the proposed tuning code, the algorithm will wait for receiving a packet on that channel. As discussed in Section V, we assume that the approximation ΔF^{approx} is good enough so that the errors in synthesizing a channel are below 1MHz. If this is the case, a packet can be received on that channel and the IF offset can be computed.

After a packet reception, the error ϵ with respect to the channel's center frequency is computed and the tuning code value is adjusted, as well as the channel spacing between the current channel and the previous.

C. DISCUSSION

For evaluating the performance of the proposed algorithms in an ideal case we run the algorithms on simulation data. This data was created by smoothing the irregularities from experimentally collected frequency data for each tuning code, when the crystal-free radio was placed in a temperature chamber. The temperature was changed from 5°C to 55°C in 5°C steps. The performance of the algorithms with experimental data is discussed in Section VIII.

Figs. 7 and 8 show the evolution of ϵ (the frequency error before IF calibration) when using the RLS-based
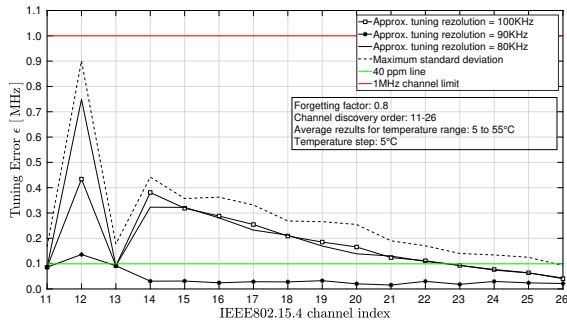
(a)
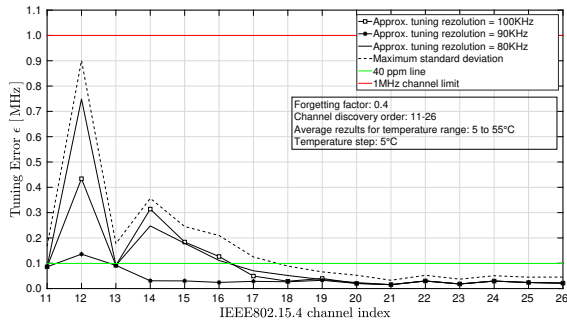
(b)

FIGURE 7: Startup phase: RLS algorithm evaluation for a) $\lambda = 0.8$ and b) $\lambda = 0.4$. Tuning error vs discovered channel for three approximations of ΔF^{approx}: 100kHz, 90KHz and 80 kHz. Performance averaged over startups at environmental temperature between $5 - 55°$ C.
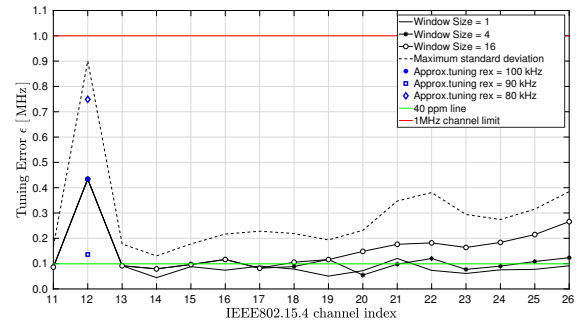

FIGURE 8: Startup phase: MA algorithm evaluation for window sizes of 1, 4 and 16. Tuning error vs discovered channel for three approximations of ΔF^{approx}: 100kHz, 90KHz and 80 kHz (blue markers). Performance averaged over startups at environmental temperature between $5-55°$C.

and MA-based algorithms respectively for discovering each IEEE802.15.4 channel at startup temperature. The algorithms were applied to startups at each of the evaluated temperatures and the average performance was plotted. The red line represents the 1 MHz error limit, above which the packets sent on a channel cannot be detected by the receiver. The green line represents the 40 ppm error, as imposed to the RF accuracy at all times, by the IEEE802.15.4 standard. All presented results represent the frequency synthesizing error before IF corrections.

For the RLS-based algorithm, the impact of the forgetting factor λ can be seen when comparing Figure 7a ($\lambda = 0.8$) to Figure 7b ($\lambda = 0.4$). A higher λ means that the system has a larger memory and so, the function coefficients are updated slower and the convergence rate of ϵ is slower. A lower value of λ gives more weight to recent data samples, improving the tracking ability of the algorithm. Still, when the approximation of ΔF^{approx} is closer to reality (see results for $\Delta F^{approx} = 90kHz$, as $\Delta F^{real} = 93kHz$), the value of λ does not impact the obtained results, as the algorithm starts with small initial error ϵ. For this case, starting with channel 13, all following channels can be synthesized with an error below 40 ppm. When startup errors are high (as it is the case for $\Delta F^{approx} = 80kHz$), a smaller λ is able to faster bring

the tuning errors below the 40 ppm limits imposed by the IEEE802.15.4 standard. For this case, we see in Figure 7b that channels following channel 16 are synthesized with very good accuracy, no matter the high startup errors (as long as they are below the red 1MHz limit line).

The Moving Average based algorithm is analyzed for window sizes of 1, 4 and 16, in Figure 8. For this algorithm, the initial approximation of the frequency resolution of a tuning code (ΔF^{approx}) only impacts the accuracy of synthesizing channel 12, as seen in Figure 8 with blue markers. After IF correction, the spacing between channel 11 and channel 12 is adjusted and used to determine the tuning code of the next channel, so ΔF^{approx} value has no further impact on the results. For determining the tuning code value of each new channel, the average of the past W channel spacing values is used. The simulation results in Figure 8 show that smaller window sizes W tend to approximate better the real channel spacing values and are able to synthesize new channels with an error $\epsilon \approx 40$ ppm. As for the RLS-based algorithm, the presented results are the average performance of the algorithm when applied to startups at temperatures between $5°$C and $55°$C, in $5°$C steps.

VII. NORMAL OPERATION PHASE
Once the tuning codes of all 16 IEEE802.15.4 channels are known at startup temperature, the crystal-free device enters the normal operation phase. In this phase, the device will communicate according to a dynamic communication schedule and it has to maintain the communication ability even if the environmental temperature evolves. Figure 9a shows the effect of environmental temperature on the communication accuracy (with experimental data). If the crystal-free radio is tuned on channel 13 for example, if the tuning code is not adjusted, the frequency will drift outside the channel. As seen in Figure 9a, for a $50°$C temperature difference, the synthesized frequency has drifted more than one communication channel (> 5MHz), to a value below channel 12.
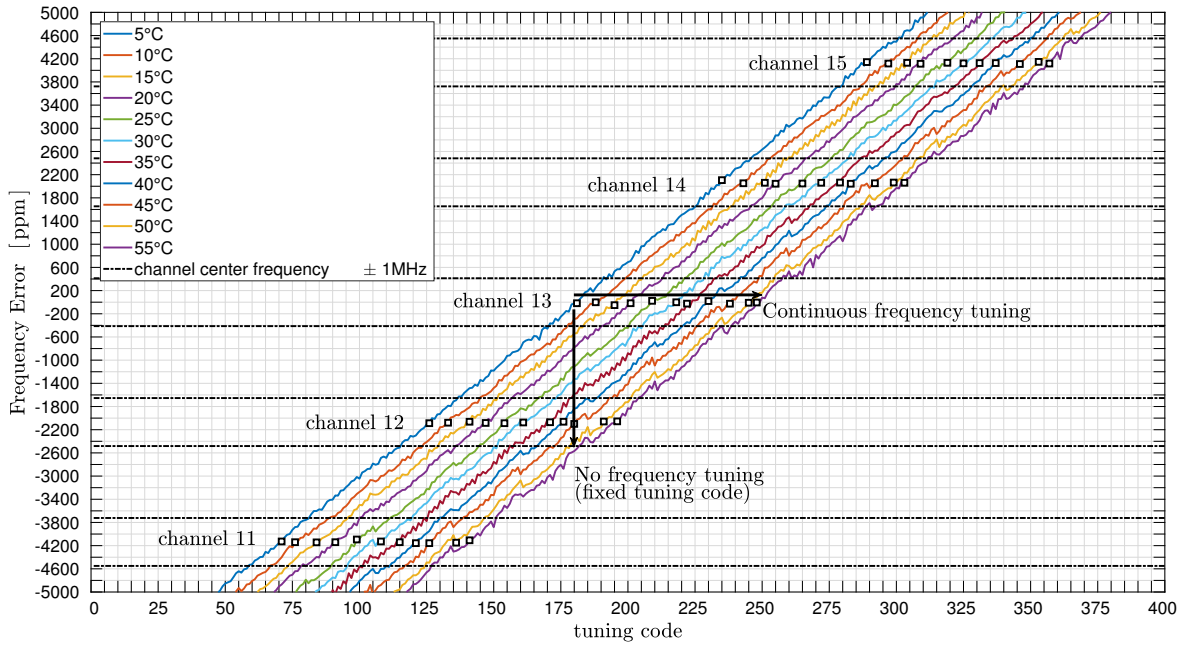
(a)
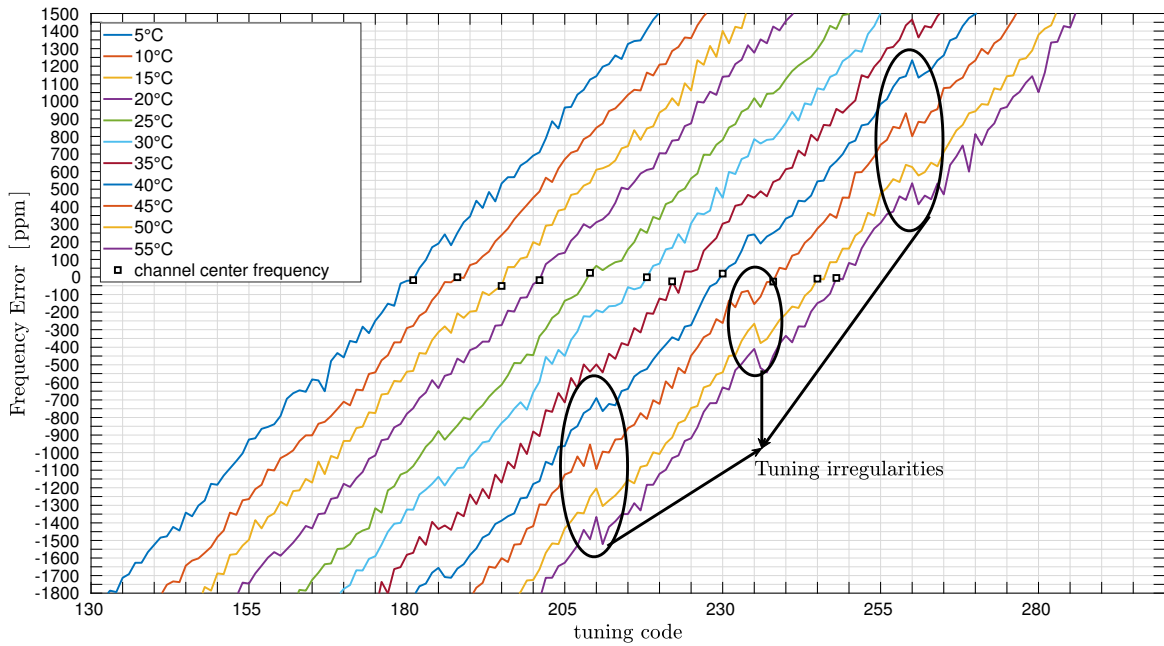
(b)

FIGURE 9: a) Radio tuned on the center frequency of channel 13. The temperature effect can be compensated by continuous adjusting the tuning codes of a channel. Experimental data. b) Zoom in: Output frequency irregularities on experimental data.

This section analyzes the ability of keeping communication accuracy on all 16 IEEE802.15.4 channels, when the environmental temperature changes in a range of 5 to 55°C. We assume that at each temperature, the crystal-free device will communicate on a random sequence of channels and that the temperature change rate is slow enough so that at least one channel can be accurately known (synthesized) as the temperature changes, by the means of IF-based calibration, described in Section V. The effectiveness of IF-based calibration for adjusting the tuning codes of a channel so as to maintain the synthesized frequency constant is detailed in our previous work, [23]. The challenge treated in this section is tuning to the remaining channels in random order (as the communication schedule will require) with an as-low-as-possible frequency error, at any temperature, when using techniques based on RLS and MA, respectively.

A. RLS CHANNEL TUNING

The RLS-based algorithm in normal operation mode makes use of the tuning function approximation y_{fit} determined at startup temperature T_0 (Section VI), and of the tuning code value U of any channel Y at the new environmental temperature, T_1, expressed by (13).

$$y_{fit}(x, T_0) = a_3 \times x^2 + a_2 \times x + a_1$$
$$x = U : \text{tuning code for channel Y at } T_1 \tag{13}$$

The tuning function is further adapted for the new environmental temperature, T_1, using (14).

$$y_{fit}(x, T_1) = a_3 \times x^2 + a_2 \times x + a_{1T_1}$$
$$a_{1T_1} \approx a_1 - f\_shift \tag{14}$$
$$f\_shift = y_{fit}(U, T_0) - F_Y$$

Where a_{1T_1} represents the new coefficient approximation at temperature T_1, F_Y represents the frequency value of channel Y, and f_shift is an approximation of the frequency shift caused by T_1 with respect to the function approximation at T_0.

The new function approximation $y_{fit}(x, T_1)$ and its coefficients a_3, a_2 and a_{1T_1} are used by the RLS-based algorithm (Fig. 5) to predict the new tuning codes to be used by any channel demanded by the communication schedule (with frequency value y_{target}), in any order. Coefficients a_3 and a_2 are updated by the algorithm to the values corresponding to the new environmental temperature, a_{3T_1} and a_{2T_1}, respectively. The tuning function for T_1 thereby changes as expressed in (15):

$$y_{fit}(x, T_1) = a_{3T_1} \times x^2 + a_{2T_1} \times x + a_{1T_1} \tag{15}$$

The same technique is used as the temperature evolves from T_1 to T_2 and to any other temperature. As a good function approximation $y_{fit}(x, T_0)$ is provided after the startup phase, the impact of the forgetting factor λ is limited.
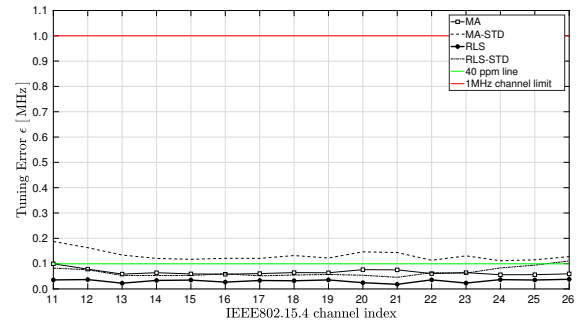

FIGURE 10: Normal operation. Monte Carlo simulation over the performance of RLS and MA based algorithms when synthesizing channels in random order as the environmental temperature slowly changes from 5 to 55°C.

B. MA CHANNEL TUNING

The MA-based algorithm enters the normal operation phase with the complete knowledge of vector $tuned\_codes(channel, T_0)$, which holds the values of the tuning codes for synthesizing each IEEE802.15.4 channel at startup temperature T_0. As for the RLS case, by the means of IF calibration, the tuning code value U for synthesizing channel Y at T_1 is known and is stored as $tuned\_codes(Y, T_1)$. For channel Y, the shift in tuning code caused by the temperature changing from T_0 to T_1 is expressed by (16).

$$shift = tuned\_codes(Y, T_0) - tuned\_codes(Y, T_1) \tag{16}$$

The algorithm detailed in Fig. 6 is still valid, but in normal operation mode, the variables inside the averaging window W are shifts in tuning code caused by the changing temperature, and not the channel spacing as in startup phase. The window contains the shifts undergone by the closest W known channels to the channel of interest, y_target. The tuning code predicted by the MA algorithm for a channel W at T_1 is expressed by (17).

$$tuned\_codes(W, T_1) = tuned\_codes(W, T_0) + avg\_shift\_W \tag{17}$$

C. DISCUSSION

We evaluate the performance of both algorithms by simulating a change in temperature from 5°C to 55°C, in steps of 5°C, using the simulation data described in Section VI-C. At each temperature value, all channels are synthesized in random order, as dictated by the communication schedule. As in the previous sections, the error ϵ represents the error in the synthesized frequency with respect to the frequency value of the channel of interest (before IF-correction). The IEEE802.15.4 standard mandates this to be below 40 ppm at all times.

Fig. 10 shows the average performance of the RLS and MA based algorithms in normal operation mode, obtained with

Monte Carlo simulation over the communication schedule at each temperature. In the figure's legend, the notation "STD" stands for "standard deviation" of the average performance of each algorithm. We see that the average frequency synthesizing error ϵ obtained with the proposed algorithms is below the 40 ppm limit for each channel. Still, for the RLS algorithm, the standard deviation of the average performance is also below 40 ppm, which is not the case for the MA algorithm.

VIII. EXPERIMENTAL DATA IRREGULARITIES

We place the crystal-free platform in a temperature chamber and vary the temperature between 5°C and 55°C, in 5°C increments. When measuring the output frequency returned by each tuning code of the crystal free platform, at each temperature, we obtain plots which are not perfectly smooth, as shown in Fig. 9b. These irregularities are the combined result of unstable temperature, limited frequency counter accuracy, oscillator phase noise and tuning function limitations (fine tuning with 5-bit codes). The immediate consequence of these irregularities is that ϵ errors are higher when applying the RLS and MA based algorithms to this data.

For the RLS-based algorithm in Fig. 7, the tuning errors are well below the 40 ppm threshold. For the startup phase with experimental data (Figs. 11a and 11b) the obtained tuning errors evolve close to this limit, with high standard deviations caused by the irregularities. Higher λ values are equivalent to less influence of new input disturbances. In Fig. 11a, the standard deviations of the error has a decreasing trend when compared to Fig. 11b. The same happens for the MA-based algorithm (Fig. 11c), that shows reduced errors for higher W values. Still, the errors evolve slightly above the green line, with higher standard deviations than the ones obtained with smoothed simulation data (Fig. 8).

In normal operation phase, the environmental temperature changes and the channel tuning needs to be adjusted with it. Figure 12 shows the average tuning error ϵ obtained with experimental data, when using the RLS-based and MA-based algorithms. Even if the average error lies on the 40ppm line, the obtained accuracy and standard deviations are worse than for the simulation data showed in Figure 10. The standard deviation of the error obtained with the RLS algorithm tends to be worse than that of the MA algorithm for experimental data. A cause for this may be that the RLS based algorithm continuously tries to fit data to a second degree function and the irregularities shown in Figure 9b (that worsen with the temperature) lead to fitting on a function different from the real tuning function.

## IX. ENABLING PACKET TRANSMISSION WITH CRYSTAL-FREE RADIOS

This section characterizes the behaviour of the RF oscillator in transmission (TX) mode and describes the steps to follow in order to be able to reliably transmit packets on IEEE802.15.4 channels using a crystal-free radio. From the best of our knowledge, this work is the first one to address this challenge.
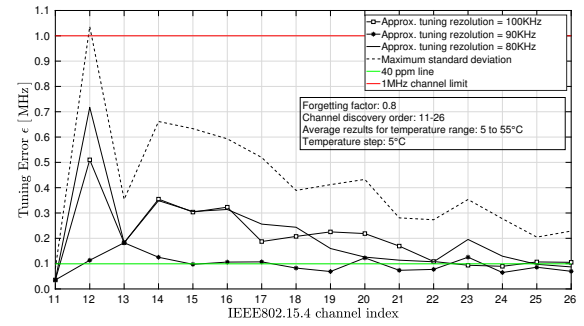
(a)
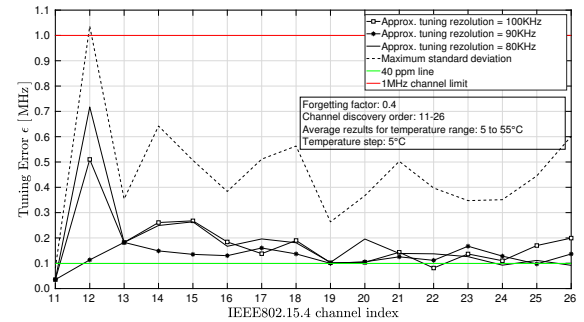
(b)
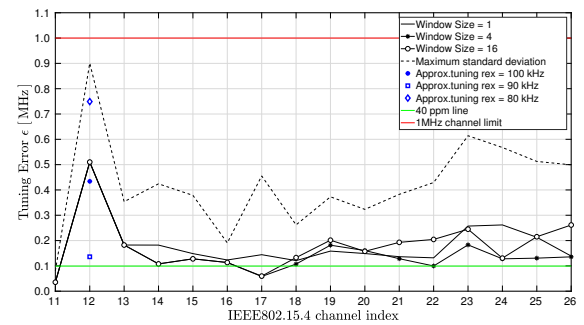
(c)

FIGURE 11: Tuning error vs discovered channel for three approximations of ΔF^{approx}: 100kHz, 90KHz and 80 kHz. Performance averaged over startups at environmental temperature between $5-55°$ C. Startup phase on measurement data: a) RLS algorithm evaluation for $\lambda = 0.8$; b) RLS algorithm evaluation for $\lambda = 0.4$; c) MA algorithm evaluation for window sizes of 1, 4 and 16.

A. CHALLENGES

Up until this point in the paper we treated techniques to initially calibrate the RF oscillator and the remaining on-chip oscillators of a crystal-free radio, followed by methods for synthesizing the IEEE 802.15.4 channels in both constant and variable temperature environment. During all these phases, the crystal-free radio was set to reception (RX) mode and RF oscillator tuning corrections were possible while measuring the IF offset during reception of network beacons. The crystal-free radio is able to store the tuning codes needed
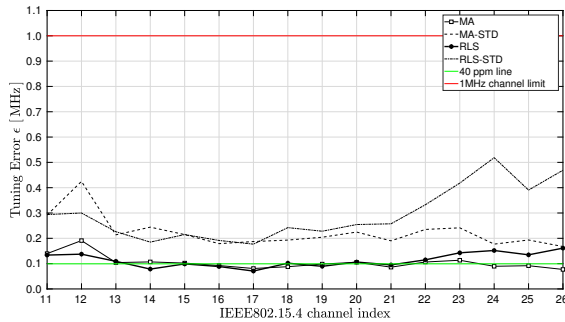
FIGURE 12: Normal operation on measurement data. Monte Carlo simulation over the performance of RLS and MA based algorithms when synthesizing channels in random order as the environmental temperature slowly changes from 5 to 55°C.

for being able to receive on any of the 16 IEEE802.15.4 channels.

However, when switching to TX mode, the tuning codes used for synthesizing channels in RX mode are not of use anymore. As different circuitry is involved when transmitting data, the same RF oscillator in TX mode behaves differently than in RX mode. This is best shown in Figure 13, where the tuning codes needed for synthesizing channels 11 to 26 when the RF oscillator is in RX mode and then TX mode are plotted (experimental data). Knowing the tuning code difference when switching from RX mode to TX mode for channel 11 for example (50 codes in Figure 13a), cannot be extrapolated to the remaining channels, as this difference tends to increase with frequency (75 codes for channel 26 in Figure 13a). This difference caused by switching from RX to TX mode is also influenced by environmental temperature. We can see in Figure 13b how at $50°C$ the tuning code difference that needs to be applied when switching from RX mode to TX mode for synthesizing the same channel is now smaller for all channels when compared to Figure 13a: only 28 codes for channel 11 and 55 codes for channel 26.

As a generalization for the difference in tuning codes for synthesizing the same channel in RX mode and then TX mode, we plotted in Figure 14a this trend across channels 11-26, accumulated over a 60 °C temperature change. The RX mode - TX mode tuning code difference is higher for higher frequency channels.

Figure 14b shows the sum of differences (RX mode - TX mode) in tuning codes for all 16 channels and how it evolves with temperature. We see that as the temperature increases, the RX mode - TX mode gap decreases for all channels, as also noticed in Figure 13b.

B. APPROACH

Because the RF oscillator in TX mode is characterized by a tuning function with different coefficients than in RX mode, both with frequency and temperature, a similar channel dis-

covery as done in RX mode is advisable (Sections VI - VII), with a few differences:

1) the RF oscillator needs to be calibrated in RX mode (Startup phase completed) so as to ensure Acknowledgement (ACK) reception in case of successful transmission attempt (see Figure 15).
2) a good starting value for ΔF^{approx} is already provided by analyzing the channel spacing in RX mode (specifically, for RLS, the a_2 term returned after running the algorithm and for MA the average of $CH\_spacing$ vector).
3) the tuning code B^{TX} corresponding to channel 11 in TX mode is to be found using a calibrated on-chip frequency reference.
4) further corrections to be applied to the frequency synthesized using the tuning code B^{TX} in TX mode are based on the frequency offset (signed) value included in the ACK received from the destination device (replacing the function $IF\_correct\_tuning\_codes$ in Section VI).

We see that the above mentioned differences impact only the initialization of the Startup phase for TX mode. The Normal Operation phase is as described in Section VII for RX mode, but with using the frequency offset information included in ACKs for frequency corrections, and not the IF value (available only at packet reception - RX mode). For these reasons, we will include in this section the results of the algorithms running on simulation data and measurement data for the Startup Phase only. We will see that these results match the ones obtained for RX mode, for the case of good approximations of ΔF^{approx} (see Section VI).

The flowchart represented in Figure 15 places the right moment to start calibrating the RF oscillator for transmission to be only after the on-chip oscillators are calibrated at current temperature and the device is able to receive network beacons and acknowledgements.

In order to determine the tuning code B^{TX} corresponding to channel 11 when the RF oscillator is in TX mode the following approximations are used:

$$f^{TX\_mode(B)} \approx \frac{ticks_{RF\_oscillator}}{ticks\_reference} \times f_{reference} [\text{Hz}]$$
$$B^{TX} \approx B - \frac{f^{TX\_mode(B)} - 2405 \, [\text{MHz}]}{\Delta F^{approx}} \qquad (18)$$

where $f^{TX\_mode(B)}$ is an approximation of the frequency output of the RF oscillator in TX mode when using tuning code B. The accuracy of this approximation is directly influenced by the accuracy of the on-chip reference used, $f_{reference}$. The tuning code B^{TX} to be used for attempting transmission on channel 11 is determined by adjusting the tuning code B so that $f^{TX\_mode(B)}$ matches the frequency of channel 11 (2405MHz).

Figure 16 shows the simulation results for the average frequency error when synthesizing the 16 IEEE802.15.4 channels with the RF oscillator in TX mode (Startup Phase). Even
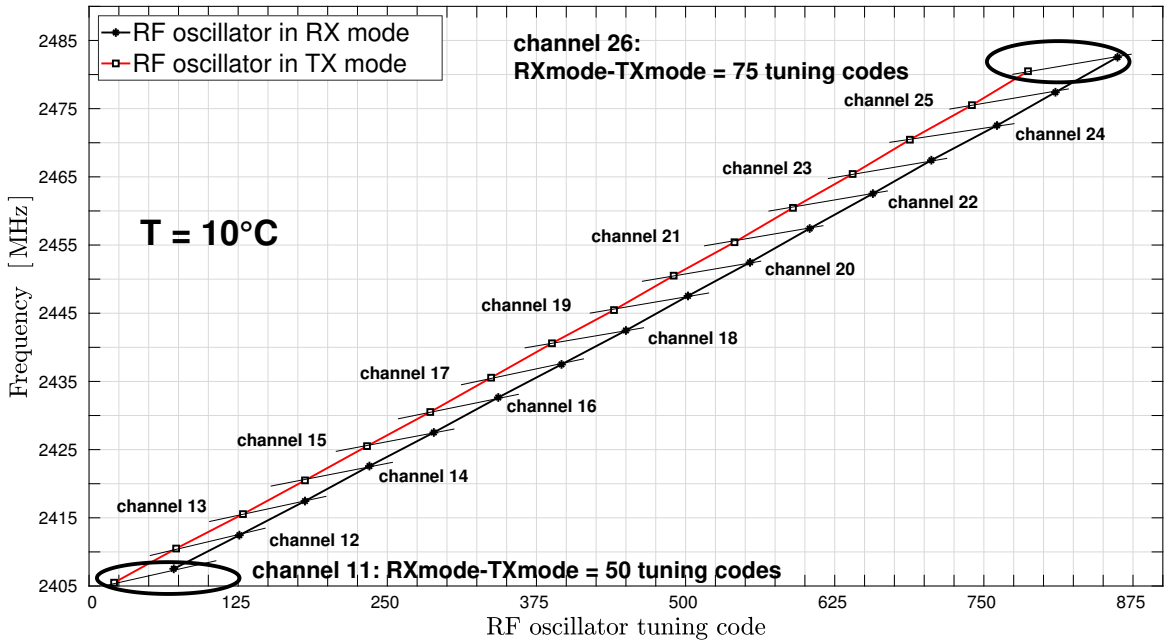
(a)
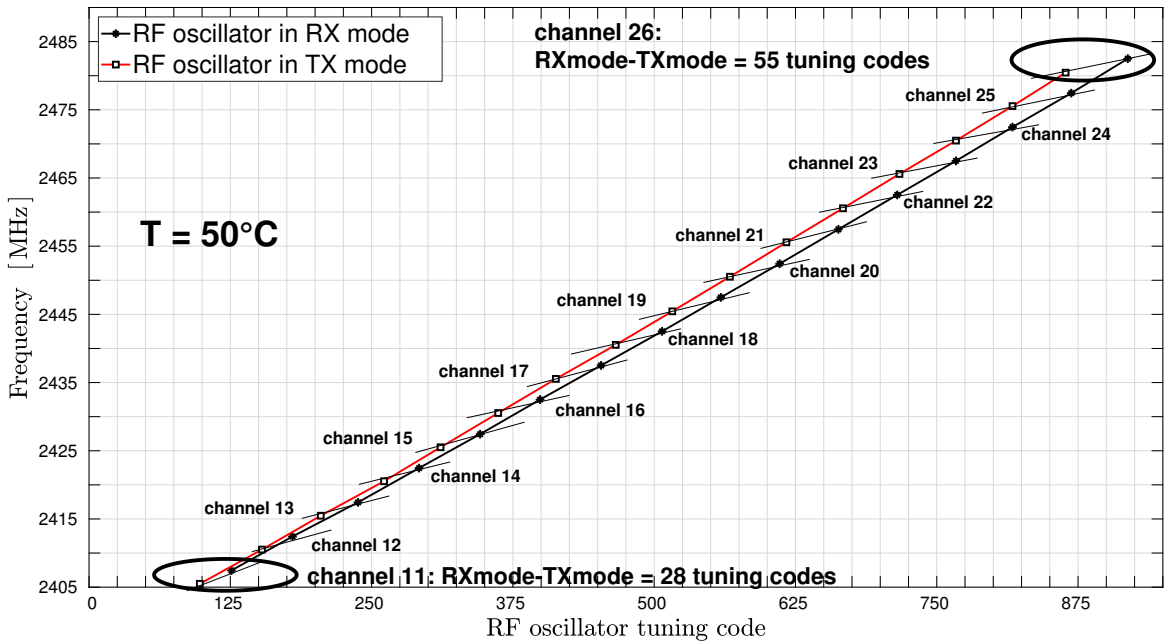
(b)

FIGURE 13: Crystal-free radio: RF oscillator tuning codes for synthesizing the 16 IEEE802.15.4 channels in RX mode and TX mode at a) 10° C and b) 50°C. Experimental data.
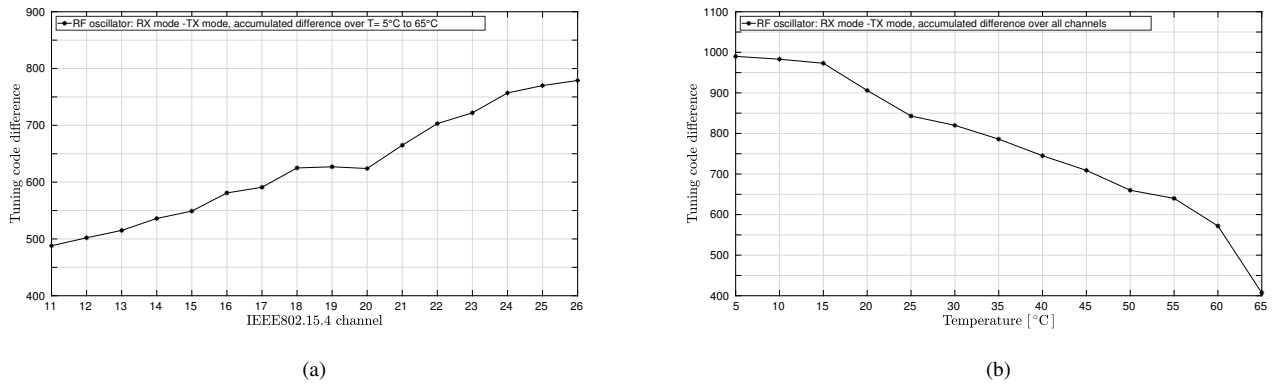
(a)

(b)

FIGURE 14: Crystal-free radio, RF oscillator, experimental data. a) For higher frequency channels, the tuning code difference needed when switching from RX mode to TX mode is higher than for lower frequency channels, at all temperatures. b) The trend with temperature of the aforementioned switching difference.
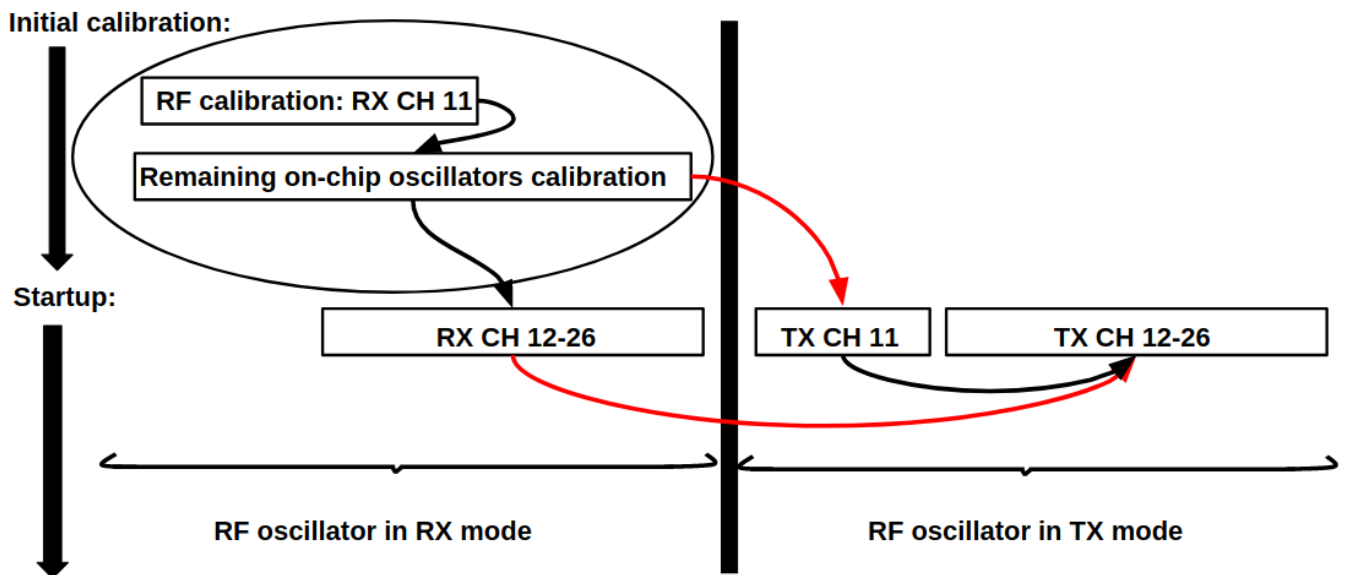

FIGURE 15: Crystal-free radio: determining the tuning codes for transmission on channels 11-26 is possible after the Initial Calibration phase has been completed and ACK reception is possible (RF oscillator calibrated in Startup Phase for RX mode).
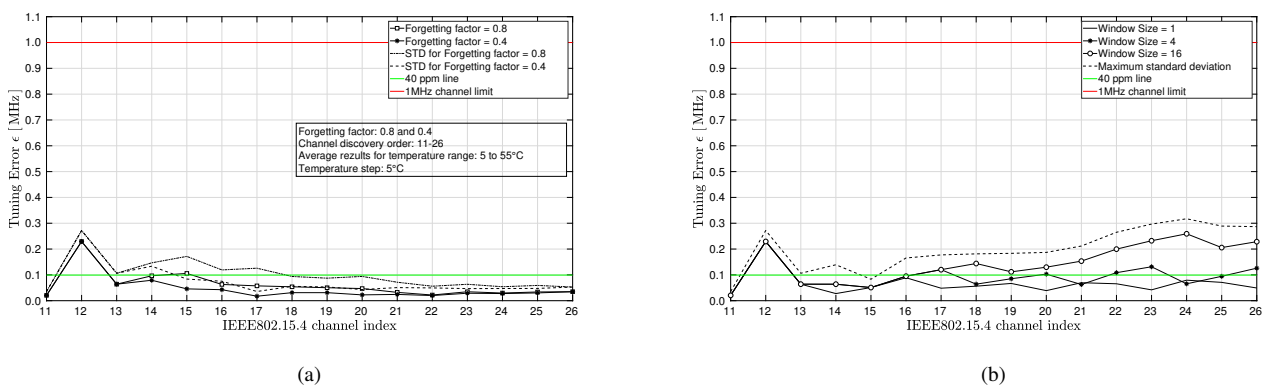

(a)

(b)

FIGURE 16: Startup phase applied to RF oscillator in TX mode (simulation data): a) RLS algorithm evaluation for $\lambda = 0.8$ and $\lambda = 0.4$ and b) MA algorithm for window sizes of 1, 4 and 16. Performance averaged over startups at environmental temperature between $5 - 55°C$.
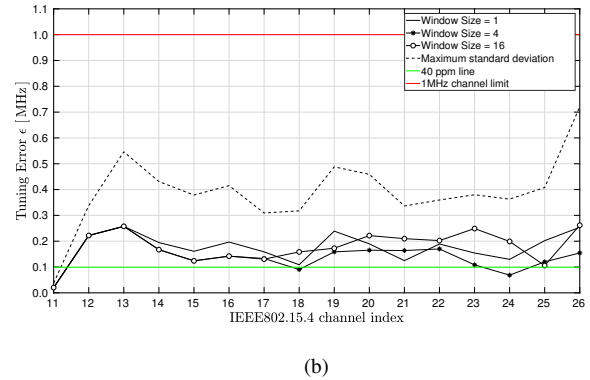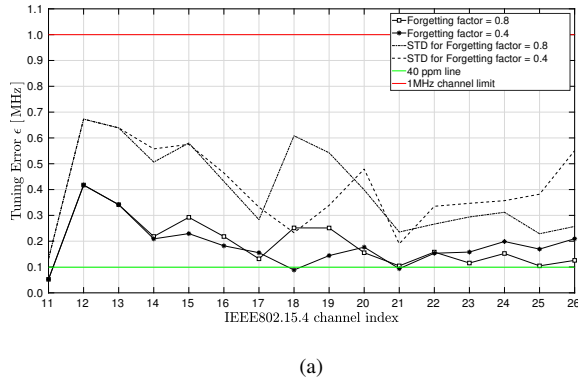
(a)

(b)

FIGURE 17: Startup phase applied to RF oscillator in TX mode (measurement data): a) RLS algorithm evaluation for $\lambda = 0.8$ and $\lambda = 0.4$ and b) MA algorithm for window sizes of 1, 4 and 16. Performance averaged over startups at environmental temperature between $5 - 55°C$.

if in TX mode the RF oscillator is characterized by different function coefficients and a different frequency resolution of a tuning code, using ΔF^{approx} from the RX mode is a good startup approximation. While the results returned by the RLS and MA algorithms applied to simulation data are promising, with errors dropping below 40 ppm, when applying them on measurement data, the average frequency error is above the green 40 ppm line (Figure 17). The average performance and the standard deviations obtained match the ones obtained for RX mode and detailed in Section VIII.

X. CONCLUSIONS

This article addresses the challenge of RF local oscillator calibration on a single chip, crystal-free, IEEE802.15.4 device. It presents a full solution for compensating the inherent drift of the internal oscillators, and making it possible for the IEEE802.15.4 device to tune its radio at the specific channels, keeping the drift boundaries at the values imposed by the standard. The proposed method is divided in three phases: initial calibration, startup phase and normal operation phase. The on-chip oscillator settings for synthesizing the 16 IEEE802.15.4 communication channels are discovered at a constant temperature during the startup phase, for which we evaluate the use of RLS or MA-based algorithms. Once all the communication channels settings are determined and stored, the device switches to normal operation phase. This phase deals with maintaining the communication accuracy on all channels taking into account temperature variation and other inherent clock drifts.

The proposed algorithms are evaluated through both simulations and experimentation. While the obtained accuracy on simulation data satisfies the 40 ppm accuracy required by the IEEE802.15.4 standard, the results obtained on experimental data show higher errors, performing in average close to the 40 ppm limit. The higher errors are the consequence of irregularities in the tuning function across frequencies and increasing temperatures.

It is important to keep in mind that these frequency errors

(obtained during startup phase or normal operation phase) impact the accuracy with which a channel will be synthesized for the first time at a certain temperature and will be immediately corrected to be below 40 ppm after the reception of a beacon (for IF-based frequency correction in RX mode) or of an ACK (for frequency correction in TX mode).

REFERENCES

[1] D. Shaviv, A. Ozgur, and A. Arbabian, "Communication With Crystal-Free Radios," IEEE Transactions on Communications, vol. 66, no. 10, pp. 4513–4520, Oct 2018.

[2] O. Khan, D. Burnett, F. Maksimovic, B. Wheeler, S. Mesri, A. Sundararajan, B. L. Zhou, A. Niknejad, and K. Pister, "Time keeping ability of crystal free radios," IEEE Internet of Things Journal, pp. 1–1, 2018.

[3] Texas Instruments, "Crystal Or Crystal Oscillator Replacement with Silicon Devices," application Report SNAA217 - July 2014.

[4] C. Lam, "A review of the recent development of MEMS and crystal oscillators and their impacts on the frequency control products industry," 2008 IEEE Ultrasonics Symposium, pp. 694–704, 2008.

[5] B. Wheeler, F. Maksimovic, N. Baniasadi, S. Mesri, O. Khan, D. Burnett, A. Niknejad, and K. Pister, "Crystal-free narrow-band radios for low-cost IoT," in 2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC), June 2017, pp. 228–231.

[6] Song Li (Microchip Technology), "Revolutionary Timing for Auto-Qualified MEMS Oscillators," https://www.sensorsmag.com/components/revolutionary-timing-for-auto-qualified-mems-oscillators, [Online, accessed April 2, 2019].

[7] Epson, "Comparison of Crystal Oscillator and Si-MEMS Oscillators," https://www5.epsondevice.com/en/information/technical_info/pdf/wp_e20140911_osc.pdf, [Online, accessed April 2, 2019].

[8] Paul Pickering (Mouser Electronics), "MEMS Oscillators Make Inroads," https://eu.mouser.com/applications/timer-mems-oscillators/, [Online, accessed April 3, 2019].

[9] SiTime, "MEMS Oscillators: Enabling Smaller, Lower Power IoT& Wearables," https://www.sitime.com/sites/default/files/_legacy/images/stories/applications/SiTimes-MEMS-Oscillators-Lower-Power--Reduce-Size-2016.pdf, [Online, accessed April 2, 2019].

[10] "Texas Instruments," http://www.ti.com/, [Online, accessed April 5, 2019].

[11] "Texas Instruments: CC2652RB (PREVIEW) SimpleLink crystal-less BAW wireless MCU ," http://www.ti.com/product/CC2652RB, [Online, accessed April 5, 2019].

[12] Baker Lawley (All About Circuits), "The End of the Crystal? TI Introduces Two New Products Using Breakthrough BAW Resonator Technology," https://www.allaboutcircuits.com/news/ti-introduces-two-new-products-using-breakthrough-baw-resonator-technology/, [Online, accessed April 5, 2019].

[13] A. Paidimarri, N. Ickes, and A. P. Chandrakasan, "A 0.68V 0.68mW 2.4GHz PLL for ultra-low power RF systems," in 2015 IEEE Radio

Frequency Integrated Circuits Symposium (RFIC), May 2015, pp. 397–400.

[14] A. Y. Jou, H. Shan, H. Pajouhi, J. Peterson, and S. Mohammadi, "A Single-Chip Wireless Powered RFID Antenna and Transceiver," IEEE Journal of Radio Frequency Identification, vol. 1, no. 3, pp. 219–227, Sep. 2017.

[15] H. Bhamra, Y. Kim, J. Joseph, J. Lynch, O. Z. Gall, H. Mei, C. Meng, J. Tsai, and P. Irazoqui, "A24 μW, Batteryless, Crystal-free, Multinode Synchronized SoC "Bionode" for Wireless Prosthesis Control," IEEE Journal of Solid-State Circuits, vol. 50, no. 11, pp. 2714–2727, Nov 2015.

[16] M. Tabesh, N. Dolatsha, A. Arbabian, and A. M. Niknejad, "A Power-Harvesting Pad-Less Millimeter-Sized Radio," IEEE Journal of Solid-State Circuits, vol. 50, no. 4, pp. 962–977, April 2015.

[17] L. Chuo, Y. Shi, Z. Luo, N. Chiotellis, Z. Foo, G. Kim, Y. Kim, A. Grbic, D. Wentzloff, H. Kim, and D. Blaauw, "A 915MHz asymmetric radio using Q-enhanced amplifier for a fully integrated $3 \times 3 \times 3mm^2$ wireless sensor node with 20m non-line-of-sight communication," in 2017 IEEE International Solid-State Circuits Conference (ISSCC), Feb 2017, pp. 132–133.

[18] S. Mesri, "Design and user guide for the single chip mote digital system," Master's thesis, EECS Department, University of California, Berkeley, May 2016.

[19] F. Maksimovic, B. Wheeler, D. Burnett, O. Khan, S. Mesri, I. Suciu, L. Lee, A. Moreno, A. Sundararajan, B. Zhou, R. Zoll, A. Ng, T. Chang, X. Vilajosana, T. Watteyne, A. Niknejad, and K. Pister, "A Crystal-Free Single-Chip Micro Mote with Integrated 802.15.4 Compatible Transceiver, Sub-mW BLE Compatible Beacon Transmitter, and Cortex M0," in 2019 Symposium on VLSI Circuits, 2019.

[20] Dust Networks, "LTC5800-IPM:SmartMesh IP Node 2.4GHz, 802.15.4e Wireless Mote-on-Chip," https://www.analog.com/media/en/technical-documentation/data-sheets/5800ipmfa.pdf, [Online, accessed April 3, 2019].

[21] IEEE, "Ieee standard for low-rate wireless networks," IEEE, Tech. Rep., April 2016.

[22] O. Khan, B. Wheeler, D. Burnett, F. Maksimovic, S. Mesri, K. Pister, and A. Niknejad, "Frequency reference for crystal free radio," in 2016 IEEE International Frequency Control Symposium (IFCS), May 2016, pp. 1–2.

[23] I. Suciu, F. Maksimovic, D. Burnett, O. Khan, B. Wheeler, A. Sundararajan, T. Watteyne, X. Vilajosana, and K. Pister, "Experimental clock calibration on a Crystal-Free Mote-on-a-Chip," in 2019 IEEE INFOCOM WKSHPS: CNERT 2019: Computer and Networking Experimental Research using Testbeds (INFOCOM 2019 WKSHPS - CNERT 2019), Paris, France, Apr. 2019.

[24] L. Peng, Y. Cao, X. Pan, and X. Zhao, "A low-power relaxation oscillator with improved thermal stability," in 2017 International SoC Design Conference (ISOCC), Nov 2017, pp. 115–116.

[25] S. Sakphrom, T. Limpiti, N. Wichaipanich, and A. Thanachayanont, "Sub-30$ppm/^{\circ}C$ High-frequency Temperature-compensated CMOS Relaxation Oscillator," in 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), July 2018, pp. 305–308.

[26] D. Cherniak, R. Nonis, and F. Padovan, "A precision 140MHz relaxation oscillator in 40nm CMOS with $28ppm/^{\circ}C$ frequency stability for automotive SoC applications," in 2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC), June 2017, pp. 57–60.

[27] H. V. Nguyen, J. Kim, and S. Kim, "Temperature And Supply Voltage Insensitive Relaxation Oscillator Using Voltage Reference," in TENCON 2018 - 2018 IEEE Region 10 Conference, Oct 2018, pp. 0555–0558.

[28] D. Burnett, B. Wheeler, L. Lee, F. Maksimovic, A. Sundararajan, O. Khan, and K. Pister, "CMOS oscillators to satisfy 802.15.4 and bluetooth LE PHY specifications without a crystal reference," in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (IEEE CCWC 2019), Las Vegas NV, USA, Jan. 2019.

[29] R. B. Staszewski, P. Cruise, and D. Leipold, "Crystal Drift Compensation in a Mobile Phone," in 2006 IET Irish Signals and Systems Conference, June 2006, pp. 241–245.

[30] J. Bae, K. Song, H. Lee, H. Cho, and H. Yoo, "A Low-Energy Crystal-Less Double-FSK Sensor Node Transceiver for Wireless Body-Area Network," IEEE Journal of Solid-State Circuits, vol. 47, no. 11, pp. 2678–2692, Nov 2012.

[31] D. Griffith, P. T. Roine, T. Kallerud, B. Goodlin, Z. Hughes, and E. T. Yen, "A $\pm$10ppm -40 to 125°C BAW-based frequency reference system for crystal-less wireless sensor nodes," in 2017 IEEE International Symposium on Circuits and Systems (ISCAS), May 2017, pp. 1–4.

[32] X. Vilajosana, P. Tuset-Peiro, T. Watteyne, and K. Pister, "Openmote: Open-source prototyping platform for the industrial iot," in International Conference on Ad Hoc Networks (AdHocNets), September 2015.

[33] K. Wang, J. Koo, R. Ruby, and B. Otis, "21.7 A 1.8mW PLL-free channelized 2.4GHz ZigBee receiver utilizing fixed-LO temperature-compensated FBAR resonator," in 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), Feb 2014, pp. 372–373.

[34] L. Chuo, Y. Shi, Z. Luo, N. Chiotellis, Z. Foo, G. Kim, Y. Kim, A. Grbic, D. Wentzloff, H. Kim, and D. Blaauw, "A 915MHz asymmetric radio using Q-enhanced amplifier for a fully integrated $3 \times 3 \times 3mm^3$ wireless sensor node with 20m non-line-of-sight communication," in 2017 IEEE International Solid-State Circuits Conference (ISSCC), Feb 2017, pp. 132–133.

[35] Y. Tu, J. Liu, K. Cheng, and C. Chang, "Low supply voltage and multiphase all-digital crystal-less clock generator," IET Circuits, Devices Systems, vol. 12, no. 6, pp. 720–725, 2018.

[36] H. Foundation, "IEC 62591:2016 Industrial networks - Wireless communication network and communication profiles - WirelessHART," HART Foundation, Tech. Rep., 2016.

[37] P. Thubert, T. Watteyne, R. Struik, and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-6tisch-architecture-08, 2015.

[38] X. Vilajosana, T. Watteyne, M. Vucinic, T.Chang, and K. Pister, "6TiSCH: Industrial Performance for IPv6 Internet of Things Networks," Special issue Real-Time Networks and Protocols for Factory Automation and Process Control Systems. IEEE Proceedings Journal, 2019.

[39] M. Vučinić, J. Simon, K. Pister, and M. Richardson, "Minimal Security Framework for 6TiSCH," Internet Engineering Task Force, Internet-Draft draft-ietf-6tisch-minimal-security-10, Apr. 2019, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-minimal-security-10

[40] N. Salman, I. Rasool, and A. H. Kemp, "Overview of the IEEE 802.15.4 standards family for Low Rate Wireless Personal Area Networks," in Symposium on Wireless Communication Systems, Sep. 2010, pp. 701–705.

[41] "LC Oscillator Basics," https://www.electronics-tutorials.ws/oscillator/oscillators.html, [Online, accessed May 20, 2019].

[42] H. Gavin, "Fitting Models to Data, Generalized Linear Least Squares, and Error Analysis," CEE 629. System Identification, Duke University, Spring 2019, 2019.

[43] MIT, "Introduction to Recursive-Least-Squares (RLS) Adaptive Filters," Massachusetts Institute of Technology, Department of Mechanical Engineering, 2.161 Signal Processing - Continuous and Discrete, 2008.

• • •