Evolving Hardware Instinctive Behaviors in Resource-scarce Agent Swarms Exploring Hard-to-reach Environments

Martin Andraud
ESAT - MICAS, Microelectronics and Sensors
Kasteelpark Arenberg 10 - box 2443
Leuven, Belgium
martin.andraud@kuleuven.be

Ahmed Hallawa
Chair of Integrated Signal Processing Systems
Kopernikusstraße 16
Aachen, Germany
hallawa@ice.rwth-aachen.de

Jaro De Roose
ESAT - MICAS, Microelectronics and Sensors
Kasteelpark Arenberg 10 - box 2443
Leuven, Belgium
jaro.deroose@kuleuven.be

Eugenio Cantatore
TU Eindhoven, MSM group
Groene loper
Eindhoven, The Netherlands
e.cantatore@tue.nl

Gerd Ascheid
Chair of Integrated Signal Processing Systems
Kopernikusstraße 16
Aachen, Germany
ascheid@ice.rwth-aachen.de

Marian Verhelst
ESAT - MICAS, Microelectronics and Sensors
Kasteelpark Arenberg 10 - box 2443
Leuven, Belgium
marian.verhelst@esat.kuleuven.be

ABSTRACT

This work introduces a novel adaptation framework to energy-efficiently adapt small-sized circuits operating under scarce resources in dynamic environments, as autonomous swarm of sensory agents. This framework makes it possible to optimally configure the circuit based on three key mechanisms: (a) an off-line optimization phase relying on R2 indicator based Evolutionary Multi-objective Optimization Algorithm (EMOA), (b) an on-line phase based on hardware instincts and (c) the possibility to include the environment in the optimization loop. Specifically, the evolutionary algorithm is able to simultaneously determine an optimal combination of static settings and dynamic instinct for the hardware, considering highly dynamic environments. The instinct is then run on-line with minimal on-chip resources so that the circuit efficiently react to environmental changes. This framework is demonstrated on an ultrasonic communication system between energy-scarce wireless nodes. The proposed approach is environment-adaptive and enables power savings up to 45% for the same performance on the considered case studies.

CCS CONCEPTS

•Computing methodologies → Evolvable hardware; •Hardware → Sensors and actuators; *Wireless integrated network sensors;*

KEYWORDS

Evolutionary multi-objective optimization, instinct evolution, swarm intelligence, wireless sensors

GECCO '18 Companion, Kyoto, Japan

DOI: 10.1145/3205651.3208255

ACM Reference format:
Martin Andraud, Ahmed Hallawa, Jaro De Roose, Eugenio Cantatore, Gerd Ascheid, and Marian Verhelst. 2018. Evolving Hardware Instinctive Behaviors in Resource-scarce Agent Swarms Exploring Hard-to-reach Environments. In *Proceedings of Genetic and Evolutionary Computation Conference Companion, Kyoto, Japan, July 15–19, 2018 (GECCO '18 Companion),* 8 pages.
DOI: 10.1145/3205651.3208255

1 INTRODUCTION

Technology advancements allow for an always increased miniaturization of electronics devices. The level of integration that these systems can achieve is always pushed forward, which allows for new application opportunities, from large consumer markets to specific industrial needs. For instance, one can think about the internet-of-things, with plenty of small devices, composed of sensors, processing and/or communication capabilities, interconnected to each other, that can now be used in various applications in our everyday life. Furthermore, it opens a new path for other types of applications, as the sensory exploration of hard-to-reach environments where only small-size nodes can enter. Those can be water distribution systems [19], the human body [12], or also more generally unknown environments [2]. However, this level of integration shifts the system constraints from pure performance to low power consumption. From the hardware perspective, designers have to make the best use of various trade-offs to obtain a maximum performance while dealing with extremely limited resources and power, and pushing these trade-offs is challenging.

Specifically, optimal circuit performances depend on a large variety of parameters as system specifications, current operating conditions, etc. System specifications may be defined by the application *off-line*, fixing general performance boundaries. In addition, operating conditions are by nature changing *on-line*, and acting optimally requires the circuit to modify its settings directly in the field. At design time, it usually results in worst-case design, suboptimal in many cases on-line [3]. The circuit over-performs most of the time, wasting power, pushing for the use of adaptation techniques.

But implementing energy-efficient adaptation techniques requires knowing the relationships between system specifications, performances, power consumption, and the associated circuit design variables (currents, voltages, etc.), which are usually very complex. This can be done for instance by relying on machine-learning based algorithms adapting directly the circuit regarding its performances [4, 18] or driven by more high-level specifications regarding the overall application, as in [17]. However, the current state-of-the-art focuses on relatively large-scale systems, such as transceiver blocks for Radio Frequency (RF) circuits [4, 5, 18], or full processors [15].

The situation shifts when adaptation algorithms have to be applied to extremely resource-scarce small-sized systems, for which saving power is very critical. For instance, one can consider a mesh network of cm-size battery-powered wireless sensing nodes moving in a given environment. One the one hand, the optimal settings of the circuit strongly depend on the environment to be explored by the nodes. If the nodes are not configured properly, they might not fulfill their task correctly, by a lack of either performance or available power. On the other hand, the adaptation mechanisms embedded in the circuit need to consume the least amount of resources and power to be effective. Thus, smart solutions are required to develop resource-efficient adaptation mechanisms, at circuit level together with automatically configuring the nodes depending on the environment to be explored.

In this work, we therefore propose a new adaptation framework targeting nodes operating in extremely resource-scarce conditions using Evolutionary Algorithms (EAs) to evolve on-line hardware adaptation mechanism, denoted as *instincts*. In biological systems, an instinctive behavior is a behavior that is evolved on hereditary bases rather than prior self-experience. Similarly, an instinct for the circuit can be seen as an on-line reaction to changes in its surrounding environment, learned off-line, which does not require intensive computational resources and power to be executed. In the context of resource-scarce systems, this is an imperative constraint. Thus, the proposed methodology, based on black-box optimization and inspired from biological systems is composed of three key features:

- **Off-line optimization through EA:** a black box optimizer attempts to find the optimum solution through manipulating the solution's genes via reproduction operators, i.e. tuning the circuit without any access to the direct relationships between parameters, performances, etc. to be optimized. This optimizer learns both static hardware parameters and the configuration of the dynamic instinct.
- **On-line hardware instinct:** the off-line learned hardware instinct is implemented in hardware to adapt the circuit using as less resources as possible.
- **Optimization with environment in the loop:** the EA framework allows to evolve the circuit's instincts directly including the environment in the loop. This enables to obtain always the best performance and power for the circuit, depending on the current environment.

The rest of this paper is organized as follows: Section 2 will give an overview of the related state-of-the-art. The proposed methodology will be detailed in section 3 and its application to an ultrasound communication circuit will be presented in section 4.

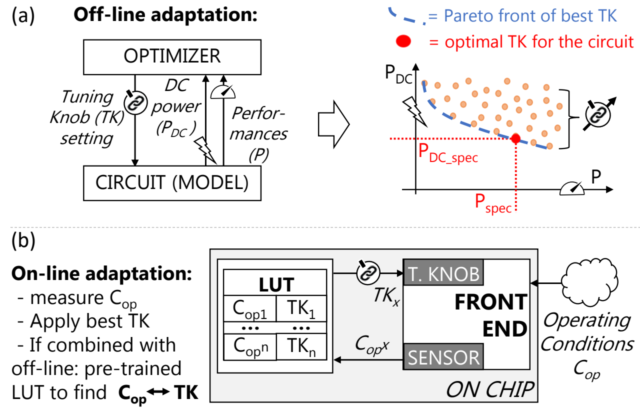


Figure 1: (a) example of off-line adaptation methodology (b) example of on-line adaptation methodology (possibly combined with off-line)

Finally, section 5 will compare the benefits of this new adaptation framework relative to convention methods.

2 CURRENT ADAPTATION PRINCIPLES

Off-line circuit adaptation: classical off-line adaptation is illustrated in Fig. 1(a).

The key elements are: 1.) a measurement of the circuit's performances 2.) tuning knobs, used to tune circuit's performances, and 3.) an optimizer, which finds the Pareto front of the trade-off between *objectives*, such as performance or power consumption. Performance measurements can be performed by directly measuring the circuit, or by using a representative model. These performances are for instance the communication quality of the wireless nodes, or the sensing accuracy of embedded sensors. Tuning knobs usually consists of available variables in the circuit, as bias currents or power supply voltages for instance. The optimizer explores the tuning space and evaluates the objectives to converge to the best trade-off. Among this Pareto front, a point is chosen meeting the current specifications. The corresponding tuning settings are applied once to the circuit and remain fixed at run-time. Several types of optimizers and algorithms can be used, such as gradient-decent search, or genetic algorithms [11, 13]. As it is performed off-line, this adaptation can be run with a massive amount of computing power. However, optimal tuning settings are chosen only static and set once off-line.

On-line circuit adaptation: adaptation can also be performed directly on-line, as seen in Fig. 1(b). This happens without external control, and it is hence the circuit itself which self-adapts according to the operating conditions C_{op}, which also have to be measured on-line by the circuit itself. The circuit can for instance measure signal statistics, as signal-to-noise ratios (SNR) or sensed signal streams, which gives an estimate of the current operating conditions. On-line, the circuit can then run on different modes (e.g. n modes), by dividing the set of operating conditions in clusters and find optimal tuning settings TK_x, where $x = 1, \ldots n$ for each cluster of conditions $C_{op}x$, where $x = 1, \ldots n$ [4, 18]. As an example, the circuit could assign several supply voltages and bias currents depending on the measured SNR, or digitize sensory data using more or less accuracy

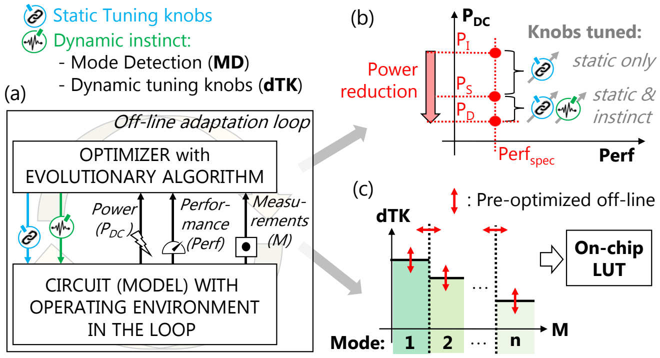


Figure 2: Principle of the proposed hardware adaptation

regarding measured signal statistics. A Look-up table (LUT) can be used to store the list of TK and C_{op}. On-line adaptation offers more effectiveness, but the algorithm needs to be embedded on-chip which requires many resources available.

Combining off-line and on-line adaptation: finding the optimal tuning strategy requires to a priori establish the relationships between performances, on-chip measurements and tuning settings. Those are usually complex, thus it has been proposed to rely on pre-trained machine-learning algorithms, which have been successfully applied to adapt circuits for different environmental circumstances, workloads, process variations, or a combination of them [1, 4, 14, 18]. In this case, the machine-learning algorithm learns a control law off-line, to map the optimal tuning knob settings to on-chip measurements. This law is then used in the off-line optimizer, or stored in an on-chip LUT and used at run time in a loop, as seen in Fig. 1(b). In essence, the clusters defines previously can be learned automatically, by building a law between SNR and TK, or between sensor statistics and TK.

Limitations of the previous work: implementing on-chip adaptation algorithms comes with overhead and power costs, that are not crucial for large-scale systems. For instance, a LUT can use megabytes of memory, and the adaptation algorithm may run on a baseband processor. If those resources are already available on-chip, the adaptation cost is minimized. However, for resource constrained systems, they need to be added, thus smart solutions with less than kilobytes of memory and μW level of power are required. This limits for instance the use of the LUT to only a few values and require the algorithm to run on a very low-power system, limiting the effectiveness of the adaptation as well. It is then crucial to make the best use of the few resources available on-chip, by for instance optimizing each cluster of a small on-chip LUT. In that way, the adaptation effectiveness is enhanced, without consuming more resources. This is the focus of this work.

3 PROPOSED METHODOLOGY

3.1 Instinct-based adaptation framework

The proposed instinct-based adaptation framework, which is based on the work of [10], combines off-line and on-line adaptation mechanisms in a novel way, as illustrated in Fig. 2. Specifically, it facilitates the ability of the optimizer, realized with an Evolutionary Algorithm (EA), to off-line learn optimal yet efficient on-line tuning rules, to achieve an optimal adaptation efficiency mimicking biological instincts.

Learning optimal tuning rules: in biology, the learning process of an instinct is done on the genetic level. This is usually manifested in the form of a primitive fast reactions to a condition experienced in the field, i.e. an biological element reacting without wasting valuable time or energy. Using this analogy, learning the instinct of a circuit refers to a strategy to optimally detect on-line circuit *modes* with the lowest amount of resources and power available. A mode targets to group possible operating conditions in which the circuit will operate in clusters. This approach is analogous to LUT techniques described in section 2 but each cluster will be pre-optimized off-line before being used in the field. This technique will be referred as Mode Detection MD. For each mode, a value for a dynamic tuning knob setting dTK will be assigned. To add flexibility and optimize the adaptation, the number, the characteristics of each mode, and the corresponding optimal dTK values are also part of the adaptation process off-line, as depicted in Fig. 2(c). In addition, the circuit can be be equipped with static tuning knob settings sTK that are the same for each cluster, also learned during the adaptation process. The combination of static tuning knobs and dynamic instinct increases the solution space and enable the optimizer to find different trade-offs for the adaptation. For example as depicted in Fig. 2(b), the power consumption can be reduced for the same performance specification.

Overall framework: The general framework is illustrated in Fig. 3. It is composed of two main phases: an off-line adaptation loop and an on-line validation in the real environment. First, the circuit is optimized off-line, regarding the procedure described in section 3.2. This optimization is performed using a simulation environment in the loop, representing the real environment to be explored. In this way the methodology can find settings that fit best for the particular environment to be explored, achieving extra power savings. Once the adaptation objectives are reached, the agents explore the real environment in an on-line validation phase, using the instincts learned in the off-line phase to adapt themselves. If needed, the overall adaptation can be refined by launching another off-line phase and start the overall procedure.

3.2 Off-line optimization based on EA

General methodology: During the off-line phase, the optimization is conducted via a EA-based optimizer, mimicking the biological instinct scheme. This process is known as black box optimization, since it only has access to the performance of the proposed solution relative to the population, which then decides if that solution shall survive to share its genes or not. As a prerequisite for the optimization, a fitness function, which serves as a basis for the EA, needs to be formulated. One can consider two objectives to be optimized: 1.) maximizing a given circuit performance $F_{\text{perf}}(C_{op})$, given by system specifications, and subjected to the operating conditions (C_{op}) while concurrently 2.) minimizing the circuit's power consumption $P_{DC}(TK)$, which is function of the circuit tuning knobs

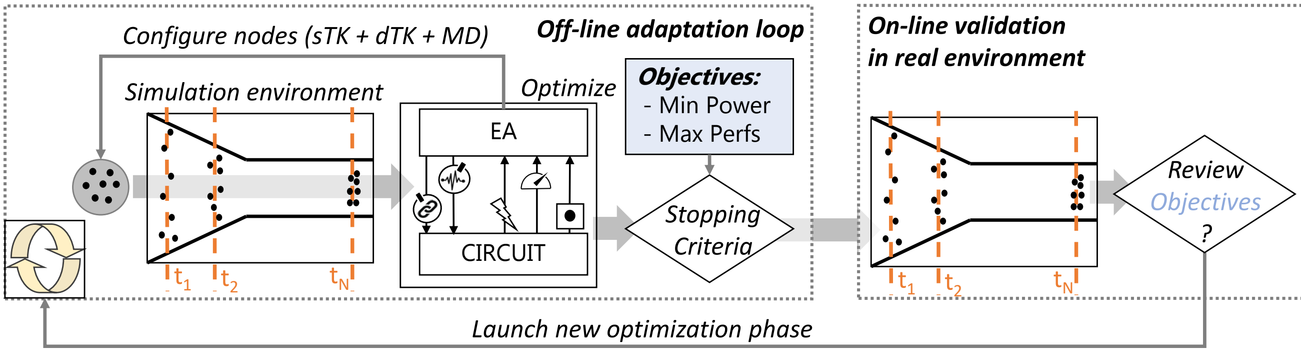


Figure 3: Illustration of the overall framework

$TK = \{sTK, dTK, MD\}$. The optimization problem can be formulated as:

$$\max_{\mathbf{TK}} F_{\text{perf}}(C_{op}) \quad \text{and} \tag{1a}$$

$$\min_{\mathbf{TK}} P_{DC}(TK) \tag{1b}$$

$$\text{s. t. } TK_{i,\min} \leq TK_i \leq TK_{i,\max} \; \forall i = 1, \ldots, TK \tag{1c}$$

As given by equations 1a and 1b, performance $F_{perf}(C_{op})$ is function of C_{op} and controlled by TK, and function P_{DC} is directly determined by TK. The EA will learn the optimal values of TK regarding the objectives, considering all possible operating conditions C_{op}. However, unlike traditional techniques, in instincts the modes of operations and the dynamic parameters are also part of the optimization process.

Integration of on-line parameters: the EA must completely optimize the instincts by configuring sTK, dTK and MD, as explained in section 3.1. Optimizing the instincts means (a) defining how to switch between different modes, by defining optimal thresholds in the operating conditions C_{op} and (b) defining an optimal dTK for each of these modes. Equations 1a and 1b become:

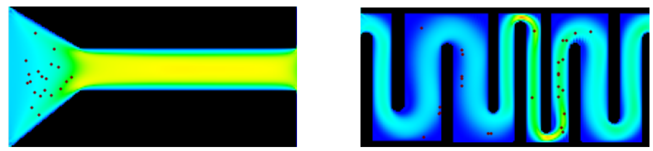
$$\max_{\mathbf{sTK,dTK,MD}} F_{\text{perf}}(C_{op}) \quad \text{and} \tag{2a}$$

$$\min_{\mathbf{sTK,dTK,MD}} P_{DC}(sTK, dTK, MD) \tag{2b}$$

Using these new parameters offers more tuning possibilities to the EA. This enables to shift optimization constraints from an on-line resource constrained environment, to an off-line phase where computing power is easily available.

3.3 Hardware requirements & implementation

As computational power is shifted in the off-line phase, running the instinct on-line will consist on implementing relatively simple hardware circuits. Those circuits are able to detect the change of modes (e.g. thresholds in measurement values) and assign the appropriate tuning setting corresponding to the current mode, as depicted in Fig. 1(b). As these modes are pre-optimized off-line, the adaptation effectiveness is optimized considering on-chip resources. However, this integration is submitted to additional constraints. In order to extract C_{op} directly on-line, the circuit will rely on an on-chip measurement M that can be selected to reflect accurately C_{op}. This selection can be made driven by design constraints, or analogously to alternate test procedures, that correlates circuit



(a) Environment 1 **(b) Environment 2**

Figure 4: (a) (b) The two $20\text{m} \times 10\text{m}$ **environments used in the case-study**

performances to low-cost measurements [1, 3, 21]. In addition, it has to be ensured that the use of M to obtain C_{op} in the optimization procedure still enables the EA to find the optimal tuning knob settings off-line. This can for instance be achieved by obtaining a complete independence between M and dTK [1].

3.4 Evolutionary multi-objective algorithms

Evolutionary multi-objective algorithms (EMOAs) are a family within the EA family of algorithms that attempt to do black-box optimization in multi-objective problems. This is more challenging than the single objective case, as the quality of a single solution relative to a population of solutions can be determined in a more straightforward manner relative to the multi-objective case. In the later case, a lot of factors such as non-dominance and diversity must be quantitatively considered. In our framework, the proposed EMOA belongs to the indicator based class of EMOA. The reason behind this choice is the fact that the number of tunned parameters are relatively large (ten parameters), given that each evaluation is computationally expensive, in addition, it is planned to expand the number of objectives to include even more than the introduced two, thus other EMOA classes, such as non-dominated sorting (e.g NSAG-2[8]), is not a good fit. Within the indicator based EMOA class, R2 indicator is picked [6] as it is less computationally demanding, relative to hyper-volume (HV) indicator based algorithms [22] for example. This is critical as each CFD simulation is computationally costly and we can not afford highly demanding indicator calculations.

4 CASE STUDY AND SCENARIO

This case study uses very small wireless nodes able to communicate distance ranging packets among each other to explore two fluid based environments, for instance in the context of unknown environment exploration [2]. These nodes communicate through

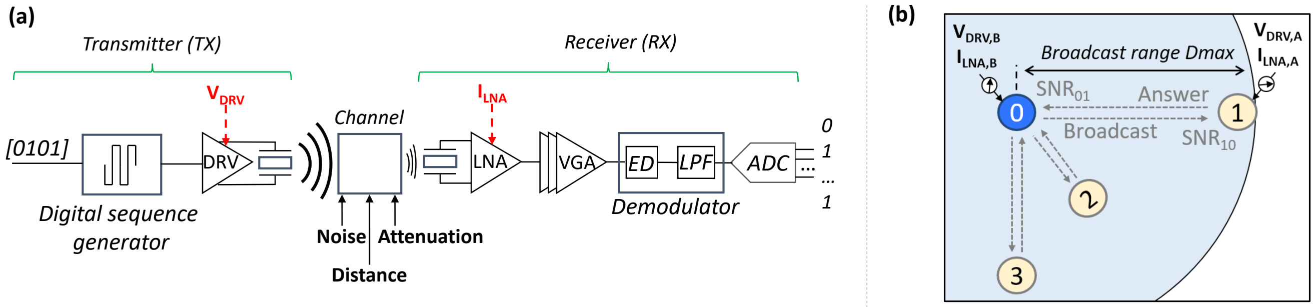


Figure 5: (a) Model overview of the ultrasound communicating, (b) Illustration of the representative simulation scenario

ultrasound (US), which is their major source of power consumption, limiting their exploration lifetime. Finding optimal tuning settings is not straightforward, as they depend on the environment, node's relative distances, conditions, etc.

Ultrasound communication model: a behavioral model of the communication has been implemented, as seen in Fig. 5(a), with a communication scheme analogous to On-Off Keying (OOK). The transmitter consists of a digital sequence generator, which converts the bit stream into a series of pulses, followed by a driver (DRV) that excites the US transducer, modeled with an electrical equivalent model [16]. The channel model considers a fluid medium, where noise and attenuation are function of the attenuation coefficient and spreading losses [20]. The receiver consists of a Low-Noise Amplifier (LNA), a Variable Gain Amplifier (VGA), a demodulator composed of a envelop detector (ED), a low pass filter (LPF), and an Analog-to digital converter (ADC). Channel model has been validated by performing path loss measurements with existing US transducers. The transceiver is equipped with two tuning knobs: 1.) the supply voltage V_{DRV}, that trades off the node's power consumption versus the transmit power, and 2.) the bias current I_{LNA}, that trades off power consumption versus receiver noise. The communication follows the protocol described in [9], based on broadcast and answers, as seen in Fig. 5(b). A node broadcasts a message and gathers all answers from its neighboring nodes. This information is subsequently used to calculate their respective distances through time-of-flight. This process repeats with other broadcasts until the time-of-flights of whole node swarm are computed. It is assumed that all nodes perform omni-directional communication and all receivers are constantly active. The transmitter consumes power only when transmitting.

Environments: two environments will be explored, as depicted in Fig. 4. The first one consists of a tunnel, with a decreasing diameter. The second one is composed of several sections and bends of different diameters. Each environment is simulated using a Computational Fluid Dynamics (CFD) simulator based on Lattice-Boltzmann algorithm, with nodes traveling inside.

Off-line optimization set up: The simulation scenario is illustrated in Fig. 7. A group of 20 nodes is chosen and their relative distances numbis computed for N different frames spanning the environment. This enables the optimization to be run on a representative inter-node distance vector. The EA algorithm then concurrently optimizes the static settings and the dynamic instinct. Referring also to Fig. 2 the EA objectives are 1). To maximize the system performance $Perf$, calculated as the minimal SNR over all communications between the set of nodes, and 2). To minimize the power consumption P_{DC}, averaged per node. The following results will be presented regarding a communication specification of $SNR \geq 21dB$. This specifications enable to keep the bit-error rate of the communication below 10^{-3} which is sufficient in this case [9].

Table 1: R2-indicator EMOA Settings

Environment	Population	Number of Evaluations
Environment 1	20	100
Environment 2	30	100

List of tunable parameters: following the example depicted in Fig. 5(b), tuning settings will be separated for broadcast and answer. The broadcast must reach nodes in a predefined radius D_{max}. The tuning parameters are set to $V_{DRV} = V_{DRV,B}$ and $I_{LNA} = I_{LNA,B}$, B referring to Broadcast. These parameters are static, as the broadcast node will not have additional run-time information of the other nodes locations before the ranging action. When the node receives a message, it changes its settings to $V_{DRV,A}$ and $I_{LNA,A}$, A referring to Answering. Yet, opportunities exist for on-line adaptation. As the node received a packet from the broadcast, it has information on the signal strength to answer with lower power. $V_{DRV,A}$ will then be a dynamic tuning knob. It is also assumed that receiver settings are the same for each node, i.e. $I_{LNA,A}$ is static.

Instinct implementation and tuning: referring to section 3, Mode Detection MD will be performed by measuring the received SNR, denoted SNR_{rec}. MD consequently consists of a set of SNR_{rec} threshold values. The choice of SNR_{rec} is motivated by the constraints listed in section 3.2: 1.) SNR_{rec} directly reflects the communication quality, and can be measured at low-cost on-chip (for example using a power detector). 2.) the answer to the broadcast $V_{DRV,A}$ is independent SNR_{rec} received by the node. In our implementation of the R2-indicator EMOA, simulated binary crossover (SBX)[7] was adopted and its rate was set as 0.85. On the other hand, polynomial based mutation was used with a mutation rate of 1/n, where n is the population number. Furthermore, The population size and the maximum number of evaluations adopted in our experiments are shown in Table 1.

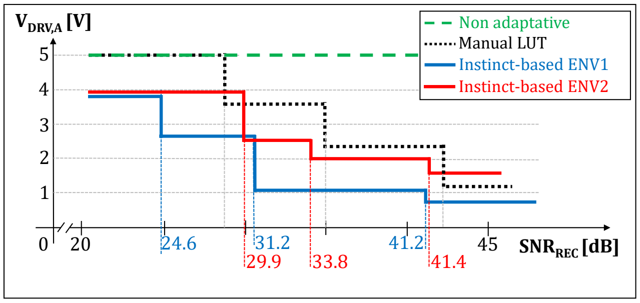


Figure 6: Comparison of the split between modes for the two case studies

5 RESULTS

5.1 Baselines for comparison

Design without adaptation: as a baseline, a design approach is first considered, without adaptation (selected knobs are: $V_{DRV,A}$,$V_{DRV,B}$,$I_{LNA,A}$,$I_{LNA,B}$). The transmitter and receiver settings will be identical, i.e. $V_{DRV,A} = V_{DRV,B}$ and $I_{LNA,A} = I_{LNA,B}$. By performing a link budget analysis considering a water medium, the path loss at Dmax=3.5m is estimated to be 90dB. The transmitter voltage is fixed to $V_{DRV,A} = V_{DRV,B} = 5V$ to reach nodes at this distance. The received power at the receiver is then equal to -95dBm and the noise power must be lower than -106dBm to achieve the required SNR. This is equivalent to a noise voltage $V_{RMS,in} = 5\mu V$ introduced by the LNA. By considering a Noise Efficiency Factor (NEF) of 1.5 for the LNA, we can estimate a bias current of around $I_{LNA} = 45\mu A$, taking a small design margin. The chosen tuning knobs are then $V_{DRV,A} = V_{DRV,B} = 5V$ and $I_{LNA,A} = I_{LNA,B} = 45\mu A$. The nodes are configured with those tuning settings and send out in each environment. Chosen settings, performances and power consumption for the considered group of nodes can be observed in Table 3.

On-line adaptation with manual LUT: with a small amount of on-chip resources, a standard practice would be to manually find the LUT settings. For instance, the designer may choose to reduce $V_{DRV,A}$ in favorable conditions, relying on SNR measurements. A possible approach consists in spanning the communication distance to extract the range of received SNR for worst conditions, and divide this range in clusters. In this case, the range is $21dB < SNR_{REC} < 49dB$. Dividing this space in equal clusters, and assigning a cluster to each mode leads to: $21dB < mode1 < 28dB$, $28dB < mode2 < 35dB$, $35dB < mode3 < 42dB$ and $42dB < mode4 < 49dB$. Then, simulations can be carried out to find the corresponding voltage for each mode. In this case, voltages found for modes 1 to 4 to ensure $SNR_{REC} \geq 21dB$ are respectively $V1 = 5V$,$V2 = 3.6V$,$V3 = 2.3V$ and $V1 = 1.2V$ as seen in Fig. 6. This approach is used to compare our proposed methodology to an on-line adaptation with the same amount of hardware resources.

5.2 Instinct-based adaptation

In the instinct-based adaptation, the EA will simultaneously tune: 1.) static tuning knobs $V_{DRV,B}$, $I_{LNA,B}$ and $I_{LNA,A}$ and 2.) the on-line instinct, that consists in choosing the best mode detection, by setting SNR_{rec} thresholds (i.e. T_{12}, T_{23} and T_{34}), and the dynamic tuning knob value $V_{DRV,A}$ assigned per mode, (i.e. V_1 to V_4) considering 4 modes. It represents 10 different parameters.

Table 2: Hypervolume indicator (HV) mean and variance for 30 runs

Environment	HV mean [%]	HV variance [%]
Environment 1	83.97	1.79
Environment 2	78.47	2.44

Analysis of the EA performance: EMOA has a stochastic nature, thus it requires statistical analysis over n-trials to test its produced Pareto-front quality for both consistency of the convergence and diversity. One widely used performance metric is hyper-volume (HV) indicator [22]. In this indicator, the HV of the solutions of the Pareto front proposed by the EMOA is calculated relative to a proposed reference point. In this work, 30 simulations have been conducted, each with different random seeds. Table 2 summarizes the hyper-volume's mean and variance relative to point (5×10^{-4},15). The table shows that the variance is relatively small (1.79% and 2.44%), this highlights the consistency of the algorithm performance. In addition, environment 2 has less HV % than environment 1, due to its dynamic nature and complexity.

General performance of instinct-based adaptation: after training, a list of optimal settings can be obtained and compared to the non-adaptive design and the manual LUT approaches, as seen in Table 3. A comparative of the chosen splits between modes is also depicted in Fig. 6. Several observations can be made: *The proposed methodology is adaptive to the current environment:* as it can be seen, the split between modes depends on the environment. For instance, in environment 1, the group nodes stays relatively close to each other. Since the probability of nodes to be very far from each other is low, a finer zone split is chosen by the EA for lower distance, where the amount of nodes is maximal.

The off-line configuration of the dynamic instinct enables to find better trade-offs: this is illustrated by the balance between transmitter and receiver settings in answering mode (the current in the receiver is increased compared to other approaches). Indeed, the overall power consumption is dominated by the amount of answers transmitted (several nodes answer to one broadcast), and reducing the maximum voltage is then more power efficient than increasing the receiver currents. This is possible since the EA explores the whole tuning space and finds the right balance between static and dynamic settings regarding the operating environment.

The use of resources is maximized: comparing to the manual LUT approach, that uses the same amount of resources, power consumption is reduced ensuring just the required SNR performance. In environment 1, other approaches over perform in terms of SNR, because the effective maximum distance between nodes is reduced in the environment. The EA capture this and adapt the tuning settings accordingly to bring the SNR at the required level and consume less.

Comparison of power consumption for different nodes: 3 nodes were randomly selected to assess the impact of the adaptation strategy. Results are seen in Fig. 8, where it can be observed that for

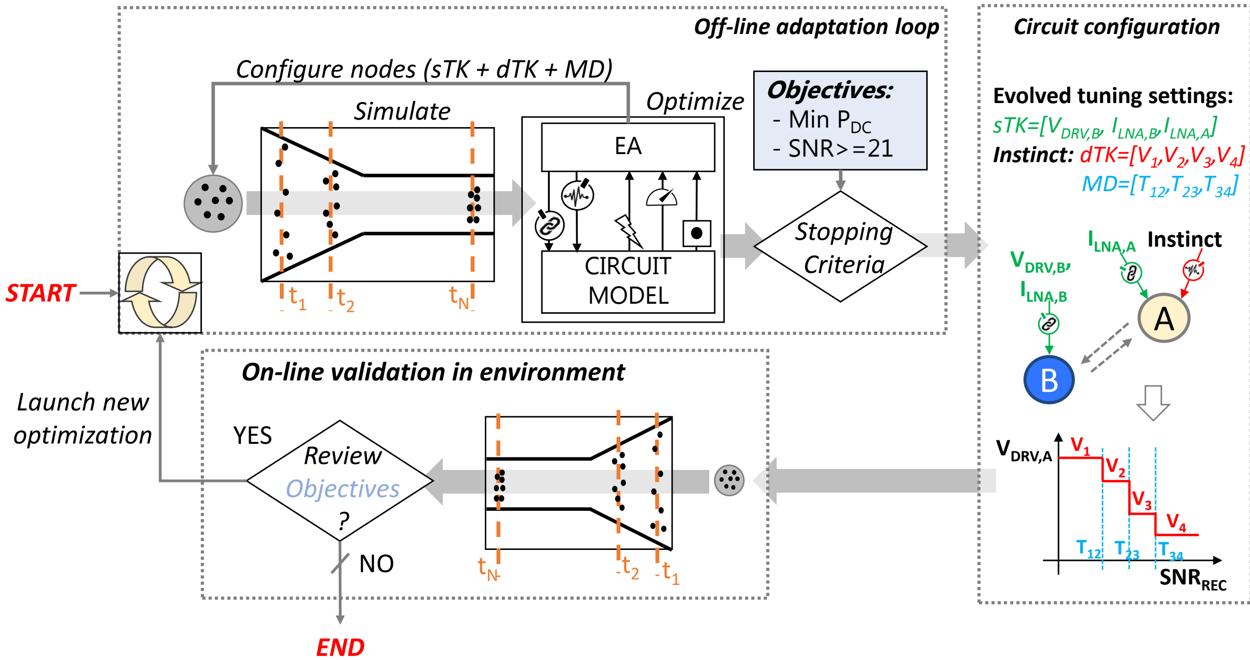


Figure 7: Illustration of the adaptation procedure for the proposed case study

Table 3: Example of obtained tuning knob settings for the two case studies

Tuning Knob settings	$V_{DRV,B}$ [V]	$I_{LNA,B}$ [μA]	$I_{LNA,A}$ [μA]	V_1 [V]	T_{12} [dB]	V_2 [V]	T_{23} [dB]	V_3 [V]	T_{34} [dB]	V_4 [V]	P_{DC} [μW]	SNR_{min} [dB]
Non-adaptive ENV1	5	45	45	5	-	5	-	5	-	5	393	22.5
Manual LUT ENV1	5	45	45	5	23.7	3.6	30.7	2.3	37.7	1.2	246	22.5
Instinct-based ENV1	4.7	40.3	69.4	3.8	24.6	2.7	31.2	1.1	41.2	0.7	141	21
Non-adaptive ENV2	5	45	45	5	-	5	-	5	-	5	420	21
Manual LUT ENV2	5	45	45	5	28	3.6	35	2.3	42	1.2	282	21
Instinct-based ENV2	5	44.9	85.3	3.9	29.9	2.6	33.8	2	41.4	1.5	234	21

every node the instinct-based adaptation saves significant power. For environment 1, the non-adaptive and manual LUT approaches over perform, i.e. SNR is above 21dB and the circuit consumes more power. Applying the instinct-based methodology enable to find the required performance level and reduce the power consumption up to 70% compared to the non-adaptive design approach, and 40% regarding manual LUT. For environment 2, all methodologies lead to the same SNR performance, which enable direct power saving comparison. As it can be seen, power is reduced by 45% compared to non-adaptive design and by 20% compared to the manual LUT.

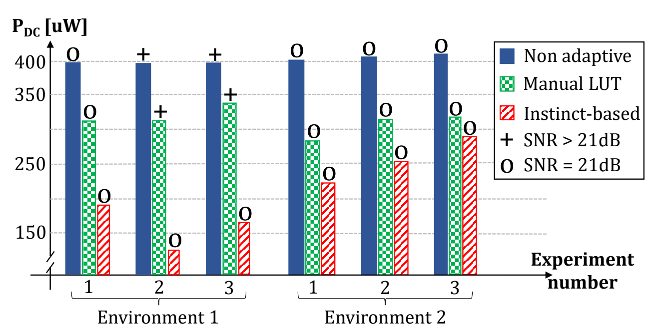


Figure 8: Power consumption comparisons for the 2 environments

6 CONCLUSION

This work introduces the instinct-based adaptation framework based on an evolutionary algorithm, with instincts being adaptive mechanisms efficiently executed on-chip. On-line instincts are integrated in the off-line optimizer which enables to find different trade-offs between performances and achieve significant power savings. Results show an improvement up to 45% compared to conventional techniques used to achieve on-line adaptive behavior.

REFERENCES

[1] M. Andraud, H.-G. Stratigopoulos, and E. Simeu. 2016. Post Manufacturing One-Shot Calibration of RF Circuits Based on Non-Intrusive Sensors. *IEEE Transactions on Circuits and Systems-I: Regular Papers* (2016).

[2] M. Andraud et al. 2017. Exploring the unknown through successive generations of low power and low resources versatile agents. In *Proc. of Design, Automation & Test in Europe.* 1–4.

[3] D. Banerjee, S. Sen, and A. Chatterjee. 2015. Self Learning Analog/Mixed-Signal/RF Systems: Dynamic Adaptation to Workload and Environmental Uncertainties. In *Proc. of IEEE/ACM Design, International Conference on Computer-Aided Design.* 59–64.

[4] D. Banerjee et al. 2015. Real-time use-aware adaptive RF transceiver systems for energy efficiency under BER constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2015).

[5] D. Banerjee et al. 2017. Self-Learning RF Receiver Systems: Process Aware Real-Time Adaptation to Channel Conditions for Low Power Operation. *IEEE Transactions on Circuits and Systems-I: Regular Papers* 64, 1 (2017), 195–208.

[6] Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. 2015. 2 indicator-based multiobjective search. *Evolutionary Computation* 23, 3 (2015), 369–395.

[7] Kalyanmoy Deb and Ram B Agrawal. 1994. Simulated binary crossover for continuous search space. *Complex Systems* 9, 3 (1994), 1–15.

[8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[9] E.H.A Duisterwinkel et al. 2016. Mapping Swarms of Resource-Limited Sensor Motes: Solely Using Distance Measurements and Non-Unique Identifiers. In *IEEE symposium Series on Computational Intelligence.*

[10] Ahmed Hallawa, Jaro De Roose, Martin Andraud, Marian Verhelst, and Gerd Ascheid. 2017. Instinct-driven dynamic hardware reconfiguration: evolutionary algorithm optimized compression for autonomous sensory agents. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion.* ACM, 1727–1734.

[11] Tetsuya Higuchi, Masaya Iwata, Didier Keymeulen, Hidenori Sakanashi, Masahiro Murakawa, Isamu Kajitani, Eiichi Takahashi, Kenji Toda, N Salami, Nobuki Kajihara, and others. 1999. Real-world applications of analog and digital evolvable hardware. *IEEE transactions on evolutionary computation* 3, 3 (1999), 220–235.

[12] I.S.M. Khalil et al. 2014. MagnetoSperm: A microrobot that navigates using weak magnetic fields. *Applied Physics Letters* 104, 22 (2014).

[13] Jörg Langeheine, Joachim Becker, Simon Folling, Karlheinz Meier, and Johannes Schemmel. 2001. A CMOS FPTA chip for intrinsic hardware evolution of analog electronic circuits. In *Evolvable Hardware, 2001. Proceedings. The Third NASA/DoD Workshop on.* IEEE, 172–175.

[14] C. Maxey, G. Creech, S. Raman, J. Rockway, K. Groves, T. Quach, L. Orlando, and A. Mattamana. 2012. Mixed-Signal SoCs with In Situ Self-Healing Circuitry. *IEEE Design & Test of Computers* 29, 6 (2012), 27–39.

[15] Bert Moons and Marian Verhelst. 2017. An Energy-Efficient Precision-Scalable ConvNet Processor in 40-nm CMOS. *IEEE Journal of solid-state Circuits* 52, 4 (2017), 903–914.

[16] R.J. Przybyla et al. 2015. 3D Ultrasonic Rangefinder On a Chip. *IEEE Journal of Solid-State Circuits* 50, 1 (2015).

[17] Stephan Schlupkothen, Ahmed Hallawa, and Gerd H. Ascheid. 2018. Evolutionary Algorithm Optimized Centralized Offline Localization and Mapping. In *2018 International Conference on Computing, Networking and Communications (ICNC): Wireless Ad hoc and Sensor Networks (ICNC'18 WAHS).* Maui, USA.

[18] S. Sen, V. Natarajan, Devarakond S., and A. Chatterjee. 2014. Process-Variation Tolerant Channel-Adaptive Virtually Zero-Margin Low-Power Wireless Receiver Systems. *IEEE Transactions of Computer-Aided Design of Circuits and Systems* 33, 12 (2014), 1764–1777.

[19] I. Stoianov, L. Nachman, and S. Madden. 2007. PIPENET: A Wireless Sensor Network for Pipeline Monitoring. In *Proc. of International Conference on Information Processing and Sensor Networks.* 264–273.

[20] M. Stojanovic. 2008. Underwater Acoustic Communications: Design Considerations on the Physical Layer. In *Proc. of Conference on Wireless on Demand Network Systems and Services.* 1–10.

[21] P. N. Variyam, S. Cherubal, and A. Chatterjee. 2002. Prediction of Analog Performance Parameters Using Fast Transient Testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 21, 3 (2002), 349–361.

[22] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. 2007. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization.* Springer, 862–876.